

# LabVIEW Lecture 1

Ertugrul Karademir

# What is LabVIEW?

- Graphical programming environment
- Measurement, testing, control applications
- Hardware control
- Flowchart representation
- Object Oriented Approach

# What is LabVIEW?

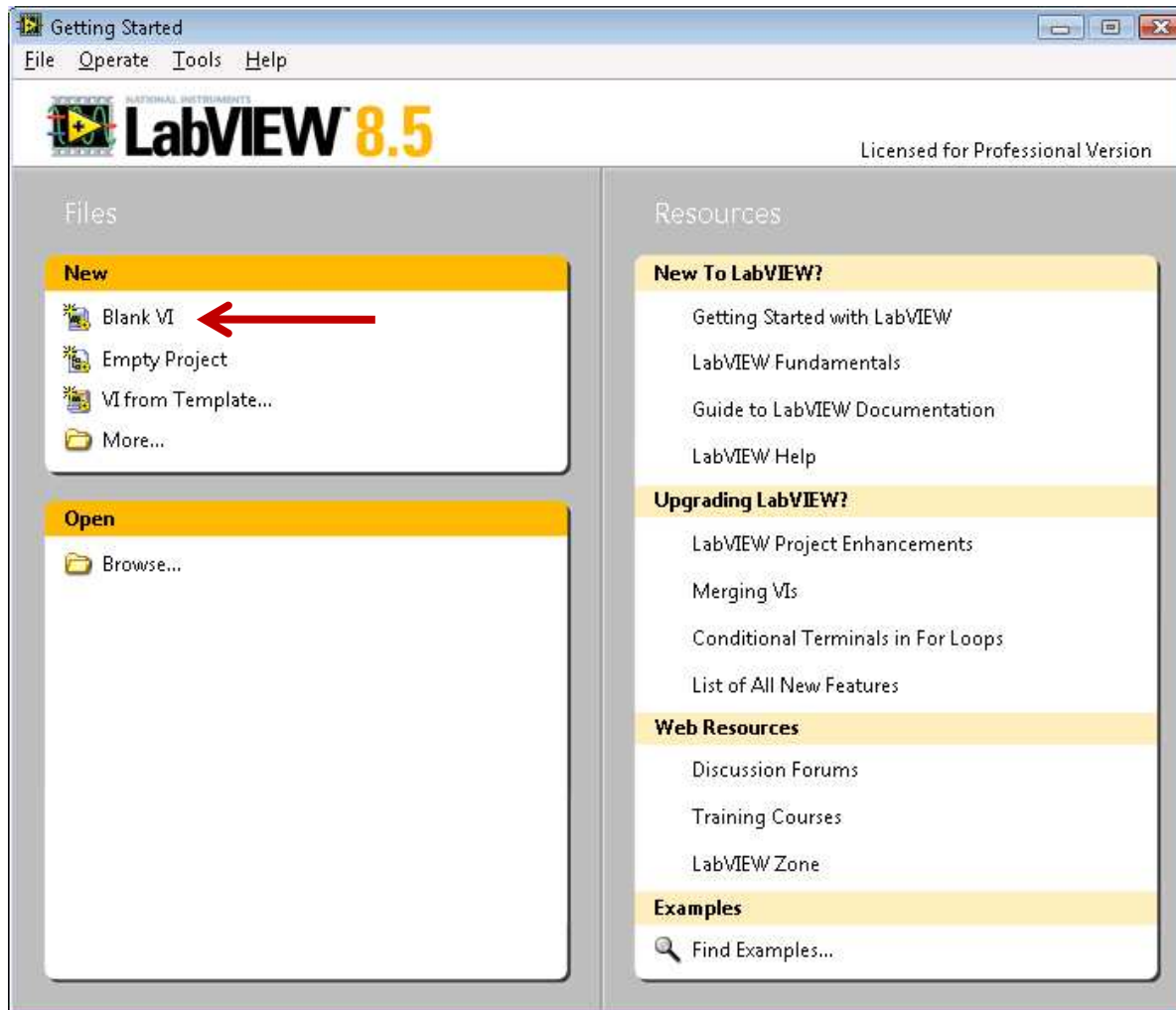


Ertugrul  
Bilkent  
F44G44444  
LabVIEW Professional Development System



[ni.com/labview](http://ni.com/labview)

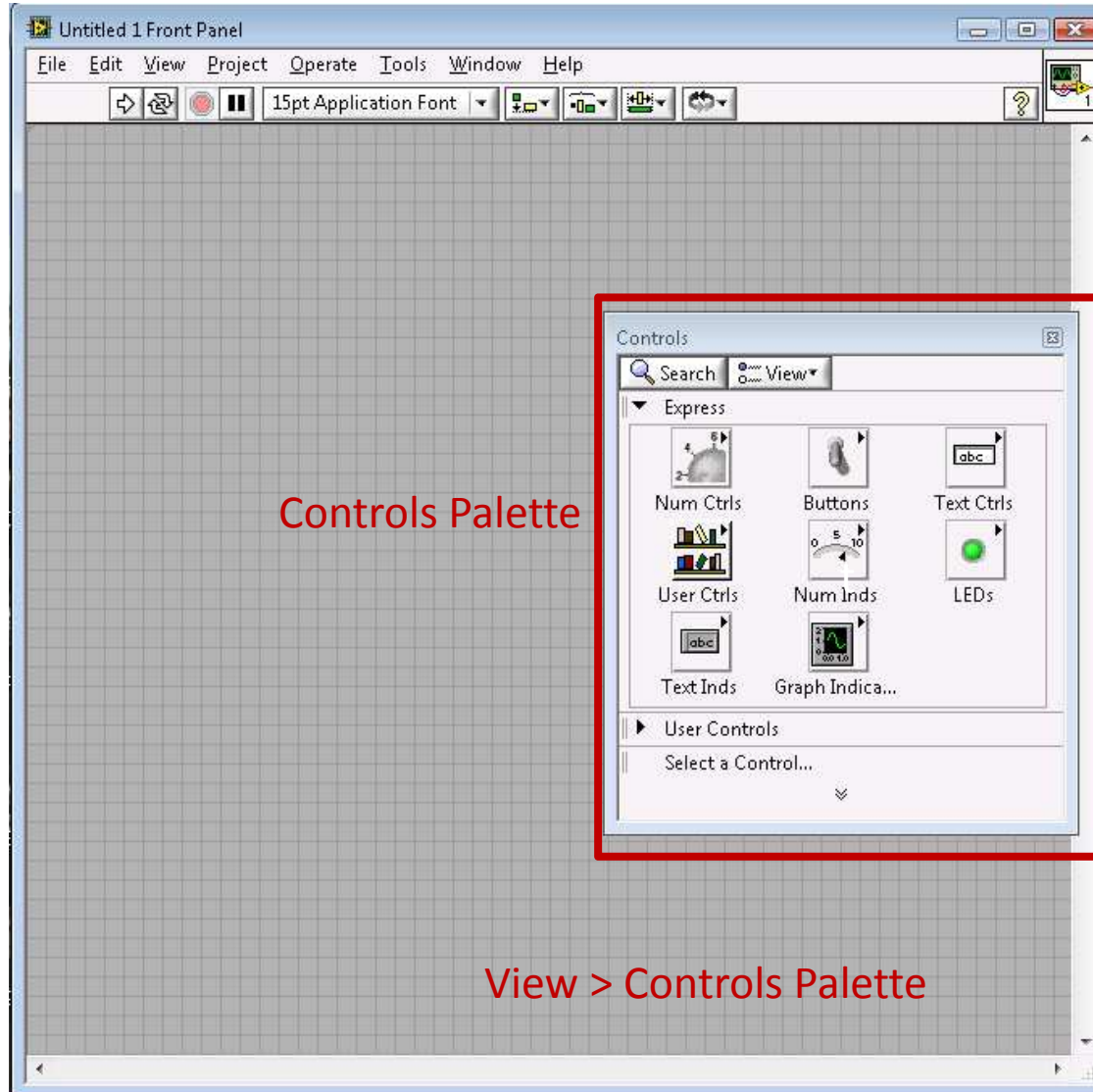
# What is LabVIEW?



# Virtual Instrument

- LabVIEW programming paradigm
  1. Program control with a front panel
  2. Algorithm with block diagram
- Easy to debug
- Hard to program

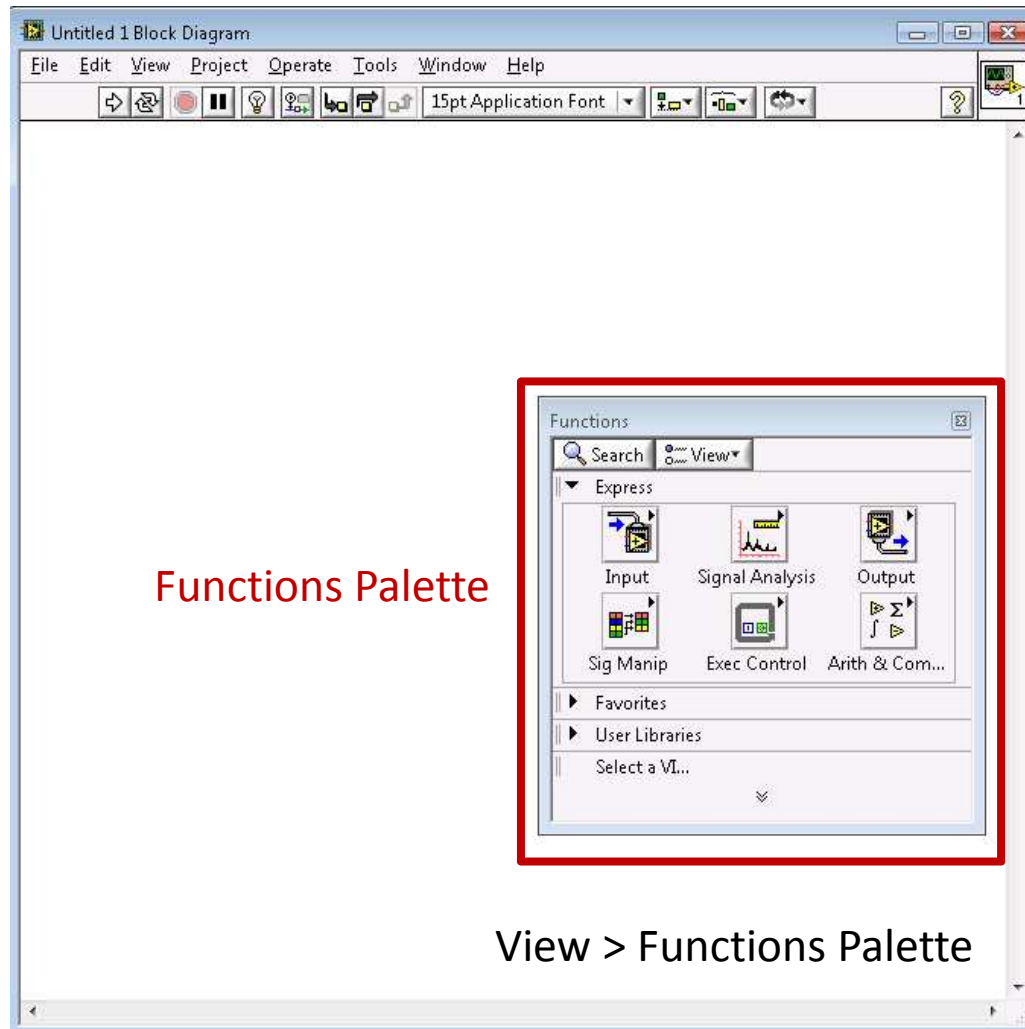
# Front Panel



Controls Palette

View > Controls Palette

# Block Diagram



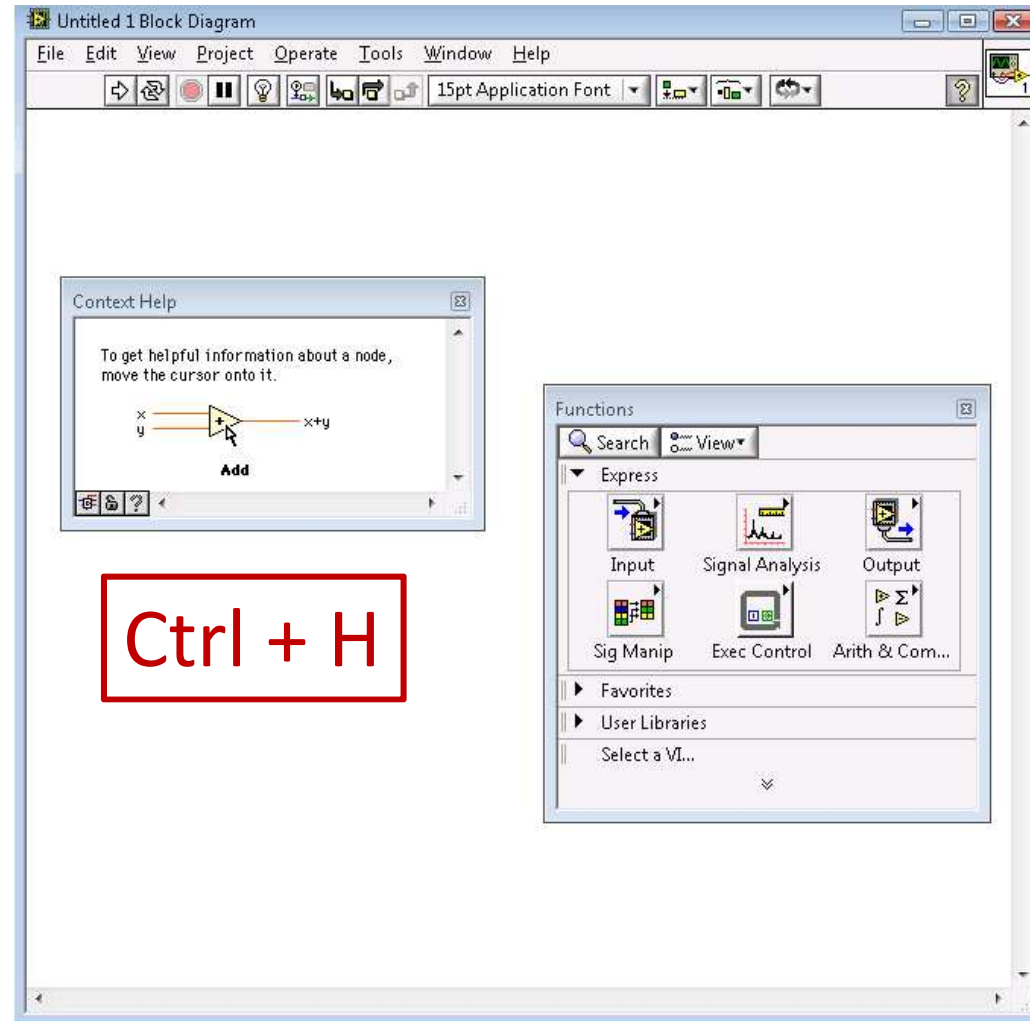
# Front Panel – Block Diagram

- To toggle between two panels

Ctrl + E



# Context Help

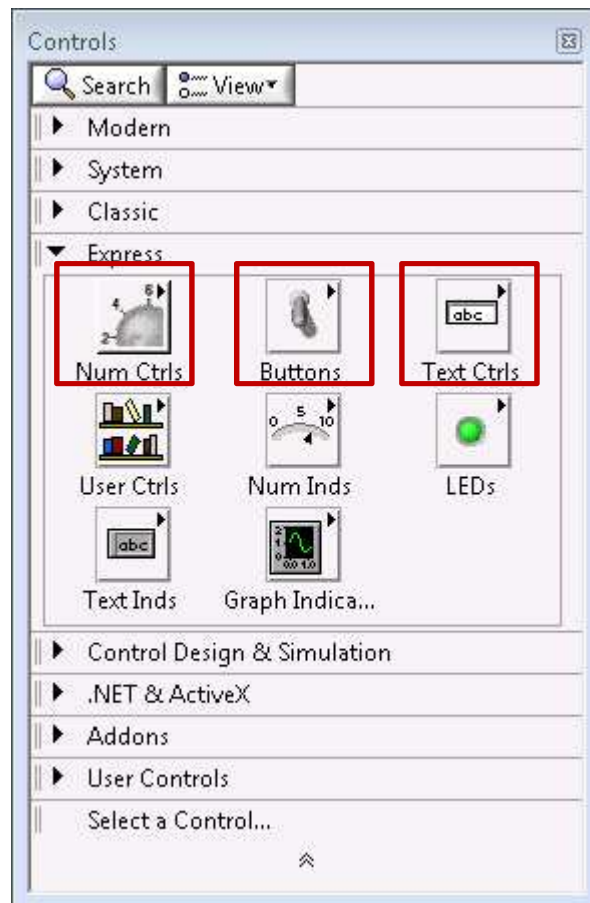




# Three fashions of variables

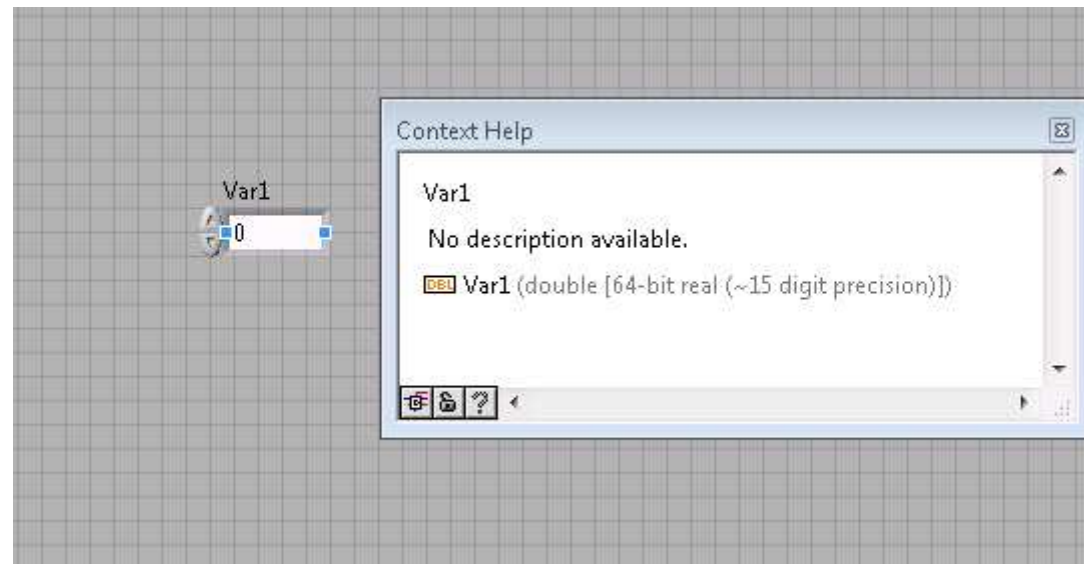
- Mixture of GUI and console
  1. Controls
    - User input (In Java: TextBox GUI Component)
  2. Indicators
    - Program output (In Java: Label GUI Component)
  3. Constants
    - Predefined constants (In Java: Class variables with **const** definer)

# Controls



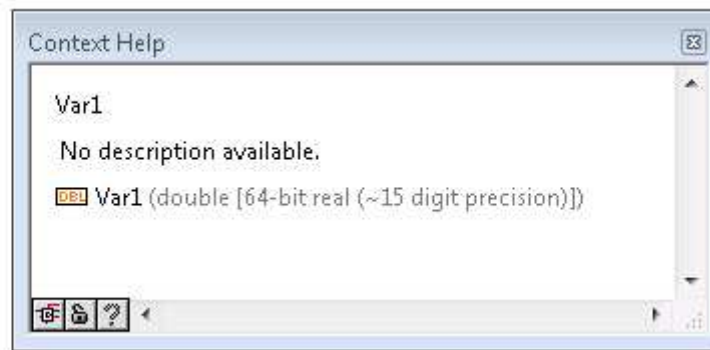
- Selected from Controls Palette
- Usually Express Palette is enough

# Controls

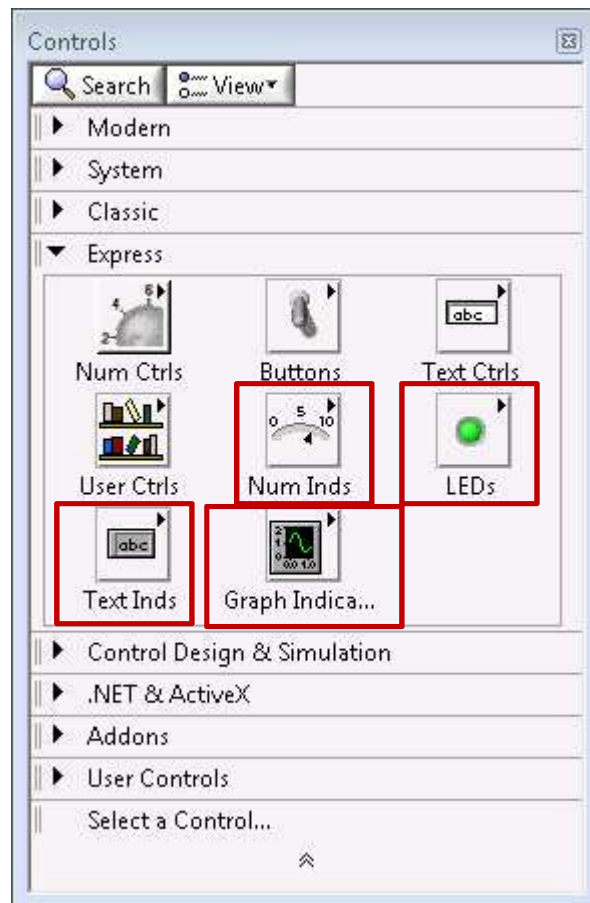


# Controls

- Each control has a proxy in the Block diagram
- Note that proxy has only OUTPUT node

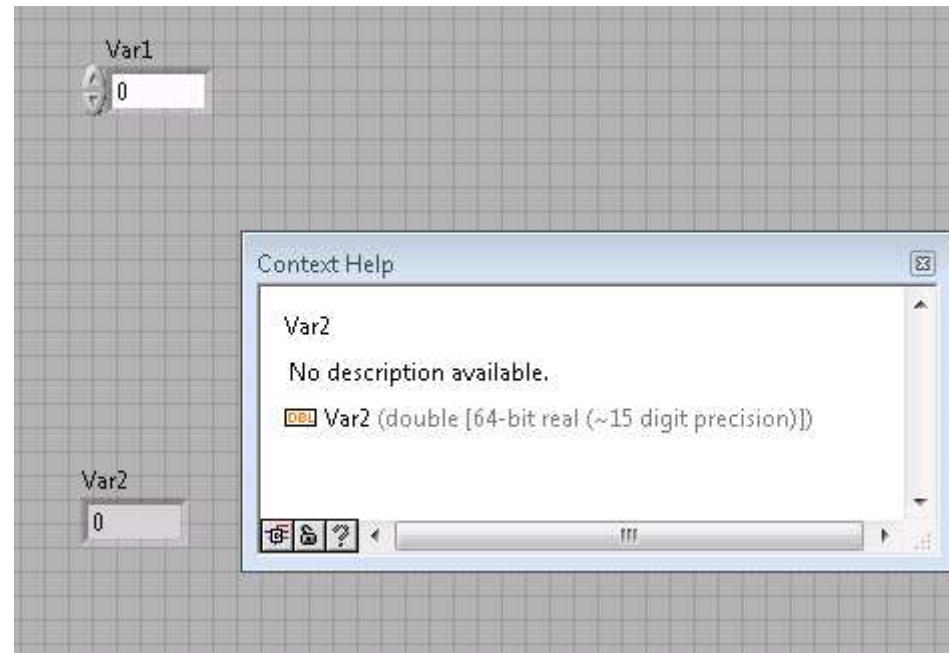


# Indicators



- Also Selected from Controls Palette
- Usually Express Palette is enough

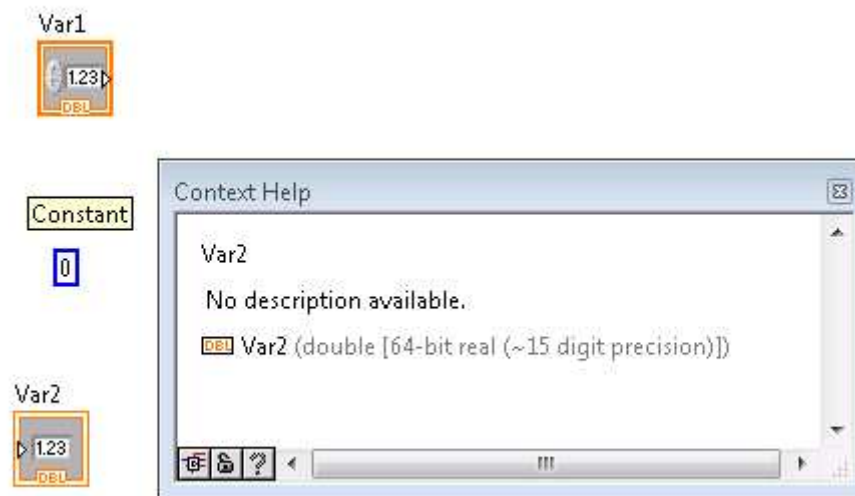
# Indicators



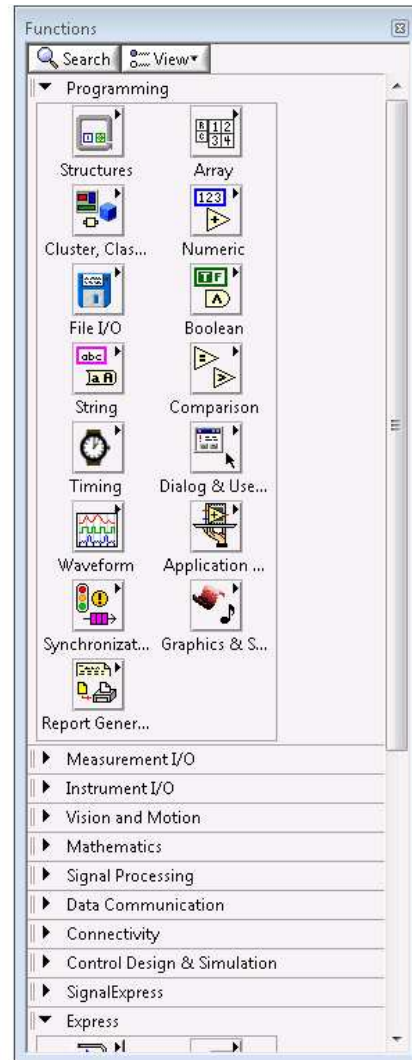


# Indicators

- Each indicator also has a proxy in the Block diagram
- Note that proxy has only INPUT node



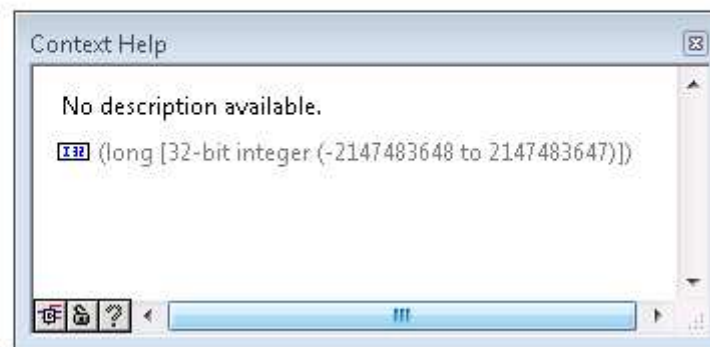
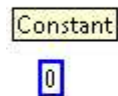
# Constants



- You can use functions palette to insert constants

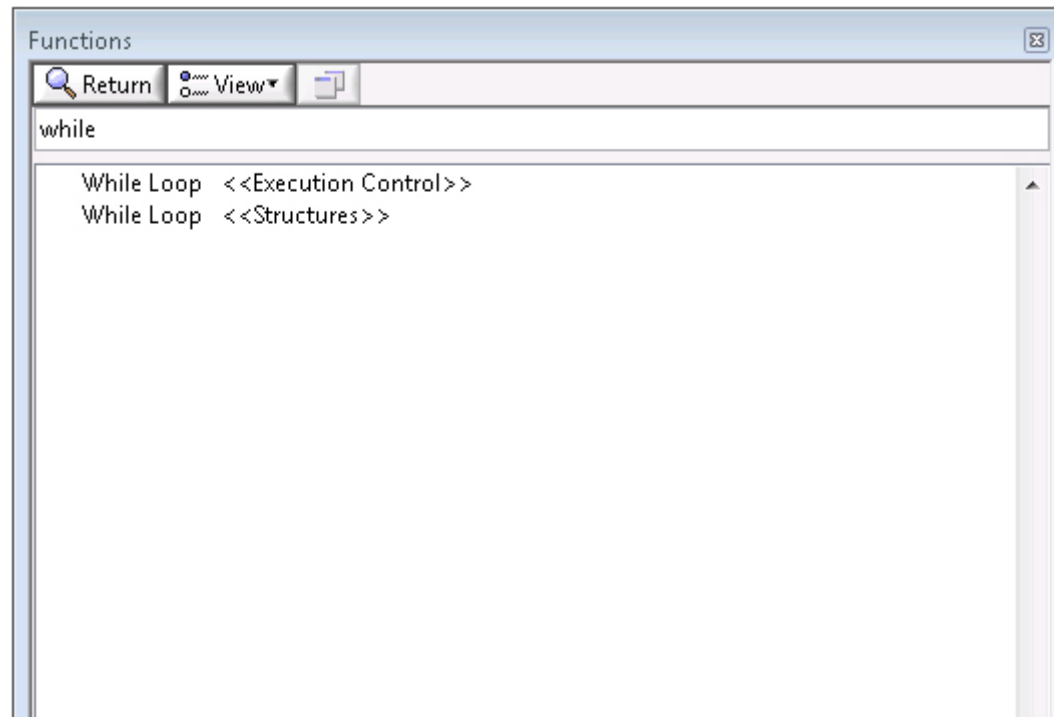
# Constants

- Constants only live in Block diagram



# Search

- You can always search for the item
- Click on the search button on the palettes

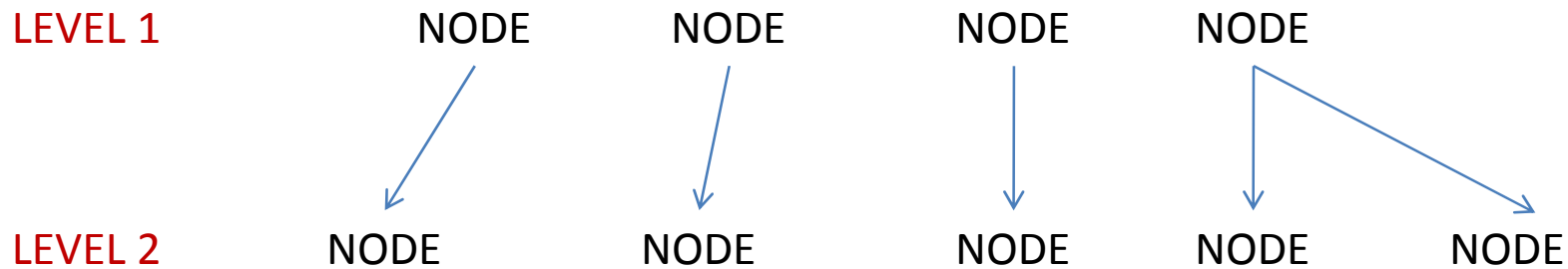


# Types

- All standard types are present
  - Integer (signed, unsigned, long, word, byte, quad)
  - Floating point (single precision, double precision, extended precision)
  - Boolean
  - String
- All of above can be arranged in arrays, matrices, clusters

# Flow Chart Paradigm

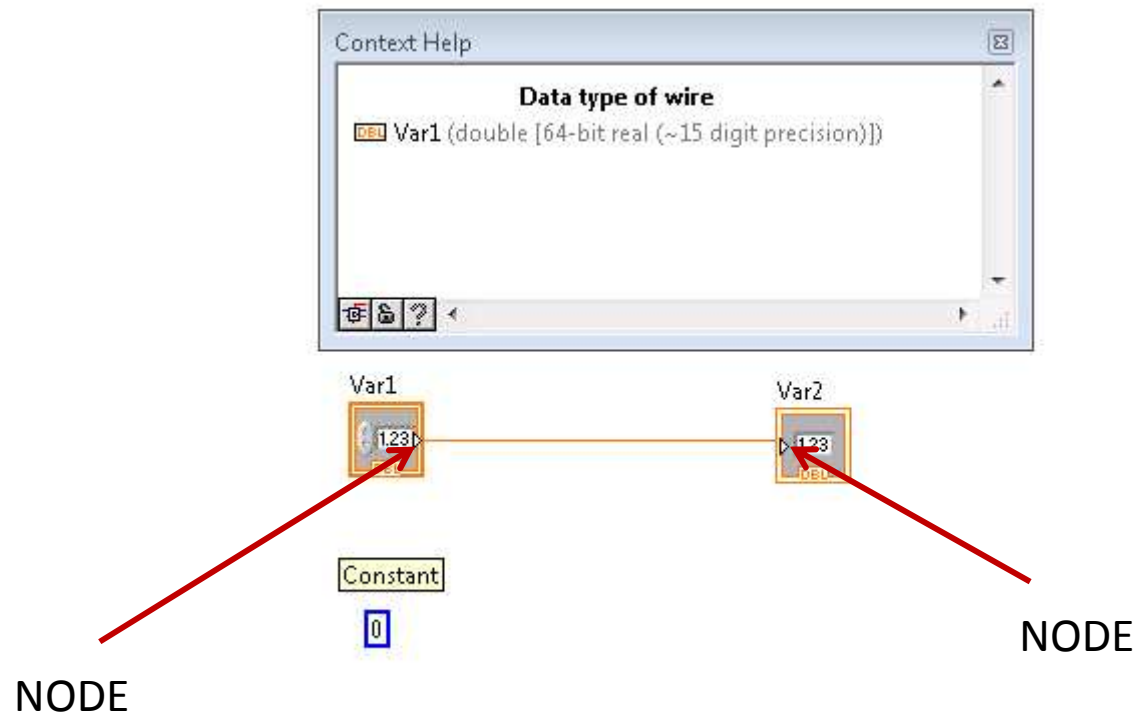
- Flow of execution is done by following nodes in a flow diagram



And so on...

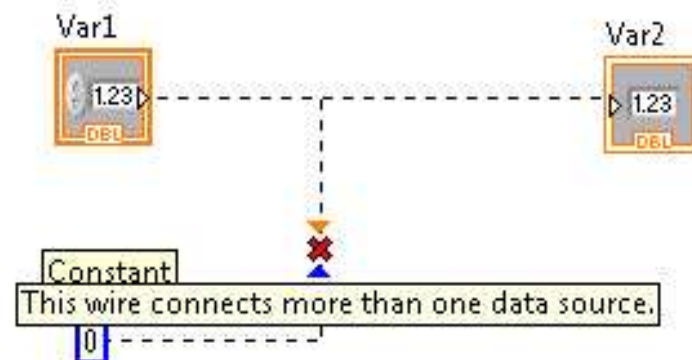
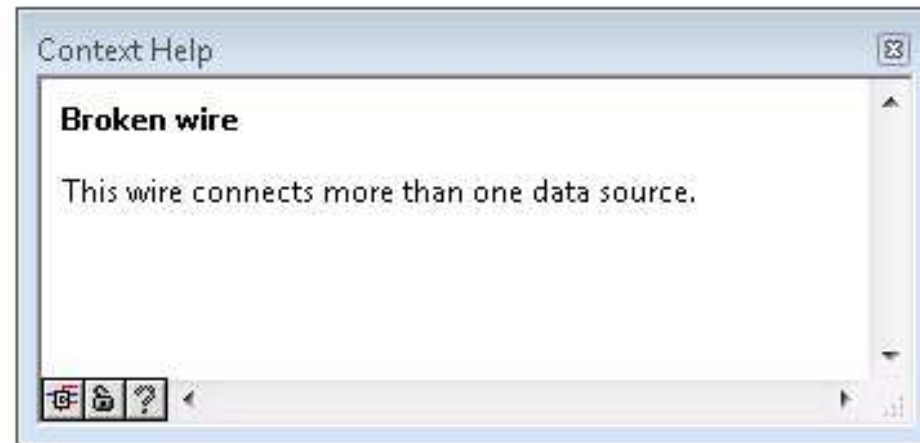
# Wiring

- Defines the direction of flow



# Wiring

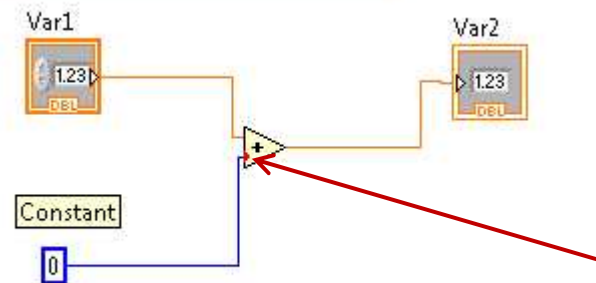
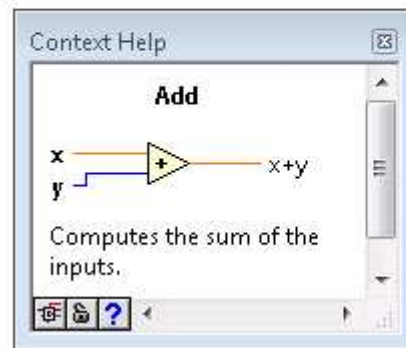
- One to many connection is acceptable
- Many to one connection is illegal



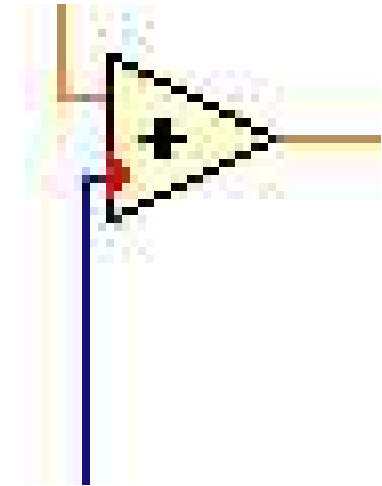


# Wiring

- Application of algorithm is done by wiring
- Color of the wire indicates type
  - Blue: Integer , Orange: Floating Point
  - Purple: String, Green: Boolean

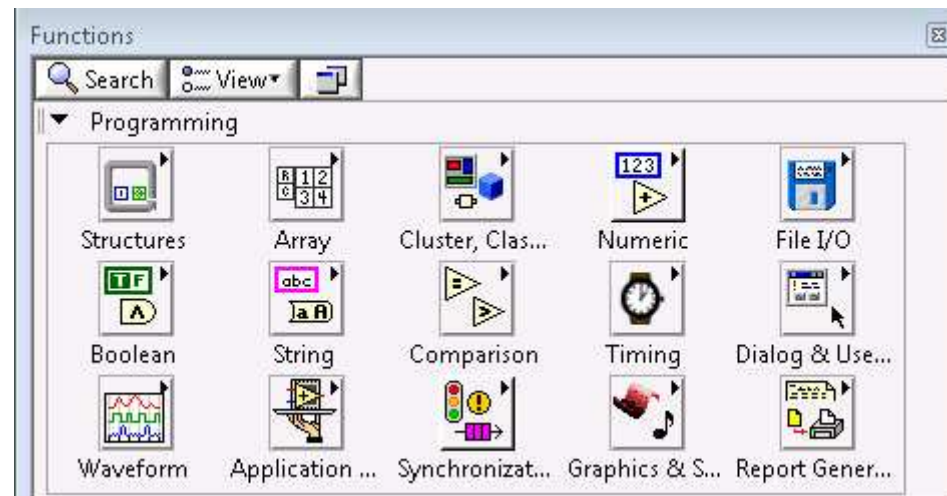


Red dot indicates that “wrong type has wired but it’s OK, he has made the type-casting”.



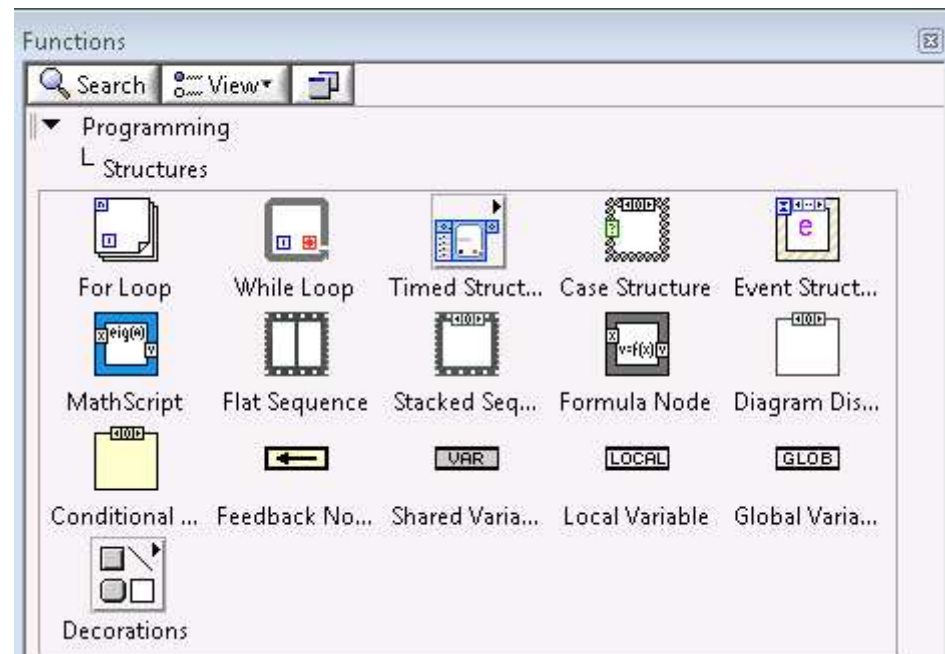
# Algorithm Construction

All algorithm structures lies in Programming Sub-Palette

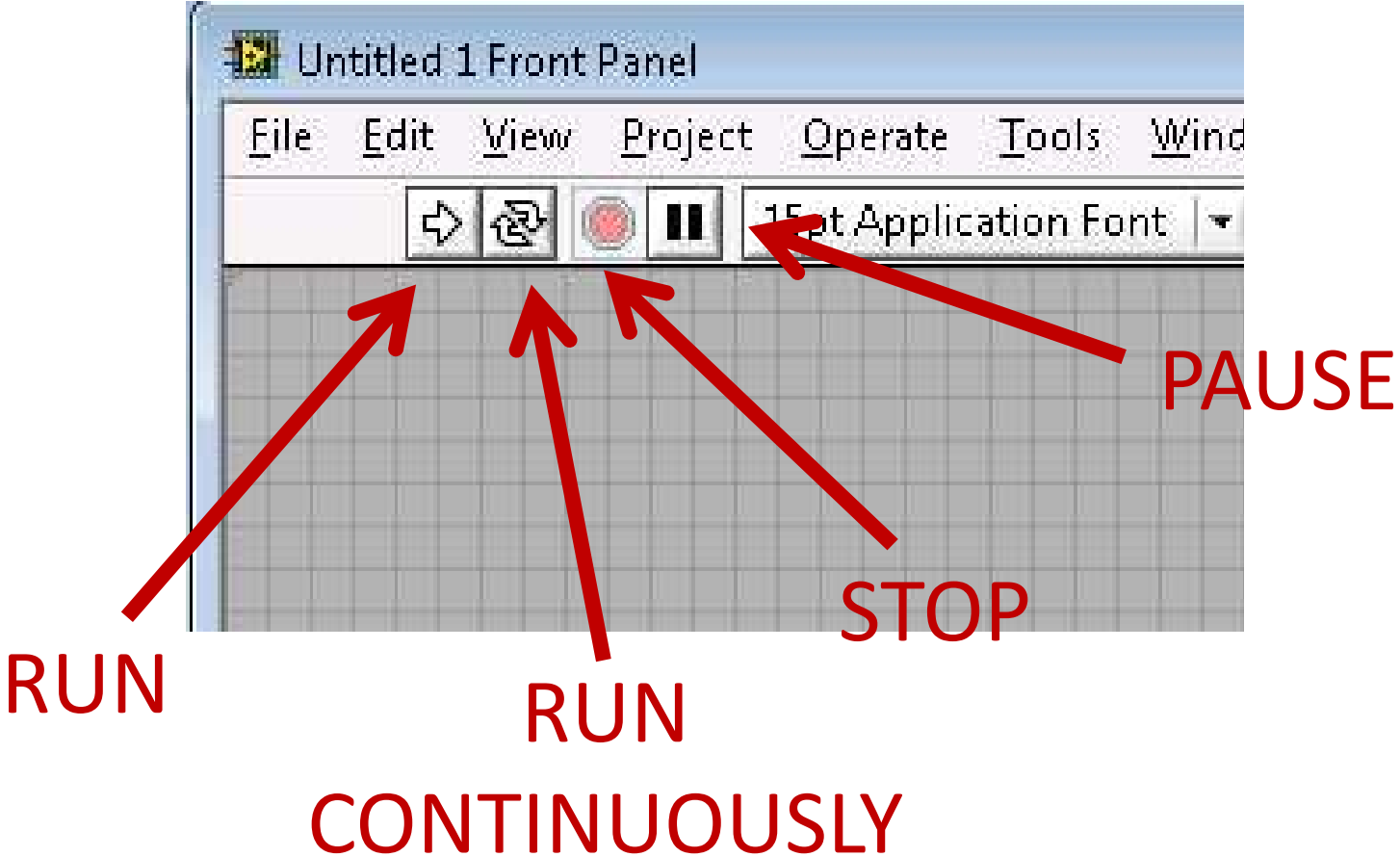


# Algorithm Construction

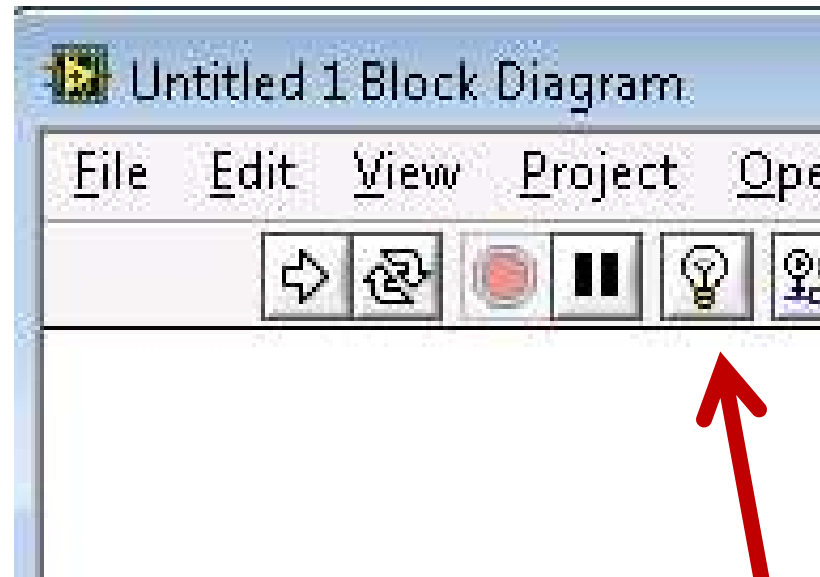
Program flow structures are under Structures  
Sub-sub-palette



# Program Control

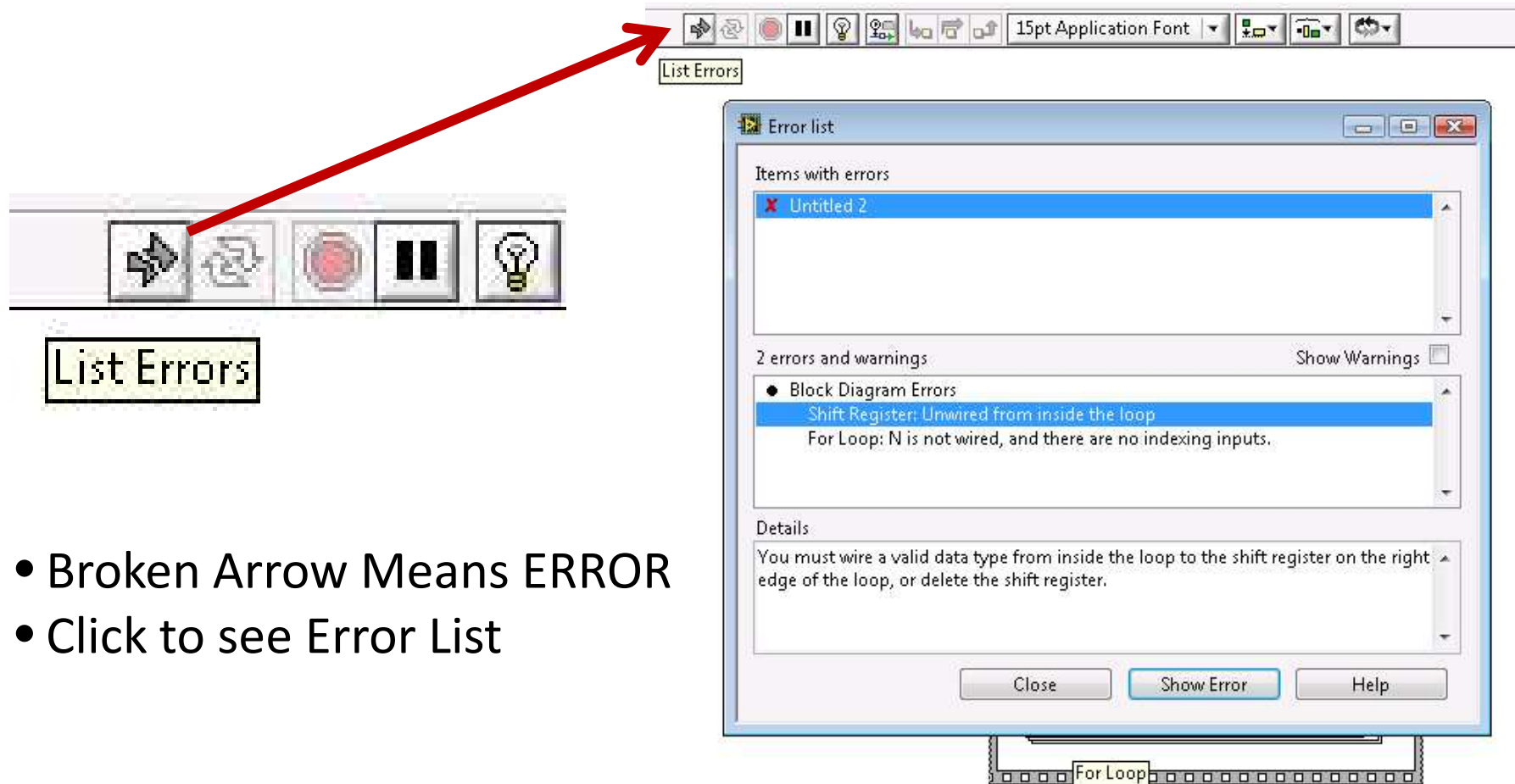


# Debugging



**HIGHLIGHT  
EXECUTION**

# Debugging

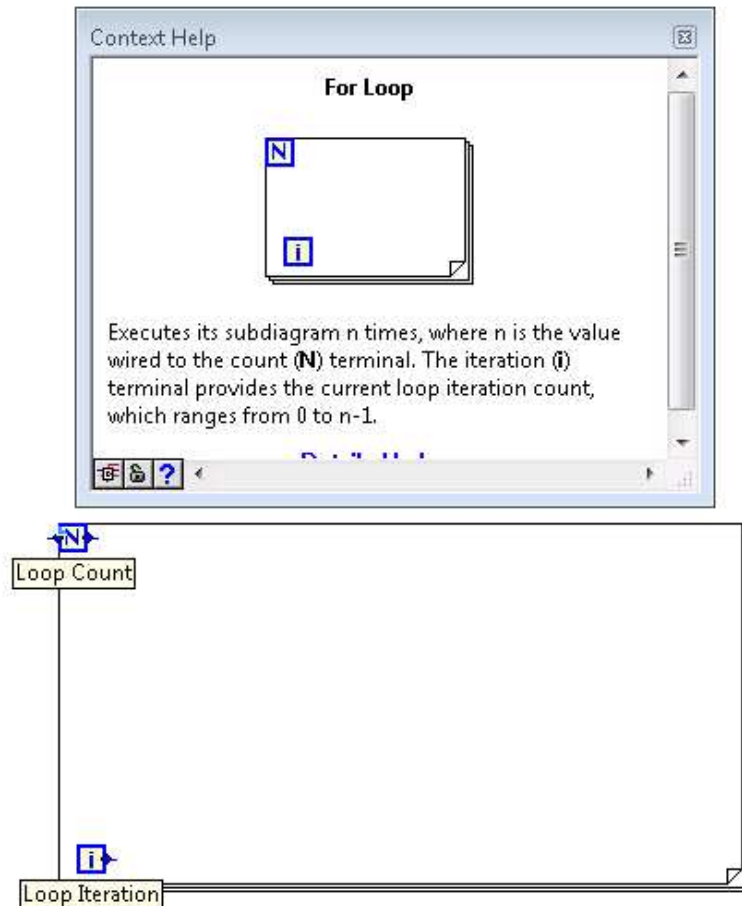


- Broken Arrow Means ERROR
- Click to see Error List

# Program Flow Control

- For loops
- While loops
- Sequences
  - Flat sequence
  - Stacked sequence
- And many more
- Compansates for: Event handling, Top-to-bottom execution, OOP, etc.

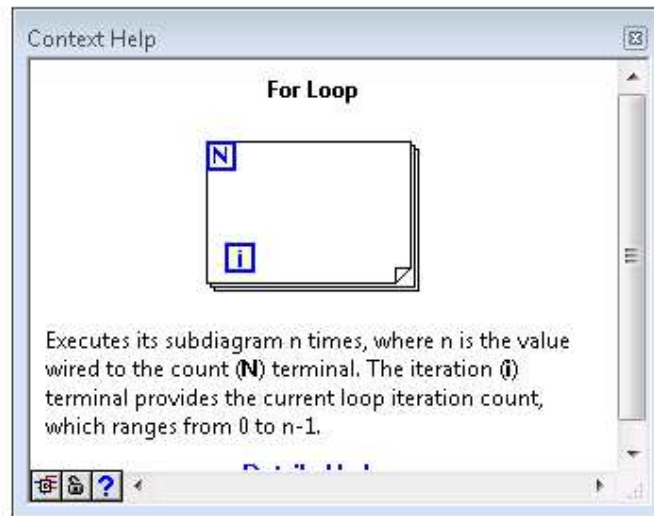
# For loop



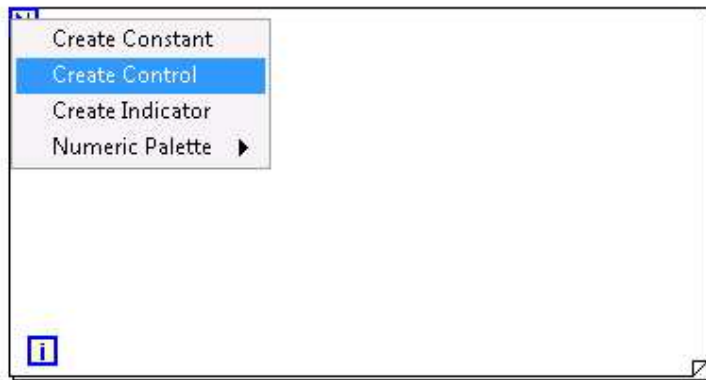
- Loop for limited iterations
- Must know the iteration amount before-hand
- Loop count can-not be change once set



# For loop



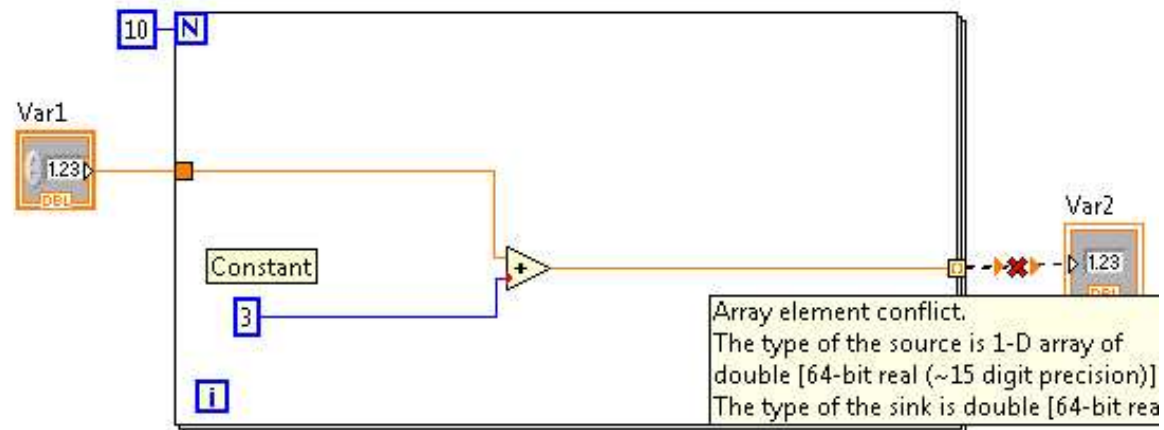
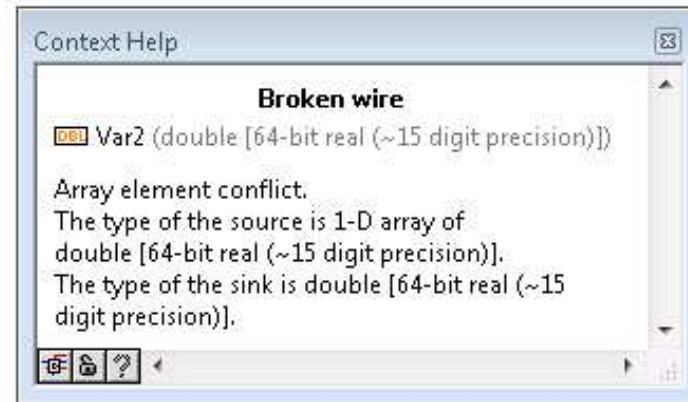
Another way to create variables: Right clicking onto the node



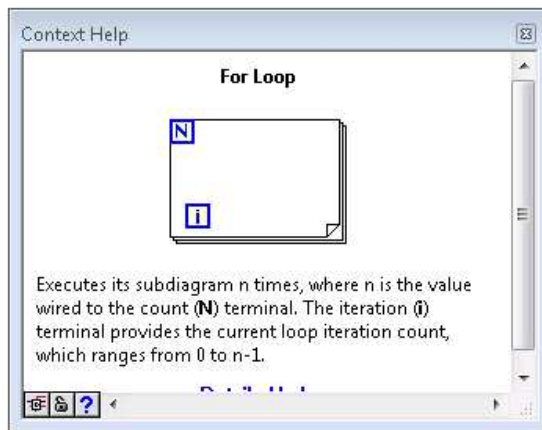
# For loop example

- Get some number from the user
- Add 3 to it 10 times
- Display the result

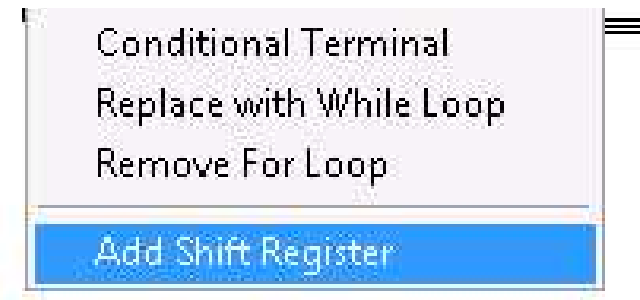
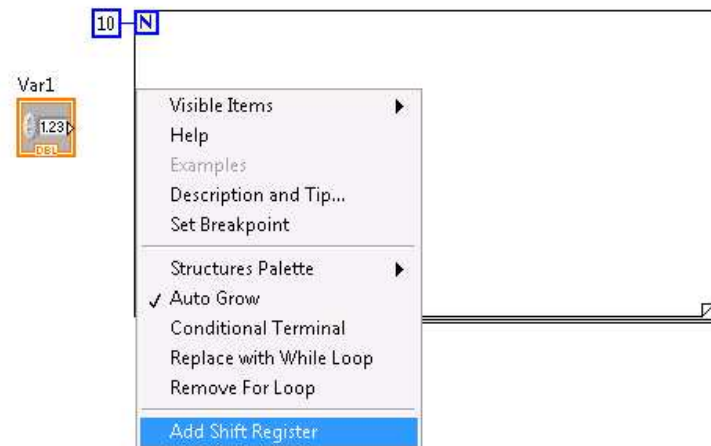
# For loop example



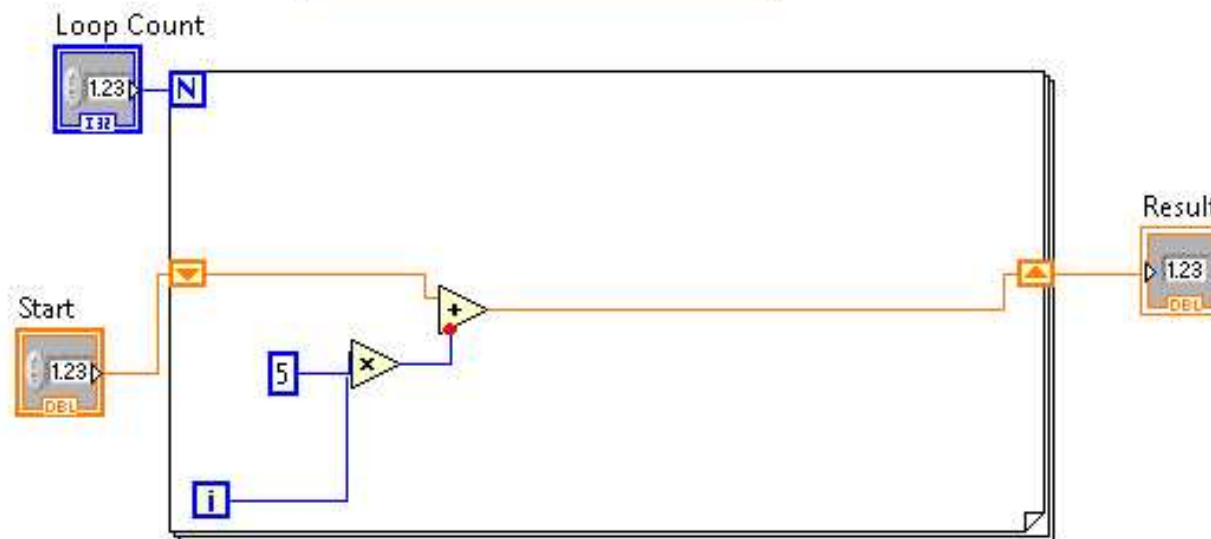
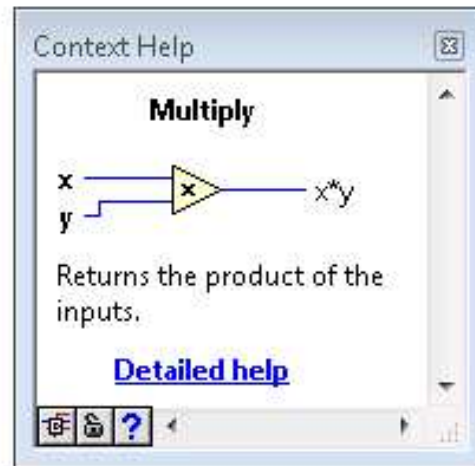
# For loop example



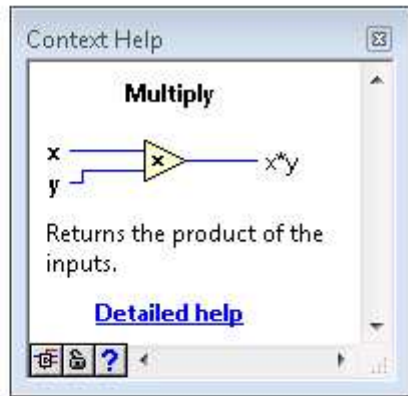
Shift registers convey result of one iteration to the next iteration



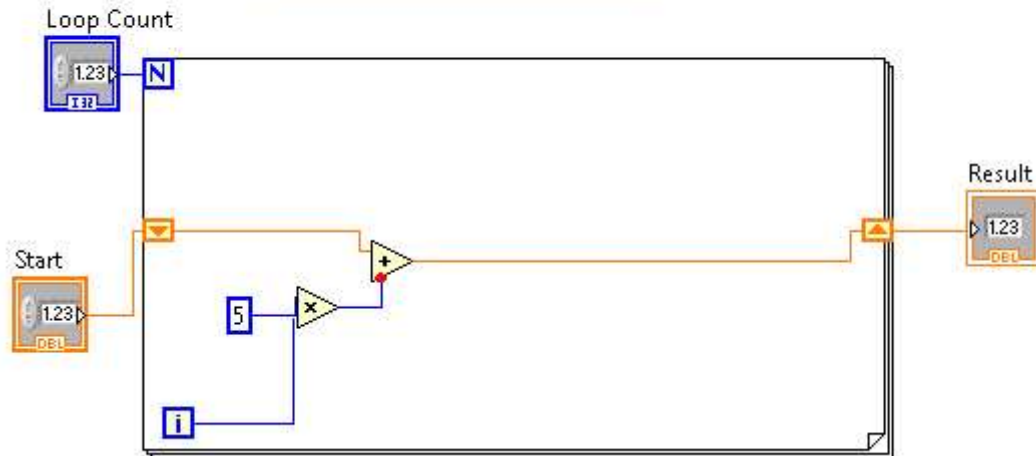
# For loop example



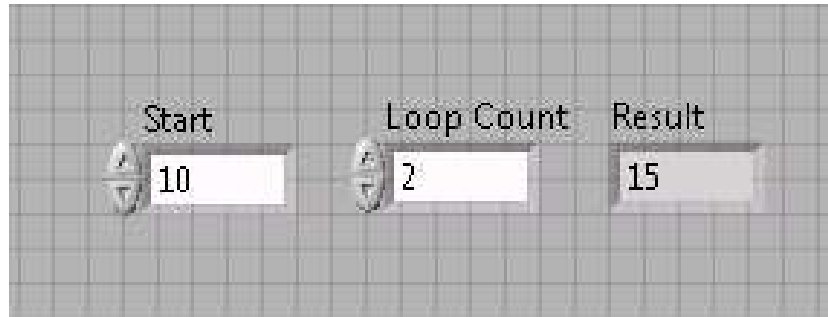
# For loop example



Do not limit your imagination

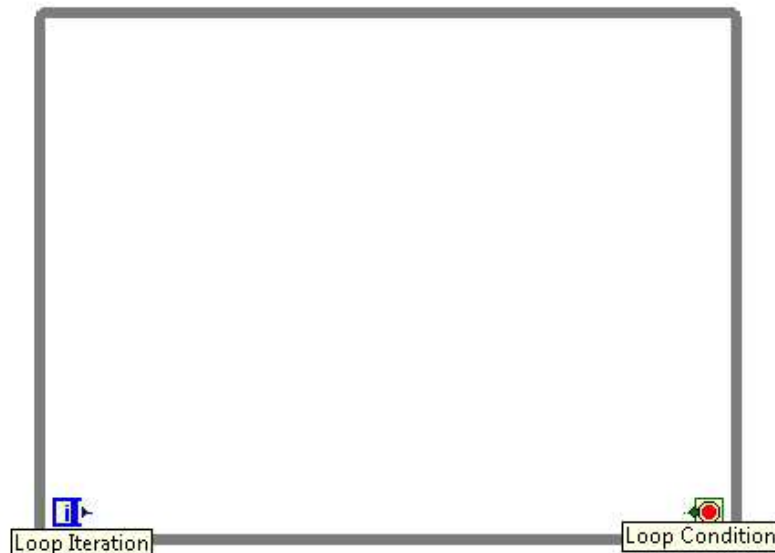
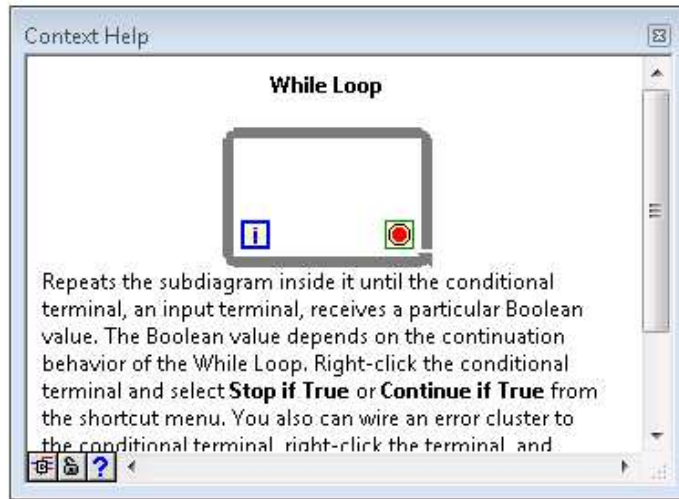


# For loop example

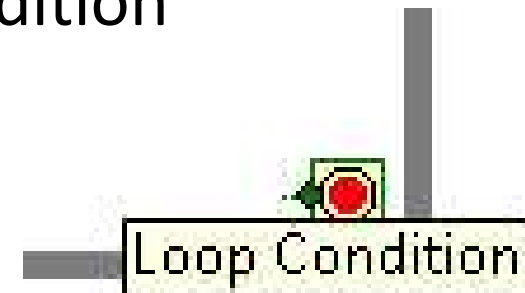


User only interacts with three objects

# While loop

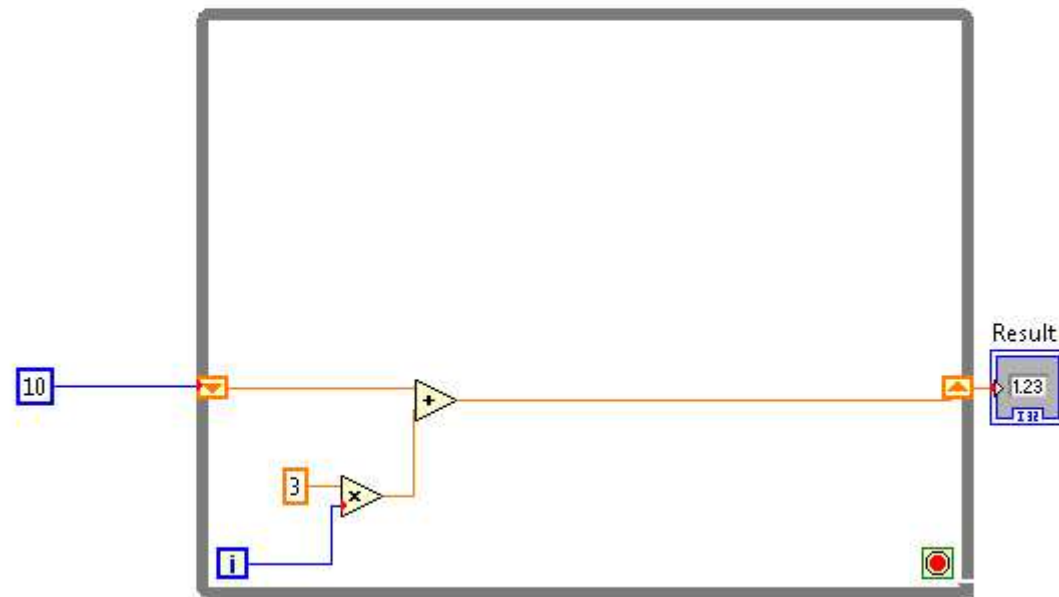


- Loops until the loop condition is satisfied
- Or while the loop condition is not satisfied
- Select by clicking on the loop condition

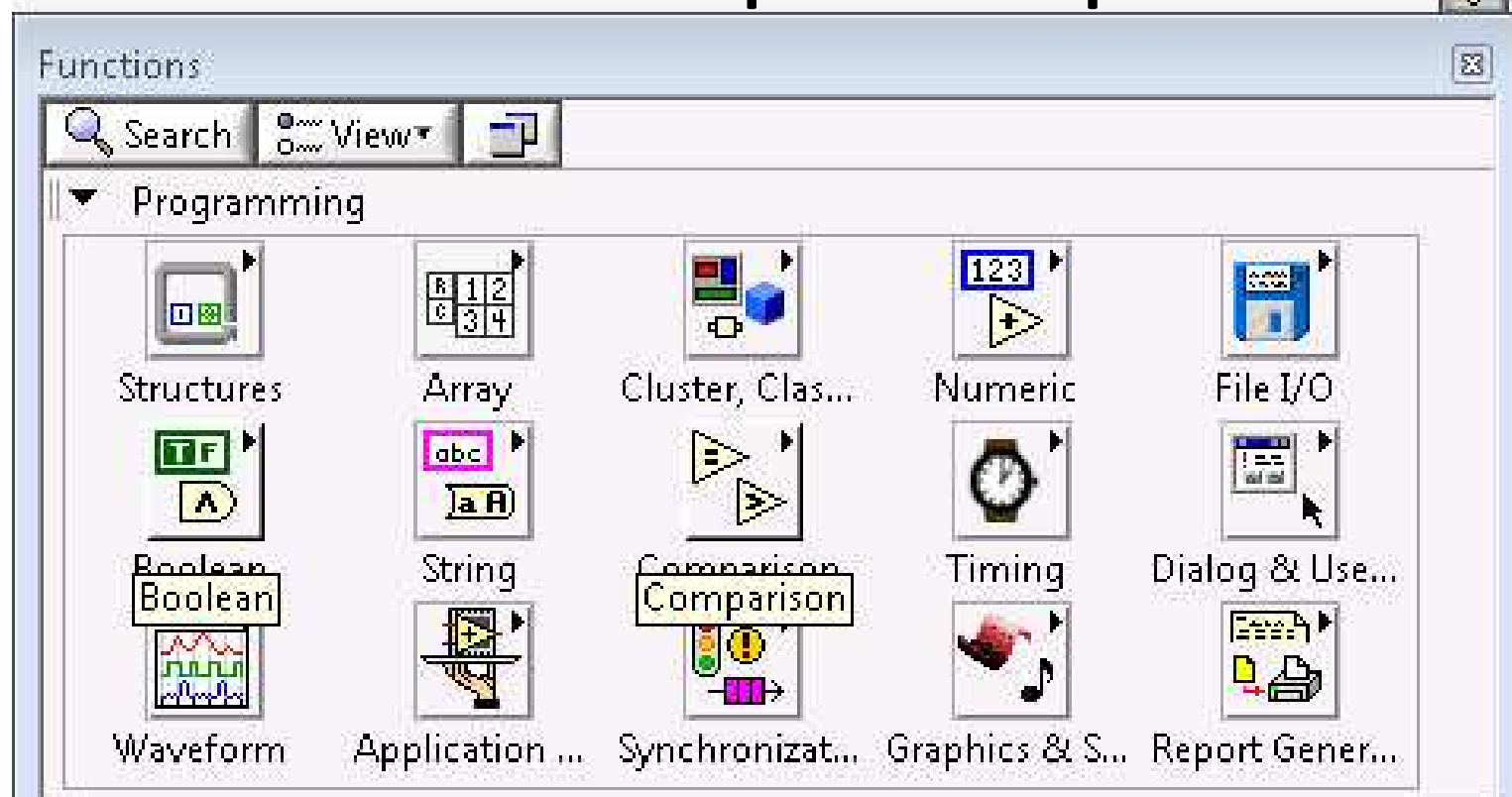




# While loop example



# While loop example



Employ boolean and comparison palette controls to manage loop condition

# While loop example

Context Help

**Equal?**

$x = y?$

Returns TRUE if  $x$  is equal to  $y$ . Otherwise, this function returns FALSE. You can change the comparison mode of this function.

[Detailed help](#)

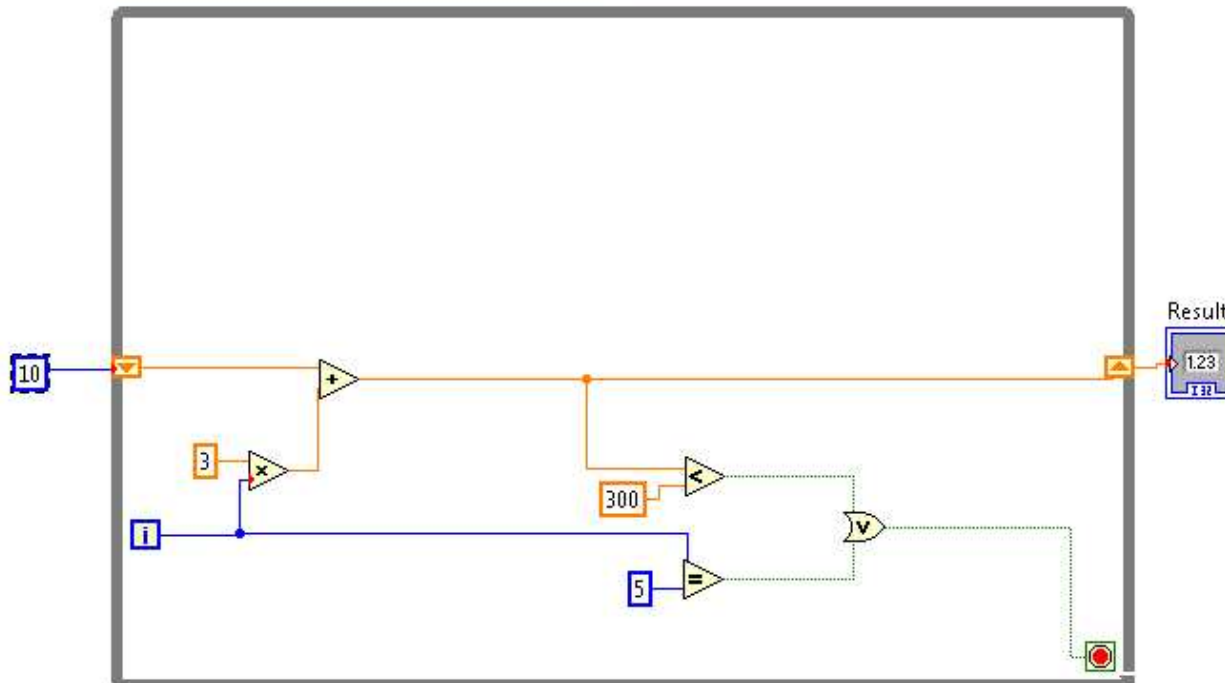
Context Help

**Less?**

$x < y?$

Returns TRUE if  $x$  is less than  $y$ . Otherwise, this function returns FALSE. You can change the comparison mode of this function.

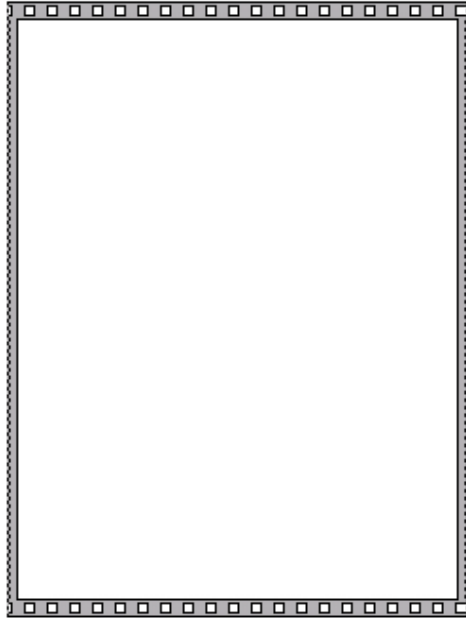
[Detailed help](#)



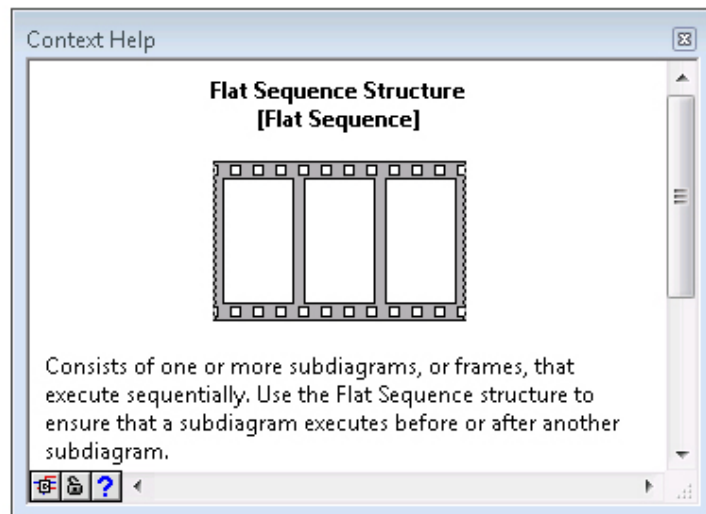
Result

10

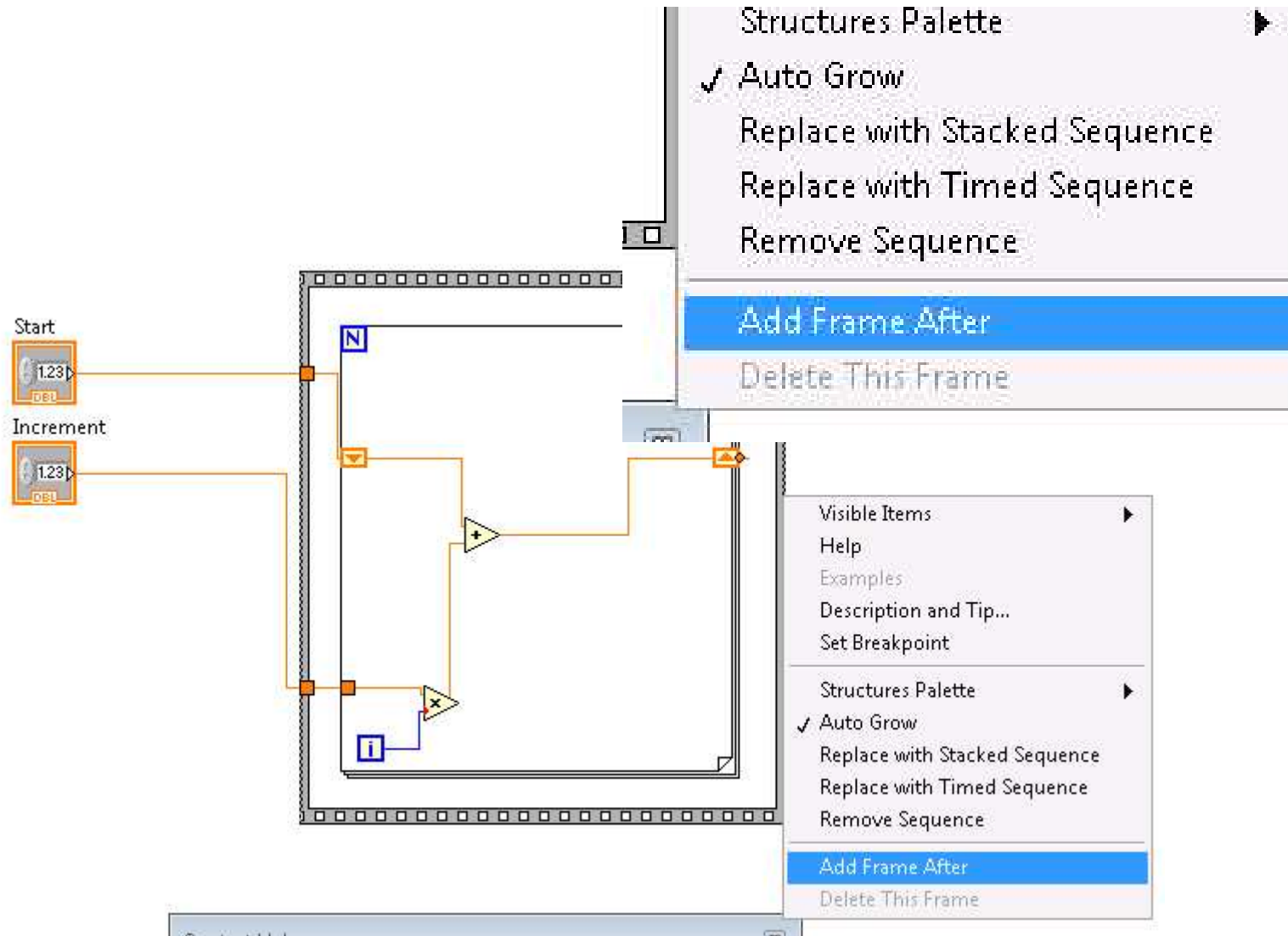
# Flat sequence



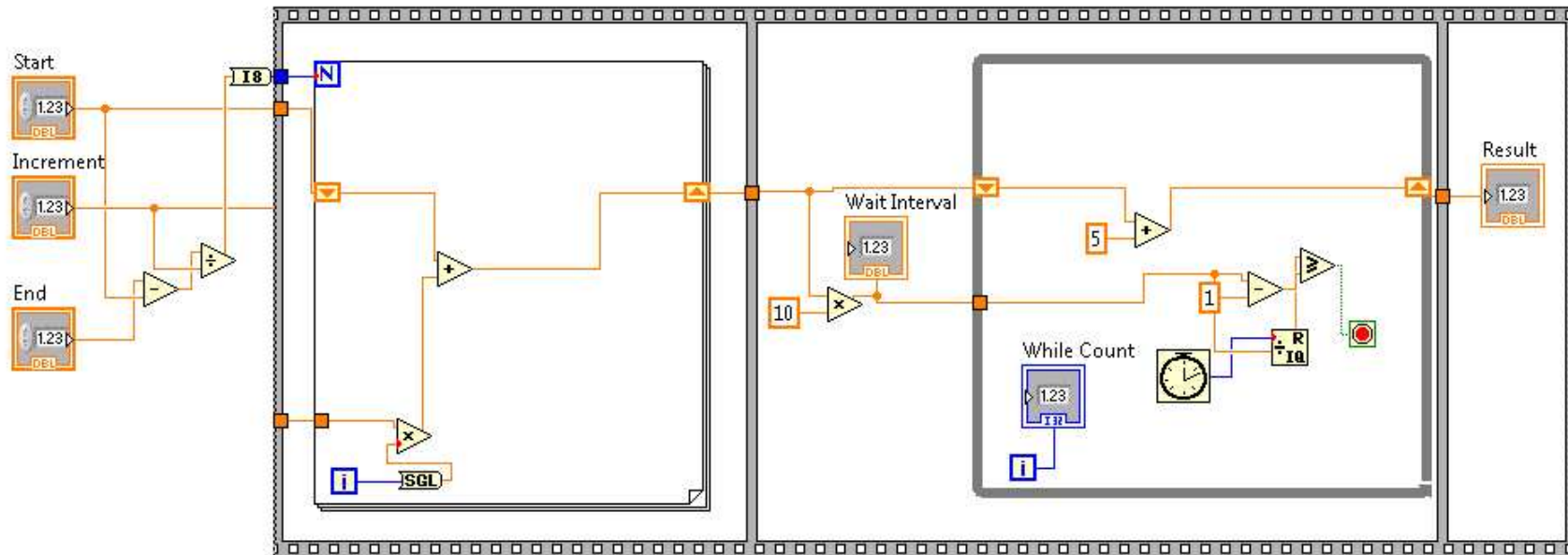
Executes contents of each frame one by one



# Flat sequence



# Flat sequence



Start	2	While Count	134084594
Increment	0,5	Wait Interval	40025
End	200	Result	2,37494E+7