SOLUTION

AUTUMN MID-SEMESTER EXAMINATION-2015

DIGITAL ELECTRONIC CIRCUITS

[EC-2009]

---

1. a) BCD Subtraction: 476.7 -297.8 **[1]**

$$\begin{array}{rrrrr} & 0100 & 0111 & 0110 & .\ 0111 \\ - & 0010 & 1001 & 0111 & .\ 1000 \\ \hline & 0001 & 1101 & 1110 & .\ 1111 \\ - & & 0110 & 0110 & .\ 0110 \\ \hline & 0001 & 0111 & 1000 & .\ 1001 \end{array}$$

b) 101010 **[1]**

Sign-Magnitude Form: **-10**
1's Compliment Form: **-21**
2's Compliment Form: **-22**

c) $\mathbf{U\bar{V} + UW + V\overline{W} = U + V\overline{W}}$ **[1]**

L.H.S: $\mathbf{U\bar{V} + UW + V\overline{W}}$

$= \mathbf{U(\bar{V} + W) + V\overline{W}}$

$= \mathbf{U(\overline{V.\overline{W}}) + V\overline{W}}$

$= \mathbf{U + V\overline{W}}$ =R.H.S [Since, $A + \bar{A}.B = A + B$]

d) Advantage of ***Look-Ahead Carry Adder : Faster, Less Propagation Delay(Ripple Delay)*** **[1]**

$C_1 = P_0C_0 + G_0$

$C_2 = P_1P_0C_0 + P_1G_0 + G_1$

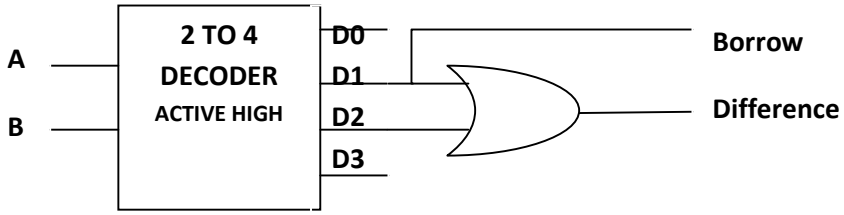
$C_3 = P_2P_1P_0C_0 + P_2P_1G_0 + P_2G_1 + G_2$

$C_4 = P_3P_2P_1P_0C_0 + P_3P_2P_1G_0 + P_3P_2G_1 + P_3G_2 + G_3$

e) Half Subtractor using 2 *to 4 Active High DECODER* **[1]**

Differnce= $\sum m(1,2) = A \oplus B$

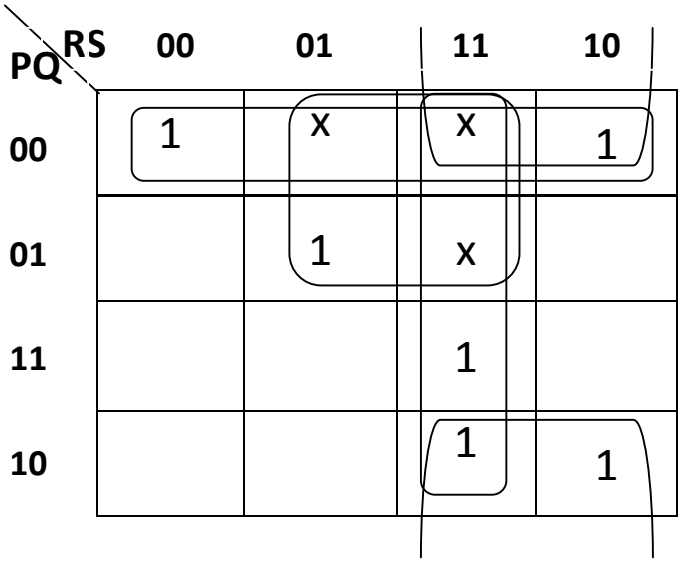
Borrow $= \sum m(1) = \bar{A}B$



2. $F(P,Q,R,S) = \prod M(4,6,8,9,12,13,14) \cdot d(1,3,7)$ [POS Form]

$= \sum m(0,2,5,10,11,15) \cdot d(1,3,7)$ [SOP Form] **[1]**

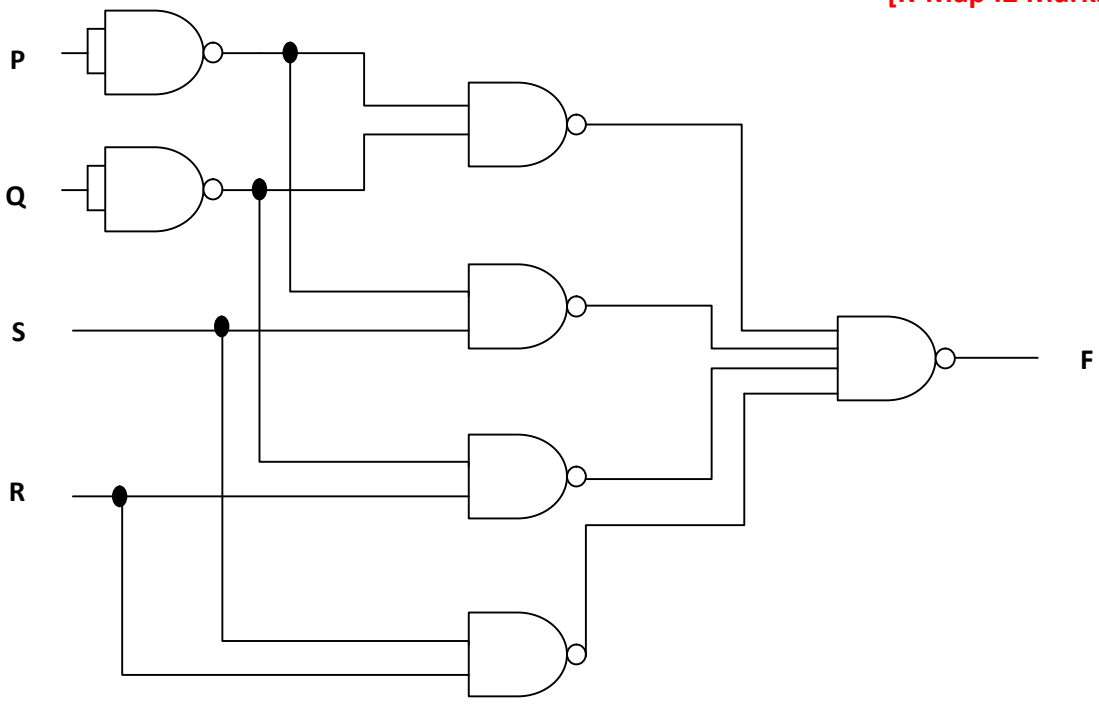
[SOP Form is for "***NAND-NAND***" Two-Level Universal gate implementation]



After K-MAP simplification:

$$F(P,Q,R,S) = \bar{P}\bar{Q} + \bar{P}S + \bar{Q}R + RS$$

[K-Map :2 Marks]



[NAND-NAND Circuit implementation 2 Marks]

3. a) F(A,B,C,D)= $\bar{A}B + BC + B\bar{D}$ [Ordinary SOP Form]

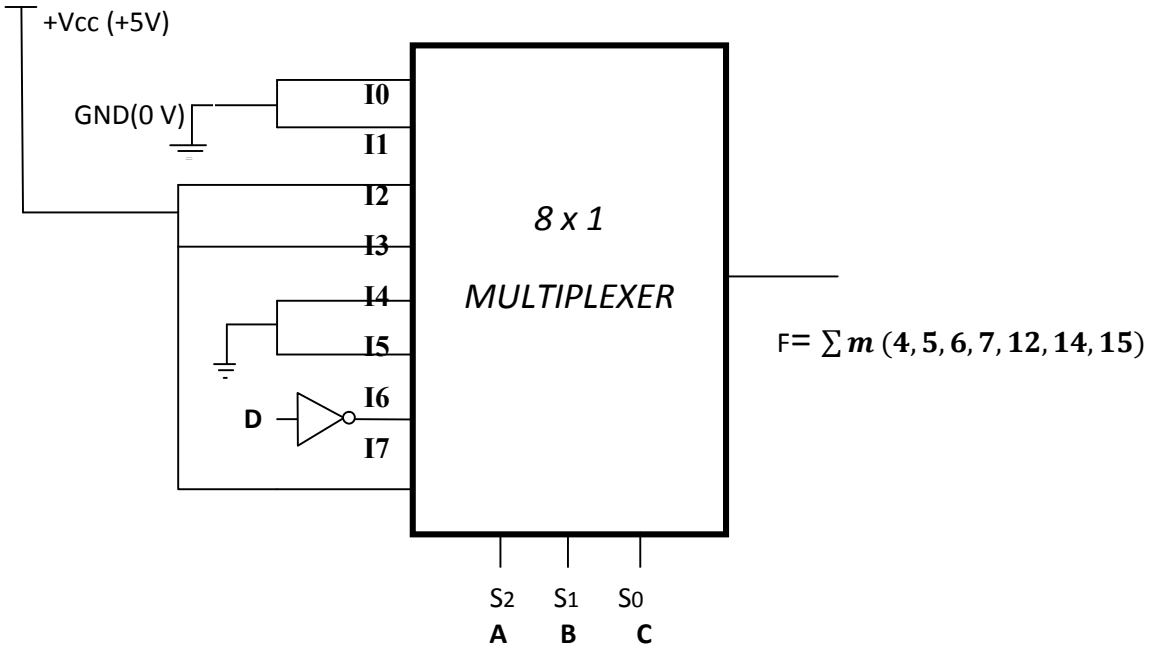
- MULTIPLEXER will implement Min-terms for the given input variables connected to its SELECT LINES A,B,C will be connected to S2,S1 and S0 respectively and D will be connected to input lines according to the requirement.
- Thus, for implementation using Decoder, Ordinary SOP has to be converted to Standard SOP Form.

F(A,B,C,D)= $\bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD + BC\bar{A}\bar{B} + BC\bar{A}B + BCA\bar{B} + BCAB +$
$B\bar{D}\bar{A}\bar{C} + B\bar{D}\bar{A}C + B\bar{D}A\bar{C} + B\bar{D}AC$

$= m4 + m5 + m6 + m7 + m6 + m7 + m14 + m15 + m4 + m6 + m12 + m14$

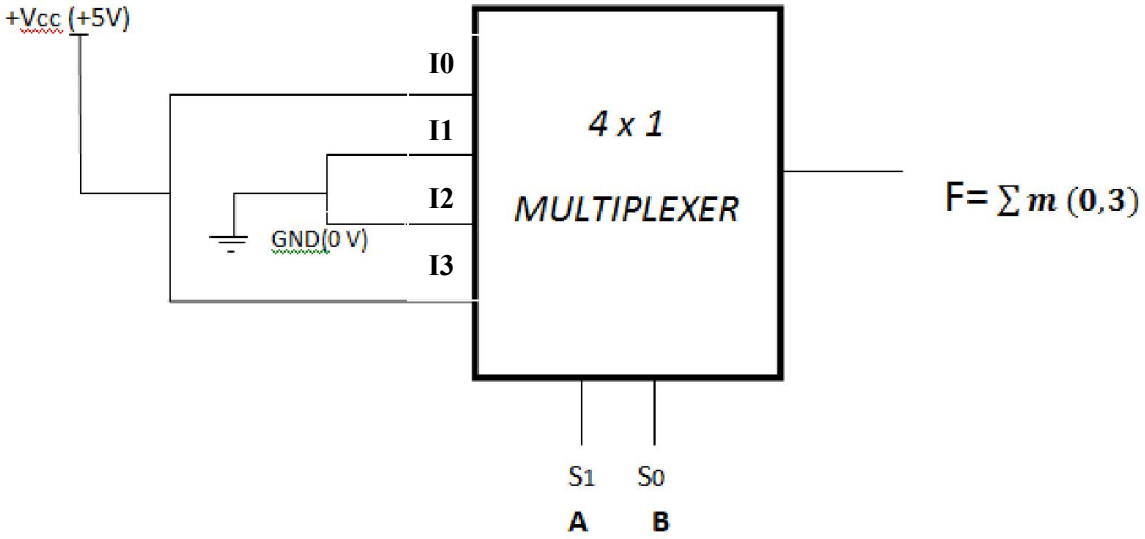
$= \sum m\,(4, 5, 6, 7, 12, 14, 15)$ [1]+Table [2] + circuit [1]

S2S1S0	A	B	C	D	F	
000	0	0	0	0	0	F=0
	0	0	0	1	0	
001	0	0	1	0	0	F=0
	0	0	1	1	0	
010	0	1	0	0	1	F=1
	0	1	0	1	1	
011	0	1	1	0	1	F=1
	0	1	1	1	1	
100	1	0	0	0	0	F=0
	1	0	0	1	0	
101	1	0	1	0	0	F=0
	1	0	1	1	0	
110	1	1	0	0	1	F=$\bar{D}$
	1	1	0	1	0	
111	1	1	1	0	1	F=1
	1	1	1	1	1	



b) $F(A,B) = \overline{A \oplus B}$

$= \sum m\,(0,3)$ **[1]**



4. a) **[0.5]**

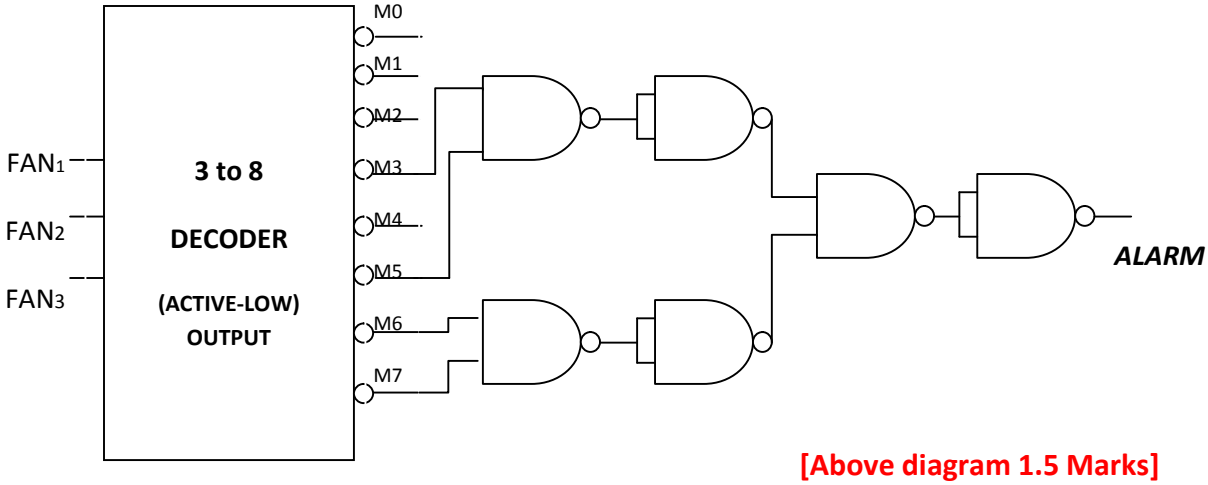
FAN_1	FAN_2	FAN_3	ALARM
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

ALARM= $\sum m\,(0,1,2,4)$ **[1]**

$= \prod M\,(3,5,6,7)$

- DECODER having ACTIVE LOW outputs will give *MAX-TERMS*

- Implementation can be done by NAND gates, We need product of MAX-TERMS so each stage NAND gate has to be converted into AND gate so we get 4 stage logic. NAND-NAND-NAND-NAND , which will work like AND-AND ≈ AND i.e Product of MAX-TERMS



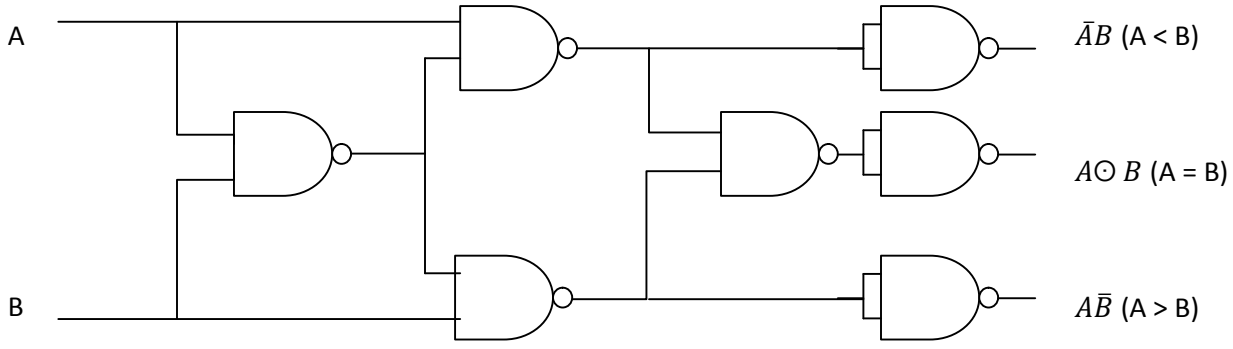
[Above diagram 1.5 Marks]

b) **1 bit Magnitude Comparator:**

Explanation: **[0.5 MARKS]**

A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0
Boolean Function: →	$\sum m\,(1) = \bar{A}B$	$\sum m\,(0,3) = \bar{A}\bar{B} + AB$ = A ʘ B	$\sum m\,(2) = A\bar{B}$	

[0.5 Marks for Correct Expression above table]



[Implementation using Minimum 5 NAND Gates 1 Marks]

5. a) ***2 bit Priority Encoder***; where the given order of the priorities for 4 inputs are
D3 > D2 > D1 > D0. **[Table: 0.5]**

D3	D2	D1	D0	A	B	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

A **[Each Corect expression A,B & V 1 Mark Each]**

D_3D_2 \ D_1D_0	00	01	11	10
00	X	1	1	1
01		1	1	1
11		1	1	1
10		1	1	1

A= $D_3 + D_2$

B

D_3D_2 \ D_1D_0	00	01	11	10
00	X	1	1	
01	1	1	1	
11	1	1	1	
10		1	1	

B= $D_3 + \overline{D_2}\, D_1$

Valid : **V=** $D_3 + D_2 + D_1 + D_0$

- **Draw the logic diagram using basic gates for above simplified expressions.** **[0.5]**

b) Received ***7-bit Hamming code*** is '***1110100***' then find the 4-bit data word. (Even Parity) **[1]**

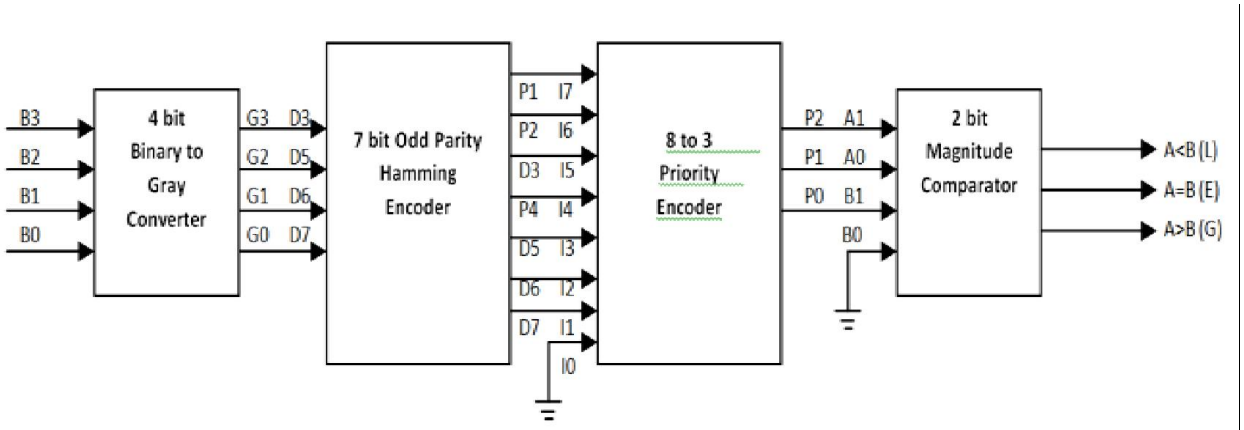
P1	P2	D3	P4	D5	D6	D7
1	1	1	0	1	0	0

$$c1 = p1 \oplus \boldsymbol{D3} \oplus \boldsymbol{D5} \oplus \boldsymbol{D7}$$
$$c2 = p2 \oplus \boldsymbol{D3} \oplus \boldsymbol{D6} \oplus \boldsymbol{D7}$$
$$c4 = p4 \oplus \boldsymbol{D5} \oplus \boldsymbol{D6} \oplus \boldsymbol{D7}$$

C1	C2	C4
1	0	1

- Thus, D5 is in error; so instead of '1' it should be '0' (1 bit is in error)
- Correct 4 bit data word will be : ***1000***

6. a) **Each Correct Block Answer : 1 Mark each Block]**



1st Block:

Binary (input)				Gray (output)			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
1	0	1	0	1	1	1	1

2nd Block:

Gray code = Data word (input)				Odd Hamming Code (Output)						
D3	**D2**	**D1**	**D0**	**P1**	**P2**	**D3**	**P4**	**D5**	**D6**	**D7**
1	1	1	1	0	0	1	0	1	1	1

3rd Block:

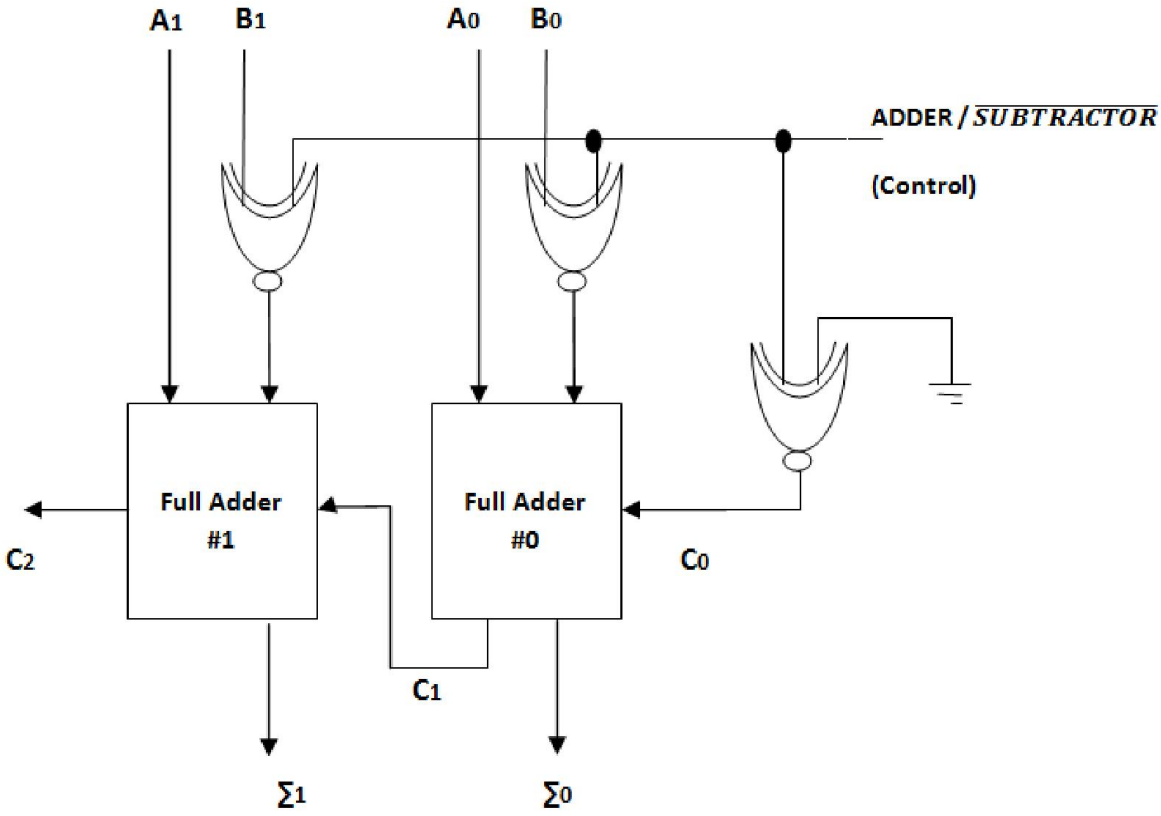
Odd Hamming Code = (I7 – I0) Input to Priority Encoder (input)							Priorities (Output)		
P1	**P2**	**D3**	**P4**	**D5**	**D6**	**D7**	**P2**	**P1**	**P0**
0	0	1	0	1	1	1	1	0	1

Note: D3 (data) is connected to I5 of Priority encoder so the Output of Priority Encoder is ***101***

th Block:

A and B two 2 bit Numbers				2 bit Magnitude Comparator (Output)		
A1	**A0**	**B1**	**B0**	**A < B**	**A = B**	**A > B**
0	0	1	0	**0**	**1**	**0**

b) **[0.5]**



[0.5]

Control (X) $\overline{ADDER / SUBTRACTOR}$	Mode of Operation
1	Adder
0	Subtractor