CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - I

| Sub: | DIGITAL SYSTEM DESIGN | | | | | | | Code: | 17EE35 |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 10/09/2018 | Duration: | 90 mins | Max Marks: | 50 | Sem: | 3rd | Branch: | EEE |

Answer For FIFTY marks

| | | Marks | OBE | |
|---|---|---|---|---|
| | | | CO | RBT |
| 1. | Solve four variable Quine McCluskey minimization techniques.<br>S=f(w, x, y, z)=∑(1,3,13,15)+d(8,9,10,11) | 10 | CO1 | L3 |
| 2. | Simplify the Boolean expression using a 3-variable VEM, with 'd' as MEV<br>F(a, b, c, d)= ∑(2,9,10,11,13,14,15) | 10 | CO1 | L3 |
| 3. | Define Combinational logic.<br>Solve the following Boolean equation using four variable Karnaugh map and implement the simplified equation using minimum number of logic gates.<br>A=f(w, x, y, z)=π(2,3,8,9,10,11,12,13,14,15) | 10 | CO1 | L2 |
| 4. | What are Multiplexers? Implement the function using<br>i) 8:1 multiplexer with a, b, c as select lines.<br>ii) 4:1 multiplexer with a, b as select lines.<br>F=f(a, b, c, d)= ∑(0,1,5,6,7,9,10,15) | 10 | CO2 | L2 |
| 5. | What is a decoder? Design a keypad interface to a digital system using 10 line BCD encoder. | 10 | CO2 | L2 |
| 6. | What is the problem associated with the parallel adder? Explain the method of correcting it, with suitable circuit equations. | 10 | CO2 | L4 |

******All the Best*****

Internal Assessment Test-1

sub: Digital system Design (17EE35)

solution

Q.1

$S = f(w,x,y,z) = \Sigma(1,3,13,15) + \Sigma d(8,9,10,11)$

| INPUT | | | | OUTPUT |
|---|---|---|---|---|
| w | x | y | z | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | d |
| 1 | 0 | 0 | 1 | d |
| 1 | 0 | 1 | 0 | d |
| 1 | 0 | 1 | 1 | d |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

step 1:

| Group | Minterm | variables w x y z | |
|---|---|---|---|
| 1 | 1 | 0 0 0 1 | ✓ |
| 1 | 8* | 1 0 0 0 | ✓ |
| 2 | 3 | 0 0 1 1 | ✓ |
| 2 | 9* | 1 0 0 1 | ✓ |
| 2 | 10* | 1 0 1 0 | ✓ |
| 3 | 11* | 1 0 1 1 | ✓ |
| 3 | 13 | 1 1 0 1 | ✓ |
| 4 | 15 | 1 1 1 1 | ✓ |

step 2:

| Group | Minterm | variables w x y z | |
|---|---|---|---|
| 1 | 1, 3 | 0 0 - 1 | |
| 1 | 1, 9* | - 0 0 1 | |
| 1 | 8*, 9* | 1 0 0 - | |
| 1 | 8*, 10* | 1 0 - 0 | |
| 2 | 3, 11* | - 0 1 1 | |
| 2 | 9, 11* | 1 0 - 1 | |
| 2 | 9, 13 | 1 - 0 1 | |
| 2 | 10*, 11* | 1 0 1 - | |
| 3 | 11*, 15 | 1 - 1 1 | |
| 3 | 13, 15 | 1 1 - 1 | |

step 3:

| Group | Minterms | variables w | x | y | z |
|---|---|---|---|---|---|
| 1 | 1, 3, 9*, 11* | – | 0 | – | 1 |
| 1 | 8*, 9*, 10*, 11* | 1 | 0 | – | – |
| 2 | 9*, 13, 11*, 15 | 1 | – | – | 1 |

step 4:

| PI Term | Decimal |
|---|---|
| $x'z$ | 1, 3, 9*, 11* |
| $wz$ | 9*, 11*, 13, 15 |

| | Minterm 1 | 3 | 13 | 15 |
|---|---|---|---|---|
| | (x) | (x) | | |
| | | | (x) | (x) |

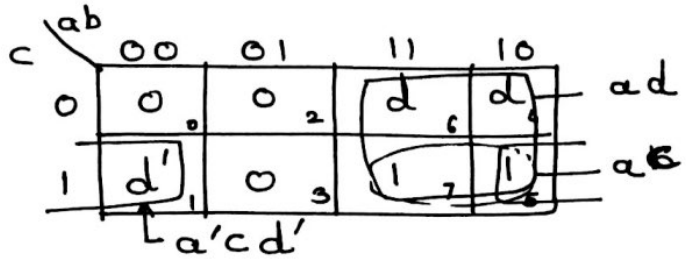
$$S = x'z + wz$$

Q. 2

$F = f(a, b, c, d) = \Sigma(2, 9, 10, 11, 13, 14, 15)$

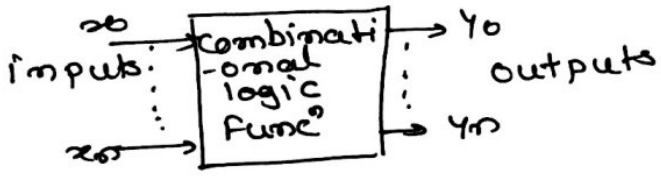
| Decimal term MEV | std. | variables a | b | c | d (MEV) | output |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 2 | 0 | 0 | 1 | 0 | 1 |
| 1 | 3 | 0 | 0 | 1 | 1 | 0 |
| 2 | 4 | 0 | 1 | 0 | 0 | 0 |
| 2 | 5 | 0 | 1 | 0 | 1 | 0 |
| 3 | 6 | 0 | 1 | 1 | 0 | 0 |
| 3 | 7 | 0 | 1 | 1 | 1 | 0 |
| 4 | 8 | 1 | 0 | 0 | 0 | 0 |
| 4 | 9 | 1 | 0 | 0 | 1 | 1 |
| 5 | 10 | 1 | 0 | 1 | 0 | 1 |
| 5 | 11 | 1 | 0 | 1 | 1 | 1 |
| 6 | 12 | 1 | 1 | 0 | 0 | 0 |
| 6 | 13 | 1 | 1 | 0 | 1 | 1 |
| 7 | 14 | 1 | 1 | 1 | 0 | 1 |
| 7 | 15 | 1 | 1 | 1 | 1 | 1 |

③ K-MAP for MEV(d)



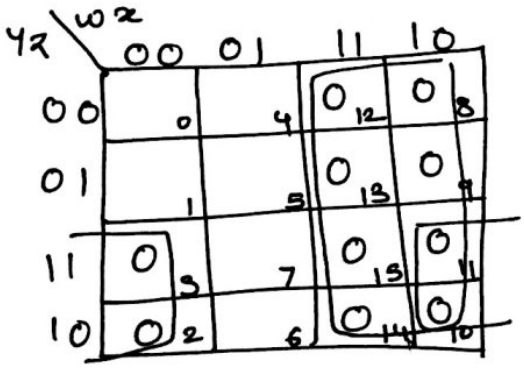

$= ad + ac + a'cd'$

Q.3 Logic circuits without feedback from output to the input (constructed from a functionally complete gate set) are said to be combinational.




$A = f(w, x, y, z) = \Sigma(0,1,5,6,7,11) = \Pi$

$= \Pi(2, 3, 8, 9, 10, 11, 12, 13, 14, 15)$




Q.4 Multiplexer select single data line from several data-input lines and the data from the selected data line would be available on the output.

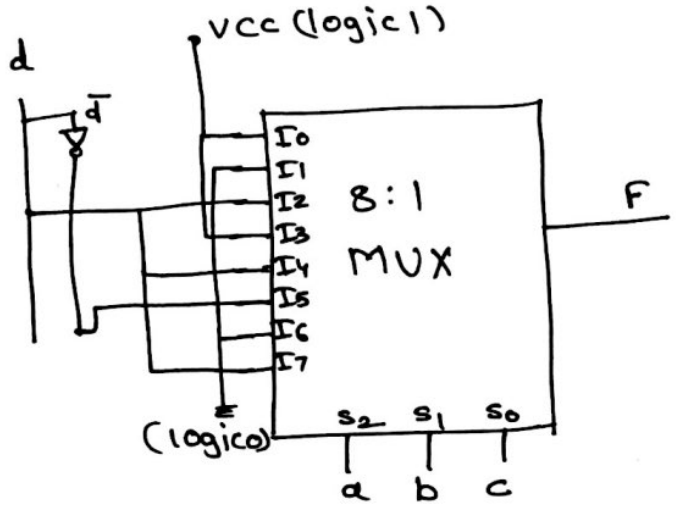
The selection of a particular input line is controlled by a set of selection lines.

(a)

i) 8:1 MUX

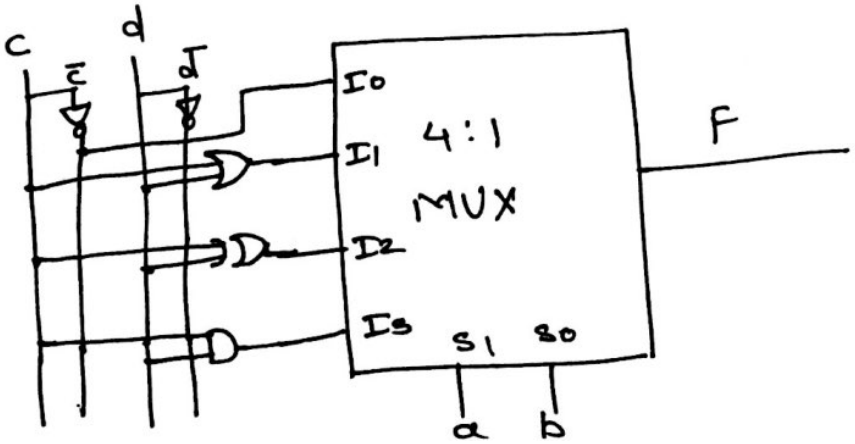
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| d | (1) | 3 | (5) | (7) | (9) | 11 | 13 | (15) |
| $\bar{d}$ | (0) | 2 | 4 | (6) | 8 | (10) | 12 | 14 |
| | $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ |
| | 1 | 0 | d | 1 | d | $\bar{d}$ | 0 | d |

| Decimal digit | INPUT a | b | c | d | OUTPUT |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | $\phi$ | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 1 |

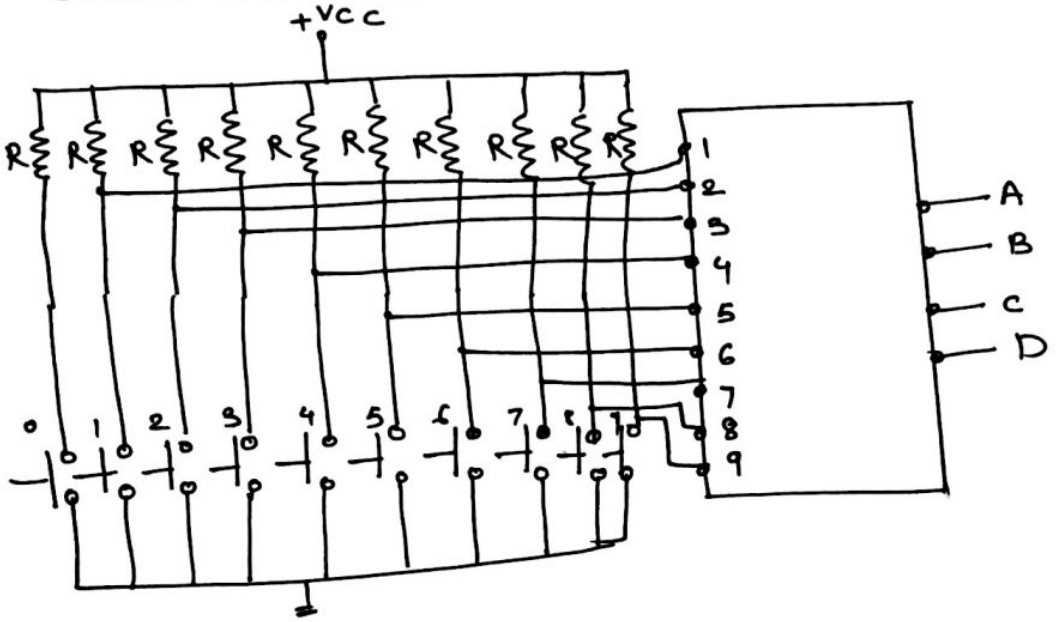



ii) 4:1 MUX

| | $\bar{c}\bar{d}$ | $\bar{c}d$ | $c\bar{d}$ | cd | |
|---|---|---|---|---|---|
| $I_0$ | (0) | (1) | 2 | 3 | $\bar{c}$ |
| $I_1$ | 4 | (5) | (6) | (7) | $c+d$ |
| $I_2$ | 8 | (9) | (10) | 11 | $\bar{c}d + c\bar{d} = c \oplus d$ |
| $I_3$ | 12 | 13 | 14 | (15) | cd |

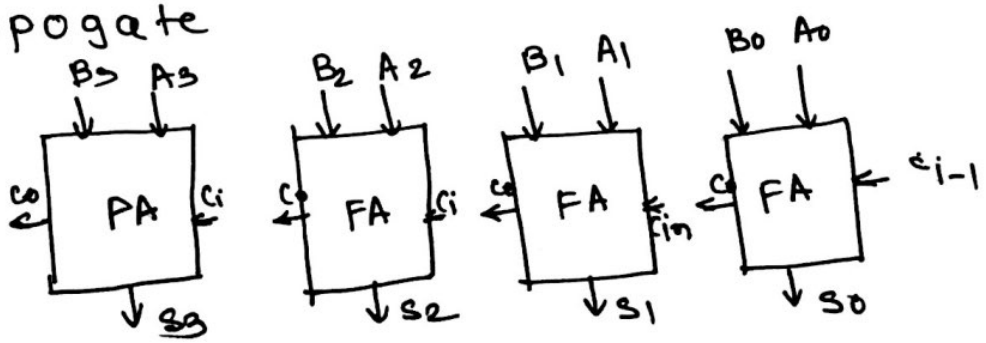


Q.5. Decoder is a multiple-input, multiple-output logic circuit which converts coded inputs into coded output where the input and output codes are different.




Q.6. In parallel adder carry output of each full-adder stage is connected to the carry input of the next higher-order stage. The sum and carry outputs of any stage cannot be produced until the input carry occurs. This leads to a time delay in addition process. This is known as carry propogation delay.

speeding up this, by eliminating inter stage carry delay is called look ahead-carry addition. It uses two functions: carry generate and carry propogate



Truth Table for carry

| A | B | Ci | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$C_o = A.B + (A \oplus B).C_{in}$

carry Gen. carry pro.

$C_i = G_i + P_i C_{i-1}$

when i=0

$C_o = G_0 + P_0.C_{-1}$ —①

i=1

$C_1 = G_1 + P_1.C_0$

$= G_1 + P_1 (G_0 + P_0 C_{-1})$

$= G_1 + P_1 G_0 + P_1.P_0 C_{-1}$ —②

i=2

$C_2 = G_1 + P_2.C_1$

$= G_2 + P_2 (G_1 + P_1.G_0 + P_1.P_0.C_{-1})$

$= G_2 + P_2 G_1 + P_2.P_1.G_0 + P_2.P_1 P_0.C_{-1}$ —③

i=3

$C_3 = G_3 + P_3.C_2 = G_3 + P_3(G_2 + P_2 G_1 + P_2.P_1.G_0 + P_2 P_1 P_0 C_{-1})$

$= G_3 + P_3 G_2 + P_3.P_2.G_1 + P_3.P_2.P_1.G_0 + P_3.P_2.P_1.P_0 C_{-1}$ —④

carry out is depend on numbers and cin. so addition is speed up. and delay is reduced.