

NIDays

THE LabVIEW CONFERENCE

Extending Your LabVIEW Skills to
LabVIEW Real-Time and
LabVIEW FPGA

<Presenter Name>

<Presenter Title>

We all have a challenge to solve...



Power Distribution
and Control



Turbine Control



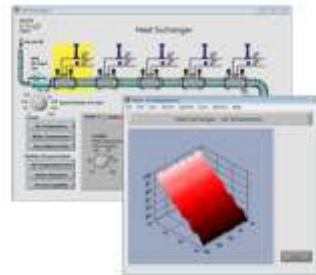
Industrial Machine
Control



Medical Device
Test



Structural
Monitoring



Process Control



Oil and Gas
Applications



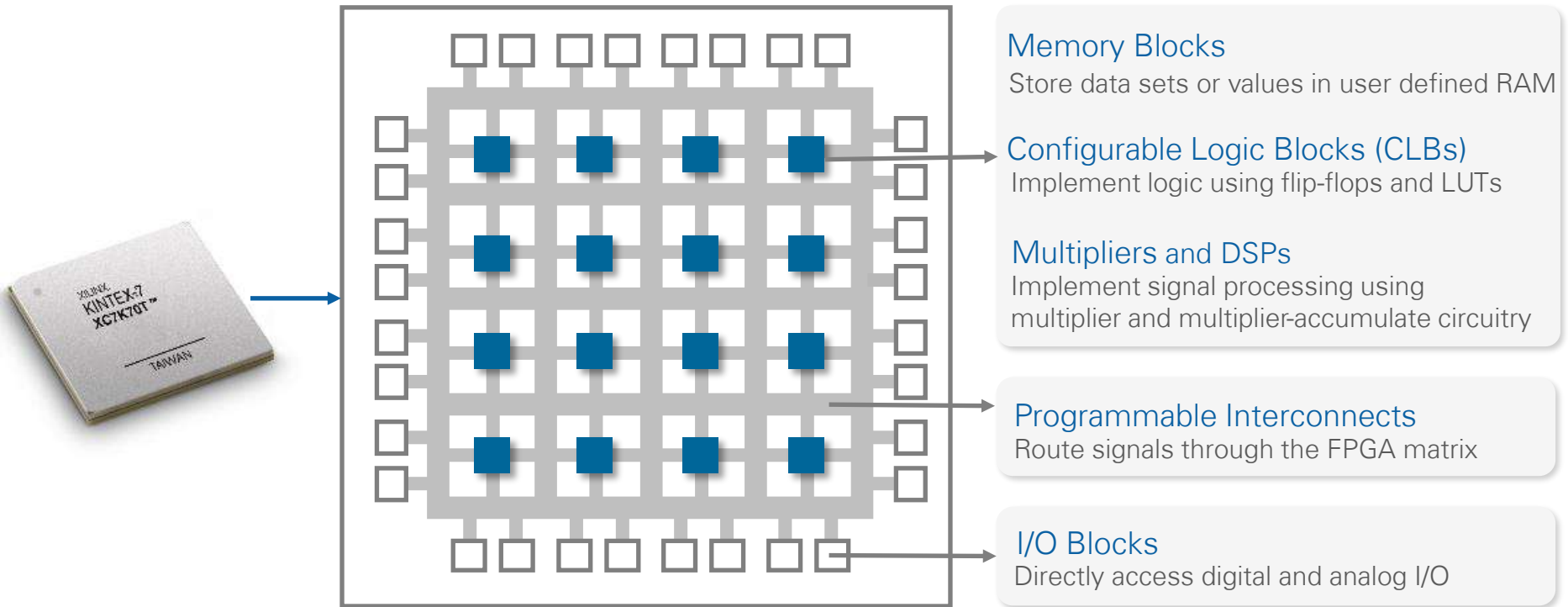
Power Monitoring
and Control

What is Real-Time?

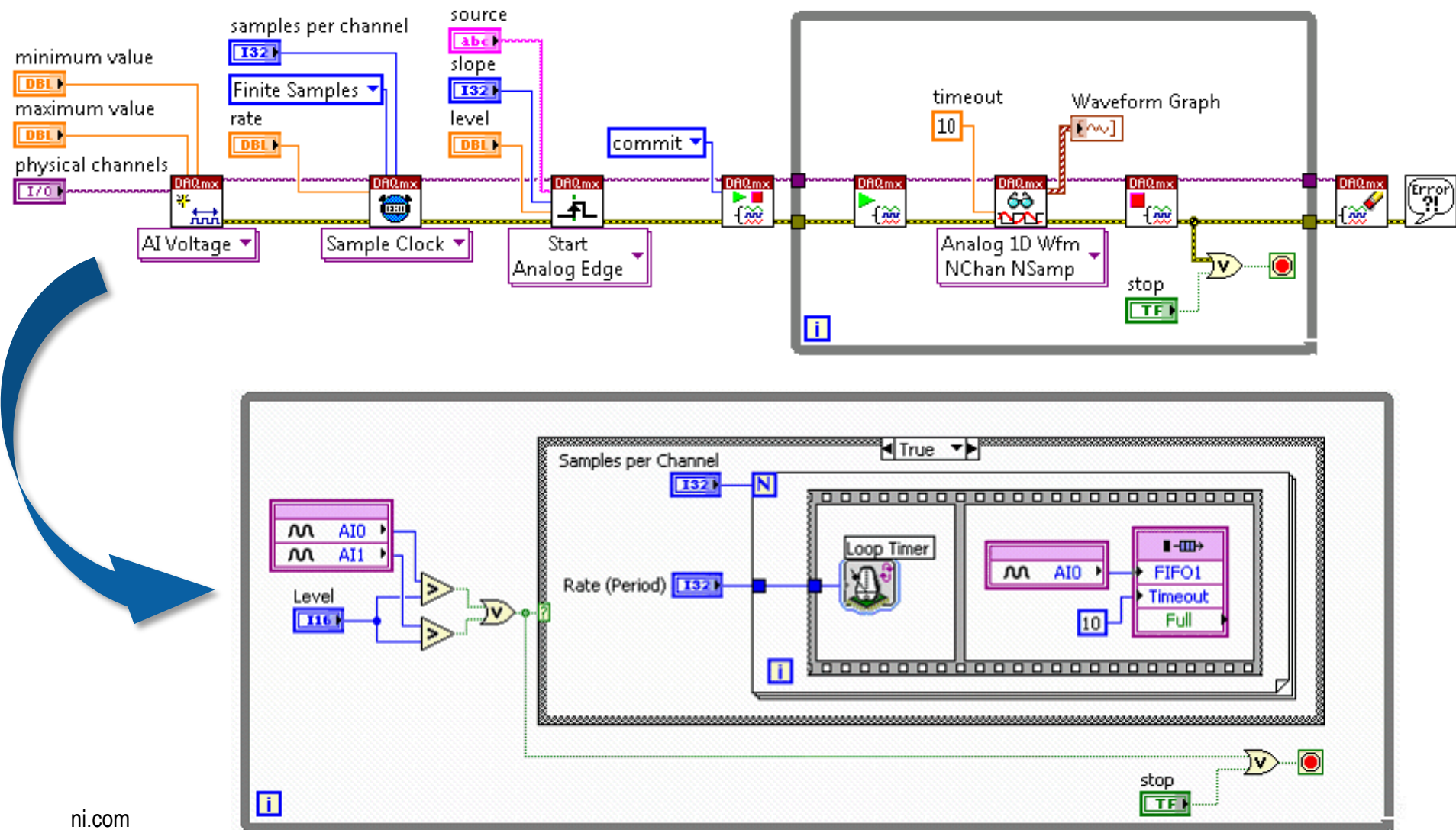
- Real-time **does not** always mean really fast
- Real-time means **absolute reliability**
- Real-time systems have timing constraints that must be met to avoid failure
- Determinism is the ability to complete a task within a fixed amount of time



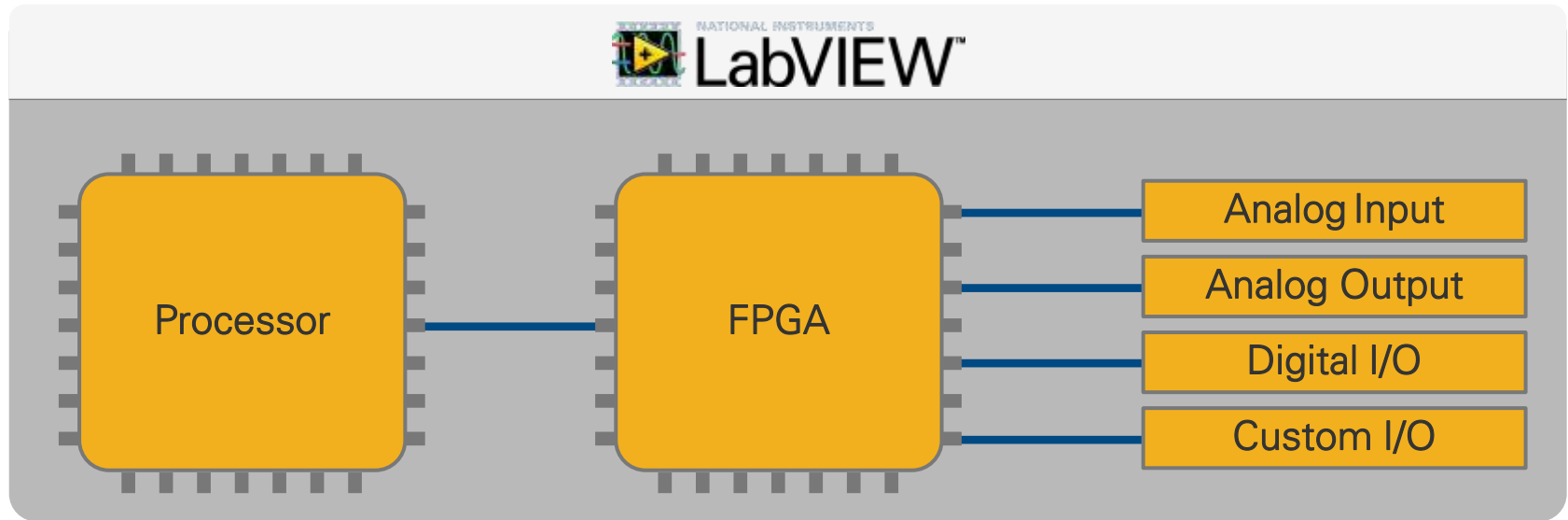
Field-Programmable Gate Array (FPGA)



Changing from the LabVIEW/DAQ Mindset



The LabVIEW RIO Architecture



CompactRIO & Single-Board RIO



Value



Ultra Rugged



Performance

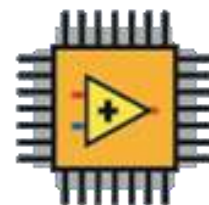
PXI, PC RIO (RSeries, FlexRIO)



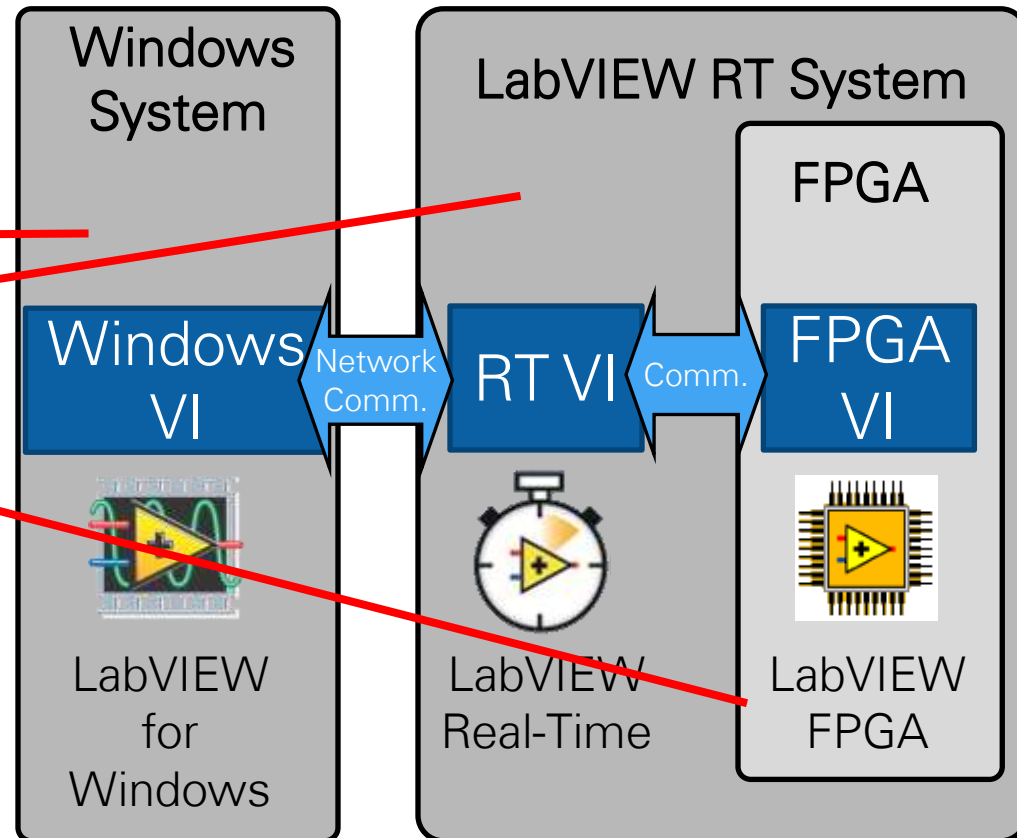
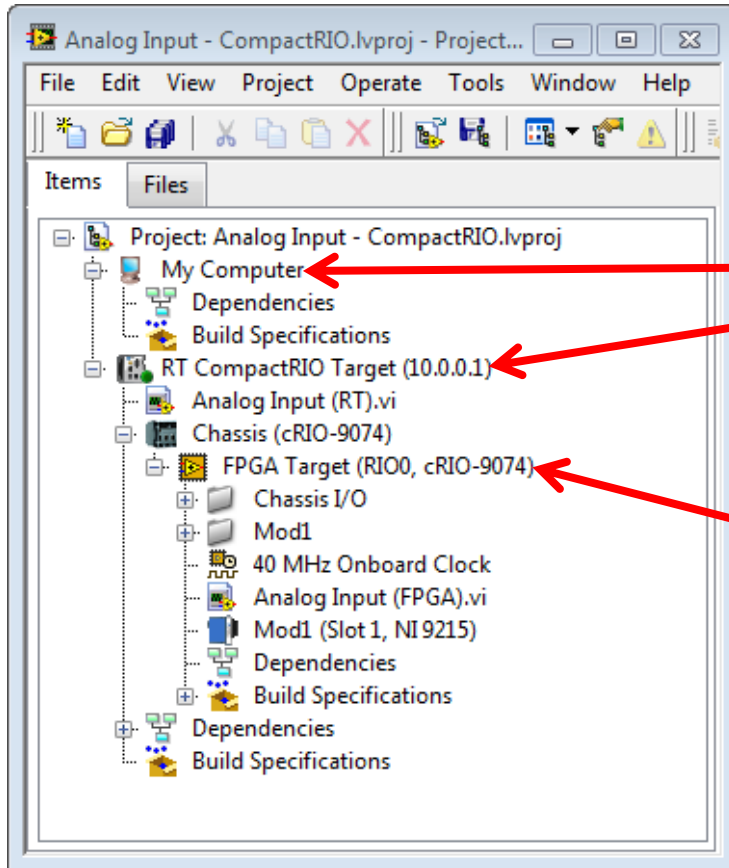
High Performance

Agenda

1. Using the LabVIEW Project
2. LabVIEW FPGA I/O Nodes
3. Working with the fixed-point data type
4. Understanding hardware clocks and concurrency
5. FPGA <--> RTOS communication
6. Test and compile
7. Working with constrained resources
8. Implementing high-priority tasks
9. Planning for the worst case scenario
10. Additional resources

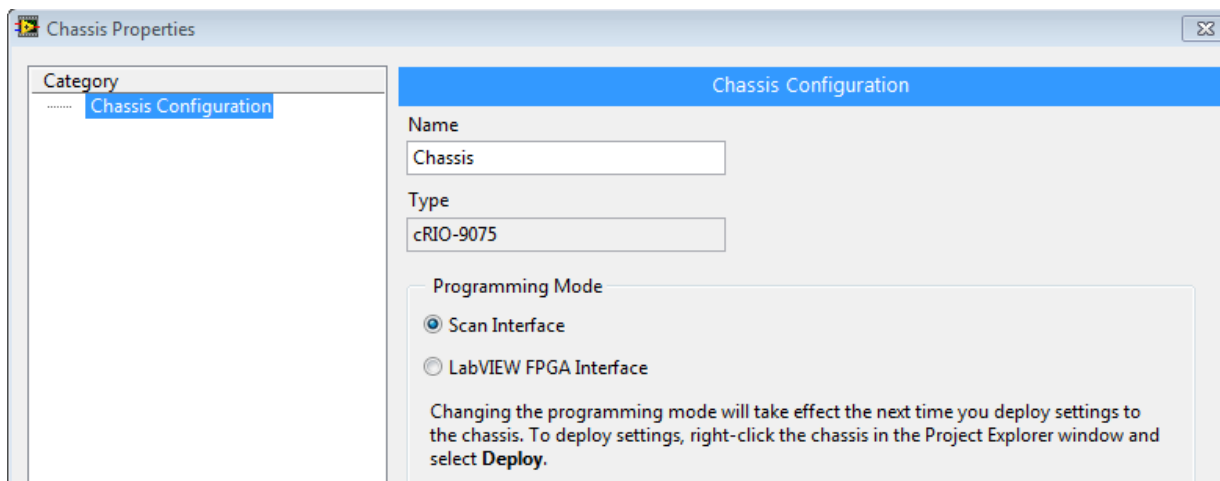
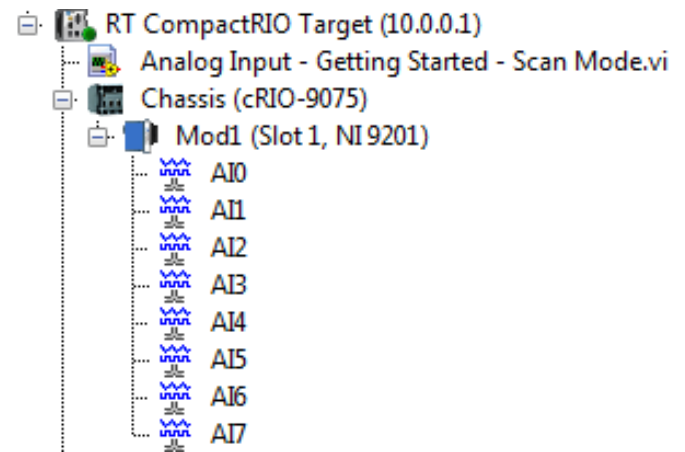


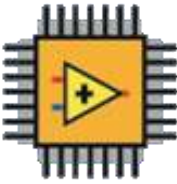
1 - Using the LabVIEW Project Demonstration



1 - Using the LabVIEW Project

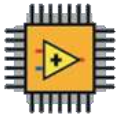
- Scan Interface Mode
 - Single-point access to I/O channels
 - Updates values at a single rate
 - Scan rate usually $< 1\text{kHz}$



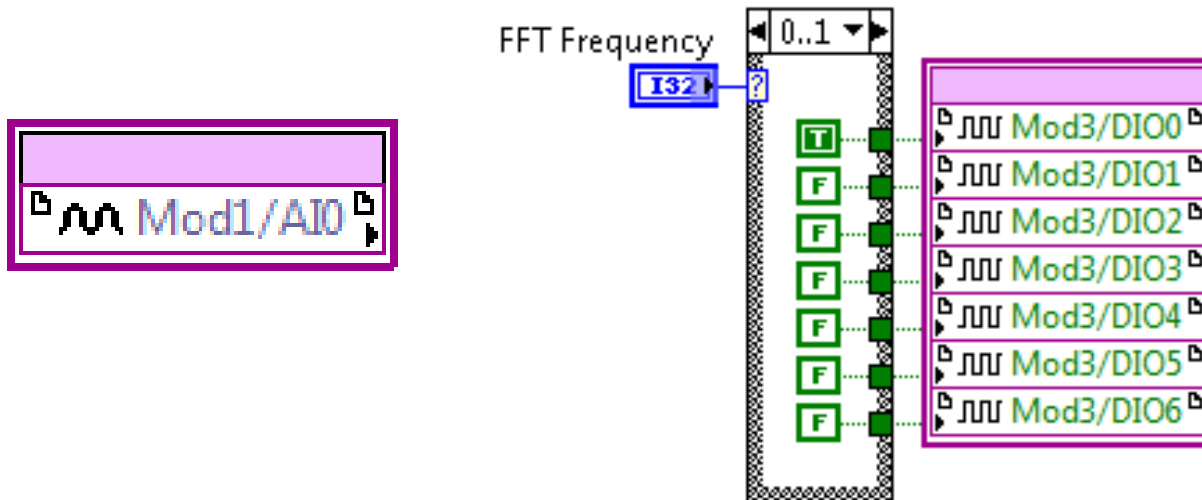


LabVIEW FPGA

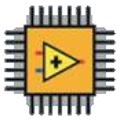
2 - LabVIEW FPGA I/O Nodes Demonstration



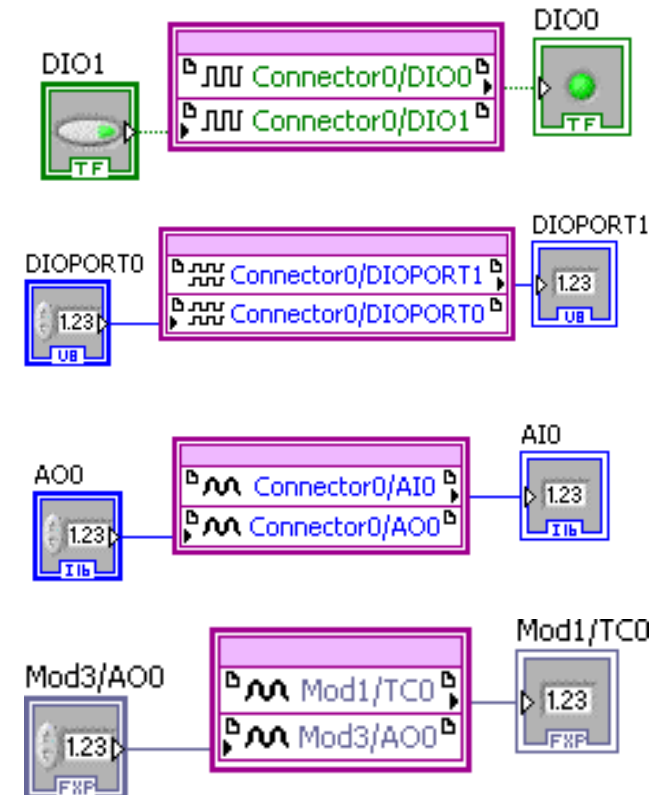
- Directly connected to I/O pins
- Use I/O Nodes to acquire and generate data
- Data rates are defined by the Analog and Digital modules
- With DAQmx, you get multiple samples per loop iteration
- With LabVIEW FPGA, the FPGA acquires one data value per loop iteration.

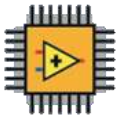


2 - LabVIEW FPGA I/O Nodes



- **Digital Line** – Writes/reads Boolean value to/from digital line
- **Digital Port (grouping of digital lines)**– Writes/reads unsigned integer value to/from digital port
- **Analog I/O** – Writes/reads data to/from an analog channel
 - R Series – Integer values
 - CompactRIO – Fixed-point values
- **Other**
 - Motion
 - CAN

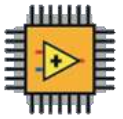




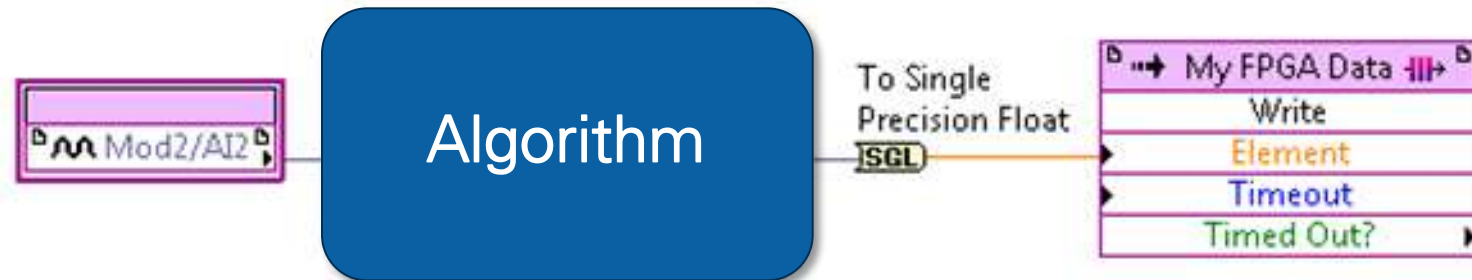
3 - Fixed-Point Math

- Floating Point: $0.5 * 0.5 = 0.25$
- Fixed Point: 1.1
 - [sign=+ | integer =1 | fraction = 1]

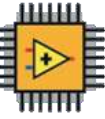
Rational Number	Fixed-Point Equivalent	Bits represented
1.5	0001.1000	8 total bits, 4 integer bits
1.5	00000001.10000000	16 total bits, 8 integer bits



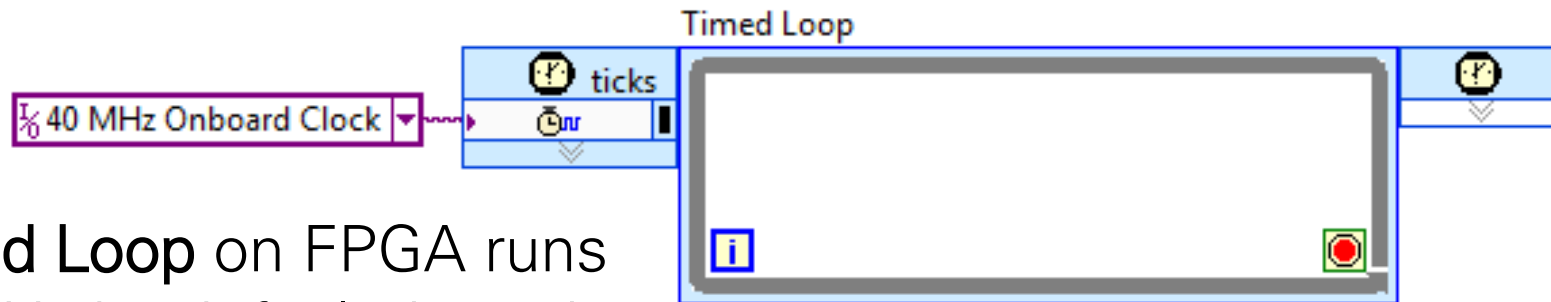
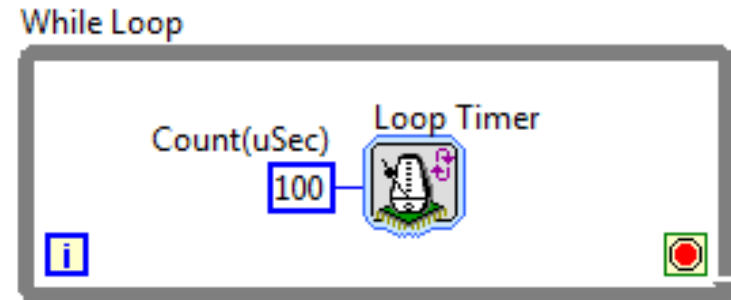
Use Cases for Floating Point Math



4 - Understanding Clocks and Hardware Concurrency

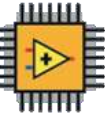


- A **While Loop** will execute at the rate specified in the **Loop Timer** function, either in ticks, ms, or μ s.



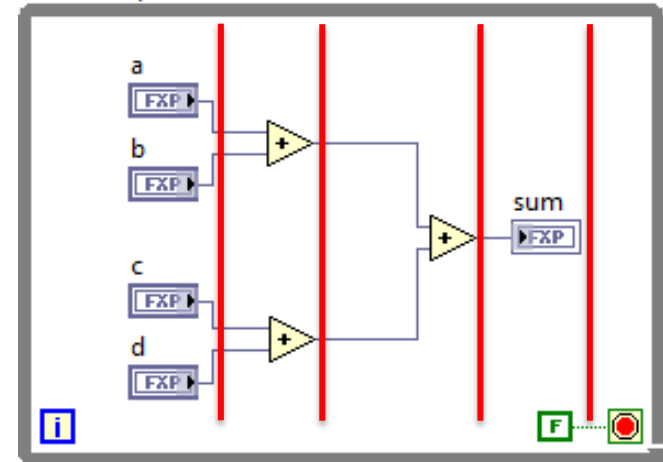
- A **Timed Loop** on FPGA runs at 40MHz by default, based on the Onboard Clock

4 - Understanding Clocks and Hardware Concurrency

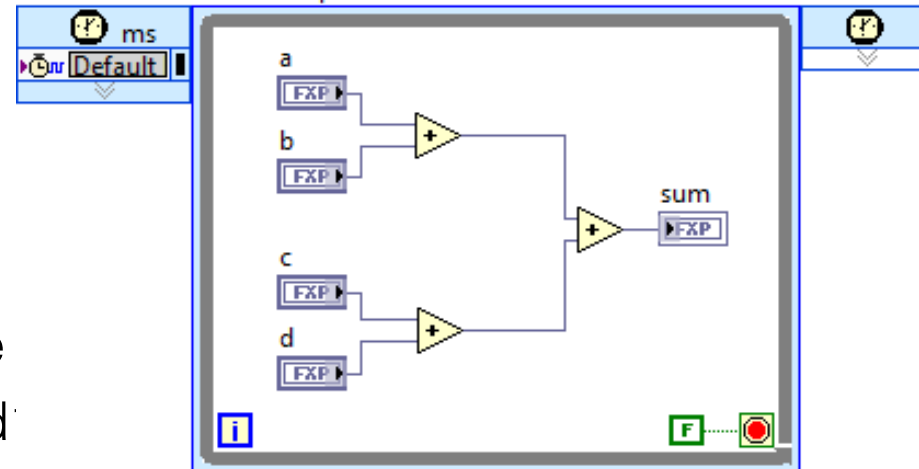


- The enable chain includes registers between each node that store values and execute at the rising edge of the clock
- A Timed Loop on FPGA is called a **Single Cycle Timed Loop (SCTL)**
 - Removes registers
 - Uses less resources
 - Code executes in 1 clock cycle
 - Not all functions are supported

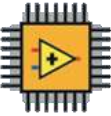
While Loop



Timed Loop

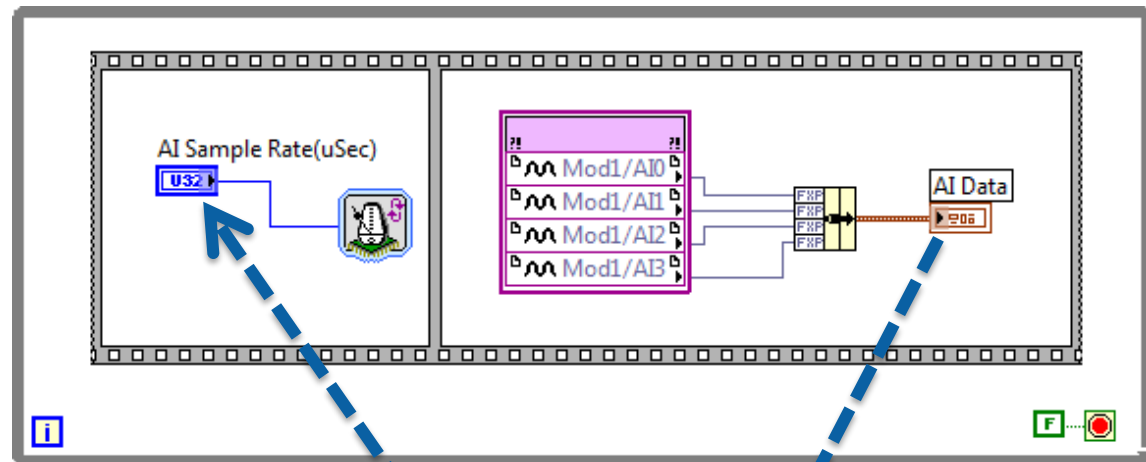


5 - Transferring Data to the RT Microprocessor

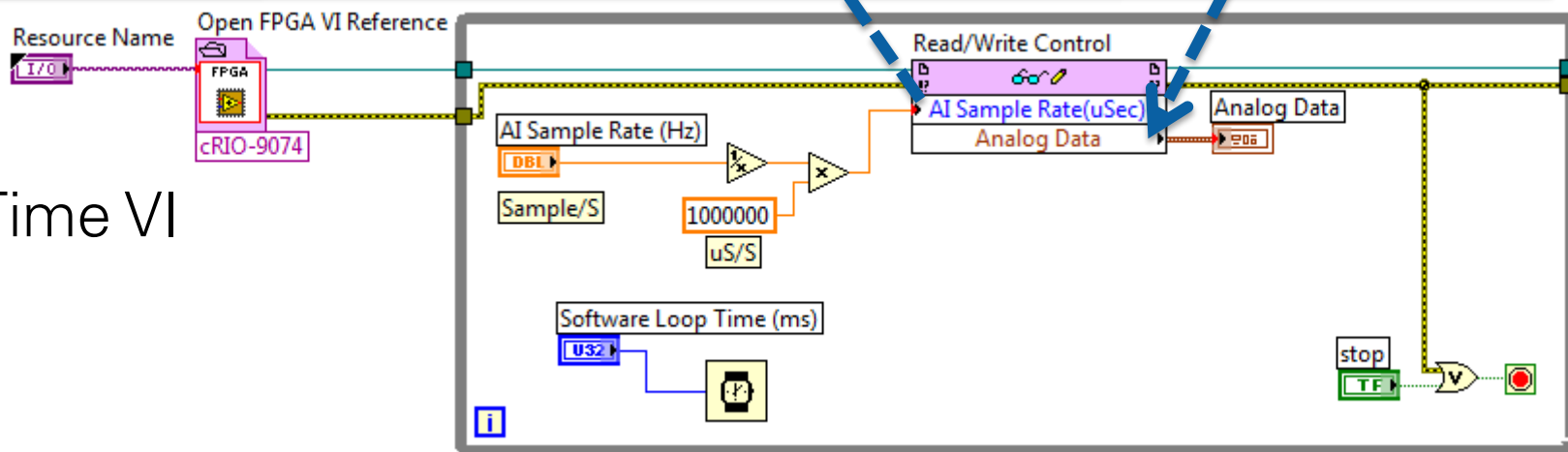


- The Read/Write Controls method is recommended for communicating current value data

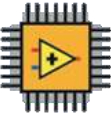
FPGA VI



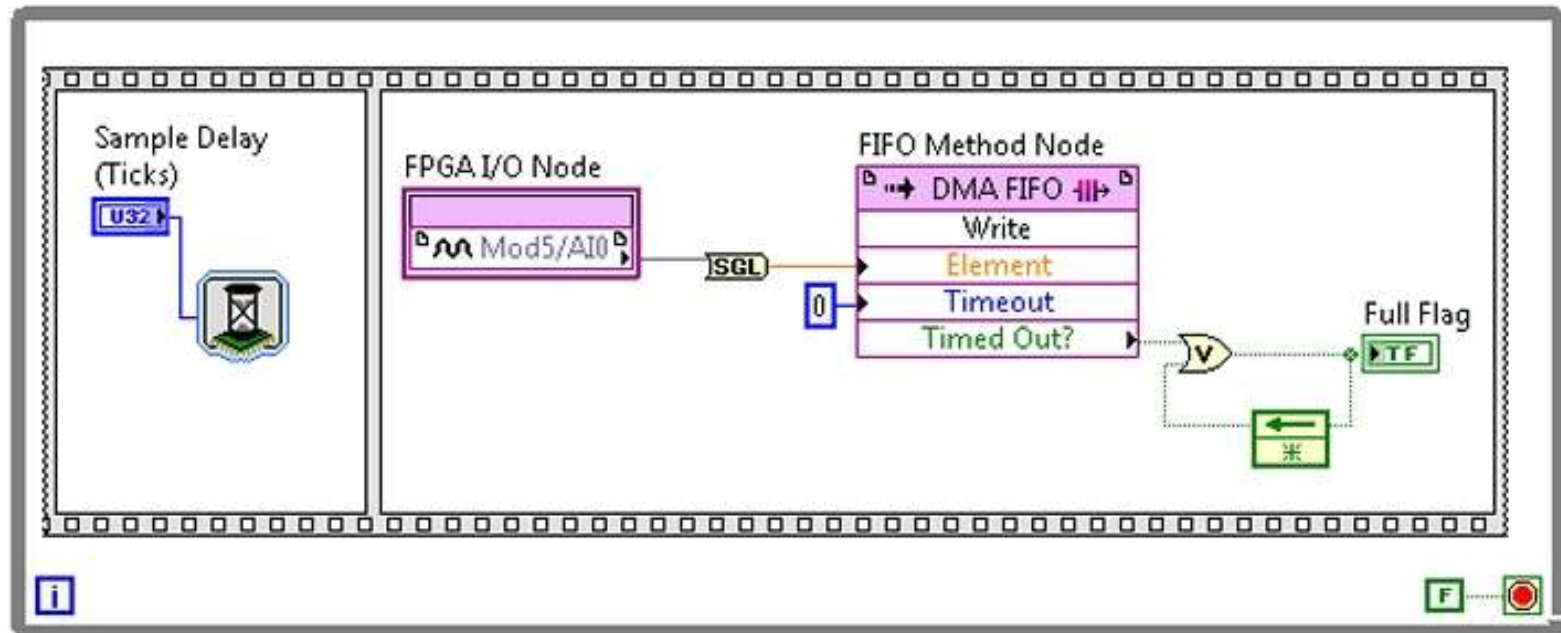
Real-Time VI



5 - Transferring Data to the RT Microprocessor



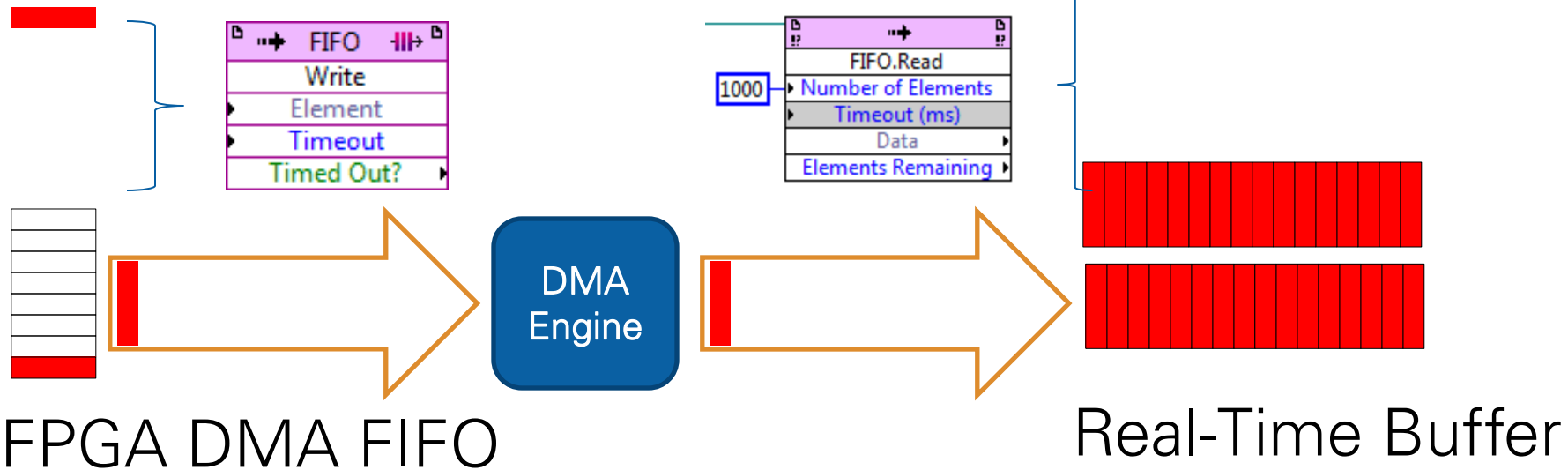
- Direct Memory Access (DMA) FIFOs are an efficient mechanism for streaming data from the FPGA to RTOS
- Does not involve processor resources
- Most RIO hardware targets have 3 dedicated DMA channels



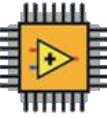
FPGA \leftrightarrow RT: DMA FIFOs



Data Element

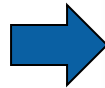
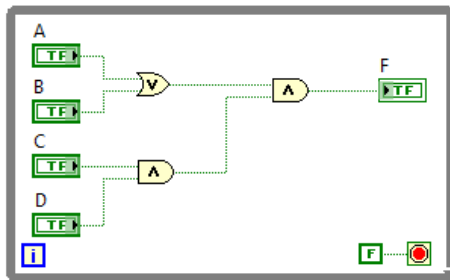


6 - Test and Compile



LabVIEW FPGA Code Compile VHDL through Xilinx

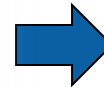
FPGA Logic Implementation



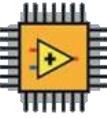
```
-- Process: synchronous logic
-- Then we keep track of what the digital input was on the previous
-- clock cycle by inserting another flip flop
previousDigitalInputFF;
process( areset, Clk )
begin
    if areset then
        previousDigitalInputFF <= false;
    elsif rising_edge(Clk) then
        previousDigitalInputFF <= coDigitalInput;
    end if;
end process previousDigitalInputFF;

-- Then we have a little combinational logic to detect a rising edge
risingEdgeDetected <= coDigitalInput and not previousDigitalInputFF;

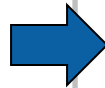
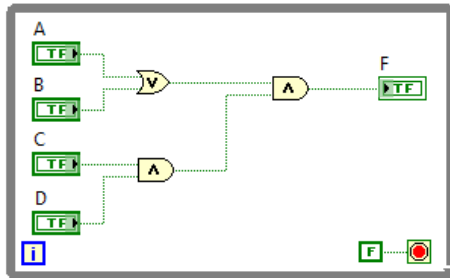
-- And finally we have a register that increments when that rising
-- edge is detected.
counterRegister;
process( areset, Clk )
```



6 - Test and Compile



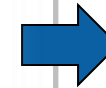
LabVIEW FPGA Code Compile VHDL through Xilinx



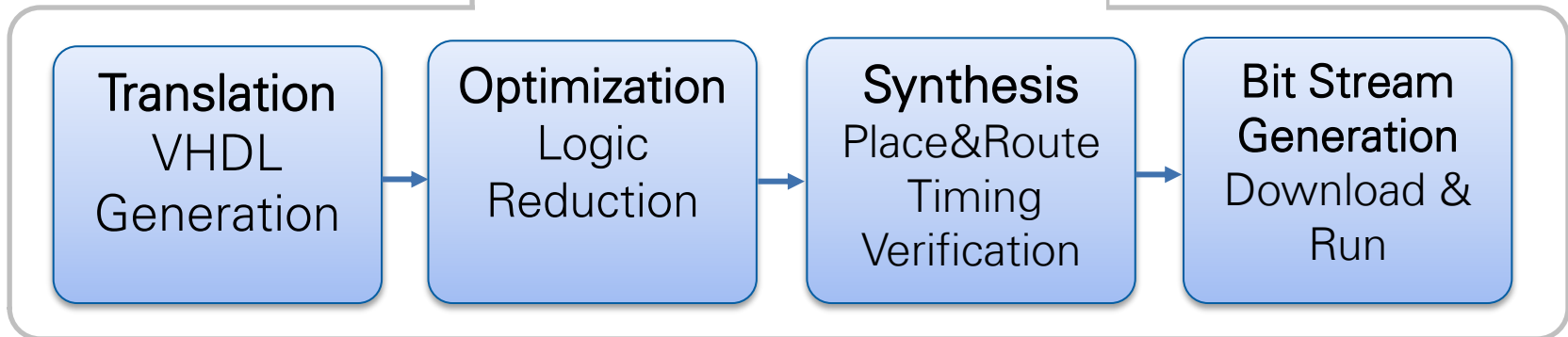
```
-- Process: Synthesizable logic
-- When we keep track of what the digital input was on the previous
-- clock cycle by inserting another flip flop
previousDigitalInputFF;
process( areset, Clk )
begin
    if areset then
        cprevDigitalInput <= false;
    elsif rising_edge(Clk) then
        cprevDigitalInput <= coDigitalInput;
    end if;
end process previousDigitalInputFF;

-- Then we have a little combinatorial logic to detect a rising edge
risingEdgeDetected <= coDigitalInput and not cprevDigitalInput;

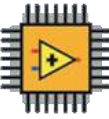
-- And finally we have a register that increments when that rising
-- edge is detected.
counterIncrement;
counter <= counter + 1;
```



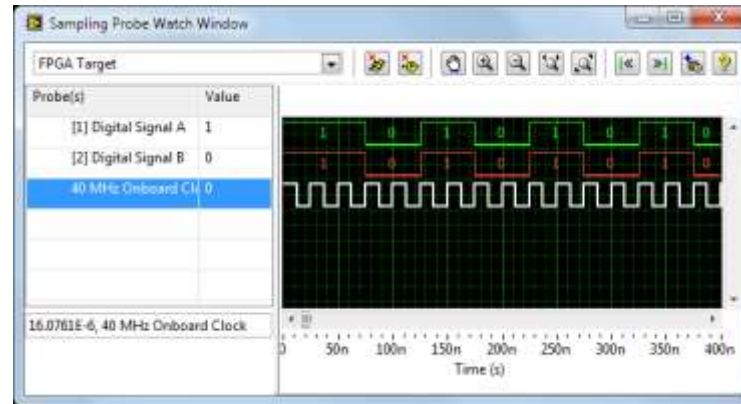
FPGA Logic Implementation



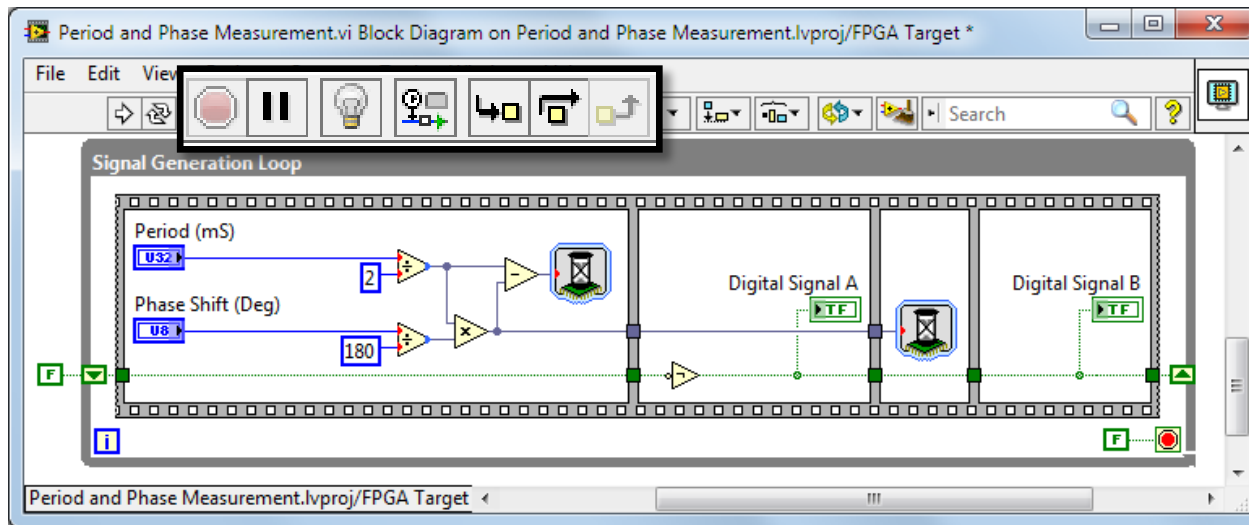
6 - Test and Compile Demonstration



Debug with standard
LabVIEW features in
simulation

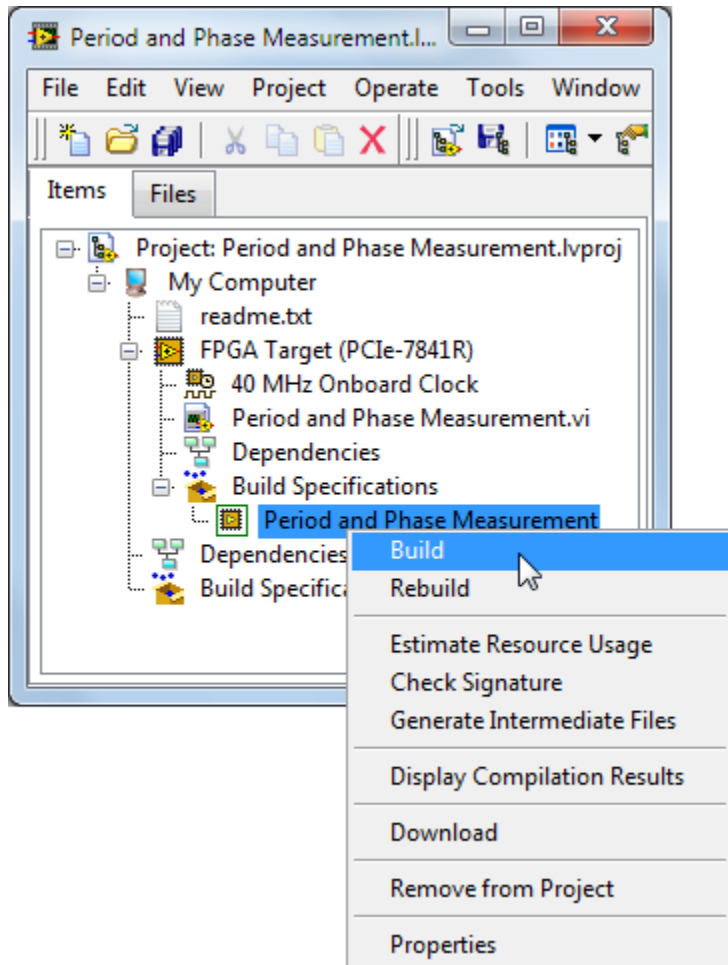
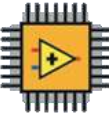


Verify signal
timing with digital
waveform probe



Simulator offers
bit-true functional
simulation

One-Click Deployment and Compilation



Development
PC



Compile
Server and
Workers



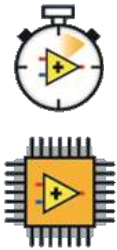
High-
Performance
Cloud





LabVIEW Real-Time

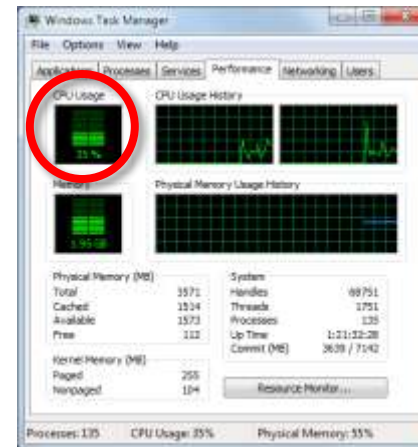
7 - Working with Constrained Resources



Disk Space



RAM



CPU Bandwidth



FPGA Gates

EFFECT

Lost Data

Crash

Starvation

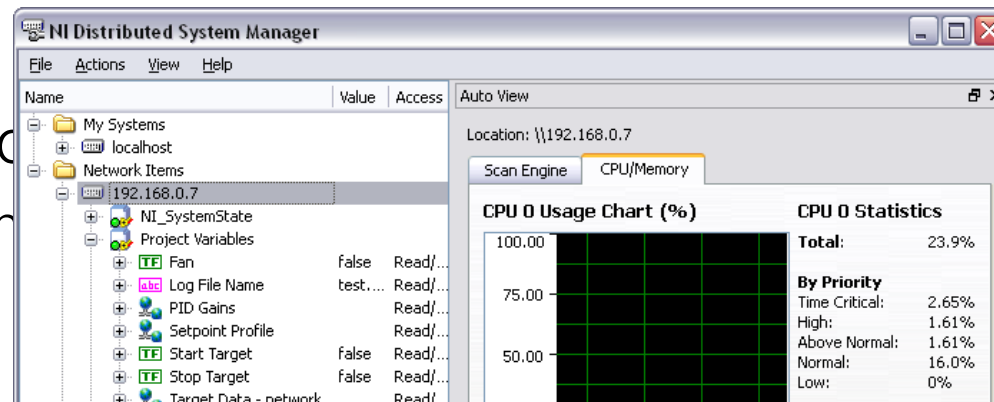
Compile Failure

7 - Monitoring Constrained Resources

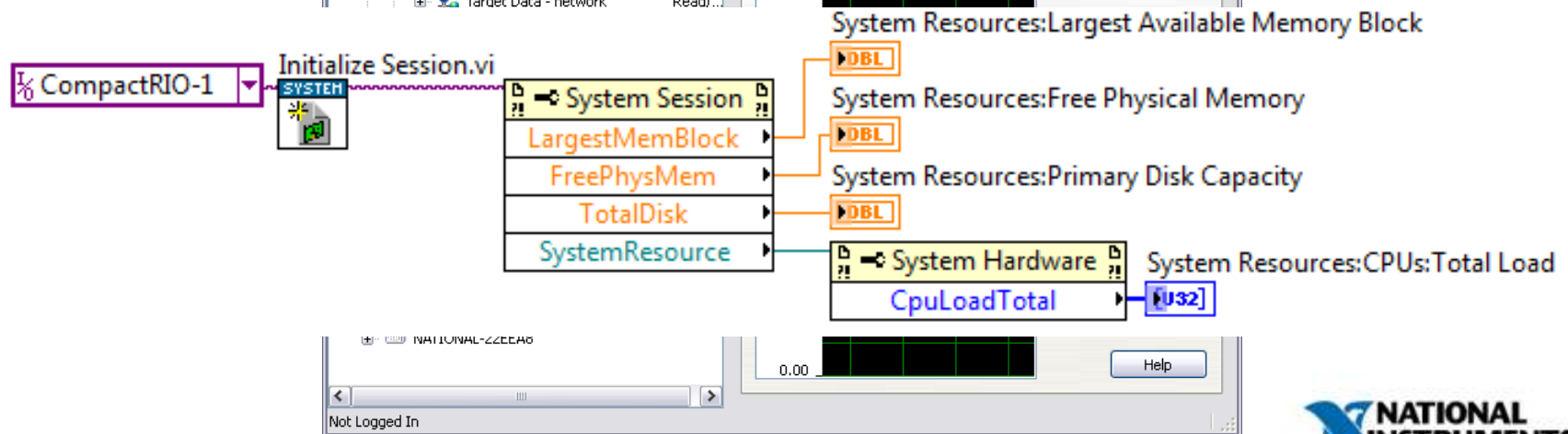


- Ensure that CPU does not exceed 70% (ideal)
- Distributed System Manager for CPU and Memory Usage
 - System State Publisher must be installed to controller

- System Configuration
- Program



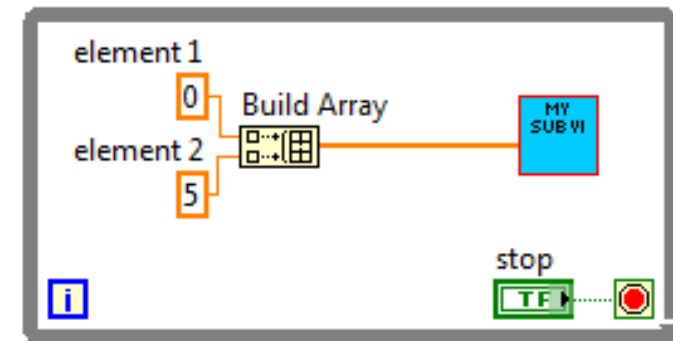
metrics



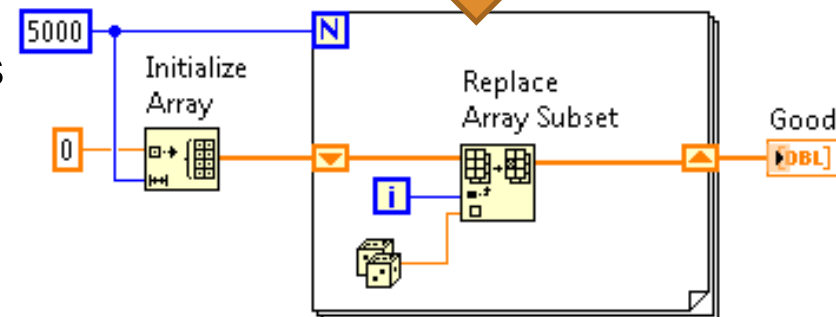
7 - Working with Constrained Resources



- LabVIEW Real-Time programs should attempt to minimize hidden memory allocation
- Common sources:
 - Queues without fixed size
 - Variable sized arrays
 - Variable sized strings
 - Variants
- Use the *In Place Element Structure*
- Use **Show Buffer Allocations** tool
 - Tools»Profile»Show Buffer Allocations
- Real-Time Execution Trace Toolkit
 - Optimize performance and analyze down to thread level



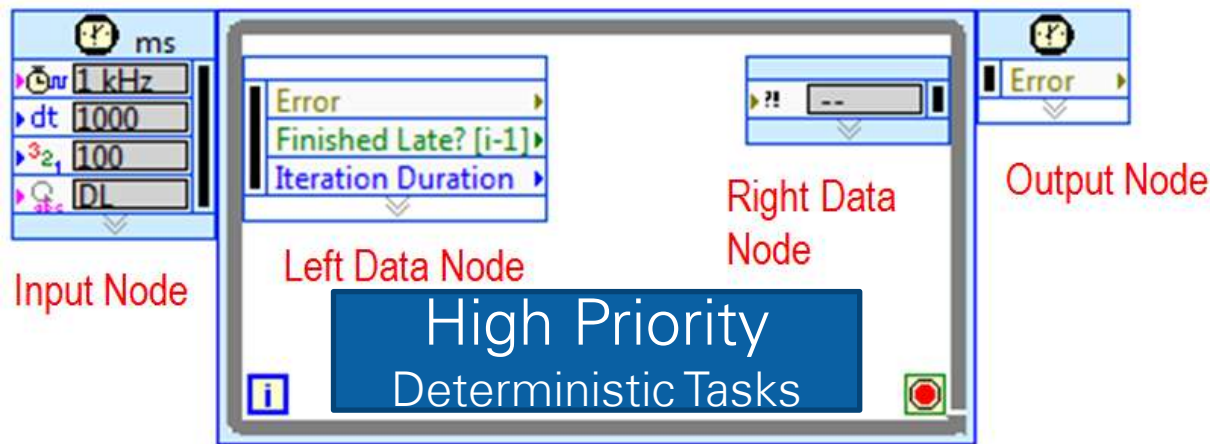
Correct



8 - Implementing High Priority Tasks



- Choose a loop structure based on the priority of your task
- Timed Loops execute at a higher priority than While Loops
 - Above High but below Time Critical
- Using multiple Timed Loops can add complexity and overhead

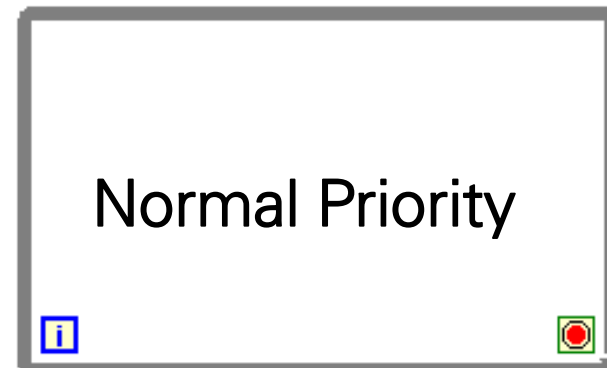


Timed Loop



While Loop

8 - Implementing High Priority Tasks



Deterministic Operations:

- Closed Loop Control
- Decision Making Logic
- Shutdown logic
- FPGA Acquisition

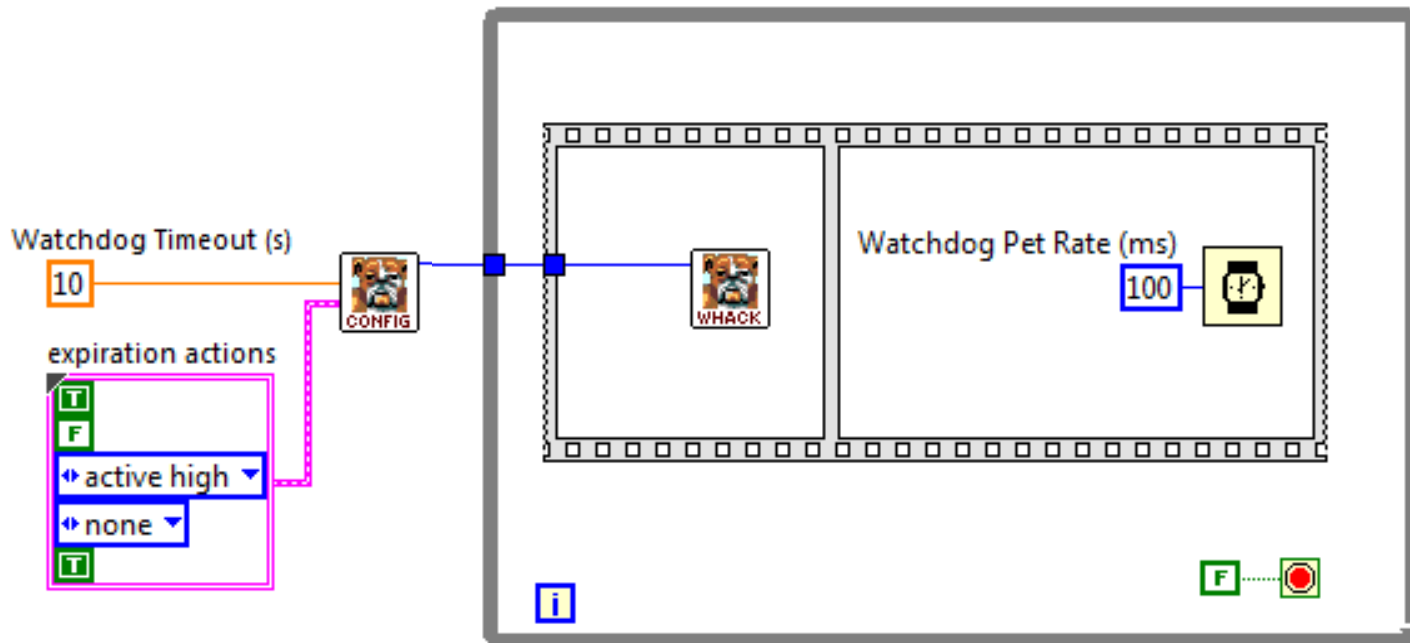
Non-Deterministic Operations:

- File I/O
- Network or serial communication
- Process involving dynamic memory allocation
 - Dynamically sized data types
- Calls to nondeterministic drivers or libraries

9 - LabVIEW RT Watchdog

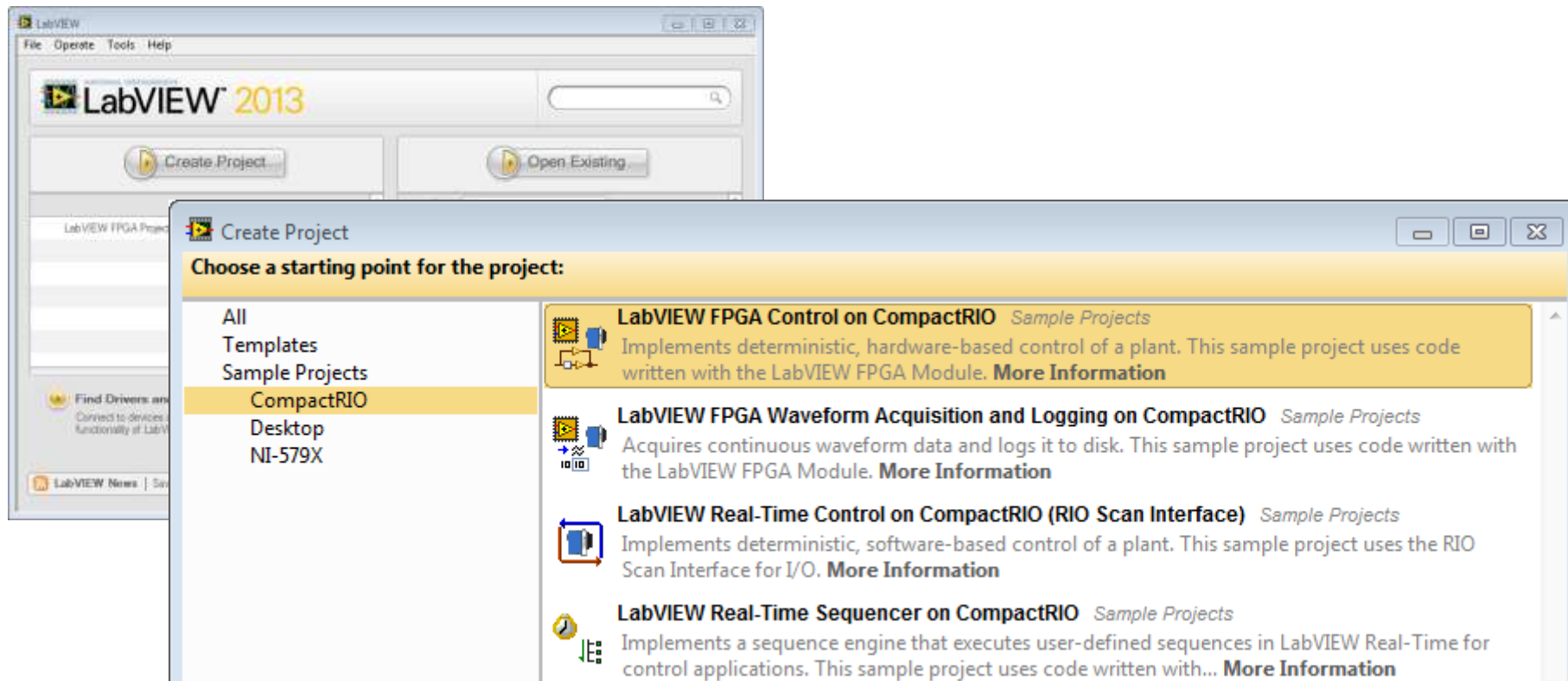


- Reboots system if the watchdog process is starved
- This could occur if:
 - CPU usage hits 100%
 - Software is hanging because it ran out of memory



Additional Resources

LabVIEW 2013 CompactRIO Sample Projects



NI LabVIEW for CompactRIO Developer's Guide

NI LabVIEW for CompactRIO Developer's Guide

Recommended LabVIEW Architectures and Development Practices
for Control and Monitoring Applications

ni.com/compactriodevguide

High Performance RIO Developer's Guide: ni.com/hprioguide

[LabVIEW Help: RT Best Practices Portal](#)

Training & Certification Path for Embedded Systems

System Prototyping

Development & Deployment

