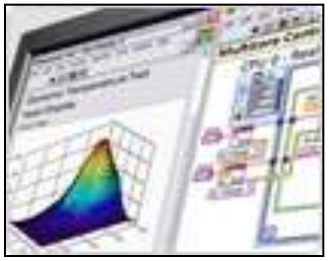
LabVIEW Online Training

Presenter:
Anders Rohde
Applications Engineer,
National Instruments DK

Webcast Structure

15:00-15:45	**NI LabVIEW Training Part 1**
15:45-16:15	Question and Answer Chat Session
16:15-16:45	Break
16:45-17:30	**NI LabVIEW Training part 2**
17:30-18:00	Question and Answer Chat Session

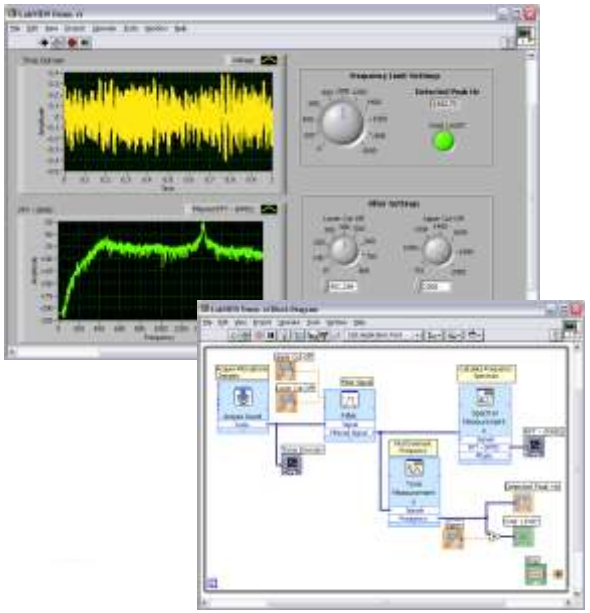
NI myDAQ

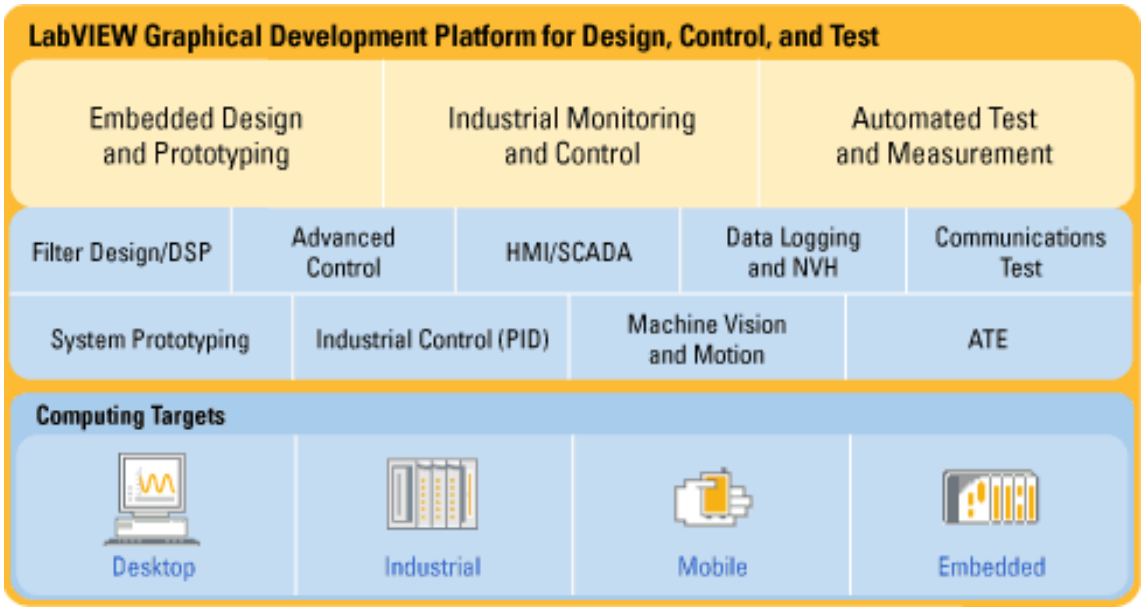


NI LabVIEW Student Edition*

LabVIEW Graphical Development System

- Graphical Programming Environment
- Compile Code for Multiple OSs and Devices
- Useful in a Broad Range of Applications





Open and Run LabVIEW

Start»All Programs»National Instruments»LabVIEW 2011

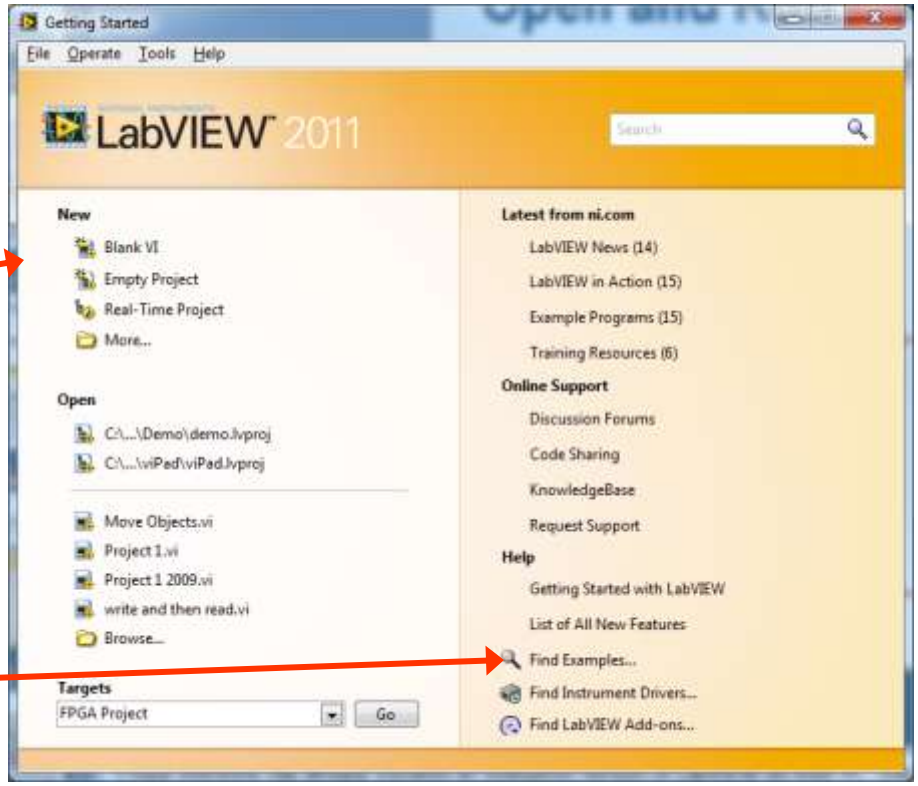

National Instruments LabVIEW

Start From a Blank VI:

New»Blank VI

Start From an Example:

Examples»Find Examples…

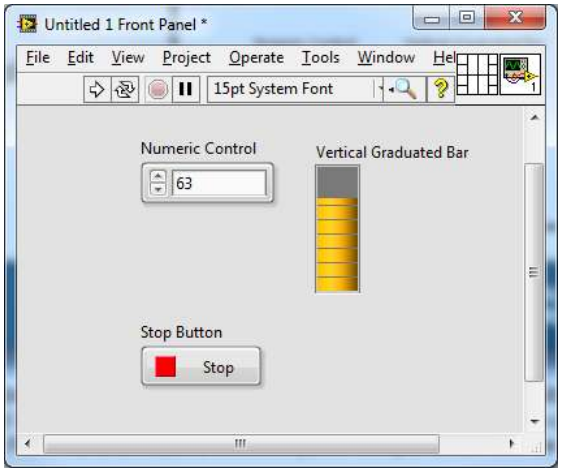


LabVIEW Programs Are Called Virtual Instruments (VIs)

Each VI Has Two Windows

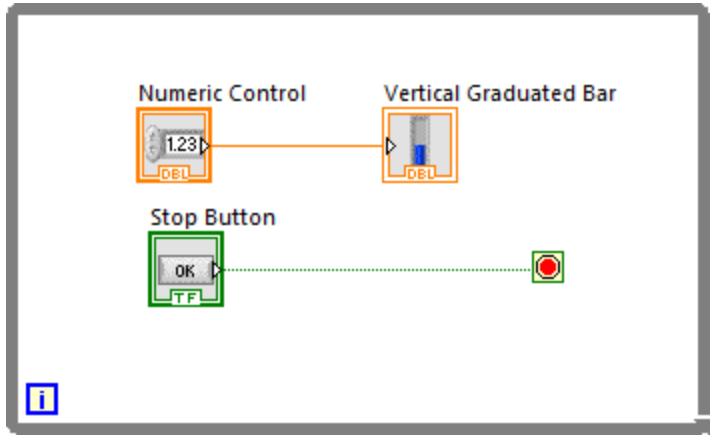
Front Panel

- User Interface (UI)
  - Controls = Inputs
  - Indicators = Outputs

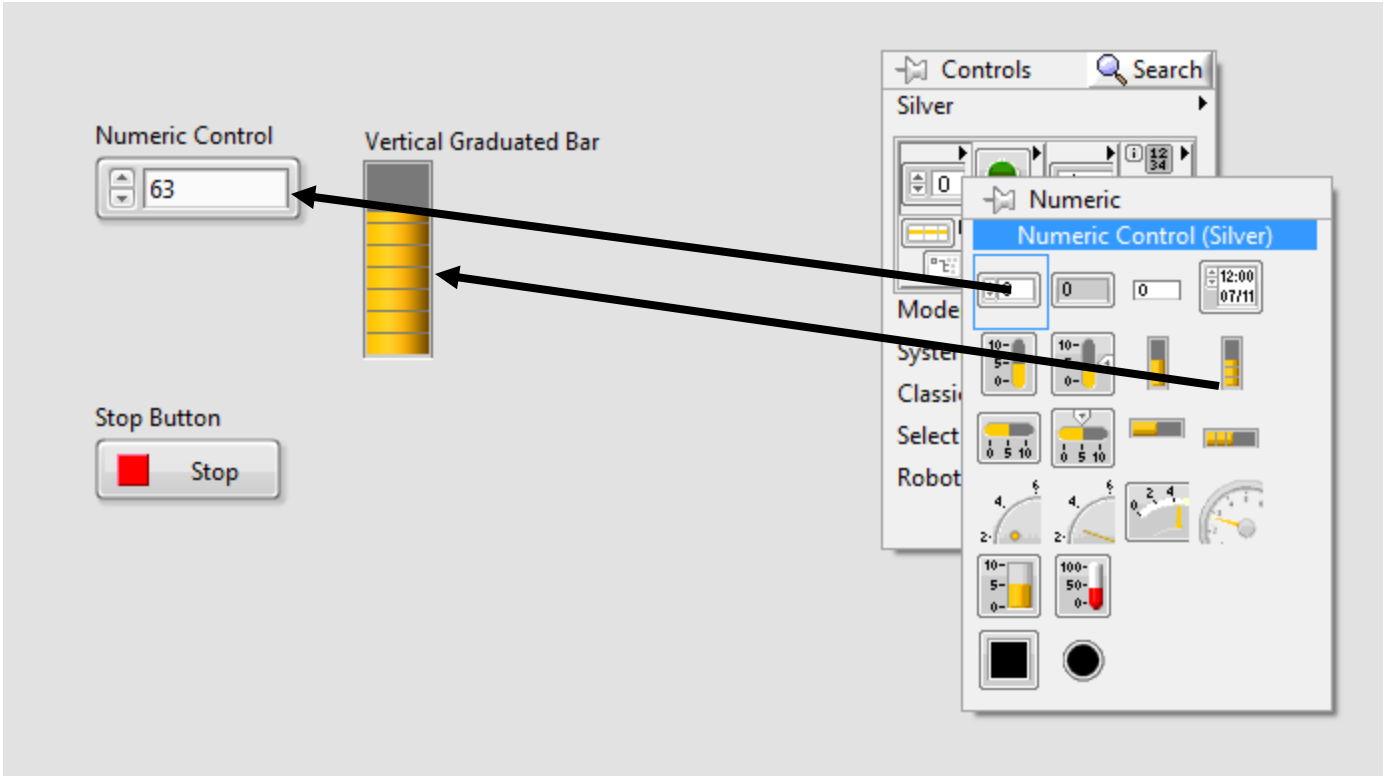


Block Diagram

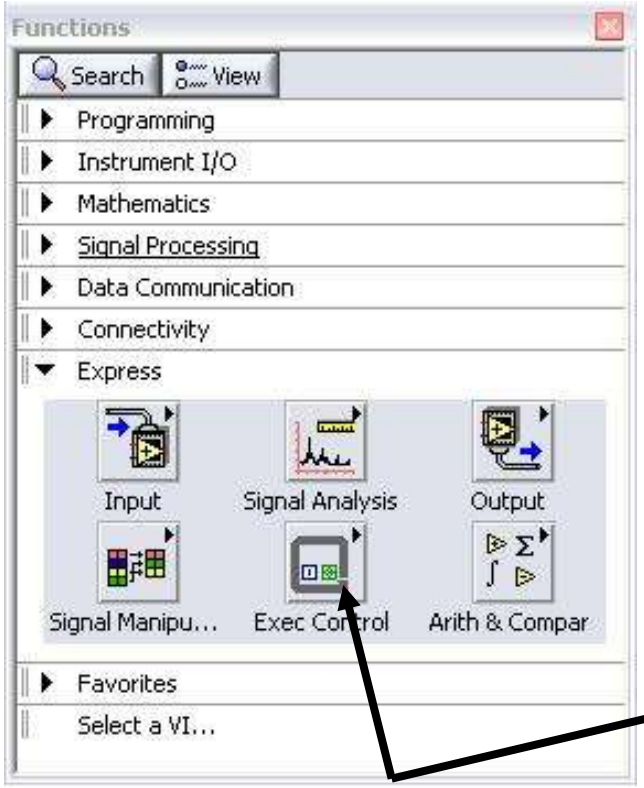
- Graphical Code
  - Data travels along wires from controls through functions to indicators
  - Blocks execute by data flow



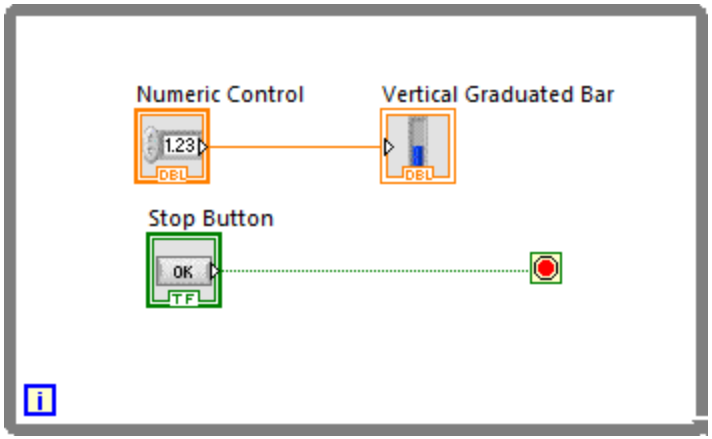
Controls Palette (Controls and Indicators)



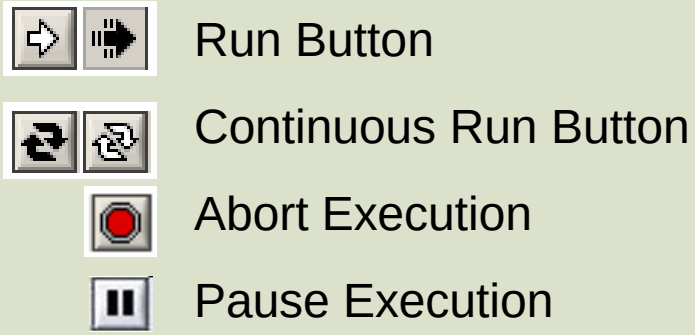
Functions (and Structures) Palette



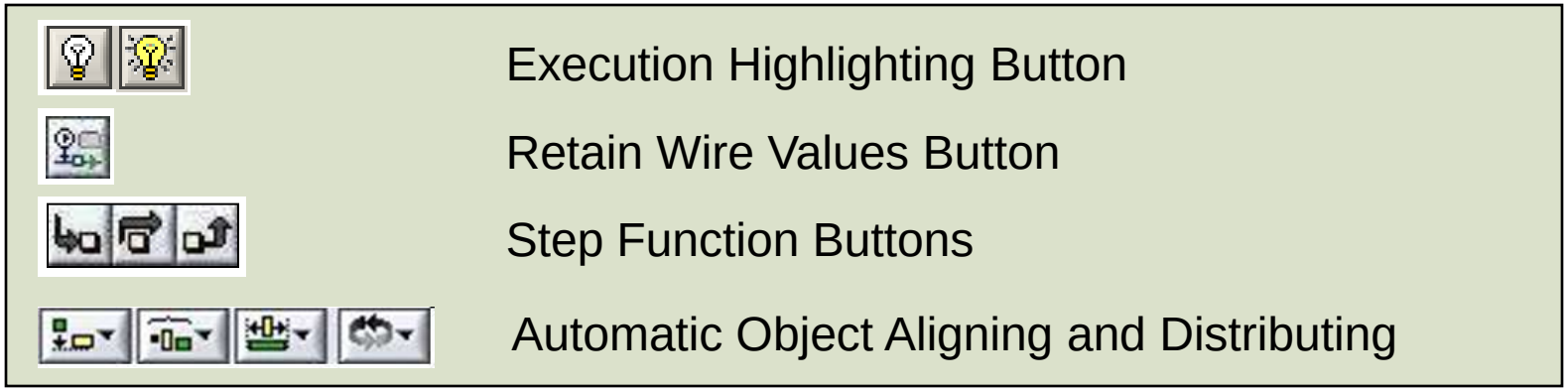
Structure:
While Loop



Status Toolbar

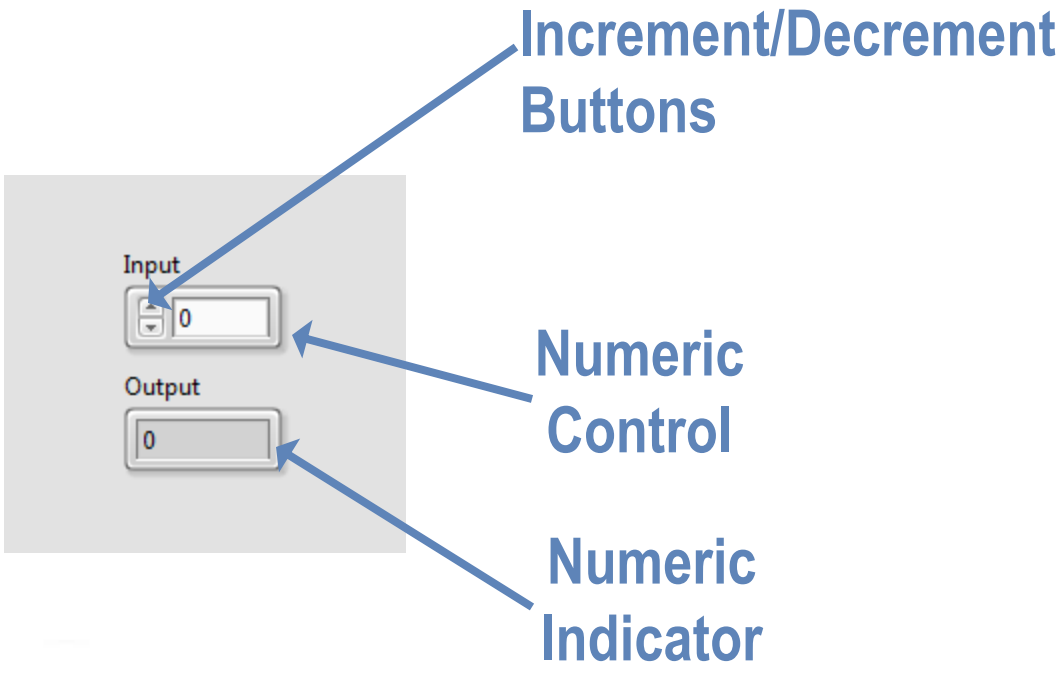


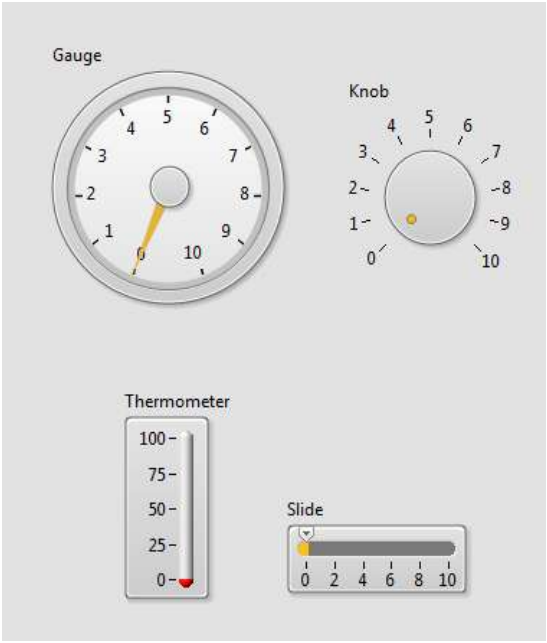
Additional Buttons on the Diagram Toolbar



Front Panel – Numeric Controls/Indicators

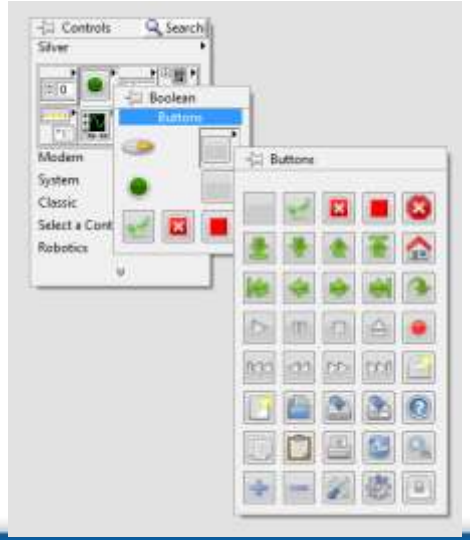
The numeric data type can represent numbers of various types, such as integer or real





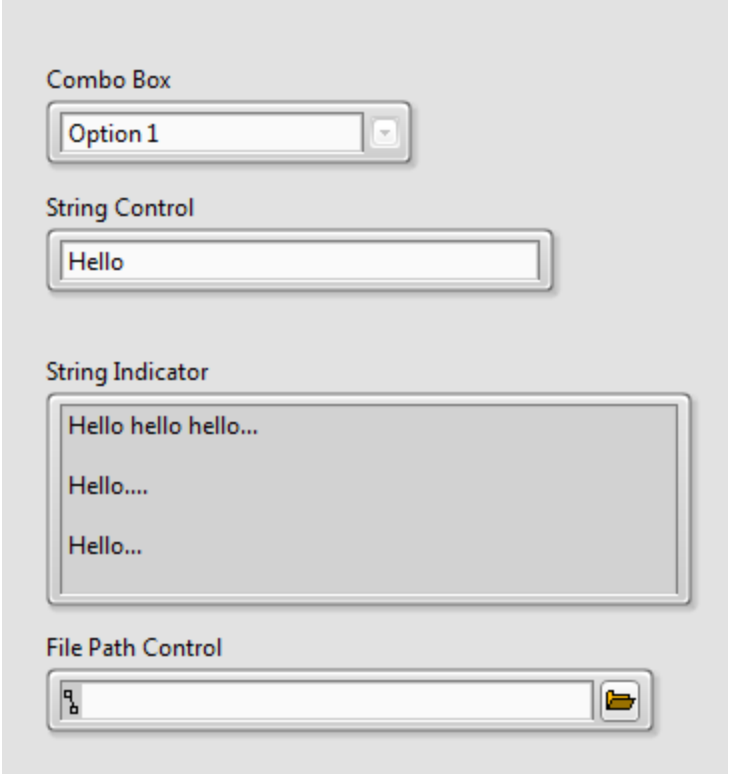
Front Panel – Boolean Controls/Indicators

- The Boolean data type represents data that has only two parts, such as True and False or On and Off
- Use Boolean controls and indicators to enter and display Boolean (True or False) values
- Boolean objects simulate switches, push buttons, and LEDs



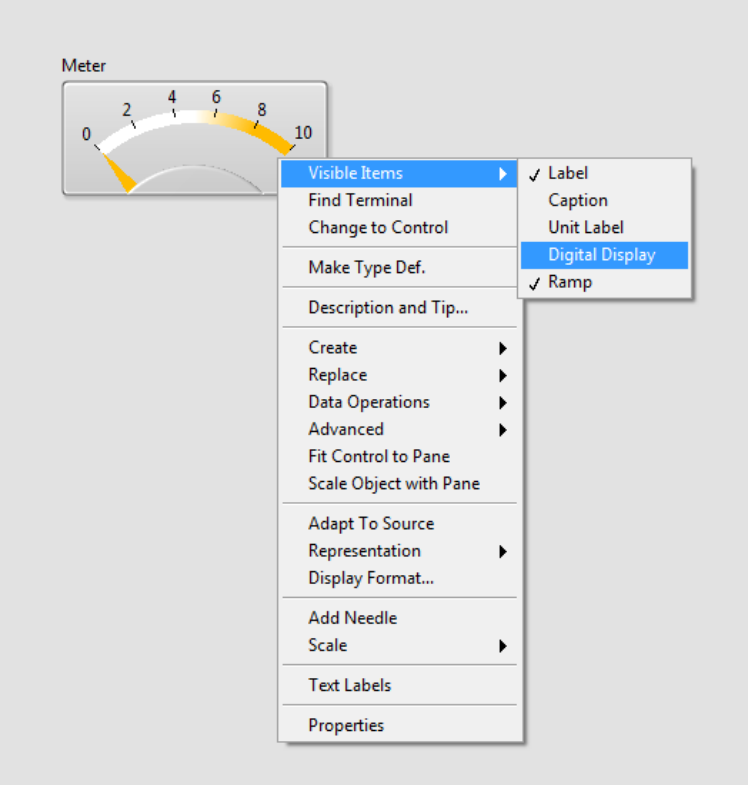
Front Panel – Strings

- The string data type is a sequence of any characters
- Use string controls to receive text from the user, such as a password or user name
- Use string indicators to display text to the user

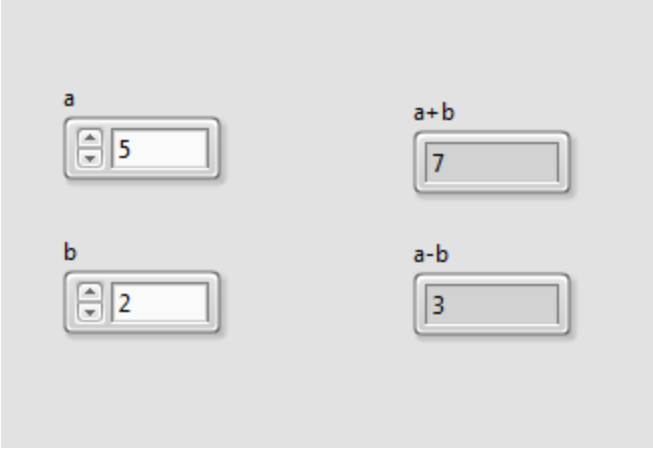


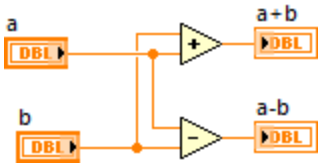
Front Panel – Shortcut Menus

- All LabVIEW objects have associated shortcut menus
- As you create a VI, use the shortcut menu items to change the look or behavior of front panel and block diagram objects
- To access the shortcut menu, right-click the object



Block Diagram



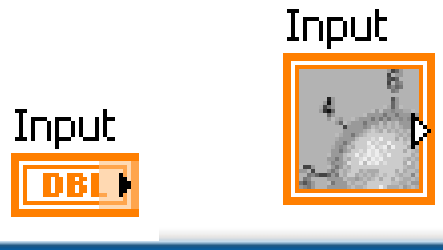


Block diagram objects include the following:

- Terminals
- SubVIs
- Nodes
- Constants
- Structures
- Wires

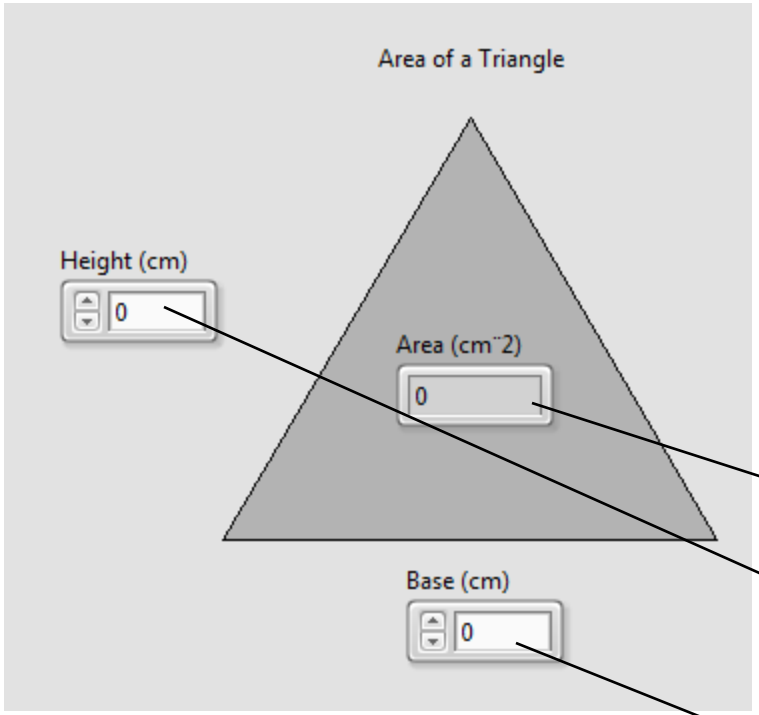
Block Diagram – Terminals

- Terminals are:
  - Block diagram appearance of front panel objects
  - Entry and exit ports that exchange information between the front panel and block diagram
  - Analogous to parameters and constants in text-based programming languages
- Change the view type of a terminal by toggling the **View as Icon** selection from the context menu

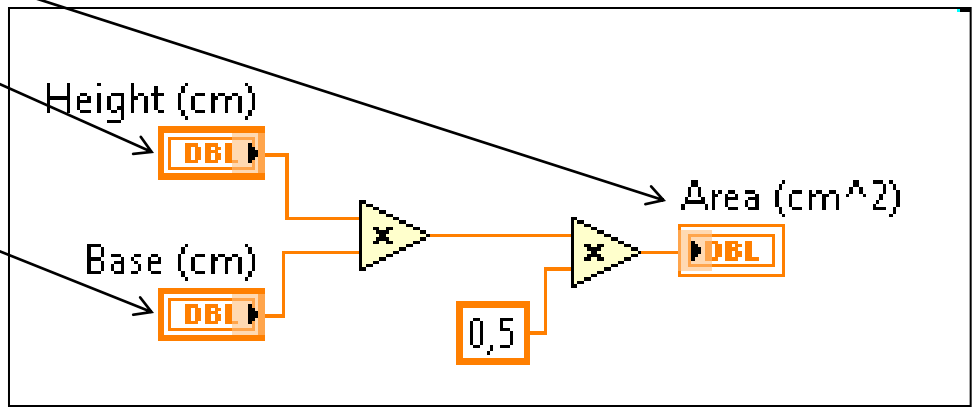


Block Diagram – Terminals

Front Panel

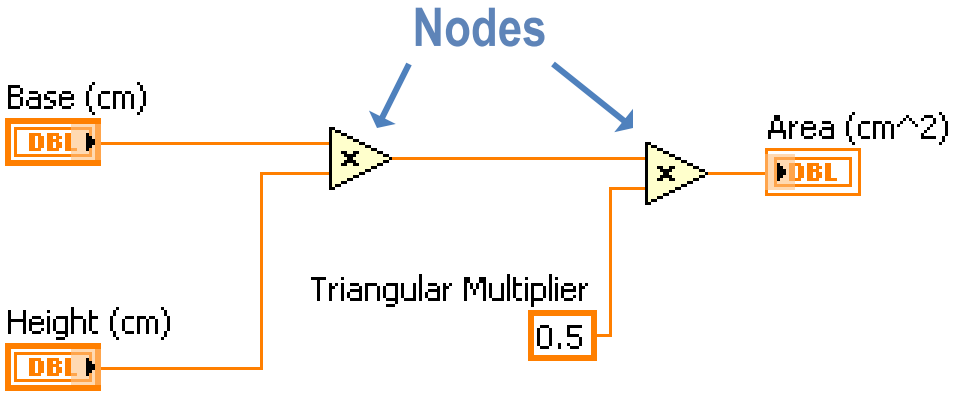


Block Diagram

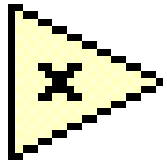


Block Diagram – Nodes

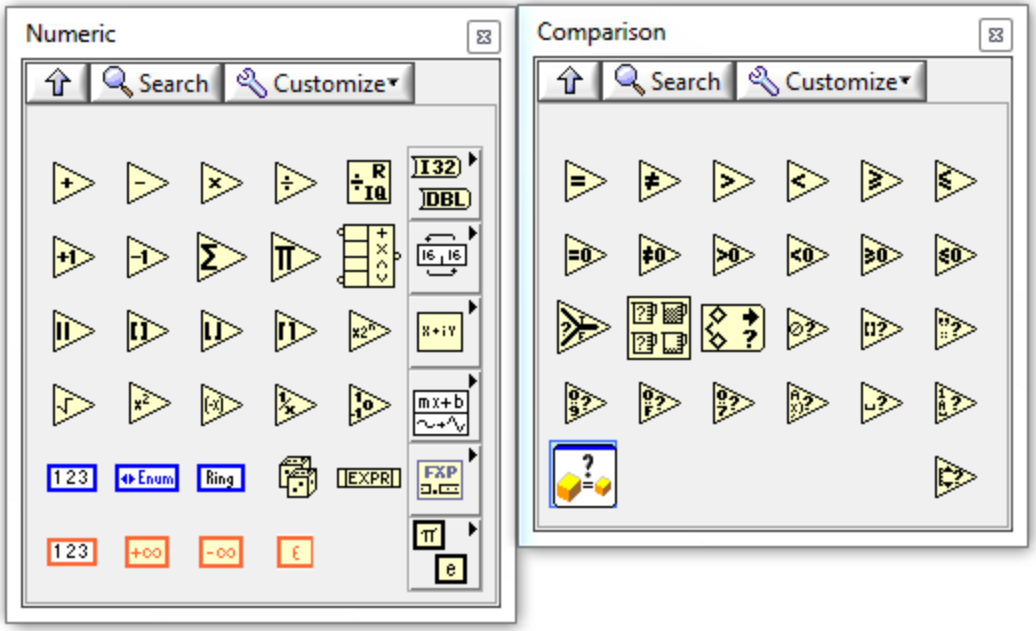
- Objects on the block diagram that have inputs and/or outputs and perform operations when a VI runs
- Analogous to statements, operators, functions, and subroutines in text-based programming languages
- Nodes can be functions, subVIs, or structures



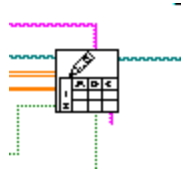
Block Diagram – Function Nodes



- Fundamental operating elements of LabVIEW
- Do not have front panels or block diagrams, but do have connector panes
- Double-clicking a function only selects the function – it does not open it like a VI
- Has a pale yellow background on its icon

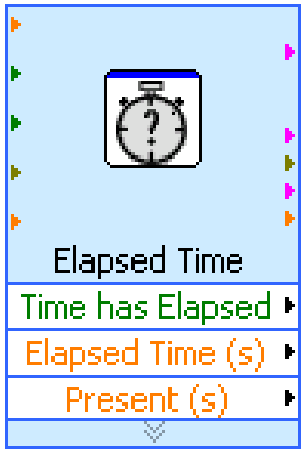


Block Diagram – SubVI Nodes



- SubVI: VIs that you build to use inside of another VI
- Any VI has the potential to be used as a subVI
- When you double-click a subVI on the block diagram, you can view the front panel and block diagram of the subVI
  - The upper right corner of the front panel and block diagram displays the icon for the current VI
  - This is the icon that appears when you place the VI on a block diagram as a subVI

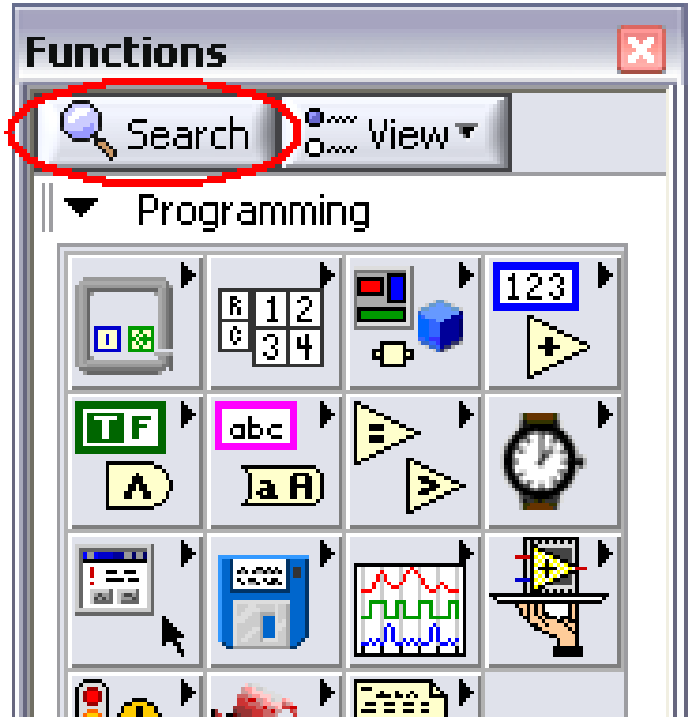
Block Diagram – Express VIs



- Express VIs are a special type of subVI
  - Require minimal wiring because you configure them with dialog boxes
  - Save the configuration of an Express VI as a subVI
- Icons for Express VIs appear on the block diagram as icons surrounded by a blue field

Searching for Controls, VIs, and Functions

Find controls, functions, and VIs using the **Search** button on the **Controls** and **Functions** palettes.



Tip: You can also use Quick Drop: <Ctrl-Space> to search through your palettes quickly.

Block Diagram – Wires

- Transfer data between block diagram objects through wires
- Wires are different colors, styles, and thicknesses, depending on their data types
- A broken wire appears as a dashed black line with a red X in the middle

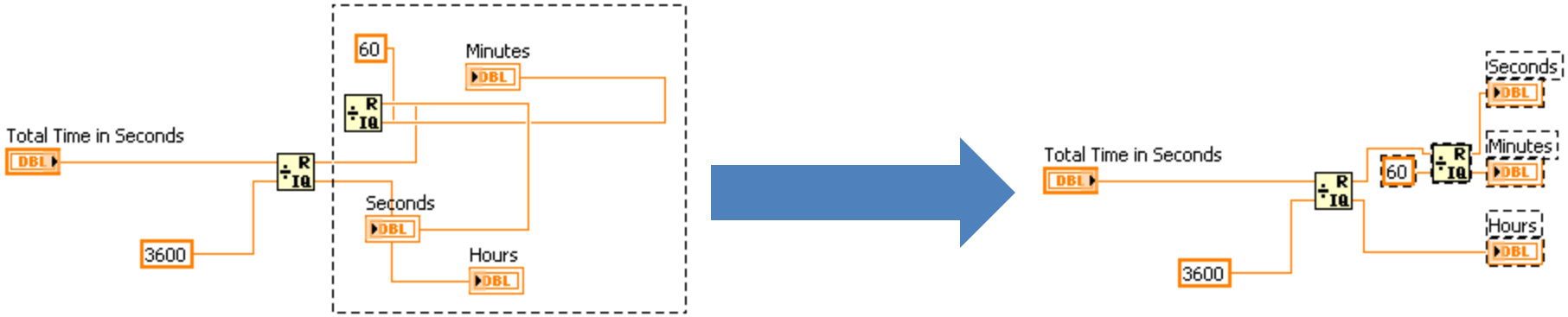
	Double Numeric	Integer Numeric	String
Scalar			
1D Array			
2D Array			

Block Diagram – Clean Up

Use the Clean Up Diagram tool to reroute multiple wires and objects to improve readability

1. Select a section of your block diagram
2. Click the Clean Up Diagram button on the block diagram toolbar
3. If you do not select anything, it will clean the entire block diagram

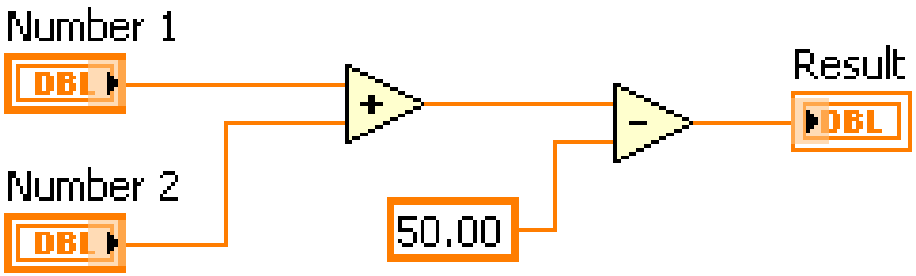
Use the <Ctrl-U> shortcut for speed

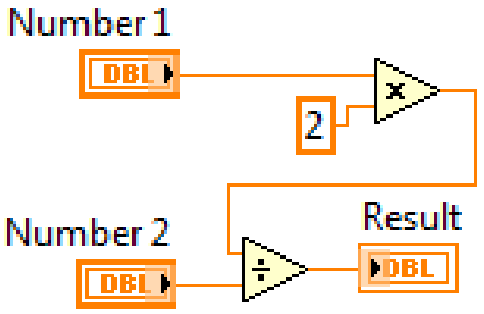


Dataflow

LabVIEW follows a dataflow model for running VIs

- A node executes only when data is available at all of its input terminals
- A node supplies data to the output terminals only when the node finishes execution





Dataflow – Quiz

Which node executes first?

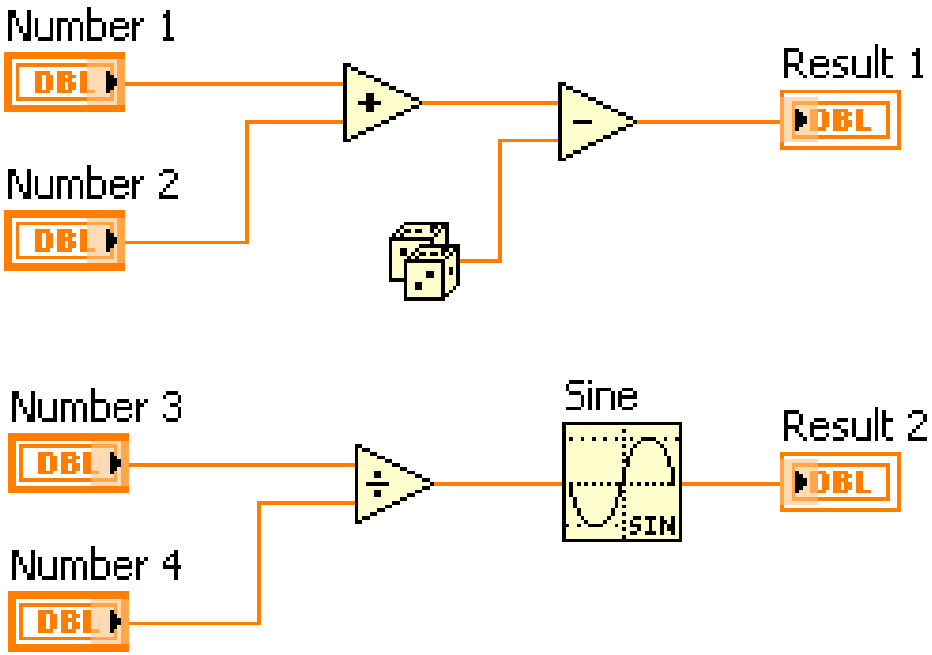
a) Add

b) Subtract

c) Random Number

d) Divide

e) Sine



Dataflow – Quiz Answers

NO CORRECT ANSWER

Which node executes first?

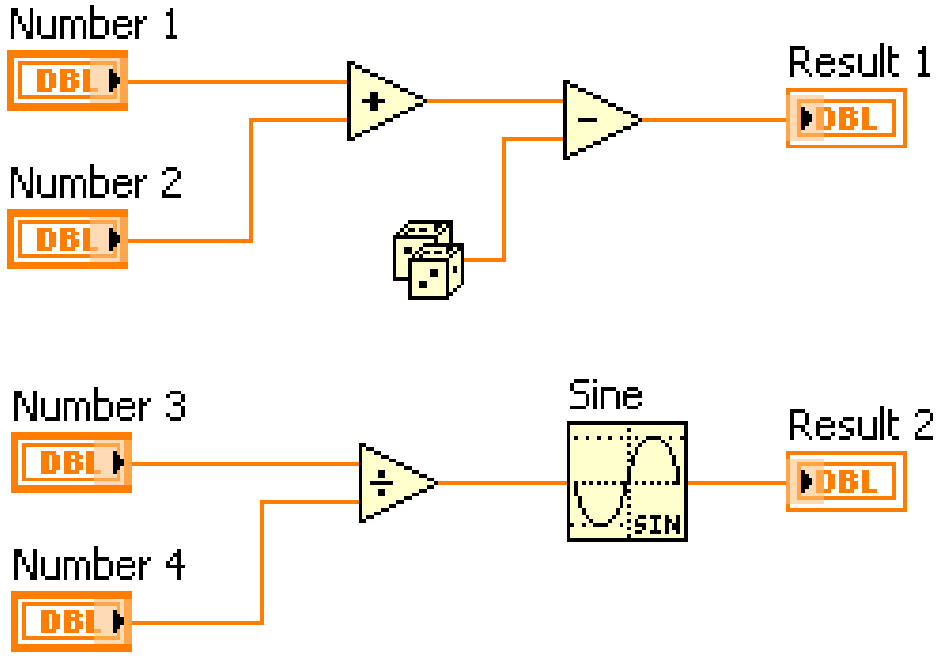
a) *Add – possibly*

b) *Subtract – definitely not*

c) *Random Number – possibly*

d) *Divide – possibly*

e) *Sine – definitely not*



Summary – Quiz

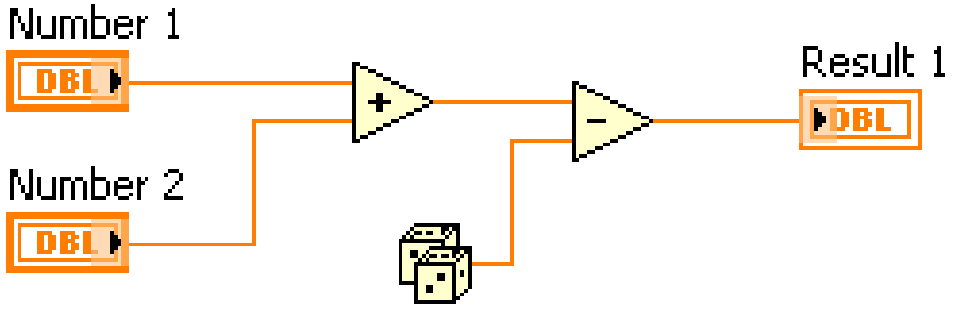
Which function executes first,

Add or Subtract?

a) Add

b) Subtract

c) Unknown



Summary – Quiz Answer

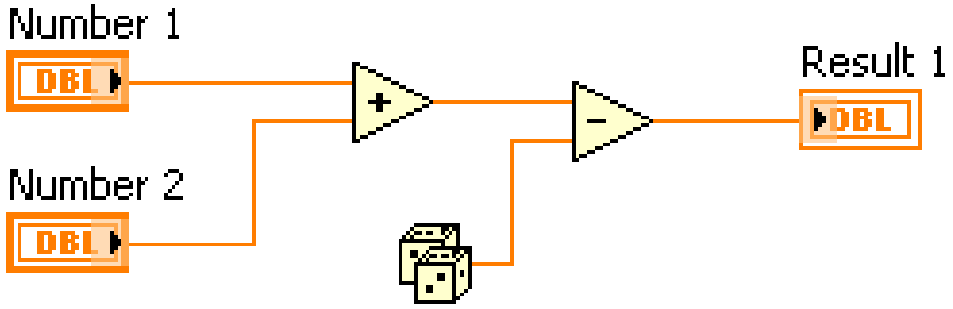
Which function executes first,

Add or Subtract?

a) **Add**

b) Subtract

c) Unknown



Summary – Quiz

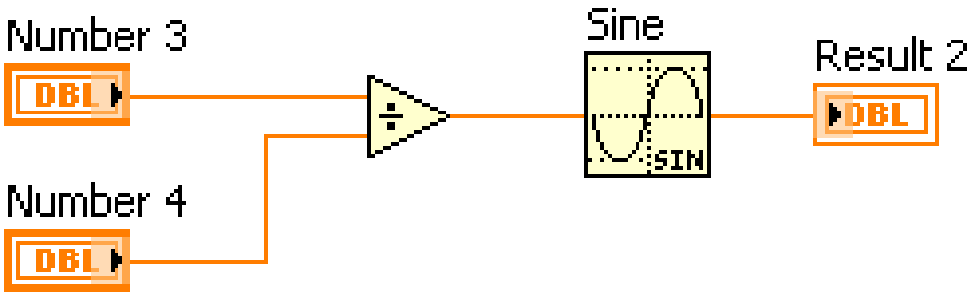
Which function executes first,

Sine or Divide?

a) Sine

b) Divide

c) Unknown



Summary – Quiz Answer

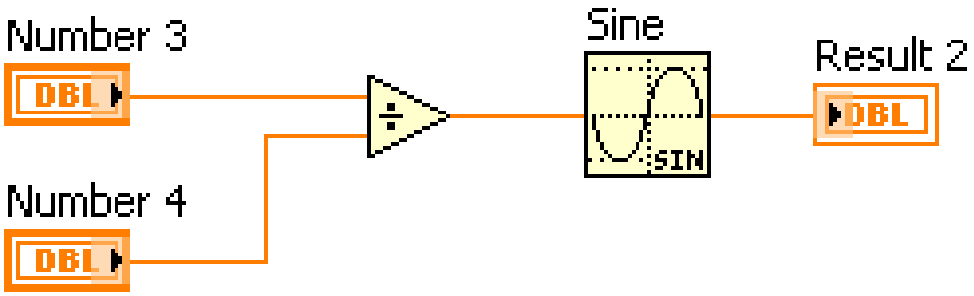
Which function executes first,

Sine or Divide?

a) Sine

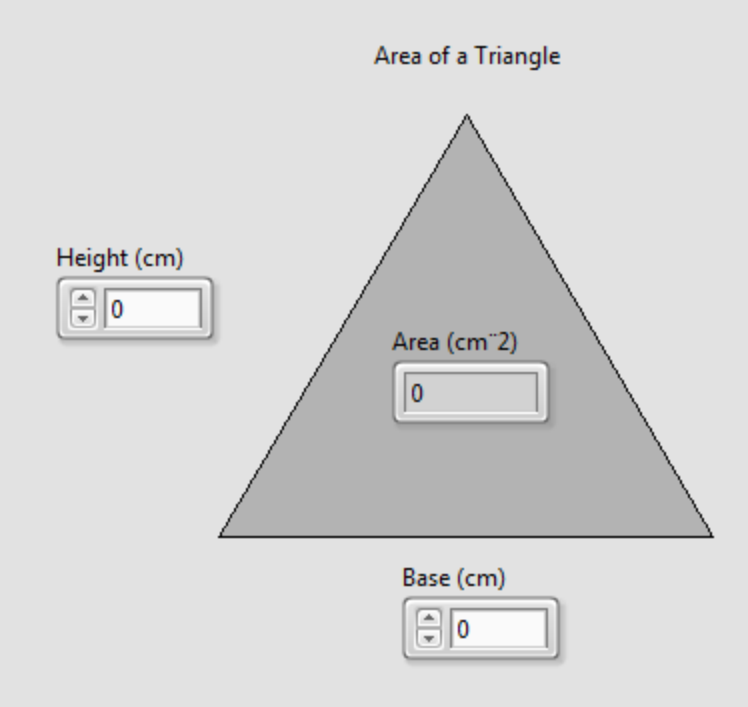
b) Divide

c) Unknown

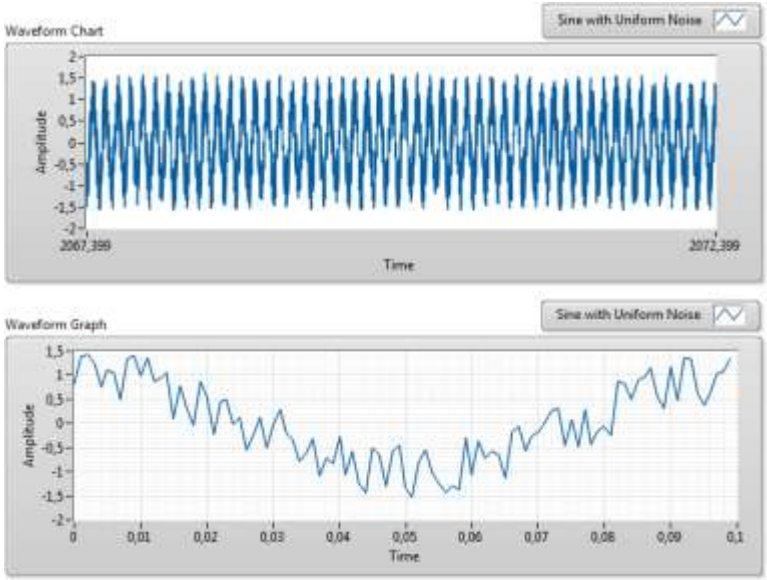


Demonstration

- Calculating the area of a triangle

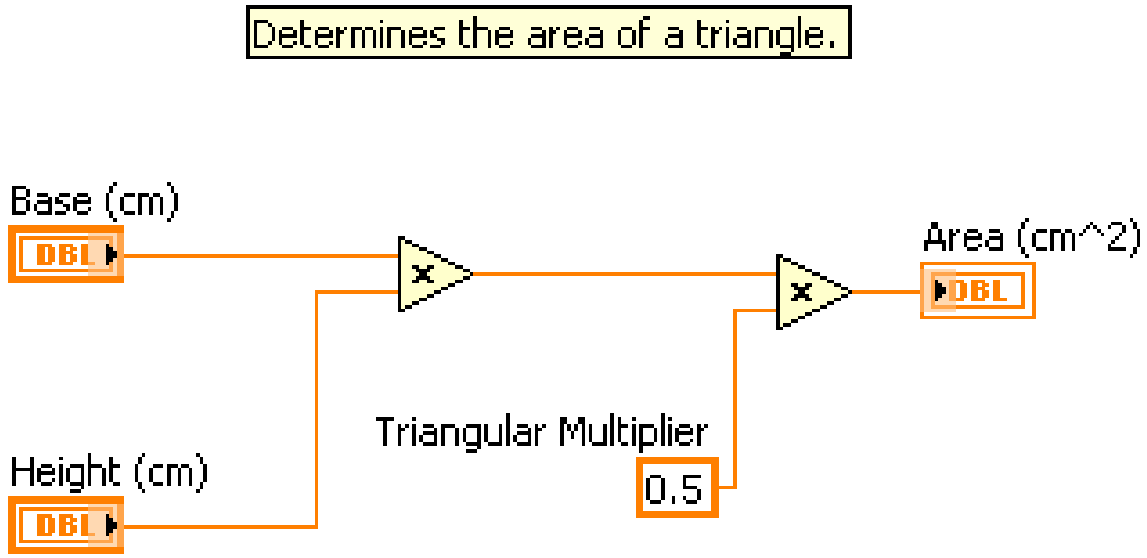


- Signal Processing



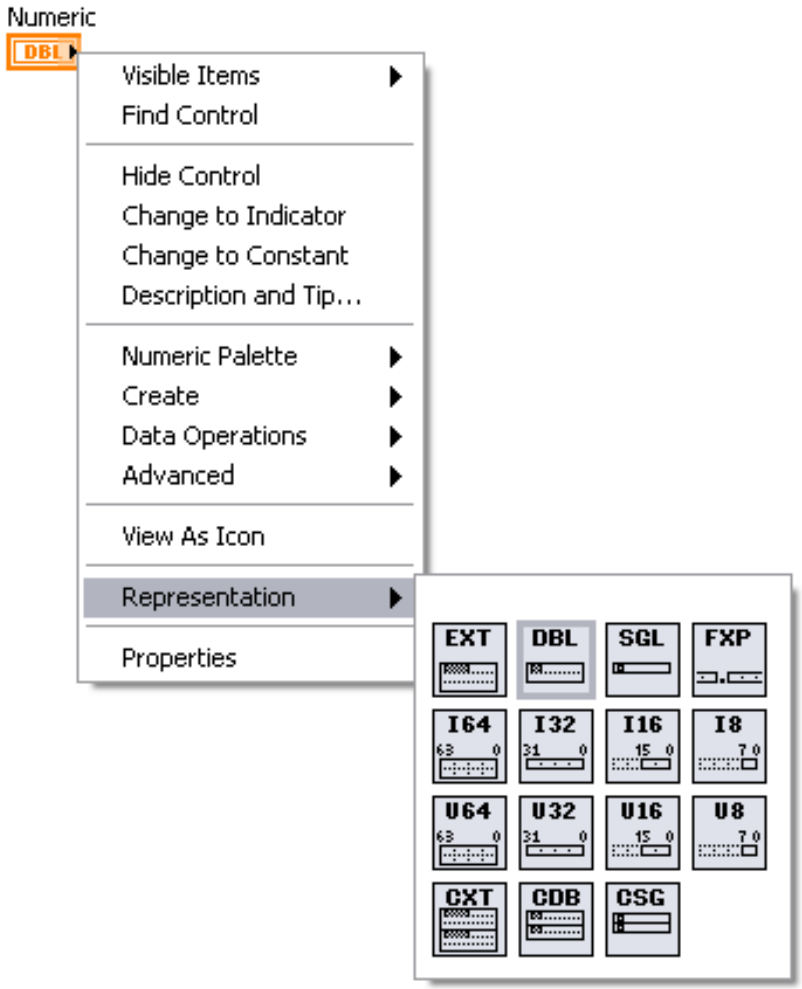
LabVIEW Data Types – Terminals

Terminals visually communicate information about the data type represented



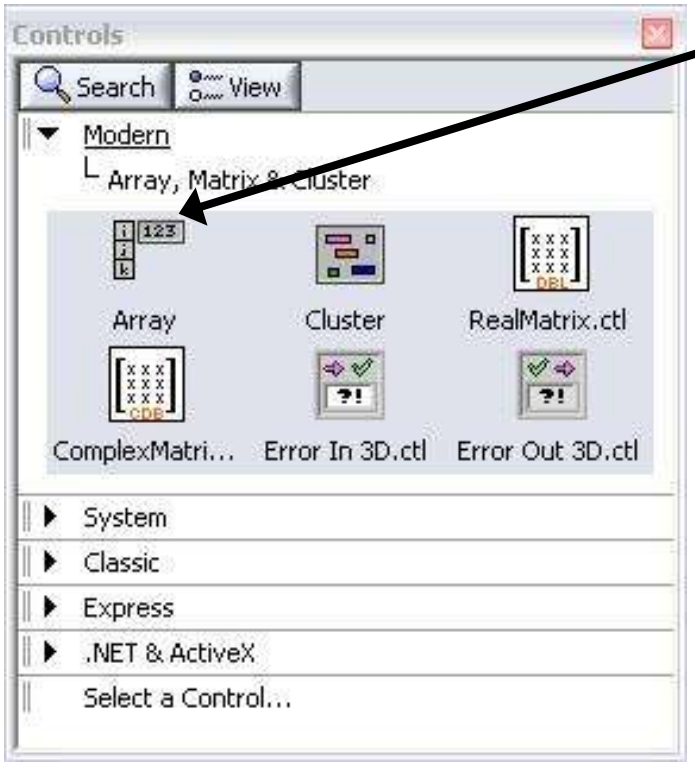
LabVIEW Data Types – Numerics

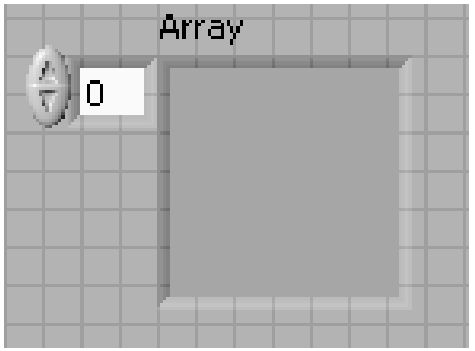
- The numeric data type represents numbers of various types
- To change the representation of a numeric, right-click the control, indicator, or constant, and select **Representation** from the shortcut menu



Creating an Array (Step 1 of 2)

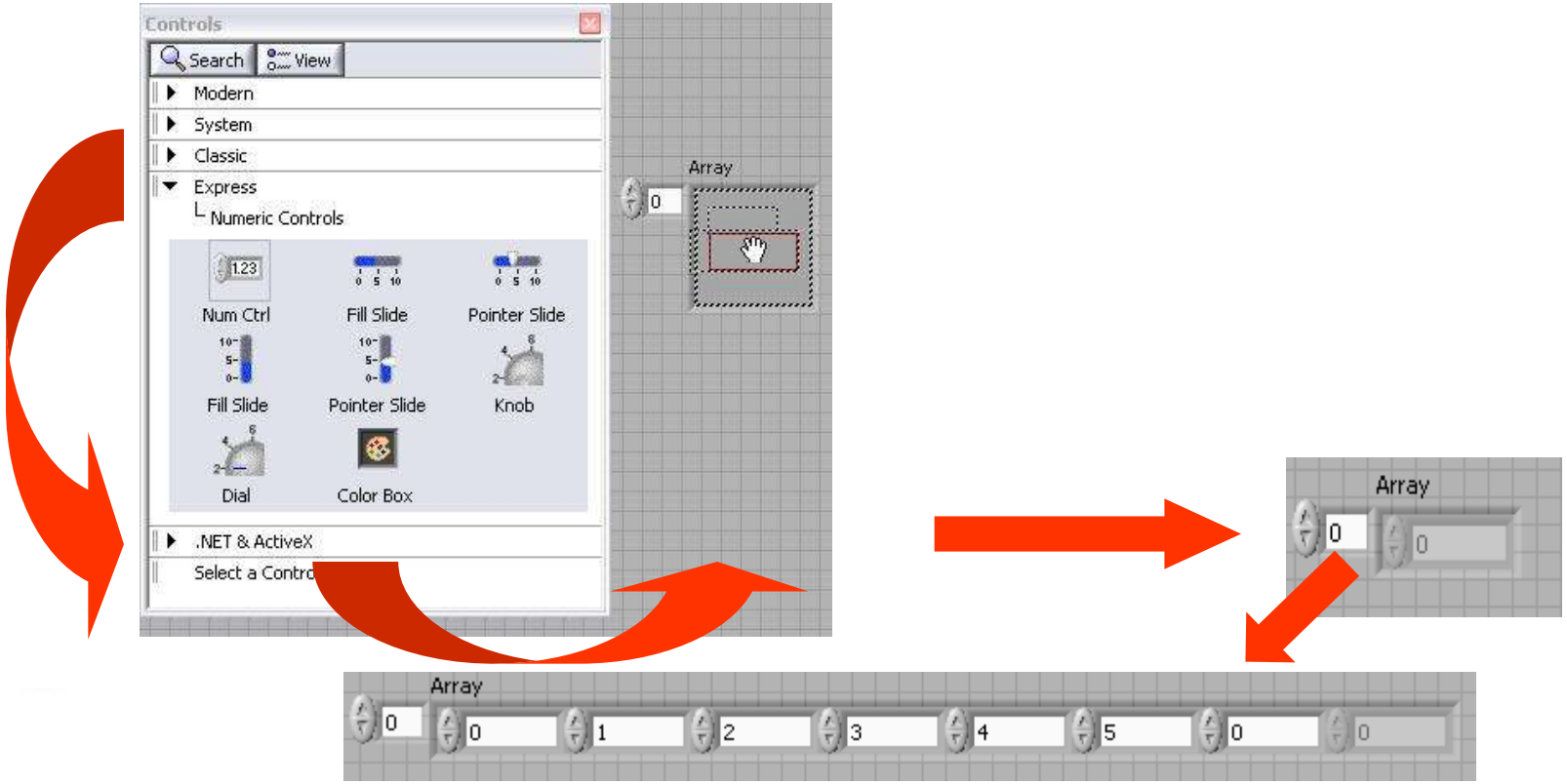
From the **Controls»Modern»Array, Matrix, and Cluster** subpalette, select the **Array** icon



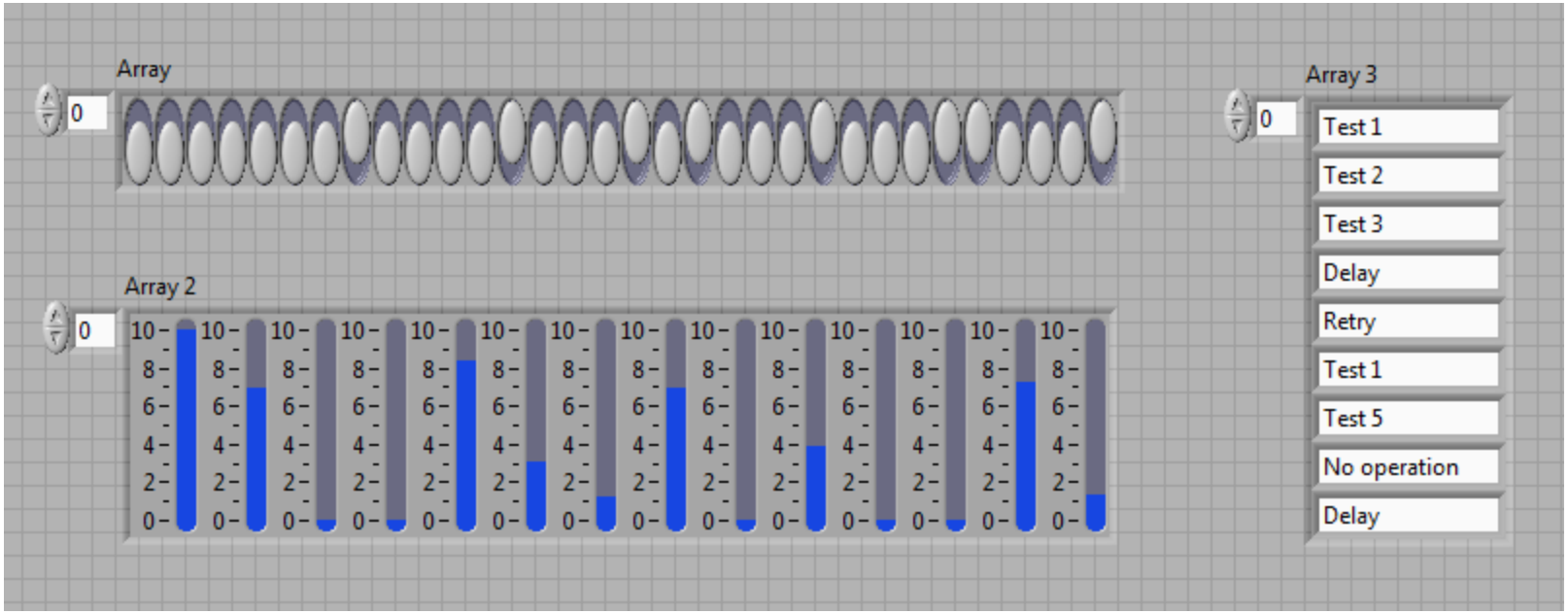


Create an Array (Step 2 of 2)

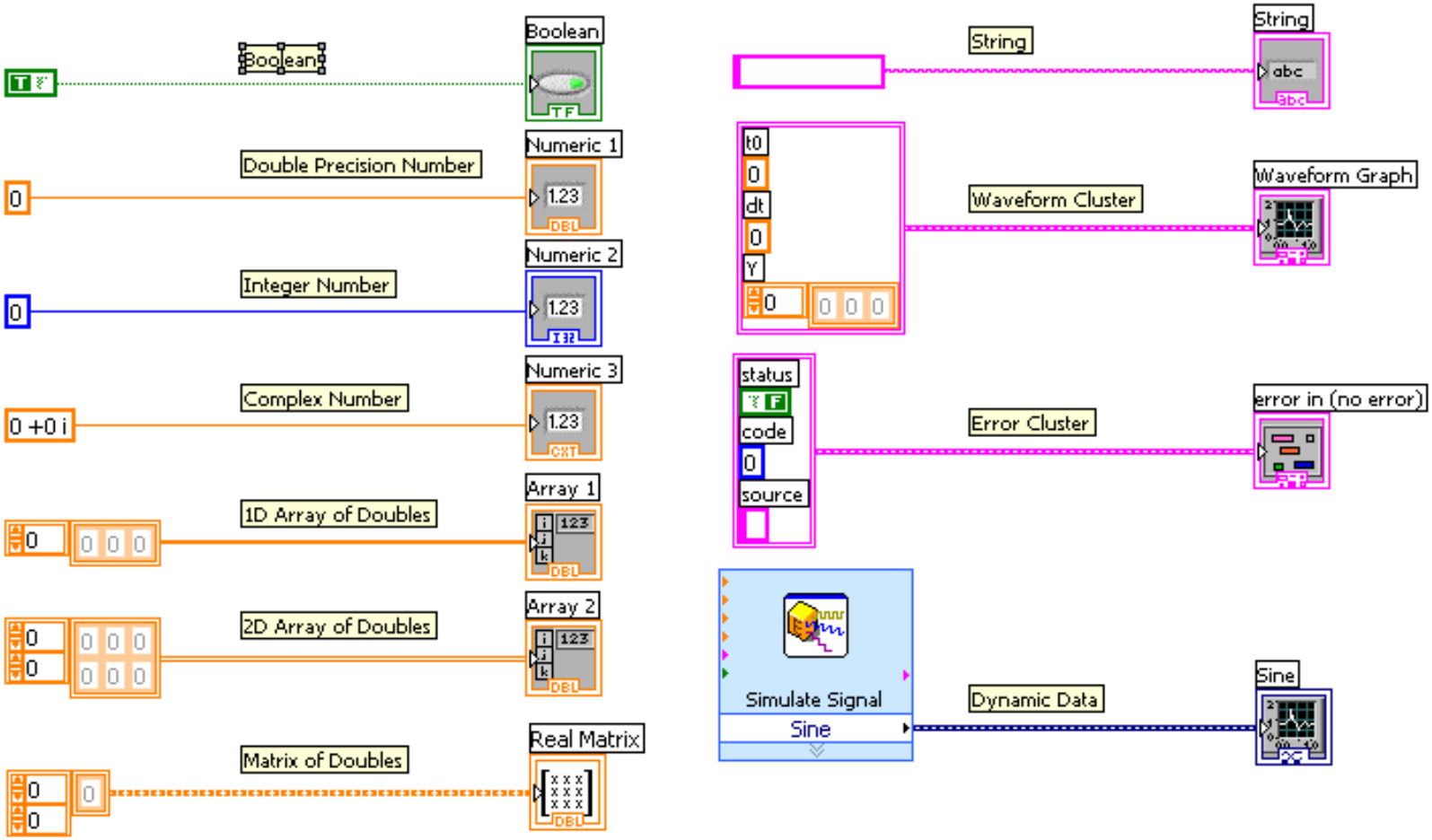
1. Place an Array Shell.
2. Insert any data type into the shell (such as Numeric Control).



Different Array Types



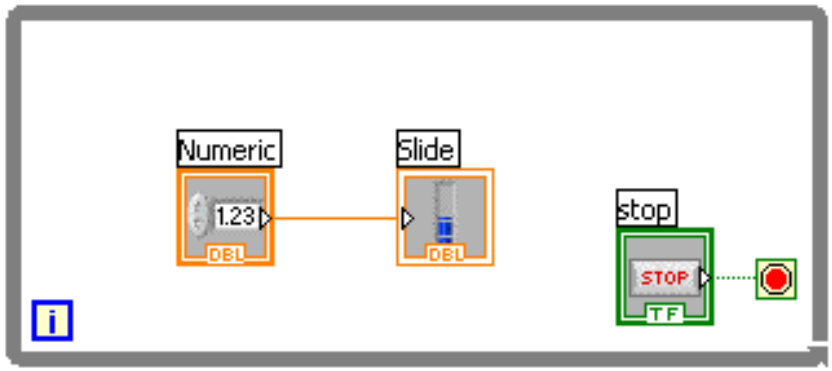
Review of Data Types Found in LabVIEW



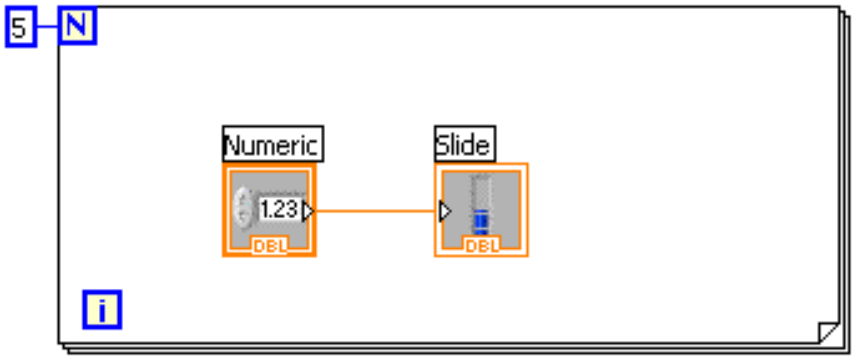
Loops

- While Loops
  - i terminal counts iteration
  - Always runs at least once
  - Runs until stop condition is met
- For Loops
  - i terminal counts iterations
  - Run according to input N of count terminal N

While Loop

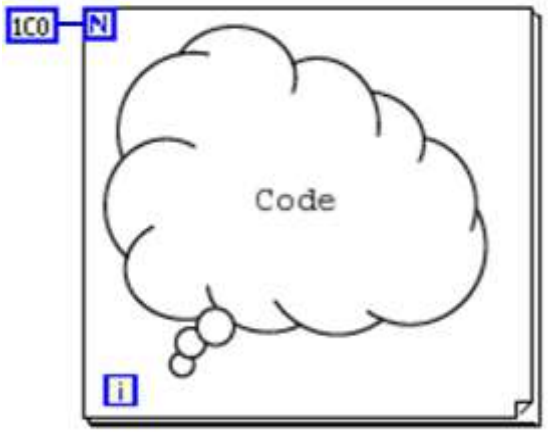


For Loop



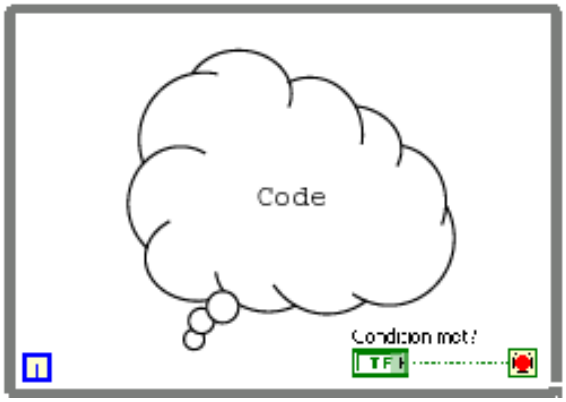
For Loop/While Loop Comparison

For Loop



- Executes a set number of times unless a conditional terminal is added
- Can execute zero times

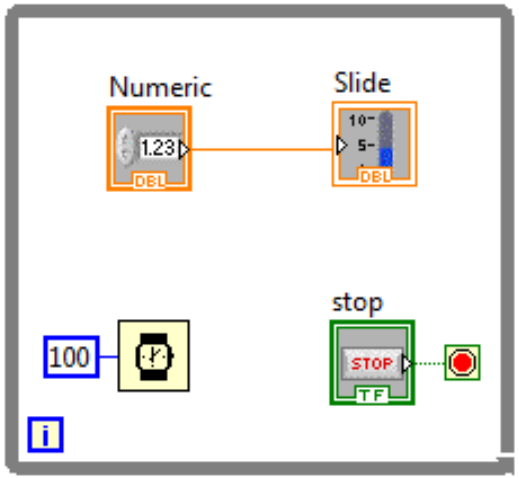
While Loop



- Stops executing only if the value at the conditional terminal meets the condition
- Must execute at least once

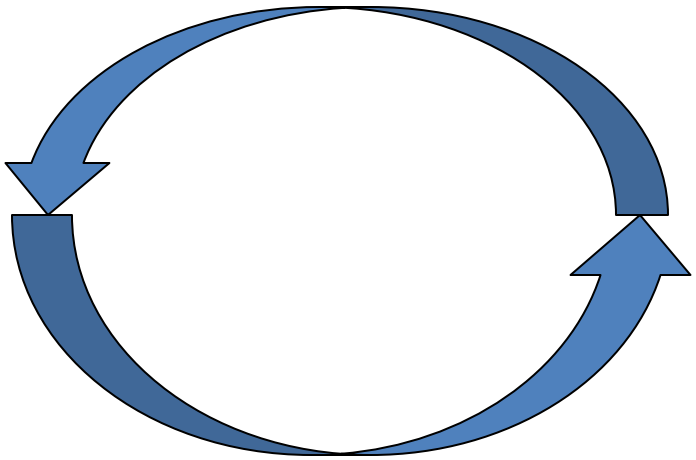
Timing a loop

- LabVIEW will run each loop as fast as possible by default
- You can decide how fast it should go by using timing functions, like ‘Wait [ms]’



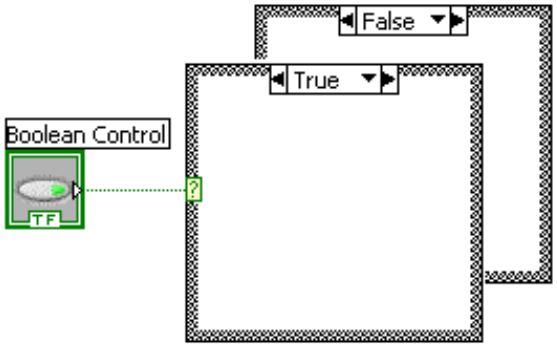
Demonstration

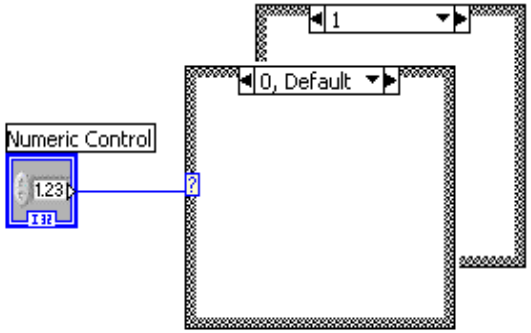
- Making previously built VIs work continously until stopped



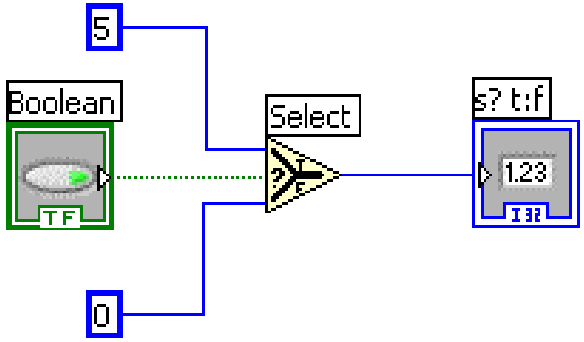
How Do I Make Decisions in LabVIEW?

1. Case Structures





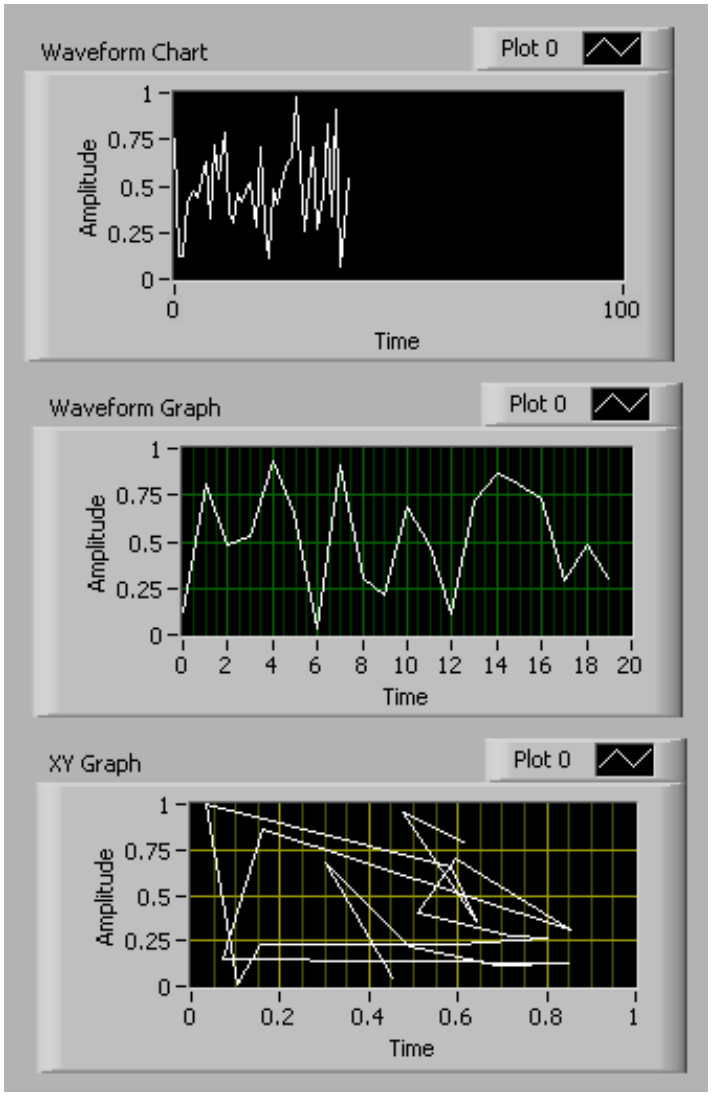
2. Select

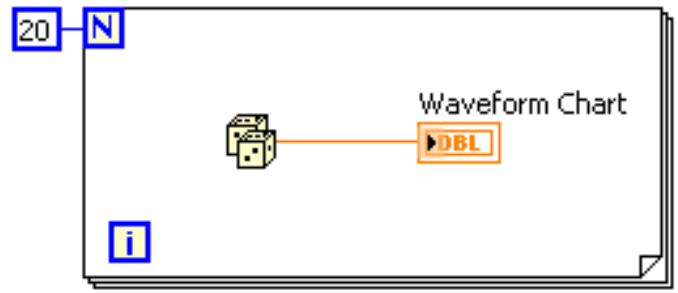


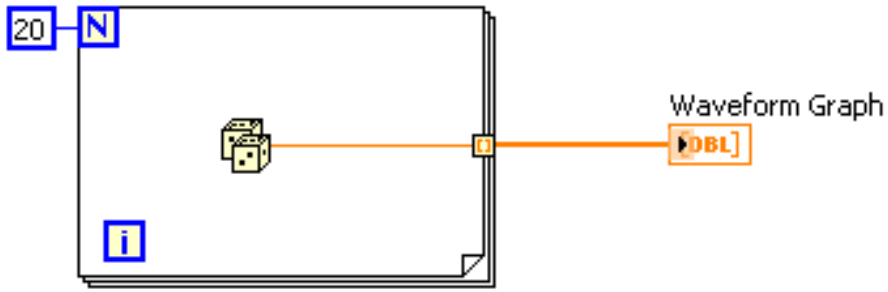
Demonstration

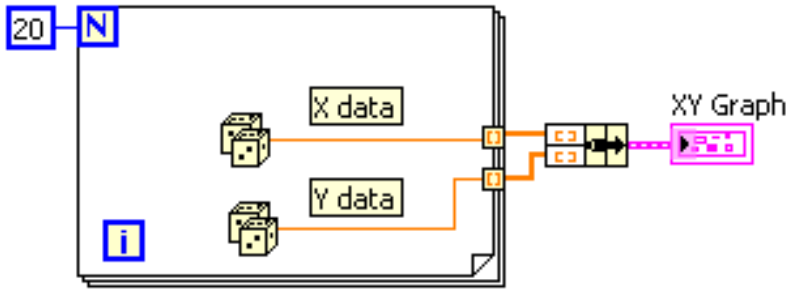
- Make it possible to turn off FFT calculation in Signal Processing example

Plotting Data



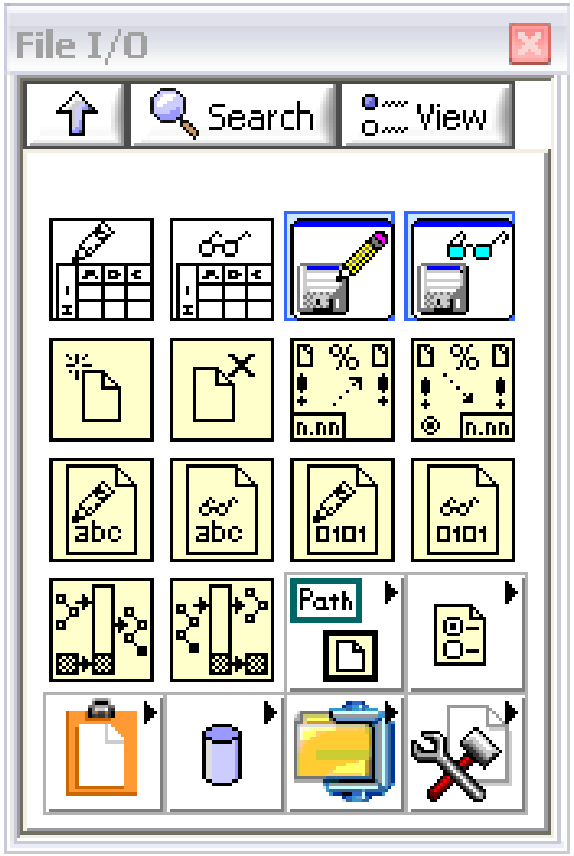






Understanding High-Level File I/O

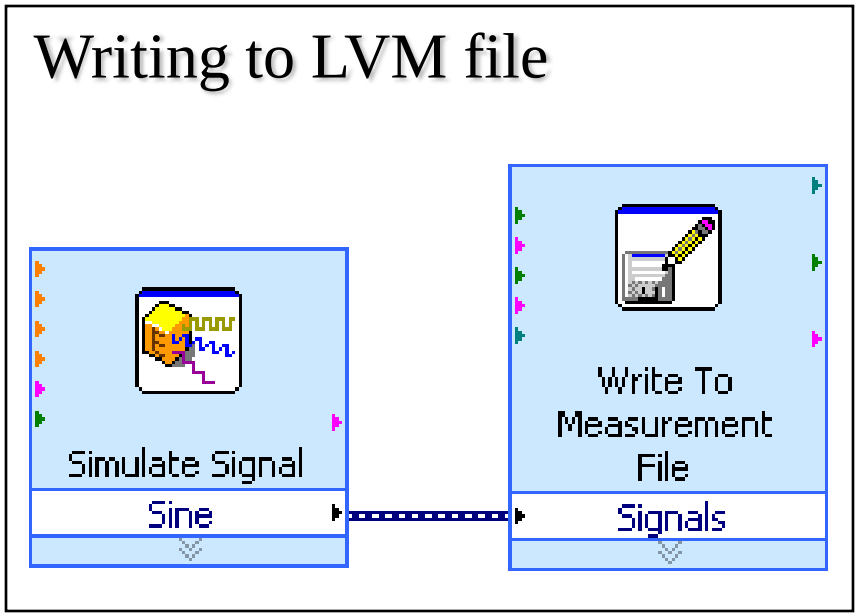
- High-Level VIs
  - Perform all 3 steps (open, read/write, close) for common file I/O operations
  - Might not be as efficient as the functions configured or designed for individual operations
- Low-Level VIs
  - Individual VI for each step
  - If you are writing to a file in a loop, use low-level file I/O functions

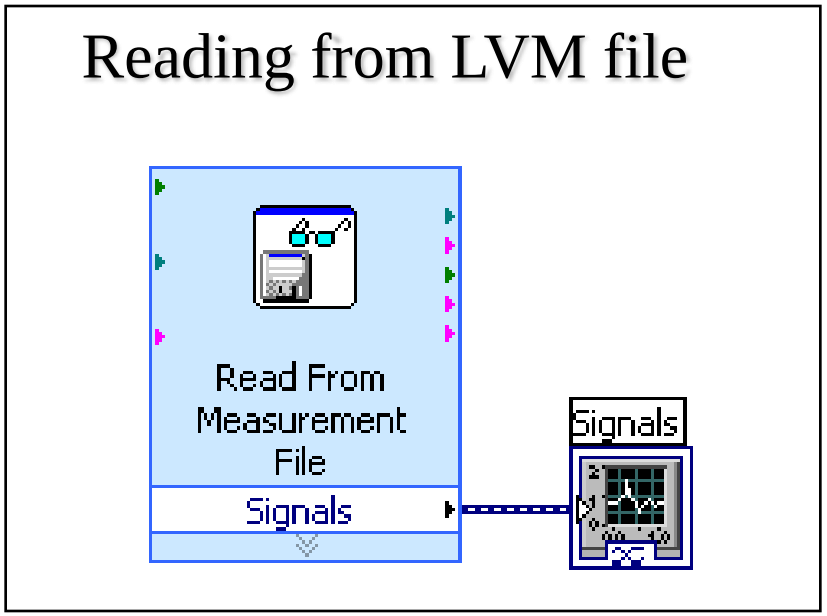


File I/O

File I/O – passing data to and from files

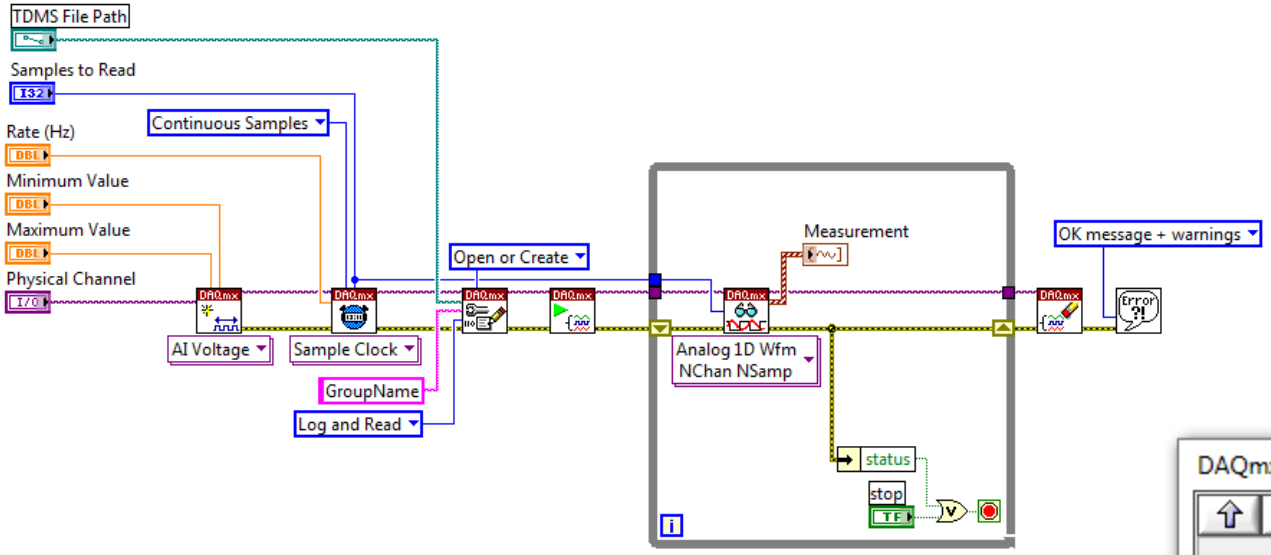
- Files can be binary, text, or spreadsheet
- Write/Read LabVIEW Measurements file (*.lvm)

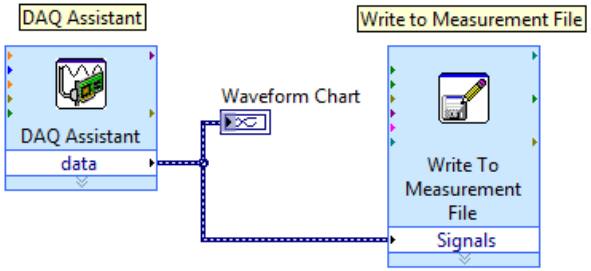


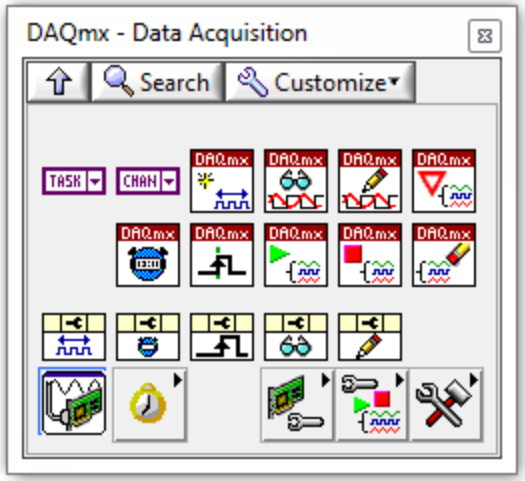


Data Acquisition

- Similar to File I/Oh igh level and low level functions available:

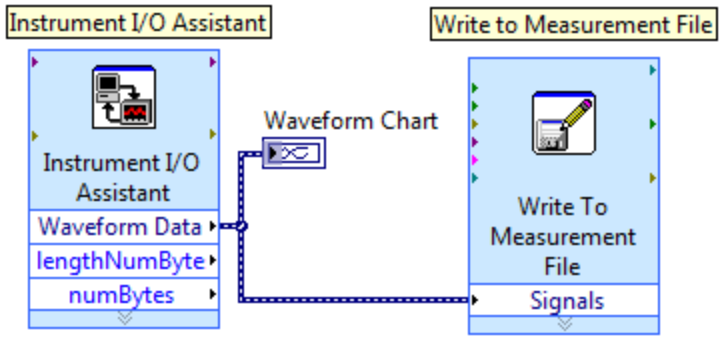


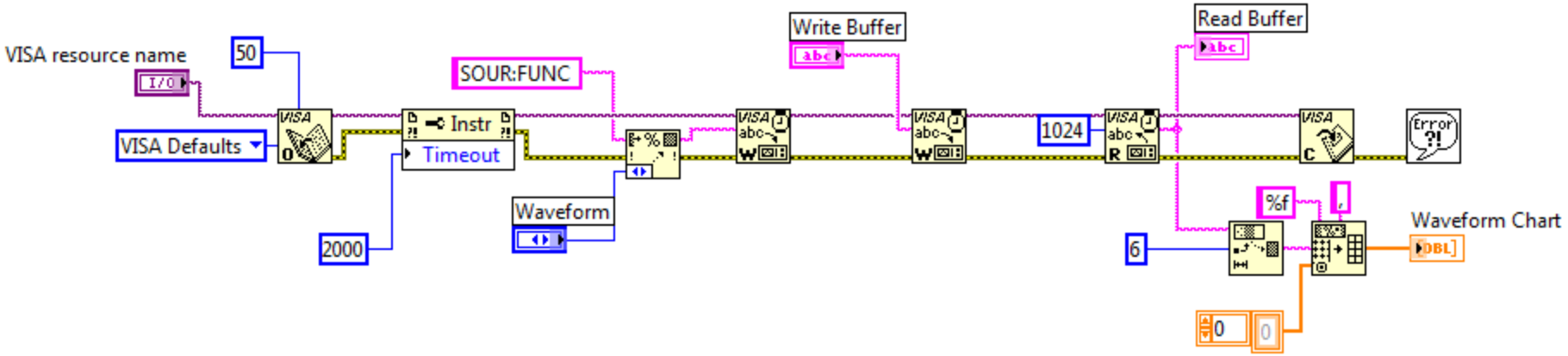




Instrument Control

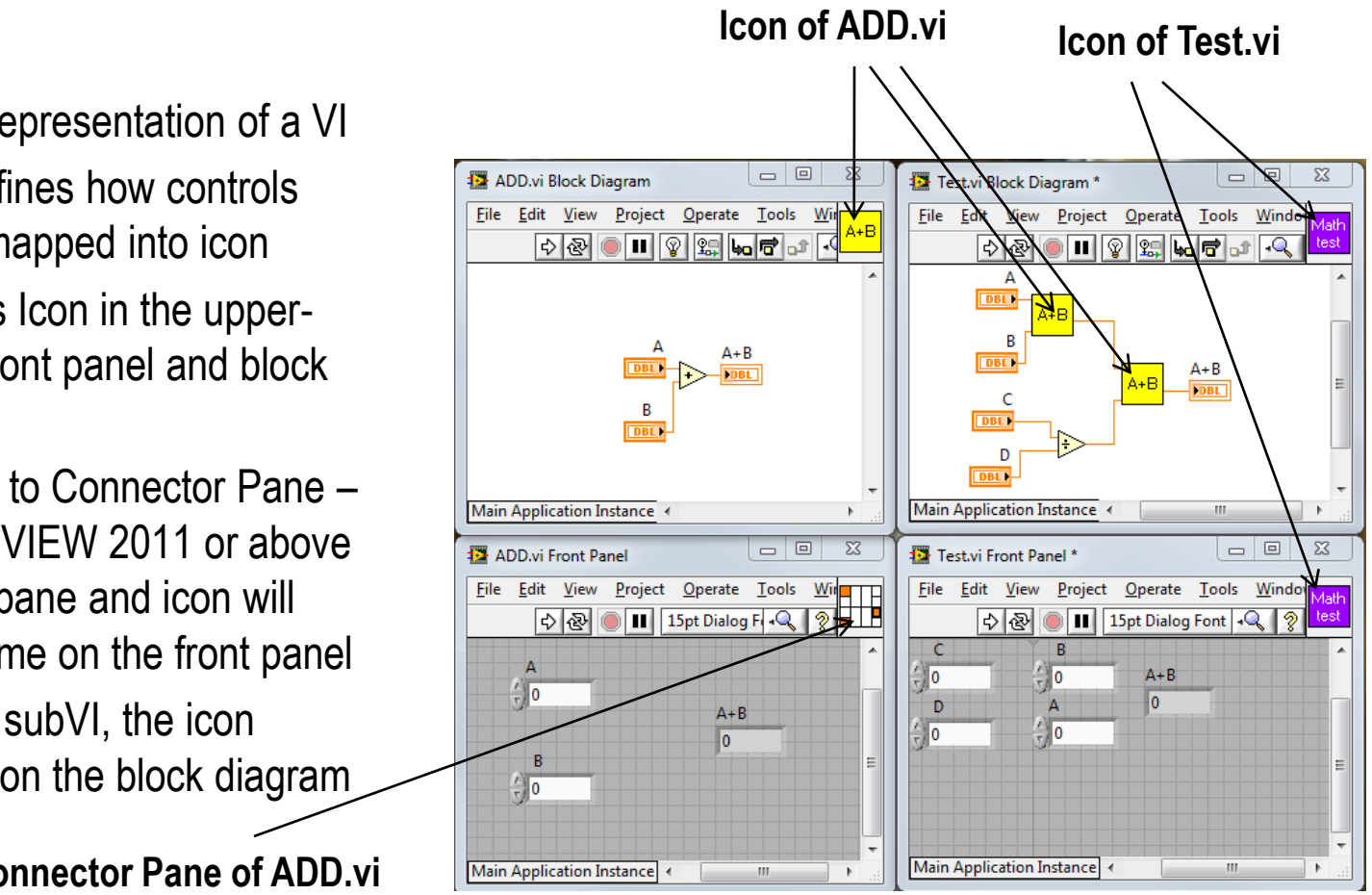
- Controlling instruments using serial communication protocols (RS-232, GPIB and similar).
- Basing on VISA – Virtual Instrument Software Architecture – generalized driver, widely used in industry





Icon and Connector Pane

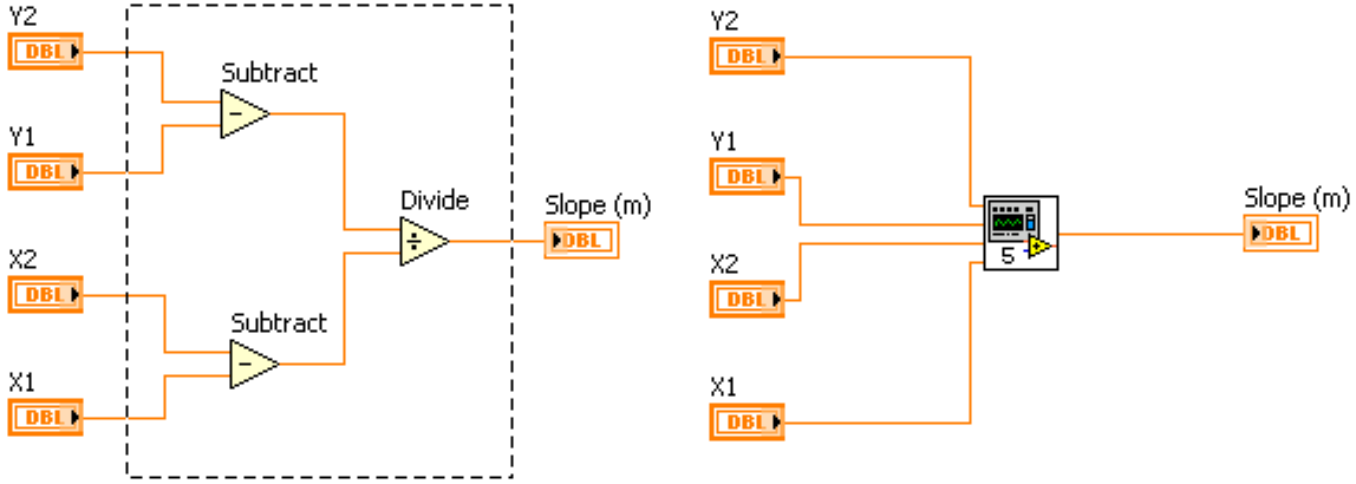
- Icon is a graphical representation of a VI
- Connector Pane defines how controls and indicators are mapped into icon
- Every VI displays its Icon in the upper-right corner of the front panel and block diagram windows
- Right click to switch to Connector Pane – If you are using LabVIEW 2011 or above both the connector pane and icon will show at the same time on the front panel
- If you use a VI as a subVI, the icon identifies the subVI on the block diagram



Using SubVIs – Section to SubVI

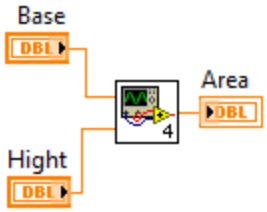
To convert a section of a VI into a subVI:

- Use the Positioning tool to select the section of the block diagram you want to reuse
- Select **Edit»Create SubVI**



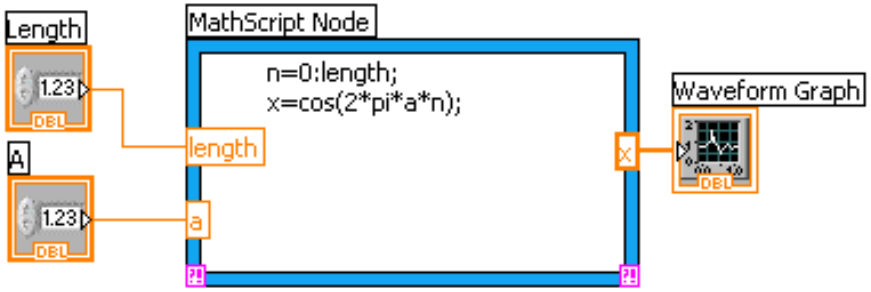
Demonstration

- Create a subVI which will calculate the area of a triangle

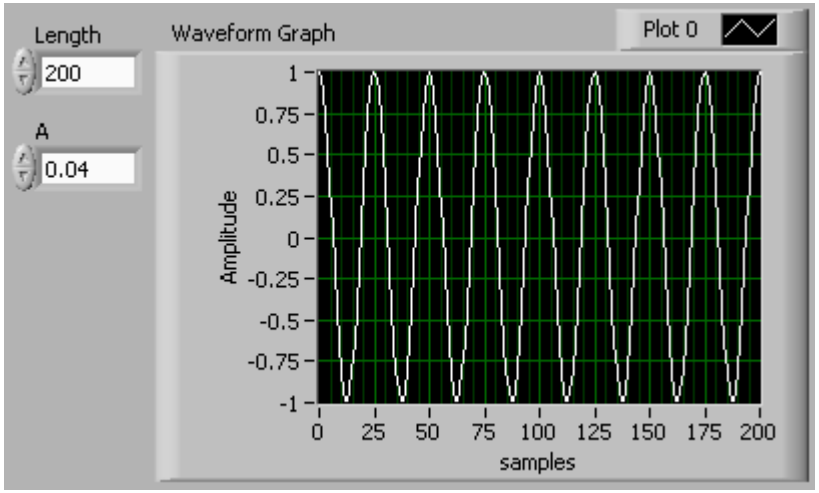


Math With the MathScript Node

- Implement equations and algorithms textually
- Input and Output variables created at the border
- Generally compatible with popular .m file scripts language
- Terminate statements with a semicolon to disable immediate output

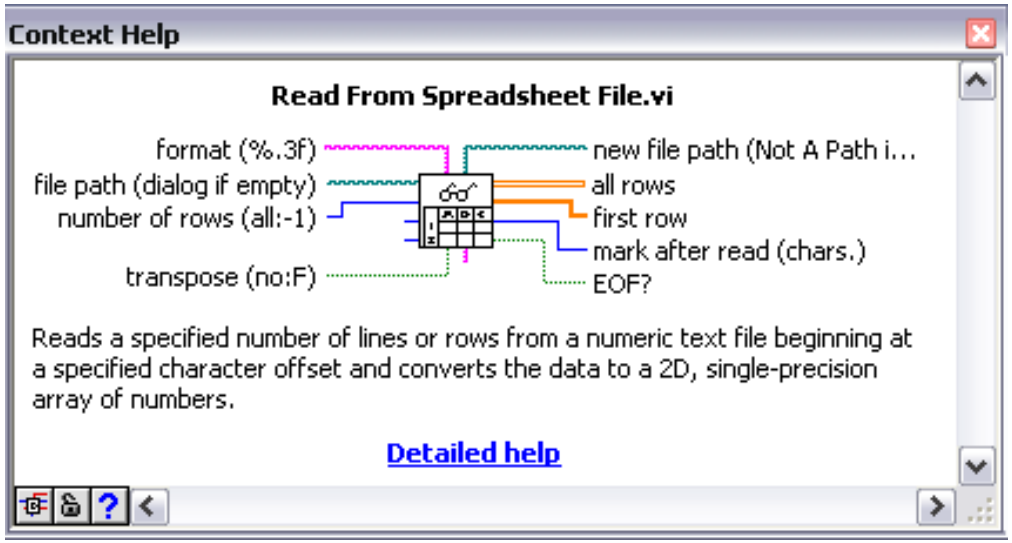


(Functions»Programming» Structures»MathScript)



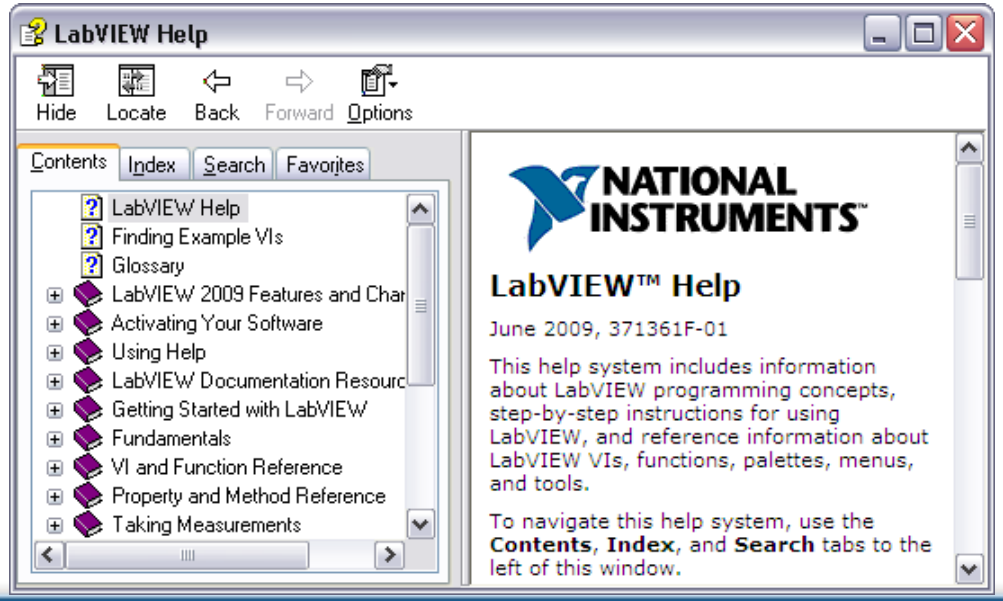
LabVIEW Help Utilities – Context Help

- Displays basic information about LabVIEW objects when you move the cursor over each object
- Select **Help»Show Context Help**, press <Ctrl-H> or
  click the **Show Context Help Window** button on the toolbar

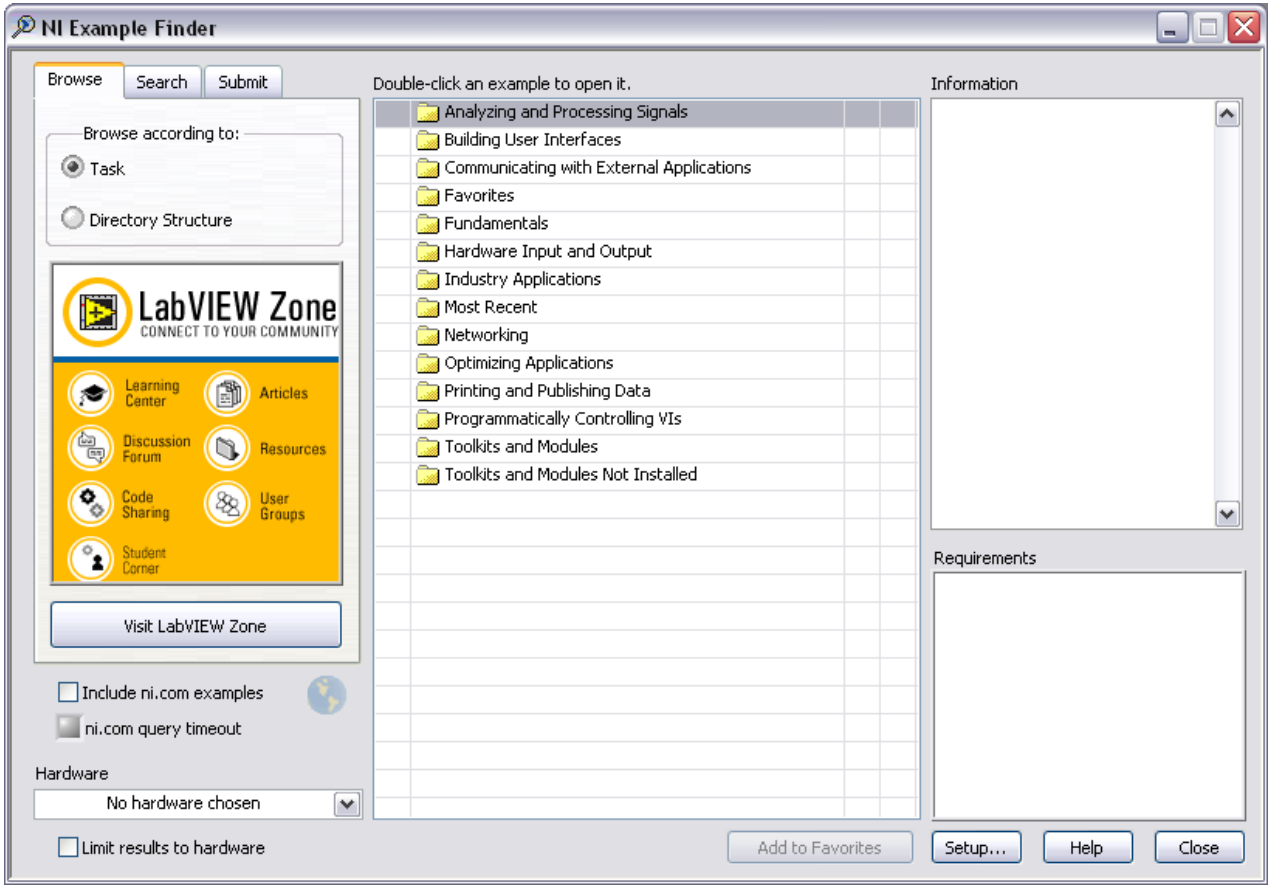


LabVIEW Help Utilities – LabVIEW Help

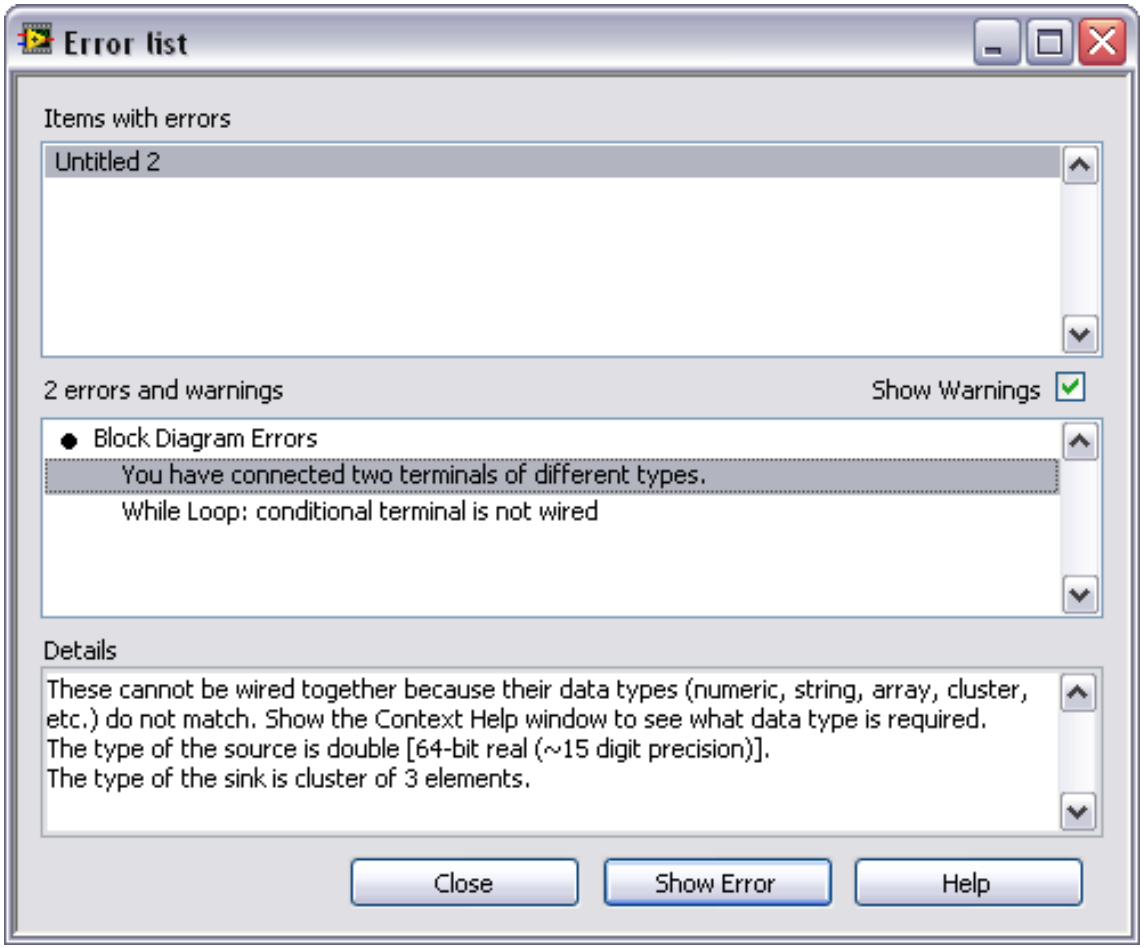
- Detailed descriptions of most palettes, menus, tools, VIs, and functions and instructions for using LabVIEW features
- Accessing the *LabVIEW Help*:
  - Select **Help»Search the LabVIEW Help**
  - Press **F1**
  - Use the **Detailed Help** link or button in the **Context Help** window



LabVIEW Help Utilities – NI Example Finder



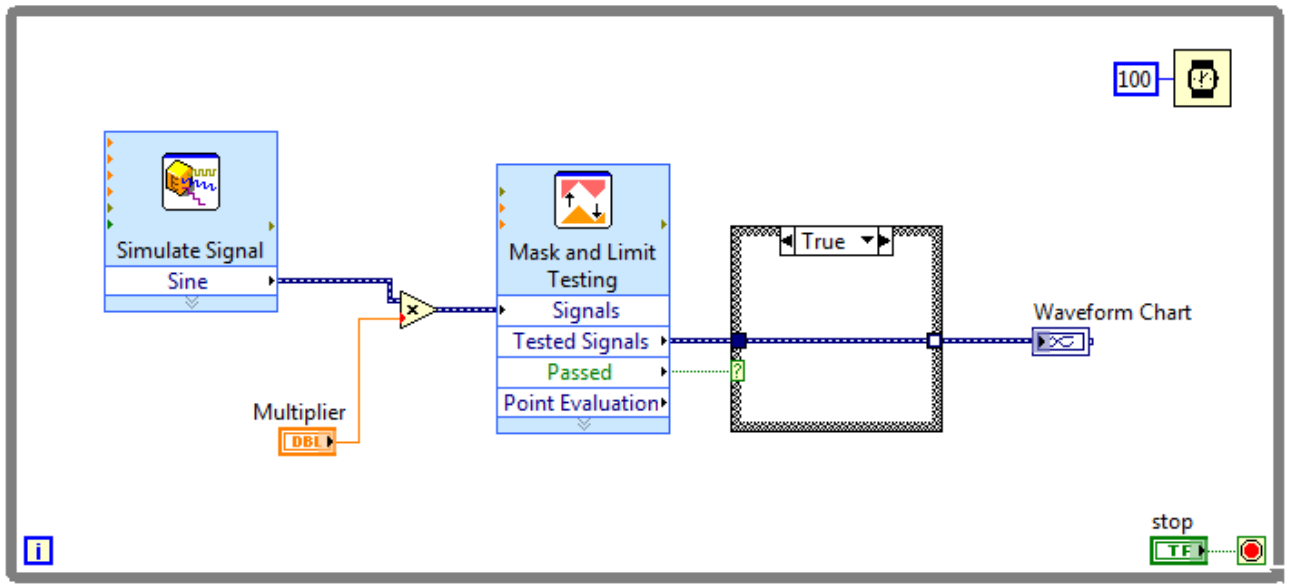
Correcting Broken VIs



Quiz

Which elements is not visible on this block diagram (multiple answers)

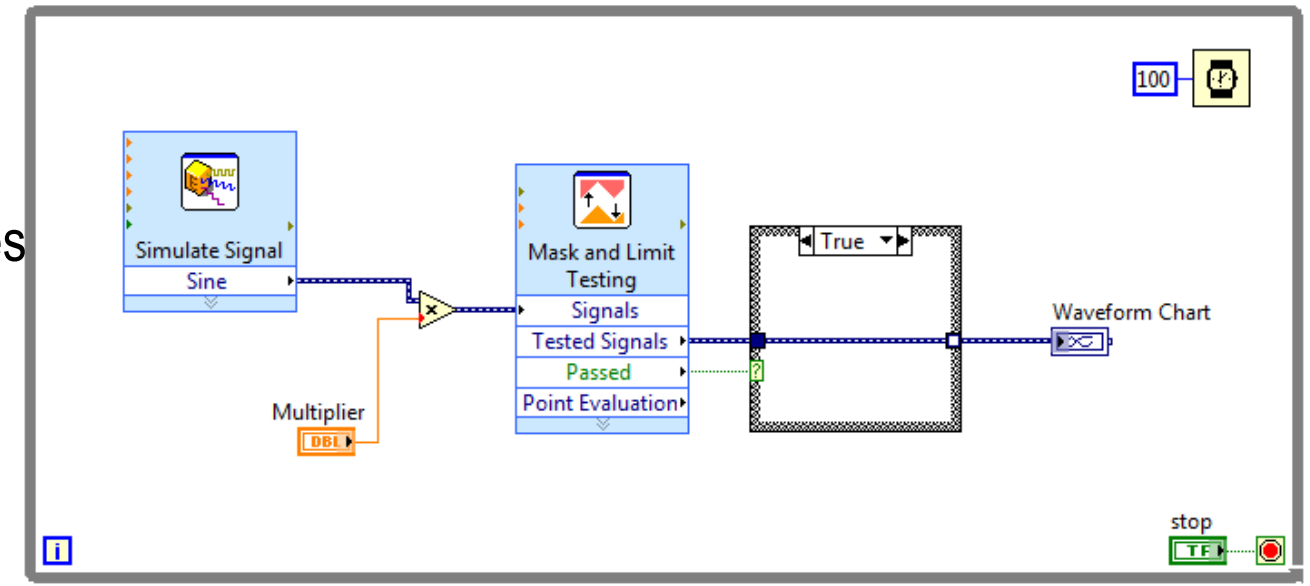
a) Constant
b) While Loop
c) Express VI
d) Function Node
e) Case Structure
f) Writing to file
g) Boolean Type wire
h) String type wire



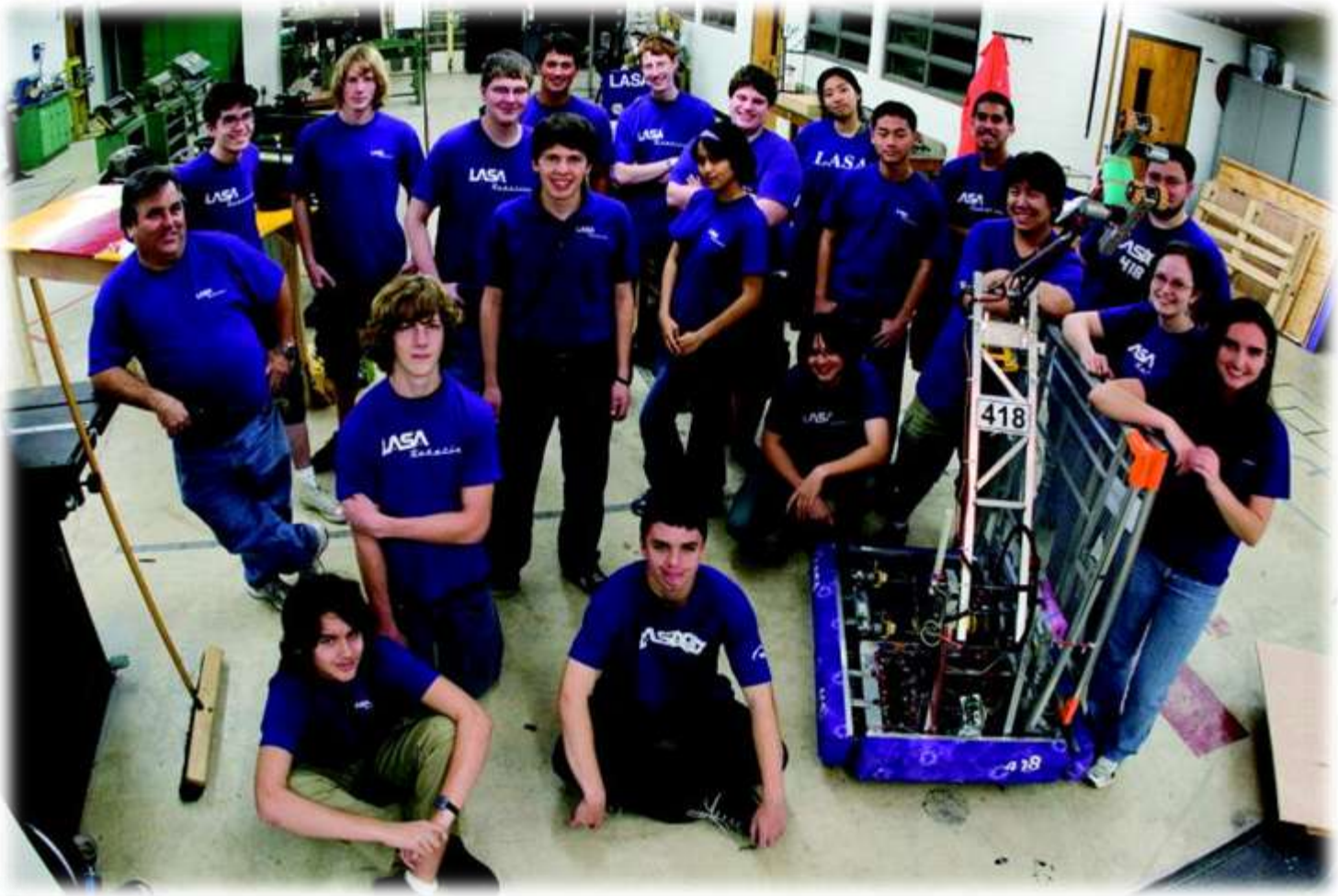
Quiz

Which elements is not visible on this block diagram (multiple answers)

a) Constant - yes
b) While Loop - yes
c) Express VI - yes
d) Function Node - yes
e) Case Structure - yes
f) Writing to file - no
g) Boolean Type wire - yes
h) String type wire - no



Use LabVIEW for your next project!



Project Assignment

- You will receive an email with a project assignment within 24 hours of this webcast
- It will contain the description of the project, hints, and information on how to submit your answer
- You should be able to finish it just by using the knowledge acquired during this webcast (and hints)
- You have time until end of May 2012 to send it back
- Only people who submit a working project will be eligible for prize drawing
- Students – please attach a scan of your student ID to get the serial number for a full LabVIEW Student Edition
- **If you don't receive the email, please contact us at** academic.projects@ni.com

Next Steps

- Learn more about LabVIEW: ni.com/gettingstarted/labviewbasics/
- See how LabVIEW is used in industry, research, and education: ni.com/casestudies
- Find more resources for students and educators at: <country>.ni.com/academic