

COMBINATIONAL LOGIC CIRCUITS

13 This chapter discusses Boolean algebra & several simplification techniques.

Boolean Laws & Theorems

Commutative Law

$$A+B = B+A$$

$$AB = BA$$

Associative Law

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

$$A(BC) = (AB)C$$

Distributive Law

$$A(CB+C) = AB+AC$$

OR operation:

$$A+0 = A$$

i.e. $0+0 = 0$
 $1+0 = 1$

$$A+A = A$$

i.e. $0+0 = 0$
 $1+1 = 1$

$$A+1 = 1$$

i.e. $0+1 = 1$
 $1+1 = 1$
 $A+\bar{A} = 1$

AND operation:

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot 0 = 0$$

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

Double Inversion & De Morgan's Theorem:

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

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

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

$$A+AB = A$$

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

$$LHS = A(1+B)$$

$$= A(1)$$

$$= A$$

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

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

$$LHS = A(B + \overline{B})$$

$$= A(1)$$

$$= A$$

Prove that $A(\overline{A}+C)(\overline{A}B+C)(\overline{A}BC+\overline{C}) = 0$

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

$$= (\underbrace{A\overline{A}}_{=0} + AC)(\overline{A}B+C)(\overline{A}BC+\overline{C})$$

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

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

$$= (A\overline{A} \cdot BC + AC)(\overline{A}BC+\overline{C})$$

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

$$= AC \cdot \overline{A}BC \stackrel{=0}{=} + \overline{C} AC \stackrel{=0}{=}$$

$$= 0 + 0$$

$$= 0$$

$$= RHS$$

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

$$\begin{aligned}
 Y &= (A+B) \overline{(\bar{B}+\bar{C})} + \bar{A}(B+C) & \overline{A} \overline{(\bar{B}+\bar{C})} &= \bar{A} + (\bar{\bar{B}} + \bar{\bar{C}}) \\
 &= (A+B)(A+BC) + \bar{A}(B+C) & &= A + \bar{B} \cdot \bar{C} \\
 &= AA + AB + ABC + BBC + \bar{A}(B+C) & &= A + BC \\
 &= A + AB + ABC + BC + \bar{A}(B+C) & 1 + B + BC &= 1 + BC \\
 &= AC(1+B+BC) + BC + \bar{A}(B+C) & &= 1 \\
 &= A + BC + \bar{A}(B+C) & \therefore (A + \bar{A}B = A + B) \\
 &= A + \bar{A}(B+C) + BC \\
 &= A + B + C + BC \\
 &= A + B + C(1+B) \\
 &= A + B + C
 \end{aligned}$$

Sum of product Method:

A product term is any group of literals that are ANDed together. A sum of product (SOP) is a group of product terms ORed together.

Four possible ways to AND 2 inputs if signals are in complemented & uncomplemented form. These ops are called fundamental products

		Fundamental Products
A	B	
0	0	$\bar{A}\bar{B}$
0	1	$\bar{A}B$
1	0	$A\bar{B}$
1	1	AB

The fundamental products are also called minterms. Products $\bar{A}\bar{B}$, $\bar{A}B$, $A\bar{B}$, AB are represented by m_0 , m_1 , m_2 and m_3 respectively.

Note: For n variable problems there can be 2^n number of minterms.

Sum of Product Equation:

Steps to get the sum of product solution:

1. Locate each o/p 1 in the truth table & write down the fundamental product
2. After identifying all the fundamental products, OR the fundamental products.

Eg:

SI	A	B	C	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1 $\rightarrow \bar{A}BC$
4	1	0	0	0
5	1	0	1	1 $\rightarrow A\bar{B}C$
6	1	1	0	1 $\rightarrow AB\bar{C}$
7	1	1	1	1 $\rightarrow ABC$

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

$$Y = F(A, B, C) = \sum m(3, 5, 6, 7)$$

' Σ ' Symbolizes summation or logical OR operation performed on corresponding minterms.

$Y = F(A, B, C)$ means Y is a function of 3 Boolean variables.

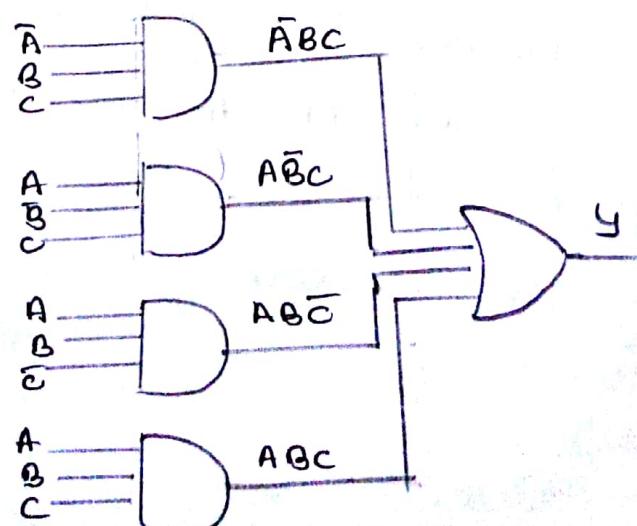
Representation of a truth table of this kind is also known as canonical sum form. ($Y = F(A, B, C)$)

Logic Circuit

For a sum of products eqn. we can derive the

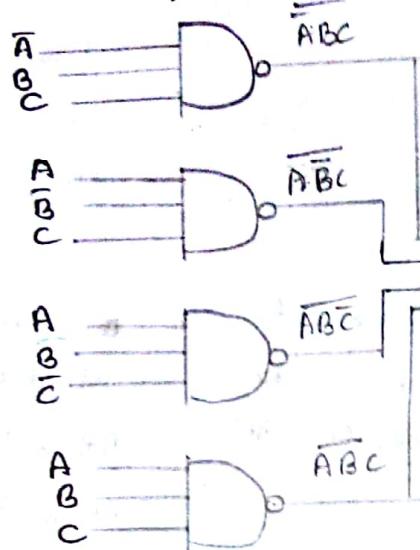
corresponding logic circuit by drawing an

AND-OR n/w.



(4)

We can't build the above circuit bcoz a 4 input OR gate is not available as TTL chip. But a 4 i/p NAND gate is available. So we have implemented using NAND-NAND n/w.



$$\overline{ABC} \cdot \overline{ABC} \cdot \overline{ABC} \cdot \overline{ABC} = \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC}$$

$$y = \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC}$$

Truth Table to Karnaugh Map:

A karnaugh map is a visual display of the fundamental products needed for a sum of products solution.

Eg: Consider the below truth table:

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

vertical column has \bar{A} followed by A
horizontal row has \bar{B} followed by B

		$\bar{B} B$		$\bar{B} B$		$\bar{B} B$	
		\bar{A}	B	\bar{A}	B	\bar{A}	B
0	\bar{A}	0	1	0	1	0	0
1	A	2	3	1	0	1	1

$$y = \sum m(2,3)$$

A 2 variable map contains $2^2 = 4$ cell.

Cell is the smallest unit k map, corresponding to 1 line of a truth table.

1 variable

map

A	\	0
		1

2 variable

map

A	\	B	0	1
			0	
			1	

3 variable

map

A	\	B	C	00	01	11	10
				0			
				1			

Available

map

AB	\	CD	00	01	11	10
			00			
			01			
			11			
			10			

\bar{A}	\	\bar{A}
A	\	A

A	\	\bar{B}	B
		$\bar{A}\bar{B}$	$\bar{A}B$
		A $\bar{A}\bar{B}$	AB

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

A	\	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
		$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	$A\bar{B}\bar{C}\bar{D}$
		$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}\bar{B}CD$	$A\bar{B}CD$
		AB $\bar{A}\bar{B}\bar{C}\bar{D}$	AB $\bar{A}\bar{B}C\bar{D}$	AB $\bar{A}\bar{B}CD$	AB $C\bar{D}$
		AB $\bar{A}\bar{B}\bar{C}\bar{D}$	AB $\bar{A}\bar{B}C\bar{D}$	AB $\bar{A}\bar{B}CD$	AB $C\bar{D}$

0	\	0
0	m ₀	m ₁
1	m ₁	

A	\	B
		0 m ₀ m ₁
		1 m ₂ m ₃

A	\	BC
		00 01 11 10
		0 m ₀ m ₁ m ₃ m ₂
A	\	BC
		00 01 11 10
		0 m ₀ m ₁ m ₃ m ₂

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

A	\	0
		1
		1

A	\	B	0	1
			0	1
			1	2 3

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

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			

Problems:

1) Plot $Y = AB\bar{C} + A\bar{B}C + \bar{A}\bar{B}C$ on the K map2) Plot $Y = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C\bar{D} + A\bar{B}\bar{C}D$ on the K map

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

$\bar{A}\bar{B}$	\	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
		0 0 0 0			
		1 0 0 1			
		0 1 0 0			
		0 0 1 1			

Entered Variable Map: is map in which one of the I/P variable is placed inside Karnaugh map. This reduces the K map size by one degree, i.e. a 3 variable prob. that requires 2³ locations in K map will require 2³⁻¹=4 location in entered variable map.

A	B	Y		\bar{B}	B		B	C	Y		\bar{C}	C
0	0	0		0	\bar{C}		0	0	0		0	0
0	1	\bar{C}		0	1		0	1	0		\bar{B}	(1)
1	0	0		0	1		1	0	1		B	A
1	1	1		1	1		1	1	A			

$y = AB + BC$

Entered variable map of the truth table.

Pairs, Quads, and Octets:

Pairs, Quads, and Octets are grouping technique in K map to simplify boolean function.

Pairs: Grouping of adjacent ones. (horizontally / vertically adjacent). Here we can eliminate the variable that changes form.

Ex:

$$\begin{aligned} 1) \quad Y &= \bar{A}B\bar{C} + \bar{A}BC \\ &= AB(\bar{C} + C) \\ &= \bar{A}B \end{aligned}$$

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

$= \bar{A}B$

$$\begin{aligned} 2) \quad Y &= ABCD + ABC\bar{D} \\ &= ABC(D + \bar{D}) \\ &= ABC \end{aligned}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
AB	0	0	0	0
AB	0	0	(1)	(1)
$A\bar{B}$	0	0	0	0

$$y = ABC$$

①

$$3) Y = \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}D \\ = \bar{B}\bar{C}(\bar{A}+A) \\ = \bar{B}\bar{C}D$$

	$\bar{C}D$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	0	0	0
$A\bar{B}$	0	1	0	0
AB	0	0	0	0

$$Y = \underline{\bar{B}\bar{C}D}$$

$$4) Y = \bar{A}\bar{B}C + \bar{A}BC + ABC \\ = \bar{A}\bar{B}C + \bar{A}BC + \bar{A}BC + ABC \\ [\because A+A=A] \\ = \bar{A}C(\bar{B}+B) + BC(\bar{A}+A) \\ = \underline{\bar{A}C + BC}$$

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

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

$$5) Y = A\bar{B}\bar{C} + A\bar{B}C \\ = A\bar{C}(\bar{B}+B) \\ = A\bar{C}$$

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

$$Y = \underline{A\bar{C}}$$

$$6) Y = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + AB\bar{C}\bar{D} + \\ AB\bar{C}D \\ = ABD(\bar{C}+C) + A\bar{C}\bar{D}(B+\bar{B}) \\ = \underline{\bar{A}BD + A\bar{C}\bar{D}}$$

	$\bar{C}D$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	1	1	0
$A\bar{B}$	1	0	0	0
AB	1	0	0	0

$$Y = \underline{\bar{A}BD + A\bar{C}\bar{D}}$$

Grouping of 4 adjacent ones is called quad. Elimination of 2 variables & their complements.

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

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

$$Y = A$$

	$\bar{C}D$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	0
$\bar{A}B$	0	0	1	0
$A\bar{B}$	0	0	1	0
AB	0	0	1	0

$$Y = CD$$

$$\begin{aligned}
 3) \quad Y &= \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \\
 &\quad A\bar{B}\bar{C}D + AB\bar{C}D \\
 &= \bar{A}BD(\bar{C}+C) + \\
 &\quad ABD(\bar{C}+C) \\
 &= \bar{A}BD + ABD \\
 &= BD(\bar{A}+A) = \underline{\underline{BD}}
 \end{aligned}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	1	1	0
$A\bar{B}$	0	1	1	0
AB	0	0	0	0

$$Y = \underline{\underline{BD}}$$

$$\begin{aligned}
 4) \quad Y &= A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + \\
 &\quad ABC\bar{D} + A\bar{B}CD \\
 &= A\bar{C}\bar{D}(B+\bar{B}) + AC\bar{D}(B+\bar{B}) \\
 &= A\bar{C}\bar{D} + AC\bar{D} \\
 &= A\bar{D}(\bar{C}+C) \\
 &= \underline{\underline{AD}}
 \end{aligned}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
$A\bar{B}$	1	0	0	1
AB	1	0	0	1

$$Y = \underline{\underline{AD}}$$

$$\begin{aligned}
 5) \quad Y &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}CD \\
 &= \bar{A}\bar{B}\bar{D}(\bar{C}+C) + A\bar{B}\bar{D}(\bar{C}+C) \\
 &= \bar{A}\bar{B}\bar{D} + A\bar{B}\bar{D} \\
 &= \bar{B}\bar{D}(\bar{A}+A) \\
 &= \underline{\underline{BD}}
 \end{aligned}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	0	0	1
$\bar{A}B$	0	0	0	0
$A\bar{B}$	0	0	0	0
AB	1	0	0	1

$$Y = \underline{\underline{BD}}$$

Octet: Grouping of 8 adjacent ones in K map. This eliminates 3 variables and their complements.

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0 ₀	0 ₁	0 ₃	0 ₂
$\bar{A}B$	4	1 ₅	4	1 ₆
$A\bar{B}$	1 ₁₂	1 ₁₃	1 ₁₅	1 ₁₄
AB	0 ₈	0 ₉	0 ₁₀	0 ₁₁

$$Y = B$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	0	0	0	0
$A\bar{B}$	0	0	0	0
AB	1	1	1	1

$$Y = \bar{B}$$

Overlapping Groups:

We are allowed to use the same 1 more than once.

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

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

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0	1	1	0
$\bar{A}B$	0	1	1	0
$A\bar{B}$	0	1	1	0
AB	0	1	1	0

$$Y = D$$

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

$$Y = \bar{D}$$

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

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	1	0	0
$A\bar{B}$	1	1	1	1
AB	1	1	1	1

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

Rolling the Map:

If possible, roll and overlap to get largest groups you can find.

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

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

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

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

(10)

Eliminating Redundant Group:

After you have finished encircling groups, eliminate any redundant group. This is a group whose 1's are already used by any other group.

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

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0 0	1 1	0 2	
$\bar{A}B$	1 1	1 1	0 0	
$A\bar{B}$	0 1	1 1	1 1	
AB	0 0	1 0	0 0	

Summary:

- 1) Enter a 1 on the Karnaugh-map method for each fundamental product that produces a 1 o/p in the truth table. Enter 0s elsewhere.
- 2) Encircle the octets, quads, and pairs. Remember to roll & overlap to get the largest groups possible.
- 3) If any isolated 1s remains, encircle each.
- 4) Eliminate any redundant group
- 5) Write the Boolean eqn by ORing the products corresponding to the encircled group.

Problems:

1) $Y = F(A, B, C, D) = \sum m(7, 9, 10, 11, 12, 13, 14, 15)$

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

$$Y = AB + AD + AC + BCD$$

$$2) Y = F(A, B, C, D) = \sum m(1, 2, 3, 6, 8, 9, 10, 12, 13, 14)$$

	$\bar{B}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}B$	0 ₀	1 ₁	1 ₃	1 ₂
$\bar{A}B$	0 ₄	0 ₅	0 ₇	1 ₆
AB	1 ₁₂	1 ₁₃	0 ₁₅	1 ₄
$A\bar{B}$	1 ₈	1 ₁	0 ₁₁	1 ₀

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

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

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

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

$$4) Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	1
$\bar{A}B$	1	1	0	0
$A\bar{B}$	1	1	0	0
AB	0	1	0	0

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

$$5) Y = \bar{A}\bar{B}C\bar{D} + ABC\bar{D} + A\bar{B}C\bar{D} + A\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}\bar{D} + \bar{A}\bar{B}CD$$

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

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	0	1	1
$\bar{A}B$	0	0	0	0
$A\bar{B}$	1	0	0	1
AB	1	0	1	1

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

$$6) Y = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}\bar{D} + ABCD + A\bar{B}CD$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	1	1
$A\bar{B}$	1	1	0	0
AB	0	0	1	0

eliminate
redundancy
 \Rightarrow

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	1	1
$A\bar{B}$	1	1	1	0
AB	0	0	1	0

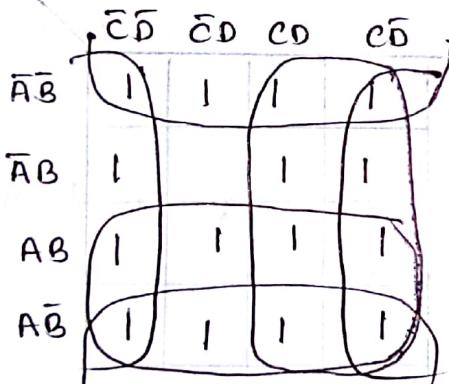
$$Y = AB\bar{C} + \bar{A}\bar{C}D$$

$$+ C\bar{A}B + ACD$$

(12)

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

$$\begin{aligned}
 Y &= \bar{A}BC(\bar{D}+\bar{D}) + AD(B+\bar{B})(C+\bar{C}) + B\bar{D}(A+\bar{A})(C+\bar{C}) + C\bar{D}(A+\bar{A}) \\
 &\quad (B+\bar{B}) + A\bar{C}(B+\bar{B})(D+\bar{D}) + \bar{A}\bar{B}(C+\bar{C})(D+\bar{D}) \\
 &= \bar{A}BCD + \bar{A}BC\bar{D} + AD(BC + B\bar{C} + \bar{B}C + \bar{B}\bar{C}) + B\bar{D}(AC + A\bar{C} + \bar{A}C + \bar{A}\bar{C}) \\
 &\quad + C\bar{D}(AB + A\bar{B} + \bar{A}B + \bar{A}\bar{B}) + A\bar{C}(BD + B\bar{D} + \bar{B}D + \bar{B}\bar{D}) \\
 &\quad + \bar{A}\bar{B}(CD + C\bar{D} + \bar{C}D + \bar{C}\bar{D}) \\
 &= \bar{A}BCD + \bar{A}BC\bar{D} + ABCD + AB\bar{C}D + A\bar{B}CD + A\bar{B}\bar{C}D + \\
 &\quad ABC\bar{D} + AB\bar{C}\bar{D} + \bar{A}BC\bar{D} + \bar{A}B\bar{C}\bar{D} + ABC\bar{D} + A\bar{B}CD + \\
 &\quad \bar{A}BC\bar{D} + \bar{A}\bar{B}C\bar{D} + AB\bar{C}D + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + \\
 &\quad \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}
 \end{aligned}$$



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

Simplification of Enternd Variable Map:

Consider below truth table.

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

Draw enterned variable map:

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

$$\therefore (\bar{C}+1=1)$$

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

Don't Care Condition:

An input-output condition that never occurs during normal operation. Since the condition never occurs, we can use an X on the truth table/k map. This X can be a 0 or a 1 whichever we prefer. The X is called a don't-care condition.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	X	X	X	X
$A\bar{B}$	0	1	X	X

$$Y = AD$$

→ don't form pair/quad
Wrong octet's only using X

I → the 1 is included in a quad.
the largest group we can find if we
visualize all X's as 1s

II → After the 1 has been enclosed all X's
outside the quad are visualized as
0's.

⇒ the X's are used to the best possible
advantage

A circuit designer is free to make the output for any
don't care condition either a '0' or a '1' in order to
produce the simplest o/p expression.

Summary:

- Given the truth table, draw a k map with 0s, 1s and don't care
- Encircle the actual 1s on the K map in the largest groups we
can find by treating the don't cares as 1s.
- After the actual 1s have been included in groups, discard
the remaining don't cares by visualizing them as 0s.

Simplify using K map and implement using logic gates.

$$1) F(A, B, C, D) = \sum m(1) + d(10, 11, 12, 13, 14, 15)$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
$A\bar{B}$	X	X	X	X
AB	0	0	X	X

$$Y = BCD$$

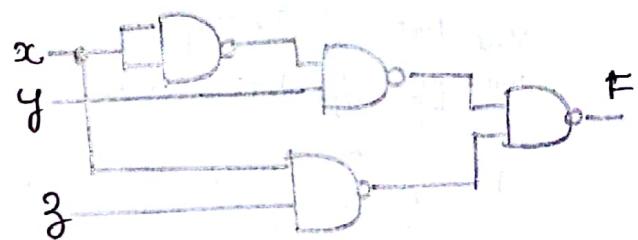
$$2) F(w, x, y, z) = w'xz + w'yz + x'yz' + wx'yz$$

$$d = wyz = w'xz(y+y') + w'yz(x+x') + x'yz'(w+w') + wx'yz$$

$$\bar{y}\bar{z} \quad \bar{y}z \quad y\bar{z} \quad yz = wyz(x+x')$$

	$\bar{y}\bar{z}$	$\bar{y}z$	$y\bar{z}$	yz
$w\bar{x}$	0	0	1	1
wx	0	1	1	0
wx	0	1	X	0
$w\bar{x}$	0	0	X	1

$$Y = xz + \bar{x}y$$



$$3) F(A, B, CD) = \sum m(6, 7, 9, 10, 13) + d(1, 4, 5, 11)$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0	X	0	0
$\bar{A}B$	X	X	1	1
$A\bar{B}$	0	1	0	0
AB	0	1	X	0
$\bar{A}B$	0	1	X	1

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

$$4) F(P, Q, R, S) = \sum m(0, 1, 4, 8, 9, 10) + d(2, 11)$$

	$\bar{R}\bar{S}$	$\bar{R}S$	RS	$R\bar{S}$
$\bar{P}\bar{Q}$	1	1	X	0
$P\bar{Q}$	1	0	0	0
PQ	0	1	0	0
$P\bar{Q}$	0	1	X	1
PQ	1	1	X	1
$\bar{P}Q$	1	1	X	1

$$Y = P\bar{Q} + \bar{Q}\bar{R} + \bar{P}\bar{R}\bar{S}$$

$$y = f(A, B, C, D) = \sum m(0, 1, 2, 4, 5, 12, 14) + d(8, 10)$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	1, 0	1, 0	0, 1	1, 0
$\bar{A}B$	1, 4	1, 5	0, 7	0, 6
$A\bar{B}$	1, 0	0, 13	0, 15	1, 14
AB	0, 9	0, 11	X, 10	

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

Entered Variable Map:

Rules for entering values in Entered Variable Map

MEV - Map entered variable.

Rule No	MEV if if op. MEV Map	Entry in Entered Variable Map	Comments:
1	0, 0	0	if functions equal 0 for both values of MEV, enter 0 in appropriate cell in MEV map
2	0, 1	1	if $f = 1$ for both values of MEV, enter 1
3	0, 0	MEV	if $f = \text{MEV}$ values, enter MEV
4	0, 1	$\bar{\text{MEV}}$	if $f = \text{complement of MEV}$ enter $\bar{\text{MEV}}$
5	0, -	-	if $f = \text{don't care}$ for both values of MEV enter -
6	0, -	0	if $f = 0$ for $\text{MEV} = 0$ & $f = 1$ for $\text{MEV} = 1$ enter 0
7	0, 0	0	if $f = 0$ and $f = -$ enter $f = 0$ in MEV
8	0, -	1	if $f = -$ and $f = 1$ enter $f = 1$ in MEV
9	0, 1	1	if $f = 1$ and $f = -$ enter $f = 1$ in MEV

$$Y = F(A, B, C, D) = \{0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14\}$$

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

When $D=0$: enter y values into K map

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

When $D=1$: enter y values into K map

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

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

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

Product of Sums Method:

A product of sums is any group of sum terms ANDed together

Given truth table, identify the fundamental sums needed for a logic design. Then by ANDing these sums we get the product of sums equation. But with the product of sums method, the fundamental sum produces an output 0 for corresponding I/P condition.

Converting a Truth Table to an Equation:

- Locate each output 0 in the truth table and write down its fundamental sum.
- Each variable is complemented when corresponding I/P variable is 1; the variable is uncomplemented when the

Corresponding SOP variable is 0.

3. All fundamental sum terms are ANDed together

In Product of Sum (POS) each sum term is called maxterm and is designated by M_i ;

Consider the below truth Table

A	B	C	y	Maxterm
0	0	0	0	$A+B+C$
0	0	1	1	M_0
0	1	0	1	M_1
0	1	1	0	M_2
1	0	0	1	M_3
1	0	1	1	M_4
1	1	0	0	M_5
1	1	1	1	M_6
			1	M_7

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

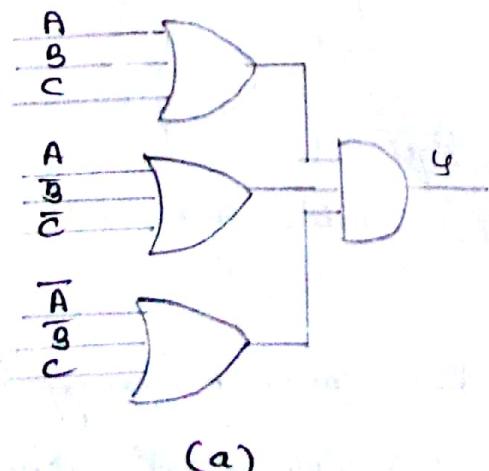
$$Y = F(A, B, C) = \prod M(0, 3, 6)$$

where \prod symbolizes product i.e AND operation

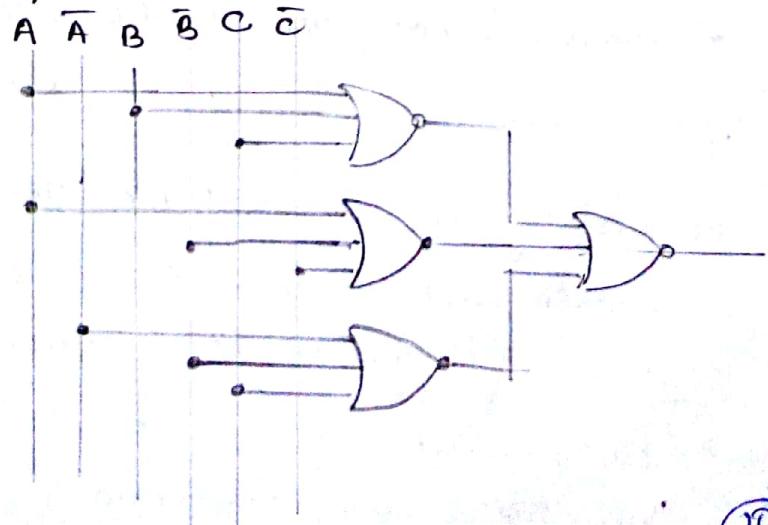
This kind of representation of a truth table is also known as canonical product form.

Logic circuit

We can get the logic circuit by drawing an OR- AND n/w or a NOR-NOR n/w. A 3-input OR gate is not available as a TTL chip. So circuit (a) is not practical.



(a)



(b)

Conversion between SOP and POS

SOP representation is obtained by considering 1s in a truth table while POS come considering 0s. Thus SOP and POS occupy complementary locations in a truth table and one representation can be obtained from the other by

- * identify complementary locations
- * changing minterms to maxterms or inverse and finally
- * changing summation by product or inverse

$$\text{Eg: 1) } Y = F(A, B, C) = \prod M(0, 3, 6) \\ = \sum m(1, 2, 4, 5, 7)$$

$$2) Y = F(A, B, C) = \sum m(3, 5, 6, 7) \\ = \prod M(0, 1, 2, 4)$$

This is known as conversion b/w canonical forms.

Product of Sum Simplification

In POS expression we assign maxterms to the cell of a Karnaugh map.

A boolean expression in the product of sums can be plotted on the Kmap by placing a 0 in each cell corresponding to a term in the expression. Remaining cells are filled with ones.

		B	\bar{B}
		A	\bar{A}
A	A_0	$A+B$	$A+\bar{B}$
\bar{A}	\bar{A}_1	$\bar{A}+B$	$\bar{A}+\bar{B}$

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

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

	B	\bar{B}
A	M_0	M_1
\bar{A}	M_2	M_3

	$B+C$	$B+\bar{C}$	$\bar{B}+C$	$\bar{B}+\bar{C}$	$C+\bar{D}$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+\bar{D}$
A	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
\bar{A}	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}

{ Steps in POS simplification:

- 1) Plot the K map and place Os in those cell corresponding Os in the truth table / maxterm in the POS expression & mark all with 1s.
- 2) Check & Encircle the octets, quads and pairs of adjacent Os. Remember to roll & overlap to get the largest groups possible.
- 3) If any isolated Os remains, encircle each.
- 4) Eliminate any redundant group.
- 5) Write the boolean expression by ANDing the sums corresponding to the encircled groups

Problems:

- 1) Minimize the expression.

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

$B+C$ $B+\bar{C}$ $\bar{B}+\bar{C}$ $\bar{B}+C$

A	0	1	1	0
\bar{A}	1	0	0	1
	4	5	6	7
	8	9	10	11
	12	13	14	15

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

$$2) Y = (\bar{A}+\bar{B}+C+D) (\bar{A}+\bar{B}+\bar{C}+D) (\bar{A}+\bar{B}+\bar{C}+\bar{D}) (A+\bar{B}+C+D) (A+\bar{B}+\bar{C}+D)$$

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

$C+D$ $C+\bar{D}$ $\bar{C}+\bar{D}$ $\bar{C}+\bar{D}$

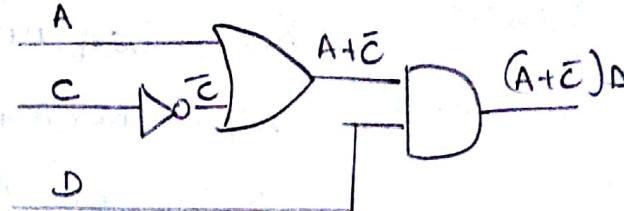
$A+B$	0	1	1	1	0
$A+\bar{B}$	1	1	0	0	1
$\bar{A}+\bar{B}$	0	0	1	1	0
$\bar{A}+B$	0	1	1	0	1
	8	9	10	11	12

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

$$3) f(A, B, C, D) = \pi M(0, 3, 4, 7, 8, 10, 12, 14) + d(2, 6)$$

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0 ₀	1 ₁	0 ₃	X ₂
$A+\bar{B}$	0 ₄	1 ₅	0 ₇	X ₆
$\bar{A}+\bar{B}$	0 ₁₂	1 ₁₃	1 ₁₅	0 ₁₄
$\bar{A}+B$	0 ₈	1 ₉	1 ₁₁	0 ₁₀

$$y = (C+D)(\bar{C}+D)(A+\bar{C})$$



$$4) f(A, B, C, D) = \pi M(0, 2, 4, 10, 11, 14, 15)$$

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0 ₀	1 ₁	1 ₃	0 ₂
$A+\bar{B}$	0 ₄	1 ₅	1 ₇	1 ₆
$\bar{A}+\bar{B}$	1 ₁₂	1 ₁₃	0 ₁₅	0 ₁₄
$\bar{A}+B$	1 ₈	1 ₉	0 ₁₀	0 ₁₁

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

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

$$5) f(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 5, 7)$$

$$= \pi M(6, 8, 9, 10, 11, 12, 13, 14, 15)$$

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	1 ₀	1 ₁	1 ₃	1 ₂
$A+\bar{B}$	1 ₄	1 ₅	1 ₇	0 ₆
$\bar{A}+\bar{B}$	0 ₁₂	0 ₁₃	0 ₁₅	0 ₁₄
$\bar{A}+B$	0 ₈	0 ₉	0 ₁₁	0 ₁₀

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

$$6) f(a, b, c, d) = \sum m(6, 7, 9, 10, 13) + d(1, 4, 5, 11)$$

$$= \pi M(0, 2, 3, 8, 12, 14, 15) + d(1, 4, 5, 11)$$

	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0 ₀	X ₁	0 ₃	0 ₂
$A+\bar{B}$	X ₄	X ₅	1 ₇	1 ₆
$\bar{A}+\bar{B}$	0 ₁₂	1 ₁₃	0 ₁₅	0 ₁₄
$\bar{A}+B$	0 ₈	1 ₉	X ₁₁	1 ₁₀

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

7) $y = A + BC\bar{D}$ using POS & SOP circuit

$$\begin{aligned}
 y = A + BC\bar{D} &= A(B + \bar{B})(C + \bar{C})(D + \bar{D}) + (\bar{A} + \bar{A})BC\bar{D} \\
 &= (AB + A\bar{B})(C + \bar{C})(D + \bar{D}) + ABC\bar{D} + \bar{A}BC\bar{D} \\
 &= (ABC + A\bar{B}\bar{C} + AB\bar{C} + A\bar{B}\bar{C})(D + \bar{D}) + ABC\bar{D} + \bar{A}BC\bar{D} \\
 &= ABCD + ABC\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + ABC\bar{D} + A\bar{B}\bar{C}D \\
 &\quad A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + ABC\bar{D} + \bar{A}B\bar{C}\bar{D}
 \end{aligned}$$

SOP :

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	1
$A\bar{B}$	1	1	1	1
AB	1	1	1	1

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

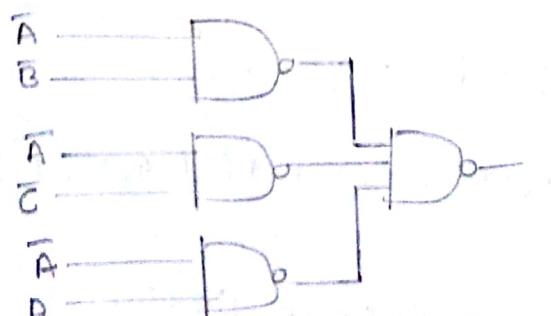
$$y = A\bar{B}C\bar{D}$$

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

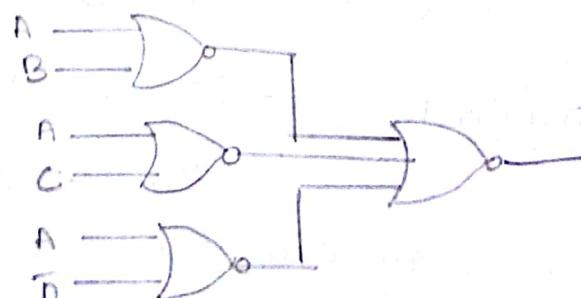
Basic gates



NAND - NAND



NOR - NOR



Given a logic circuit, we can find its dual circuit as follows.

1. Change each AND gate to an OR gate & vice versa & complement all inputs and outputs
2. Change each NAND gate to a NOR gate & vice versa & complement all inputs & outputs.

3) Change all NAND gates to NOR gate & complement all i_{ip} and o_{op} signal to get NOR-NOR circuit from NAND-NAND circuit.

Simplification By Quine-McClusky Method:

K map simplification method becomes complicated for 5 or more variables. Quine McClusky method is a systematic approach for logic simplification that does not have these limitations and can easily be implemented in a digital computer.

Quine-McClusky method involves preparation of 2 tables.

- determines prime implicants
- selects essential prime implicants to get minimal expression.

Prime implicants are expressions with least no. of literals that represents all the terms given in a truth table.

Prime implicants are examined to get essential prime implicants for a particular expression that avoids any type of duplication.

Generating prime implicants

1) Find out all the terms that gives output 1 from truth table / SOP equation

2) List all minterms in binary form.

3) Arrange the minterms in different groups depending on how many 1's in variable combinations have.

C.i.e 1st group has no 1 in 1st combination

2nd group has only one, 3rd two ls, 4th three ls

5th four ls etc...)

4) Compare each binary number with every term in the adjacent next higher category & if any differ only by

one position, put a check mark and copy the term in the next column with '1' in the position that they differed
 5) Apply the same process described in previous step for the resultant column and continue these cycles until a single pass through cycle yields no further elimination of literals.

Selection of prime Implicants

1. List all prime implicants
2. Select the minimum number of prime implicants which must cover all the minterms.

$$\text{Problem: } F(A, B, C, D) = \sum m(0, 2, 3, 6, 7, 8, 10, 12, 13)$$

stage: 0	1	2	3
ABCD	ABCD	ABCD	A B C D
0000 (0)	0000 (0) ✓	00-0 - 000	0-0 (0, 2, 8, 10)
0010 (2)	0010 (2) ✓	- 000 (0, 8) ✓	0-0 (0, 8, 2, 10)
0011 (3) ✓	1000 (8) ✓	001- (2, 3) ✓	0-1 (2, 3, 6, 7) ✗
0110 (6) -	0010 (3) ✓	0-10 (2, 6) ✓	
0111 (7)	0110 (6) ✓	- 010 (2, 10) ✓	
1000 (8) ✓	1010 (10) ✓	10-0 (8, 10) ✓	
1010 (10) ✓	1100 (12) ✓	1-00 (8, 12) ✓	
1100 (12) ✓	0111 (7) ✓	0-11 (8, 7) ✓	
1101 (13) ✓	1101 (13) ✓	011- (6, 7) ✓	
		110- (12, 13) ✓	

Prime implicants: List

Prime implicant	m_0	m_2	m_3	m_6	m_7	m_8	m_{10}	m_{12}	m_{13}
$A\bar{C}D$ (8, 12)						✓	✓		
$A\bar{B}\bar{C}$ (12, 13)								✓	✓
$\bar{B}\bar{D}$ (0, 2, 8, 10)	✓	✓					✓	✓	
$\bar{A}\bar{C}$ (2, 3, 6, 7)		✓	✓	✓	✓	✓			

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

$$Y = F(A, B, C, D) = \sum m(0, 1, 2, 3, 10, 12, 11, 13, 14, 15)$$

A B C D	A B C D	A B C D	A B C D
0 0 0 0 (0)	0 0 0 0 (0)	0 0 0 - (0, 1) ✓	0 0 - - (0, 2, 1, 3)
0 0 0 1 (1)	0 0 0 1 (1)	0 0 - 0 (0, 2) ✓	0 0 - - (0, 1, 2, 3)
0 0 1 0 (2)	0 0 1 0 (2)	0 0 - 1 (1, 3) ✓	- 0 1 - (2, 3, 10, 11)
0 0 1 1 (3)	0 0 1 1 (3)	0 0 1 - (2, 3) ✓	- 0 1 - (2, 10, 3, 11)
0 0 1 0 (10)	1 0 1 0 (10)	- 0 1 0 (2, 10) ✓	1 - 1 - (10, 11, 4, 15)
1 1 0 0 (12)	1 1 0 0 (12)	- 0 1 1 (3, 11) ✓	1 - 1 - (10, 14, 11, 15)
1 0 1 1 (11)	1 0 1 1 (11)	1 - 1 0 (10, 11) ✓	1 1 - - (12, 13, 14, 15)
1 1 0 1 (13)	1 1 0 1 (13)	1 1 0 - (10, 13) ✓	1 1 - - (12, 14, 13, 15)
1 1 1 0 (14)	1 1 1 0 (14)	1 1 - 0 (12, 14) ✓	
1 1 1 1 (15)	1 1 1 1 (15)	1 - 1 1 (11, 15) ✓	
		1 1 - 1 (13, 15) ✓	
		1 1 1 - (14, 15) ✓	

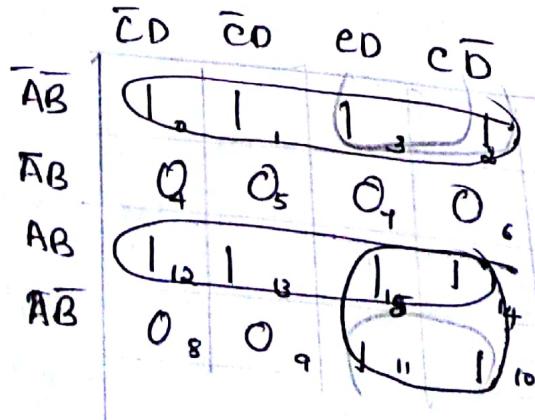
$\bar{A}\bar{B}$ (0, 2, 13)
 $\bar{B}C$ (2, 3, 10, 11)
 AC (10, 11, 14, 15)
 AB (12, 13, 14, 15)

0 1 2 3 10 11 12 13 14 15

$$Y = AB + \bar{A}\bar{B} + \bar{B}C \quad \text{or}$$

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

By K map:



$$\begin{aligned}
 Y &= \bar{A}\bar{B} + AB + A\bar{B} \quad \text{or} \\
 Y &= \bar{A}\bar{B} + AB + \bar{B}C
 \end{aligned}$$

$$Y = F(A, B, C, D) = \sum m(1, 2, 8, 9, 10, 12, 13, 14)$$

A B C D	A B C D	A B C D	A B C D
0 0 0 1 (1)	0 0 0 1 (1)	0 0 1 0 (2)	1 - 0 - (8, 12, 9, 13)
0 0 1 0 (3)	0 0 1 0 (2)	0 1 0 0 (8)	1 - 0 0 (8, 12, 10, 14)
1 0 0 0 (8)	1 0 0 0 (8)	1 - 0 0 (8, 12)	
1 0 0 1 (9)	1 0 0 1 (9)	1 - 0 1 (9, 13)	
1 0 1 0 (10)	1 0 1 0 (10)	1 - 1 0 (10, 14)	
1 1 0 0 (12)	1 1 0 0 (12)	1 1 0 - (12, 13)	
1 1 0 1 (13)	1 1 0 1 (13)	1 1 - 0 (12, 14)	
1 1 1 0 (14)	1 1 1 0 (14)		

Prime Implicants:

	1	2	8	9	10	12	13	14
$\bar{B}\bar{C}D$	(1, 9)	✓		✓				
$\bar{B}C\bar{D}$	(2, 10)		✓		✓			
$A\bar{C}$	(8, 12, 9, 13)			✓	✓	✓	✓	
$A\bar{D}$	(8, 12, 13, 14)			✓	✓	✓	✓	

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

Hazards and Hazard Covers:

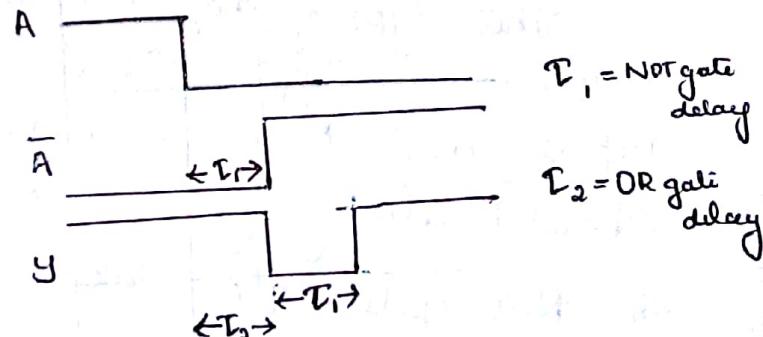
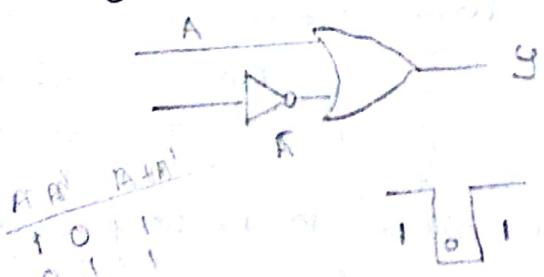
Various simplification techniques give minimal expression for a logic equation which in turn requires minimum h/w for realization. But due to some practical problems in certain cases we may prefer to include more terms than given by simplification techniques.

So far we have considered that gate generates o/p's instantaneously. But practical circuit always offer finite propagation delay though very small in nanosecond order. This give rise to several hazards & hazard covers additional terms for an eqn that prevents occurring of them.

Static 1 Hazard:

In a combinational circuit, if o/p goes momentarily 0 when it should remain 1, the hazard is known as static 1 hazard.

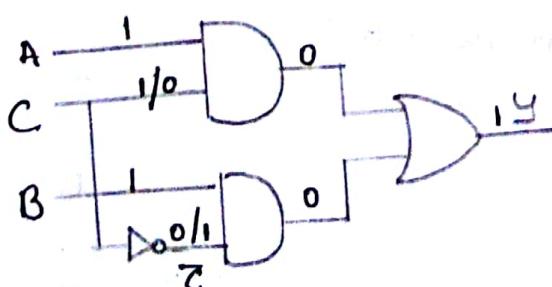
This type of hazard occurs when $Y = A + A'$ type of situation appears for a logic circuit for certain combination of other i/p's and A makes a transition $1 \rightarrow 0$. An $A + A'$ condition should always generate 1 at the o/p i.e. static 1. But the NOT gate o/p takes finite time to become 1 following $1 \rightarrow 0$ transition of A. Thus for the OR gate there are 2 zeros appearing at its i/p for that small duration, resulting a 0 at its o/p. The width of this zero is in nanosecond order and its called a glitch.



Consider the below K-map and minimize (simplify & draw circuit using basic gates)

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

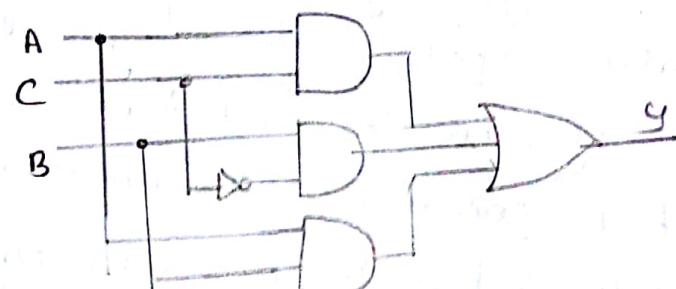
$$Y = B\bar{C} + AC$$



For this circuit input $B=1$ and $A=1$ and then C makes transition $1 \rightarrow 0$. The o/p shows a glitch. (i.e. when $B=1$ & $A=1$ C is $1 \rightarrow 0$, Y should be 1 but it will 0 for few nanoseconds becoz of delay in NOT gate)

If we consider another grouping for the same map. This includes 1 additional term AB and o/p $y = BC' + AC + AB$. This circuit requires more time than minimal representation is hazard free.

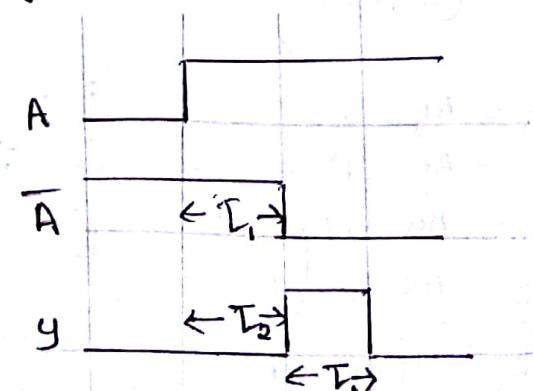
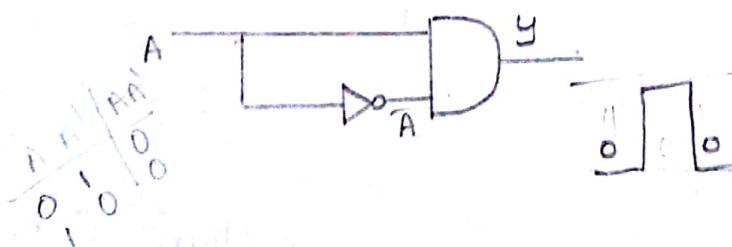
	\bar{C}	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	1	0
AB	1	1
$A\bar{B}$	0	1



Hazard free circuit

Static-0 Hazard: In a combinatorial circuit, if o/p goes momentarily 1 when it should remain 0, the hazard is known as static-0 hazard.

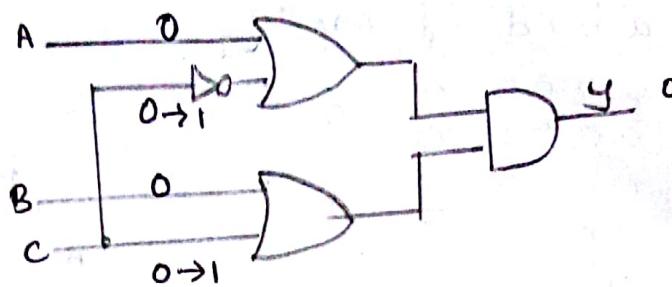
This type of hazard occurs when $y = AA'$. kind of situations occurs in a logic circuit for certain combination of other inputs and A makes a transition $0 \rightarrow 1$. AA' condition should always generate 0 at the o/p (i.e. static-0). But the NOT gate o/p takes similar time to become 0 following a $0 \rightarrow 1$ transition of A. Thus for final AND gate there are 2 ones appearing at o/p for a small duration resulting a 1 at its o/p.



Consider following K map and corresponding logic circuit $y = (B+C)(A+\bar{C})$. But if $B=0$, $A=0$ and C makes transition from $0 \rightarrow 1$ there will be static 0 hazard at o/p.

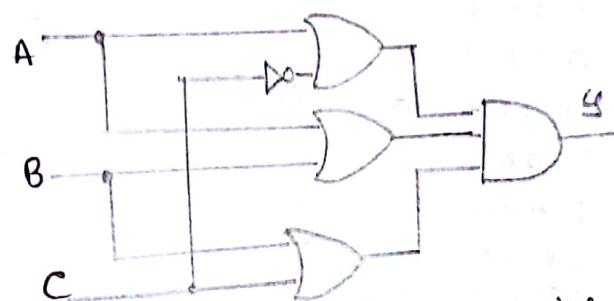
To avoid that one more sum term is added to POS form & corresponding logic circuit is given below.

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



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

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



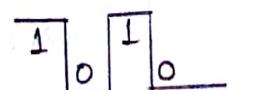
hazard free circuit

Dynamic Hazard: occurs when circuit o/p makes multiple transitions before it settles to a final value while the logic eqn. asks for only one transition.

"A dynamic hazard is the possibility of an o/p changing more than once as a result of a single i/p change"

Dynamic hazards often occur in larger logic circuit where are different routes to the o/p (from the i/p)

eg: A logic circuit is meant to change o/p state from 1 to 0, but instead changes from 1 to 0 then 1 and finally just at the correct value '0'. This is a dynamic hazard.



Dynamic hazard take a complex method to

remove
If all static hazards have been eliminated from a circuit then dynamic hazard can't occur.

Solve using entired variable map

$$F(A, B, C, D) = \Sigma(2, 3, 4, 5, 13, 15) + \Sigma_d(8, 9, 10, 11)$$

M _i	a	b	c	d	f	Map Entry
0	0	0	0	0	0	
	1	0	0	1	0	
1	2	0	0	1	1	
	3	0	0	1	1	
2	4	0	1	0	1	
	5	0	1	0	1	
3	6	0	1	1	0	
	7	0	1	1	0	0
4	8	1	0	0	-	
	9	1	0	0	-	-
5	10	1	0	1	-	-
	11	1	0	1	-	-
6	12	1	1	0	0	
	13	1	1	0	1	
7	14	1	1	0	0	d
	15	1	1	1	1	d

$$\bar{B}\bar{C} \quad \bar{B}C \quad BC \quad B\bar{C}$$

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

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

$$Y = F(A, B, CD) = \Sigma_m(3, 4, 5, 7, 8, 11, 12, 13, 15)$$

M _i	A	B	C	D	f
0	0	0	0	0	0
	1	0	0	1	0
1	2	0	0	0	0
	3	0	0	1	1
2	4	0	1	0	1
	5	0	1	0	1
3	6	0	1	0	0
	7	0	1	1	1
4	8	1	0	0	1
	9	1	0	0	0
5	10	1	0	1	0
	11	1	0	1	1
6	12	1	1	0	1
	13	1	1	0	1
7	14	1	1	0	0
	15	1	1	1	1

$$\bar{B}\bar{C} \quad \bar{B}C \quad BC \quad B\bar{C}$$

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

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

HDL Implementation Model.

There are different ways of describing combinational circuit in Verilog.

1) Dataflow Modeling

2) Behavioural Modeling

3) Structural/Gate level Modeling.

Dataflow Modeling:

Data flow describes how the signals flow from the I/p's to the O/p's.

'assign' is the keyword used to assign a value to the left hand side of a signal assignment statement.

Verilog Operator List

Relational Operators : $<$, \leq , $>$, \geq , \neq

Logical Operation : $!$, $\&\&$, $\|$

Bitwise Operators : \sim , $\&$, \mid , \wedge

Arithmetical Operators : $+$, $-$, $*$, $/$

Ex: i) $Y = (A+B) \cdot (C+D)$

```
module testexp (A,B,C,D,Y);
```

```
input A,B,C,D;
```

```
output Y;
```

```
assign Y = (A|B)&(C|D);
```

```
endmodule
```

ii) $Y = \overline{C} + \overline{A}\overline{D} + \overline{B}\overline{D}$

```
module testexpl (A,B,C,D,Y)
```

```
input A,B,C,D;
```

```
output Y;
```

```
assign Y =  $\sim C | (\sim A \& \sim D) | (\sim B \& \sim D);$ 
```

endmodule.

Behavioural Modeling:

In this model statements are executed sequentially following arithmetic disruption. Ideally suited for sequential circuit.

Keyword always is executed only if any variable within the sensitivity list changes its value.

Procedure assignment / output variable within always must be of register type defined by reg keyword.

Ex: $y = AB + CD$ ($y=1$ if $AB=11$ or if $CD=11$ else $y=0$)

module test (A, B, C, D, Y);

input A, B, C, D;

output Y;

reg Y;

always @ (A or B or C or D)

if ((A==1) && (B==1))

Y=1;

else if ((C==1) && (D==1))

Y=1;

else

Y=0;

endmodule.

Structural / Gate level Modeling:

VHDL uses components or gates to model the system.

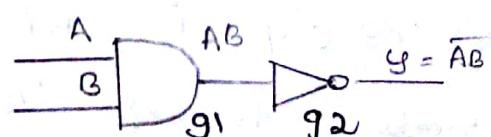
module circuit (P, Q, R);

input P, Q;

output R;

with g;

and g1(s, P, Q);



```
not g2(y, s);  
endmodule.
```

Write Verilog code using

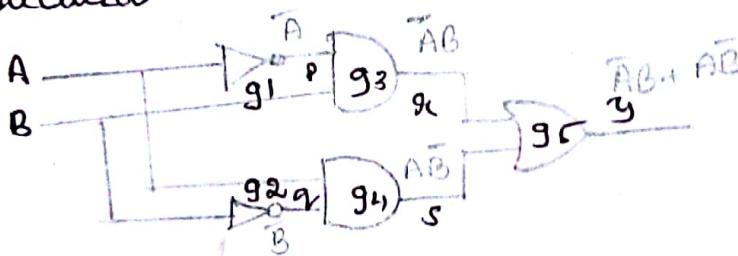
- i) Structural model
- ii) Dataflow model
- iii) Behavioural model.

for a) $y = \bar{A}\bar{B} + AB$

b) $y = \bar{A}B + A\bar{B}$

b) $y = \bar{A}B + A\bar{B}$

i) Structural model



```
module prob(A, B, y);
```

```
input A, B;
```

```
output y;
```

```
wire p, q, r, s;
```

```
not g1(p, A);
```

```
not g2(q, B);
```

```
and g3(r, p, B);
```

```
and g4(s, q, A);
```

```
or g5(y, r, s);
```

```
endmodule.
```

ii) Dataflow Model

```
module prob1(A, B, y);
```

```
input A, B;
```

```
output y;
```

```
;
```

assign $y = (\bar{A} \cdot \bar{B}) \mid (A \cdot \bar{B})$;

endmodule

Behaviour Model

```
module prob2( A,B,Y )
    input A,B;
    output Y;
    reg Y;
    always @ ( A or B )
        if((A==0)&&(B==1))
            Y=1;
        else if ((A==1)&&(B==0))
            Y=1;
        else
            Y=0;
endmodule.
```

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

Problems:

- 1) Using Q-M method simplify following expression
- $F(A, B, C, D) = \sum_m(0, 3, 5, 6, 7, 11, 14)$
 - $F(A, B, C, D) = \sum_m(0, 1, 2, 3, 10, 11, 12, 13, 14, 15)$

Problems:

- 1) Using K Map simplify the following expression:
- $F(A, B, C, D) = \sum_m(0) + d(10, 11, 12, 13, 14, 15)$
 - $F(A, B, C, D) = \sum_m(9) + d(10, 11, 12, 13, 14, 15)$

Product Of Sum Simplification:

$$Y = F(A, B, C, D) = \sum m(0, 1, 2, 3, 10, 11, 12, 13, 14, 15)$$

Step 8:

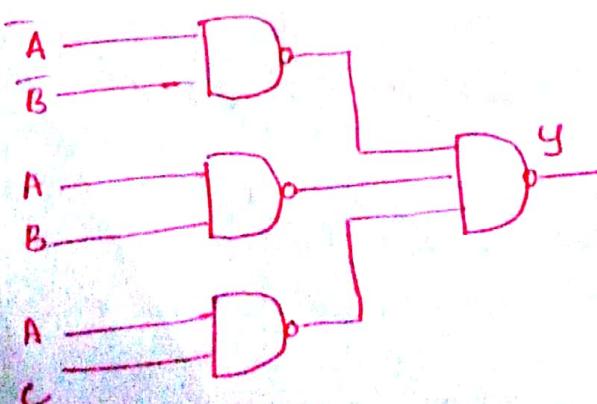
- 1) Convert the truth table into a K map. Group the 1's and write SOP equations and draw NAND-NAND circuit
- 2) Complement the K map. Group the 1's, write SOP equations and draw NAND-NAND circuit for \bar{Y}
- 3) Convert the complementary NAND-NAND circuit to dual NOR-NOR circuit by changing all NAND gates to NOR gates and complementing all signals
- 4) Compare the NAND-NAND circuit with NOR-NOR circuit with final gate is used for implementation.

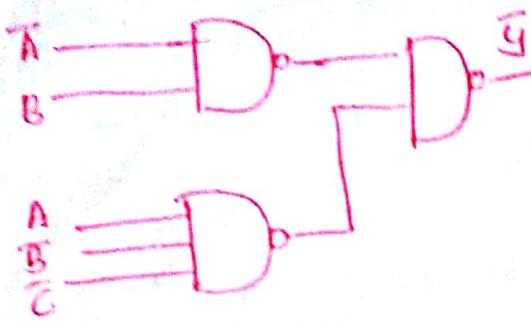
	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	1	1	1	1
AB	0	0	0	0
$A\bar{B}$	1	1	1	1
$\bar{A}B$	0	0	1	1

$$y = \bar{A}\bar{B} + AB + AC$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	1	1	1
$A\bar{B}$	0	0	0	0
$A\bar{B}$	1	1	0	0

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





$$\begin{aligned}
 & \overline{A+B} \\
 & \overline{B} \rightarrow D \rightarrow Y \\
 & \overline{A} \rightarrow D \rightarrow \overline{\overline{A+B+C}} \\
 & \overline{B} \rightarrow D \rightarrow \overline{\overline{A+B+C}} \\
 & \overline{C} \rightarrow D \rightarrow \overline{\overline{A+B+C}} \\
 & Y = (\overline{A+B}) + (\overline{\overline{A}} + \overline{B} + \overline{C}) \\
 & = (\overline{A+B}) \cdot (\overline{\overline{A}} + \overline{B} + \overline{C}) \\
 & = \underline{(\overline{A+B}) \cdot (\overline{\overline{A}} + \overline{B} + \overline{C})}
 \end{aligned}$$

Problems:

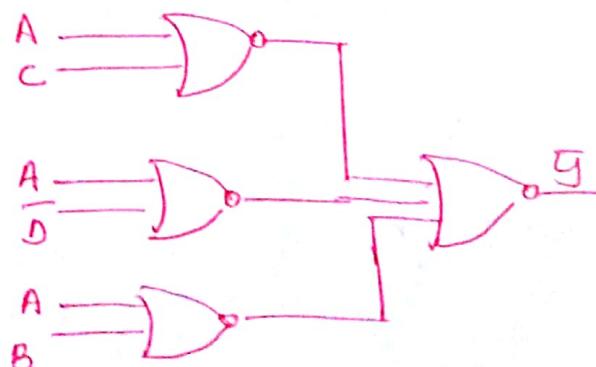
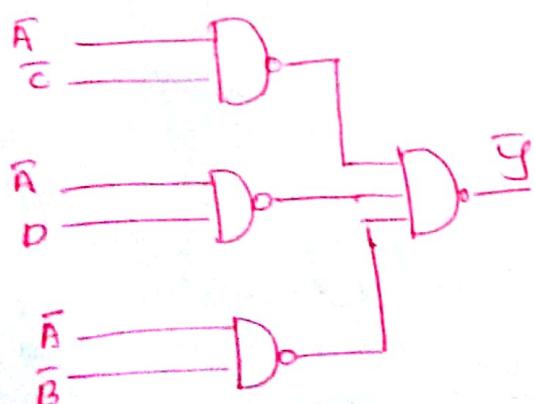
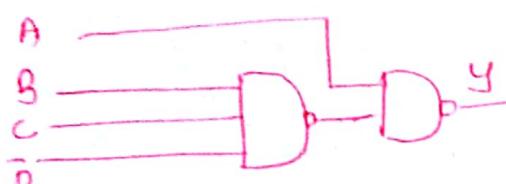
$$Y = F(A, B, C, D) = \sum m(6, 8, 9, 10, 11, 12, 13, 14, 15)$$

	\overline{CD}	$\overline{C}D$	$C\overline{D}$	CD
\overline{AB}	1			
$\overline{A}B$		1		
AB	1	1	1	1
$A\overline{B}$	1	1	1	1

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

	\overline{CD}	$\overline{C}D$	$C\overline{D}$	CD
\overline{AB}	1	1	1	1
$\overline{A}B$	1	1	1	0
AB	0	0	0	0
$A\overline{B}$	0	0	0	0

$$\overline{Y} = \overline{AB} \cdot \overline{A}\overline{C} + \overline{A}\overline{D} + \overline{A}\overline{B}$$



Quine-McClusky Method & Don't Care Condition:

$$1) F(A, B, C, D) = \sum m(2, 3, 10, 11, 12, 13, 14, 15) + d(0, 1)$$

0 1 2 3 10 11 12 13 14 15
 0000 0001 0010 0011 1010 1011 1100 1101 1110 1111

A B C D	A B C D	A B C D
0 0 0 0 (0) ✓	0 0 0 - (0,1) ✓	0 0 -- (0,1,2,3)
0 0 0 1 (1) ✓	0 0 - 0 (0,2) ✓	0 0 - - (0,2,1,3)
0 0 1 0 (2) ✓	0 0 - 1 (1,3) ✓	- 0 1 - (2,3,10,11)
0 0 1 1 (3) ✓	0 0 1 - (2,3) ✓	- 0 1 - (2,10,3,11)
1 0 1 0 (10) ✓	- 0 1 0 (2,10) ✓	1 - 1 - (10,14,11,15)
1 1 0 0 (12) ✓	- 0 1 1 (3,11) ✓	1 - 1 - (10,11,14,15)
1 0 1 1 (11) ✓	1 0 1 - (10,11) ✓	1 1 - - (12,13,14,15)
1 1 0 1 (13) ✓	1 - 1 0 (10,14) ✓	1 1 - - (12,14,13,15)
1 1 1 0 (14) ✓	1 - 1 0 - (12,13) ✓	
1 1 1 1 (15) ✓	1 1 - 0 (12,14) ✓	
	1 - 1 1 (11,15) ✓	
	1 1 - 1 (13,15) ✓	
	1 1 1 - (14,15) ✓	

Prime Implicants: $\bar{A}\bar{B}$, $\bar{B}C$, AC , AB

ABCD	2 3 10 11 12 13 14 15	0 1
00--	$\bar{A}\bar{B}$ (0,1,2,3)	✓
-01-	$\bar{B}C$ (2,3,10,11)	✓
1-1-	AC (10,11,14,15)	✓
11--	AB (12,13,14,15)	✓

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

Module II: ADE

Introduction:

Analog and digital circuits are used to transmit and process the information like sound, light from an environment.

Analog electronic circuit:

- * used with small signal
- * includes analog signals which are continuously changeable.

- * linear in nature
- * Analog signals represent infinite set of values.

Digital electronic circuit:

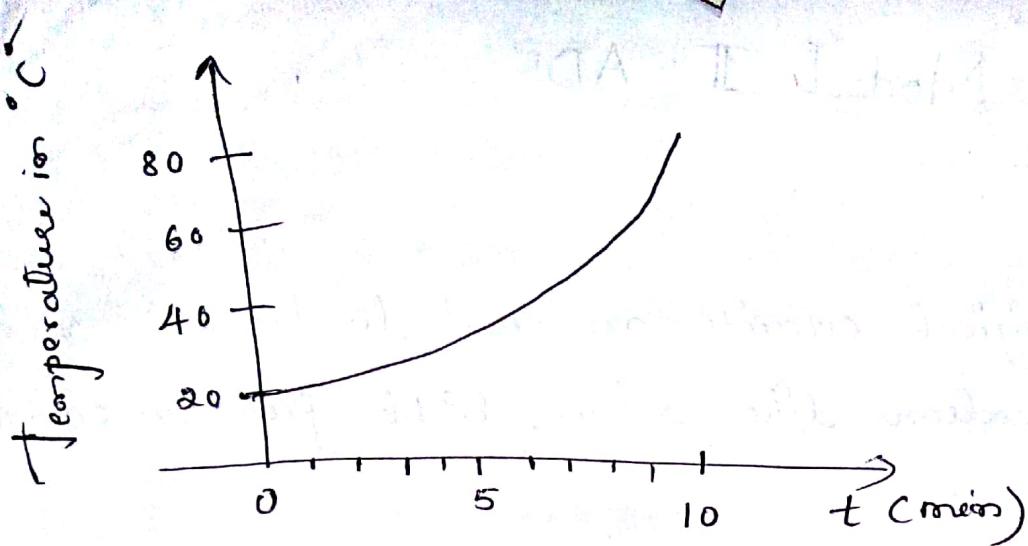
- * used with large signal
- * Is a circuit where the signal should be one of two discrete levels.
- * non linear in nature
- * digital signal represent finite set of values.

Eg: Heat a container of water

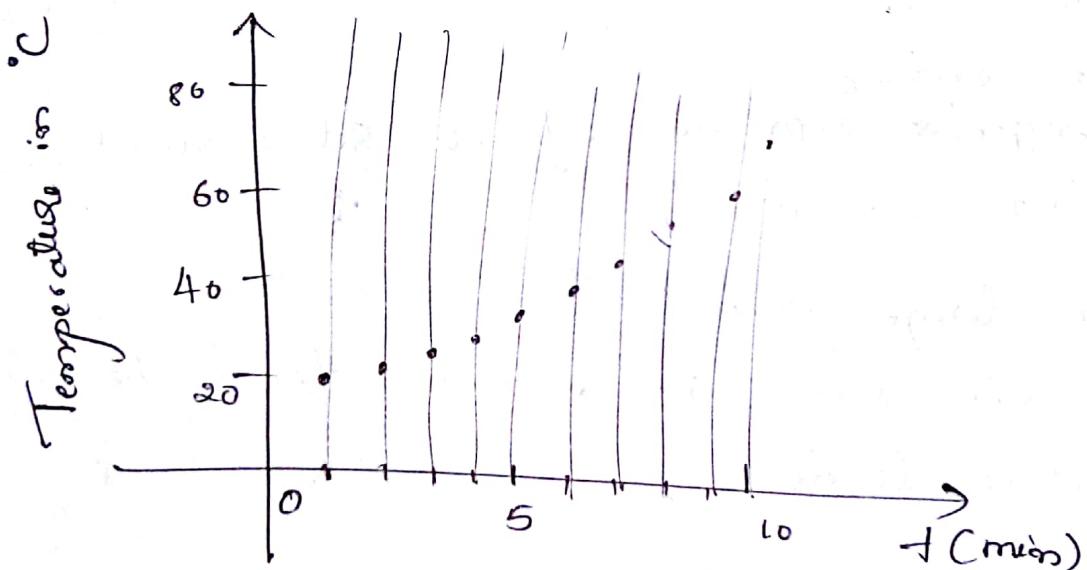
2 ways of recording the temperature of water over a period of time

1. temperature recorded continuously which changes smoothly from 20°C to 80°C

→ Example for analog signal.



2. Record the temperature for particular time interval
Example for digital signal



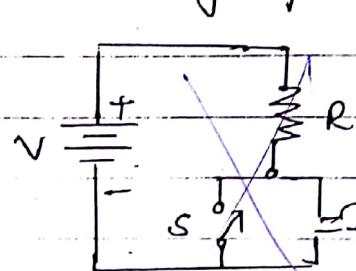
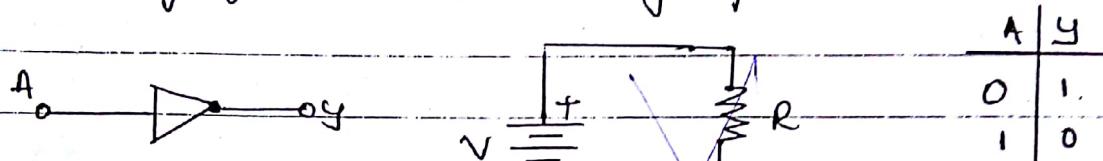
All naturally occurring physical phenomena are analog signals.

The basic Gates:

A digital circuit having one/more i/p signals but only one o/p signal is called gate. Gates simulate mental processes, hence they are called logic circuit. Three logic circuits, the inverter, the OR gate and the AND gate can be used to produce any digital system.

The inverter / NOT gate :

The o/p of is a complement of i/p.



A	Y
0	1
1	0

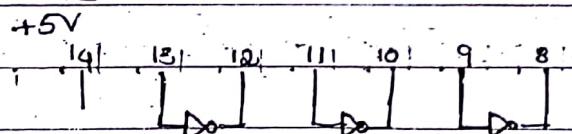
Truth Table

Switch equivalent

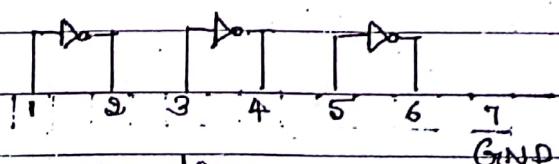
Equation form :

$$Y = \text{not } A$$

$$Y = A'$$



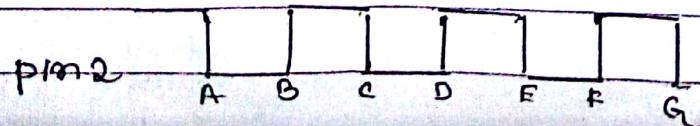
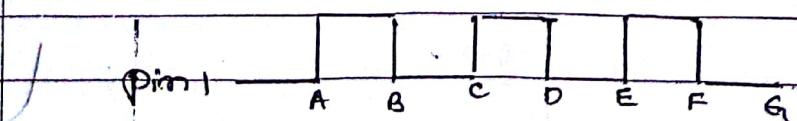
7404



pin diagram.

Used to complement the digital signal.

Prob: A 1-kHz square wave drives pin 1 of a 7404. What does the voltage waveform at pin 2 look like?



Timing diagram

OR Gate:

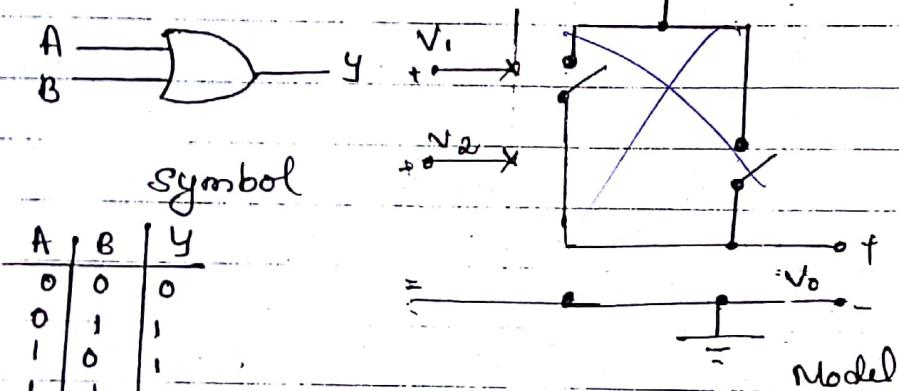
An OR gate has 2 or more input signals but only one output signal and voltage is high if any or all of the input voltages are high.

2 input OR gate : $Y = A \text{ OR } B$

$$Y = A + B$$

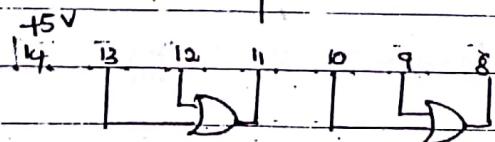
'+' sign represents logic operation OR

$+5 \text{ Vdc}$

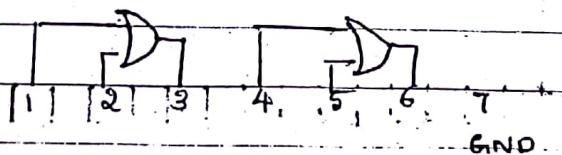


Truth table

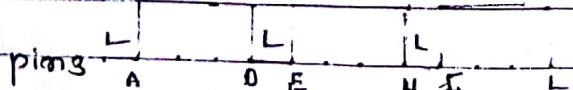
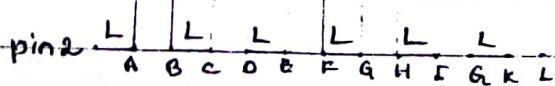
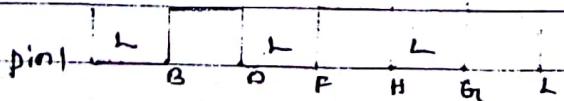
Used to implement logical OR operation.



7432

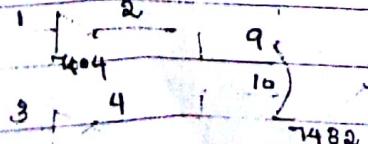


Pinout diagram.



Timing Diagram

Prob: Work out truth table for below fig.



A	B	Y
L	L	L
H	L	H
L	H	H
H	H	L

A digital timing diagram is a representation of set of signals in the time domain

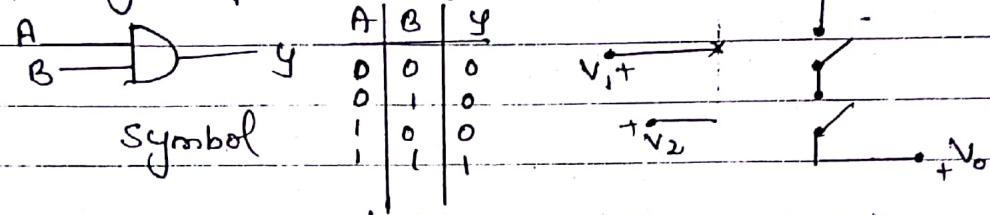
AND gate:

An AND gate has 2/more input signals but only one of signal of voltage is high if any all the i/p's are high

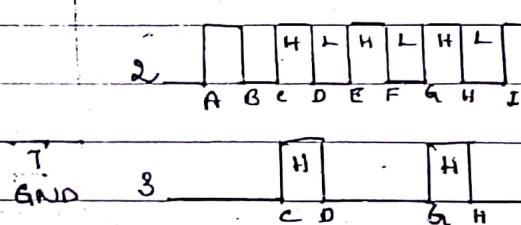
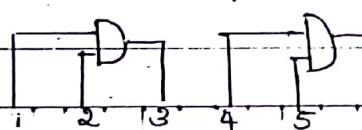
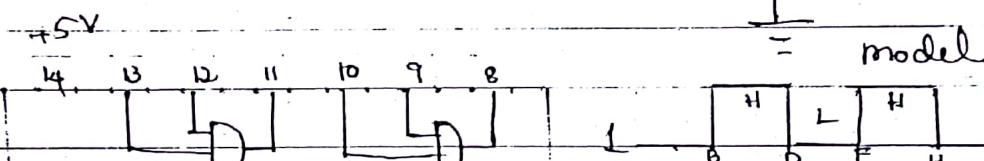
2 i/p AND gate: $Y = A \text{ AND } B$

$$Y = A \cdot B = AB$$

∴ sign represents logic operation AND



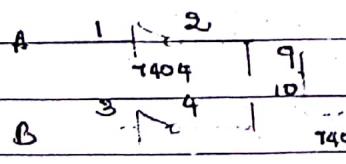
truth table



pinout diagram

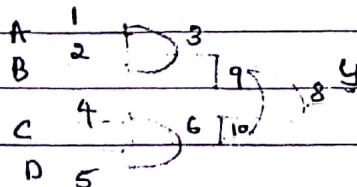
Timing diagram

Prob: Work out truth table for below fig.



A	B	C	D	Y = AB
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Prob: What is the Boolean eqn for the logic circuit below?



$$A = 1 \\ B = 2 \\ C = 3 \\ D = 4 \\ P = 5 \\ Y = A \cdot B + C \cdot D$$

$$Y = AB + CD$$

$$Y = (A+B) \cdot (C+D)$$

What is the logic circuit whose boolean eqn is $Y = A \bar{B} C + A \bar{B} \bar{C}$

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

6.2

Universal Logic Gates:

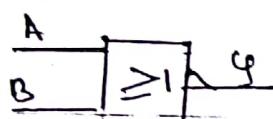
The NAND and NOR gates are known as universal gates, since any logic function can be implemented using NAND or NOR gates.

NOR gate: A NOR gate has 2/more i/p signals but one o/p signal and o/p voltage is ^{high} when ~~any~~ all of the i/p voltages are low.

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

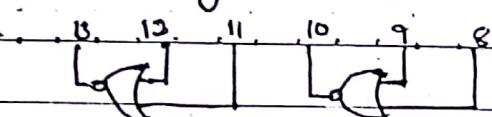


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

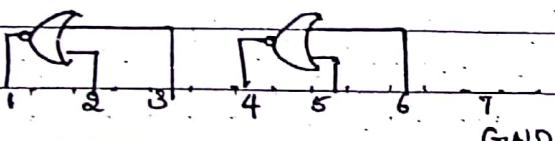


IEEE rectangular symbol

+5V



7402



$$2 \rightarrow \geq 1 \rightarrow 1$$

$$3 \rightarrow \geq 1 \rightarrow 4$$

$$8 \rightarrow \geq 1 \rightarrow 10$$

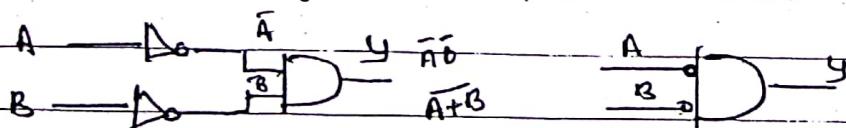
$$11 \rightarrow \geq 1 \rightarrow 13$$

IEEE symbol

pencil diagram

Bubbled AND gate

Bubbles on the i/p's are remainder of the inversion that takes place before AND operation



Bubbled AND gate & NOR gate are equivalent & interchangeable

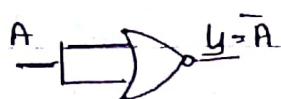
De Morgan's First Theorem:

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

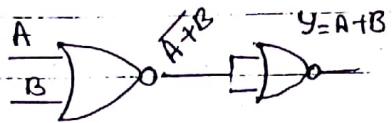
The complements of a sum equals the product of the complements

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

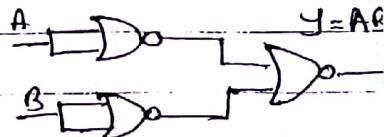
Universality of NOR Gate :



NOT from NOR



NOR from NOR

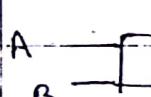


AND from NOR

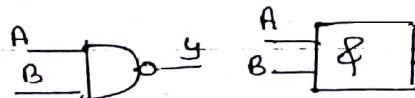
Above fig. shows how all other logic gates can be obtained from NOR gates.

NAND gate : The o/p is high when one of the i/p is low

$$Y = \overline{AB}$$

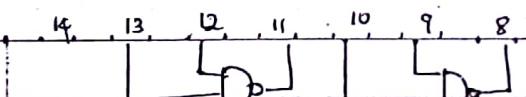


$$Y = \overline{AB}$$

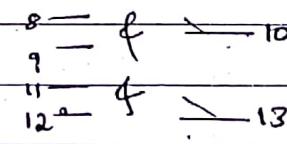
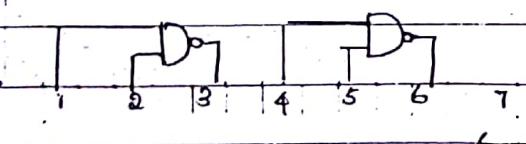
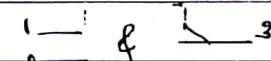


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

+5V



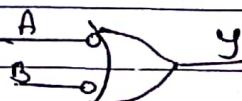
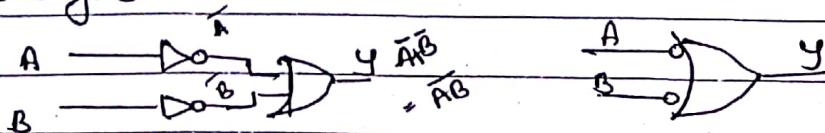
7400



pinout diagram

IEEE symbol

Bubbled OR gate



Bubbled OR gate & NAND gate are equivalent & interchangeable.

D'Morgan's Second Theorem :

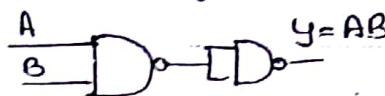
$$\overline{AB} = \overline{A} + \overline{B}$$

The complement of a product equals the sum of the complements.

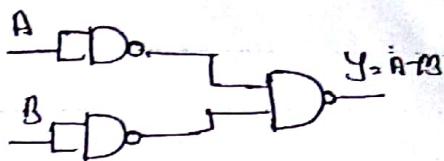
Universality of NAND gate:



NOT from NAND



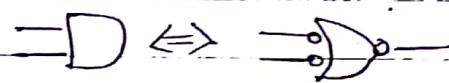
AND from NAND



OR from NAND

Above fig shows how all other gates can be implemented using NAND gate.

Useful logic equivalence:



TTL NAND Gate:

NAND gate is the backbone of the 7400 TTL series bcoz most devices in this family are derived from it. It is least expensive & widely available TTL gate.

Positive and Negative Logic:

An OR gate in a +ve logic system becomes an AND gate in -ve logic system.



Negative AND

A	B	Y
L	L	H
L	H	H
H	L	H
H	H	H

If we convert the above table

in a +ve logic and -ve logic as shown in table a & b.

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

table a

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

table b

OR gate

+ve logic

AND gate

-ve logic

+ve	OR	\leftrightarrow	-ve	AND
+ve	AND	\leftrightarrow	-ve	OR
+ve	NOR	\leftrightarrow	-ve	NAND
+ve	NAND	\leftrightarrow	-ve	NOR

Voltage Definitions of Basic Gates :

+ve OR / -ve AND	O/p is high if any i/p is high
+ve AND / -ve OR	O/p is high when all i/p's are high
+ve NOR / -ve NAND	O/p is low if any i/p is high
+ve NAND / -ve NOR	O/p is low when all i/p's are high

Assertion Level Logic :

A variable / signal is said to be true when it performs the desired action

In digital system there are 2 options to define a variable, either logic 0 or logic 1. This means that each variable has active level associated with it.

A variable is said to be active low when its true value is logic 0 and it is said to be active high when its true value is logic 1.

A signal / variable is said to be asserted when it is at its active level and it is deasserted / negated when it is not at its active level.

assert \Rightarrow to activate. If an i/p line has a bubble on it, we assert the i/p by making it low. If there is no bubble we assert the i/p by making it high.

(16)

Introduction to HDL:

A hardware description language (HDL) is a language which describes the ~~sys~~ of digital systems.

The description is in textual form. The boolean expression, logic diagrams, digital system can be represented in this using HDL.

Why HDL

- i) describe a large complex design requiring hundreds of logic gates in convenient manner in smaller space.
- ii) use s/w test-bench to detect functional error.
- iii) get h/w implementation detail.

2 widely used HDLs are

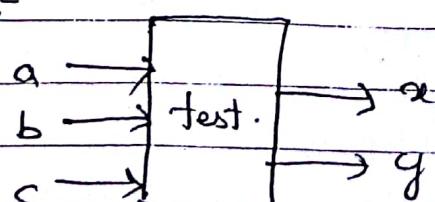
* Verilog

* VHDL : Very High Speed Integrated circuit HDL

Verilog HDL

Introduced in 1980 as simulation & verification package/tool by Gateway Design Automation
Later acquired by Cadence Data Systems
Now acquired by Open Verilog International
module test (x,y,a,b,c); // module name with port list.
input a, b, c; // defines i/p port
output x, y; // defines o/p port

// module body



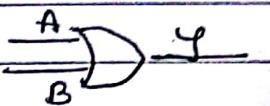
// module body ends
end module.

case sensitive, identifier starts with - or a letter
`/* ... */` multiline comment of any length.
 module, endmodule, input, output are keywords
`//` is for single line comment
`; mol` end statement
 can take up to 16 inputs.

A module describes a design entity with a name or identifier selected by user followed by input output port list.

Example : two input OR gate:

```
module orgate (A,B, Y);
```

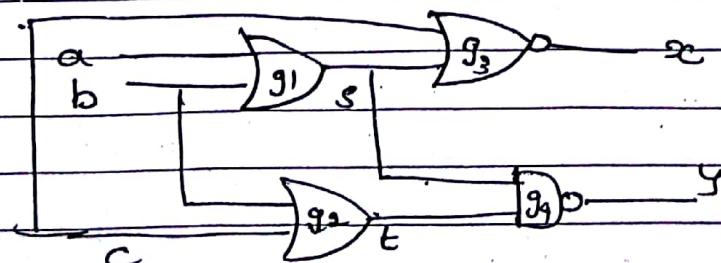


```
    input A,B; // defines 2 i/p port
    output Y; // defines output port
    or g1(Y,A,B); // or predefined keyword
endmodule
```

for OR operation of g1 is user defined gate identifier

Keywords and, or, mol, oxand, mol, xor, xnor, etc

Write Verilog HDL code for following logic circuit



```
module testcircuit(a,b,c, x,y)
```

```
input a, b, c;
```

```
output x, y;
```

```
wire s t;
```

```
or g1 (s,a,b);
```

```
or g2 (t,b,c);
```

#20 // gate delay of 20ns.

```

        nor g3(xc, c, s);
        nand g4(yj, s, t);
endmodule.           Hnzdelay
```

Test Bench:

```
module testor;
```

```
reg A, B;
```

```
wire xc; // storage of data for passing it to
or-gate g1(A, B, xc);           module.
```

```
initial // Simulation begins
```

```
begin
```

```
A = 1'b0; B = 1'b0;
```

```
#20
```

$1'b0 = 0$

```
A = 1'b0; B = 1'b0;
```

$1'b1 = 1$

```
#20
```

```
A = 1'b0; B = 1'b1;
```

```
#20
```

```
A = 1'b1; B = 1'b0;
```

```
#20
```

```
A = 1'b1; B = 1'b1;
```

```
#20
```

```
end
```

```
endmodule
```

```
module Or-gate (A, B, xc)
```

```
input A, B;
```

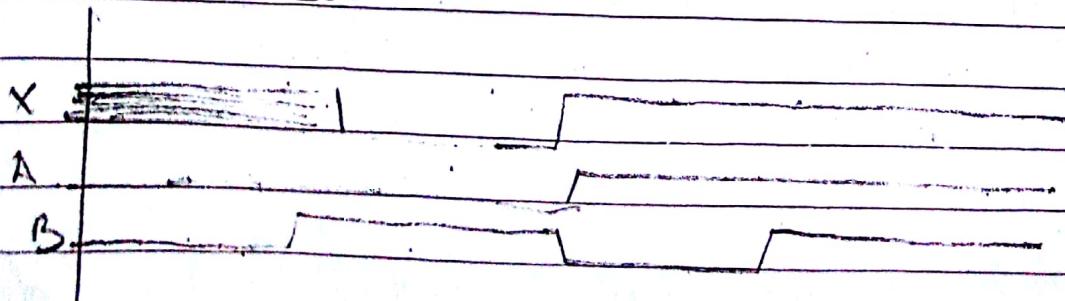
done in.

```
output xc;
```

(unit 2)

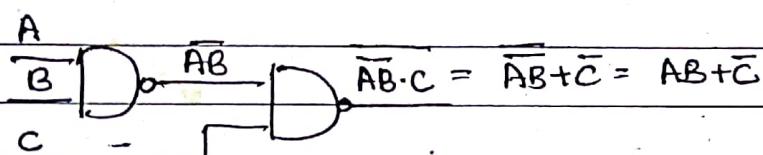
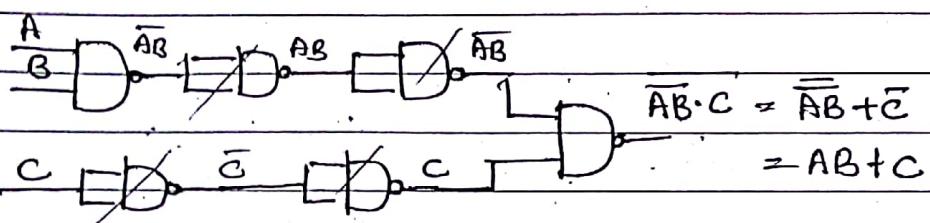
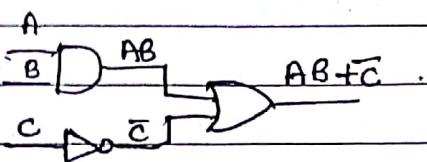
```
or # (20) g1 (xc, A, B);
```

```
endmodule.
```



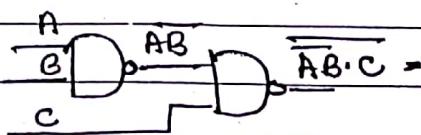
Realize $Y = AB + \bar{C}$ using NAND/NOR
NAND only.

Method 1:



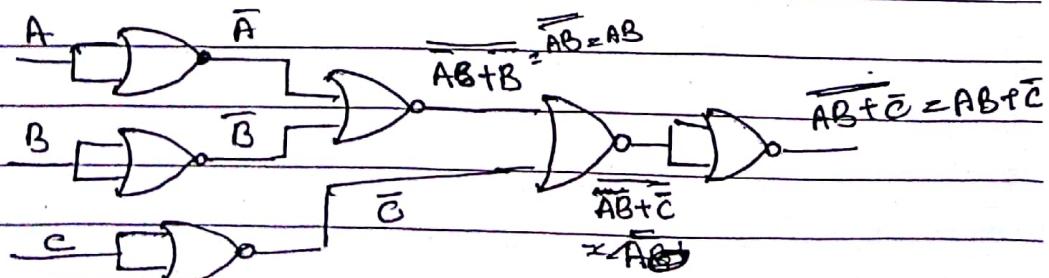
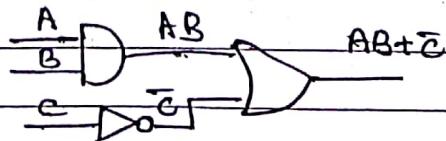
Method 2: $Y = AB + \bar{C}$

$$\begin{aligned} &= \overline{\overline{AB} + \bar{C}} \\ &= \overline{\overline{AB} \cdot \bar{C}} \\ &= \overline{\overline{AB} \cdot C} \end{aligned}$$



NOR only

Method 1:



86 copy → 48.w

Method 2: $\overline{Y} = \overline{AB} + \overline{C}$

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

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

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

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

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

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

