

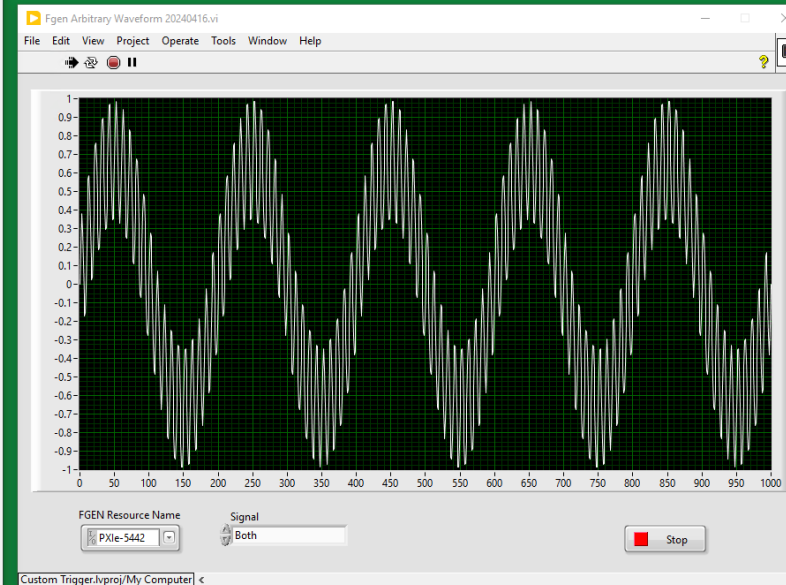
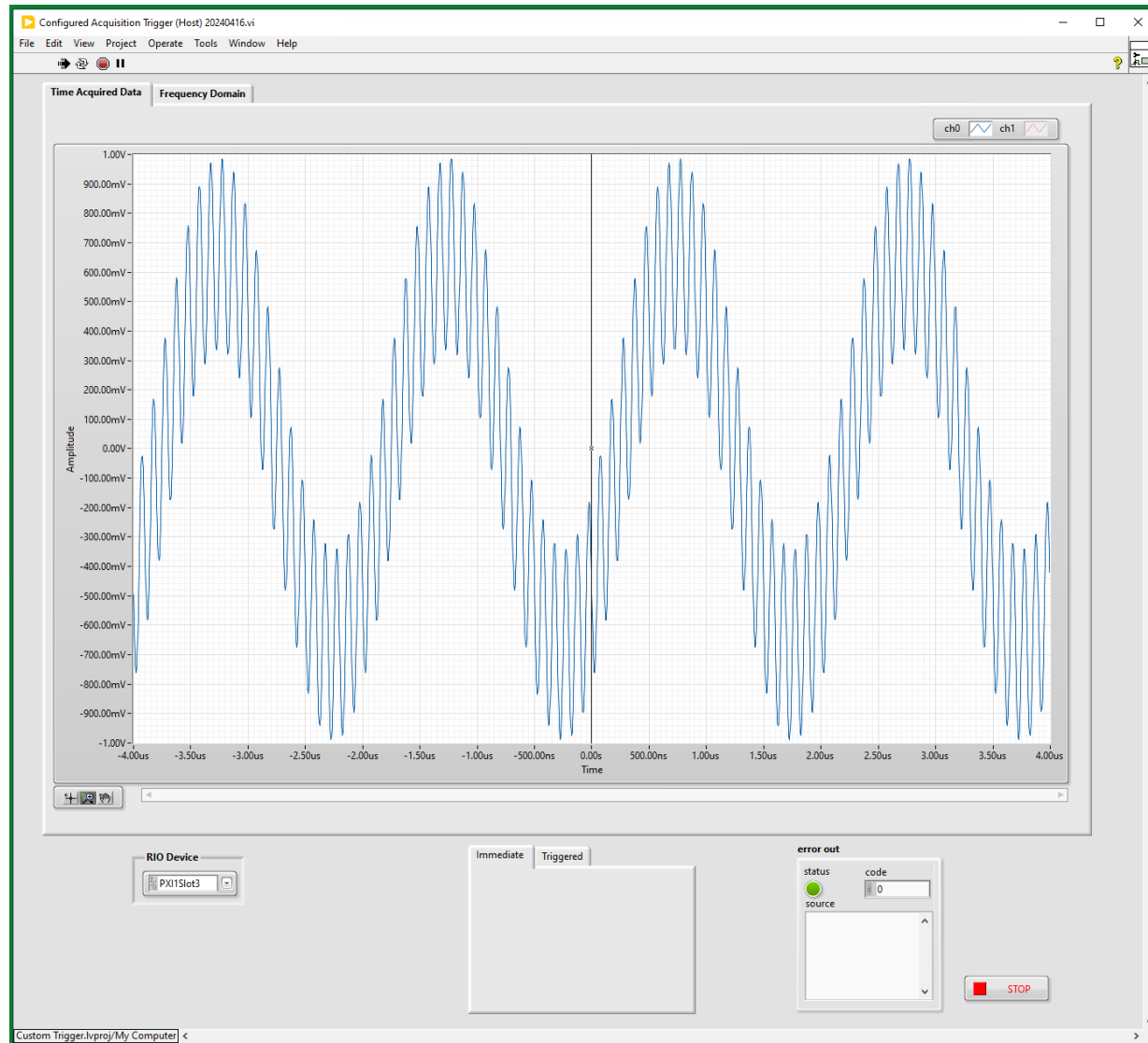
High Speed “Real Time” Data Analysis with LV FPGA

On Reconfigurable Scopes

Scenario

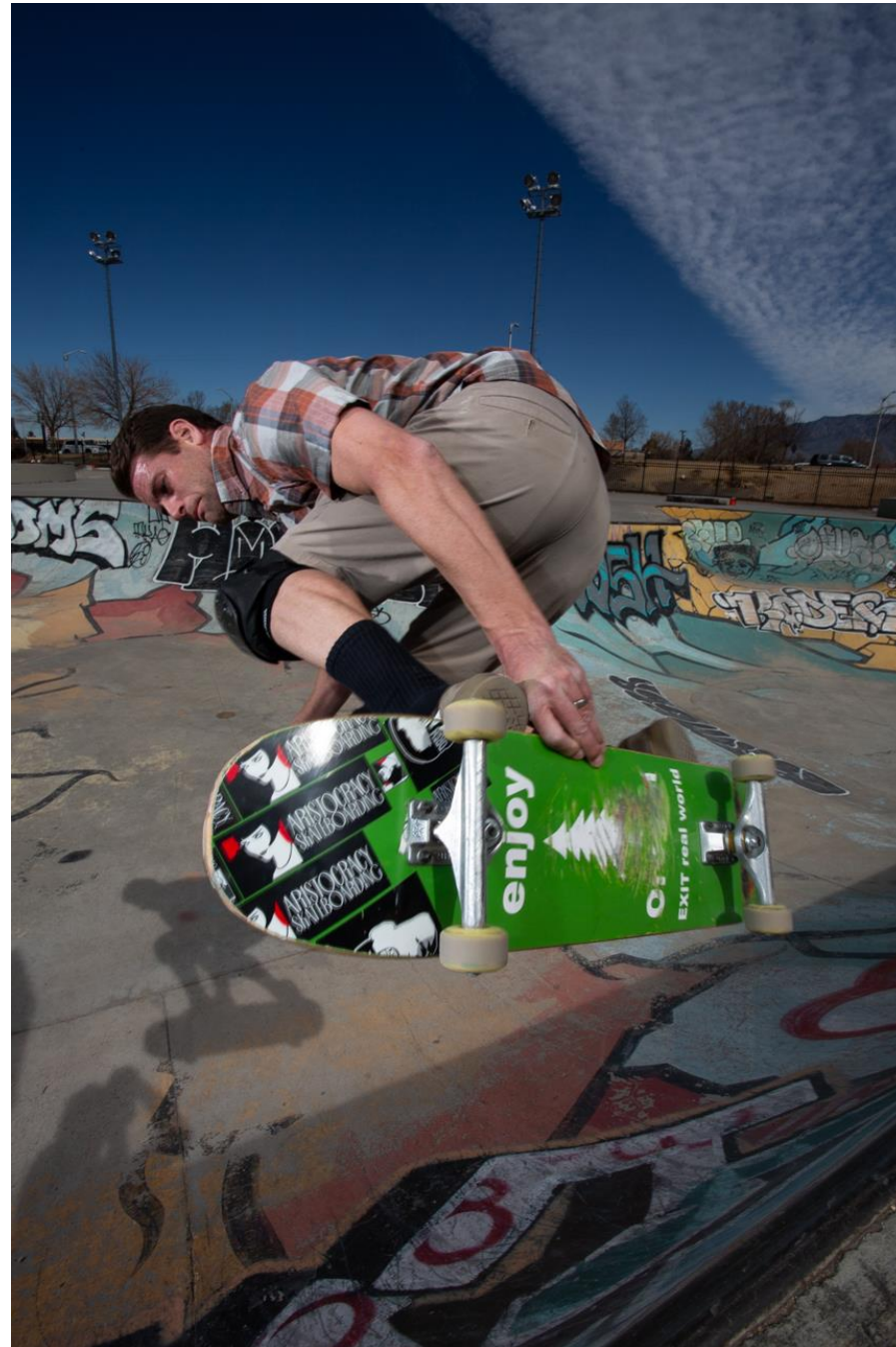
1. Normally, my device has a smooth sine wave
2. Occasionally, high frequency noise is mixed in
3. Need to capture the transition to diagnose

Demo



Luke Graham

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



My Objective

- **Expand your imagination**
- **Inform your questions**

- **Part 2 of 2**

Glossary

- **FPGA**
- **Latency**
- **Signal Processing (DSP)**
- **Real Time**

Glossary

- **FPGA** – “Programmable” Hardware
- **Latency** – How long it takes for something to respond. Today, how long it takes data to move through an algorithm.
- **Signal Processing** (DSP) – Examples include:
 - Apply filters: Low pass filter, notch filter
 - Decimate: Reduce data rate by 1:10
 - Transforms: Fourier transform to move to frequency domain
 - Data reduction: Acquire waveform but only keep amplitude and frequency.
- **Real Time** – In this context, “in line” or “before we’ve logged the data to disk”
 - (Nothing to do with RTOS)

What's a “Reconfigurable Scope”?
What's a “FlexRIO”?

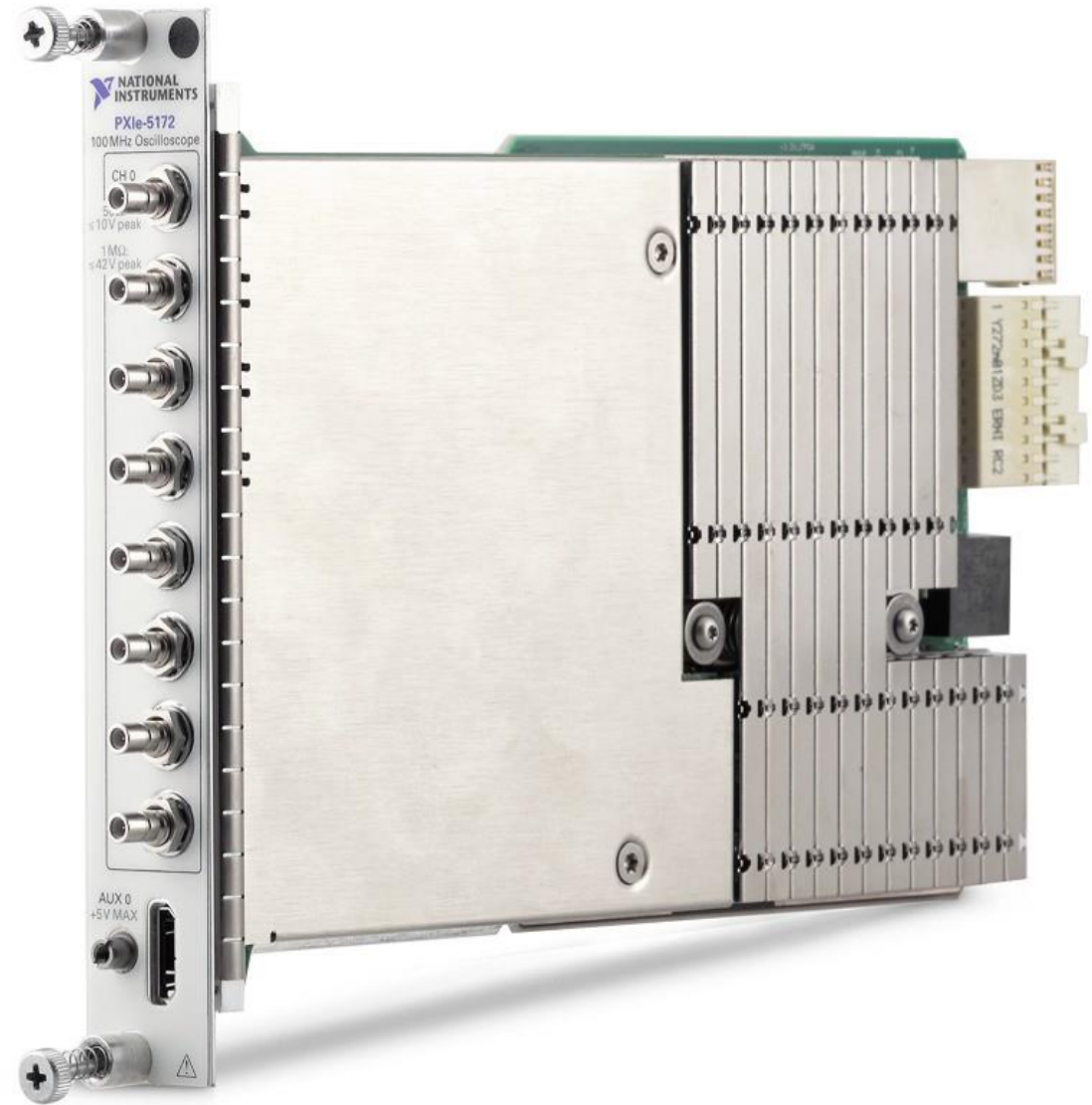
FlexRIO

- User-accessible FPGA
- Many versions available
 - Digitizers
 - Digital Only
 - Coprocessor



Reconfigurable Scope

- User-accessible FPGA
- NI-Scope Driver (no modification)
- Low-level access to NI-Scope functions
- NI-Scope examples & Instrument Studio
- “Instrument”



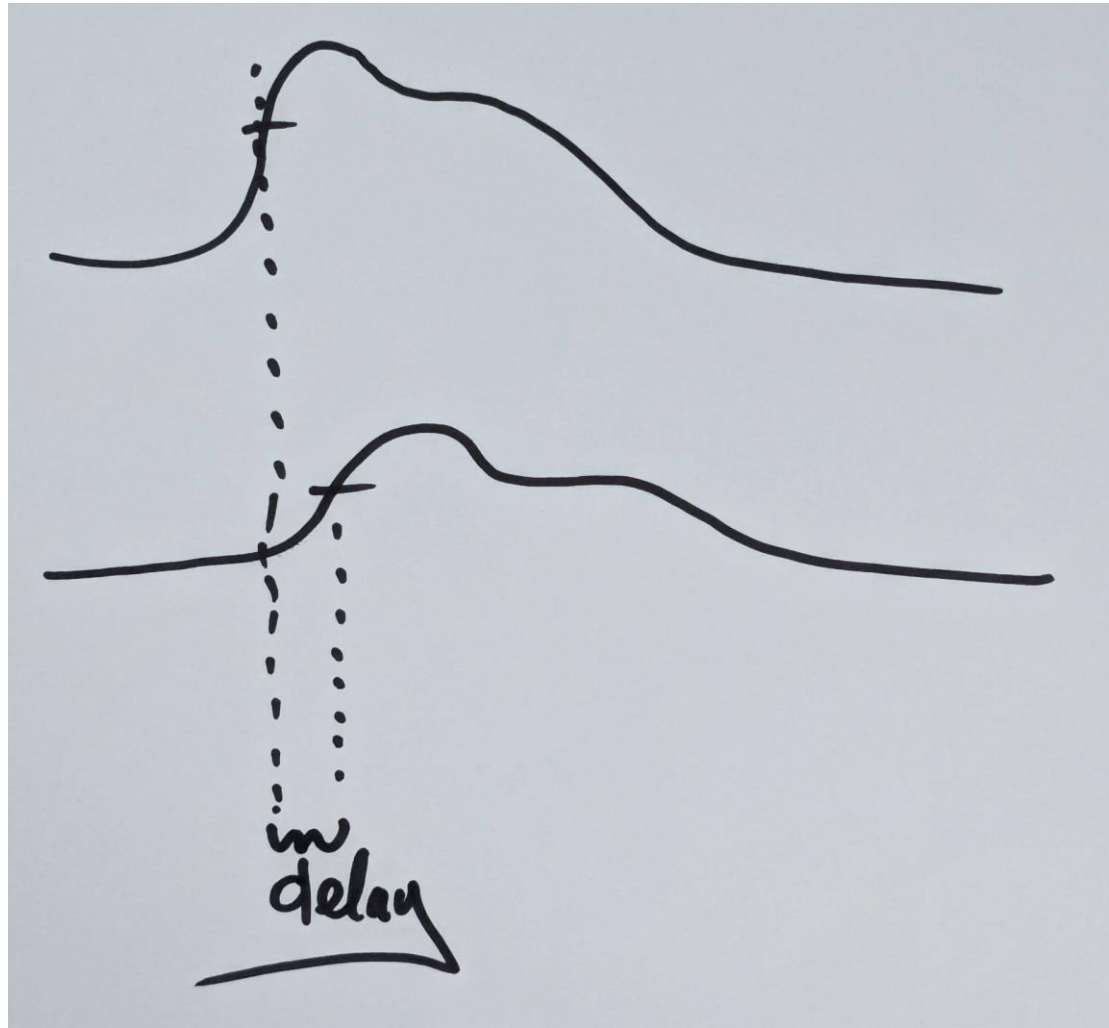
Why use a Reconfigurable Scope?

- Exotic trigger
- Truly independent triggers
- In-line DSP
- Massive Data Reduction
- Evolving project requirements

You **DON'T** need Reconfigurable Scopes if:

- You're using a fiducial
- The entire chassis triggers at the same time
- Someone else wants all the data to run through THEIR algorithm
- This will never change, even for diagnostics

Data Reduction



Data Reduction

- Measure the delay between two pulses on two channels.
- Pulses occur every 3 seconds.
- Delay between the two lines can be up to 2 seconds apart.
- We need to measure the delay with 10 nanosecond resolution.

Data Reduction

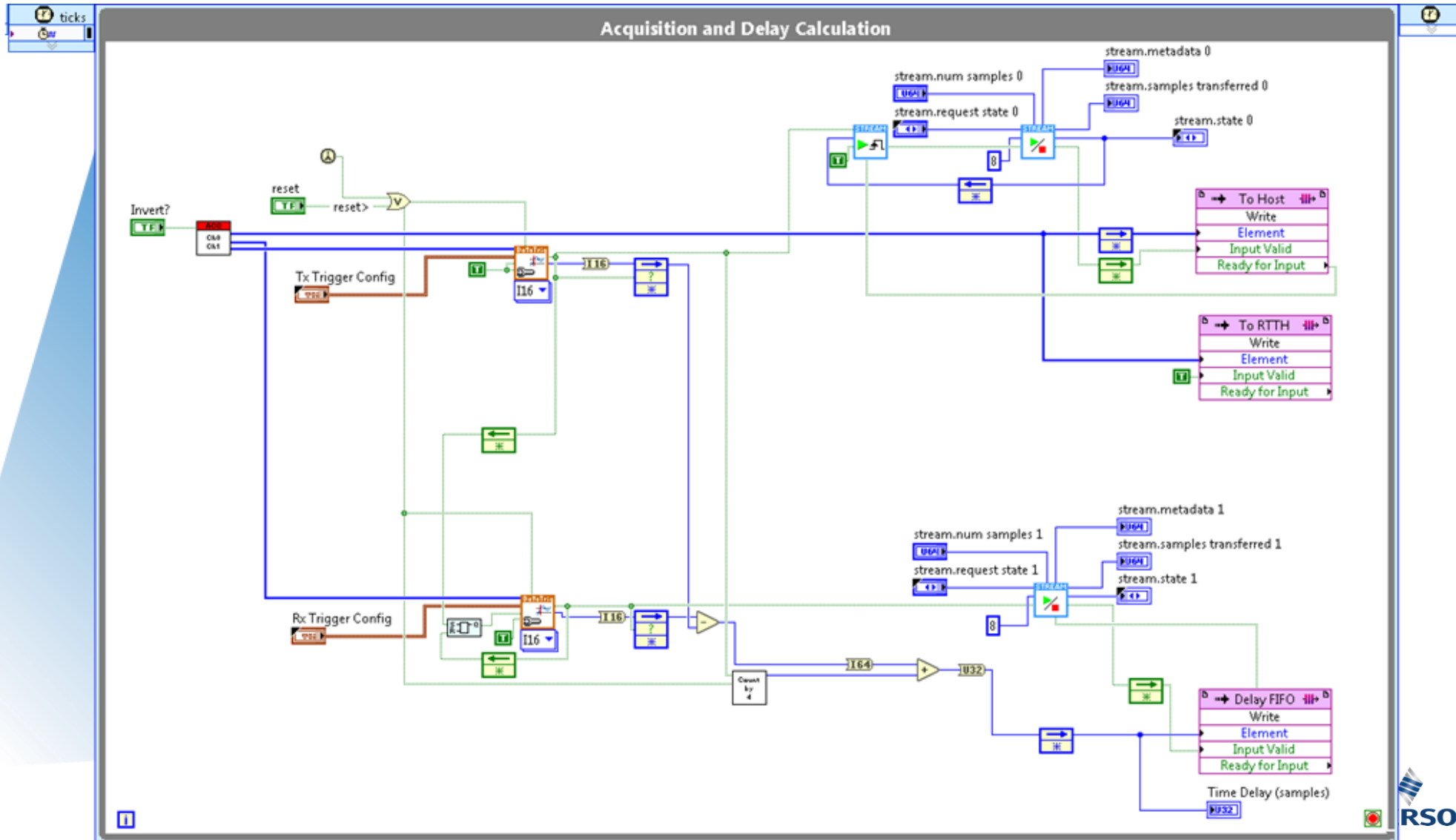
- Measure the delay between two pulses on two channels.
- Pulses occur every 3 seconds.
- Delay between the two lines can be up to 2 seconds apart.
- We need to measure the delay with 10 nanosecond resolution.

- How many samples per measurement?
- How much data do we care about?

Pseudo Code

- In FPGA: “Trigger” Channel 0 and Channel 1 independently.
- Note the clock count of each.
- $(\text{Channel 1 clock count}) - (\text{Channel 0 clock count}) = \text{pulse delay}$

Example Application – Data Reduction



Libraries

The screenshot displays the NI Package Manager application window. The title bar reads "NI Package Manager". The interface is divided into a left sidebar and a main content area. The sidebar, titled "BROWSE PRODUCTS", lists categories: Programming Environments, Application Software, Add-Ons, Drivers (highlighted), Utilities, Software Suites, Tools Network, and DataPlugins. The main content area shows the product "LabVIEW Instrument Design Libraries for Reconfigurable Oscilloscopes". At the top of this area, there are tabs for "INSTALLED 1154" and "UPDATES 33", and a search bar containing the text "reconfigurable". The product details include the NI logo, a version dropdown set to "21.5", bitness options "32-bit, 64-bit", and language "English". A green "INSTALL" button is prominently displayed. Below the product information are three tabs: "Overview" (selected), "Details", and "Support". The "Overview" tab contains a descriptive paragraph about the software's purpose and capabilities.

NI Package Manager

BROWSE PRODUCTS

INSTALLED 1154 | UPDATES 33

reconfigurable

LabVIEW Instrument Design Libraries for Reconfigurable Oscilloscopes

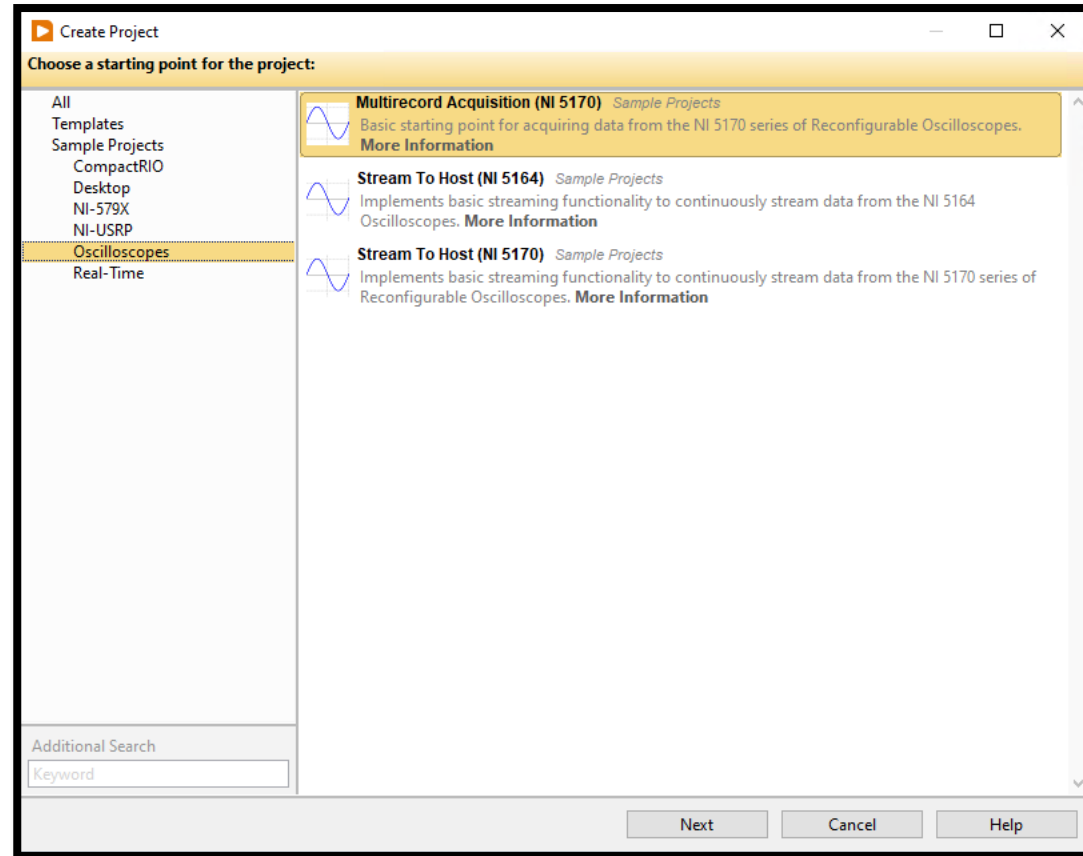
Version: 21.5 Bitness: 32-bit, 64-bit Language: English

INSTALL

Overview Details Support

LabVIEW Instrument Design Libraries for Reconfigurable Oscilloscopes is driver software for LabVIEW. You can use this software to create and develop oscilloscope applications that use programmable FPGAs. The software provides an interface between the user and vendor code on both the host and FPGA, with sample projects to use as-is or as starting points for creating application-specific instrument designs such as custom triggers or in-line FPGA processing. Each sample project included with LabVIEW Instrument Design Libraries for Reconfigurable Oscilloscopes contains host VIs which run on the host computer, and FPGA VIs which run on the device itself.

Create Project



Walk through project

- Host code overview
- FPGA code
- Host code again to look at triggering and FIFO

Modified SW

- Massive Data Reduction
- Custom trigger
 - Frequency domain trigger
- Independent triggering channels

Challenge

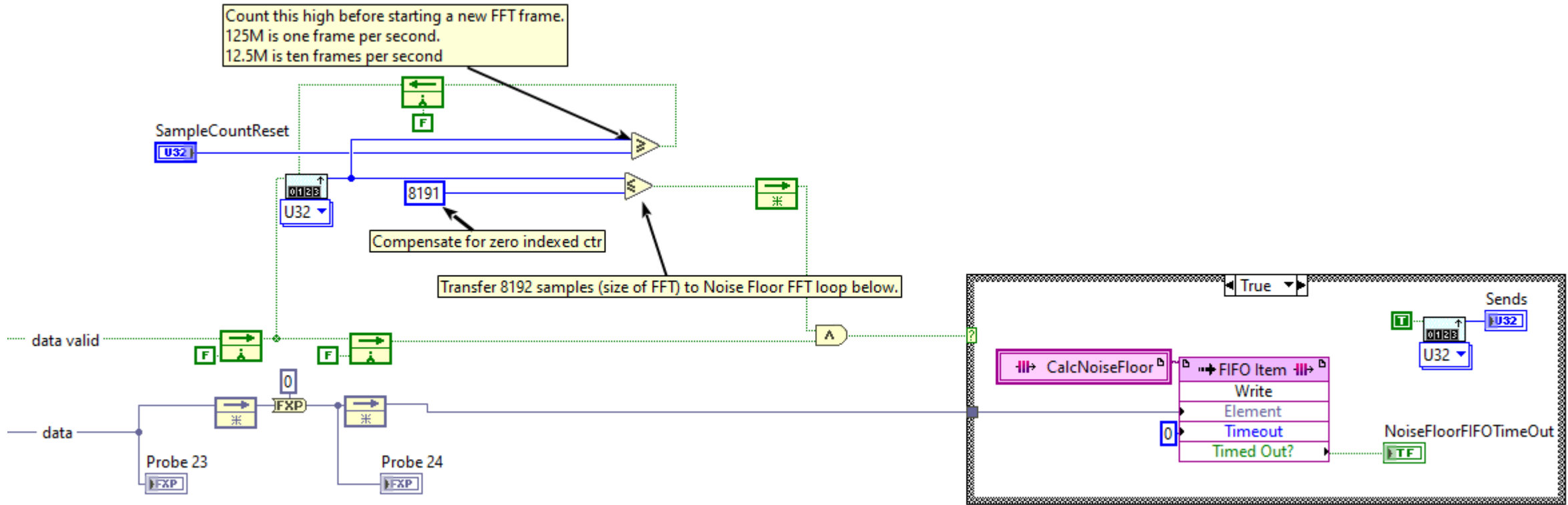
- User wants to create a running average of FFT traces over the course of a day.
- They want to see if an unexpected transmitter shows up in their frequency band of interest.
- We want one FFT trace of the full digitizer bandwidth “every so often”. It’s user configurable, but let’s say once every second.
- Streaming time-domain data to the processor all the time is going to be very intensive.

Challenge

- User wants to create a running average of FFT traces over the course of a day.
- They want to see if an unexpected transmitter shows up in their frequency band of interest.
- We want one FFT trace of the full digitizer bandwidth “every so often”. It’s user configurable, but let’s say once every second.
- Streaming time-domain data to the processor all the time is going to be very intensive.
- Point-by-point acquisition, but FFT is bin-by-bin. (8,192 bins)

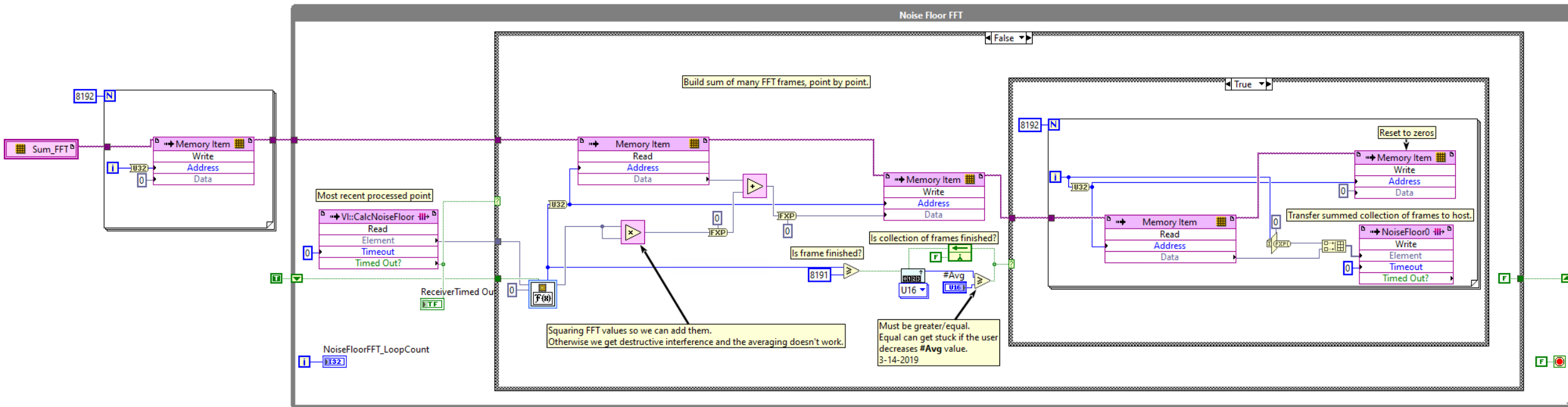
Massive Data Reduction

- Capture one FFT “frame” per second/minute/hour



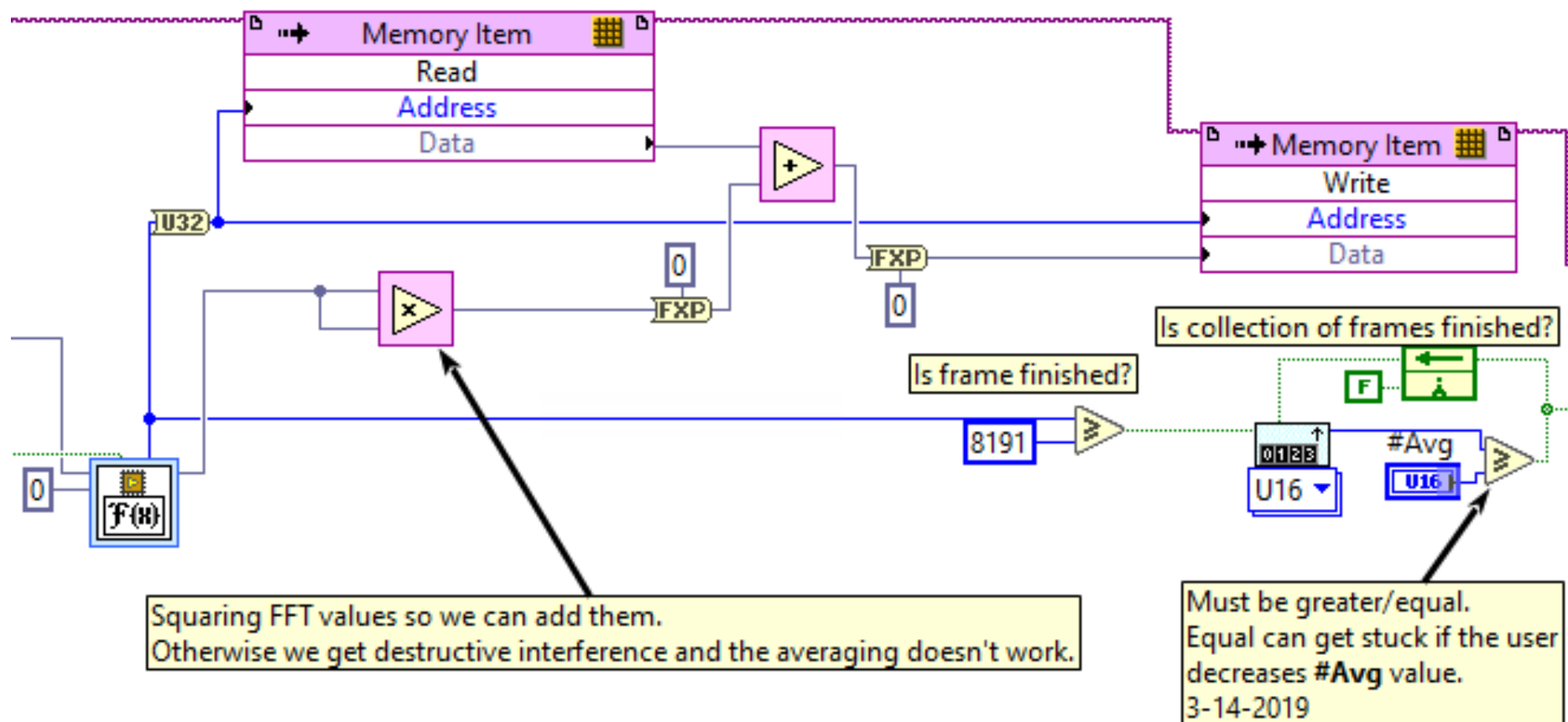
Massive Data Reduction

- Sum FFT frames and transfer to host

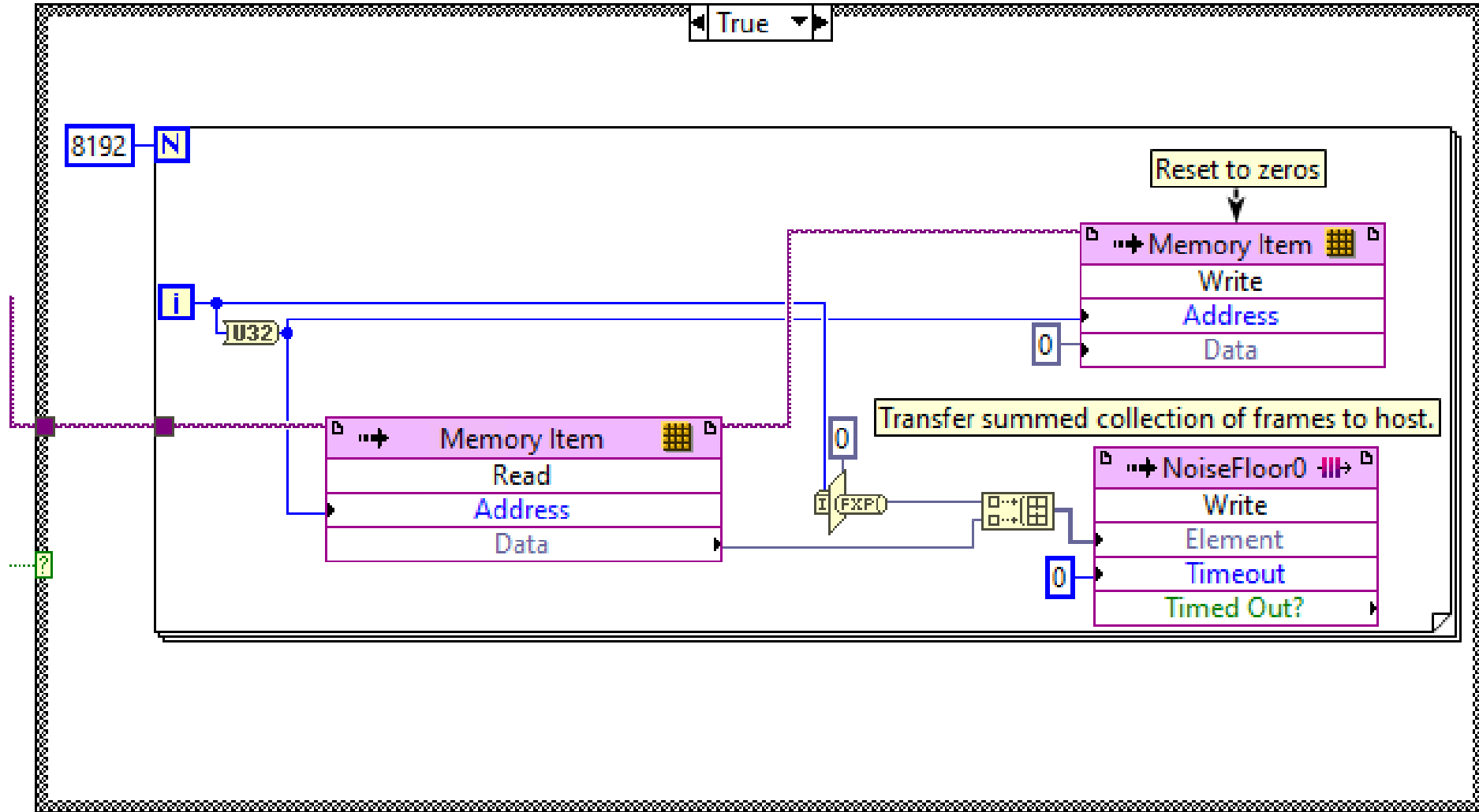


Massive Data Reduction

Build sum of many FFT frames, point by point.



Massive Data Reduction



FlexRIO Example

- Walk through project