Oslo - Stockholm - Utrecht - Brussels - Copenhagen - Helsinki

ni.com/nidays

NATIONAL INSTRUMENTS

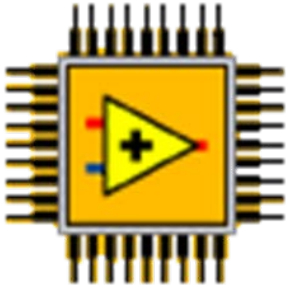
ADVANCED DATA ACQUISITION TECHNIQUES WITH INTELLIGENT DAQ

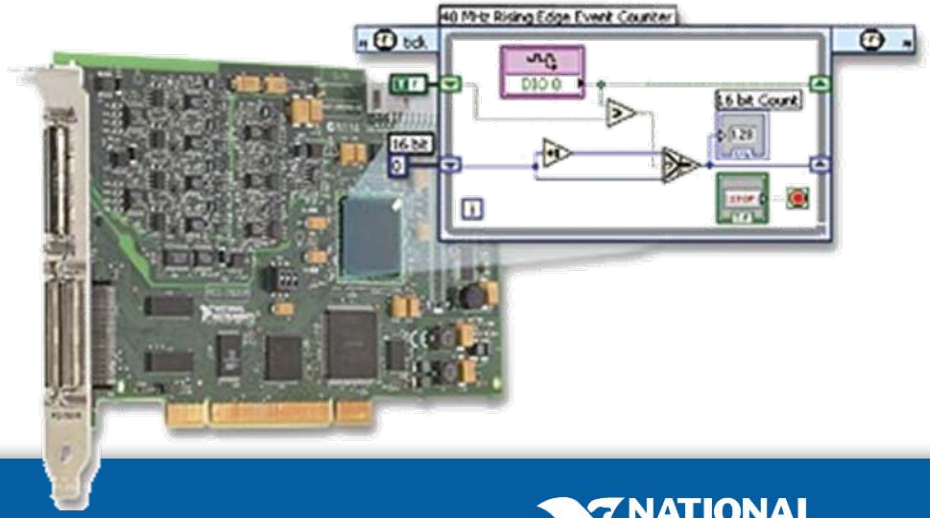
Agenda

- **Intro to Intelligent DAQ**
- LabVIEW FPGA for Test
- Implementing Custom Timing and Triggering
- High-Speed DMA Streaming with LabVIEW FPGA
- Building a Custom DAQ Personality

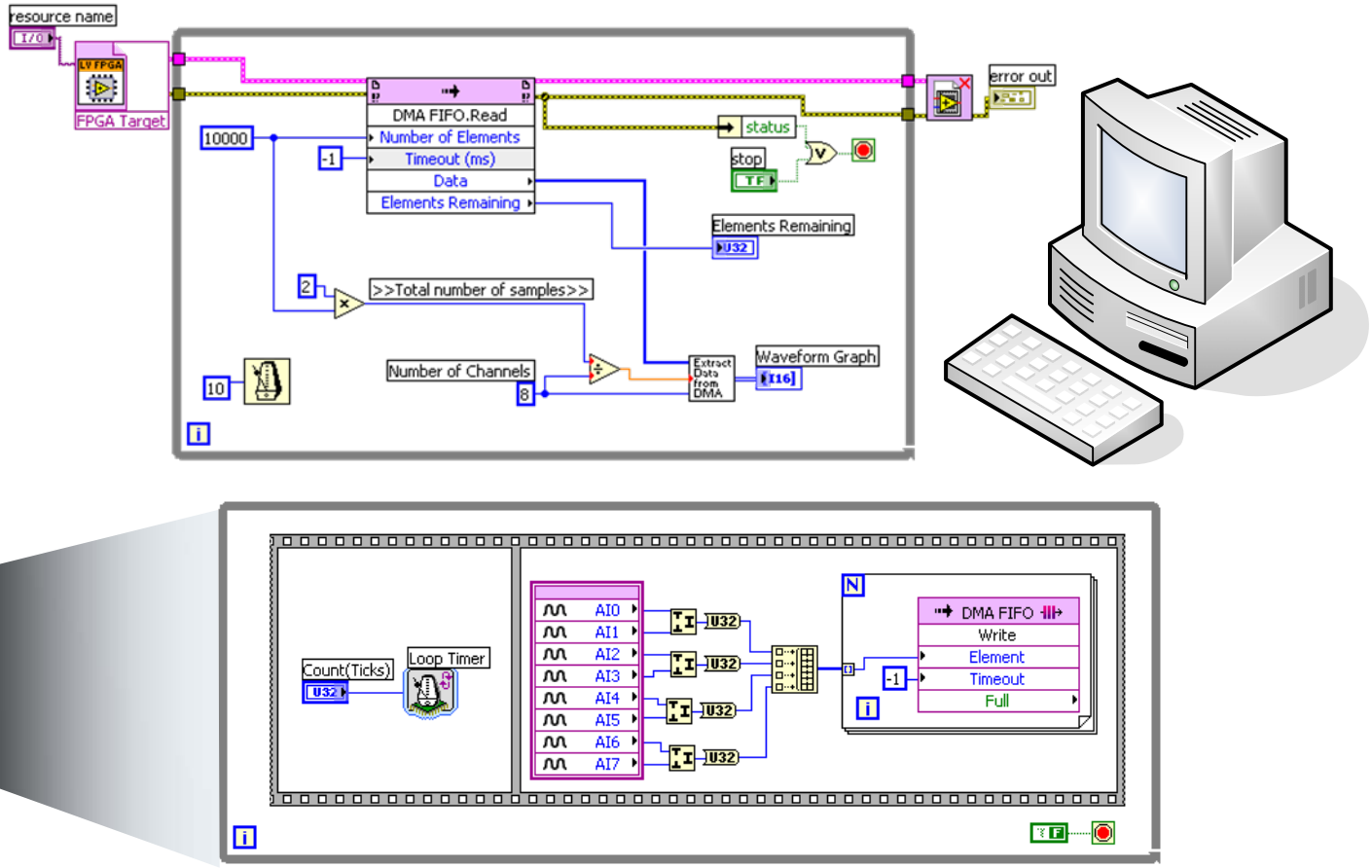
What Is Intelligent DAQ?

- FPGA-based I/O timing
- User-defined onboard processing
- Hardware-timed speed and reliability





LabVIEW FPGA Module



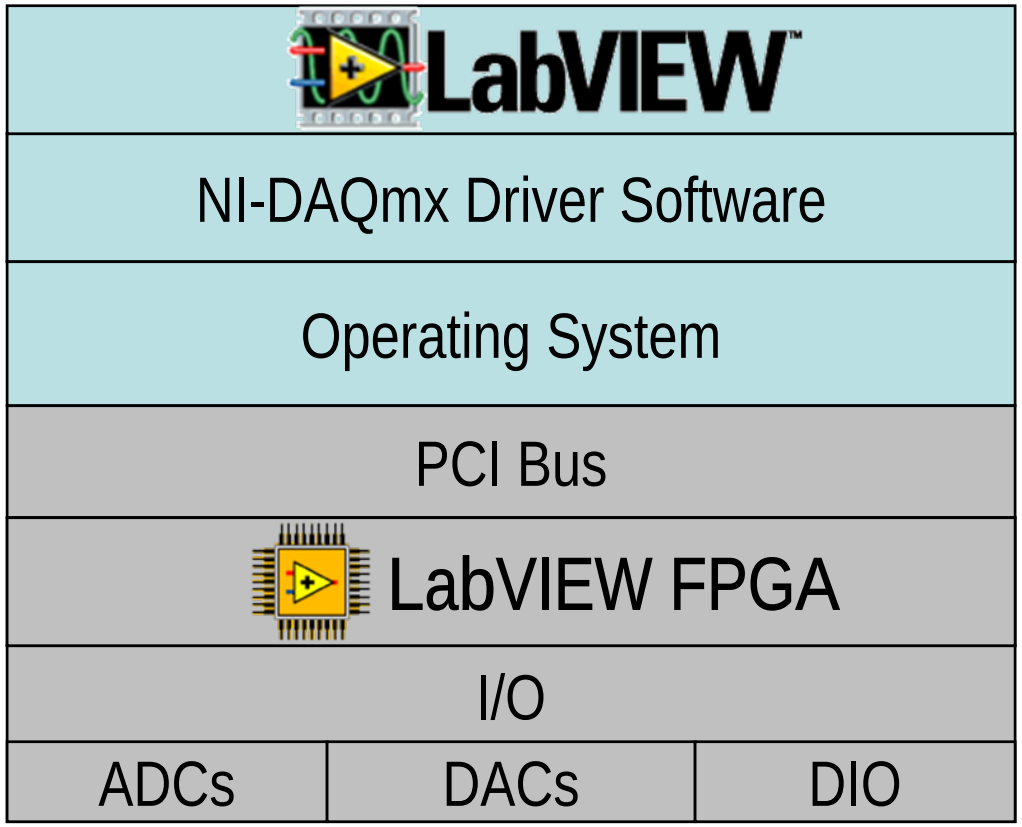
Typical MIO DAQ Intelligent DAQ

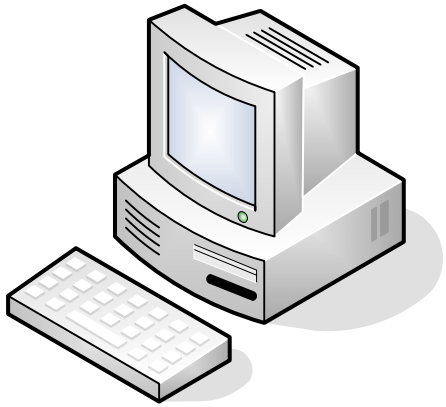
I/O Control	Fixed ASIC for timing and triggering	⟷	Open FPGA for timing and triggering
Analog I/O	Multiplexed, shared sample clock	⟷	Simultaneous or independent operation
Digital I/O	Up to 32 lines, correlated DIO	⟷	Up to 160 lines, hardware timed
Onboard Counters	2 general-purpose counters	⟷	Custom counters on any digital line

Layers of Abstraction

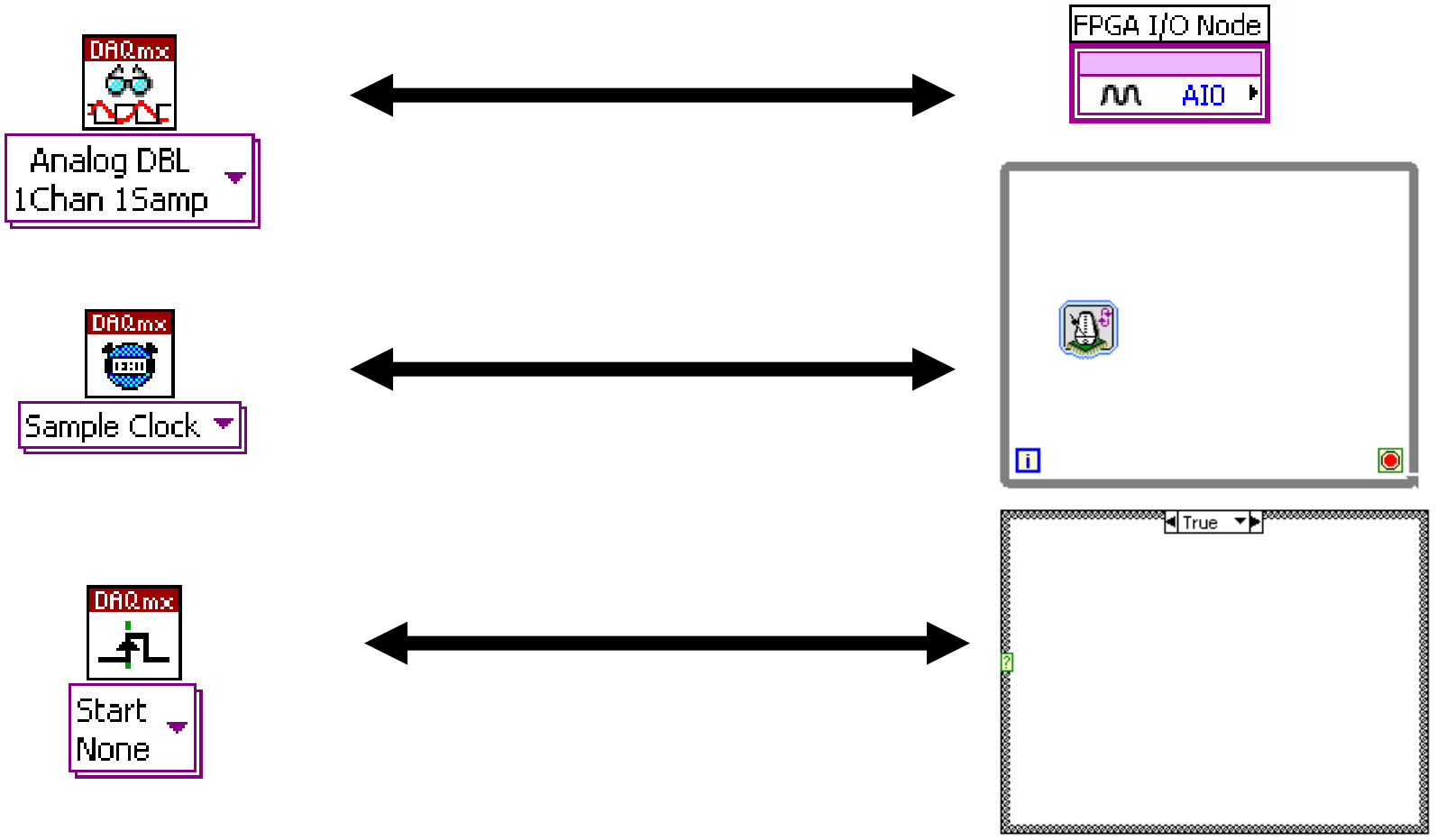
Software

Hardware





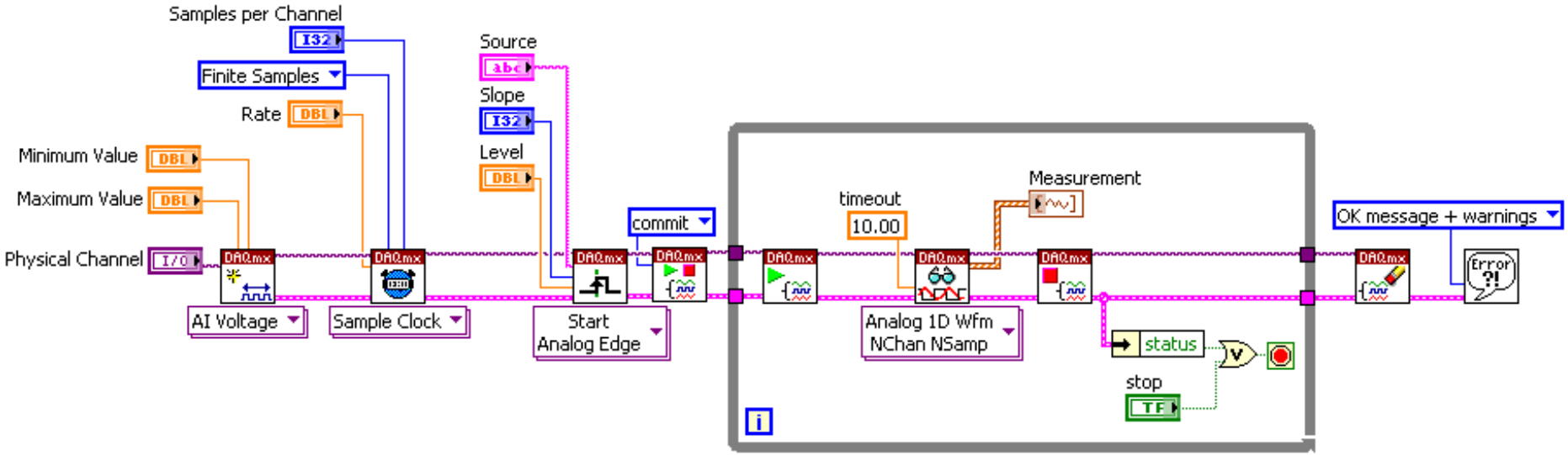
Intelligent DAQ Programming



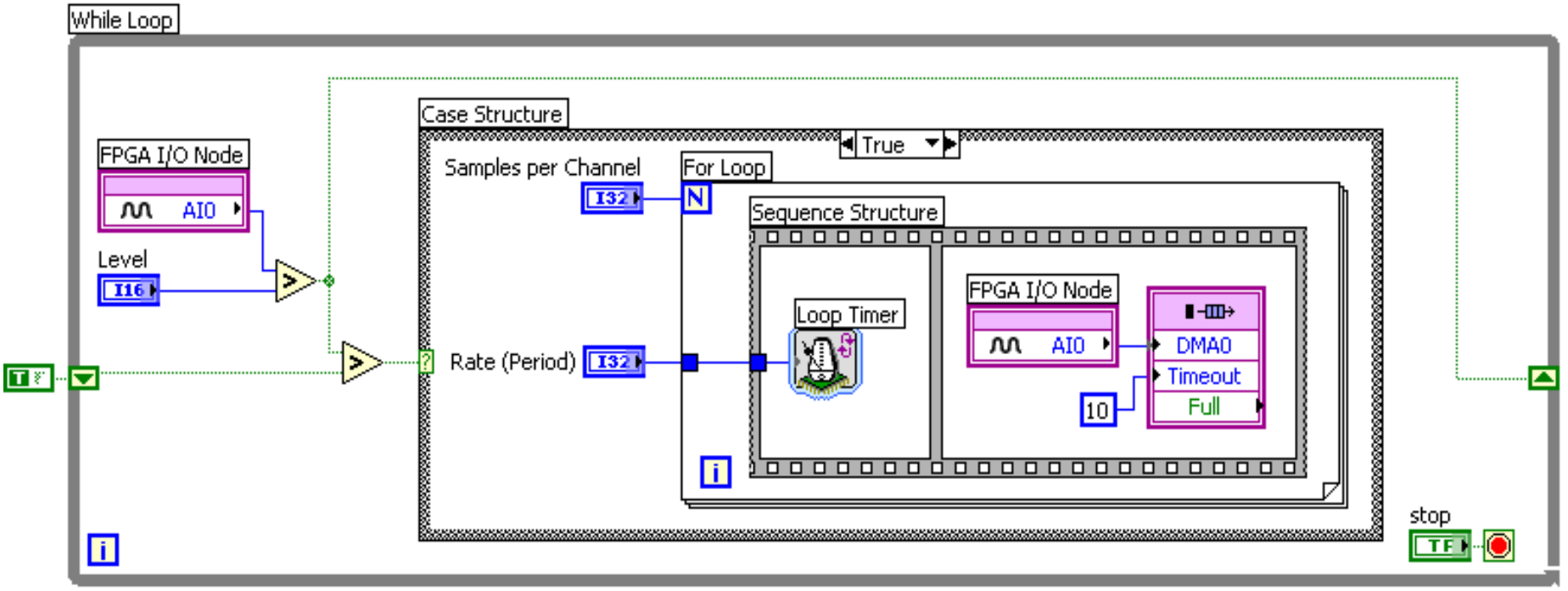
Agenda

- Intro to Intelligent DAQ
- LabVIEW FPGA for Test
- **Implementing Custom Timing and Triggering**
- High-Speed DMA Streaming with LabVIEW FPGA
- Building a Custom DAQ Personality

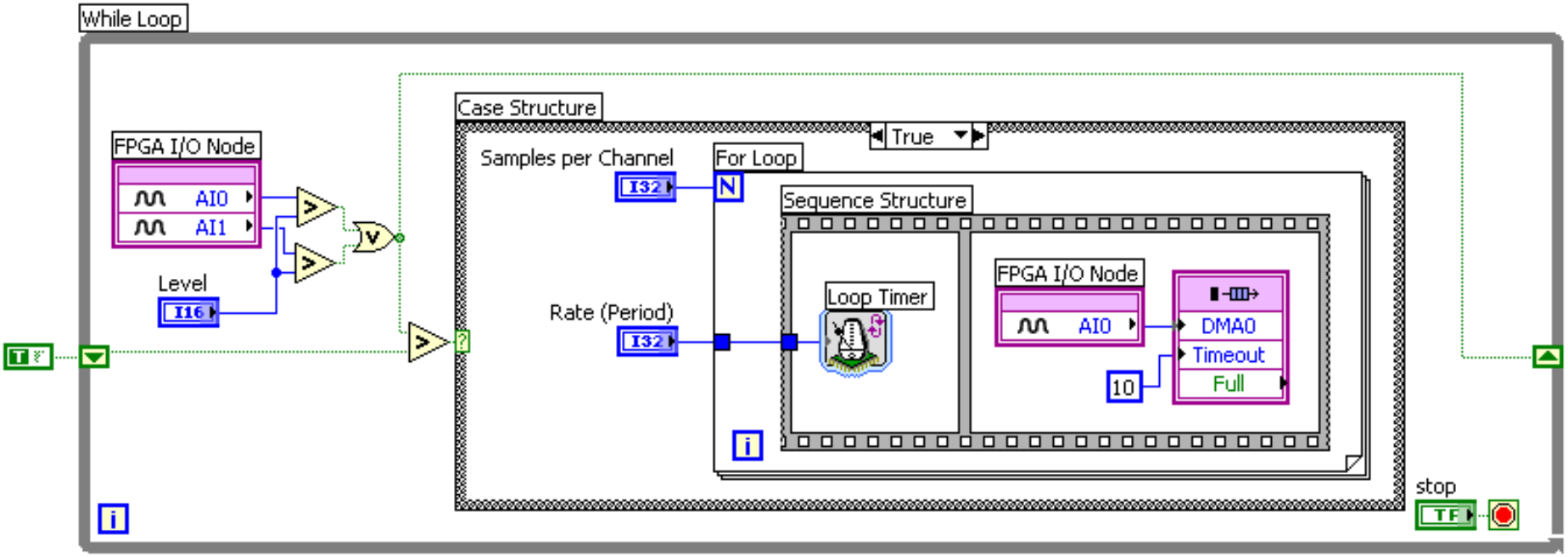
Triggered Analog Input
with NI-DAQmx



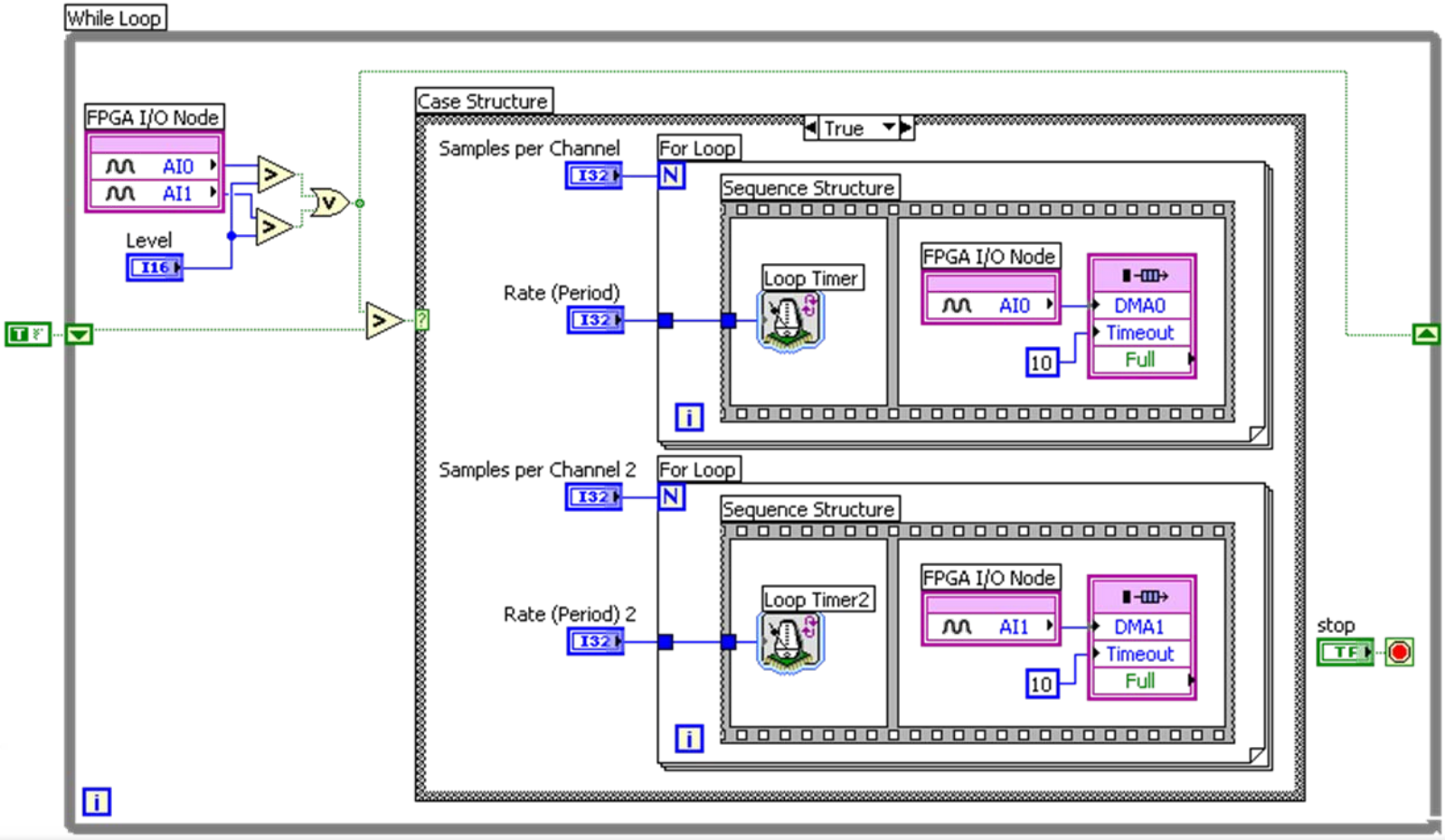
Triggered Analog Input
with Intelligent DAQ



Custom Triggered Analog Input
with Intelligent DAQ

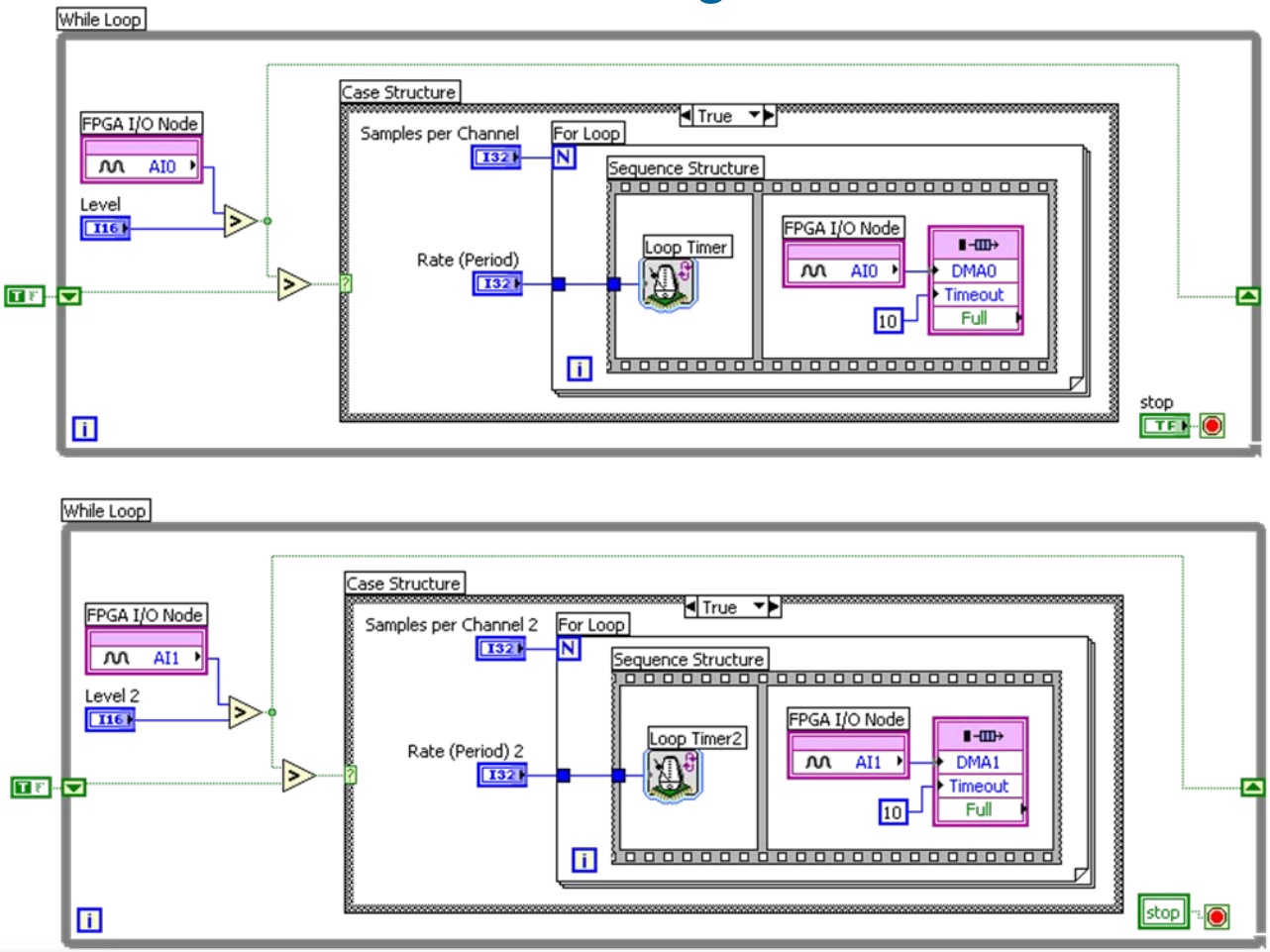


Triggered Multirate Analog Input
with Intelligent DAQ



NATIONAL INSTRUMENTS

Triggered Multirate Analog Input
with Intelligent DAQ



Agenda

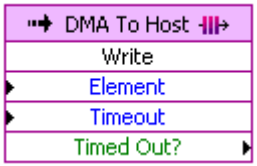
- Intro to Intelligent DAQ
- LabVIEW FPGA for Test
- Implementing Custom Timing and Triggering
- **High-Speed DMA Streaming with LabVIEW FPGA**
- Building a Custom DAQ Personality

DMA Theory of Operation

- A distributed FIFO between the FPGA and host
  - FIFO is lossless
  - FIFO can be full or empty
- FIFO composed of two pieces of memory
  - Block memory on the FPGA
  - RAM on the host
- Memories are connected by an engine
  - Engine can be started and stopped
  - Data is immediately moved when space is available

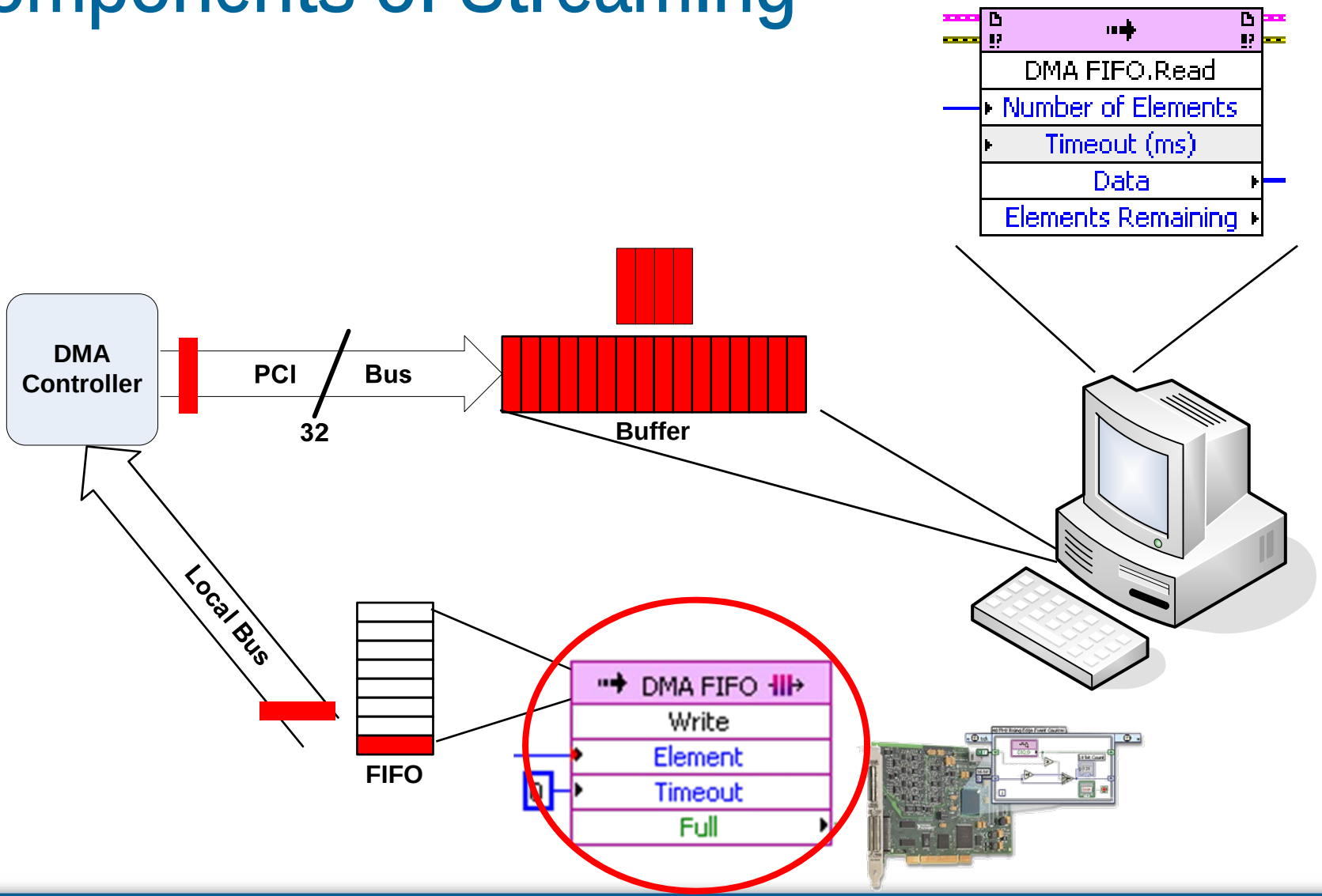
DMA API on the FPGA

- Timeout input
  - -1: Wait forever
  - 0: Succeed or fail immediately
  - Positive value: Time to wait in ticks
  - Typically use 0 or -1
    - 0 must be used inside of a single cycle loop

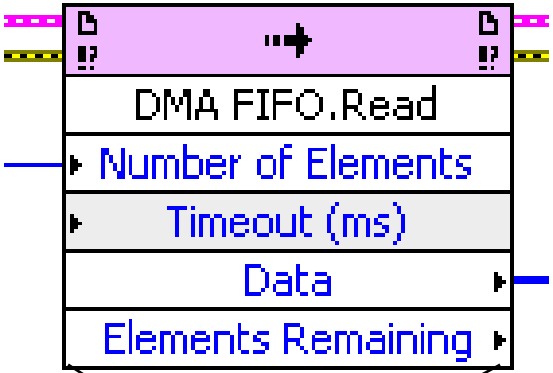


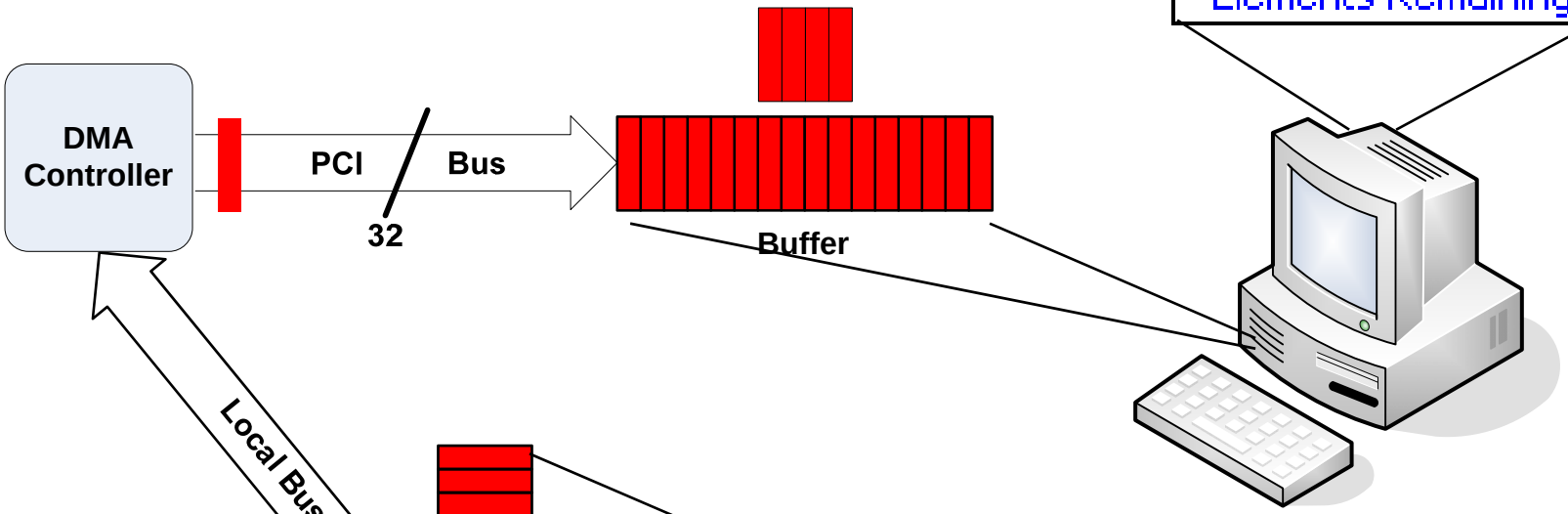
- Timeout output
  - Must be handled unless timeout is -1
  - For timeouts other than -1, some options would be to retry or cause a fatal error
    - Ignoring it is usually never the right option

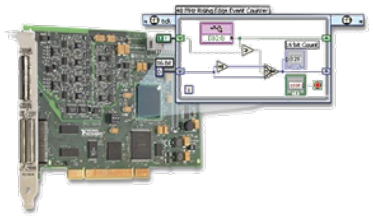
Components of Streaming



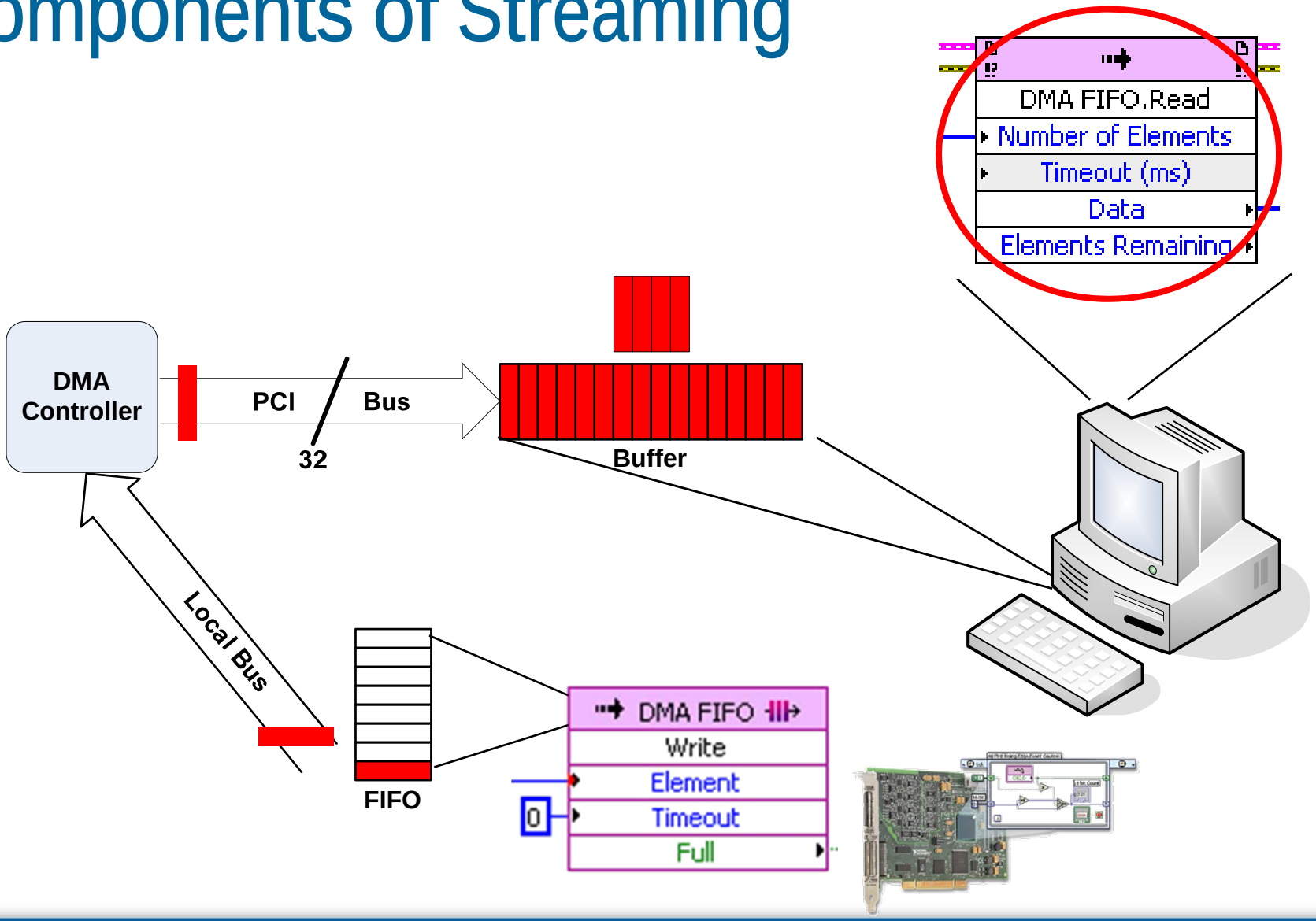
Components of Streaming





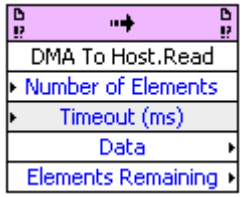


Components of Streaming

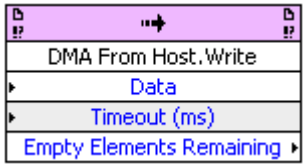


DMA Read and Write on the Host

- Number of elements input (Read)
  - Host can access multiple elements in the FIFO in a single operation



- Timeout input
  - Legal values are -1, 0, positive integer
  - Positive integer is time to wait in milliseconds
  - Timeout is always a fatal error on the host



- Elements remaining/empty elements remaining output
  - A way for the application to monitor progress

Agenda

- Intro to Intelligent DAQ
- LabVIEW FPGA for Test
- Implementing Custom Timing and Triggering
- High-Speed DMA Streaming with LabVIEW FPGA
- **Building a Custom DAQ Personality**

Building a DAQ Personality

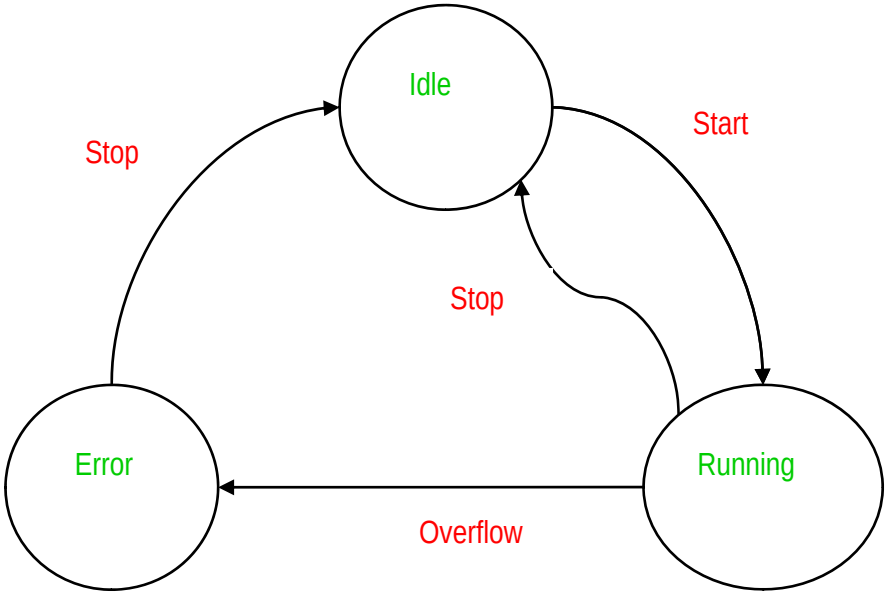
- Step 1: Define personality requirements
- Step 2: Design your state machine
  - The state machine pattern models hardware very well
- Step 3: Develop your state machine on the host
  - Goal is to debug on the host and minimize compiles

Building a DAQ Personality: Step 1

- Example personality requirements:
  - Two simultaneous analog input channels
  - Custom trigger – trigger on either of two analog values
  - Configurable sample rate
  - Continuous acquisition

Building a DAQ Personality: Step 2

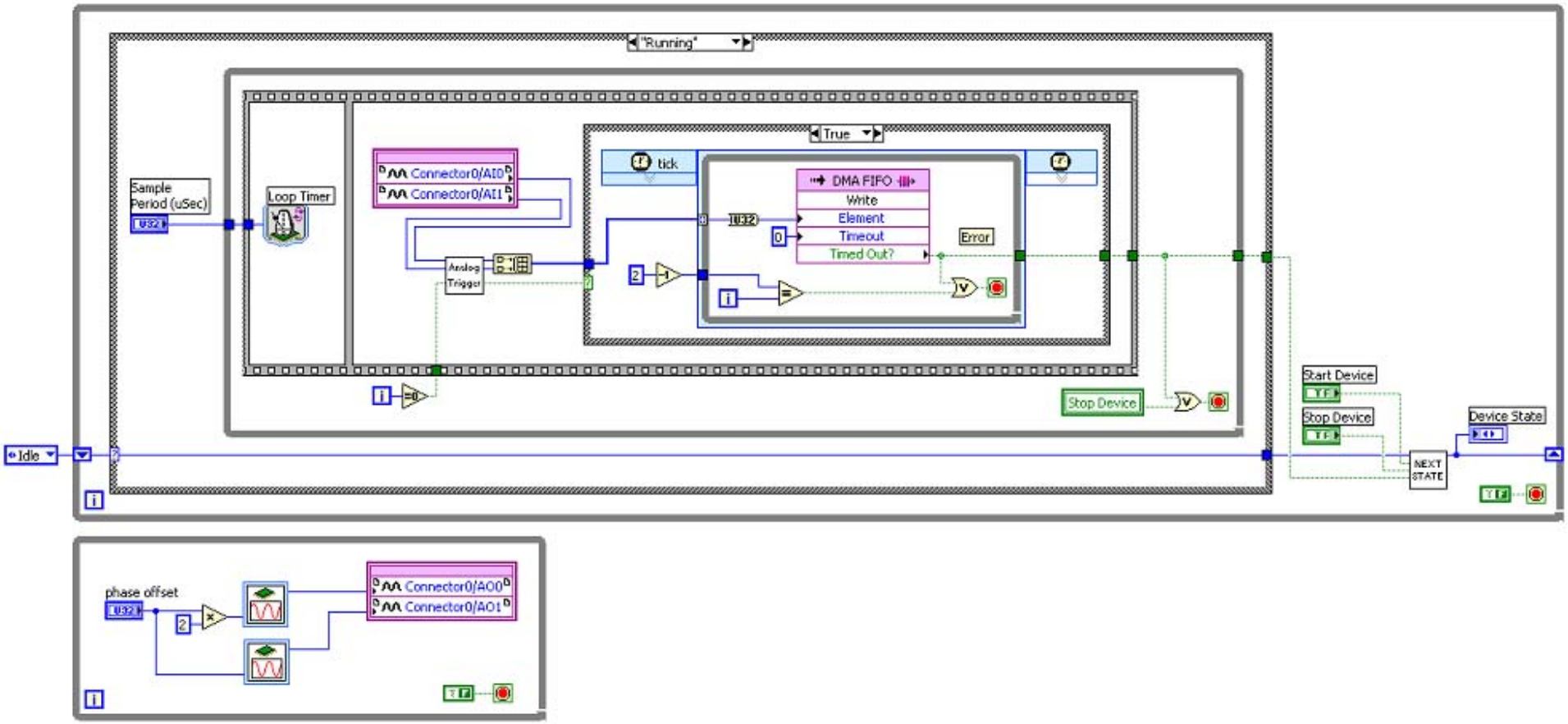
- Design your state machine
  - The host will write commands to the device and read the state of the device



Building a DAQ Personality: Step 3

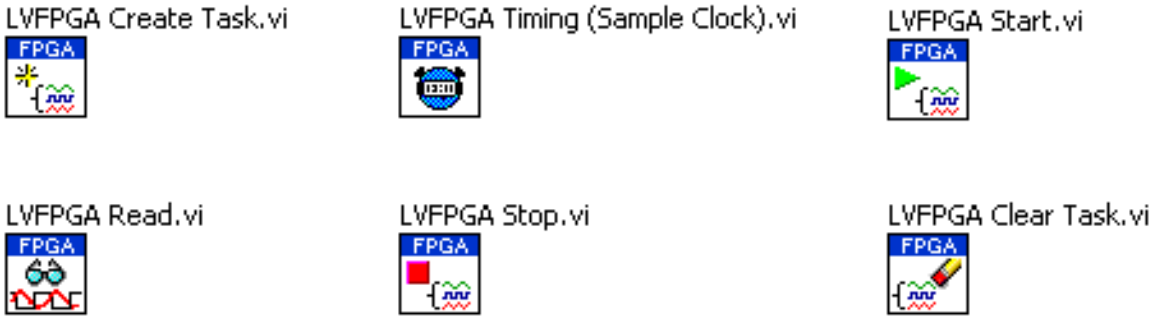
- Develop your state machine rules on the host
- Use simulation to verify logic and state transitions
- Implement each state in your state machine
- Compile

Screenshot of FPGA Block Diagram

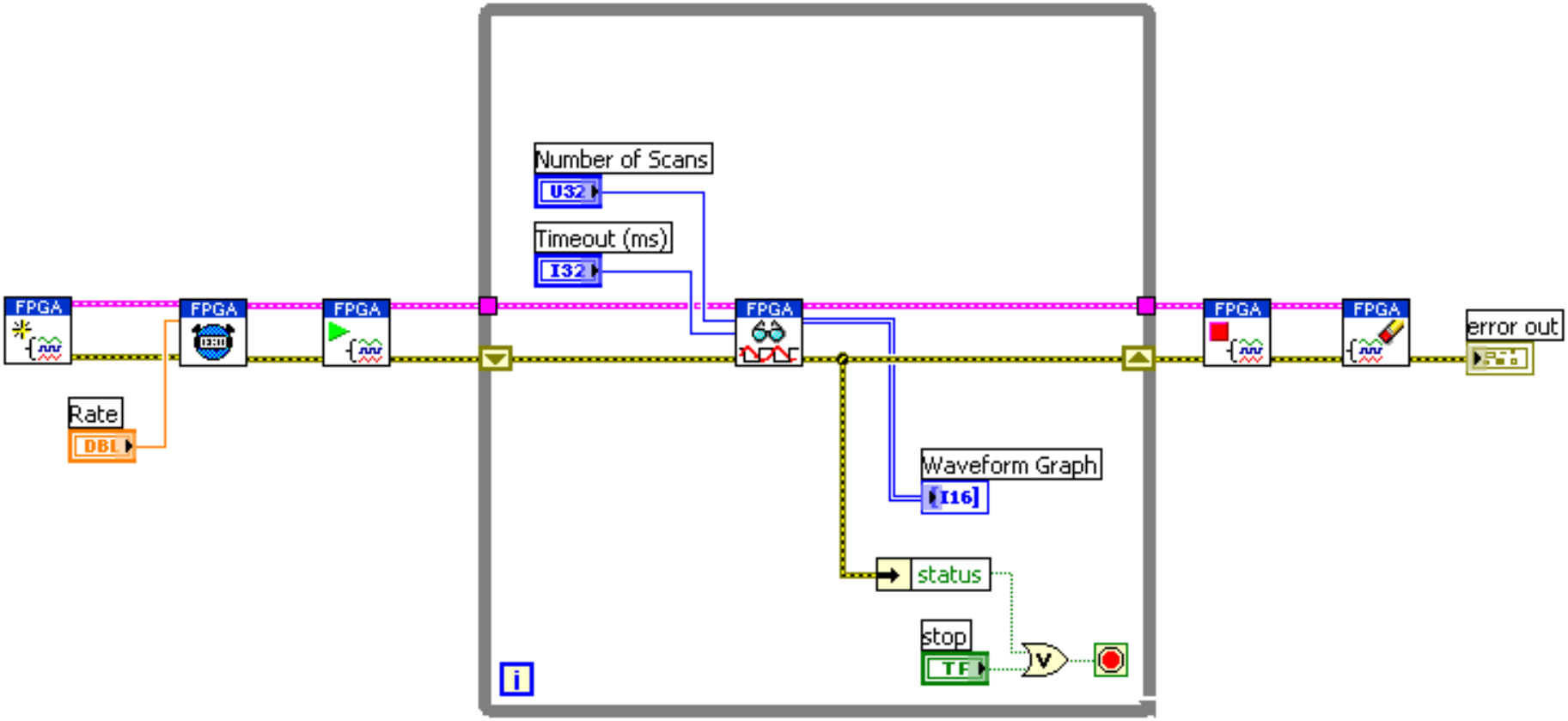


Creating the Host Application

- Design your API
  - We've chosen to use a subset of the NI-DAQmx API



Host Application



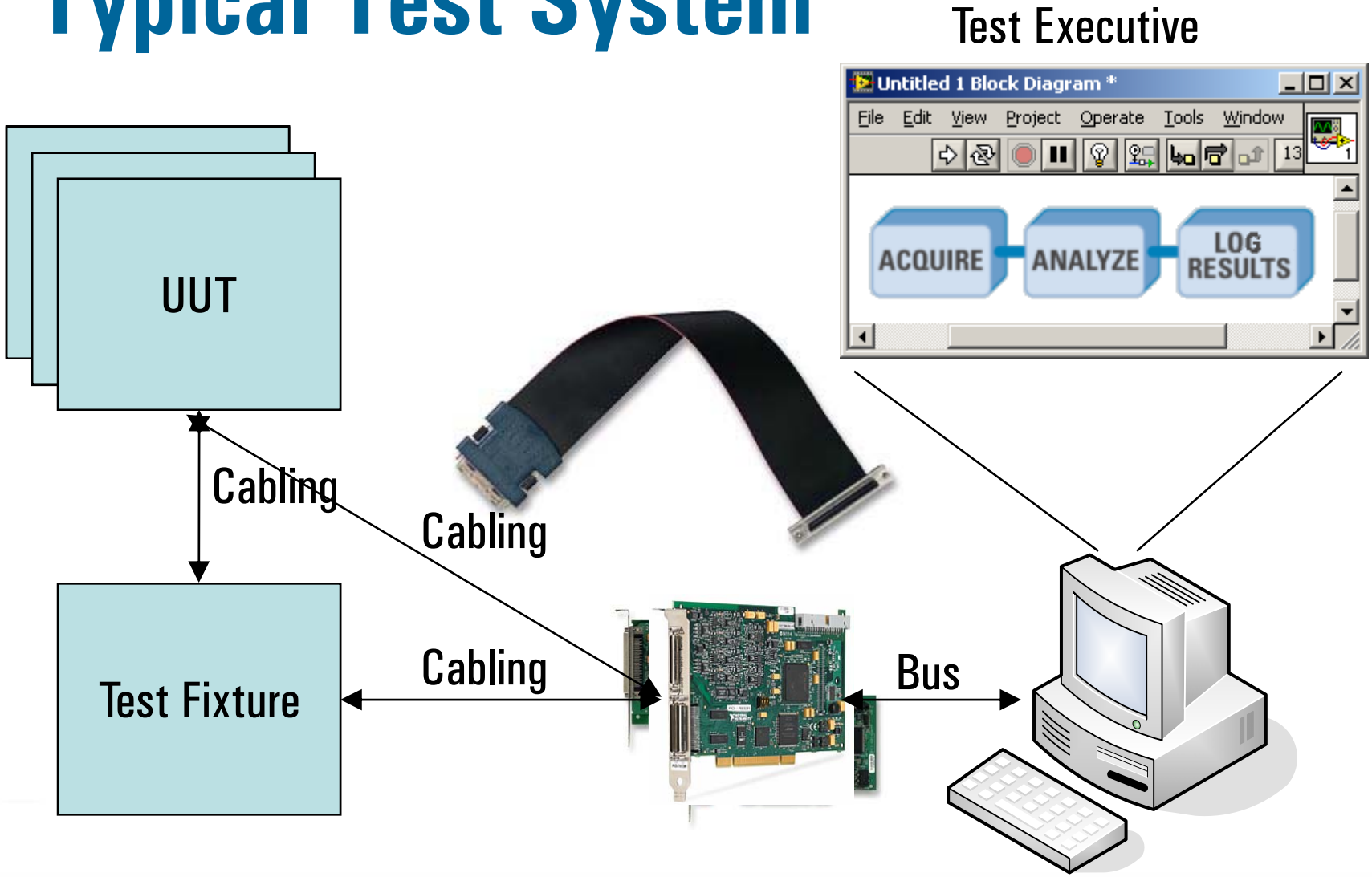
Summary

- Intelligent DAQ operates at register level with IO nodes
- LabVIEW FPGA allows customized timing and triggering as well as DMA operations
- Three steps in building the **D**AQ personality: **D**efine , **D**esign, **D**evelop

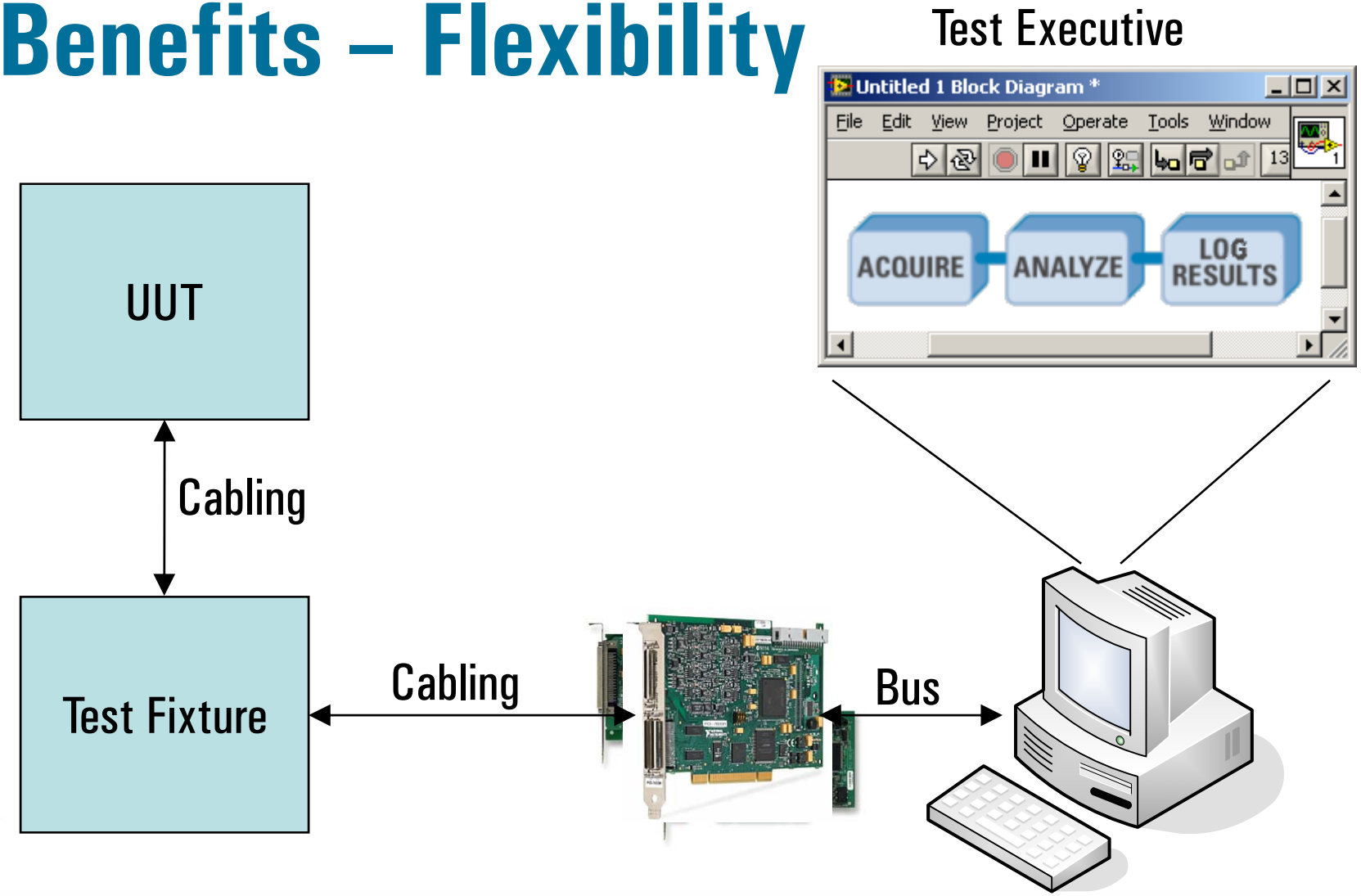
Beginning of Slide Appendix

The Benefits of FPGAs for Test

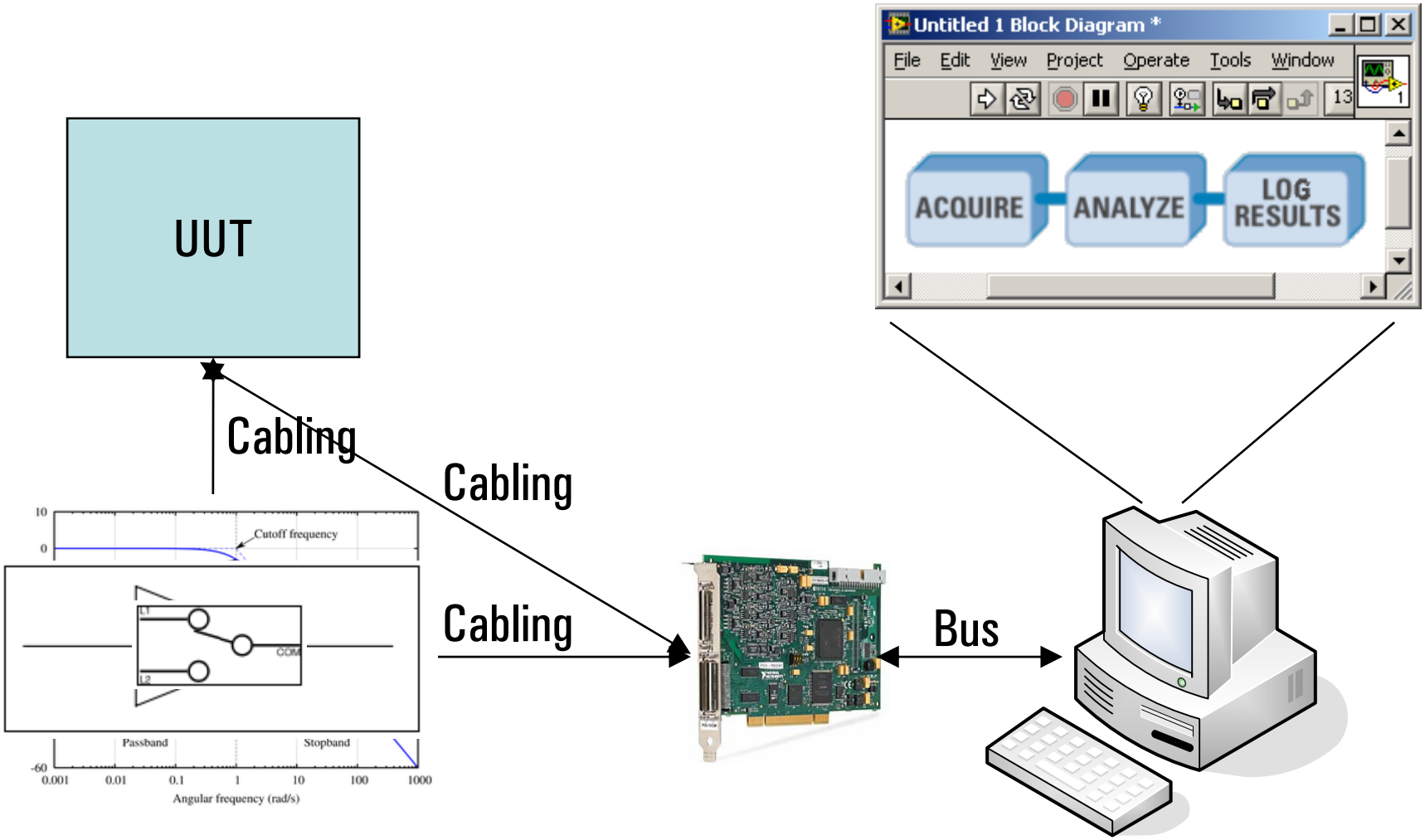
Typical Test System



Benefits – Flexibility

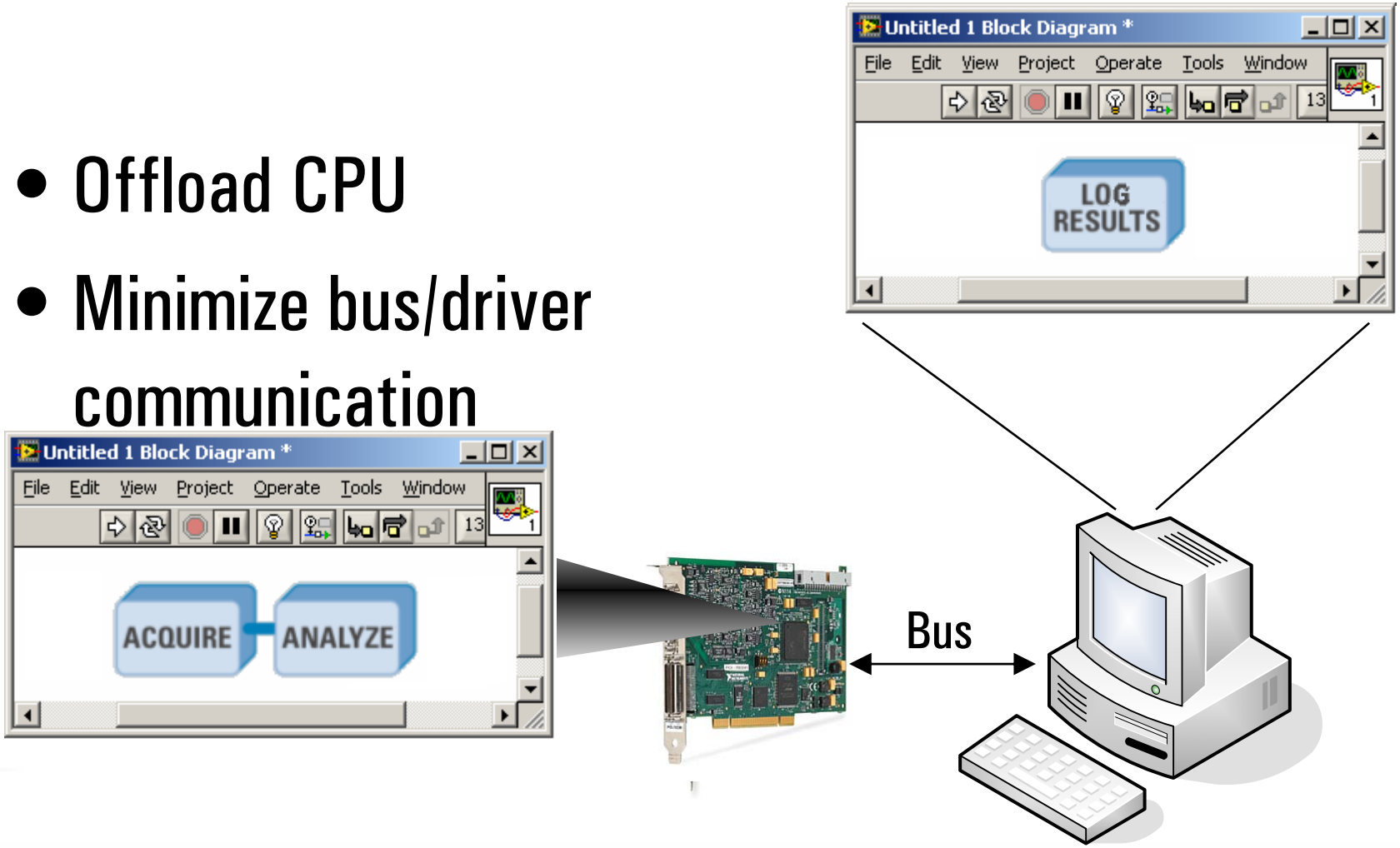


Benefits – Simplified Fixturing

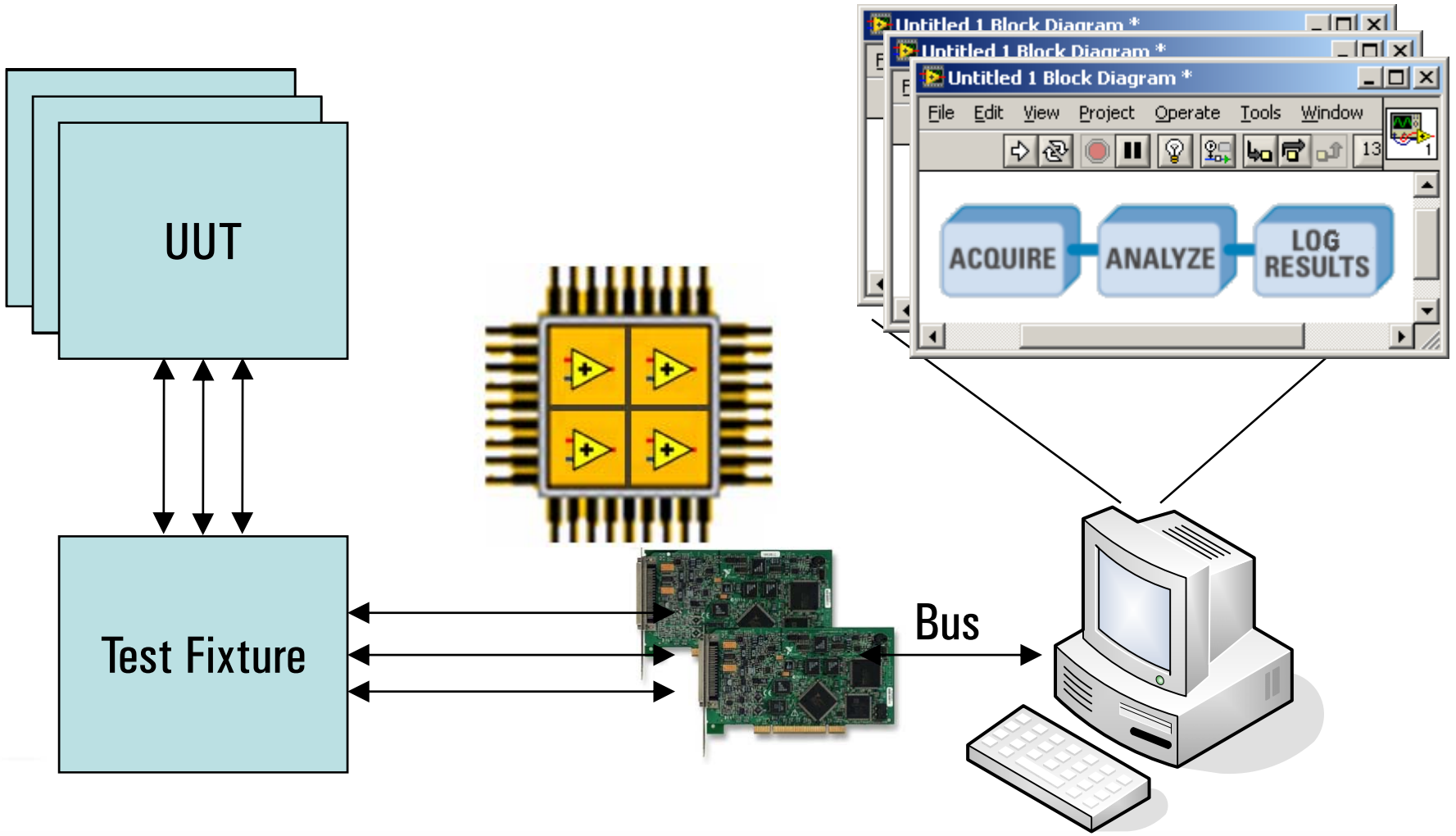


Benefits – Performance Test Executive

- Offload CPU
- Minimize bus/driver communication



Benefits – Parallelization



Design Considerations For Test

- Functionality
- Performance
- Flexibility
- Scalability
- Reusability
- Complexity
- Development Cost
- Maintenance Cost
- System Cost
- Robustness