
NATIONAL
INSTRUMENTS™

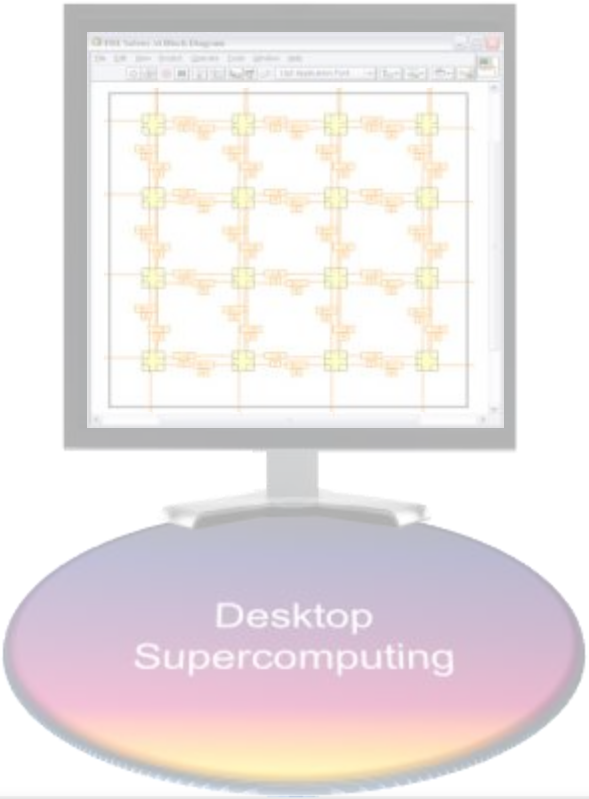
Introduction: Using FPGAs in Test Applications

Remco Krul

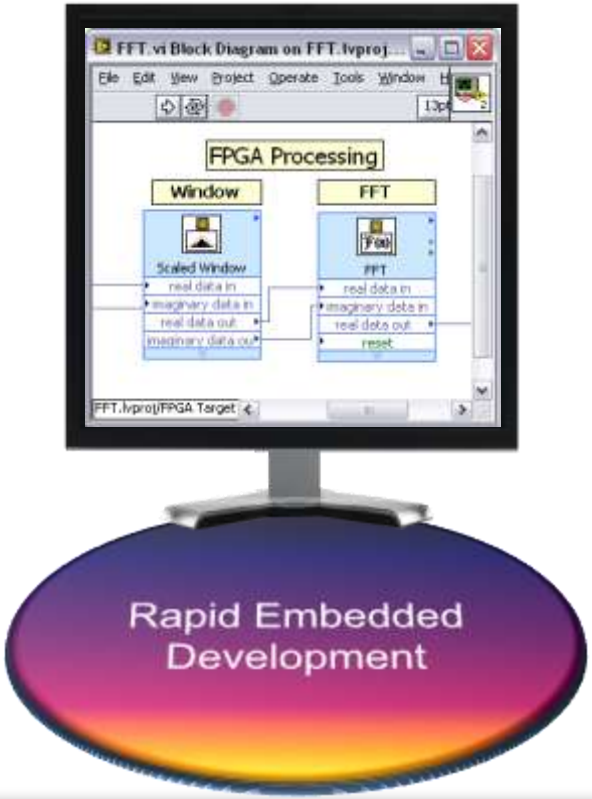
Marketing manager

Key Technologies

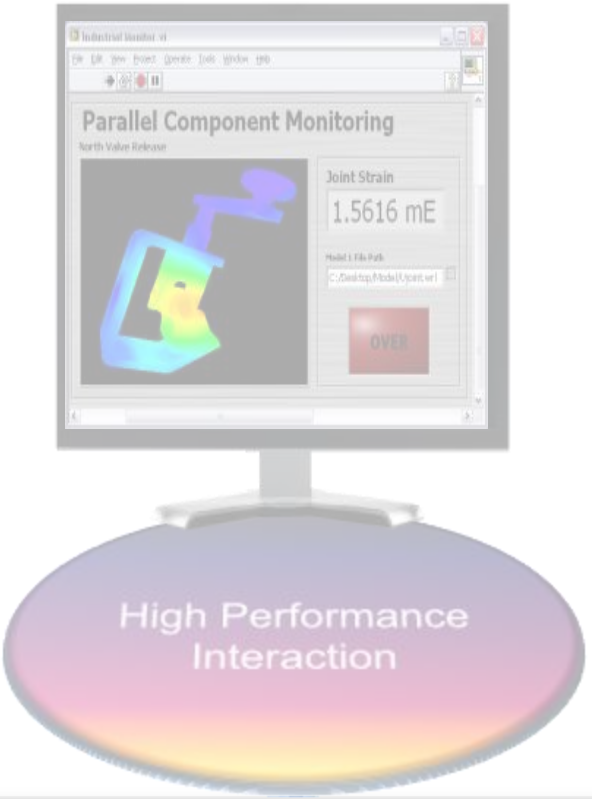
Multicore

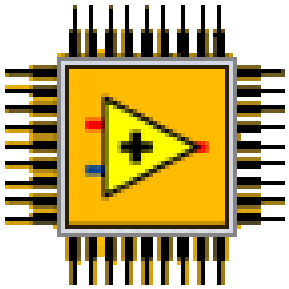


FPGA



Modular I/O



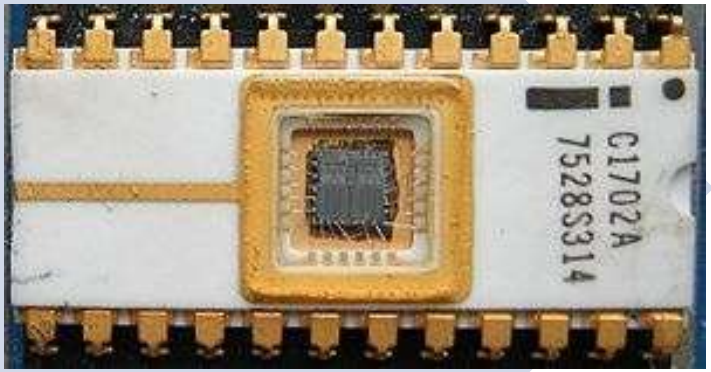


History of FPGAs

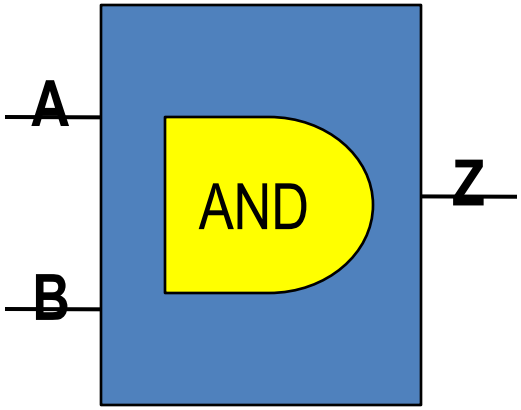
History of the FPGA

1956
ROM Invented

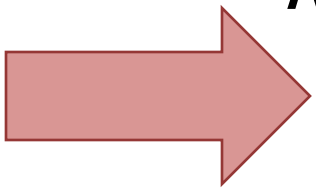
1970
TI uses IBM memory to implement logic

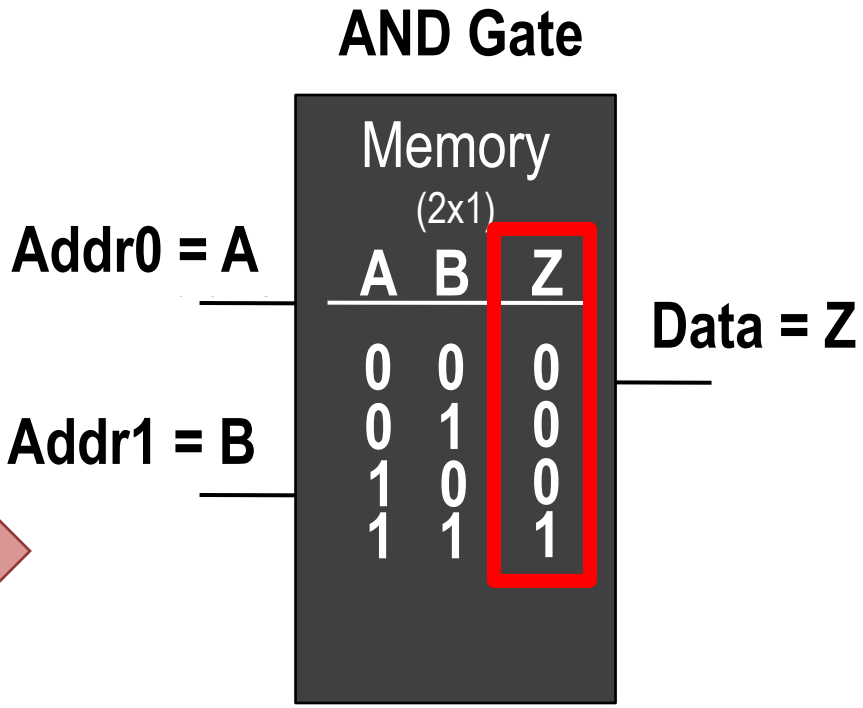


FPGA History

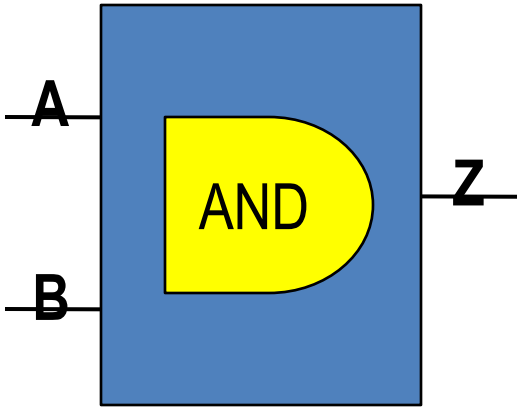


A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

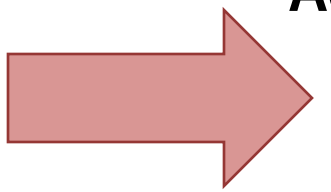




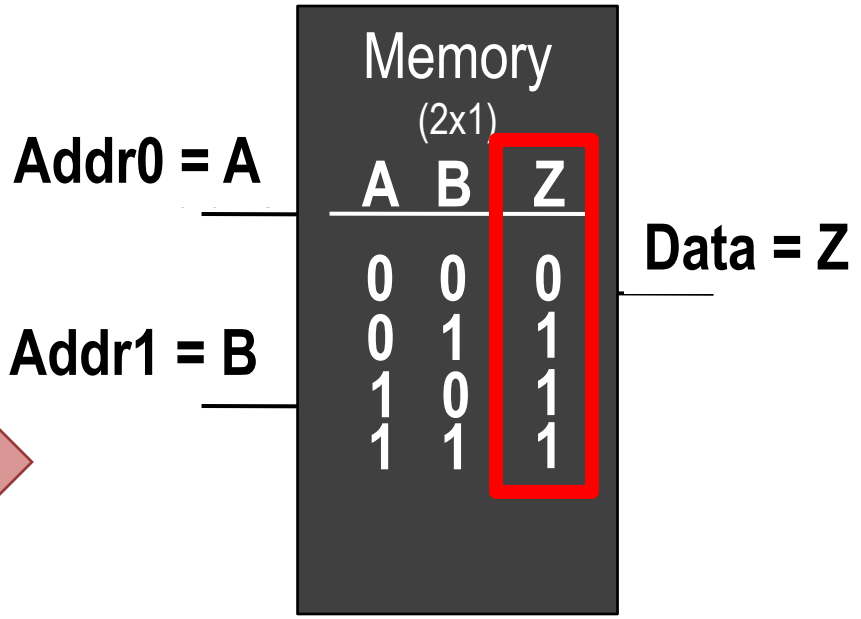
FPGA History



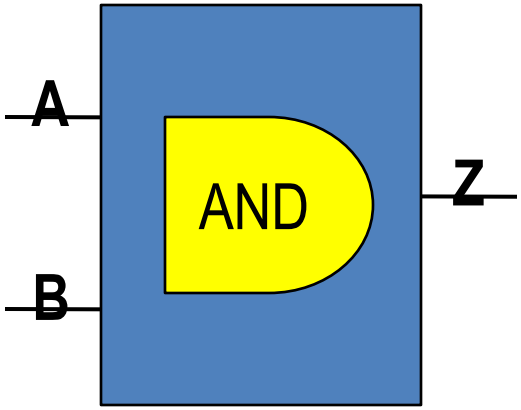
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



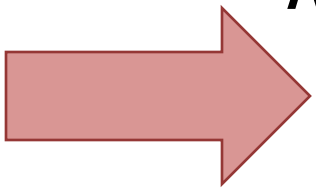
OR Gate



FPGA History



A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



XOR Gate

Addr0 = A

Addr1 = B

Memory
(2x1)

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

Data = Z

History of the FPGA

Ross Freeman

1956
ROM Invented

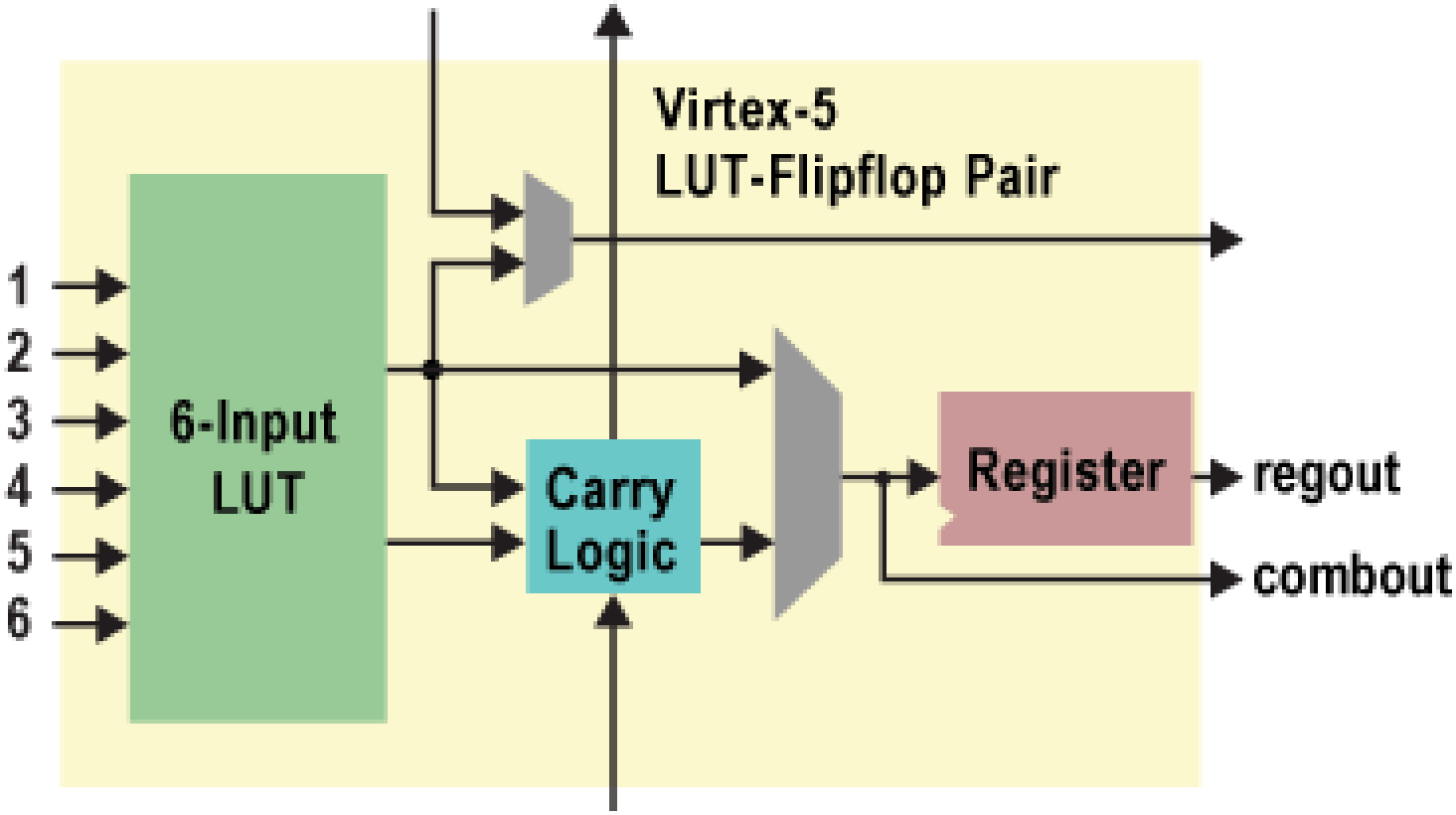
1970
TI uses IBM memory to implement logic

1971
EPROM invented

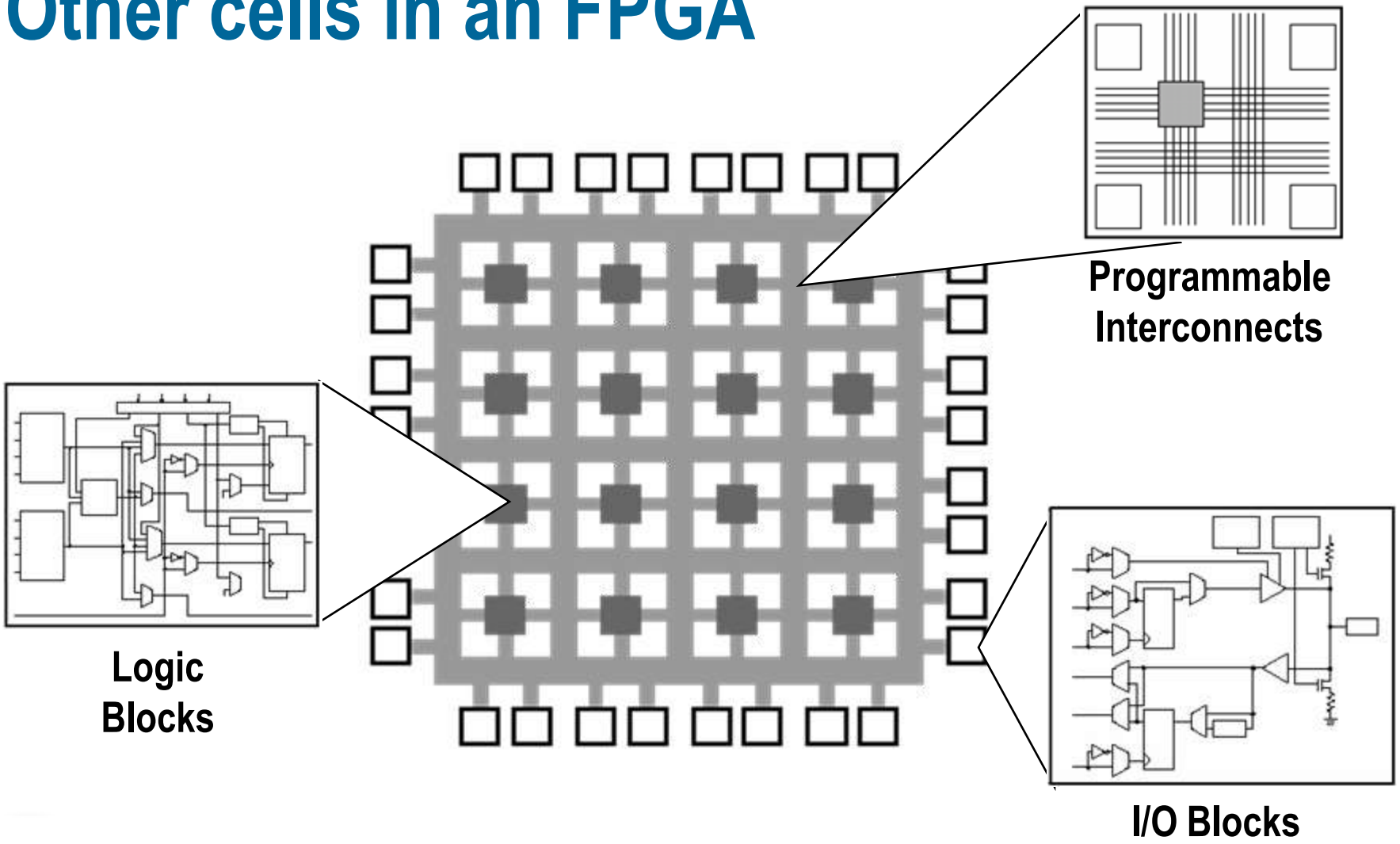
1973
Xilinx invents FPGA
64, 3-input macrocells

2009
Xilinx announces Virtex-6
176,000 6-input logic cells

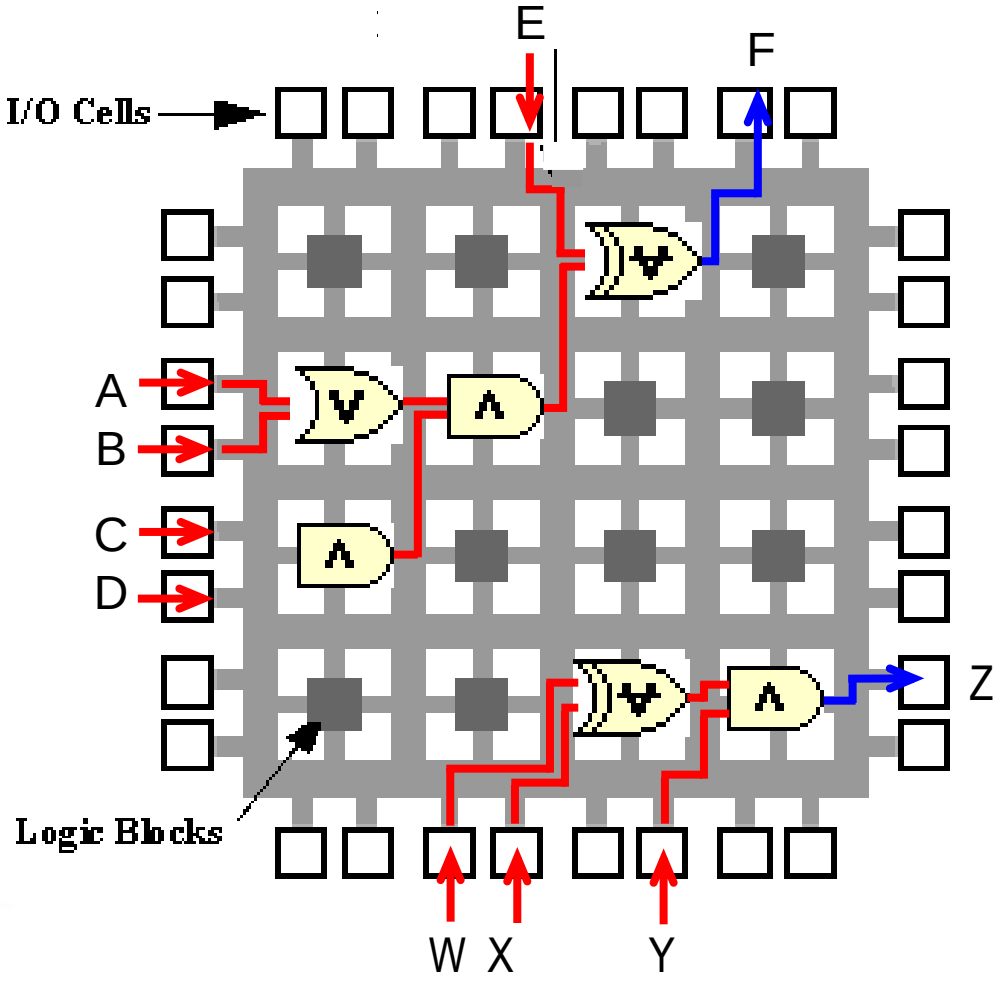
Modern FPGA Logic Cell (Virtex 5)

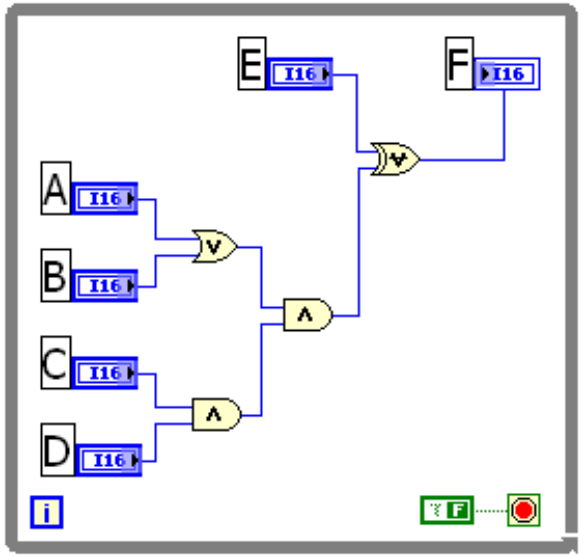


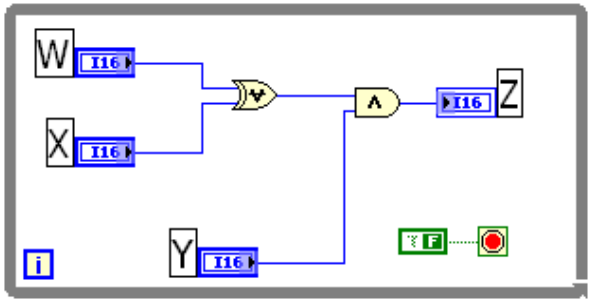
Other cells in an FPGA



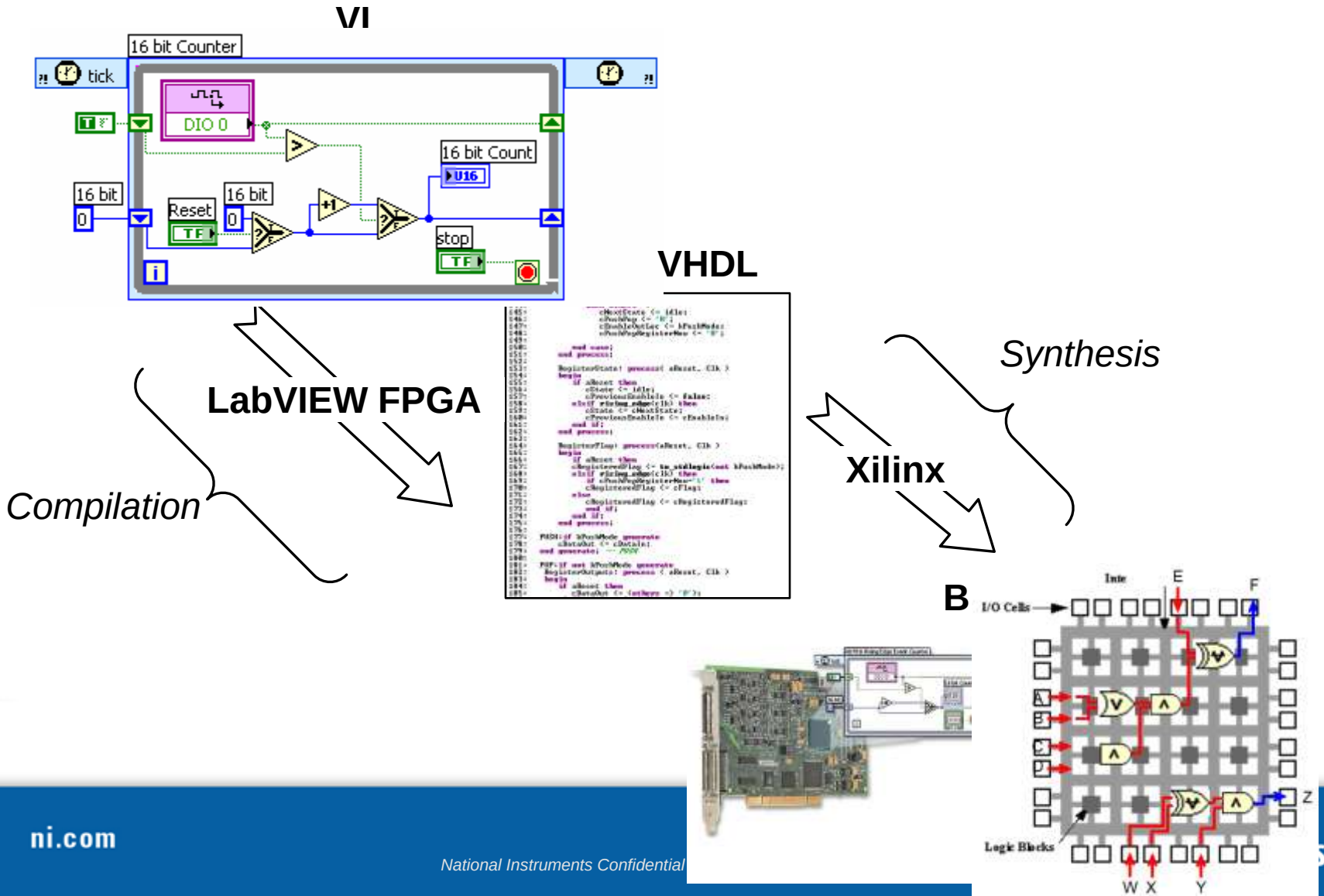
Mapping LabVIEW to FPGAs



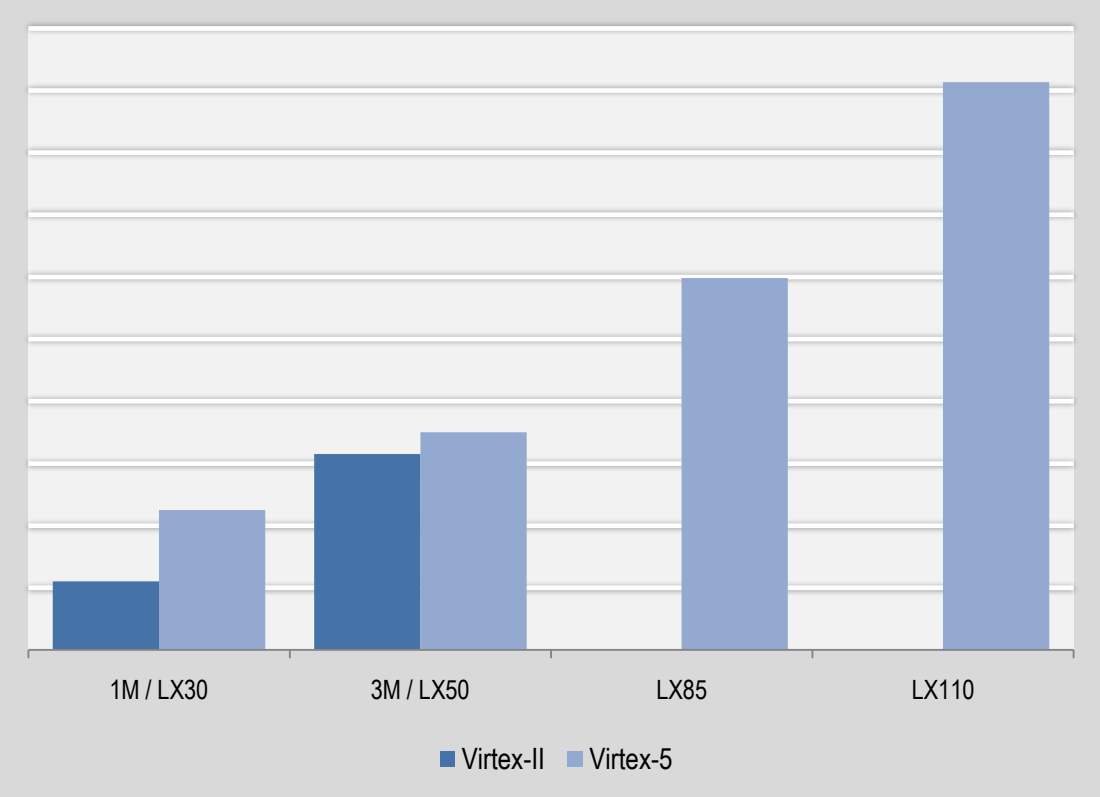




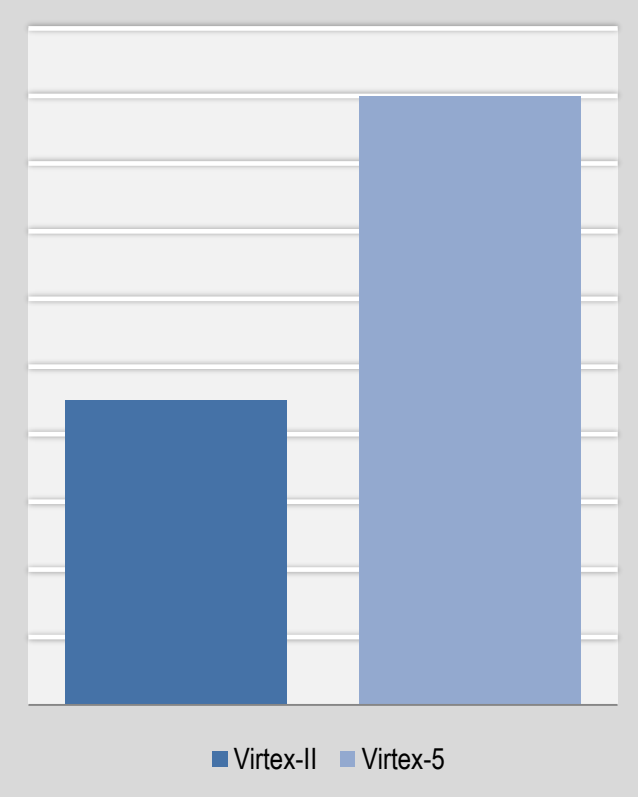
Compiling G for a FPGA



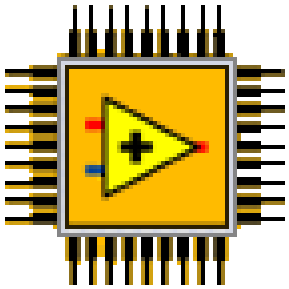
FPGA Families / Sizes



FPGA Logic Resources



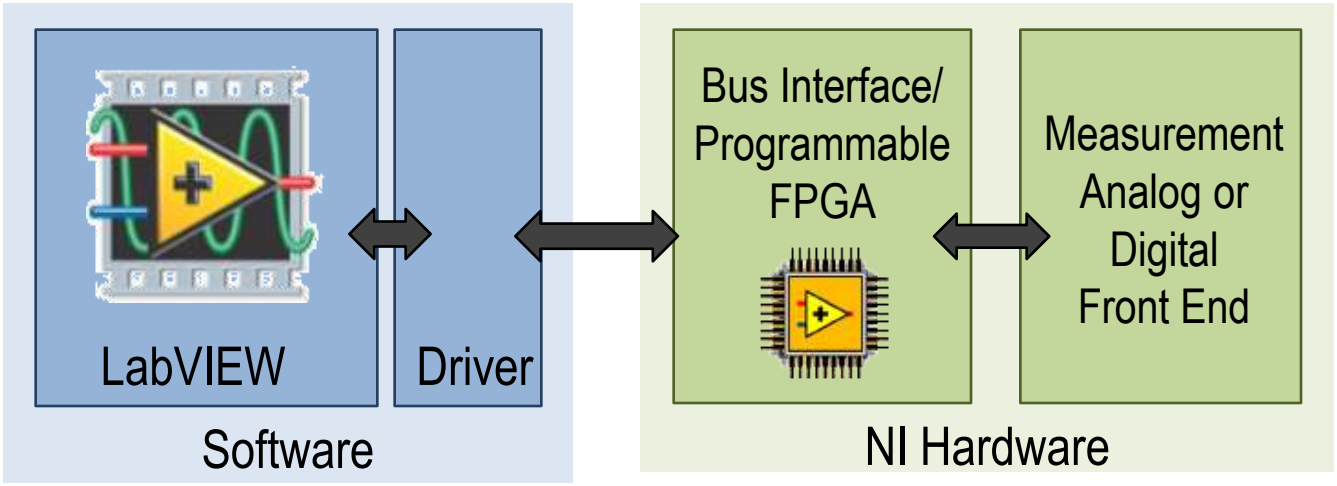
FPGA Execution Speed



FPGAs and Virtual Instrumentation

Virtual instrumentation model

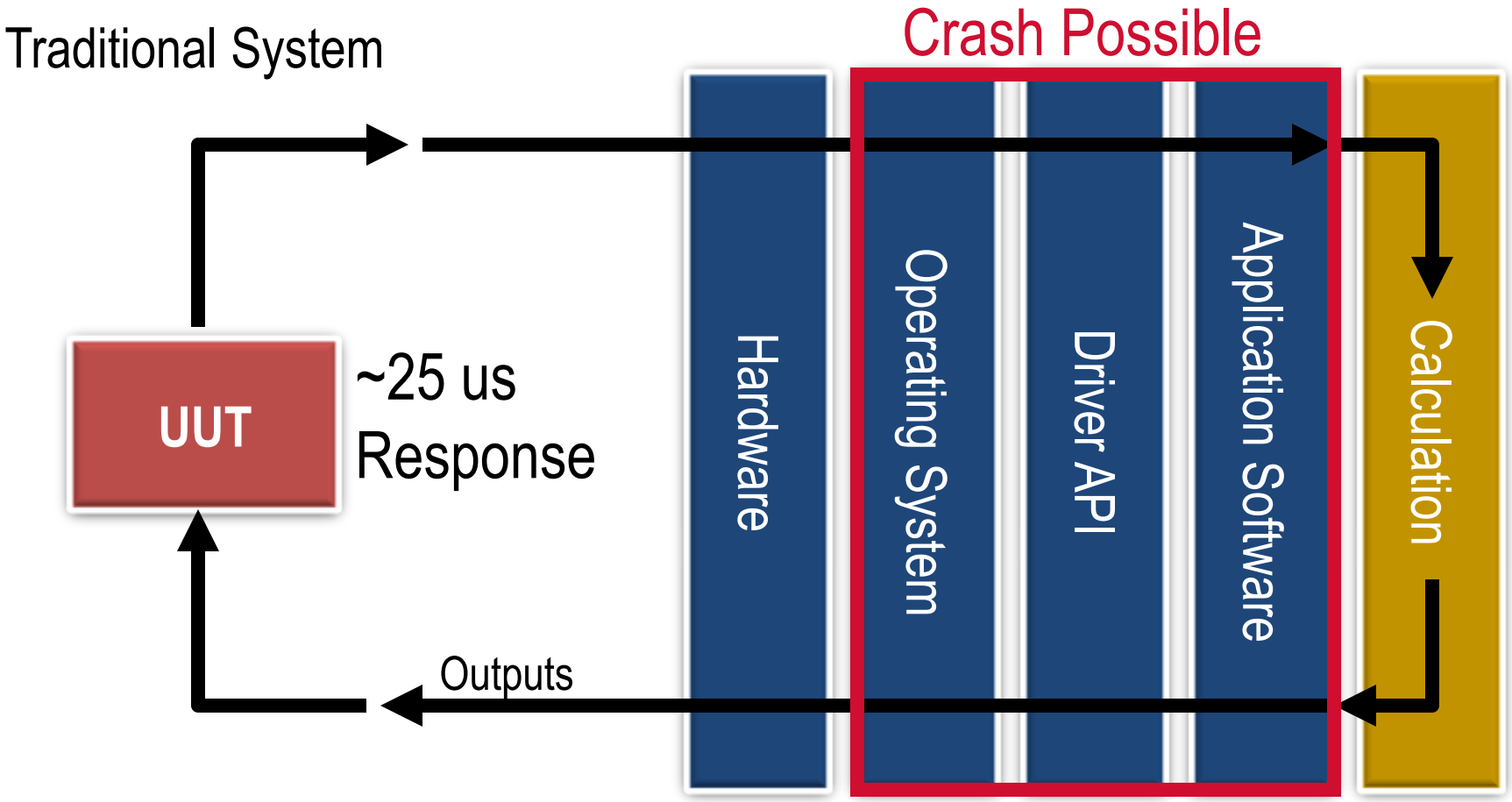
LabVIEW FPGA System Model



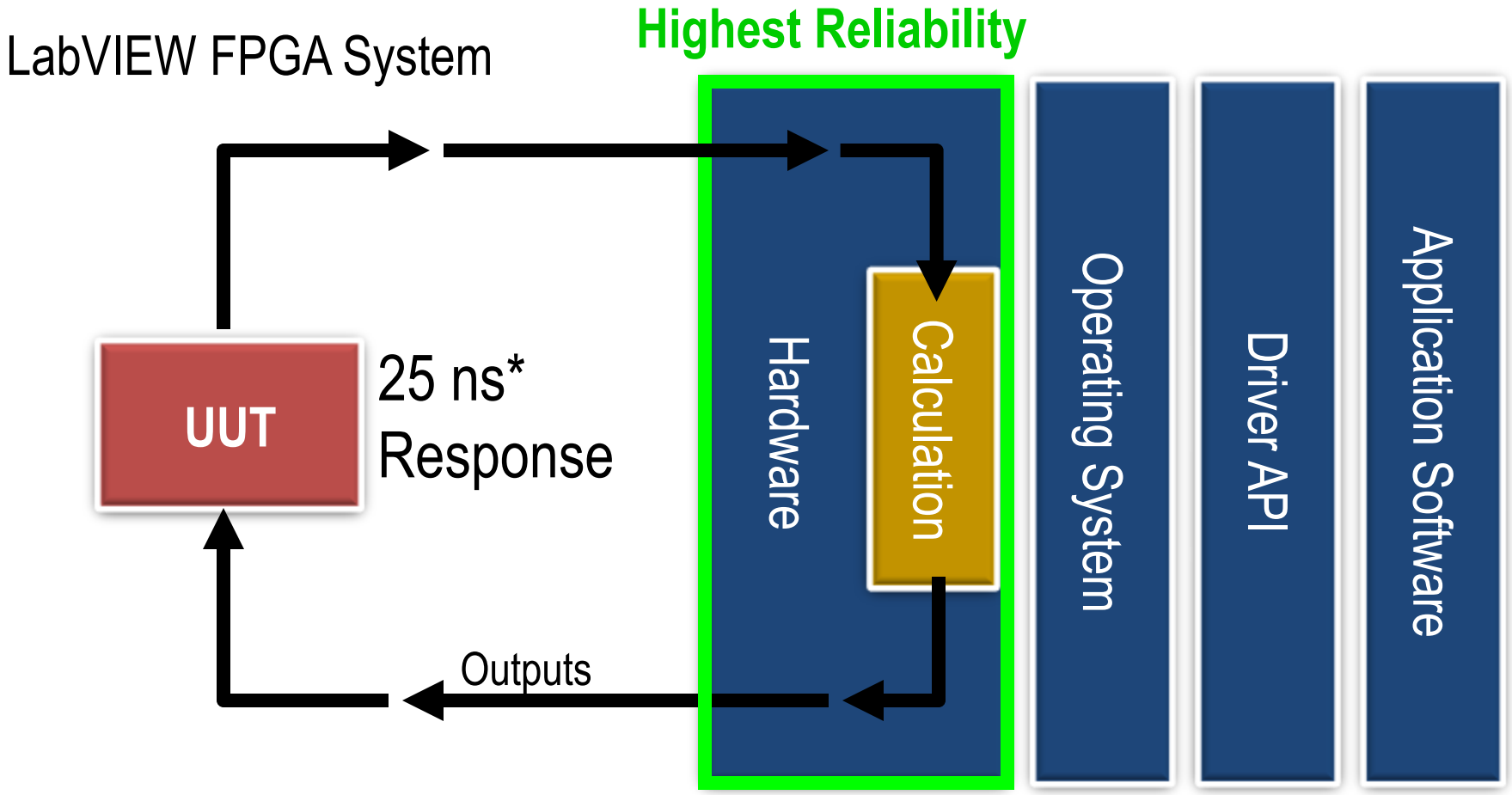
Importance of FPGAs in Test Systems

- ***High Reliability*** – Designs become a custom circuit
- ***High Determinism*** – Runs algorithms at deterministic rates down to 5 ns
- ***True Parallelism*** – Enables parallel tasks and pipelining and thus reduces test times.
- ***Reconfigurable*** – Create new and alter existing task-specific personalities

Robustness (SW applications)



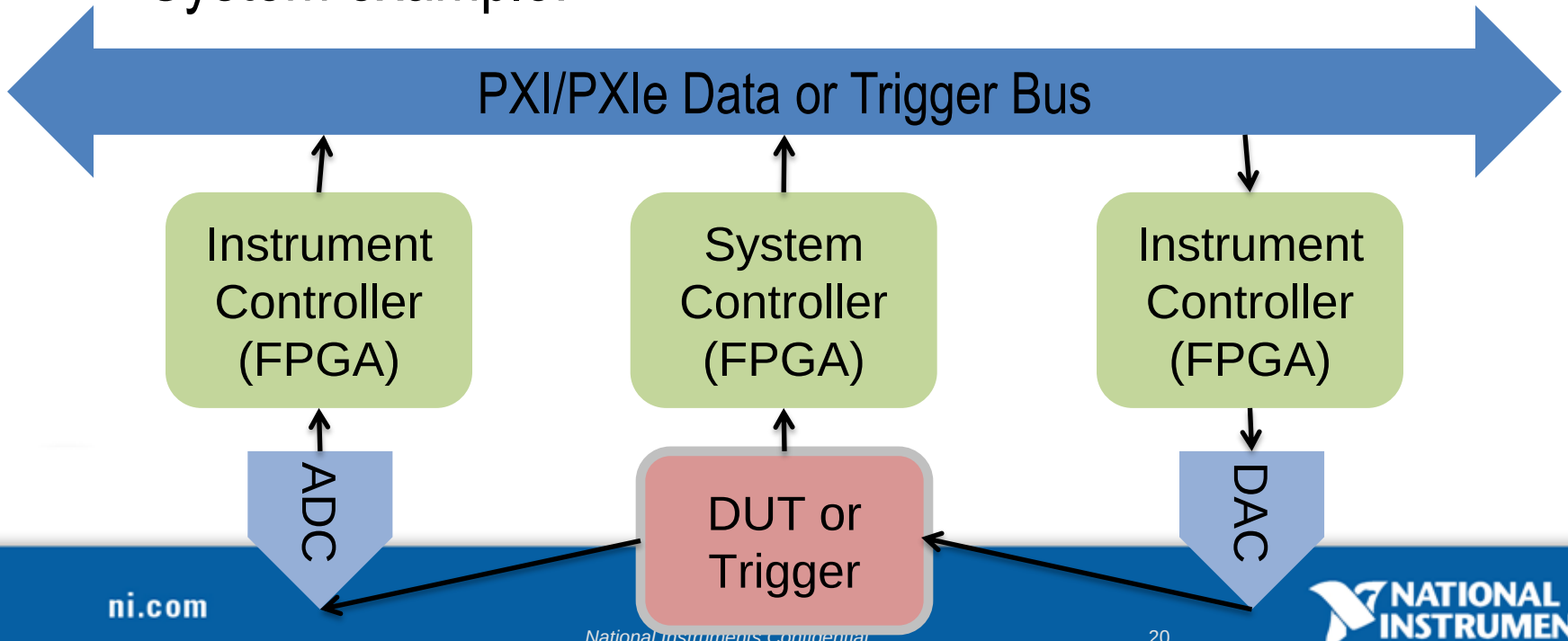
Robustness (Hardware applications)



* Faster response for 80 and 120 MHz clocks

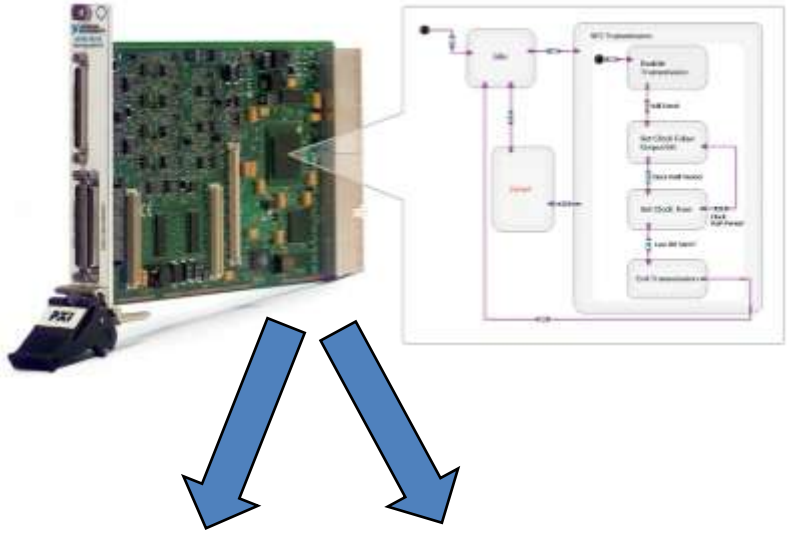
System Control

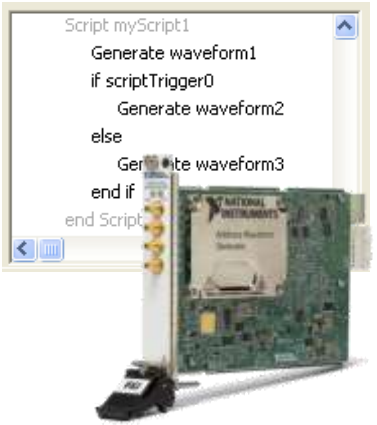
- Goal: move the system timing and decision making of your system from software to hardware
- Benefit: increased determinism, faster response
- System example:



System Control Example

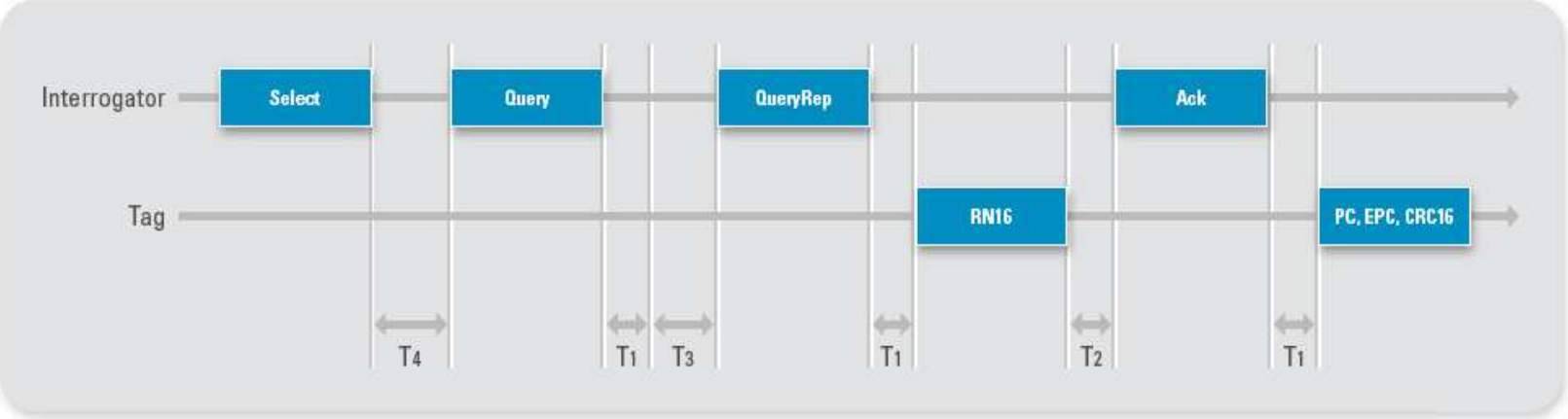
- State machine deployed to FPGA for system control
- Triggers sent from FPGA to modular instruments
  - Instruments programmed with measurement lists and/or scripts
  - Use PXI trigger lines for communication





Closed-Loop I/O Example: RFID

- Testing an RFID tag requires emulating the tag reader
  - Interrogate and respond to tag within microseconds
- Coding/decoding, modulation/demodulation, and decision making must be done in hardware to meet timing

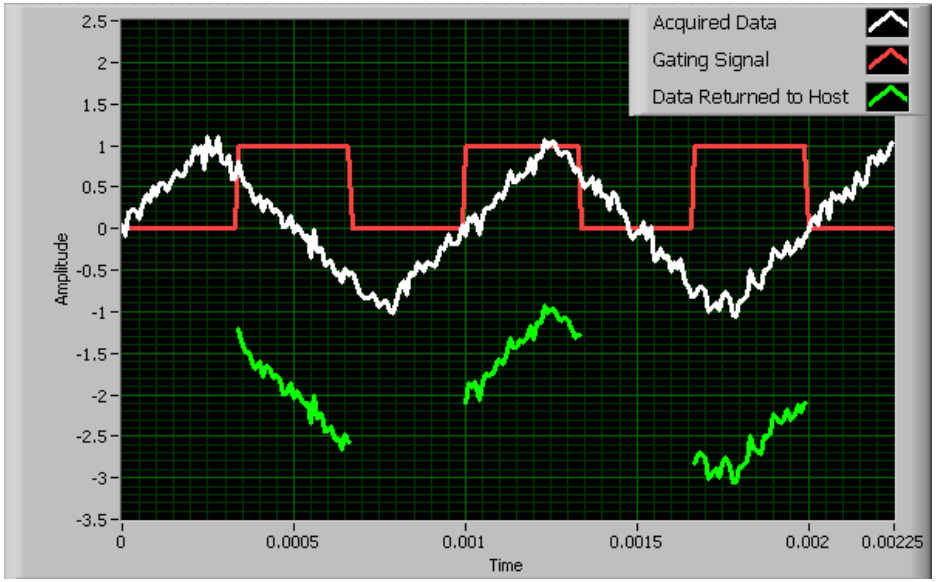


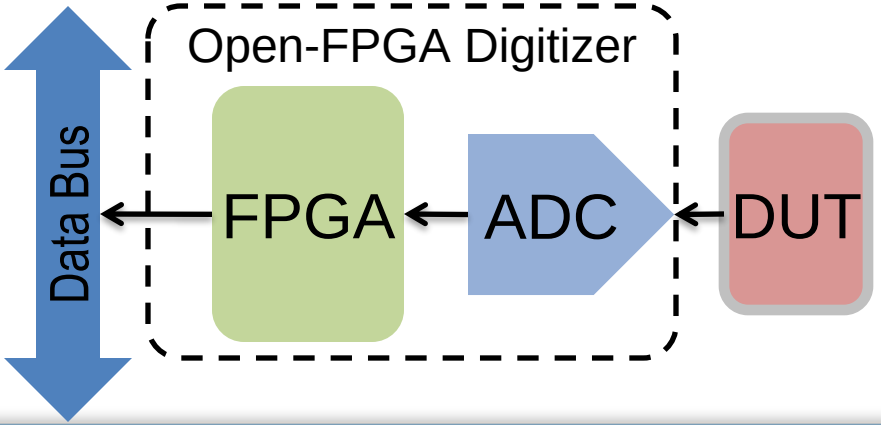
Data Reduction or Signal Processing

- Goal: perform more processing on the instrument instead of the host
- Benefit: free data bus for other transfers, reduce burden on CPU, get more done consuming less resources
- Examples
  - Filtering
  - Peak-detect
  - Fast Fourier Transforms (FFTs)
  - Custom triggering
  - Algorithmic pattern generation

Data Reduction Example

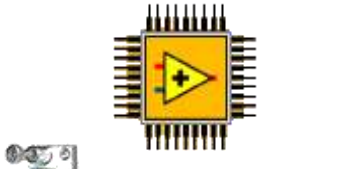
- Continuous acquisition with a digitizer; only need data that arrives between triggers
- Rearm time of digitizer's native trigger too slow
- Use FPGA to eliminate data outside trigger window before bus transfer





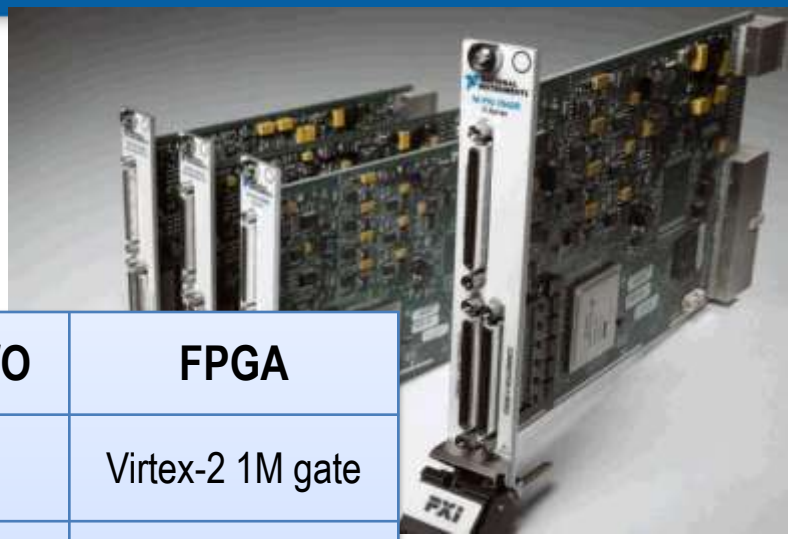
NI LabVIEW FPGA Targets

- R Series DAQ
- NI IF-RIO
- NI FlexRIO
- NI CompactRIO
- Compact Vision Systems





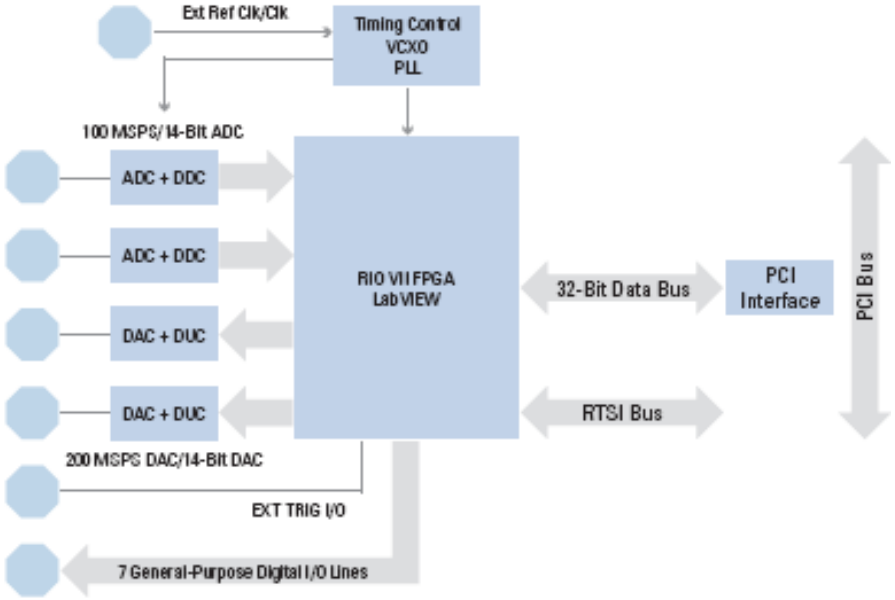

NI PXI RIO Products



	Analog Input	Analog Output	Digital I/O	FPGA
PXI-7831R	8ch, 16-bit, 200kS/sec	8ch, 16-bit, 1MS/sec	96 lines	Virtex-2 1M gate
PXI-7833R	8ch, 16-bit, 200kS/sec	8ch, 16-bit, 1MS/sec	96 lines	Virtex-2 3M gate
PXI-7841R	8ch, 16-bit, 200kS/sec	8ch, 16-bit, 1MS/sec	96 lines	**Virtex-5 LX30**
PXI-7842R	8ch, 16-bit, 200kS/sec	8ch, 16-bit, 1MS/sec	96 lines	**Virtex-5 LX50**
PXI-7851R	8ch, 16-bit, **750kS/sec**	8ch, 16-bit, 1MS/sec	96 lines	**Virtex-5 LX30**
PXI-7852R	8ch, 16-bit, **750kS/sec**	8ch, 16-bit, 1MS/sec	96 lines	**Virtex-5 LX50**
NEW PXI-7853R	8ch, 16-bit, **750kS/sec**	8ch, 16-bit, 1MS/sec	96 lines	**Virtex-5 LX85**
NEW PXI-7854R	8ch, 16-bit, **750kS/sec**	8ch, 16-bit, 1MS/sec	96 lines	**Virtex-5 LX110**

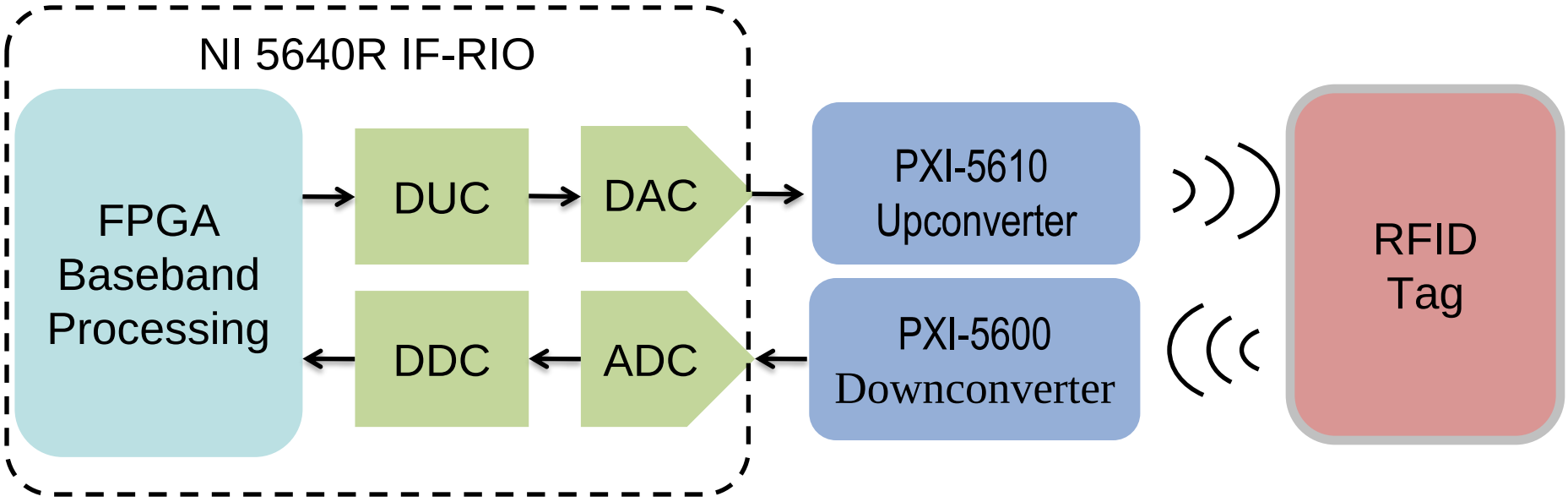
PCI-5640R IF-RIO Transceiver

- 2 input channels, 100 MS/s, 14 bits with built-in digital downconversion
- 2 output channels, 200 MS/s, 14 bits with built-in digital upconversion
- 250 kHz to 80 MHz analog frequency range
- Targeted at
  - RF stimulus response tests (dynamic test),
  - Inline modulation and demodulation (SDR),
  - User-defined IF applications

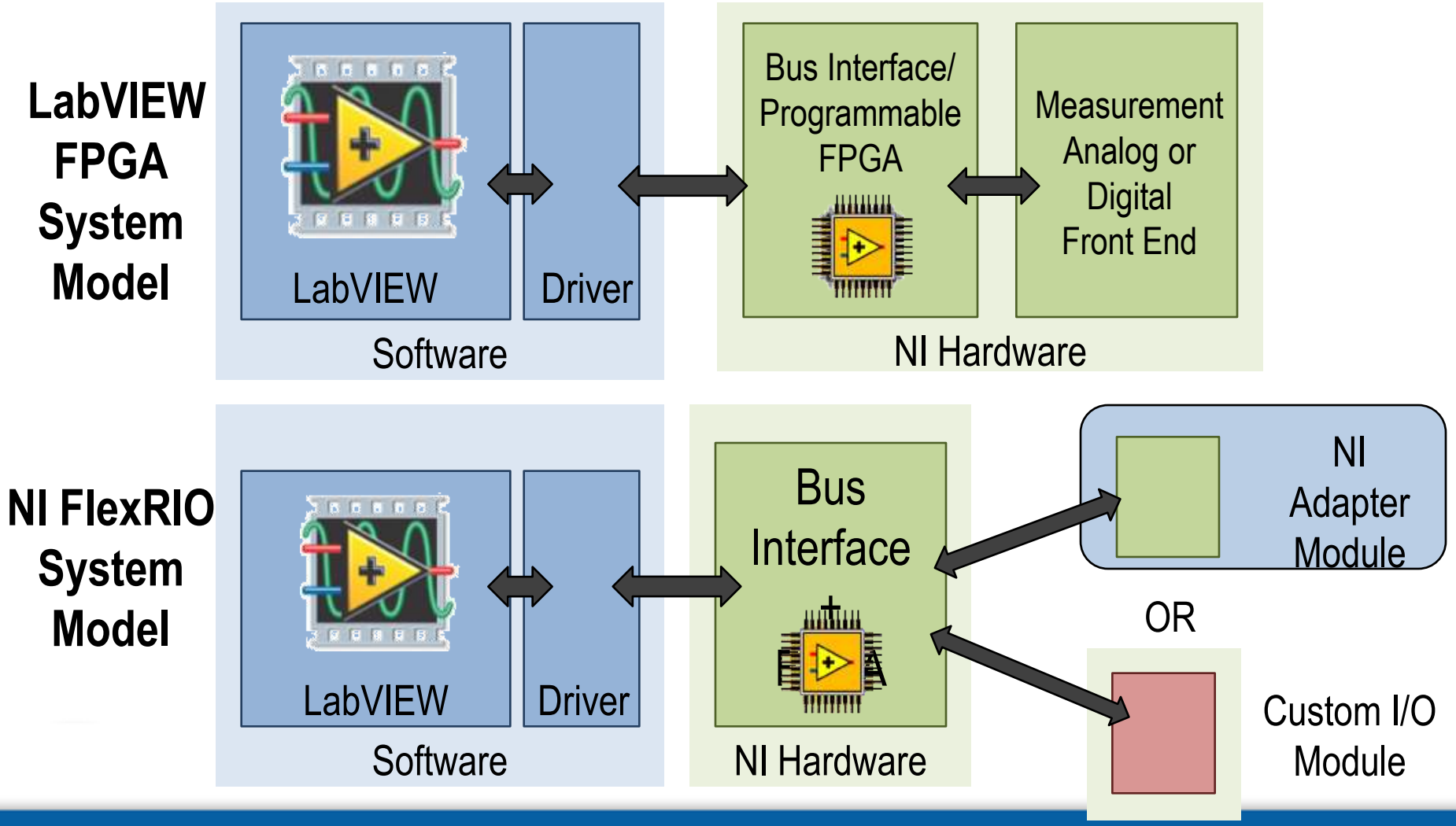


Closed-Loop I/O Example: RFID

- IF-RIO device can perform necessary processing
- Upconverter and downconverter condition the signal for the correct RF frequency of the tag



Progression of Models



NI FlexRIO

FlexRIO Adapter Module
- Interchangeable I/O
- Customizable by users
- Module Development Kit

FlexRIO FPGA Module
- Up to 132 channels
- Up to 1 Gb/s per pair
- Up to 128 MB of DDR2 DRAM

Mimas
Gigabit Ethernet Adapter

- Generate and analyze low level High Speed Ethernet data traffic in realtime.
- Supports channel simulation and fault injection features.
- High Speed 2,5Gbps fiber optical point-to-point data link.

The Mimas adapter includes:

- 2x PHY Ethernet circuits for 10/100 Mbit or 1Gbit (RJ45 with integrated magnetics)
- 2,5 Gbps XAUI tranciever with optical interface for RX/TX
- EEPROM for FlexRIO identification
- Glue logic (filters, drivers etc.)

Included software:

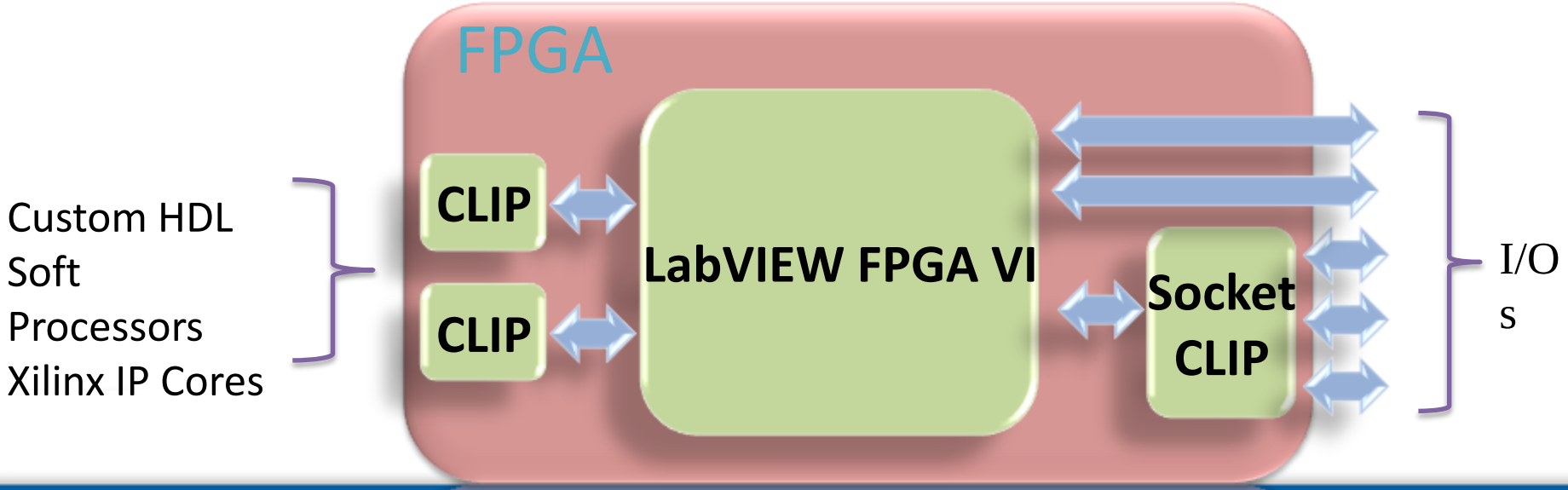
- Software code for the NI FlexRIO FPGA to handle:
  - MAC,
  - Ethernet frames and
  - Fault injection application.
- LabVIEW Software GUI for the system

Application areas:

- Customized IT based protocol
- Channel simulations
- Fault injection
- Customized MAC address adaptations

Component-Level IP (CLIP)

- Use any HDL IP on LabVIEW FPGA targets
- Access full capability of FPGA technology
- Use in parallel with LabVIEW FPGA VIs
- Examples: soft processors, IP cores, custom HDL



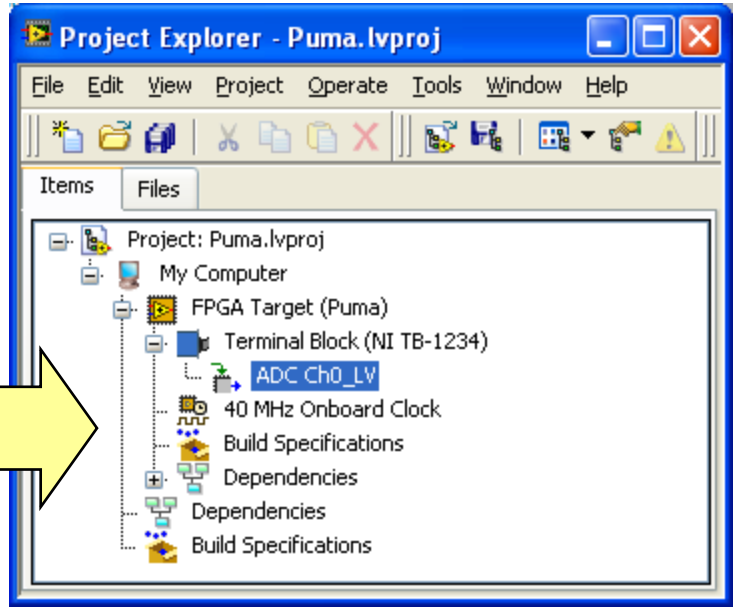
Adding a CLIP to the LabVIEW Project

AnalogFrontEndCLIP.vhd

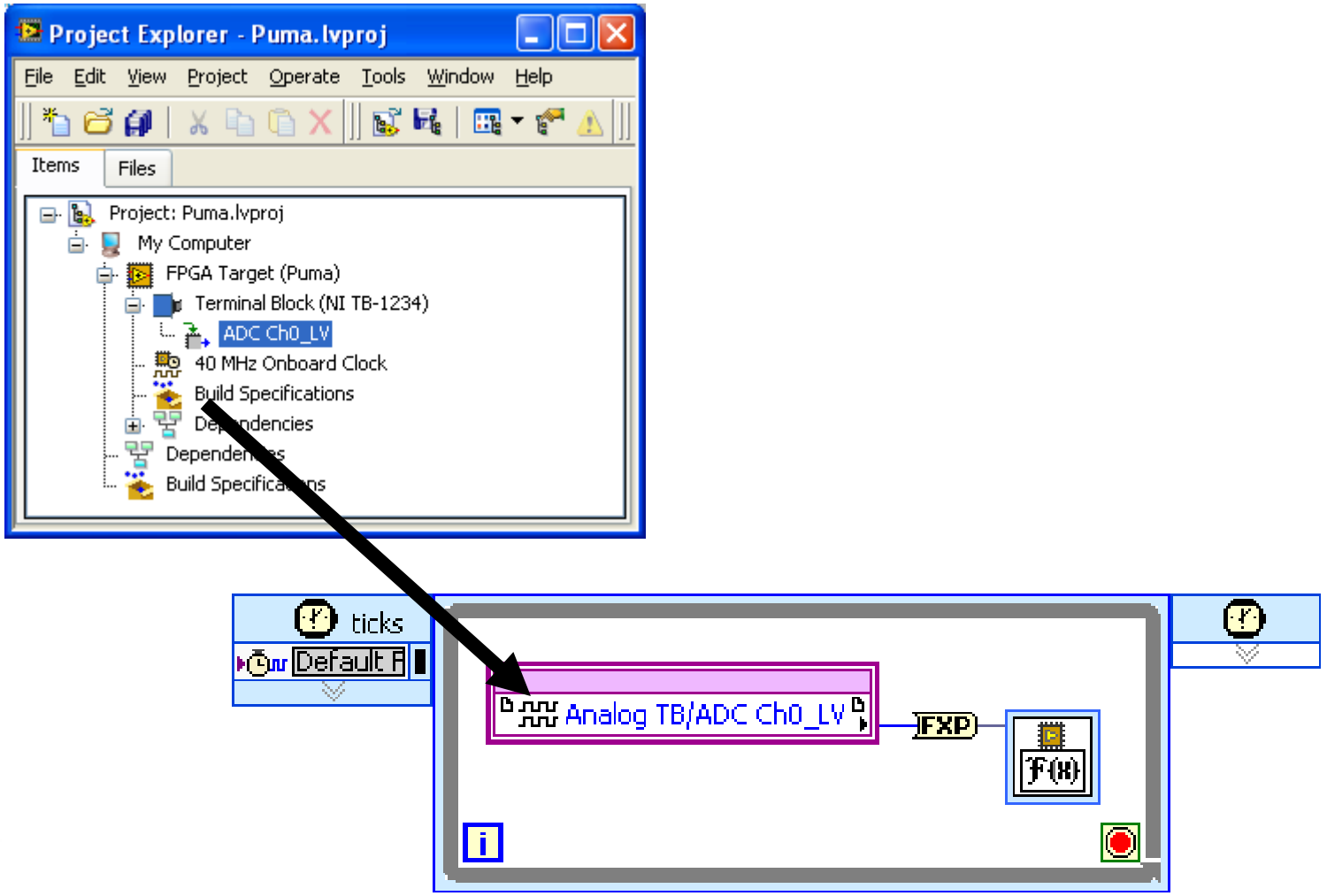
```
entity AnalogFrontEnd is
   port (
      adcCh0_LV : out std_logic_vector(15 downto 0);
      adcCh0_TB : in  std_logic_vector(15 downto 0) := x"0000"
   );
end AnalogFrontEnd;

architecture rtl of AnalogFrontEnd is
begin

     adcCh0_LV <= adcCh0_TB;
     ...

end rtl;
```

AnalogFrontEndCLIP.xml

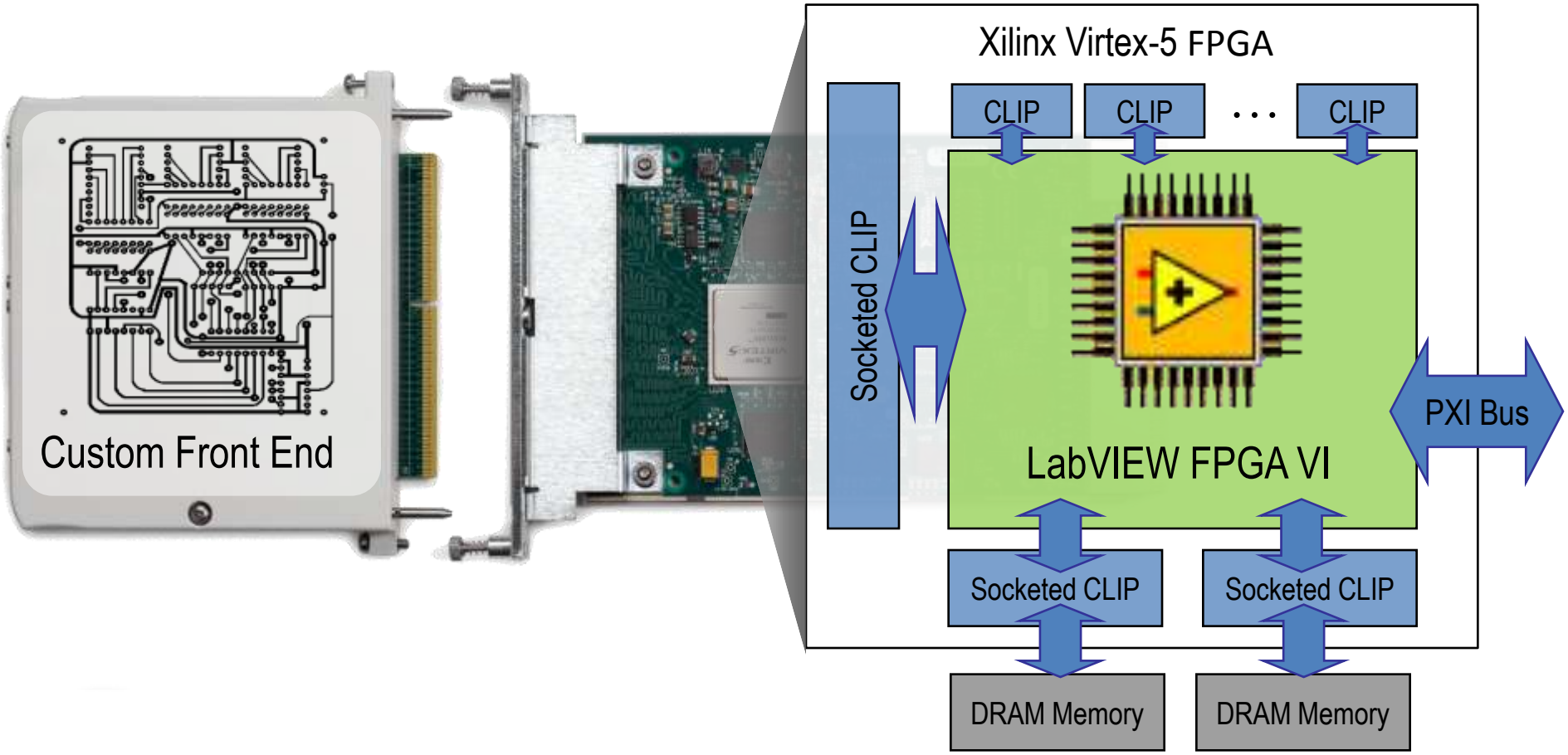
```
<?xml version="1.0" encoding="utf-8"?>
<CLIPDeclaration Name="AnalogFrontEnd">
  <HDLName>AnalogFrontEnd</HDLName>

  <InterfaceList>
    <Interface Name="AnalogIO">
      <InterfaceType>LabVIEW</InterfaceType>
      <Signal Name="ADC Ch0_LV">
        <HDLName>adcCh0_LV</HDLName>
        <DataType><I16/></DataType>
        <Direction>FromCLIP</Direction>
        <SignalType>data</SignalType>
      </Signal>
    </Interface>
 </InterfaceList>
</CLIPDeclaration>
```



Adding a CLIP to the LabVIEW Project



Custom Modules



PCI-5640R and PXIe-5641R

- 2 inputs and 2 outputs
- Identical data converters and similar analog performance
- Different FPGAs and bus interfaces

FPGAs and Busses

PCI-5640R

- Virtex II Pro FPGA (p30)
  - 30,816 logic cells
  - 136 multipliers
  - 2,448 Kbits block ram
  - 100 MHz maximum clock
- PCI bus
- DMA
  - To host > 100 MB/s
  - From host > 100 MB/s

PXIe-5641R

- Virtex 5 FPGA (sx95)
  - 94,208 logic cells
  - 640 multipliers
  - 8,784 Kbits block ram
  - 250 MHz maximum clock
- PXI Express (PCI Express) bus
- DMA
  - To host > 100 MB/s
  - From host > 50 MB/s

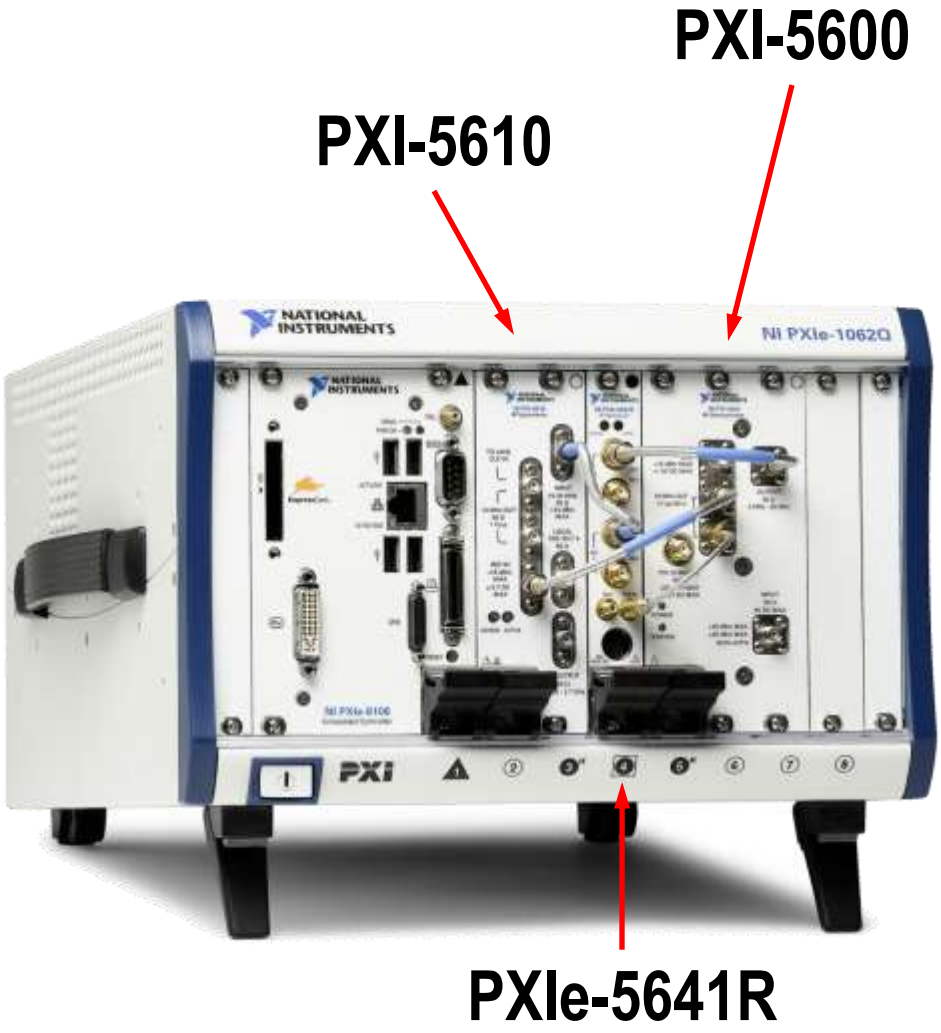
PXI Configuration

PXI-5600 Downconverter

- 9 kHz to 2.7 GHz
- 20 MHz bandwidth at 15 MHz IF
- -130 to 30 dBm input level
- 80 dB SFDR

PXI-5610 Upconverter

- 250 kHz to 2.7 GHz
- 20 MHz bandwidth at 25 MHz IF
- -110 to 10 dB output level



Choose the Right Model for Your Application

Host PC or PXI Controller

LabVIEW

Instrument-Specific Drivers or NI-RIO FPGA API

NI Modular Instrument	NI Modular Instrument	NI R Series DAQ	NI IF-RIO	NI FlexRIO	NI FlexRIO
Bus Interface and I/O Control	Bus Interface and I/O Control	Bus Interface + User-FPGA	Bus Interface + User-FPGA	Bus Interface + User-FPGA	Bus Interface + User-FPGA
NI-SCOPE Front End	NI Digital Front End	NI DAQ Front End	NI IF Front End	Partner Front End	Customer Front End

Summary

- FPGA Technology can increase test system
  - Functionality
  - Performance
  - Reliability
- NI FlexRIO family is designed to be NI's best test solution for
  - Processing performance
  - Diversity of I/O (including custom I/O)
- NI views FPGA technology as disruptive for test system design

Questions ?