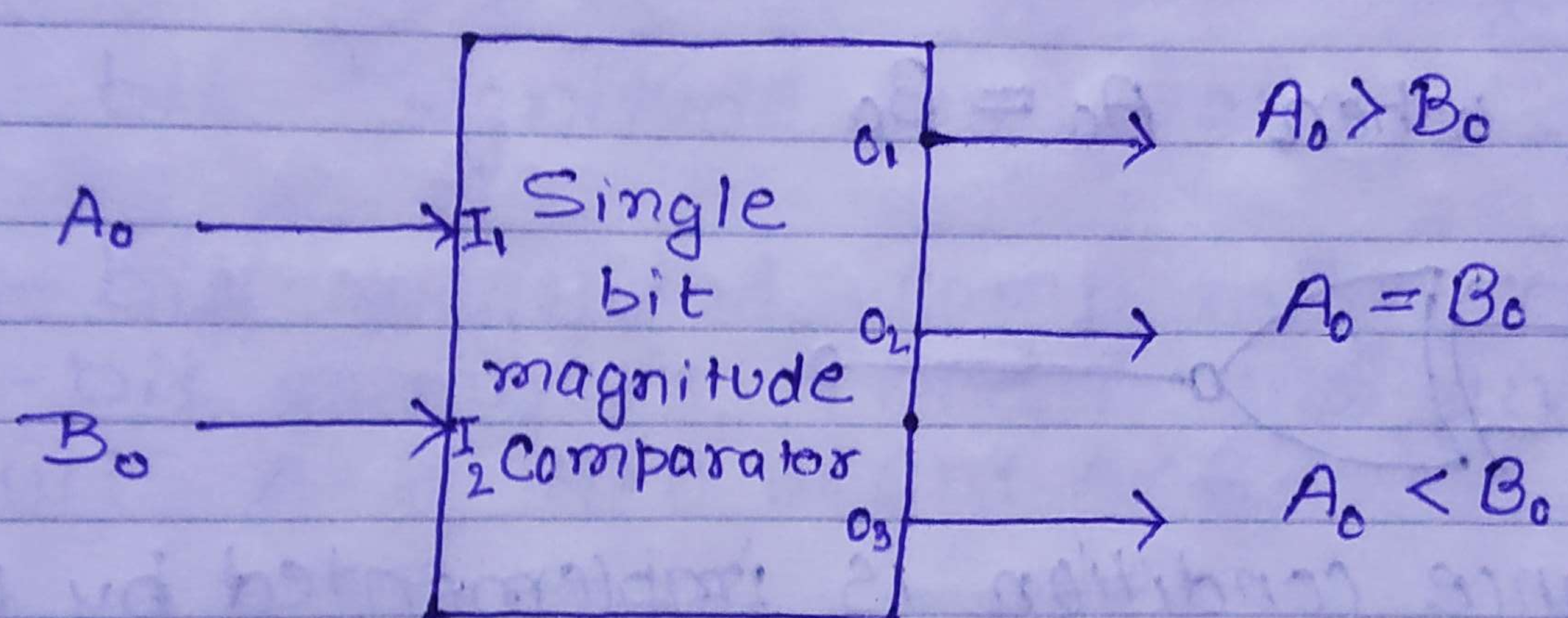
Magnitude Comparator

A Magnitude Comparator is a combinational circuit that compares the magnitude of two numbers A and B and generate one of the following outputs : $A=B$, $A<B$ and $A>B$.

The block diagram of a single - bit magnitude comparator is given below



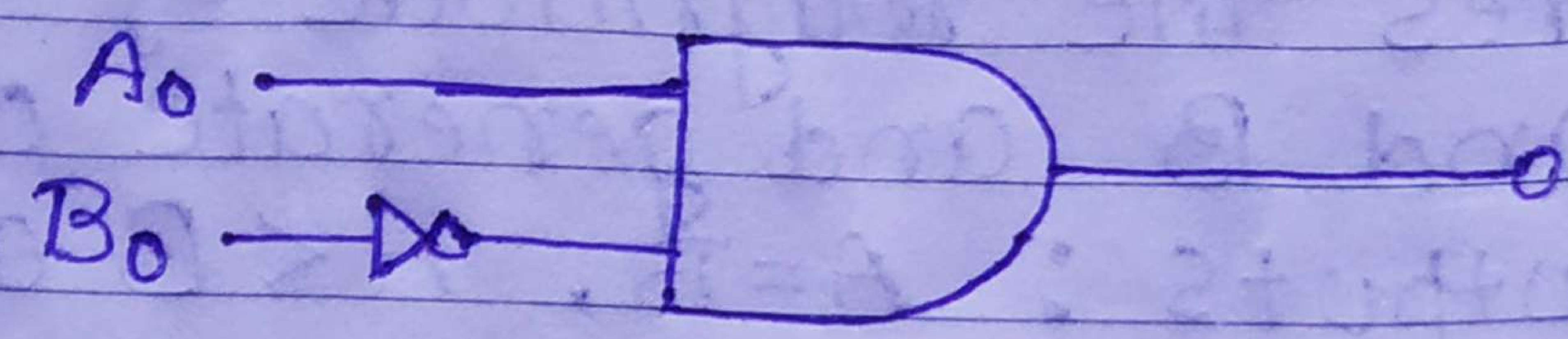
A_0 and B_0 will be provided at input port I_1 & I_2 respectively.

If A_0 is greater than B_0, i.e $A_0 > B_0$ then output port 1 (O_1) will be high i.e '1'

If A_0 is equal to B_0, ie $A_0 = B_0$, then output port 2 (O_2) will be high, i.e '1'

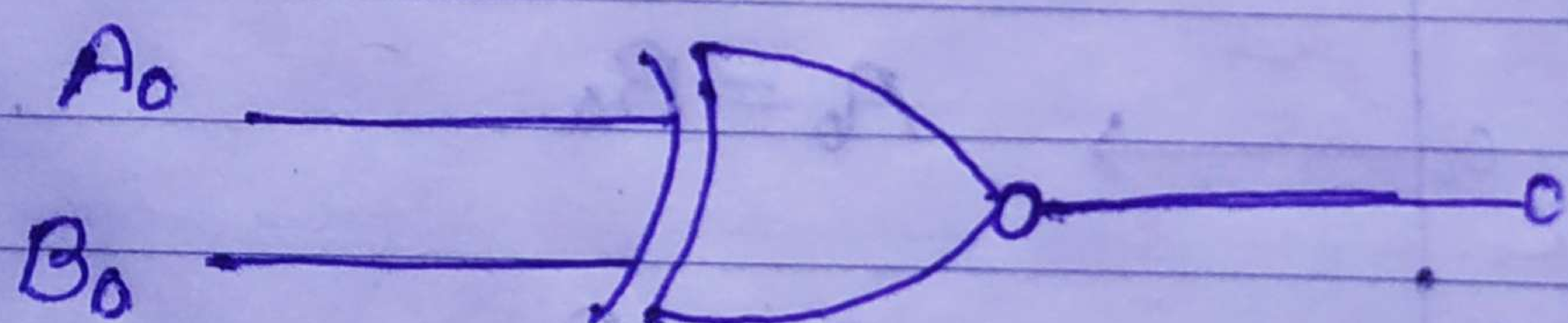
If A_0 is smaller than B_0, then output port 3 (O_3) will be high.

In order to implement the magnitude comparator
case I, when $A_0 > B_0$



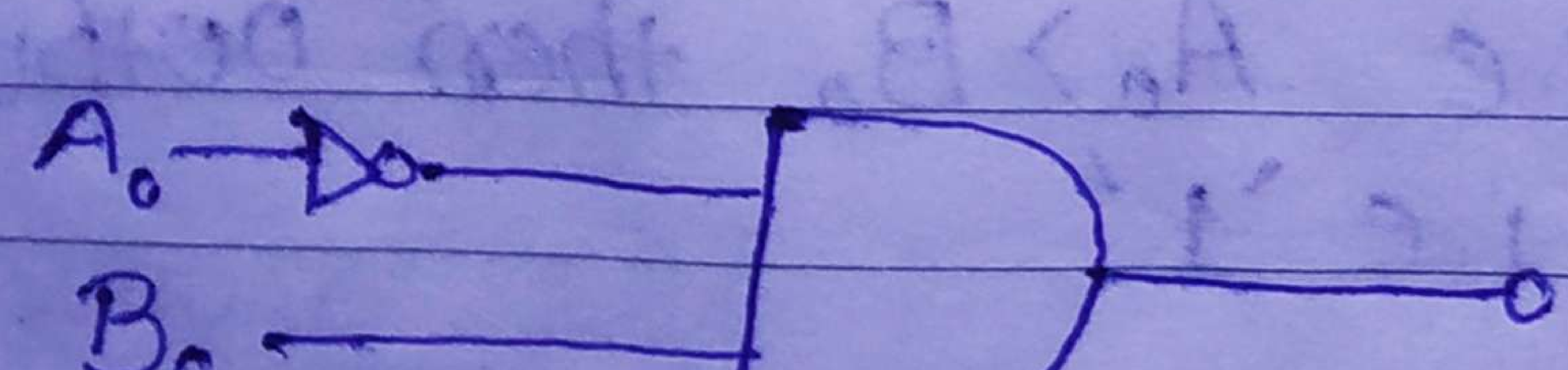
In the above logic setup output will be high only if $A_0 = 1$ and $B_0 = 0$ i.e $A_0 > B_0$

Case II, when $A_0 = B_0$

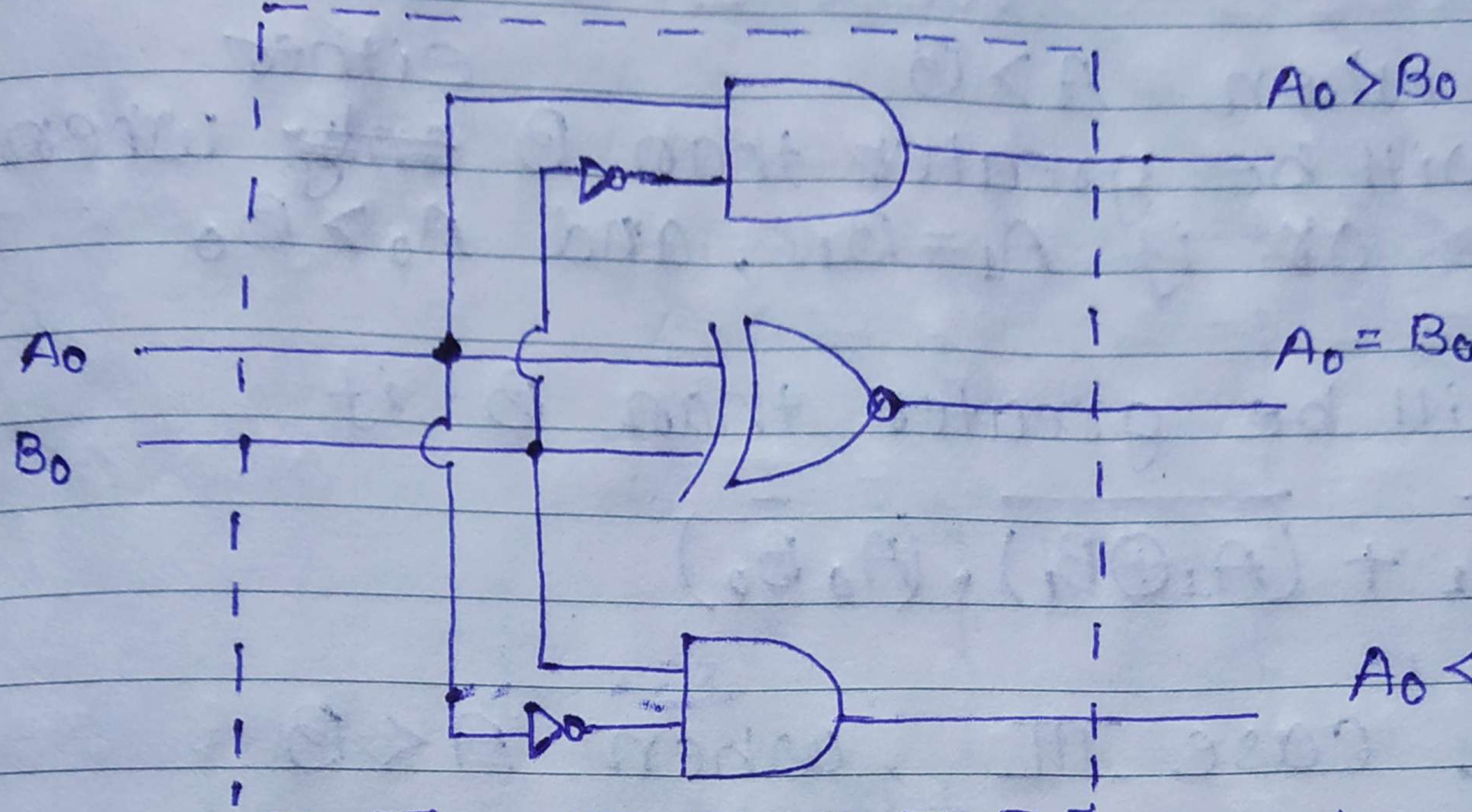


The equivalance condition is implemented by using EXNOR gate, as output of EX-NOR is '1' only when both inputs are equal.

case III, When $A_0 < B_0$



In the above logic setup output will be '1' only if $A_0 = 0$ and $B_0 = 1$ i.e $A_0 < B_0$



single bit magnitude comparator.

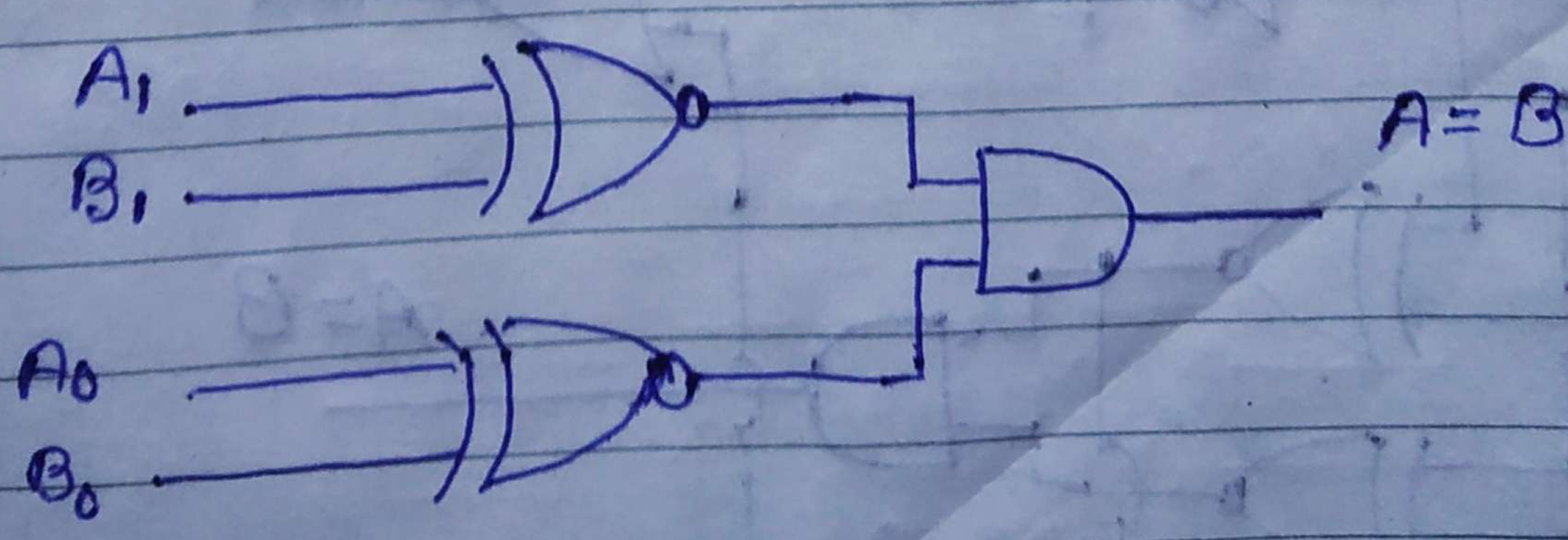
2-bit Magnitude Comparator.

A 2-bit magnitude comparator compares two 2-bit numbers A and B, and gives following result $A=B$, $A>B$ ~~or~~ and $A<B$.

Let $A = A_1 A_0$ (MSB A_1, LSB A_0)
$B = B_1 B_0$

Case I, when $A=B$
$A=B$, only when $A_1 = B_1$ and $A_0 = B_0$

i.e



Case II, when $A > B$ either

A will be greater than B ~~only~~ when $A_1 > B_1$ or if $A_1 = B_1$, and $A_0 > B_0$

i.e A will be greater than B if

$$A_1 \bar{B}_1 + \overline{(A_1 \oplus B_1)} \cdot (A_0 \bar{B}_0)$$

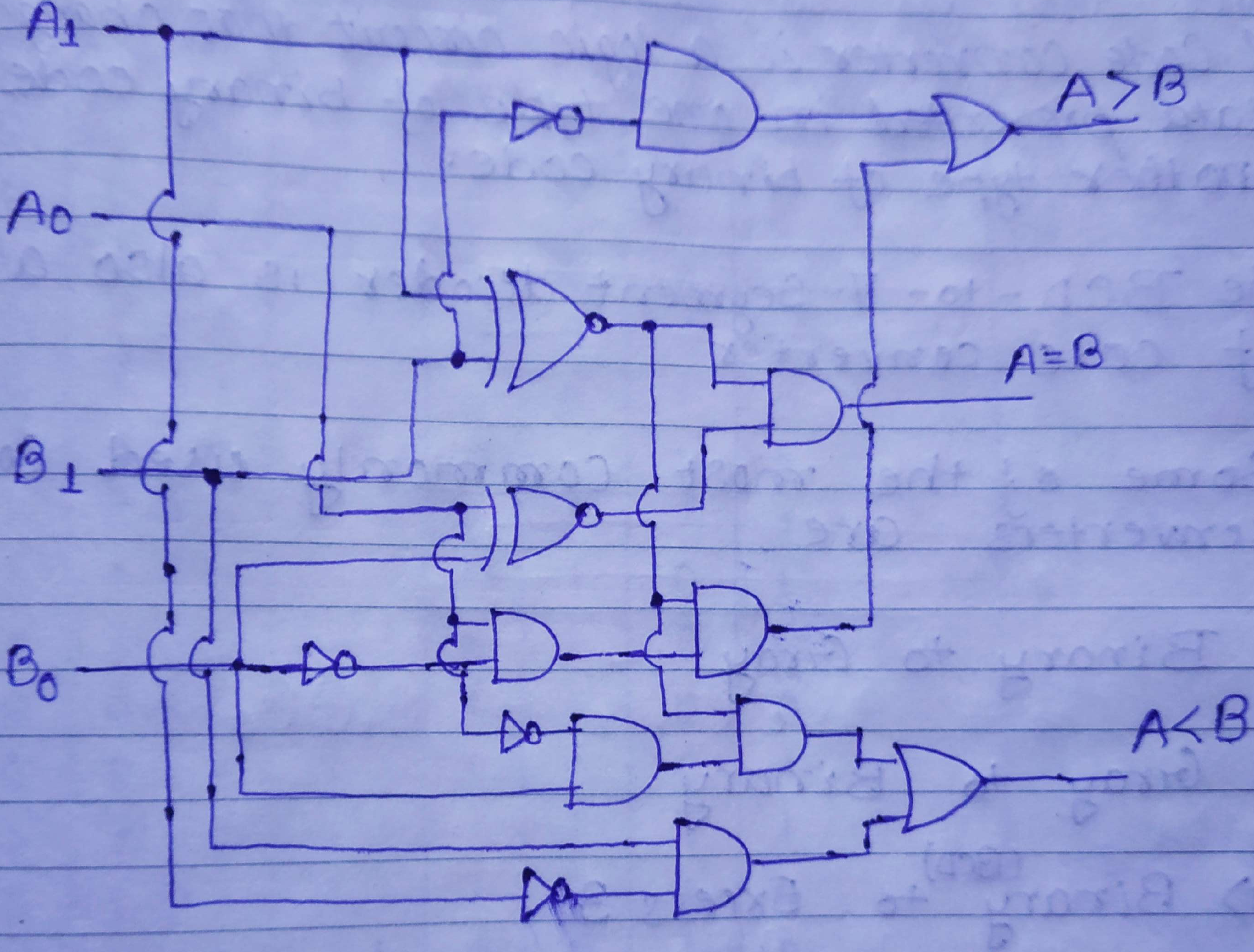
Similarly, case III, when $A < B$

Compare the MSB, if $A_1 < B_1$, then $A < B$

If $A_1 = B_1$, then compare LSB or next bit to MSB, if $A_0 < B_0$ then $A < B$

So $A < B$ if

$$\bar{A}_1 B_1 + \overline{(A_1 \oplus B_1)} \cdot (\bar{A}_0 \cdot B_0)$$



2-bit Magnitude Comparator.

CODE CONVERTERS

A Code converter is a logic circuit that changes data presented in one type of binary code to another type of binary code.

The BCD-to-7-Segment decoder is also a kind of code converter

Some of the most commonly used Code Converters are

i) Binary to Gray

ii) Gray to Binary

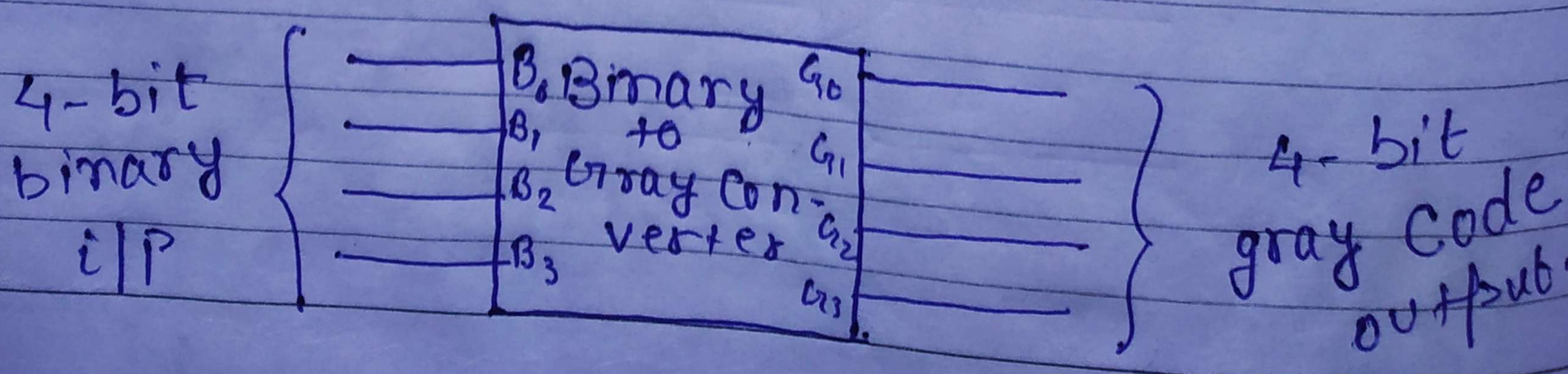
iii) Binary (BCD) to Excess 3

iv) ~~ASCII~~ ASCII to EBCDIC etc.

Binary to Gray Code Converters

In this code converter input will be in binary form, the converter will convert the binary data into Gray code.

SUNDAY



3

Truth table of binary to gray code converter

	Binary i/p				Gray Code output			
	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

From the truth table above, G_0, G_1, G_2 & G_3 can be expressed in minterm as

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

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

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

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

Simplifying for G_0, using K-map we get

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

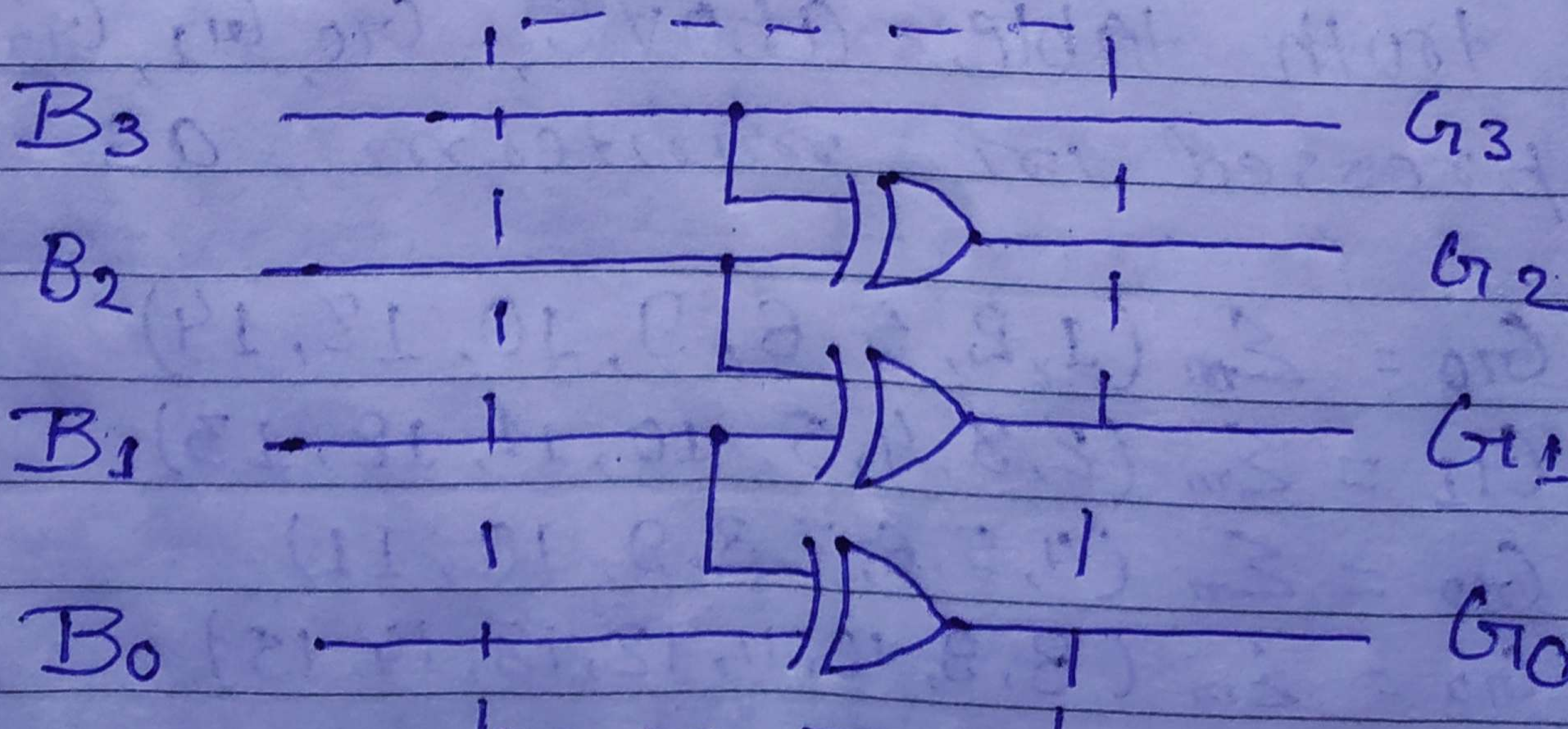
$$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0 = B_1 \oplus B_0$$

Similarly

$$G_1 = B_2 \oplus B_1$$

$$G_2 = B_3 \oplus B_2$$

$$G_3 = B_3$$



Logic diagram of 4-bit Binary to Gray Converter.

BCD to Excess-3 Code Converter

Truth Table of BCD to Excess-3 Code Converter is given below

	BCD input				Excess-3 Code output			
	B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0
10	1	0	1	0	d	d	d	d
11	1	0	1	1	d	d	d	d
12	1	1	0	0	d	d	d	d
13	1	1	0	1	d	d	d	d
14	1	1	1	0	d	d	d	d
15	1	1	1	1	d	d	d	d

As the input is BCD, so we need to consider the case only for 0 to 9, rest will be treated as don't care conditions.

So, E_0, E_1, E_2 & E_3 can be written in ~~tear~~ terms of minterm as

$$E_0 = \Sigma_m(0,2,4,6,8) + \Sigma_d(10,11,12,13,14,15)$$
$$E_1 = \Sigma_m(0,3,4,7,8) + \Sigma_d(10,11,12,13,14,15)$$
$$E_2 = \Sigma_m(1,2,3,4,9) + \Sigma_d(10,11,12,13,14,15)$$
$$E_3 = \Sigma_m(5,6,7,8,9) + \Sigma_d(10,11,12,13,14,15)$$

Solving E_0, using K-map, we get

B_1B_0 \ B_3B_2	00	01	11	10
00	1_{0}	1_{4}	d_{12}	1_{8}
01	$_{1}$	$_{5}$	d_{13}	$_{9}$
11	$_{3}$	$_{7}$	d_{15}	d_{11}
10	1_{2}	1_{6}	d_{14}	d_{10}

$$E_0 = \overline{B_0}$$

Solving for E_1

B_1B_0 \ B_3B_2	00	01	11	10
00	1_{0}	1_{4}	d_{12}	1_{8}
01	$_{1}$	$_{5}$	d_{13}	$_{9}$
11	1_{3}	1_{7}	d_{15}	d_{11}
10	$_{2}$	$_{6}$	d_{14}	d_{10}

$$E_1 = \overline{B_1}\,\overline{B_0} + B_1B_0$$
$$= B_1 \odot B_0$$

Solving for E_2

B_1B_0 \ B_3B_2	00	01	11	10
00	0	1 4	d 12	8
01	1 1	5	d 13	1 9
11	1 3	7	d 15	d 11
10	1 2	6	d 14	d 10

$$E_2 = \overline{B_2}B_0 + \overline{B_2}B_1 + B_2\overline{B_1}\,\overline{B_0}$$

Solving for E_3

B_1B_0 \ B_3B_2	00	01	11	10
00	0	4	d 12	1 8
01	1	1 5	d 13	1 9
11	3	1 7	d 15	d 11
10	2	1 6	d 14	d 10

$$E_3 = B_3 + B_2B_0 + B_2B_1$$

Logic diagram of BCD to Excess-3 Code converter