| | |
|---|---|
| STUDENT'S NAME Prateek Pati l | |
| CLASS III-Sem BTech | 2VX22CB037 |
| ROLL No. | DATE |

# DIGITAL DESIGN AND COMPUTER ORGANIZATION

1. Simplify the following Boolean Expressions to minimum literals.

a) $xy + x\bar{y}$

→ $x(y+\bar{y})$

$x(1)$ $[\because y+\bar{y}=1]$

$x$

b) $(\overline{A.B})(\overline{A+B})$

By applying De-morgans law

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

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

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

$0 + A\bar{B} + B\bar{A} + 0$ $[\because A.\bar{A}=0]$

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

c) $xyz + \bar{x}y + xy\bar{z}$

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

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

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

$y$

d) $xy + x(wz + w\bar{z})$

$xy + xw(z+\bar{z})$ $[z+\bar{z}=1]$

$xy + xw$

$x(y+w)$

e) $(\bar{a}+\bar{c})(a+\bar{b}+\bar{c})$

$a.\bar{a} + \bar{b}\bar{a} + \bar{a}\bar{c} + a\bar{c} + \bar{b}\bar{c} + \bar{c}.\bar{c}$

$0 + (a+\bar{a})\bar{c} + \bar{b}\bar{a} + \bar{b}\bar{c} + \bar{c}.\bar{c}$

$\bar{c} + \bar{c} + \bar{b}\bar{a} + \bar{b}\bar{c}$

$\bar{c} + \bar{b}\bar{c} + \bar{b}\bar{a}$

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

$\bar{c}+\bar{b}\bar{a}$

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

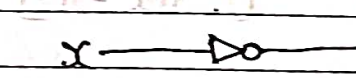
$\bar{A}B\bar{D}+\bar{A}B\bar{C}D+AB+\bar{A}BCD$

2. What do you mean by Universal gates? Illustrate Universal property of NAND as well as NOR gate by implementing basic gates.

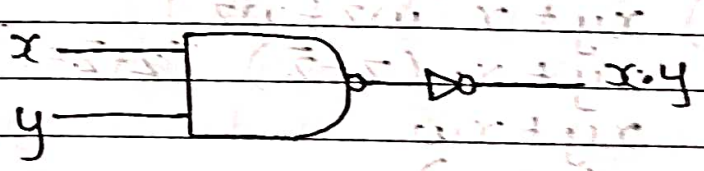
→ A gate is said to be Universal gate because any logic circuit can be implemented with it. We need to show logical operations of AND, OR and complement can be obtained with that gate alone.

i) NAND as Universal gate

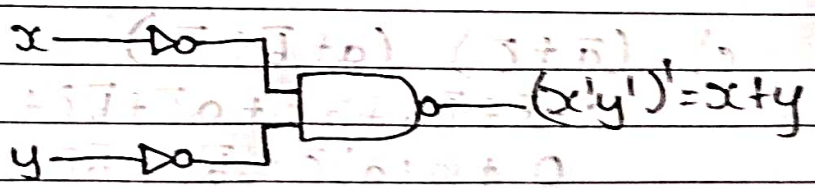
Inverter




AND



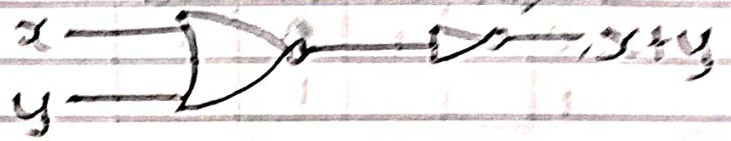

OR

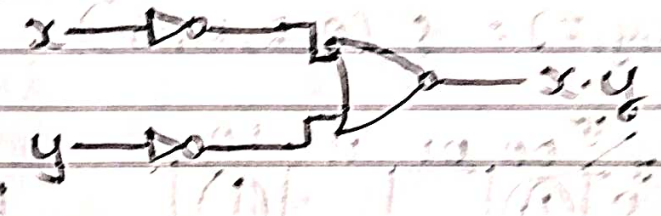


| STUDENT'S NAME | |
|---|---|
| CLASS | |
| ROLL NO. | DATE |

ii) NOR as Universal gate.

Inverter $x \longrightarrow x'$

OR 

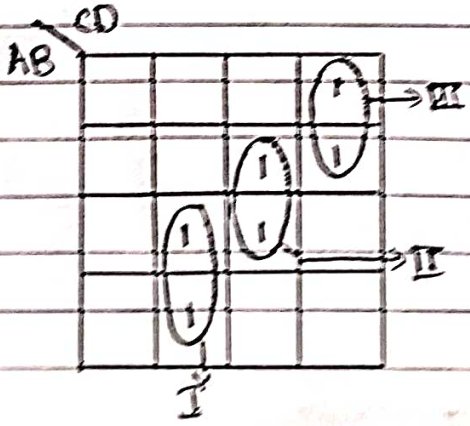

AND 


3. Define the terms implicant, prime implicant & essential prime implicants.

→ Any minterm or combination of minterms which makes an SOP function equal to 1, is called implicant. The group must contain either 1, 2, 4, 8, 16... ones.

Prime implicant :- The largest possible group of 1 which can be used to cover given 1 is called prime implicant

Essential prime implicant :- Essential prime implicant involves atleast two if single 1 which cannot be combined in any other way



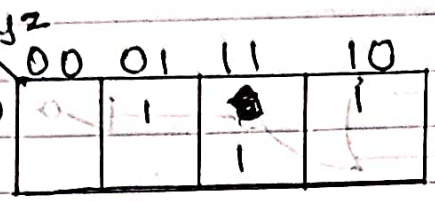

I → Prime, implicant, essential prime implicant

II → Prime implicant

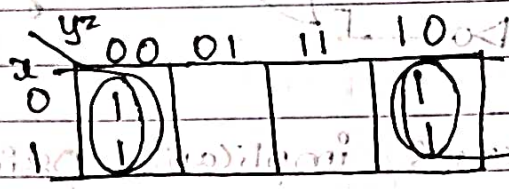
III → P.I, E.P.I

4. Simplify the following using K-Map.

a) $F(x,y,z) = \Sigma(1,2,3,7)$



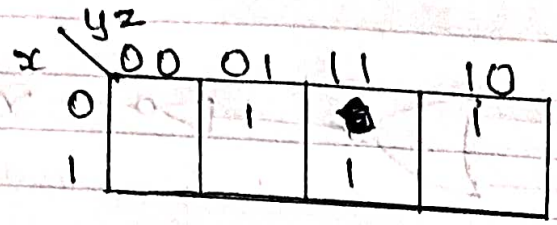

b) $F(x,y,z) = \Sigma(0,2,4,6)$



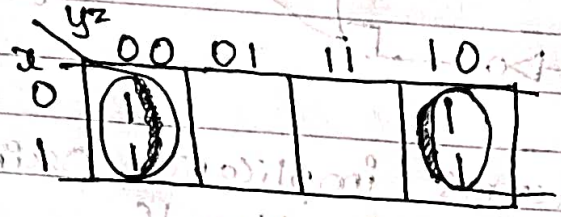

$F = \bar{z}$

4. Simplify the following using K-Map.

a) $F(x,y,z) = \Sigma(1,2,3,7)$

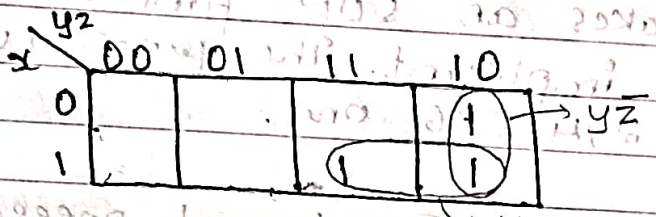



b) $F(x,y,z) = \Sigma(0,2,4,6)$




$F = \bar{z}$

c) $xy + \bar{x}y\bar{z} + \bar{x}y\bar{z}$




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

∴ The simplified version is

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

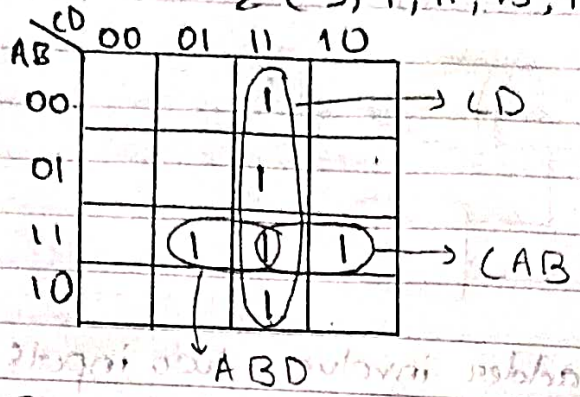
d) $F(xyz) = \bar{x}y + y\bar{z} + \bar{y}\bar{z}$


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

$\therefore F = \bar{z} + y$

| STUDENT'S NAME | |
|---|---|
| CLASS | |
| ROLL No. | DATE |

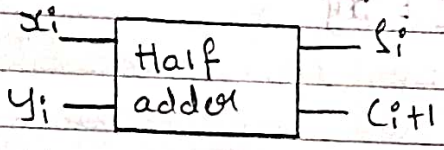
c) $F(A,B,C,D) = \Sigma(3,7,11,13,14,15)$




$\therefore F = ABC + AB'D + CD$

5. Write truth table of half adder & full adder & Explain working of both.

→

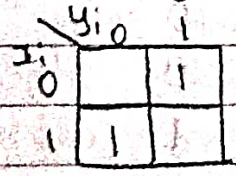



Truth table

| $x_i$ | $y_i$ | $S_i$ | $C_{i+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

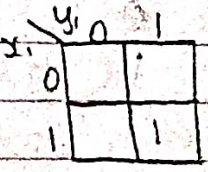
where, $x_i$ & $y_i$ are the inputs of half adder & $S_i$ is the sum of input bits and $C_{i+1}$ is the carry bit.

By using K-map




$S_i$

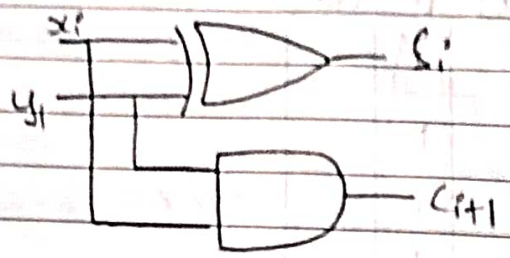
$F = x_i\bar{y}_i + \bar{x}_i y_i$




$C_{i+1}$

$F = x_i y_i$

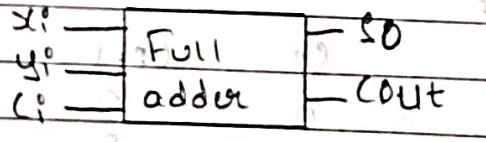
The circuit diagram looks like this




The half adder involves two inputs i.e. addend and augend and produces output $S_i$ and Carry. The simplified boolean functions can be taken by using K-map i.e.

$$S = xy' + x'y$$
$$C = xy$$

Full adder

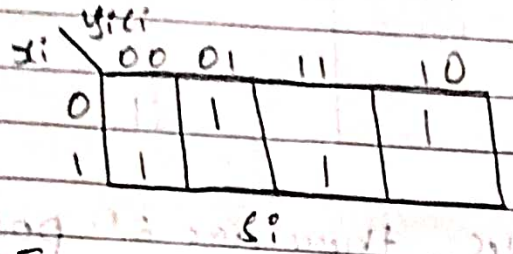


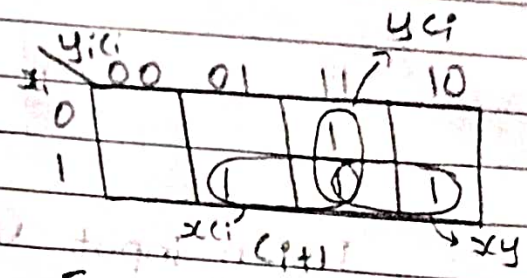

Truth table

| $x_i$ | $y_i$ | $C_i$ | SO | Cout |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

| STUDENT'S NAME | |
|---|---|
| CLASS | |
| ROLL No. | DATE |

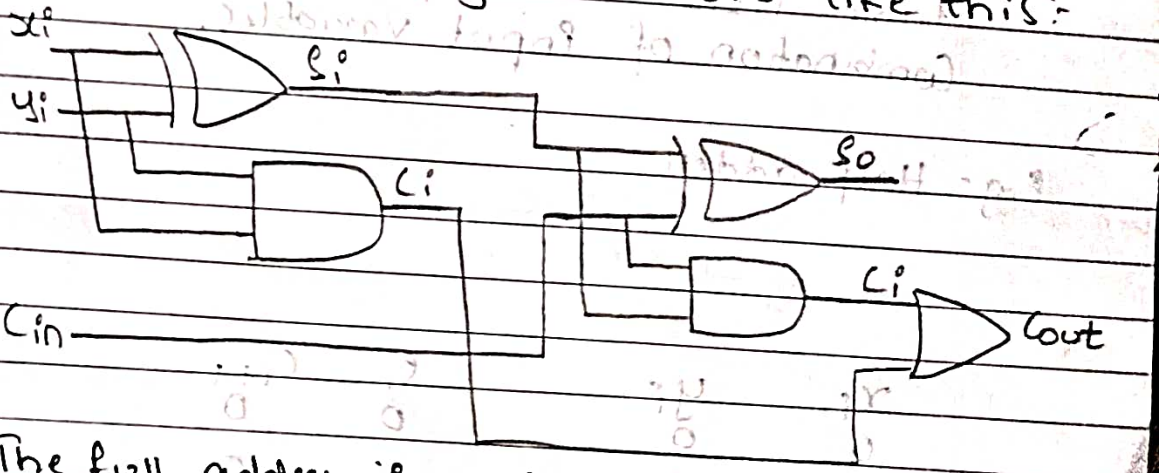
By using K-Map




$F = x\bar{y}\bar{z} + x\bar{y}z + xyz + \bar{x}y\bar{z}$ $F = xC_i + xy + yC_i$

The circuit diagram looks like this:-



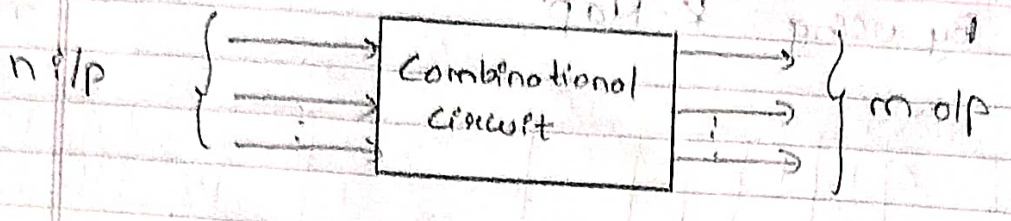

The full adder is a combinational circuit which takes three input & give 2 output i.e. So & Cout. The simplified boolean function can be taking using K-map i.e.

$$S = x\bar{y}_i\bar{C}_i + x\bar{y}_i C_i + xyC_i + \bar{x}y_i\bar{C}_i$$

6. What do you mean by combinational circuits explain with example.

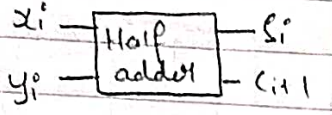
→ A combinational circuit consists of logic gates whose outputs are determined by the present combination of inputs. i.e. no previously stored value or state is taken into consideration here.

Block diagram for combinational circuits.



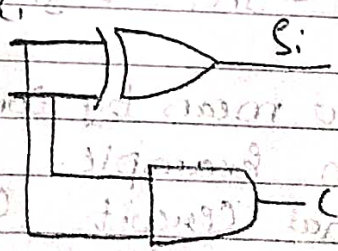

For n, input variables, there are $2^n$ possible combinations of the binary inputs. For each binary input combination there is one possible value for each output. The truth table lists the output values for each combination of input variables.

E.g:- Half adder




| $x_i$ | $y_i$ | $S_i$ | $C_{i+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

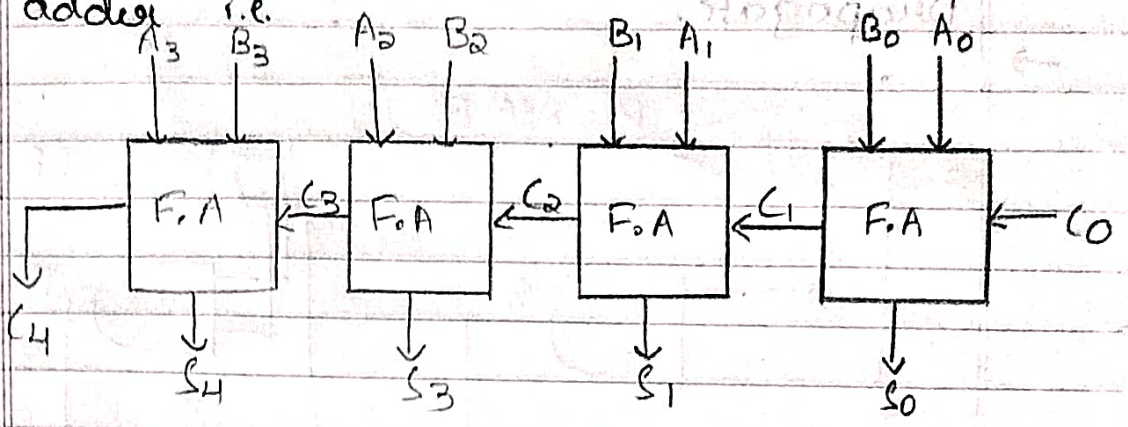
∴ The circuit diagram is



| STUDENT'S NAME | |
|---|---|
| CLASS | |
| ROLL No. | DATE |

7. Explain working of a 4-bit parallel binary adder

→ Considering, a four bit parallel binary adder i.e.




where $A_0, B_0$ are the input to the Full adder & $C_0$ is the carry bit.

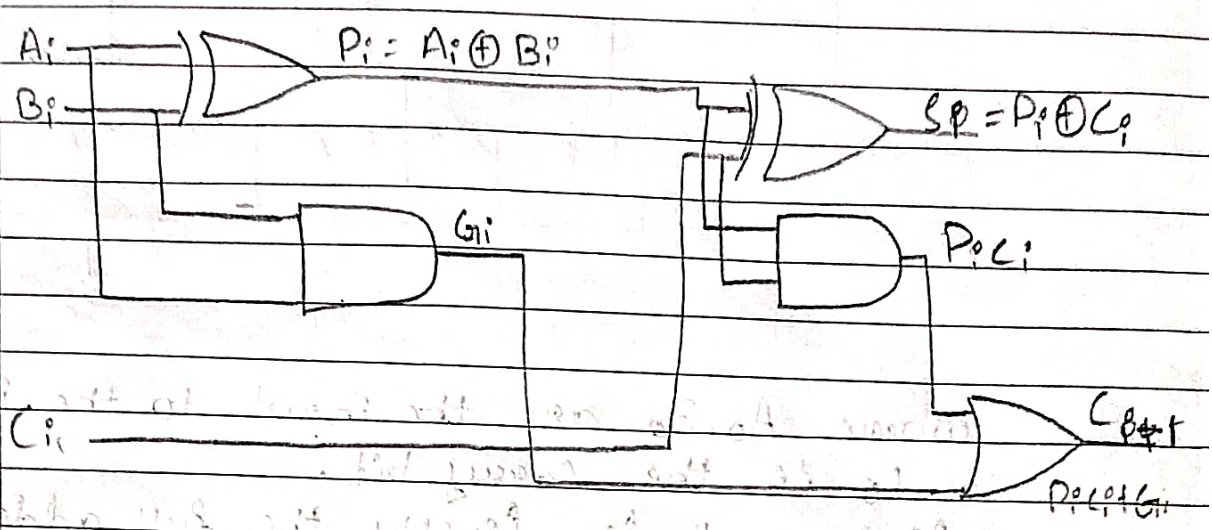
* As shown in fig, firstly the full adder FA I adds $A_0$ $B_0$ along with carry $C_0$ to generate Sum $S_0$. The carry $C_1$ which is connected to the next adder in chain
* Next, the full adder FA 2 uses the carry bit $C_1$ with the input bits $A_1, B_1$ to generate Sum $S_1$ & further carry $C_2$ is taken to the next adder in chain
* The process continues till the last full adder F.A4 uses the carry bit $C_3$ to add with its input $A_3$ & $B_3$ to generate the last bit of output along last carry bit $C_4$.

E.g:

| Subscript i: | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|
| Input Carry | 0 | 1 | 1 | 0 | $C_i$ |
| Augend | 1 | 0 | 1 | 1 | $A_i$ |
| Addend | 0 | 0 | 1 | 1 | $B_i$ |
| Sum | 1 | 1 | 1 | 0 | $S_i$ |
| Output Carry | 0 | 0 | 1 | 1 | $C_{i+1}$ |

8. With diagram & Expression, Explain the gate level delays in n-bit adder & how it is minimized using generate & propogate.

→




where, $P_i$ = Propogate = $A_i \oplus B_i$
$g_i$ = generate term = $A_i B_i$

The n-bit adders sum output depends on the value of the input carry i.e. the value of $S_i$ at any given stage in the adder is the steady state final value only after the i/p carry to that stage is propogated.

$$\therefore C_{i+1} = P_i C_i + G_i$$

$$C_1 = P_0 C_0 + G_0$$
$$C_2 = P_1 C_1 + G_1$$
$$C_2 = P_1 (P_0 C_0 + G_0) + G_1$$
$$C_2 = P_1 P_0 C_0 + P_1 G_0 + G_1$$
$$C_3 = P_2 C_2 + G_2$$
$$C_3 = G_2 + (P_1 P_0 C_0 + P_1 G_0 + G_1) P_2$$
$$C_3 = G_2 + P_2 G_1 + P_1 P_2 G_0 + P_0 P_1 P_2 C_0$$