13-08-2014

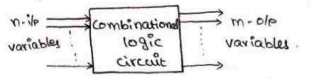
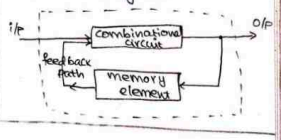
# UNIT - II

## COMBINATIONAL LOGIC

there are two types of digital logic circuits.

(i) combinational circuit.

(ii) sequential circuit.

| Combinational circuits | Sequential circuit. |
|---|---|
| * o/p depends on present i/p's. | * o/p depends on present i/p's & past o/p's. |
| * contains no internal memory | * contains internal memory |
| * It consists of logic gates | * It consists of logic gates & memory elements. |
| * Block diagram.   | * Block diagram   |

### Design procedure of a combinational circuit :-

Step 1 :- From the specifications of the circuit, determine the required number of i/p's & o/p's and assign a symbol to each.

Step 2 :- Derive the truth table that defines the required relation b/w i/p's & o/p's.

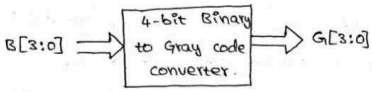
Step 3 :- Obtain the simplified boolean functions for each o/p as a function of the i/p variables.

Step 4 :- Draw the logic diagram and verify the correctness of the design.

### * Code converters :-

* → construct 4-bit binary to Gray code converter

4-bit Binary to Gray code converter converts 4-bit binary to 4 bit Gray code.




B[3:0] ⟶ 4-bit Binary number.

$B = B_3 B_2 B_1 B_0$

G[3:0] ⟶ 4-bit Gray number.

$G = G_3 G_2 G_1 G_0$

Truth Table :-

| | i/p (Binary) | | | | o/p (Gray) | | | |
|---|---|---|---|---|---|---|---|---|
| | $B_3$ | $B_2$ | $B_1$ | $B_0$ | $G_3$ | $G_2$ | $G_1$ | $G_0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 11 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

In Gray numbers, 1's are minterms, 0's are Maxterms.

$G_0 = \Sigma m(1,2,5,6,9,10,13,14)$

$G_1 = \Sigma m(2,3,4,5,10,11,12,13)$

$G_2 = \Sigma m(4,5,6,7,8,9,10,11)$

$G_3 = \Sigma m(8,9,10,11,12,13,14,15)$

## K-Map Simplification :-

For $G_0$ :-

| $B_3B_2 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 (1) | 3 | 1 (2) |
| 01 | 1 (4) | 1 (5) | 1 (7) | 1 (6) |
| 11 | 1 (12) | 1 (13) | 1 (15) | 1 (14) |
| 10 | 1 (8) | 1 (9) | 1 (11) | 1 (10) |

$\therefore \boxed{G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0 = B_1 \oplus B_0}$

For $G_1$ :-

| $B_3B_2 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | 1 | 1 | | |
| 11 | 1 | 1 | | |
| 10 | | | 1 | 1 |

$\therefore \boxed{G_1 = B_1\bar{B}_2 + \bar{B}_1 B_2 = B_2 \oplus B_1}$

For $G_2$ :-

| $B_3B_2 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | 1 | 1 | 1 |
| 11 | | | | |
| 10 | 1 | 1 | 1 | 1 |

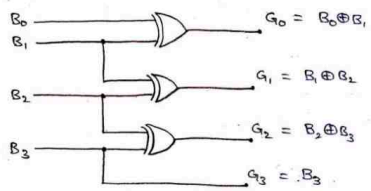
$\therefore \boxed{G_2 = B_3\bar{B}_2 + \bar{B}_3 B_2 = B_3 \oplus B_2}$

For $G_3$ :-

| $B_3B_2 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

$\therefore \boxed{G_3 = B_3}$

## Logic circuits :-




## * Gray to Binary converter :-

$G[3:0] \Rightarrow$ **4-bit Gray to Binary converter** $\Rightarrow B[3:0]$

$G = G_3G_2G_1G_0$ $\qquad$ $B = B_3B_2B_1B_0$

Truth table :-

| | i/p (Gray) $G_3\ G_2\ G_1\ G_0$ | o/p (Binary) $B_3\ B_2\ B_1\ B_0$ |
|---|---|---|
| 0 | 0 0 0 0 | 0 0 0 0 |
| 1 | 0 0 0 1 | 0 0 0 1 |
| 2 | 0 0 1 0 | 0 0 1 0 |
| 3 | 0 0 1 1 | 0 0 1 1 |
| 4 | 0 1 1 0 | 0 1 0 0 |
| 5 | 0 1 1 1 | 0 1 0 1 |
| 6 | 0 1 0 1 | 0 1 1 0 |
| 7 | 0 1 0 0 | 0 1 1 1 |
| 8 | 1 1 0 0 | 1 0 0 0 |
| 9 | 1 1 0 1 | 1 0 0 1 |
| 10 | 1 1 1 1 | 1 0 1 0 |
| 11 | 1 1 1 0 | 1 0 1 1 |
| 12 | 1 0 1 0 | 1 1 0 0 |
| 13 | 1 0 1 1 | 1 1 0 1 |
| 14 | 1 0 0 1 | 1 1 1 0 |
| 15 | 1 0 0 0 | 1 1 1 1 |

| | i/p (Gray) $G_3\ G_2\ G_1\ G_0$ | o/p (Binary) $B_3\ B_2\ B_1\ B_0$ |
|---|---|---|
| 0 | 0 0 0 0 | 0 0 0 0 |
| 1 | 0 0 0 1 | 0 0 0 1 |
| 2 | 0 0 1 0 | 0 0 1 1 |
| 3 | 0 0 1 1 | 0 0 1 0 |
| 4 | 0 1 0 0 | 0 1 1 1 |
| 5 | 0 1 0 1 | 0 1 1 0 |
| 6 | 0 1 1 0 | 0 1 0 0 |
| 7 | 0 1 1 1 | 0 1 0 1 |
| 8 | 1 0 0 0 | 1 1 1 1 |
| 9 | 1 0 0 1 | 1 1 1 0 |
| 10 | 1 0 1 0 | 1 1 0 0 |
| 11 | 1 0 1 1 | 1 1 0 1 |
| 12 | 1 1 0 0 | 1 0 0 0 |
| 13 | 1 1 0 1 | 1 0 0 1 |
| 14 | 1 1 1 0 | 1 0 1 1 |
| 15 | 1 1 1 1 | 1 0 1 0 |

$B_0 = \cancel{\Sigma m(1,3,5,7,9,11,13,15)} = \Sigma m(1,2,4,7,8,11,13,14)$

$B_1 = \cancel{\Sigma m(2,3,6,7,10,11,14,15)} = \Sigma m(2,3,4,5,8,9,14,15)$

$B_2 = \cancel{\Sigma m(4,5,6,7,12,13,14,15)} = \Sigma m(4,5,6,7,8,9,10,11)$

$B_3 = \cancel{\Sigma m(8,9,10,11,12,13,14,15)} = \Sigma m(8,9,10,11,12,13,14,15)$

## K-Map simplification :-

For $B_0$ :-

| $G_3G_2 \backslash G_1G_0$ | $\bar{G}_1\bar{G}_0$ 00 | $\bar{G}_1G_0$ 01 | $G_1\bar{G}_0$ 11 | $G_1G_0$ 10 |
|---|---|---|---|---|
| $\bar{G}_3\bar{G}_2$ 00 | - 0 | 1 1 | 3 | 1 2 |
| $\bar{G}_3G_2$ 01 | 1 4 | 5 | 1 7 | 6 |
| $G_3G_2$ 11 | 12 | 1 13 | 15 | 1 14 |
| $G_3\bar{G}_2$ 10 | 1 8 | 9 | 1 11 | 10 |

$\therefore B_0 = (G_0 \oplus G_1) \oplus (G_2 \oplus G_3)$

For $B_1$ :-

| $G_3G_2 \backslash G_1G_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | 1 | 1 | | |
| 11 | | | 1 | 1 |
| 10 | 1 | 1 | | |

$\therefore B_1 = \bar{G}_2\bar{G}_3G_1 + G_2\bar{G}_3\bar{G}_1 + G_2G_3G_1 + \bar{G}_2G_3\bar{G}_1$

$= \bar{G}_3(G_1\bar{G}_2 + \bar{G}_1G_2) + G_3(G_1G_2 + \bar{G}_1\bar{G}_2)$

$\therefore B_1 = G_1 \oplus G_2 \oplus G_3$

For $B_2$ :-

| $G_3G_2 \backslash G_1G_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | 1 | 1 | 1 |
| 11 | | | | |
| 10 | 1 | 1 | 1 | 1 |

$\therefore B_2 = G_2\bar{G}_3 + \bar{G}_2G_3$

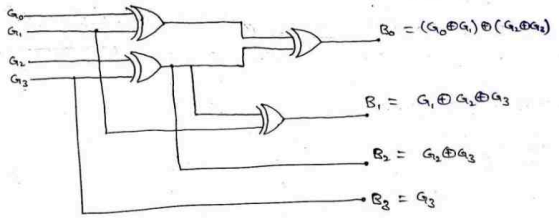
$\therefore B_2 = G_2 \oplus G_3$

For $B_3$ :-

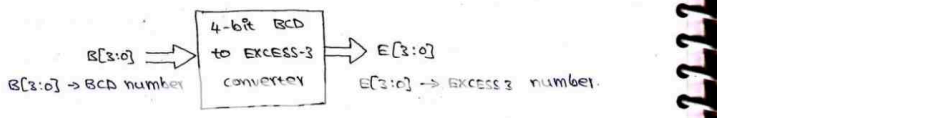
| $G_3G_2 \backslash G_1G_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

$\therefore B_3 = G_3$

## Logic circuit :-




## * → BCD to EXCESS-3 converter :-




Truth table :-

| i/p (BCD) | | | | o/p (EXCESS-3) | | | |
|---|---|---|---|---|---|---|---|
| $B_3$ | $B_2$ | $B_1$ | $B_0$ | $E_3$ | $E_2$ | $E_1$ | $E_0$ |
| 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 |

BCD numbers are 0-9

∴ consider 10-15 as don't cares.

$E_0 = \Sigma m(0,2,4,6,8) + \Sigma d(10,11,12,13,14,15)$

$E_1 = \Sigma m(0,3,4,7,8) + \Sigma d(10,11,12,13,14,15)$

$E_2 = \Sigma m(1,2,3,4,9) + \Sigma d(10,11,12,13,14,15)$

$E_3 = \Sigma m(5,6,7,8,9) + \Sigma d(10,11,12,13,14,15)$

K-Map simplification :-

For $E_0$ :-

| $B_3B_2 \backslash B_1B_0$ | $\bar{B}_1\bar{B}_0$ 00 | $\bar{B}_1B_0$ 01 | $B_1B_0$ 11 | $B_1\bar{B}_0$ 10 |
|---|---|---|---|---|
| $\bar{B}_3\bar{B}_2$ 00 | 1 | | | 1 |
| $\bar{B}_3B_2$ 01 | 1 | | | 1 |
| $B_3B_2$ 11 | X | X | X | X |
| $B_3\bar{B}_2$ 10 | 1 | | X | X |

$\therefore \boxed{E_0 = \bar{B}_0}$

For $E_1$ :-

| $B_3B_2 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | 1 | |
| 01 | 1 | | 1 | |
| 11 | X | X | X | X |
| 10 | 1 | | X | X |

$\therefore \boxed{E_1 = B_1 \oplus B_0 = \bar{B}_0\bar{B}_1 + B_0B_1}$

For $E_2$ :-

| $B_3B_2 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | 1 | 1 | 1 |
| 01 | 1 | | | |
| 11 | X | X | X | X |
| 10 | | 1 | X | X |

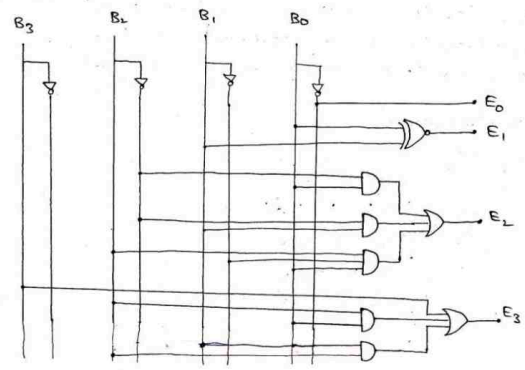
$\therefore \boxed{E_2 = \bar{B}_2B_0 + \bar{B}_2B_1 + B_2\bar{B}_1\bar{B}_0}$

For $E_3$ :-

| $B_3B_2 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | 1 | 1 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 1 | X | X |

$\therefore \boxed{E_3 = B_3 + B_2B_0 + B_2B_1}$

logic circuit :-




20/08/2014

→ EXCESS-3 to BCD converter :-

Truth table :-

| i/p (Excess-3) | o/p (BCD) |
|---|---|
| $E_3\ E_2\ E_1\ E_0$ | $B_3\ B_2\ B_1\ B_0$ |
| 0 0 0 0 | X X X X |
| 0 0 0 1 | X X X X |
| 0 0 1 0 | X X X X |
| 0 0 1 1 | 0 0 0 0 |
| 0 1 0 0 | 0 0 0 1 |
| 0 1 0 1 | 0 0 1 0 |
| 0 1 1 0 | 0 0 1 1 |
| 0 1 1 1 | 0 1 0 0 |
| 1 0 0 0 | 0 1 0 1 |
| 1 0 0 1 | 0 1 1 0 |
| 1 0 1 0 | 0 1 1 1 |
| 1 0 1 1 | 1 0 0 0 |
| 1 1 0 0 | 1 0 0 1 |
| 1 1 0 1 | X X X X |
| 1 1 1 0 | X X X X |
| 1 1 1 1 | X X X X |

## Binary to BCD converter :-

Ex :- 1 5 → 0001 0101

for 10 - 15 no.'s 3 MSB's are 0's.

∴ neglect 3 MSB's.

Truth table :-

| i/p (Binary) | o/p (BCD) |
|---|---|
| $B_3$ $B_2$ $B_1$ $B_0$ | $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ |
| 0 0 0 0 | 0 0 0 0 0 |
| 0 0 0 1 | 0 0 0 0 1 |
| 0 0 1 0 | 0 0 0 1 0 |
| 0 0 1 1 | 0 0 0 1 1 |
| 0 1 0 0 | 0 0 1 0 0 |
| 0 1 0 1 | 0 0 1 0 1 |
| 0 1 1 0 | 0 0 1 1 0 |
| 0 1 1 1 | 0 0 1 1 1 |
| 1 0 0 0 | 0 1 0 0 0 |
| 1 0 0 1 | 0 1 0 0 1 |
| 1 0 1 0 | 1 0 0 0 0 |
| 1 0 1 1 | 1 0 0 0 1 |
| 1 1 0 0 | 1 0 0 1 0 |
| 1 1 0 1 | 1 0 0 1 1 |
| 1 1 1 0 | 1 0 1 0 0 |
| 1 1 1 1 | 1 0 1 0 1 |

## BCD to Binary converter :-

| Input (BCD) | o/p (Binary) |
|---|---|
| $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | $B_3$ $B_2$ $B_1$ $B_0$ |
| 0 0 0 0 0 | 0 0 0 0 |
| 0 0 0 0 1 | 0 0 0 1 |
| 0 0 0 1 0 | 0 0 1 0 |
| 0 0 0 1 1 | 0 0 1 1 |
| 0 0 1 0 0 | 0 1 0 0 |
| 0 0 1 0 1 | 0 1 0 1 |
| 0 0 1 1 0 | 0 1 1 0 |

| $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | $B_3$ $B_2$ $B_1$ $B_0$ |
|---|---|
| 0 1 0 0 0 | 1 0 0 0 |
| 0 1 0 0 1 | 1 0 0 1 |
| 1 0 0 0 0 | 1 0 1 0 |
| 1 0 0 0 1 | 1 0 1 1 |
| 1 0 0 1 0 | 1 1 0 0 |
| 1 0 0 1 1 | 1 1 0 1 |
| 1 0 1 0 0 | 1 1 1 0 |
| 1 0 1 0 1 | 1 1 1 1 |

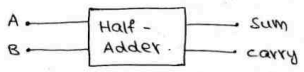
## * ADDERS :-

Adders is a digital circuit which performs addition operations.

### 1. Half Adders :-

The logic circuit which performs addition of 2 bits is called a "Half Adder". It contains two Binary i/p (Augend & Addend) and two Binary o/p (sum & carry).

Block diagram :-

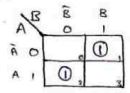



Truth table :-

| i/p's | | o/p's | |
|---|---|---|---|
| A | B | sum | carry |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

K-Map simplification :-

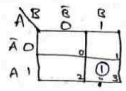
For sum :-




∴ Sum $= \bar{A}B + A\bar{B}$

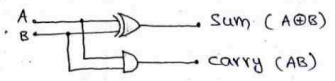
$= A \oplus B$

For carry :-




∴ carry $= AB$.

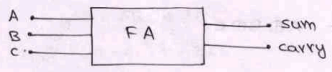
logic circuit :-



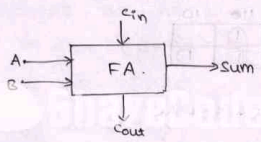
## 2. Full Adder :-

Full Adder is a combinational circuits that forms the arithmetic sum of 3-input bits. It consists of 3 i/p's & 2 o/p. The third input is a $C_{in}$ which represents the carry from the previous significant position.

Block diagram :-




(or)




Truth table :-

| i/p's | | | o/p's | |
|---|---|---|---|---|
| A | B | C | Sum | carry |
| 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 |

K-Map simplifications :-

For sum

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

$\therefore$ sum $= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

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

$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$ $\quad (\because \bar{x}y + x\bar{y} = x \oplus y)$

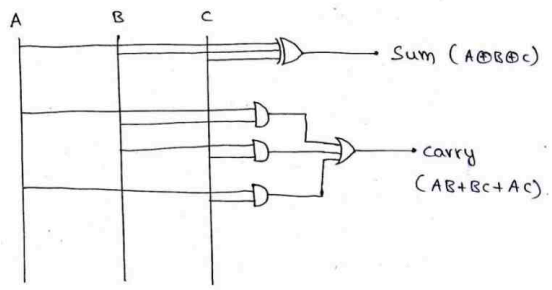
$= A \oplus B \oplus C.$

For carry

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

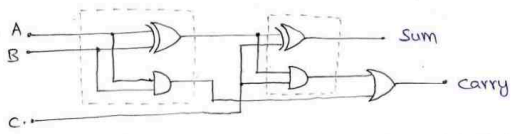
$\therefore$ carry $= AB + BC + AC.$

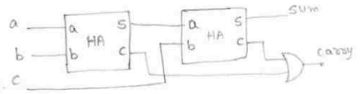
logic circuit :-




Note :- A full adder can also be implemented by using two half-adders and 1 OR gate.

## Full Adder using Half Adders:-




$\therefore$ Sum $= A \oplus B \oplus C$.

$$\text{carry} = AB + (A \oplus B)C$$
$$= AB + (\bar{A}B + A\bar{B})C$$
$$= AB + \bar{A}BC + A\bar{B}C$$
$$= B(A + \bar{A}C) + A\bar{B}C$$
$$= B((A+\bar{A})(A+C)) + A\bar{B}C$$
$$= AB + BC + A\bar{B}C$$
$$= AB + C(\bar{B} + A\bar{B}) = AB + C((B+A)(B+\bar{B}))$$
$$= AB + BC + AC$$

$\therefore$ carry $= AB + BC + AC$.

## 3. Parallel Adder (Binary Adder):-

It is a combinational circuit which performs addition of 2, n-bit number.

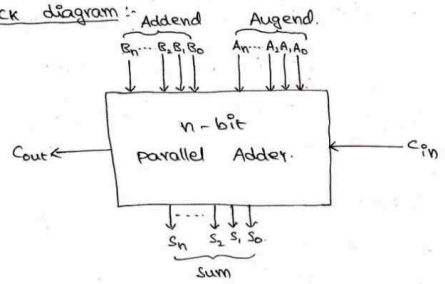
A & B are n+1 bit numbers.

$A \rightarrow A_n \cdots A_3 A_2 A_1 A_0$

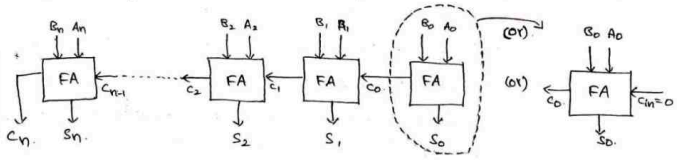
$B \rightarrow B_n \cdots B_3 B_2 B_1 B_0$

$$\begin{array}{lll} & c_{n-1} \quad c_2 \; c_1 \; c_0 & \\ & A_n \cdots\cdots A_2 \; A_1 \; A_0 & \longrightarrow \text{Augend} \\ + & B_n \cdots\cdots B_2 \; B_1 \; B_0 & \longrightarrow \text{Addend} \\ \hline & c_n \; S_n \qquad S_2 \; S_1 \; S_0 & \longrightarrow \text{Sum.} \end{array}$$

## Block diagram:-




## Logic diagram:-




26/08/2014

**Ex:-** Design 4-bit parallel adder using full-Adders.

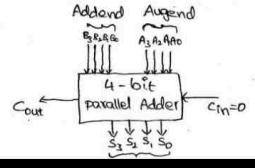
**Sol:-** A, B $\longrightarrow$ 4-bit Binary numbers.

$A = A_3 A_2 A_1 A_0$

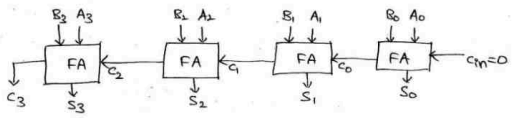
$B = B_3 B_2 B_1 B_0$

$$\begin{array}{lll} & c_2 \; c_1 \; c_0 \; c_{in}=0 & \\ & A_3 \; A_2 \; A_1 \; A_0 & \longrightarrow \text{Augend} \\ & B_3 \; B_2 \; B_1 \; B_0 & \longrightarrow \text{Addend} \\ \hline & c_3 \; S_3 \; S_2 \; S_1 \; S_0 & \longrightarrow \text{result.} \end{array}$$

Block diagram:-



Logic diagram :-

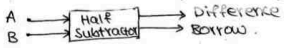



* **Subtractors:-** Subtractor is a digital circuit which performs subtraction operations.

1. **Half Subtractor :-**

It is a combinational circuit which performs subtraction of two binary bits. It's has two inputs (minuend & subtrahend) and two outputs (difference & borrow).

Block diagram :-

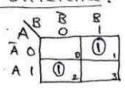



Truth table :-

| i/p's | | o/p's | |
|---|---|---|---|
| A | B | Diff. | Borrow |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

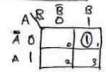
K-Map Simplification :-

for Difference :-



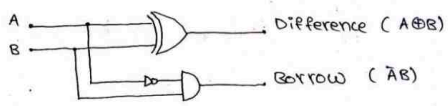

$\therefore$ Diff. $= \bar{A}B + A\bar{B}$

for Borrow :-




$\therefore$ Borrow $= \bar{A}B$

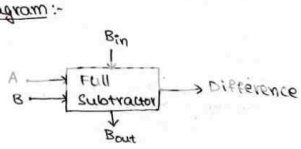
logic circuit :-




2. **Full Subtractor :-**

Full Subtractor is a combinational circuit which performs subtraction of 2 binary bits by considering borrow of the previous stage. It has 3 inputs and 2 outputs.

Block diagram :-




Truth table :-

| i/p's | | | o/p's | |
|---|---|---|---|---|
| A | B | $B_{in}$ | Diff | $B_{out}$ |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

## K-map simplifications :-

For Difference :-

| A \ $B B_{in}$ | $\bar{B}\bar{B}_{in}$ 00 | $\bar{B}B_{in}$ 01 | $B B_{in}$ 11 | $B\bar{B}_{in}$ 10 |
|---|---|---|---|---|
| $\bar{A}$ 0 | 0 | 1 | 0 | 1 |
| A 1 | 1 | 0 | 1 | 0 |

$\therefore$ $\boxed{\text{Difference} = A \oplus B \oplus B_{in}}$

$\therefore$ Difference $= \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + ABB_{in}$

$= \bar{A}(\bar{B}B_{in} + B\bar{B}_{in}) + A(\bar{B}\bar{B}_{in} + BB_{in})$

$= \bar{A}(B \oplus B_{in}) + A(B \odot B_{in}) = \bar{A}(B \oplus B_{in}) + A(\overline{B \oplus B_{in}})$

$= A \oplus B \oplus B_{in} = A \oplus B \oplus B_{in}$.

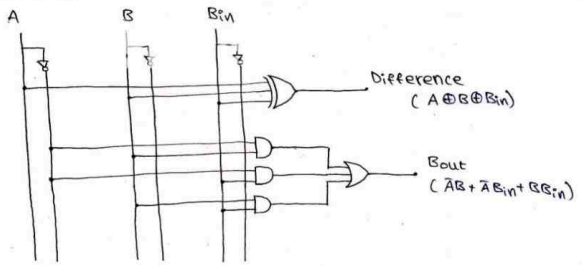
For $B_{out}$ :-

| A \ $B B_{in}$ | $\bar{B}\bar{B}_{in}$ 00 | $\bar{B}B_{in}$ 01 | $B B_{in}$ 11 | $B\bar{B}_{in}$ 10 |
|---|---|---|---|---|
| $\bar{A}$ 0 | | 1 | 1 | 1 |
| A 1 | | | 1 | |

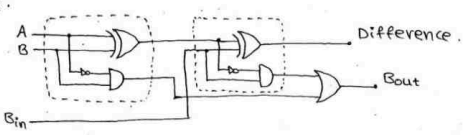
$\therefore$ $\boxed{B_{out} = \bar{A}B + \bar{A}B_{in} + B B_{in}}$

$\therefore$ $B_{out} = \bar{A}B + \bar{A}B_{in} + BB_{in}$

logic circuit :-




Note :- A full subtractor can also be implemented with two half - subtractors and 1 OR gate.




$\therefore$ Difference $= A \oplus B \oplus B_{in}$.

$\therefore$ $B_{out} = (\overline{A \oplus B})B_{in} + \bar{A}B$

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

$= \bar{A}\bar{B}B_{in} + ABB_{in} + \bar{A}B$

$= \bar{A}\bar{B}B_{in} + B(AB_{in} + \bar{A})$ $\quad (\because A + BC = (A+B)(A+C))$

$= \bar{A}\bar{B}B_{in} + B((A+\bar{A})(B_{in}+\bar{A}))$ $\quad (\because A + \bar{A} = 1)$

$= \bar{A}\bar{B}B_{in} + BB_{in} + B\bar{A}$

$= B_{in}(\bar{A}\bar{B} + B) + B\bar{A}$ $\quad (\because A + BC = (A+B)(A+C))$

$= B_{in}((\bar{A}+B)(B+\bar{B})) + B\bar{A}$ $\quad (\because B + \bar{B} = 1)$

$= \bar{A}B_{in} + BB_{in} + B\bar{A}$.



---

Ex: Design 4-bit (binary) parallel subtractor using full adders.

Sol: A, B $\rightarrow$ 4-bit binary no.'s.

$A = A_3\ A_2\ A_1\ A_0$

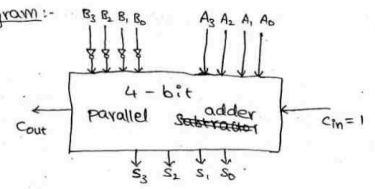
$B = B_3\ B_2\ B_1\ B_0$

$A - B = A +$ 2's complement of B

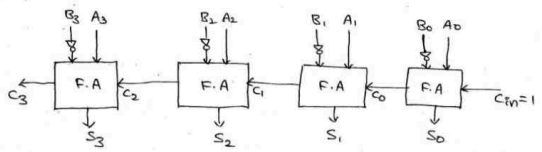
$= A +$ 1's comp. of B $+ 1$

$$
\begin{array}{rccccl}
 & c_2 & c_1 & c_0 & & \\
 & A_3 & A_2 & A_1 & A_0 & \longrightarrow \text{Minuend} \\
 & \bar{B}_3 & \bar{B}_2 & \bar{B}_1 & \bar{B}_0 & \longrightarrow \text{1's comp. of B} \\
+ & & & & 1 & \longrightarrow 1. \\
\hline
 & c_3\ S_3 & S_2 & S_1 & S_0 &
\end{array}
$$

Block diagram :-



logic diagram :-




Parallel adder / Subtractor (or) Binary adder / Subtractor :-

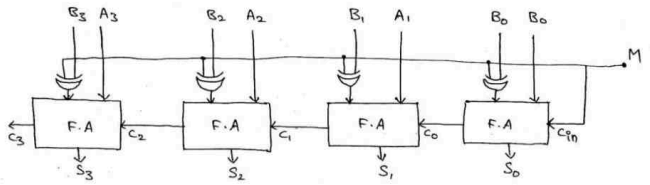



Fig:- 4-bit parallel adder / subtractor.
where M - controlling input.

if $M = 0$ then output $= A_3A_2A_1A_0 + B_3B_2B_1B_0 + C_{in}$ $(\because A \oplus 0 = A)$

$= A_3A_2A_1A_0 + B_3B_2B_1B_0$ $(\because C_{in} = 0)$ $M=0$

$= A + B.$

$\therefore$ (i.e) this circuit act as adder.

If $M = 1$ then output $= A_3A_2A_1A_0 + \overline{B_3}\,\overline{B_2}\,\overline{B_1}\,\overline{B_0} + C_{in}$ $(\because A \oplus 1 = \bar{A})$

$= A_3A_2A_1A_0 + \overline{B_3}\,\overline{B_2}\,\overline{B_1}\,\overline{B_0} + 1$ $(\because M = 1 \Rightarrow C_{in} = 0)$

$= A - B$

$\therefore$ (i.e) this circuit act as Subtractor.

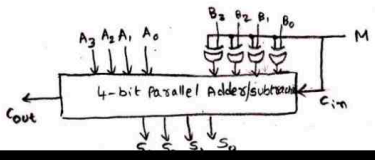



fig: Block diag

Look ahead carry generator :-

* The sum and carry outputs of any stage of n-bit parallel adder cannot be produced until the input carry occurs. This leads to time delay in addition process. This delay is known as "carry propagation delay."

* LACG is used to eliminate the carry delay.

* LAC addition uses two functions.
  (i) carry generate ($G_i$)
  (ii) carry propagate ($P_i$)

$$\begin{array}{rrrr} & C_2 & C_1 & C_0 \\ & A_2 & A_1 & A_0 \\ + & B_2 & B_1 & B_0 \\ \hline C_3 & S_2 & S_1 & S_0 \end{array}$$

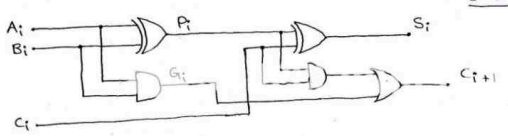



Fig:- FA using 2 HA's and OR gate with P & G shown.

$\therefore$ $P_i = A_i \oplus B_i$

$G_i = A_i \cdot B_i$

using P & G, sum & carry can be expressed as

$S_i = P_i \oplus C_i$

$C_{i+1} = G_i + (P_i C_i)$

$\therefore$ $C_0 = $ i/p carry.

$C_1 = G_0 + P_0 \cdot C_0$

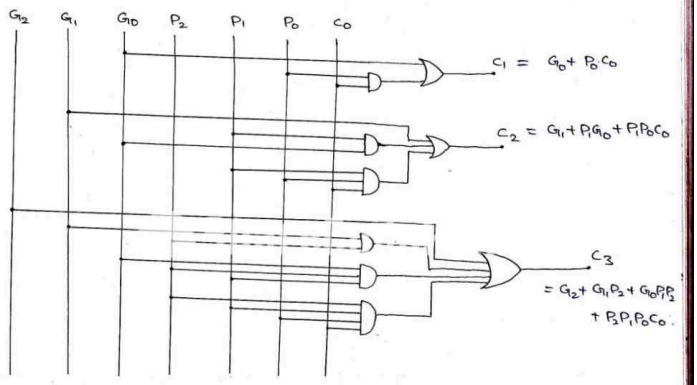
$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0)$

$= G_1 + P_1 \cdot G_0 + P_1 P_0 C_0$

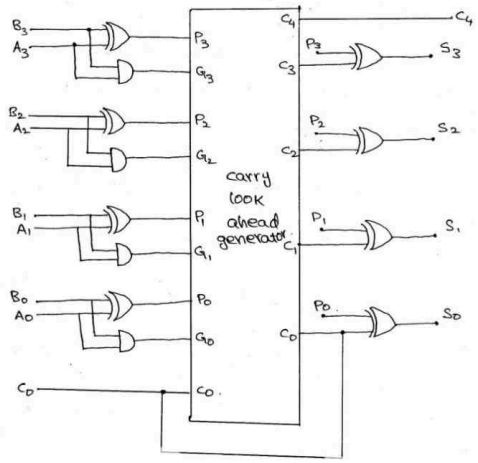
$$\therefore C_3 = G_2 + P_2C_2$$
$$= G_2 + P_2(G_1 + P_1G_0 + P_1P_0C_0)$$
$$= G_2 + G_1P_2 + G_0P_1P_2 + P_2P_1P_0C_0$$

From the above boolean equations (Expressions) it can be seen that $C_1, C_2, C_3$ propagate at the same time and depends only on $C_0$.

3-bit look ahead carry generator logic diagram :-




4-bit adder with carry look ahead :-






* BCD adder / Decimal adder :-

* The digital systems handles the decimal no. in the form of BCD number.

* A BCD adder is a circuit that adds two BCD digits and produces a sum digit also in BCD.

* To implement BCD adder we required,

(i) 4-bit binary adder for initial Addition.

(ii) logic circuit to detect sum is greater then 9 (or) not.

(iii) 1 more 4-bit binary adder to add $(0110)_2$

to the sum if sum is >9 or carry is 1.

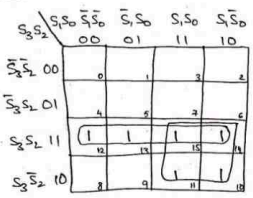
Truth table to implement logic circuit to detect sum is >9 or not :-

| i/p's | | | | o/p |
|---|---|---|---|---|
| $S_3$ | $S_2$ | $S_1$ | $S_0$ | 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 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

o/p is '1' when i/p is >9.

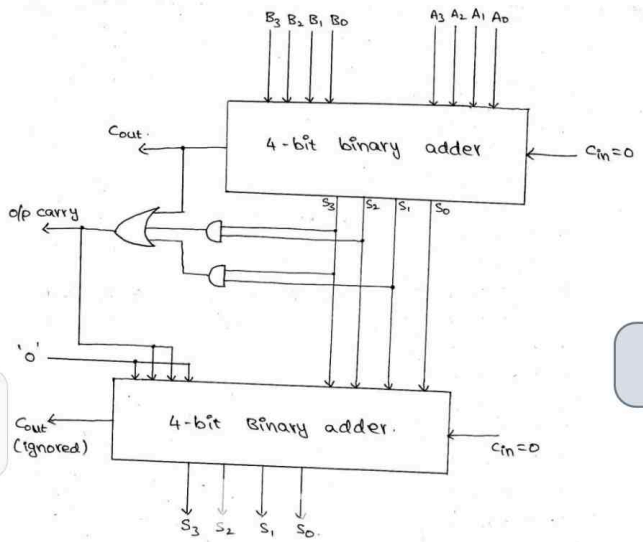
K-Map simplification :-

for y :-




$\therefore\ y = S_3S_2 + S_3S_1$

logic diagram :-

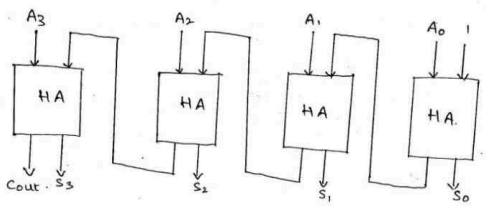



* 4-bit Binary Incrementer :-

$A = A_3A_2A_1A_0$ 4-bit binary no.

$A+1 = A_3A_2A_1A_0 + 1$

$$\begin{array}{r} c_2\ c_1\ c_0 \\ A_3\ A_2\ A_1\ A_0 \\ +\quad 1 \\ \hline C_{out}\ S_3\ S_2\ S_1\ S_0 \end{array}$$



## * 4-bit Binary Decrementer :-

$A = A_3 A_2 A_1 A_0$ 4-bit Binary number.

$A - 1 = A_3 A_2 A_1 A_0 - 0001$

$= A_3 A_2 A_1 A_0 +$ 2's comp. of 0001

$= A_3 A_2 A_1 A_0 +$ 1's comp. of 0001 + 1

$= A_3 A_2 A_1 A_0 + 1110 + 1$

## ** Binary Multiplier :-

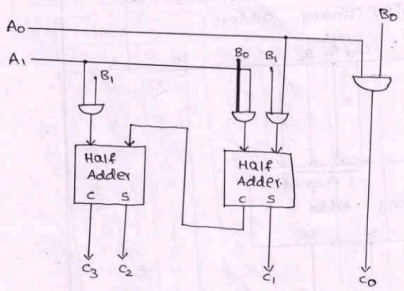
### (i) 2 X 2 Binary Multiplier :-

$B = B_1 B_0$ $A = A_1 B_0$ 2-bit binary no.'s.

(B X A = Simple multiplication)

```
B X A ⇒      B1  B0   → Multiplicand
           X A1  A0   → Multiplier
           ----------
             A0B1 A0B0 } Partial products.
          A1B1 A1B0
          ----------
          C3 C2 C1 C0  → Product.
```

logic diagram :-




### (ii) 4 X 3 Binary Multiplier :-

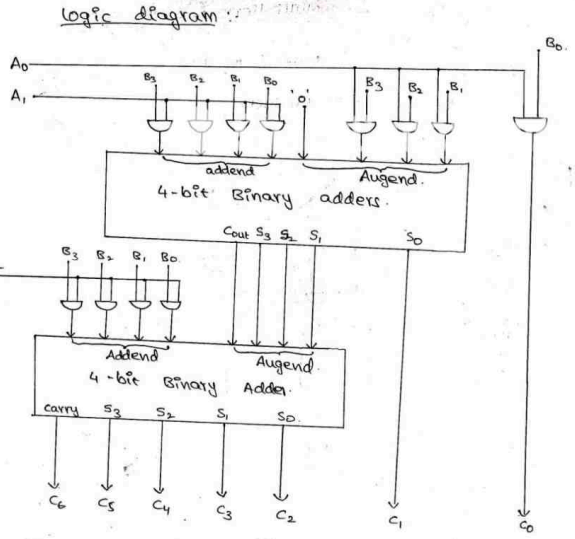
$B = B_3 B_2 B_1 B_0$ → 4-bit Binary no.

$A = A_2 A_1 A_0$ → 3-bit Binary no.

B X A

```
      B3  B2  B1  B0   → Multiplicand
    X A2  A1  A0       → Multiplier
    ------------------
      A0B3 A0B2 A0B1 A0B0 }
   A1B3 A1B2 A1B1 A1B0    } Partial products.
 A2B3 A2B2 A2B1 A2B0      }
 ------------------------
 C6 C5 C4 C3 C2 C1 C0   → Product
```

logic diagram :-



16/08/2014

## * Magnitude comparator (Digital comparator) :-

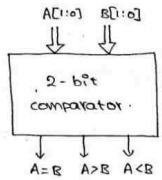
* A comparator is a special combinational circuit designed primarily to compare the relative magnitudes of two Binary numbers.

* It receives two n-bit numbers, A & B are inputs and produces 3 outputs A>B, A<B, A=B.

**Design 2-bit comparator using logic gates.**

2-bit comparator compares 2, 2-bit binary numbers.

**Block diagram:-**




A, B are 2 bit Binary numbers.

**Truth table:-**

| | i/p's | | | | o/p's | | |
|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_0$ | $B_1$ | $B_0$ | A=B | A>B | A<B |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 12 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 13 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 16 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

$(A=B) = \Sigma m(0,5,10,15)$

$(A>B) = \Sigma m(4,8,9,12,13,14)$

$(A<B) = \Sigma m(1,2,3,6,7,11)$

**K-map simplification :-**

**For A=B :-**

$(A=B) = \Sigma m(0,5,10,15)$

| $A_1A_0 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | | |
| 01 | | 1 | | |
| 11 | | | 1 | |
| 10 | | | | 1 |

$(A=B) = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$

$= \bar{A}_0\bar{B}_0(\bar{A}_1\bar{B}_1 + A_1B_1) + A_0B_0(\bar{A}_1\bar{B}_1 + A_1B_1)$

$= (\bar{A}_1\bar{B}_1 + A_1B_1)(\bar{A}_0\bar{B}_0 + A_0B_0).$

$\therefore \boxed{(A=B) = (A_0 \odot B_0)(A_1 \odot B_1)}$

**For (A>B) :-**

$(A>B) = \Sigma m(4,8,9,12,13,14)$

| $A_1A_0 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | | | |
| 11 | 1 | 1 | | 1 |
| 10 | 1 | 1 | | |

$(A>B) = A_1\bar{B}_1 + \bar{B}_1\bar{B}_0A_0 + A_1A_0\bar{B}_0$

$\therefore \boxed{(A>B) = A_1\bar{B}_1 + A_1A_0\bar{B}_0 + A_0\bar{B}_1\bar{B}_0}$

**For (A<B) :-**

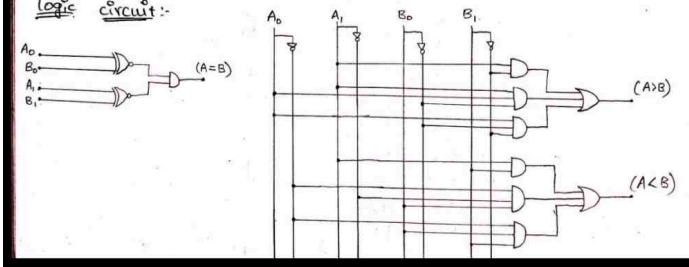
$(A<B) = \Sigma m(1,2,3,6,7,11)$

| $A_1A_0 \backslash B_1B_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | 1 | 1 | |
| 01 | | | 1 | 1 |
| 11 | | | | |
| 10 | | | 1 | |

$(A<B) = \bar{A}_1B_1 + B_0\bar{A}_1\bar{A}_0 + \bar{B}_1B_0\bar{A}_0$

$\therefore \boxed{(A<B) = \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0}$

**Logic circuit :-**








25/08/2014

26/08/2014

* **Decoder** :- A Decoder is a multiple input, multiple o/p circuit, which converts coded inputs into coded outputs, where the i/p & o/p codes are differents.

* the i/p code generally has fewer bits than them o/p codes.

logic diagram :-

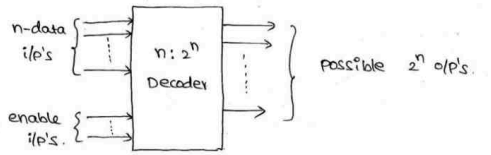
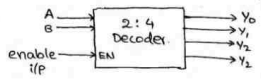



Fig:- General Structure of Decoder.

(i) 2 - to - 4 (2:4) Binary Decoder :-

Block diagram :-



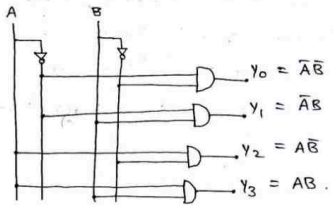

Truth table :-

| i/p's | | | o/p's | | | |
|---|---|---|---|---|---|---|
| EN | A | B | $Y_0$ | $Y_1$ | $Y_2$ | $Y_3$ |
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

(∵ enable i/p is '0' then there is no i/p's.)

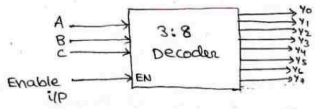
from the above truth table, $Y_0 = \bar{A}\bar{B}$, $Y_1 = \bar{A}B$, $Y_2 = A\bar{B}$, $Y_3 = AB$ (∵ K-map Simplification)

logic diagram :-




(ii) 3 - to - 8 (3:8) Binary decoder :-

Block diagram :-




Truth table :-

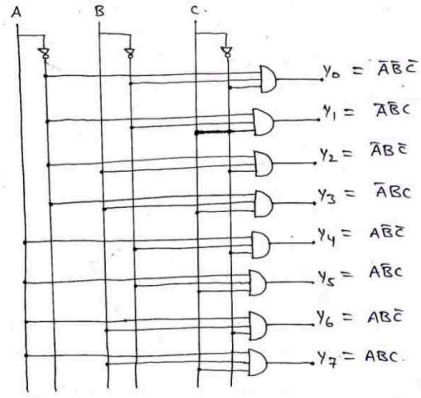
| i/p's | | | | o/p's | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EN | A | B | C | $Y_0$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_5$ | $Y_6$ | $Y_7$ |
| 0 | X | X | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

From the above truth table (using k-map simplification)

$Y_0 = \bar{A}\bar{B}\bar{C}$, $Y_1 = \bar{A}\bar{B}C$, $Y_2 = \bar{A}B\bar{C}$, $Y_3 = \bar{A}BC$

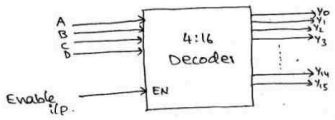
$Y_4 = A\bar{B}\bar{C}$, $Y_5 = A\bar{B}C$, $Y_6 = AB\bar{C}$, $Y_7 = ABC$

logic diagram :-




Ex:- construct 4:16 decoders using 3:8 decoders.

Sol:- Block diagram of 4:16 decoder :-




Truth table of 4:16 decoder :-

| i/p's | | | | o/p |
|---|---|---|---|---|
| A | B | C | D | Y(High) |
| 0 | 0 | 0 | 0 | $Y_0$ |
| 0 | 0 | 0 | 1 | $Y_1$ |
| 0 | 0 | 1 | 0 | $Y_2$ |
| 0 | 0 | 1 | 1 | $Y_3$ |
| 0 | 1 | 0 | 0 | $Y_4$ |
| 0 | 1 | 0 | 1 | $Y_5$ |
| 0 | 1 | 1 | 0 | $Y_6$ |
| 0 | 1 | 1 | 1 | $Y_7$ |
| 1 | 0 | 0 | 0 | $Y_8$ |
| 1 | 0 | 0 | 1 | $Y_9$ |
| 1 | 0 | 1 | 0 | $Y_{10}$ |
| 1 | 0 | 1 | 1 | ⋮ |
| 1 | 1 | 0 | 0 | |

4:16 Decoder using 3:8 Decoders :-

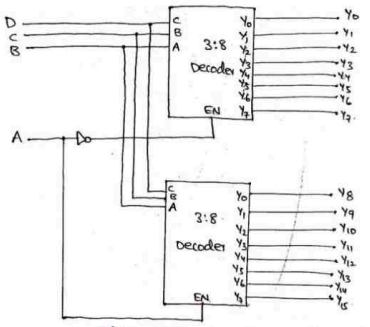
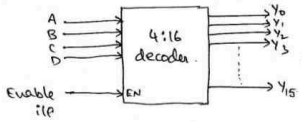



Fig:- 4:16 decoder using 3:8 decoders.

Ex:- construct 4:16 Decoder using 2:4 decoder.

Sol:- Block diagram of 4:16 decoder :-




Truth table of 4:16 decoder :-

| i/p's | | | | o/p |
|---|---|---|---|---|
| A | B | C | D | Y(High) |
| 0 | 0 | 0 | 0 | $Y_0$ |
| 0 | 0 | 0 | 1 | $Y_1$ |
| 0 | 0 | 1 | 0 | $Y_2$ |
| 0 | 0 | 1 | 1 | $Y_3$ |
| 0 | 1 | 0 | 0 | $Y_4$ |
| 0 | 1 | 0 | 1 | $Y_5$ |
| | ⋮ | | | ⋮ |
| 1 | 1 | 1 | 0 | $Y_{14}$ |
| 1 | 1 | 1 | 1 | $Y_{15}$ |

## 4:16 decoder using 2:4 decoders :-

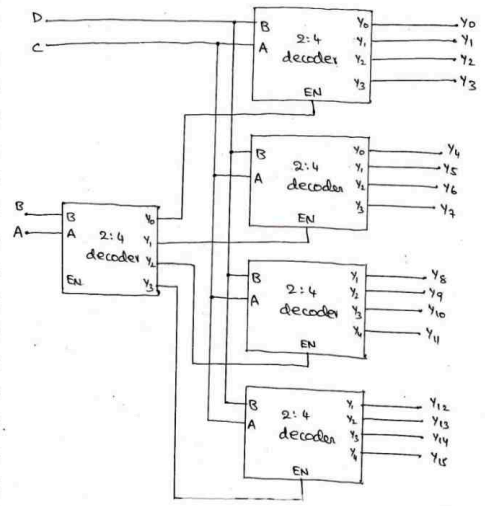
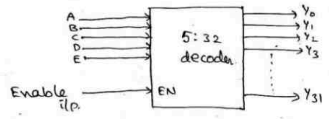



Fig:- 4:16 decoder using 2:4 decoders.

Ex:- construct 5:32 decoder using 4, 3:8 decoders & 1, 2:4 decoder

Sol:- Block diagram of 5:32 decoder:-




Truth table of 5:32 decoder:-

| i/p's | | | | | o/p |
|---|---|---|---|---|---|
| A | B | C | D | E | Y (high) |
| 0 | 0 | 0 | 0 | 0 | $Y_0$ |
| 0 | 0 | 0 | 0 | 1 | $Y_1$ |
| 0 | 0 | 0 | 1 | 0 | $Y_2$ |
| | | ⋮ | | | ⋮ |
| 1 | 1 | 1 | 1 | 0 | $Y_{30}$ |

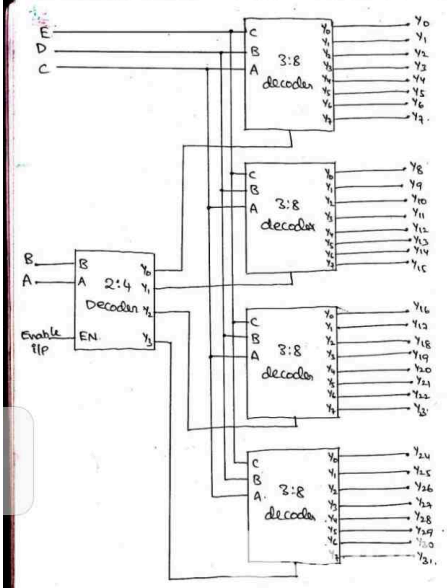



Fig:- 5:32 decoder using 4, 3:8 decoders & 1, 2:4 decoders.



* Realization of Boolean Expressions using Binary

→ The combination of decoder and external logic gates can be used to implement single or multiple output functions.

→ The decoder generates minterms for input variables. Thus by logically oring specified minterms we can implement the given function.

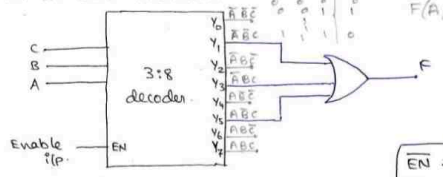
**Ex:-** $F(A,B,C) = \Sigma m(1,3,5)$ Implement by using decoder.

for SOM write Truth table

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

$F(A,B,C) = \overset{001}{\bar{A}\bar{B}C} + \overset{011}{\bar{A}BC} + \overset{101}{A\bar{B}C}$

**Sol:-**

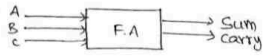



$\overline{EN} = 0 \rightarrow$ Active

$\overline{EN} = 1 \rightarrow$ disable.

**Ex:-** Implement full adder circuit using decoder.

**Sol:-** Block diagram of full adder :-



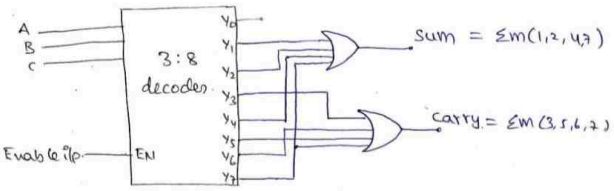

Truth table of full adder:-

| i/p's | | | o/p's | |
|---|---|---|---|---|
| A | B | C | sum | carry |
| 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 |

$\therefore$ Sum $= \Sigma m(1,2,4,7)$

carry $= \Sigma m(3,5,6,7) = \Sigma m(3,5,6,7)$

Full adder using Decoder & external gates :-

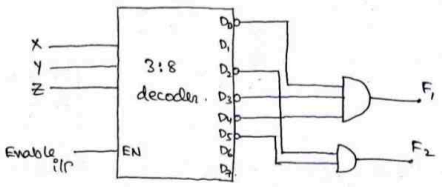



**Ex:-** Implement the following boolean expressions using decoder.

$F_1(X,Y,Z) = \pi M(0,3,4)$

$F_2(X,Y,Z) = \pi M(2,5)$

**Sol:-**




from the above circuit,

$F_1 = \pi M(0,3,4)$

$F_2 = \pi M(2,5)$.



02/09/2014

*** BCD to 7-Segment Display Decoder :-**

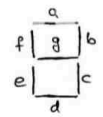



fig:- 7-segment display.

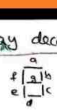
→ there are two types of 7-segment displays.

(i) common anode → segment i/p must be active low to glow the segment.

(ii) common cathode → segment i/p must be active high to glow the segment.

→ Most commonly used one is common cathode type.

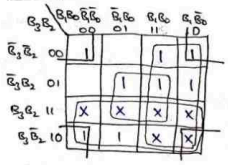
Truth table of common cathode 7-segment display decoder:-




| Decimal | i/p (BCD) $B_3$ $B_2$ $B_1$ $B_0$ | o/p a b c d e f g | |
|---|---|---|---|
| 0 | 0 0 0 0 | 1 1 1 1 1 1 0 | 0 |
| 1 | 0 0 0 1 | 0 1 1 0 0 0 0 | 1 |
| 2 | 0 0 1 0 | 1 1 0 1 1 0 1 | 2 |
| 3 | 0 0 1 1 | 1 1 1 1 0 0 1 | 3 |
| 4 | 0 1 0 0 | 0 1 1 0 0 1 1 | 4 |
| 5 | 0 1 0 1 | 1 0 1 1 0 1 1 | 5 |
| 6 | 0 1 1 0 | 1 0 1 1 1 1 1 | 6 |
| 7 | 0 1 1 1 | 1 1 1 0 0 0 0 | 7 |
| 8 | 1 0 0 0 | 1 1 1 1 1 1 1 | 8 |
| 9 | 1 0 0 1 | 1 1 1 1 0 1 1 | 9 |

K-Map Simplification

for a :-




$\therefore \quad a = B_3 + B_2B_0 + B_1 + \bar{B}_2\bar{B}_0$

Similarly, $b =$

$c =$

$d =$

$e =$

$f =$

$g =$

Logic diagram:

* ENCODER :-

An encoder is a digital circuit that performs the inverse operation of the decoder. It has "$2^n$" (or) fewer) input lines and 'n' output lines. The o/p lines generate the binary code corresponding to the i/p value.

Block diagram :-

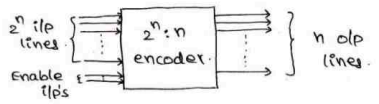


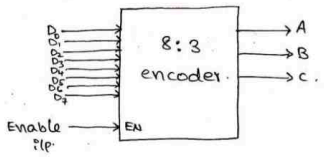

Fig:- General structure of Encoder.

Octal to Binary Encoder :- (8-to-3) (8:3 Encoder)

→ It has 8 inputs. (i.e) 1 for each octal digit and 3 o/p's that generate the corresponding Binary code.

→ In Encoders it is assumed that only one i/p has a value of '1' at any given time, otherwise the circuit is meaningless.

Block diagram :-



Truth table (or) Function table :-

| | i/p's | | | | | | | | o/p's | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | A | B | C |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| EN = 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | X | X | X |

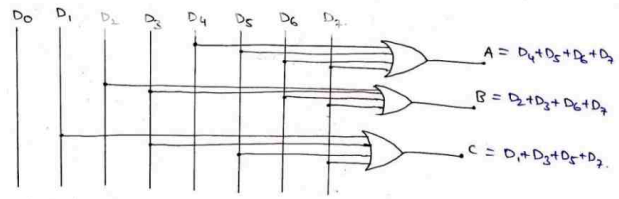
→ if Enable i/p is '0' there is no o/p.

$\therefore$ $A = D_4 + D_5 + D_6 + D_7$

$B = D_2 + D_3 + D_6 + D_7$

$C = D_1 + D_3 + D_5 + D_7$

logic diagram:-




* Priority Encoder:- ( 4-bit priority encoder)

A priority encoder is a encoder circuit that includes the priority function. In priority encoder, if two or more i/p's are equal to '1' at the same time, the i/p having the highest priority will take precedance.

Truth table:-

| | i/p's | | | | o/p's | | |
|---|---|---|---|---|---|---|---|
| | $D_0$ | $D_1$ | $D_2$ | $D_3$ | $Y_1$ | $Y_0$ | V |
| | 0 | 0 | 0 | 0 | X | X | 0 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 01 | X | 1 | 0 | 0 | 0 | 1 | 1 |
| 10 | X | X | 1 | 0 | 1 | 0 | 1 |
| 11 | X | X | X | 1 | 1 | 1 | 1 |

→ $D_3$ i/p has highest priority and $D_0$ i/p has lowest priority. when $D_3$ i/p is high, regardless of other i/p's output is 11.

→ 'V' is a valid bit indicator, that is set to '1' when 1 or more i/p's are equal to "1".

K-map simplifications:-

for $Y_1$ :-

| $D_0D_1$ \ $D_2D_3$ | $\bar{D}_2\bar{D}_3$ 00 | $\bar{D}_2D_3$ 01 | $D_2D_3$ 11 | $D_2\bar{D}_3$ 10 |
|---|---|---|---|---|
| $\bar{D}_0\bar{D}_1$ 00 | X | 1 | 1 | 1 |
| $\bar{D}_0D_1$ 01 | | 1 | 1 | 1 |
| $D_0D_1$ 11 | | 1 | 1 | 1 |
| $D_0\bar{D}_1$ 10 | | 1 | 1 | 1 |

$\therefore$ $Y_1 = D_2 + D_3$

for $Y_0$ :-

| $D_0D_1$ \ $D_2D_3$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | |
| 01 | 1 | 1 | 1 | |
| 11 | 1 | 1 | 1 | |
| 10 | | 1 | 1 | |

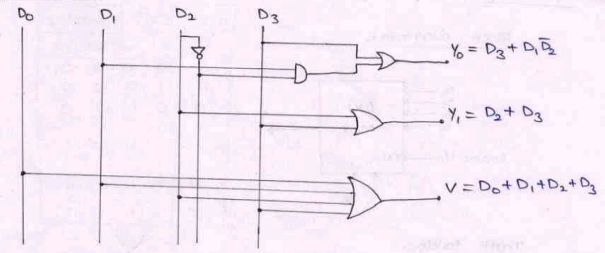
$\therefore$ $Y_0 = D_3 + D_1\bar{D}_2$

for V :-

| $D_0D_1$ \ $D_2D_3$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | 1 | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | | 1 | 1 |

$\therefore$ $V = D_0 + D_1 + D_2 + D_3$

logic circuit :-



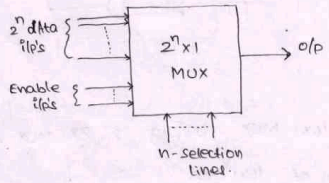

## * Multiplexer (or) Data selector :-

→ Multiplexer is a combinational logic circuit that selects binary information from one of many input lines and directs it to a single o/p line.

→ The selection of particular i/p line is controled by a set of selection lines.

→ MUX has $2^n$ i/p lines, n selection lines & one o/p.

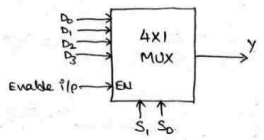
Block diagram :-




03/10/2014

## 4x1 MUX :-

Block diagram :-

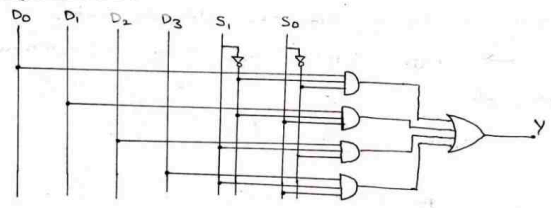



Truth table :-

| i/p's | | | o/p |
|---|---|---|---|
| EN | $S_1$ | $S_0$ | Y |
| 0 | x | x | x |
| 1 | 0 | 0 | $D_0$ |
| 1 | 0 | 1 | $D_1$ |
| 1 | 1 | 0 | $D_2$ |
| 1 | 1 | 1 | $D_3$ |

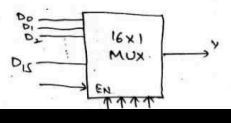
$\therefore Y = D_0\bar{S}_1\bar{S}_0 + D_1\bar{S}_1 S_0 + D_2 S_1\bar{S}_0 + D_3 S_1 S_0$

logic circuit :-




Ex:- Construct a 16x1 MUX using 2, 8x1 MUX & 1, ~~2x1 MUX~~

Sol:- Block diagram of 16x1 MUX :-



Truth table :-

| selection i/p's | | | | o/p |
|---|---|---|---|---|
| $S_3$ | $S_2$ | $S_1$ | $S_0$ | Y |
| 0 | 0 | 0 | 0 | $D_0$ |
| 0 | 0 | 0 | 1 | $D_1$ |
| 0 | 0 | 1 | 0 | $D_2$ |
| ⋮ | | | | ⋮ |
| 1 | 1 | 1 | 0 | $D_{14}$ |
| 1 | 1 | 1 | 1 | $D_{15}$ |

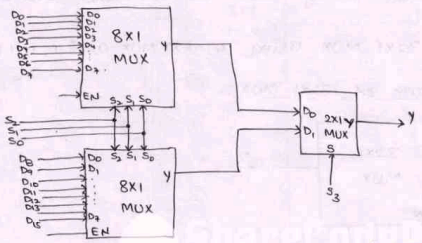
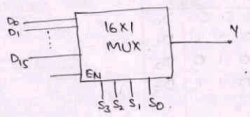



Fig:- 16X1 MUX using two 8X1 MUX & one 2X1 MUX.

Ex:- construct a 16X1 MUX using two 8X1 MUX and OR gate.

Sol:- Block diagram of 16X1 MUX :-




Truth table of 16X1 MUX :-

| selection i/p's | | | | o/p |
|---|---|---|---|---|
| $S_3$ | $S_2$ | $S_1$ | $S_0$ | Y |
| 0 | 0 | 0 | 0 | $D_0$ |
| 0 | 0 | 0 | 1 | $D_1$ |
| ⋮ | | | | ⋮ |
| 1 | 1 | 1 | 0 | $D_{14}$ |



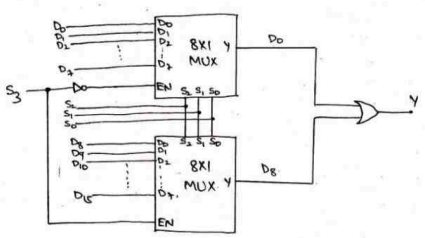
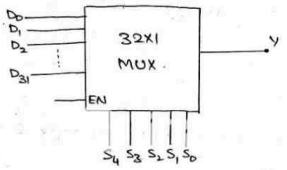



fig:- 16X1 MUX using two 8X1 MUX and one OR gate.

Ex:- construct a 32X1 MUX using 4, 8X1 MUX and 1, 4X1 MUX.

Sol:- Block diagram of 32X1 MUX :-




Truth table of 32X1 MUX :-

| selection i/p's | | | | | o/p |
|---|---|---|---|---|---|
| $S_4$ | $S_3$ | $S_2$ | $S_1$ | $S_0$ | Y |
| 0 | 0 | 0 | 0 | 0 | $D_0$ |
| 0 | 0 | 0 | 0 | 1 | $D_1$ |
| 0 | 0 | 0 | 1 | 0 | $D_2$ |
| 0 | 0 | 0 | 1 | 1 | $D_3$ |
| ⋮ | | | | | ⋮ |
| 1 | 1 | 1 | 0 | 1 | $D_{29}$ |
| 1 | 1 | 1 | 1 | 0 | $D_{30}$ |
| 1 | 1 | 1 | 1 | 1 | $D_{31}$ |

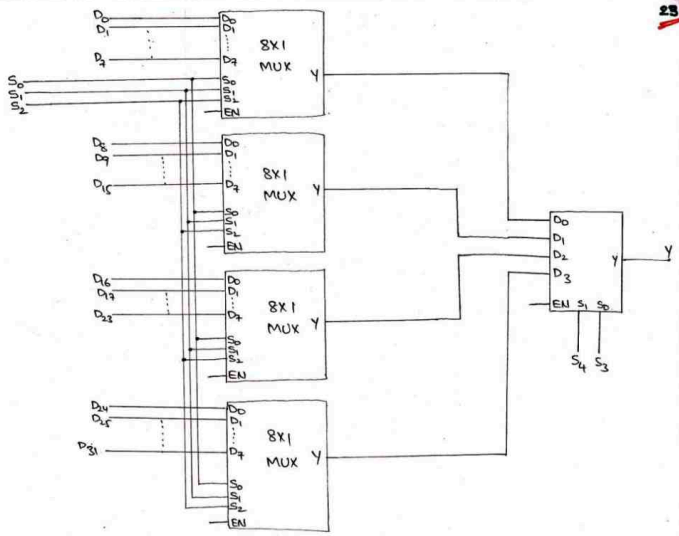
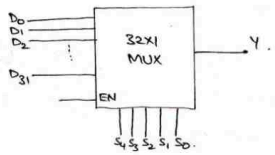



Fig:- 32x1 MUX using four 8x1 MUX & one 4x1 MUX.

Ex:- construct 32x1 MUX 4, 8x1 MUX & 1, 2-to-4 decoder, 1 OR-gate.

Sol:- Block diagram of 32x1 MUX :-




Truth table of 32x1 MUX :-

| Selection i/p's | o/p |
|---|---|
| $S_4$ $S_3$ $S_2$ $S_1$ $S_0$ | Y |
| 0 0 0 0 0 | $D_0$ |
| 0 0 0 0 1 | $D_1$ |
| ⋮ | ⋮ |
| 1 1 1 1 0 | $D_{30}$ |

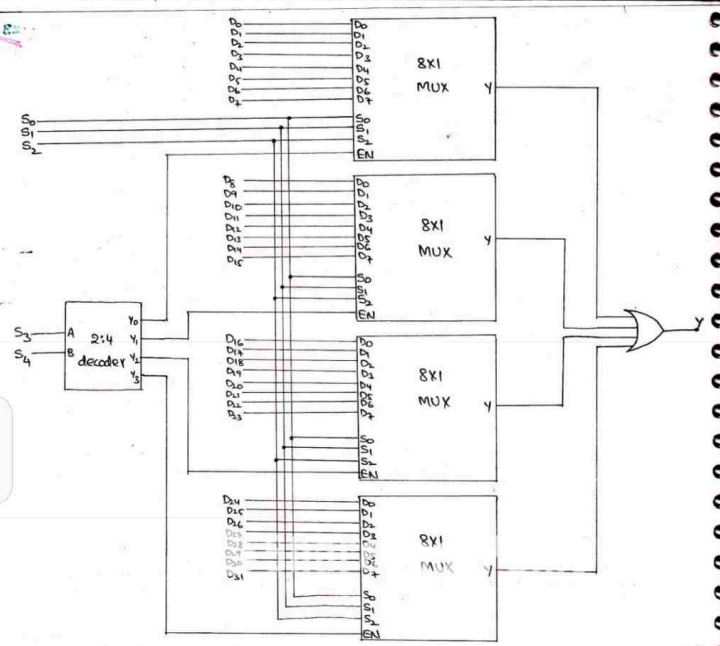



Fig:- 32x1 MUX using four 8x1 MUX, one 2:4 decoder & one OR-gate.

* Implementation of combinational logic using Multiplexer:-

Ex:- Implement the following boolean function using

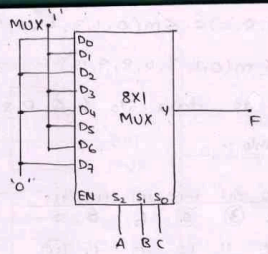
(a) 8x1 MUX (b) 4x1 MUX

(i) $F(A,B,C) = \Sigma m(1,3,5,6)$

Sol:- Truth table:-

| i/p's | o/p |
|---|---|
| A B C | F |
| 0 0 0 | 0 |
| 0 0 1 | 1 |
| 0 1 0 | 0 |
| 0 1 1 | 1 |
| 1 0 0 | 0 |
| 1 0 1 | 1 |

(i) using 8X1 MUX

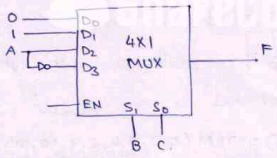



(ii) using 4X1 MUX

→ considering 'A' as data i/p & B,C as selection i/p's.

implementation table :-

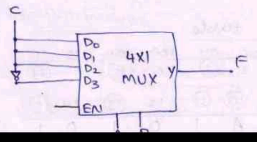
| A \ BC | $D_0$ $\bar{B}\bar{C}$ 00 | $D_1$ $\bar{B}C$ 01 | $D_2$ $B\bar{C}$ 10 | $D_3$ $BC$ 11 |
|---|---|---|---|---|
| $\bar{A}$ 0 | 0 | 1 | 2 | 3 |
| A 1 | 4 | 5 | 6 | 7 |
| | 0 | 1 | A | $\bar{A}$ |




→ considering 'C' as data i/p & A,B as selection i/p's.

implementation table :-

| C \ AB | $D_0$ $\bar{A}\bar{B}$ 00 | $D_1$ $\bar{A}B$ 01 | $D_2$ $A\bar{B}$ 10 | $D_3$ $AB$ 11 |
|---|---|---|---|---|
| $\bar{C}$ 0 | 0 | 2 | 4 | 6 |
| C 1 | 1 | 3 | 5 | 7 |
| | C | C | C | $\bar{C}$ |



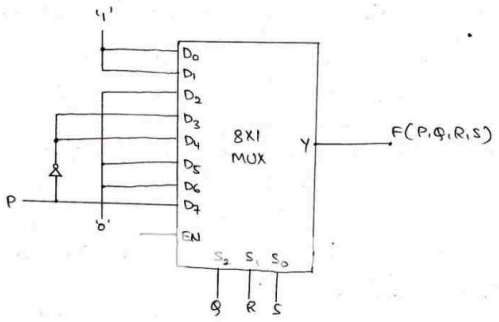

Ex: Implement $F(P,Q,R,S) = \Sigma m(0,1,3,4,8,9,15)$ using 8X1 MUX.

Sol:- $F(P,Q,R,S) = \Sigma m(0,1,3,4,8,9,15)$.

considering 'P' as data i/p & Q,R,S as Selection i/p's.

Implementation table :-

| P \ QRS | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| $\bar{P}$ 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| P 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | 1 | 1 | 0 | $\bar{P}$ | $\bar{P}$ | 0 | 0 | P |




Ex:- Implement $F(A,B,C,D) = \Pi M(0,1,2,4,6,9,12,14)$ using 8X1 MUX.

Sol:- $F(A,B,C,D) = \Pi M(0,1,2,4,6,9,12,14)$.

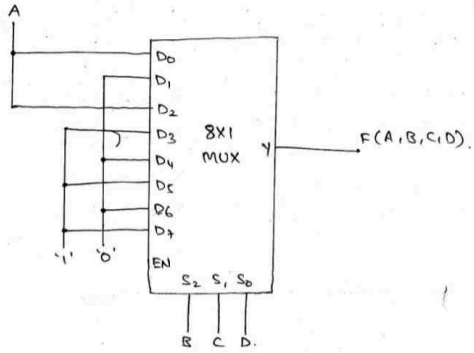
considering 'A' as data i/p & B,C,D as selection i/p's.

Here instead of minterms, maxterms are specified. thus we have to circle maxterms which are not included in the boolean function.

$\therefore F(A,B,C,D) = \Pi M(0,1,2,4,6,9,12,14) = \Sigma m(3,5,7,8,10,11,13,15)$

implementation table :-

| A \ BCD | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| $\bar{A}$ 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| A 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | A | 0 | A | 1 | 0 | 1 | 0 | 1 |

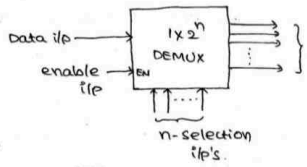



05/09/2014

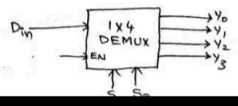
## Demultiplexer (or) Data distributor :-

A demultiplexer is a combinational circuit that receives information on a single i/p line and transmits that information on one of $2^n$ possible o/p lines. The selection of specific o/p line is controlled by the values of 'n' selection lines.

**Block diagram :-**




(i) 1x4 demultiplexer




**Truth table :-**

| i/p's | | | o/p's | | | |
|---|---|---|---|---|---|---|
| EN | $S_1$ | $S_0$ | $Y_3$ | $Y_2$ | $Y_1$ | $Y_0$ |
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | $D_{in}$ |
| 1 | 0 | 1 | 0 | 0 | $D_{in}$ | 0 |
| 1 | 1 | 0 | 0 | $D_{in}$ | 0 | 0 |
| 1 | 1 | 1 | $D_{in}$ | 0 | 0 | 0 |

$\therefore Y_0 = D_{in} \cdot \bar{S}_1 \bar{S}_0$

$Y_1 = D_{in} \cdot \bar{S}_1 S_0$

$Y_2 = D_{in} \cdot S_1 \bar{S}_0$

$Y_3 = D_{in} \cdot S_1 S_0$

**Ex:-** Construct 1x4 DEMUX using 2, 1x2 DEMUX.

**Sol:-**

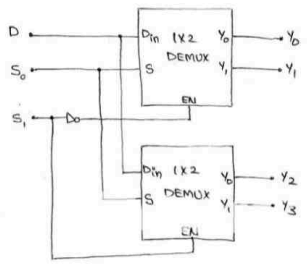



fig:- 1x4 DEMUX using two 1x2 demux.

09/09/2014

**Ex:-** Implement $F(w,x,y,z) = \Sigma m(1,2,5,6,9,11,13,15)$ using 4x1 MUX

**Sol:-** Considering w, x as data i/p's & y, z as section i/p's of 4x1 MUX.

Implementation table.

| wx \ yz | $D_0$ 00 | $D_1$ 01 | $D_2$ 10 | $D_3$ 11 |
|---|---|---|---|---|
| $\bar{w}\bar{x}$ 00 | 0 | 1 | 2 | 3 |
| $\bar{w}x$ 01 | 4 | 5 | 6 | 7 |
| $w\bar{x}$ 10 | 8 | 9 | 10 | 11 |
| $wx$ 11 | 12 | 13 | 14 | 15 |
| | 0 | 1 | $\bar{w}\bar{x}+\bar{w}x = \bar{w}$ | $w\bar{x}+wx = w$ |