Digital Electronics:

Sarala .T
Asst. Prof.
E&CE Dept.
SVCE

Module 1:

Principles of combinational logic:

Combinational logic:

-> Boolean Expressions are implemented using logic gates.

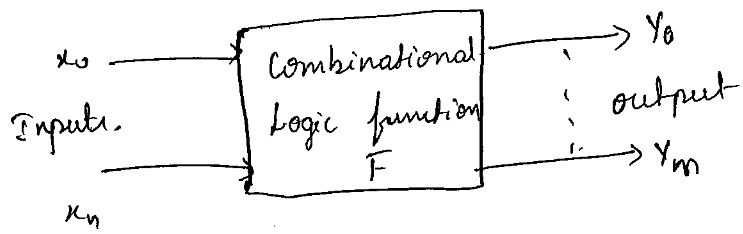
-> The AND, OR & NOT are reffered as basic gates.

-> complex functional structures such as arithmetic ckts, decoders, encoders, multiplexers & so on. are realized using basic gates (logic) functions.

Combinational logic:

Combinational logic deals with the techniques of combining the basic gates into circuits that perform some desired function.

Ex: Adders, Subtractors, Encoders, Decoders etc.



Let $X \in \{x_0, x_1, x_2 \dots\dots x_n\}$ be a set of i/ps

$Y \in \{y_0, y_1, y_2 \dots\dots y_m\}$ be a set of o/ps.

O/P is related to I/P as

$$\boxed{y = F(x)}$$

where, F = combinational function.

Basics:

-> Basic logic gates are fundamental building blocks from which all of the logic circuits & digital systems are constructed.

-> Logic gates produce one output level when some combinations of input levels are given.

-> The interconnection of gates to perform a variety of logic operations is called logic design.

Basic terms:

Truth table: A tabular shows the "truth" relationship between set of input & output variables. For all possible input variable combinations, the output is Examined. Truth is defined as 1 (+ve logic) or 0 (-ve logic) of binary variable.

Logical / Boolean Expression: Two logical variables are combined with logical operator to form logical Expression.

Boolean Laws & Theorems:

Useful for design the digital circuit are given below.

1) law of Intersection.

$A \cdot 1 = A$

I/P		O/P
A	1	
0	1	0
1	1	1

$\} \Rightarrow A$

$A \cdot 0 = 0$

I/P		O/P
A	0	
0	0	0
1	0	0

$\} \Rightarrow 0$

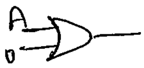
2) Law of union:

$A + 1 = 1$

I/P		O/P
A	1	
0	1	1
1	1	1

$\} \rightarrow 1$

$A + 0 = A$

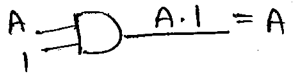


I/P		O/P
A	0	
0	0	0
1	0	1

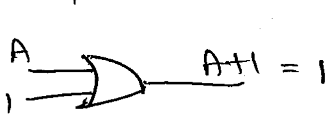
$\} \rightarrow A$

3) Law of Identity:

$A.1 = A$

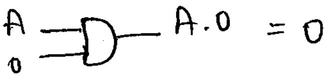


$A+1 = A$

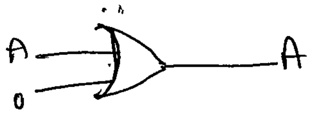


4) Law of Null:

$A.0 = 0$

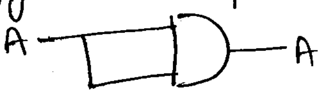


$A+0 = A$



5) Law of tautology or Indepotence:

$A \cdot A = A$



I/P		O/P
A	A	
0	0	0
1	1	1

$\} - A$

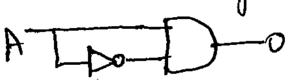
$A + A = A$

I/P		O/P
A	A	
0	0	0
1	1	1

$\}$ A

6) Law of complement or Negation:

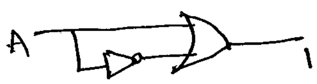
$A \cdot \overline{A} = 0$



I/P		O/P
A	$\overline{A}$	
0	1	0
1	0	0

$\} - 0$

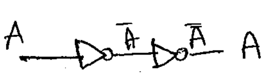
$A + \overline{A} = 1$



I/P		O/P
A	$\overline{A}$	
0	1	1
1	0	1

7) Law of Double Negation or Involution:

$\overline{\overline{A}} = A$



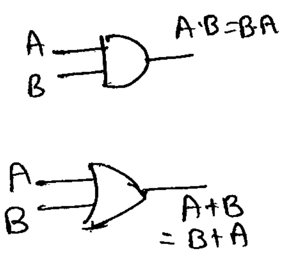
A	$\overline{A}$	$\overline{\overline{A}}$
0	1	0
1	0	1

$\} - A$

8) Law of commutation:

$$A + B = B + A$$

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

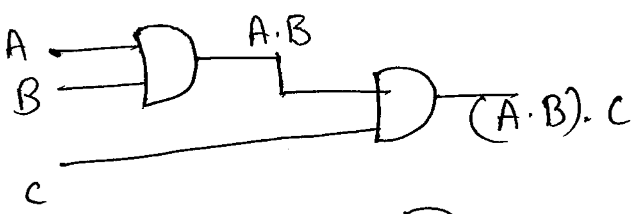


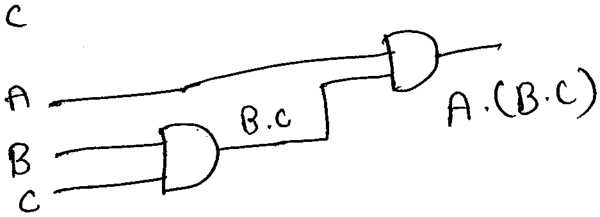
I/P		O/P		O/P	
A	B	A·B	B·A	A+B	B+A
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	0	1	1
1	1	1	1	1	1

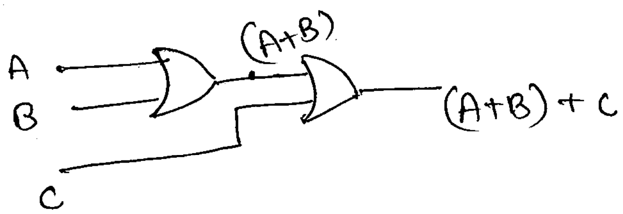
9) Law of Association:

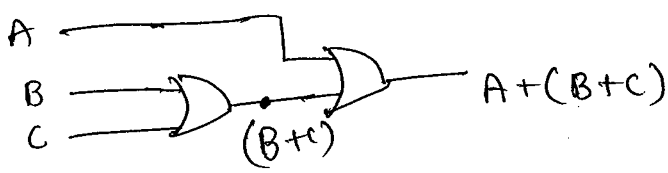
$$(A.B).C = A.(B.C)$$

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









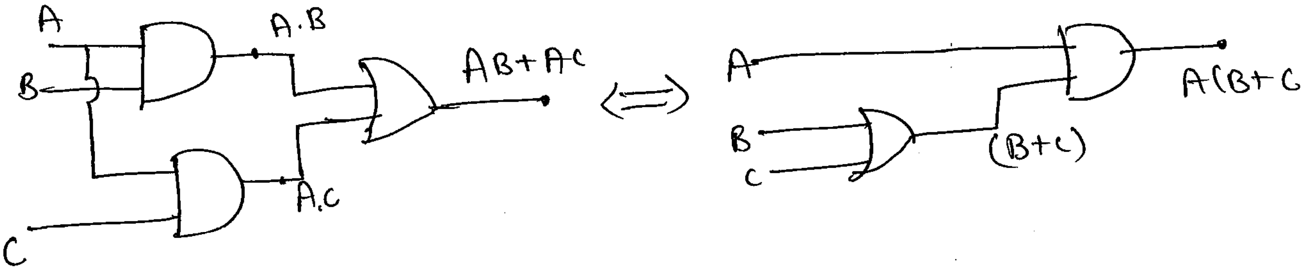
I/P			O/P			
A	B	C	(A.B).C	A.(B.C)	AB	BC
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	1	1
1	1	1	1	1	1	1

I/P			Intermediate o/p		O/P	
A	B	C	(A+B)	(B+C)	(A+B)+C	A+(B+C)
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

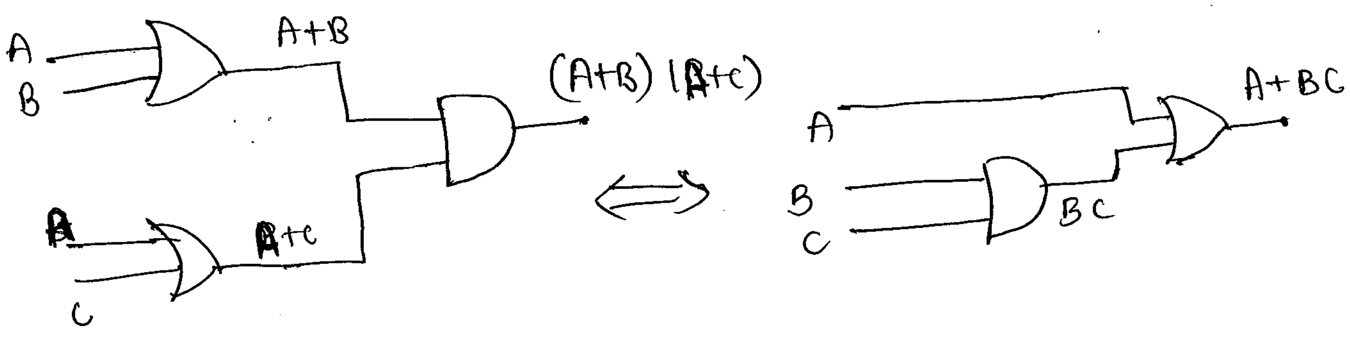
10) Law of Distribution:

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

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



I/p.			Intermediate o/p			o/p.	
A	B	C	AB	AC	B+C	AB+AC	A(B+C)
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	0	1	0	0
0	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1



Proove that $(A+B)(A+C) = A+BC$

$$\underbrace{(A+B)(A+C)}_{LHS} = \underbrace{A+BC}_{RHS}$$

$LHS = A \cdot A + A \cdot B + A \cdot C + B \cdot C$ ∵ multiplication, $A \cdot A = A$.

$A + A \cdot B + A \cdot C + B \cdot C$

$A(1+B) + A \cdot C + BC$ ∵ $1+A = 1$

$A + A \cdot C + BC$

$A(1+C) + BC$ ∵ $1+C = 1$

$\underline{A+BC} = RHS$

ii) Law of Absorption:

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

Proof: $A(A+B) = A \cdot A + A \cdot B$ ∵ $A \cdot A = A$

$= A + A \cdot B$ ∵ $1+A = 1$

$= A(1+B)$

$= \underline{A} = RHS$

ii) $A + AB = A$

Proof: $A \cdot 1 + AB = A(1+B)$ ∵ $1+A = 1$

$= \underline{A}$ $\underline{RHS}$

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

Proof: $AB + \overline{B} = (A + \overline{B})(B + \overline{B})$ $\because (A + BC) = (A+B)(A+C)$

$= \underline{(A + \overline{B})}$ $\because B + \overline{B} = 1$

$= RHS$

iv) $A\overline{B} + B = A + B$

Proof: $A\overline{B} + B = (A+B)(\overline{B}+B)$ $\because (A+BC) = (A+B)(A+C)$

$= \underline{A+B} = RHS$ $\because B+\overline{B} = 1$ & $A \cdot 1 = A$

12) Consensus Theorem:

1) $AB + \overline{A}C + BC = AB + \overline{A}C$

Proof: $AB + \overline{A}C + BC = AB + \overline{A}C + BC(\overline{A} + A)$

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

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

$= AB(1+C) + \overline{A}C(1+B)$

$= \underline{AB + \overline{A}C}$

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

Proof: $(A+B)(\overline{A}+C)(B+C) = (A\overline{A} + AC + B\overline{A} + BC)(B+C)$

$= ACB + \overline{A}BB + CBB + ACC + \overline{A}BC + BCC$

$\because AA = A$ $= ACB + \overline{A}B + \underline{CB} + AC + \overline{A}BC + \underline{BC}$

$\because BC + BC = BC$ $\Rightarrow ACB + \overline{A}B + AC + \overline{A}BC + BC$

$\because 1 + B = 1$ $= AC(1+B) + \overline{A}B + BC(\overline{A}+1)$

$LHS = \underline{AC + \bar{A}B + BC}$

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

$= \cancel{A\bar{A}} + AC + B\bar{A} + BC$

$RHS = \underline{AC + \bar{A}B + BC}$

$\underline{LHS = RHS}$

13) <u>Transposition</u> <u>Theorem</u>:

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

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

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

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

$= AB + \bar{A}C + ACB + \bar{A}CB$

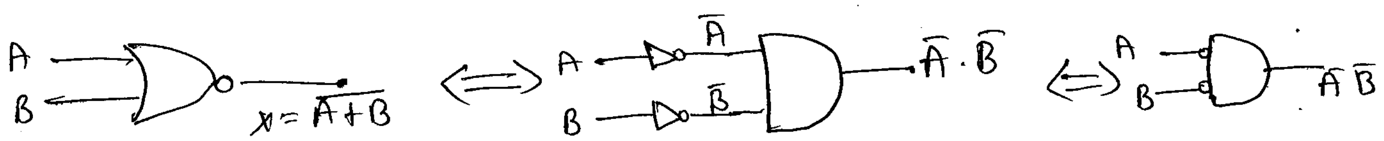
$= AB(1+C) + \bar{A}C(1+B)$

$= \underline{AB + \bar{A}C} = RHS$

✩ De Morgan's theorem-1

State: NOR logic is equivalent to AND logic with inverted inputs.

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

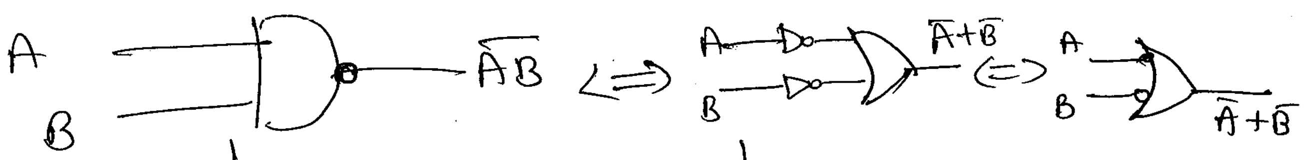


I/P		Intermediate O/P			O/P	
A	B	A+B	$\bar{A}$	$\bar{B}$	$\overline{A+B}$	$\bar{A}\ \bar{B}$
0	0	0	1	1	1	1
0	1	1	1	0	0	0
1	0	1	0	1	0	0
1	1	1	0	0	0	0

✩ De Morgan's Theorem-II

State: NAND logic is equivalent to OR logic with inverted inputs.

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



I/P		Intermediate O/P			O/P	
A	B	AB	$\bar{A}$	$\bar{B}$	$\overline{AB}$	$\bar{A}+\bar{B}$
0	0	0	1	1	1	1
0	1	0	1	0	1	1
1	0	0	0	1	1	1
1	1	1	0	0	0	0

Rule to De-Morganize the Expression:

1) Identify the different terms.
2) Find the dual of the expression → (Dual of OR operator is AND operator or vice versa).
3) Complement each term.
4) Complement the expression.
5) Repeat the process considering term(s) as expressions till all the terms are De-Morganized.

Examples:

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

Sln:

→ terms → A, B, C

→ find dual of expression → $\overline{A\cdot B\cdot C}$

→ complement each term → $\overline{\bar{A}\cdot\bar{B}\cdot\bar{C}}$

→ Complement expression → $\overline{\overline{\bar{A}\cdot\bar{B}\cdot\bar{C}}}$

$$\overline{A+B+C} \Rightarrow \bar{A}\cdot\bar{B}\cdot\bar{C}$$

2) $\overline{ABC} = \bar{A}+\bar{B}+C$

Sln:

→ term → A, B, C

→ dual of expression → $\overline{A+B+C}$

→ complement each term → $\overline{\bar{A}+\bar{B}+\bar{C}}$

→ complement Expression → $\overline{\overline{\bar{A}+\bar{B}+\bar{C}}}$

$$\overline{ABC} \Rightarrow \bar{A}+\bar{B}+\bar{C}$$

Problems :

1) De Morganize the Expression $\overline{AB+CD+EFG}$

Sln:

→ Expn has three terms i.e AB , CD, EFG

→ Find dual of term. $\overline{(AB)\cdot(CD)\cdot(EFG)}$

→ complement each term $\overline{(\overline{AB})\cdot(\overline{CD})\cdot(\overline{EFG})}$

→ complement expression → $(\overline{AB})\cdot(\overline{CD})\cdot(\overline{EFG})$

→ Further AB – A, B, CD → C, D & EFG → E, F, G terms.

→ dual each term → $(\overline{A+B})\,(\overline{C+D})\cdot(\overline{E+F+G})$

→ complement each term → $(\overline{\bar{A}+\bar{B}})\,(\overline{\bar{C}+\bar{D}})\,(\overline{\bar{E}+\bar{F}+\bar{G}})$

→ Complement expression → $\overline{(\overline{\bar{A}+\bar{B}})\,(\overline{\bar{C}+\bar{D}})\,(\overline{\bar{E}+\bar{F}+\bar{G}})}$

→ $(\overline{\overline{\bar{A}+\bar{B}}})\cdot(\overline{\overline{\bar{C}+\bar{D}}})\cdot(\overline{\overline{\bar{E}+\bar{F}+\bar{G}}})$

→ $(\bar{A}+\bar{B})\cdot(\bar{C}+\bar{D})\cdot(\bar{E}+\bar{F}+\bar{G})$

2) De morganize Expression $x+y$

→ term x, y

→ dual of Expression → $x \cdot y$

→ complement each term → $\bar{x} \cdot \bar{y}$

→ complement Expression → $\overline{\bar{x} \cdot \bar{y}}$

$$x+y = \overline{\bar{x} \cdot \bar{y}}$$

3) De-morganize $xy + xz$

→ terms are $(xy)\ (xz)$
→ dual the expression → $(xy)\cdot(yz)$
→ Complement each term → $(\overline{xy})\cdot(\overline{xz})$

→ complement expression → $\overline{(\overline{xy})(\overline{xz})}$

→ further $xy \to x \,\&\, y$, $\overline{xz} \to x \,\&\, z$ terms.

→ dual the expression → $\overline{(\overline{x+y})\ (\overline{x+z})}$

→ complement each term → $\overline{(\overline{\bar{x}+\bar{y}})(\overline{\bar{x}+\bar{z}})}$

→ complement expression → $\overline{(\overline{\overline{\bar{x}+\bar{y}}})(\overline{\overline{\bar{x}+\bar{z}}})}$

⇒ $\overline{(\bar{x}+\bar{y})\ (\bar{x}+\bar{z})}$

☆ Solve De morgan's theorem:

1) $W+Q$
2) $(A+B+C)D$
3) $[\overline{(\overline{A+B})+\overline{C}}]$

Reduction of Switching equations.

1) $F = xy'z + xyz$
2) $P = x'yz' + x'yz + xyz' + xyz$
3) $F = a'b + ab$ (Ans $F = b$)
4) $w = abc' + abc$ $(w = ab)$
5) $Q = a'bc + a'b'c + a'b'c' + abc$ $(Q = a'b' + bc)$
6) $P = x(y+z')(x'y')$ $(P = xy'z)$
7) $R = (x'+y')(x+y')$ $(R = xy)$
8) $S = ab + a'b + ab' + a'b'$ $(S = 1)$

4) Using Boolean algebra, write an equivalent expression for $\bar{Y} \cdot Z + 1$.

$\Rightarrow \cancel{\bar{Y}(Z+1)}$

$\Rightarrow (\bar{Y}+1)(Z+1)$

$\Rightarrow (\cancel{Y})\ (1)$

5) $(X+\bar{Y}+Z)\cdot 1 = (X+\bar{Y}+Z)$ $\qquad A \cdot 1 = A$

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

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

7) $XZ + \overline{XZ}$ $\qquad \because XZ+\bar{X} = (X+\bar{X})(\cancel{Z}+\bar{X})$

$= XZ + \bar{X} + \bar{Z}$

$= (X+\bar{X})(Z+\bar{X}) + \bar{Z}$

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

$= (Z+\bar{Z}) + \bar{X}$

$= 1 + \bar{X}$

$= \cancel{X}\ 1$

8) $\overline{ab.c} + a\bar{b}\bar{c} + a\bar{b}c$

9) $(x(\overline{xy})) + (y(\overline{xy}))$

10) $f = xyz + \bar{x}z$

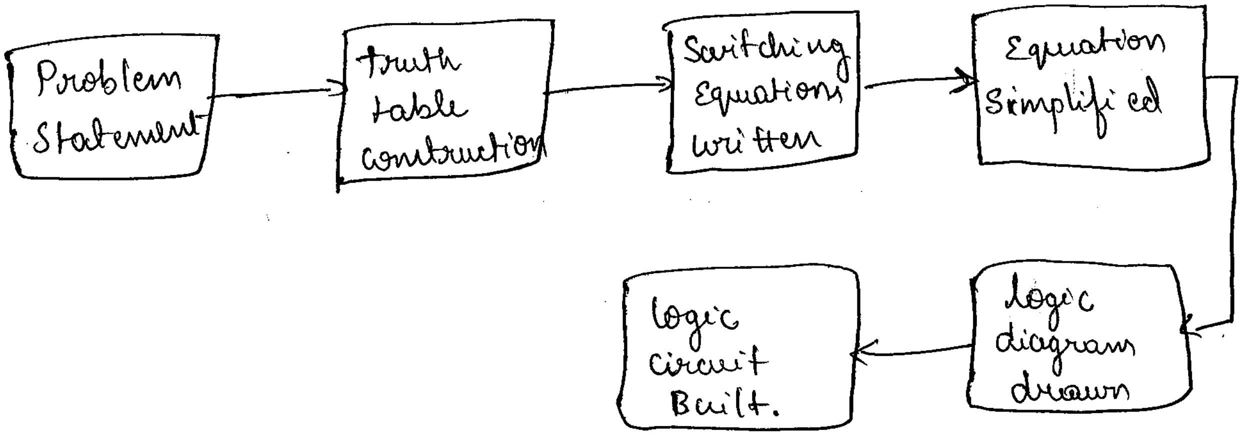
11) $f = abc + a\bar{b}c + \bar{a}$

12) $f = \overline{rst}\ (\overline{r+s+t})$

13) $f = (b+\bar{c})(\bar{b}+c) + (\overline{\bar{a}+b+c})$

General logic Design sequence:

List of various steps in designing a combinational logic system.



-> Proper statement of a problem is the most important part of ~~the~~ any digital design task.

-> Problem is rewritten in the form of a truth table after deciding the no. of input & o/p variables.

-> Switching Equations can be derived from truth table & can be ~~on~~ written in terms ~~of~~ SOP or POS ~~I/p & o/p variables~~.

-> Equations are simplified use various simplification techniques like Boolean algebra, K-map, MEV etc.

-> Finally the logic diagram is drawn & realized using any one of the three main digital integrated circuit families:

1) transistor Transistor logic (TTL)
2) Emitter Coupled logic (ECL)
3) Complementary metal oxide silicon (CMOS).

~~Canonical~~ forms:

Truth Table Representation:

Using ~~proplem~~ problem statement realization of truth table.

1. Develop the truth table for a system which accepts two 2-bit binary numbers & generates three outputs. The first output indicates when the 2 no.s differ by 2 or more, the second output indicates when the 2 numbers are identical & the 3rd O/P indicates when the first no. exceeds the second no.

I/P		O/P		
a b	c d	x	y	z
0 0	0 0	0	1	0
0 0	0 1	0	0	0
0 0	1 0	[illegible]	0	0
0 0	1 1	1	0	0
0 1	0 0	0	0	1
0 1	0 1	0	1	0
0 1	1 0	[illegible]	0	0
0 1	1 1	[illegible]	0	0
1 0	0 0	[illegible]	0	1
1 0	0 1	0	0	1
1 0	[illegible] 0	0	1	0
1 0	1 1	0	0	0
1 1	0 0	1	0	1
1 1	0 1	[illegible]	0	1
1 1	1 0	0	0	1
1 1	1 1	0	1	0

x

0 – 2

0 – 3

1 – 2 → 1

1 – 3

2 – 0

3 – 0

3 – 1

3 – 2 → 1

2) Give the truth table representation of a system which takes two 2-bit binary no.'s as its inputs & generates an output to indicate, when the sum of the two numbers is odd.

I/P				O/P
a	b	c	d	y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

3) Represent the no. of days in a month for a non-leap year by a truth table, indicating the o/p of invalid i/p's if any by 0.

Soln: Four bits are required to represent 12 months, with 0000 for January, 0001 for Feb, & so on.

No. of day are possibly

$$(28)_{10} = (11100)_2$$

$$(30)_{10} = (11110)_2$$

$$(31)_{10} = (11111)_2$$

	I/p					O/p				
	a	b	c	d		v	w	x	y	z
Jan	0	0	0	0	$\rightarrow$	1	1	1	1	1
Feb	0	0	0	1	$\rightarrow$	1	1	1	0	0
Mar	0	0	1	0	$\rightarrow$	1	1	1	1	1
Apr	0	0	1	1	$\rightarrow$	1	1	1	1	0
May	0	1	0	0	$\rightarrow$	1	1	1	1	1
June	0	1	0	1	$\rightarrow$	1	1	1	1	0
July	0	1	1	0	$\rightarrow$	1	1	1	1	1
Aug	0	1	1	1	$\rightarrow$	1	1	1	1	1
Sep	1	0	0	0	$\rightarrow$	1	1	1	1	0
Oct	1	0	0	1	$\rightarrow$	1	1	1	1	1
Nov	1	0	1	0	$\rightarrow$	1	1	1	1	0
Dec	1	0	1	1	$\rightarrow$	1	1	1	1	1
INV	1	1	0	0	$\rightarrow$	0	0	0	0	0
INV	1	1	0	1	$\rightarrow$	0	0	0	0	0
INV	1	1	1	0	$\rightarrow$	0	0	0	0	0
INV	1	1	1	1	$\rightarrow$	0	0	0	0	0

4) Write the truth table for a 4 I/p system indicating when majority of its i/p's are true.

5) Write the truth table of a four input logic system to indicate when ~~num~~ numbers divisible by 3 or 5 occur

6) An Electric motor powering a convayor used to move material is to be turned on when one of two operators is in position, if material is present to be moved & if the protective interlock switch is not open. i/p & o/p variables are to be Expressed in binary; i.e, if operator 1 is in position then the associated variable is a logic 1. The motor is running (on) if its output control variable is a 1. & the motor is off if the variable is 0. fig shows a simple diagram of convayor system.

-> The first task is to identify the i/p & o/p variables & to assign names to them.

Page no. 86

Digital logic text book.

★ TVUM table: (truth table)

The truth table is a breakdown of a logic function by listing all possible values the function can attain. Such a table typically contains several rows & columns, with the top row representing the logical variables & combinations, in increasing complexity leading up to the final function.

CANONICAL FORMS

Canonical is the word used to describe a condition of a switching equation. In normal use the word means " conforming to a general rule ".

-> The rule for switching Equation is that each term used in switching equation must contain all of the available input variables.

-> Two formats.

1) Sums of minterms -> canonical Sum of products (SOP)
2) products of maxterms. -> canonical Product of Sum (POS)

Representation of Minterms & maxterms in TT

Input variables			minterms		Maxterms	
a	b	c	term	Designation	term	designation.
0	0	0	$a'b'c'$	m_0	$a+b+c$	M_0
0	0	1	$a'b'c$	m_1	$a+b+c'$	M_1
0	1	0	$a'bc'$	m_2	$a+b'+c$	M_2
0	1	1	$a'bc$	m_3	$a+b'+c'$	M_3
1	0	0	$ab'c'$	m_4	$a'+b+c$	M_4
1	0	1	$ab'c$	m_5	$a'+b+c'$	M_5
1	1	0	abc'	m_6	$a'+b'+c$	M_6
1	1	1	abc	m_7	$a'+b'+c$	M_7

Definitions:

Minterm: A minterm is a special case product (AND) term. A minterm is a product term that contains all of the input variables (can't repeat) that make up a boolean expression.

Maxterm: A maxterm is a special case sum (OR) term. A maxterm is a sum term that contains all of the input variables (can't repeat) that makeup a Boolean Expression.

Canonical sum of products (Sum of minterms):

A canonical sum of products is a complete set of minterms that defines when an o/p variable is logic 1.

Canonical product of sums (Product of maxterms):

A canonical product of sums is a complete set of maxterms that defines when the o/p variable is logic 0.

~~SOP~~: ~~Sum of products~~ is a ~~co set of variables arranged in sum of product terms~~.

~~POS~~:

SOP: Expressions is a set of product terms connected with logical sum operators.

POS - expressions is a set of sum (OR) terms connected with logical product (AND) operators.

★ To place SOP eqn into canonical form using boolean algebra, we do the following.

1. Identify the missing variable(s) in each AND term.
2. AND the missing term & its complement with the original AND term.

$$xy(z+z') \quad \because (z+z') = 1 \rightarrow$$ value not changed.

3) Expand the term by application of the property of distribution.

$$xyz + xyz'$$

★ To place POS eqn into canonical form using boolean algebra, we do the following.

1. Identify the missing variables in each (OR) term.
2. OR the missing terms & its complement with the original OR term, $(x+y'+zz')$.
Because $zz' = 0$, the original OR term value is not changed.
3. Expand the term by application of the property of Distribution.

$$(x+y'+z)(x+y'+z')$$

Problem:

Place the following eqn into the proper canonical form:

1) $P = f(a,b,c) = ab' + ac' + bc$ (SOP)

2) $G = f(w,x,y,z) = w'x + yz'$ (SOP)

3) $T = f(a,b,c) = (a+b')(b'+c)$ (POS)

4) $J = f(A,B,C,D) = \cancel{(A+B'+C+D)}\ (A+B'+C)(A'+D)$ (POS)

Generation of switching equations from truth tables (SOP & POS) forms.

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

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

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

$$= (111 + 110 + 101 + 000 + 100)$$

$$= m_7 + m_6 + m_5 + m_0 + m_4$$

$\Sigma m(0, 4, 5, 6, 7)$ — SOP

for POS

$\Pi M(1, 2, 3)$

Represent in terms of truth table values.

1) Expand $(\bar{A}+B)(B+\bar{C})$ to maxterms.

Example.

Decimal value	Inputs $x\ y\ z$	O/P f	fundamental Products	Minterm	fundamental Sum	maxterm
0	0 0 0	0	$x'y'z'$	m_0	$x+y+z$	M_0
1	0 0 1	0	$x'y'z$	m_1	$x+y+z'$	M_1
2	0 1 0	1	$x'yz'$	m_2	$x+y'+z$	M_2
3	0 1 1	1	$x'yz$	m_3	$x+y'+z'$	M_3
4	1 0 0	0	$xy'z'$	m_4	$x'+y+z$	M_4
5	1 0 1	0	$xy'z$	m_5	$x'+y+z'$	M_5
6	1 1 0	1	xyz'	m_6	$x'+y'+z$	M_6
7	1 1 1	1	xyz	m_7	$x'+y'+z'$	M_7

A canonical SOP is a complete set of minterms that defines when an O/P variable is a logical 1.

$$f_1 = x'yz' + x'yz + xyz' + xyz$$

$$= m_2 + m_3 + m_6 + m_7$$

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

A canonical POS is a complete set of maxterms that defines when an o/p is a logical 0.

$$f_2 = (x+y+z)(x+y+z')\cdot(x'+y+z)(x'+y+z')$$

$$= M_0 \cdot M_1 \cdot M_4 \cdot M_5$$

$$POS = \Pi M(0, 1, 4, 5)$$

Problems:

Express the following SOP equation in a minterm list

a) $f(A,B,C) = A'BC + A'B'C + ABC$

$= 011 + 001 + 111$

$= m_3 + m_1 + m_7$

$= \Sigma(1,3,7)$

b) $f(w,x,y,z) = wxyz' + wx'yz' + w'xyz' + w'x'yz'$

c) $T = f(a,b,c) = (a+b'+c)(a+b'+c')(a'+b'+c)$

d) $f(A,B,C,D) = (A+B'+C+D)(A+B'+C+D')(A'+B+C+D)$
$(A'+B'+C+D)(A'+B+C'+D)(A'+B'+C'+D)$

e) from truth table find SOP for o/p A & A'
POS for o/p. A & A'.

I/p's			O/P	
x	y	z	A	A'
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

Problem

* Write the truth table for function & algebraic form of expression POS or SOP form

1) $f = \pi(2, 7)$
2) $f = \pi(8, 14, 15)$
3) $f_{(a,b,c)} = \pi(0, 3, 5)$
4) $f(a,b,c,d) = \pi(0, 3, 5)$
5) $f(a,b,c,d) = \Sigma(3, 4, 9, 12)$

* Represent the following in both canonical maxterm & minterm forms in decimal notation.

1) $f = \bar{x}y + yz$
2) $f = (\bar{a}+b)(b+\bar{c})$
3) $f = \bar{a}b + c\bar{d}$
4) $f = p + \bar{q}r$

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

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

---

* SOP $\longrightarrow$ POS

$\Sigma \longrightarrow \pi$

Example: $M_i = \overline{m_i}$

$\Sigma(1, 4, 7) = \pi(0, 2, 3, 5, 6)$

KARNAUGH MAPS (K-MAP)

-> It is originated from the 'map method' proposed by Edward Veitch also called the 'Veitch Diagram' & then modified by "Maurice Karnaugh" 1953.

-> Developed by Karnaugh in 1953 that he presented in his paper entitled "The map method for synthesis of combinational logic circuit."

-> Boolean expressions can be simplified algebraically. ~~The effectiveness~~ of ~~algebraically~~. The effectiveness of algebraic simplification depends on the knowledge and ability to apply Boolean algebraic rules, laws & theorems.

-> The K-map is a systematic method of simplifying Boolean Expressions.

-> It is a practical form of truth table & could handle up to six variables.

-> It is an array of cells/squares in which each cell represents a binary value of the input variables.

-> The K-map is a chart or a graph, composed of an arrangement of adjacent cells, each representing a particular combination of variables in SOP form.

-> K-Map can easily be applied for problems involving up to six variables.

-> An 'n' variables function can have 2^n possible combinations of product terms in SOP form or 2^n possible combinations of POS forms.

★ Since the K-map is a graphical representation of Boolean Expressions.

- a two-variable K-map has $2^2 = 4$ cells or squares.
- a three-variable K-map has $2^3 = 8$ cells.
- a Four-variable K-map has $2^4 = 16$ cells
- a Five-variable K-map has $2^5 = 32$ cells & so on.

K-map Setup:

-> The number of cell/square in a K-map is equal to 2^n, where 'n' is the number of input variables.

-> The map is drawn to ~~all~~ show the relationship between squares & I/p's variables.

-> variables are assigned to row & column.

-> Binary marking are placed in each row & column using reflected (Gray) code sequence.

-> A Karnaugh map compraises a box for every line in the truth table. the binary value for each box is the binary value of the input terms in the corresponding table row & column.

-> Unlike a truth table, in which the input values typically follow a standard binary sequence (00, 01, 10, 11). ~~A~~ (the Karnaugh map's input values must be ordered such that the values for adjacent columns vary by only a single bit, for Example 00, 01, 11 & 10. this ordering is known as a Gray code.)

2 variable K-map:

LSB

MSB

A \ B	0	1
0	0	1
1	2	3

minterms Decimal values.

$2^2 = 4$ Squares

3 & 4 variable K-map:

LSB — Gray code.

MSB

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

$2^3 = 8$ Squares

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

$2^4 = 16$ squares.

Imp.
★ Terms:

Implicants: It is the minterm corresponding to the cell which is having '1' ~~to the~~ in the K-map (Single cell)

Prime Implicants: All the possible grouping in a K-map is called as Prime Implicant.

Essential prime Implicants: It is a prime Implicant which is having atleast a single one which is only one time grouping. & not covered by any other prime implicants.

Redundent PI: It is a prime Implicant with or without it there is no effect in ckt resultant.

Example problem:

1) $Y = f(a,b,c) = \Sigma(0,1,4,5)$

Simplify using K-map method.

Soln: given minterm values.

3 variable I/p function $\therefore$ $2^3 = 8$ squares.

a \ bc	00	01	11	10
0	1 (m_0)	1 (m_1)	0 (m_3)	0 (m_2)
1	1 (m_4)	1 (m_5)	0 (m_7)	0 (m_6)

Grouping of 1's done for 2, 4, 8 ... so on.

→ all rows are covered so no 'a' terms

→ two columns of the K-maps are grouped.

common term of two-column. $\bar{b}\bar{c}$ 00, $\bar{b}c$ 01 is $(\bar{b})$

$Y = f(a,b,c) = \bar{b}$

Prime Implicants are: PI: ~~{0,1}~~, ~~{4,5}~~, {0,1,4,5}

Essential prime Implicant: EPI {0,1,4,5}

RPI = {0,1} {4,5}

2) Simplify the following expression using a three variable K-map.

$Q = f(a,b,c) = \Sigma(1,2,3,6,7)$

a \ bc	00	01	11	10
0	0 (m_0)	1 (m_1)	1 (m_3)	1 (m_2)
1	0 (m_4)	0 (m_5)	1 (m_7)	1 (m_6)

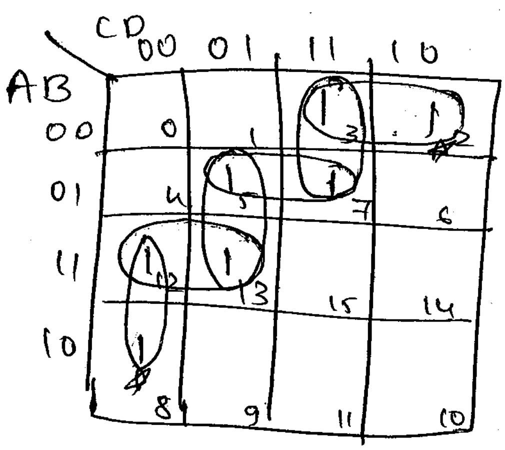
PI: ~~{3,2}~~ ~~{7,6}~~ {1,3}

{3,2,7,6}

EPI: {1,3} {3,2,7,6}

Examples:

1) $f(A,B,C,D) = \Sigma(2,3,5,7,8,12,13)$



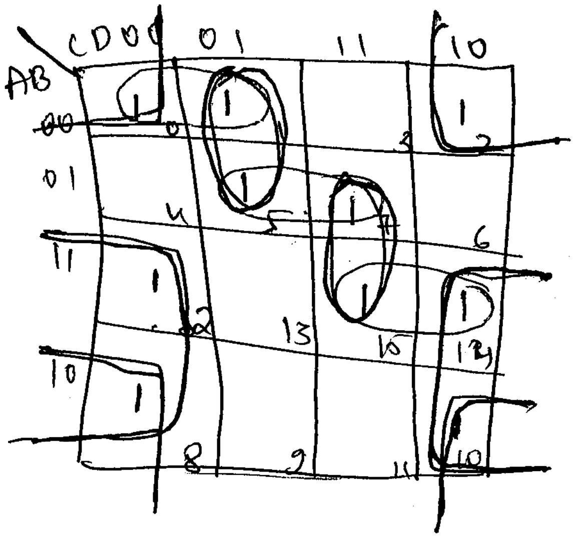
PI - 6 = {3,2}, {5,7}
{3,7}, {5,13}
{12,13}, {12,8}

EPI = 2 = {3,2}, {12,8}

RPI = 2 = {5,13}, {3,7}

$Y = AC'D' + A'B'C + ABC' + A'BD$

2) $f(A,B,C,D) = \Sigma(0,1,2,5,7,8,10,12,14,15)$



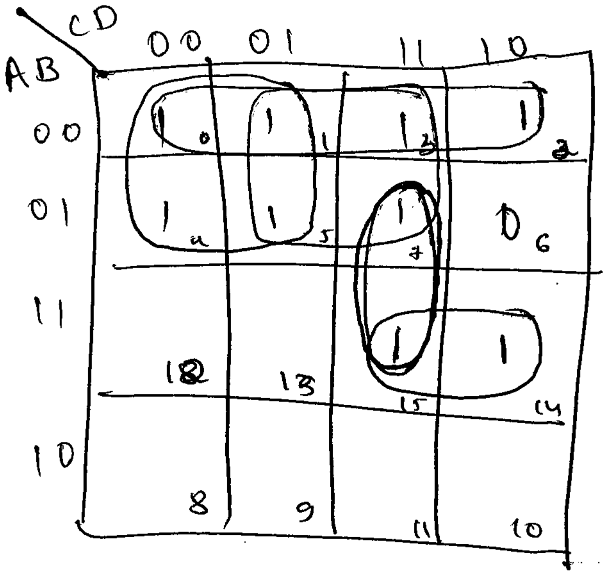
PI - 7

EPI = 2

RPI = 3

$Y = A\bar{D} + \bar{B}\bar{D} + \bar{A}\bar{C}D + BCD$

3) $f(A,B,C,D) = \Sigma(0,1,2,3,4,5,6,7,15,14)$



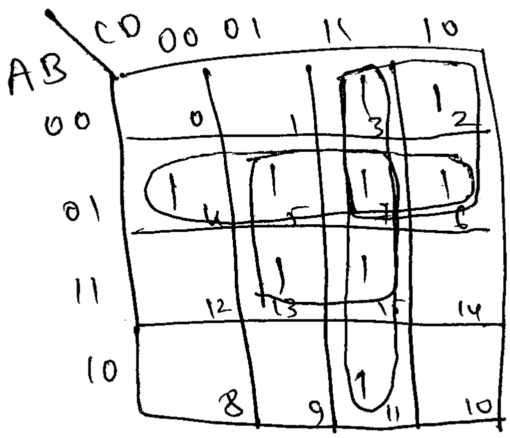
PI = 5

EPI = 3

RPI = 1

$Y = A'C' + A'D + A'B' + ABC$

4) $f(A,B,C,D) = \Sigma\{2,3,4,5,6,7,11,13,15\}$



PI = 4

EPI = 4

RPI = 0

$Y = A'B + BD + CD + A'C$

Problem:

Reduce the following Equation using a four variable K-map.

1) $L = f(a,b,c,d) = \Sigma(0,2,5,7,8,10,13,15)$

2) $K = f(w,x,y,z) = \Sigma(0,1,4,5,9,11,13,15)$

3) $P = f(r,s,t,u) = \Sigma(1,3,4,6,9,11,12,14)$

4) $D = f(w,x,y,z) = \Sigma(5,7,8,9,13)$

5) $P = f(a,b,c,d) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$

6) $S = f(a,b,c,d) = \Sigma(1,3,4,5,7,8,9,11,15)$

7) $U = f(w,x,y,z) = \Sigma(1,5,7,8,9,10,11,13,15)$

8) $f(w,x,y,z) = \pi M(1,5,6,7,11,12,13,15)$

9) $f(w,x,y,z) = \Sigma m(1,5,6,7,11,12,13,15)$

Reduce the following equation using a 2 - 3 variable K-map.

1) $f(x,y) = \pi M(2,3)$

2) $f(x,y) = \Sigma m(1,2)$

3) $f(x,y,z) = \Sigma m(0,2,5,7)$

4) $f(x,y,z) = \Sigma m(1,2,5,6)$

5) $f(x,y,z) = \pi M(0,1,6,7)$

6) $f(x,y,z) = \pi M(1,3,5,7)$

4 variable K-map:

1) $f(x,y,z,w) = \pi M(0,1,4,5,7,10,11,13,14,15)$

2) $f(w,x,y,z) = \pi M(1,3,5,7,8,9,10,12,13,14)$

✯ 5 & 6 Variable K-map:

$\bar{C}$ $\quad$ C → adjacent colomn.

AB \ CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

DE \ ABC	000	001	011	010	110	111	101	100
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

Problems:

1) Five variable K-map for

$T = f(a, b, c, d, e) = \Sigma(0, 2, 8, 10, 16, 18, 24, 26)$

ab \ cde	000	001	011	010	110	111	101	100
00	1 (0)	(1)	(3)	1 (2)	(6)	(7)	(5)	(4)
01	1 (8)	(9)	(11)	1 (10)	(14)	(15)	(13)	(12)
11	1 (24)	(25)	(27)	1 (26)	(30)	(31)	(29)	(28)
10	1 (16)	(17)	(19)	1 (18)	(22)	(23)	(21)	(20)

$T = c'd'e' + c'de'$

2) $w = f(a, b, c, d, e) = \Sigma(1, 3, 4, 6, 9, 11, 12, 14, 17, 19, 20, 22, 25, 27, 28, 30)$

ab \ cde	000	001	011	010	110	111	101	100
00	(0)	1 (1)	1 (3)	(2)	1 (6)	(7)	(5)	1 (4)
01	(8)	1 (9)	1 (11)	(10)	1 (14)	(15)	(13)	1 (12)
11	(24)	1 (25)	1 (27)	(26)	1 (30)	(31)	(29)	1 (28)
10	(16)	1 (17)	1 (19)	(18)	1 (20)	(21)	(23)	1 (22)

$w = \cancel{d'e} + c'e + ce'$

Solve:

1) $J = f(v, w, x, y, z) = \Sigma(4, 5, 6, 7, 9, 11, 13, 15, 25, 27, 29, 30)$

2) $f(v, w, x, y, z) = \pi M(0, 1, 2, 3, 5, 7, 8, 10, 13, 15, 16, 17, 18, 19, 21, 23, 24, 25, 26, 27, 29, 31)$

3) $f(v, w, x, y, z) = \Sigma m(0, 2, 5, 7, 8, 10, 13, 15, 16, 18, 21, 23, 24, 26, 29, 31)$

* In complelty Specified functions (Don't care terms)

-> When an output value is known for every Possible combination of input variables, the fun is said to be completely specified.

-> When ~~the~~ an O/P value is not known for every combination of I/P variables, because all combinations cannot occur, the function is said to be incompletly specified

-> The truth table does not generate an O/P value for every possible combination of input variables.

-> They are not called as minterm, maxterm -> function are called "don't care terms"

For Example (BCD) binary to Ex-3 BCD conversions.

Binary W	X	Y	Z	Ex-3 A	B	C	D
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	⋮			
⋮ 1	1	1	1	X	X	X	X

→ only BCD → EX-3 can be done only for single digit no., 0 → 9.
9 → we 4 digit value. but 10 – 15 no O/P produced for EX-3 function.

∴

$A = f(w,x,y,z) = \Sigma(5,6,7,8,9) + \Sigma d(10,11,12,13,14,15)$

$B = f(w,x,y,z) = \Sigma(1,2,3,4,9) + \Sigma d(10,11,12,13,14,15)$

$C = f(w,x,y,z) = \Sigma(0,3,4,7,8) + \Sigma d(10,11,12,13,14,15)$

$D = f(w,x,y,z) = \Sigma(0,2,4,6,8) + \Sigma d(10,11,12,13,14,15)$

Simplify using K-map.

wx \ yz	00	01	11	10
00	0	1	3	2
01	4	1 (5)	1 (7)	1 (6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	1 (9)	X (11)	X (10)

xz, xy, w

$A = w + xz + xy$

wx \ yz	00	01	11	10
00	0	1 (1)	1 (3)	1 (2)
01	1 (4)	5	7	6
11	X (12)	X (13)	X (15)	X (14)
10	8	1 (9)	X (11)	X (10)

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

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

wx \ yz	00	01	11	10
00	1 (0)	1	1 (3)	2
01	1 (4)	5	1 (7)	6
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	9	X (11)	X (10)

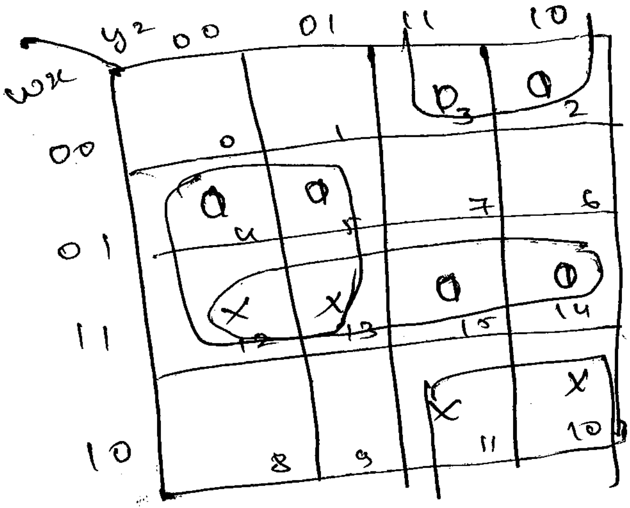
$C = y'z' + yz$ ~~$w'x'z$~~

wx \ yz	00	01	11	10
00	1 (0)	1	3	1 (2)
01	1 (4)	5	7	1 (6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	9	X (11)	X (10)

$D =$ ~~$y'z'$~~ $+$ ~~yz'~~ z'

☆ Solve

$f(w,x,y,z) = \Pi M(2,3,4,5,14,15) + d(10,11,12,13)$



$f(w,x,y,z) = wx + \underline{xy' + x'y}$

$= (w'+x')\ \underline{(x'+y)(x+y')}$

Problem:

1) $f(a,b,c,d) = \Sigma(0,1,2,5,8,15) + \Sigma d(6,7,10)$
2) $f(a,b,c,d) = \Pi(2,8,11,15) + \Pi d(3,12,14)$
3) $f(a,b,c,d) = \Sigma(7,9,11,12,13,14) + \Sigma d(3,5,6,15)$
4) $f(a,b,c,d) = \Pi(1,2,3,4,9,10) + \Pi d(0,14,15)$
5) $f(a,b,c,d) = \Sigma(6,7,9,10,13) + \Sigma d(1,4,5,11,15)$

☆ Quine - McCluskey Minimization procedure:

→ It is a algorithm used to reduce or minimization of Boolean functions which ~~has~~ was developed by W.V. Quine & Edward J. McCluskey in 1956.

→ It is a tabular form makes it more efficient for use in computer algorithms.

→ It gives a deterministic way to check that the minimal form of a boolean function has been reached..

The procedure for minimization is as follows:

Step 1: Describe individual minterms/maxterms & don't care terms of the given boolean Expression by their Equivalent binary numbers. Differentiate don't care terms using ☆ or - or X terms.

Step 2: Form a table by grouping numbers representing minterms/ maxterms having Equivalent number of 1's & arrange them in ascending order i.e. first no's having no. 1's, then number having one 1, then having two 1's, etc.

Grouping:

Group	no. of 1's in minterm	Four bit binary no. representing miterms.
I	zero	0000,
II	one	0001, 0010, 0100, 1000
III	two	0011, 0101, 0110, 1001, 1010, 1100
IV	Three	0111, 1011, 1110, 1101,
V	Four	1111.

<u>Step3:</u> Compare each number in the <u>mth</u> <u>top group</u> with each minterm/maxterm including don't care term in the next $(m+1)^{th}$ lower group looking for one variable change.

If the two numbers are the same in every position but one place is different, then mark a check sign (✓) to the right of both numbers to show that these numbers have been paired. Then enter the newly founded number in the next column (new). The new no. replaces the old paired numbers but where the literal differ an 'x' is placed in the position of that literal.

0000 (0) ✓
0001 (1) ✓ ⇒ 000X (0,1)

0101 (5) ✓
0111 (7) ✓ ⇒ 0X1X (0,3)

1101 (13)
1111 (15) ⇒ 11X1 (13,15)

01X1 (5,7)
11X1 (13,15) ⇒ X1X1 (5,7,13,15)

Step 4: using step 3 above, form a second table & repeat the process again untill no further pairing is possible. ~~one~~ on second repeat, compare number of m^{th} to numbers in the next $(m+1)^{th}$ group that have the same 'x' position.

<u>Step5:</u> Terms which were not covered are the prime implicants & are ORed & ANDed together to form a final function. Name prime implicants ~~& as~~ P_j, where j is integer number.

Step 6: Construct the prime-implicants map & determine essential prime implicants. Don't consider don't care term part of group.

Select the row corresponding to column ~~full~~ that has one 'x' & mark the minterms included by that row & columns. Repeat the process of selection of rows & columns till all the minterms grouped by all the prime implicants are marked.

Reduction of prime implicant table:

1) check the essential columns & remove them,
2) check for row dominance & remove all dominating rows
3) check for column dominance & remove all dominate columns.
4) Repeat 1, 2, 3 if there is any removal occurs.

☆ Essential columns: A column is essential if it covers one 1-entry that cannot be covered by any other columns.

☆ Row ~~R_i~~ r_i dominates row r_j if r_i has all the 1 entries in r_j, r_i is dominating & r_j is dominated.

☆ column P_i dominates P_j, if P_i has all the 1 entries in P_j, P_i is dominating & P_j is dominated.

Problem:

1) Simplify the given Boolean expression using Quine-McCluskey procedure $f(w,x,y,z) = \bar{w}\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}xy\bar{z} + \bar{w}xyz + w\bar{x}\bar{y}z + w\bar{x}y\bar{z} + wx\bar{y}z + wxy\bar{z} + wxyz$

minterm	Binary repn.	Number of 1's in minterm	Assign Group Number
$\bar{w}\bar{x}\bar{y}\bar{z}$	0000	0	1
$\bar{w}x\bar{y}z$	0101	2	2
$\bar{w}xy\bar{z}$	0110	6	2
$\bar{w}xyz$	0111	7	3
$w\bar{x}\bar{y}z$	1001	9	2
$w\bar{x}y\bar{z}$	1010	10	2
$wx\bar{y}z$	1101	13	3
$wxy\bar{z}$	1110	14	3
$wxyz$	1111	15	4

Group		
1	0000 ✓	0x0x (I)
2	0101 ✓ 0110 ✓ 1001 ✓ 1010 ✓	0xx0 (I) x00x (I) x0x0 (I)
3	0111 1101 1110	
4	1111	

* Start pairing off each element of first group with the next, however since m_0 has no 1's, it and the next group of numbers with one 1's are missing, therefore they cannot be paired off.

Group	Step 1	
1	0000	(P_1)
2	(5) 0101 ✓	
	(6) 0110 ✓	
	(9) 1001 ✓	
	(10) 1010 ✓	
3	(7) 0111 ✓	
	(13) 1101 ✓	
	(14) 1110 ✓	
4	(15) 1111	

Step 2		
01X1	(5,7)	✓
X101	(5,13)	✓
011X	(6,7)	✓
X110	(6,14)	✓
1X01	(9,13)	P_2
1X10	(10,14)	P_3
X111	(7,15)	✓
11X1	(13,15)	✓
111X	(14,15)	✓

Step 3		
X1X1	(5,7,13,15)	P_4
X1X1	(5,13,7,15)	×
X11X	(6,7,14,15)	P_5
X11X	(6,~~14~~,7,15)	×

minimal prime Implicants.

Prime Implicant		Decimal	0 ✓	5 ✓	6 ✓	7	9 ✓	10 ✓	13	14	15
P_1	$\bar{w}\bar{x}\bar{y}\bar{z}$	(0)	⊗								
P_2	$w\bar{y}z$	(9,13)					⊗		X		
P_3	$wy\bar{z}$	(10,14)						⊗		X	
P_4	xz	(5,7,13,15)		⊗		X			X		X
P_5	xy	(6,7,14,15)			⊗	X				X	X

Select row-wise & column wise so that all the minterm represented by prime-implicants are covered. Select columns 0, 5, 6, 9 & 10 because there is only one 'x' in these columns, select P_1, P_4, P_5, P_2 & P_3 rows corresponding to 0, 5, 6, 9, & 10 columns, ~~repre~~ respectively. Therefore, write the minimized SOP as given below.

$$F(w,x,y,z) = \bar{w}\bar{x}\bar{y}\bar{z} + w\bar{y}z + wy\bar{z} + x\cancel{y}z + xy.$$

2) Simplify the given boolean expression using Quine - McCluskey procedure

$$F(A,B,C) = \Sigma m(0,1,2,5,6,7)$$

Group	minterm	Step 1		
1	(0) 000 ✓	00×	(0,1)	P_1
		0×0	(0,2)	P_2
2	(1) 001 ✓	×01	(1,5)	P_3
2	(2) 010 ✓	×10	(2,6)	P_4
3	(5) 101 ✓	1×1	(5,7)	P_5
3	(6) 110 ✓	11×	(6,7)	P_6
4	(7) 111			

minimal prime Implicant table:

Prime Implicant	Decimal	Minterm	0 ✓	1 ✓	2 ✓	5 ✓	6 ✓	7 ✓
P_1	(0,1)	$\bar{A}\bar{B}$	×	×				
P_2	(0,2)	$\bar{A}\bar{C}$	×		×			
P_3	(1,5)	$\bar{B}C$		×		×		
P_4	(2,6)	$B\bar{C}$			×		×	
P_5	(5,7)	AC				×		×
P_6	(6,7)	AB					×	×

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

Another posiblity can also be explained as given in table.

Prime Impliant	Decimal	minterms	0 ✓	1 ✓	2 ✓	5 ✓	6 ✓	7 ✓
P_1	(0,1)	$\bar{A}\bar{B}$	X	X				
P_2	(0,2)	$\bar{A}\bar{C}$	X		X			
P_3	(1,5)	$\bar{B}C$		X		X		
P_4	(2,6)	$B\bar{C}$			X		X	
P_5	(5,7)	AC				X		X
P_6	(6,7)	AB					X	X

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

3) Simplify the given Boolean expression using Quine-McCluskey procedure.

$$F(A,B,C,D) = \pi M(4,5,6,8,9) \cdot \pi d(0,7,15)$$

Soln. POS Prime implicants are Sum rather than product form. Don't care terms included in group of stage 1. to find prime implicants.

Group:	minterm.
1	0000 (0) ✓
2	0100 (4) ✓
2	1000 (8) ✓
3	0101 (5) ✓
3	0110 (6) ✓
3	1001 (9) ✓
3	1010 (10) ✓
4	0111 (7) ✓
4	1101 (13) ✓
5	1111 (15)

Step 1

0X00	(0,4)	P_1
X000	(0,8)	P_2
010X	(4,5) ✓	
01X0	(4,6) ✓	
100X	(8,9)	P_3
10X0	(8,10)	P_4
011X	(6,7) ✓	
X101	(5,13) ✓	
01X1	(5,7) ✓	
1X01	(9,13)	P_5
X111	(7,15) ✓	
11X1	(13,15) ✓	

Step (2)

01XX	(4,5,6,7)	P_6
01XX	(4,6,5,7)	X
X1X1	(5,13,7,15)	X
X1X1	(5,7,13,15)	P_7

minimal prime implicant map:

	Prime Implicant	Decimal	4 ✓	5 ✓	6 ✓	8 ✓	9 ✓	10 ✓	13 ✓
P_1	$A+C+D$	(0,4)	X						
P_2	$B+C+D$	(0,8)				X			
P_3	$\bar{A}+B+C$	(8,9)				X	X		
P_4	$\bar{A}+B+D$	(8,10)				X		X	
P_5	$\bar{A}+C+\bar{D}$	(9,13)					X		X
P_6	$A+\bar{B}$	(4,5,6,7)	X	X	X				
P_7	$\bar{B}+\bar{D}$	(5,7,13,15)		X					X

Don't care terms 0, 7 & 15 are not considered as columns to find Essential prime Implicants as given in table column 6 & 10 have only one 'x' So select corresponding row P_4 & P_6 ~~as~~ selected. mark the columns covered by P_4 & P_6 rows. only column 9 & 13 are not selected. So select P_5 to complete the selection of columns.

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

★ **Procedure to select Prime implicants to obtain a Single minimal Sum.**

1) Identify all the essential prime implicants. i.e, look for columns with only one 'x' and the essential prime implicant can be found in the corresponding row called the essential row. Score out all the rows & all the columns which have 'x' in the essential row.

2) Score out all dominating columns & dominated rows. Dominated rows are deleted subject to the cost conditions stated. ~~in~~ ~~secti~~ in Example.

3) If any column that is left has a single x, identify the corresponding essential prime implicants along the row called the secondary essential row. Score out the secondary essential row & all columns with single 'x', but more than one 'x', the we have a cyclic table.

4. At this point if all the columns have been scored out, all the essential prime Implicants consistitute the minimal sum.

If all the columns are not scored out, repeat steps 2&3 until either all the columns are scored out or a cyclic table is obtained.

Example:

	m_0	m_1	m_2	m_5	m_{10}	m_{11}	m_{14}	m_{15}
A	X		X					
B	X	X						
C		X		X				
D							X	
E					X	X	X	X

Prime Implicant table to illustrate column reduction.

→ m_{10}, m_{11} & m_{15} have only one 'x' & equal to each other.

→ A column is said to dominate another, if in addition to having 'x''s in the same row's as that of the other column, it has 'x''s in rows in which the other column does not ~~have~~ have X's.

→ In column m_{14} dominates columns m_{10}, m_{11}, m_{15}; column m_0 dominates column m_2 & column m_1 dominates m_5 ...

A column C_i in a prime implicant table can be deleted if

a) C_i Equals C_j

b) C_i dominates C_j.

where C_j is some other column in the table. In other words, Equal & dominating columns can be removed.

column m_0 is deleted as it dominates column m_2,

column m_1 is deleted as it dominates column m_5

column m_{10} & m_{11} are deleted as they equal column m_{15}

column m_{14} is deleted as it dominates column m_{15}

after reduction.

	m_2	m_5	m_{15}
A	x		
B		x	
C			
D			x
E			

Since the remaining columns have only one 'x' each, these correspond to the essential prime implicants.

$\therefore$ the irredundant disjunctive normal Expression which is also the minimum sum is

$$f = A + B + C$$

Similar rules could also ~~be~~ applied to ~~detect~~ delete rows in a prime Implicant table.

If two rows have 'x's at exactly the same columns, they are said to be Equal.

A row is said to dominate another if in addition to having 'x's in the same columns as that of the other row, it has x's in columns in which the other row has no x's.

Ex:

	m_1	m_5	m_7	m_9	m_{10}	m_4	cost
A	x		x	x			3
B	x		x				4
C		x			x		4 —
D						x	3
E		x			x		3

row C & E are Equal in which row A dominates row B.

1) when 2 rows are Equal remove the row with higher or Equal cost.

2) Remove a dominated row if it costs more or Equal to the dominating row.

Problem:

1) Find the minimal cover for the prime implicant table shown below.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	cost
r_1			×		×		×	3
r_2				×		×		4
r_3		×	×			×		4
r_4	×	×				×		4
r_5			×		×		×	4
r_6	×	×			×	×		6
r_7			×	×	×			6

→ no column has single '×', hence there are no essential prime implicants.

→ Delete C_2 as C_2 dominates C_1

→ Delete C_6 as C_6 dominates C_1 & C_2

→ Delete C_3 as C_3 dominates C_7

→ Delete C_5 as C_5 dominates C_7

The table reduces to.

	C_1	C_4	C_7	cost
r_1			×	3
r_2		×		4
r_4	×			4
r_5			×	4
r_6	×			6
r_7		×		6

Delete r_5 as $r_1 = r_5$ & cost of r_5 > cost of r_1

Delete r_7 as $r_7 = r_2$ & cost of r_7 > cost of r_2

delete r_6 as $r_6 = r_4$ & cost of r_6 > cost of r_4

The table reduces to..

	C_1	C_4	C_7	cost
r_1			×	
r_2		×		
r_4	×			

minimal cover is r_1, r_2, r_4.

minimal sum. $f = \underline{r_1 + r_2 + r_4}$

Problem:

Find the minimal cover for the prime implicant table shown below.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	cost
r_1		×		×	×		×					3
r_2						×				×		3
r_3											×	4
r_4	×	×						×	×			4
r_5	×			×			×					4
r_6				×					×	×		5
r_7	×						×	×				6
r_8			×			×			×	×		7
r_9		×	×		×				×			7

column Delete - $C_1 \xrightarrow{\text{Dominates}} C_8$ ☆ C_{81} is one 'x' r_3 is essential Prime Implicant.

$C_2 \rightarrow C_5$

$C_9 \rightarrow C_3$

$C_{10} \rightarrow C_6$

row delete - $r_5 \rightarrow r_1$ cost $r_5 > r_1$

$r_6 \rightarrow r_1$ $r_6 > r_1$

$r_2 \rightarrow r_8$ $r_2 < r_8$ — Delete r_2

$r_4 \xrightarrow[\text{dominated by.}]{} r_7$

table becomes: ☆ C_4 has one 'x', r_1 is prime Implicant. delete all the columns which has 'x' in row 'r_1'. delete C_5 & C_7

	C_3	C_6	C_8	cost
r_2		X		3
r_4			X	4
r_7			X	6
r_8	X	X		7
r_9	X			7

$r_7 = r_4$ & $r_7 > r_4$ delete r_7

r_9 is dominated by r_8 $\rightarrow$ $r_9 = r_8$

delete r_8.

	C_3	C_6	C_8	cost
r_2		X		3
r_4			X	4
r_8	X	X		7

→ C_3 has single 'x' along r_8 and c_8 has single 'x' along r_4.

→ Hence r_4 & r_8 figure in minimal cover.

→ The minimal cover so far is $\{r_1, r_3, r_4, r_8\}$

→ ~~C_3~~ C_3 & C_8 are deleted. C_6 is also deleted as it has an 'x' along row r_8.

→ The minimal cover is $\{r_1, r_3, r_4, r_8\}$