

Fast & Flexible: LabVIEW FPGA for Programmers

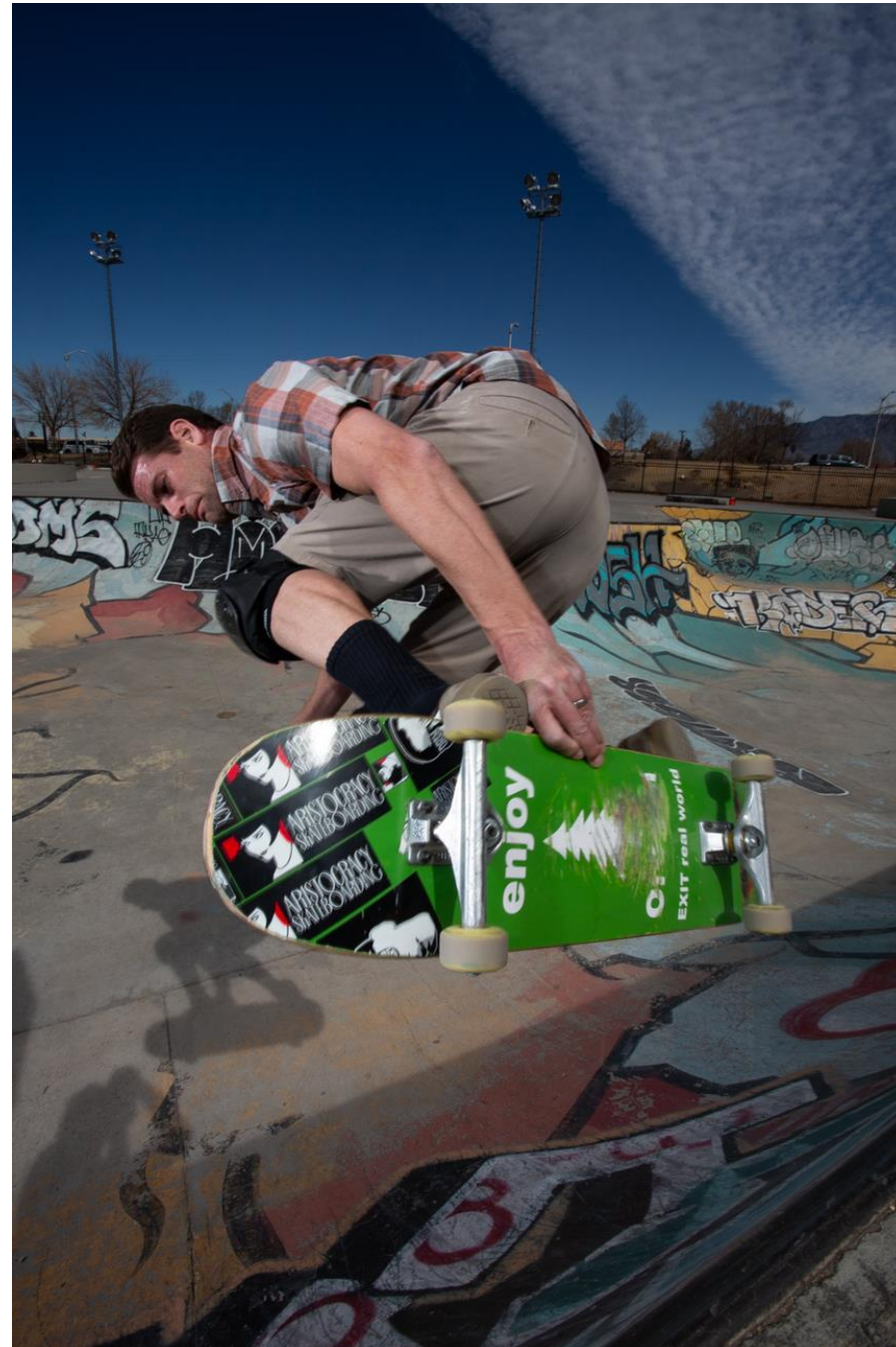
Luke Graham

Who is this for?

- This is not simple. ￣_(\ツ)_/￣
- LabVIEW programmers who want to learn more about LV FPGA
- HDL programmers who want to learn more about LV FPGA
- Part 1 of 2

Luke Graham

- I live in New Mexico
- Test & Measurement since 2006
- LabVIEW Architect since 2012
- Field AE for NI

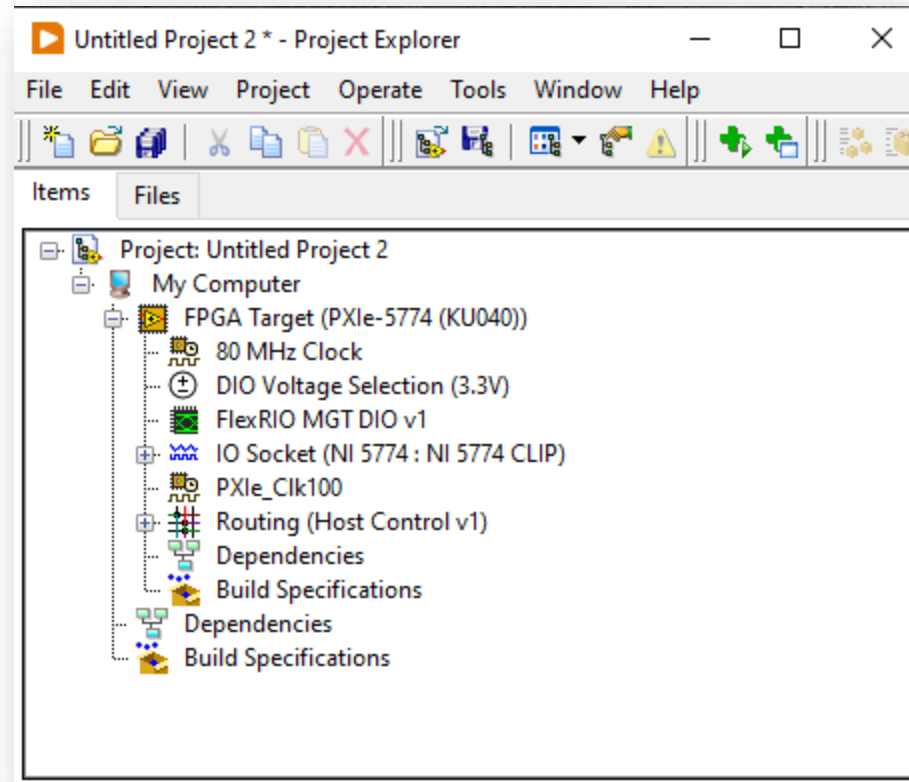
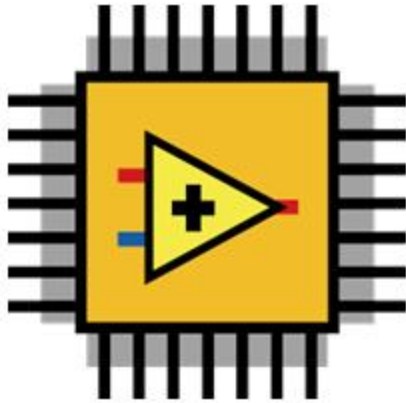


My Objective

- **Really Good Building Blocks**
- **Inform your questions**

LabVIEW FPGA Development

Walkthrough



Why use LabVIEW FPGA?

Why use LabVIEW FPGA?

1. The platform
2. The performance
3. The customization

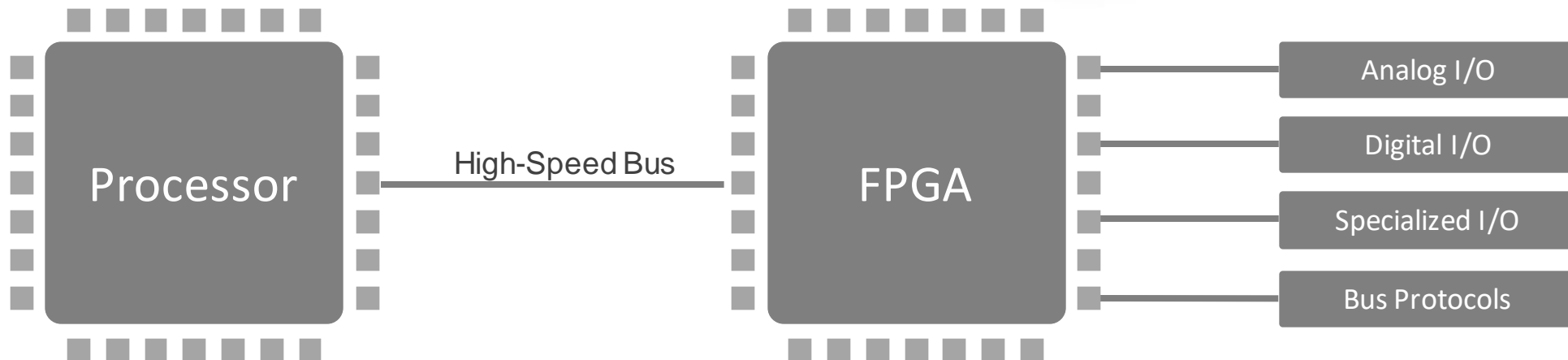
Platforms



OR



CompactRIO

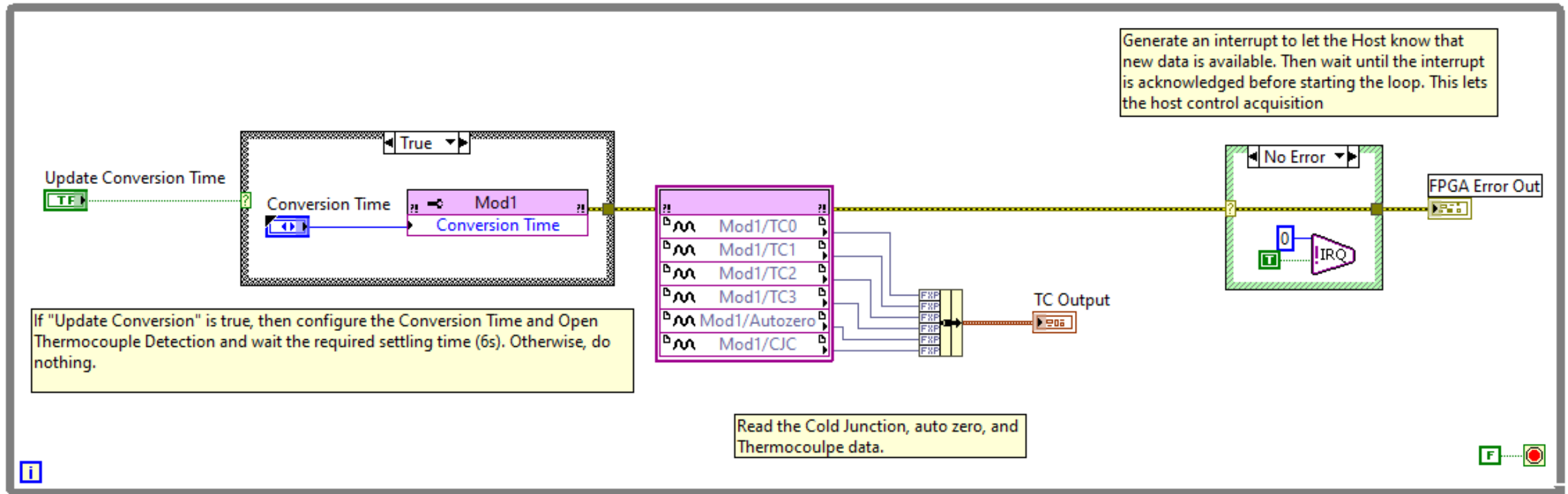




CompactRIO

- Wave energy research one mile off the coast of Bolivia
- Molten lead circuit for heat transfer research
- Emissions reductions for large diesel power generators
- Thermal chamber control for simulating earth orbit
- Geologic imaging ¼ mile underground
- Energy monitoring for cold fusion research 🙄

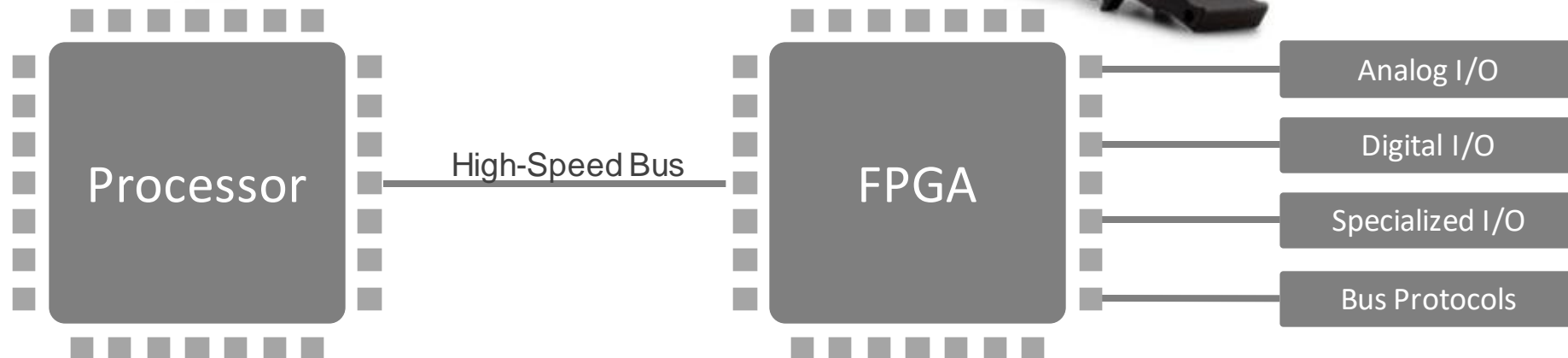
CompactRIO



FlexRIO or Reconfigurable Scopes



FlexRIO or Reconfigurable Scopes



LabVIEW FPGA Concepts

- Single-Cycle Timed Loops
- Pipelining
- Fixed-Point
- High Throughput Math
 - Handshaking
- Fixed Arrays
- Memory
- Clock Domains
- Xilinx IP
- Data Transfer to/from Host: Fast & Slow

Glossary

- FPGA
- FlexRIO or Reconfigurable Scopes
- Target
- Host



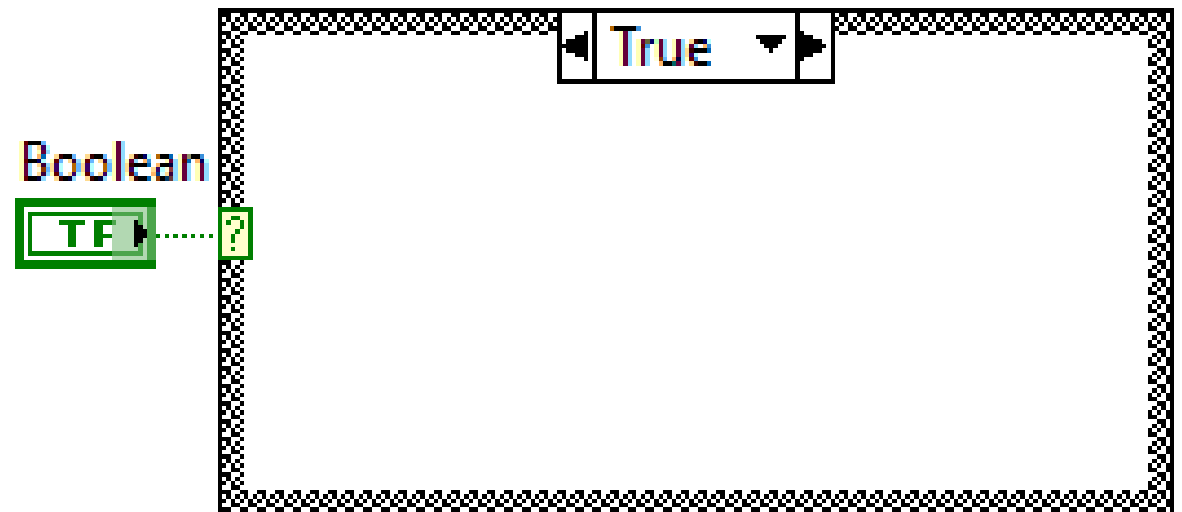
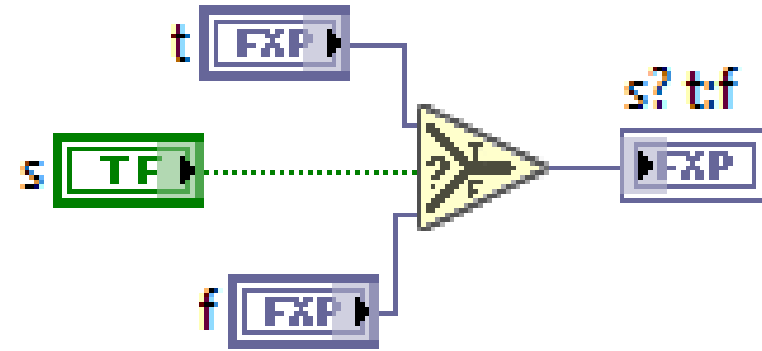
FPGAs: Flexible & Inflexible

- **Flexible:** I can define how I want my digitizer to work.
- **Inflexible:** At run time, everything must be defined. Can't "allocate" anything.
 - Includes Data type, array sizes
- We need to return data on every clock cycle, even if it's not valid.

FPGAs: Flexible & Inflexible

Everything must be defined

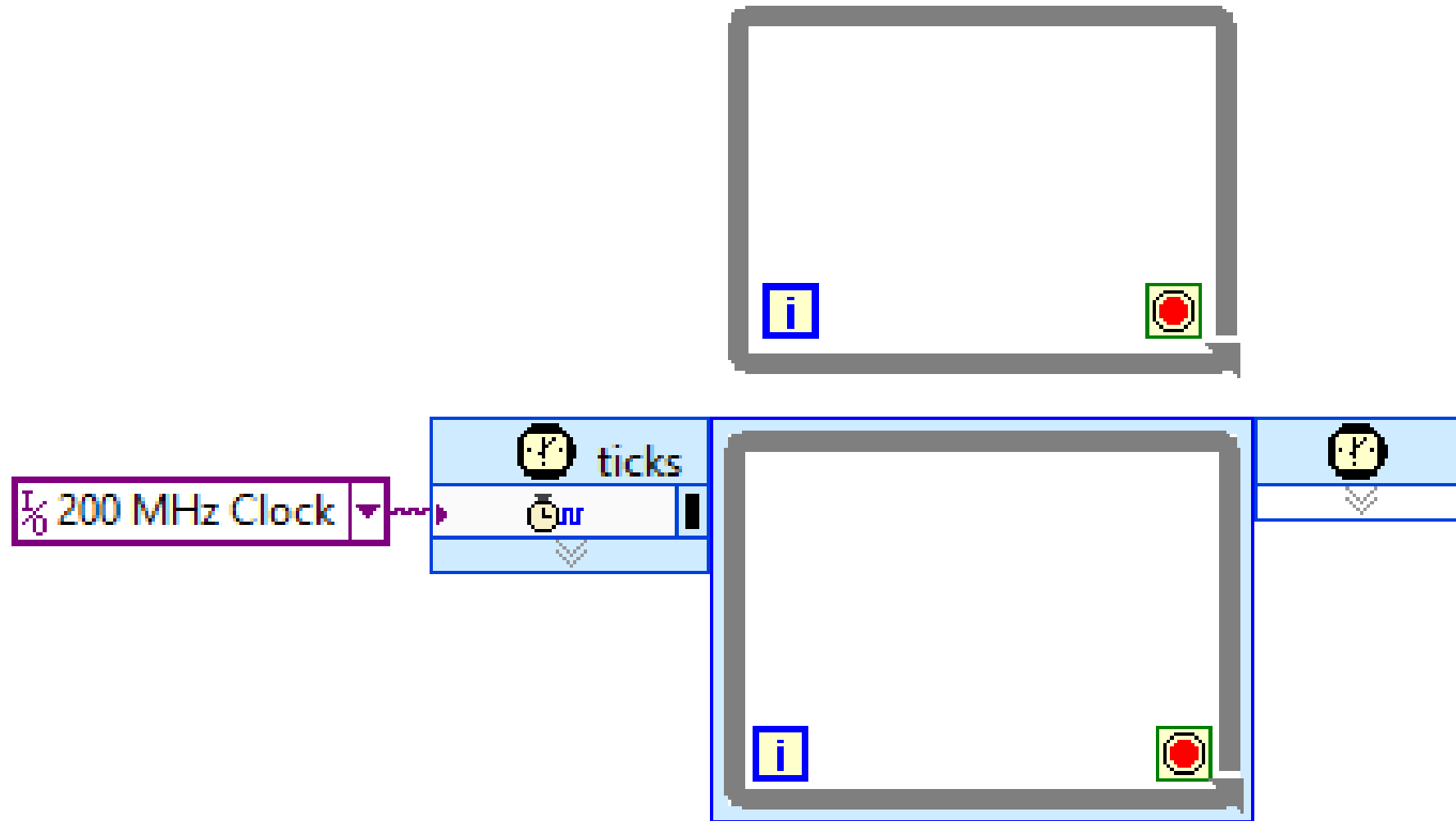
- Exceptions:
 - Trigger scenarios
 - Enable/disable
 - Compare to elapsed count
 - Etc.



FPGA: Keep in Mind

- Compile Time
- Available Resources
- Ability to “meet timing”

Single-Cycle Timed Loops



Pipelining

$$7+7+7+7+7+7+7+7+7+7=?$$

Pipelining

$7+7=?$

$\text{Ans}+7=?$

$\text{Ans}+7=?$

$\text{Ans}+7=?$

$\text{Ans}+7=?$

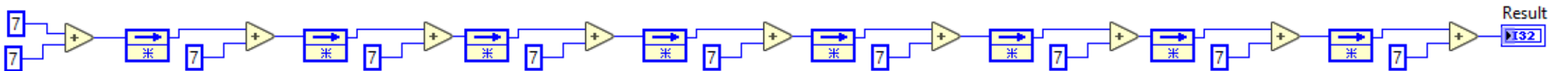
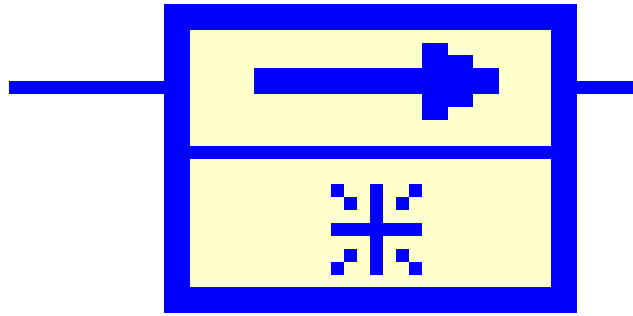
$\text{Ans}+7=?$

$\text{Ans}+7=?$

$\text{Ans}+7=?$

Pipelining

- Feedback Nodes!



Vocabulary Word

- Latency: The number of cycles before a valid answer is returned
- What is the latency of the previous example?


Fixed-Point Data

1 1.001 1.0009765625

Numeric Constant Properties: Value

Appearance | **Data Type** | Display Format | Documentation

Representation

 ☐ Adapt to entered data

Fixed-Point Configuration

Encoding	Range
<input checked="" type="radio"/> Signed <input type="radio"/> Unsigned	Minimum -128
Word length 12 bits	Maximum 127.9375
Integer word length 8 bits	Delta 0.0625
<input type="checkbox"/> Include overflow status	

OK Cancel Help

Fixed-Point Data

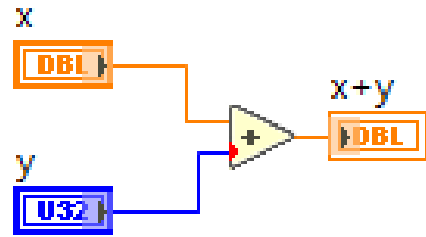
- Fixed-Point Data Type errors are my most common bug in FPGA.
 - Saturated Data
 - LSB underflow

Fixed-Point Data

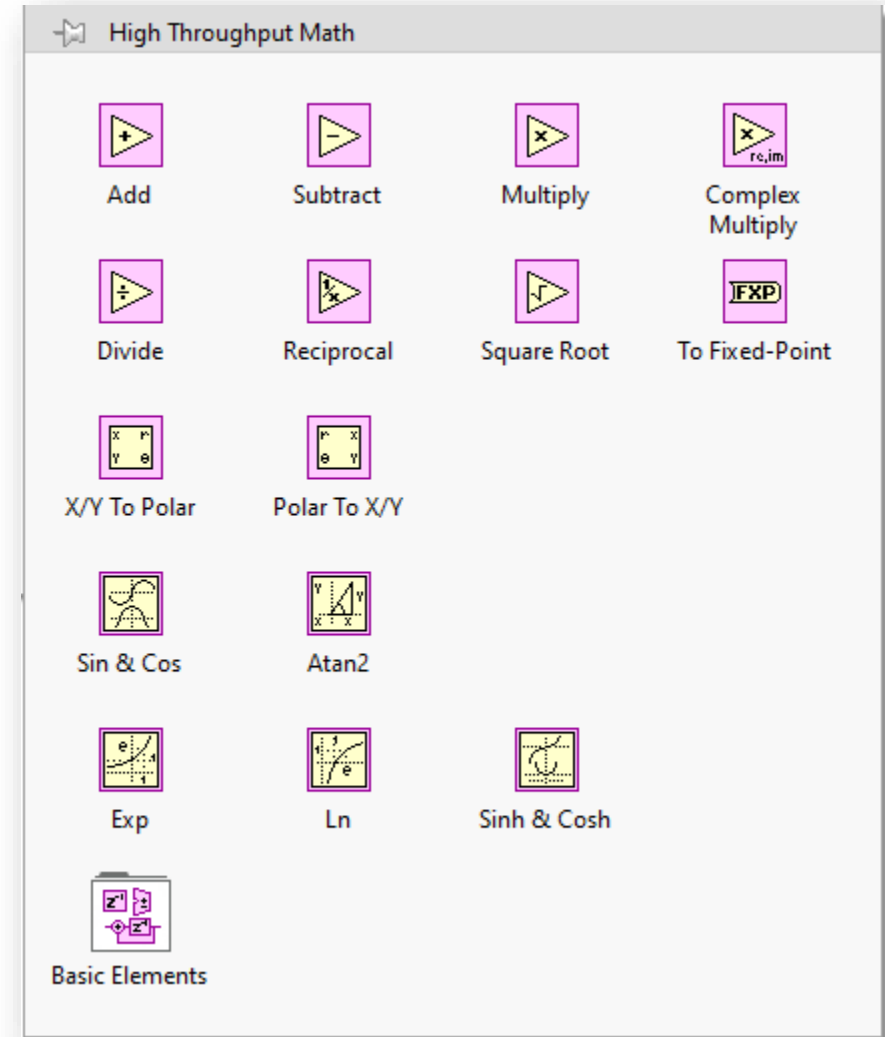
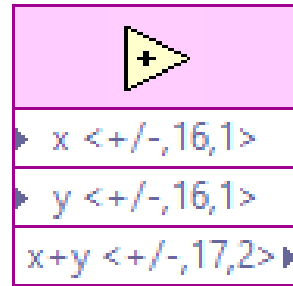
- Fixed-Point Data Type errors are my most common bug in FPGA.
 - Saturated Data
 - LSB underflow

(I drop indicators at every step of my algorithm for debugging)

High Throughput Math

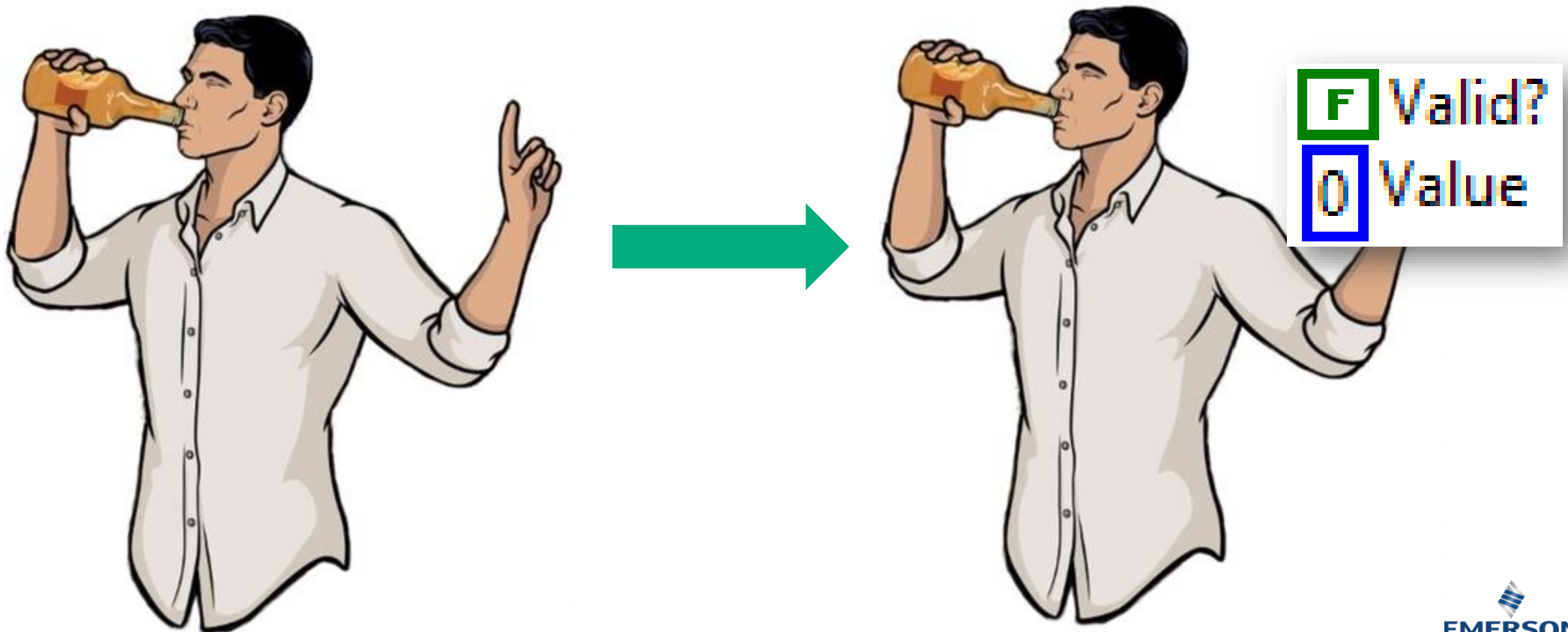


VS

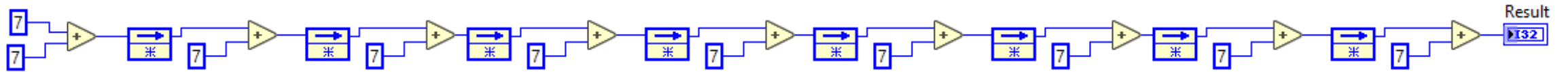


Handshaking

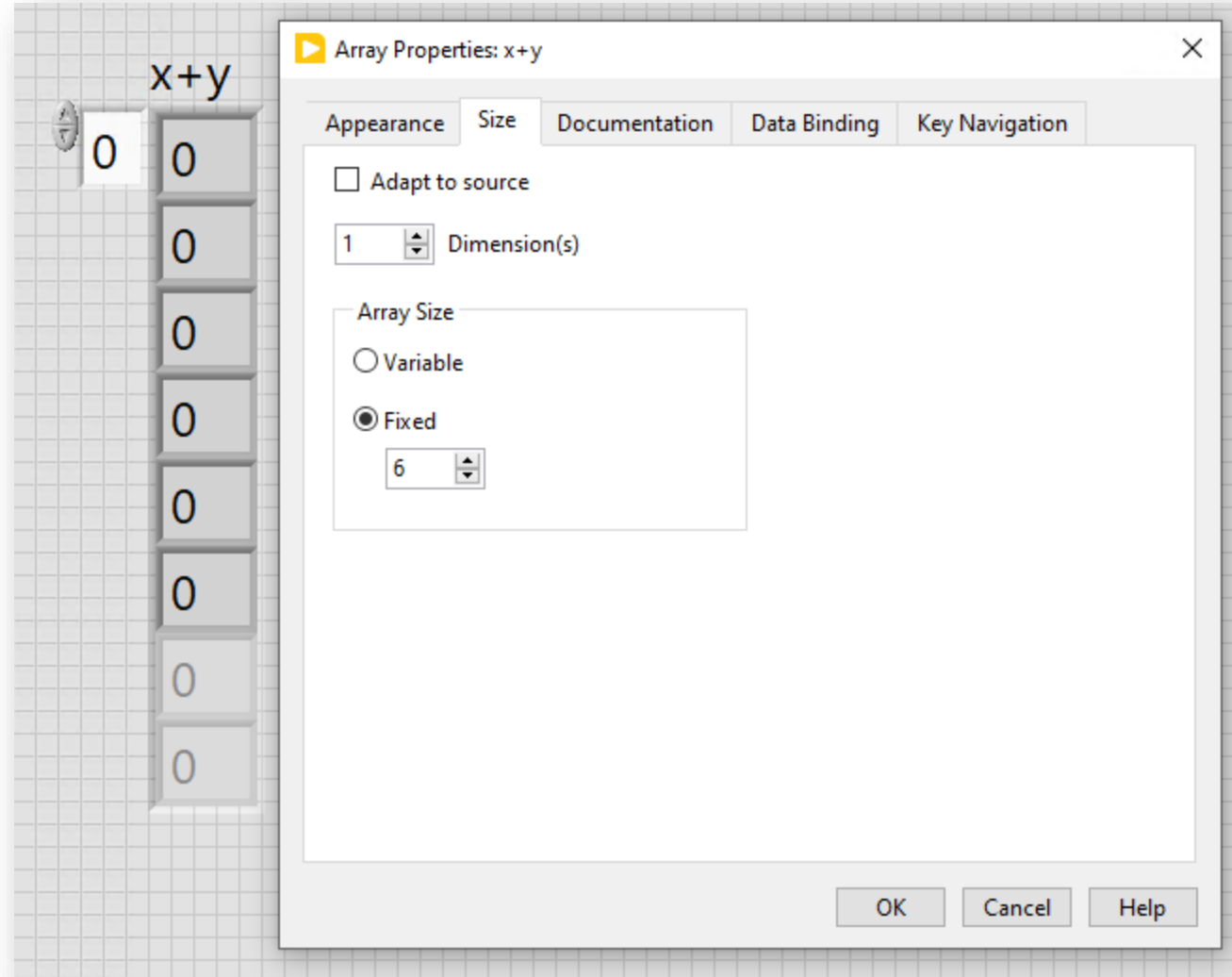
- We MUST return a value on every cycle, even if we have no data.
- (No such thing when using a processor)



Handshaking



Fixed Arrays

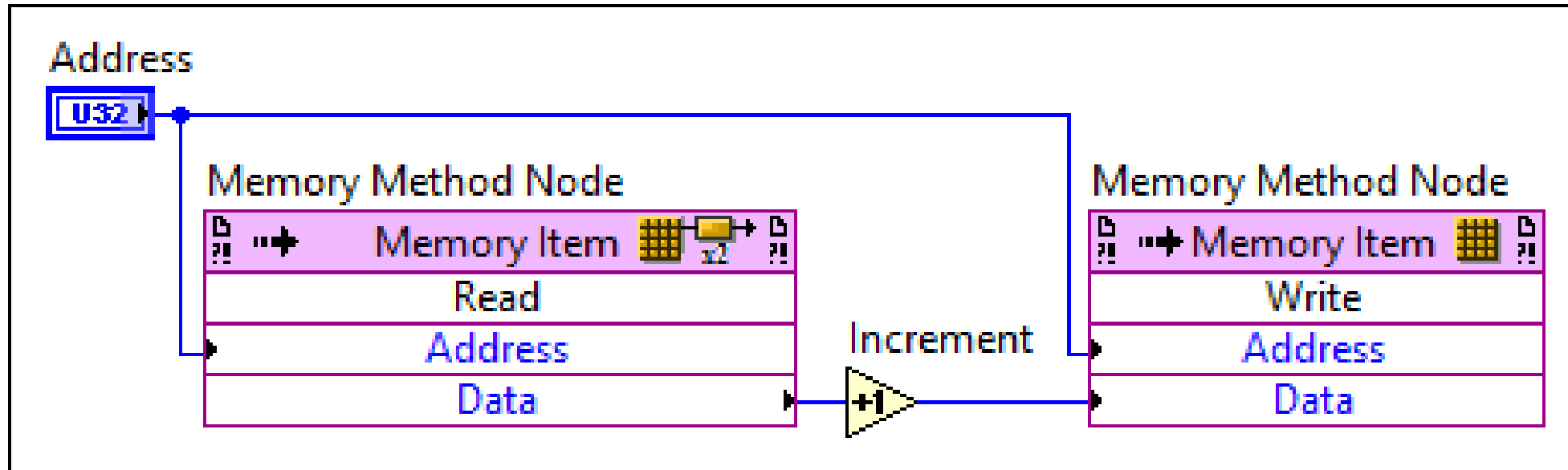


Memory

Use Cases:

- Buffer data that is acquired point-by-point
- Share data between loops or clock domains
- Use data sets without multiple copies

Memory



Memory

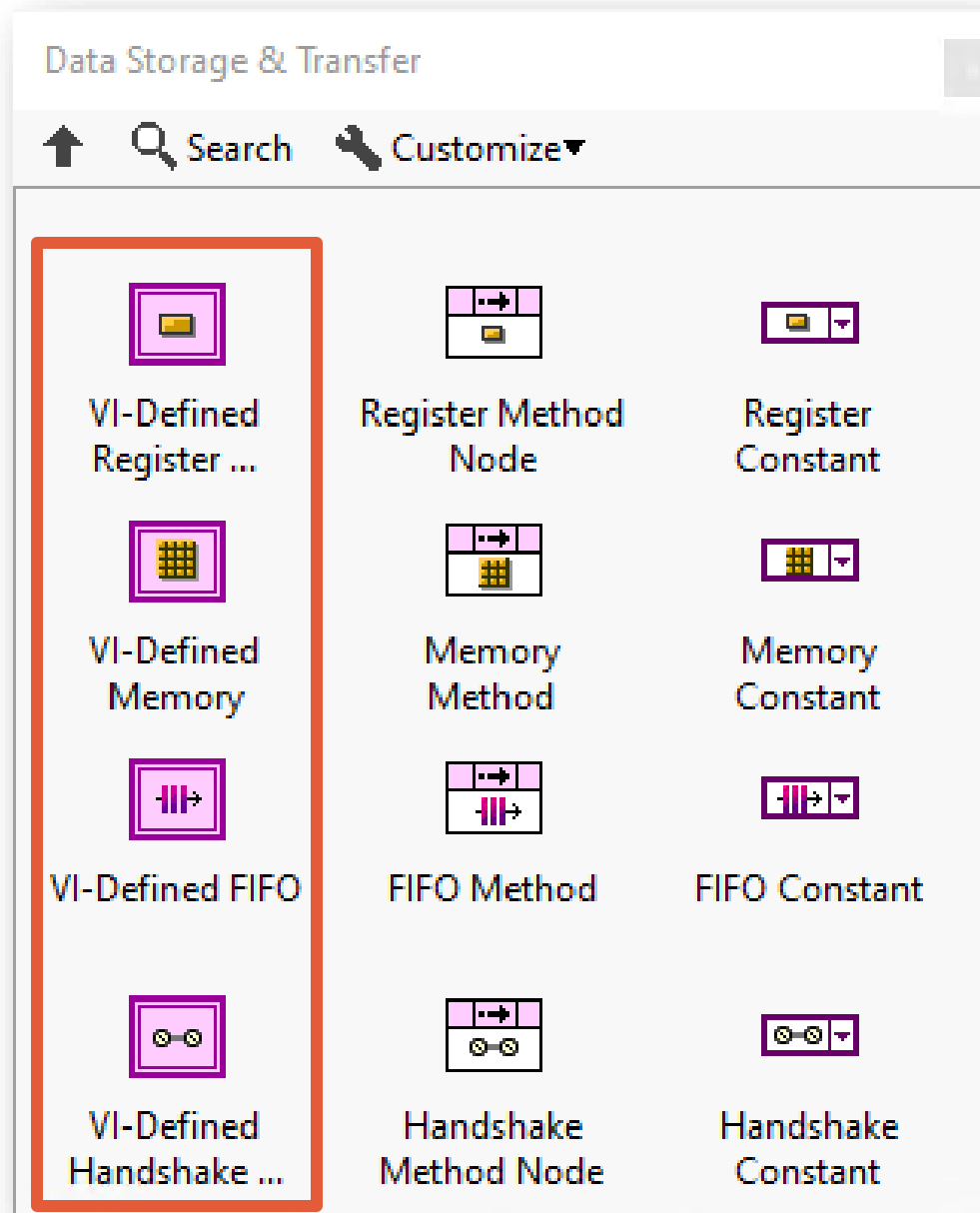
Distinct types of Memory

- VI-Defined
- Target-Scoped

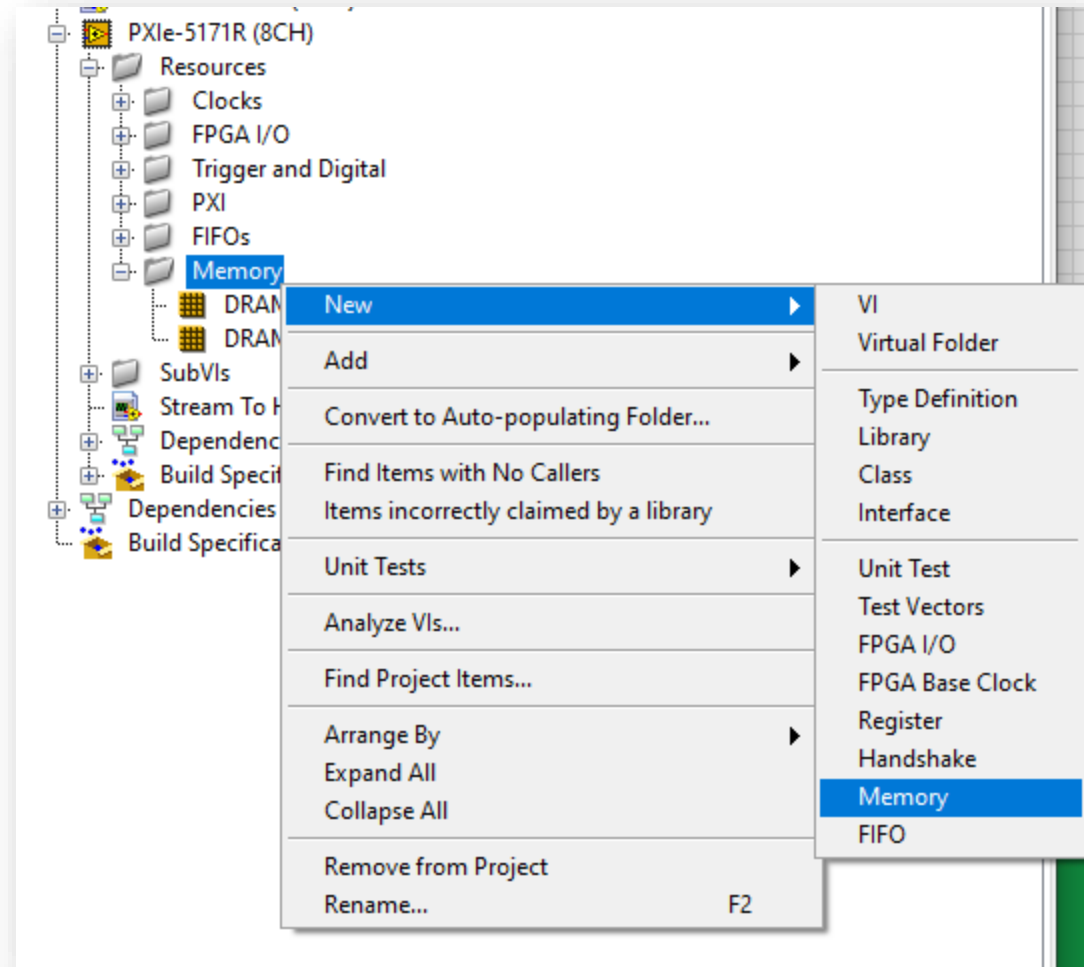
Physical Implementation

- Look-Up Table
- Block Memory
- DRAM

Memory

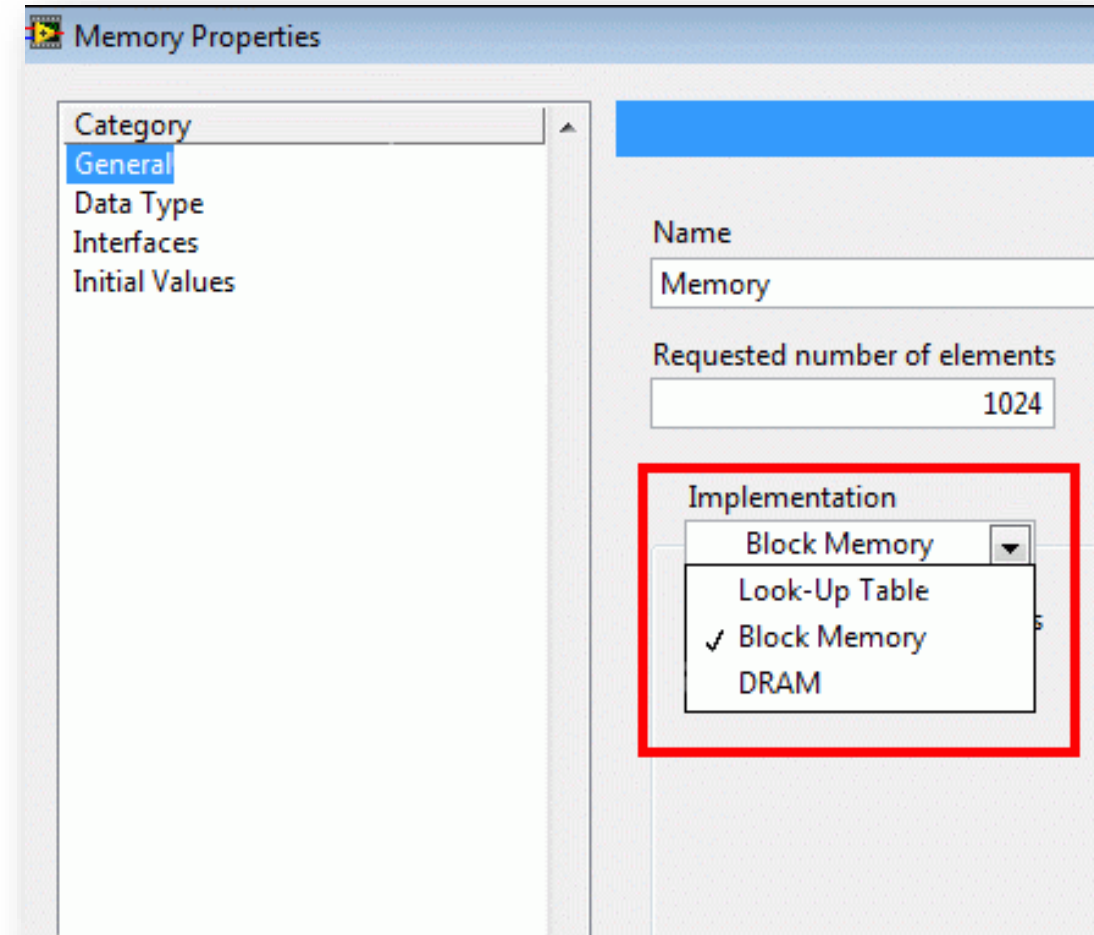


Memory

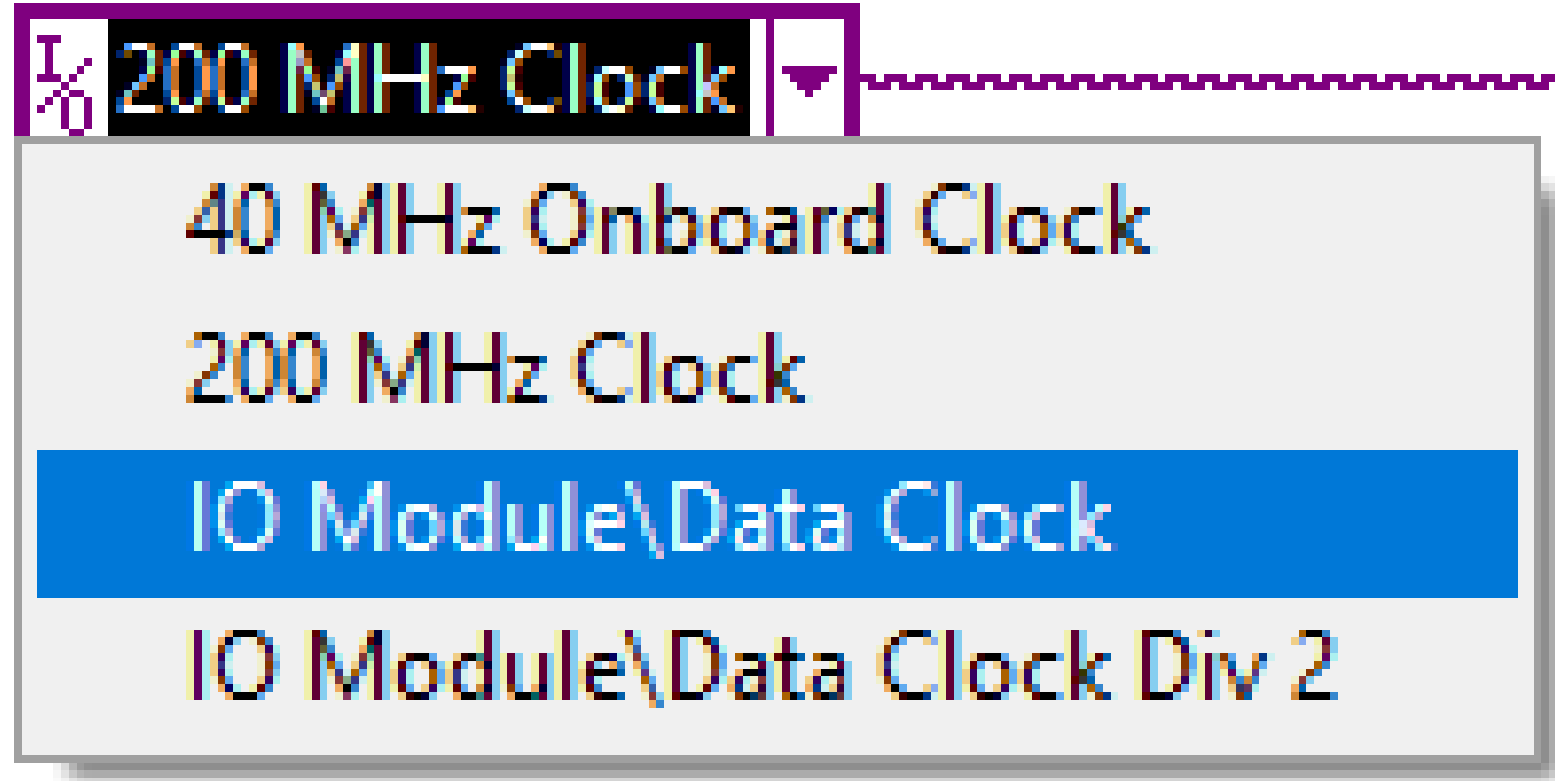


Memory

- Look-Up Table
 - Logic Gates that are hard-wired on the FPGA.
 - Consumes FPGA resources (can be used as FPGA resources OR memory)
 - Use when:
 - Needed in a SCTL with no latency.
 - Limited remaining Block Memory
- Block Memory
 - Can cross clock domains
 - Does not consume FPGA resources (use this first when possible)
- DRAM
 - External to the FPGA
 - HUGE data sets are possible.
 - Requires multiple clock cycles.
 - Sequential access (only one command per cycle)



Clock Domains



Clock Domains

FPGA Derived Clock Properties

Category
General

General

Name: 35MHz

Parent Clock Name: 200 MHz Clock

Desired Derived Frequency: 35 MHz

Actual Derived Configuration

Parent Frequency	Multiplier	Derived Frequency
200MHz	7	35MHz
	Divisor	
	40	

Parent Period: 5ns

Derived Period: 28.57142857142857ns

Message

The desired derived frequency is available.

OK Cancel Help

Clock Domains

- Slower clock rates can meet timing more easily.
- Use after decimating data to run more efficiently.
(Rather than “invalid” 9,999 times out 10,000.)
- Use to facilitate communication with other devices.
- Use to run at desired control loop rates.

Xilinx IP

IP Symbol

Freq. Response

Implementation Details

Coefficient Reload

Ideal

Quantized

Frequency Response (Magnitude)

Magnitude (dB)

Normalized Frequency (x PI rad/sample)

(0.250,-8.12)

Set to Display

1

[1 - 1]

Filter Analysis

Pass Band

Range : 0.0 - 0.5

Min	-78.421369 dB
Max	0.000000 dB
Ripple	78.421369 dB

Stop Band

Range : 0.5 - 1.0

Min	
Max	-47.003569 dB
Ripple	

Component Name

FIR_Compiler_0456C0B5E3864BD48D527D6610ADC2C0

Filter Options

Channel Specification

Implementation

Detailed Implementation

Interface

Summary

Filter Coefficients

Select Source

Vector

Coefficient Vector

-0.003673448703933533,-0.005724159328994389,-0.005069756523949517,0.0130488523674651

Coefficient File

no_coe_file_loaded

Number of Coefficient Sets

1

[1 - 1024]

Number of Coefficients (per set): 15

☐ Use Reloadable Coefficients

Filter Specification

Filter Type

Decimation

Inferred Coefficient Structure(s) : Non Symmetric

Rate Change Type

Integer

Interpolation Rate Value

1

[1 - 1]

Decimation Rate Value

10

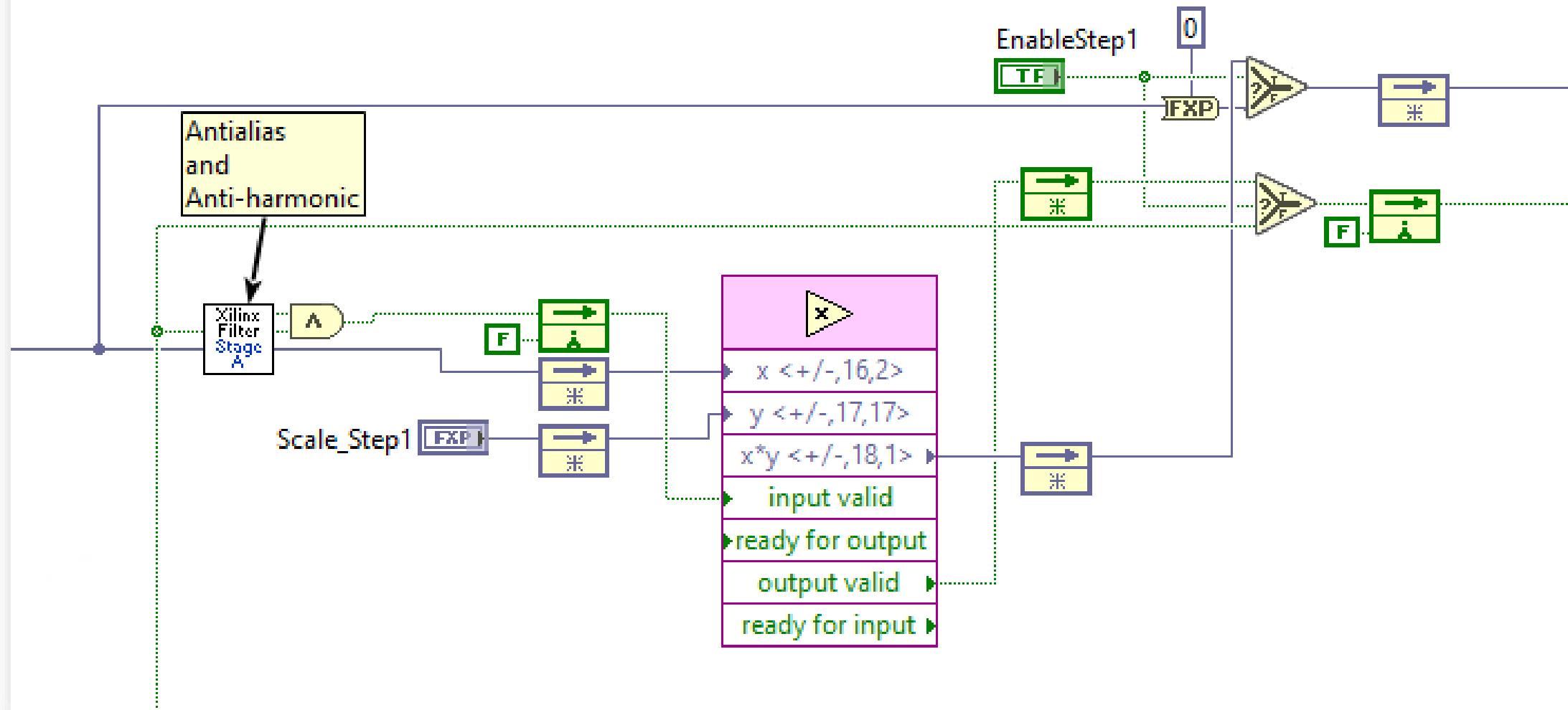
[2 - 1024]

Zero Pack Factor

1

[1 - 1]

All Together Now



Last Topic! Communicating FPGA/Host

- **Slow Data**

- Lossy
- Non-Deterministic

- **Fast Data**

- Lossless
- Deterministic

Communicating FPGA/Host

Slow Data

- **User Configured:** Sample Rates, Decimation Settings, Trigger Configuration
- **User Interpreted:** Status Indicators, Debug Indicators,

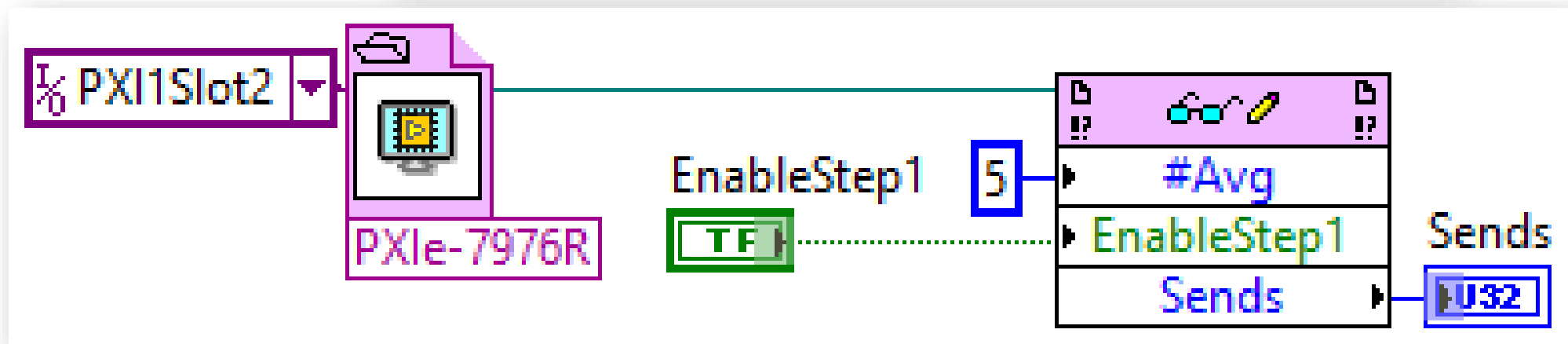
Communicating FPGA/Host

Slow Data

FPGA Front Panel



Host Block
Diagram



Communicating FPGA/Host

Fast Data

- **Acq Data:** Digitizer Data, Stuff you can't miss.
- **Generator Data:** Stream to AWG
- **Programmatic Transitions:** Alarms, Thresholds, Things that need timestamps

Communicating FPGA/Host

Fast Data

FIFO Properties

Category
General
Data Type
Interfaces

General

Name
ScaleMeas_AI0

Type
Target to Host - DMA ☐ Disable on Overflow

Requested Number of Elements
1023

Implementation
Block Memory

Actual Number of Elements
1023

Control Logic
Slice Fabric

OK Cancel Help

Communicating FPGA/Host

Fast Data

FIFO Properties

Category
General
Data Type
Interfaces

General

Name
ScaleMeas_AI0

Type
Target to Host - DMA ☐ Disable on Overflow

Requested Number of Elements
1023

Implementation
Block Memory

Actual Number of Elements
1023

Control Logic
Slice Fabric

OK Cancel Help

Communicating FPGA/Host

Fast Data

Data Type

Data Type

FXP

Fixed-Point Configuration

Encoding

☒ Signed

☐ Unsigned

Word length

16 bits

Integer word length

1 bits

Range

Minimum

-1

Maximum

0.999969

Delta

3.05176E-5

Communicating FPGA/Host

Fast Data

Interfaces

Arbitration for Read

Arbitrate if Multiple Requestors Only

Number Of Elements Per Read

1

Arbitration for Write

Never Arbitrate

Number Of Elements Per Write

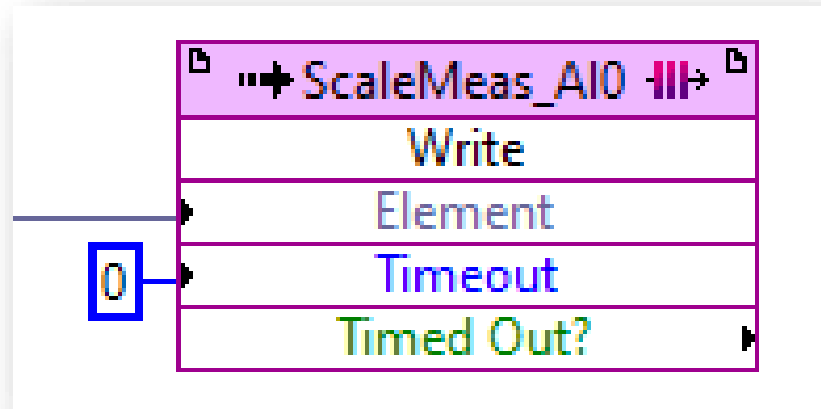
1

☒ 1
☐ 2
☐ 4
☐ 8
☐ 16

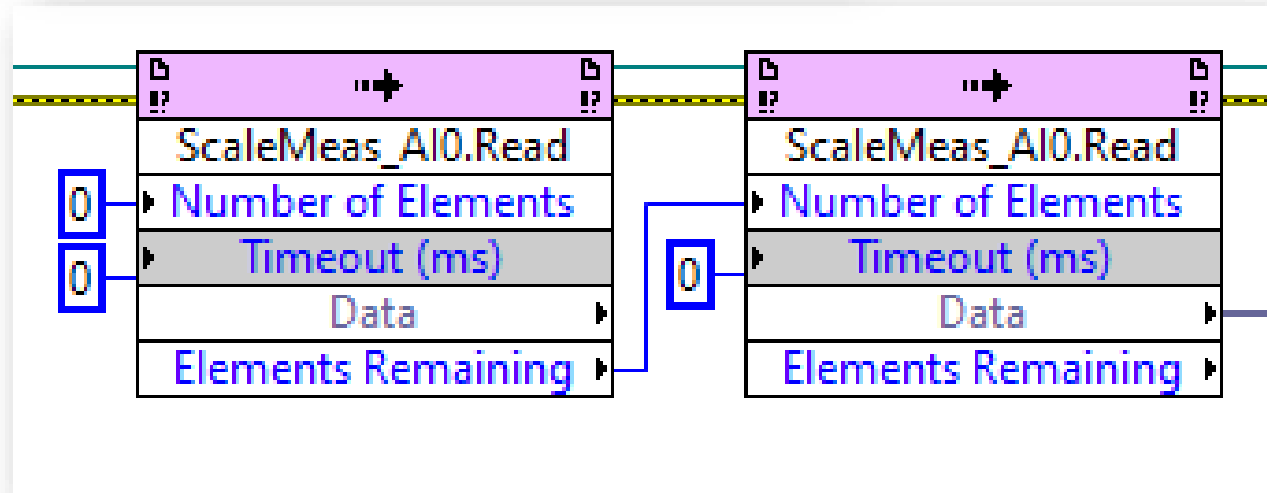
Communicating FPGA/Host

Fast Data

FPGA



Host



Questions?!

- Luke Graham – Field AE – luke.graham@ni.com
- Jacob Sees – Product Manager – jacob.sees@ni.com
- R&D Person?

Thank you!

Dataflow Diagram

