

Dated:

DLD Homework 3

Ali Muhammad Asad

aa0719D

Q1) 9's Complement of a BCD Digit.

| Decimal No. | BCD | 9's Comp. | W: | C |
|-------------|---------|-----------|--------------------------|---|
| | A B C D | w u y z | AB' CD' | |
| 0 | 0 0 0 0 | 1 0 0 1 | 00 1 1 0 0 | |
| 1 | 0 0 0 1 | 1 0 0 0 | 01 0 0 0 0 | |
| 2 | 0 0 1 0 | 0 1 1 1 | 1 A { 1 0 X X X X | |
| 3 | 0 0 1 1 | 0 1 1 0 | 1 0 0 0 X X | |
| 4 | 0 1 0 0 | 0 1 0 1 | 00 X X X X | |
| 5 | 0 1 0 1 | 0 1 0 0 | W = A' B' C' | |
| 6 | 0 1 1 0 | 0 0 1 1 | | |
| 7 | 0 1 1 1 | 0 0 1 0 | X: AB' CD' 00 01 11 10 | |
| 8 | 1 0 0 0 | 0 0 0 1 | 00 0 0 1 1 | |
| 9 | 1 0 0 1 | 0 0 0 0 | 01 1 1 0 0 | |

$$d(A, B, C, D) = \sum(10, 11, 12, 13, 14, 15)$$

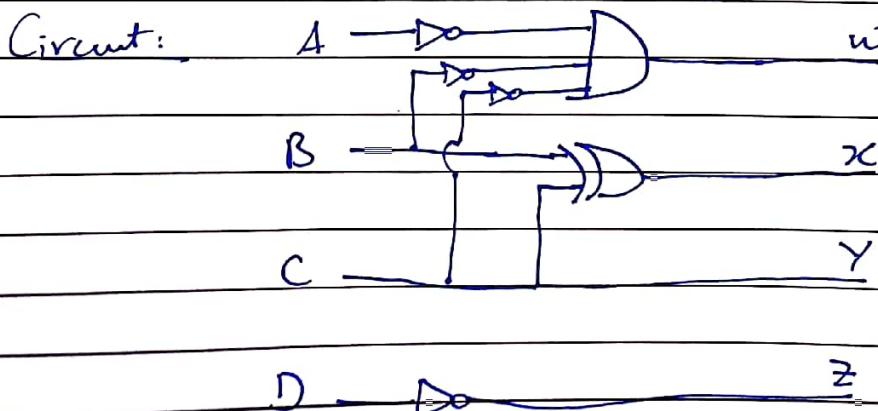
| y: | | AD | 00 | 01 | 11 | 10 | |
|----|--|----|----|----|----|----|-------|
| | | | 00 | 0 | 1 | 1 | y = c |
| | | | 01 | 0 | 1 | 1 | |
| | | | 11 | X | X | X | |
| | | | 10 | 0 | 0 | X | |

$$w = A' B' C'$$

$$x = BC' + B'C = B \oplus C$$

$$z = D'$$

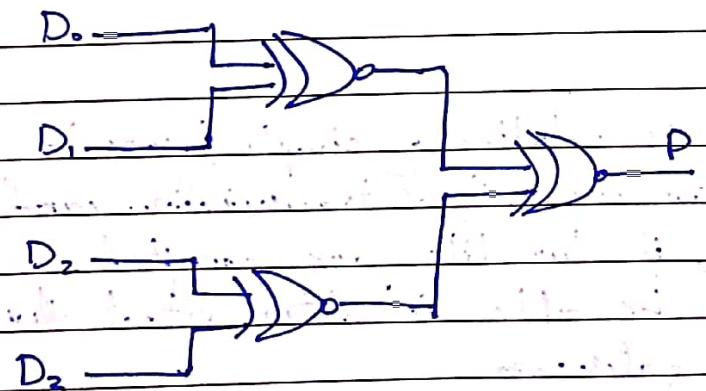
$$w = A' B' C' \quad x = BC' + B'C = B \oplus C \quad y = c \quad z = D'$$



Dated:

Q2) (a) ODD Parity D_0, D_1, D_2, D_3 Inputs, P parity bit

| D_0, D_1, D_2, D_3 | P. | $D_0 D_1 \overset{D_2 D_3}{\oplus}$ | 00 | 01 | 11 | 10 |
|----------------------|----|--|----|----|----|----|
| 0 0 0 0 | 1 | 00 | 1 | 0 | 1 | 0 |
| 0 0 0 1 | 0 | 01 | 0 | 1 | 0 | 1 |
| 0 0 1 0 | 0 | 11 | 1 | 0 | 1 | 0 |
| 0 0 1 1 | 1 | 10 | 0 | 1 | 0 | 1 |
| 0 1 0 0 | 0 | | | | | |
| 0 1 0 1 | 1 | $D_2 D_3 D_0 D_1 + D_0 D_1 D_2 D_3 + D_0 D_1 D_2 D_3$ | | | | |
| 0 1 1 0 | 1 | $D_0 D_1 D_2 D_3 + D_0 D_1 D_2 D_3 + D_0 D_1 D_2 D_3 + D_0 D_1 D_2 D_3$ | | | | |
| 0 1 1 1 | 0 | $D_0 D_1 D_2 D_3 + D_0 D_1 D_2 D_3$ | | | | |
| 1 0 0 0 | 0 | | | | | |
| 1 0 0 1 | 1 | *From book, even parity is | | | | |
| 1 0 1 0 | 1 | $(x \oplus y) \oplus (z \oplus p)$ | | | | |
| 1 0 1 1 | 0 | *Then by this logic, odd parity is | | | | |
| 1 1 0 0 | 1 | $(x \oplus y) \oplus (z \oplus p)$ where \oplus is $\bar{x} \cdot \bar{N} \cdot P$. | | | | |
| 1 1 0 1 | 0 | | | | | |
| 1 1 1 0 | 0 | $P = (D_0 \oplus D_1) \oplus (D_2 \oplus D_3)$ | | | | |
| 1 1 1 1 | 1 | as for odd ls. parity bit is D_3 , then any odd number of assignments would give 0, however for any even number of assignment, 1 would be generated, as required for parity. | | | | |



Q2 (b)

Design even parity tester circuit that receives 05 bits (i.e., D_0, D_1, D_2, D_3, P) and generates a flag F such that $F = 1$, when there is error in transmission and $F = 0$ when there is no error in transmission. Provide the truth table, Boolean expression, and Logic Diagram for generating F

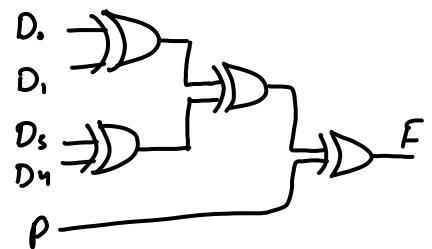
| D_0 | D_1 | D_2 | D_3 | P | F |
|-------|-------|-------|-------|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

F_{sr} 4 bit Even Parity,

$(A \oplus B) \oplus (C \oplus D)$

Then for 5 bit observing the generated truth table,

$$F = \underline{(D_0 \oplus D_1) \oplus (D_2 \oplus D_3) \oplus P}$$

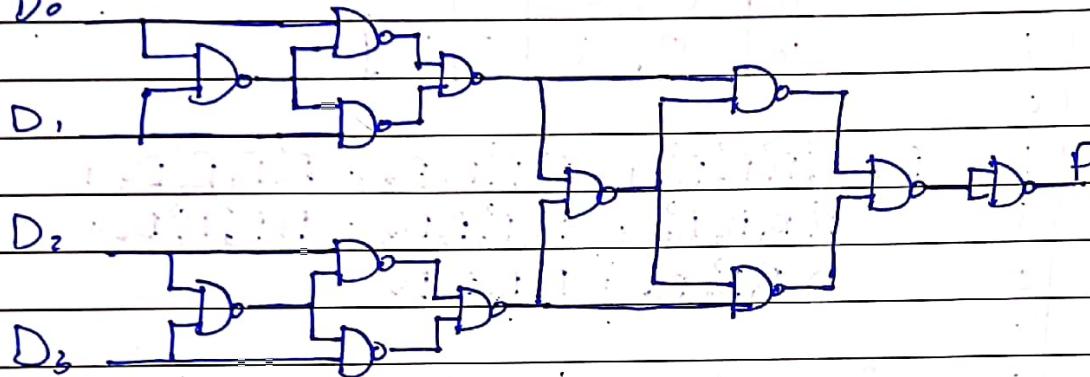


Dated:

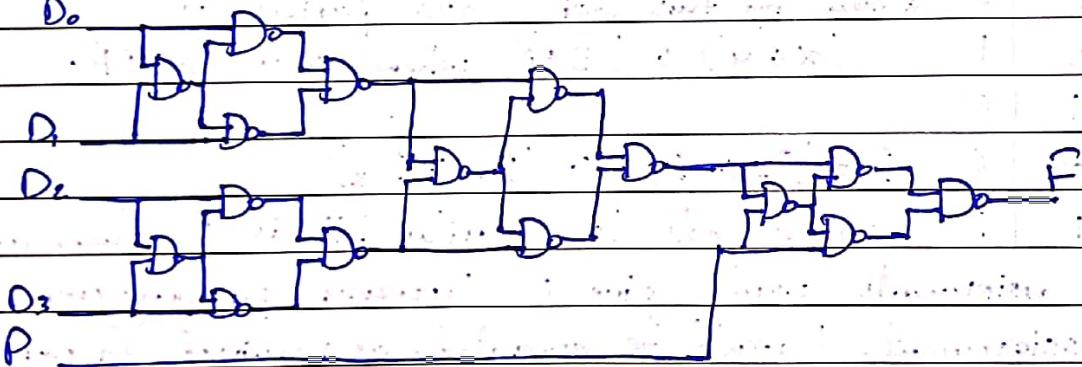
(b) Even Parity Test Circuit \rightarrow 5 bits (D_0, D_1, D_2, D_3, P)
if generates flag P , $P = 1 \rightarrow$ error in transmission.

(c) NAND logic implementation of (a) & (b).

(a) D_0



(b) D_0

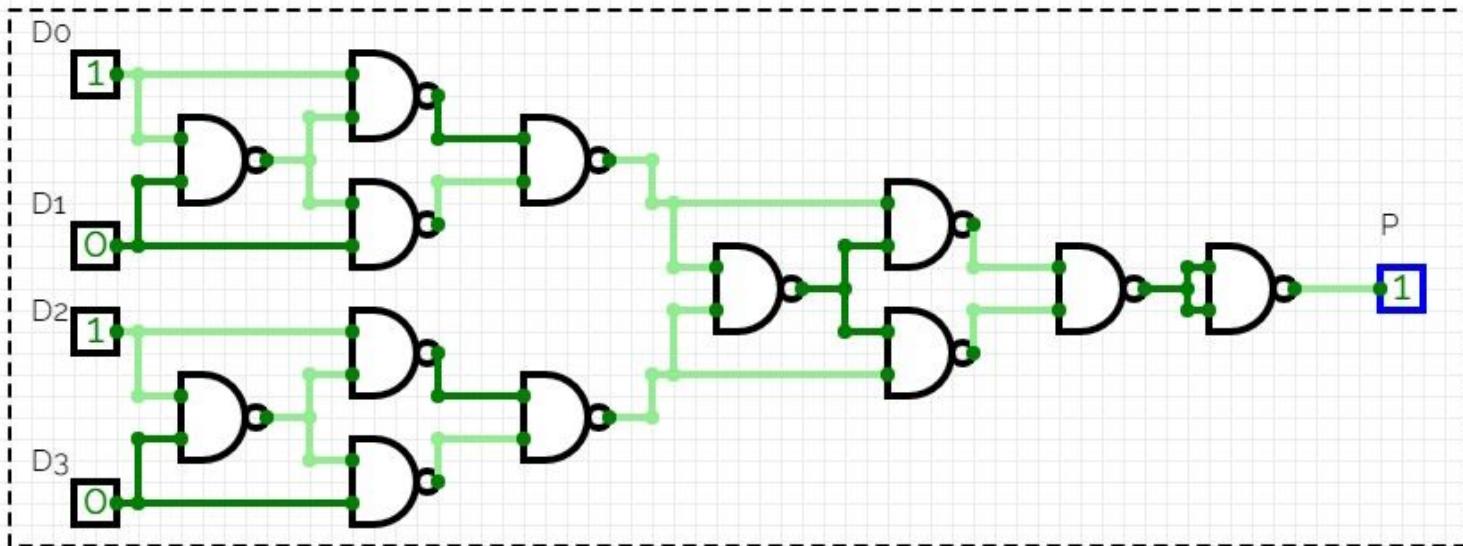


c) Conclusion:

The main advantage is that it is relatively simple & easy to use & will detect any error in one bit of transmission and due to any disturbance. However, if two bits are corrupted simultaneously, then it may not be able to detect the error. Also, it uses many gates.

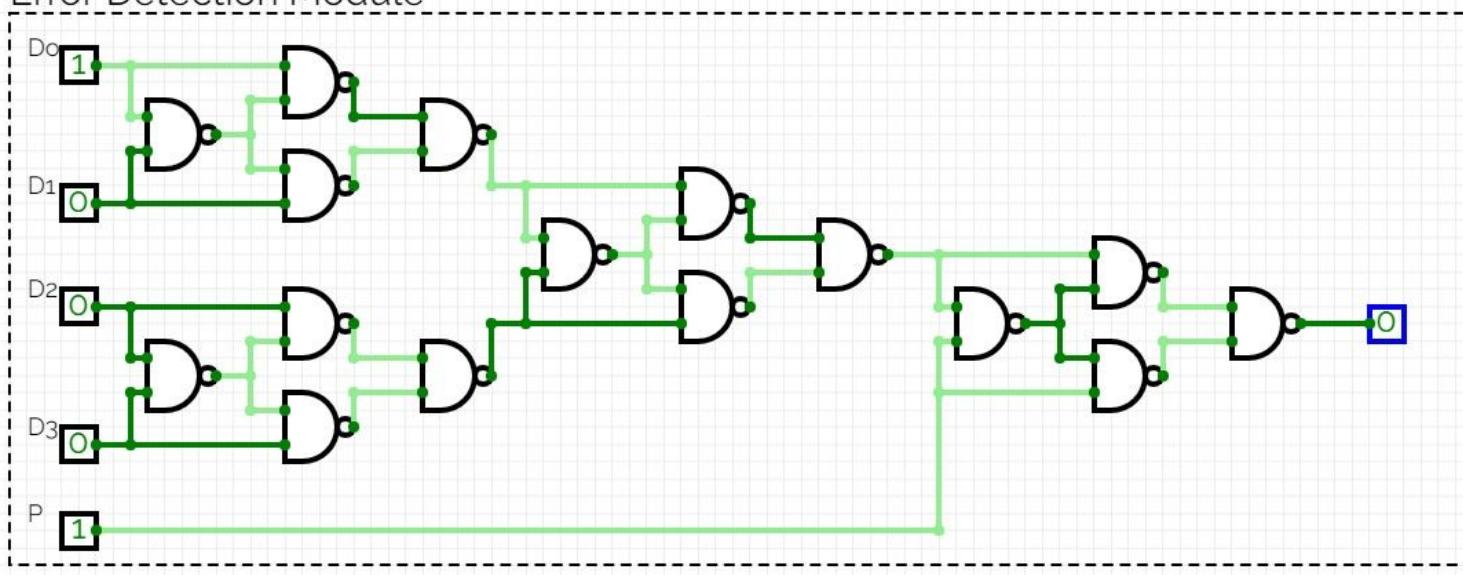
Test Case I: Parity Bit Generation

Parity Generation Module



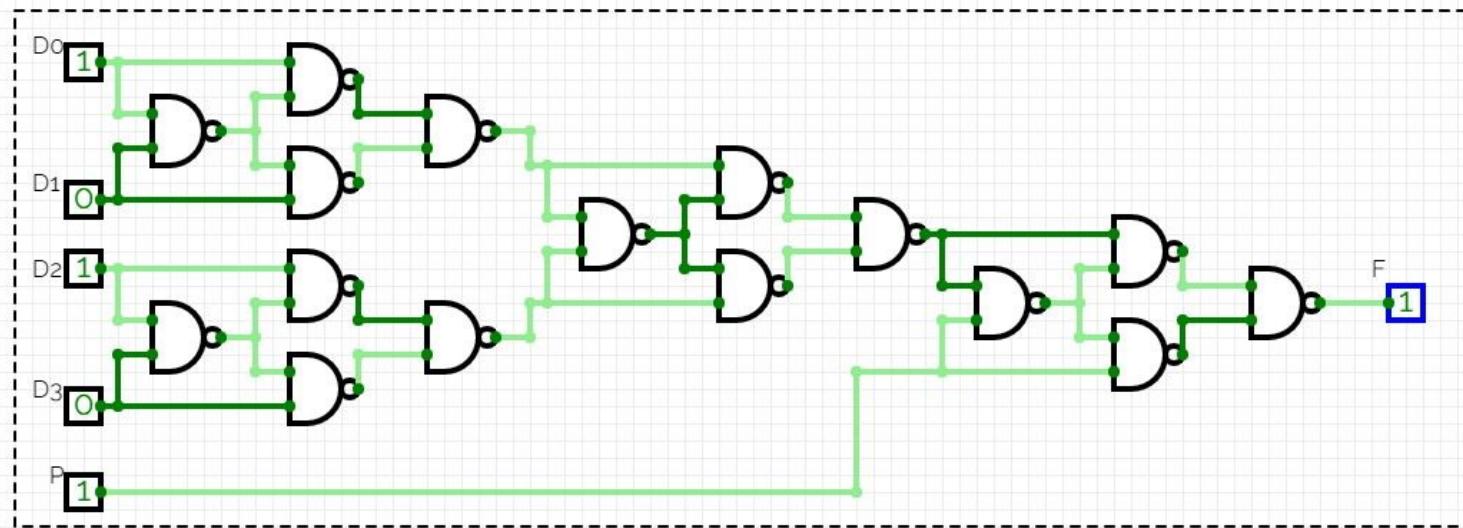
Test Case II: Flag Generation → No Error

Error Detection Module



Test Case III: Flag Generation → Error

Error Detection Module



Q3 (a)

Analysis Procedure i.e., fine the Boolean representation of the logic circuit and obtain its truth table.

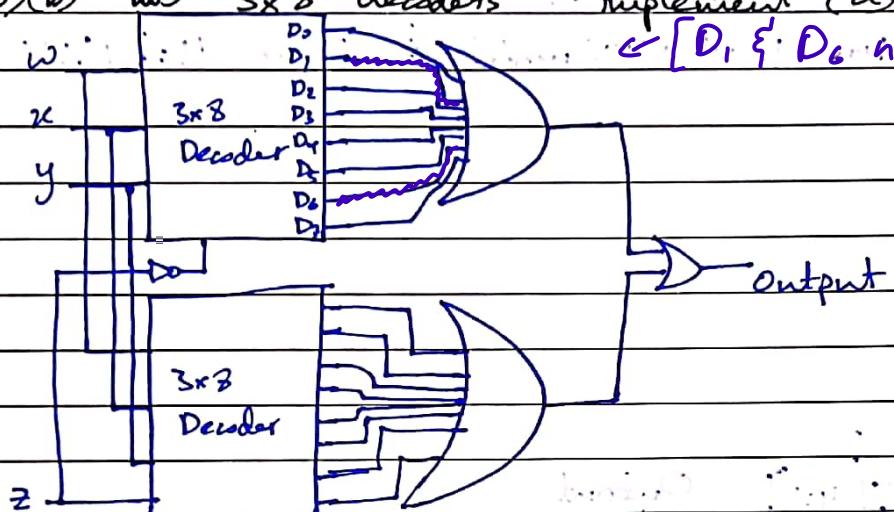
Solution:

$$\begin{aligned}
 &\implies [(w'(x \oplus z))(w'x' + w')(y \oplus z')]' \\
 &\implies [(w'(xz' + x'z))(w')(yz' + y'z)]' \\
 &\implies [(w'xz' + w'x'z)(w')(yz' + y'z)]' \\
 &\implies [(w'xz' + w'x'z)(yz' + y'z)]' \\
 &\implies [w'xyz' + w'xy'z'z + w'x'yzz' + w'x'y'z]' \\
 &\implies [w'xyz' + 0 + 0 + w'x'y'z]' \\
 &\implies [w'xyz' + w'x'y'z]' \\
 &\implies (w'xyz')'(w'x'y'z)' \\
 &\implies (w + x' + y' + z)(w + x + y + z') \implies \Pi(1, 6)
 \end{aligned}$$

| w | x | y | z | Output |
|---|---|---|---|--------|
| F | F | F | F | T |
| F | F | F | T | F |
| F | F | T | F | T |
| F | F | T | T | T |
| F | T | F | F | T |
| F | T | F | T | T |
| F | T | T | F | F |
| F | T | T | T | T |
| T | F | F | F | T |
| T | F | F | T | T |
| T | F | T | F | T |
| T | F | T | T | T |
| T | T | F | F | T |
| T | T | F | T | T |
| T | T | T | F | T |
| T | T | T | T | T |

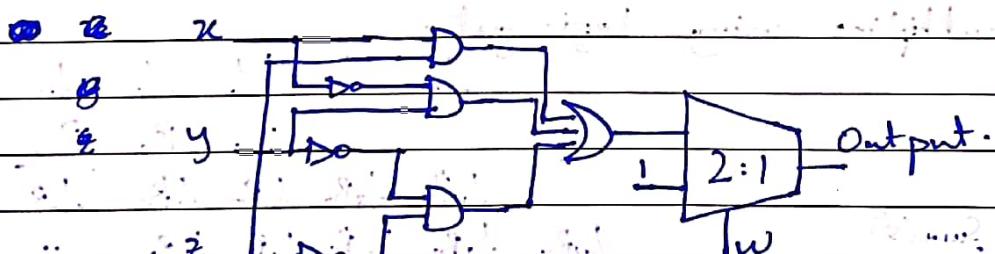
Dated:

Q3(b) two 3×8 decoders \rightarrow implement (a) $\Leftarrow [D_1 \& D_6 \text{ not being used}]$

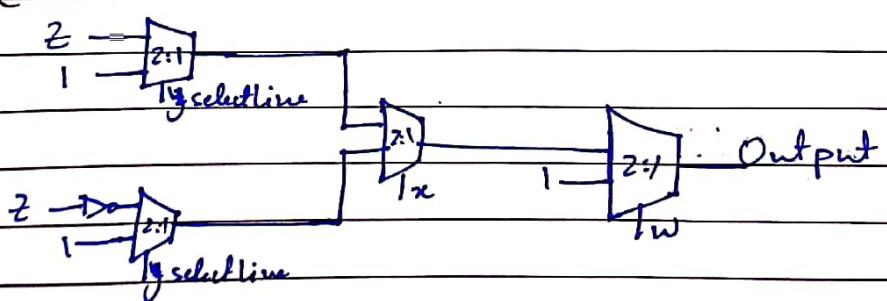


(c) 2×1 Multiplexer \rightarrow implement (a)

| wz | 00 | 01 | 11 | 10 | wz^1 | wz^0 |
|------|----|----|----|----|--------------------------|--------------------------------|
| 00 | 11 | 0 | 11 | 11 | $w + y'z'$ | $y'z' + xz + w^2x^2y$ |
| 01 | 1 | 11 | 10 | 0 | $w \Rightarrow y'z'$ | $y'z' + xz + \cancel{w^2x^2y}$ |
| 11 | 11 | 11 | 11 | 11 | $w \Rightarrow 1$ | \dots |
| 10 | 11 | 1 | 1 | 11 | { w is selection line} | \dots |



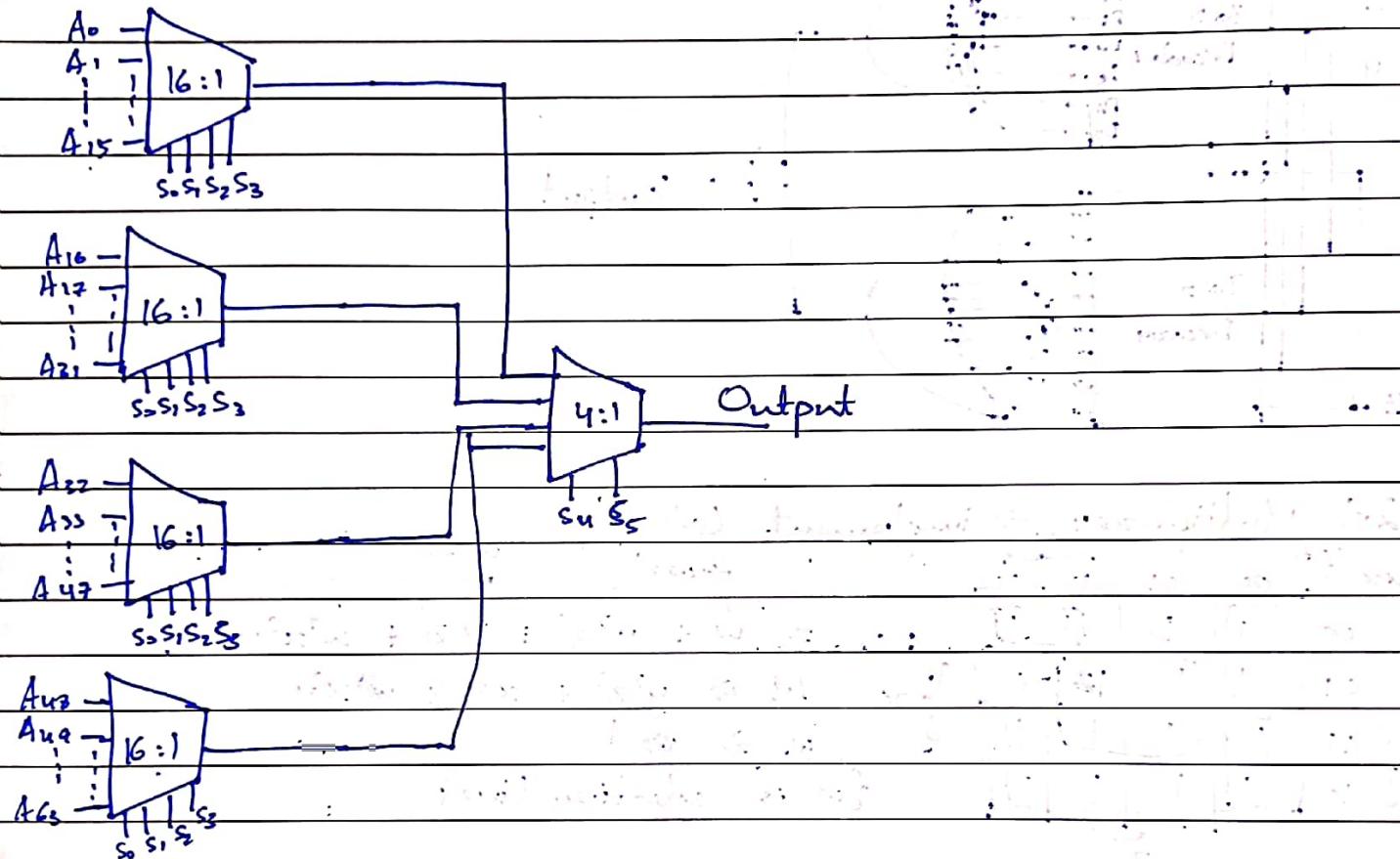
(d) 2×1 Mux & invertors.



Dated:

Q4) (a) 64x1 MUX using 16x1 MUX

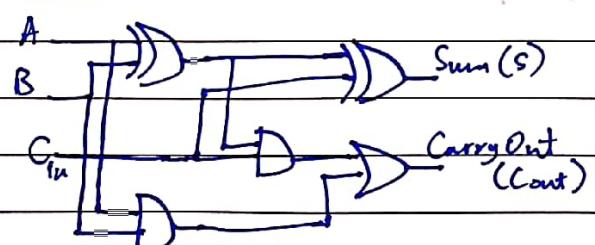
$64 \div 16 = 4 \Rightarrow$ Four 16x1 MUX needed & one 4:1 MUX



(b) 5 bit by 3bit multiplier using logic gates

B

Full-Adder:

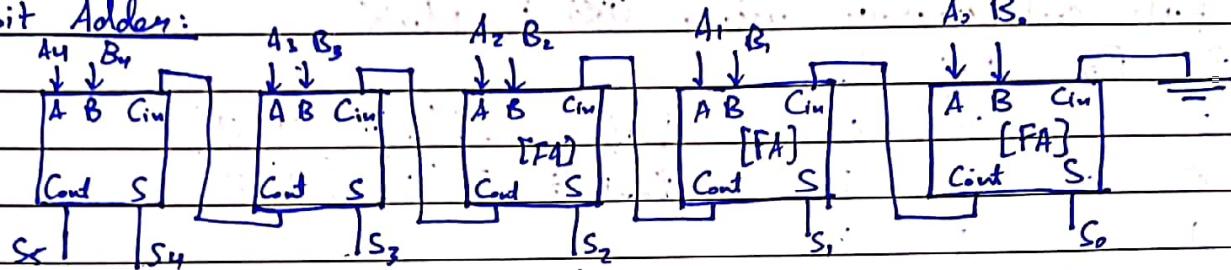


Let $\begin{bmatrix} A & B & C_{in} \\ [FA] & S \\ C_{out} & S \end{bmatrix}$ represent a Full Adder:

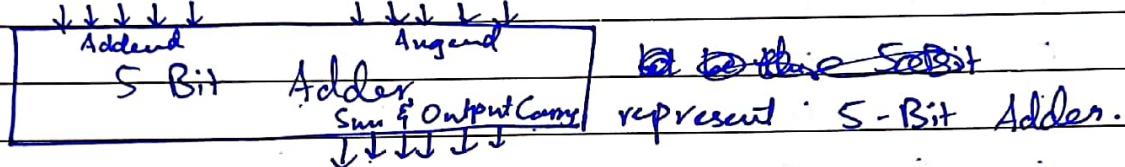
$$\begin{array}{r}
 1 \quad A_4 \quad A_3 \quad A_2 \quad A_1 \quad A_0 \\
 \times \quad B_3 \quad B_2 \quad B_1 \quad B_0 \\
 \hline
 A_4 B_3 \quad A_3 B_3 \quad A_2 B_3 \quad A_1 B_3 \quad A_0 B_3 \\
 A_4 B_2 \quad A_3 B_2 \quad A_2 B_2 \quad A_1 B_2 \quad A_0 B_2 \\
 A_4 B_1 \quad A_3 B_1 \quad A_2 B_1 \quad A_1 B_1 \quad A_0 B_1 \\
 A_4 B_0 \quad A_3 B_0 \quad A_2 B_0 \quad A_1 B_0 \quad A_0 B_0 \\
 \hline
 O_6 \quad O_5 \quad O_4 \quad O_3 \quad O_2 \quad O_1 \quad O_0
 \end{array}$$

Dated:

5-Bit Adder:

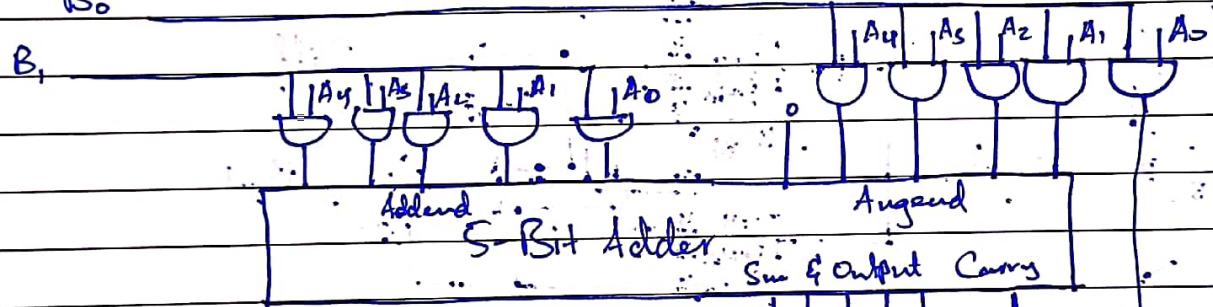


Let

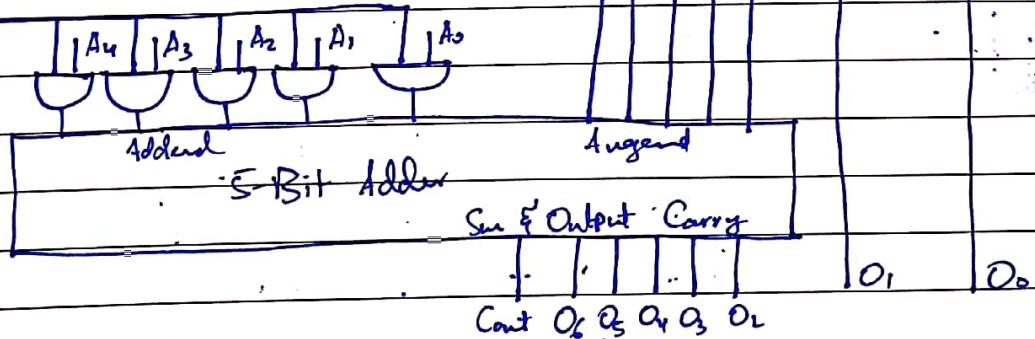


Then 5 bit by 3 bit Multiplier:

B₀



B₂



Dated:

(c) A full subtractor using ~~4:1~~ ^{1x4} DeMux

A B: Bi_i Diff B_{out}

0 0 0 0 0

$$\text{Diff} \rightarrow \sum(1, 2, 4, 7)$$

0 0 1 1 1

$$B_{out} \rightarrow \sum(1, 2, 3, 7)$$

0 1 0 1 1

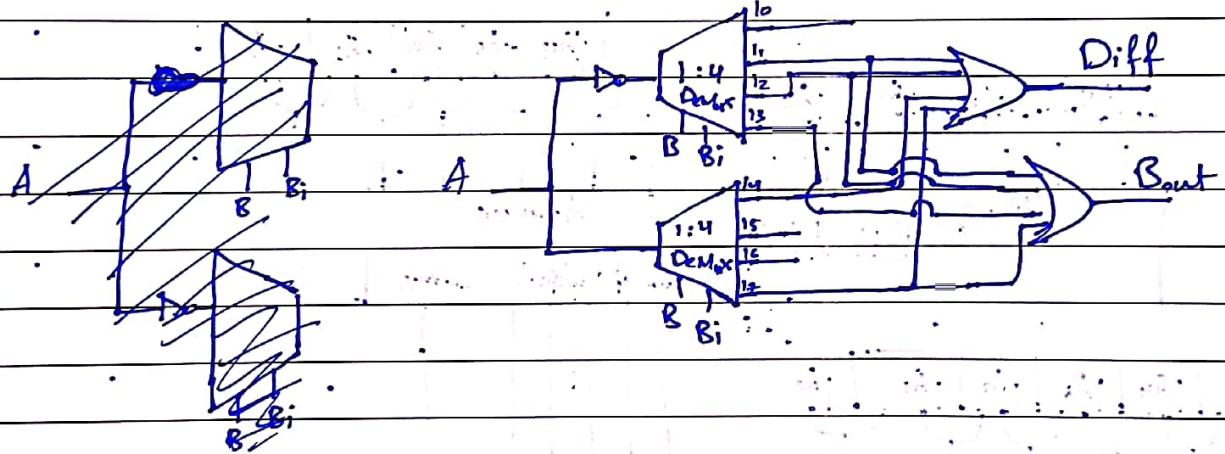
0 1 1 0 1

1 0 0 1 0

1 0 1 0 0

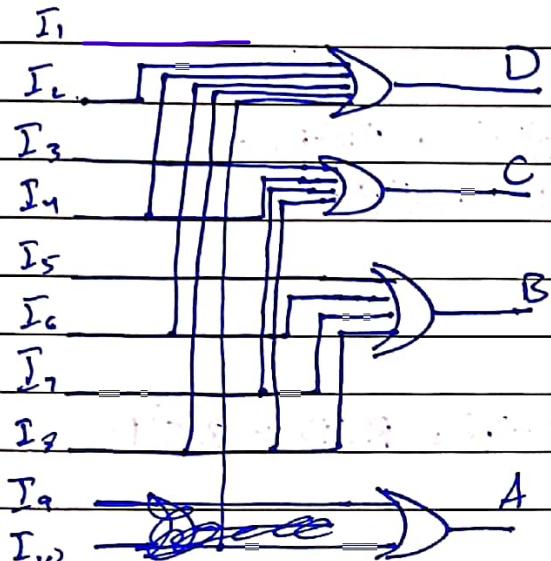
1 1 0 0 0

1 1 1 1 1



Dated:

$$Q5) \text{ (a)} \quad A = I_9 + I_{10} \quad B = I_5 + I_6 + I_7 + I_8 \\ C = I_1 + I_4 + I_7 + I_9 \quad D = I_2 + I_4 + I_6 + I_8 + I_9$$



Q5 (b)

The fault is that if multiple input lines are turned active, then the circuit will produce ambiguous results as the circuit is designed to work at a single input line. So multiple active input lines give ambiguous results. A Priority Encoder can be used to solve the problem as it will give priority to the input line with greater subscript. Also, the inputs can be labelled from $I_0 \dots I_9$ instead of $I_1 \dots I_{10}$ since the truth table represents Decimal to BCD conversion.

| I_0 | I_1 | I_2 | I_3 | I_4 | I_5 | I_6 | I_7 | I_8 | I_9 | A | B | C | D |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| x | x | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| x | x | x | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| x | x | x | x | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| x | x | x | x | x | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| x | x | x | x | x | x | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| x | x | x | x | x | x | x | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| x | x | x | x | x | x | x | x | 1 | 0 | 1 | 0 | 0 | 0 |
| x | x | x | x | x | x | x | x | x | 1 | 1 | 0 | 0 | 1 |

$$\begin{aligned} A &= \Sigma(8, 9) \\ \Rightarrow A &= I_9 + I_8 I'_9 \end{aligned}$$

$$\begin{aligned} B &= \Sigma(4, 5, 6, 7) \\ \Rightarrow B &= I_4 I'_5 I'_6 I'_7 I'_8 I'_9 + I_5 I'_6 I'_7 I'_8 I'_9 + I_6 I'_7 I'_8 I'_9 + I_7 I'_8 I'_9 \end{aligned}$$

$$\begin{aligned} C &= \Sigma(2, 3, 6, 7) \\ \Rightarrow C &= I_2 I'_3 I'_4 I'_5 I'_6 I'_7 I'_8 I'_9 + I_3 I'_4 I'_5 I'_6 I'_7 I'_8 I'_9 + I_6 I'_7 I'_8 I'_9 + I_7 I'_8 I'_9 \end{aligned}$$

$$\begin{aligned} D &= \Sigma(1, 3, 5, 7, 9) \\ \Rightarrow D &= I_1 I'_2 I'_3 I'_4 I'_5 I'_6 I'_7 I'_8 I'_9 + I_3 I'_4 I'_5 I'_6 I'_7 I'_8 I'_9 + I_5 I'_6 I'_7 I'_8 I'_9 + I_7 I'_8 I'_9 + I_9 \end{aligned}$$

Dated:

(Q6)(a) From the truth table,

$$\text{Group 1} = \sum(0, 3, 6, 9, 12, 15)$$

$$d = \text{Group 2} = \sum(1, 4, 7, 10, 13)$$

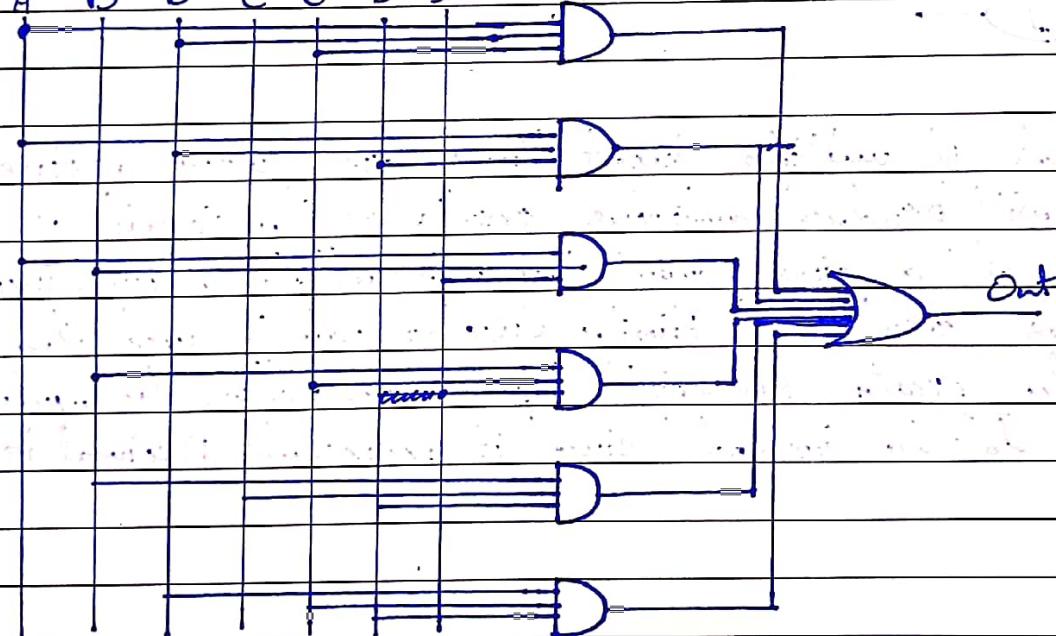
| AB\CD | 00 | 01 | 10 | 11 | |
|-------|----|----|----|----|---|
| 00 | 1 | X | 1 | 1 | |
| 01 | X | | X | 1 | B |
| 10 | 1 | X | 1 | X | |
| 11 | 1 | X | 1 | X | |

$$\text{Out} = A'B'C' + A'B'D +$$

$$A'BCD + ABCD + B'C'D$$

$$\text{Out} = A'B'C' + A'B'D + A'BD' + BC'D' + BCD + B'C'D$$

$$A' B B' C C' D D'$$



Dated:

(b) Let an ID be 4digit.

$$A = A_3 A_2 A_1 A_0 \dots B = B_3 B_2 B_1 B_0$$

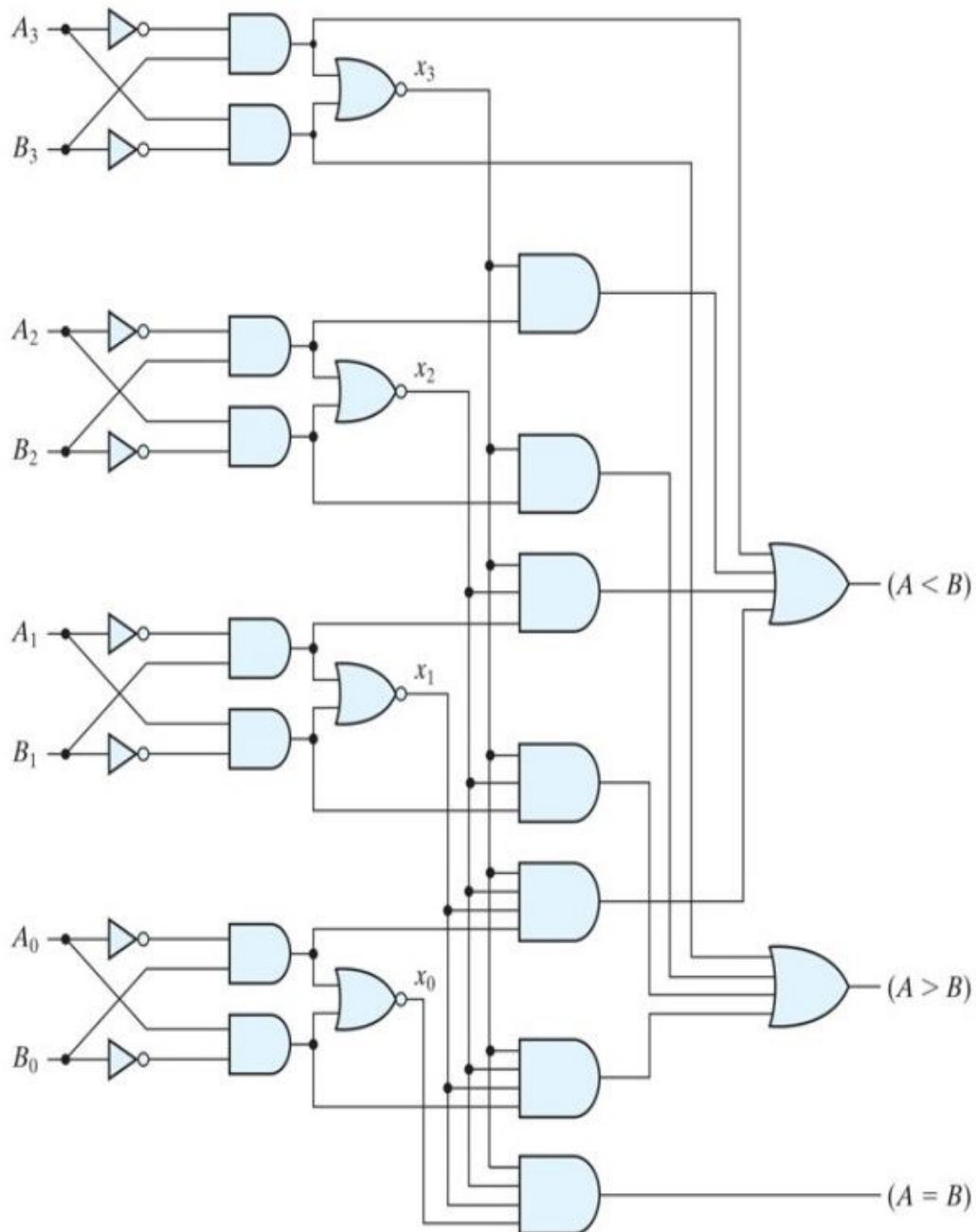
$A \geq B$ if $A_i \geq B_i$: $A_3 \geq B_3, A_2 \geq B_2, A_1 \geq B_1, A_0 \geq B_0$

$$x_i = A_i B_i + A_i' B_i' [i = 0, 1, 2, 3] \text{ [XNOR]}$$

$A \geq B$ if $x_3 x_2 x_1 x_0$

$$A > B : A_3 B_3' + x_3 A_2 B_2' + x_3 x_2 A_1 B_1' + x_3 x_2 x_1 A_0 B_0'$$

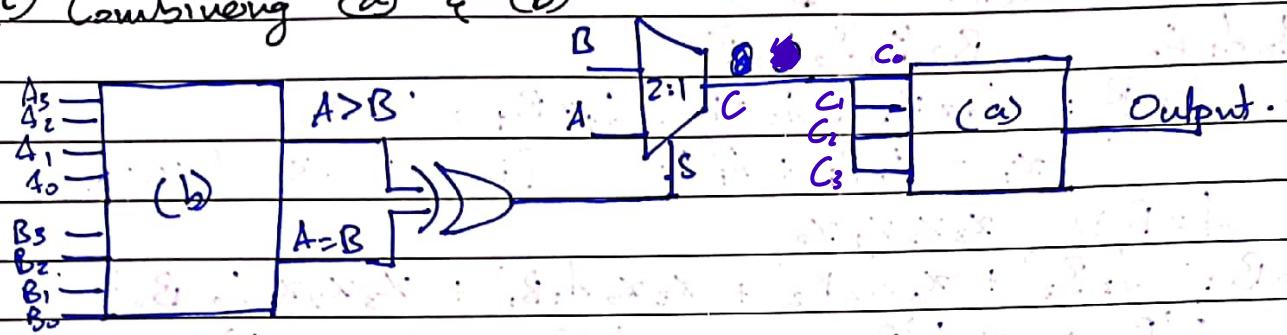
$$A < B : A_3' B_3 + x_3 A_2' B_2 + x_3 x_2 A_1' B_1 + x_3 x_2 x_1 A_0' B_0$$



Circuit inspired from reference book

Dated:

(c) Combining (a) & (b)



A is either $= B$ or $A > B$. If both are false or true, then hence the output of B is XORed & acts as selector line for selecting A or B . If $A > B$ & $A = B$ both are same, then neither is true & $B > A$ is true so B is chosen. Else A is chosen.

Then the output C is selected using a 2x1 MUX & sent to component from (a) which gives output accordingly.

(d) We can use multiple as (b) components to compare for multiple student IDs at the same time. The output of 2 IDs will be XORed & then compared with the third ID again in (b) Component. Then that will act as selector bit S_0 & S_1 for those 3 IDs. The same process repeats. Then the appropriate ID is sent to (a) Component.

The same flow process repeats for any n number of IDs & then the appropriate ID will be selected using n¹ MUX & sent to (a) Component.