Exp No: 1 (a)

BASIC ARITHMETIC OPERATIONS

AIM:

To perform basic arithmetic operations using LabVIEW. Addition, Subtraction, Multiplication and division.

THEORY:

To give the 2 inputs numeric controls are selected in Front panel and all arithmetic operators can be selected from Functions palette. Addition, subtraction, multiplication and division are the basic arithmetic operations.

PROCEDURE:

Step 1: Start the LabVIEW and create new project and select the blank VI. And click finish button.

Step 2: Press ctrl +T button to align Create front and block diagram panel.

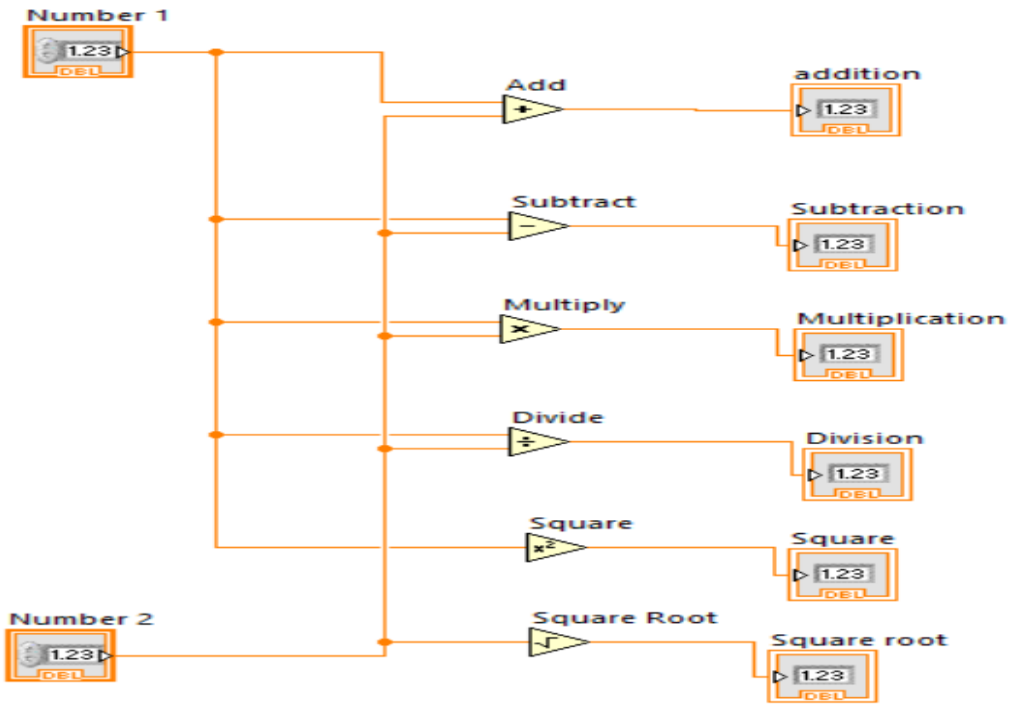
Step 3: Numeric controls are given as inputs and numeric indicators are given as output they are selected by right clicking on the front panel.

Step 4: Different arithmetic operators such as addition, subtraction, multiplication and division are generated in block diagram panel.

Step 5: Using wiring operation inputs and outputs are connected to the respective operators in the block diagram panel.

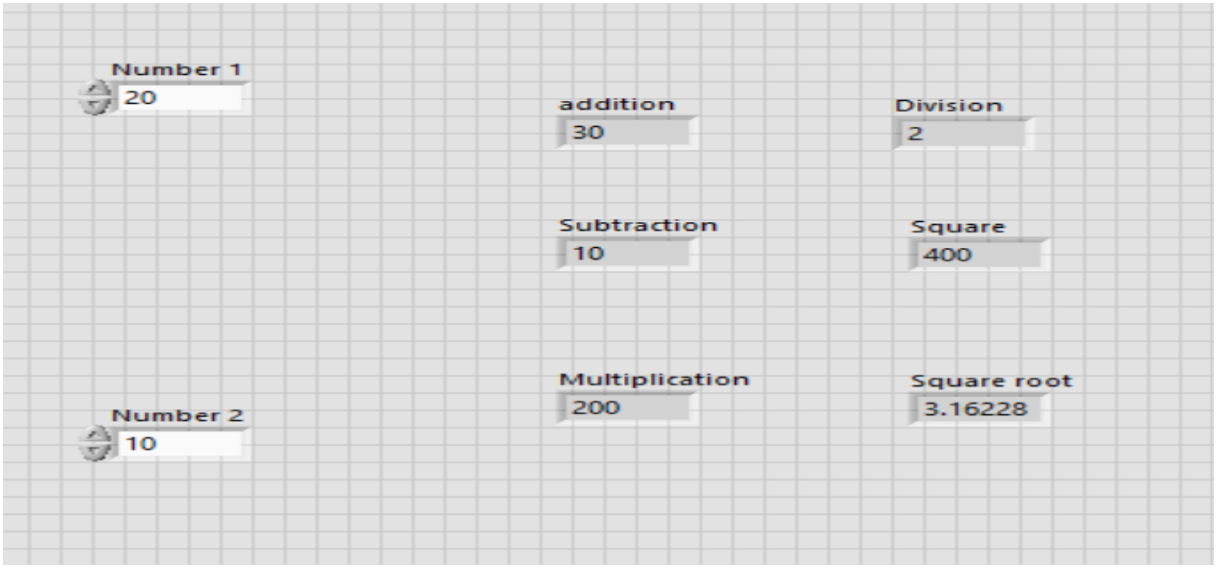
Step 6: Input values are given in the front panel and the program is executed. Hence the output is generated.

BLOCK DIAGRAM:



The block diagram for basic arithmetic operations using LabVIEW is shown below in Fig.1

OUTPUT:



RESULT:

The arithmetic operations were performed and the result is verified using LabVIEW.

Exp No: 1(b)

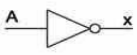
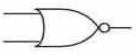
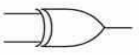
BOOLEAN OPERATIONS

AIM: To perform Boolean operations using LabVIEW. AND,OR,NOT,XOR , NAND

THEORY:

The truth table of Boolean operations like AND, OR,NOT,XOR ,and NAND is shown below in Fig 2.1. The Boolean inputs are given by push buttons in the control palette .All the Boolean operators can be taken from the functions palette.

Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$
Symbol						

Truth Table

A	X
0	1
1	0

B	A	X
0	0	0
0	1	0
1	0	0
1	1	1

B	A	X
0	0	1
0	1	1
1	0	1
1	1	0

B	A	X
0	0	0
0	1	1
1	0	1
1	1	1

B	A	X
0	0	1
0	1	0
1	0	0
1	1	0

B	A	X
0	0	0
0	1	1
1	0	1
1	1	0

PROCEDURE:

Step 1: Start the LabVIEW and select the blank VI.

Step 2: Create front and block diagram panel.

Step 3: To perform Boolean operation push buttons are taken as inputs and round LED as output.

Step 4: Different Boolean operations such as AND, OR, XOR, NOT, NAND are selected from the block diagram panel.

Step 5: Boolean inputs and outputs are wired in the block diagram panel.

Step 6: Logic values 0 & 1 are given in the front panel and the program is executed.

BLOCK DIAGRAM:

The block diagram of Boolean operations is shown below

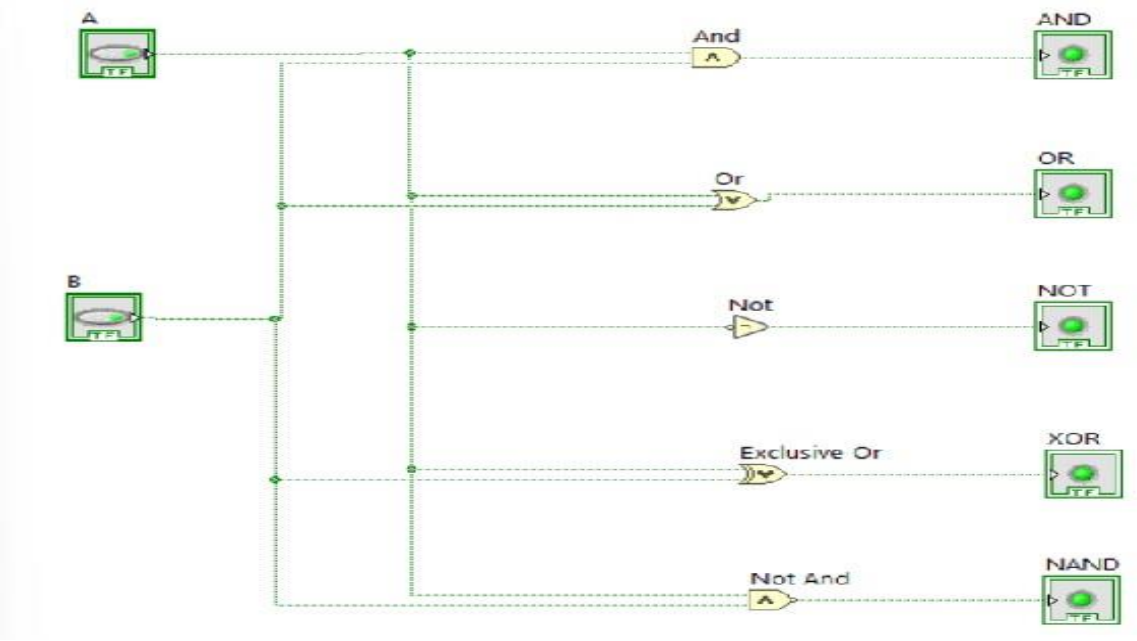
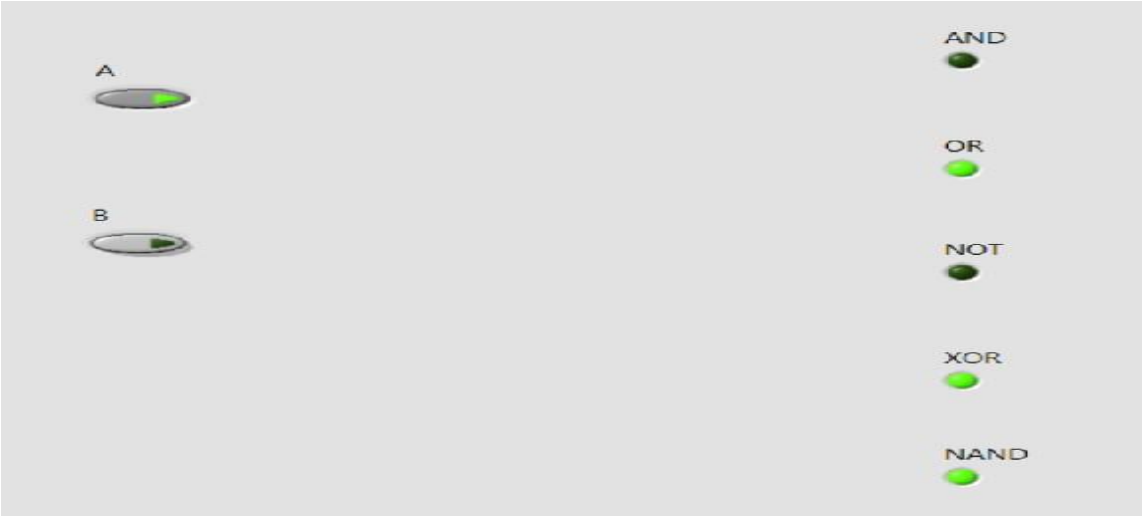


Fig: Block diagram of Boolean operations

OUTPUT/FRONT PANEL:

The output of Boolean operations is shown below in Fig.2.3.



RESULT: The truth table of Boolean operations AND, OR, NOT, XOR , and NAND gates are verified.

Exp No: 1(c)

Half Adder and Full Adder

AIM: To perform half adder and Full Adder using LabVIEW.

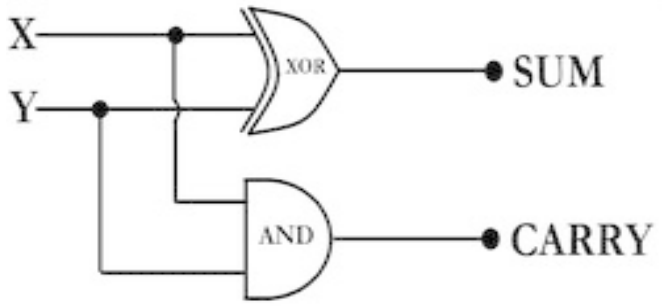
THEORY: Half Adder is a combinational logic circuit that is designed by connecting one EX-OR gate and one AND gate. The half-adder circuit has two inputs: A and B, which add two input digits and generate a carry and a sum. Full Adder is a combinational circuit that adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. The Logical Expression for half adder and Full adder is given as

Sum = A ⊕ B Sum = A ⊕ B ⊕ C

Carry = AB Carry = AB+BC+AC

Half Adder :

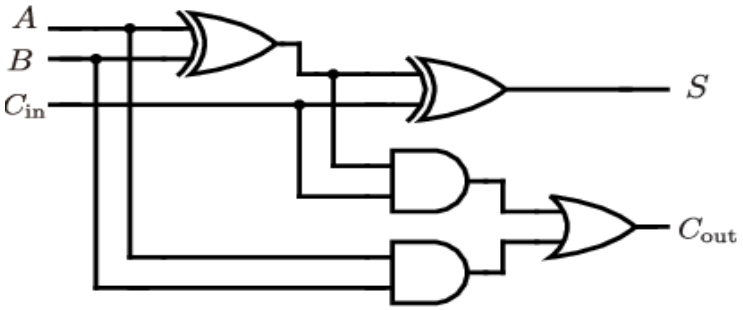
(a)



(b)

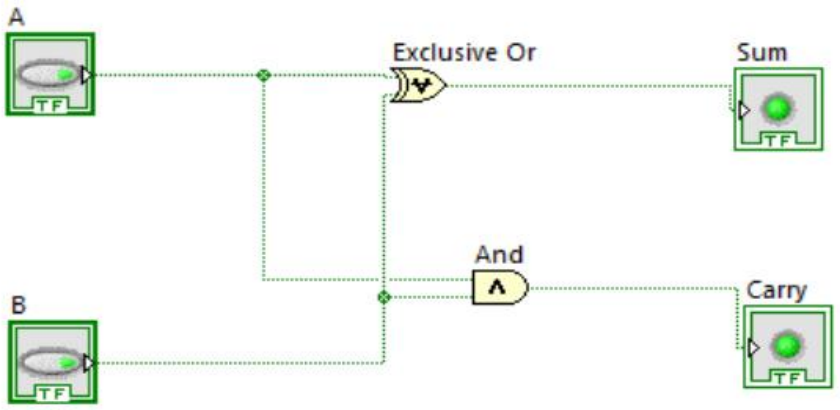
X	Y	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full Adder:

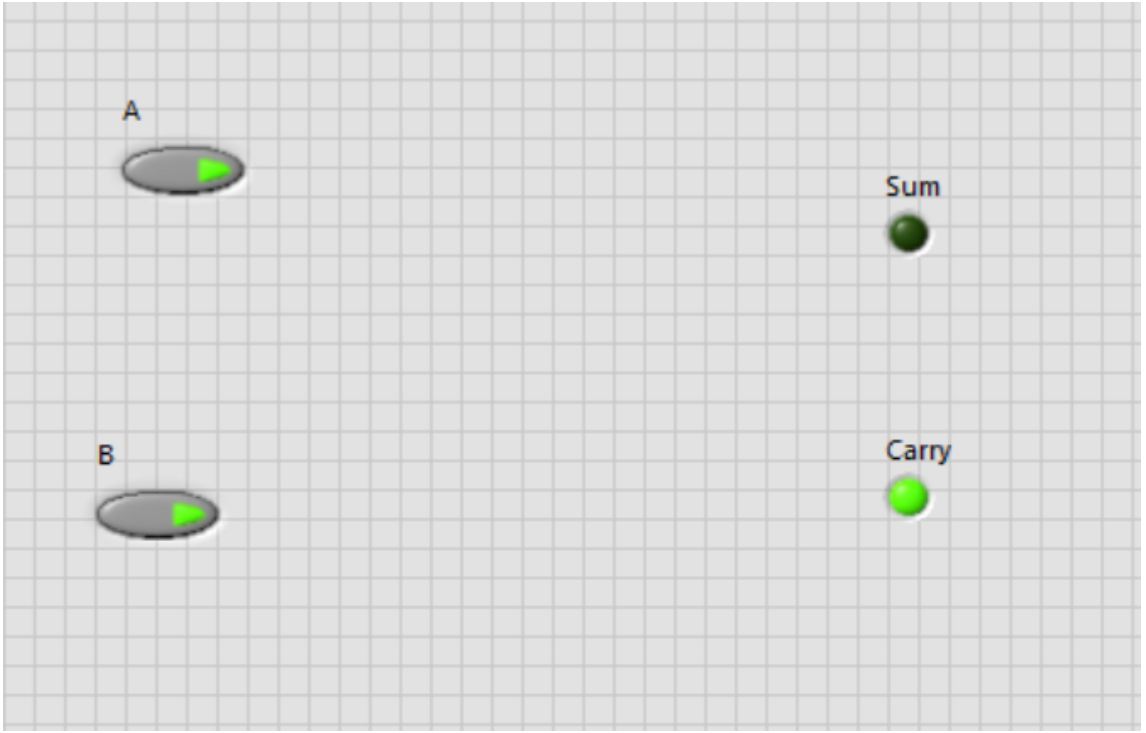


Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

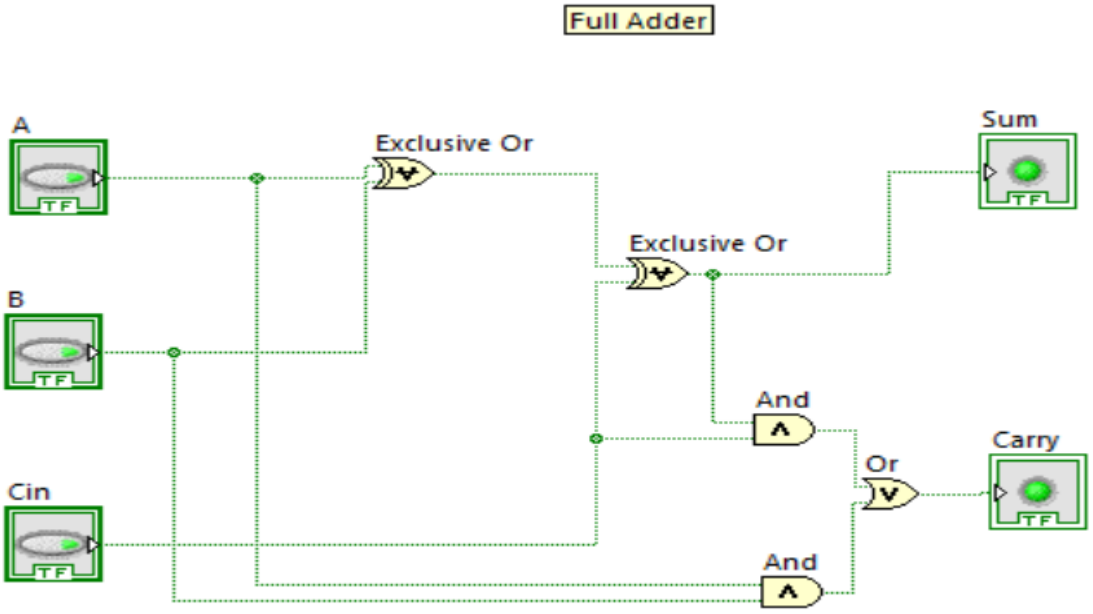
Half Adder Block diagram:



OUTPUT/FRONT PANEL:



Full Adder Block diagram



OUTPUT/FRONT PANEL:

RESULT: The truth table of Half adder and Full adder are observed and verified,

Exp No: 2(a)

SUM OF ‘n’ NUMBERS USING ‘FOR’ LOOP

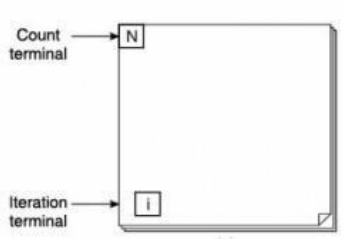
AIM:

To find the sum of ‘n’ numbers using FOR loop.

THEORY:

Usually sum of n numbers, we take for natural numbers.The equation to calculate sum of n numbers is n(n+1)/2.So we want to take sum of first 3 numbers, n=3.Then as per the equation it becomes 3(4)/2=6.i.e 3+2+1=6.

FOR loop in LabVIEW is shown below in Fig 3.1.



In FOR loop, shift registers are used if we make use of previous results. N is the count and ‘i’ is the iteration terminal and initial value is zero.

PROEDURE:

Step 1: Create blank VI.

Step 2: First of all, move to the front panel and press right then control palette and choose numeric option place in the front panel and its correspondent will show block diagram.

Step 3: Right click on the block diagram panel, select program, go to structures and select a FOR loop.

Step 4: Right click on the border of the FOR loop and select add shift register, borders are converted into shift register.

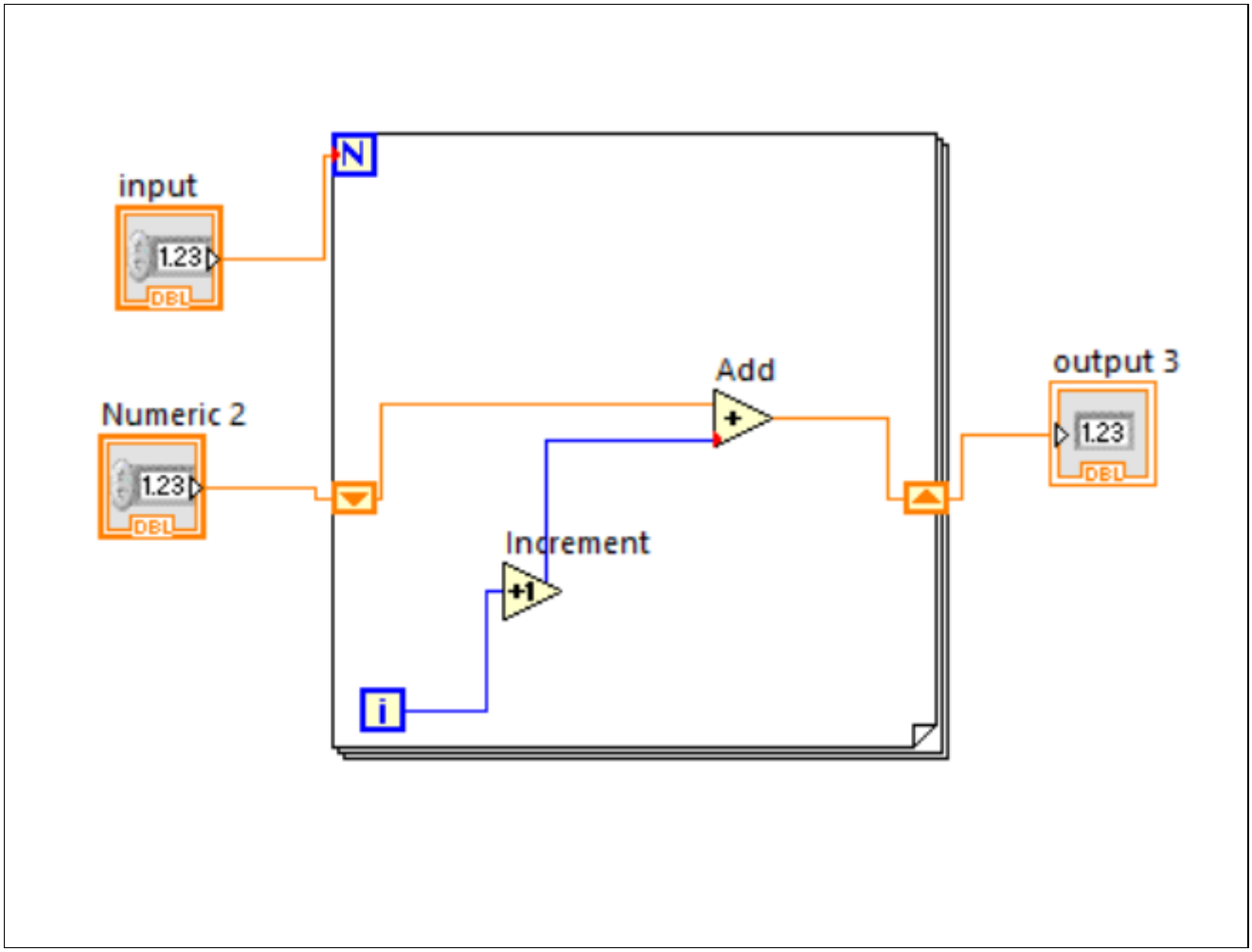
Step 5: In block diagrams select the numeric button and choose constant and connect it to register.

Step 6: Select the add button from the numeric tab. Then select the increment option from the numeric tab. Connect the output of this increment block to the remaining input of the add block.

Step 7: At the right shift register click right and from the drop down select create and then select indicator.

Step 8: Inputs are given in the front panel and the program is executed.

Block diagram:



Input:

Input

10

constant

0

Front Panel: Output:

output

55

Result: Thus the sum of 'n' natural numbers using FOR loop is performed in LABview

Exp No: 2(b)

FACTORIAL OF A GIVEN NUMBER USING WHILE LOOP

Aim: To perform the factorial of a given number using WHILE loop.

Algorithm:

Step 1: Create blank VI.

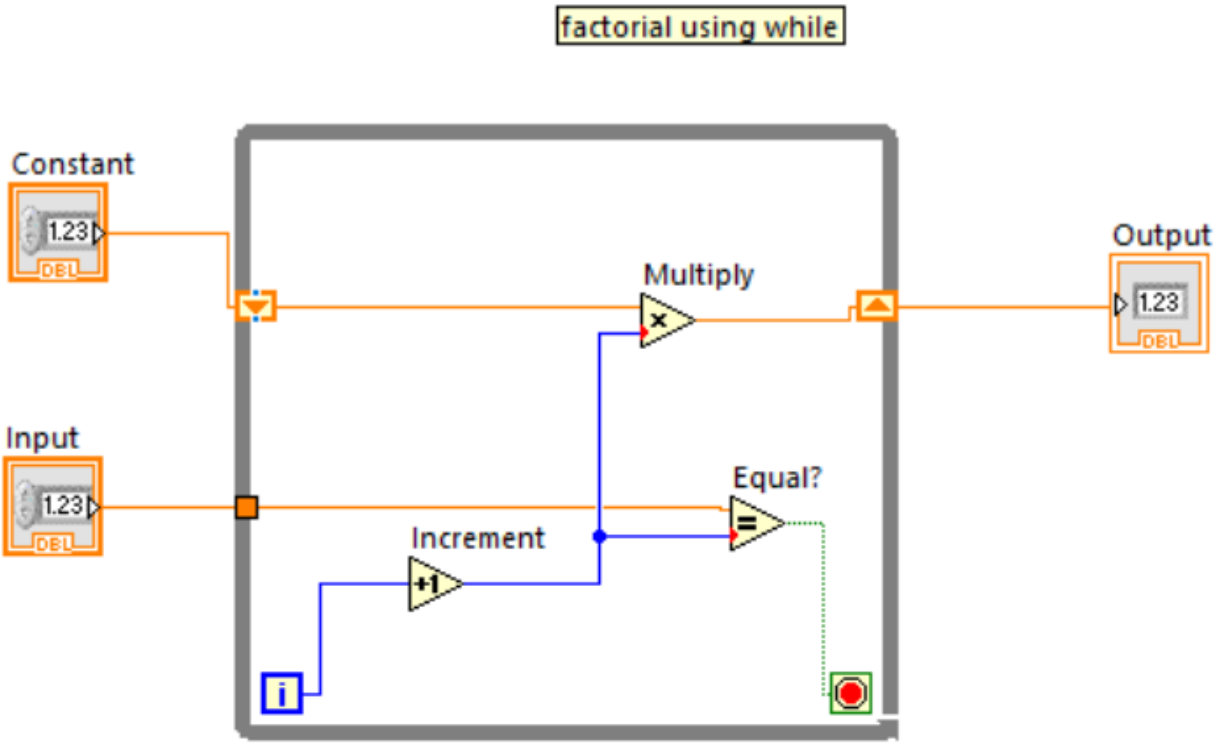
Step 2: Right click on the block diagram panel, select program, go to structures and select a WHILE loop.

Step 3: Right click on the border of the WHILE loop and select add shift register, borders are converted into shift register.

Step 4: Using wiring operations required connections are given in the block diagram.

Step 5: Inputs are given in the front panel and the program is executed.

Block diagram panel:



Output:

Result: Thus the factorial of the given number using WHILE loop is performed.

Exp No: 3(a)

SORTING EVEN NUMBERS USING WHILE LOOP IN AN ARRAY

Aim: To sort even numbers using WHILE loop in an array.

Algorithm:

Step 1: Create blank VI.

Step 2: Right click on the block diagram panel , select program , go to structures and select a WHILE loop.

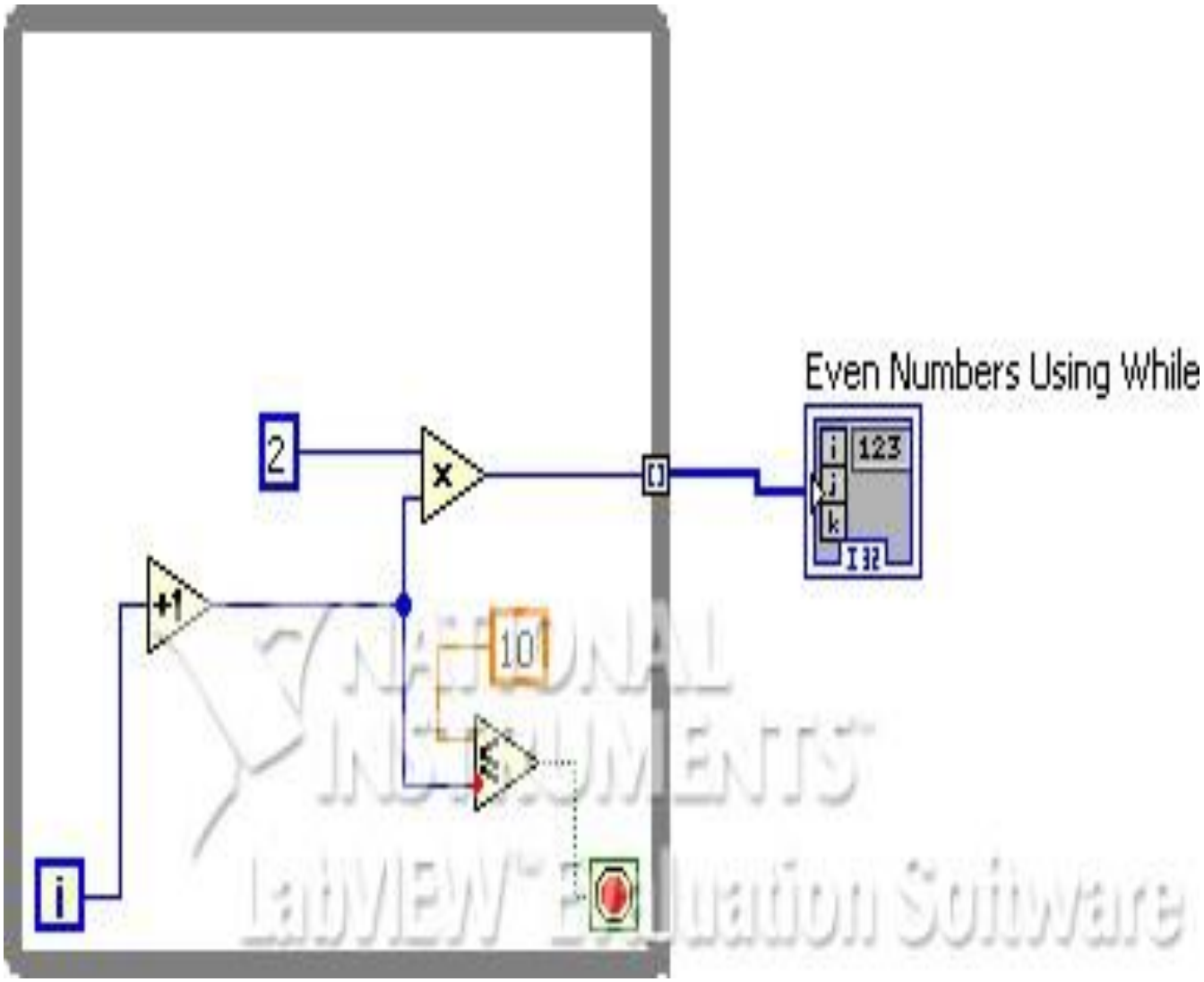
Step 3: Create an array in the front panel and add numeric indicator to it.

Step 4: Add the numeric control in the front panel.

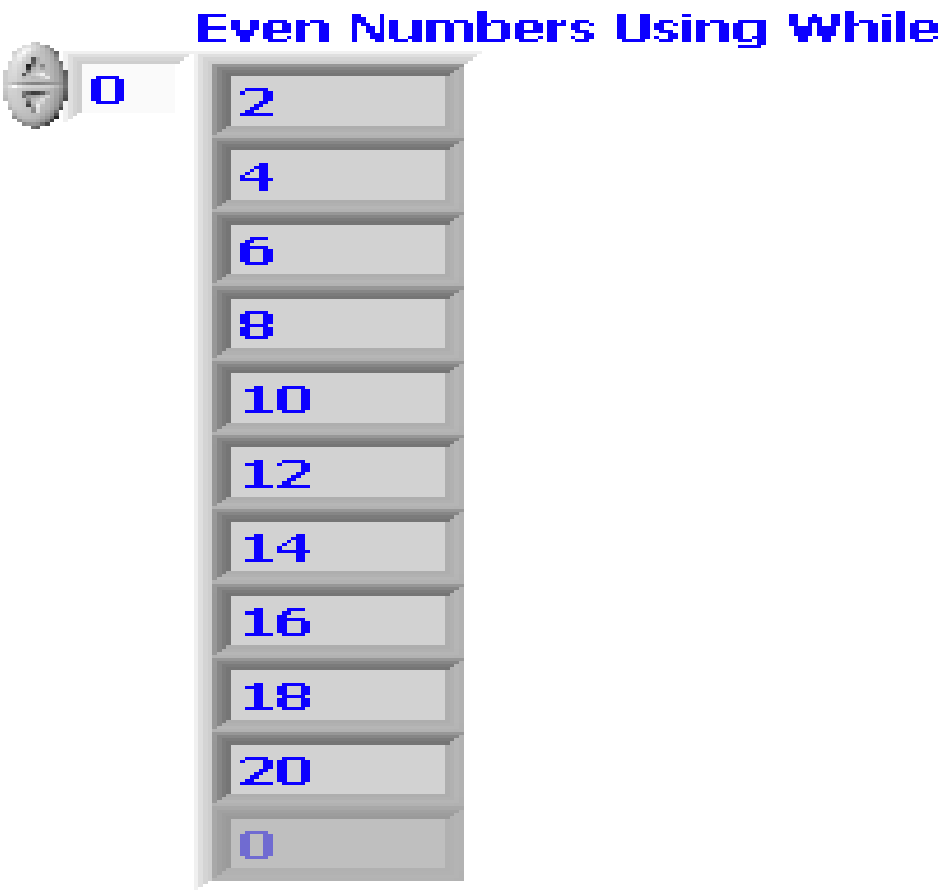
Step 5: Using wiring operations required connections are given in the block diagram.

Step 6: Inputs are given in the front panel and the program is executed.

Block diagram panel:



FRONT PANEL:



Result: Thus the even numbers from the given set of numbers is sorted using WHILE loop in an array.

Exp No: 3(b)

MAXIMUM AND MINIMUM VALUE IN AN ARRAY

Aim: To find the maximum and minimum variable from an array.

Algorithm:

Step 1: Create blank VI.

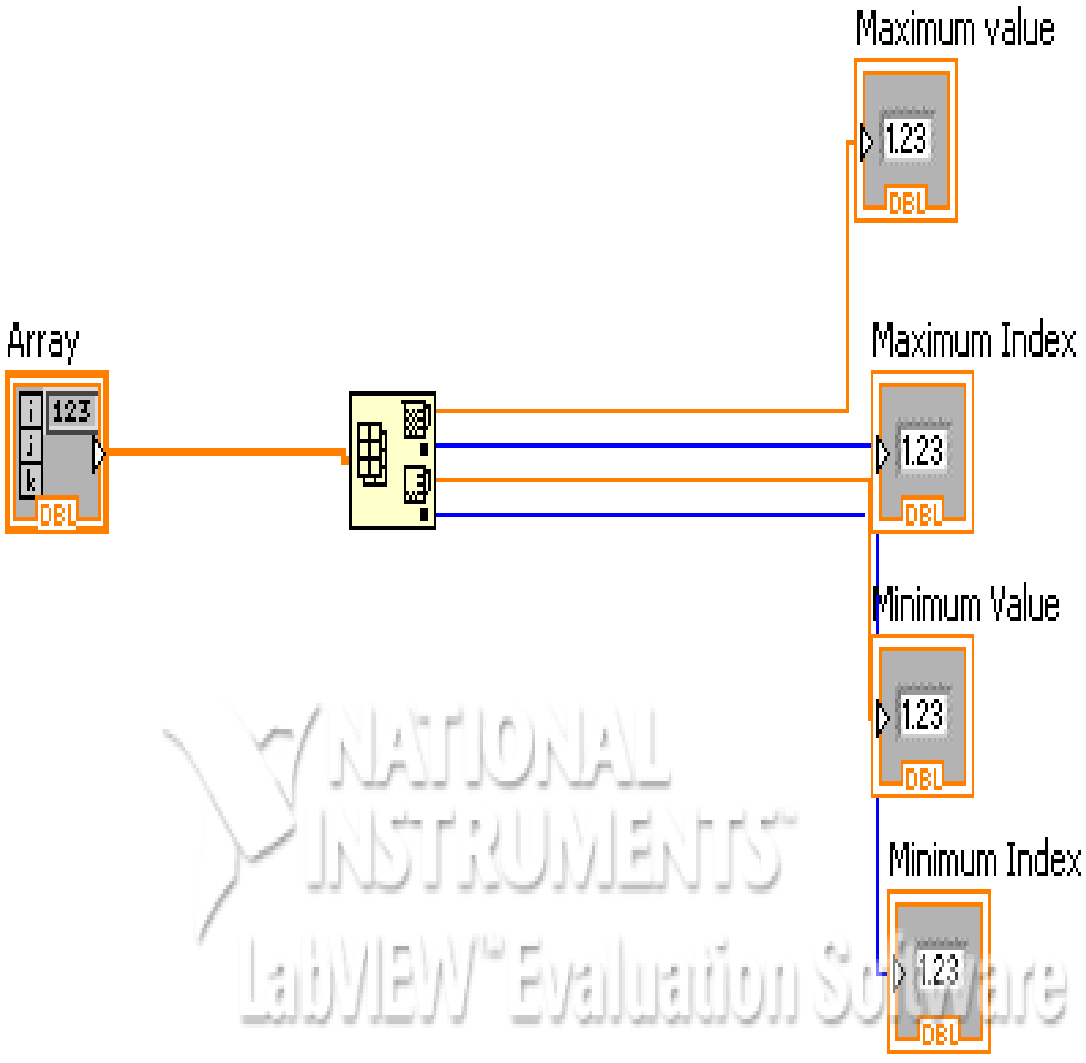
Step 2: Right click on the front panel →modern →array→ array matrix→ numeric control.

Step 3: Create four numeric indicators in the front panel for maximum variable, index, minimum variable and index.

Step 4: Using wiring operations required connections are given in the block diagram.

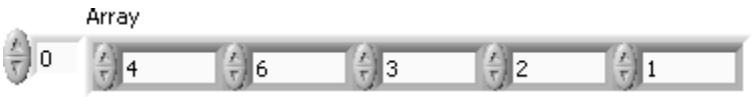
Step 5: Inputs are given in the front panel and the program is executed.

Block diagram panel:



Front Panel:

Input:



Output:

Maximum value

6

Maximum Index

1

Minimum Value

1

Minimum Index

4