Chapter 3. Structures

3.1 What are Structures?

Structure control the execution flow in a VI. It is available in Function>>Programming sub-palette.

Typical structures include:

- For Loop.
- While Loop.
- Case structure.
- Flat Sequence structure.

3.2. The For loop

For Loop controls repetitive operation until a specification number of iteration completes. It has two terminals:

- the **count terminal** (an input terminal)

- the **iteration terminal** (an output terminal). Where 0 represents first iteration. (N-1 is Nth iteration)

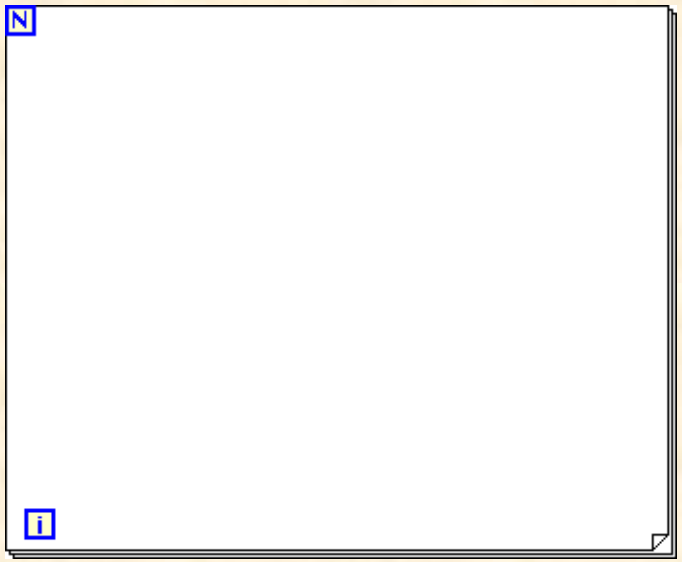


Figure 1. A For Loop at the Block Diagram

3.2.1 A For Loop Example

Write a For Loop VI: the iteration number (Count Terminal input) is input via a numerical control, and the iteration terminal is displayed via a numerical indicator.

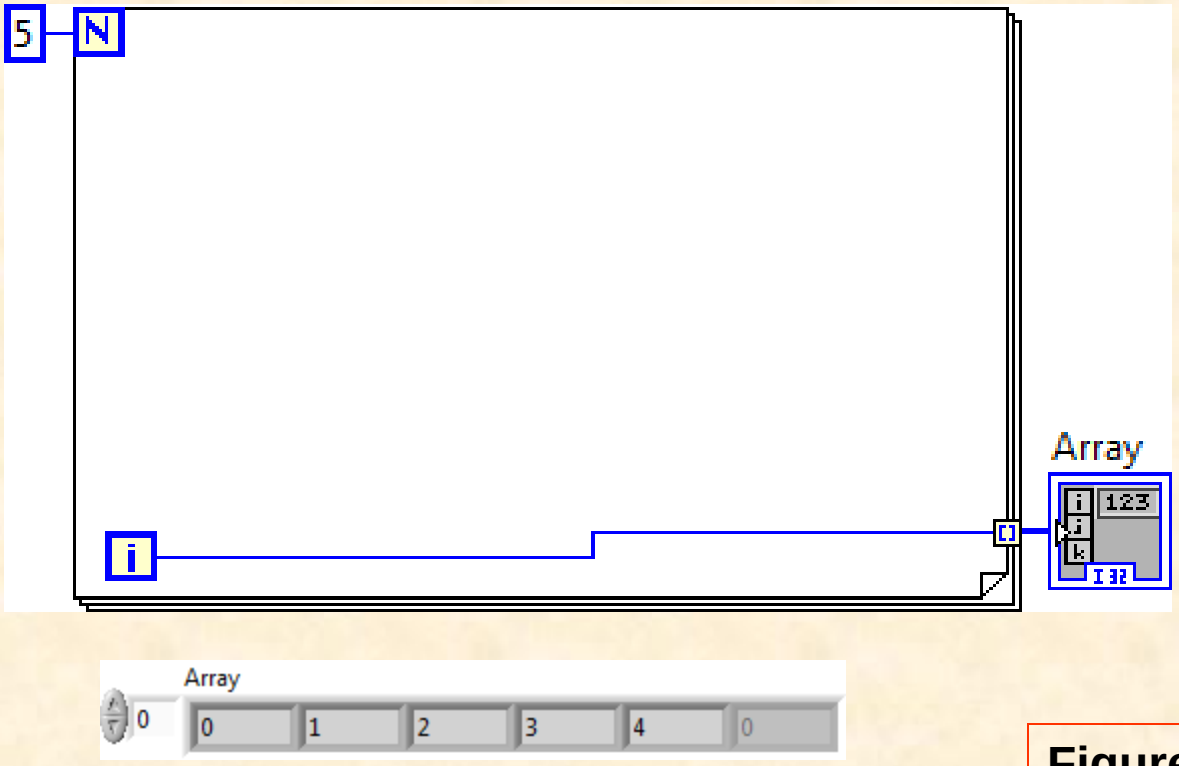


Figure 2. For loop example.

- **Shift register**

3.3 The While loop

The While Loop repeat a section operation until a certain condition is met. It has two terminals:

- The **conditional terminal** (an input terminal):
  Boolean variable: True or False

- The **iteration terminal** (an output terminal).
  Where 0 represents first iteration. (N-1 is Nth iteration)

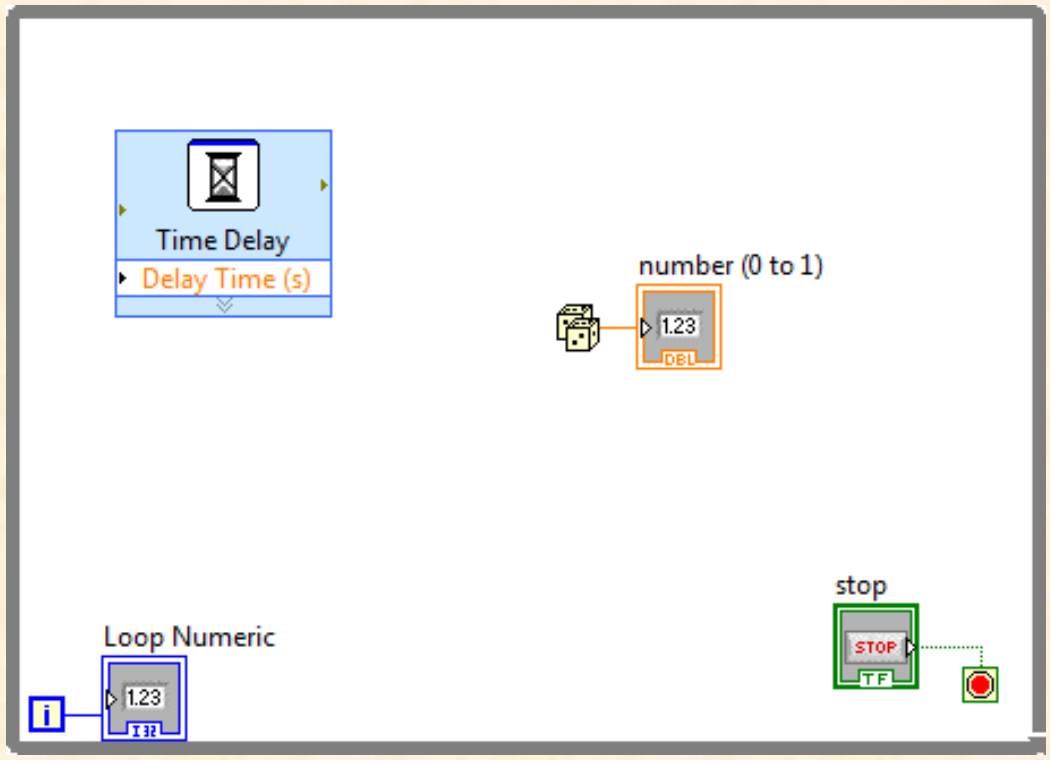
Figure 3. While loop

3.3.1 A While Loop Example

In this excise, we will place a random number inside a While Loop and display the random numbers and While Loop counter on the front panel.

Procedure:

1) Place Random Number function on the block diagram, and create an indicator to display it.
2) Place the While Loop on the block diagram from Function>>Express>>Execution Control. Note that A Stop button also create on the front panel.
3) Create an indicator to show the iteration number.
4) Click Run button to start the program, and push Stop button to stop the program.
5) You can change the property of iteration indicator by right-click it on the front panel, and change the data as I32 (integer).



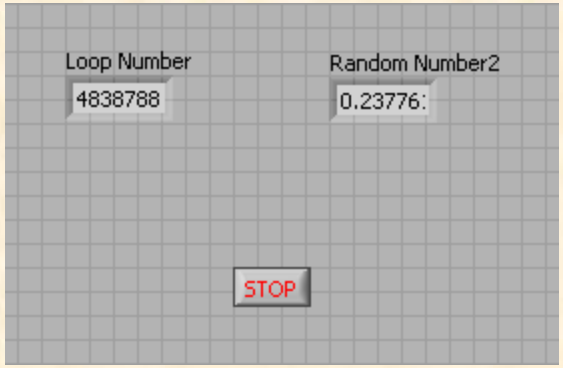


Figure 4. While loop example

3.4 Case Loop

The case structure is a method of executing conditional loop, and it is available from **Function>Programming>Structure.**

The case structure can have multiple sub-diagrams. At the top of the Case structure is the elector label. At the left is the Selector terminal that controls which sub-diagram should operate, and the selector can be numeric, Boolean, string type control. The Default case specify the case to execute if no any selector choice is chosen.

You can place selector terminal anywhere on the Case structure along the left border.

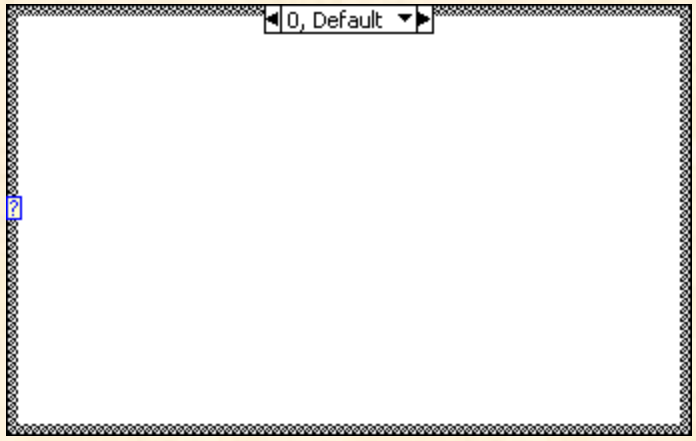
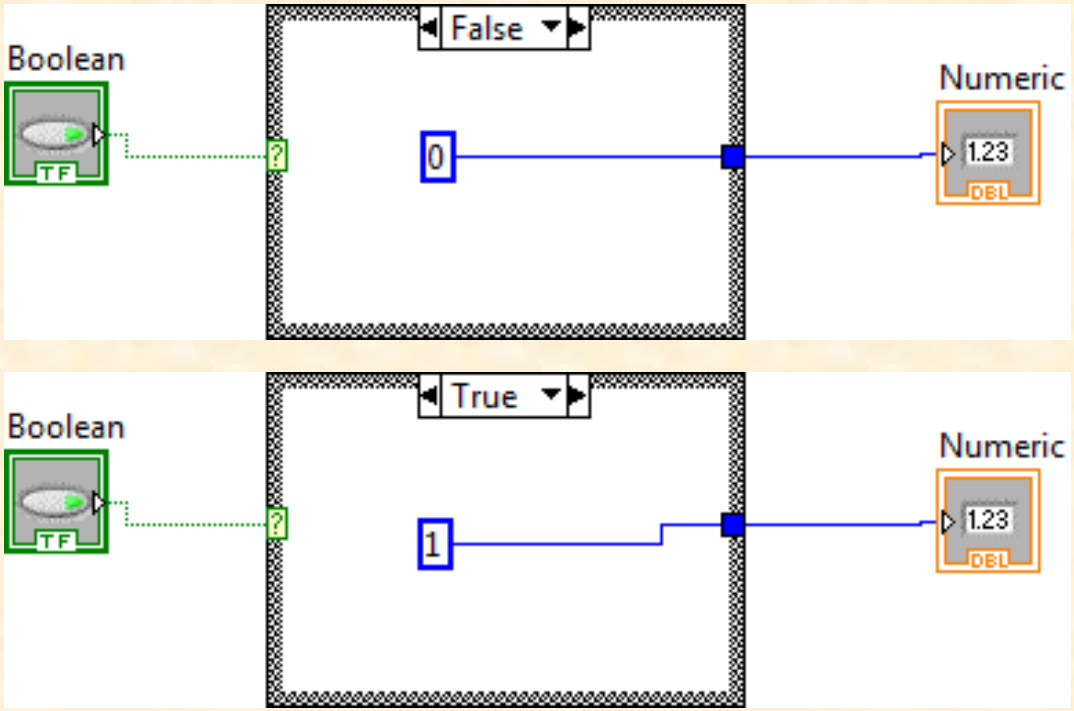


Figure 5. A case structure.

Boolean Case Structure

A Boolean case structure has only two frames (or sub-diagrams), which is controlled by the corresponding Boolean Selector.

The following is a example of Boolean structure, which outputs 0 for False case, and 1 for True case.

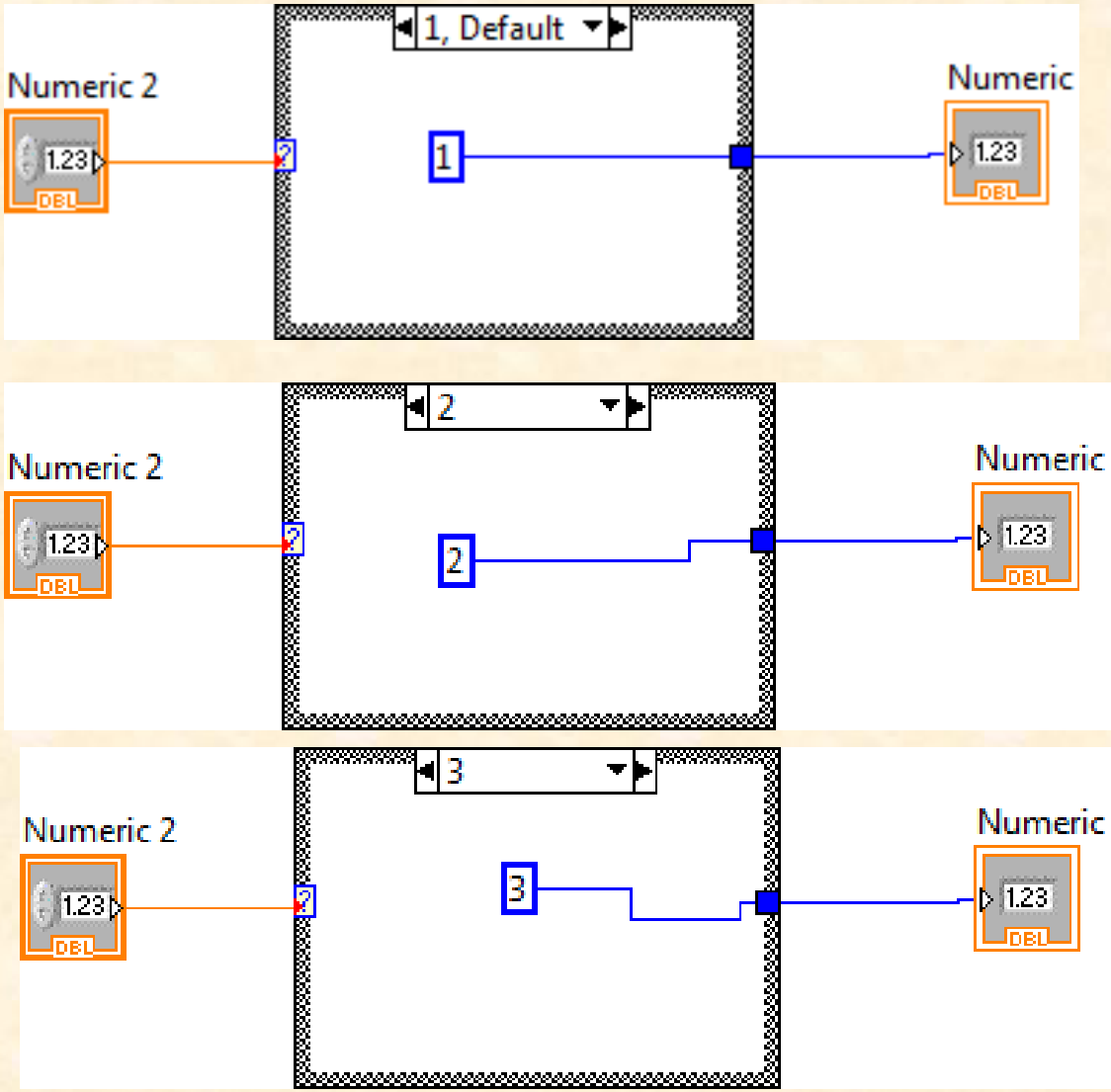


Numeric Case Structure

A Numeric case structure may have more than two frames (or sub-diagrams), which is controlled by the corresponding numeric Selector:

- Right click to add, delete frame.
- Input Data Type determine the Boolean or numeric case structure

The following is a example of numeric case structure, which outputs 1 for case 1, and 2 for case 2 and 3 for case 3.

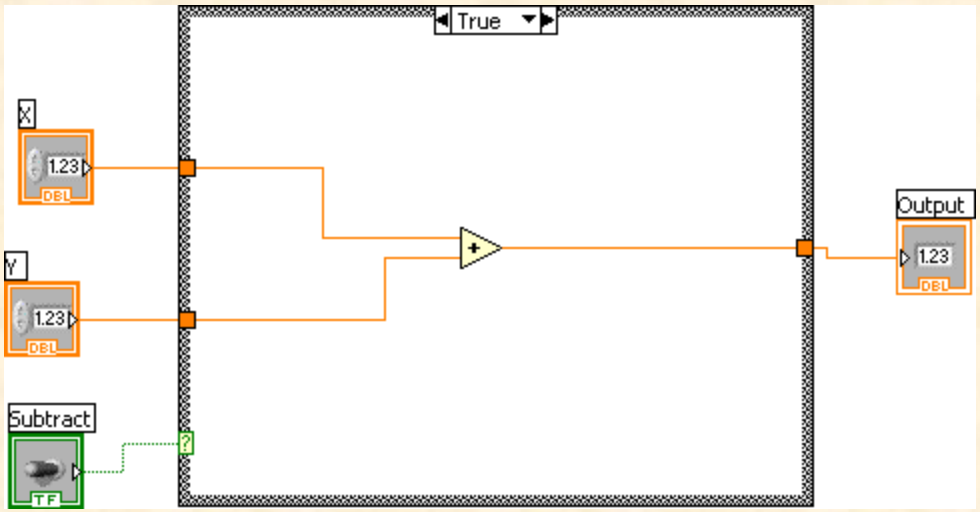


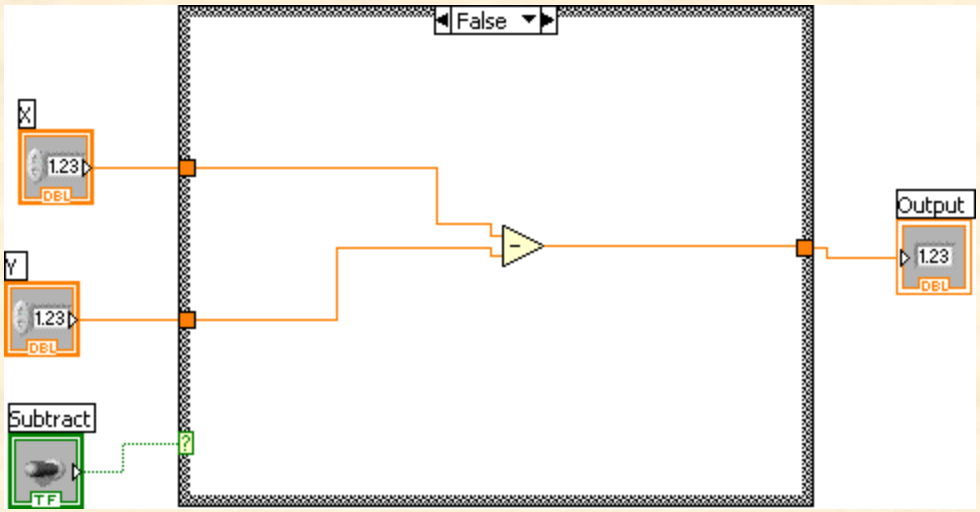
Case Example 1 (Assignment 1)

In this excise, you have two inputs and you will use Boolean Case structure to perform the following tasks: if the boolean wired to the selector terminal is True, the VI will add the numbers; otherwise, the VI will subtract the numbers. Using one controls as case selector, and one indicator to show the calculation result.

The input data numbers from the front panels will pass through the tunnels to the Case structure.

- A tunnel is a data entry point or exit point on a structure. You can wire a terminal from outside the Case structure to a terminal within the structure.





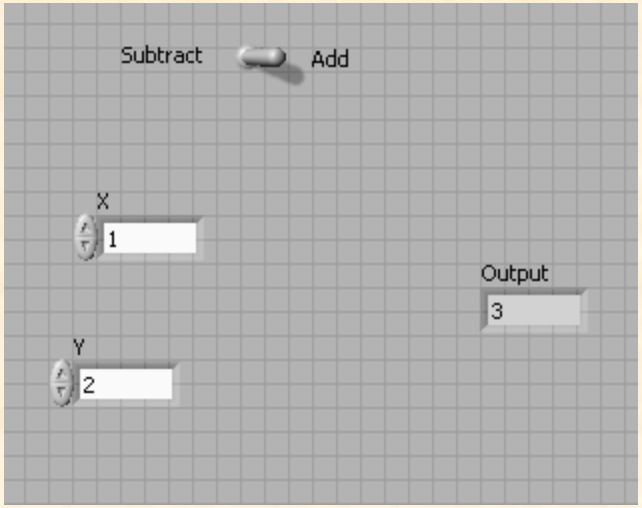


Figure 6. The case example 1

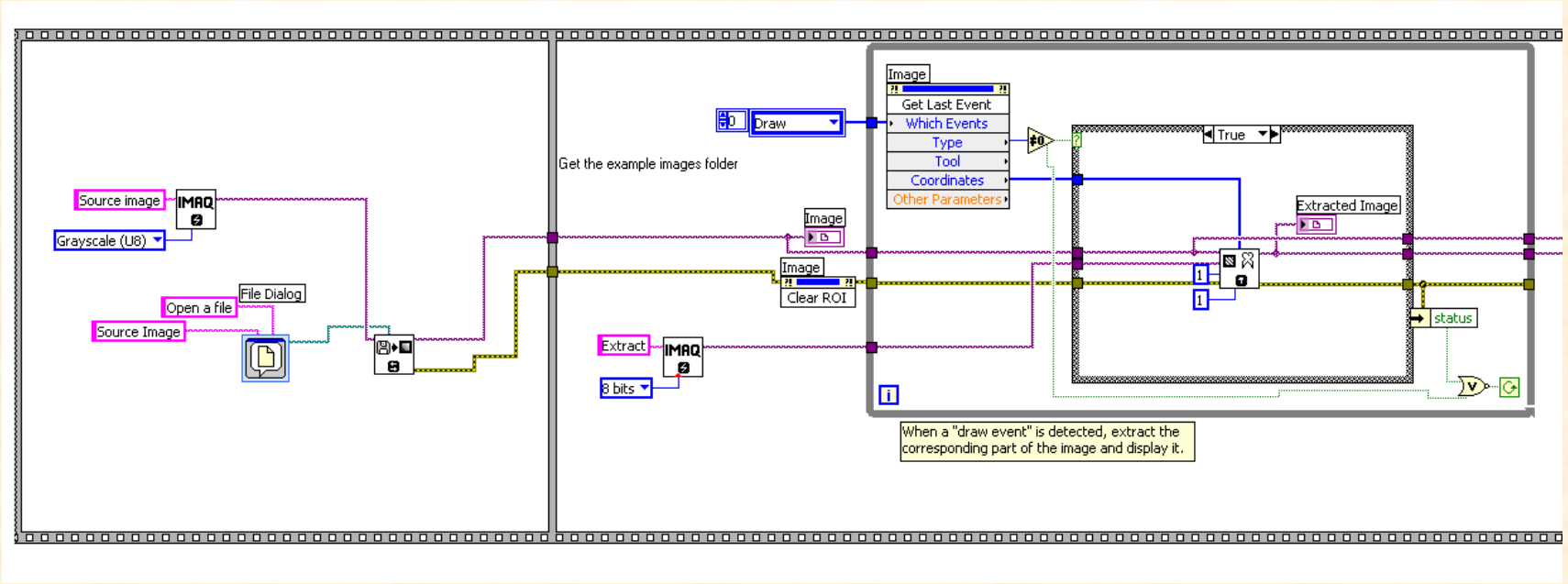
Case Example 2 (Assignment 2)

For the previous example (Case Example 1), use numeric Case structure (case 1, case 2 and Case 3) to perform the following tasks: for Case 1, the VI will add the numbers; for Case 2, the VI will subtract the numbers; for Case 3, the VI will multiply the two data. Using one control as the case selector, and one indicator to show the calculation result.

3.5 Flat Sequence Structure

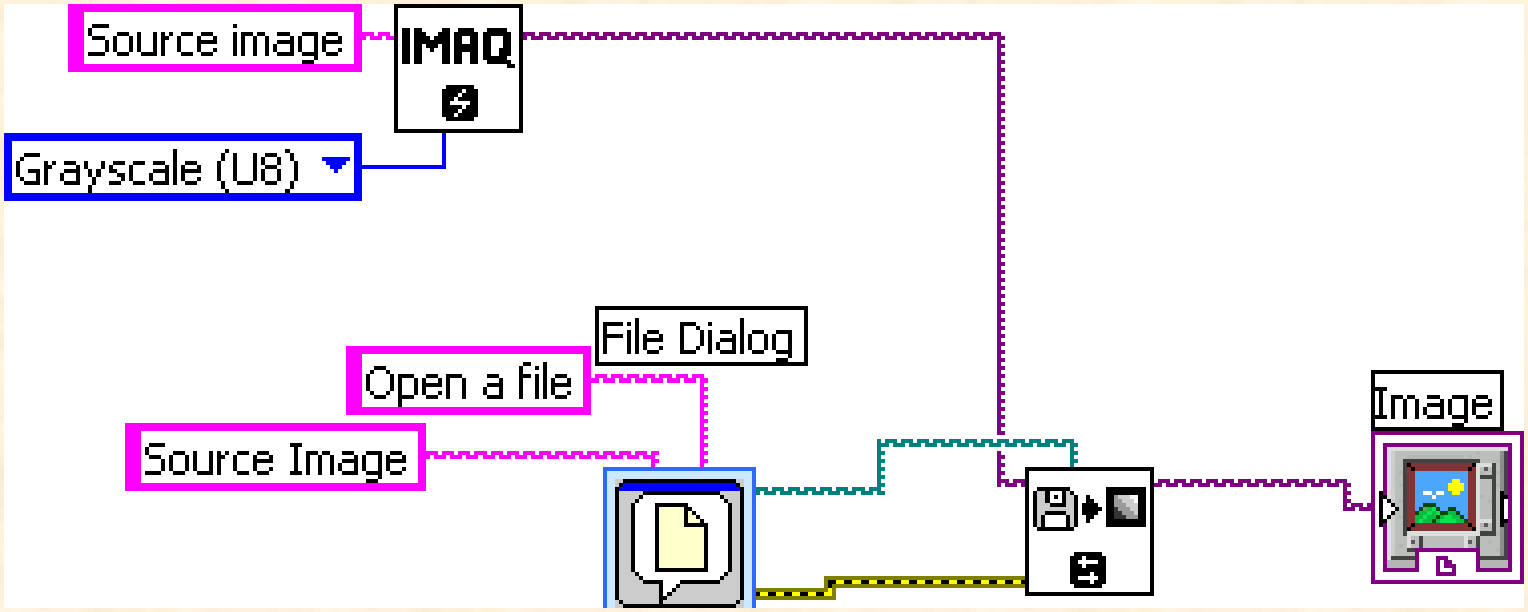
Flat Structure consists of several frames. Frames in flat structure execute in order from the left to the right.

- The VI is executed from left to right, one frame after the other.
- Data is passed between frames using tunnels.
- Right click the board to add more frame.



Read Image Example (Assignment 3)

Write a LabView VI to read an image and show the images by using Image Display function, shown in the following Block Diagram.



Flat Structure Example (Assignment 4)

Write a LabView VI to read two images one after the other by using Flat Structure. Show the two images in two display windows respectively by using Image Display function.

Assignment 5

Use a For loop to generate 100 random numbers. Determine the most current maximum and minimum number as the random numbers are being generated. This is sometimes referred to as a "running" maximum and minimum. Display the running maximum and minimum values as well as the current random number on the front panel.

Be sure to include the "Time Delay Express VI" in the For Loop so that the user is able to watch the values update as the For Loop execute.

You may need to use the For Loop "Shift Register" function and Case structure.

Assignment 6 (Extra Assignment)

Construct a VI that has 3 Round LEDs on the Front Panel. When you run the program, the first LED should turn on and stay on. After one second, the second LED should turn on and stay on. After two more seconds, the third LED should turn on and stay on. All LEDS should be on for three second, and then the program ends.

Assignment 7 (Extra Assignment)

Create a time trial program to compare the average execution time of the "Formula Node" and the native LabVIEW Math Function. This program will require a For Loop, a Flat Sequence Structure, and a Case Structure. The For Loop is required to run the "time trial" N times. The Sequence Structure is required to sample the "Tick Count" before and after the code executes. The Case Structure is required to determine whether the user would like to execute the Formula Node or the native LabVIEW Math Functions. To test the timing, using the following formulas:

$$a = X^2 / 4;$$

$$b = (2 \times X) + 1;$$

$$Y = \sin(a + b);$$

Where X is the input and Y is the output. Run the time trial for each of the cases. Which method has the fastest execution time?