

# PH2150

## Introduction to LabVIEW (1)



Gary Boorman





# LabVIEW



## Laboratory Virtual Instrument Engineering Workbench

- LabVIEW is a graphical programming environment used to develop sophisticated measurement, test and control systems
- Used by millions of scientists and engineers in thousands of laboratories and test environments
- It uses graphical icons and wires that resemble a flowchart to create Virtual Instruments (VI)

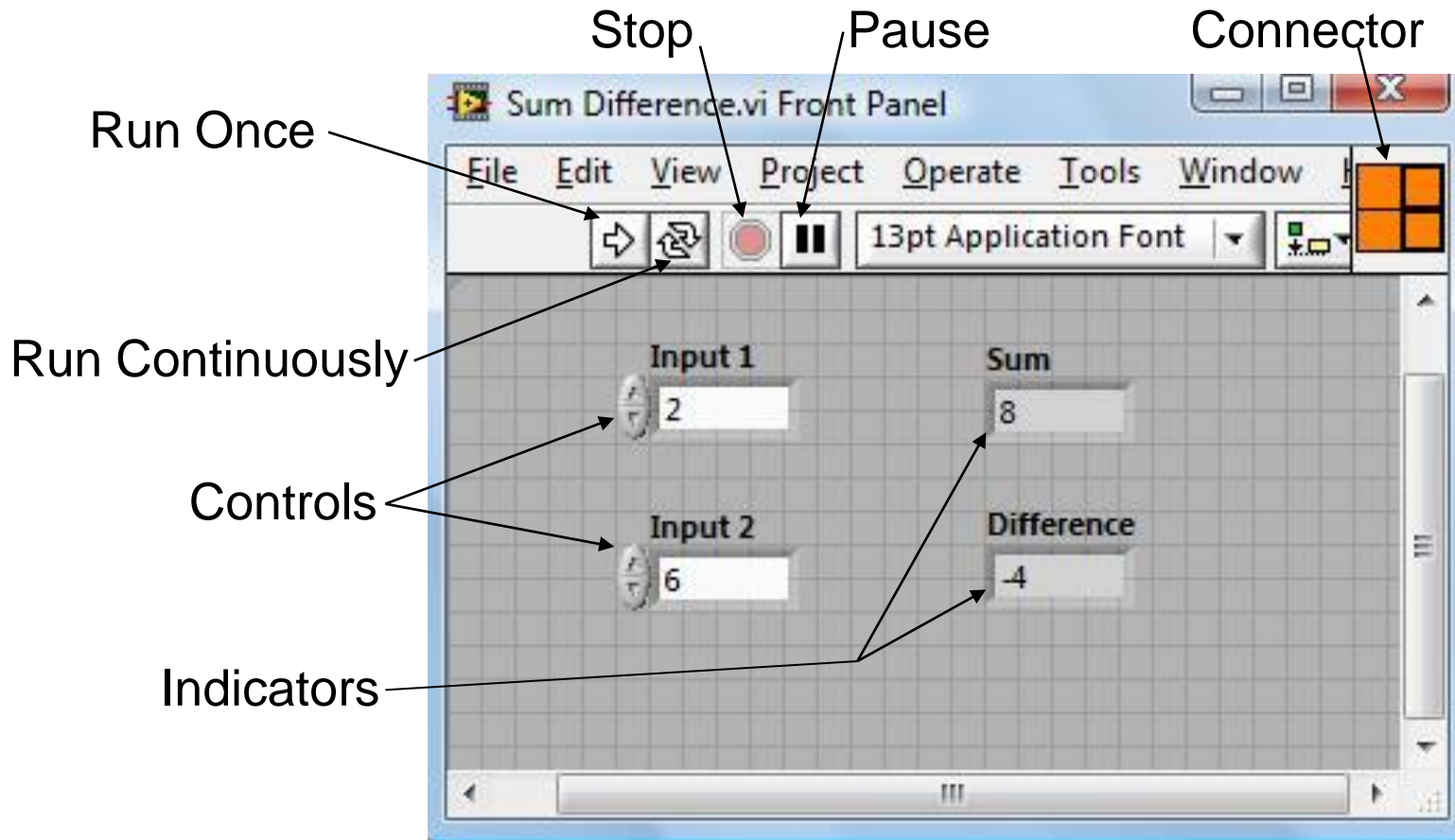


# LabVIEW - Characteristics

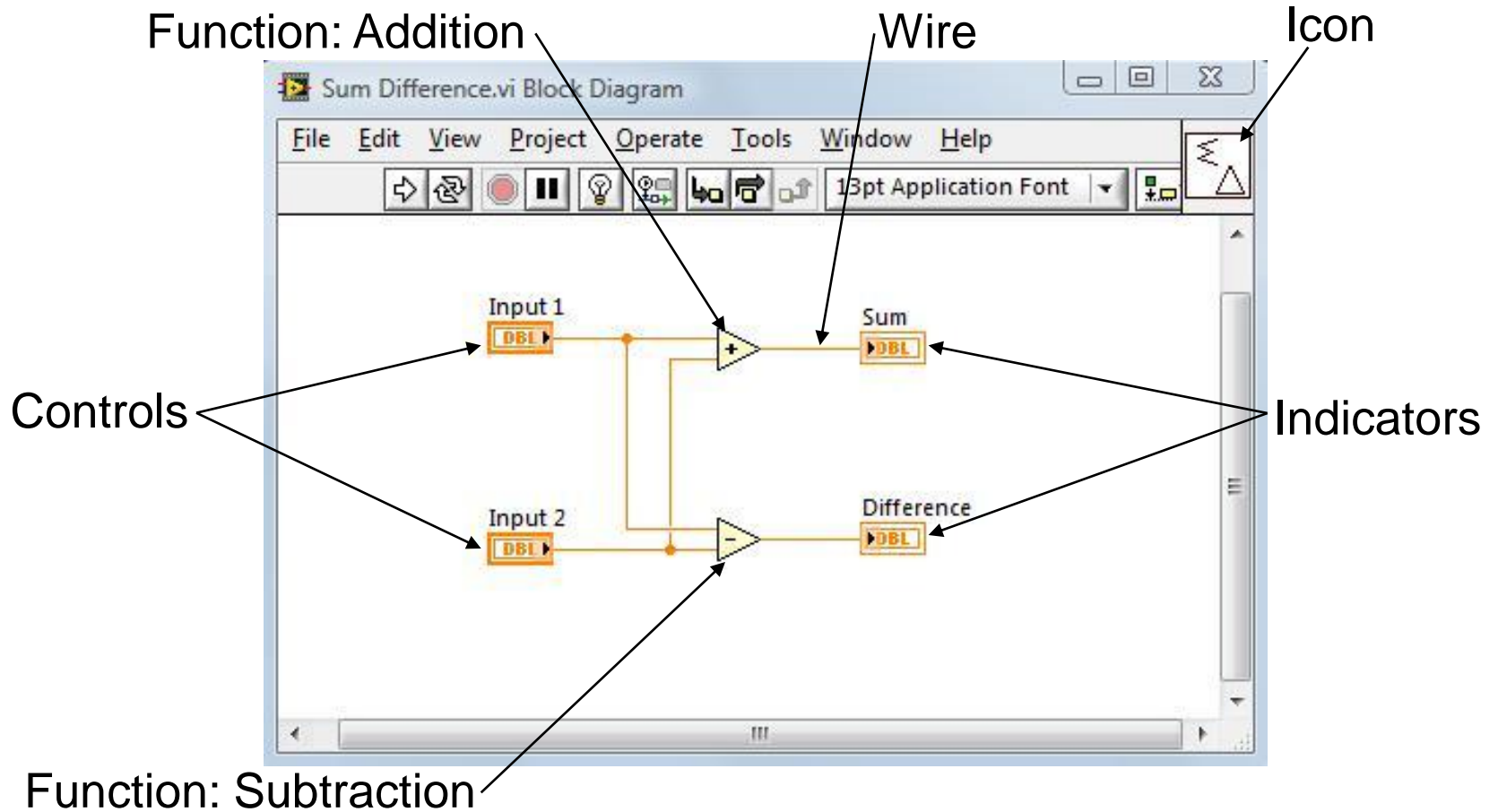
- LabVIEW is a data-flow language – each sub-VI will run (and output data) when data is ready at each input of the VI
- VI consists of a user interface (*front panel*), the operating code (*diagram*) and interface (*connector pane*)
- LabVIEW is ***the tool*** for data-acquisition, control and analysis



# LabVIEW – Front Panel

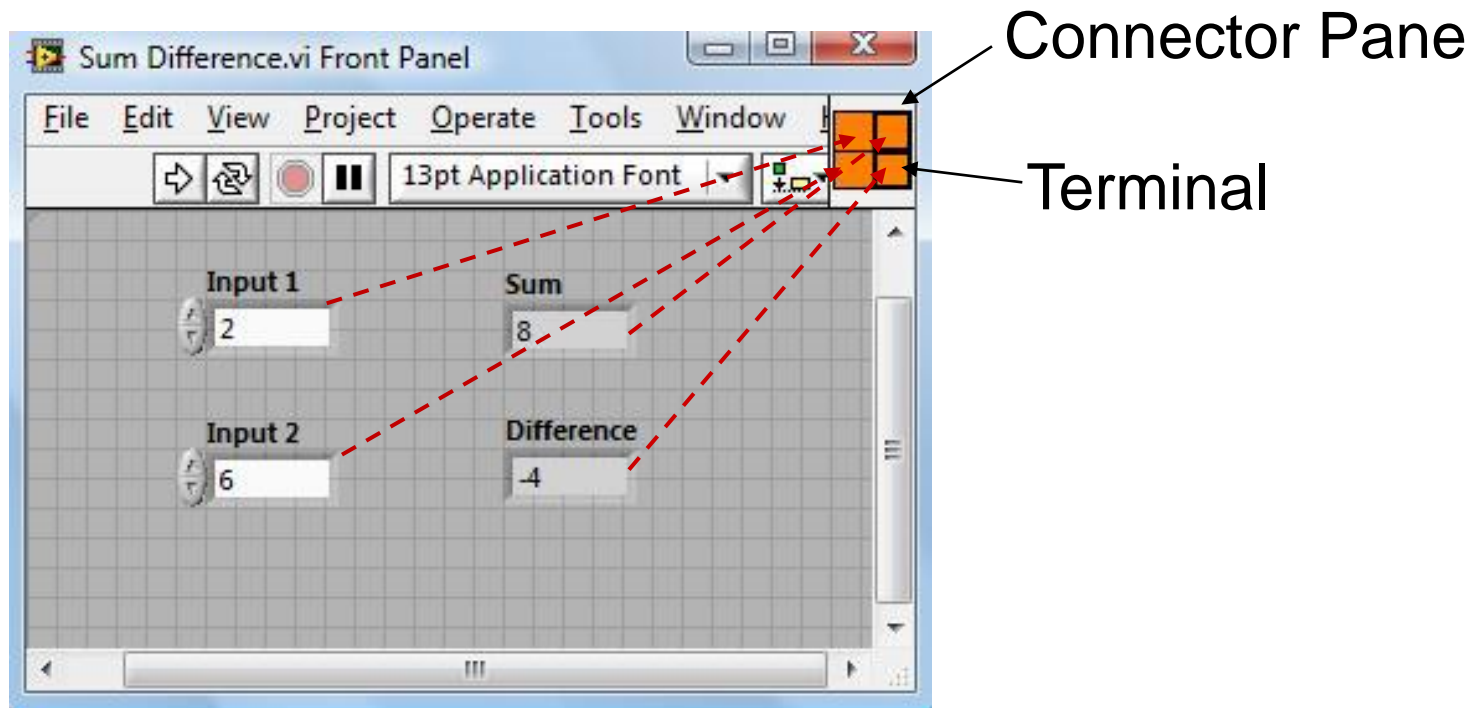


# LabVIEW - Diagram



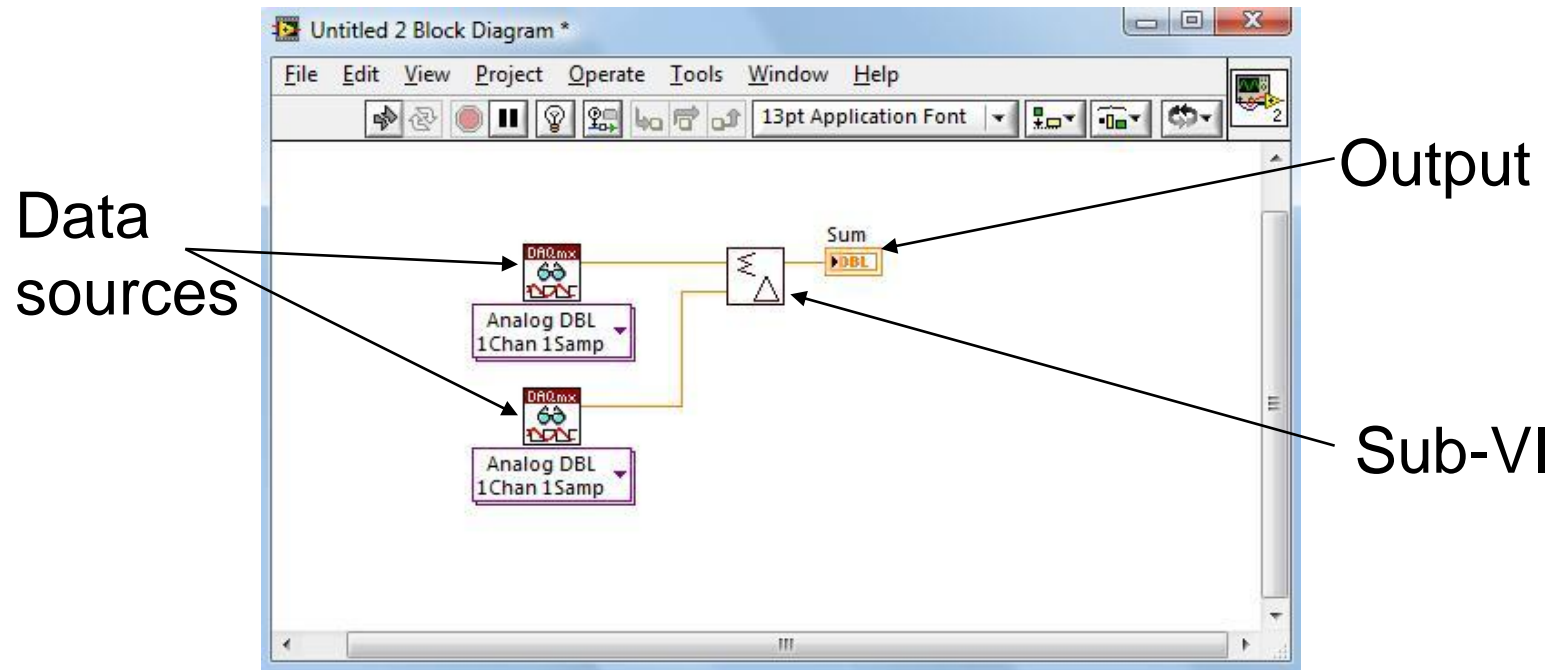
# LabVIEW – Connector Pane

Each Control/Indicator can be connected to a Terminal on the Connector Pane



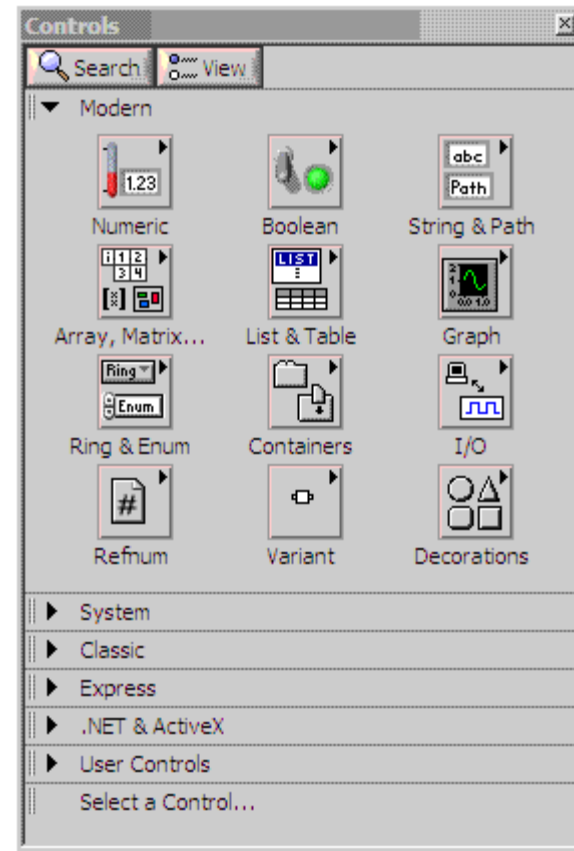
# LabVIEW – Using a Sub-VI

Drag the sub-VI to the diagram and wire its inputs and outputs



# LabVIEW – Controls Palette

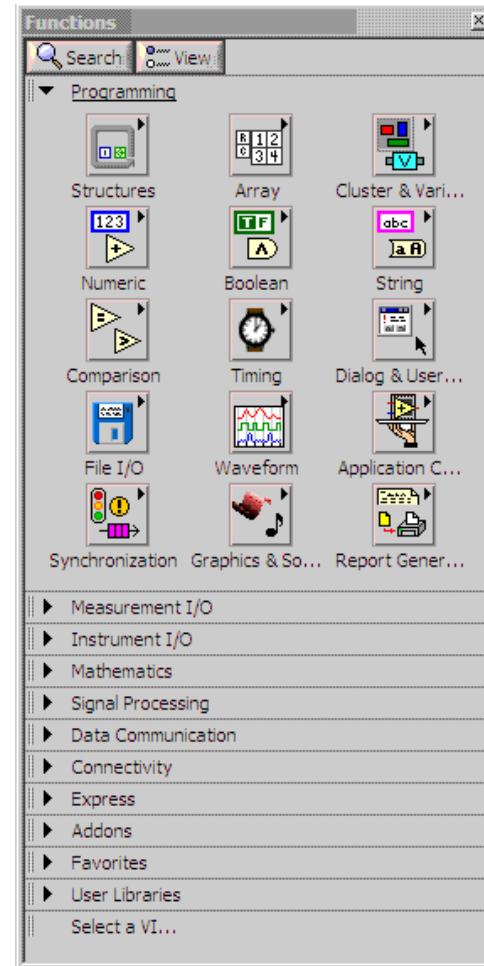
- This palette is displayed when the Front Panel is active
- Contains Controls and Indicators





# LabVIEW – Functions Palette

- This palette is displayed when the Diagram is active
- The functions you will mostly use are in the Structures, Numeric, Array, Boolean, Comparison and Timing sub-palettes



# LabVIEW – Tools Palette

- This palette is displayed when either the Diagram or Front Panel is active
- The Tool can be set to be a pointer, wiring, text, value or probe tool, or be automatic
- Automatic Tools are active if the top bar 'led' is green



# LabVIEW – Style

- Try and get the diagram on to one screen only, use sub-VIs to split the functionality
- Avoid crossing wires unnecessarily – good sub-VI connector/terminal positions can help
- Keep data flowing from left to right
- Give meaningful names to controls/indicators
- Use plenty of space & comments on the diagram




# LabVIEW – Data Types

- LabVIEW has several data types: integers, doubles, strings, paths, boolean etc
- Can be displayed as large or small icons on block diagram

Integer   
(indicator)

String   
(indicator)

Boolean   
(control)

Double   
(control) Large icon

 Small icon

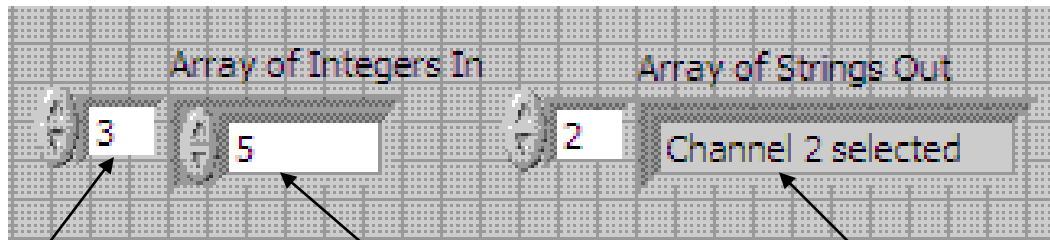
Cluster   
(control)



# LabVIEW – Arrays

[13, 8, 1, 5, 7] index0 is 13, index1 is 8, index2 is 1...

[start, end, Channel 2 selected] index0 is 'start', index1 is 'end'...



Array Index    Integer input of index 3    String output of index 2

- Arrays can hold any data type (subject to a few limitations)
- Arrays start at index 0
- Many functions to operate on arrays

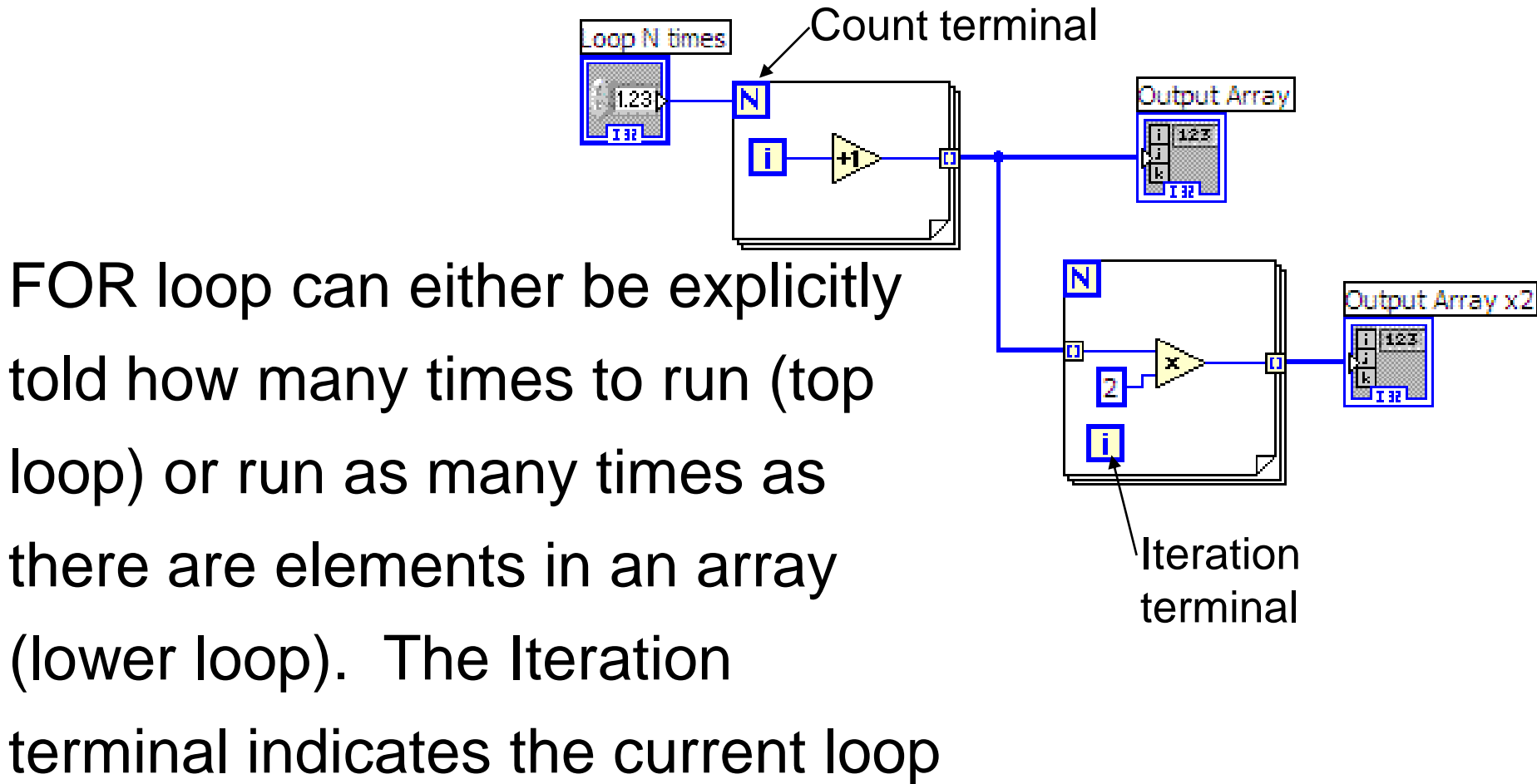


# LabVIEW – Structures

- Structures are used for repetitive tasks (FOR and WHILE loops), to make choices (CASE structure) or to do tasks in a particular order (SEQUENCE structure)
- FOR and WHILE loops are useful to operate on, or to build, arrays
- CASE structures can make decisions
- Structures can be ‘nested’

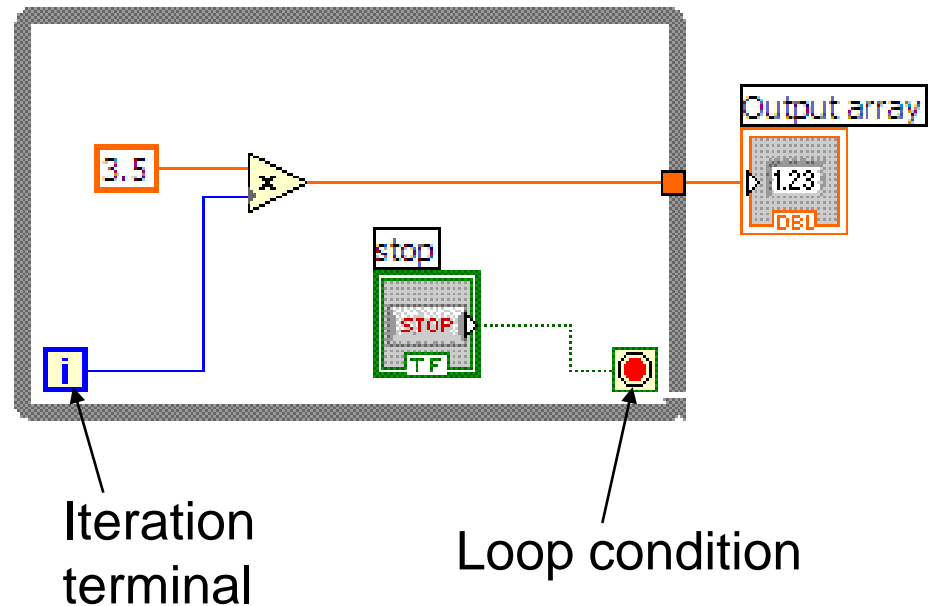


# LabVIEW – FOR Loop



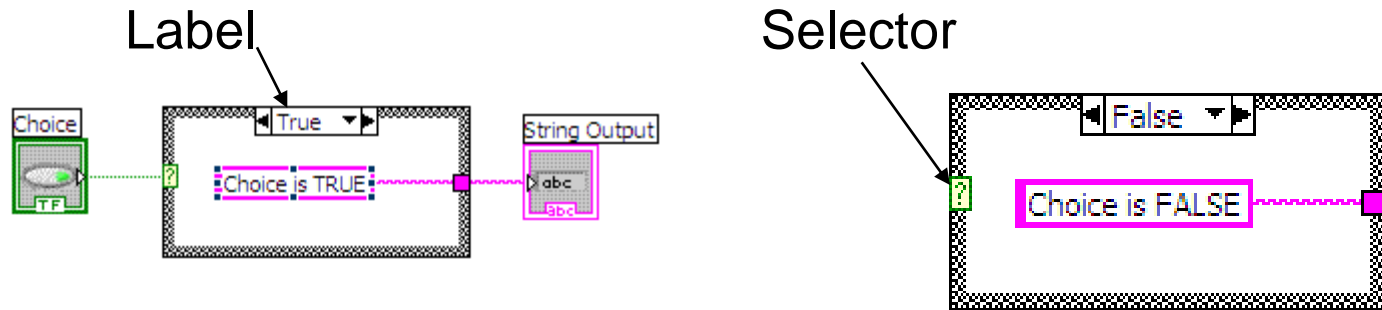
# LabVIEW – WHILE Loop

The WHILE loop will run until told to stop (in this case by a 'stop' button on the front panel). The loop condition can also be configured to stop unless told to continue





# LabVIEW – CASE Structure



The 'choice' boolean selects either the true or false case. CASE structures can have more than two choices – use an integer to select and ensure that a 'default' case is defined

