# Combinational Circuits (DC-IV) MCQs & Numerical Problems

The problems considered here are put under the following five subtopics. These are:

  (a) Arithmetic circuits
  (b) Multiplexers
  (c) Decoders and encoders
  (d) Code converters
  (e) Comparators

Some problems have been taken from previous GATE examination and some other problems are added for practice.

Special efforts are put to arrange them in the increasing order of their complexity, so that it is easier to learn progressively.

Video solution to some typical Gate problems will be given separately.

## Arithmetic Circuits

**1.** For a binary half adder having two inputs A and B the correct set of logical expressions the output
$S(= A\ plus\ B)\ and\quad C(= carry)$
Are:  (also given logic diagram)
(a) $S = AB + \bar{A}B$ & $C = \bar{A}B$
(b) $S = \bar{A}B + A\bar{B}$ & $C = AB$
(c) $S = \bar{A}B + A\bar{B}$ & $C = \bar{A}B$
(d) $S = AB + \bar{A}\bar{B}$ & $C = A\bar{B}$

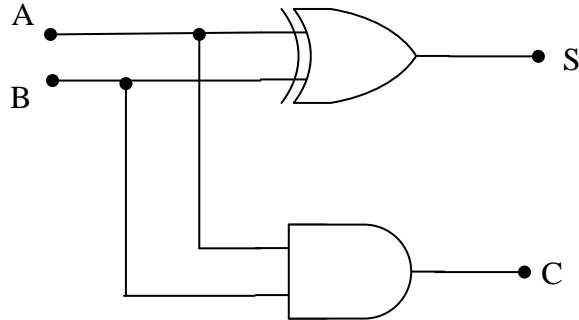**Sol.**     **Half adder:   logic circuit which adds two one bit number is called half adder.**

   **Truth table**

| Inputs | | Outputs | |
|---|---|---|---|
| A | B | S (sum) (carry) | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Expressions for sum and carry can be written from the truth table**

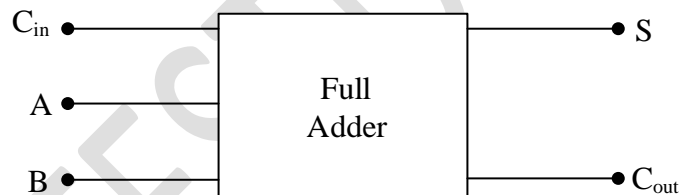$$S = \bar{A}B + A\bar{B} = (A \oplus B)$$

$$C = A.B$$



**Option (b)**

**2.** Find the expression for sum and carry for binary full adder

**Soln.** **Full adder: Half adder has only two inputs (no provision of carry from lower order bits) when multi bit addition is preformed full adder takes care of it. Block diagram is shown in the Fig.**



**It consists of 3 inputs and 2 outputs.**

**Truth table**

| Inputs | | | Outputs | |
|---|---|---|---|---|
| **A** | **B** | **$C_{in}$** | **S (sum)** | **$C_{out}$ (carry output)** |
| 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 |

$$S = \overline{A}\overline{B}\, C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in}$$

$$= \overline{A}(\overline{B}C_{in} + B\overline{C}_{in}) + A(\overline{B}\overline{C}_{in} + BC_{in})$$

$$S = \overline{A}(B \oplus C_{in}) + A\overline{(B \oplus C_{in})}$$

$$S = A \oplus B \oplus C_{in} \qquad\qquad\qquad (1)$$

$$C_{out} = \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} + ABC_{in}$$

$$= BC_{in}(A + \overline{A}) + A\overline{B}C_{in} + ABC_{in}$$

$$C_{out} = BC_{in} + A\overline{B}C_{in} + AB\overline{C_{in}}$$

**This expression can be written in the following form**

$$C_{out} = BC_{in} + AC_{in} + AB \qquad\qquad\qquad (2)$$

**K-map simplification can also be done for above expressions full adder can also be implemented using two half adders and one OR gate. Figure gives such implementations**
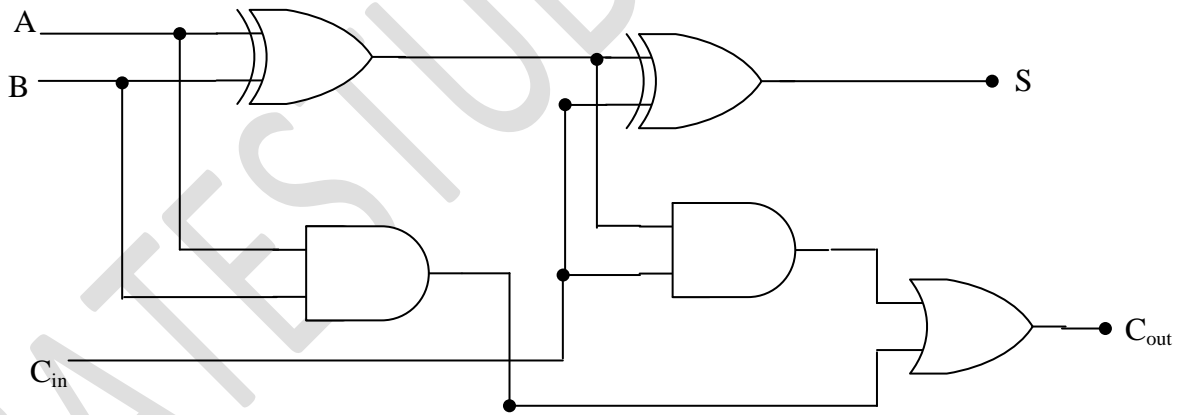


**The S$_{output}$ from the second half adder is EX-OR of C$_{in}$. C$_{out}$ is as per the expression, used modulo 2 adders instead of simple adder.**

**It is implemented with two half adders and one OR gate.**

**3.** Find the expression for difference and borrow for binary full sub tractor.

**Soln.**     **Half and full subtractors can be implemented in the same way**

**Half sub tractor**

$$D = X \oplus Y$$

$$B_{out} = \overline{X}Y$$

**Full sub tractor**

$$D = X \oplus Y \oplus B_{in}$$

$$B_{out} = \overline{X}Y + \overline{X} B_{in} + Y B_{in}$$

**Full sub tractor can be implemented using two half sub tractors.**

**4.** For a binary half sub tractor having two inputs A and B, the correct set of logical expressions for the outputs D (= A minus B) and X (= borrow) are
(a) $D = AB + \overline{A}B, X = \overline{A}B$
(b) $D = \overline{A}B + A\overline{B} + A\overline{B}, X = A\overline{B}$
(c) $D = \overline{A}B + A\overline{B}, X = \overline{A}B$
(d) $D = AB + \overline{A}\overline{B}, X = A\overline{B}$

**[GATE-1999: 2 Marks]**

**Soln.**     **Truth table for the half sub tractor can be written as**

| A | B | D (Difference) | X (Borrow) |
|---|---|---|---|
| **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **1** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** |

**Boolean equation for Col D can be written as**

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

**For     Col. X**

$$X = \overline{A}B \qquad \text{Option (C)}$$

**5.** A 2 bit binary multiplier can be implemented using
  (a) 2 input ANDs only
  (b) 2 input X-ORs and 4-input AND gates only
  (c) Two (2) input NORs and one XNOR gate
  (d) XOR gates and shift registers

<div align="right">

**[GATE-1997: 1 Mark]**

</div>

**Soln.     Multiplication of binary numbers is done in the same manner as multiplication of decimal numbers. The process is simple because the multiplier digits are either 0 or 1**

| | $B_1$ | $B_0$ | **Multiplicand** |
|---|---|---|---|
| | $A_1$ | $A_0$ | **Multiplier** |
| | $A_0 B_1$ | $A_0 B_0$ | **Partial product** |

| $A_1 B_1$ | $A_1 B_0$ | | |
|---|---|---|---|
| $C_3$ $C_2$ | $C_1$ | $C_0$ | |

**Final product is sum of partial products. So it needs AND and OR operations**

**Option (b)**

# Multiplexers

**6.** The logic function implemented by the circuit below is (ground implies a logic "0")



4 x 1 MUX
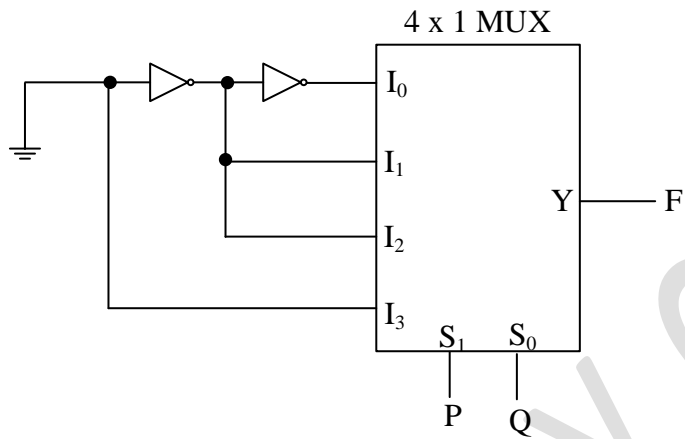
(a) F = AND (P,Q)  
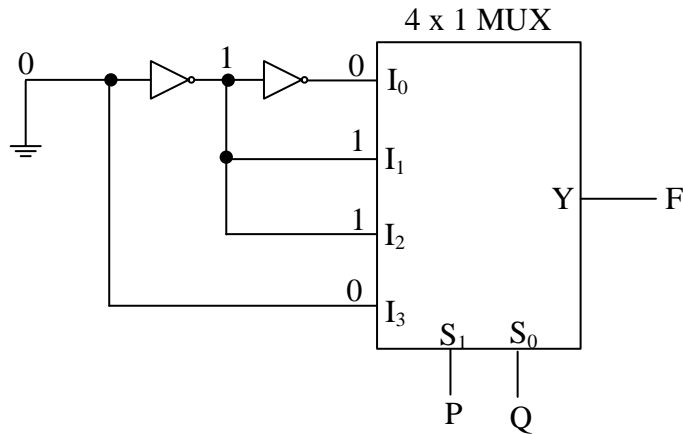(b) F = OR (P,Q)  
(c) F = XNOR (P,Q)  
(d) F = XOR (P,Q)

**[GATE -2011: 1Mark]**

**Soln.** **Above circuit is redrawn to indicate the voltage levels**

**Truth table for MUX can be written as:**
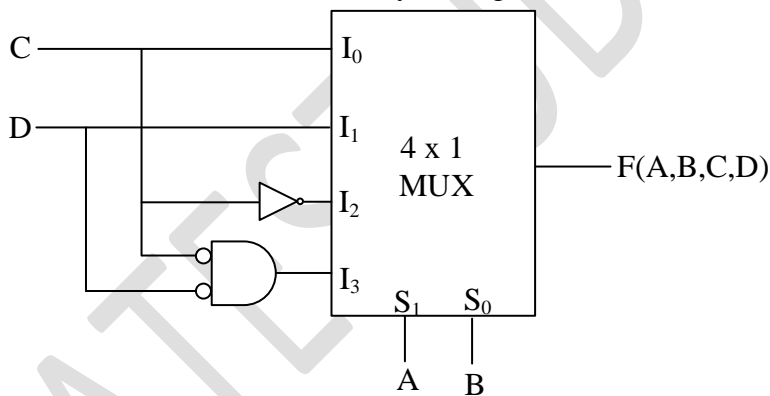
| P | Q | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**4 x 1 MUX**

**Output F can be written as**

$$F = \bar{P}Q + P\bar{Q} = P \oplus Q$$

**Option (d)**

**7.** The Boolean function realized by the logic circuit shown is



(a) $F = \Sigma m\ (0,1,3,5,9,10.14)$    (c) $F = \Sigma m\ (1,2,4,5,11,14.15)$
(b) $F = \Sigma m\ (2,3,5,7,8,12.13)$    (d) $F = \Sigma m\ (2,3,5,7,8,9.12)$

**[GATE-2010: 2 Marks]**

**Soln.**        **Output can be written as**

$$F = \bar{A}\bar{B}C + \bar{A}BD + A\bar{B}\bar{C} + AB(\bar{C}.\bar{D})$$

**Since the result is to be given in min term form, we expand the terms.**

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

**Min. terms are put in K-map**

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 |  |  | 1 | 1 |
| 01 |  | 1 | 1 |  |
| 11 | 1 |  |  |  |
| 10 | 1 | 1 |  |  |

$$F = \sum m\,(2, 3, 5, 7, 8, 9, 12)$$

**Option (d)**

**8.** What are the minimum number of 2-to-1 multiplexers required to generate a 2- input AND gate and a 2-input Ex-OR gate?
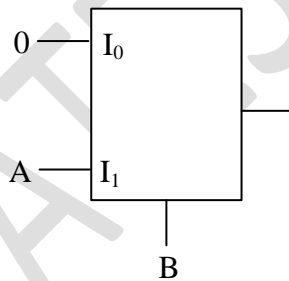
(a) 1 and 2
(b) 1 and 3

(c) 1and 1
(d) 2 and 2

[GATE-2009: 2 Marks]

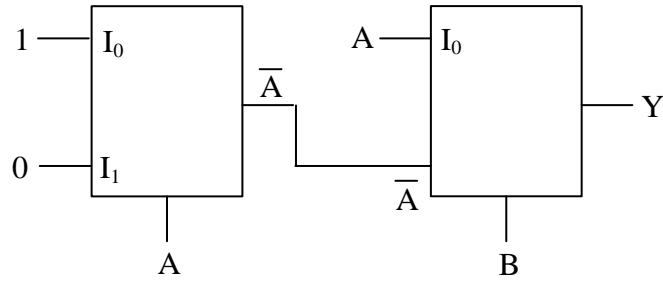**Soln.**     **Consider 2-input AND gate**

**Let the variables are A and B**

```
0 ──┤ I₀      ┐
    │         ├──
A ──┤ I₁      ┘
       │
       B
```

**B is used in select line.**

$$F = \bar{B}.\,0 + B.\,A = AB \qquad\qquad \textbf{AND gate}$$
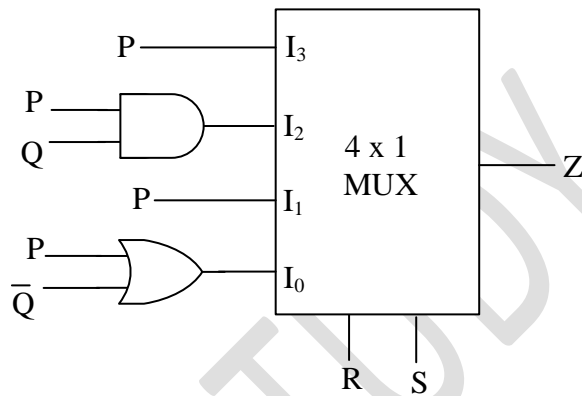
**2-input EX-OR gate**

**Let the variables be A & B for EX-OR gate complements of variable are also needed**

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

**Option (a) 1 and 2**

**9.** For the circuit shown in the following figure, $I_0 - I_3$ are input to the 4:1 multiplexer R (MSB) and S are control bits.



The output Z can be represented by

(a) $PQ + P\bar{Q}S + \bar{Q}\bar{R}\bar{S}$

(b) $P\bar{Q} + PQ\bar{R} + \bar{P}\bar{Q}\bar{S}$

(c) $P\bar{Q}\bar{R} + \bar{P}QR + PQRS + \bar{Q}\bar{R}\bar{S}$

(d) $PQ\bar{R} + PQR\bar{S} + P\bar{Q}\bar{R}S + \bar{Q}\bar{R}\bar{S}$

**[GATE-2008: 2 Marks]**

**Soln.**     $\overline{RS}.(P + \bar{Q}) + \bar{R}S.P + R\bar{S}PQ + RSP$

$= \overline{RS}P + \overline{RS}\bar{Q} + \bar{R}SP + R\bar{S}PQ + RSP$

**Plot on K-map & simplify**

|  RS<br>PQ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 |  |  |  |
| 01 |  |  |  |  |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 |  |

$$Z = PQ + P\bar{Q}S + \bar{Q}\bar{R}\bar{S}$$

**Option (a)**

**10.** In the following circuit, X given by
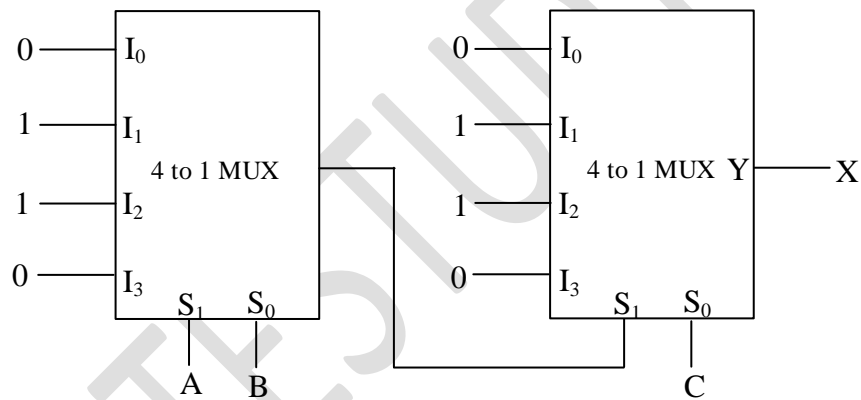


(a) $X = A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + ABC$

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

(c) $X = AB + BC + AC$

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

**[GATE-2007: 2 Marks]**

**Soln.**     **Output of first MUX is Y (say)**

$$Y = \bar{A}\bar{B}.\,0 + \bar{A}B.\,1 + A\bar{B}.\,1 + AB.\,0$$

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

$$X = \bar{Y}.\,C + Y\bar{C} = Y \oplus C$$

**Result is in expanded form.**

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

$$= \overline{\bar{A}B}.\overline{A\bar{B}}.C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

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

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

**Option (a)**

**11.** The Boolean function f implemented in the figure using two input multiplexers is
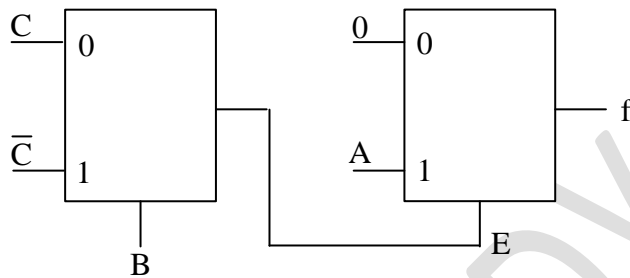


(a) $A\bar{B}C + AB\bar{C}$          (c) $\bar{A}BC + \bar{A}\bar{B}\bar{C}$
(b) $ABC + A\bar{B}\bar{C}$          (d) $\overline{AB}C + \bar{A}B\bar{C}$

**[GATE-2005: 1 Mark]**

**Soln.**    The output E can written as

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

**Output f can be written as**

$$f = \bar{E}.0 + EA = E.A$$

**So,**   $f = EA = A\bar{B}C + AB\bar{C}$

**Option (a)**

**12.** The minimum number of 2-to-1 multiplexers required to realize a 4-to-1 multiplexer is

(a) 1                                              (c) 3

(b) 2                                              (d) 4

**Soln.**      **For lager number of inputs a tree is created. It is achieved by enable/strobe signal. No of inputs is double i.e. from 2 to 4**

**So we need the output from both these OR gate is required. One could also use 2:1 MUX instead of OR gate**

**So total number 2:1 MUX required is 2+1=3**

**Option (c)**

**13.** Without any additional circuitry an 8:1 MUX can be used to obtain

(a) Some but not all Boolean functions of 3 variables

(b) All function of 3 variables but none of 4 variables

(c) All functions of 3 variables and some but not all of 4 variables
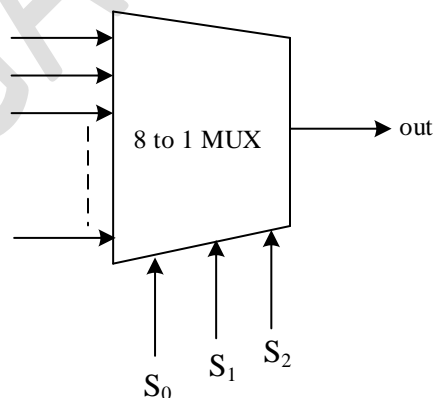
(d) All functions of 4 variables

**Soln.**      **Note that**

**A $2^n$:1 MUX can implement all logic functions of (n+1) variables without any additional circuitry.**

**Thus 8:1 MUX can implement all logic functions of (3+1) variables**

**For 4 variables there are 16 possible combinations. So to use 8:1 MUX use 3 inputs as select lines of MUX and the 4th input as input of MUX**

**14.** In the TTL circuit in the figure, $S_2$ and $S_0$ are select lines and $X_7$ and $X_0$ are input lines. $S_0$ and $X_0$ are LSBs. The output Y is
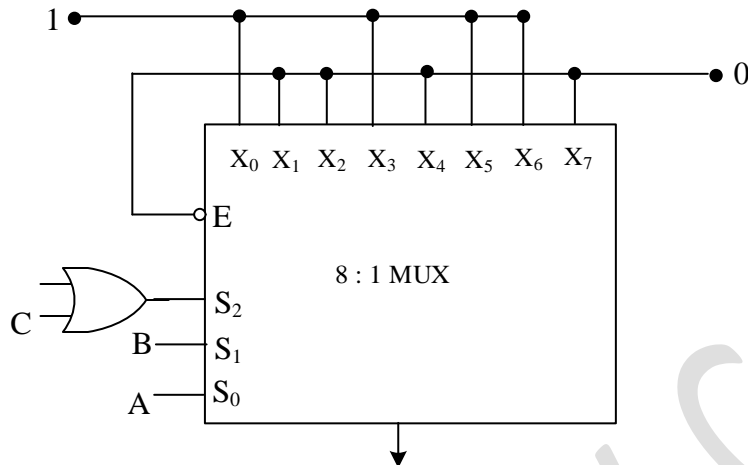


(a) Indeterminate

(b) $A \oplus B$

(c) $\overline{A \oplus B}$

(d) $\bar{C}(\overline{A \oplus B}) + C(A \oplus B)$

**[GATE-2001: 2 Marks]**

**Soln.** It is given that the MUX is made up of TTL circuit. For TTL circuit open terminal is taken high, since $S_2$ select line is connected to OR gate whose one terminal connected to C and the other is open (high) so OR gate output is $S_2 = 1 + C = 1$
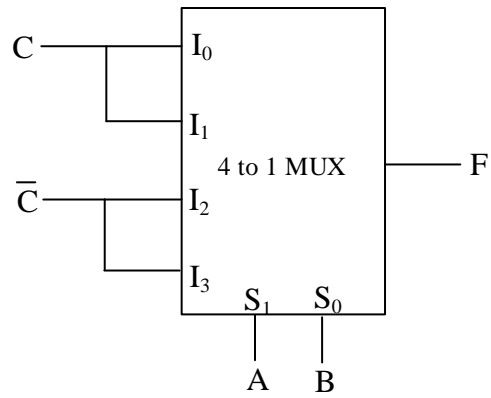
So, $S_2 = 1$

| $S_2 = 1$ | $S_{1(B)}$ | $S_{0(A)}$ | Y |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$Y = (S_1 \oplus S_0) = (A \oplus B)$

**Option (b)**

**15.** The logic realized by the circuit shown in figure is



(a) $F = A \odot C$

(b) $F = A \oplus C$

(c) $F = B \odot C$

(d) $F = B \oplus C$

**Soln.**    **Output F is given by**

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

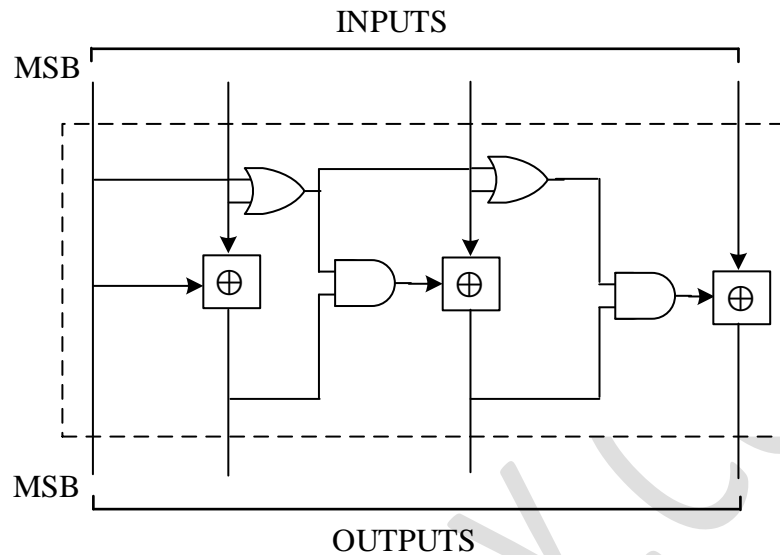$$= \bar{A}C(\bar{B} + B) + A\bar{C}(\bar{B} + B)$$

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

$$= (A \oplus C)$$

**Option (b)**

# Code Convertors

**16.** The circuit shown in the figure converts



(a) BCD to binary code          (c) Excess-3 to Gray code

(b) Binary to excess-3 code        (d) Gray to Binary code
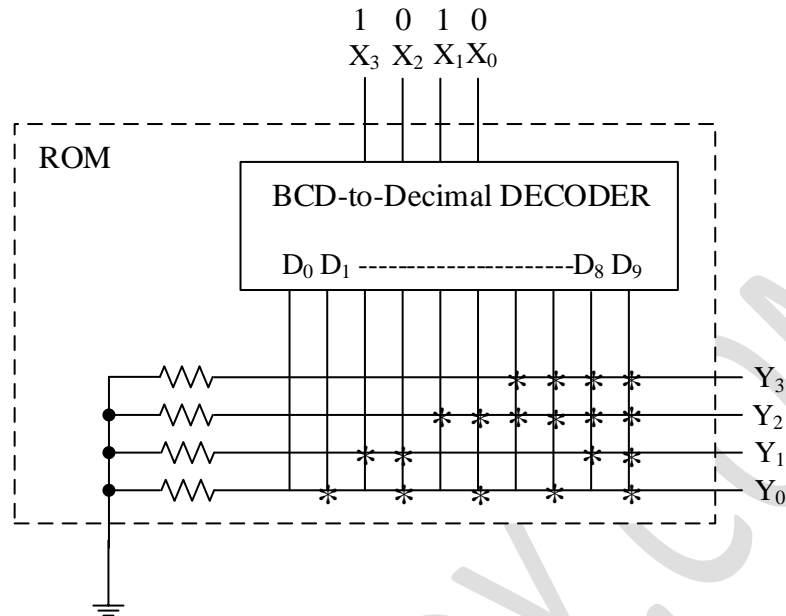
**[GATE-2003: 2 Marks]**

Soln. The circuit is a code converter i.e. converts input bit code to some other code. It uses EX-OR gates, and the inputs to EX-OR are from the input bit and output bit. So the guess is that the circuit is for gray to binary converter. Take some simple input bits and verify



Note that EX-OR gate is modulo 2 adder.

**17.** If the input $X_3$, $X_2$, $X_1$, $X_0$ to the ROM in the figure are 8, 4, 2, 1 BCD numbers, then the output $Y_3$, $Y_2$, $Y_1$, $Y_0$ are



(a) gray code numbers                    (c) Excess-3 code numbers

(b) 2 4 2 1 BCD numbers                  (d) None of the above

**[GATE-2002: 2 Marks]**

Soln.   The given circuit is for BCD to decimal decoder.

Consider input BCD code as 1001 this is equivalent to decimal $9_{10}$ i.e. $D_9$. For this the outputs are

$Y_3$      $Y_2$      $Y_1$      $Y_0$      i.e.      1111

Take another BCD code.

1000 this is equivalent to $8_{10}$ i.e. $D_8$ the output is 1110

It can be verified that this corresponds to 2421 BCD numbers

Option (b)