FPGA Based Instrumentation

Johan Olsson
Saab Account Manager

Klas Andersson
Sr Applications Engineer

Agenda

- What is an FPGA?
- Benefits of an FPGA
- Working With FPGAs
  - LabVIEW
  - VHDL
- Example Applications

FPGA Technology

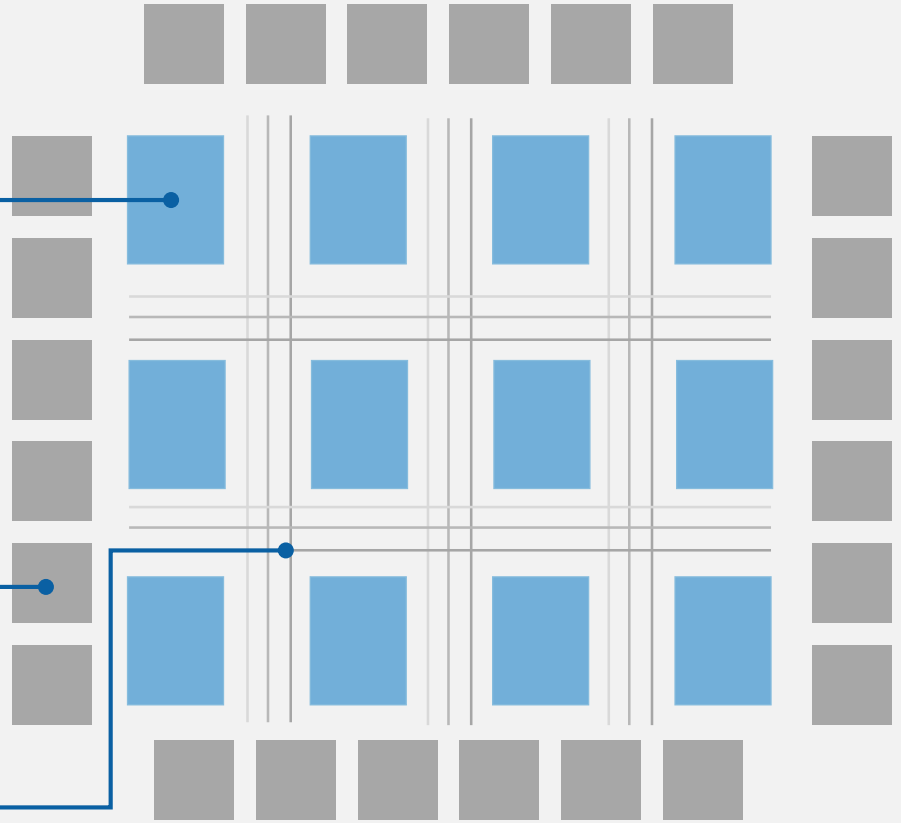
Memory Blocks
Store data sets or values in user defined RAM

Configurable Logic Blocks (CLBs)
Implement logic using flip-flops and LUTs

Multipliers and DSPs
Implement signal processing using multiplier and multiplier-accumulate circuitry

I/O Blocks
Directly access digital and analog I/O

Programmable Interconnects
Route signals through the FPGA matrix



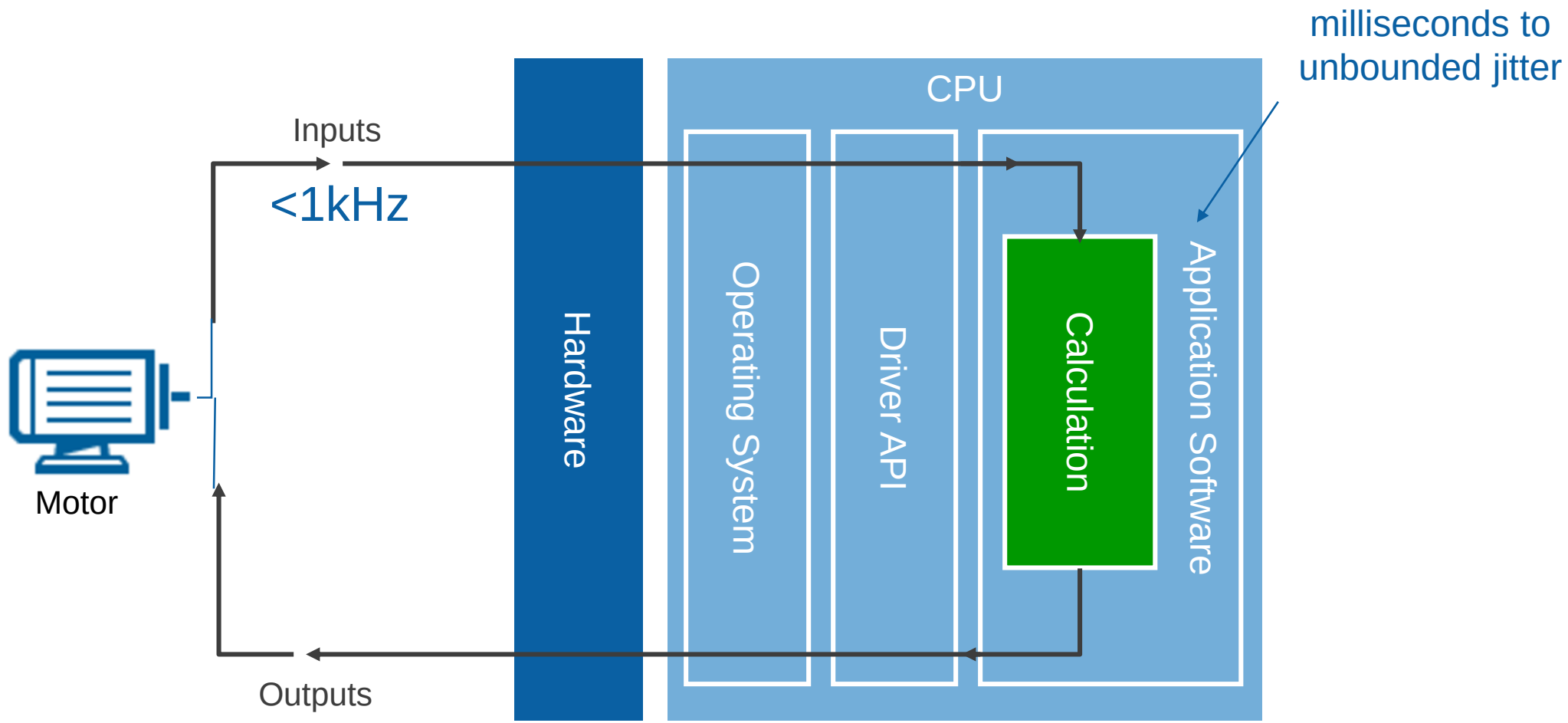
NATIONAL INSTRUMENTS

Benefits of an FPGA

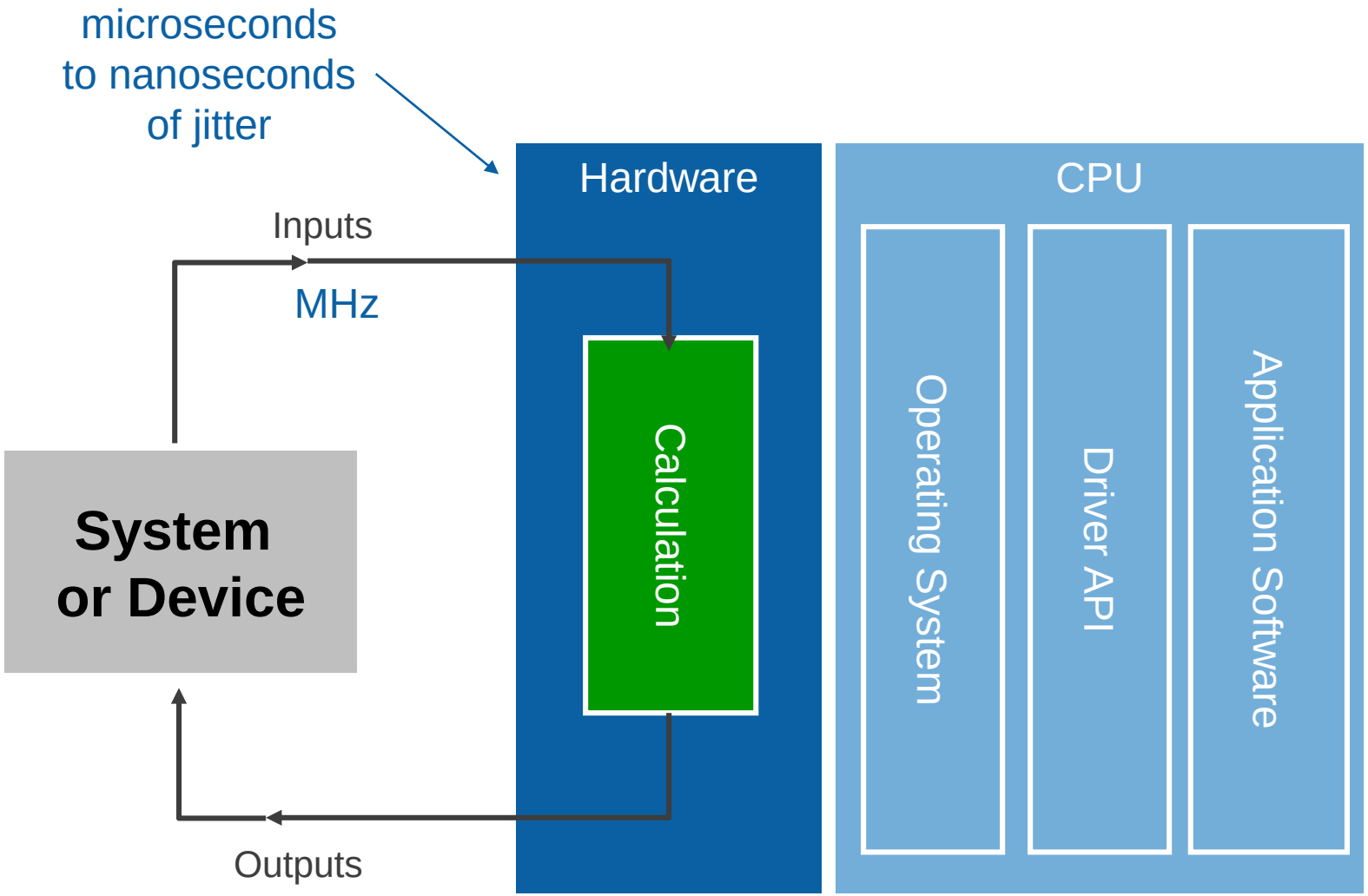
- Maximum performance
- Maximum reliability
- Reconfigurable functionality
- Offload processing
- Cost

When is an FPGA used?

- Fast I/O response times and specialized functionality
- Rapid prototyping and verification without the fabrication process of custom ASIC design
- Implementing custom functionality with the reliability of dedicated deterministic hardware
- Field-upgradable after deployment



NATIONAL INSTRUMENTS



NATIONAL INSTRUMENTS

NI’s Portfolio of FPGA-enabled Instruments

SOM

CompactRIO

Industrial Controller

Multifunction RIO

USRP RIO

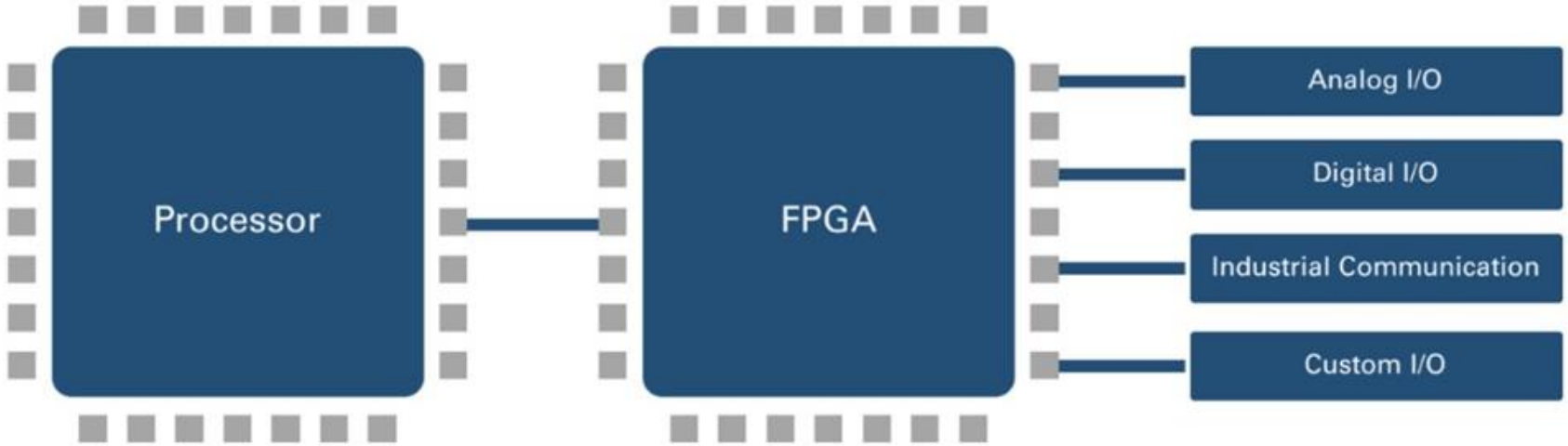
Modular Instruments

FlexRIO

RF Instruments

NATIONAL INSTRUMENTS

NI’s Portfolio of FPGA-enabled Instruments



Computation
- Real-Time Analytics
- LabVIEW Math & Analysis Libraries
- Algorithms, Decision Making

Connectivity
- Data Transfer Mechanisms
- Network Interface

Computation
- Signal Processing
- Data Reduction

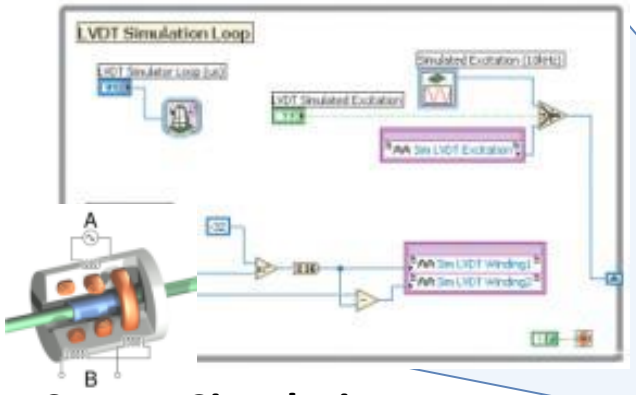
Connectivity
- Custom Timing, Triggering and Synchronization
- Custom Protocols

Control
- Fast, Deterministic, Closed-Loop Control (MHz rates)
- Safety, Reliability

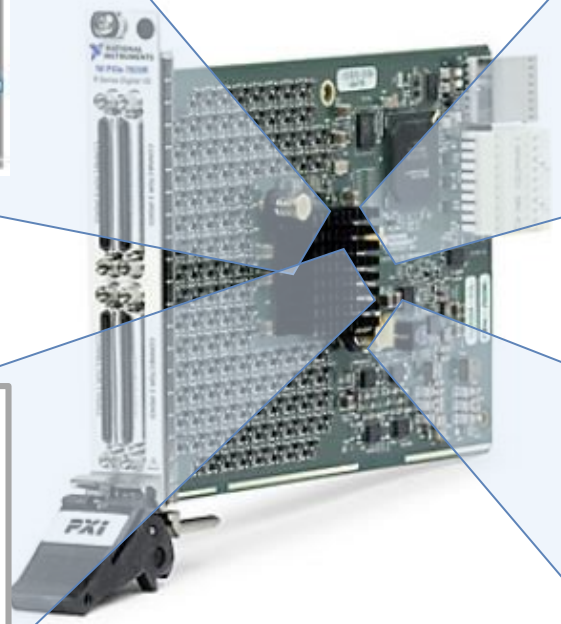
Connectivity
- Any Sensor
- Any Protocol
- Industrially Rated
- Signal Conditioning
- Cameras, Drives, Actuators
- Functional Safety

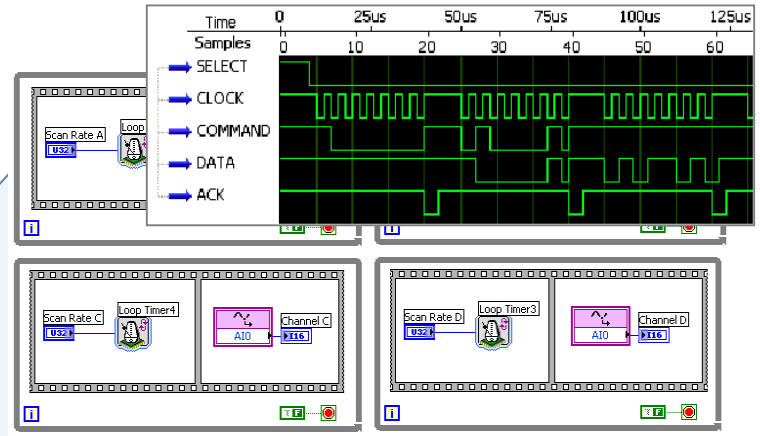
PXI Multifunction Reconfigurable I/O Module

- Featuring Built-In ADCs, DACs, and DIO

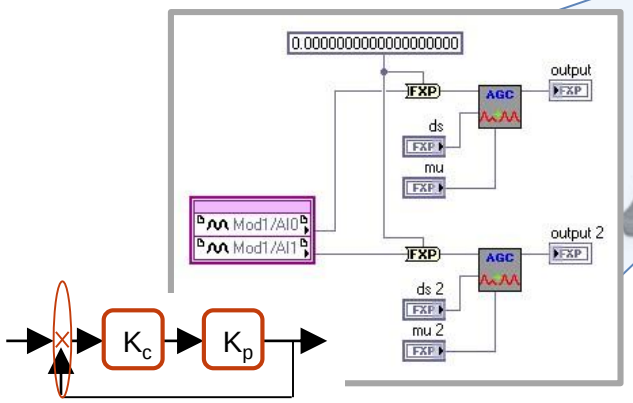


Sensor Simulation

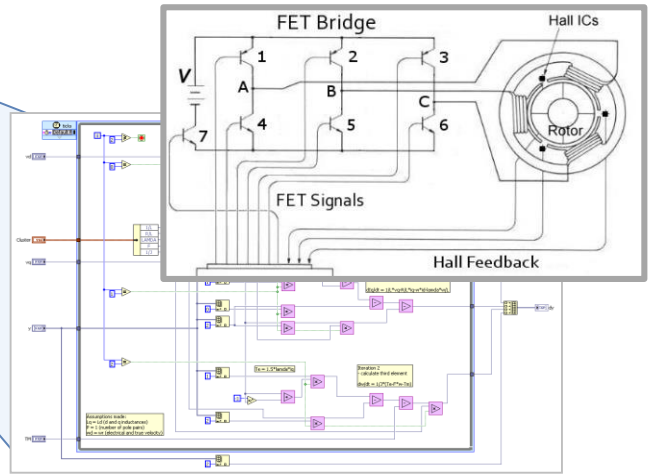




Custom I/O and Protocols



High-Speed Control

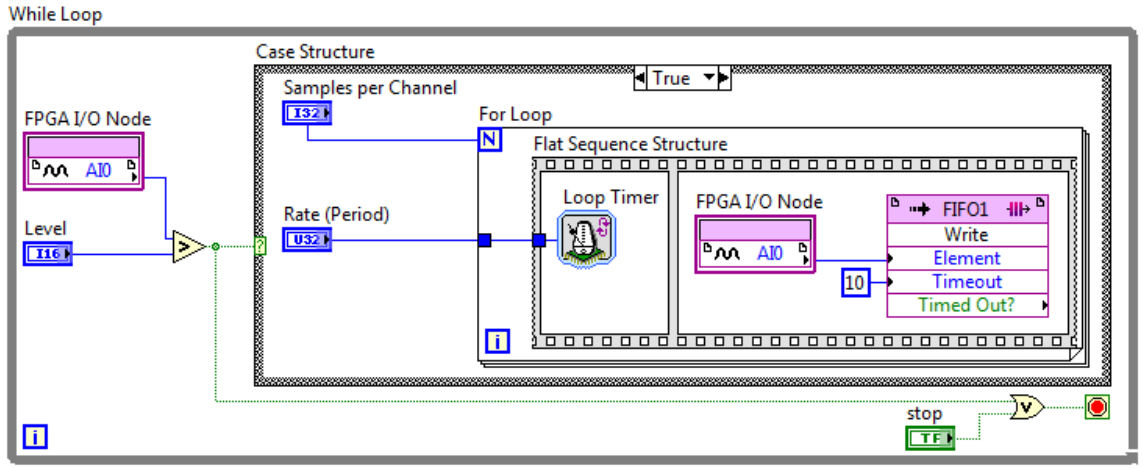


High-Speed Model Execution

NATIONAL INSTRUMENTS

Programming FPGAs

LabVIEW™

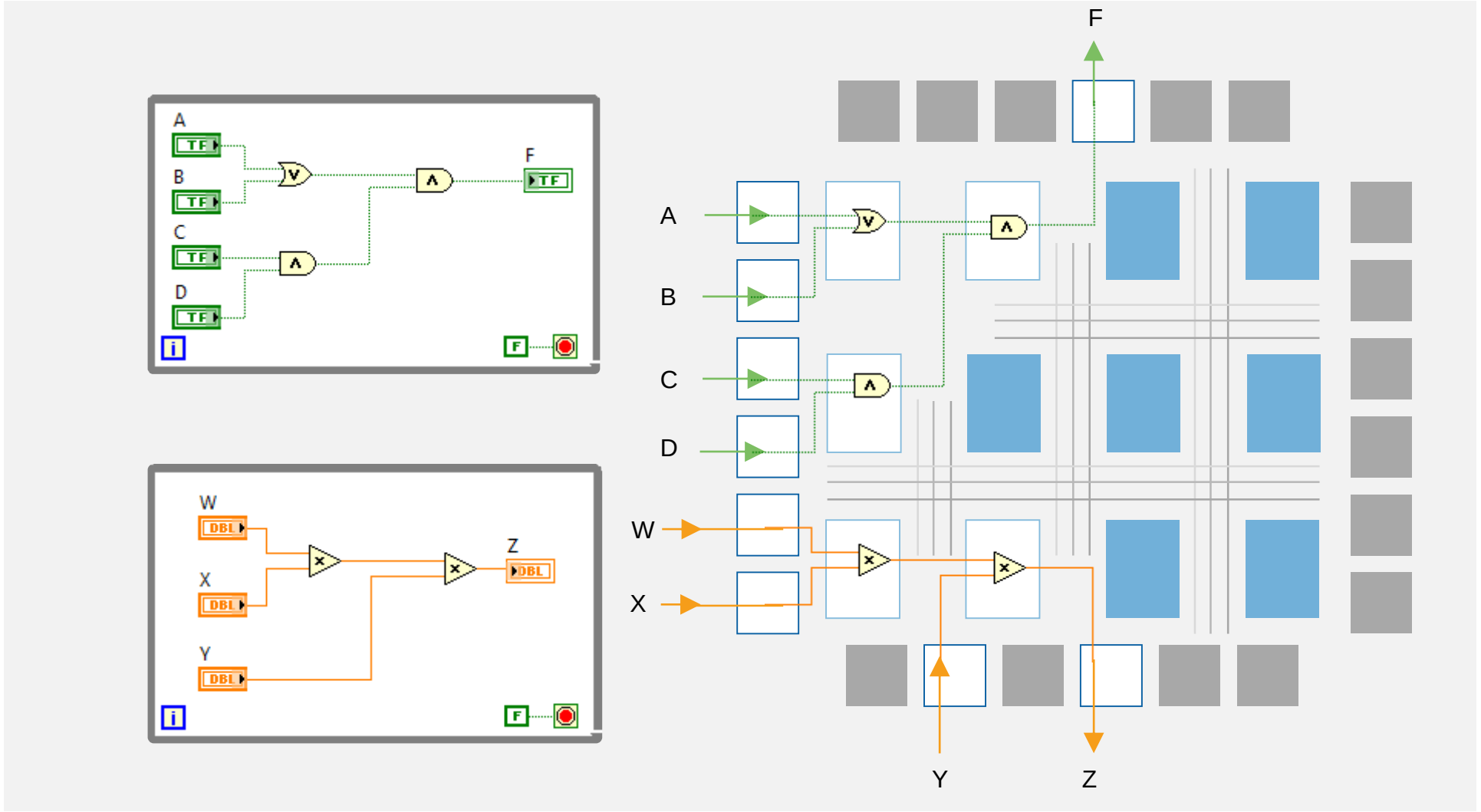


VHDL

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity signed_adder is
  port
  (
    aclr : in   std_logic;
    clk  : in   std_logic;
    a    : in   std_logic_vector;
    b    : in   std_logic_vector;
    q    : out  std_logic_vector
  );
end signed_adder;

architecture signed_adder_arch of signed_adder is
  signal q_s : signed(a'high+1 downto 0); -- extra bit wide

begin -- architecture
  assert(a'length >= b'length)
    report "Port A must be the longer vector if different sizes!"
    severity FAILURE;
  q <= std_logic_vector(q_s);

  adding_proc:
  process (aclr, clk)
    begin
      if (aclr = '1') then
        q_s <= (others => '0');
      elsif rising_edge(clk) then
        q_s <= ('0'&signed(a)) + ('0'&signed(b));
      end if; -- clk'd
    end process;

end signed_adder_arch;
```

NATIONAL INSTRUMENTS™

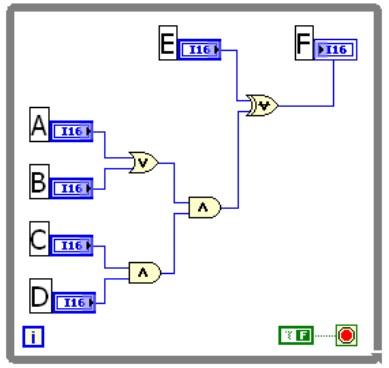
Mapping LabVIEW to an FPGA



NATIONAL INSTRUMENTS

Compilation Process

LabVIEW FPGA Code



Compile VHDL through Xilinx

```
process synchronizationFFs;

-- Then we keep track of what the digital input was on the previous
-- clock cycle by inserting another flip flop
PreviousDigitalInputFF:
process( aReset, Clk )
begin
  if aReset then
    cPrevDigitalInput <= false;
  elsif rising_edge(Clk) then
    cPrevDigitalInput <= cDigitalInput;
  end if;
end process PreviousDigitalInputFF;

-- Then we have a little combinatorial logic to detect a rising edge
cRisingEdgeDetected <= cDigitalInput and not cPrevDigitalInput;

-- And finally we have a register that increments when that rising
-- edge is detected.
counterRegister:
```

FPGA Logic Implementation

PXI High-Speed Serial Instrument

Translation
VHDL Generation

Optimization
Analyze Logic Reduction

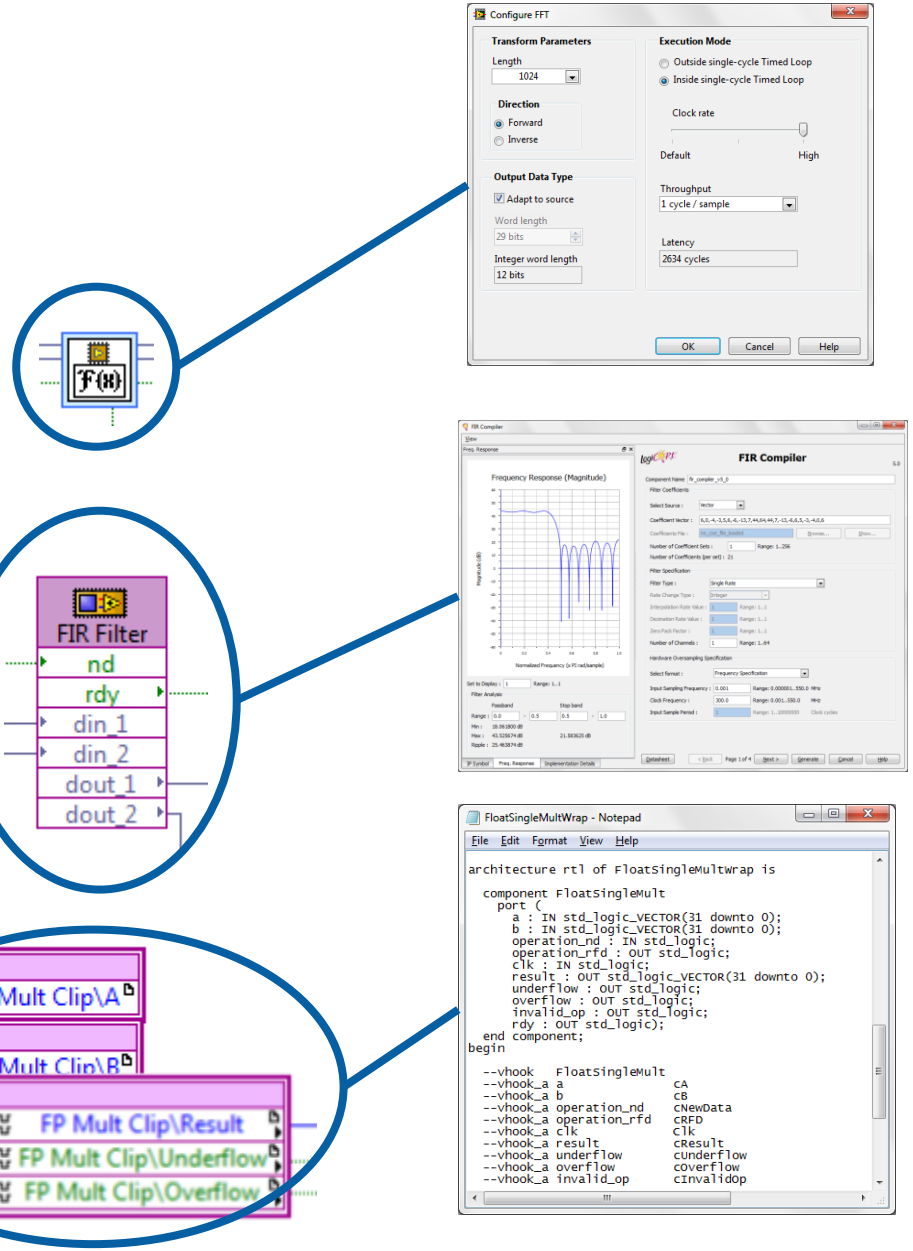
Synthesis
Place and Route Timing Verification

Bit Stream Generation
Download & Run

NATIONAL INSTRUMENTS

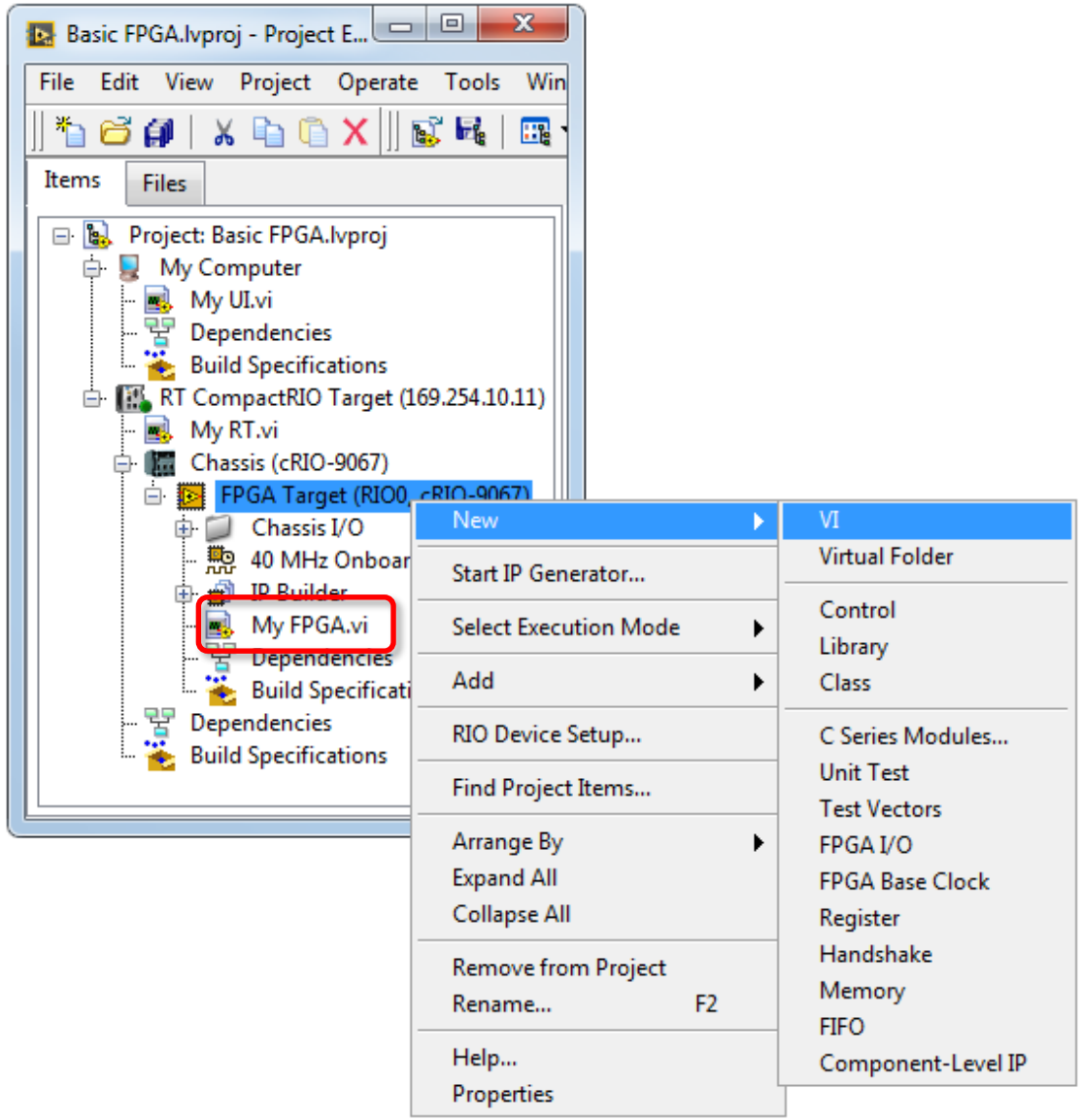
Implementation Options

- LabVIEW FPGA
  - Custom
  - Existing IP
  - Integrated
- IP Integration Node
  - Xilinx COREGEN
- Component-Level IP (CLIP)
  - Existing VHDL



NATIONAL INSTRUMENTS

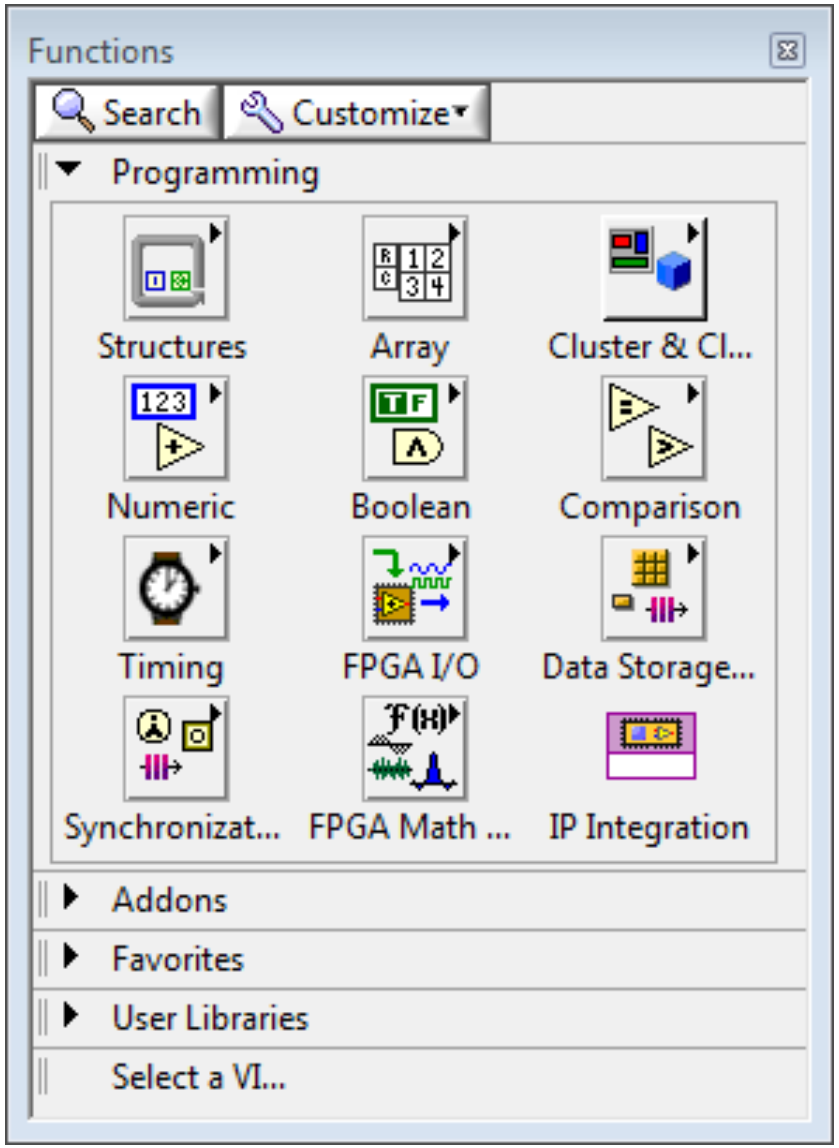
Add a VI Under the FPGA Target



NATIONAL INSTRUMENTS

FPGA Palettes

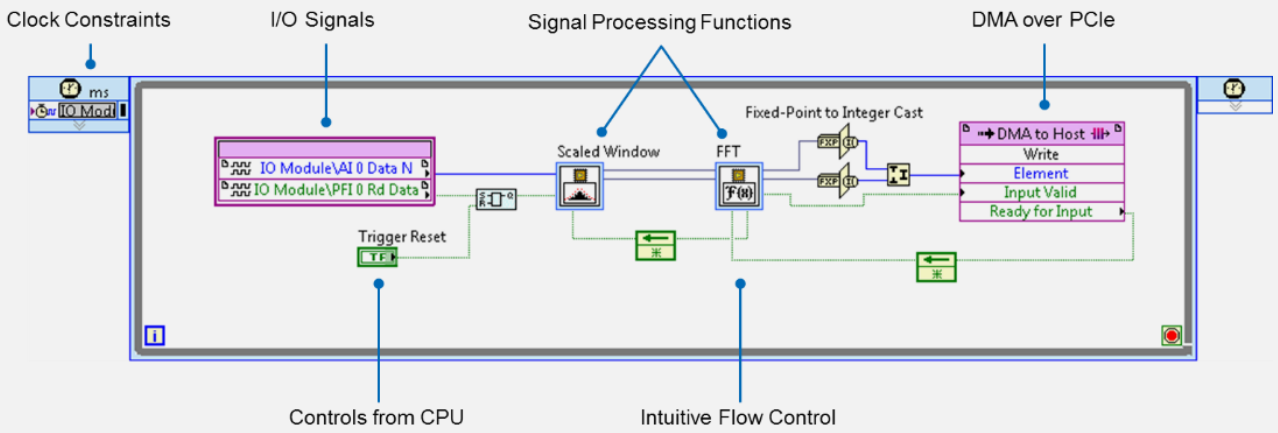
- VIs under the FPGA target inherit the FPGA Function Palette
- Many palettes are similar to LabVIEW for Windows
- FPGA-specific palettes
  - FPGA I/O
  - Data Storage & Transfer
  - Synchronization
  - FPGA Math & Analysis
  - IP Integration



NATIONAL INSTRUMENTS

Programming FPGAs with LabVIEW

Focus on your algorithms, not infrastructure



Save time with extensive libraries of FPGA IP

LabVIEW FPGA IP Commonly used with FlexRIO

- 10 Gigabit Ethernet UDP
- 3-Phase PLL
- Accumulator
- All-digital PLL
- Area measurements
- Bayer decoding
- Binary morphology
- Binary object detection
- BRAM delay
- BRAM FIFO
- BRAM packetizer
- Butterworth filter
- Centroid calculation
- Channel emulation
- Channel power
- CIC compiler
- Color extraction
- Color space conversion
- Complex multiply
- Corner detection
- Counters
- D latch
- Delay
- Digital gain
- Digital pre-distortion
- Digital pulse processing filter
- Discrete delay
- Discrete normalized integrator
- Divide
- Dot product
- DPO
- DRAM FIFO IDL
- DRAM packetizer
- DSP48 node
- DUC/DDC compiler
- Edge detection
- Equalization
- Exponential
- FFT
- Filtering
- FIR compiler
- Fixed-point filter design
- Fractional interpolator
- Fractional resampler
- Frequency domain measurements
- Frequency mask trigger
- Frequency shift
- Halfband decimator
- Handshake
- Hardware test sequencer
- I2C
- Image operators
- Image transforms
- Instruction sequencer
- IQ impairment correction
- Line detection
- Linear interpolation
- Lock-in amplifier filter
- Log
- Matrix multiply
- Matrix transpose
- Mean, Var, Std deviation
- Memory IDL
- Moving average
- N channel DDC
- Natural log
- Noise generation
- Normalized square
- Notch filter
- Persistence display
- PFT channelizer
- PID
- Pipeline frequency transform (PFT)
- Polar to X/Y conversion
- Power level trigger
- Power servoing
- Power spectrum
- Programmable filter
- Pulse measurements
- Reciprocal
- RFFE
- Rising/falling edge detect
- RS-232
- Scaled window
- Shading correction
- Sin & Cos
- Spectrogram
- SPI
- Square root
- Streaming controller
- Streaming IDL
- Synchronous latch
- Trigger IDL
- Unit delay
- VITA-49 data packing
- Waveform generation
- Waveform match trigger
- Waveform math
- X/Y to polar conversion
- Xilinx Aurora
- Zero crossing
- Zero order hold
- Z-Transform delay

NATIONAL INSTRUMENTS

How to program these Instruments with VHDL / Verilog

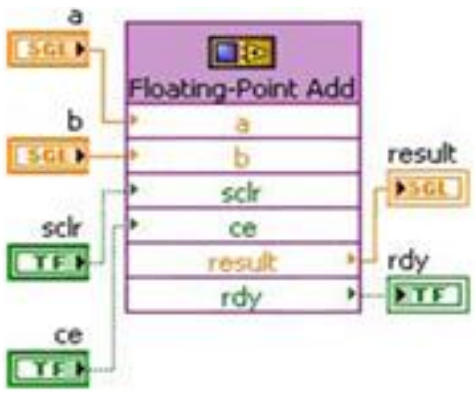
You want to leverage

- Work done by design teams
- Existing IP libraries

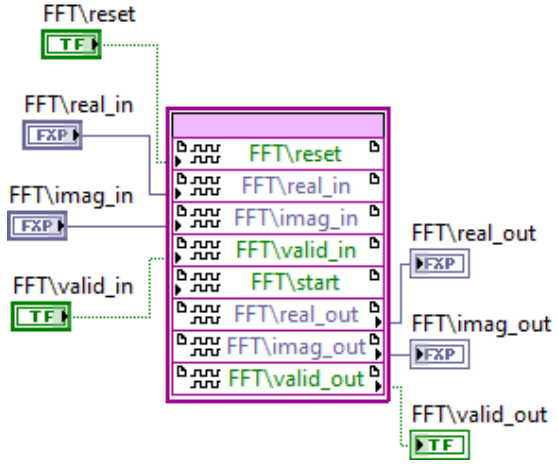
You are more comfortable/proficient with traditional digital design tools

Existing tools in LabVIEW

IP Integration Node



CLIP (Compentent Level IP)

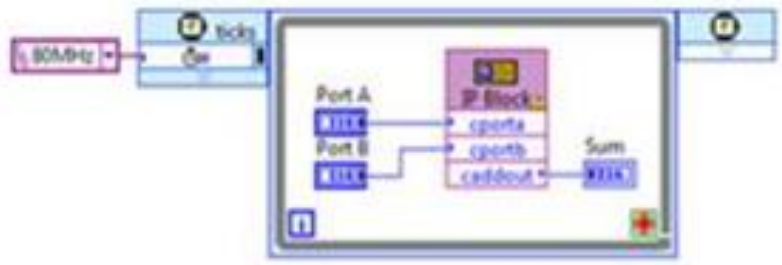


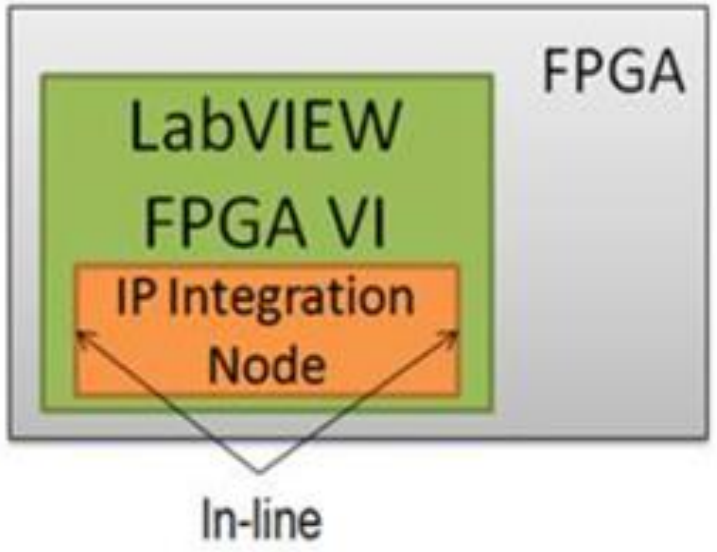
NATIONAL INSTRUMENTS

Reuse VHDL code in LabVIEW FPGA

IP Integration Node

- Executes VHDL in-line with FPGA VI code
  - Required to be placed in a SCTL
- Added and configured on FPGA VI Block diagram
  - Xilinx Compilation tools needs to be locally installed to configure the node
  - Option to execute IP faster than the SCTL
- Supports Simulation
- Limits
  - No HW access
  - IP Requirement/limitation list in LabVIEW Help



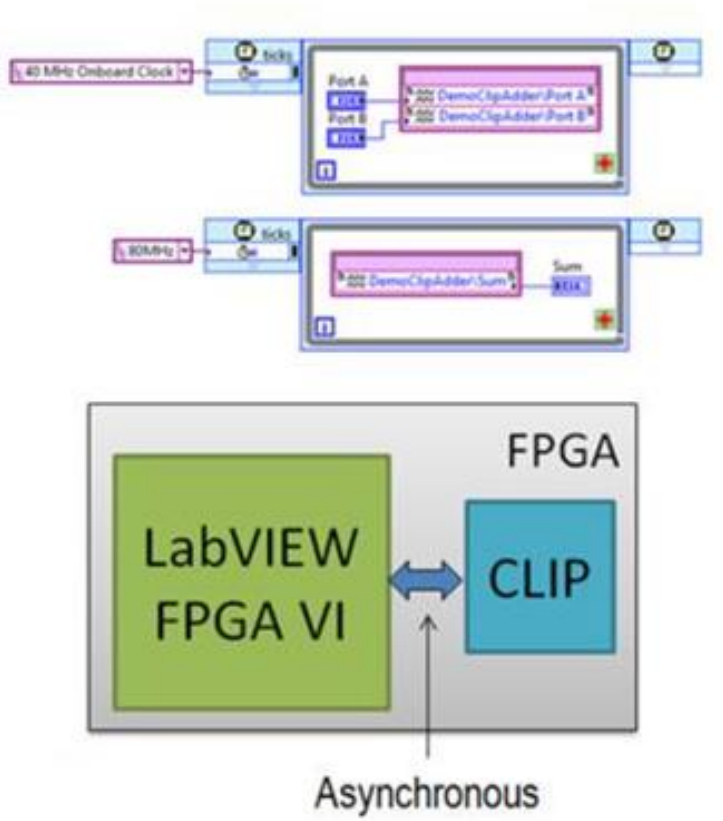


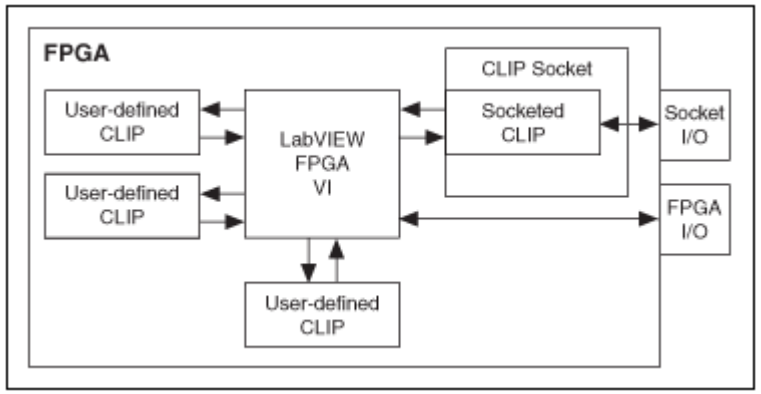
NATIONAL INSTRUMENTS

Reuse VHDL code in LabVIEW FPGA

Component-Level IP (CLIP)

- Executes VHDL parallel with LabVIEW FPGA code
- Configured and added on Project Level
  - User defines the entire interface to LV FPGA Block Diagram
  - IP Interface definition in XML file. Wizard creates this when importing IP.
- Two CLIP Types:
  - User-defined CLIP
    - Enables VHDL Code to communicate directly with an FPGA VI
  - Socketed CLIP
    - Enables VHDL Code to communicate directly with an FPGA VI and directly access FPGA clocks and I/O Pins and not available by FPGA VI (Refer to HW documentation)
- FPGA Simulation mode not supported, require third-party simulation.

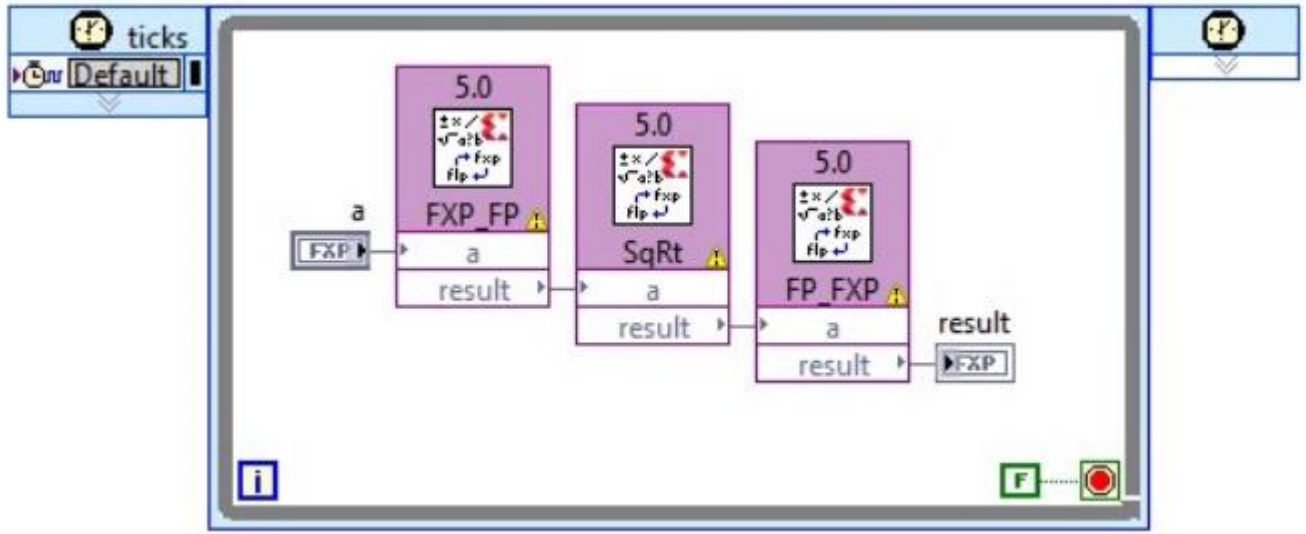




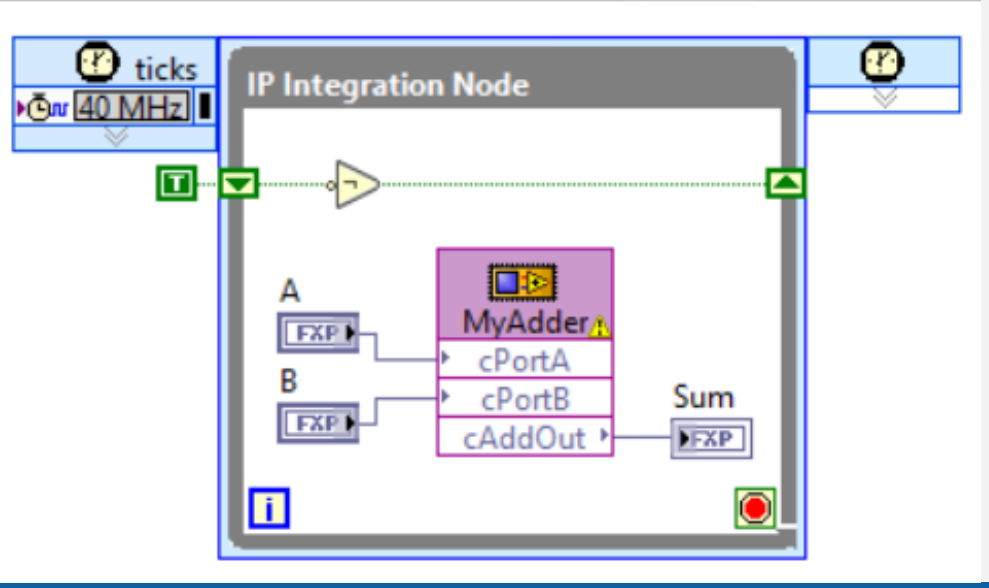
NATIONAL INSTRUMENTS

Using Xilinx Core IP

- Uses IP Integration Node
  - Configured when placed on block diagram
  - Must be placed in a SCTL
- **Programming»Xilinx IP** palette displays IP for selected FPGA Device family
- Xilinx Compilation tools needs to be locally installed
- Some Xilinx Core IP might require additional Xilinx Licenses



NATIONAL INSTRUMENTS



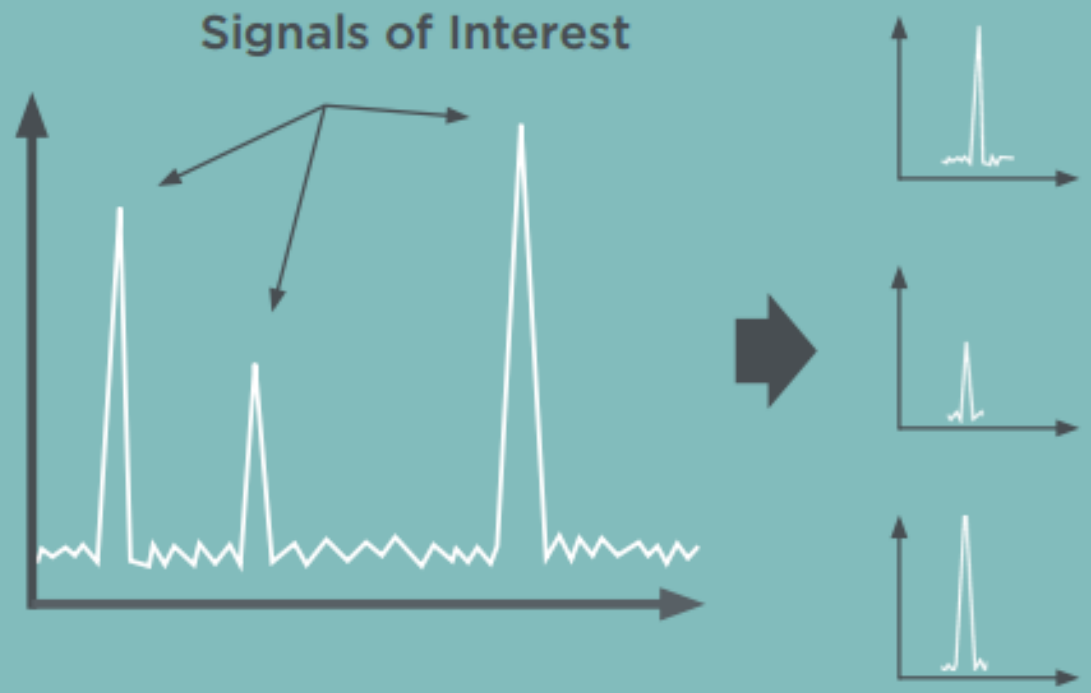
IP Integration Node Example

C:/Users/SE/Desktop/SAAB TS_CVI_FPGA/Examples/IPNodeFPGA/3rd Part

```
1  Library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity DemoClipAdder is
6      port (
7              clk    : in std_logic;
8              aReset  : in std_logic;
9              cPortA  : in std_logic_vector(15 downto 0)
10             cPortB  : in std_logic_vector(15 downto 0)
11             cAddOut : out std_logic_vector(15 downto 0
12         );
13 end DemoClipAdder;
14
15 architecture rtl of DemoClipAdder is
16 begin
17     process(aReset, clk) begin
18         if(aReset = '1') then
19             cAddOut <= (others => '0');
20         elsif rising edge(clk) then
```

NATIONAL INSTRUMENTS

Example Applications



Channelizer Solution

Up to 1024 Channels

Tune each channel for center frequency, BW, gain, filter selection and On/off status.

Combine narrow and wide channels and have overlapping channels.

“Thanks to the modular platform, if an increased channel count is needed multiple NC-10 can be daisy chained via multiple peer-to-peer streams.”

NOVATOR SOLUTIONS

NATIONAL INSTRUMENTS

Multi-Channel Record & Playback

Record live narrow/wideband RF signals

Gapless, Triggered, Timestamped

Scalable to include multiple signals in one system

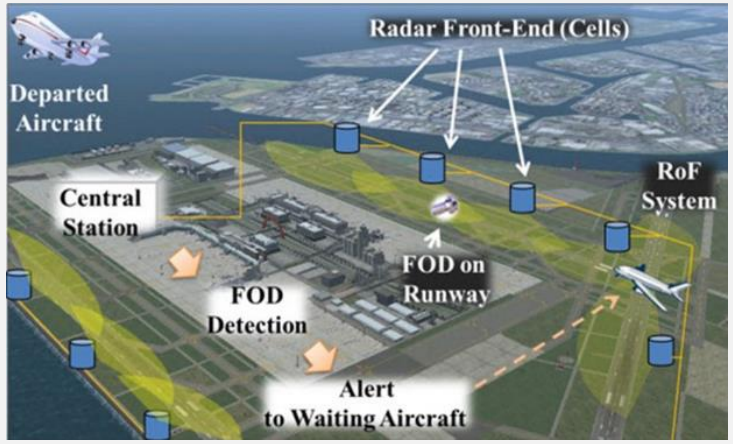
“Every customer has specific requirements. Using National Instruments hardware and software, we deliver ready-to-run systems with writing speeds greater than 3GB/s to meet our customers’ specifications.”

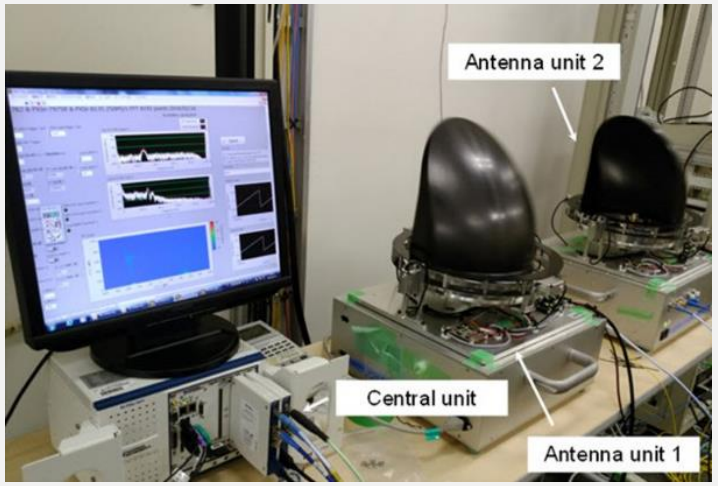
NOVATOR SOLUTIONS

NATIONAL INSTRUMENTS

CUSTOMER EXAMPLE

Creating an Airport Runway Foreign Object Debris Detection System



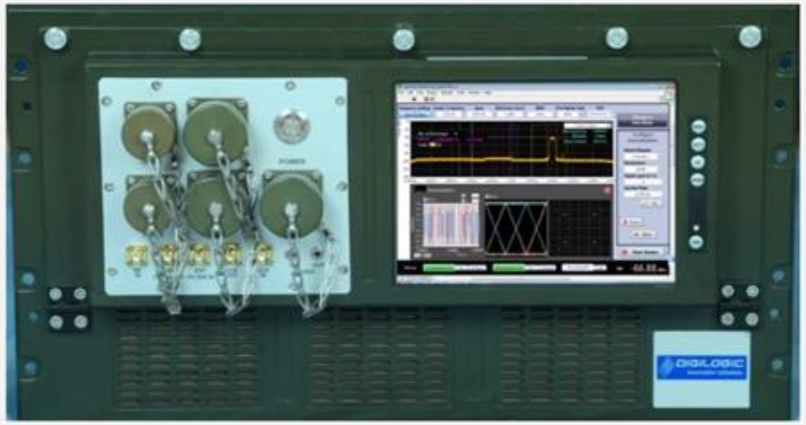


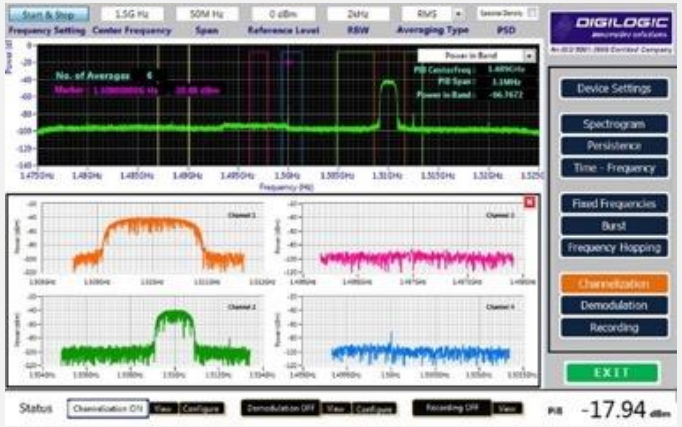
- Distributed radar system realized using PXIe and DMA FIFO for central signal processing
- Main algorithm of signal processing circuits implemented in less than one month, 90% faster than the conventional programming method

"We successfully developed the prototype of the FOD detection millimeter-wave radar system for airport runways using LabVIEW and FlexRIO."

Shunichi Futatsumori
Electronic Navigation Research Institute

Wide-Band Spectrum Monitoring System for Defense Applications



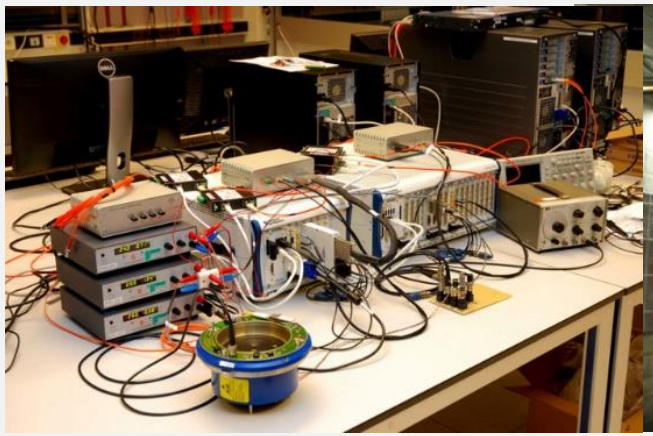


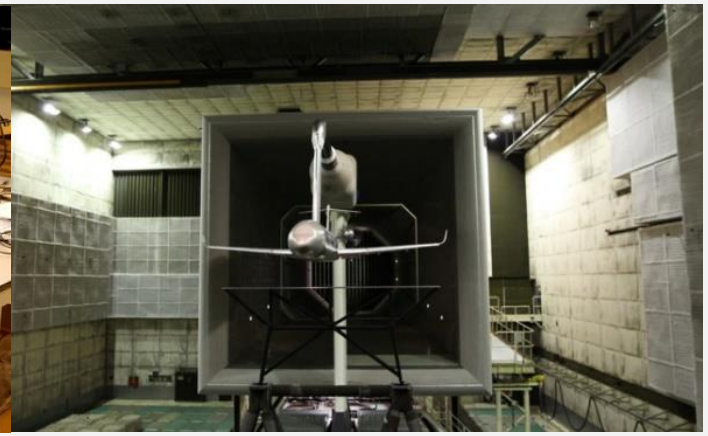
- Spectrum monitoring using PXIe 5667 VSA and NI FlexRIO
- Channelization, demodulation, and decoding performed on FlexRIO FPGA using peer-to-peer streaming

"We have developed a system that meets the requirements and with all essential monitoring capabilities in HF, VHF, and UHF bands to serve defense purposes within our scheduled time plan."

Avinash Reddy
Digilogic Systems Pvt. Ltd.

Characterizing Sound Profiles for a New Airbus Aircraft Using NI PXI





- Over 200 microphones and pressure sensors synchronized within 1 µs using PXI timing module
- NI FlexRIO allowed introduction of minor phase delays to compensate for lead wires and delta-sigma ADC delays

"We chose a National Instruments system because we had good experiences in the past with NI, and the NI modules had the capabilities we needed."

Johan de Goedge
National Aerospace Laboratory

NATIONAL INSTRUMENTS

Radar Target Simulator

- Integrated NI RF and FlexRIO hardware with LabVIEW to quickly develop a solution with an attractive GUI
- Control of individual parameters is possible using radar equation

"We decreased development time and increased overall cost savings because of NI off-the-shelf technology and a simplified programming approach."

Vahan Grigoryan
Insol LLC

Q&As