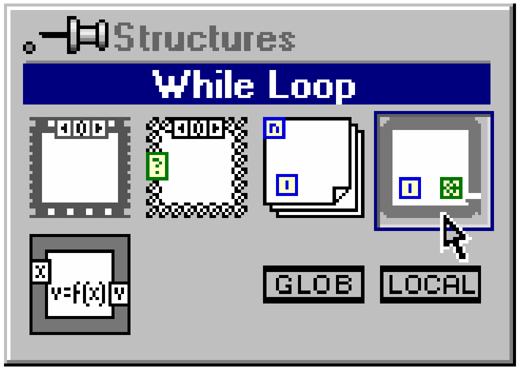
Lecture 3
(week Dec. 9-15)
Expanding the Features of a VI

You Will Learn:

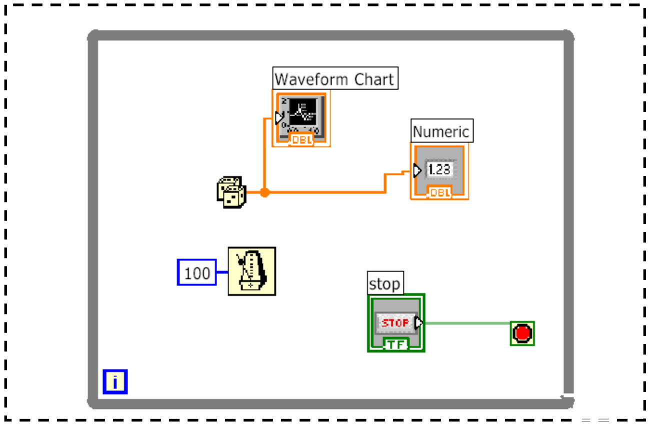
- **A. Structures--While Loop, for loop (How to use shift registers), case structure, sequence, formula node**
- **B. Arrays and Clusters**
- **C. Charts and Graphs**
- **D. Strings**

A. Structures--While Loop

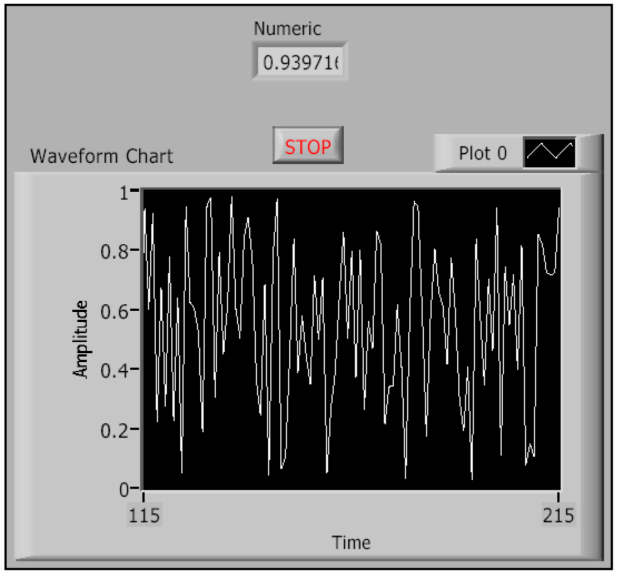
1. Select While Loop



2. Enclose code to be repeated



3. Drop or drag additional nodes and then wire



Do

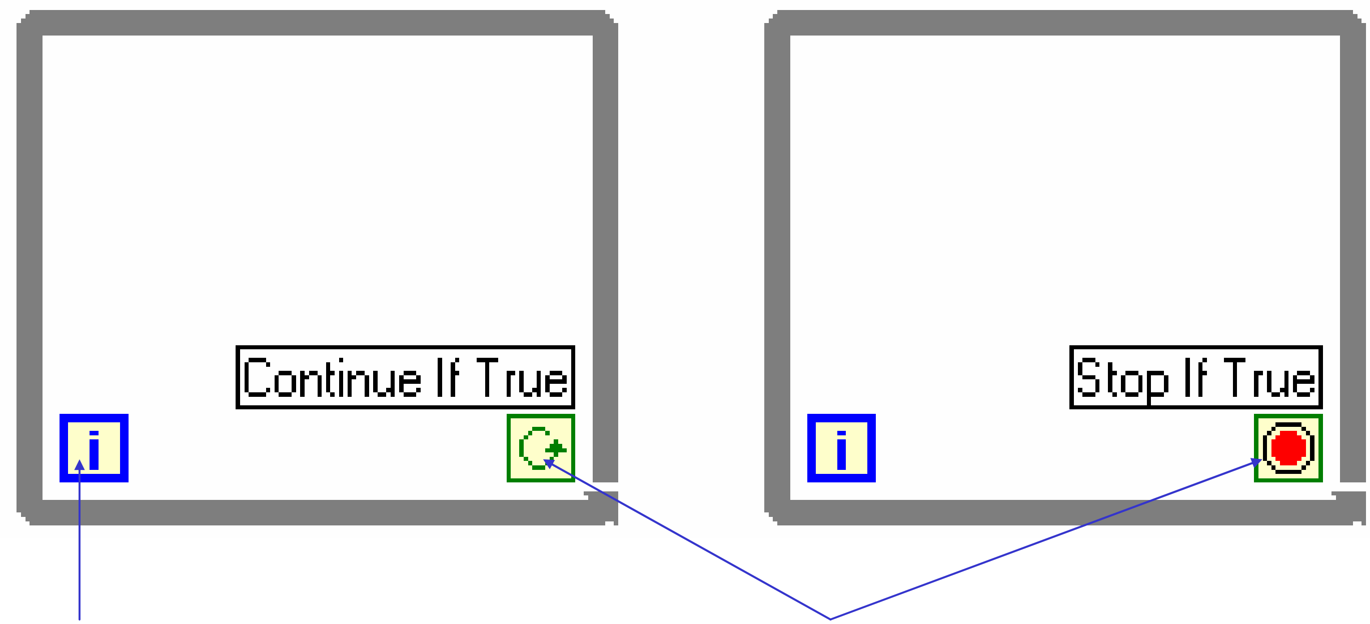
(Execute diagram inside Loop)

stop if Condition is TRUE

Save as while loop.vi

Select the Loop Condition

- **Right-click on Conditional Terminal to define when the loop stops**

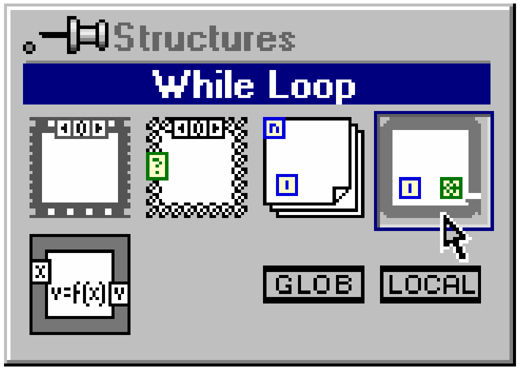


Iteration Terminal

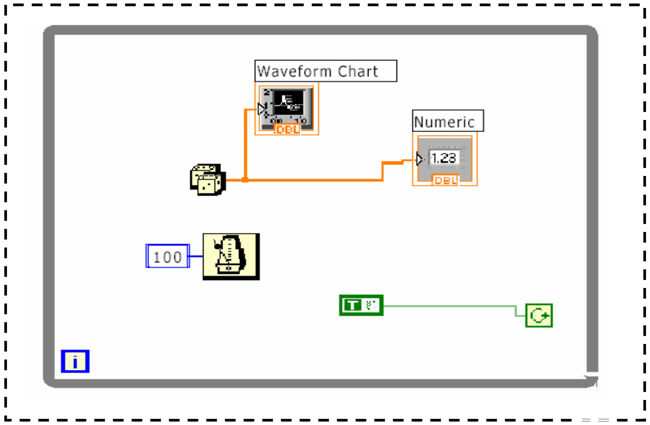
Conditional Terminal

A. Structures--While Loop

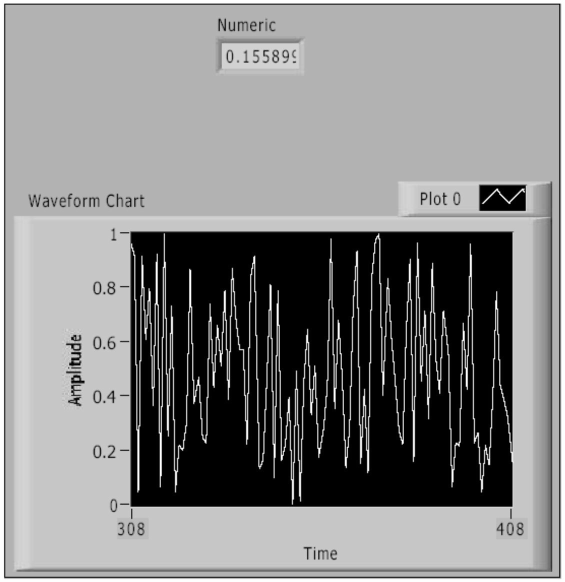
1. Select While Loop



2. Enclose code to be repeated



3. Drop or drag additional nodes and then wire



Do

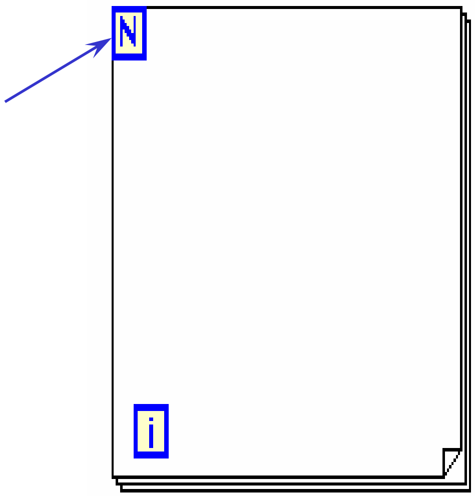
(Execute diagram inside Loop)

continue if Condition is TRUE

Save as while loop1.vi

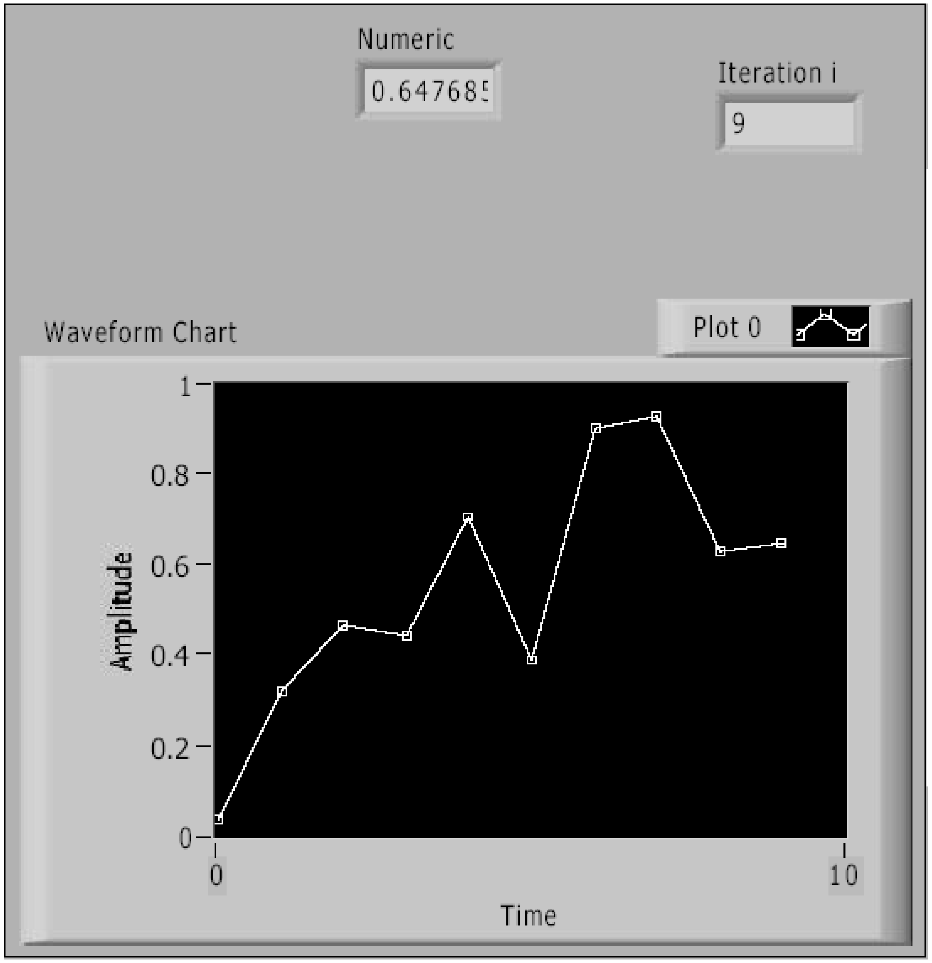
A. Structures--For Loop

- In Structures subpalette of Functions palette
- Enclose code to be repeated and/or resize and add nodes inside boundary
- Executes diagram inside of loop a predetermined number of times
- Shift registers can be created at the border

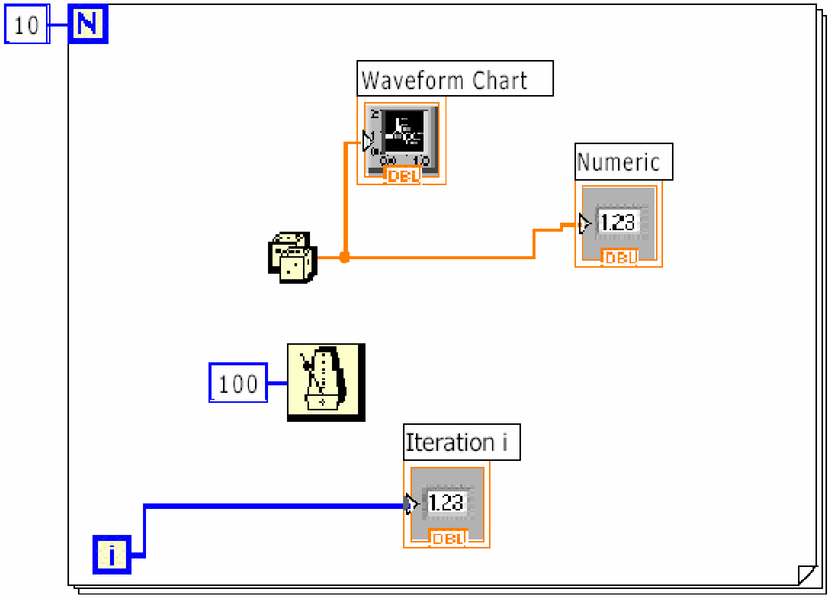


Count terminal
(Numerical input)

A. Structures--For Loop



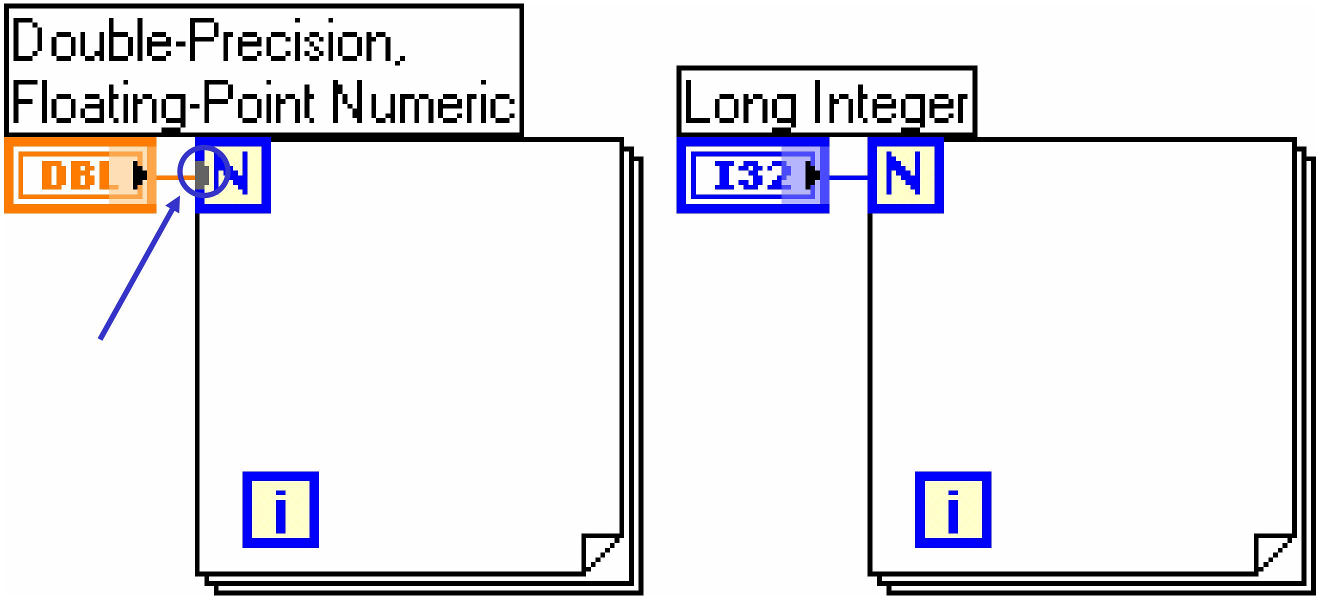
Replace wile loop with for loop



Save as for loop.vi

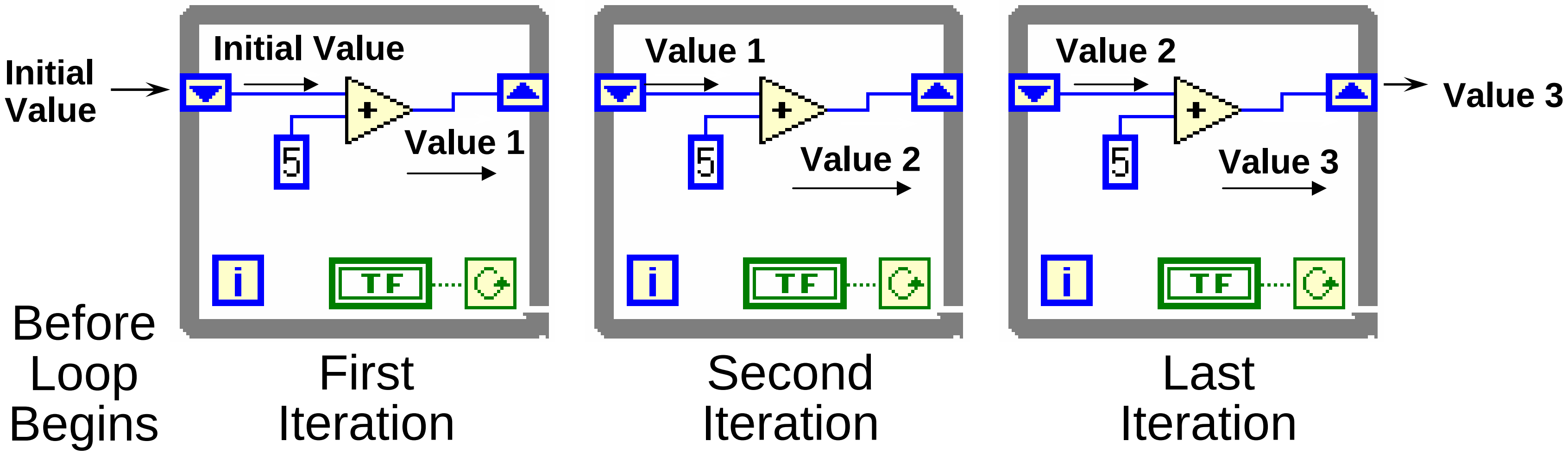
A. Structures--For Loop Numeric Conversion

- **Numeric defaults to double precision (8 bytes) or long integer (4 bytes)**
- **LabVIEW automatically converts to different representations**
- **Gray *coercion* dot on terminal indicates conversion**

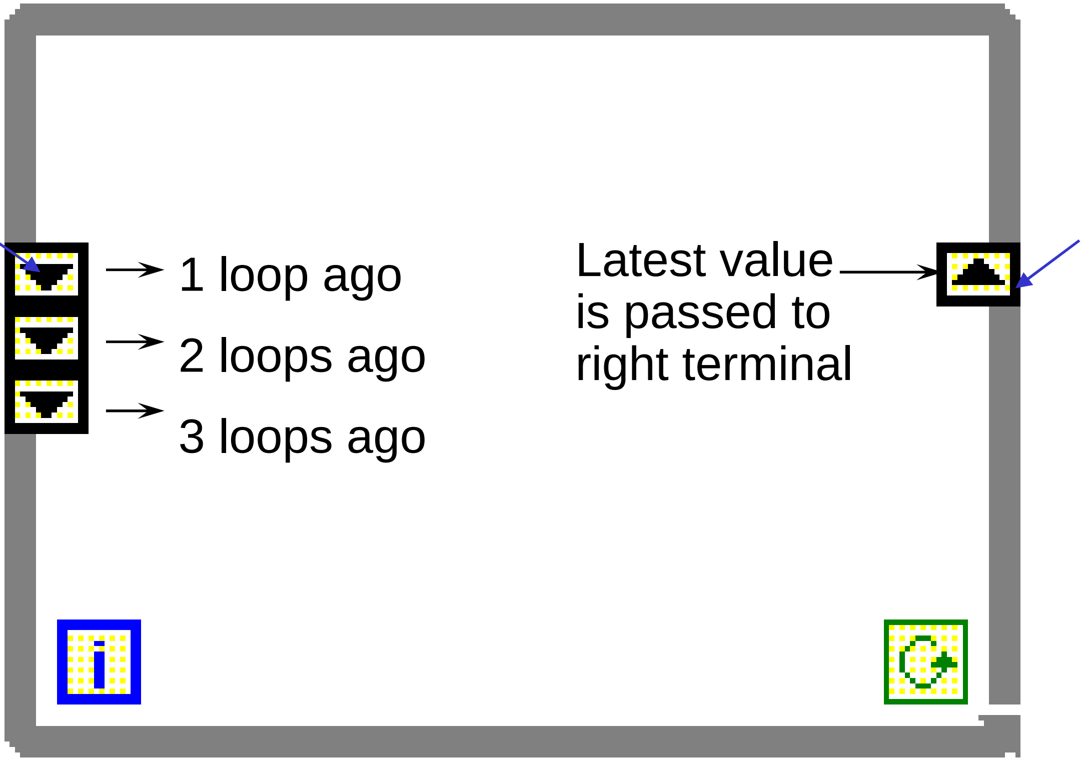


A. Structures--For Loop
Shift Registers

- Available at left or right border of loop structures
- Right-click on border and select Add Shift Register
- Right terminal stores data on completion of iteration
- Left terminal provides stored data at beginning of next iteration



Additional Shift Register Elements



Right-click on left terminal to add new elements

1 loop ago

2 loops ago

3 loops ago

Latest value is passed to right terminal

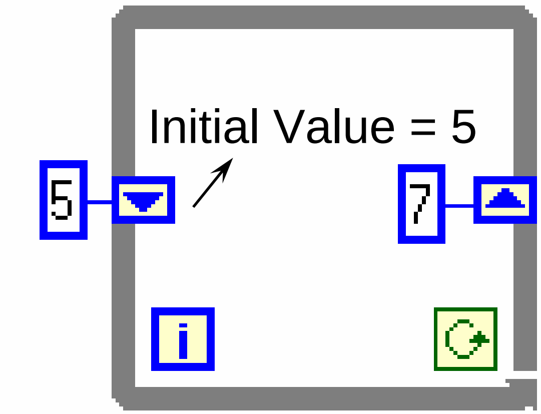
Right-click on the border for a new shift register

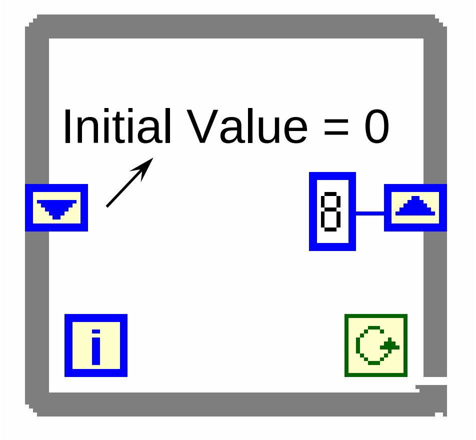
Initializing Shift Registers

Initialized

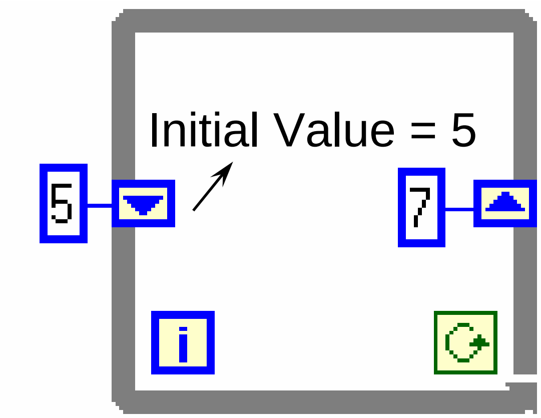
Uninitialized

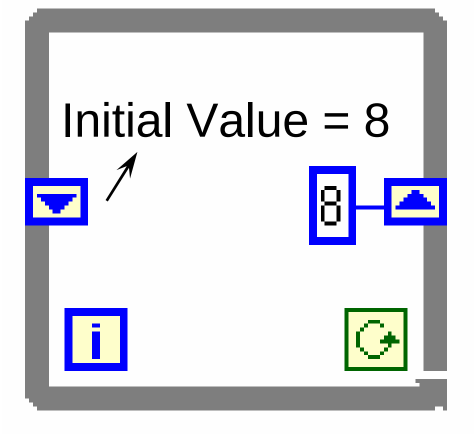
RUN 1





RUN 2





1. Find the **VI from Help»Find Examples...»Search**

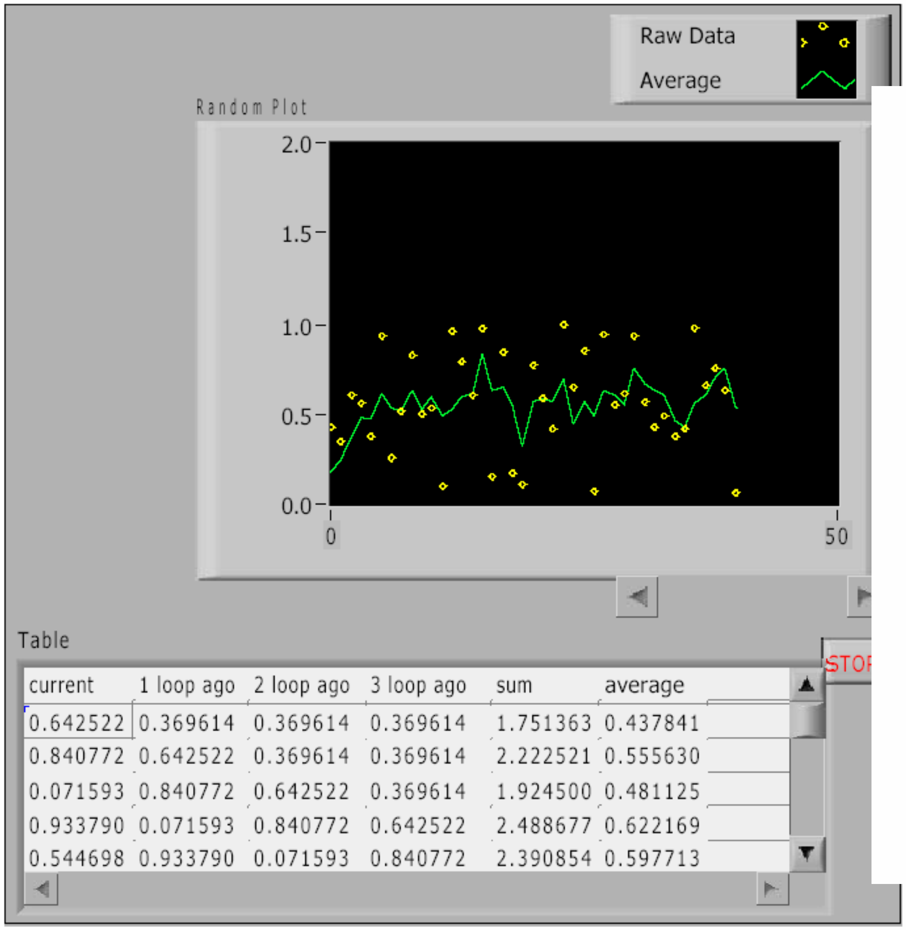
2. Type in shift registers and **search**

3. Choose **Calculate Running Average.vi**

4. Modify the VI as shown following

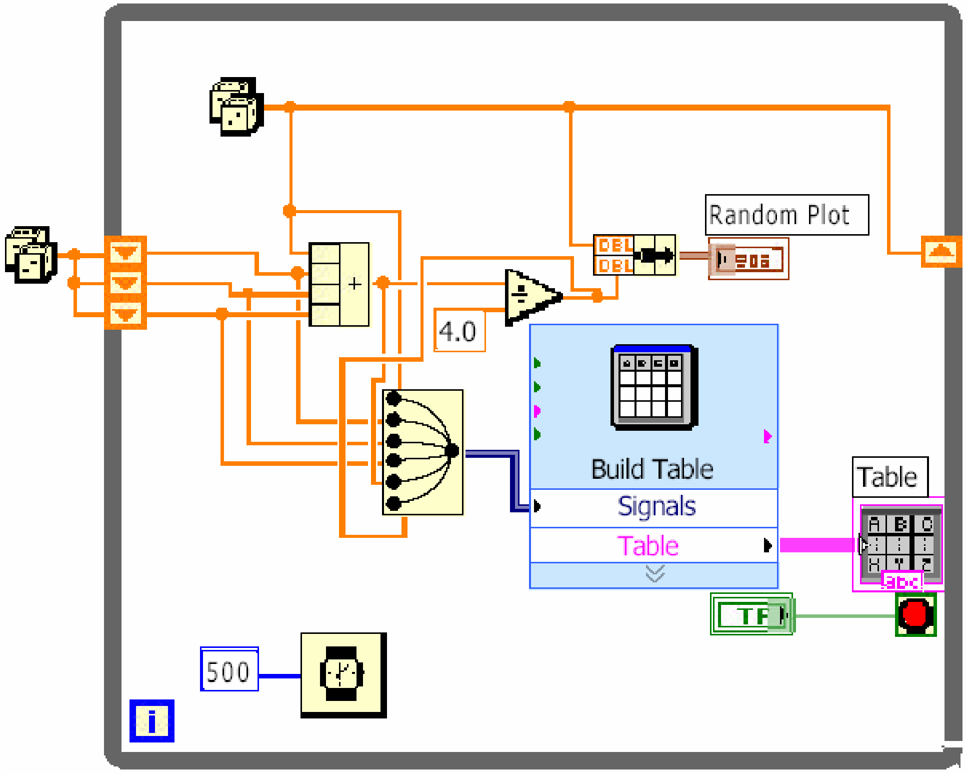
5. Save to your floppy disk

6. Change to for loop



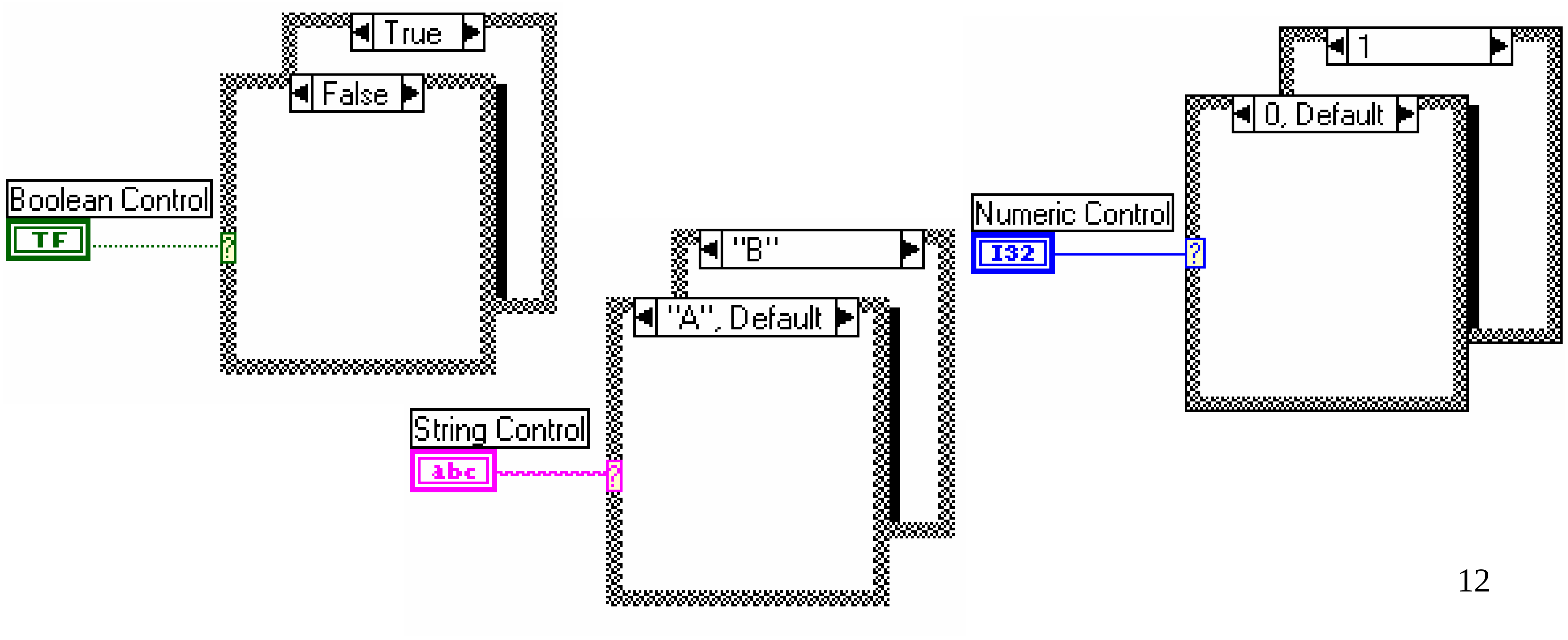
Table

current	1 loop ago	2 loop ago	3 loop ago	sum	average
0.642522	0.369614	0.369614	0.369614	1.751363	0.437841
0.840772	0.642522	0.369614	0.369614	2.222521	0.555630
0.071593	0.840772	0.642522	0.369614	1.924500	0.481125
0.933790	0.071593	0.840772	0.642522	2.488677	0.622169
0.544698	0.933790	0.071593	0.840772	2.390854	0.597713

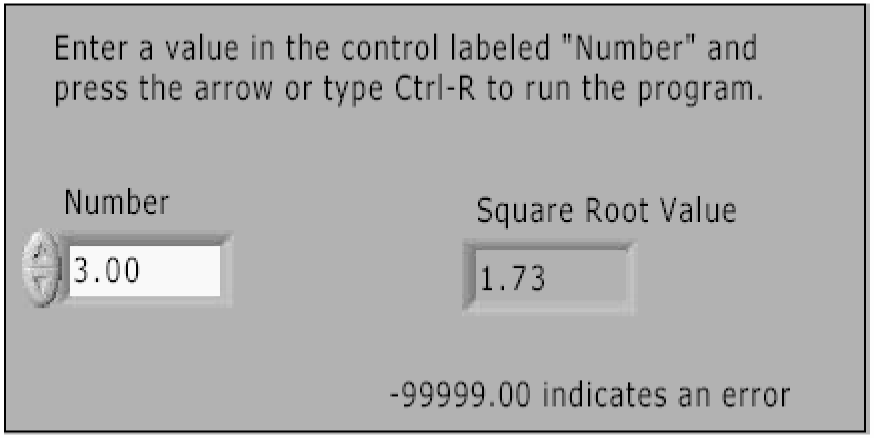


A. Structures--case structures If…Then…Else

- In the Structures subpalette of Functions palette
- Enclose nodes or drag them inside the structure
- Stacked like a deck of cards, only one case visible



1. Find the **VI from Help»Find Examples...»Search**
2. **Type in “**simple case**”** and **search**
3. Choose **Simple Case Structure.vi**



Enter a value in the control labeled "Number" and press the arrow or type Ctrl-R to run the program.

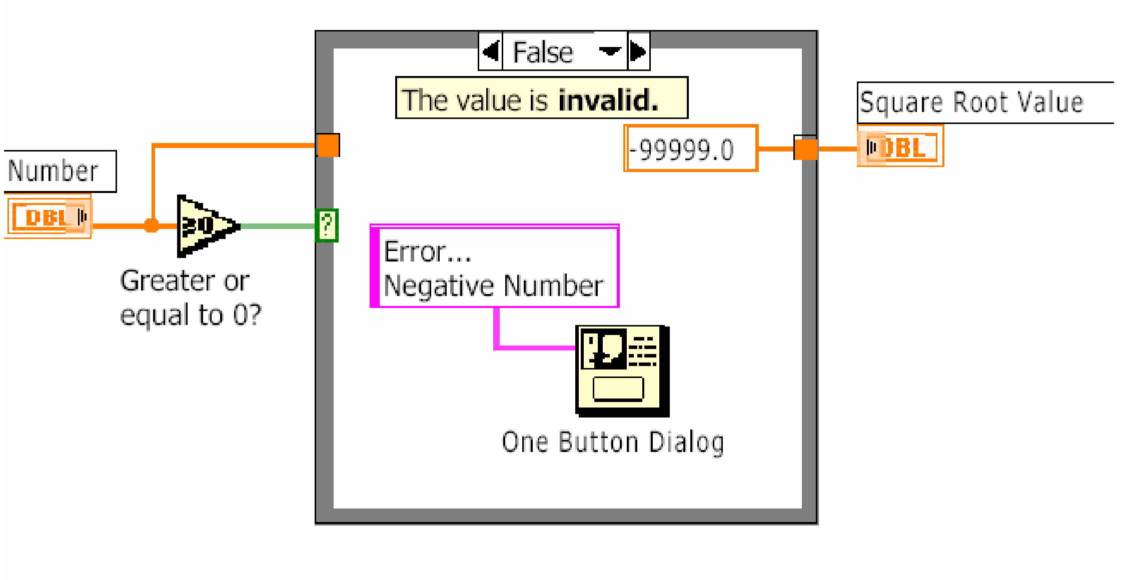
Number

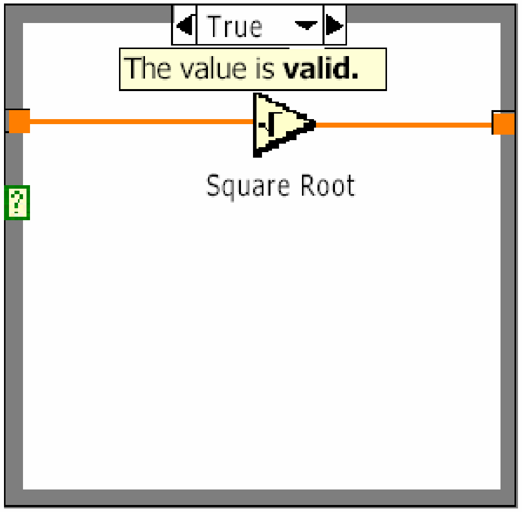
3.00

Square Root Value

1.73

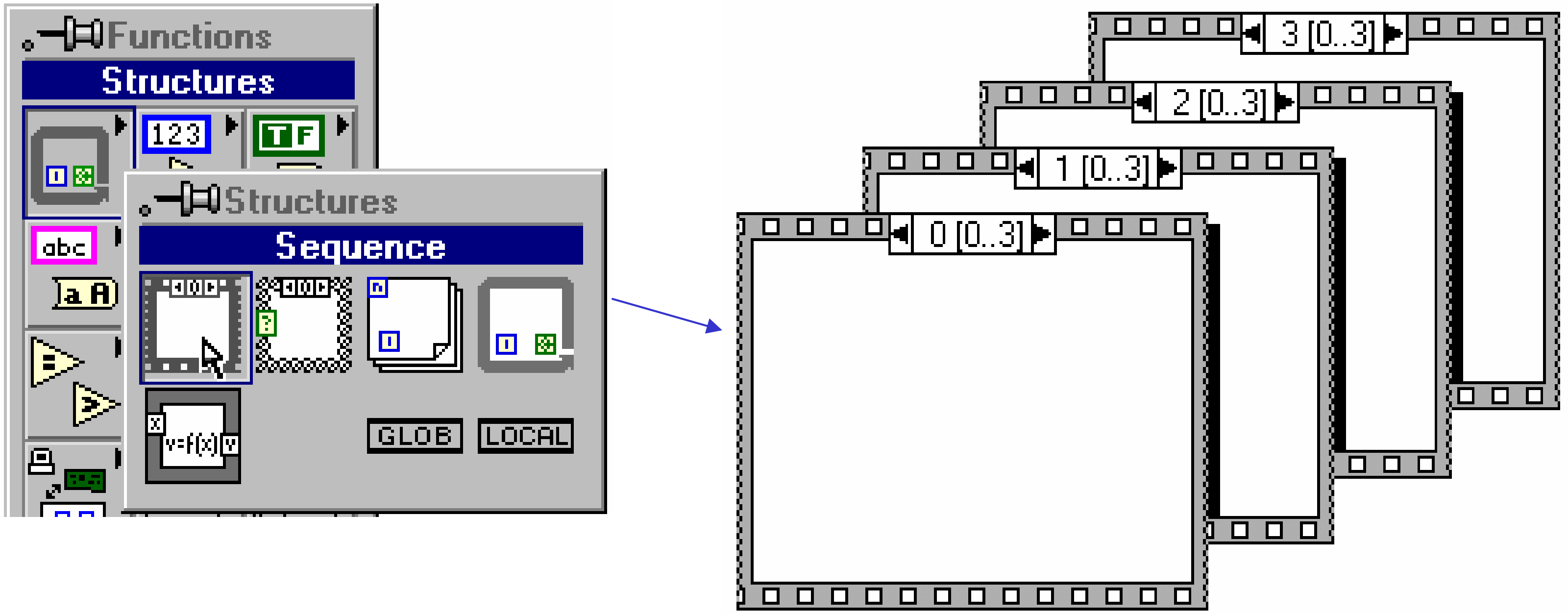
-99999.00 indicates an error





A. Structures--Sequence Structures

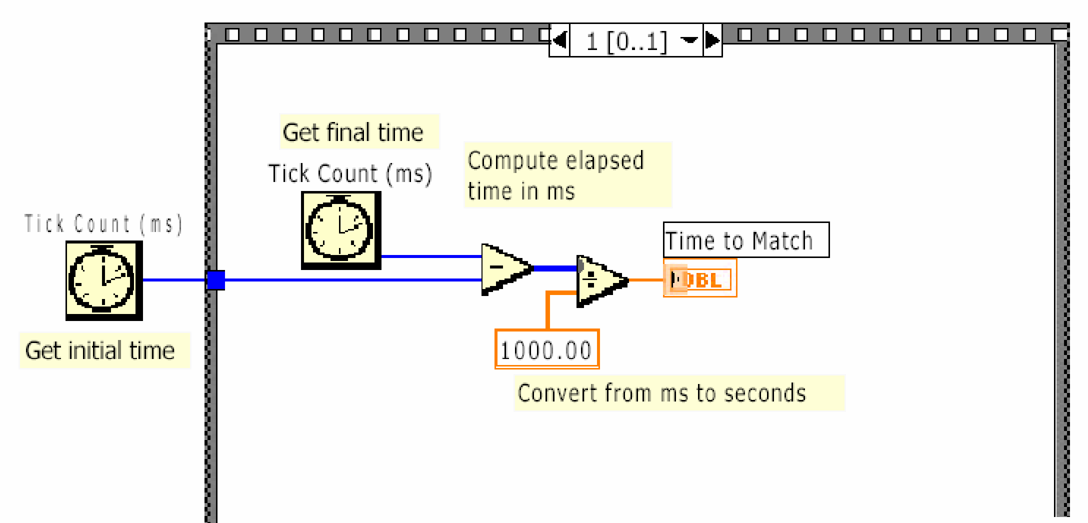
- **In the Structures subpalette of Functions palette**
- **Executes diagrams sequentially, Frame 0 (0..x), where x is the total number of frames**
- **Stacked like a deck of cards, only one frame visible**

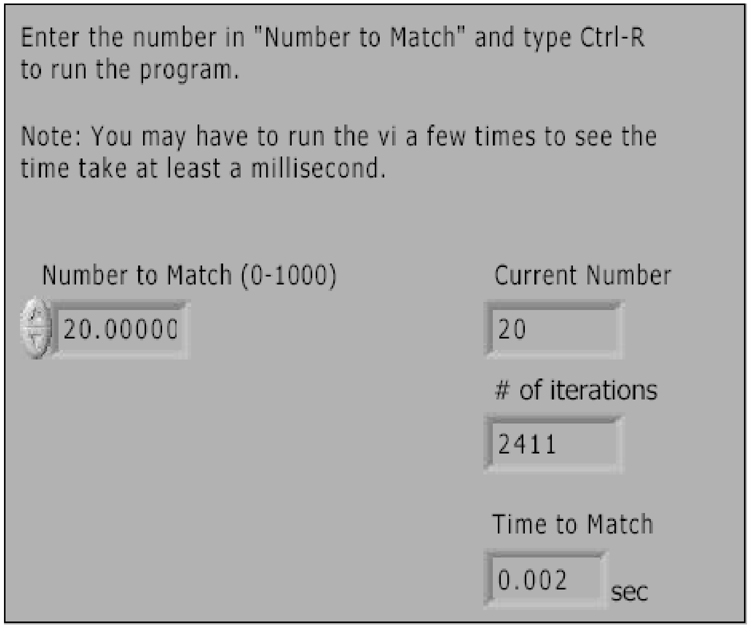


1. Find the **VI from Help»Find Examples...»Search**

2. Type in “sequence” and **search**

3. Choose Time to Match**.vi**




Enter the number in "Number to Match" and type Ctrl-R to run the program.

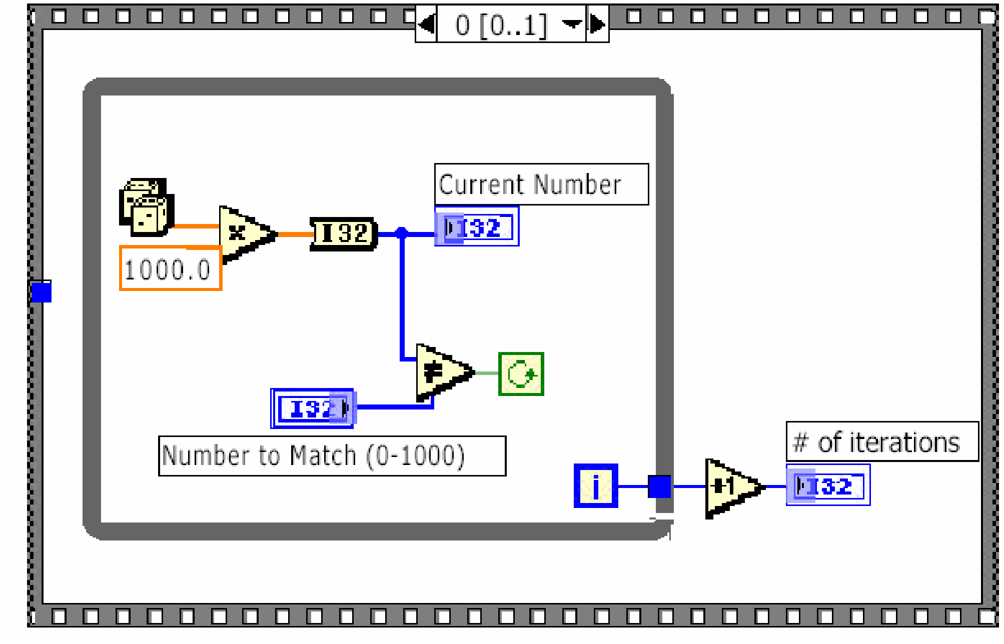
Note: You may have to run the vi a few times to see the time take at least a millisecond.

Number to Match (0-1000): 20.00000

Current Number: 20

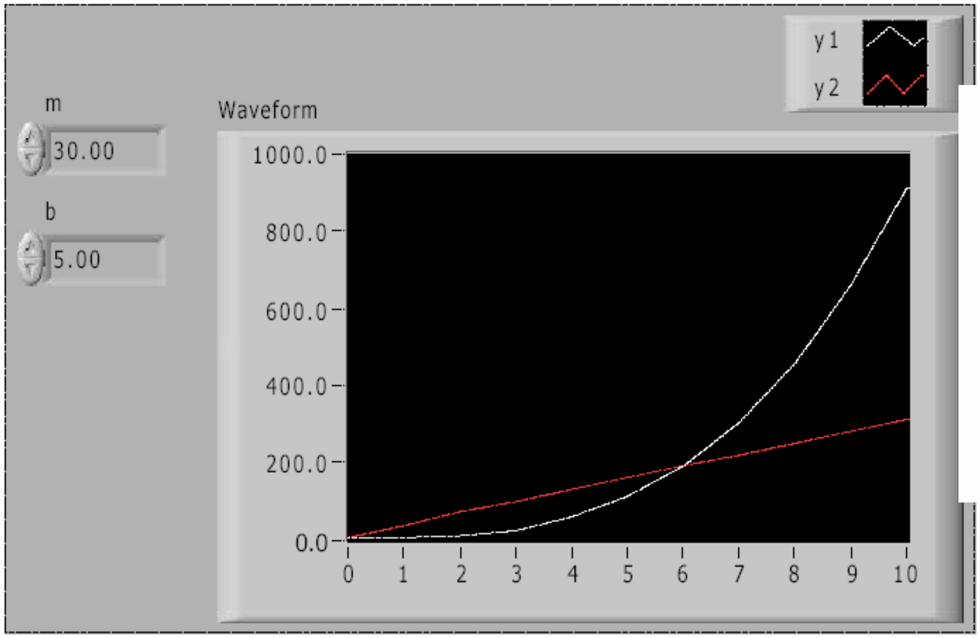
# of iterations: 2411

Time to Match: 0.002 sec



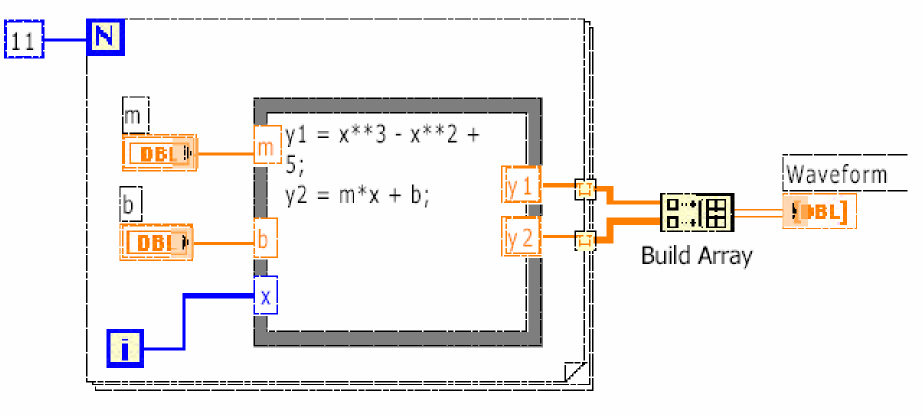
A. Structures--Formula Node

- In the Structures subpalette
- Implement complicated equations
- Variables created at border
- Variable names are case sensitive
- Each statement must terminate with a semicolon (;)
- Context Help Window shows available functions



m 30.00

b 5.00



```
y1 = x**3 - x**2 +
5;
y2 = m*x + b;
```

B. Array Controls and Indicators

A. About arrays

B. About generating arrays with loops

C. Some basic array functions

D. What polymorphism is

E. Using graphs to display data

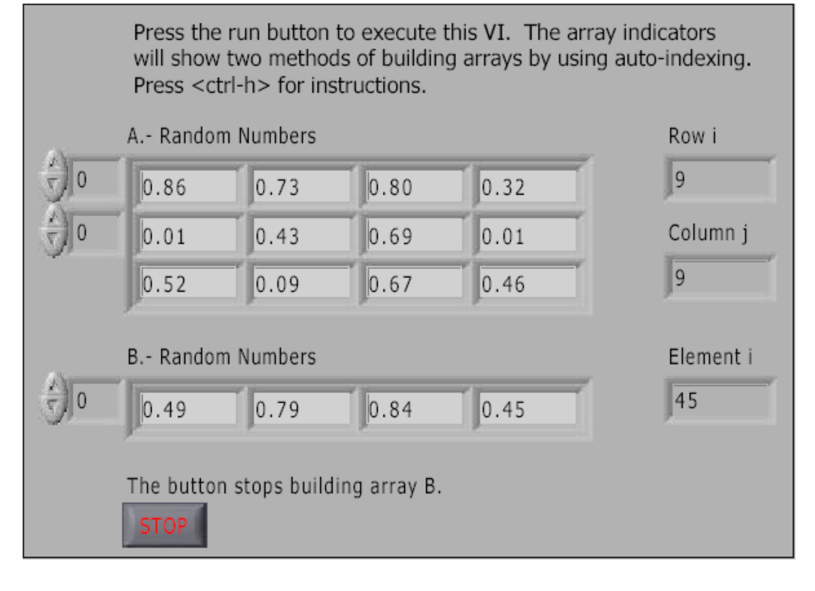
F. About clusters

G. Some cluster functions

1. Find the **VI from Help»Find Examples...»Search**

2. Type in “array” and **search**

3. Choose Time to Match**.vi**



Press the run button to execute this VI. The array indicators will show two methods of building arrays by using auto-indexing. Press <ctrl-h> for instructions.

A.- Random Numbers

0	0.86	0.73	0.80	0.32
0	0.01	0.43	0.69	0.01
	0.52	0.09	0.67	0.46

Row i: 9

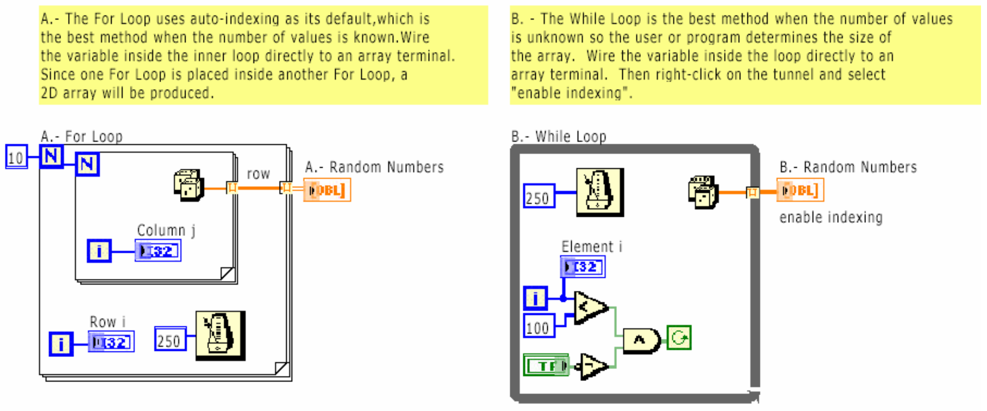
Column j: 9

B.- Random Numbers

0	0.49	0.79	0.84	0.45

Element i: 45

The button stops building array B.

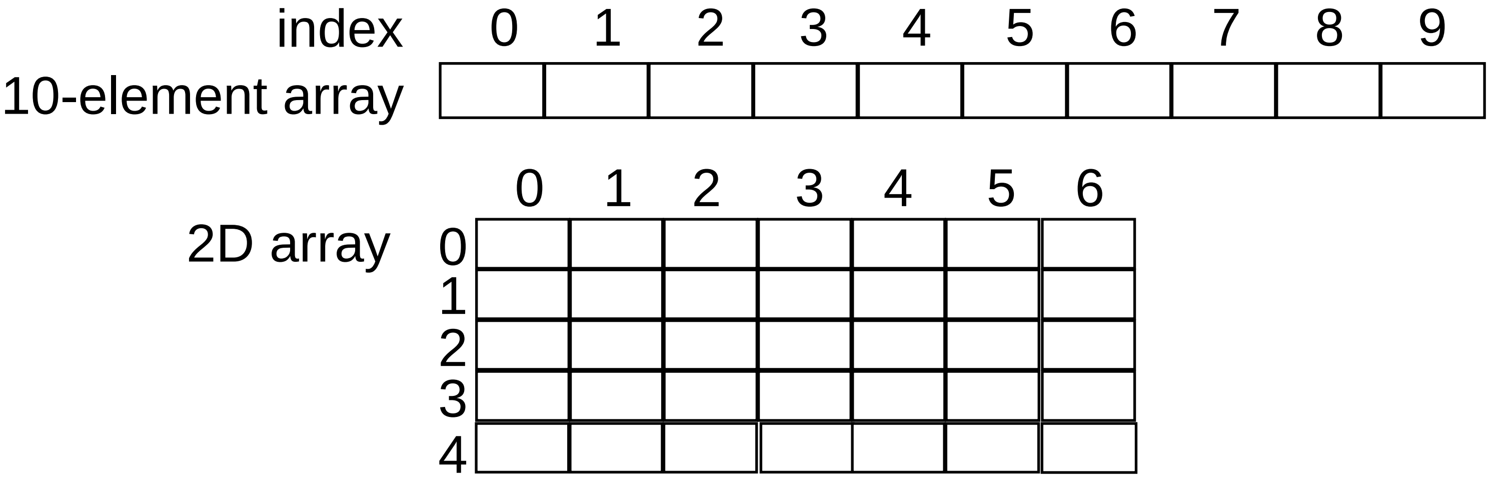


A.- The For Loop uses auto-indexing as its default,which is the best method when the number of values is known.Wire the variable inside the inner loop directly to an array terminal. Since one For Loop is placed inside another For Loop, a 2D array will be produced.

B. - The While Loop is the best method when the number of values is unknown so the user or program determines the size of the array. Wire the variable inside the loop directly to an array terminal. Then right-click on the tunnel and select "enable indexing".

Arrays

- **Collection of data elements that are of same type**
- **One or more dimensions, up to 2^{31} elements per dimension**
- **Elements accessed by their index**
- **First element is index 0**



index 0 1 2 3 4 5 6 7 8 9

10-element array

0 1 2 3 4 5 6

2D array

0

1

2

3

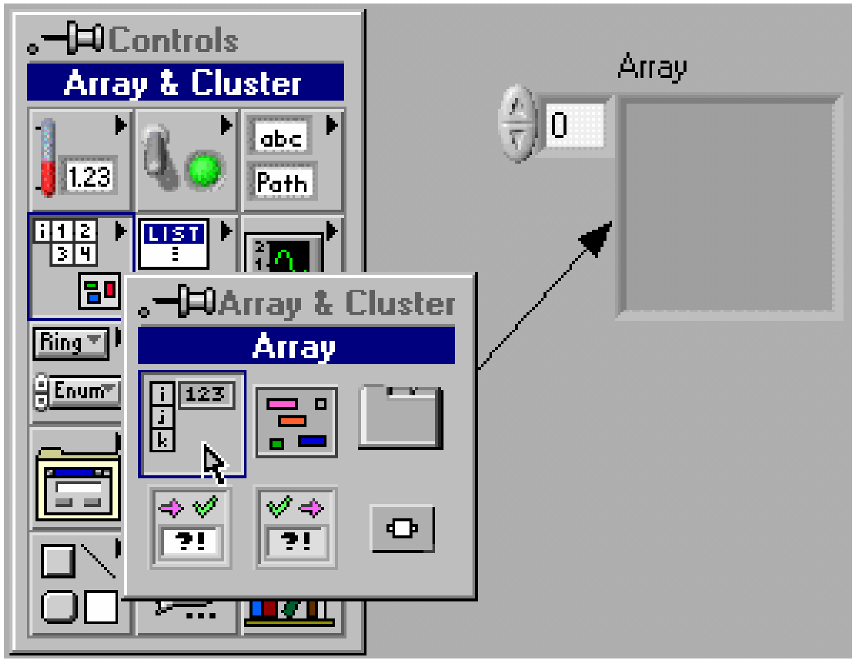
4

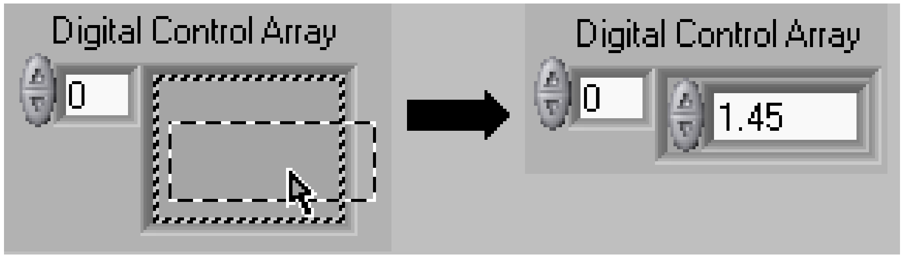
Five-row by seven column array of 35 elements

Array Controls and Indicators

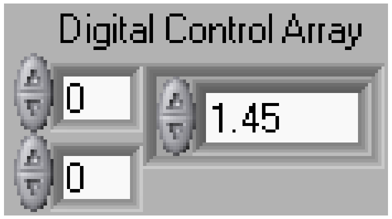
1. Select the **Array Shell** from the Controls palette

2. Place **data object** inside shell



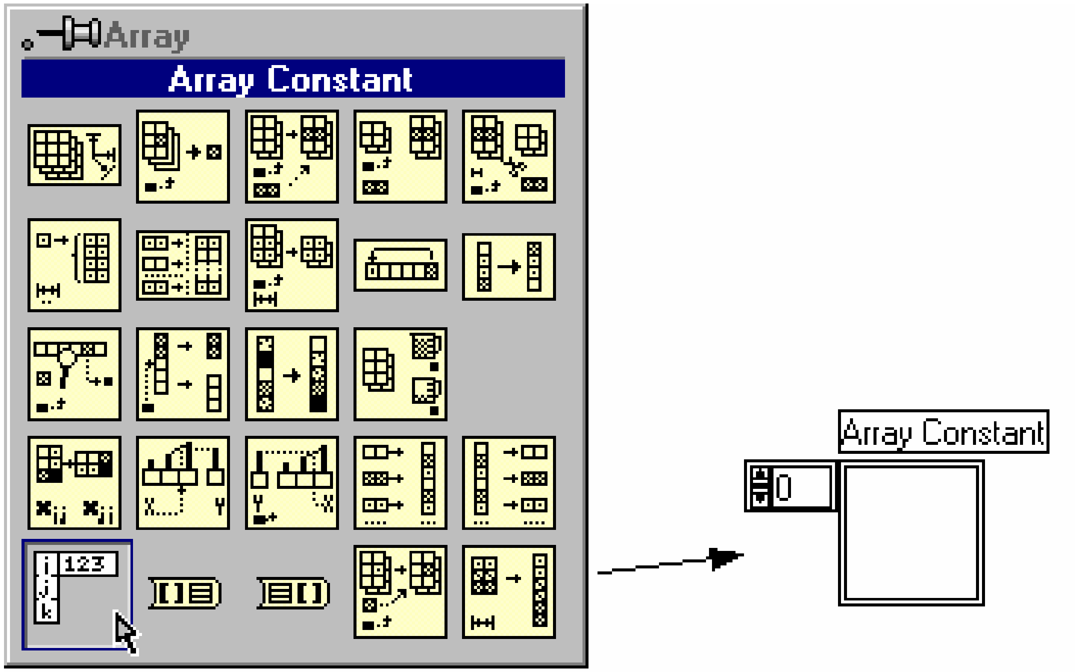


Add Dimension for 2D arrays

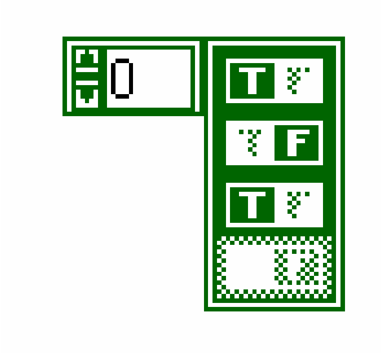


Creating Array Constants

1. Select **Array Constant** shell from the **Array** subpalette



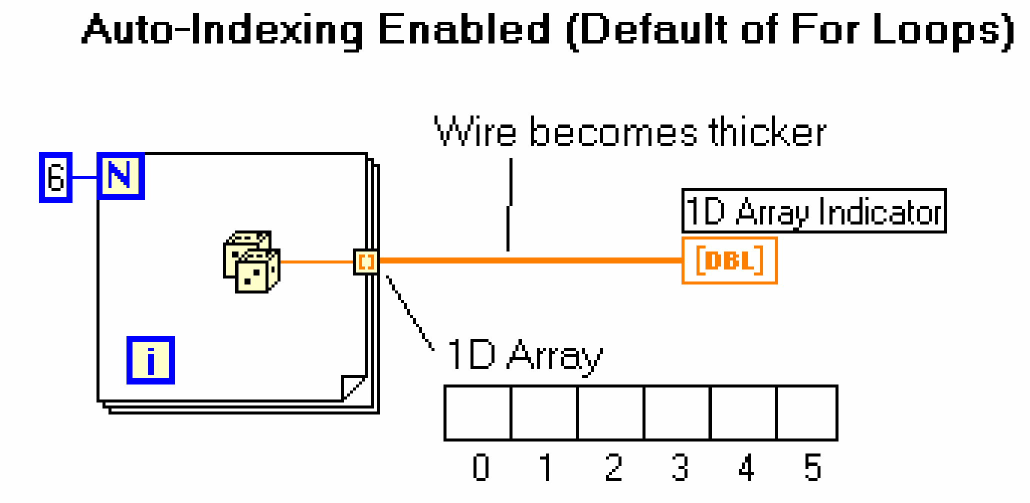
2. Place the data object in the array shell



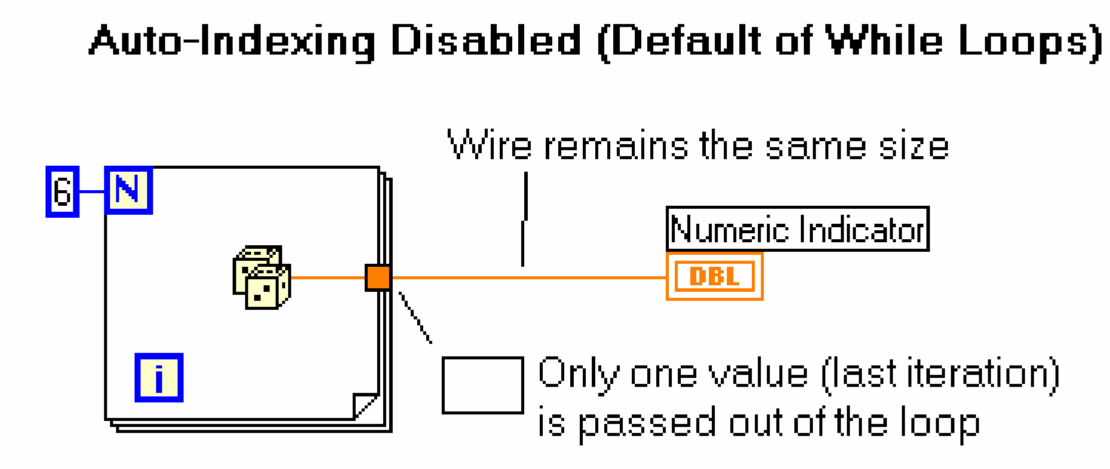
Creating and Using Arrays

- ***Auto-Indexing* – loops accumulate arrays at their boundaries**

- **For Loops auto index by default; While Loops do not**



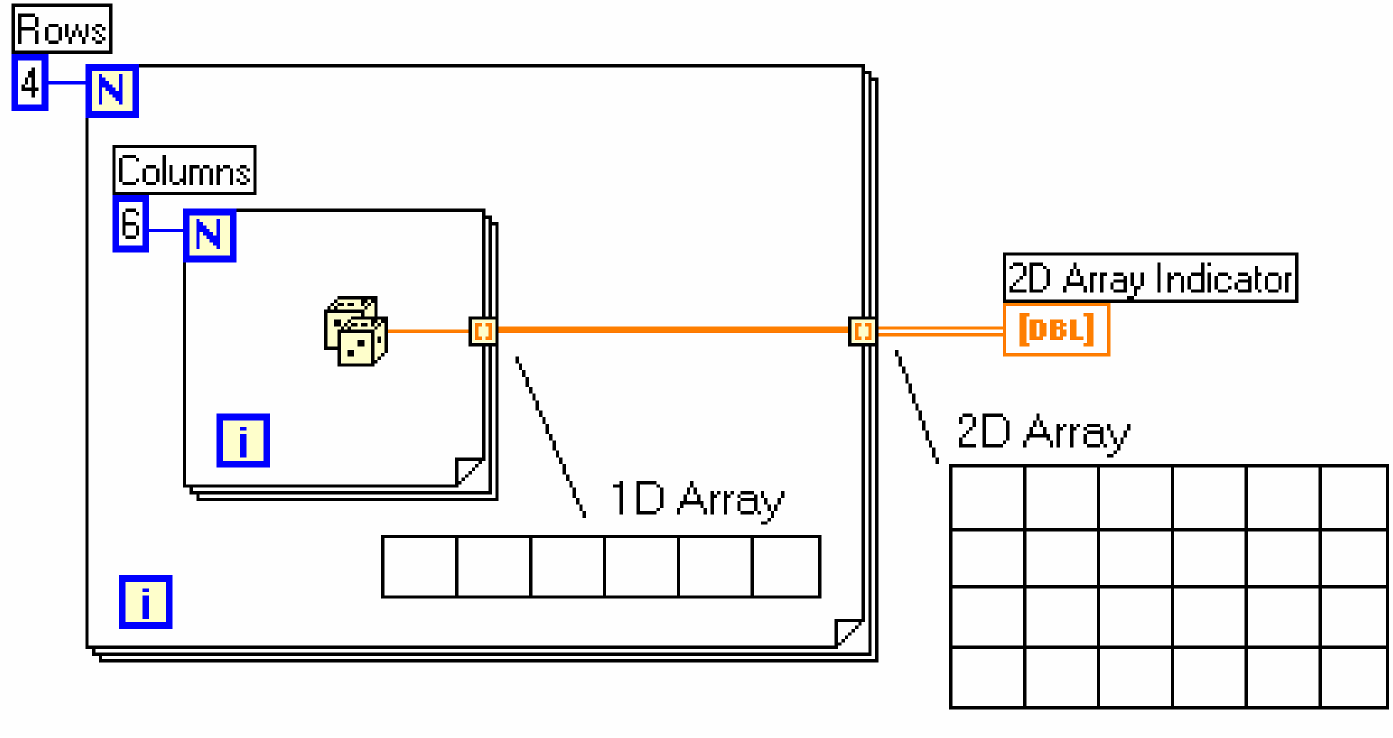
Auto-Indexing Enabled (Default of For Loops)



Auto-Indexing Disabled (Default of While Loops)

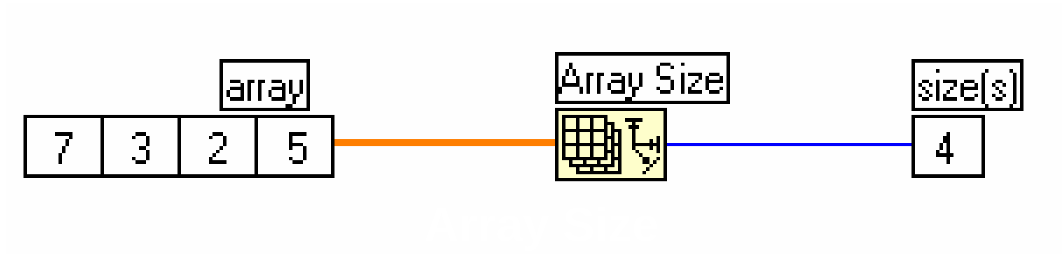
6
N
i
Wire remains the same size
Numeric Indicator
DBL
Only one value (last iteration) is passed out of the loop

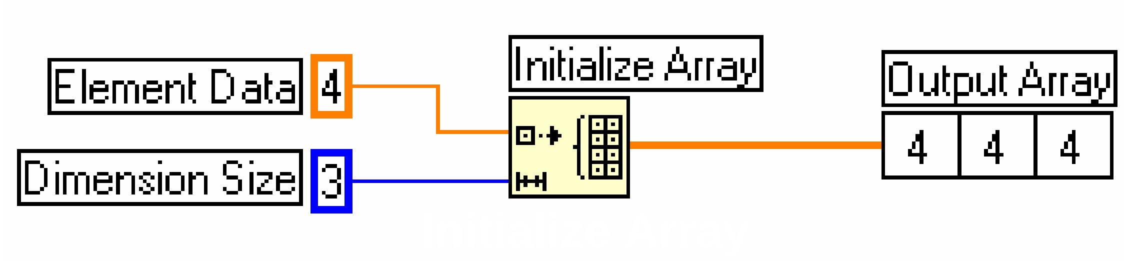
Creating 2D Arrays

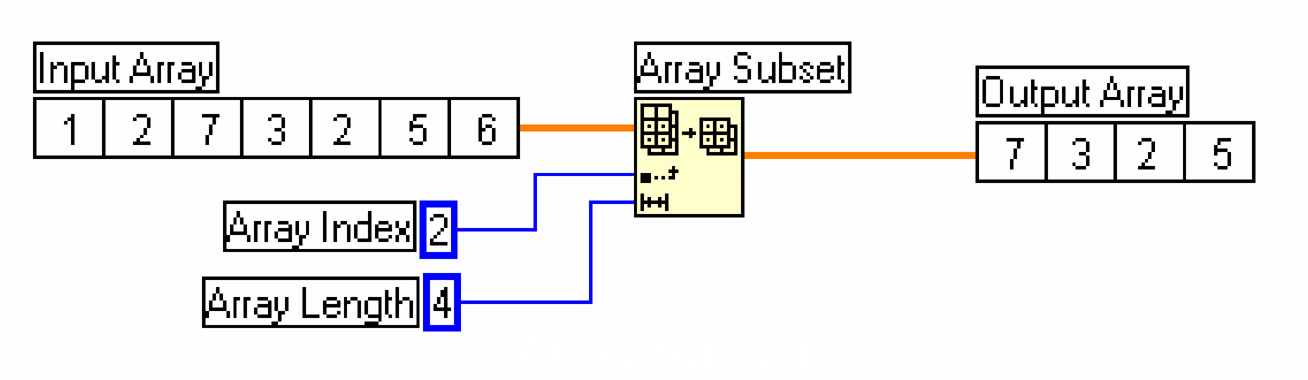


- **Inner loop creates column elements**
- **Outer loop stacks them into rows**

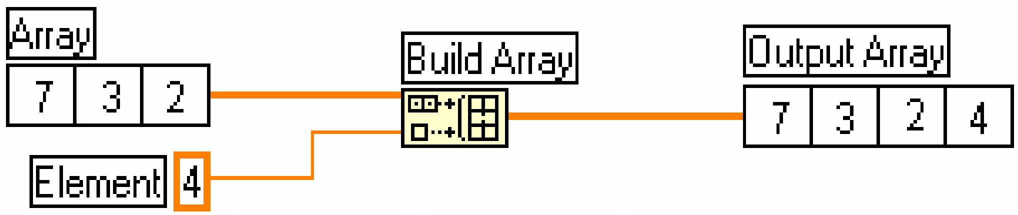
Common Array Functions



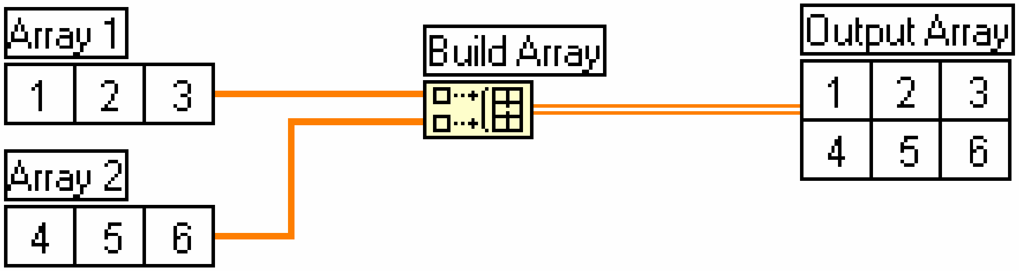




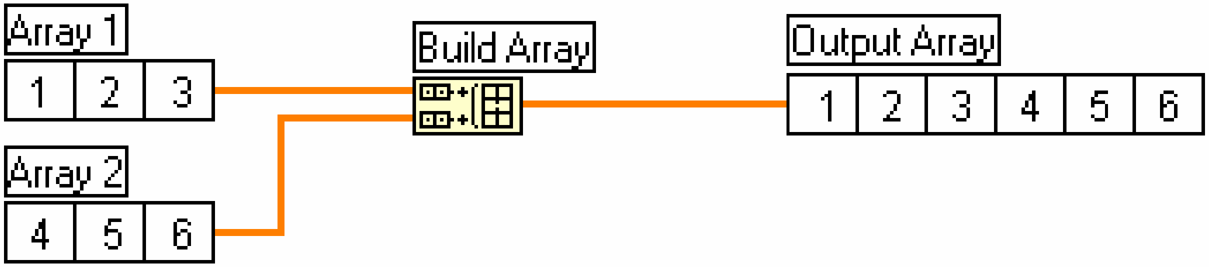
The Build Array Function



Appending an element



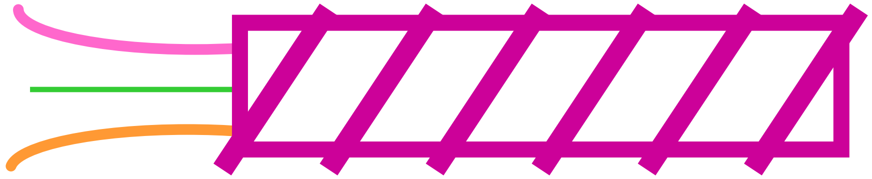
Building a higher dimension array



Concatenate Inputs (default)

B.Arrays and Clusters---- Clusters

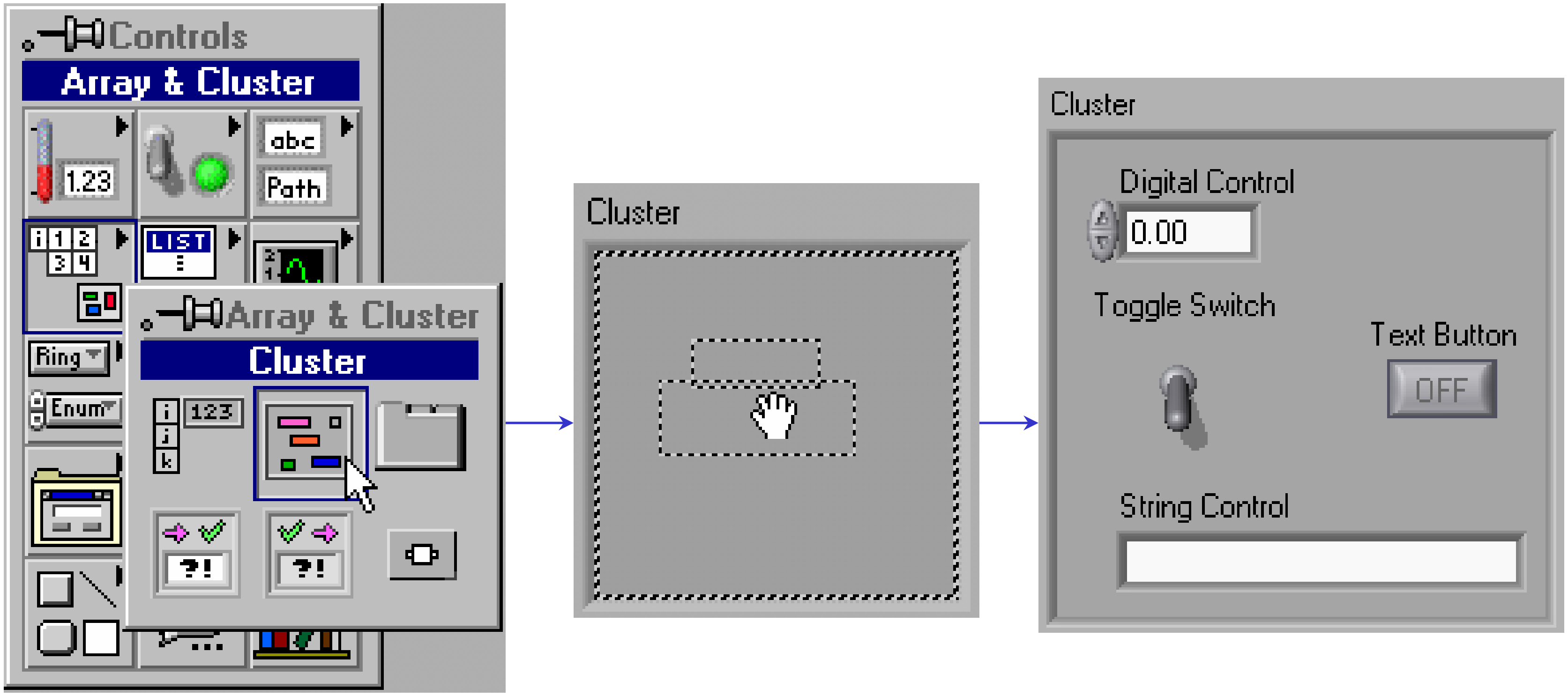
- **Data structure that groups data together**
- **Data may be of different types**
- **Analogous to *record* in Pascal or *struct* in C**
- **Elements must be either all controls or all indicators**
- **Thought of as wires bundled into a cable**



Cluster Controls and Indicators

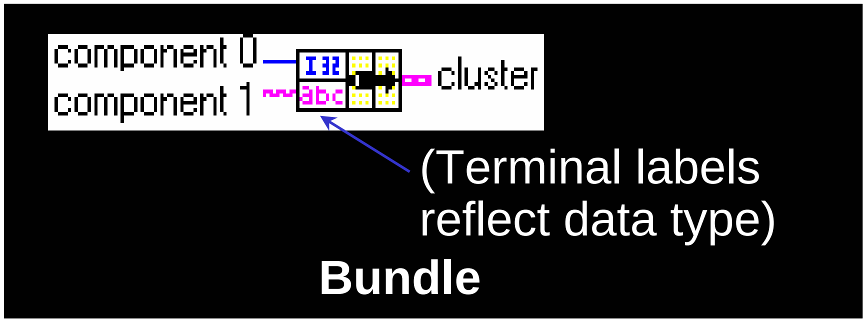
1. Select a **Cluster** shell from the **Array & Cluster** subpalette

2. Place objects inside the shell



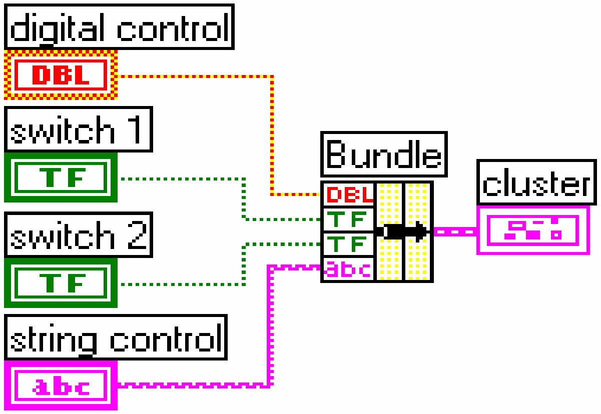
Cluster Functions

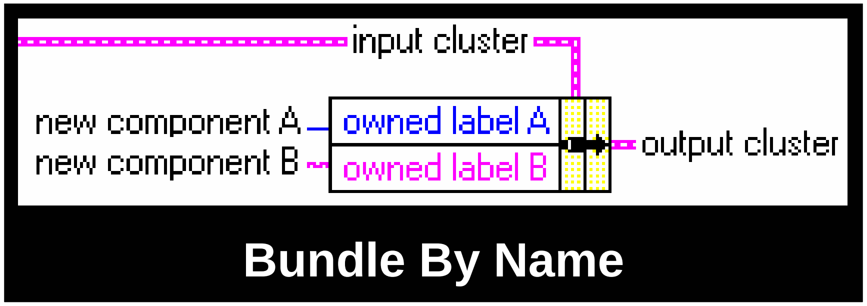
- In the **Cluster** subpalette of the **Functions** palette
- Can also be accessed by right-clicking on the cluster terminal



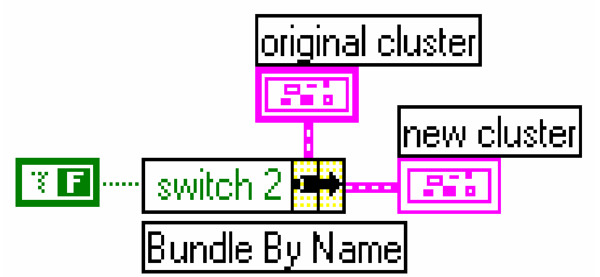
(Terminal labels reflect data type)

Bundle

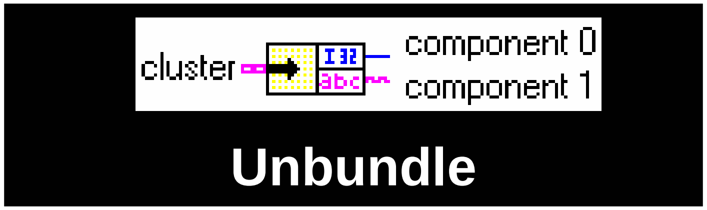




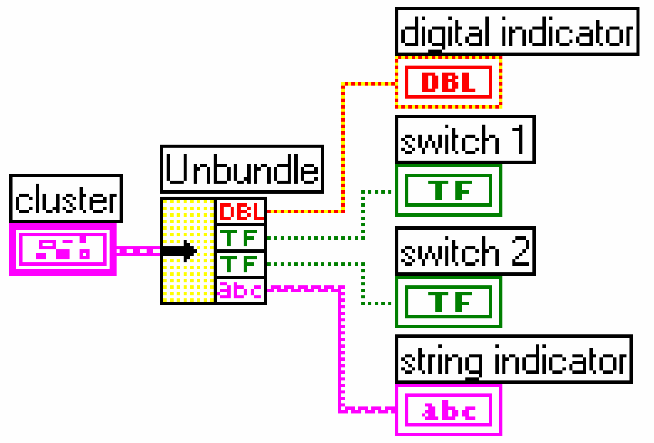
Bundle By Name

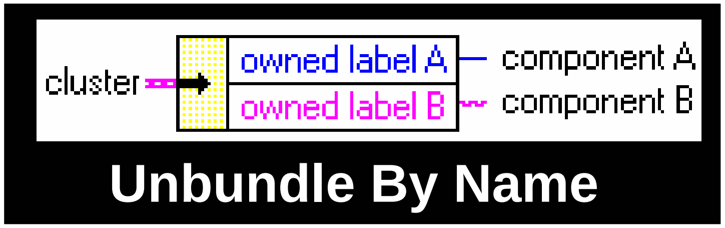


Cluster Functions

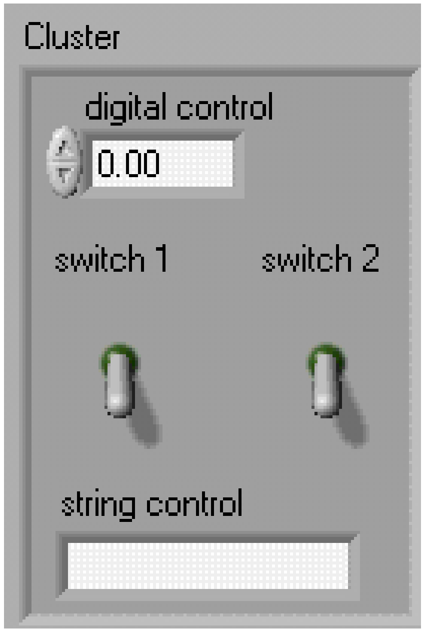


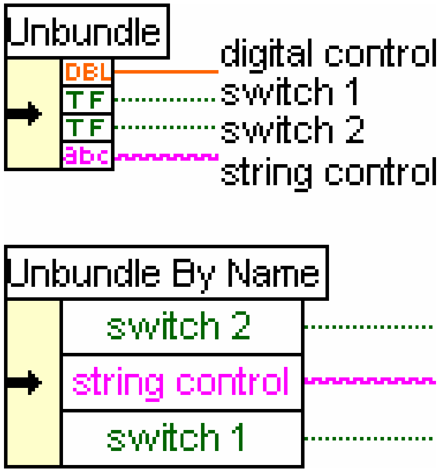
Unbundle





Unbundle By Name

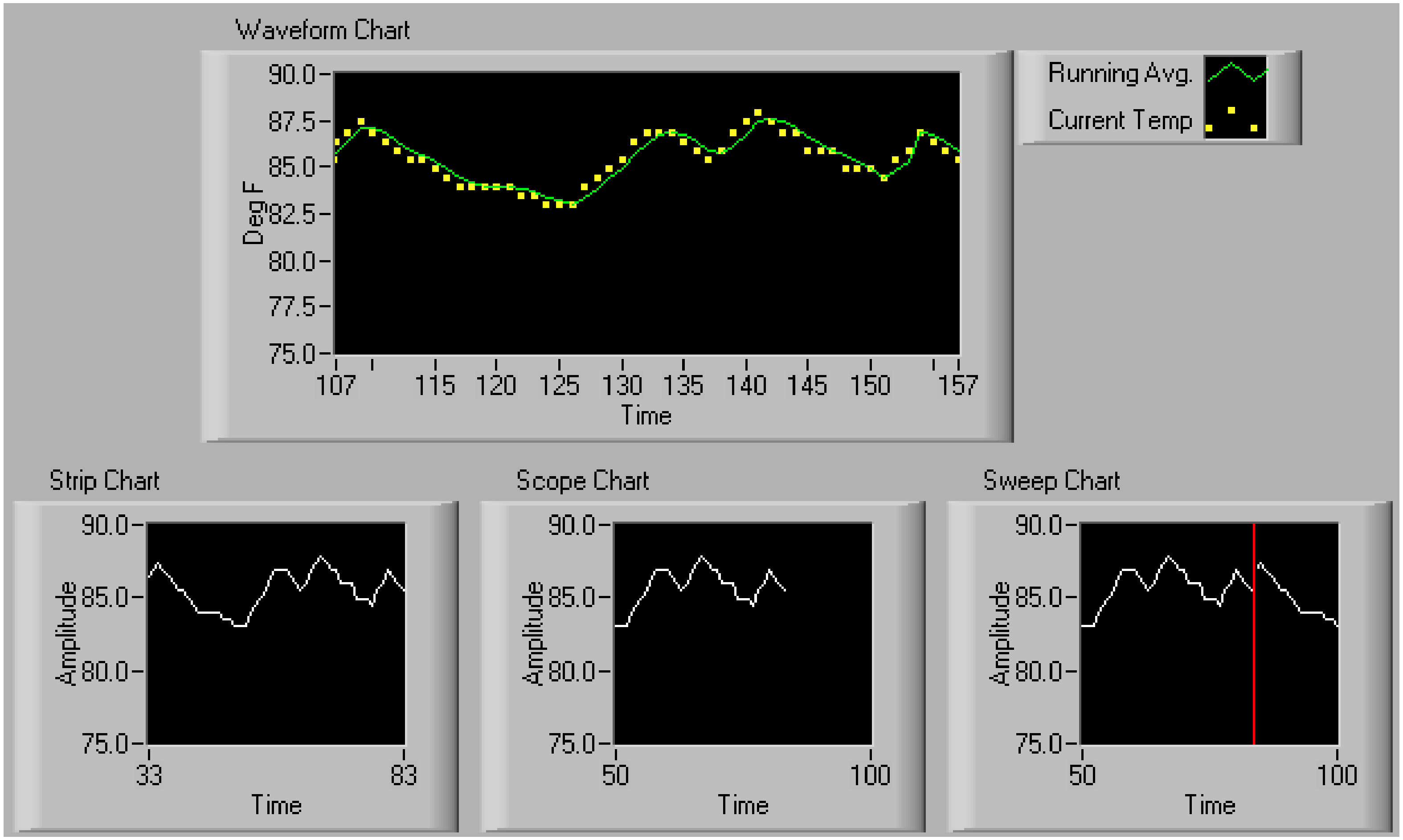




Unbundled cluster in the diagram

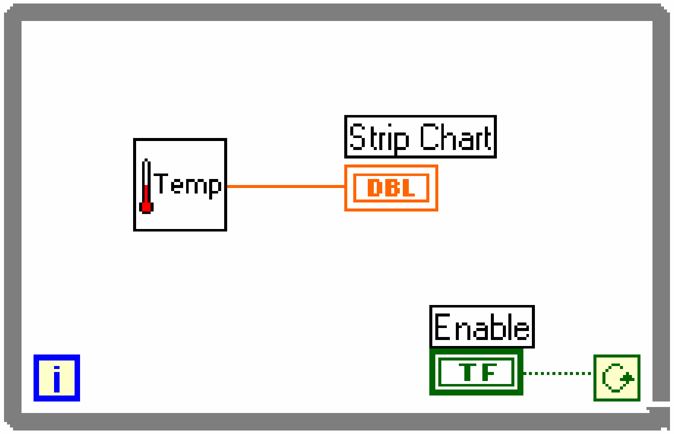
C.Charts and graphs---- charts

- **Selected from the Controls»Graph subpalette**

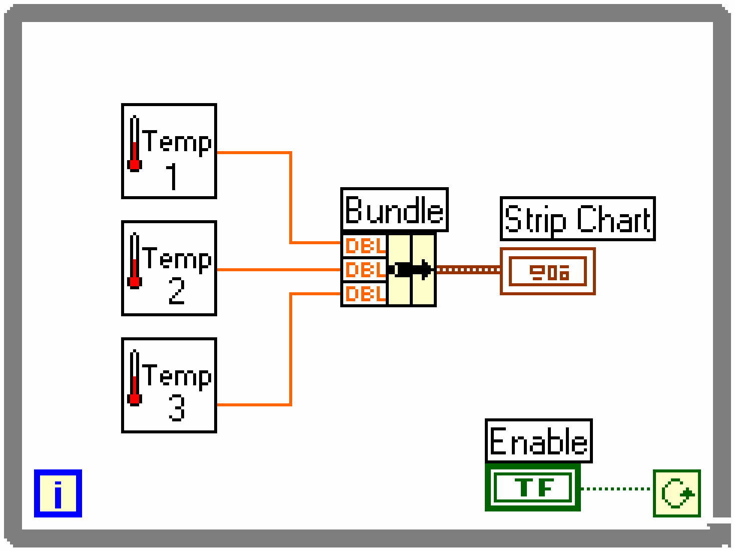


Wiring to Charts

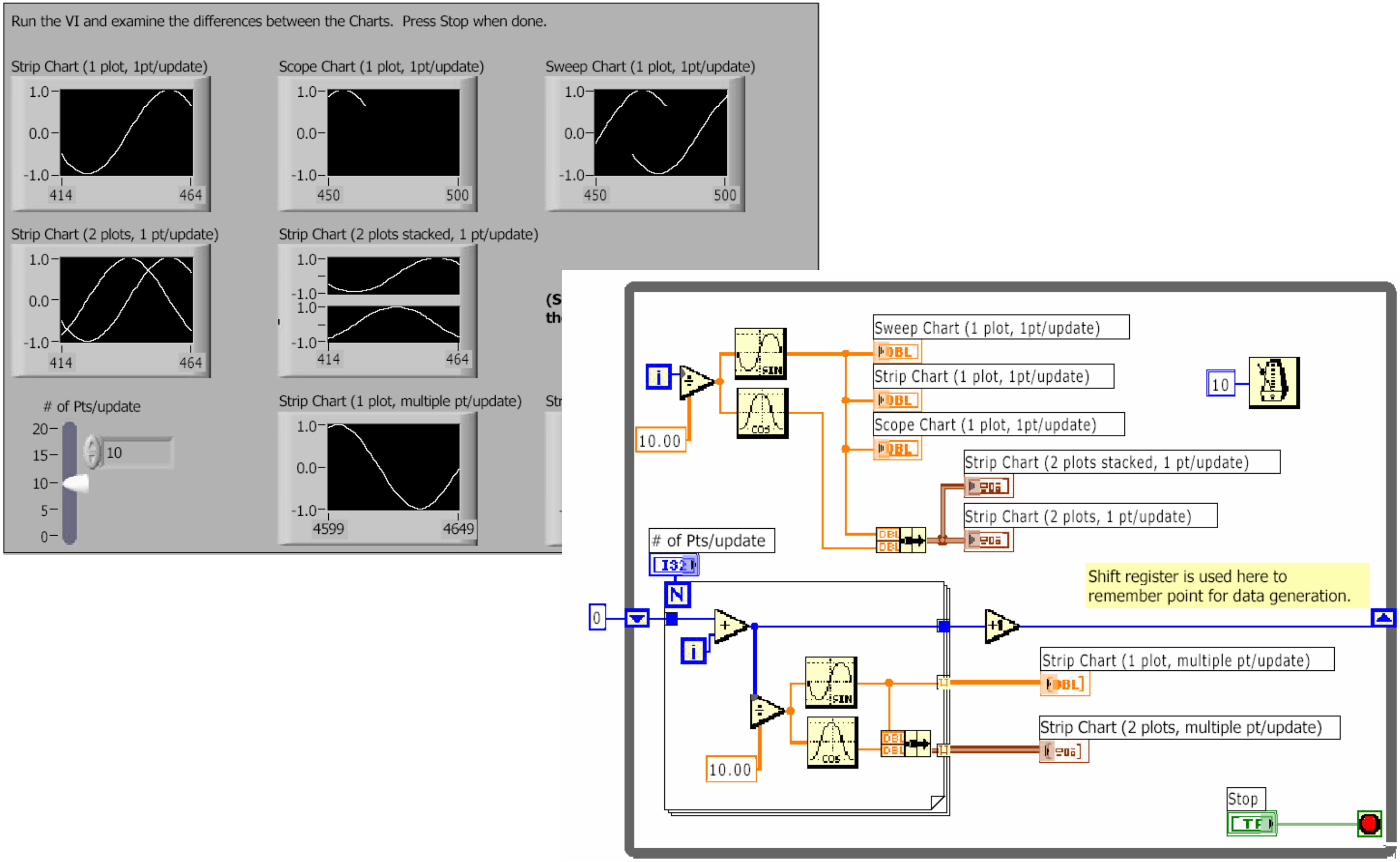
- **Single-Plot Chart**



- **Multiple-Plot Chart**



1. Find the **VI from Help»Find Examples...»Search**
2. **Type in “**chart” and **search**
3. Choose charts**.vi**



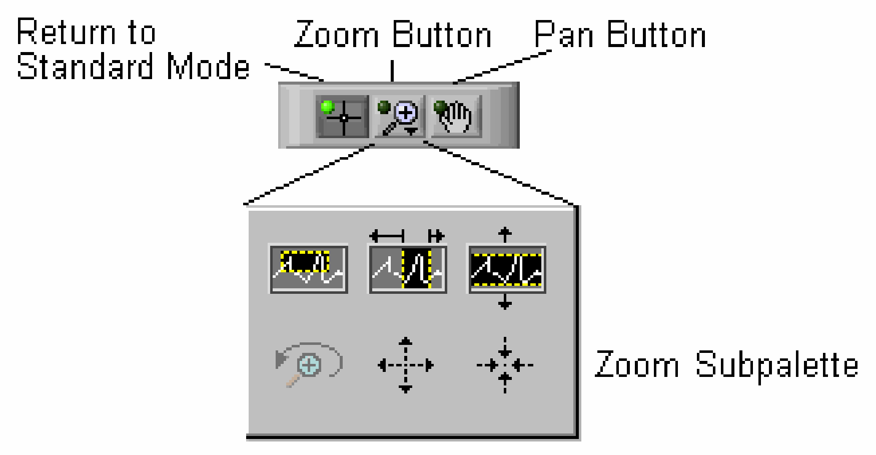
Run the VI and examine the differences between the Charts. Press Stop when done.

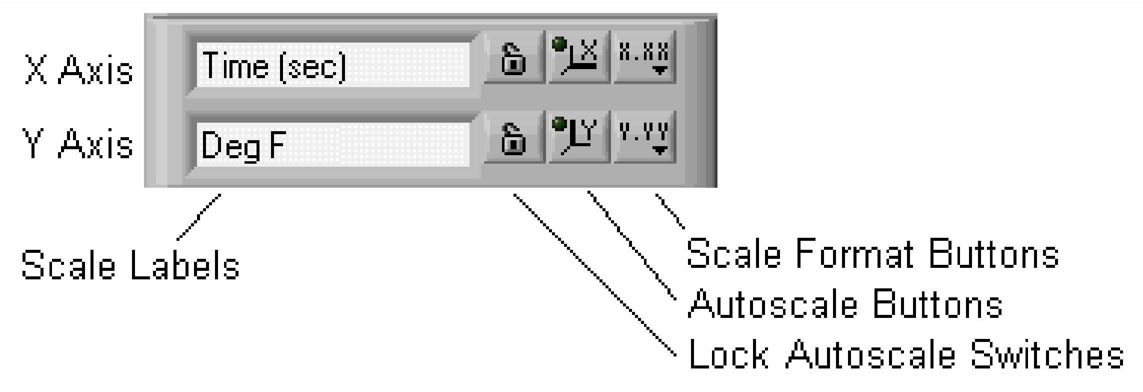
Strip Chart (1 plot, 1pt/update)
Scope Chart (1 plot, 1pt/update)
Sweep Chart (1 plot, 1pt/update)
1.0
0.0
-1.0
414
464
450
500
Strip Chart (2 plots, 1 pt/update)
Strip Chart (2 plots stacked, 1 pt/update)
# of Pts/update
20
15
10
5
0
10
Strip Chart (1 plot, multiple pt/update)
4599
4649
SIN
COS
10.00
DBL
Strip Chart (2 plots stacked, 1 pt/update)
Strip Chart (2 plots, 1 pt/update)
10
# of Pts/update
I32
N
0
i
+1
Strip Chart (1 plot, multiple pt/update)
Strip Chart (2 plots, multiple pt/update)
Stop
TF

Shift register is used here to remember point for data generation.

Customizing Charts and Graphs

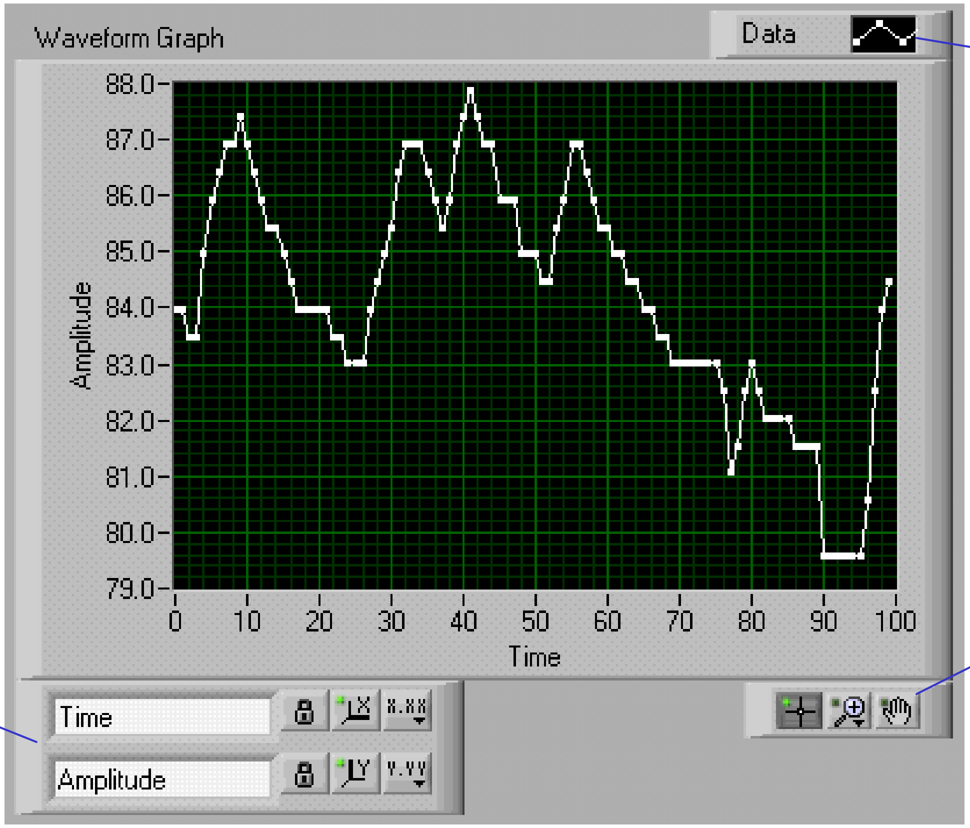
- **Plot Legend (color, point style, line style, etc.)**
- **Digital Display**
- **Scrollbar**
- **Customize the**
- **X and Y Axes**
- **Graph Palette**
- **Scale Legend**





C.Charts and graphs---- graphs

- Selected from the Graph subpalette
- Waveform Graph – Plot an array of numbers against their indices
- XY Graph – Plot one array against another

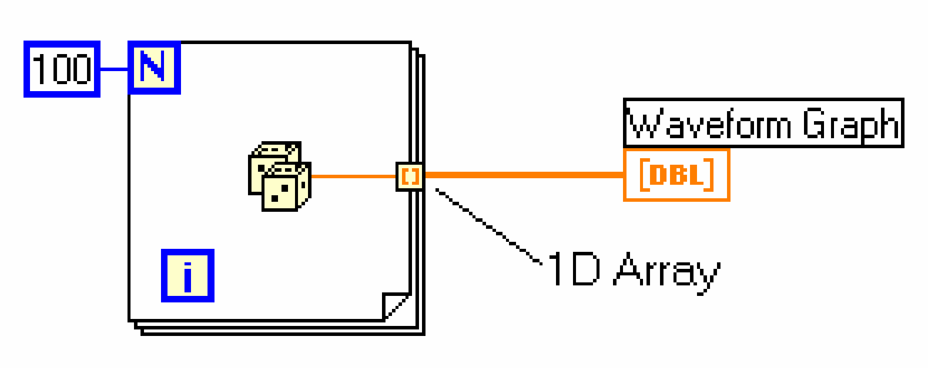


Plot Legend (point and line styles)

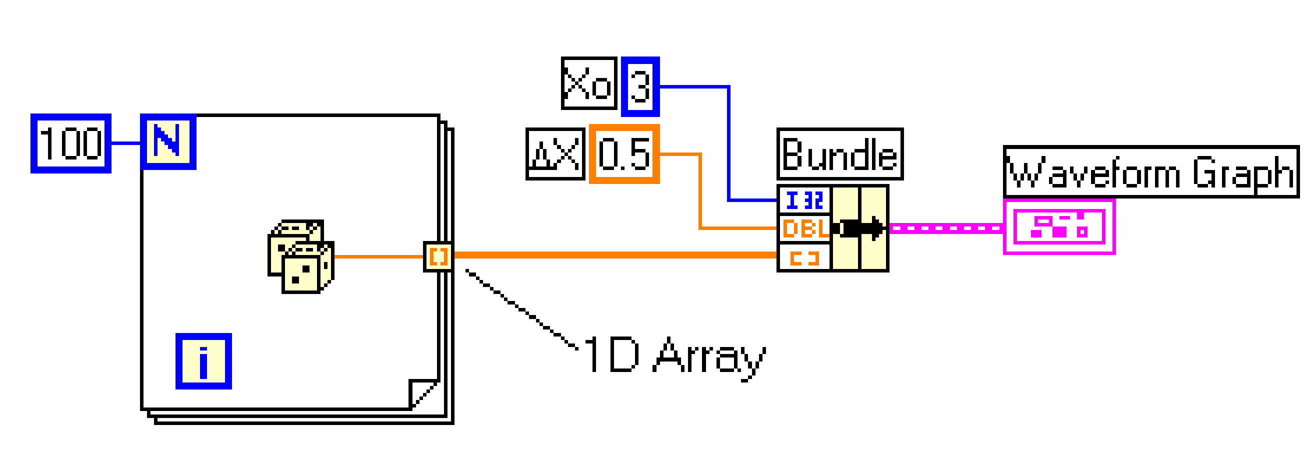
Graph Palette

Scale Legend

Single-Plot Waveform Graphs

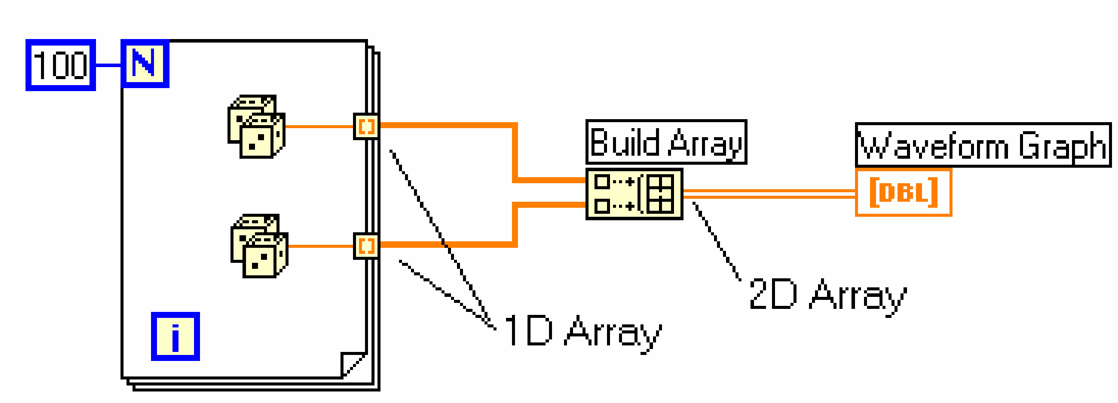


Uniform X axis - initial X = 0.0, delta X = 1.0

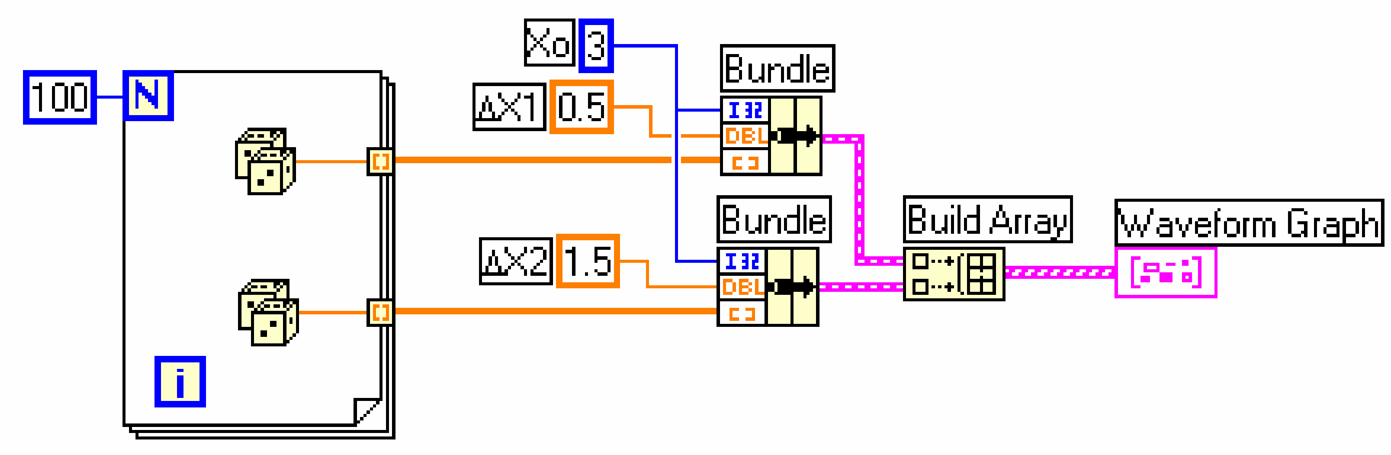


Uniform X axis - you specify point spacing

Multiple-Plot Waveform Graphs

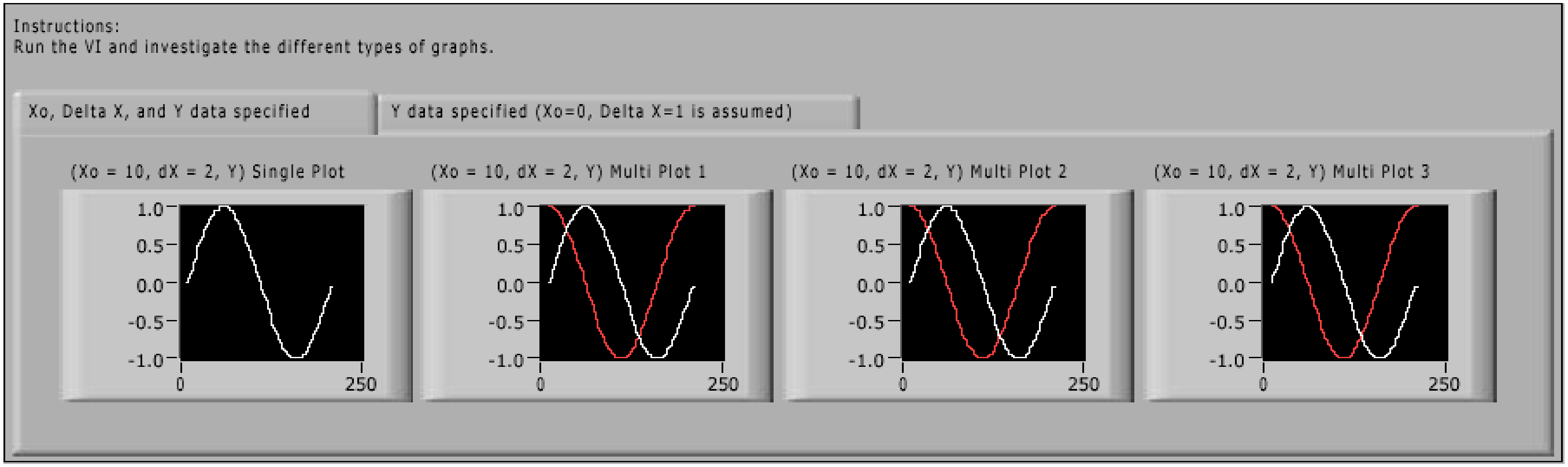


Each row is a separate plot (initial X = 0, delta X = 1)



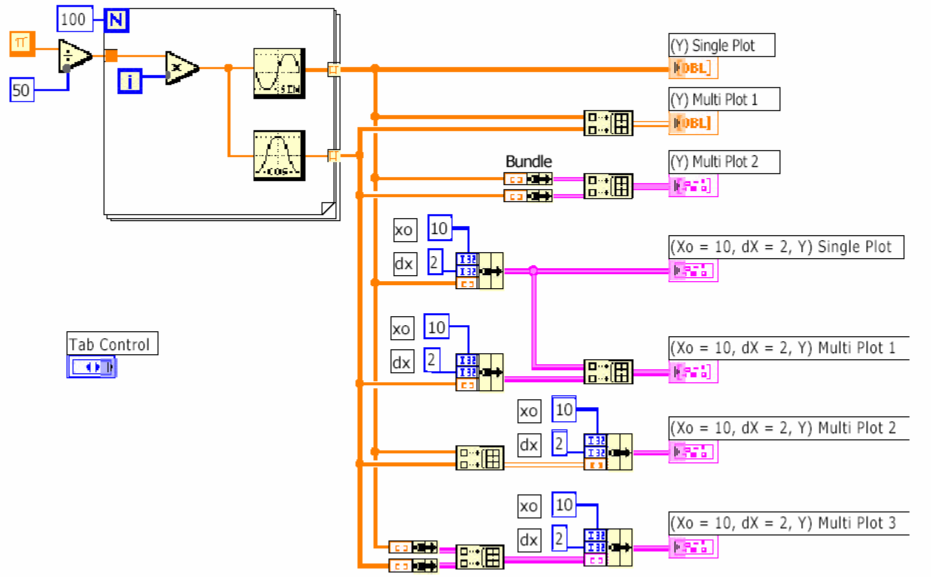
Each row is a separate plot (Bundle specifies point spacing of the X axis)

1. Find the **VI from Help»Find Examples...»Search**
2. **Type in “**graph” and **search**
3. Choose Waveform Graph**.vi**



Instructions:
Run the VI and investigate the different types of graphs.

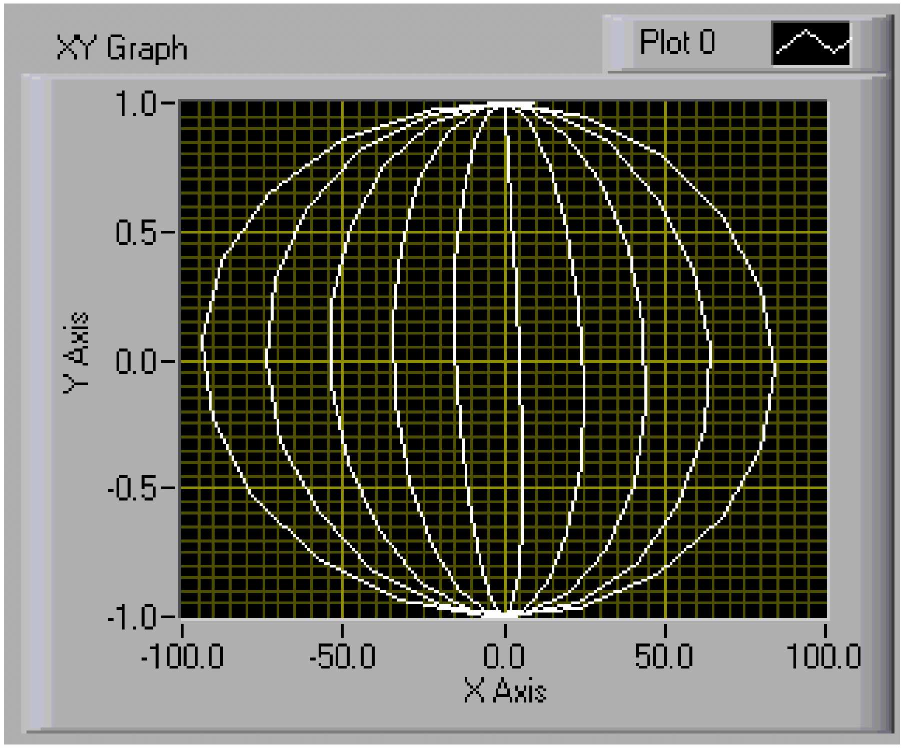
Xo, Delta X, and Y data specified
Y data specified (Xo=0, Delta X=1 is assumed)
(Xo = 10, dX = 2, Y) Single Plot
(Xo = 10, dX = 2, Y) Multi Plot 1
(Xo = 10, dX = 2, Y) Multi Plot 2
(Xo = 10, dX = 2, Y) Multi Plot 3
1.0
0.5
0.0
-0.5
-1.0
0
250

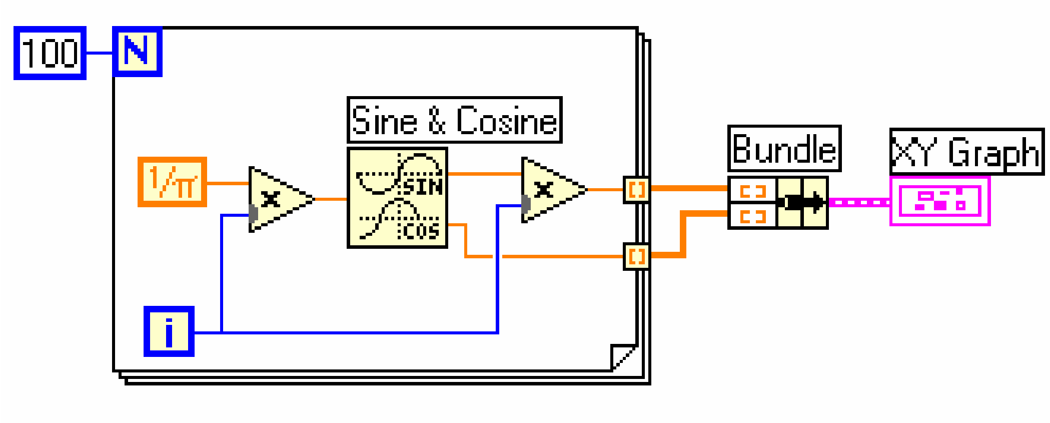


100
N
50
i
(Y) Single Plot
DBL
(Y) Multi Plot 1
DBL
Bundle
(Y) Multi Plot 2
xo
10
dx
2
(Xo = 10, dX = 2, Y) Single Plot
(Xo = 10, dX = 2, Y) Multi Plot 1
(Xo = 10, dX = 2, Y) Multi Plot 2
(Xo = 10, dX = 2, Y) Multi Plot 3
Tab Control

XY Graphs

- **Non-uniform X axis**
- **Separate X and Y arrays define data points**

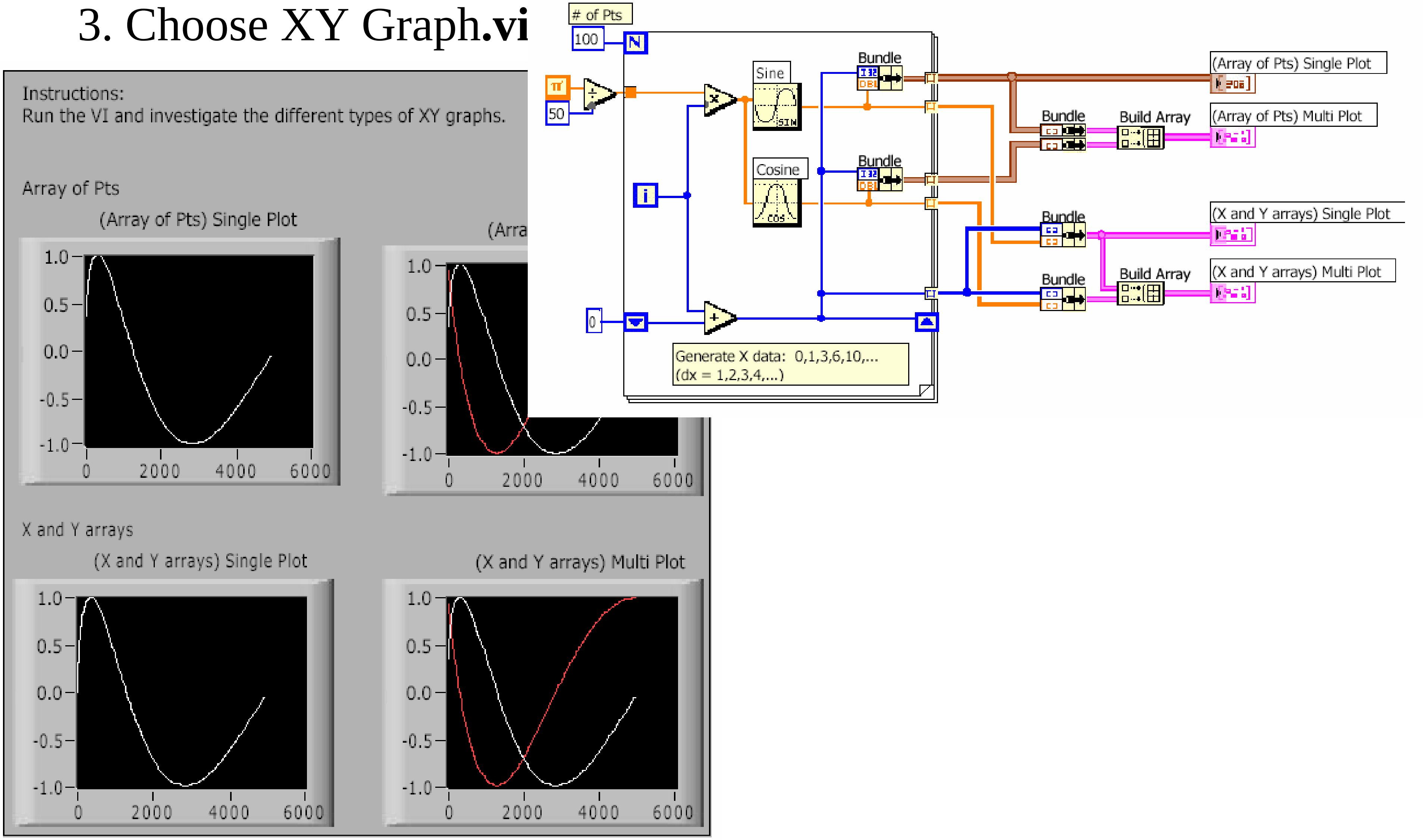




1. Find the **VI from Help»Find Examples...»Search**

2. Type in “xy graph” and **search**

3. Choose XY Graph**.vi**

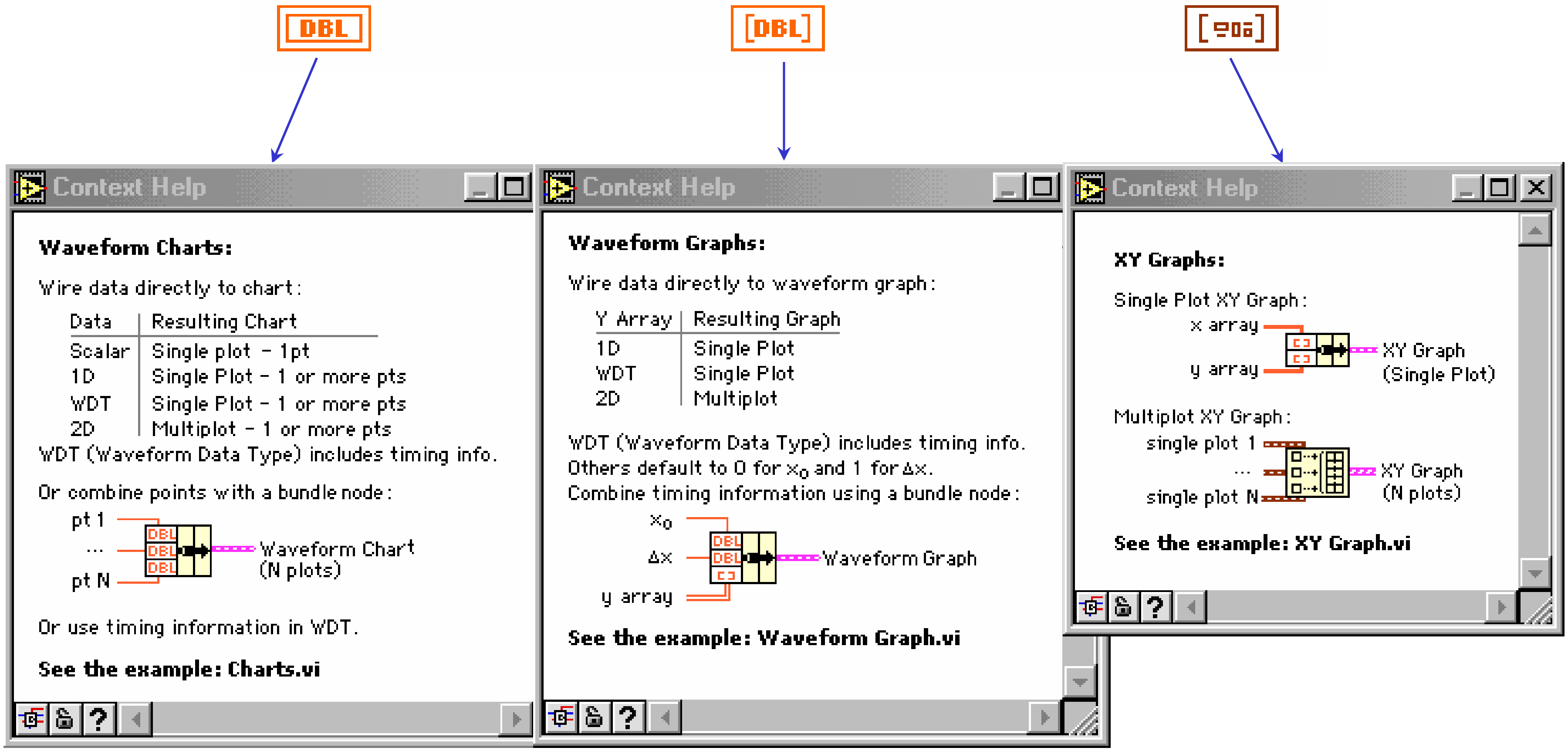


Instructions:
Run the VI and investigate the different types of XY graphs.

Array of Pts
(Array of Pts) Single Plot
(Arra
X and Y arrays
(X and Y arrays) Single Plot
(X and Y arrays) Multi Plot
1.0
0.5
0.0
-0.5
-1.0
0
2000
4000
6000
# of Pts
100
N
50
i
0
Sine
Cosine
Bundle
Build Array
Generate X data: 0,1,3,6,10,...
(dx = 1,2,3,4,...)
(Array of Pts) Single Plot
(Array of Pts) Multi Plot
(X and Y arrays) Single Plot
(X and Y arrays) Multi Plot

Chart and Graph Use Summary

- **Use the Context Help window with charts and graphs**



Waveform Charts:

Wire data directly to chart:

Data	Resulting Chart
Scalar	Single plot - 1pt
1D	Single Plot - 1 or more pts
WDT	Single Plot - 1 or more pts
2D	Multiplot - 1 or more pts

WDT (Waveform Data Type) includes timing info.

Or combine points with a bundle node:

Or use timing information in WDT.

See the example: Charts.vi

Waveform Graphs:

Wire data directly to waveform graph:

Y Array	Resulting Graph
1D	Single Plot
WDT	Single Plot
2D	Multiplot

WDT (Waveform Data Type) includes timing info.
Others default to 0 for x_0 and 1 for Δx.
Combine timing information using a bundle node:

See the example: Waveform Graph.vi

XY Graphs:

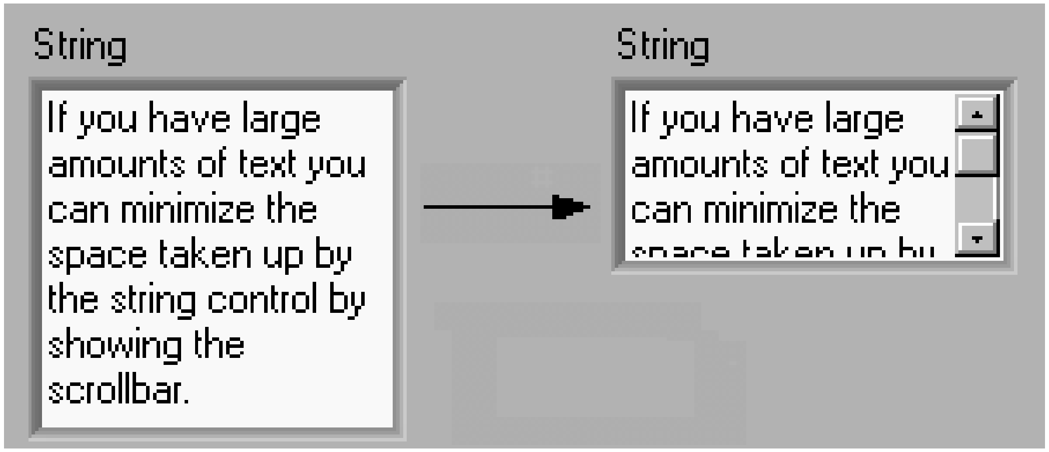
Single Plot XY Graph:

Multiplot XY Graph:

See the example: XY Graph.vi

D. Strings

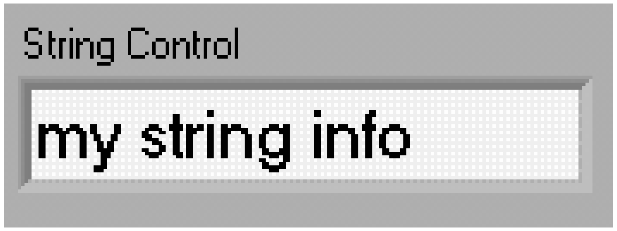
- A string is a sequence of displayable or nondisplayable characters (ASCII)
- Many uses – displaying messages, instrument control, file I/O
- String control/indicator is in the Controls»String subpalette



String
If you have large amounts of text you can minimize the space taken up by the string control by showing the scrollbar.
String
If you have large amounts of text you can minimize the

String Display Modes

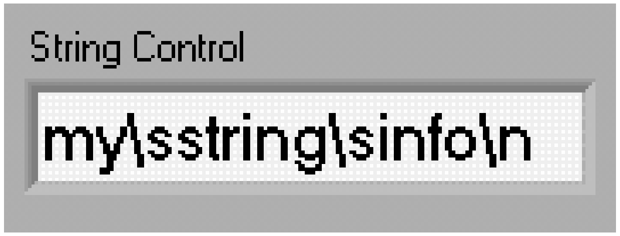
- Normal display



String Control

my string info

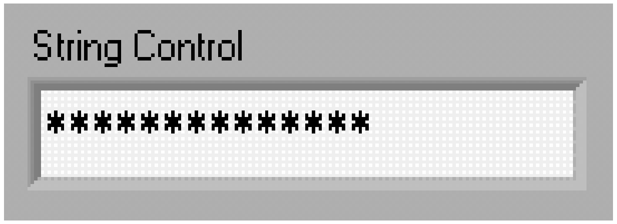
- \ code display



String Control

my\sstring\sinfo\n

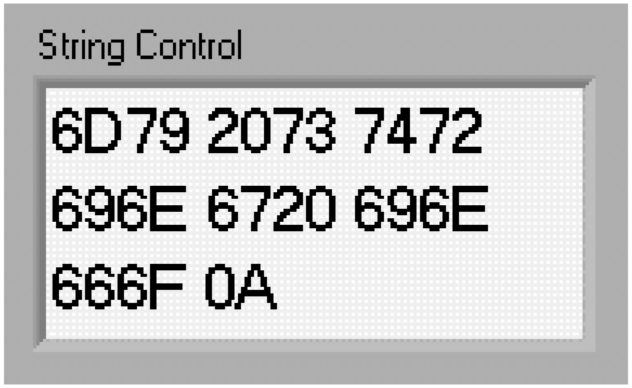
- Password display



String Control

****

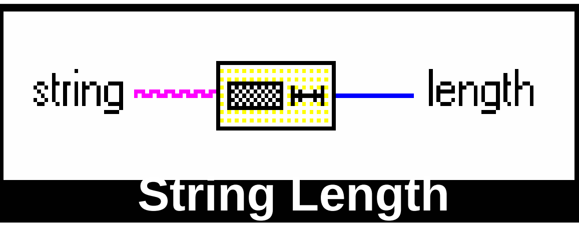
- Hex display



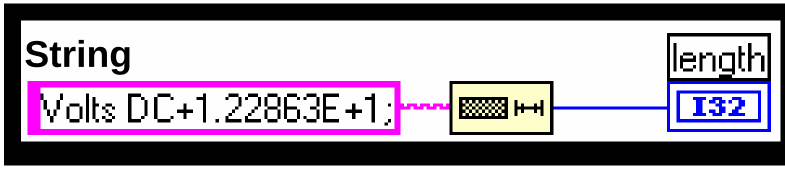
String Control

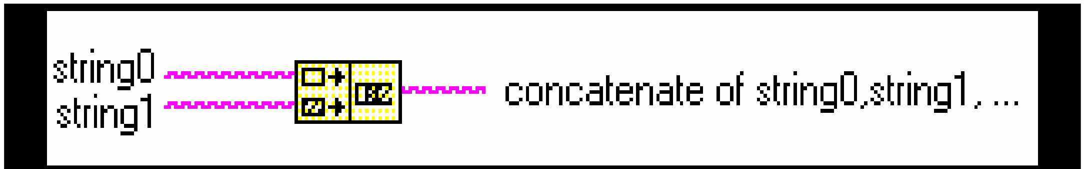
6D79 2073 7472
696E 6720 696E
666F 0A

String Functions



String Length

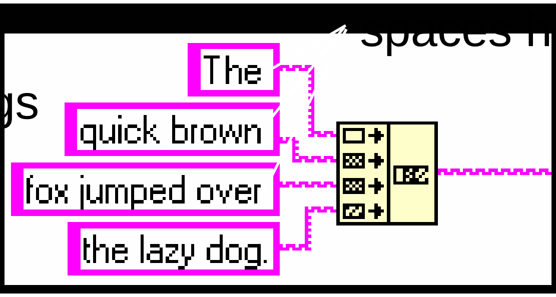




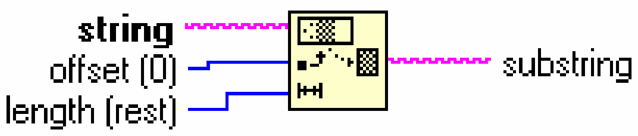
Concatenate Strings

spaces

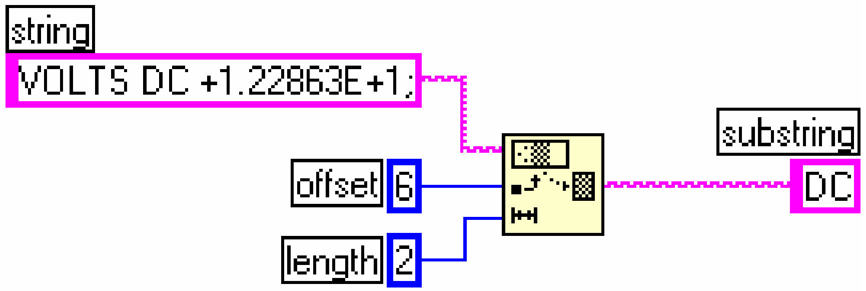
gs

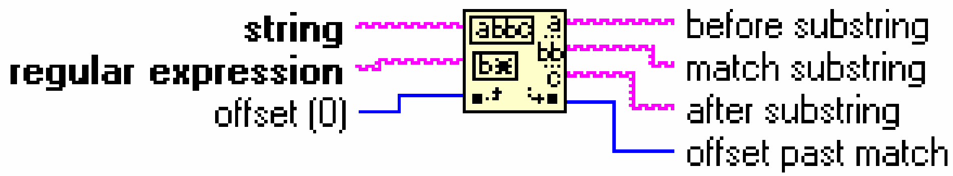


String Functions

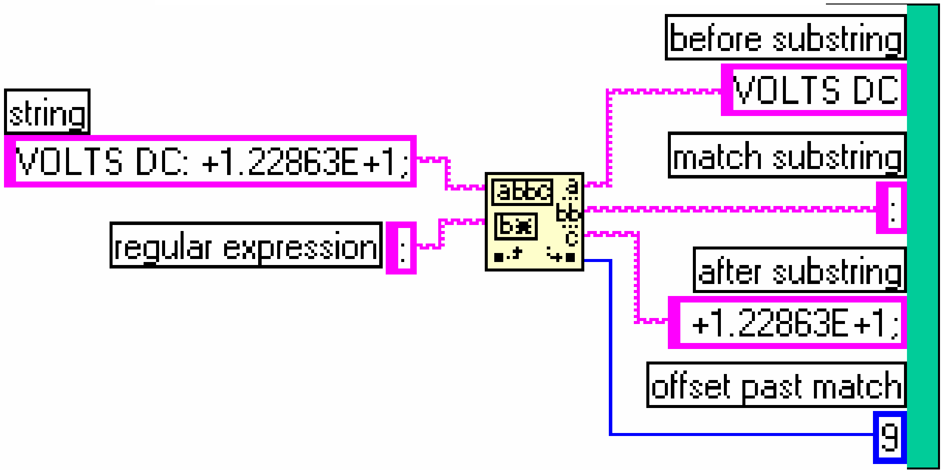


String Subset





Match Pattern

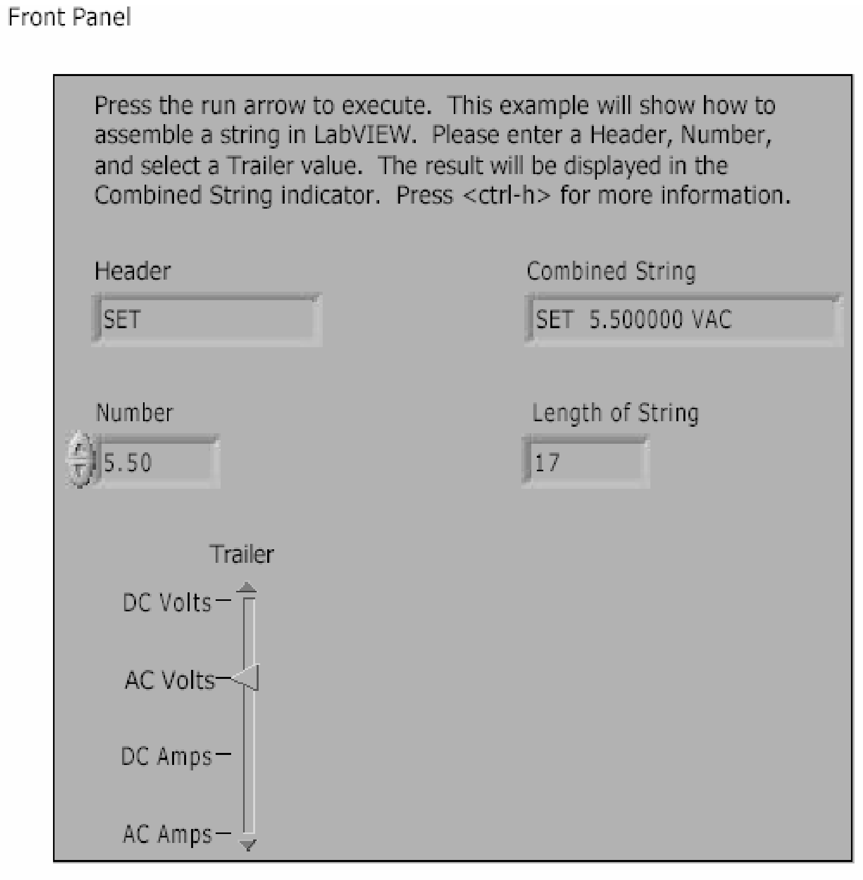


D. Strings

1. Find the **VI from Help»Find Examples...»Search**

2. Type in “string” and **search**

3. Choose Build String**.vi**



Front Panel

Press the run arrow to execute. This example will show how to assemble a string in LabVIEW. Please enter a Header, Number, and select a Trailer value. The result will be displayed in the Combined String indicator. Press <ctrl-h> for more information.

Header

SET

Combined String

SET 5.500000 VAC

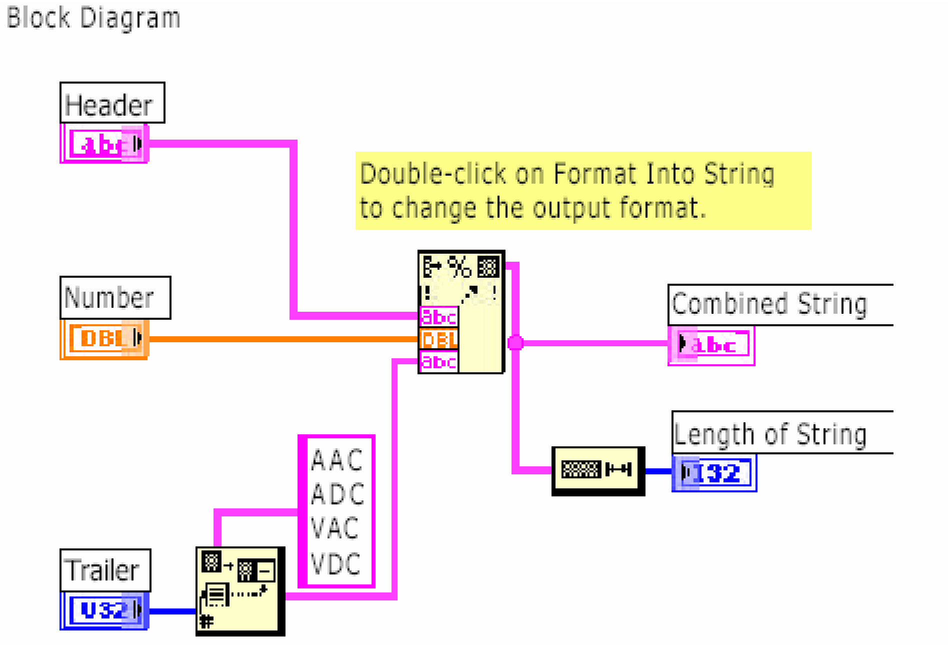
Number

5.50

Length of String

17

Trailer



Block Diagram

Double-click on Format Into String to change the output format.

Summary

- Two structures to repeat execution
- – While Loop------continue, stop if it is true
- – For Loop
- Loop timing controlled using Wait Until Next ms Multiple function
- Shift registers transfer data values from one iteration to the next
- – Adapt to any data type
- – Additional left terminals may be added to access previous iteration
- Two structures to control flow of data
- – Case structure and Sequence structure
- Case structure
- – Boolean or numeric cases – selector determines type
- – Subdiagrams placed inside case structure
- – Output from a Case structure must be defined for all cases

Summary--Cnd

- Sequence structure executes subdiagrams sequentially
- Sequence locals pass data between frames
  - Created at the border of Sequence structure
  - Data available in subsequent frames
- Formula Nodes allow direct entry of equations in the block diagram
- An array is a collection of elements of same data type – numeric, Boolean, string, etc.
- Creating array or cluster controls/indicators is a two-step process
  1. Get array or cluster shell
  2. Place desired control/indicator inside the shell
- Loops can accumulate arrays at boundaries – auto-indexing
- Array functions are in Array subpalette of Functions palette
- Clusters group data that can be different types
- The Bundle and Unbundle functions are used to create and disassemble clusters

Summary--Cnd

Three modes of waveform charts

- **Strip chart**
- **Scope chart**
- **Sweep chart**

Charts can be customized – right-click on chart for menus

•Plot data on graphs

- **Many features to manipulate plotted graph**
- **Multiple plots can be plotted on one graph**

•String is a collection of ASCII characters – many uses

Display messages

Instrument control

File I/O

•Many functions to manipulate strings – Strings palette of Functions menu

Homework for Today

1. Make a VI to calculate factorial, N!
2. Make VIs for C to F and F to C using Formula Node