Lab(2)
Faculty of Engineering,
Ain Shams University
MCT-242, Fall 2014

December 7, 2014

Ahmed Okasha
okasha1st@gmail.com

Disclaimer

- Part of this work is based on a Digital Electronics and Computer Interfacing course at the University of Alabama by Tim Mewes.

LAB Goals:

- Performing basic statistical analysis of measured data
- Understanding the use of shift registers
- Performing basic Digital Logic Operations
- Demonstrating different types of signals and how to simulate them.
- Demonstrating Sampling Theorem and Aliasing
- Understanding different types of Filters

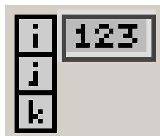
Building Arrays

What is an array?

- An array can either resemble a vector or a matrix. As does a vector and a matrix, an array groups similar pieces of data.
- Arrays may contain numeric, Boolean, string, and cluster data types. They may be used as an indicator (output) or a control (input).

Creating a One-Dimension Array.

In the Front-Panel:

- Controls Palette → Modern → Array, Matrix & Cluster → Array 

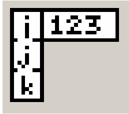
- Notice when you first put the array on the front panel that it is empty. You can determine your array type by inserting either a control or indicator inside the array.
- For example, for a numerical indicator array:

  Controls Palette → Num Inds → Num Ind → Place inside Array.

Creating a One-Dimension Array.

In the Block-Diagram (Creating Constant Array):

- Functions Palette → Programming → Array → Array Constant
- To make the array a numerical constant array:



  - Functions Palette → Mathematics → Numeric → Numeric Constant
  - Drag and drop it to the Array constant box

OR:

- Functions Palette → Programming → Array → Build Array
- Functions Palette → Mathematics → Numeric → Numeric Constant. And connect the numerical constant to the build Array box.

Exercise (1)

- Implement a VI that calculates the mean, the standard deviation for the given measurement set.

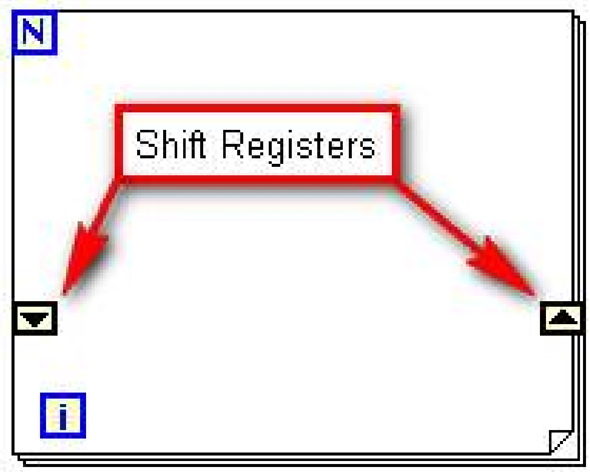
Trial	1	2	3	4	5	6	7	8	9	10
Current (mA)	21.5	22.1	21.3	21.7	22	22.2	21.8	21.4	21.9	22.1

Exercise (2)

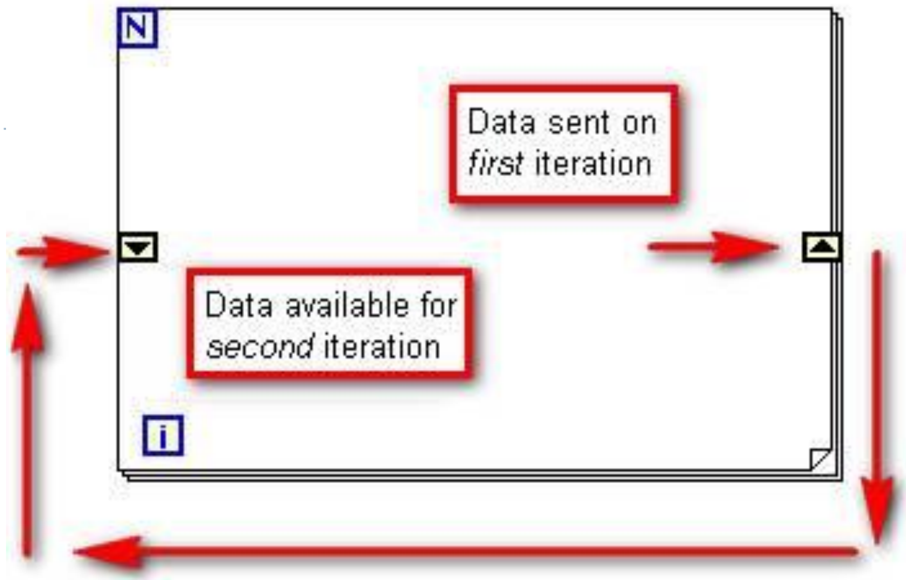
- Implement a VI that finds the best-fit straight line for the given calibration data (decreasing direction).

Input (x)	Output (y) (decreasing direction)
0	0
5	1.2
10	3.5
15	3
20	5
25	5.5
30	7
35	7.7
40	9

Shift Registers



- Selected via right-clicking the frame.
- Enables the result of an iteration to be passed to the next iteration.
- Can be used for any data type



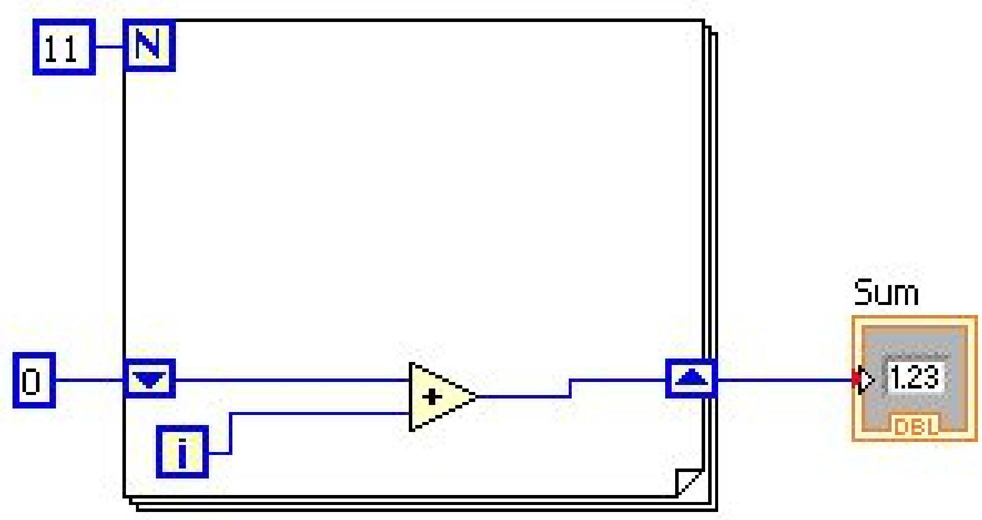
- Data comes into the loop by way of the shift register on the left side of the for loop and is passed to the next iteration of the loop through the shift register on the right side of the for loop.
- The initial value is set by wiring to the left terminal and the final iterations value is output at the right terminal.

Example

- C++ Code:

```
int i, n, sum;
n=11;
sum=0;
for (i=0; i<n; i++)
{
sum = sum + i;
}
cout << sum << endl;
```

LabVIEW VI:



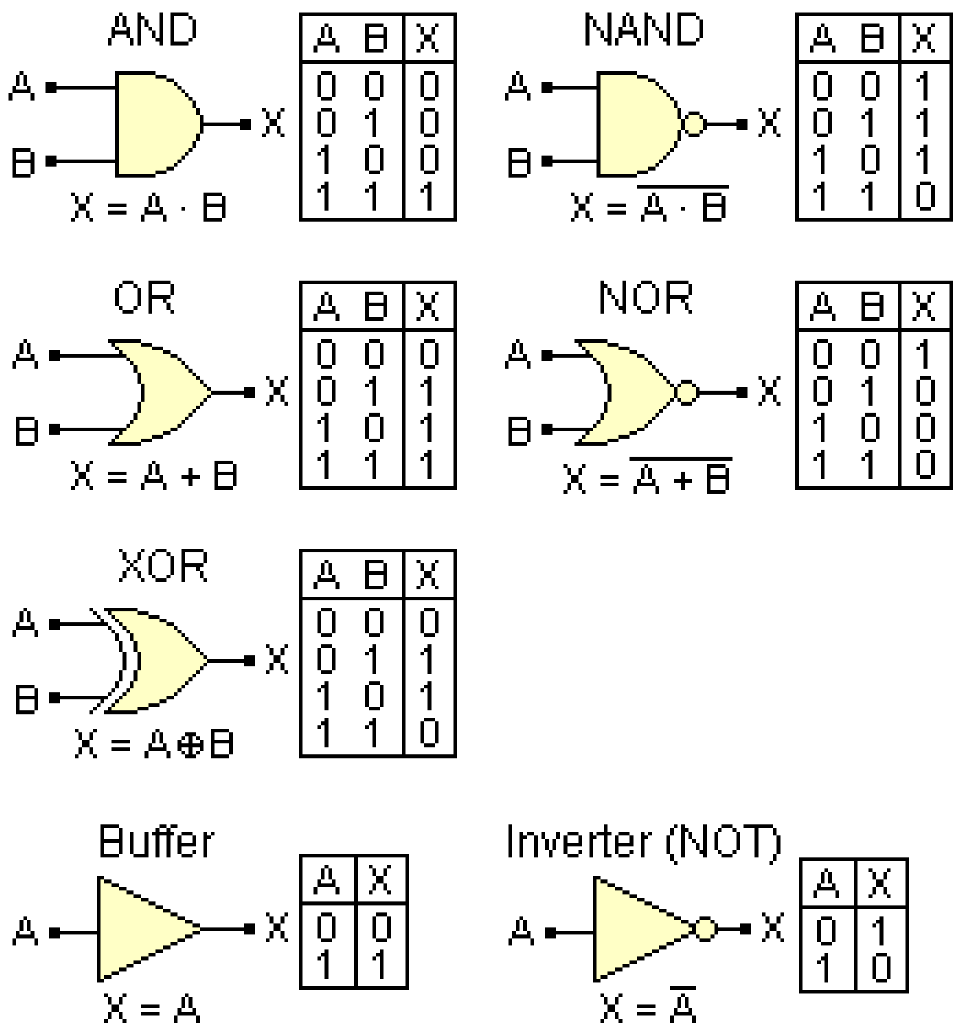
Exercise (3)

- Design a VI that generates a random number (0-100). Use shift registers with For Loop (20 iterations) to get the maximum of the generated 20 random numbers.

Note:

- Add Indicators for both the random number and the maximum number. Also add a delay (Wait function).
- You will have to use select function (Functions Palette >> Comparison>>Select)

Digital Logic Operations



AND

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

NAND

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

OR

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

NOR

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

XOR

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

Buffer

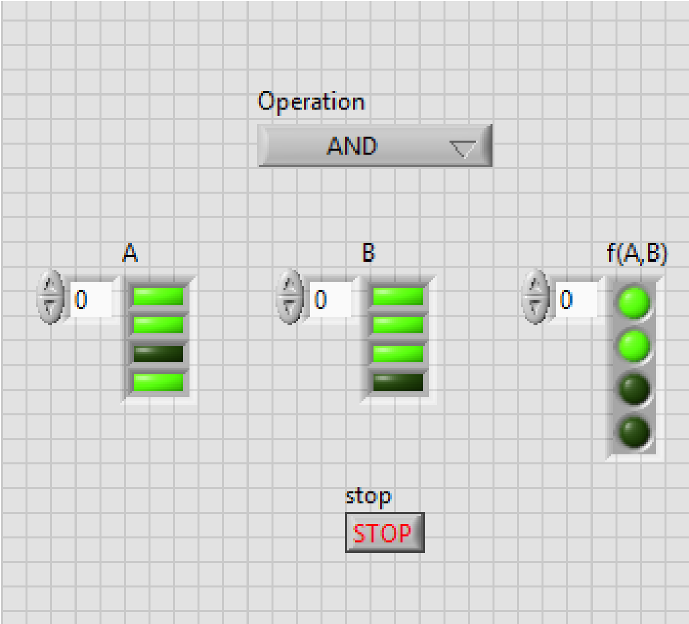
A	X
0	0
1	1

Inverter (NOT)

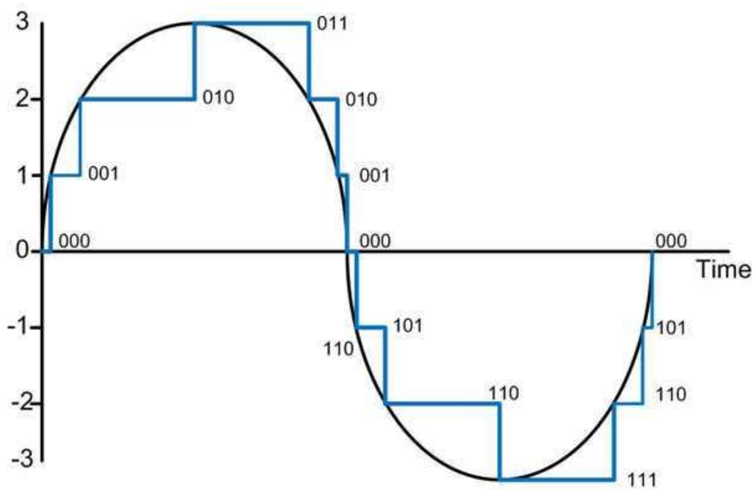
A	X
0	1
1	0

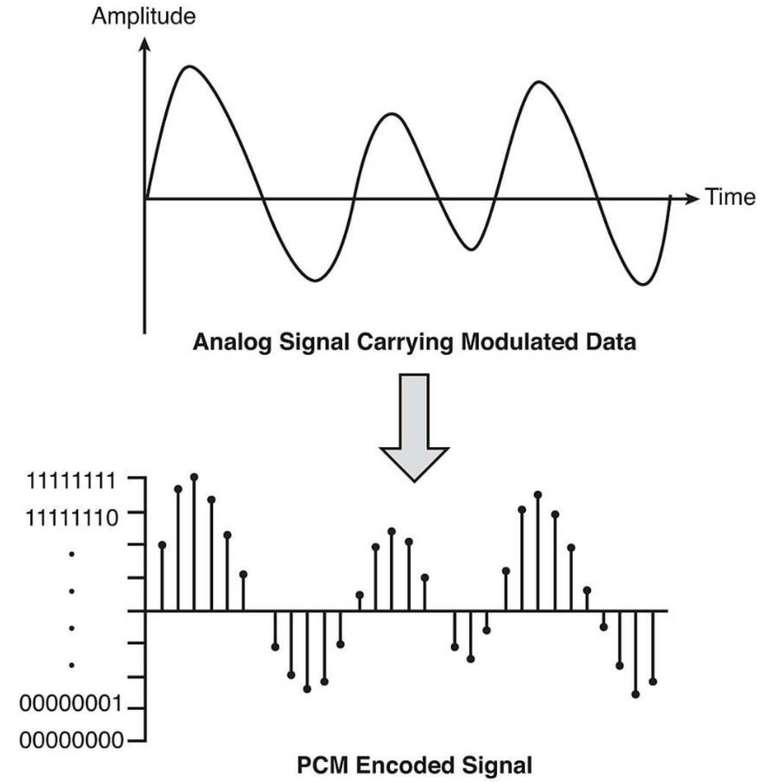
Exercise (4)

- Implement a VI that consists of two arrays of led controls (Every array consists of 4 leds). This VI is intended to calculate the appropriate Boolean operation based on the user input, and display the result in an array of led indicators. The required operations are:
  - AND
  - OR
  - XOR
  - NAND
  - NOR



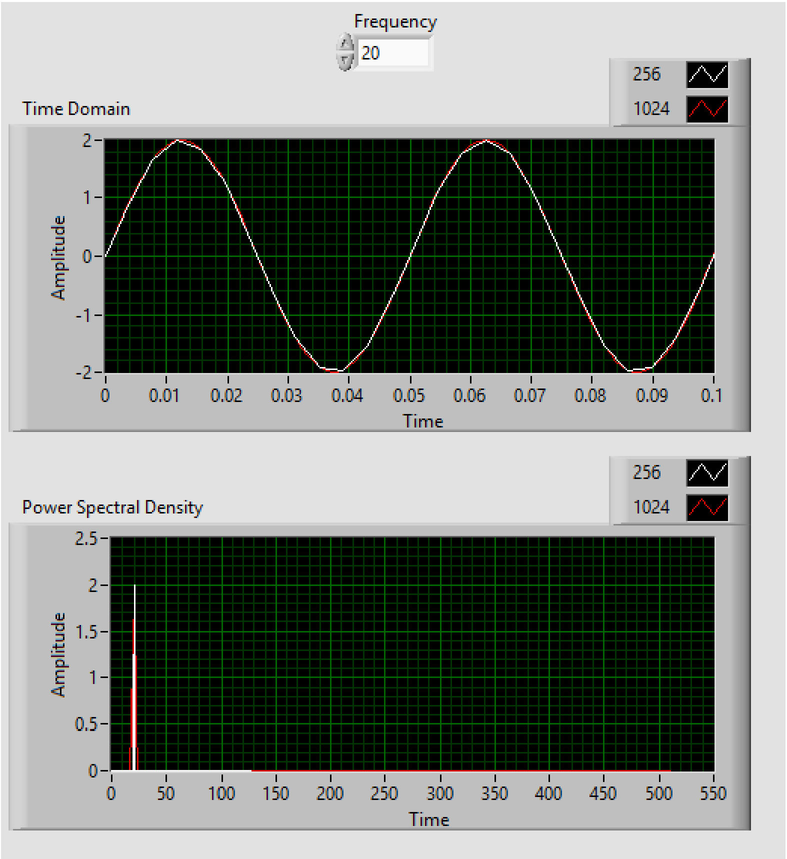
Signals

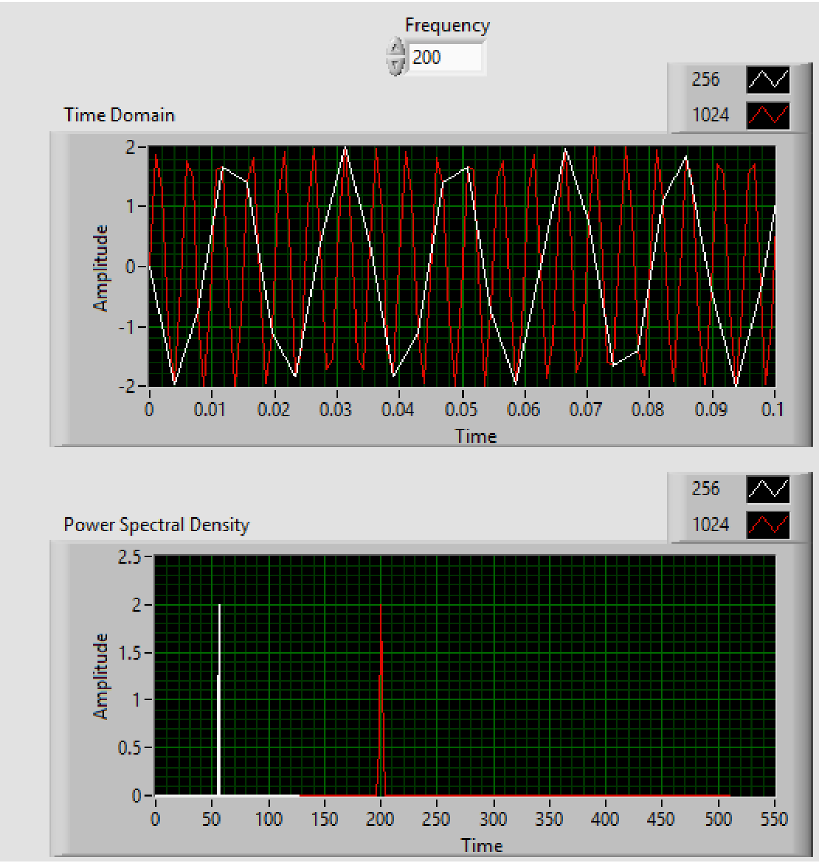

3
2
1
0
-1
-2
-3
000
001
010
011
010
001
000
101
110
110
111
110
101
000
Time


Amplitude
Time
Analog Signal Carrying Modulated Data
11111111
11111110
00000001
00000000
PCM Encoded Signal

Aliasing

- If the signal not sampled fast enough a problem known as aliasing will occur.



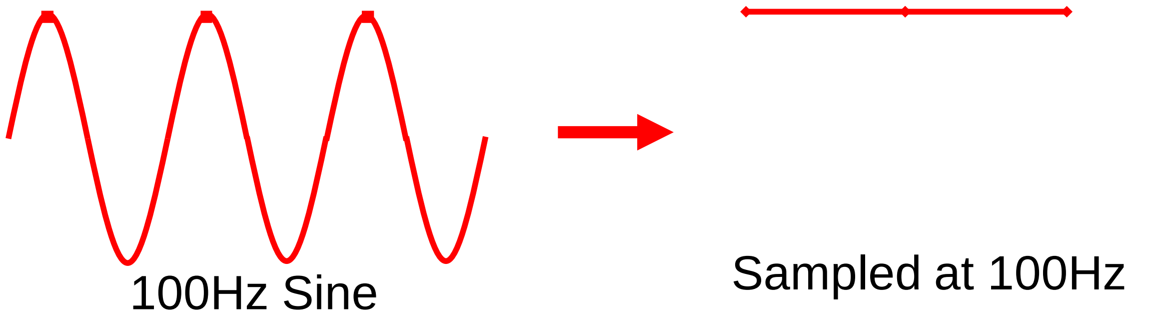
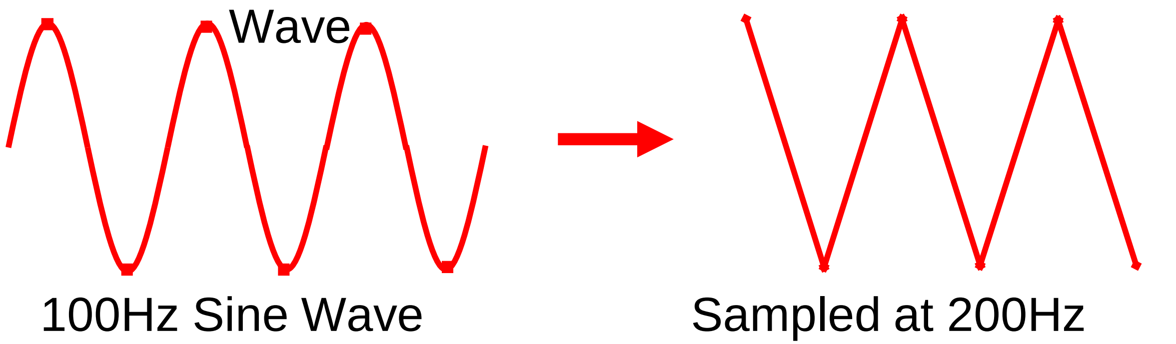
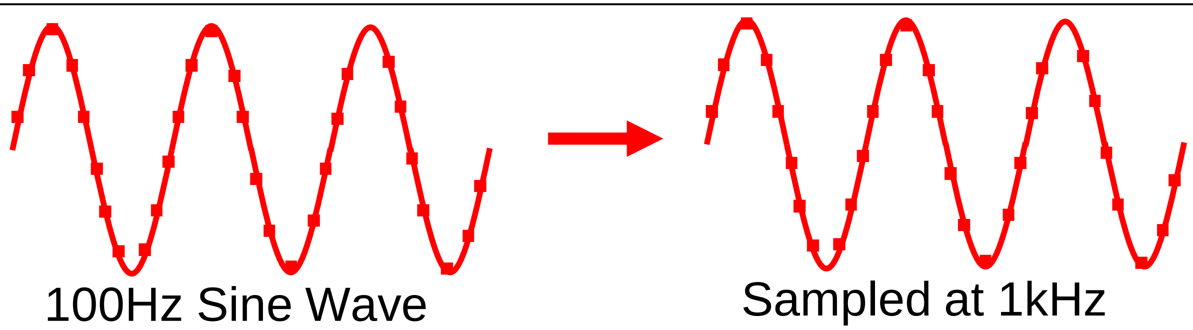


Nyquist Theorem

- You must sample at greater than 2 times the maximum frequency component of your signal to accurately represent the FREQUENCY of your signal.

- NOTE: You must sample between 5 - 10 times greater than the maximum frequency component of your signal to accurately represent the SHAPE of your signal.

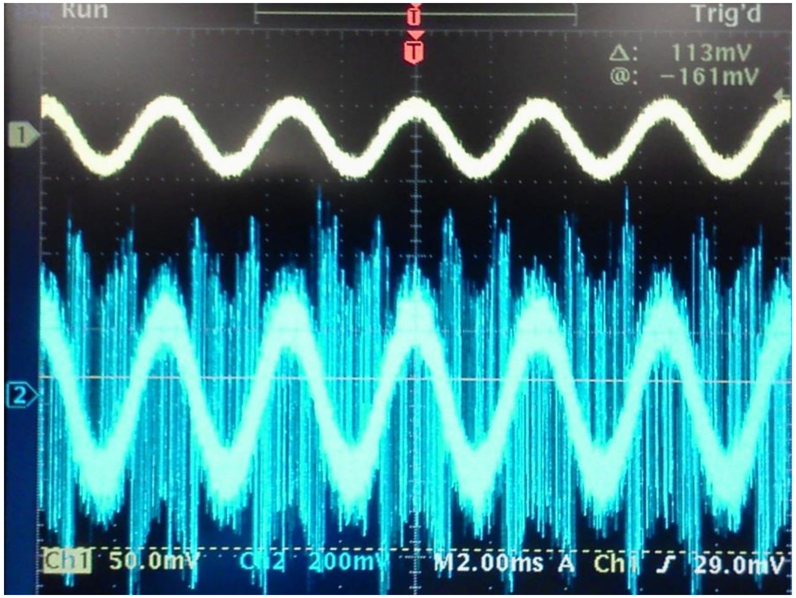
Nyquist Theorem

 100Hz Sine Wave → Sampled at 100Hz	Aliased Signal
 100Hz Sine Wave → Sampled at 200Hz	Adequately Sampled for Frequency Only (Same # of cycles)
 100Hz Sine Wave → Sampled at 1kHz	Adequately Sampled for Frequency and Shape

Filters

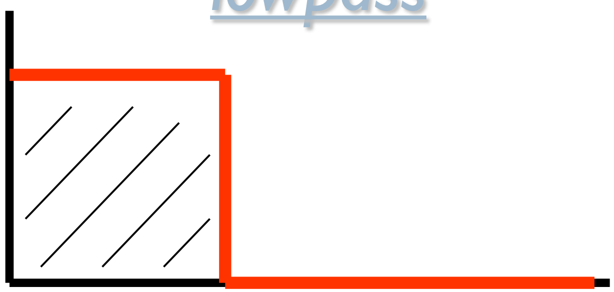
What is a Filter?

- A *filter* is a system/process that processes a signal in some desired fashion.
- Filtering is a frequency selective operation, in which some frequency components are suppressed and some others are passed.

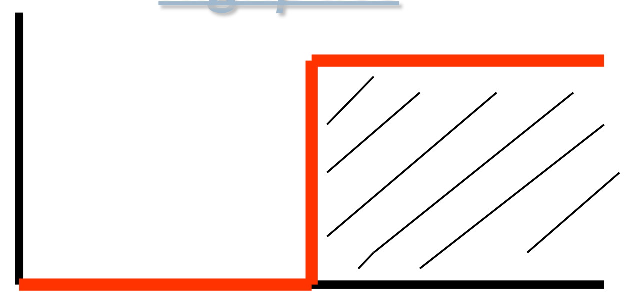


Background: Four types of filters - "Ideal"

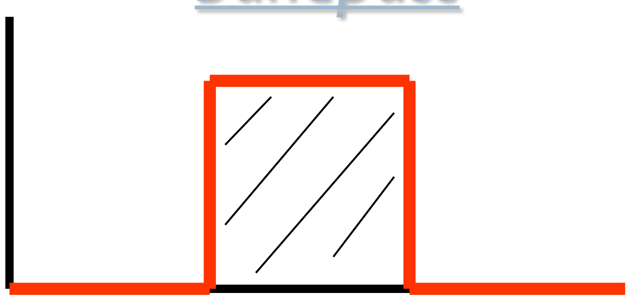
lowpass



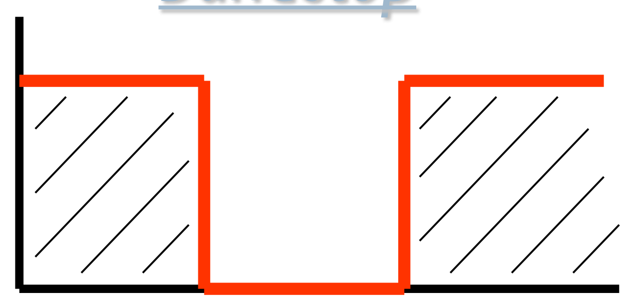
highpass



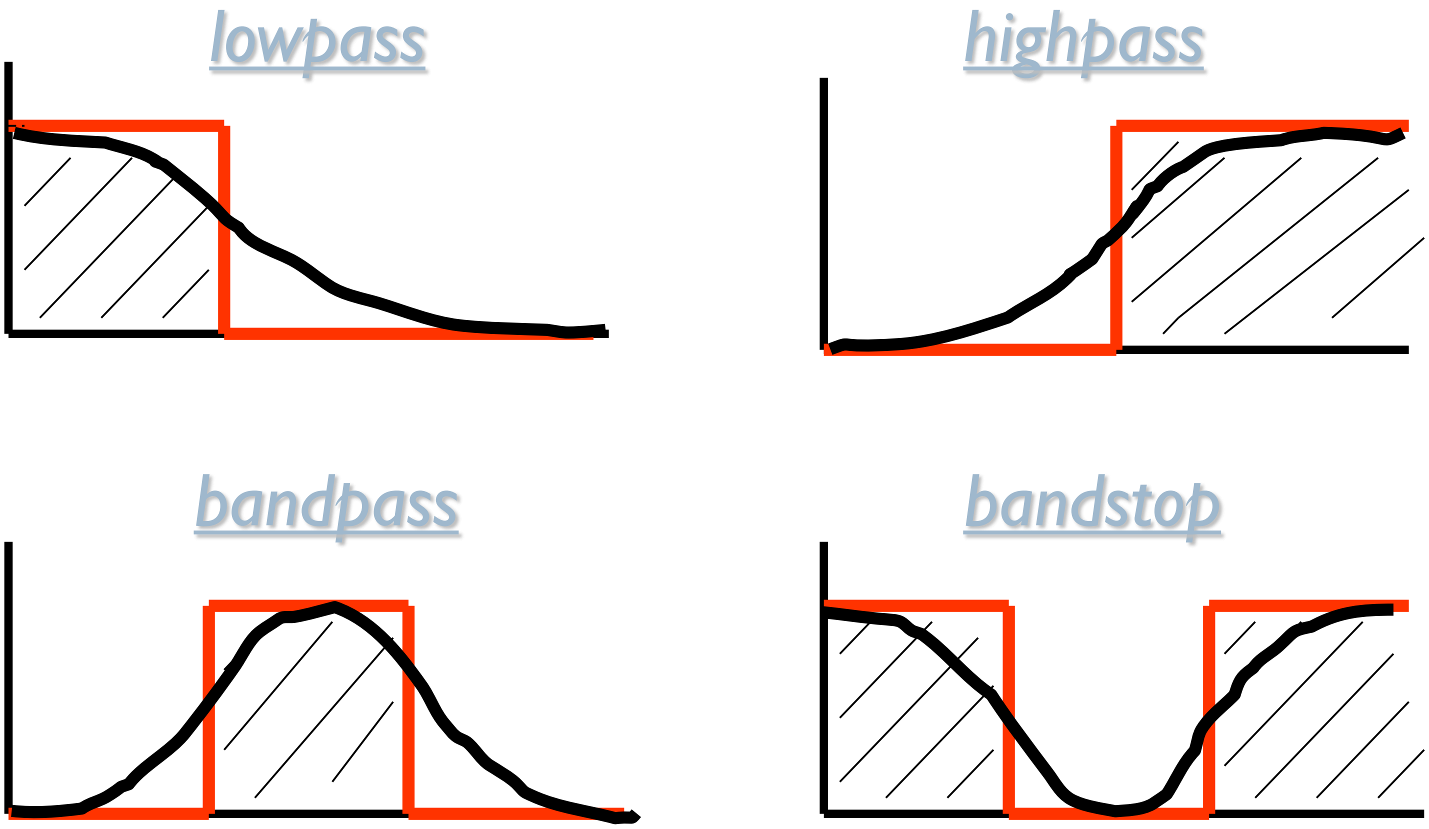
bandpass



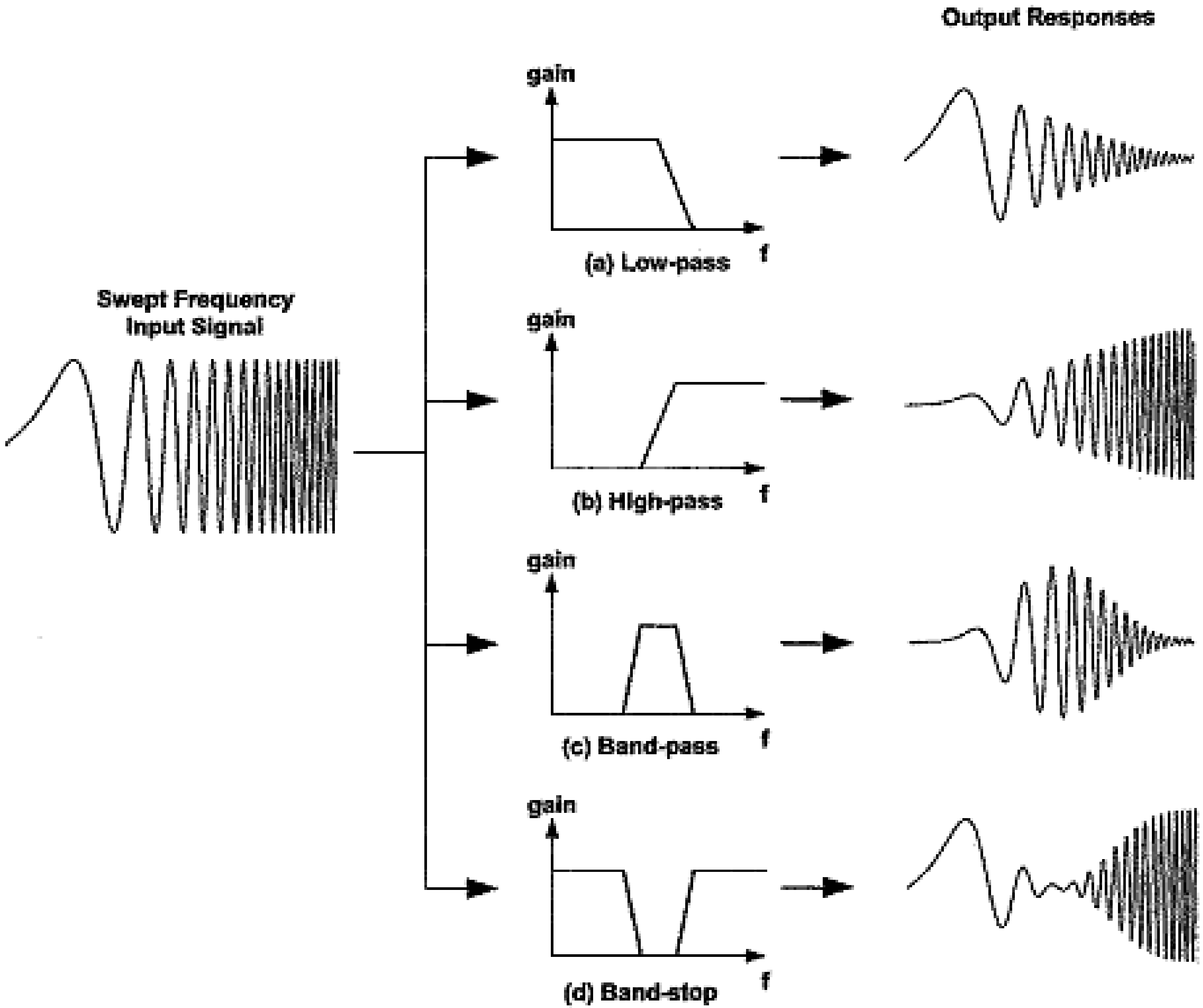
bandstop



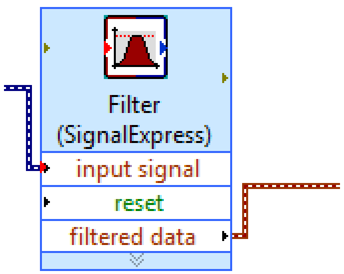
Background: Realistic Filters:

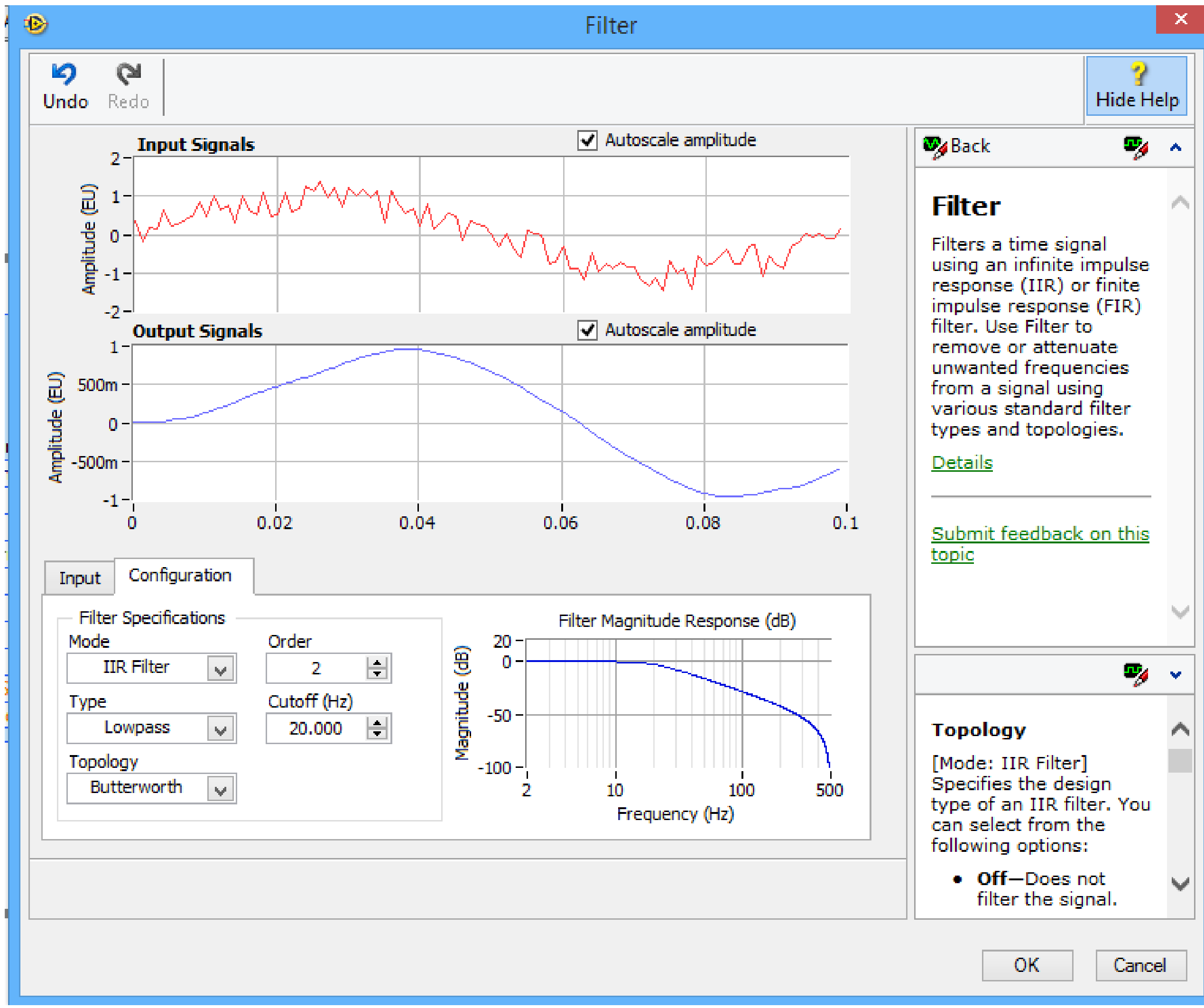


Example


Output Responses
Swept Frequency
Input Signal
gain
f
(a) Low-pass
gain
f
(b) High-pass
gain
f
(c) Band-pass
gain
f
(d) Band-stop

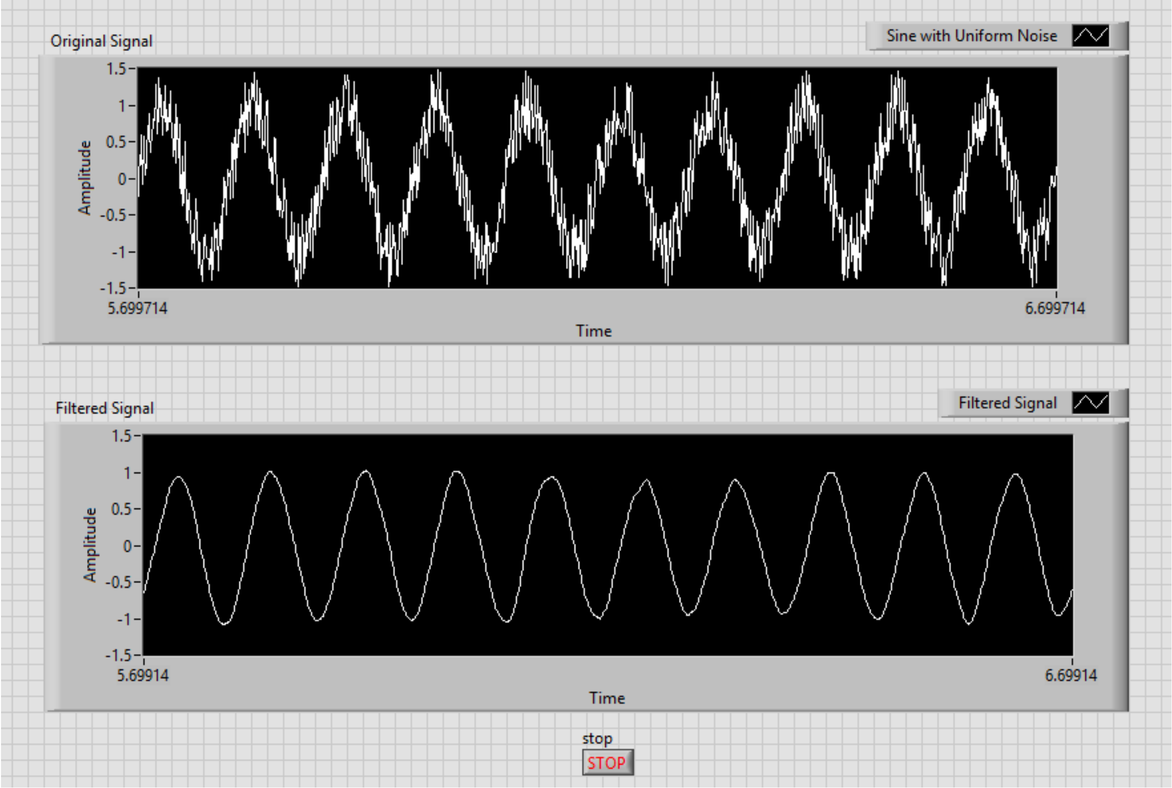
Filters in LabVIEW





Exercise (5)

- Design a VI that generates simulated sine-wave signal, add noise to it and use a filter to eliminate this noise.



Exercise (6)

Design a VI that generates three simulated sine-wave signals with user-controlled amplitude and frequency Knob. The VI adds the three sine wave signals and perform a filter operation. Show both signals (Original and the filtered one) on a single waveform chart

Notes:

- Vary the signals frequencies and also the filter type (Low Pass - High Pass- Band Pass) to see the effect of these variations on the filtered signal.
- Vary the filter characteristics such as order and cut off frequency, and observe the filtered signal.

Exercise (6) Front Panel