



LabVIEW Lab(1)

Design of Measurement Systems



Ahmed Okasha
a.okasha@eng.asu.edu.eg

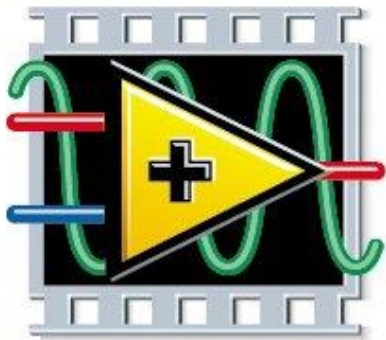
LAB Goals:

- ▶ Understand the capabilities and the functionalities of LabVIEW.
- ▶ Give general overview on LabVIEW graphical interface.
- ▶ Use LabVIEW as a programming language.



What is LabVIEW ?

- ▶ LabVIEW is a graphical programming environment used by millions of engineers and scientists to develop sophisticated measurement, test, and control systems using intuitive graphical icons and wires that resemble a flowchart



NATIONAL INSTRUMENTS

LabVIEW™

Certified Developer



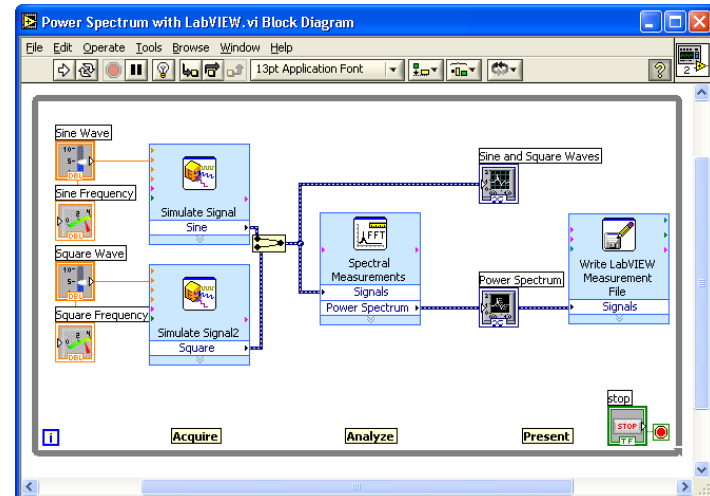
LabVIEW Programs Are Called Virtual Instruments (VIs)

▶ Front Panel

- Controls = Inputs
- Indicators = Outputs

▶ Block Diagram

- Accompanying “program” for front panel
- Components “wired” together

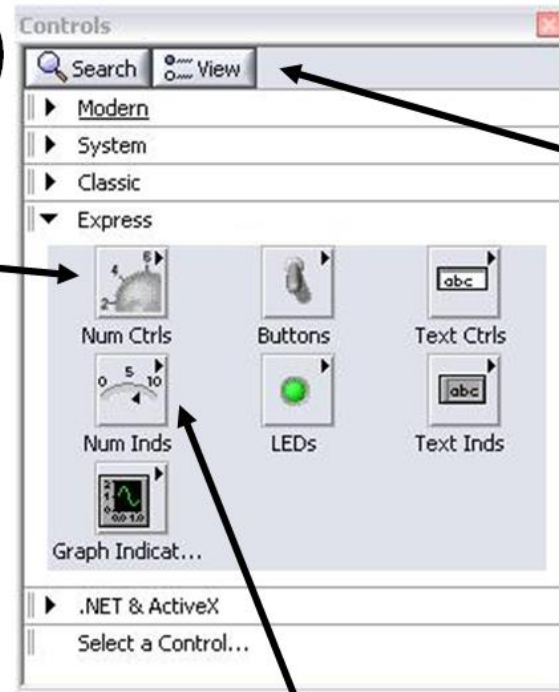
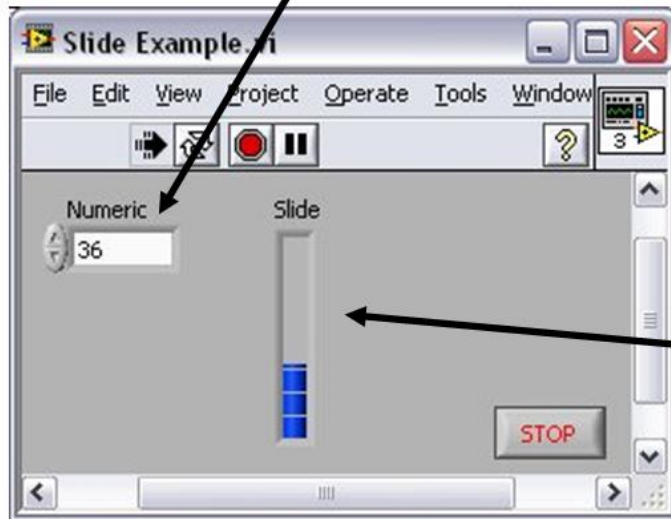


VI Front Panel

Controls Palette (Controls & Indicators) (Place items on the Front Panel Window)

**Control:
Numeric**

**Customize
Palette
View**

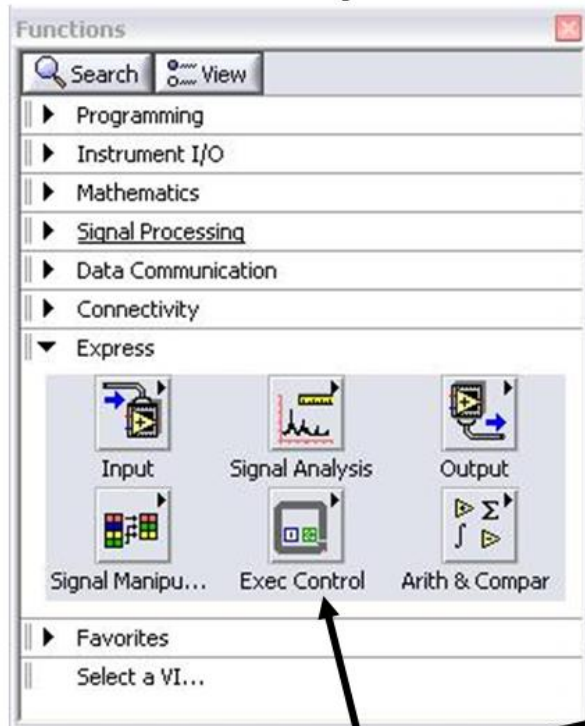


**Indicator:
Numeric Slide**

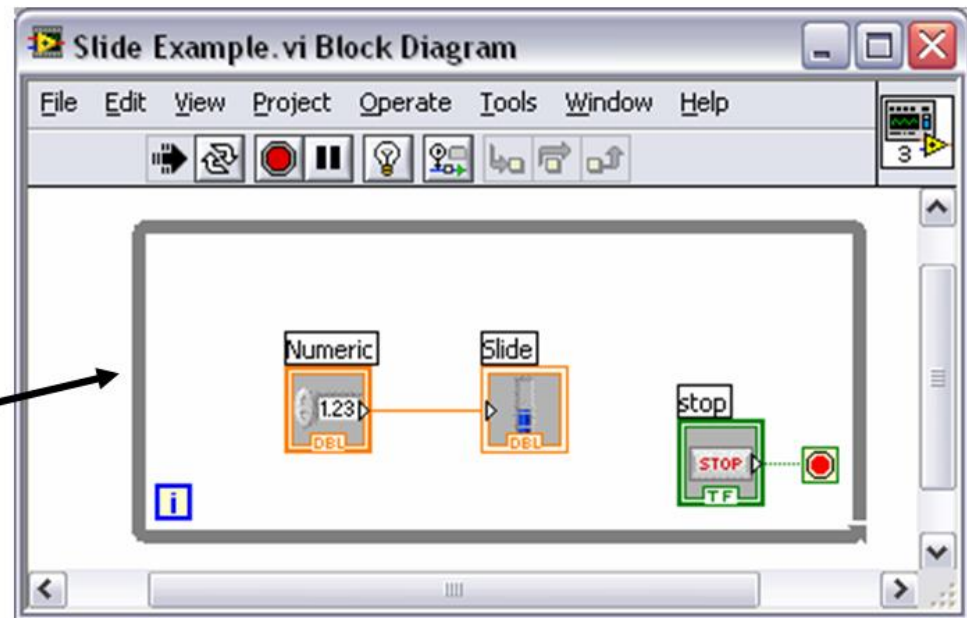


VI Block Diagram

Functions (and Structures) Palette



(Place items on the
Block Diagram Window)



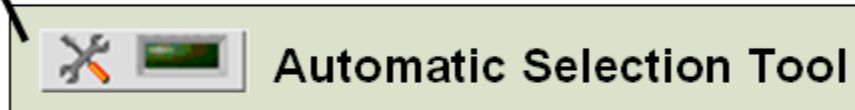
**Structure:
While Loop**

Tools Palette

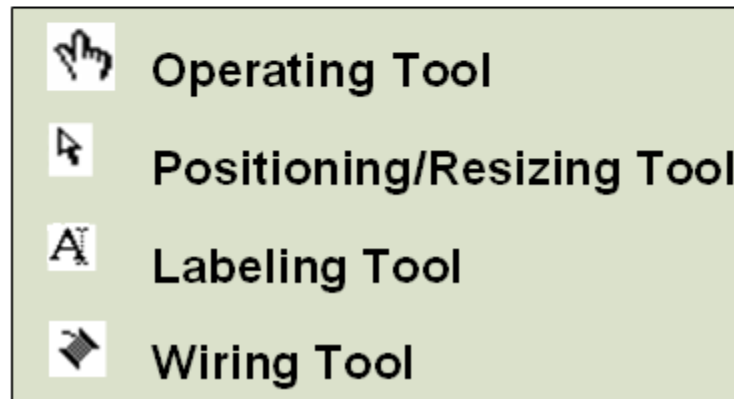
Tools Palette



- Recommended: Automatic Selection Tool
- Tools to operate and modify both front panel and block diagram objects



Automatically chooses among the following tools:



Status Toolbar



Run Button



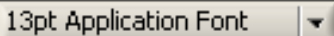
Continuous Run Button



Abort Execution



Pause/Continue Button



Text Settings



Align Objects



Distribute Objects



Reorder



Resize front panel objects

Additional Buttons on the Diagram Toolbar



Execution Highlighting Button



Step Into Button



Step Over Button

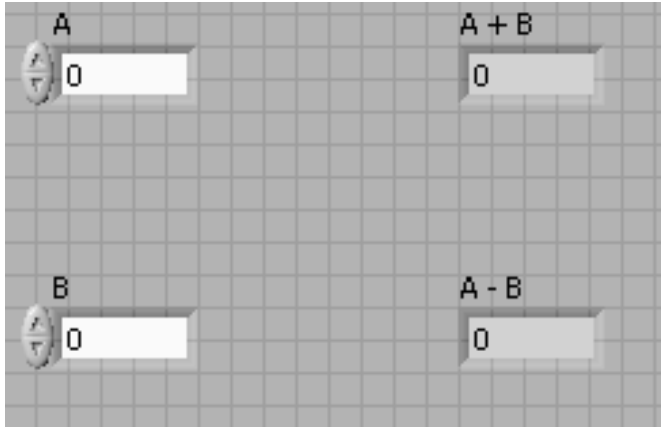


Step Out Button



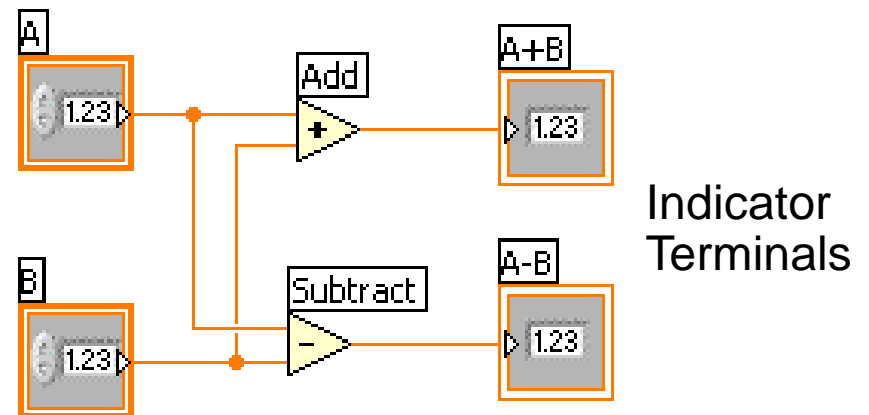
Creating a VI

▶ Example (I)



Control
Terminals

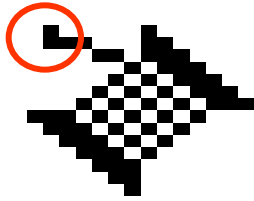
Block Diagram Window



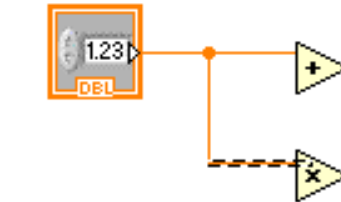
Indicator
Terminals

Wiring Tips – Block Diagram

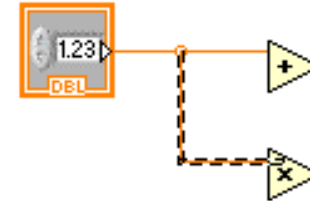
Wiring “Hot Spot”



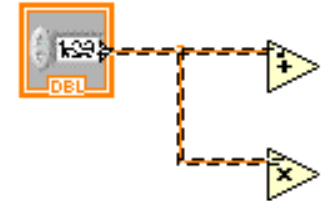
Click To Select Wires



single-click

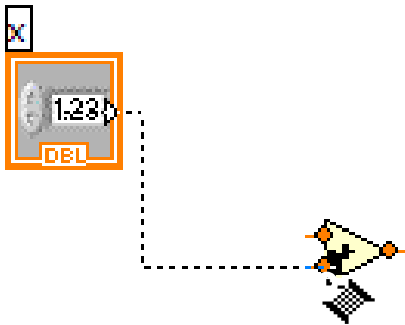


double-click

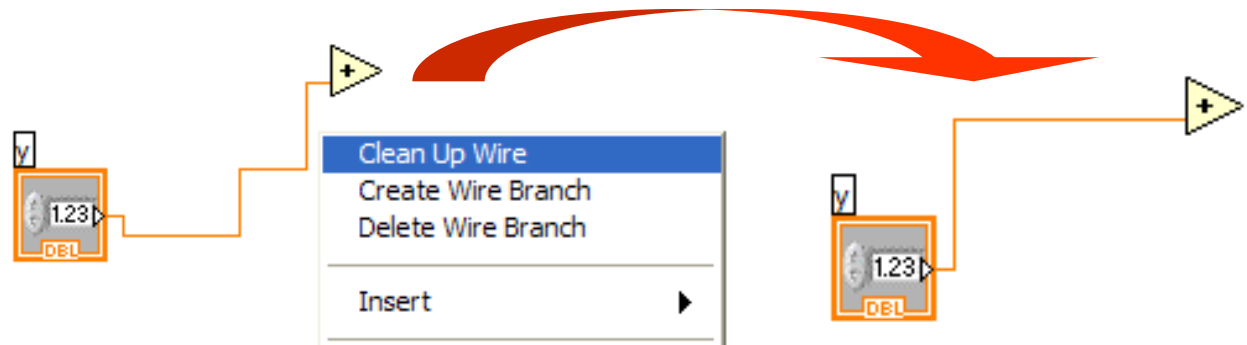


triple-click

Use Automatic Wire Routing



Clean Up Wiring



Debugging Techniques

- Finding Errors



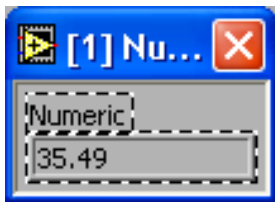
Click on broken Run button
Window showing error appears

- Execution Highlighting



Click on Execution Highlighting button; data flow is animated using bubbles. Values are displayed on wires.

- Probe



Right-click on wire to display probe and it shows data as it flows through wire segment



You can also select Probe tool from Tools palette and click on wire

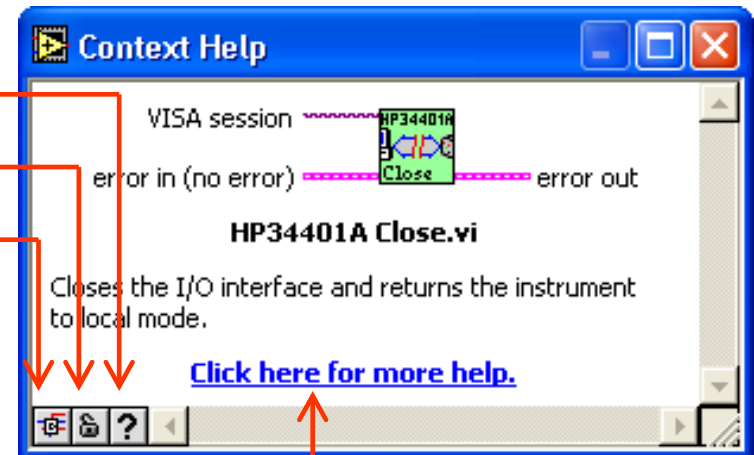


Help Options

To display the **Context Help** window, select **Help»Show Context Help** or press the <Ctrl-H> keys.

Context Help

- Online help
- Lock help
- Simple/Complex Diagram help
- Ctrl + H



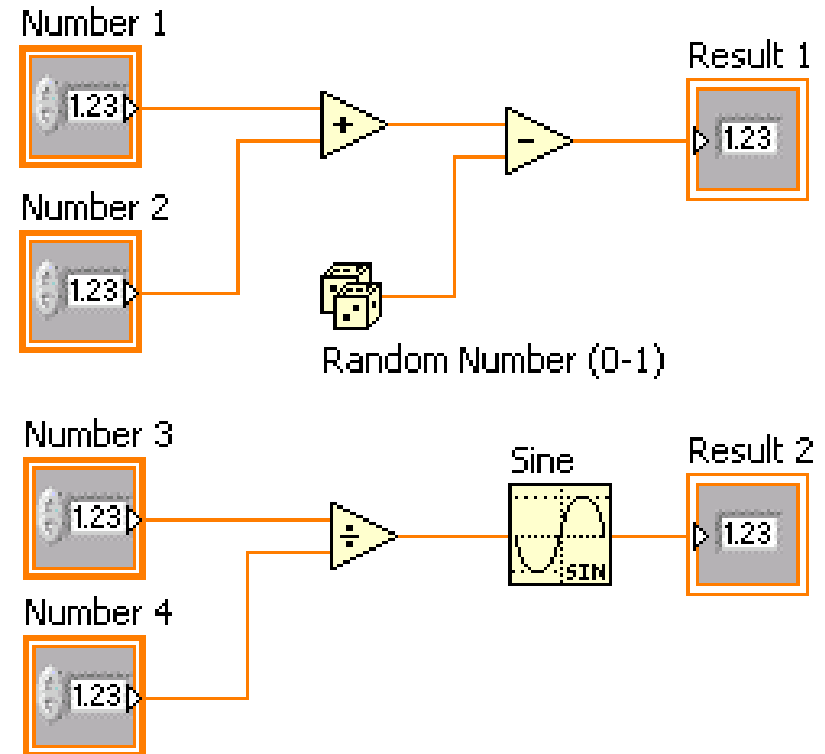
Online reference

- All menus online
- Pop up on functions in diagram to access online info directly

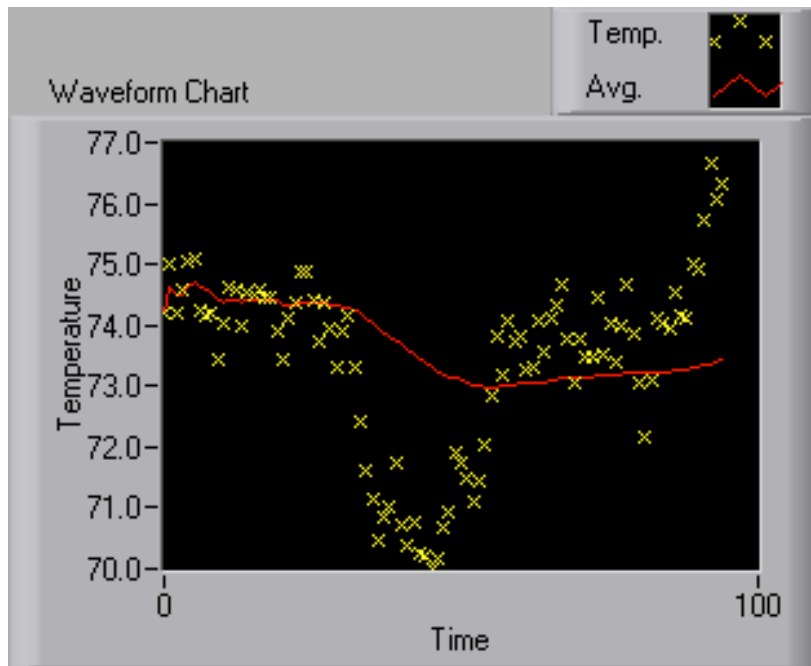
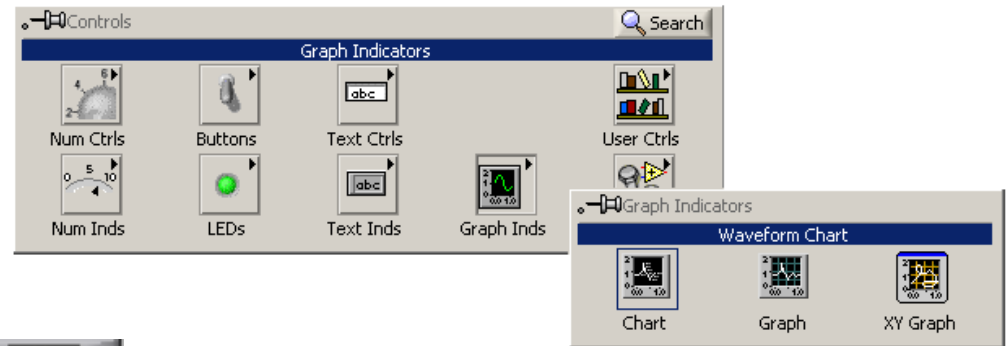


Dataflow Programming

- Block diagram executes dependent on the flow of data; block diagram does NOT execute left to right
- Node executes when data is available to ALL input terminals
- Nodes supply data to all output terminals when done



Waveform Charts



Waveform chart – special numeric indicator that can display a history of values

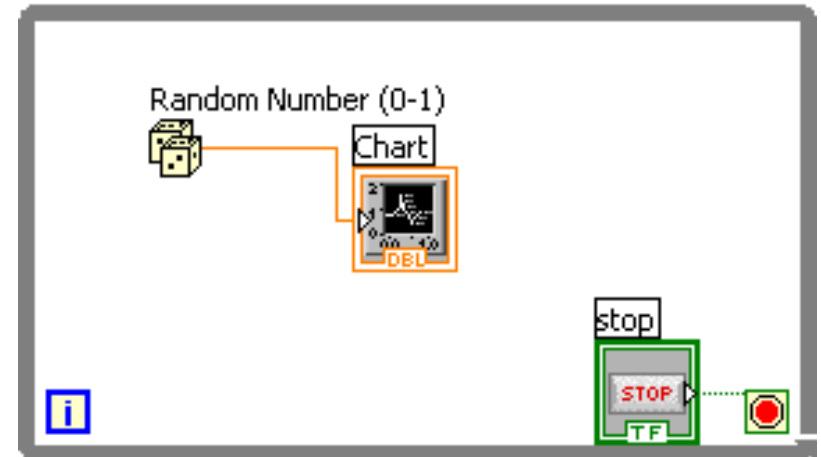
Controls >> Graph Indicators >> Waveform Chart

Loops

▶ While Loops

- ▶ Have Iteration Terminal
- ▶ Always Run at least Once
- ▶ Run According to Conditional Terminal

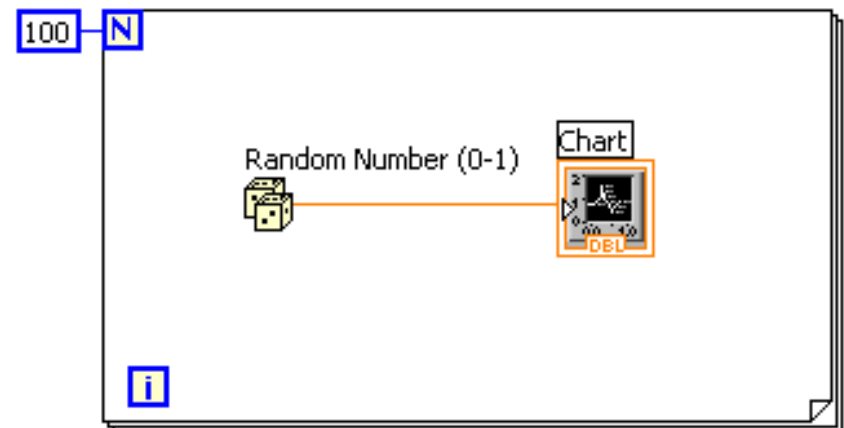
While Loop



▶ For Loops

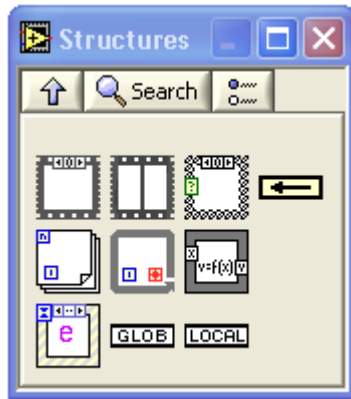
- ▶ Have Iteration Terminal
- ▶ Run According to input N of Count Terminal

For Loop

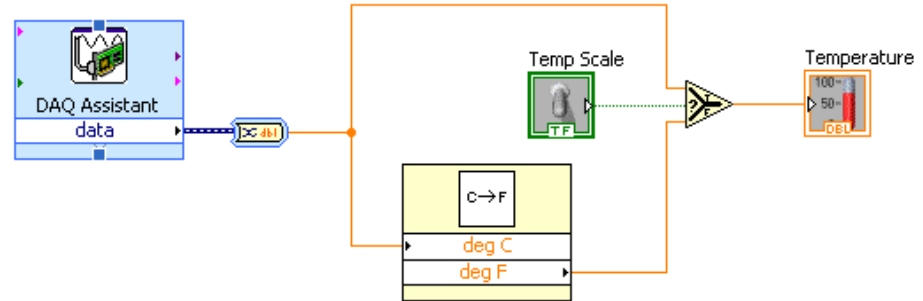


Loops (cont.)

1. Select the loop



2. Enclose code to be repeated



3. Drop or drag additional nodes and then wire

