Dataflow programming languages: LabVIEW

CC BY NC SA

Andrea Sterbini – sterbini@di.uniroma1.it

Data-flow: interconnected functional units

Functional units connected by wires

- wires represent data exchanges (i.e. variables)
- they could be typed (a different color/shape for each wire)
- many data can be aggregated in a single BUS (i.e. record)
- each functional unit has a default GUI for testing its I/O

Granularity

- functional units can be defined and reused
- circuits/networks can be packaged as new blocks

LabVIEW

Created by National Instruments to interact with digital data-acquisition and control systems

Modelled over the circuit design and testing metaphor

Each functional unit in the graphic language runs
as soon all its input data are available

Multiple cores and threads are used to schedule the parallel execution of multiple active units

The programs are compiled into an intermediate "G" language
(but can also be compiled to native code)

You normally (need to) add explanation boxes to document your ideas

Free LabView Community edition available for personal usage

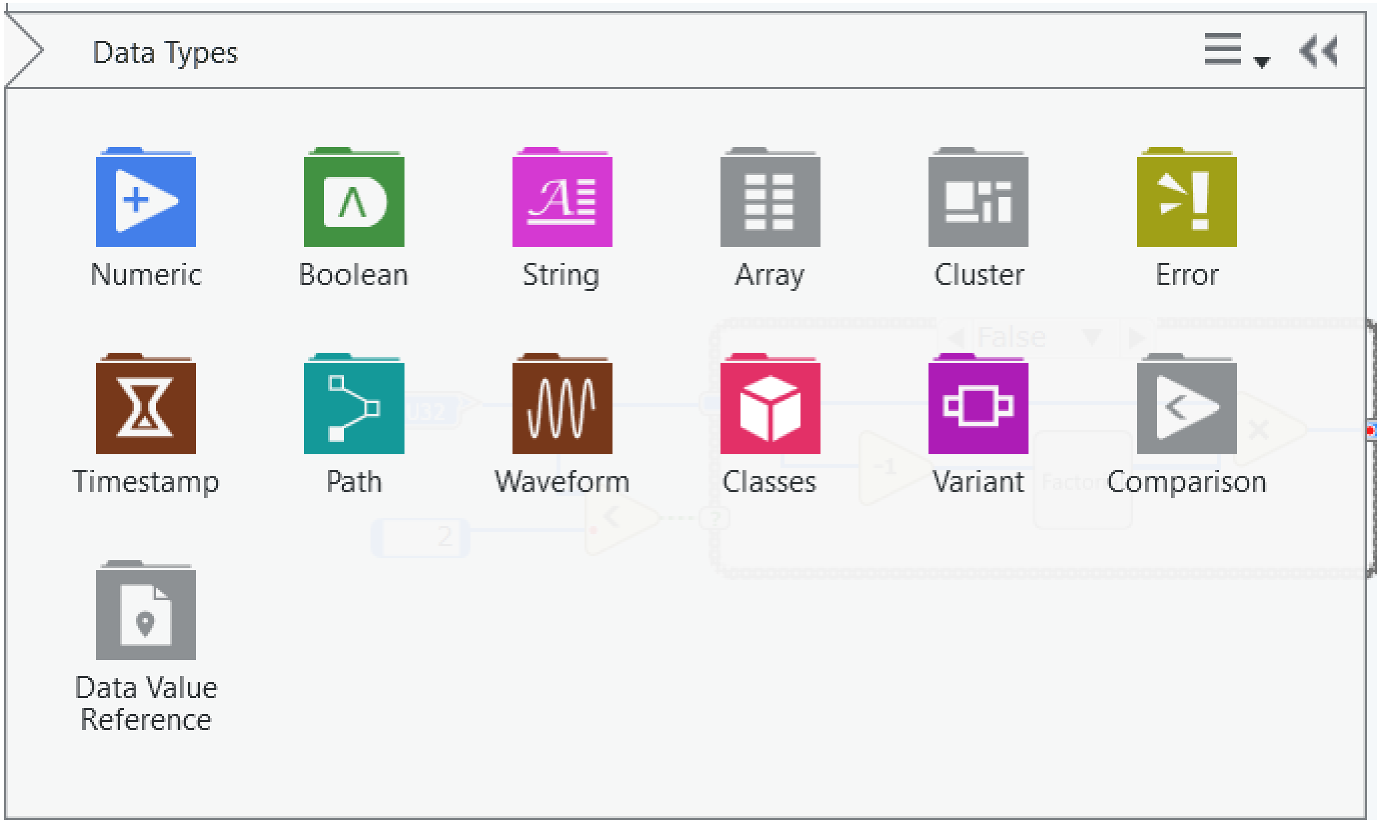
Data types

Many numeric types (to interface with hardware)

Arrays and records (Clusters)

Classes

…

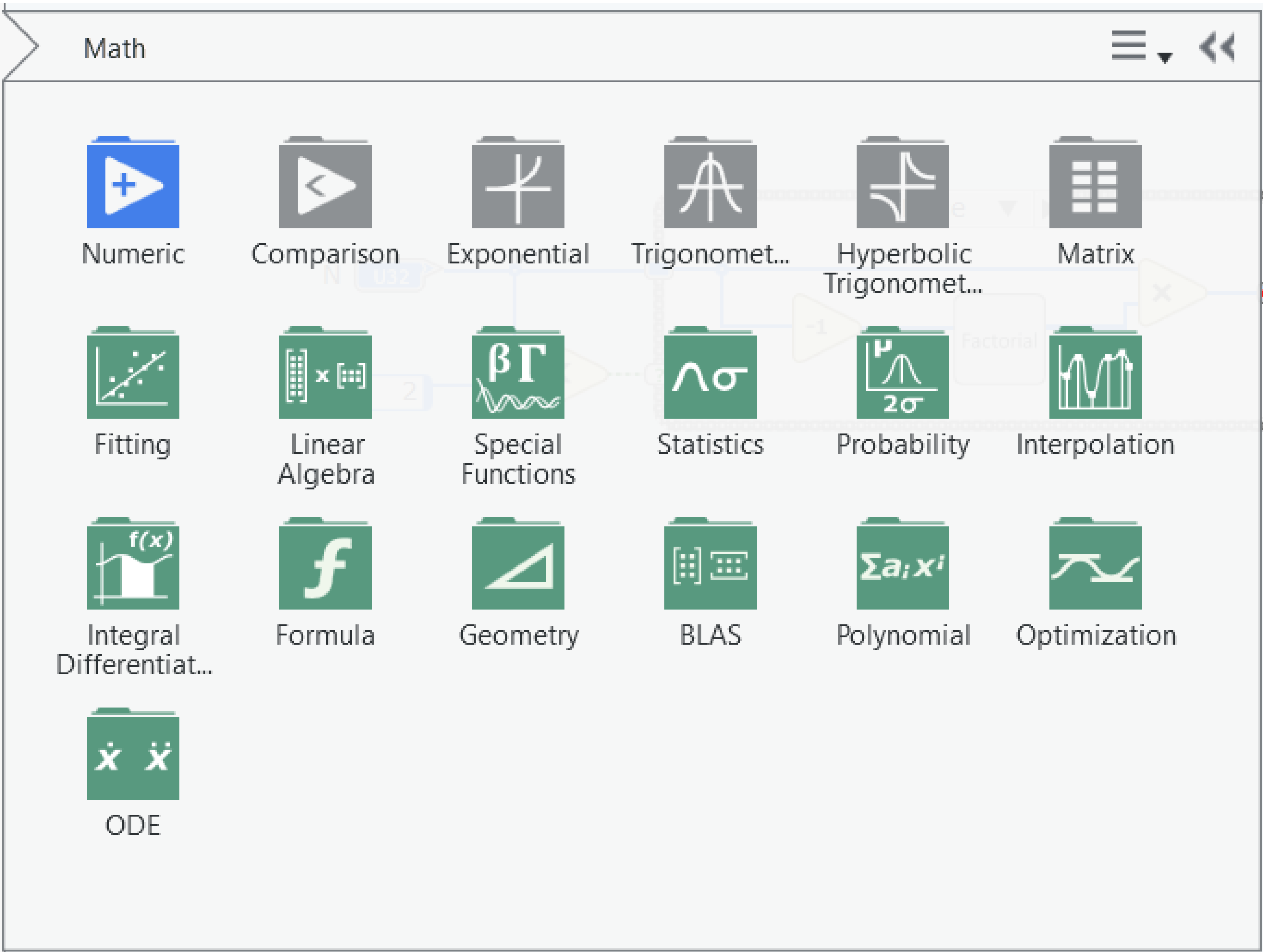


Functional units

Many numeric processing elements

Multiple values can be bundled in buses

Wires have types

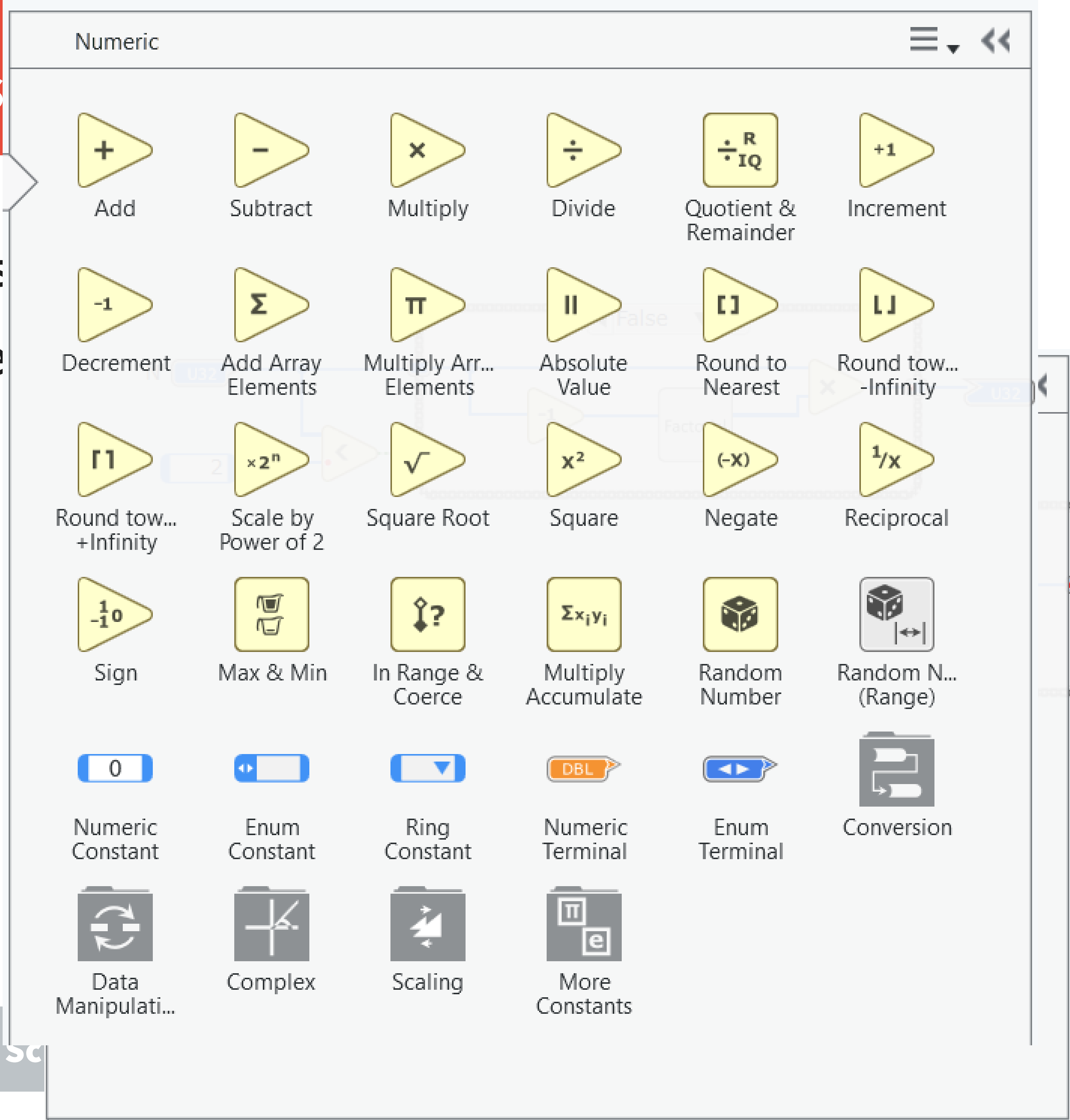


Functional units

Many numeric process

Multiple values can be bundled in buses

Wires have types

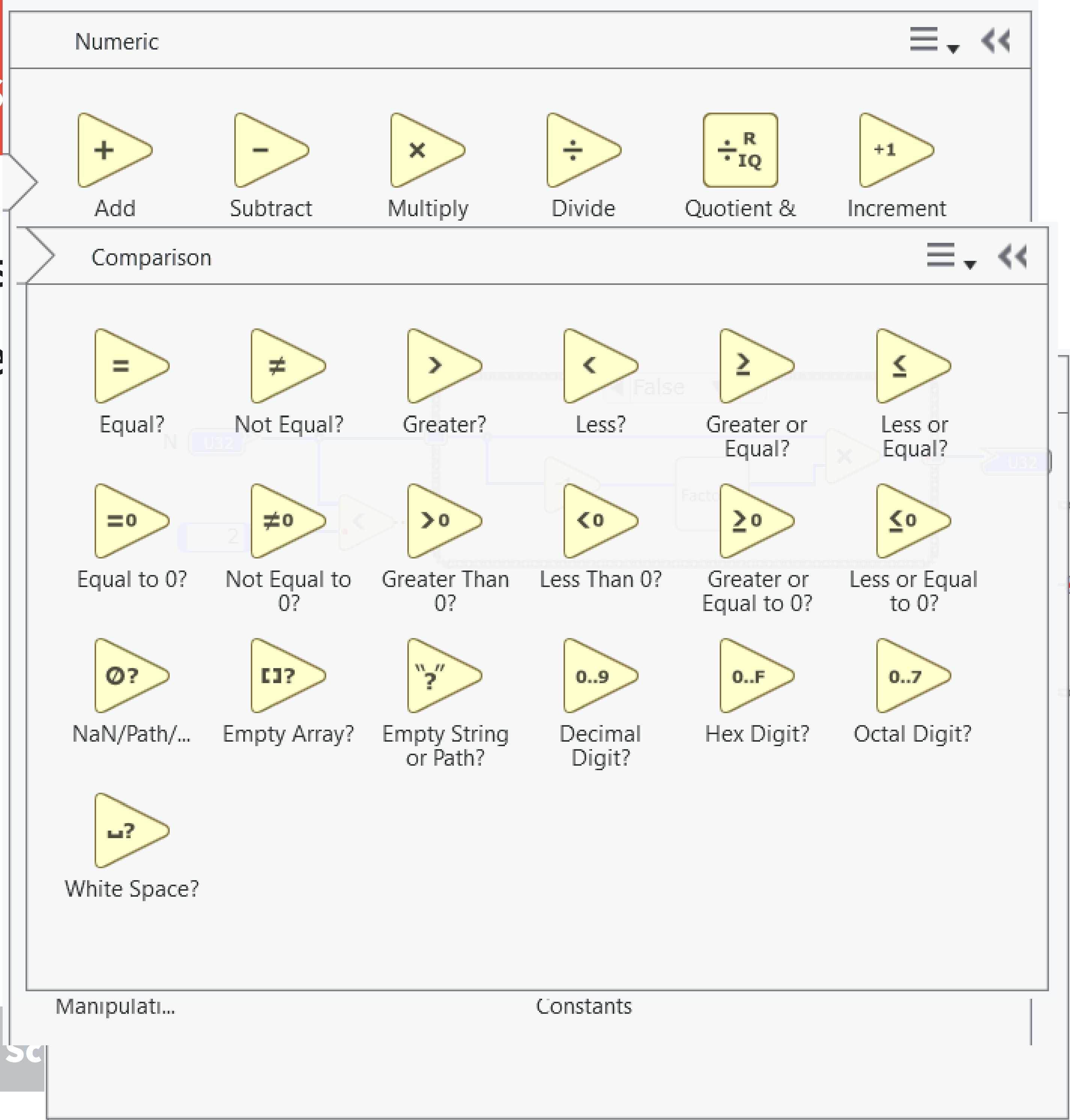


Functional units

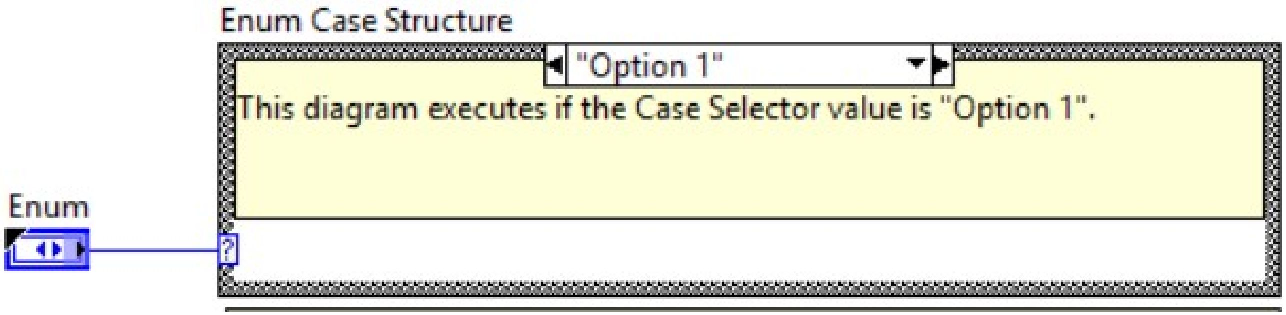
Many numeric process

Multiple values can be bundled in buses

Wires have types



Control structures and scope

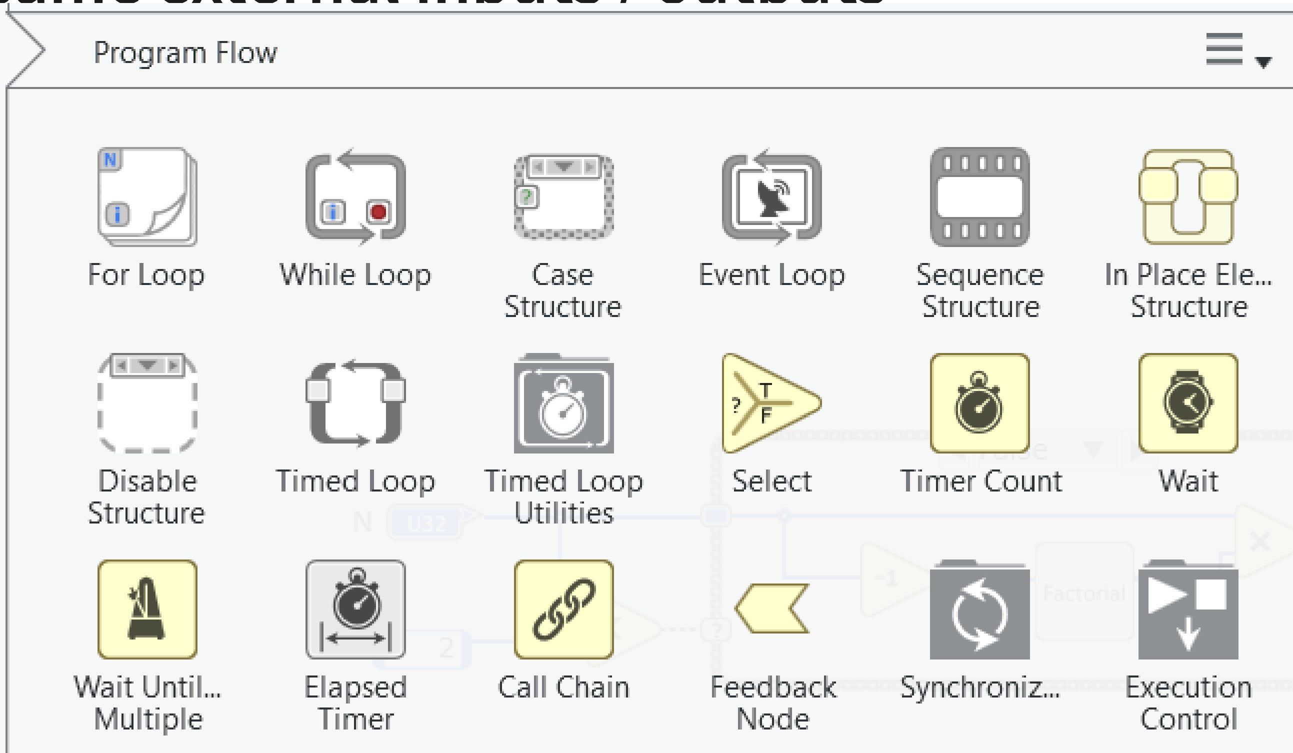


This diagram executes if the Case Selector value is "Option 1".

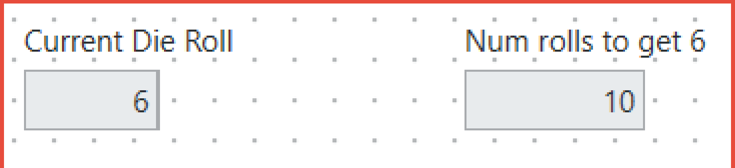
Control structures are represented as boxes

- on the border there is a <u>conditional/control input</u> connector
- the box is the equivalent of a parenthesis
- multiple cases (if-then-else, switch-case) become “pages”
- the box title contains the options of the case/condition
- all “pages” share the same external inputs / outputs
- control values (index)
  are present in all pages

There are also boxes for <u>formulas</u> or <u>external code</u> (ASM/C/C++)

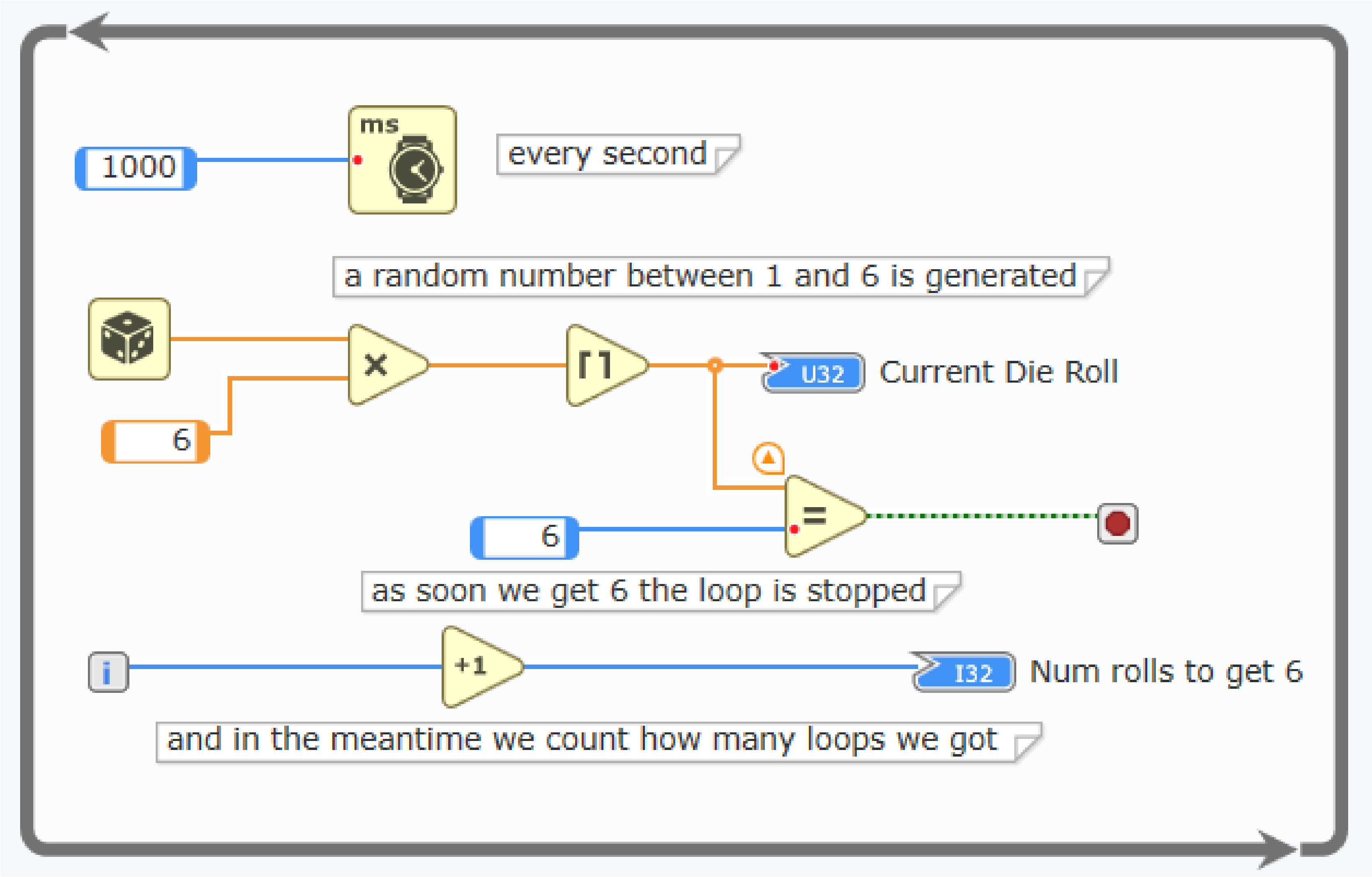


While loop example



Current Die Roll 6

Num rolls to get 6 10



every second

a random number between 1 and 6 is generated

Current Die Roll

as soon we get 6 the loop is stopped

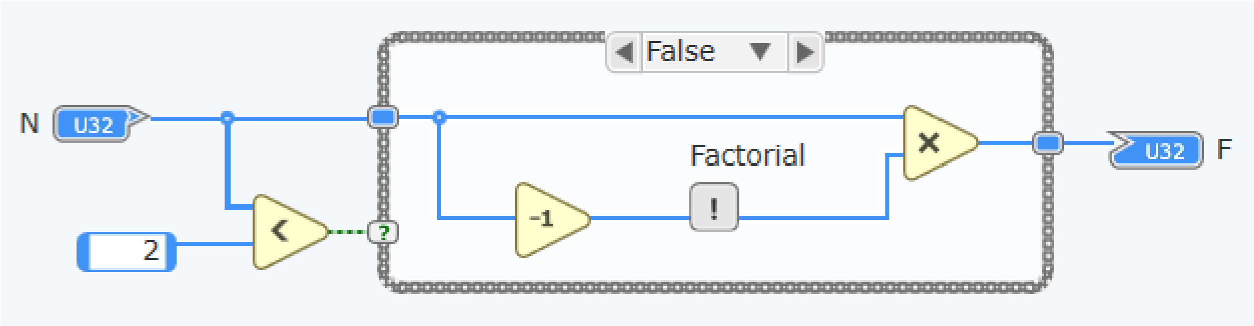
Num rolls to get 6

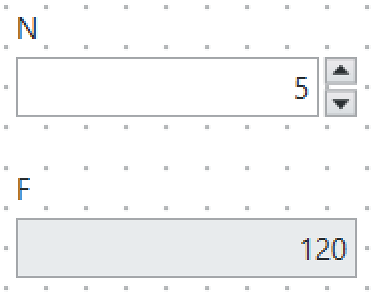
and in the meantime we count how many loops we got

Recursion? YES

Define a block as “reentrant” (i.e. allowing multiple parallel copies)

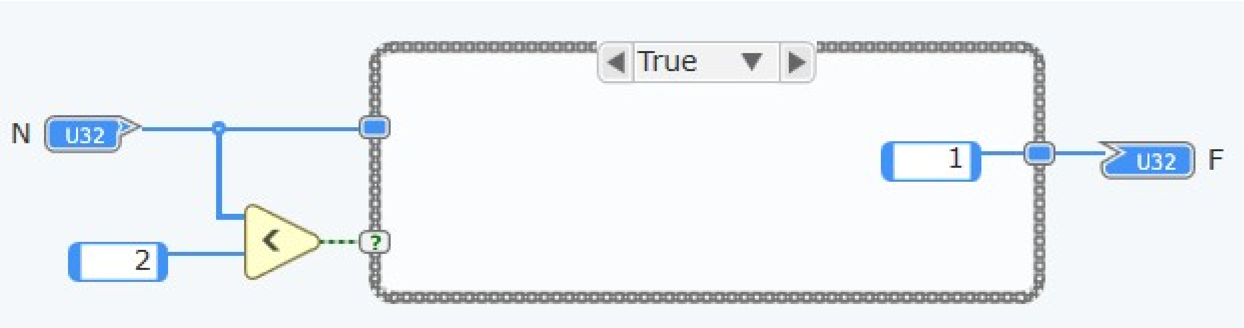
Then you can call it inside the same block or one of its sub-blocks





N: 5

F: 120



NOTE: you can also define “code” blocks with C

Concurrency

Inherently parallel

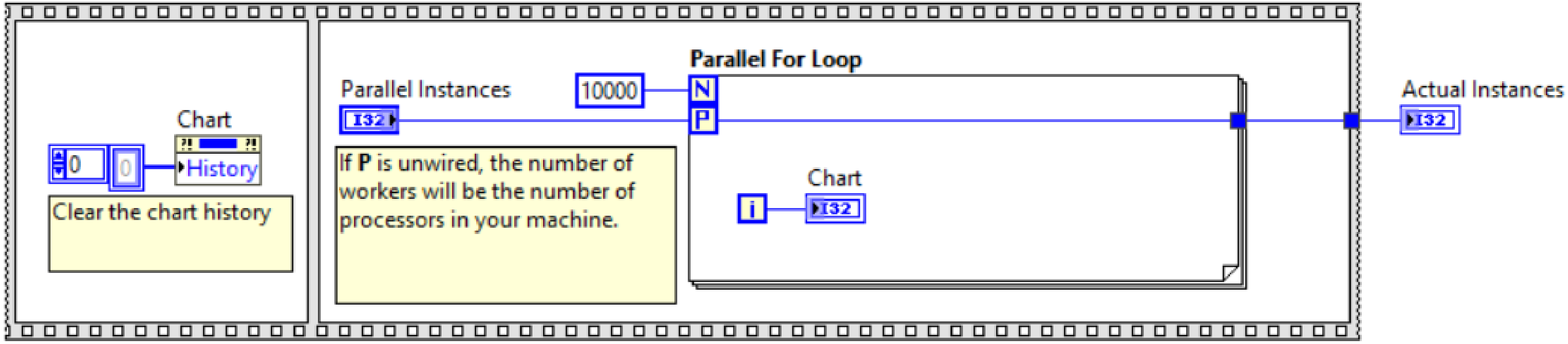
- linked units must run sequentially because of data dependency
- NON-linked units run in parallel (emulated)

Synchronization

- a block starts when all input data is available

Sequencing constraints

- data dependencies (links induce time order)
- you can add time dependencies without data exchange
  (or you could add data dependencies to do the same)



Clear the chart history

If P is unwired, the number of workers will be the number of processors in your machine.

LabView programming style

Data-flow visual design

Visual construction of the data-flow diagram

Visual test of the diagram

all blocks have their GUI showing IN/OUT data

probes can be added to show internal wires' values

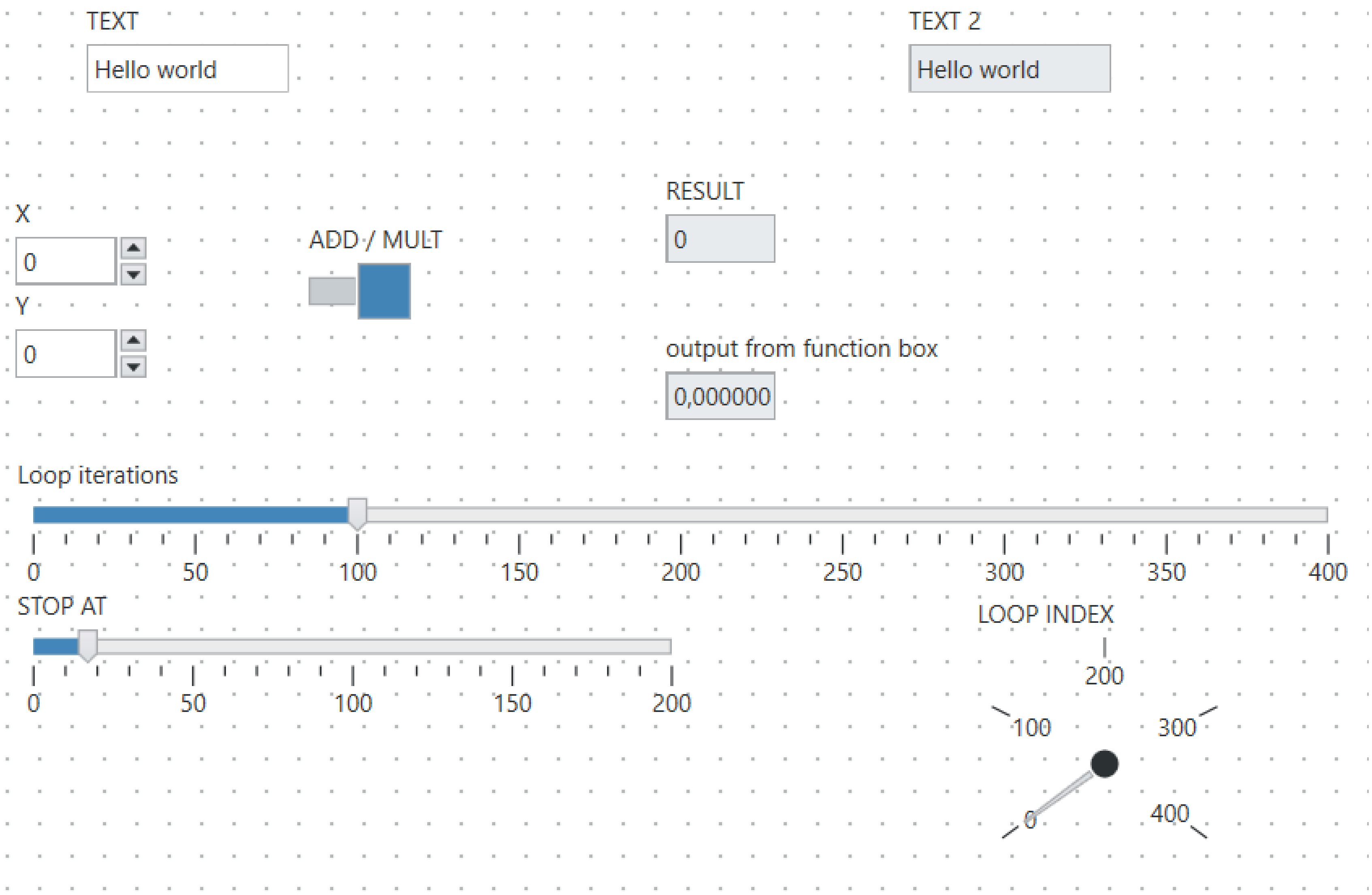
Inherently parallel (you just forget about sequentiality constraints)

Object-Oriented (classes)

Interaction with other systems:

- Function blocks for data math manipulation
- Code blocks for special algorithms
- Many libraries for Statistics, Signal analysis/manipulation, Math

Each functional unit has a GUI
many widgets are available (active or read-only)



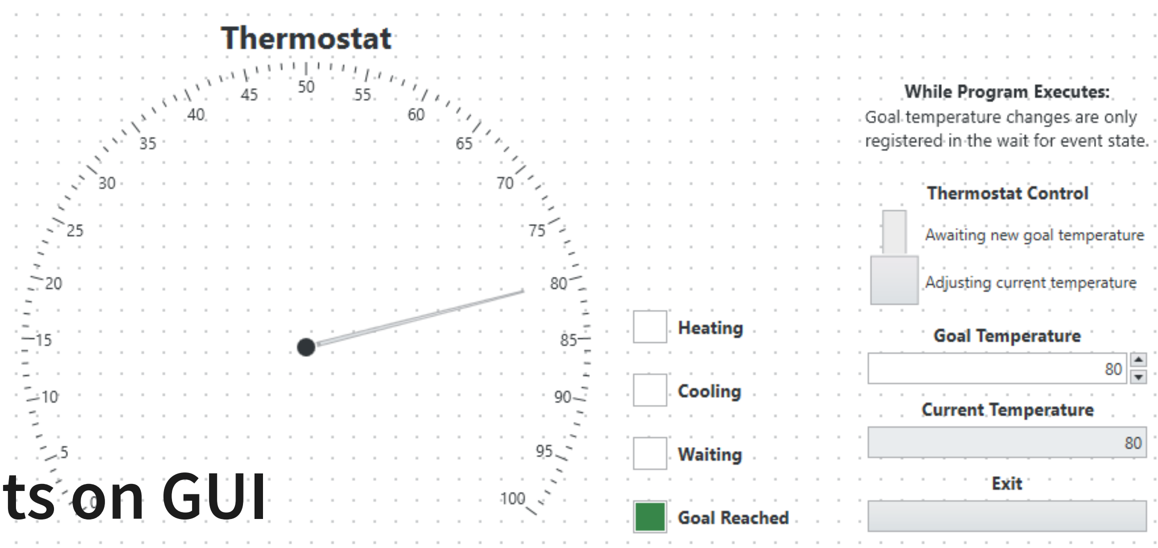
Debugging

Visual tracing of data on wires

GUI for blocks IN/OUT

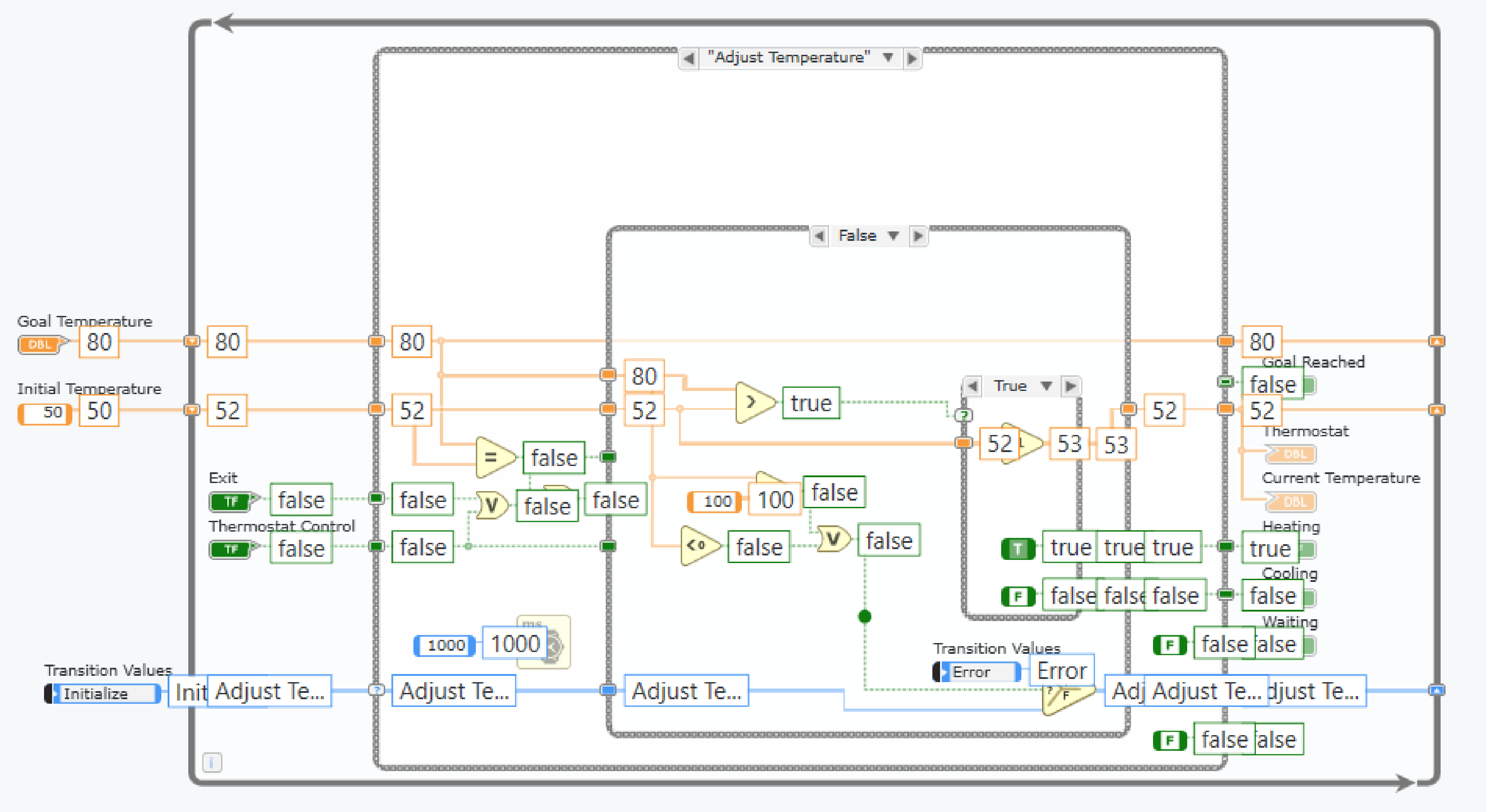
Probes on wires show as widgets on GUI

Values are shown on wires



While Program Executes:

Goal temperature changes are only registered in the wait for event state.



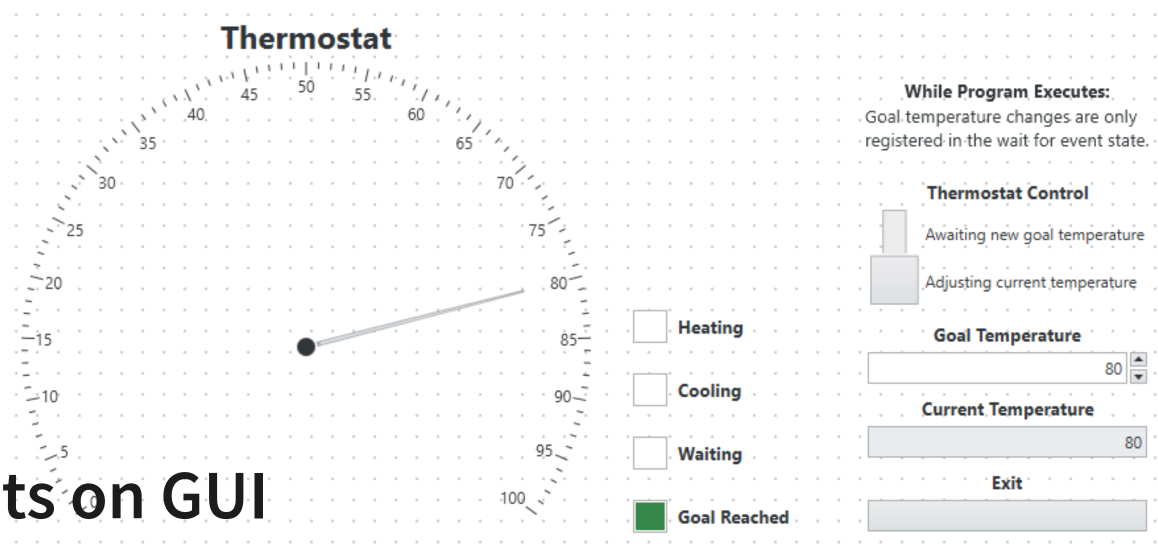
Debugging

Visual tracing of data on wires

GUI for blocks IN/OUT

Probes on wires show as widgets on GUI

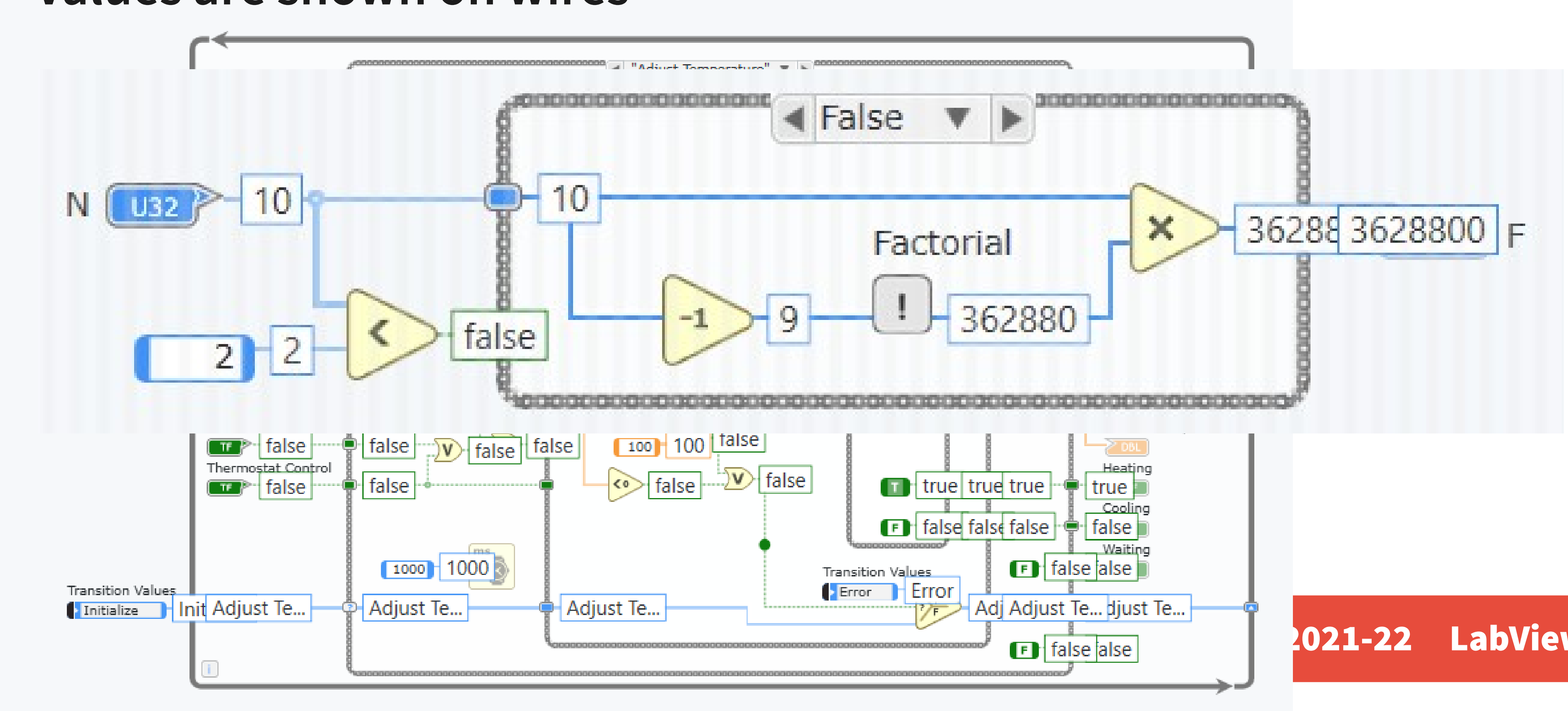
Values are shown on wires



While Program Executes:
Goal temperature changes are only registered in the wait for event state.

Goal Temperature 80

Current Temperature 80



Demo

DEMO