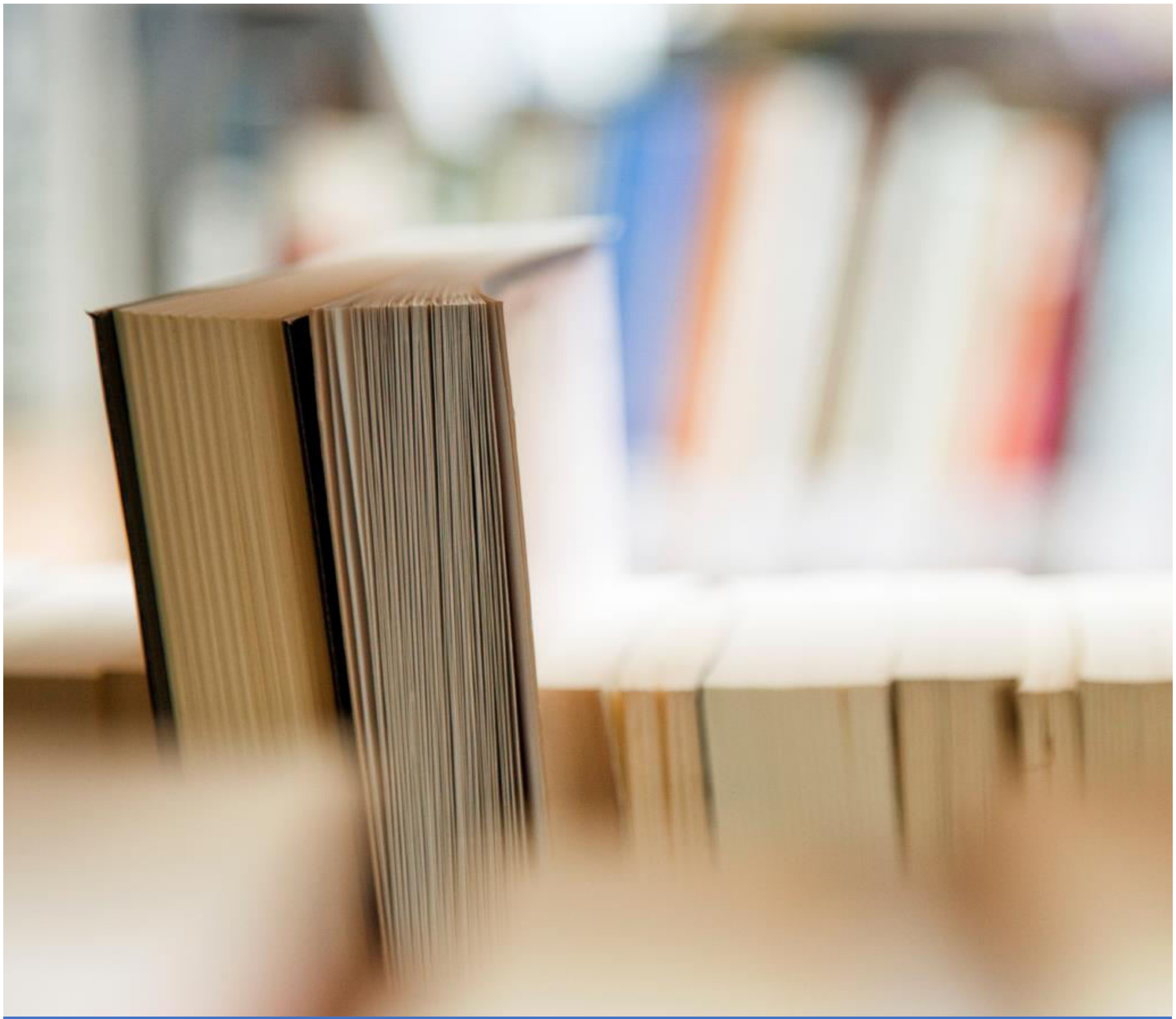


Adders/Subtractors and Multiplexers

Experiment 5 and 6

BIMAL PARAJULI (20BDS0405) 7/10/21 CSE – 1003 (LAB)

Bimal parajuli (20BDS0405)

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.

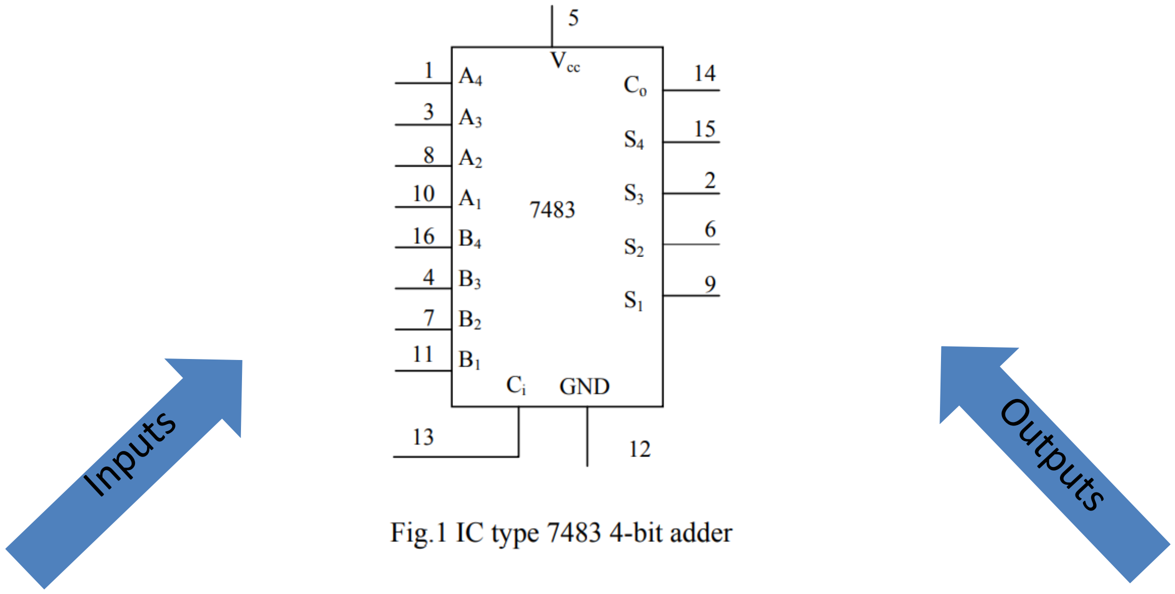
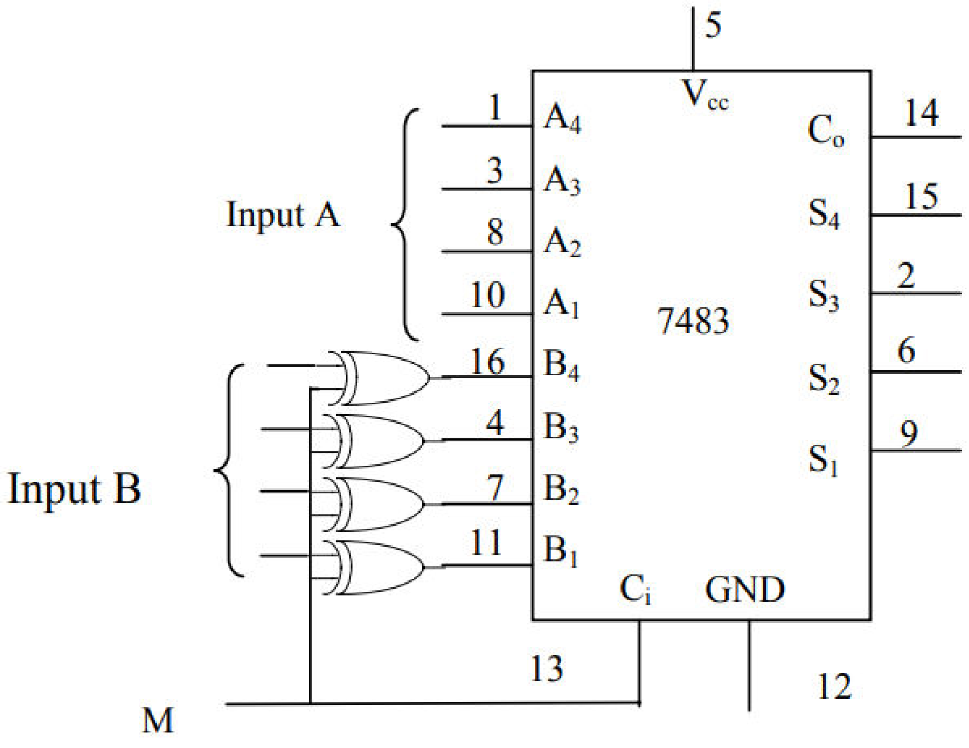


Fig.1 IC type 7483 4-bit adder

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.

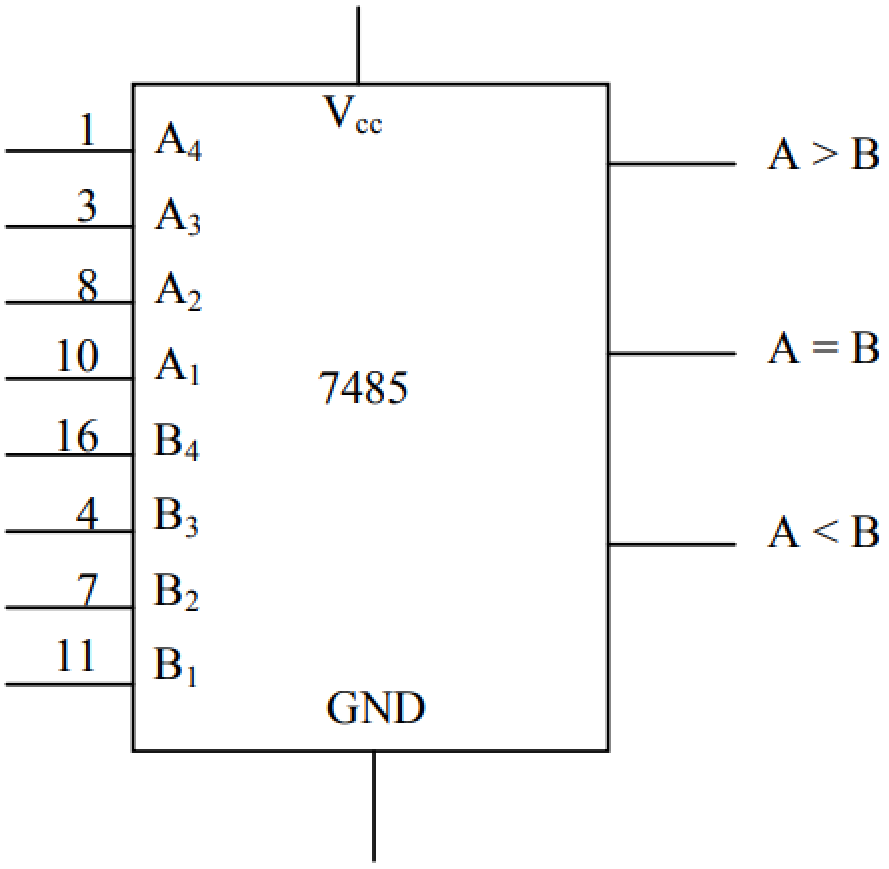


Fig. 3 4-bit magnitude comparator

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:

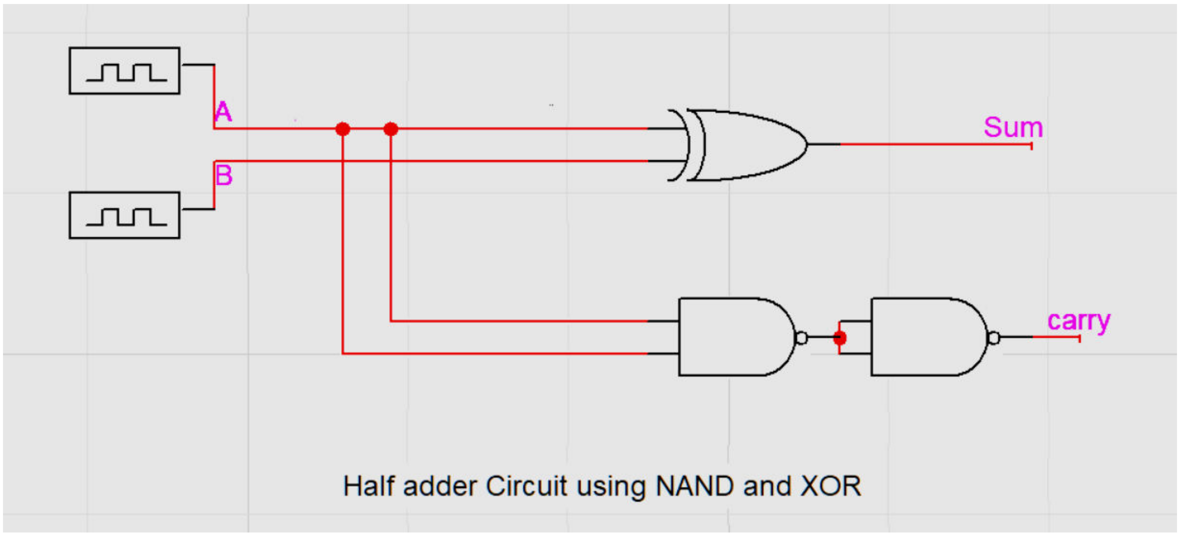
Input(A)	Input(B)	Carry (C)	Sum(S)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Here, C= A.B

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

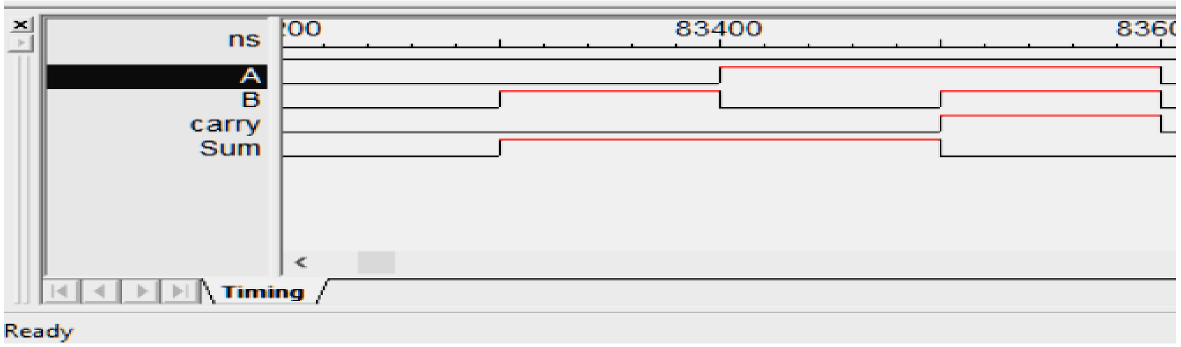
And, S=A XOR B

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

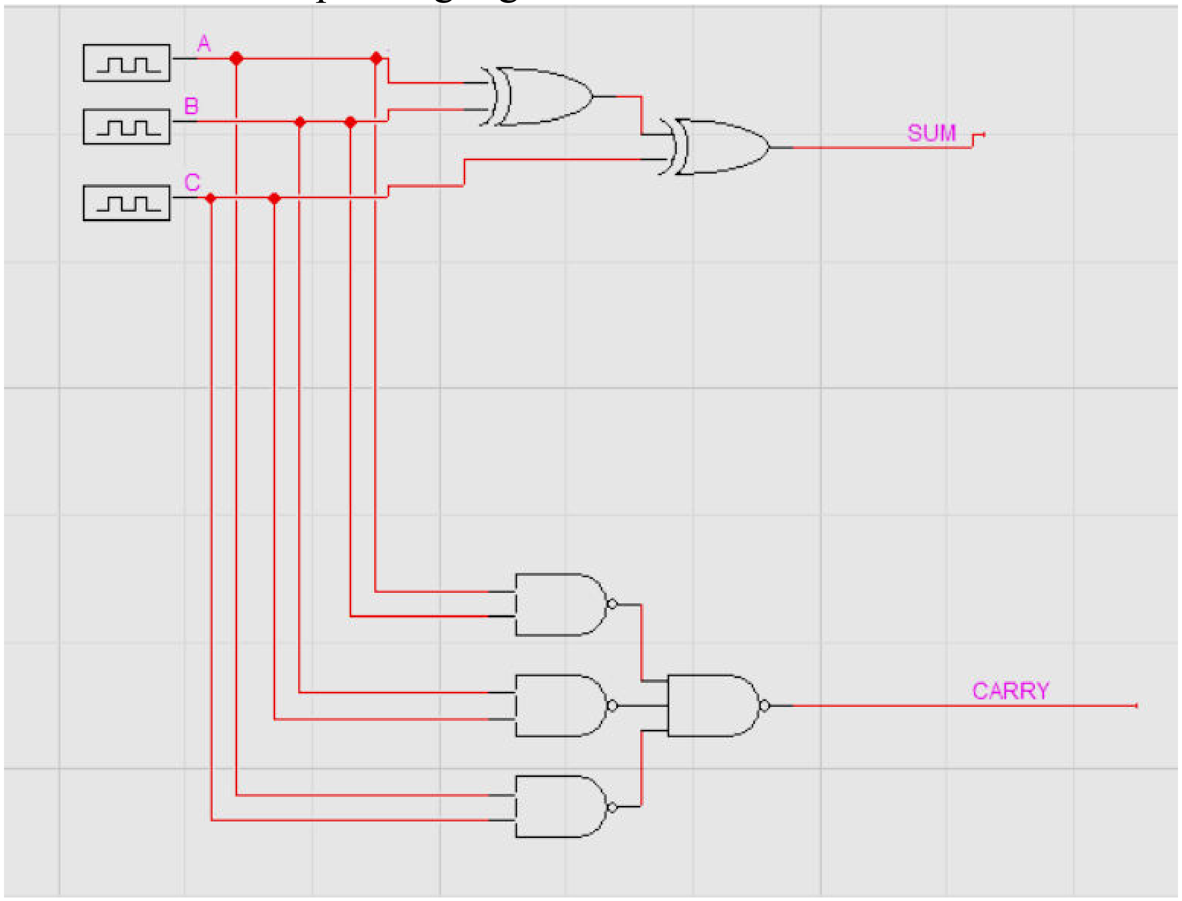
\BC A\	00	01	11	10
00	0	1	0	1
01	1	0	1	0

From above Kmap:

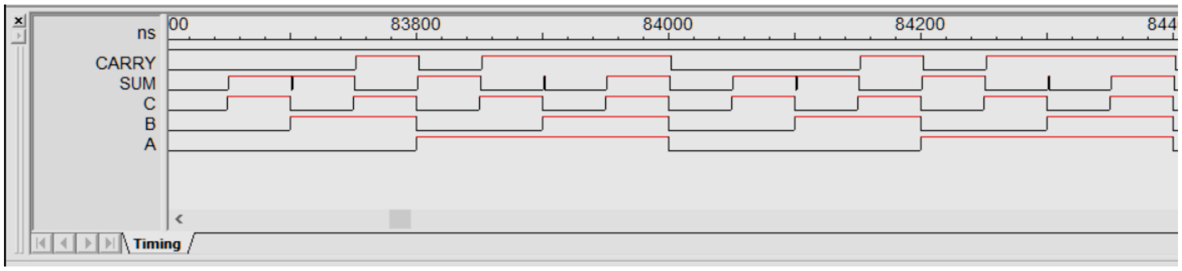
$$C_o = \bar{A}.B.C + A.\bar{B}.C + A.B.\bar{C} + A.B.C$$
$$= A.B + B.C + A.C \quad \text{[Solved from Karnaugh Map]}$$
$$= \overline{(\overline{A.B}\,.\overline{B.C}.\overline{C.A})} \quad \text{[De Morgan's Law]}$$

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

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.***

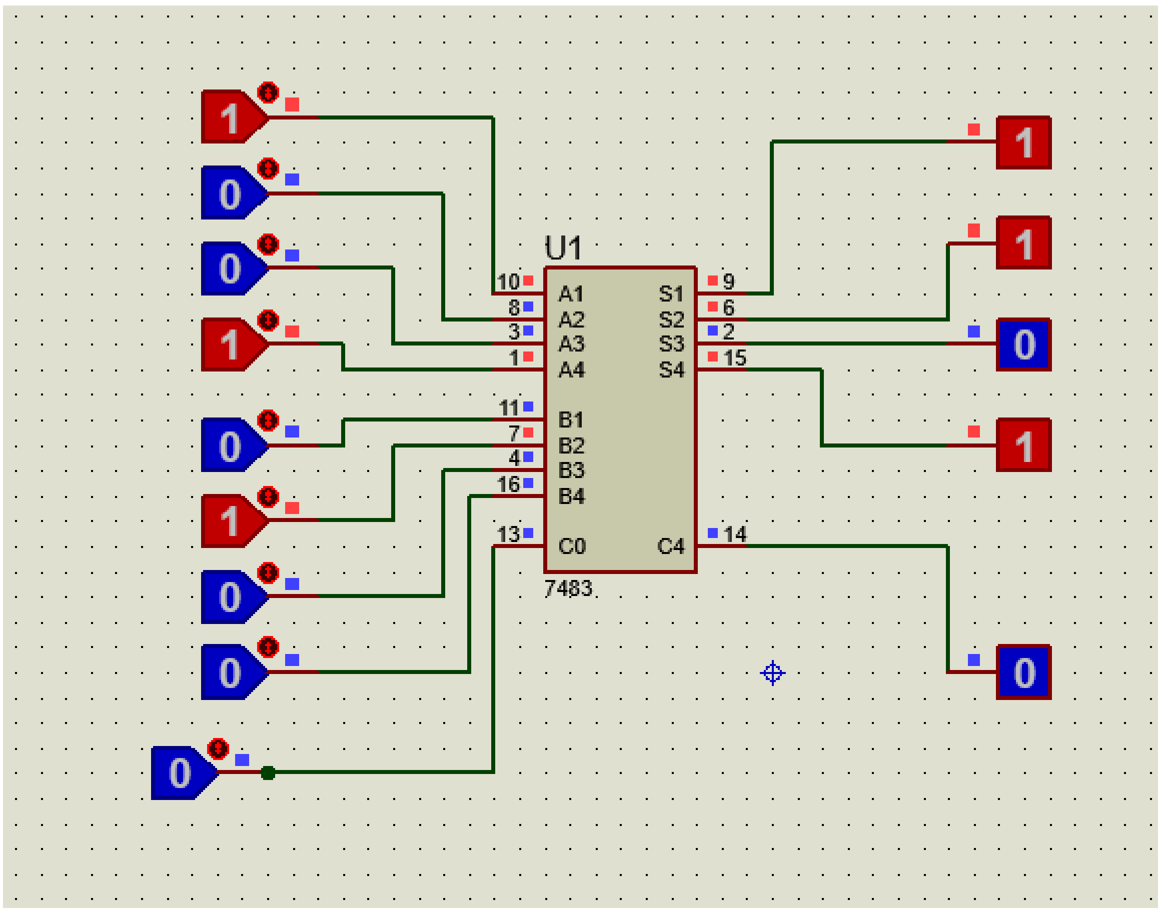
Input carry Ci is taken as logic 0. Show that if the input carry is 1, it adds 1 to the output sum.

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.

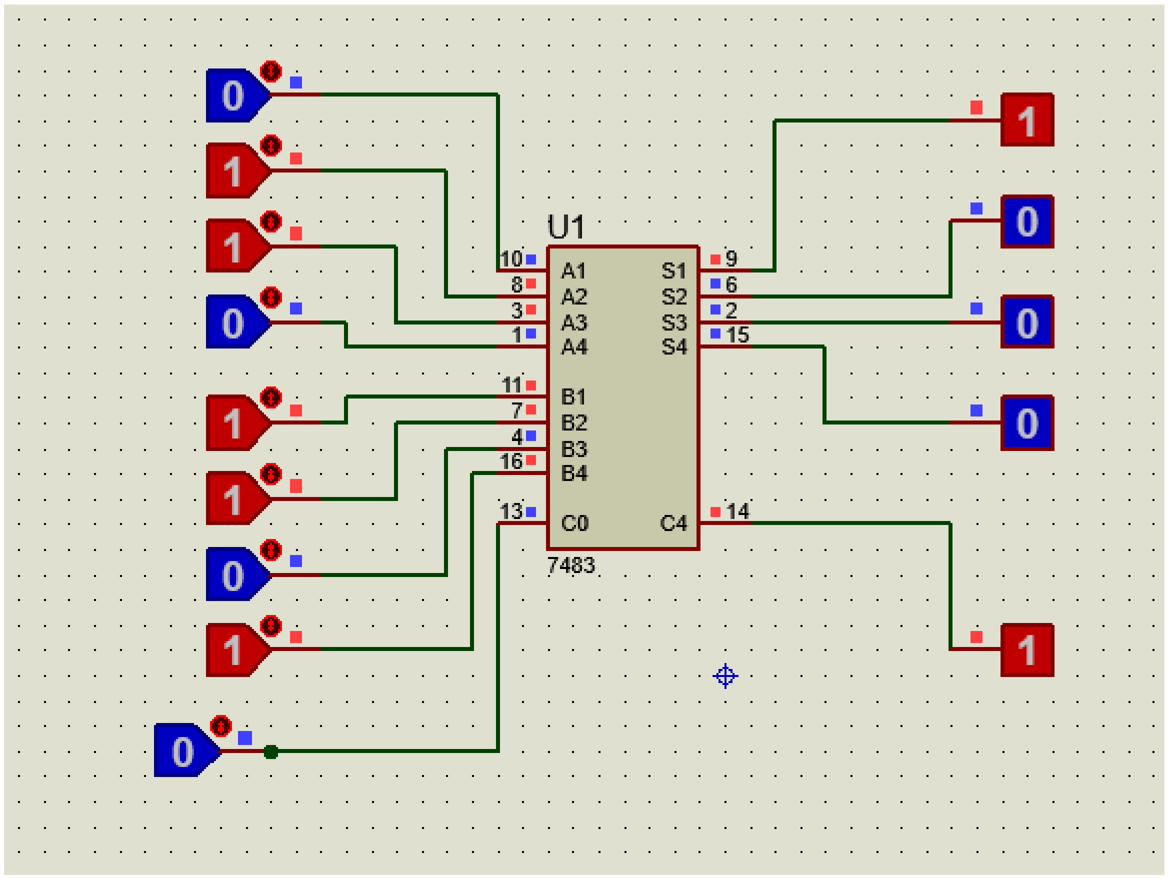
Table 1.

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

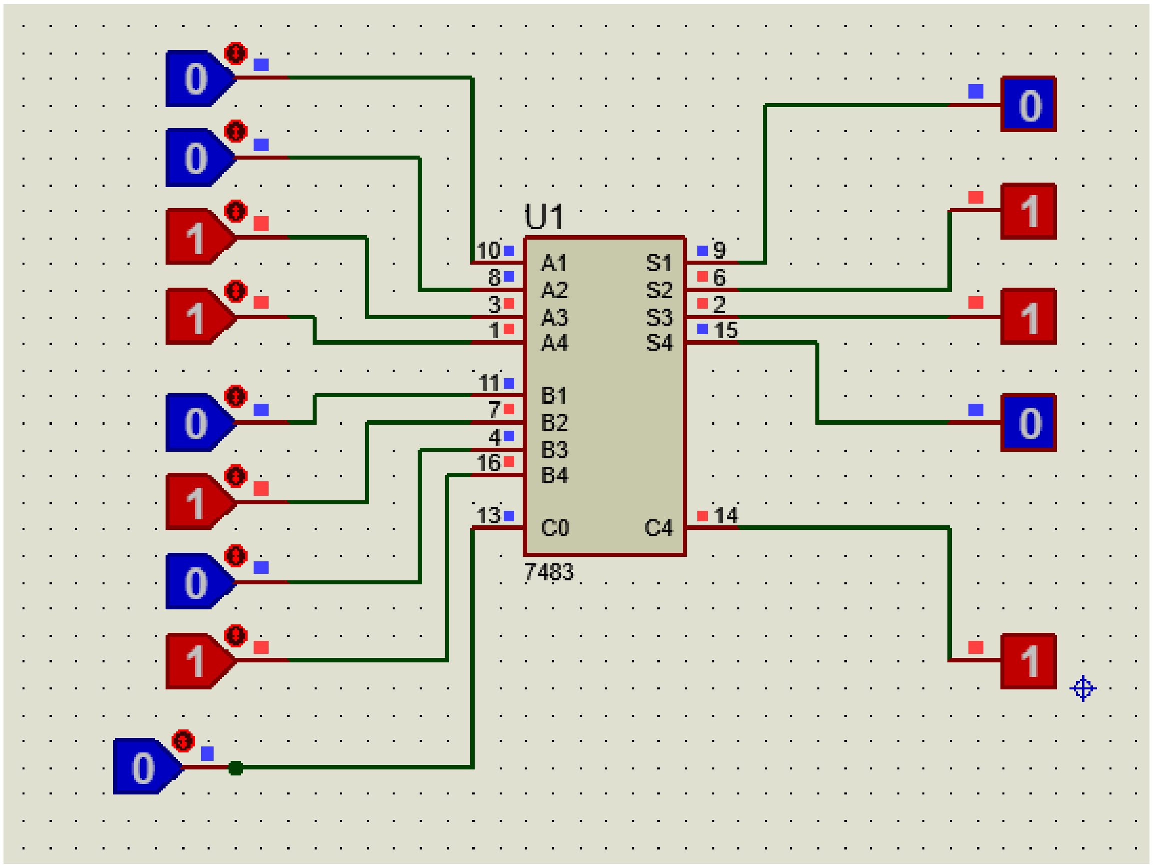
1 0 0 1 + 0 0 1 0



0 1 1 0 + 1 0 1 1



1 1 0 0 + 1 0 1 0

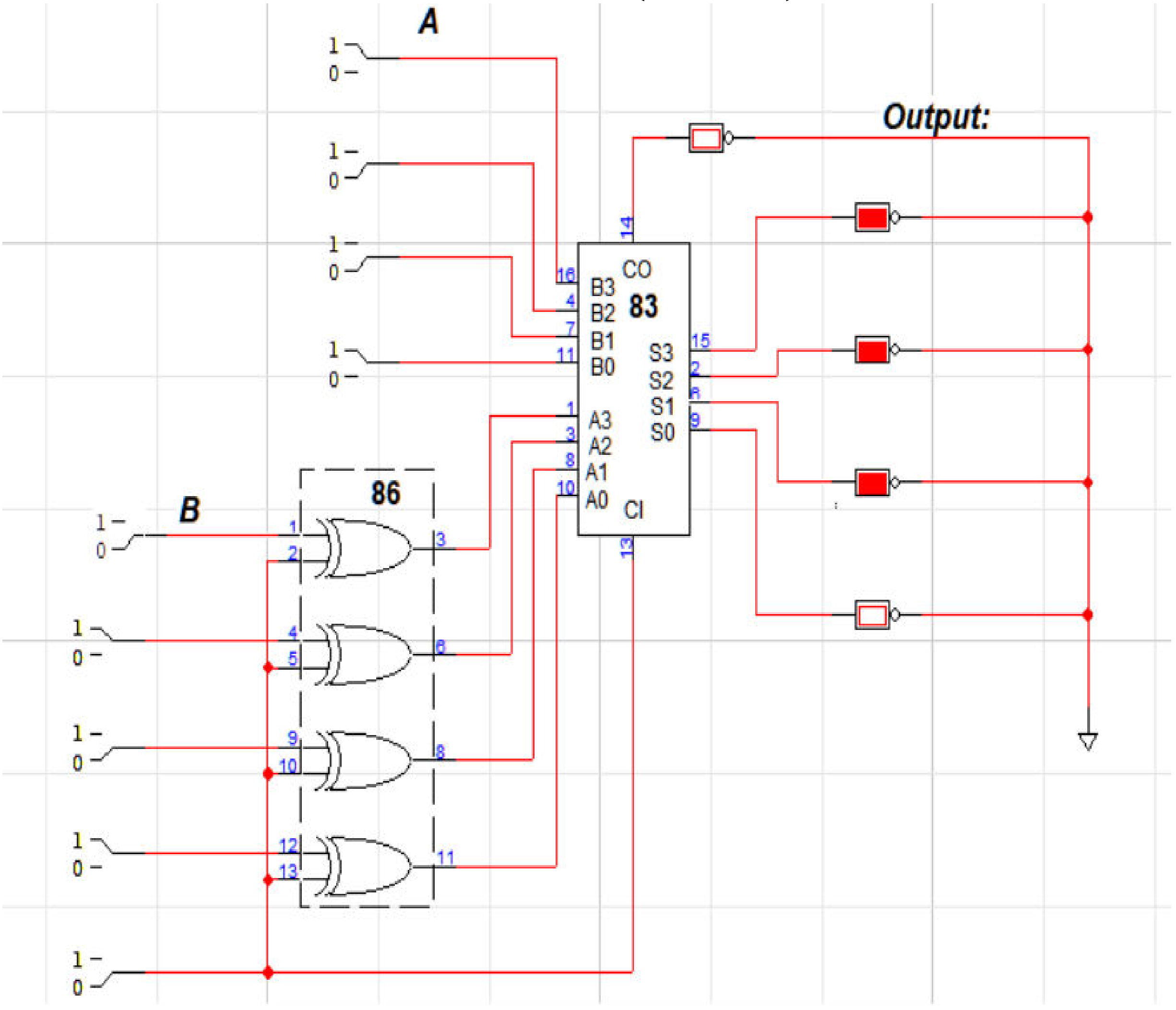


d) ***Connect the adder-subtractor circuit as shown in Fig 2. Perform the following operations and record the values of the output sum and the output carry Co.***

Table 2.

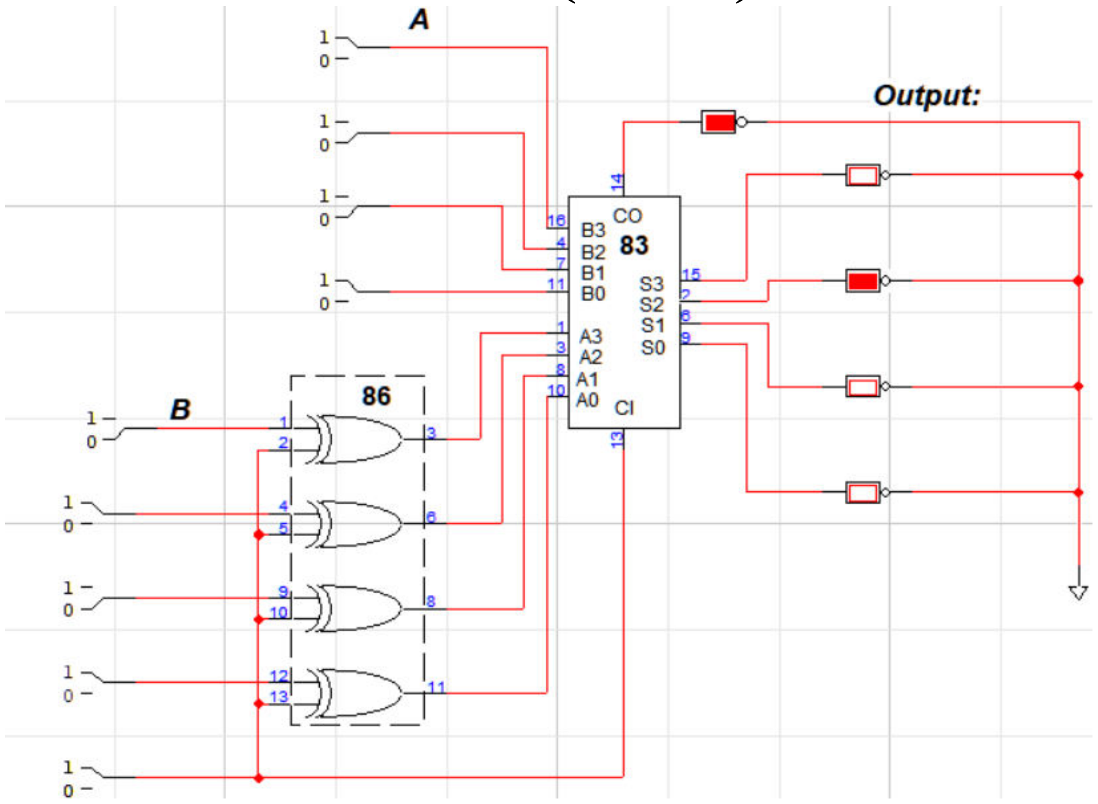
Decimal A B	**Carry Out**	**Output sum**			
9 + 5	0	1	1	1	0
9 - 5	0	0	1	0	0
9 + 13	1	0	1	1	0
9 - 9	0	0	0	0	0
10 + 6	1	0	0	0	0
6 - 10	0	1	1	0	0

9 + 5 = 14(01110):

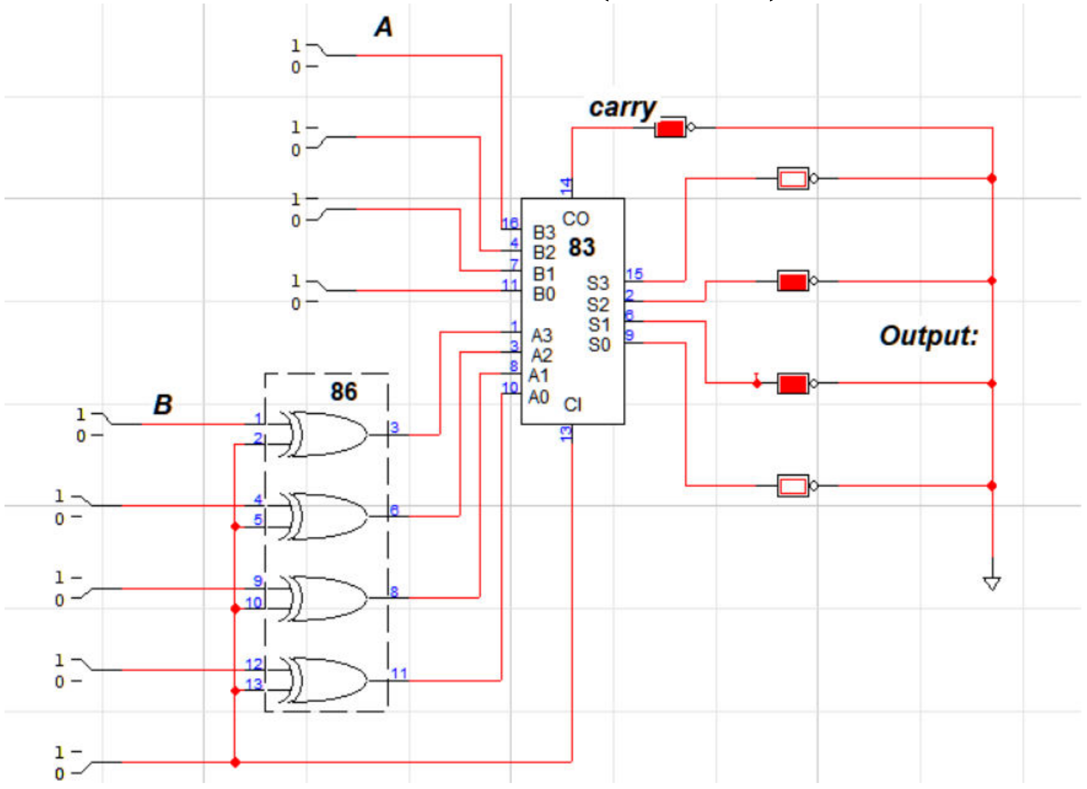


Bimal parajuli (20BDS0405)

9 - 5 = 4 (10100):

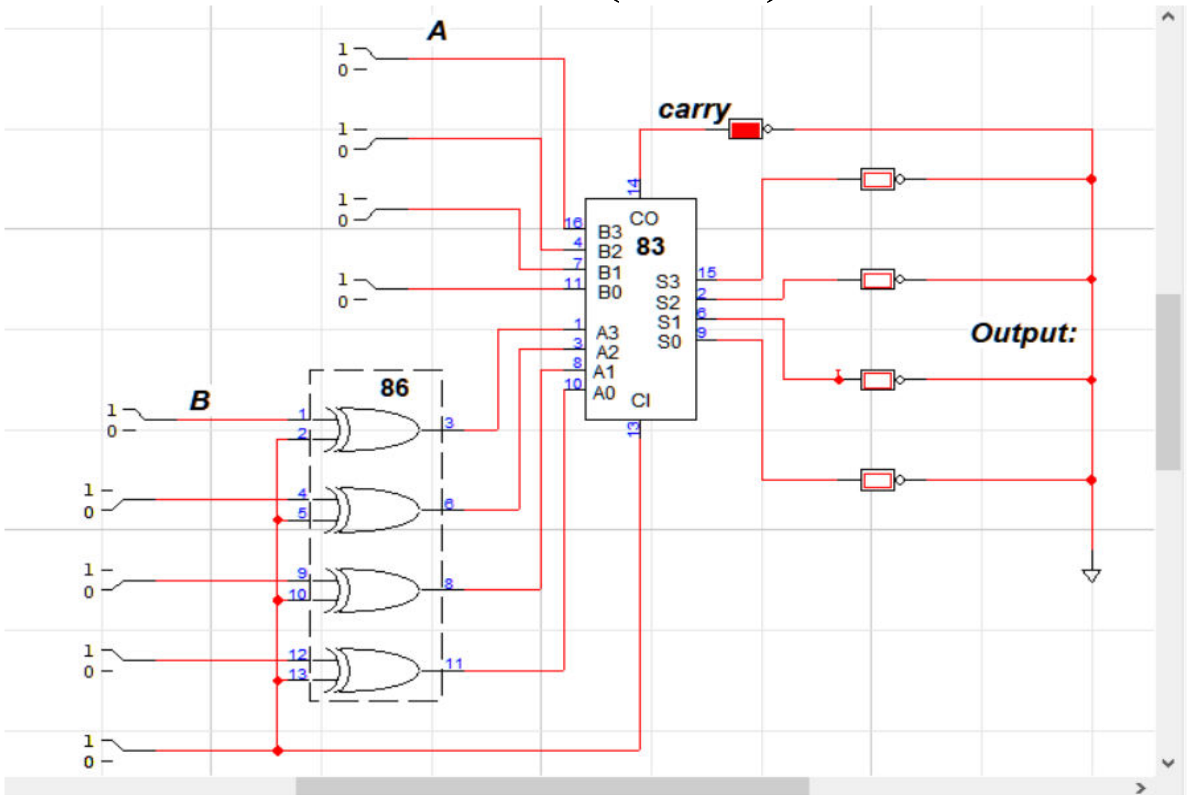


9 + 13 = 22 (10110):

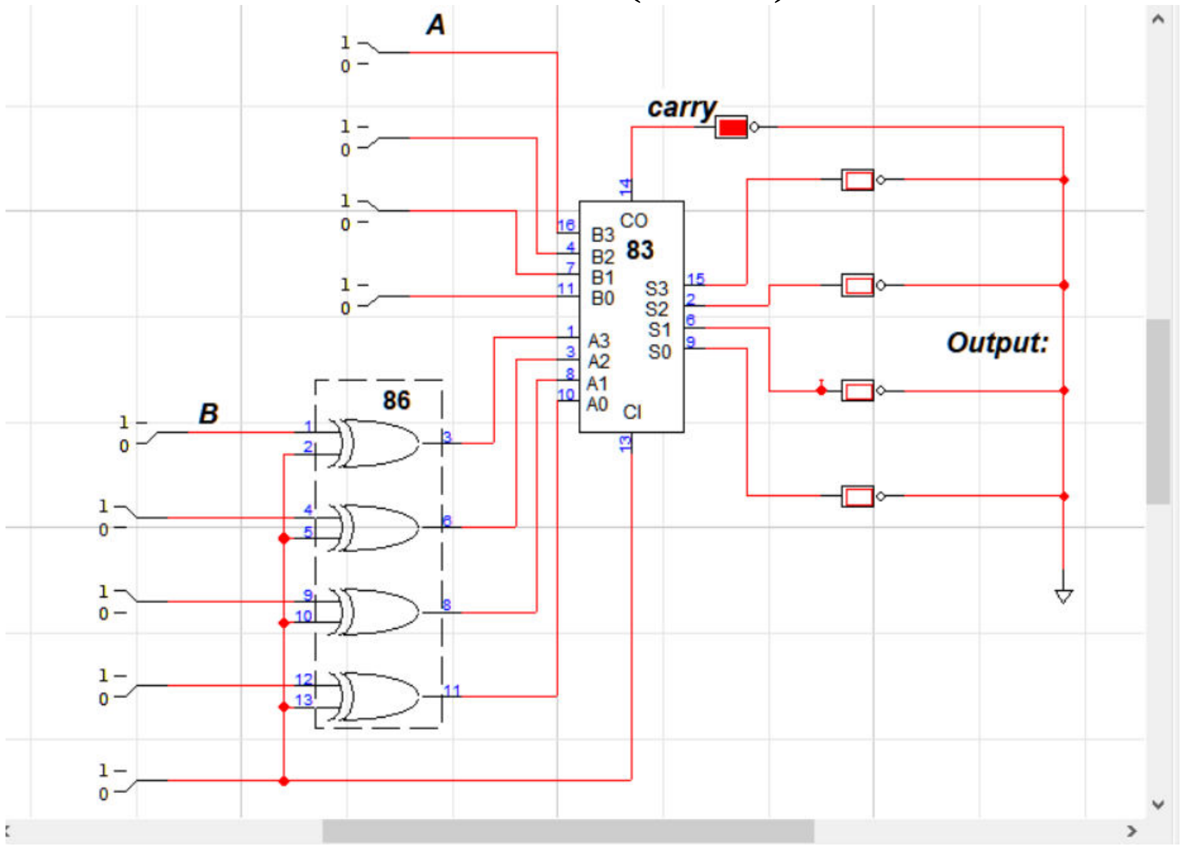


Bimal parajuli (20BDS0405)

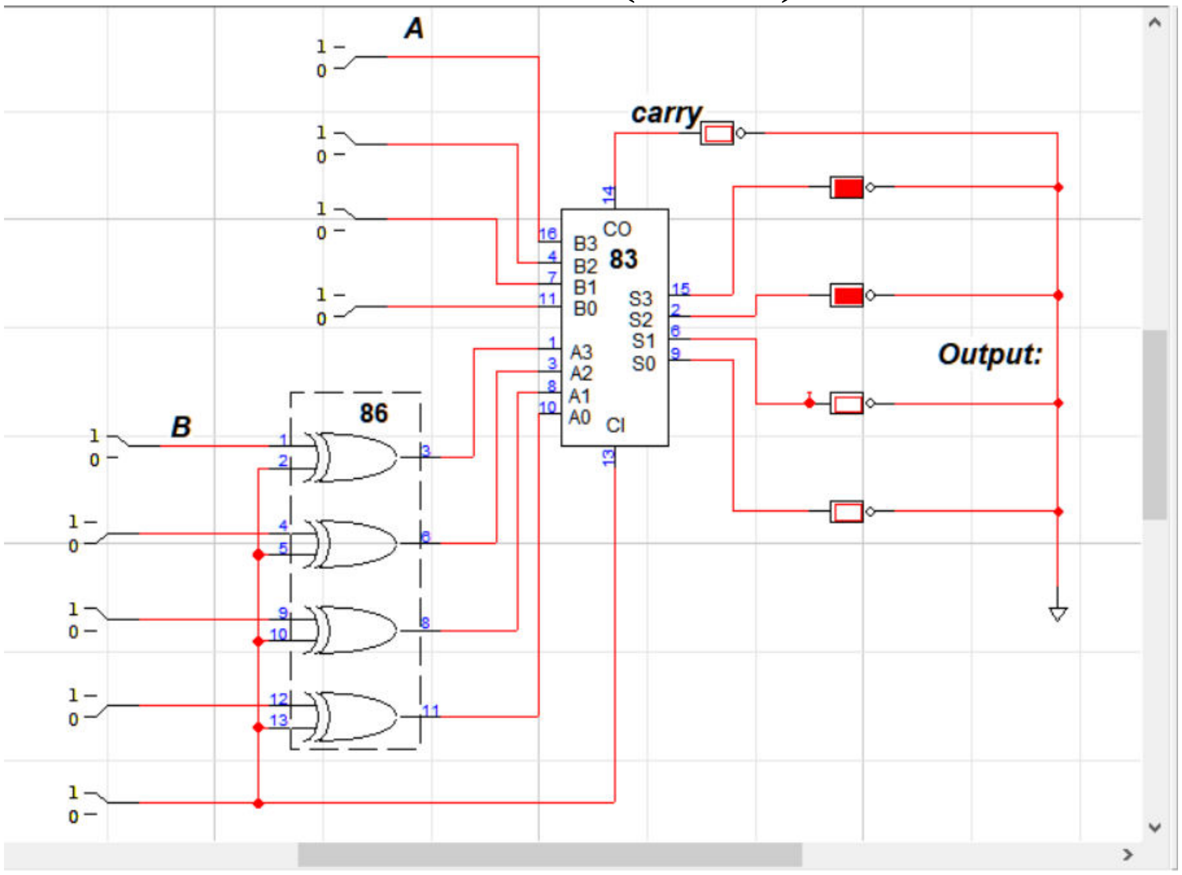
9 - 9 = 0 (10000):



10 + 6 = 16(10000):



6-10 = -4 (01100)

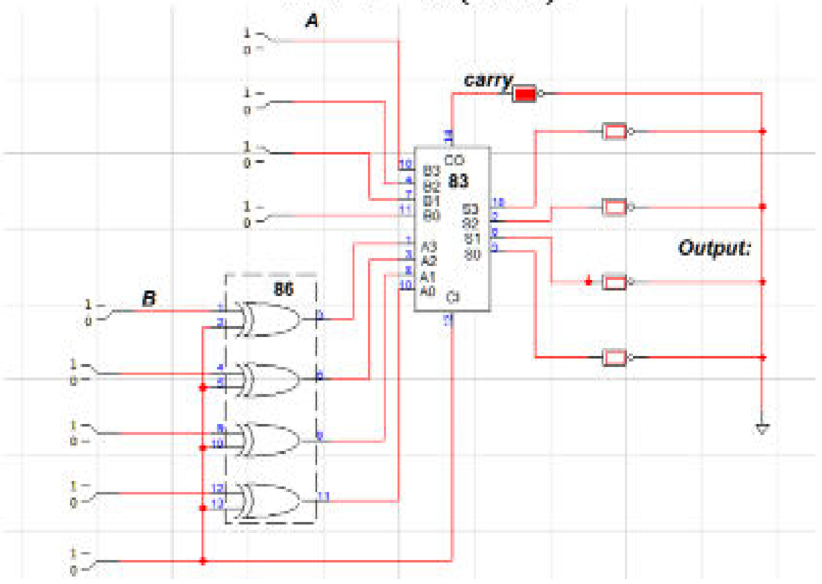


- ***Show that Co =1 when sum exceeds 15.***

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

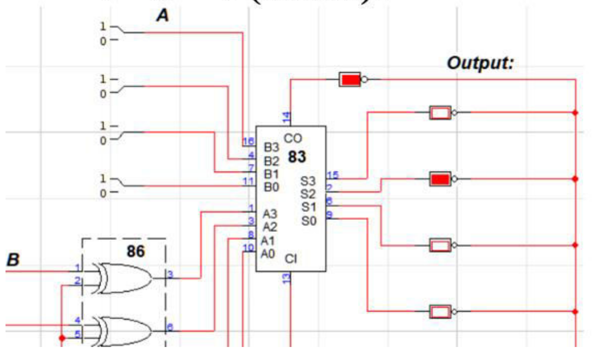
Here, since, the output of the sum is greater than 15, it needs 5 digits to be represented in equivalent binary notation and hence the carry out needs to be set high as overflow.

10 + 6 = 16(1000):



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

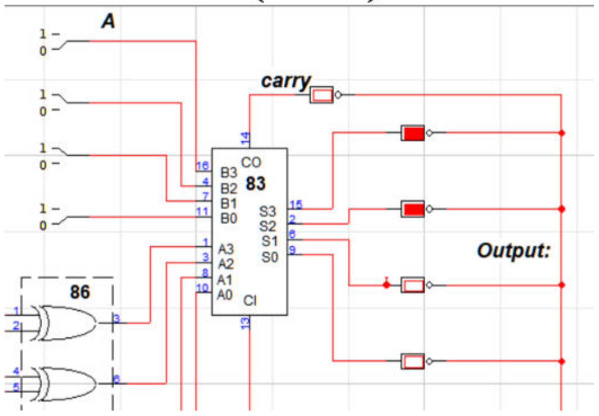
9 - 5 = 4 (10100):



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)



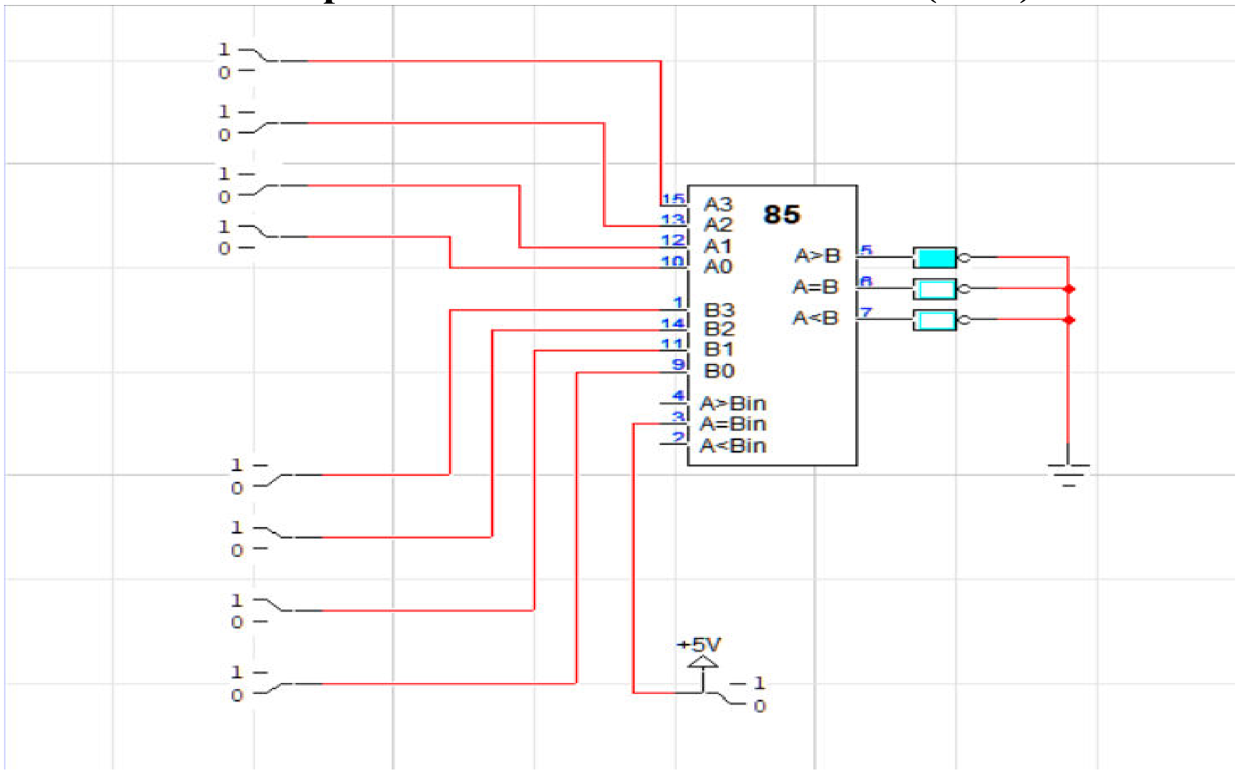
However,
When A < B, the Carry out is set low and the sum is the twos complement of the exact expected difference. For example in 6-10, the carry out is 0, so the difference must be negative, the sum is 1100. By taking it's 2's complement, we get the required answer i.e. 100 or 4
Carry => 1
Sum => must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

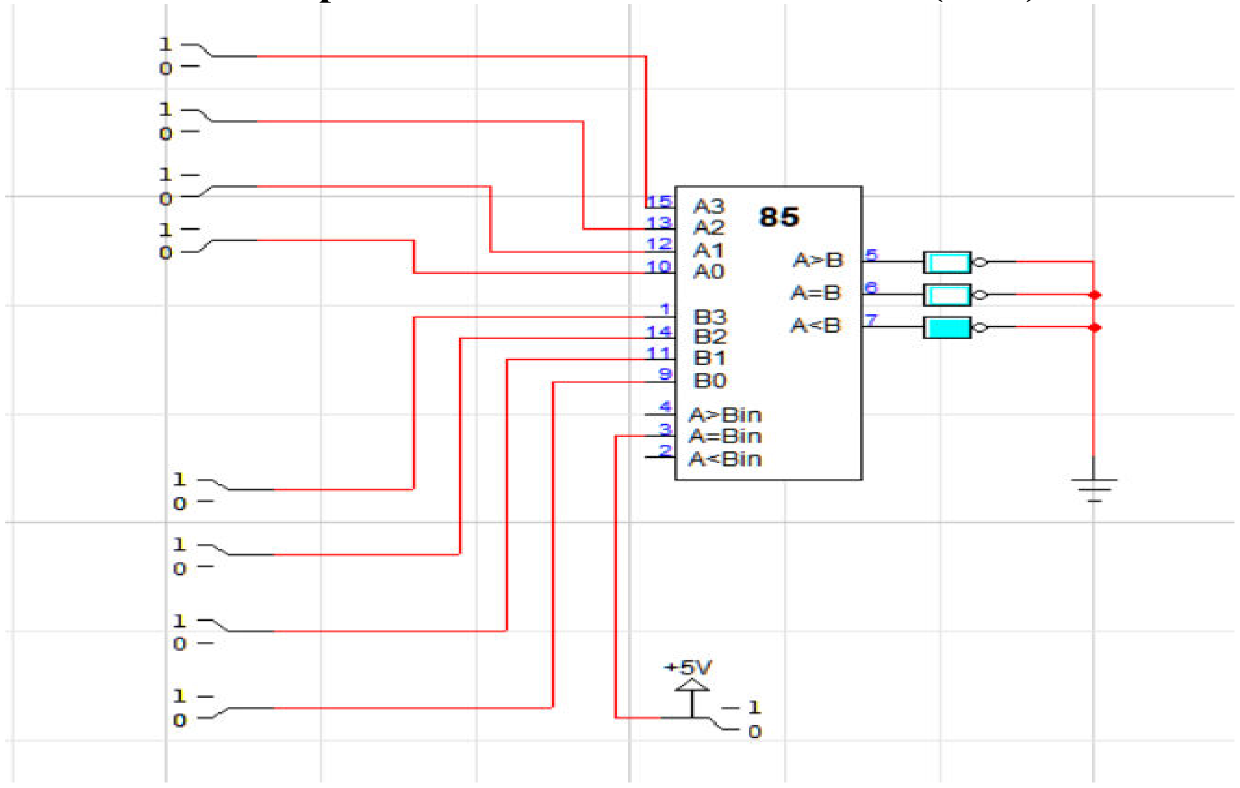
Table 3.

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

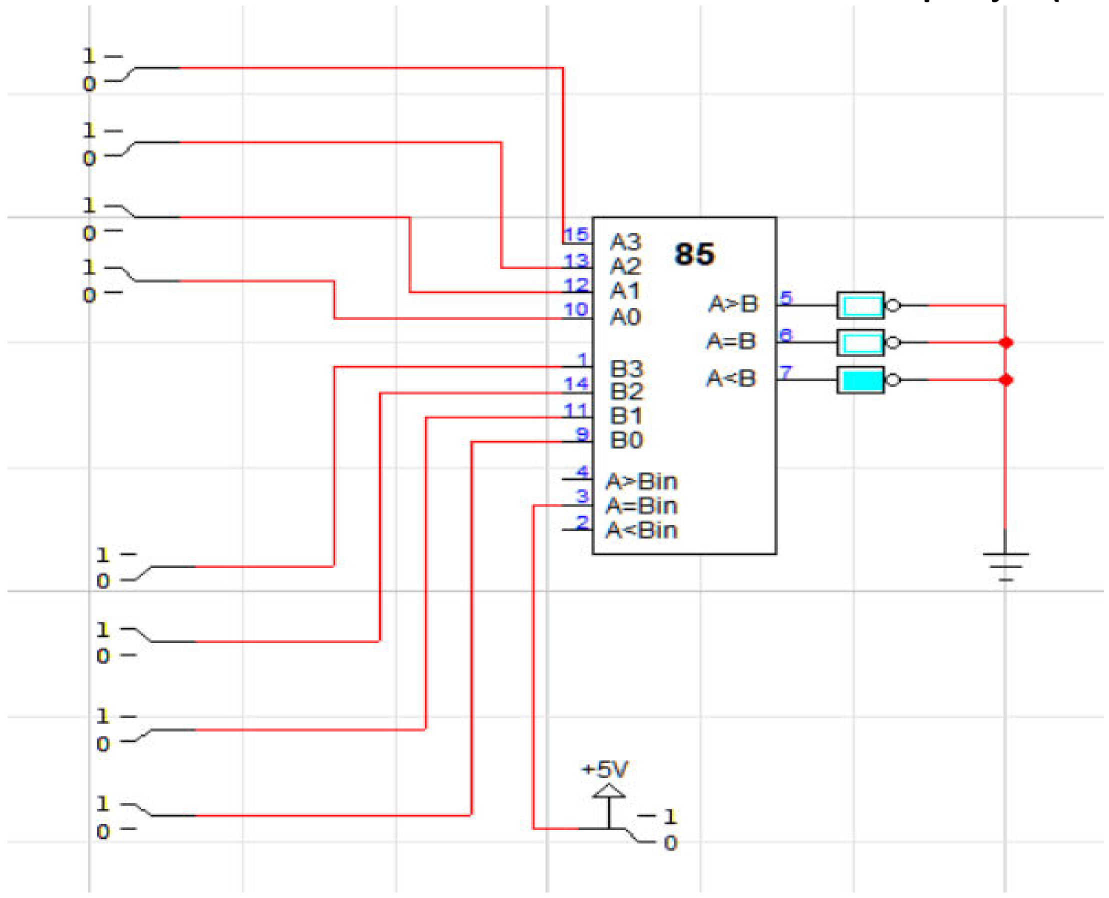
Comparison of A=1001 and B= 0110: (A>B)



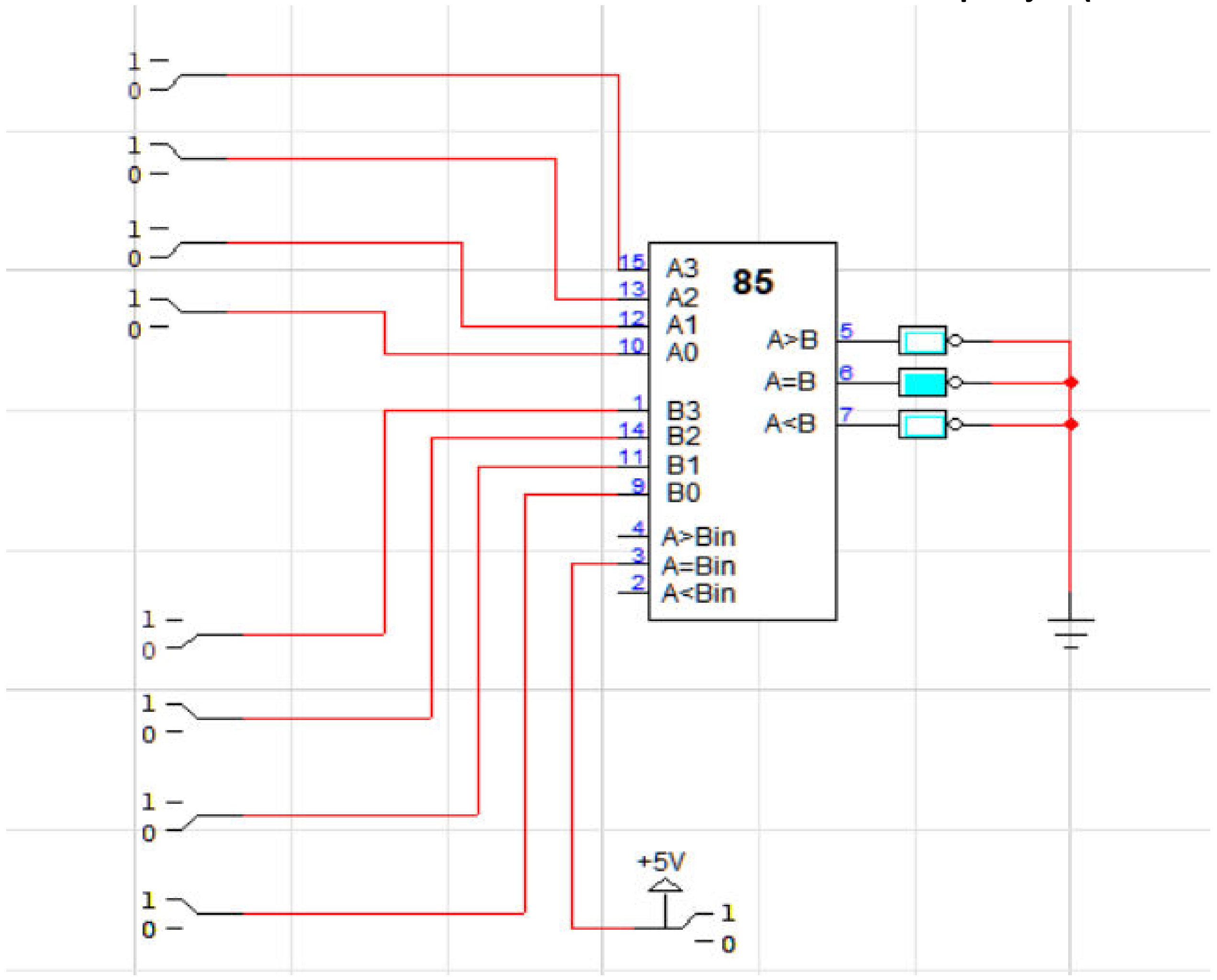
Comparison of A= 1100 and B= 1110: (A<B)



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



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.

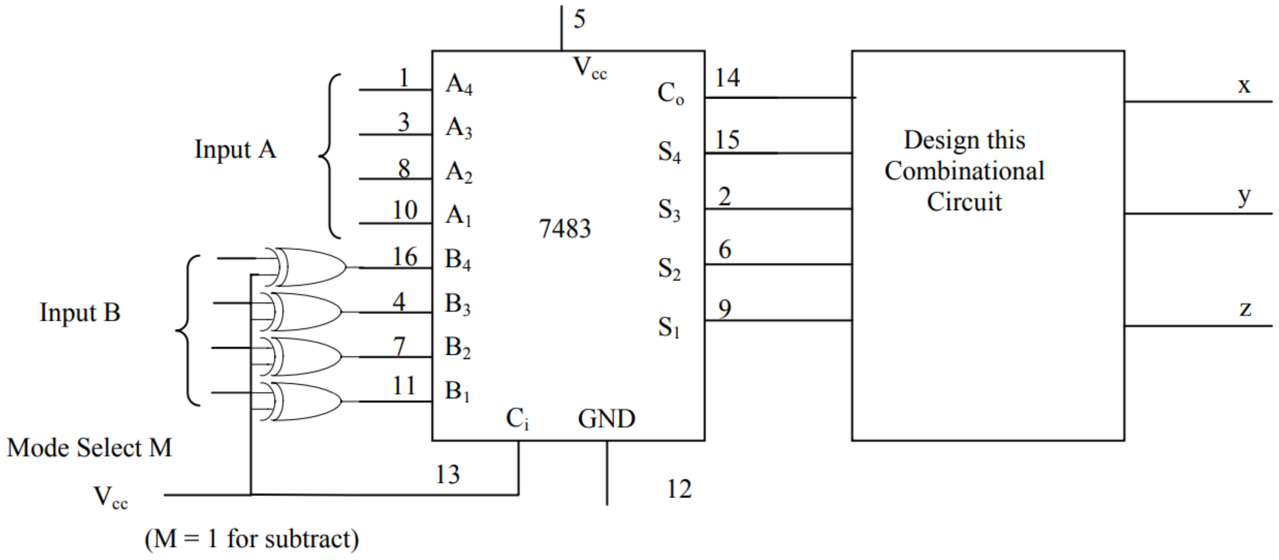


Fig.4 A magnitude comparator using a subtractor

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. $(\overline{\boldsymbol{So}+\boldsymbol{S1}+\boldsymbol{S2}+\boldsymbol{S3}})$

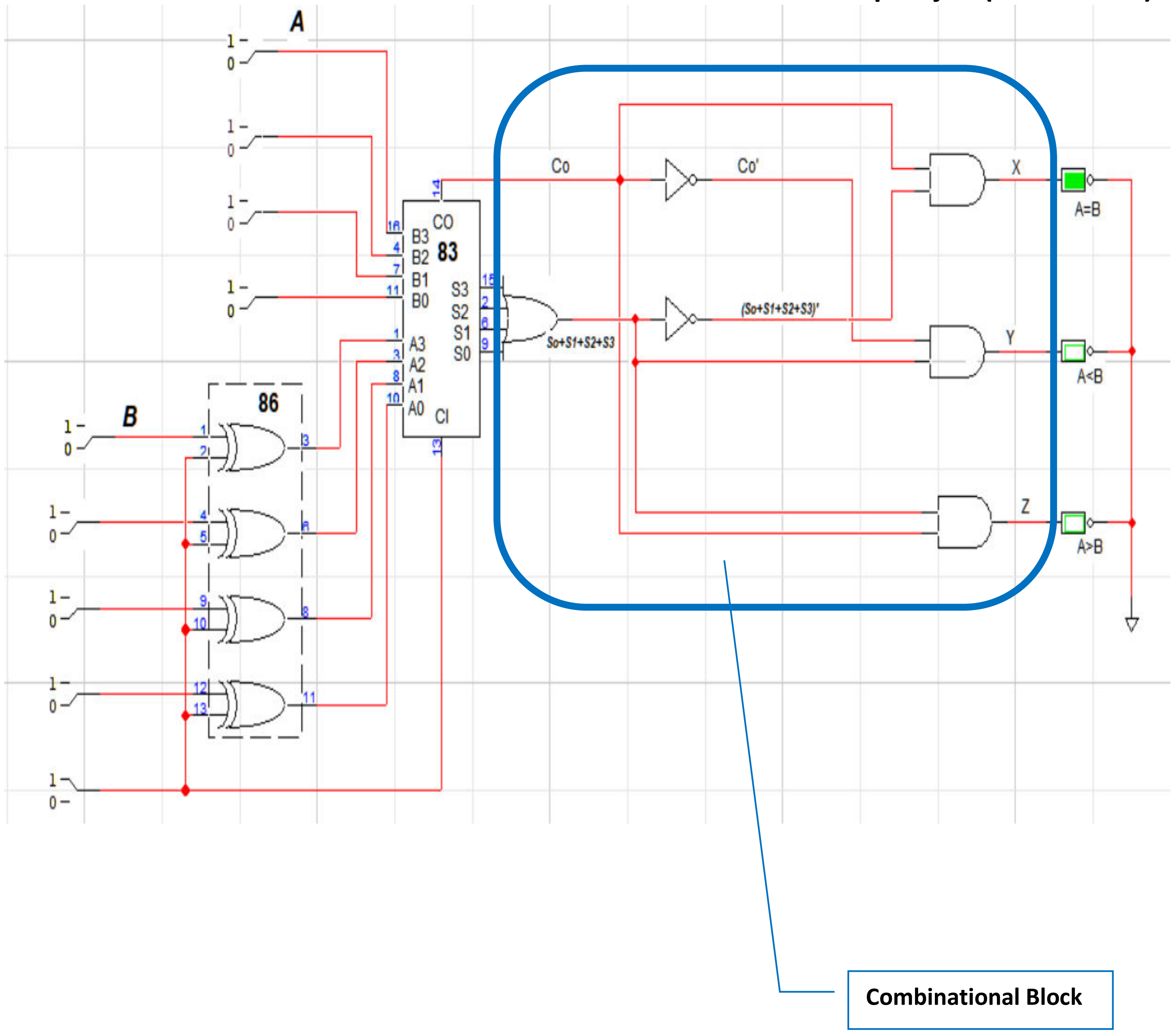
Y will be high if A < B. i.e. S ≠ 0000 and Co = 0

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

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)



Bimal parajuli (20BDS0405)

DIGITAL LOGIC DESIGN EXPERIMENT #6 DESIGN WITH MULTIPLEXERS

Objectives:

To design a combinational circuit and implement it with multiplexers. To use a demultiplexer to implement a multiple output combinational circuit from the same input variables.

Apparatus:

- IC type 7404 HEX inverter
- IC type 7408 quad 2-input AND gate
- IC type 74151 8x1 multiplexer (1)
- IC type 74153 dual 4x1 multiplexer (2)
- IC type 7446 BCD-to-Seven-Segment decoder (1)
- Resistance network (1)
- Seven-Segment Display (1)

Softwares Used:

- LogicWorks 5
- Proteus 8 pro

IC Description:

74151 is an 8 line-to-1 line multiplexer. It has the schematic representation shown in Fig 1. Selection lines S_2, S_1 and S_0 select the particular input to be multiplexed and applied to the output.

Strobe S acts as an enable signal. If strobe =1, the chip 74151 is disabled and output y = 0. If strobe = 0 then the chip 74151 is enabled and functions as a multiplexer. Table 1 shows the multiplex function of 74151 in terms of select lines.

Table 1.

Strobe	Select Lines			Output
S	S_2	S_1	S_0	Y
1	X	X	X	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7

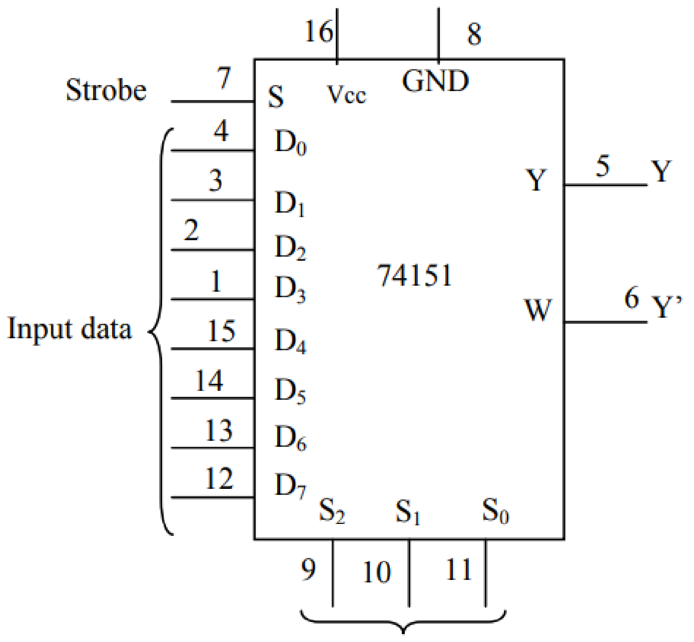


Fig.1 IC type 74151 Multiplexer 8×1

74153 is a dual 4 line-to-1 line multiplexer. It has the schematic representation shown in Fig 2. Selection lines S1 and S0 select the particular input to be multiplexed and applied to the output IY {1 = 1, 2}.
Each of the strobe signals *IG* {I = 1, 2} acts as an enable signal for the corresponding multiplexer.
Table 2. Shows the multiplex function of 74153 in terms of select lines.

Note that each of the on-chip multiplexers act independently from the other, while sharing the same select lines S_1 and S_0.

Table 2

	Multiplexer 1		
Strobe	Select lines		Output
1G	S_1	S_0	1Y
1	X	X	0
0	0	0	$1D_0$
0	0	1	$1D_1$
0	1	0	$1D_2$
0	1	1	$1D_3$

	Multiplexer 2		
Strobe	Select lines		Output
2G	S_1	S_0	2Y
1	X	X	0
0	0	0	$2D_0$
0	0	1	$2D_1$
0	1	0	$2D_2$
0	1	1	$2D_3$

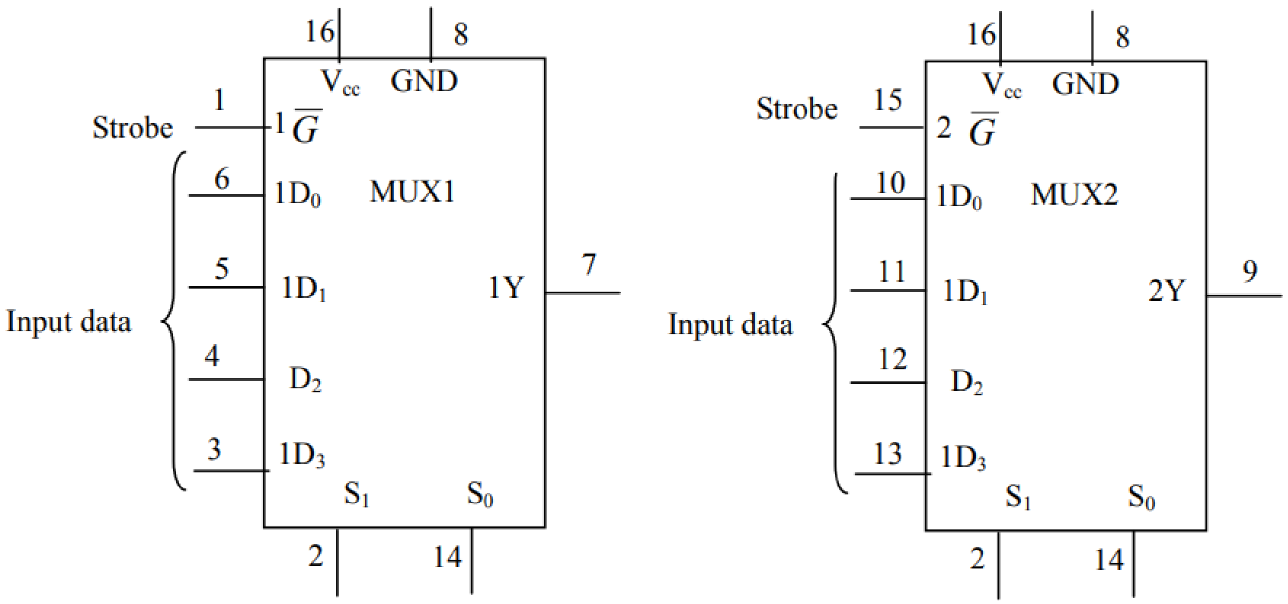


Fig.2 Chip 74153

IC 7446 is a BCD to seven segment decoder driver. It is used to convert the combinational circuit outputs in BCD forms into 7 segment digits for the 7 segment LED display units. Just like experiment #4.

Procedure:

Part I: Parity Generator:

a) Design a parity generator by using a 74151 multiplexer. Parity is an extra bit attached to a code to check that the code has been received correctly.

Odd parity bit means that the number of 1's in the code including the parity bit is an odd number. Fill the output column of the truth table in Table 2 for a 5-bit code in which four of the bits (A, B, C, D) represents the information to be sent and fifth bit (x), represents the parity bit. The required parity is an odd parity.

The inputs B, C and D correspond to the select inputs of 74151. Complete the truth table in Table 3 by filling in the last column with 0, 1, A or A'.

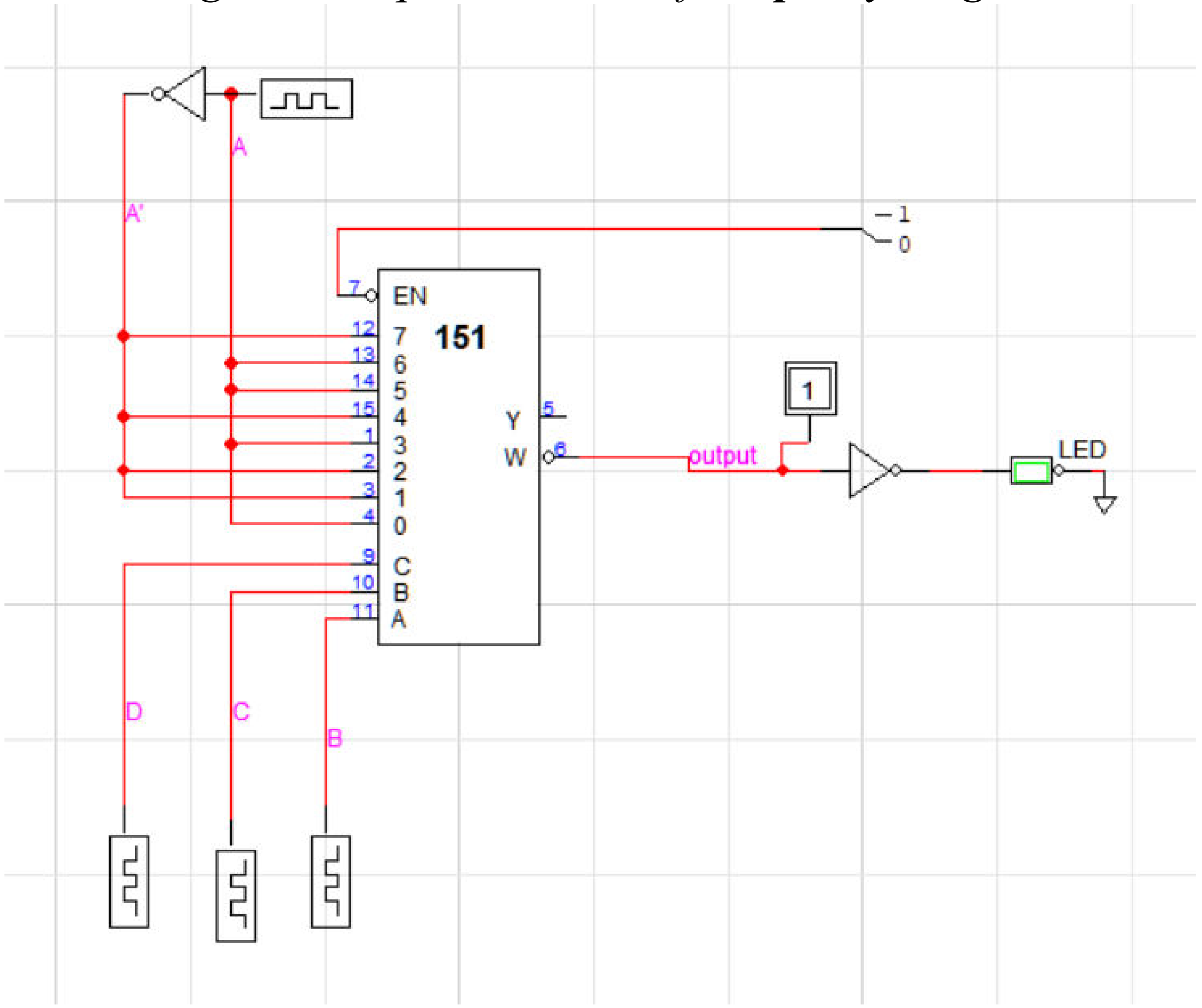
Ans:

Following is the table that demonstrates the corresponding parity bit along with the 4 original bits for **odd parity check**

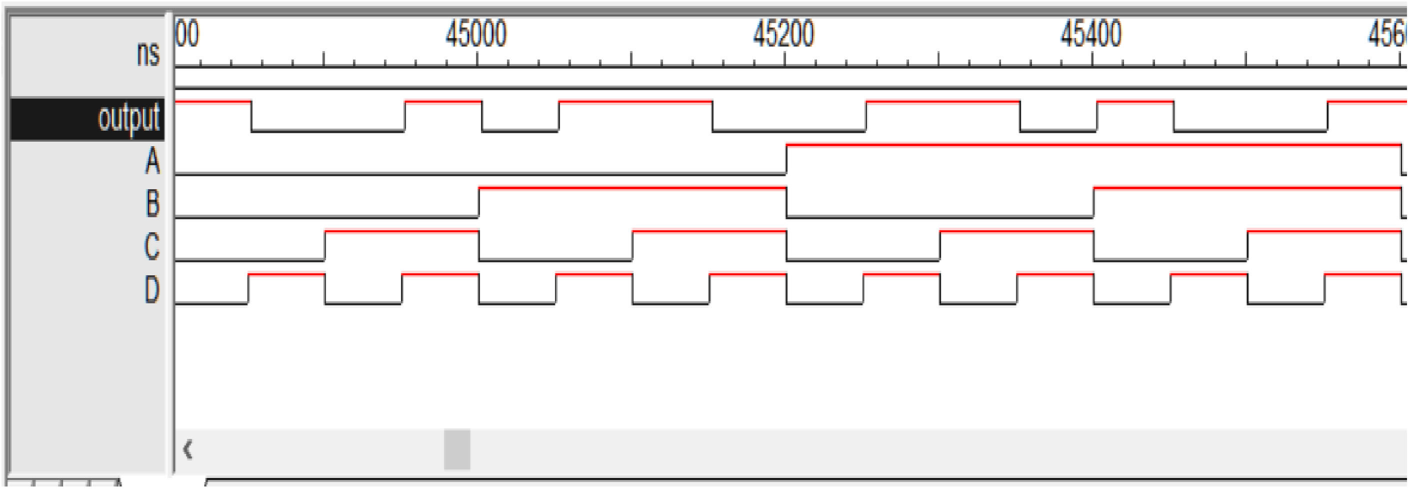
Inputs				Output	Connects Data to
A	B	C	D	x	
0	0	0	0	1	A’
0	0	0	1	0	A
0	0	1	0	0	A
0	0	1	1	1	A’
0	1	0	0	0	A
0	1	0	1	1	A’
0	1	1	0	1	A’
0	1	1	1	0	A
1	0	0	0	0	A’
1	0	0	1	1	A
1	0	1	0	1	A
1	0	1	1	0	A’
1	1	0	0	1	A
1	1	0	1	0	A’
1	1	1	0	0	A’
1	1	1	1	1	A

b) Simulate the circuit using LogicWorks, use 74-151 multiplexer and Binary switches for inputs and Binary Probes for outputs. The 74151 has one output for Y and another inverted output W. Use A and A' for providing values for inputs 0-7. The internal values "A, B, C" are used for selection inputs B, C, and D. Simulate the circuit and test each input combination filling in the table shown below. In the Lab connect the circuit and verify the operations. Connect an LED to the multiplexer output so that it represents the parity bit which lights any time when the four bits input have even parity.

Following is the required circuit of the ***parity bit generator:***



Circuit assembled in LogicWorks 5



Simulated in LogicWorks 5

Part 2: Vote Counter:

A committee is composed of a chairman (C), a senior member (S), and a member (M). The rules of the committee state that:

- The vote of the member (M) will be counted as 2 votes
- The vote of the senior member (S) will be counted as 3 votes.
- The vote of the chairman (C) will be counted as 5 votes.

Each of these persons has a switch to close ("l") when voting yes and to open ("0") when voting no.

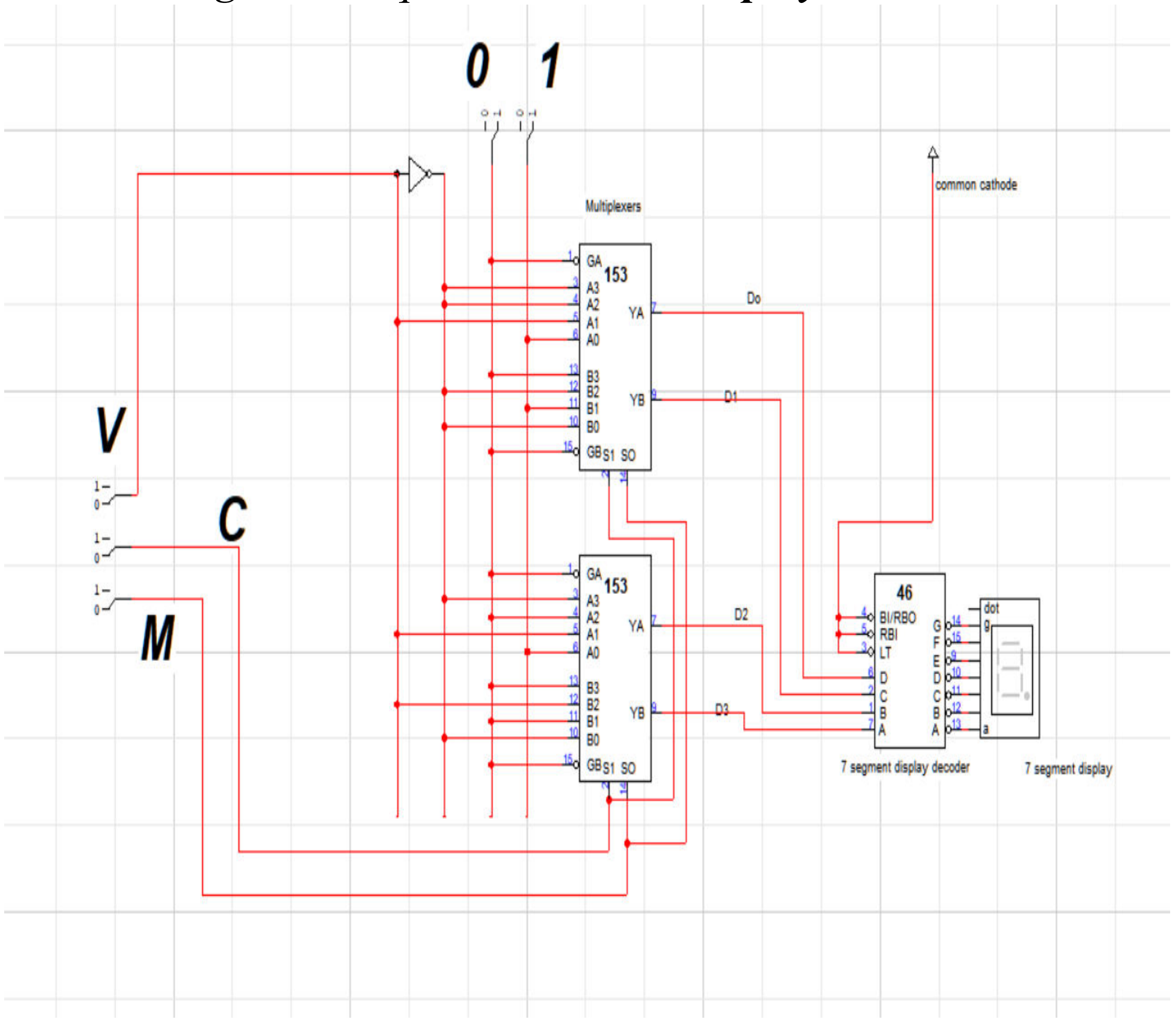
It is necessary to design a circuit that displays the total number of votes for each issue. Use a seven segment display and a decoder to display the required number.

If all members vote no for an issue the display should be blank. (Recall from Experiment #5, that a binary input 15 into the 7446 blanks all seven segments).

If all members vote yes for an issue, the display should be 0. Otherwise the display shows a decimal number equal to the number of 'yes' votes. Use two 74153 units, which include four multiplexers to design the combinational circuit that converts the inputs from the members' switch to the BCD digit for the 7446.

In LogicWorks use +5V for Logic 1 and ground for Logic 0 and use switches for C, S, and M. Use two chips 74153 and one decoder 7446 verify your design and get a copy of your circuit with the pin numbers to Lab so that you could connect the hardware in exactly the same way.

Following is the required circuit to ***display the vote count:***:



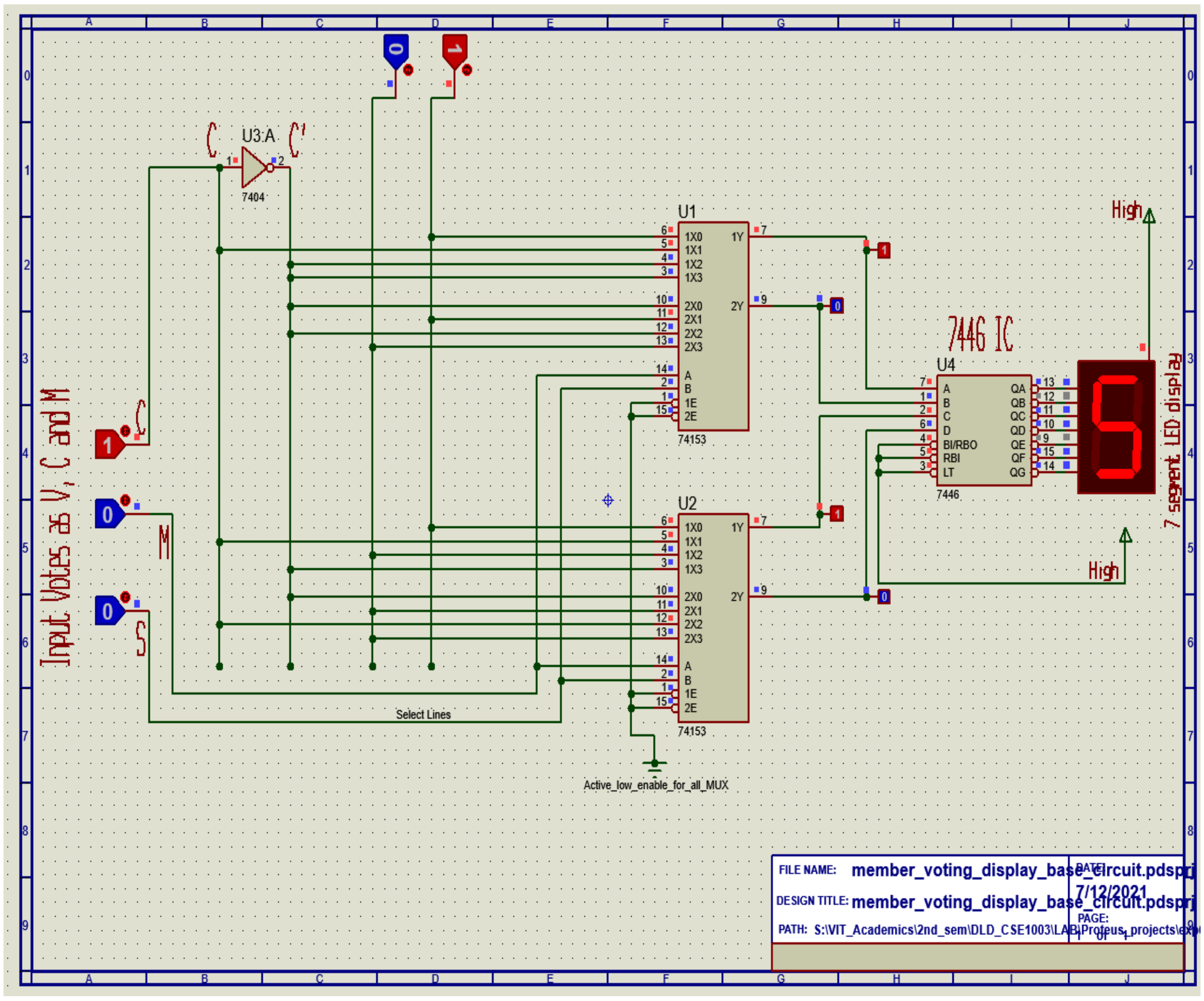
Here,

Let vote of Chairperson --> C (weightage, C = 5)

Let vote of senior member --> S (weightage, S = 3)

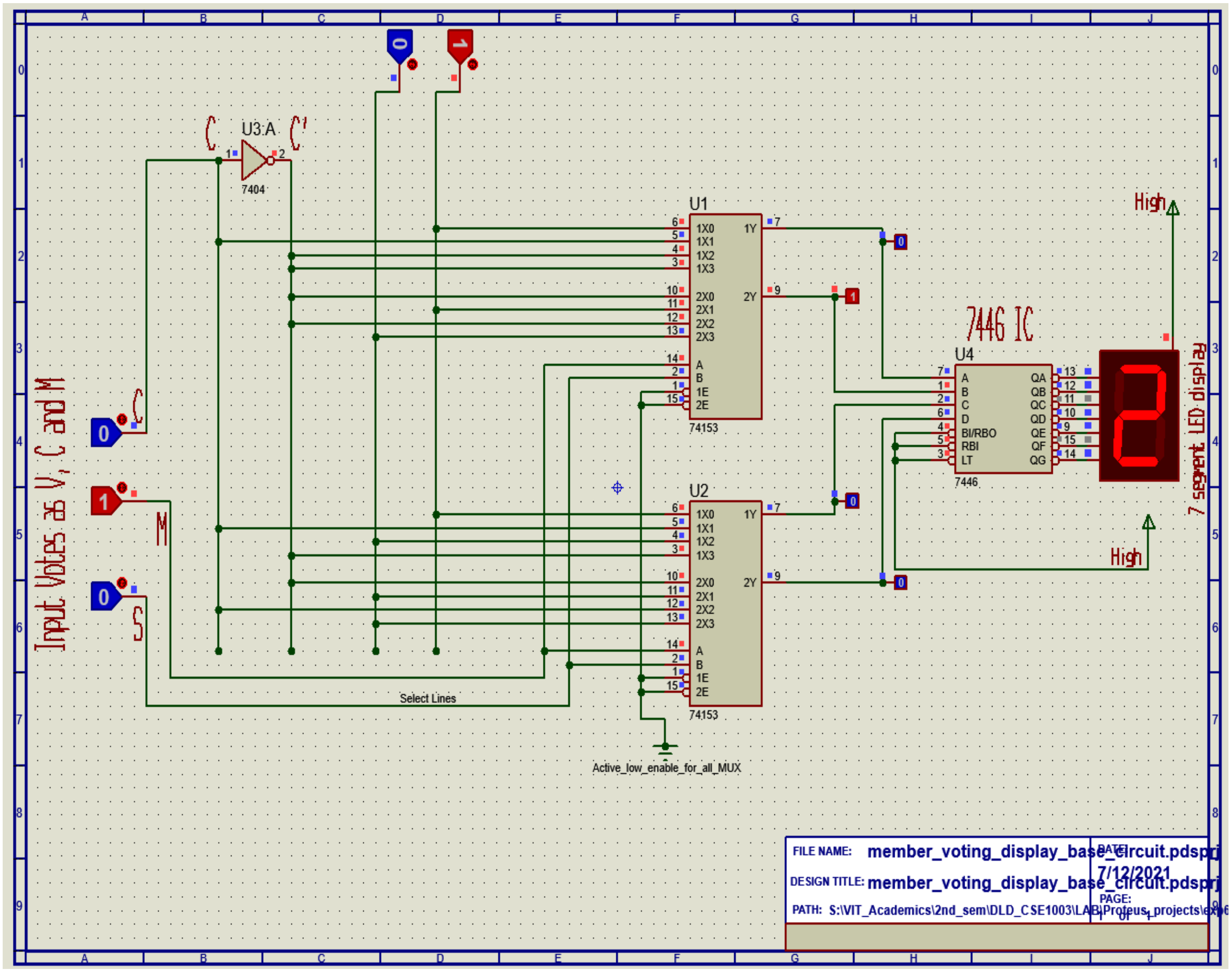
Let vote of general member --> M (weightage, M=2)

When only C votes in favor,
C=1,
S=0,
M=0
Display out =5


U3:A
7404
C
C'
U1
74153
U2
74153
U4
7446
7446 IC
7 segment LED display
High
Input Votes as V, C and M
M
S
Select Lines
Active_low_enable_for_all_MUX
FILE NAME: member_voting_display_base_circuit.pdsprj
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
DATE: 7/12/2021
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\

Simulated in Proteus 8 pro

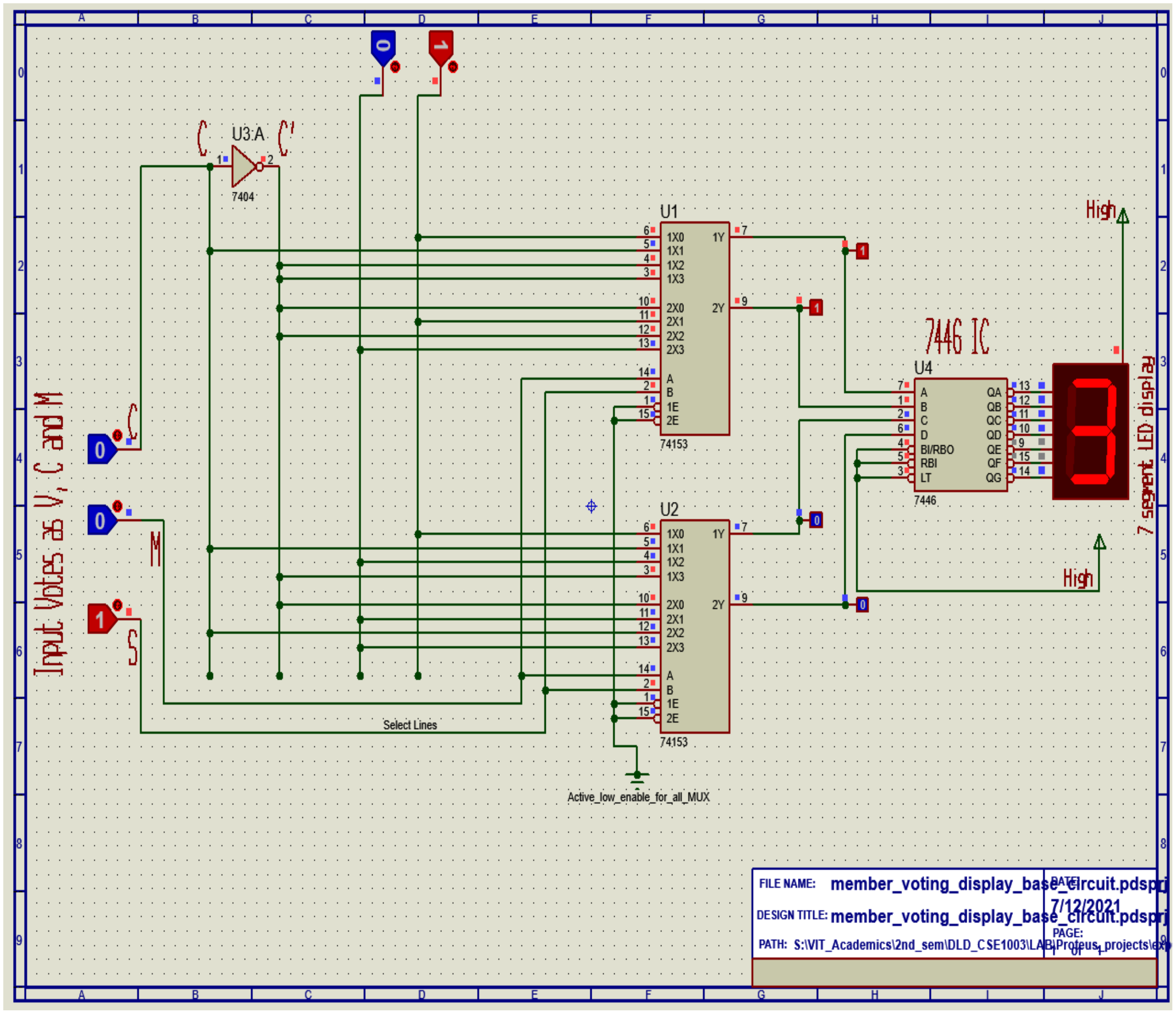
When only M votes in favor,
C=0,
M=1,
S=0
Display out =2



U3:A
7404
C
C'
U1
74153
U2
74153
7446 IC
U4
7446
High
High
7 segment LED display
Input Votes as V, C and M
C
M
S
Select Lines
Active_low_enable_for_all_MUX
FILE NAME: member_voting_display_base_circuit.pdsprj
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\ex6
DATE: 7/12/2021
PAGE: 1 of 1

Simulated in Proteus 8 pro

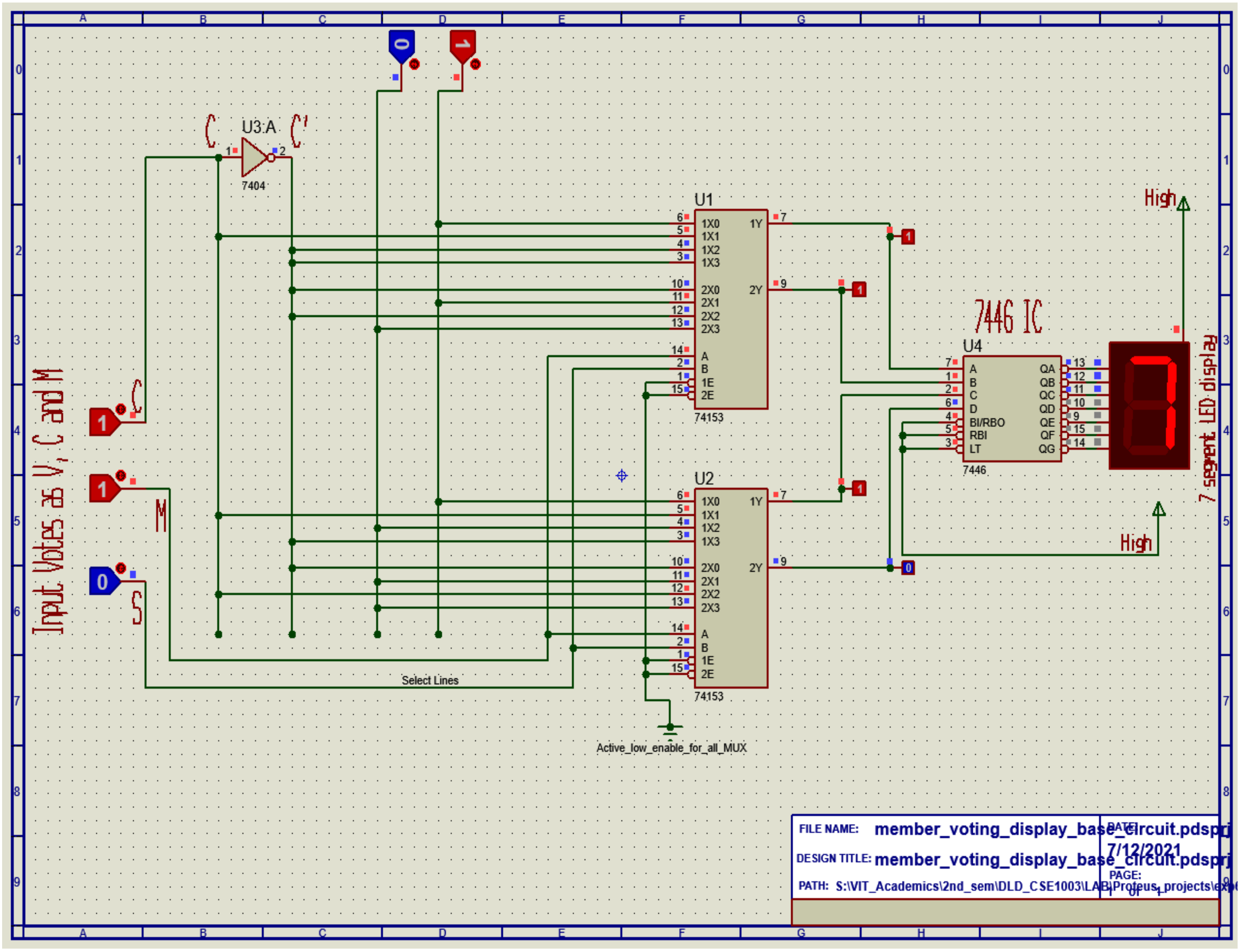
When only S votes in favor,
C=0,
M=0,
S=1
Display out = 3



FILE NAME: member_voting_display_base_circuit.pdsprj
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
DATE: 7/12/2021
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\exp
PAGE:

Simulated in Proteus 8 pro

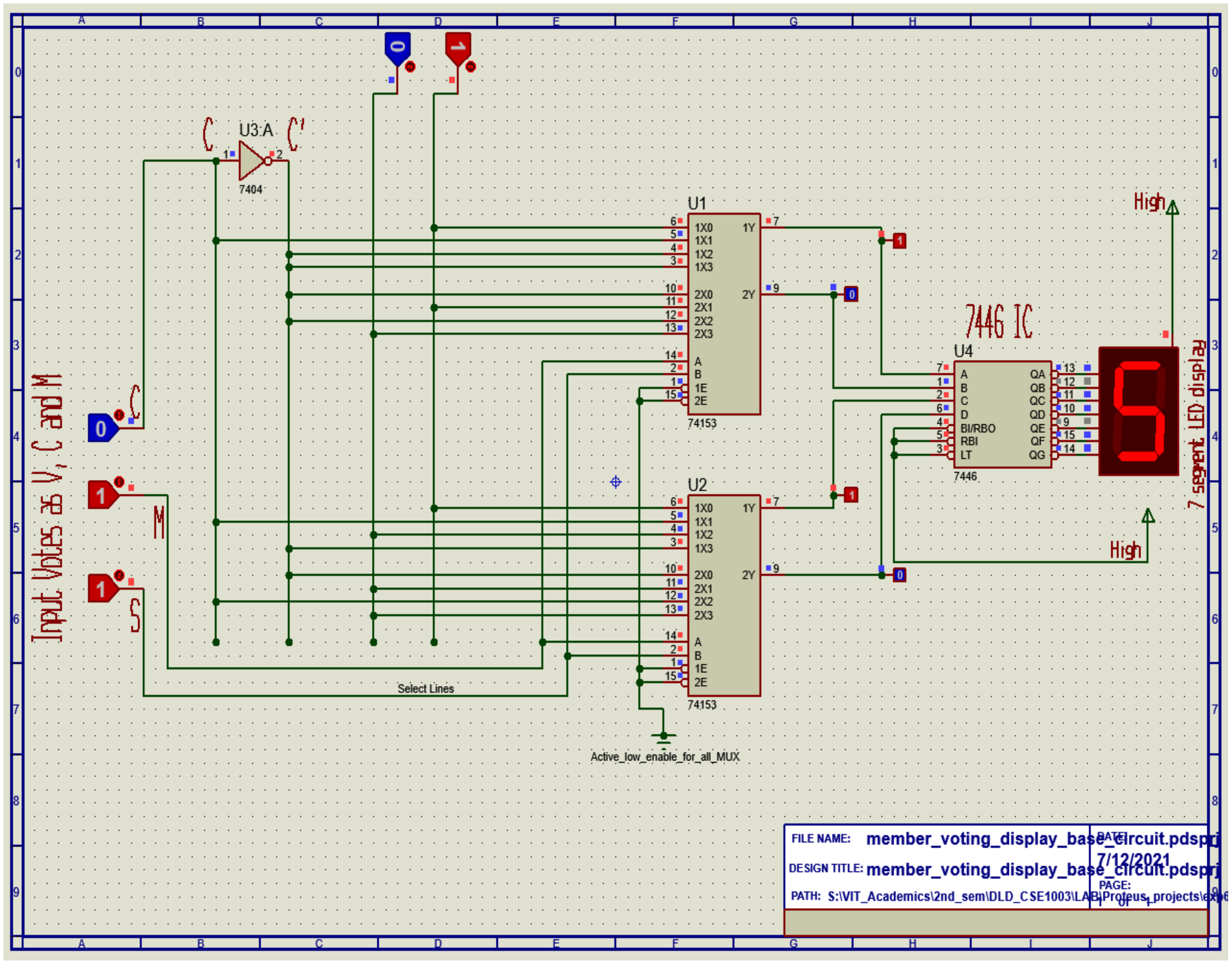
When C and M votes in favor,
C=1,
M=1,
S=0
Display out = 7



FILE NAME: member_voting_display_base_circuit.pdsprj
DATE: 7/12/2021
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
PAGE:
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\ex

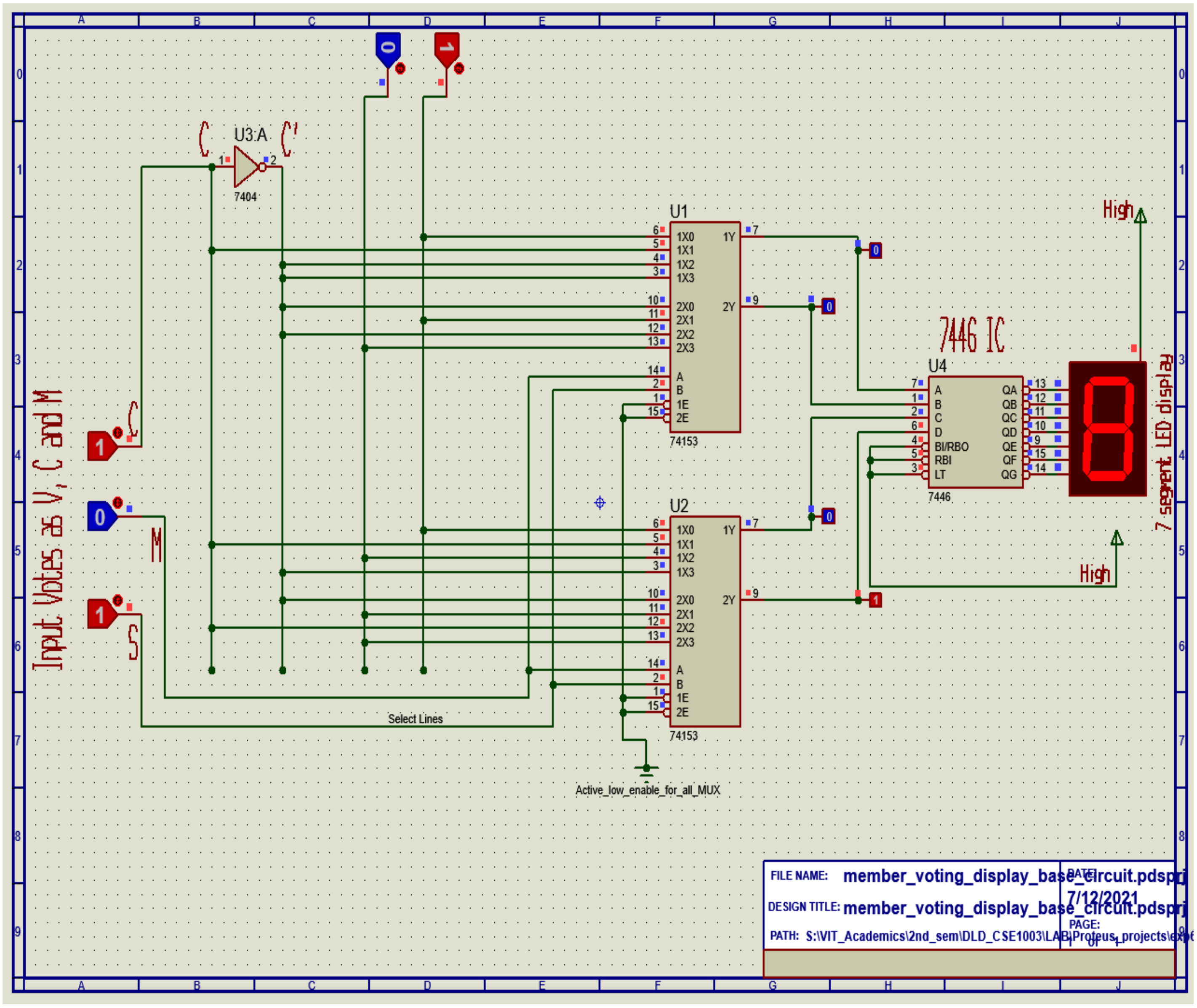
Simulated in Proteus 8 pro

When M and S votes in favor,
C=0,
M=1,
S=1
Display out = 5


Input Votes as V, C and M
C
M
S
U3:A
C'
7404
U1
74153
U2
74153
7446 IC
U4
7446
High
High
7 segment LED display
Select Lines
Active_low_enable_for_all_MUX
1X0
1X1
1X2
1X3
2X0
2X1
2X2
2X3
A
B
1E
2E
1Y
2Y
BI/RBO
RBI
LT
QA
QB
QC
QD
QE
QF
QG
FILE NAME: member_voting_display_base_circuit.pdsprj
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\ex
DATE: 7/12/2021
PAGE: 1 of 1

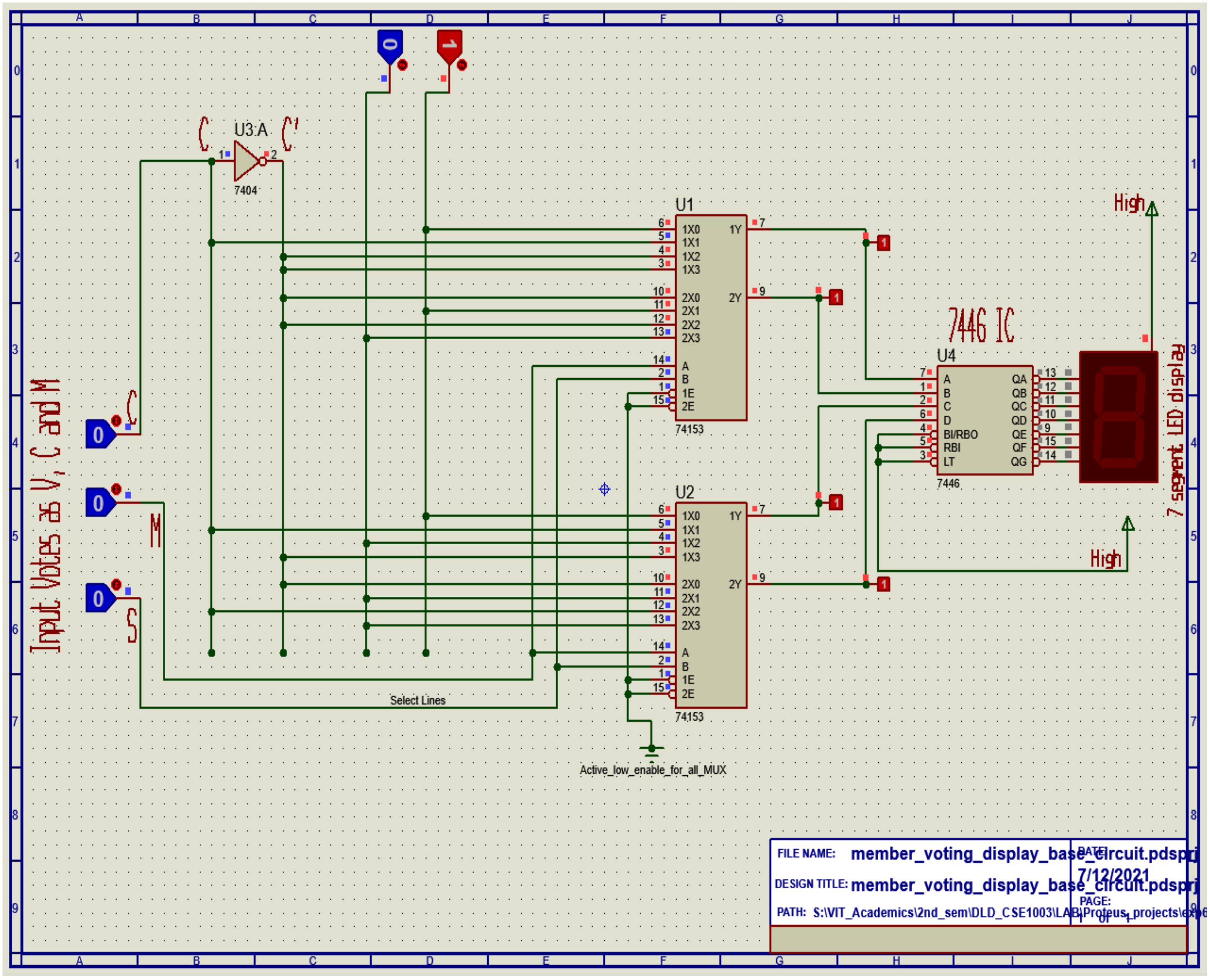
Simulated in Proteus 8 pro

When C and S votes in favor,
C=1,
M=0,
S=1
Display out = 8


U3:A
C
C'
7404
U1
1X0
1X1
1X2
1X3
2X0
2X1
2X2
2X3
1Y
2Y
A
B
1E
2E
74153
U2
74153
7446 IC
U4
7446
BI/RBO
RBI
LT
QA
QB
QC
QD
QE
QF
QG
High
High
7 segment LED display
Input Votes as V, C and M
M
S
Select Lines
Active_low_enable_for_all_MUX
FILE NAME: member_voting_display_base_circuit.pdsprj
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\exp
DATE: 7/12/2021
PAGE: 1 of 1

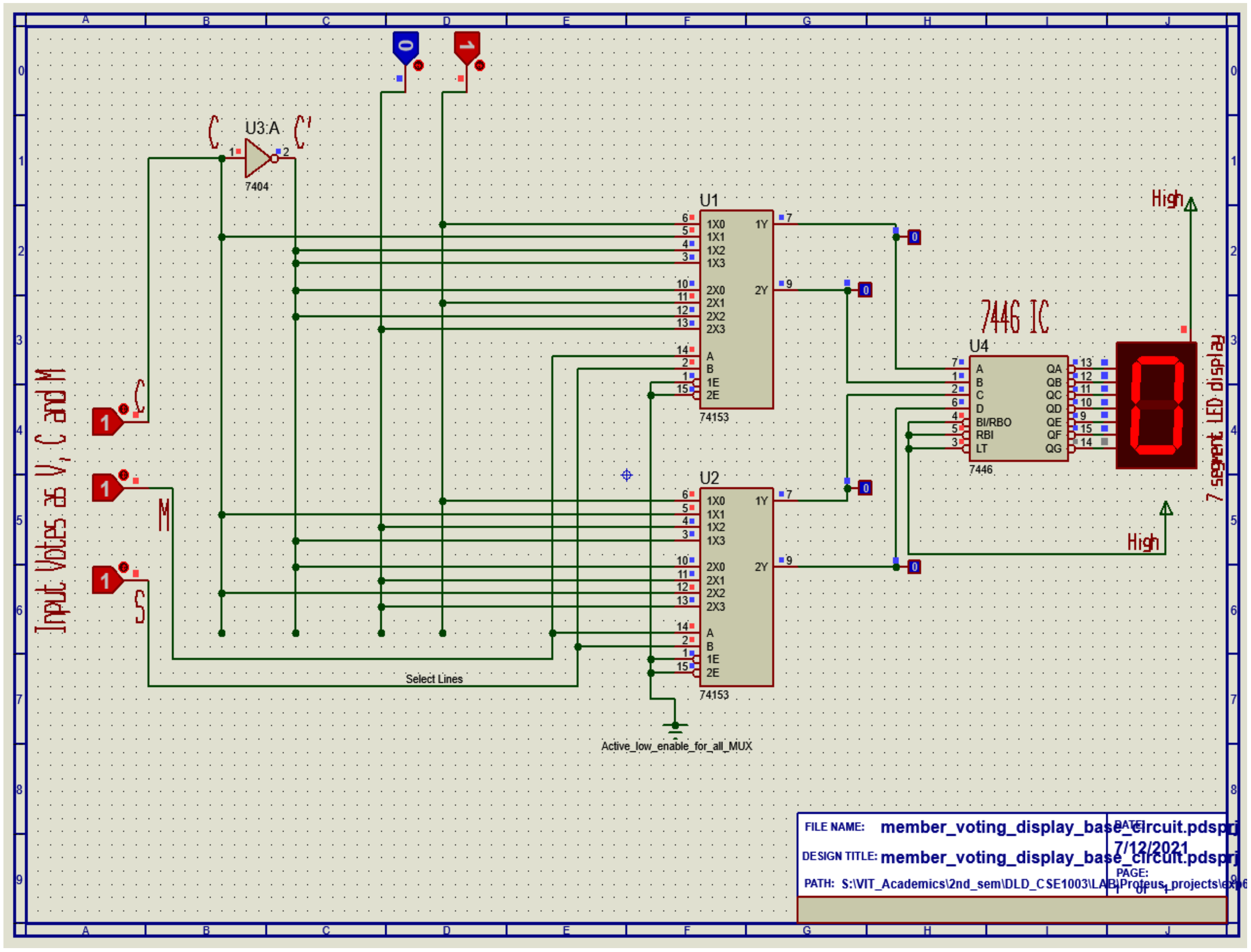
Simulated in Proteus 8 pro

When no one votes in favor,
C=0,
M=0,
S=0
Display out = null.


Input Votes as V, C and M
C
M
S
U3:A
C'
7404
U1
74153
U2
74153
1X0
1X1
1X2
1X3
2X0
2X1
2X2
2X3
1Y
2Y
A
B
1E
2E
Select Lines
Active_low_enable_for_all_MUX
7446 IC
U4
7446
BI/RBO
RBI
LT
QA
QB
QC
QD
QE
QF
QG
High
High
7 segment LED display
FILE NAME: member_voting_display_base_circuit.pdsprj
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\
DATE: 7/12/2021
PAGE: 1 of 1

Simulated in Proteus 8 pro

When all votes in favor,
C=1,
M=1,
S=1
Display out = 0 (Although it's 10, 1 as carry can't be displayed).



Input Votes as V, C and M
C
M
S
U3:A
C'
7404
U1
U2
74153
1X0
1X1
1X2
1X3
2X0
2X1
2X2
2X3
1Y
2Y
A
B
1E
2E
7446 IC
U4
BI/RBO
RBI
LT
QA
QB
QC
QD
QE
QF
QG
7446
High
7 segment LED display
Select Lines
Active_low_enable_for_all_MUX
FILE NAME: member_voting_display_base_circuit.pdsprj
DESIGN TITLE: member_voting_display_base_circuit.pdsprj
DATE: 7/12/2021
PATH: S:\VIT_Academics\2nd_sem\DLD_CSE1003\LAB\Proteus_projects\exp6
PAGE: 1 of 1

Simulated in Proteus 8 pro

--*THE END*--