Bimal parajuli (20BDS0405)

DIGITAL LOGIC DESIGN EXP #5

ADDERS, SUBTRACTORS AND MAGNITUDE COMPARATORS

Objectives:

- To construct and test various adders and subtractor circuits.
- To construct and test a magnitude comparator circuit.

Apparatus:

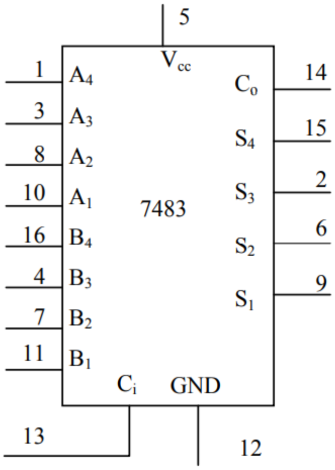
- IC type 7486 quad 2-input XOR gates
- IC type 7408 quad 2-input AND gates
- IC type 7404 HEX inverter
- IC type 7483 4-bit binary adder
- IC type 7485 4-bit magnitude comparator.

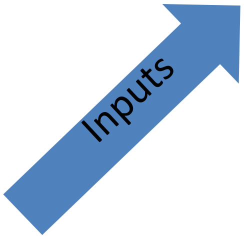
Softwares Used:

- LogicWorks 5
- Proteus 8 professional

a) *Addition:*

IC type 7483 is a 4-bit binary adder with fast carry. The pin assignment is shown in Fig 1. The two 4-bit input binary numbers are A_1 through A_4 and B_1 through B_4. The 4-bit sum is obtained from S_1 through S_4. C_i is the input carry and C_o the out carry. This IC can be used as an adder-subtractor as a magnitude comparator.





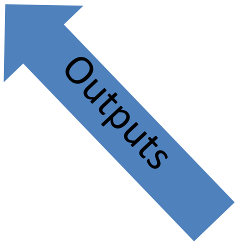
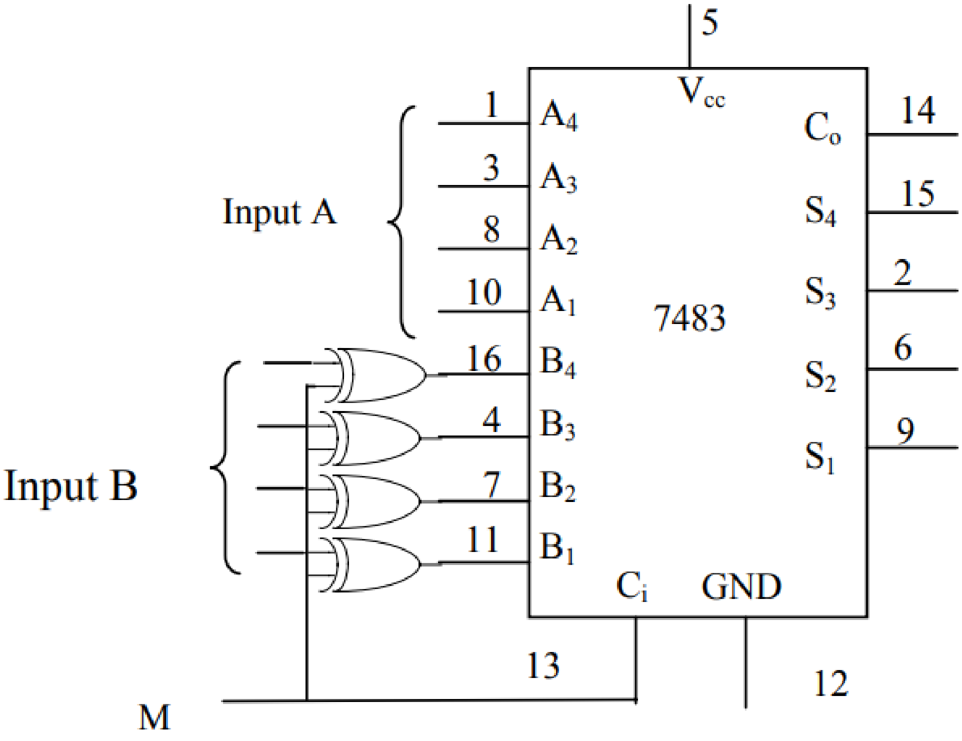


Fig.1 IC type 7483 4-bit adder

b) *Subtraction:*

The subtraction of two binary numbers can be done by taking the 2’s complement of the subtrahend and adding it to the minued. The 2’s complement can be obtained by taking the 1’s complement and adding 1.

To perform **A - B**, we complement the four bits of B, add them to the four bits of A, and add 1 to the input carry. This is done as shown in **Fig 2.**



M = 0 for add and M = 1 for subtract

Fig. 2 4-bit adder/subtractor

Four **XOR** gates **complement the bits of B** when the mode select **M = 1** (because x ⊕ 1=x’) and **leave the bits of B unchanged when M = 0** (because x ⊕ 0=x) thus, when the mode select M is equal to 1, the input carry C_i is equal to 1 and the sum output is A plus the 2’s complement of B. When M is equal to 0, the input carry is equal to 0 and the sum generates A + B.

c) *Magnitude comparison*

The comparison of two numbers is an operation that determines whether one number is greater than, equal to, or less than the other number.

The **IC 7485** is a **4 bit magnitude comparator**. It compares two 4-Bit binary numbers (labeled as A&B) generates an output of 1 at one of three outputs labeled A > B, A < B, A = B. Three inputs are available for cascading comparators. See Fig.3.

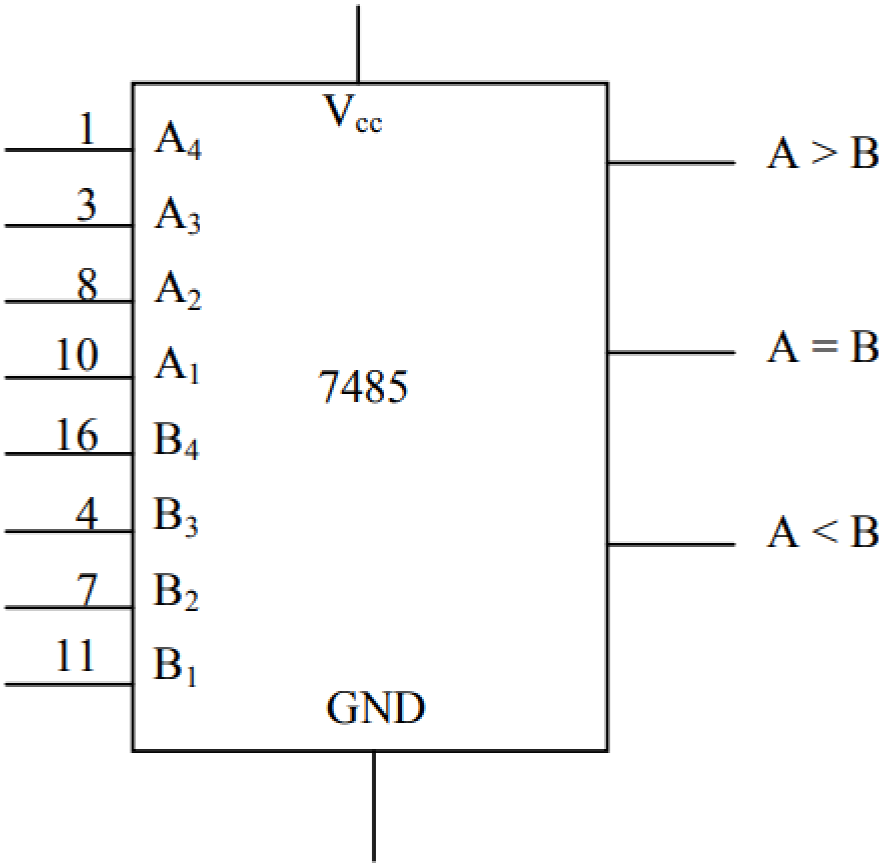


Fig. 3 4-bit magnitude comparator

Procedure:

a) ***Design using LogicWorks a half adder circuit using only XOR gates and NAND gates. Then during the Lab construct the circuit and verify its operation.***

Following is the truth table for the Half Adder circuit:

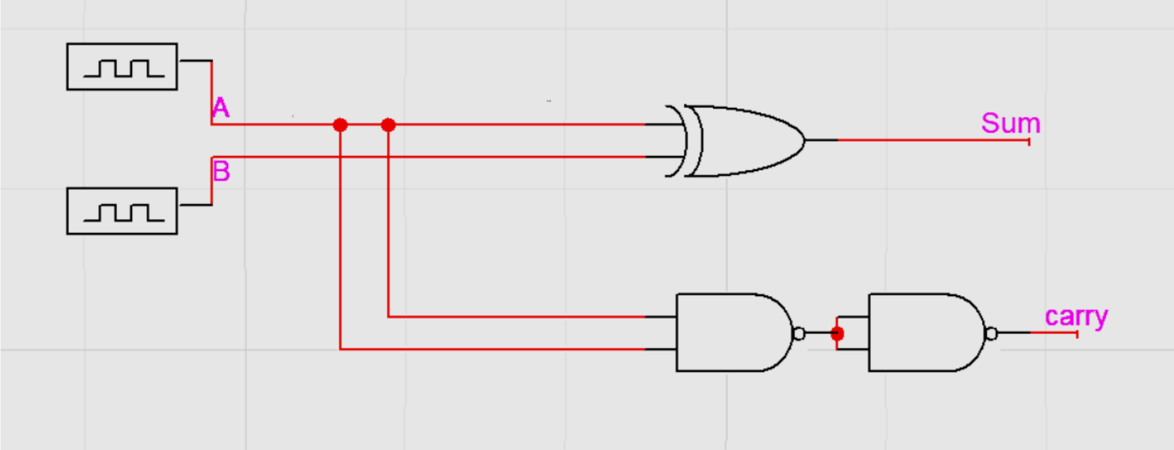
Input(A)	Input(B)	Carry (C)	Sum(S)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Here, C= A.B

C = $\overline{(\overline{A.B})}$ [Using Demorgan's law]

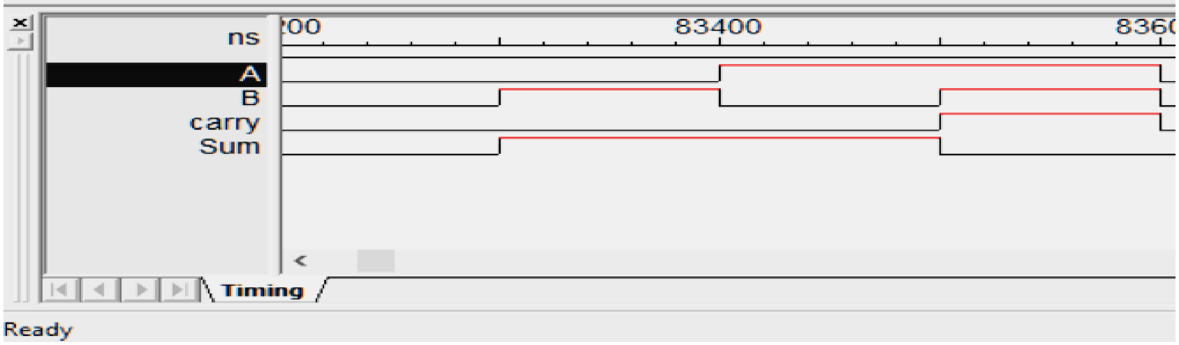
And, S=A XOR B

The corresponding logical circuit OF HALF ADDER is:



Half adder Circuit using NAND and XOR

The result of Simulation is:



b) ***Design using LogicWorks a full adder circuit using only XOR gates and NAND gates. Then during the Lab construct the circuit and verify its operation.***

Following is the truth table for the Half Adder circuit:

Input(A)	Input(B)	Carry-In (C)	Carry Out (C_o)	Sum(S)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Now,

\BC A\	00	01	11	10
00	0	0	1	0
01	0	1	1	1

\BC A\	00	01	11	10
00	0	1	0	1
01	1	0	1	0

From above Kmap:

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

$= A.B + B.C + A.C$ [Solved from Karnaugh Map]

$= \overline{(\overline{A.B}\ .\overline{B.C}.\overline{C.A})}$ [De Morgan's Law]

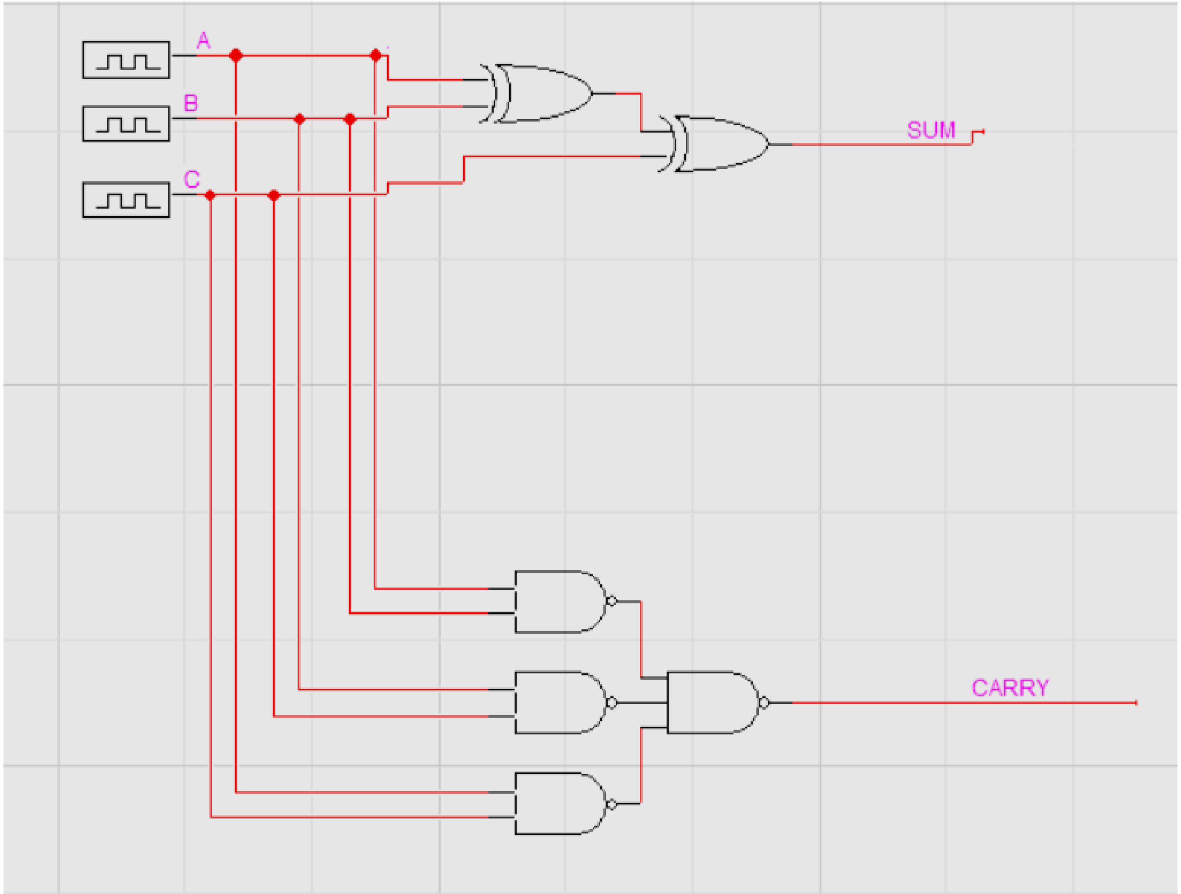
$S = \bar{A}.\bar{B}.C + \bar{A}.B.\bar{C} + A.\bar{B}.\ \bar{C} + A.B.C$

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

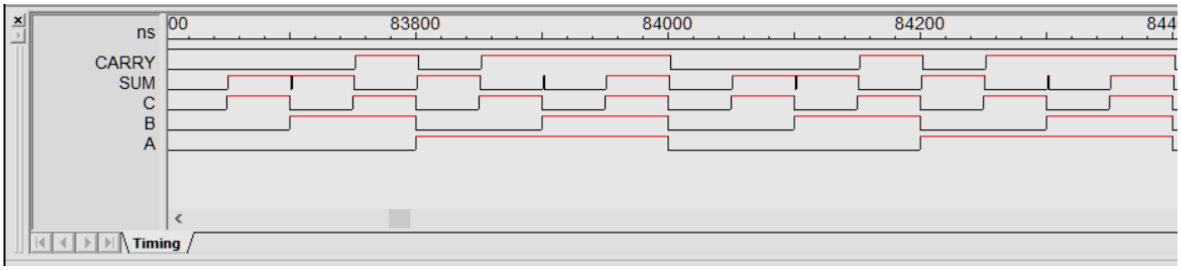
$= \bar{A}$ (B xor C) + A$\overline{\text{(B xor C)}}$

= A xor B xor C

The corresponding logical circuit of FULL ADDER is:



The result of Simulation is:



c) ***Use IC 7483 to add the two 4-bit numbers A and B shown in Table1. In LogicWorks, select the chip 74-83 and use Binary switches for the bits of the two numbers and the input carry and use Binary Probe for the sum and carry out.***

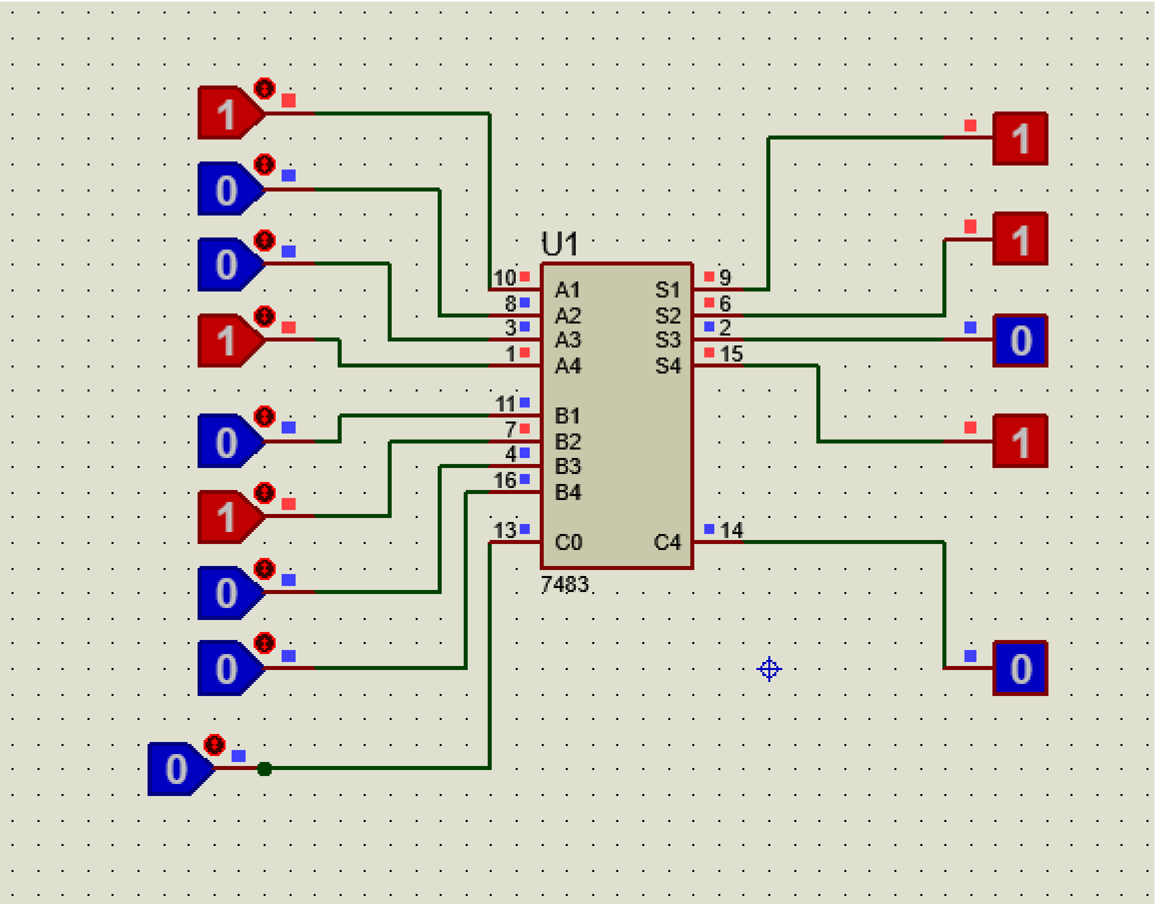
Input carry Ci is taken as logic 0. Show that if the input carry is 1, it adds 1 to the output sum.

In the Lab use switches S1-1 to S1-8 for the two numbers and use the SPDT S2 for the input carry Ci. For sum and carry out, use LED-1 to LED-5.

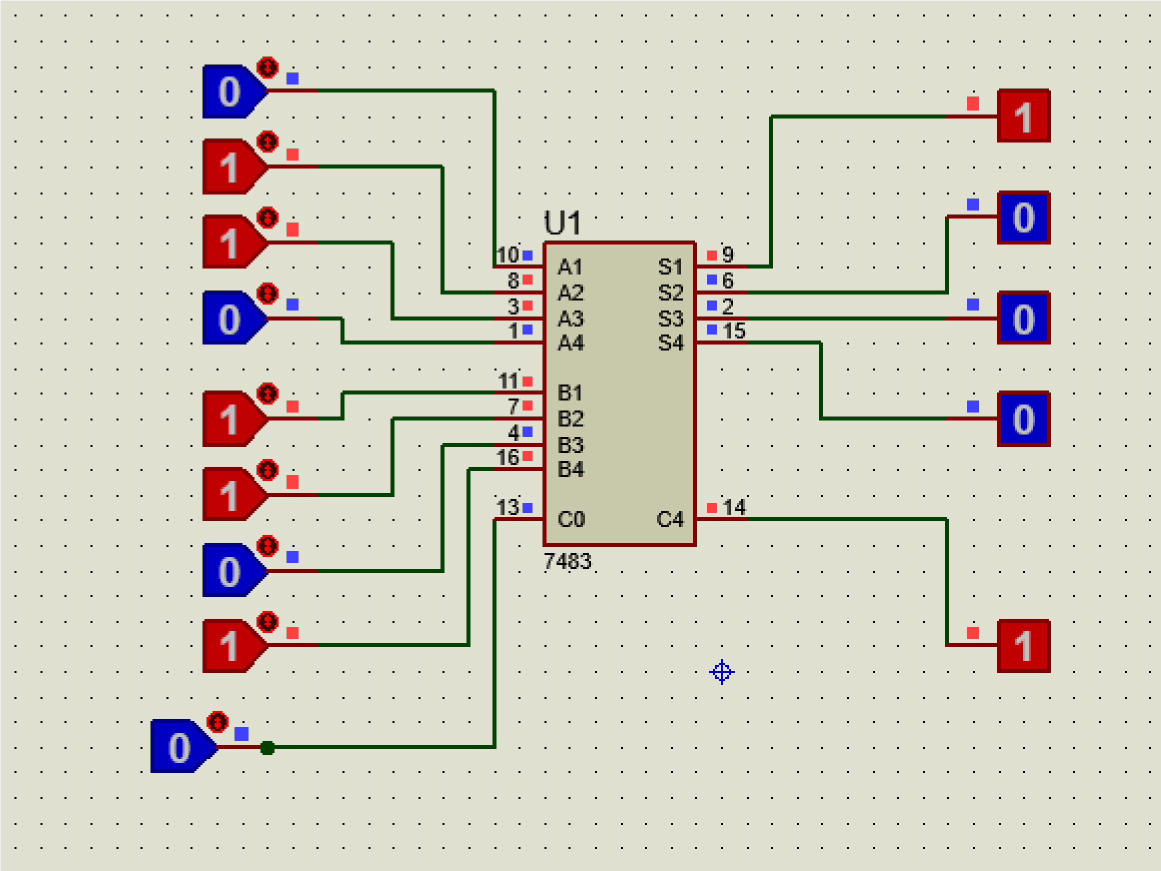
Table 1.

A4	**A3**	**A2**	**A1**	**B4**	**B3**	**B2**	**B1**	**Sum**				**Carry out**
1	0	0	1	0	0	1	0	1	0	1	1	0
0	1	1	0	1	0	1	1	0	0	0	1	1
1	1	0	0	1	0	1	0	0	1	1	0	1

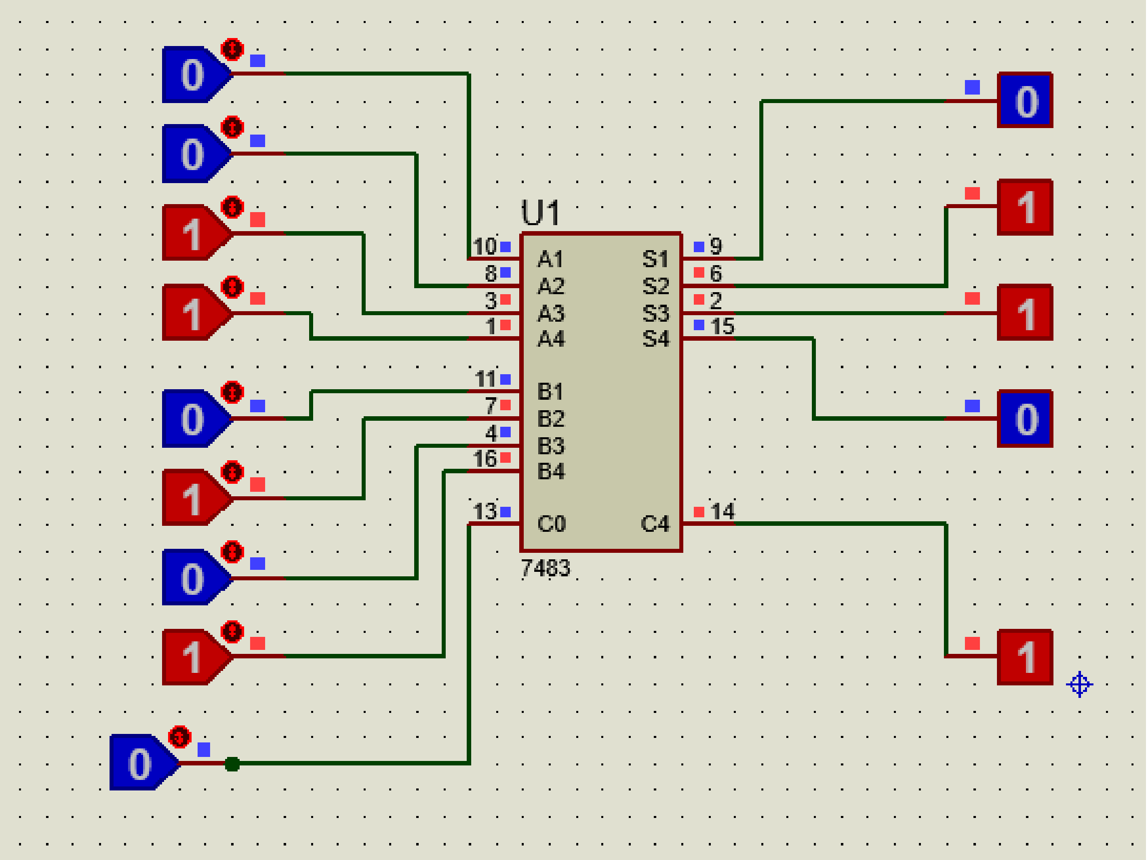
1 0 0 1 + 0 0 1 0



0 1 1 0 + 1 0 1 1



1 1 0 0 + 1 0 1 0

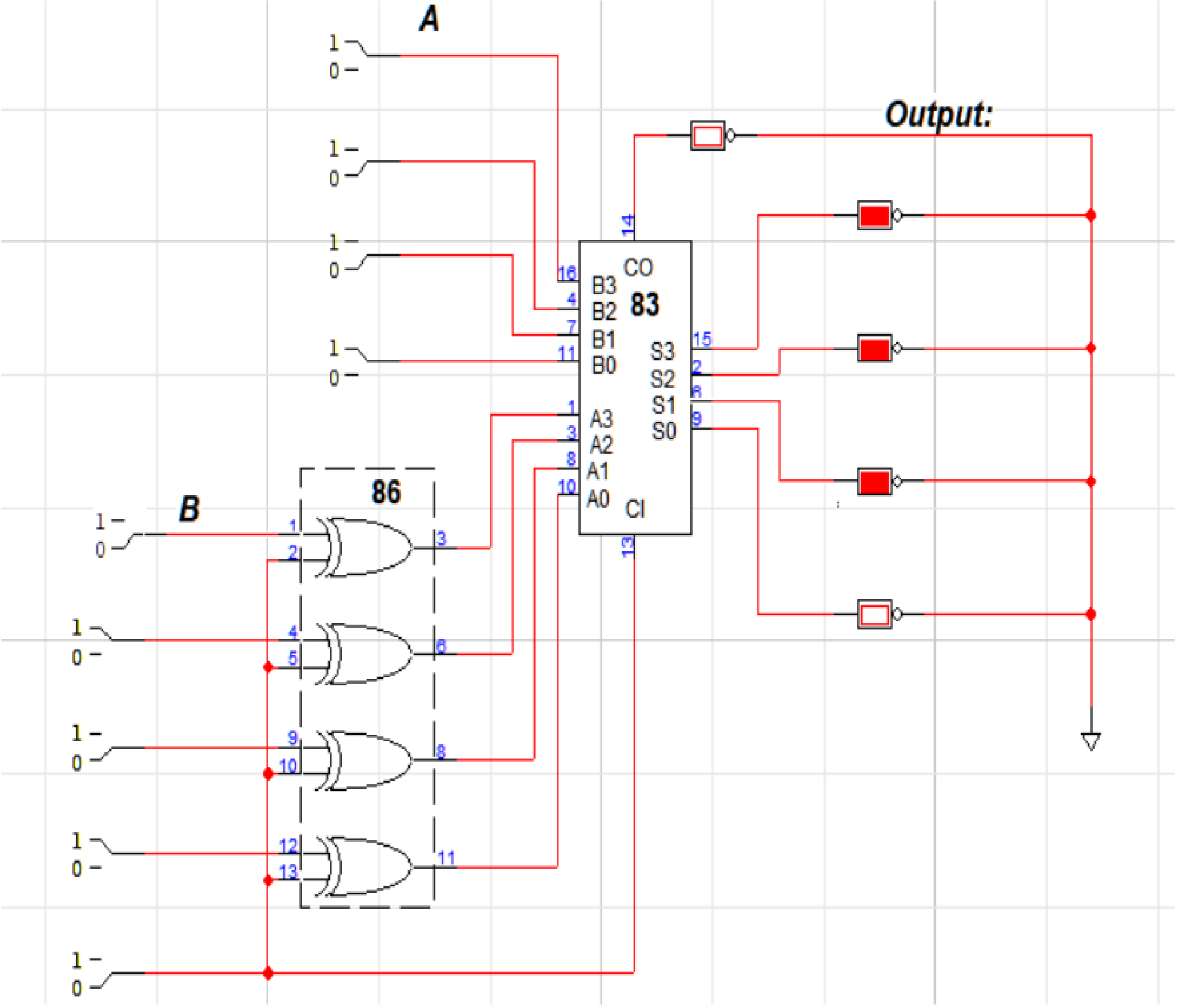


d) ***Connect the adder-subtractor circuit as shown in Fig 2. Perform the following operations and record the values of the output sum and the output carry Co.***

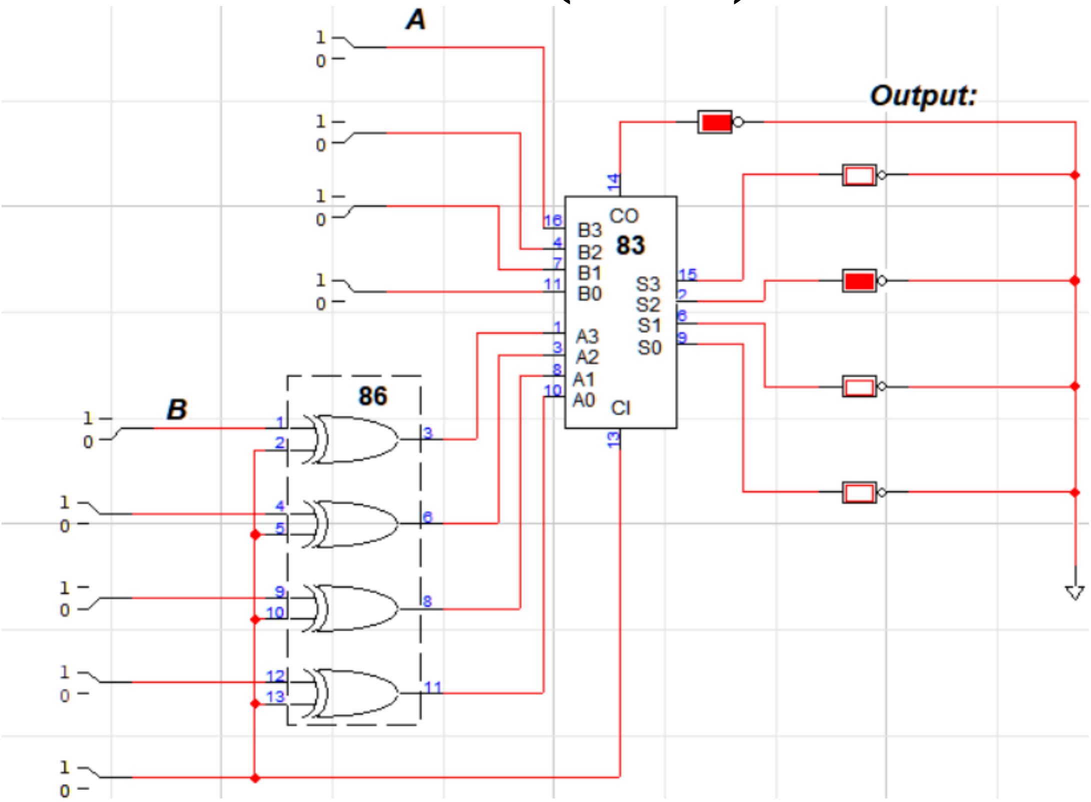
Table 2.

Decimal	**Carry**	**Output sum**			
A B	**Out**				
9 + 5	0	1	1	1	0
9 - 5	0	0	1	0	0
9 + 13	1	0	1	1	0
9 - 9	0	0	0	0	0
10 + 6	1	0	0	0	0
6 - 10	0	1	1	0	0

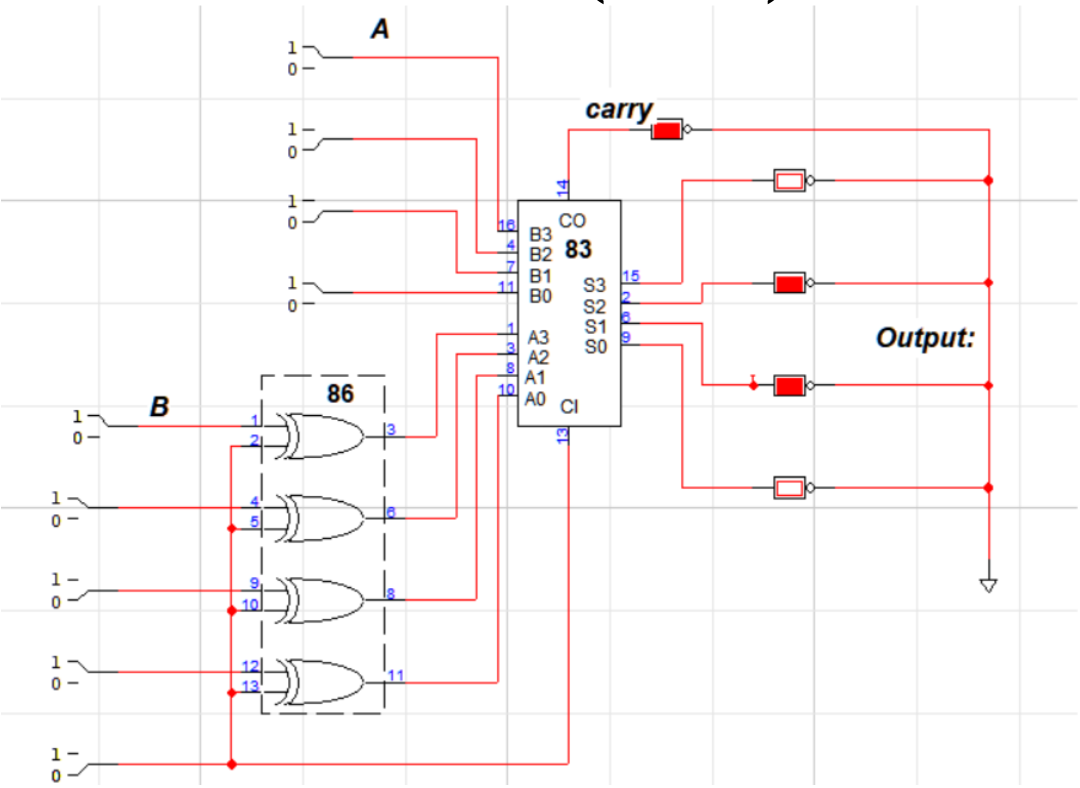
9 + 5 = 14(01110):



9 - 5 = 4 (10100):

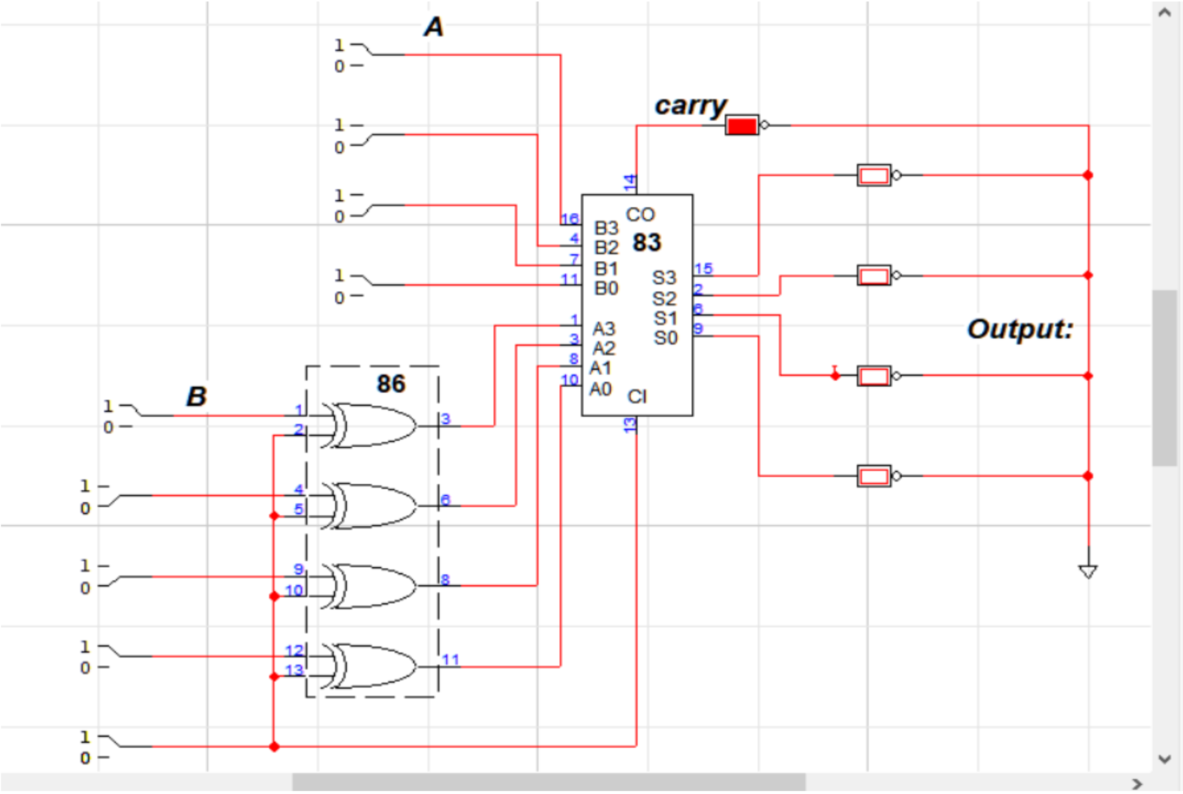


9 + 13 = 22 (10110):

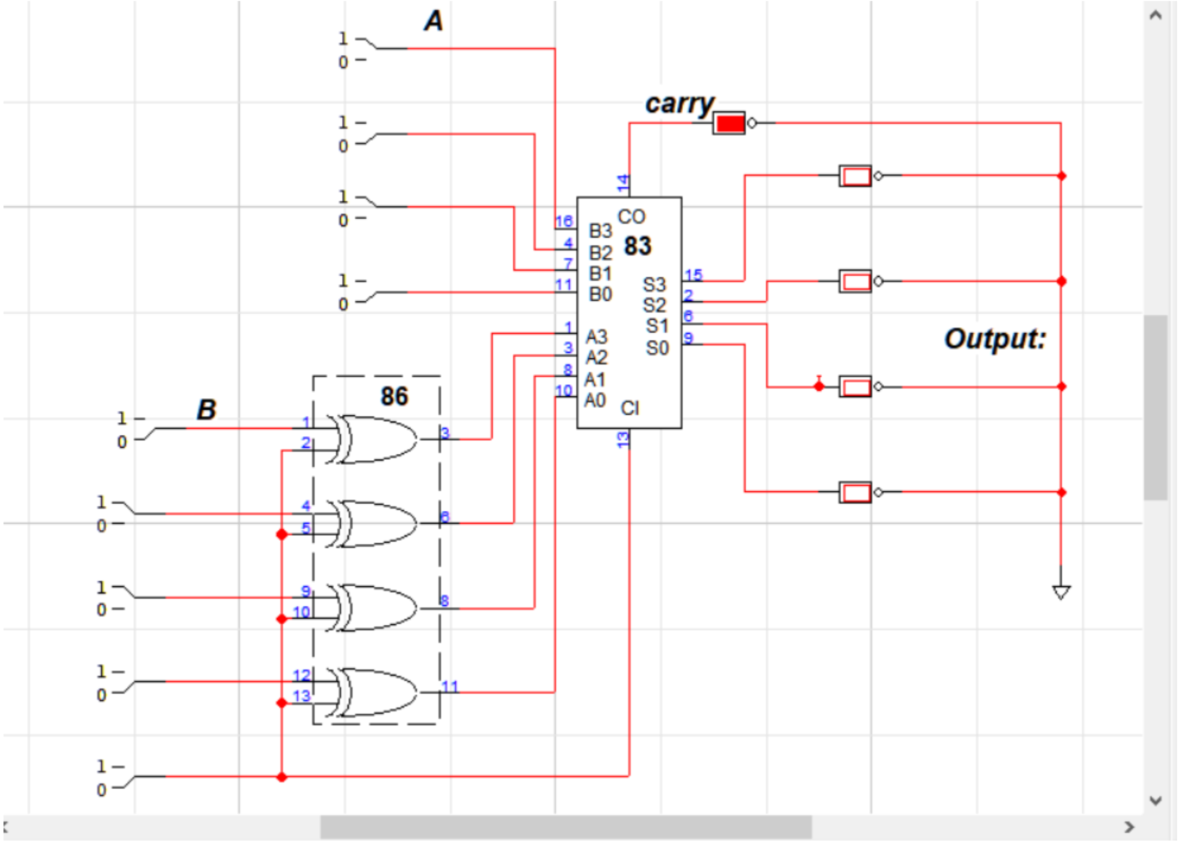


Bimal parajuli (20BDS0405)

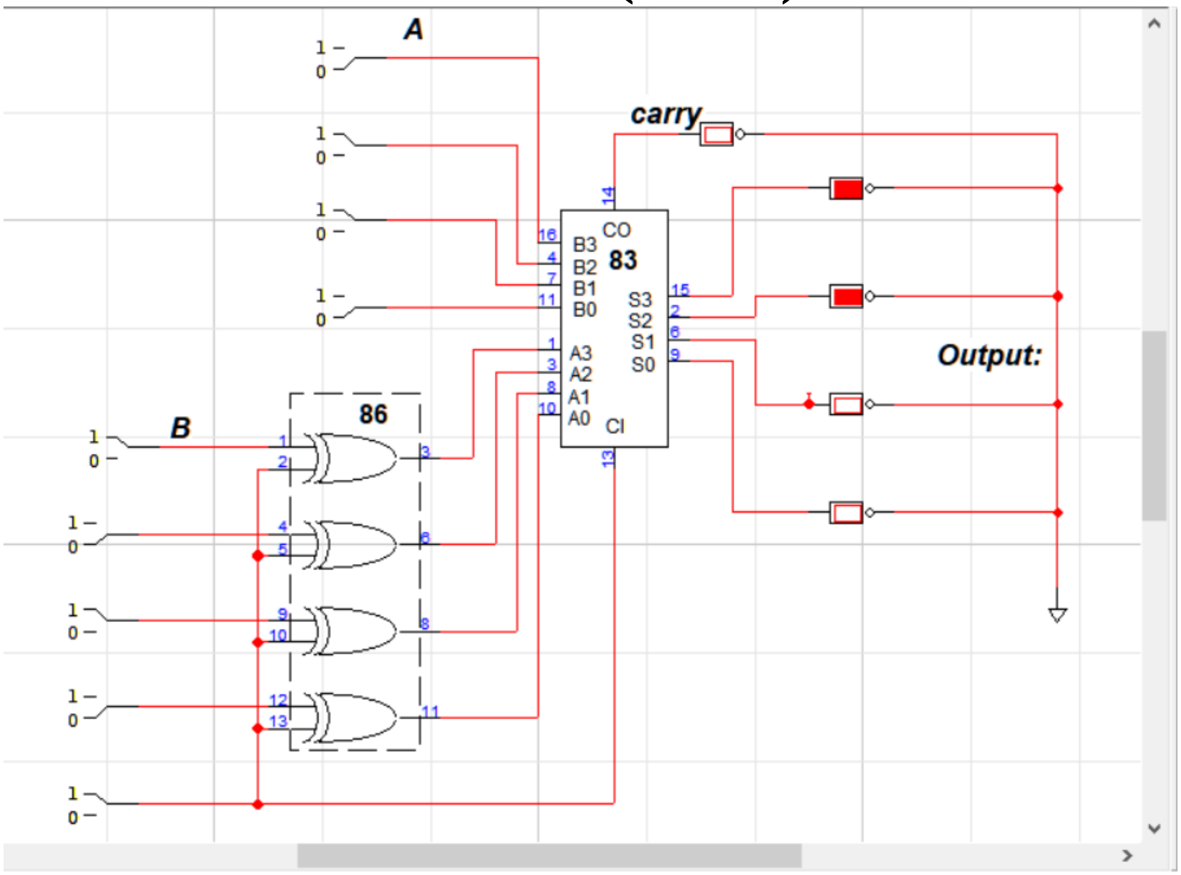
9 - 9 = 0 (10000):



10 + 6 = 16(10000):



6-10 = -4 (01100)

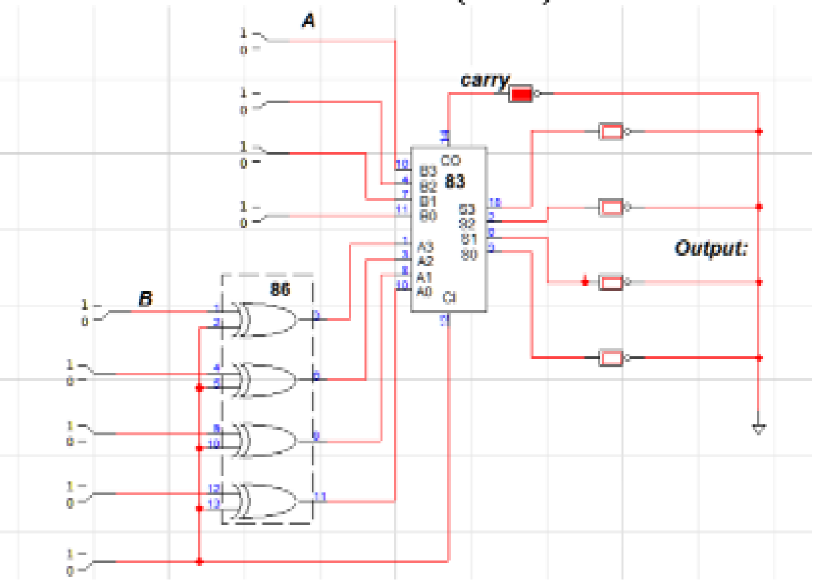


- ***Show that Co =1 when sum exceeds 15.***

In the demonstration of 6 + 10 = 16, we can see that the output carry out C_0 =1.

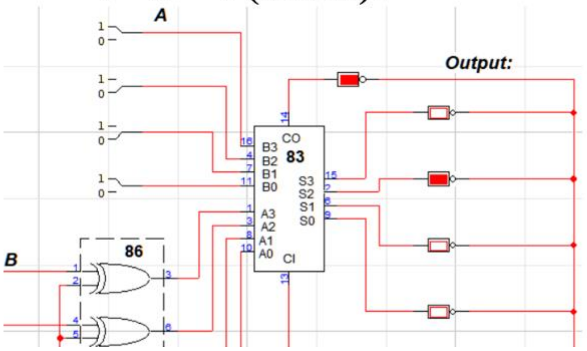
Here, since, the output of the sum is greater than 15, it needs 5 digits to be represented in equivalent binary notation and hence the carry out needs to be set high as overflow.

10 + 6 = 16(1000):



- ***Comment on sum and Co for the subtraction operations when A > B and A < B.***

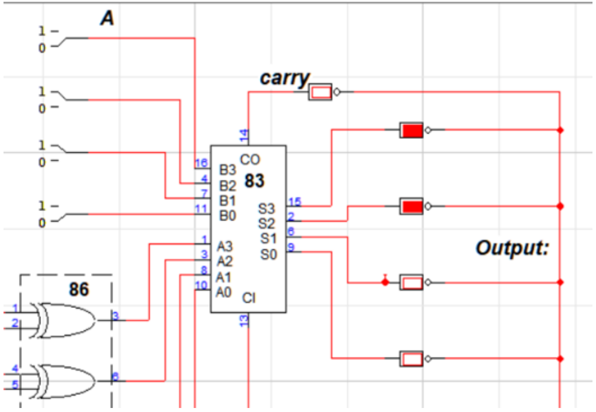
9 - 5 = 4 (10100):



When A> B, the Carry out is set high and the sum is exactly the expected output and can be accepted as the difference of two supplied numbers.

Carry => 1
Sum => can be accepted as it is

6-10 = -4 (01100)



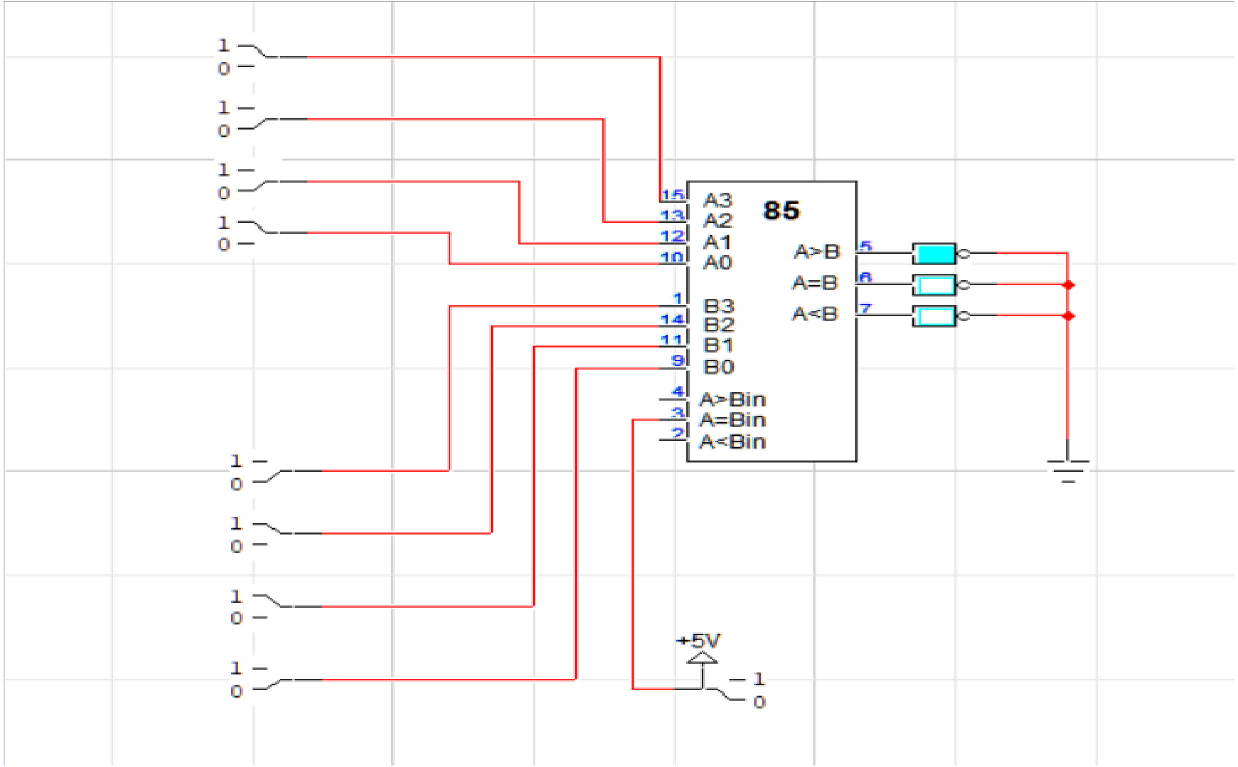
However,
When A < B, the Carry out is set low and the sum is the twos complement of the exact expected difference. For example in 6-10, the carry out is 0, so the difference must be negative, the sum is 1100. By taking it’s 2’s complement, we get the required answer i.e. 100 or 4
Carry => 1
Sum => must be calculated 2’s complement.

e) ***Use IC7485 to compare the following two 4 bit numbers A and B. Record the outputs in table 3. Note that in LogicWorks you need to connect (A = B) input to logic 1 (as an indication that previous stages are equal in multi-digit numbers) for correct results while this is not necessary for the hardware.***

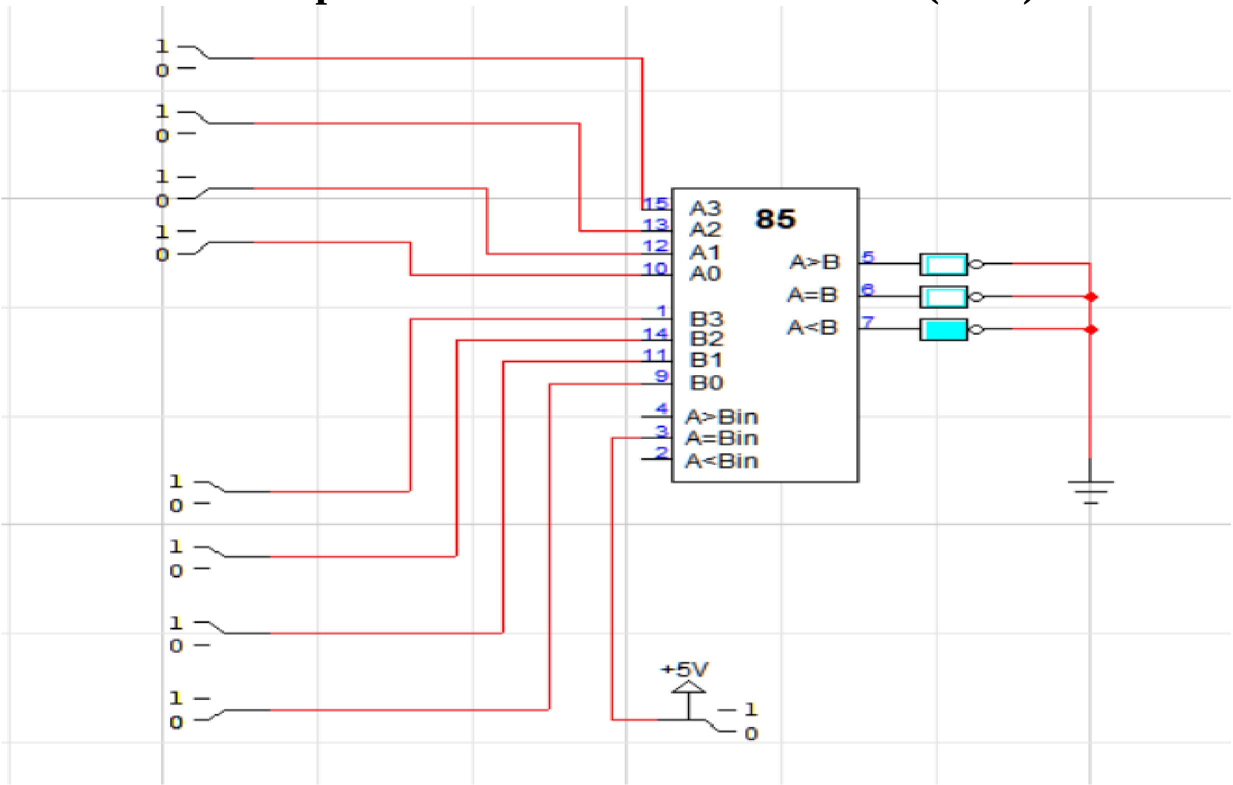
Table 3.

A	B	Outputs
1001	0110	A>B
1100	1110	A<B
0011	0101	A<B
0101	0101	A=B

Comparison of A=1001 and B= 0110: (A>B)

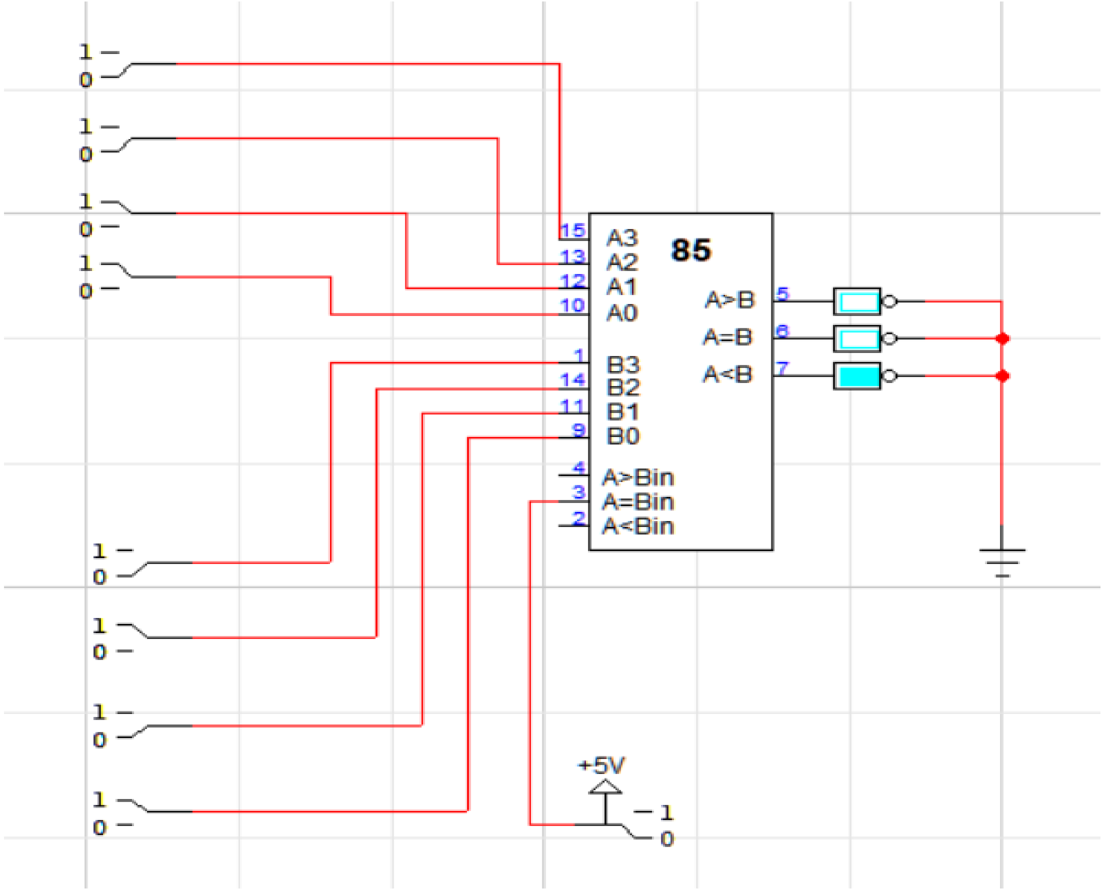


Comparison of A= 1100 and B= 1110: (A<B)

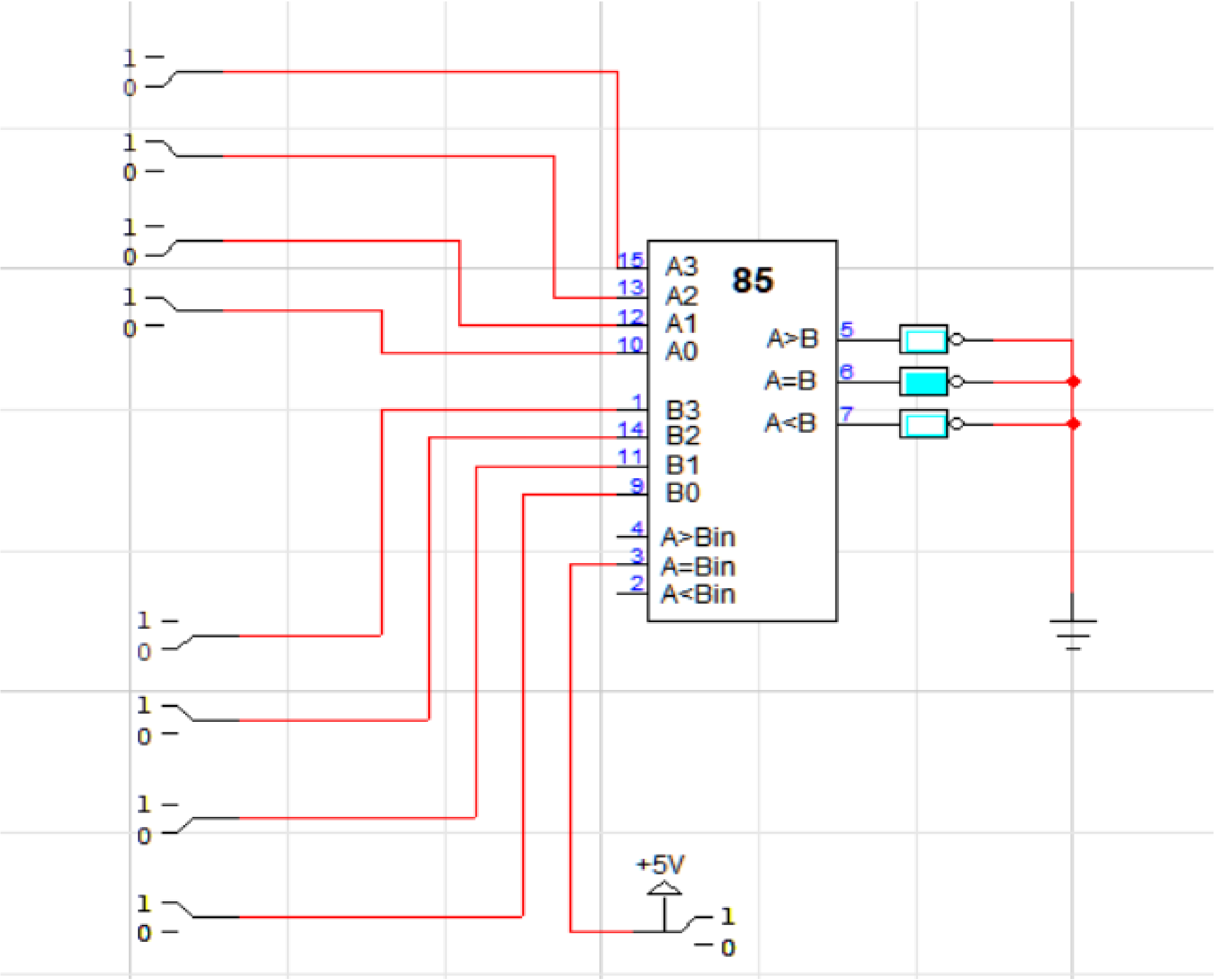


Comparison of A= 0011 and B = 0101: (A<B)

Bimal parajuli (20BDS0405)



Comparison of A=0101 and B= 0101: (A=B)



f) ***A magnitude comparator can be constructed by using a subtractor as in Fig 2. And an additional combinational circuit. This is done with a combinational circuit which has 5 inputs S1, S2, S3, S4, and Co, and three outputs X, Y, Z see Fig.4.***

X = 1 if A = B Where Co = and S = 0000
Y = 1 if A < B Where Co = 0 and S ≠ 0000
Z = 1 if A > B Where Co = 1 and S ≠ 0000

Design and construct this logic circuit with minimum number of gates. Check the comparator action using Part (e). In the Lab verify your LogicWorks simulation.

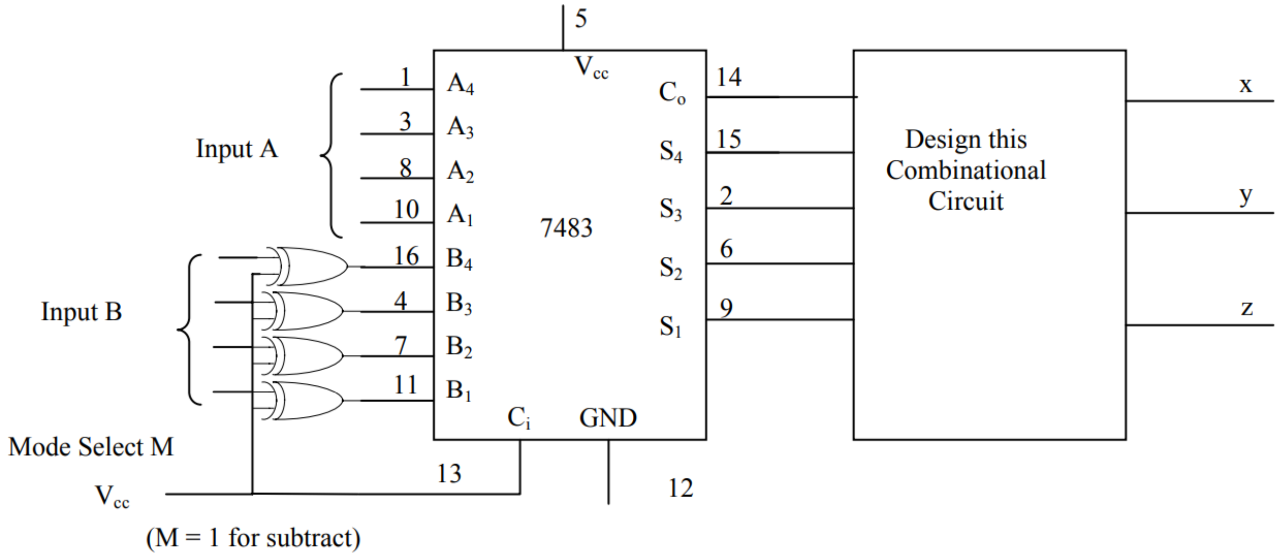


Fig.4 A magnitude comparator using a subtractor

Based on the above conditions,

X will be high if the numbers are equal i.e. S=0000 and Co=1.

So, **X** = **C**o. $(\overline{\boldsymbol{So} + \boldsymbol{S1} + \boldsymbol{S2} + \boldsymbol{S3}})$

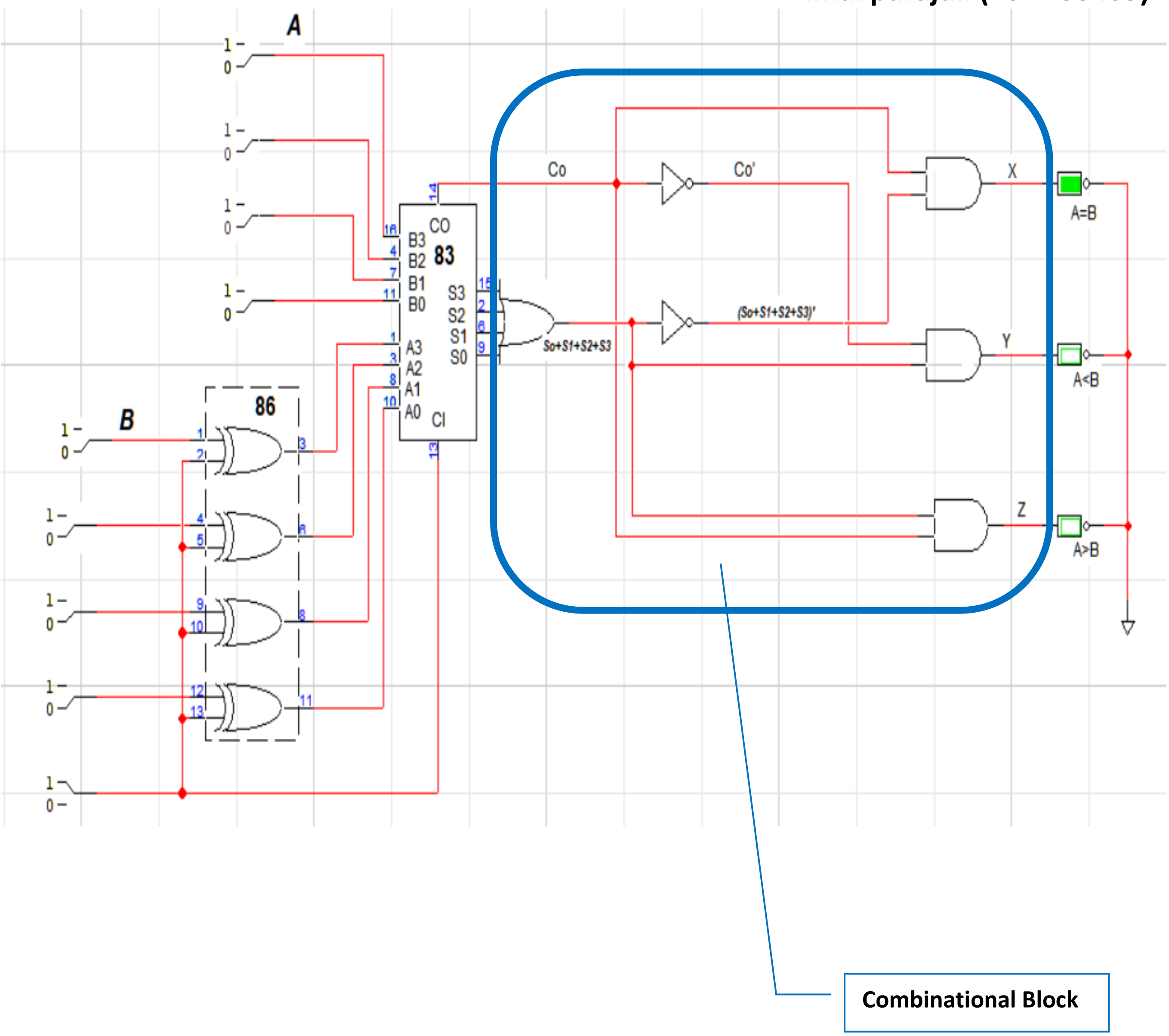
Y will be high if A < B. i.e. S ≠ 0000 and Co = 0

So, **Y** = $\overline{\mathbf{Co}}$. $(\boldsymbol{So} + \boldsymbol{S1} + \boldsymbol{S2} + \boldsymbol{S3})$

Y will be high if A > B. i.e. S ≠ 0000 and Co = 1

So, **Z** = **C**o. $(\boldsymbol{So} + \boldsymbol{S1} + \boldsymbol{S2} + \boldsymbol{S3})$

Bimal parajuli (20BDS0405)



Combinational Block