PROGRAMME: B.Tech

SEMESTER /YEAR: IV/ II

SUBJECT CODE: 20ECS2

SUBJECT NAME: Modeling, Design and Prototyping

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (Autonomous)

Affiliated to JNTUK, Kakinada & Approved by AICTE, New Delhi.

Accredited by NAAC, NBA Tier-1 for CSE, IT, ECE, EEE & ME and "CPE" status by UGC.

L.B.Reddy Nagar, Mylavaram – 521 230, Andhra Pradesh, India.

List of Experiments

1. BASIC ARITHMETIC OPERATIONS
2. BOOLEAN OPERATIONS
3. SUM OF ‘n’ NUMBERS USING ‘FOR’ LOOP
4. FACTORIAL OF A GIVE NUMBER USING FOR LOOP
5. SUM OF ‘n’ NATURAL NUMBERS USING WHILE LOOP
6. FACTORIAL OF A GIVE NUMBER USING WHILE LOOP
7. SORTING EVEN NUMBERS USING WHILE LOOP IN AN ARRAY
8. ARRAY MAXIMUM AND MINIMUM
9. BUNDLE AND UNBUNDLE CLUSTER
10. FLAT AND STACKED SEQUENCE
11. APPLICATION USING FORMULA NODE
12. MEDIAN FILTER
13. DISCRETE COSINE TRANSFORM
14. CONVOLUTION OF TWO SIGNALS
15. WINDOWING TECHNIQUE

Exp No: 1

BASIC ARITHMETIC OPERATIONS

Aim: To perform basic arithmetic operations using Labview.

Algorithm:

Step 1: Start the Lab view and select the blank VI.

Step 2: Create front and block diagram panel.

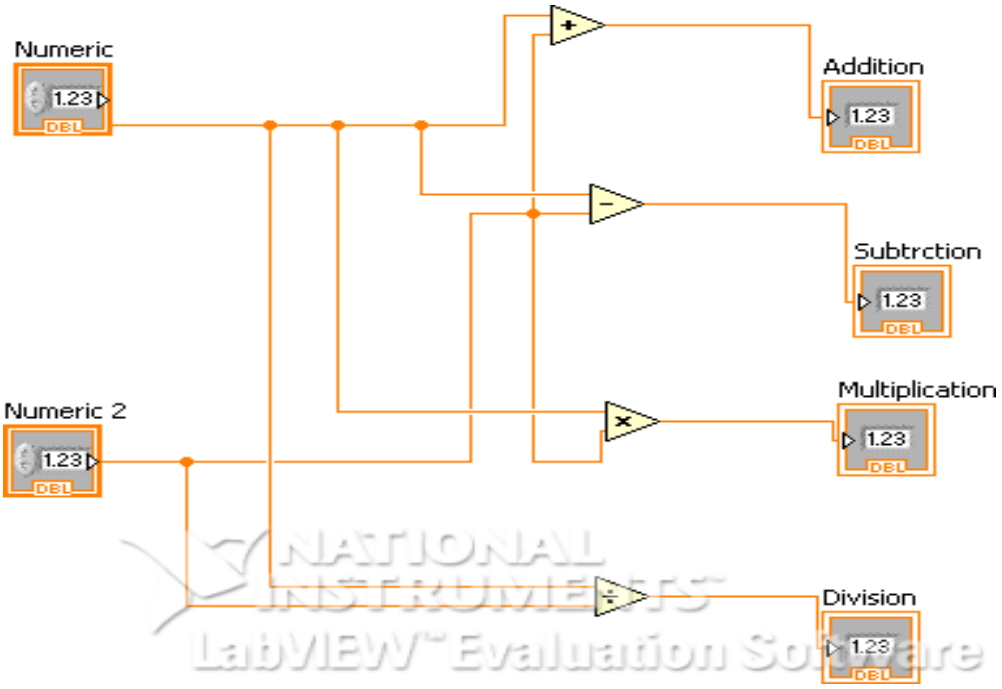
Step 3: Numeric controls are given as inputs and numeric indicators are given as output they are selected by right clicking on the front panel.

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.

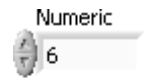
Step 6: Input values are given in the front panel and the program is executed. Hence the output is generated.

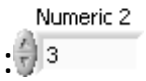
Block diagram panel:



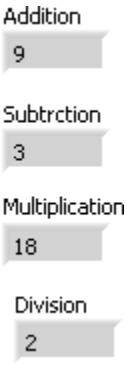
Front panel:

Input:





Output:



Result:

Thus the arithmetic operations were performed and the result is verified using Labview.

Exp No: 2

BOOLEAN OPERATIONS

Aim: To perform Boolean operations using Labview.

Algorithm:

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

Step 2: Create front and block diagram panel.

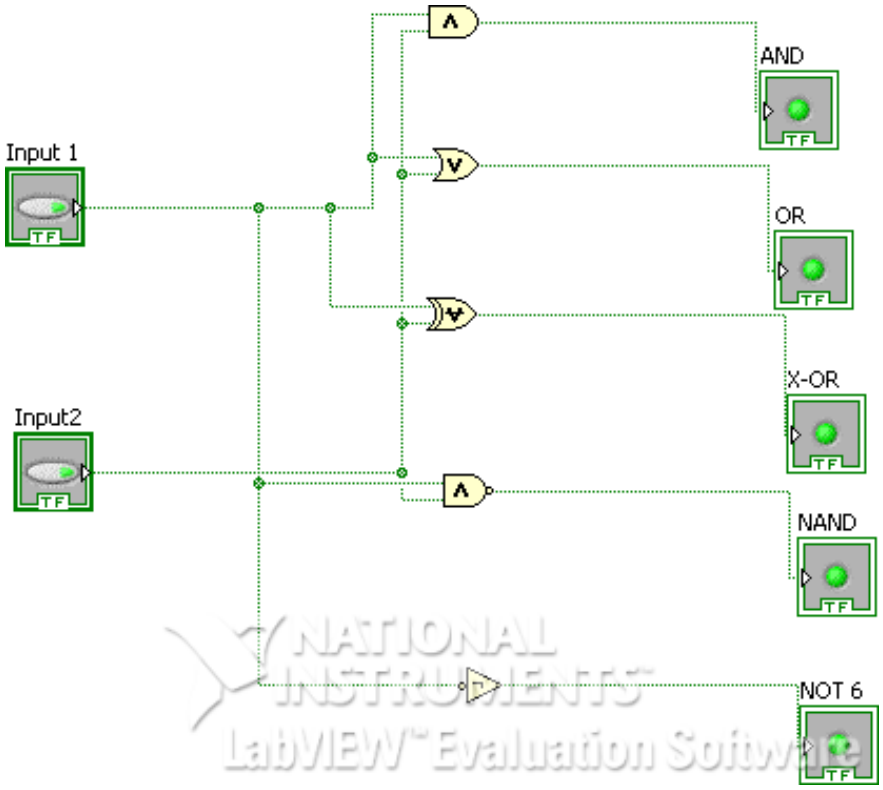
Step 3: To perform Boolean operation push buttons are taken as inputs and round LED as output.

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 panel:



Front panel:

Input:

Input 1

Input2

Output:

AND

OR

X-OR

NAND

NOT 6

Truth Table:

AND:

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR:

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR:

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

NAND:

X1	X2	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOT:

X	Y
0	1
1	0

Result: Thus the Boolean operation using Labview is performed.

Exp No: 3

SUM OF ‘n’ NUMBERS USING ‘FOR’ LOOP

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

Algorithm:

Step 1: Create blank VI.

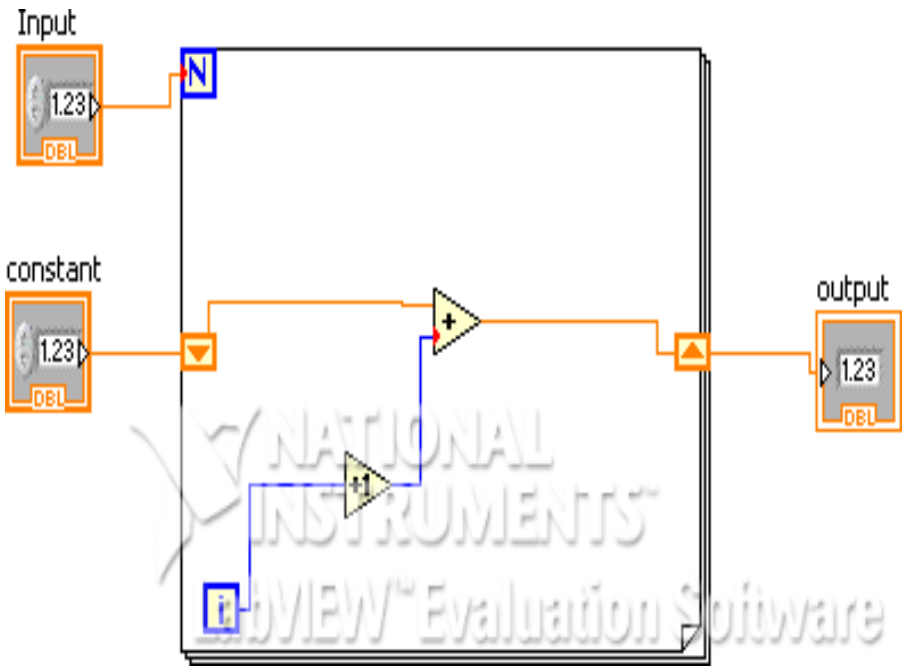
Step 2: Right click on the block diagram panel, select program, go to structures and select a FOR loop.

Step 3: Right click on the border of the FOR 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.

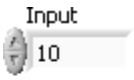
Step 5: Inputs are given in the front panel and the program is executed.

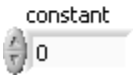
Block diagram panel:



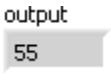
Front panel:

Input:





Output:



Result: Thus the sum of ‘n’ natural numbers using FOR loop is performed in Labview.

Exp No: 4

FACTORIAL OF A GIVE NUMBER USING FOR LOOP

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

Algorithm:

Step 1: Create blank VI.

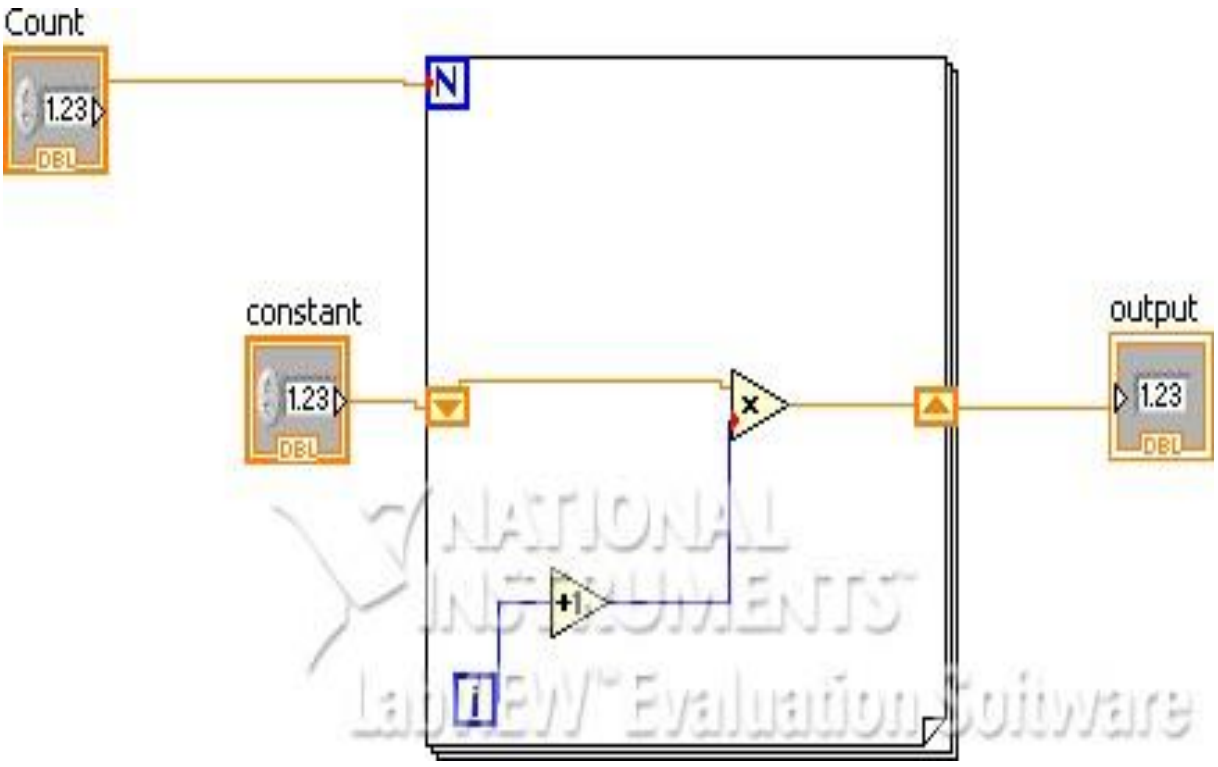
Step 2: Right click on the block diagram panel, select program, go to structures and select a FOR loop.

Step 3: Right click on the border of the FOR 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.

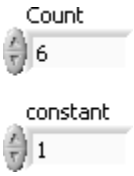
Step 5: Inputs are given in the front panel and the program is executed.

Block diagram panel:

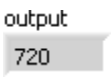


Front panel:

Input:



Output:



Result: Thus the factorial of a given number is using FOR loop is performed in Labview.

Exp No: 5

SUM OF ‘n’ NATURAL NUMBERS USING WHILE LOOP

Aim: To find the sum of n natural numbers 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.

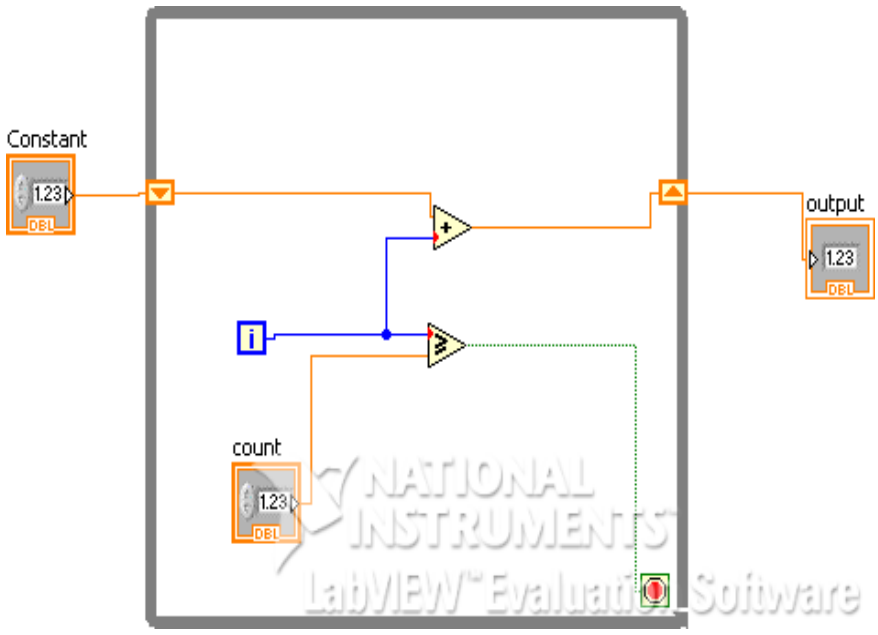
Step 3: The tunnels of the border are converted to shift register.

Step 4: Generate an adder and greater than equal to inside the WHILE loop.

Step 5: Using wiring operation required wiring is done in the block diagram panel.

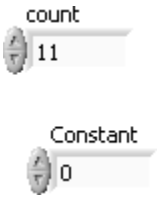
Step 6: Input values are given in front panel and the output is generated.

Block diagram panel:

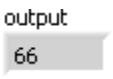


Front panel:

Input:



Output:



Result: Thus the sum of n natural numbers using WHILE loop is performed.

Exp No: 6

FACTORIAL OF A GIVE NUMBER USING WHILE LOOP

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

Algorithm:

Step 1: Create blank VI.

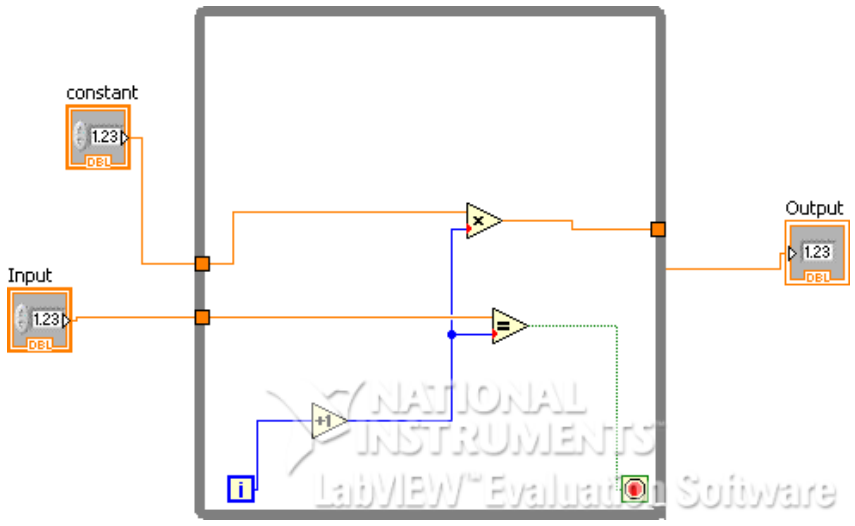
Step 2: Right click on the block diagram panel, select program, go to structures and select a WHILE loop.

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.

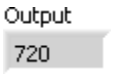
Block diagram panel:



Front panel:

Input:

Output:



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

Exp No: 7

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 structuresand select a WHILE loop.

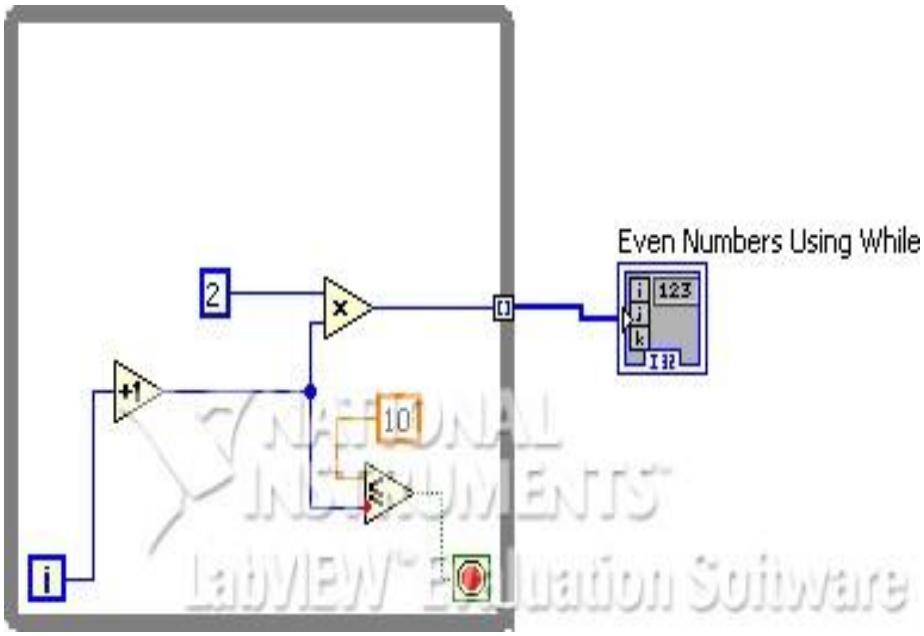
Step 3: Create an array in the front panel and add numeric indicator to it.

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: 8

ARRAY MAXIMUM AND MINIMUM

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

Algorithm:

Step 1: Create blank VI.

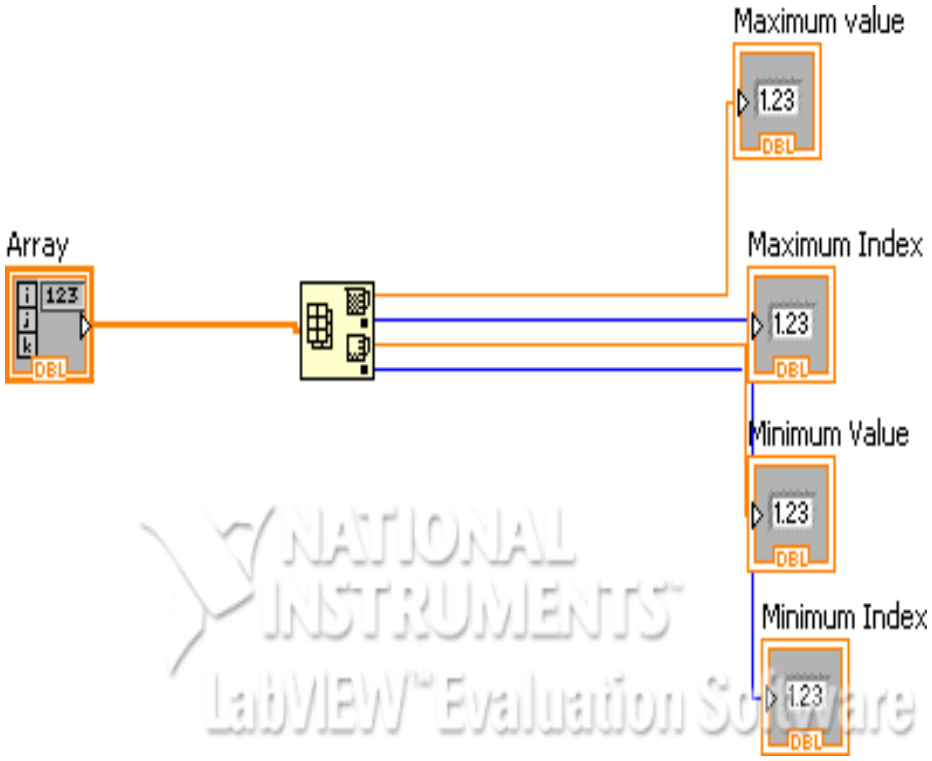
Step 2: Right click on the front panel →modern →array→ array matrix→ numeric control.

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.

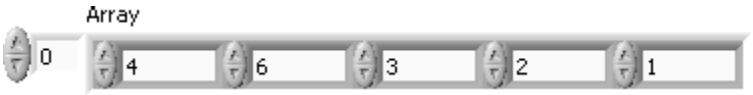
Step 5: Inputs are given in the front panel and the program is executed.

Block diagram panel:

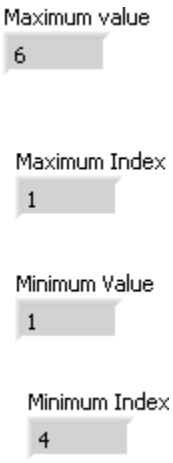


Front panel:

Input:



Output:



Result: Hence the maximum and minimum values of array were displayed using Labview.

Exp No: 9

BUNDLE AND UNBUNDLE CLUSTER

Aim: To bundle and unbundle a cluster.

Algorithm:

For bundling cluster:

Step 1: Create blank VI.

Step 2: Select numeric control, string control for inputs.

Step 3: Select bundle operation from a cluster.

Step 4: Create a new cluster add numeric indicator, string indicator and round LED.

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.

For unbundling cluster:

Step 1: Create a new VI.

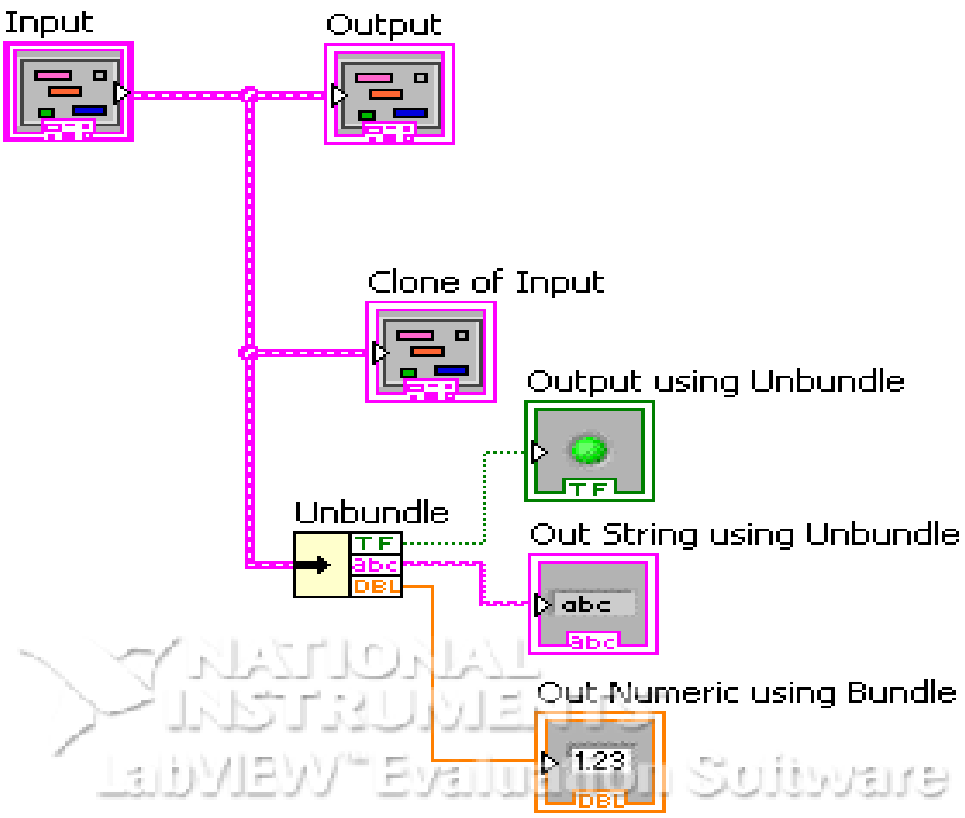
Step 2: Create a cluster with numeric control, string and round LED.

Step 3: Create numeric indicator, Boolean indictor and string indicator.

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.

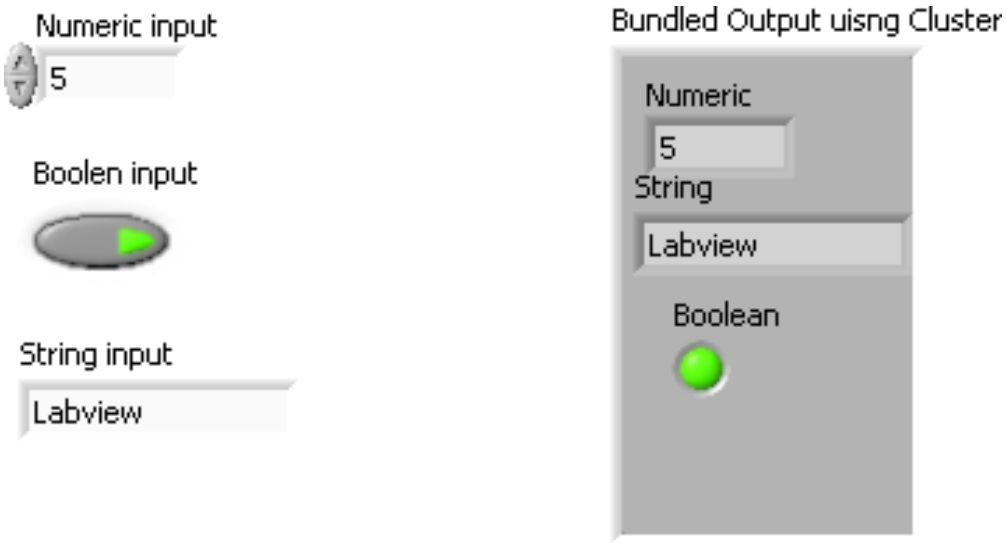
Block diagram panel:

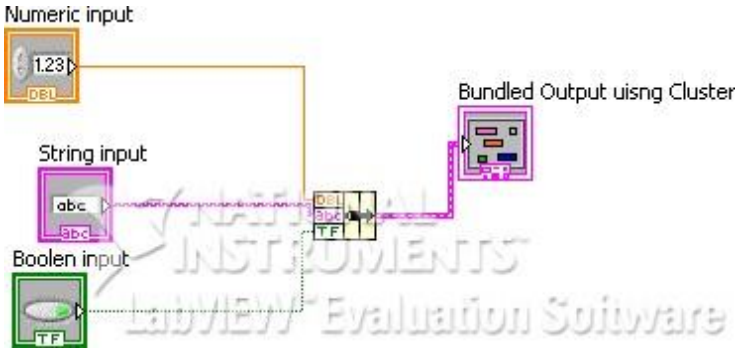


Front panel:

Bundle cluster:

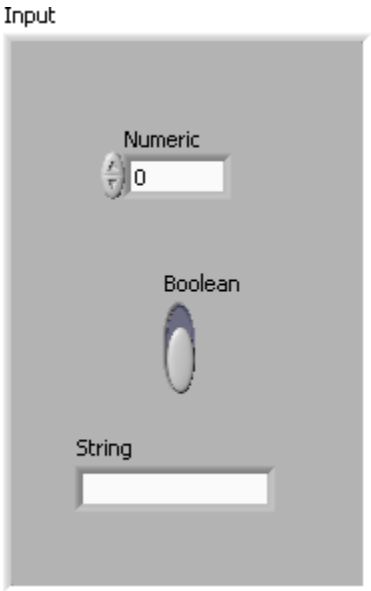
Input:



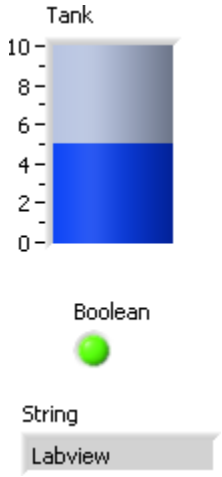


Unbundle cluster:

Input:



Output:



Result: Hence a cluster is bundled and unbundled using Labview.

Exp No: 10

FLAT AND STACKED SEQUENCE

Aim: To perform functions using flat and stacked sequence.

Algorithm:

Step 1: Create blank VI.

Step 2: Create flat and stacked sequence.

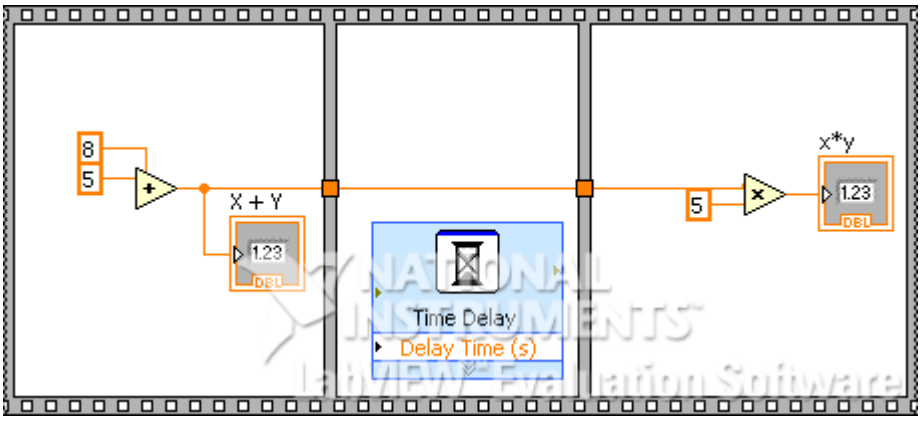
Step 3: Right click on the boundary and add sequence local connect adder with the boundary.

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.

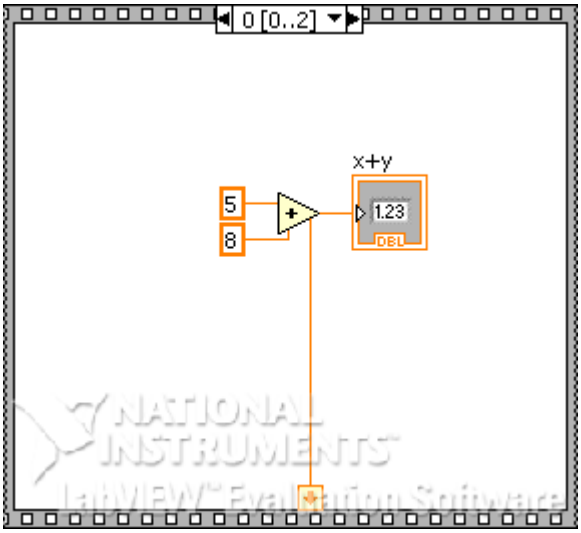
Block diagram panel:

Flat sequence:



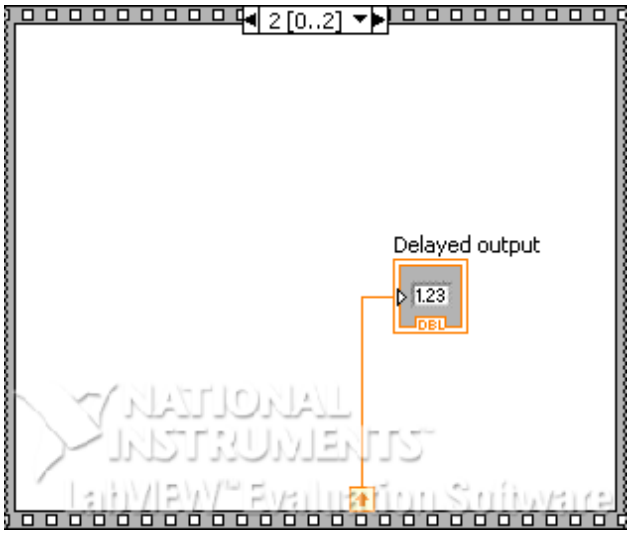
Stacked sequence:

(i)



(ii)

(iii)



Front panel:

Flat sequence:

Result: Thus various functions using flat and stacked sequence using Labview was performed.

Exp No: 11

APPLICATION USING FORMULA NODE

Aim: To create a sine wave using formula node.

Algorithm:

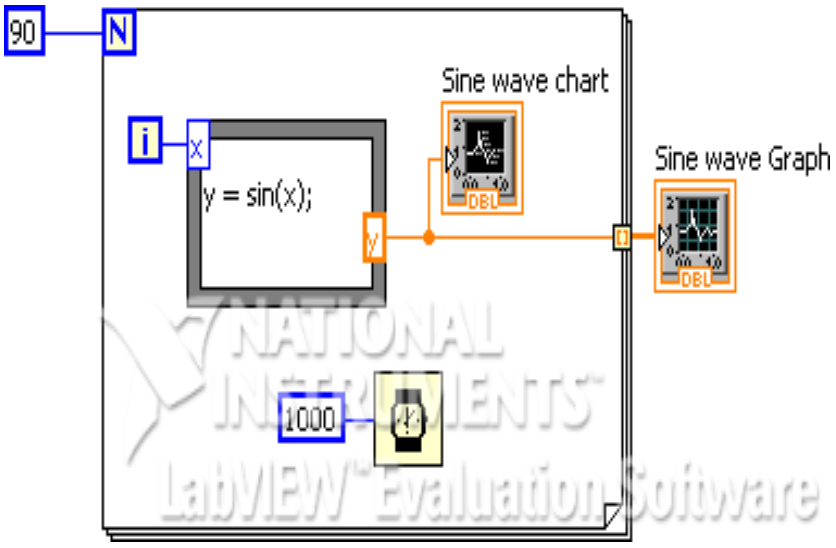
Step 1: Create blank VI.

Step 2: Create formula node, waveform chart and waveform graph in the front panel.

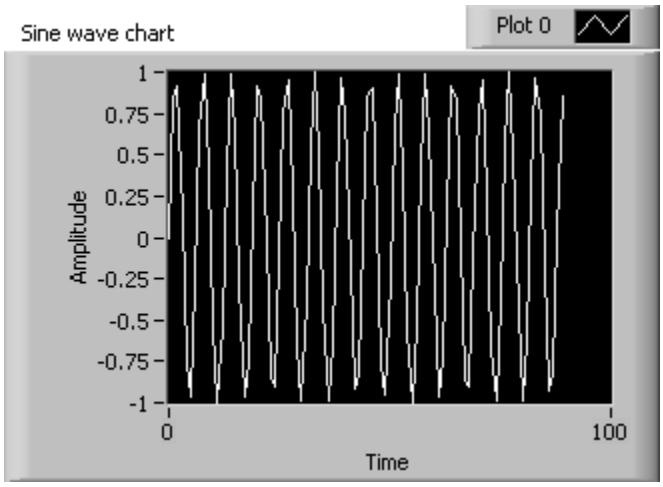
Step 3: Connections are given as per the circuit diagram with necessary wirings.

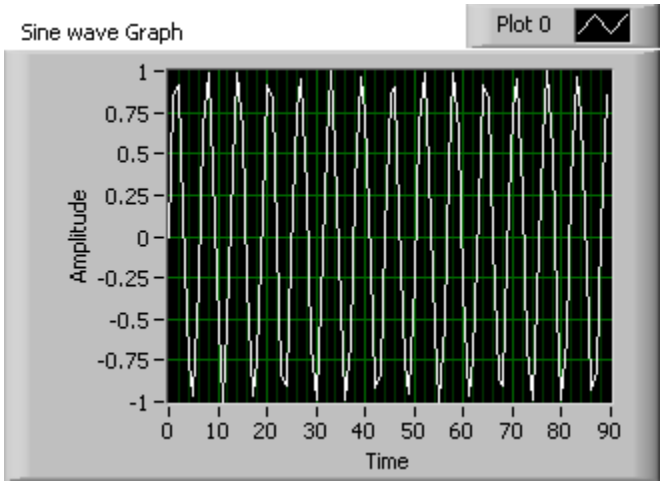
Step 4: Inputs are given in the front panel and the program is executed.

Block diagram panel:



Front panel:





Result: Thus a sine waveform is generated using formula node.

Exp No:12

MEDIAN FILTER

Aim: To apply filtering technique for a given input signal.

Algorithm:

Step 1: Create a blank VI.

Step 2: In the block diagram panel go to signal processing and create a noise signal.

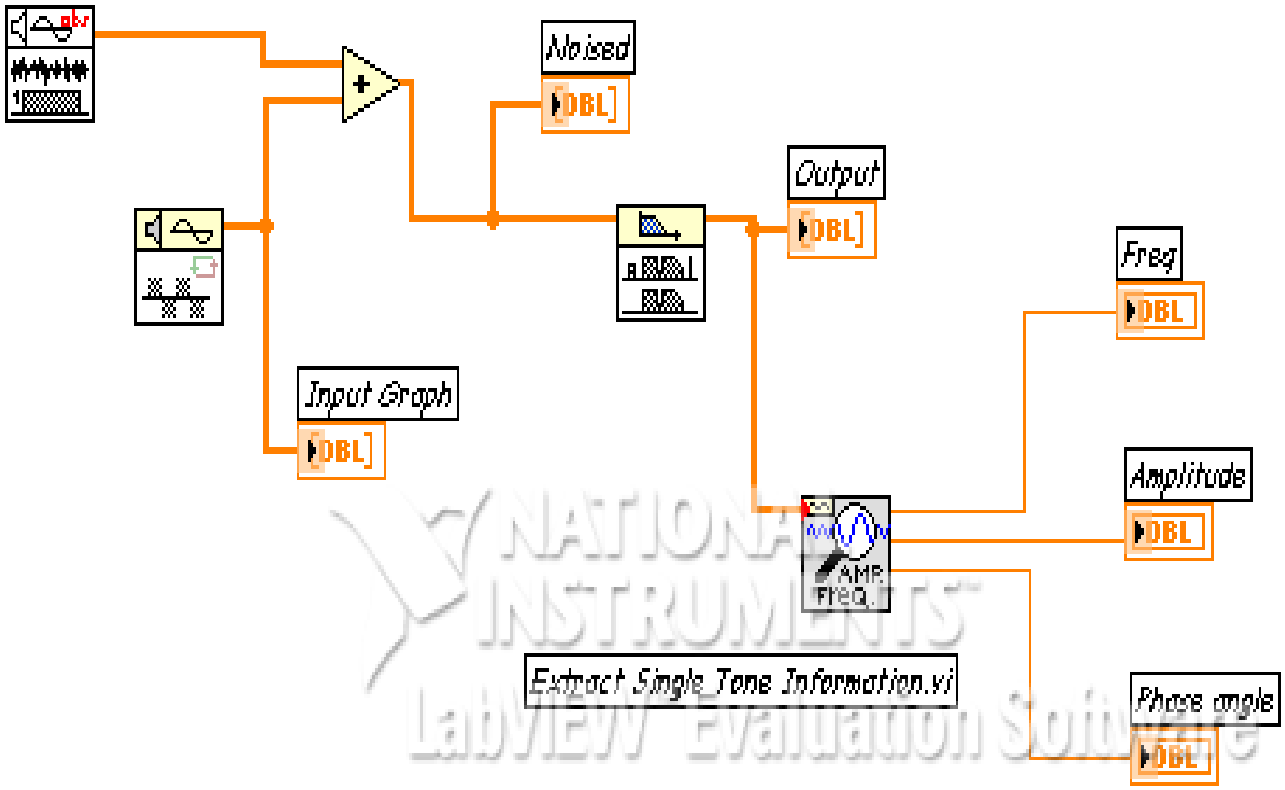
Step 3: Select any one input signal.

Step 4: Create median filter from signal processing block.

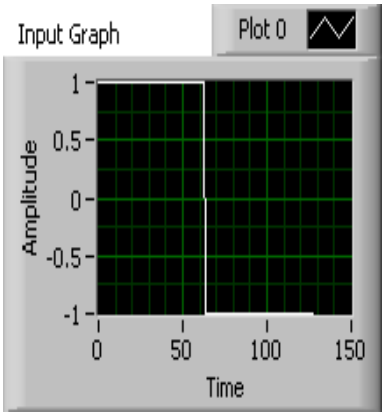
Step 5: Using wiring operations required connections are given in the block diagram.

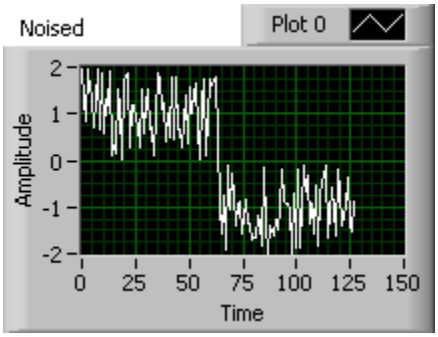
Step 6: Inputs are given in the front panel and the program is executed.

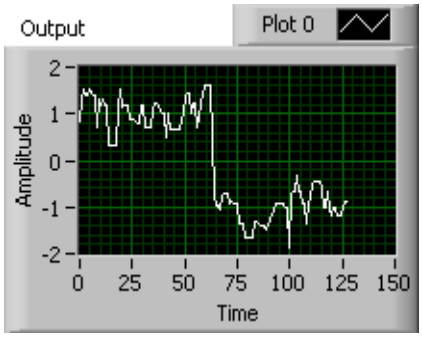
Block diagram panel:



Front panel:







Result: Hence filtering technique is applied on the input signal.

Exp No: 13

DISCRETE COSINE TRANSFORM

Aim: To perform discrete cosine transform on the given signal

Algorithm:

Step 1: Create a blank VI

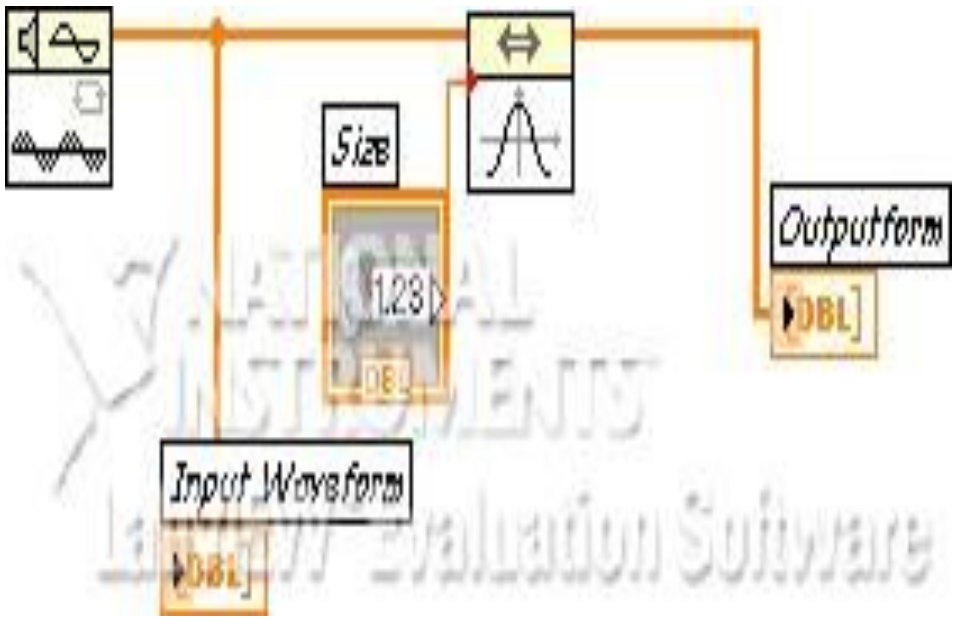
Step 2: Generate a waveform from the signal processing block.

Step 3: Create a discrete cosine transform and waveform graph in the front panel.

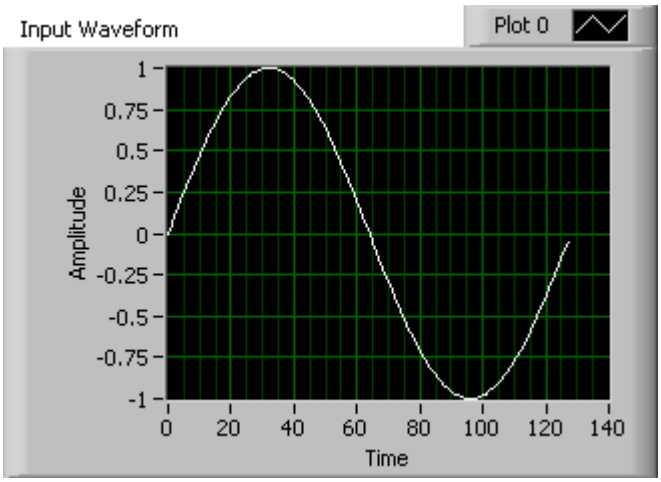
Step 4: Using wiring operations required connections are given in the block diagram.

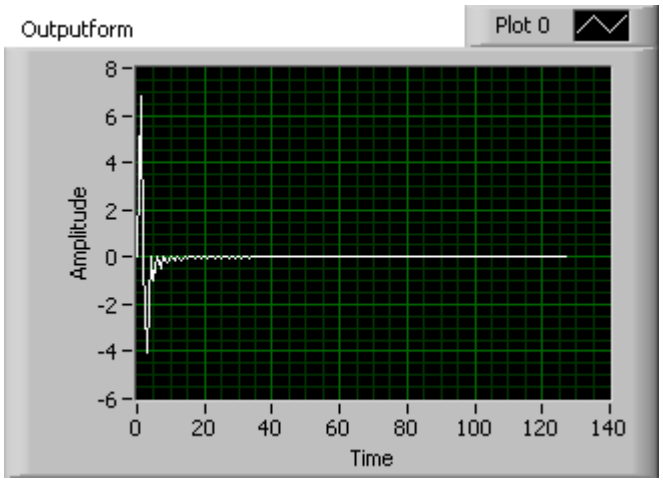
Step 5: Inputs are given in the front panel and the program is executed.

Block diagram panel:



Front panel:





Result: Hence discrete cosine transform was performed on the given input signal.

Exp No: 14

CONVOLUTION OF TWO SIGNALS

Aim: To perform convolution of two signals.

Algorithm:

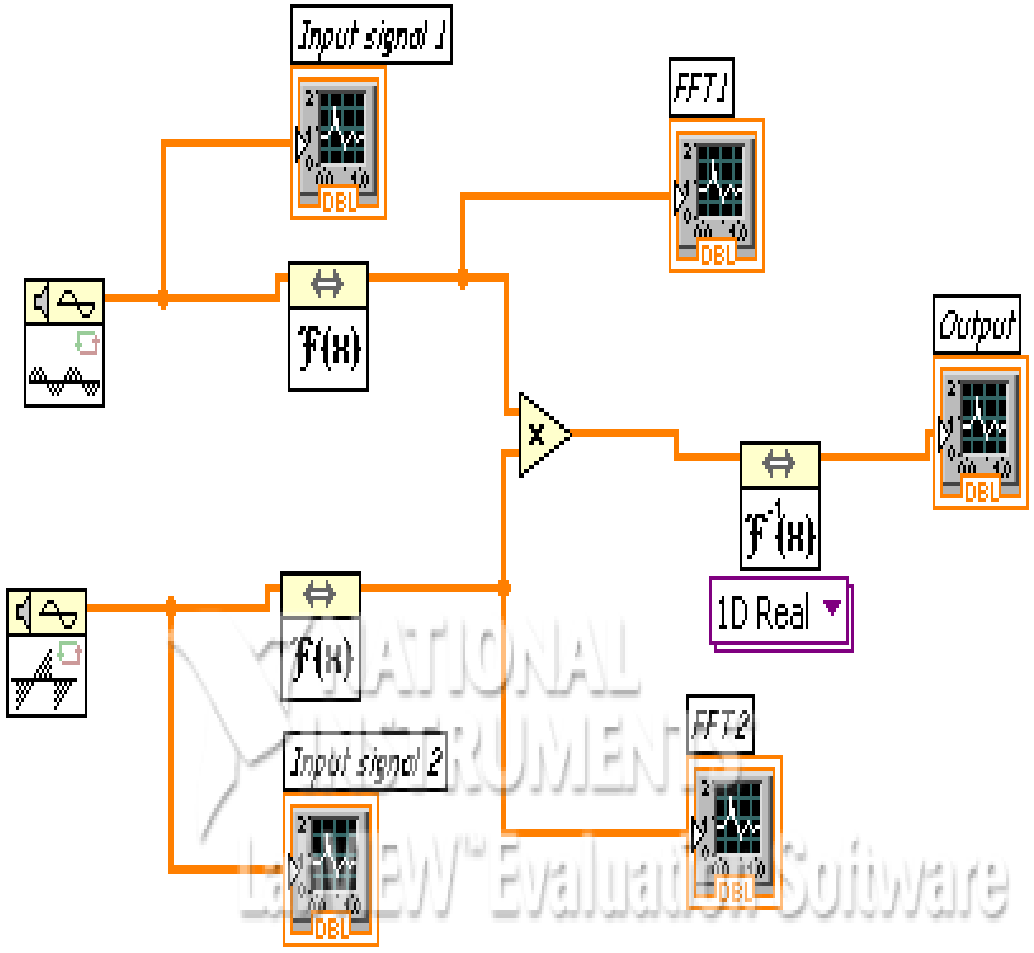
Step 1: Create a blank VI.

Step 2: Create two inputs and waveform graph.

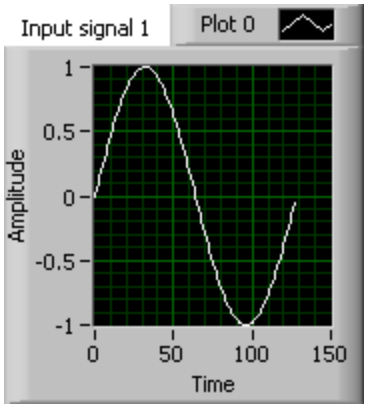
Step 3: Apply FFT for the two inputs and give it to multiplier.

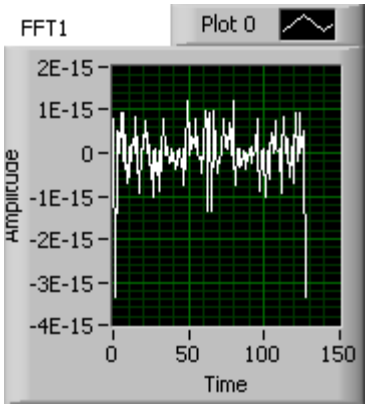
Step 4: In the receiver end IFFT is performed and the convolved output is displayed in the waveform graph.

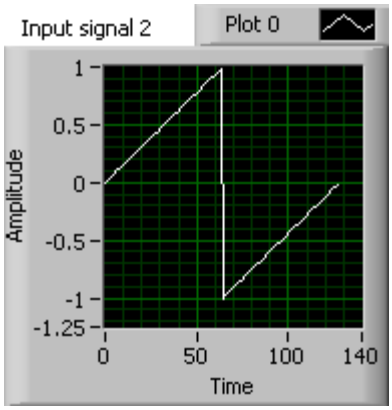
Block diagram panel:

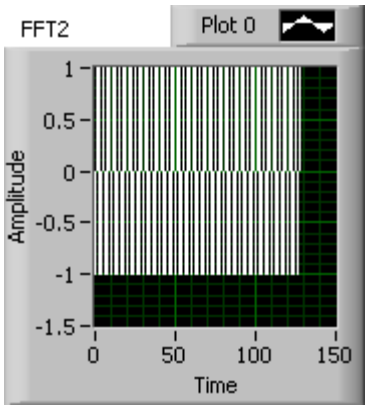


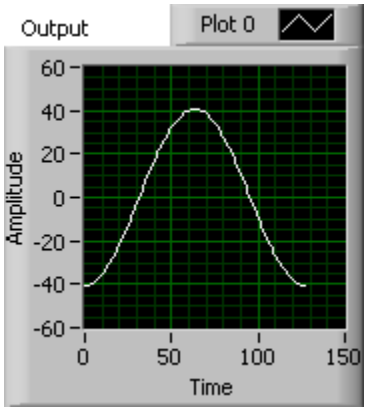
Front panel:


Input signal 1
Plot 0
Amplitude
Time
1
0.5
0
-0.5
-1
0
50
100
150


FFT1
Plot 0
Amplitude
Time
2E-15
1E-15
0
-1E-15
-2E-15
-3E-15
-4E-15
0
50
100
150


Input signal 2
Plot 0
Amplitude
Time
1
0.5
0
-0.5
-1
-1.25
0
50
100
140


FFT2
Plot 0
Amplitude
Time
1
0.5
0
-0.5
-1
-1.5
0
50
100
150



Result: Hence two signals were convolved and the result is verified using Labview.

Exp No: 15

WINDOWING TECHNIQUE

Aim: To apply different windowing technique on the give input signal

Algorithm:

Step 1: Create a blank VI.

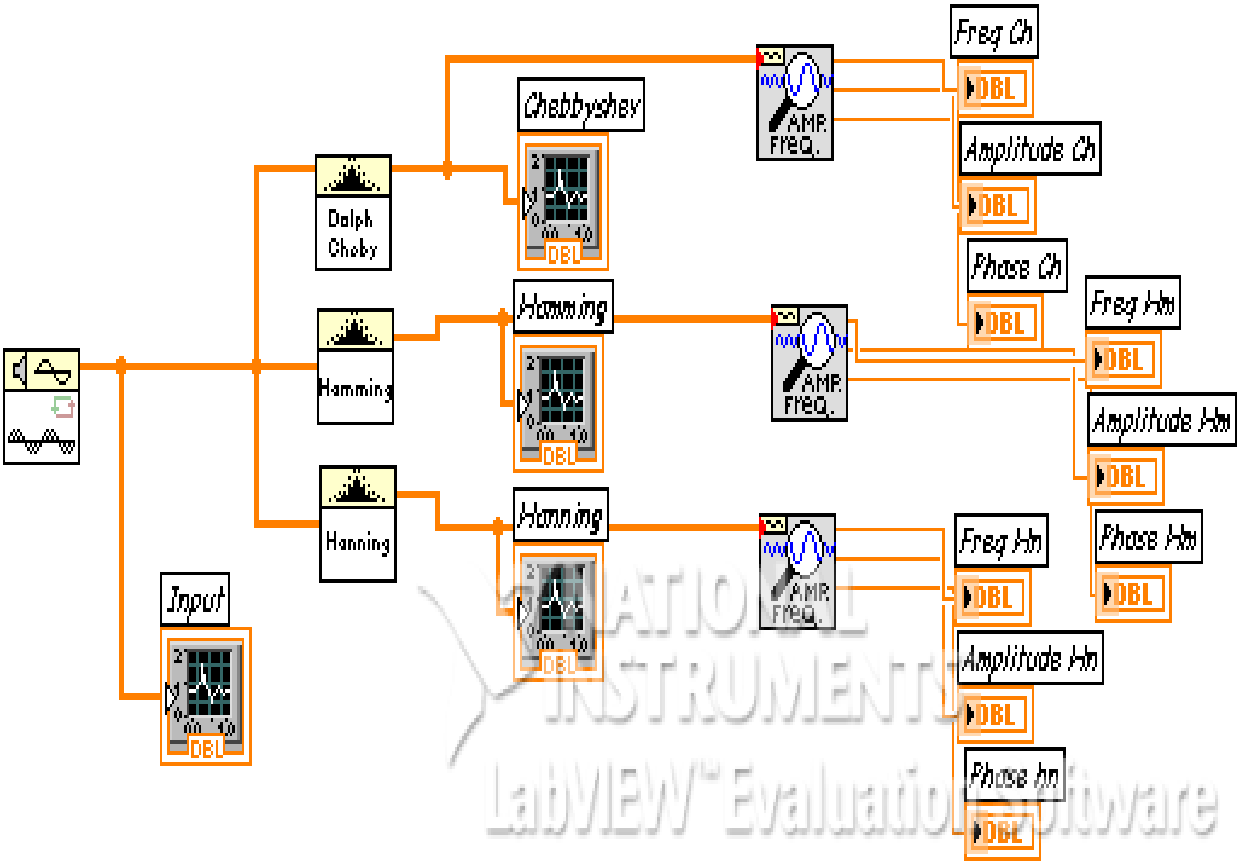
Step 2: Create Hanning, Hamming, Blackman, Chebyshev window on the block diagram panel.

Step 3: Generate an input waveform and create extract single tone for all the windows.

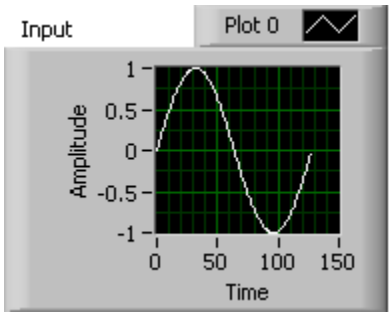
Step 4: Using wiring operations required connections are given in the block diagram.

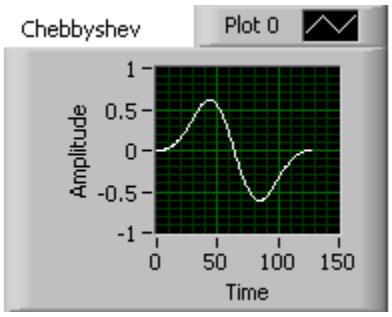
Step 5: Inputs are given in the front panel and the program is executed.

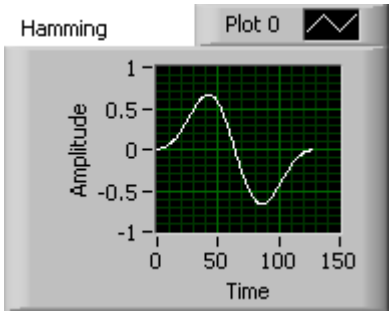
Block diagram panel:

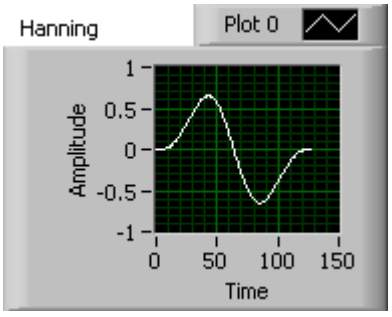


Front panel:









Result: Hence the characteristics of various windows are performed using Labview.