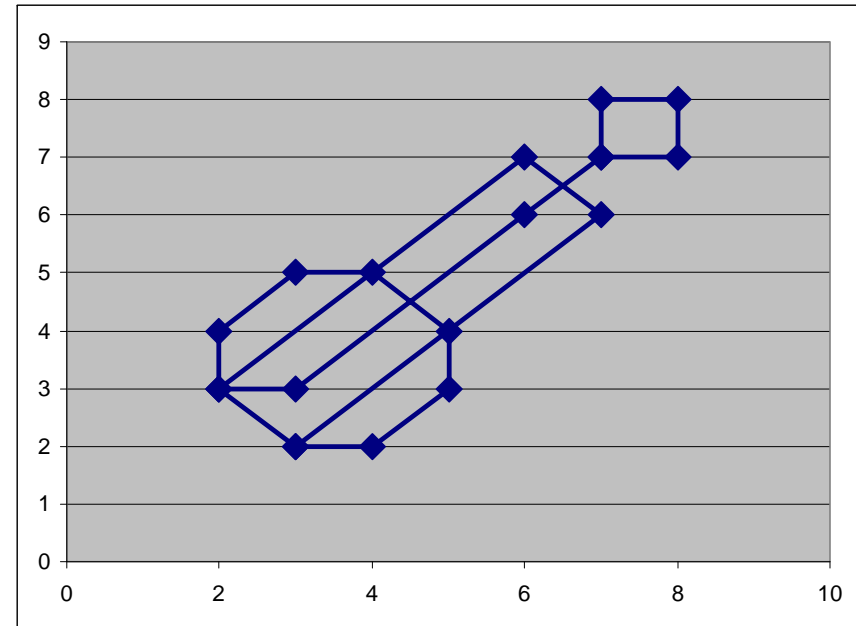
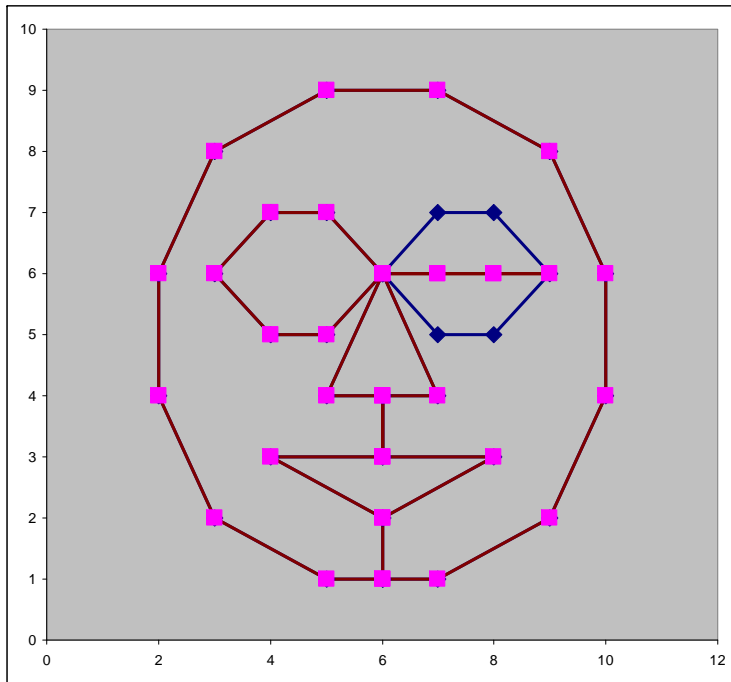


Today

- **4/5/11 Physics 262 Lecture 10**
- **Questions about Exams**
- **Homework**
 - Review your exams and review practice exam again.
- **Lab: Continue project**
 - Do pre-lab before lab. See last weeks handout.
 - Requires use of excel/matlab and some thought
 - See handout for more detail.
- **Lecture**
 - Labview Intro

Stills and Animated Movies



Show matlab animation

Show excel file

Show 2010 text files

“Cannon Fire”

```
for(ii=1:8)
  A=[3 2; 2 3; 2 4; 3 5
    4 5; 5 4; 5 3; 4 2
    3 2; 7 6; 6 7; 2 3
    3 3; 6 6;
    6+ii 6+ii;
    6+ii 7+ii;
    7+ii 7+ii;
    7+ii 6+ii;
    6+ii 6+ii];
  figure(3);clf;hold on
  N=length(A(:,1))
  for i=1:N-1;
    plot(A((i:i+1),1),A((i:i+1),2),'r-','LineWidth',10/sqrt((A(i,1)-A(i+1,1))^2+(A(i,2)-A(i+1,2))^2))
    tic
    axis([1 16 1 16]);
    while(toc<.005);end;
  end
end
```

LabVIEW

National Instruments Labview has been around for over 20 years:

1986 LabVIEW 1.0 for Macintosh

1990 LabVIEW 2.0 for Macintosh

1992 LabVIEW for Windows

Labview is a widely used tool in Physics and Engineering labs

Nearly all test, measurement, and control applications can be divided into 3 main components: the ability to

1. *Acquire data with controlled conditions.*
2. *Analyze data.*
3. *Present data.*

Acquire/Control with LabVIEW

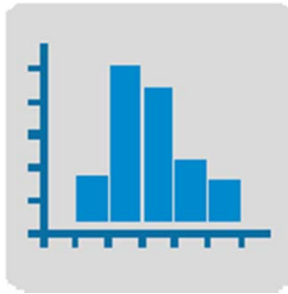


LabVIEW integrates with NI hardware and thousands of I/O devices from hundreds of different vendors.

LabVIEW can control instruments and acquire data using the following devices/ busses (and many more):

- GPIB, Serial Port, Parallel Port, Ethernet, VXI, PXI Instruments
- Data Acquisition (DAQ)
- PCI eXtensions for Instrumentation (PXI)
- Image Acquisition (IMAQ)
- Motion Control
- Real-Time (RT) PXI
- PLC (through OPC Server)
- PDA
- Modular Instruments

Analyze with LabVIEW



Analysis is built in to the LabVIEW development environment.

LabVIEW includes the following tools to help you analyze your data:

- More than 400 measurement analysis functions for Differential Equations, Optimization, Curve Fitting, Calculus, Linear Algebra, Statistics, etc.
- Express VIs (Virtual Instruments) specifically designed for measurement analysis, including filtering and spectral analysis
- Signal Processing VIs for Filtering, Windowing, Transforms, Peak Detection, Harmonic Analysis, Spectrum Analysis, etc.

Present with LabVIEW



Presentation with LabVIEW can be done realtime or post-processed with Labview.

LabVIEW includes the following tools to help you present your data:

- On your machine — Graphs, Charts, Tables, Gauges, Meters, Tanks, 3D Controls, Picture Control, 3D Graphs (Windows Only), Report Generation (Windows Only)
- Over the Internet — Web Publishing Tools, Datasocket (Windows Only), TCP/IP, VI Server, Remote Panels, Email
- Enterprise Connectivity Toolset — SQL Tools (Databases), Internet Tools (FTP, Telnet, HTML)

Text based versus Graphical based test and measurement

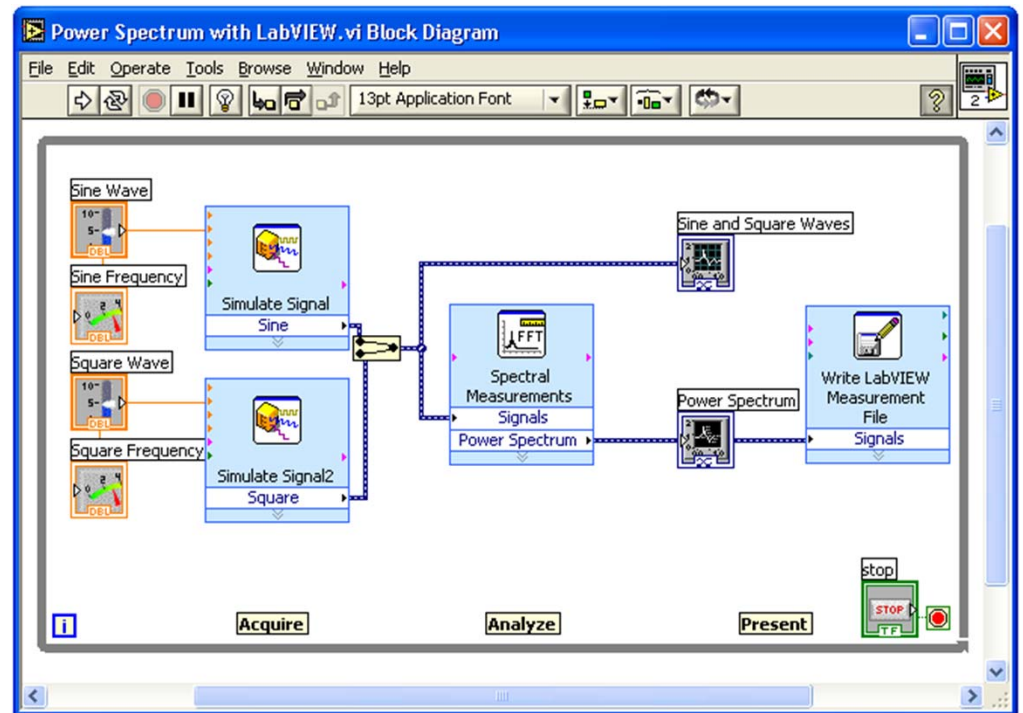
Text based programming

```
//C++ Calculator Program
#include<iostream.h>
#include<stdlib.h>

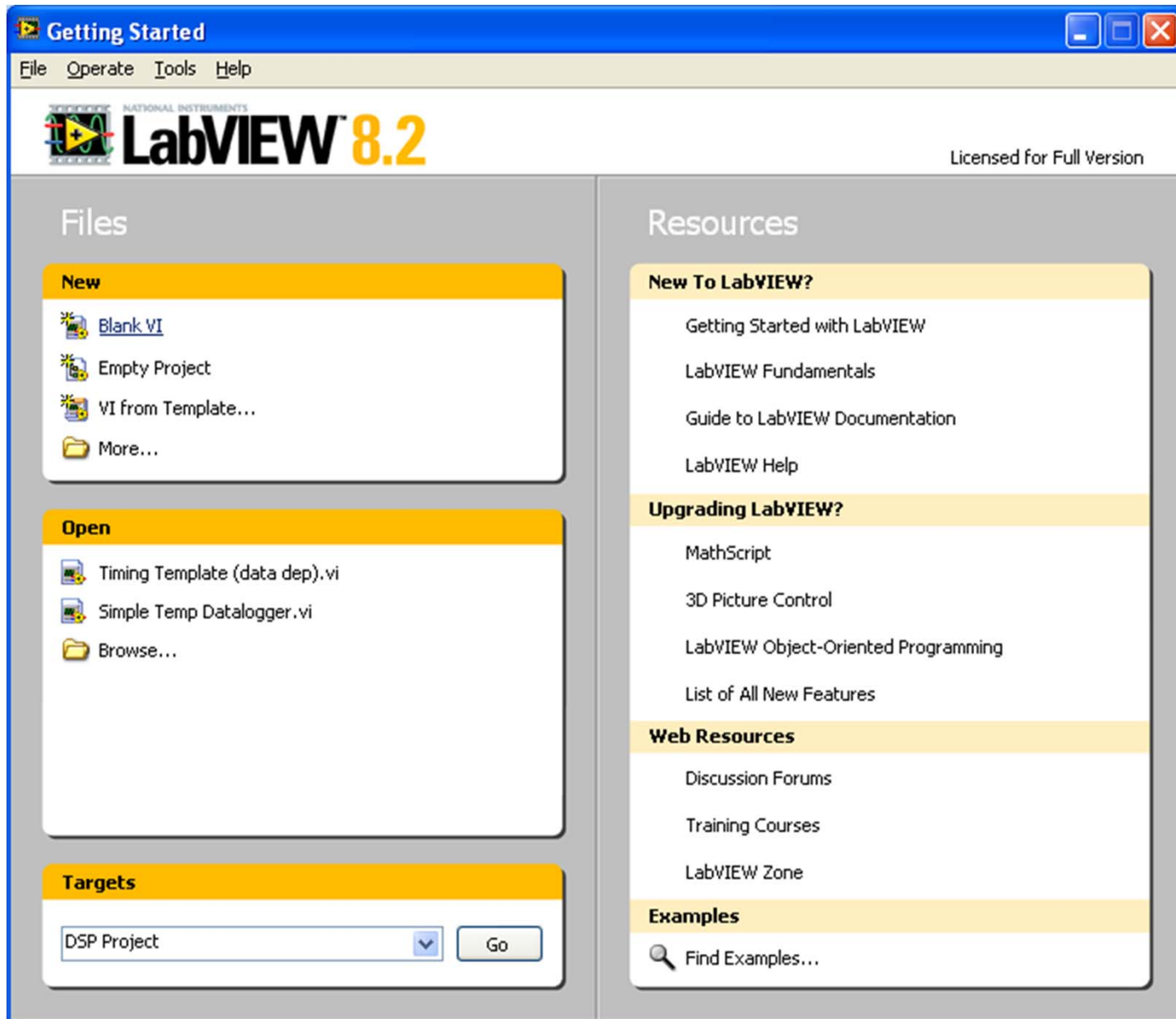
int main()
{
    int a, b;
    cout << "Enter two integers to add: ";
    cin >> a >> b;
    cout << (a + b);
    cout << "Enter two integers to subtract: ";
    cin >> a >> b;
    cout << (a - b);
    cout << "Enter two integers to multiply: ";
    cin >> a >> b;
    cout << (a * b);
    return 0;
}
```

C++

Graphical based programming



Opening National Instruments>Labview



Virtual Instruments (VIs)

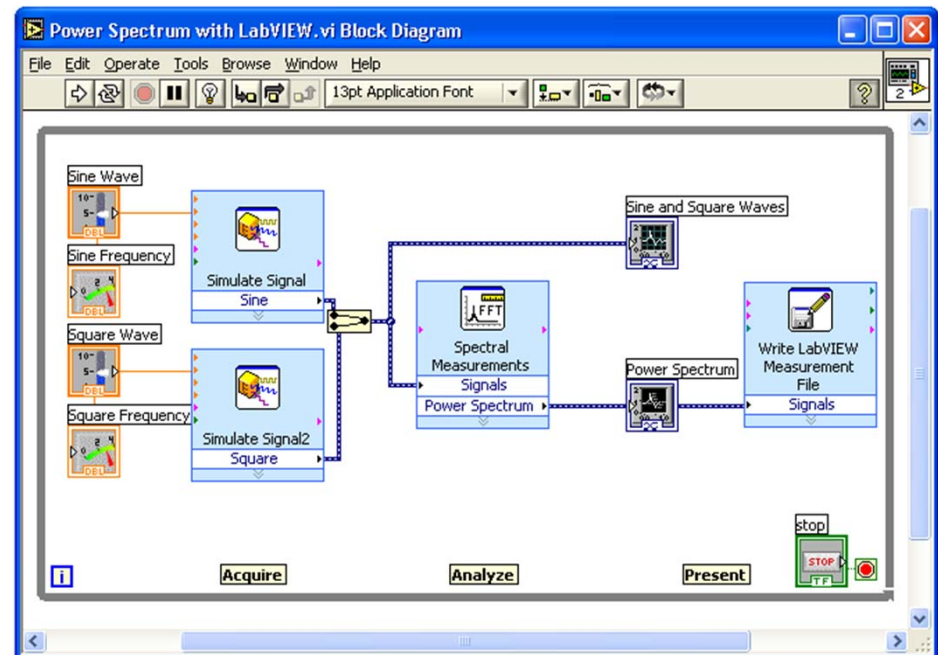
- Virtual instruments (VIs) have three main parts — the front panel, the block diagram, and the icon and connector pane

Front Panel

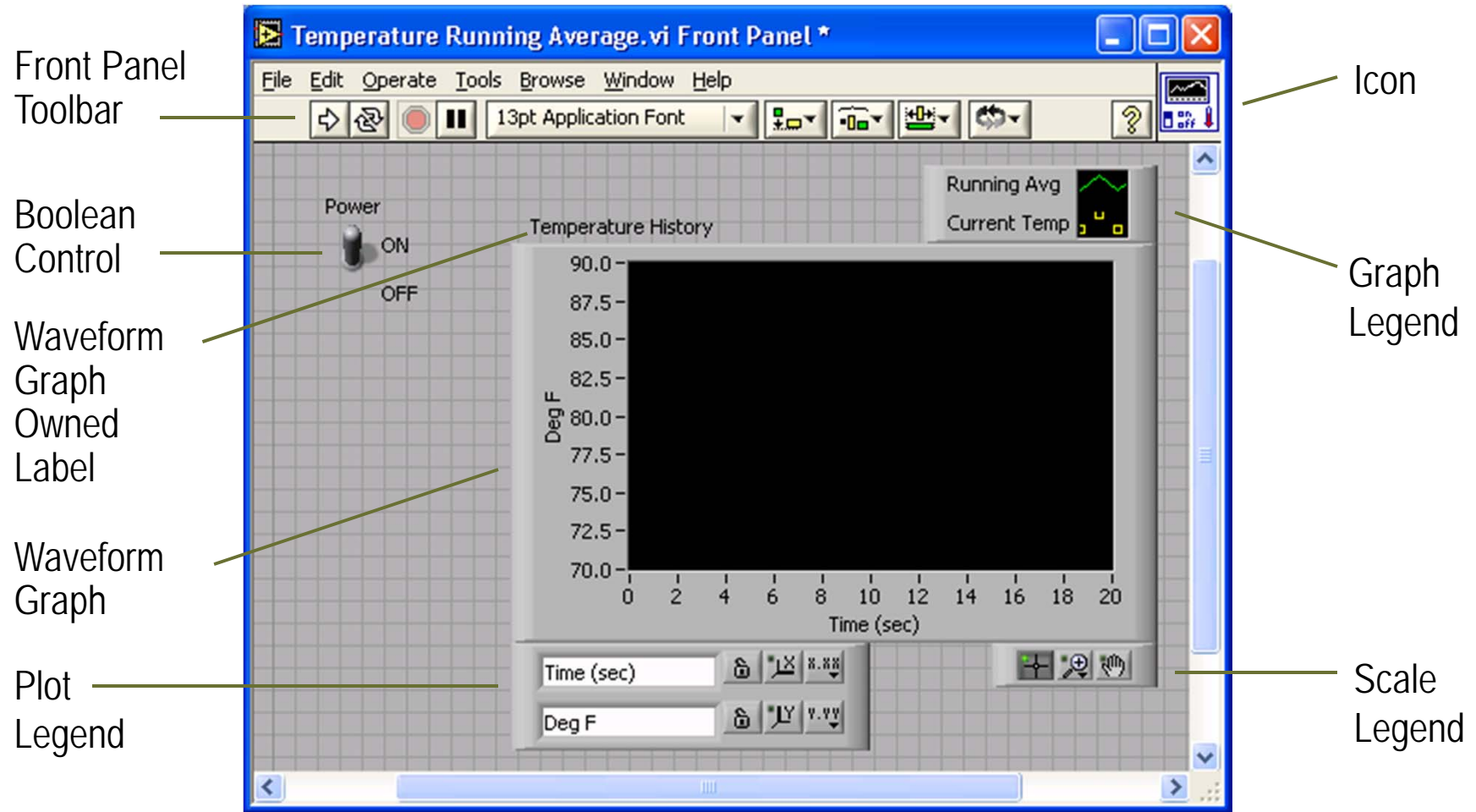
- Controls = Inputs
- Indicators = Outputs

Block Diagram

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



Front Panel Window



- The front panel is the user interface of a LabVIEW program and the block diagram is the executable code

Front Panel and Block Diagram Toolbars



Run button

Continuous Run button

Abort button

Pause/Continue button



Broken Run button

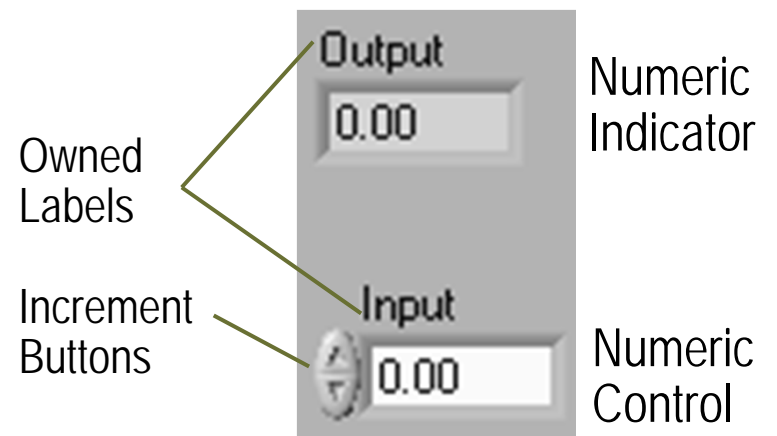
Creating a VI Front Panel

Build the front panel with controls (inputs) and indicators (outputs)

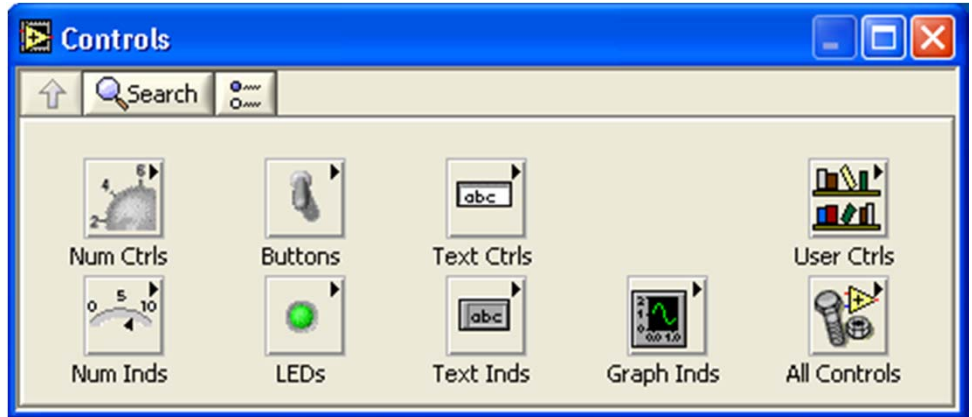


Boolean
Control

Boolean
Indicator



Front Panel – Controls Palette



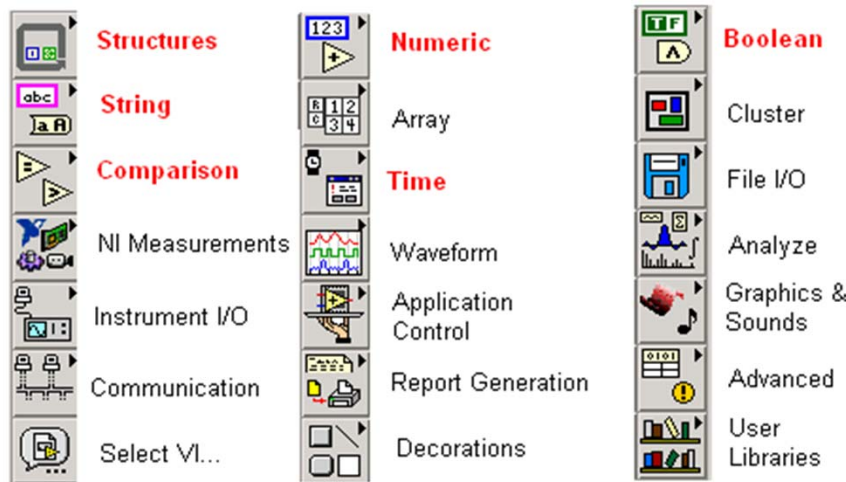
Controls Palette

Contains the most commonly used controls

Block Diagram – Functions Palette



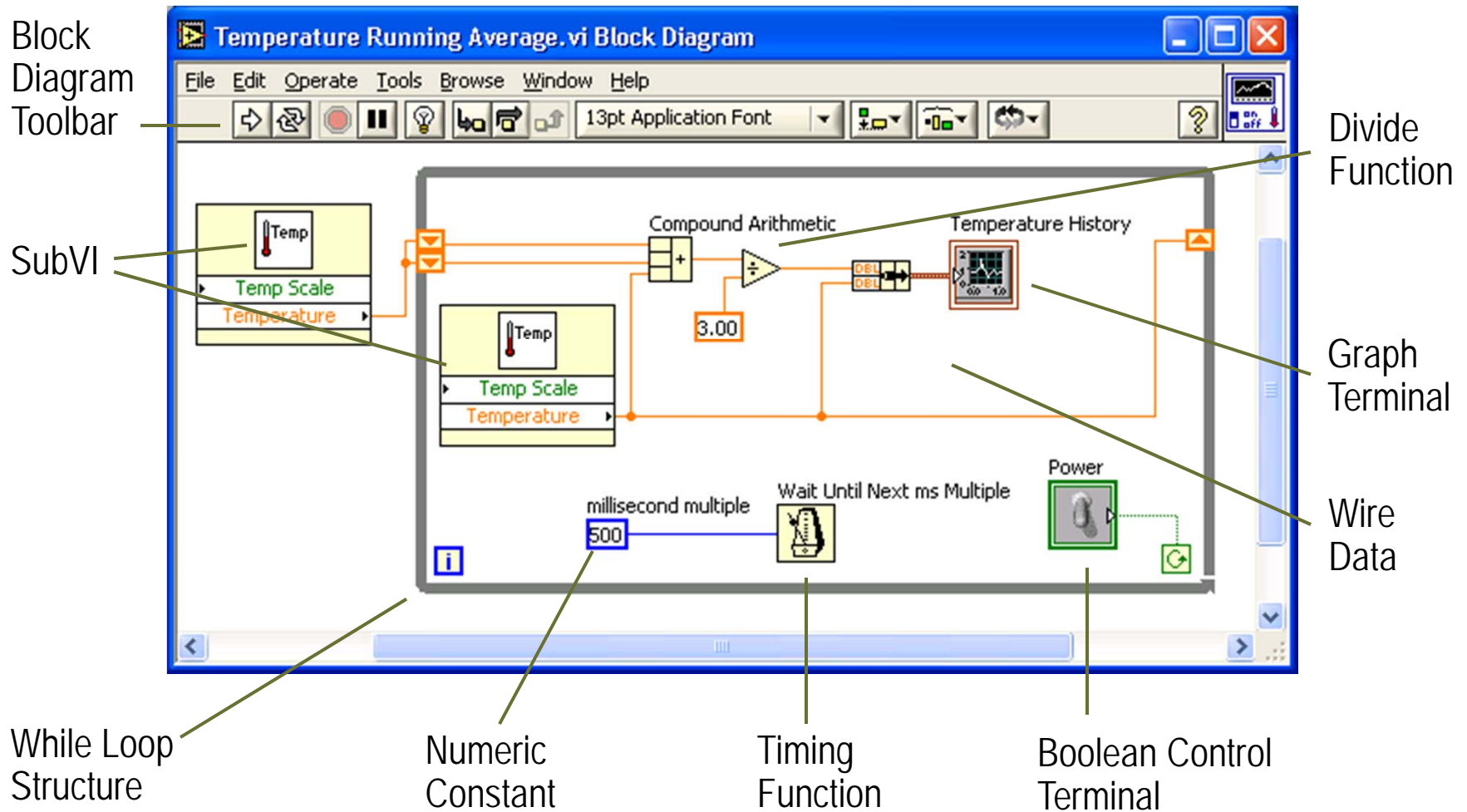
Functions Palette



Functions Palette

Contains the VIs and the most commonly used functions

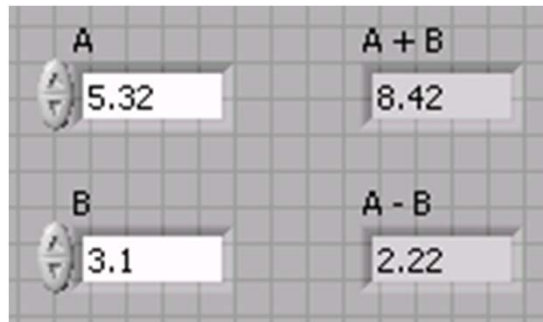
Block Diagram Window



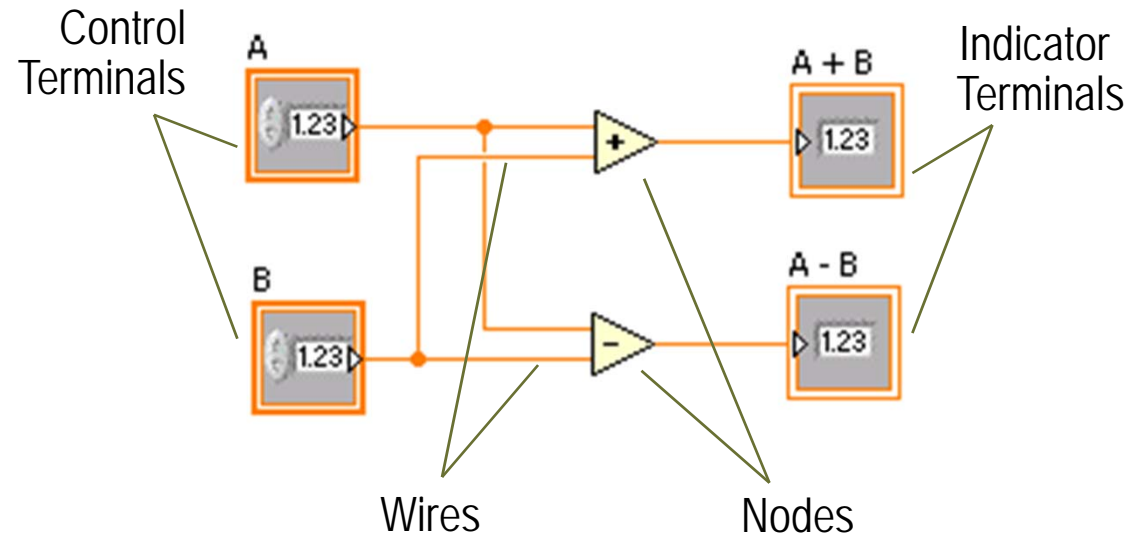
- The block diagram contains the graphical source code composed of nodes, terminals, and wires

Creating a VI Block Diagram

Front Panel



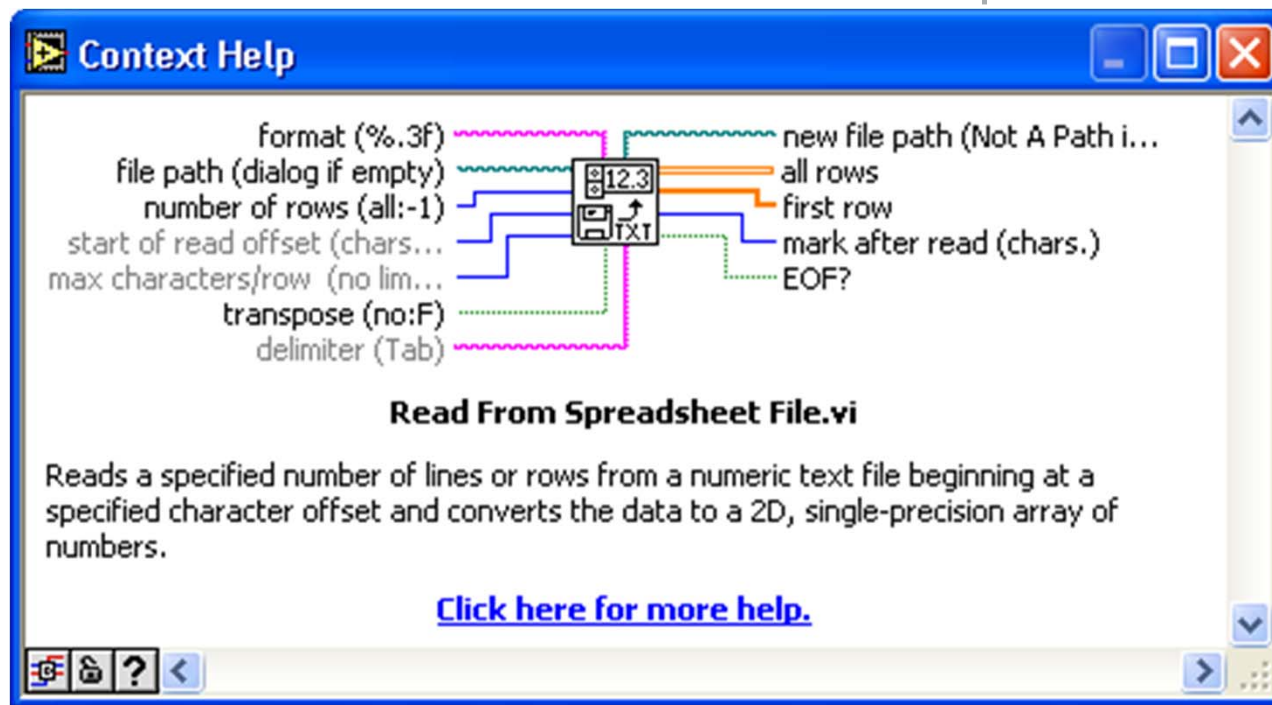
Block Diagram



- **Express VIs:** interactive VIs with configurable dialog page
- **Standard VIs:** modularized VIs customized by wiring
- **Functions:** fundamental operating elements of LabVIEW; no front panel or block diagram

Context Help

- To display the Context Help window, select **Help»Show Context Help**, press the <Ctrl-H> keys
- Move cursor over object to display help
- Connections:
Required – bold, Recommended – normal, Optional - dimmed



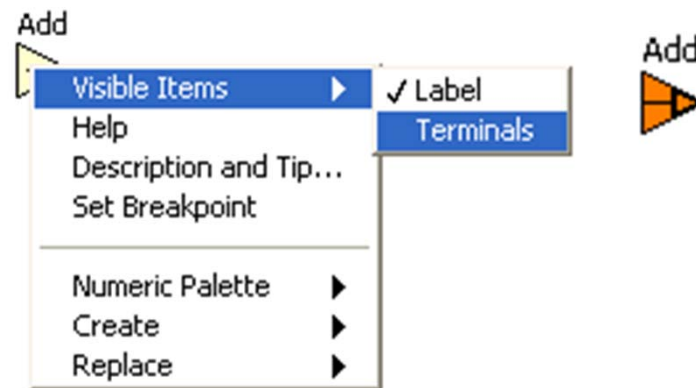
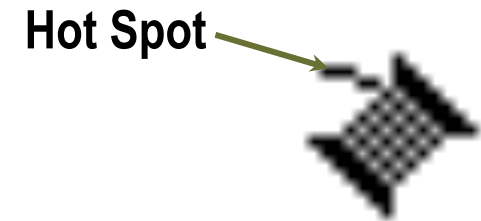
Tools Palette

- LabVIEW automatically selects the tool it thinks you need
- On both the front panel and the block diagram
- A “tool” is operating mode of the mouse cursor
- Use the tools to operate and modify front panel and block diagram objects
- To show the tools palette, select **Window»Show Tools Palette**



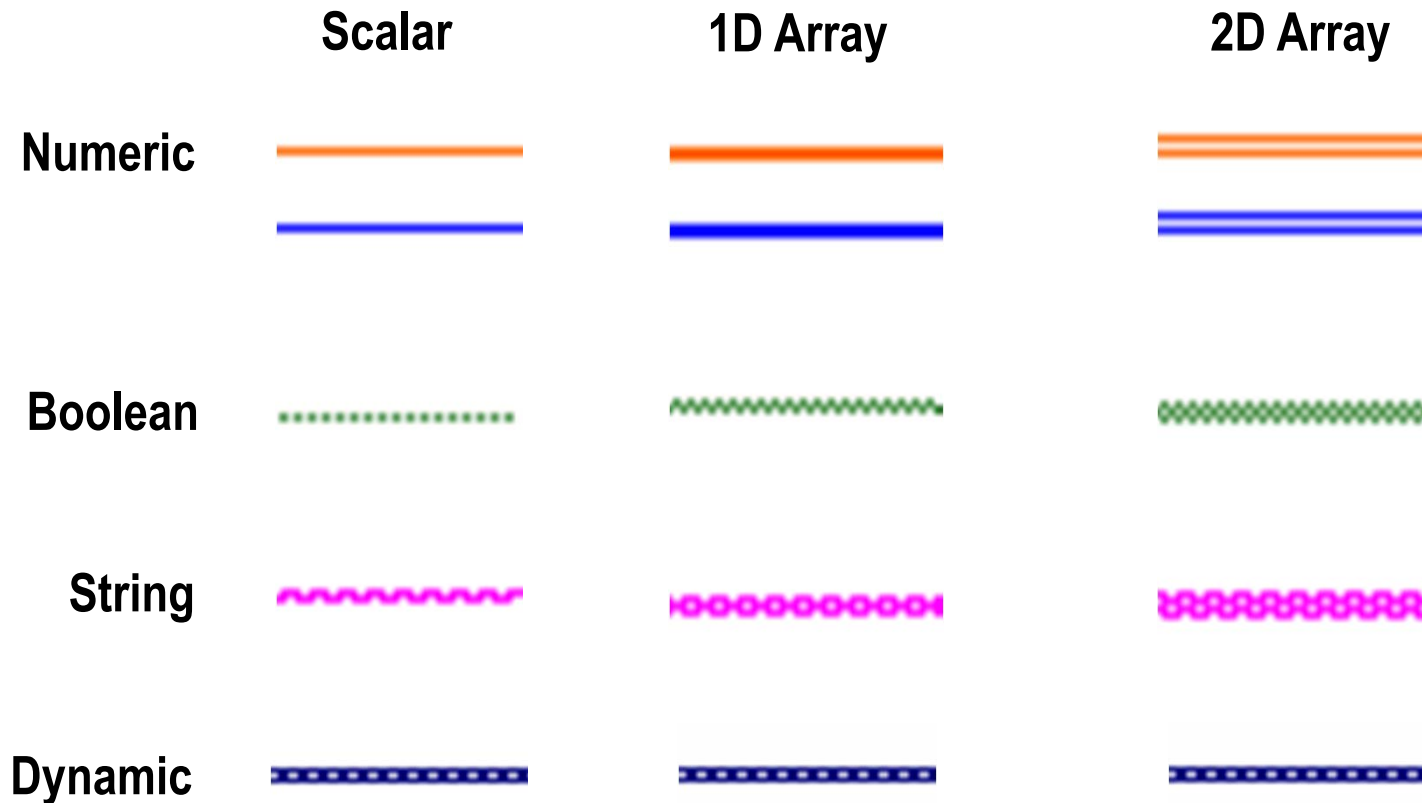
Wiring Techniques

- Automatic Wiring
- Use Context Help Window when wiring
- Right-click wire and select **Clean Up Wire**
- Automatic wire routing
- Right-click terminals and select **Visible Items»Terminals**



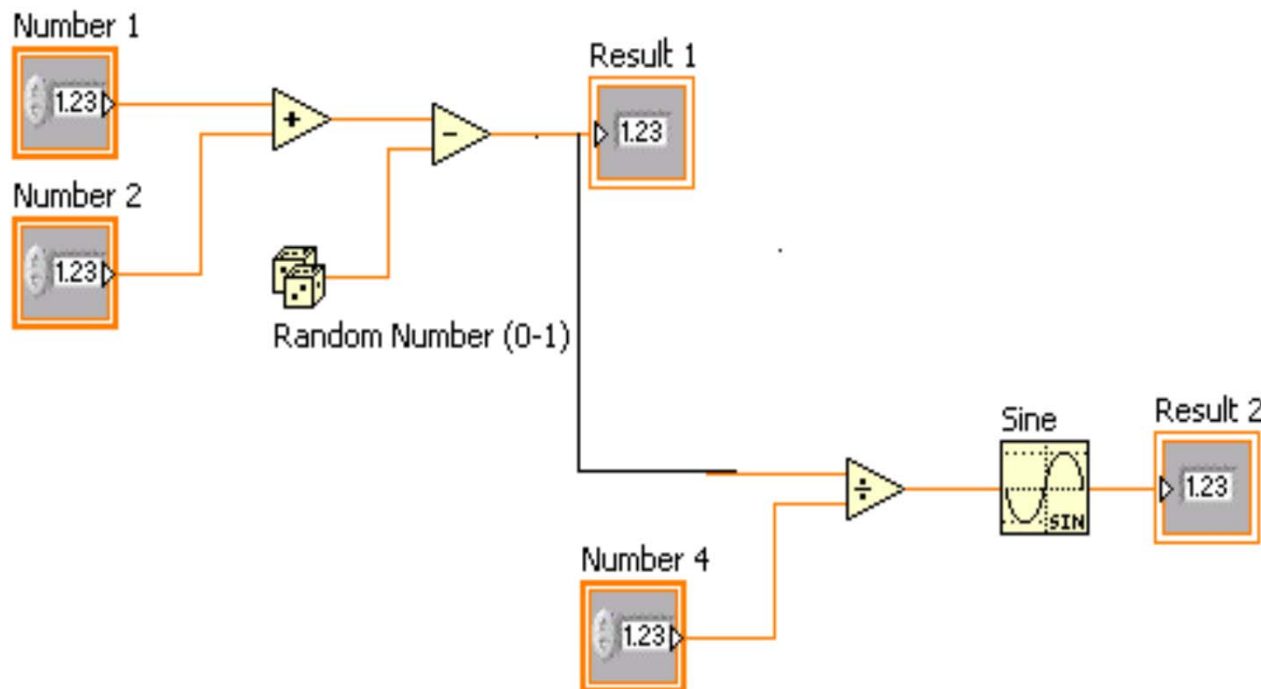
View the terminal connections to a function

Wiring the Block Diagram

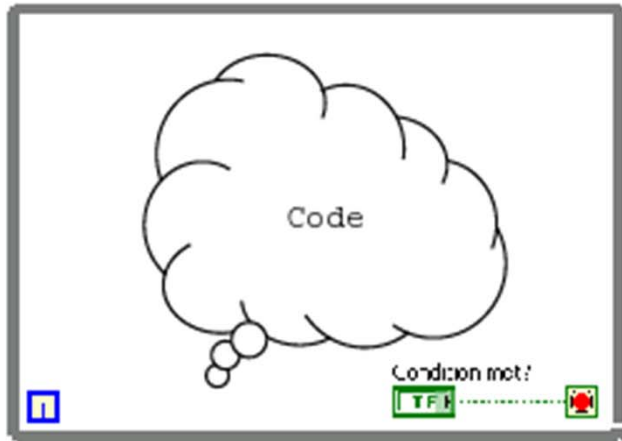


Dataflow Programming

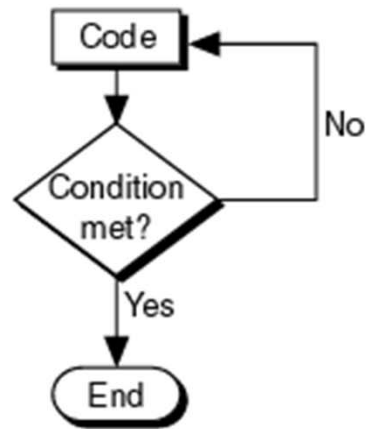
- Block diagram executes dependent on the flow of data; block diagram does NOT execute left to right
- Node executes when data is available to ALL input terminals
- Nodes supply data to all output terminals when done



While Loops



LabVIEW While Loop



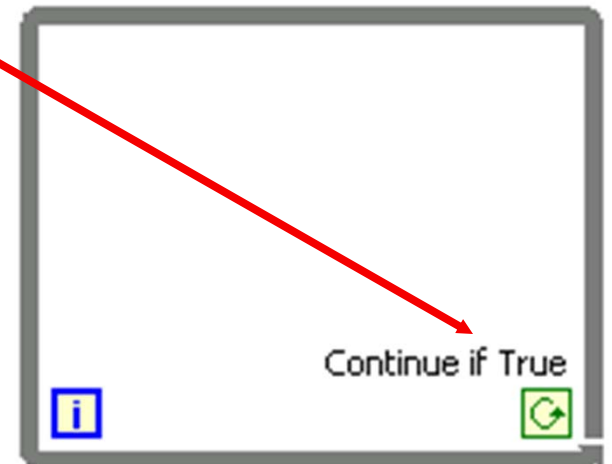
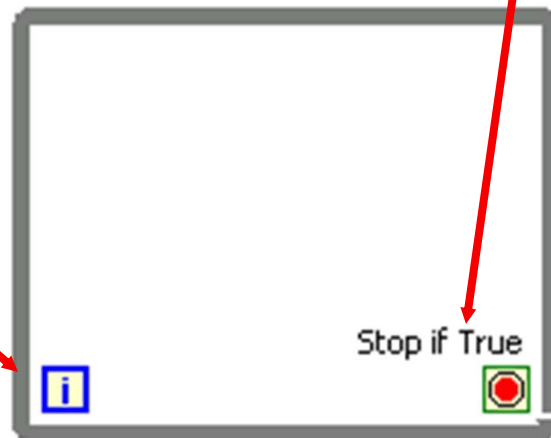
Flow Chart

```
Repeat (code);  
Until Condition met;  
End;
```

Pseudo Code

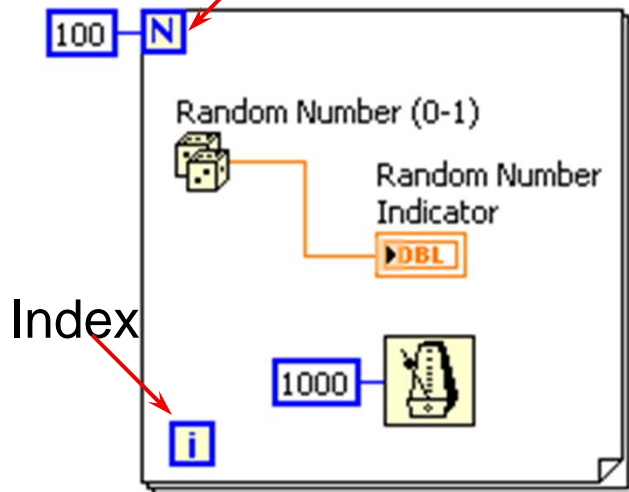
Conditional Terminal

Iteration Terminal

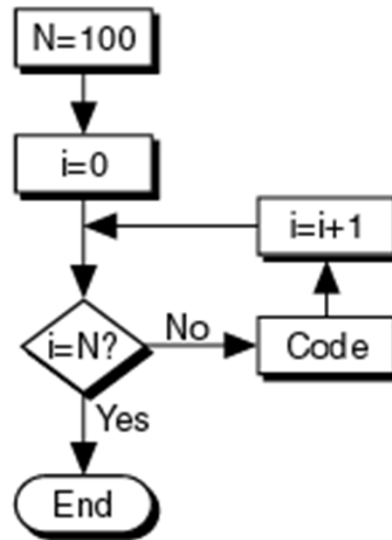


For Loops

Count terminal
(Numerical input)



LabVIEW For Loop



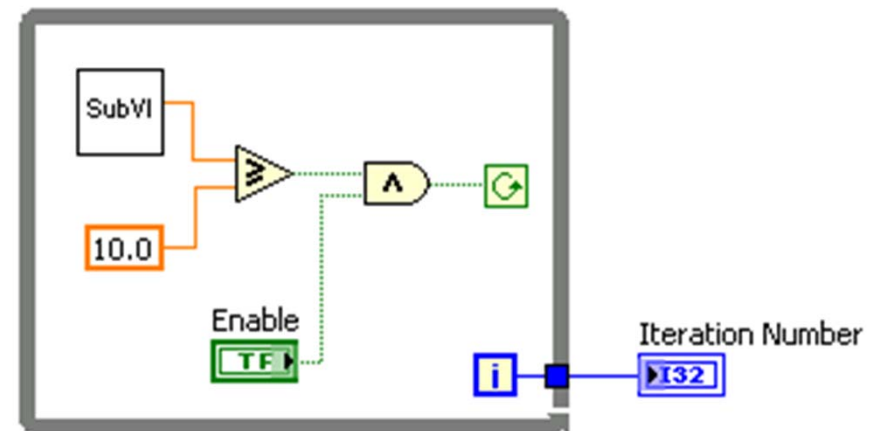
Flow Chart

```
N=100;  
i=0;  
Until i=N:  
    Repeat (code; i=i+1);  
End;
```

Pseudo Code

Structure Tunnels

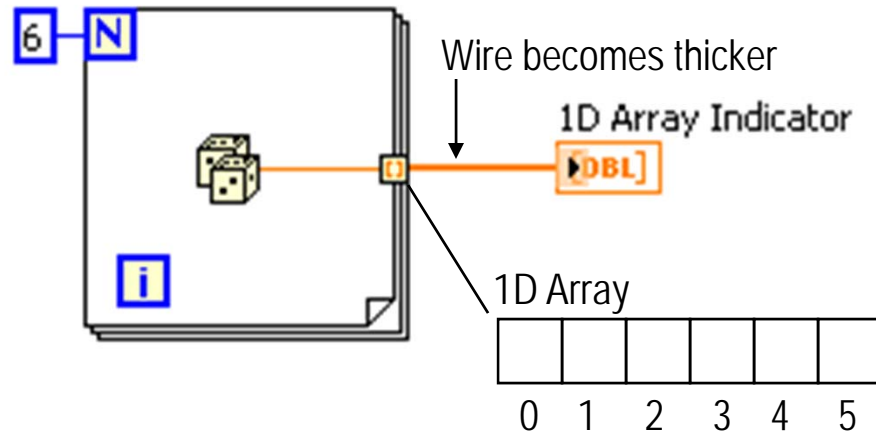
- Tunnels feed data into and out of structures.
- The tunnel is a block that appears on the border; the color of the block is related to the data type wired to the tunnel.
- When a tunnel passes data into a loop, the loop executes only after data arrive at the tunnel.
- Data pass out of a loop after the loop terminates.
- Array and indexes as they pass through tunnel, unless auto-indexing is disabled.



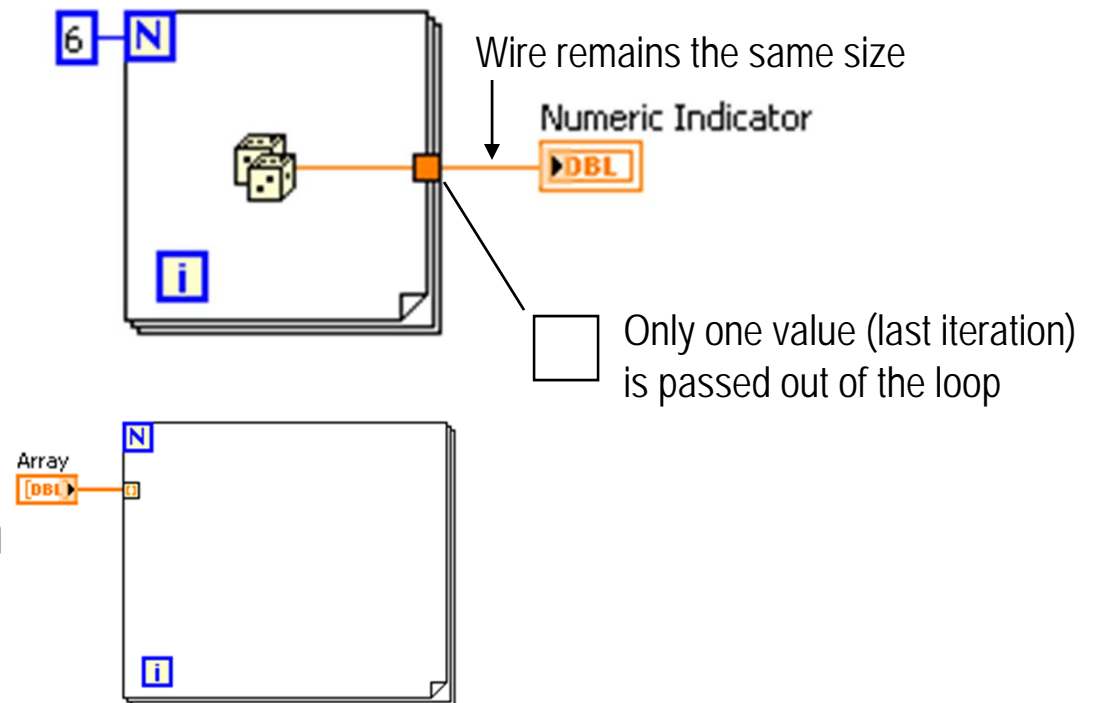
Auto-Indexing

- Loops can accumulate arrays at their boundaries with auto-indexing
- For Loops auto-index by default
- While Loops output the final value by default
- Right-click on tunnel and enable/disable auto-indexing
- Auto-indexing on Loop input converts arrays to indexed elements (default), which can be disabled at the node.

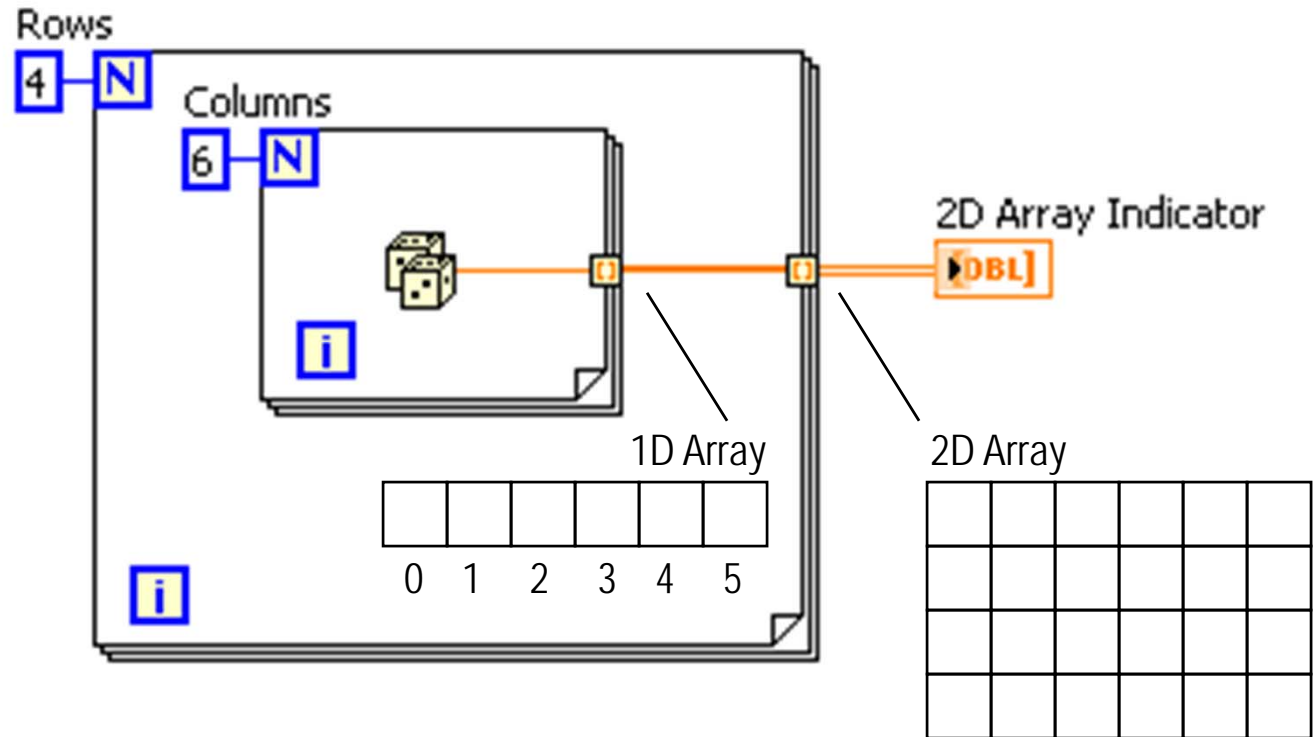
Auto-Indexing Enabled



Auto-Indexing Disabled



Creating 2D Arrays



- Inner loop creates column elements
- Outer loop stacks them into rows