Introduction to LabVIEW FPGA

Carlos Pazos
Embedded Software Product Marketing Manager
Embedded Systems Challenges

**High-speed I/O and analysis**
An oil well pump monitoring system requires high-speed I/O and analysis to catch momentary pressure spikes and vibration indications.

**System uptime and reliability**
A bio-refinery requires long system uptimes and high reliability for failsafe control systems.

**High-speed or deterministic control**
An Extremely Large Telescope (ELT) requires control of nanometric position actuators.
### Other Systems Challenges

<table>
<thead>
<tr>
<th>Automated Test</th>
<th>Channel Density</th>
<th>Programmable FPGA</th>
<th>Analog Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose Test</td>
<td>Lower Capitol Cost</td>
<td>Faster Filtering and Spectrum Analysis</td>
<td>High Accuracy, Repeatability, and Calibration</td>
</tr>
<tr>
<td></td>
<td>Improved Test Time with More Test Points</td>
<td>Complex and Deterministic Triggers</td>
<td>Best Channel and Device Synch with PXI</td>
</tr>
</tbody>
</table>

| High-Performance Embedded | | |
|---------------------------|-----------------|-------------------|-------------------|
| LIDAR | More Signals for More Precise Results | Reduce Data Processing time | >10 ENOB on 8 Channels |
| RADAR | | | |
| Signal Intelligence Beam Position Monitoring General Physics | High Density for Compact Design | Customizable Algorithms and Memory Control | High Bandwidth for Spectral Analysis |
A Processor Based Approach

Benefits:
- Can easily be programmed
- Well-suited for high-precision floating point calculations
- Well-suited for networking and peripheral I/O
Processor Based Approach

- Inputs
  - ~25 us Response

- System or Device

- Hardware

- Outputs

- CPU
  - Operating System
  - Driver API
  - Calculation
  - Application Software
Decision Making in Hardware

System or Device

Inputs

~25 ns Response

Calculation

Outputs

Hardware

CPU

Operating System

Driver API

Application Software
FPGA Technology

What is an FPGA?

- Software defined hardware
- No operating system is needed for execution of logic

---

Dedicated logic in silicon for highest reliability

I/O Blocks  Logic

FPGA

Inline signal processing with minimal latency

High-speed and deterministic control
The NI Approach

We call this the LabVIEW RIO architecture.

Processor
Real-Time or PC-Based

FPGA

Modular I/O for Any Signal

Highly Productive LabVIEW Graphical Programming Environment for Programming Host, FPGA, I/O, and Bus Interfaces
High-Speed Control Application

Plant → I/O → FPGA

Acquisition and Control → Watchdog Timer

FPGA → Processor (RTOS)

System Monitoring → Sequencer Engine → Disk

Processor (RTOS) → Processor (HMI)

Display → UI Commands
High-Throughput Test Application

- **I/O**
  - DUT
  - DUT
  - DUT

- **Triggering**

- **FPGA**
  - Signal Processing
  - Analysis
  - Analysis

- **DUT Control**

- **Processor (HMI)**
  - Display
  - Data Logging
  - Disk

ni.com
Embedded Software Development Challenge

- Embedded development requires multiple software tools
- Parallel processing increases system complexity
- Software tools don’t address system design

Long learning curves
Limited reuse
Need for “specialists”

Increased costs
Increased time-to-result

Tools
- Math (.m file script)
- Simulation (Hybrid)
- User Interface (HTML)
- FPGA (VHDL, Verilog)

Targets
- Host Control (C, C++, .NET)
- DSP (Fxd pt C, Assembly)
- H/W Driver (C, Assembly)
- System Debug

FPGAs
Multicore Processors
Graphical System Design for FPGAs

LabVIEW FPGA

Xilinx IP

HLS Technology

VHDL

NI RIO Hardware

Value

Rugged

Performance

High Performance

High Performance

ni.com
LabVIEW FPGA Module

- Use LabVIEW to design hardware
- Offload the most critical pieces of your application
  - High speed control
  - Inline signal processing
  - Custom protocols
  - Custom timing, triggering, and synchronization
  - Fast stimulus/response testing
Mapping LabVIEW to an FPGA
Using the LabVIEW Project

- Development PC Target
- Embedded Processor Target
- FPGA Target
- FPGA Resources (I/O, clocks, IP, etc.)
Abstraction of Hardware Complexities
Interactive Front Panel Communication
LabVIEW FPGA Datatypes

- The fixed-point datatype is very efficient for hardware applications (DSPs, FPGAs, etc.)
- Uses less hardware resources than floating-point
- Single precision floating-point datatype is available and recommended for certain use cases
Abstraction of Timing

- Loops can execute on the order of ticks of the 40 MHz clock (nanoseconds), microseconds, and/or milliseconds.
Single-Cycle Timed Loop

- Executes code within 1 cycle of the FPGA clock
- Can be used to optimize the performance of your code
LabVIEW FPGA Functions and IP

In Product

- Filters (Butterworth, Notch, DC-RMS, etc.)
- PID control
- Control of brushless DC motors
- Digital buses and protocols (SPI, I2C, UART, etc.)
- Image Processing
- RF communications
- Linear and nonlinear systems
- PWM
- Encryption

Online

- Data manipulation
- Device drivers (LCD display, IR sensors, etc.)
- Video processing
- Basic elements (counters, accumulators, etc.)
- Signal generation
- High-throughput math
- Transforms
- Trig functions
- Digital signal processing
- …and more

ni.com/IPnet
Reuse of Existing HDL Algorithms

• Increase application development efficiency and leverage existing team expertise
• Similar to calling a DLL in LabVIEW for the desktop

```vhdl
Library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity DemoClipAdder is
  port (  
    clk     : in std_logic;
    aReset  : in std_logic;
    cPortA  : in std_logic_vector(15 downto 0);
    cPortB  : in std_logic_vector(15 downto 0);
    cAddOut : out std_logic_vector(15 downto 0);
  );
end entity DemoClipAdder;
```
Compilation Process

LabVIEW FPGA Code  \[\text{Compile VHDL through Xilinx}\]  \[\text{FPGA Logic Implementation}\]
Compilation Process

LabVIEW FPGA Code

Compile VHDL through Xilinx FPGA Logic Implementation

Translation
VHDL Generation

Optimization
Analyze Logic Reduction

Synthesis
Place and Route Timing Verification

Bit Stream Generation
Download & Run
One-Click Deployment and Compilation

Development PC

Compile Server and Workers

High-Performance Cloud
LabVIEW FPGA Compile System

LabVIEW FPGA Compile Farm Toolkit

On-Site Compile Farm

Compile Server
- Farms out compilation jobs to available workers

Compile Workers
- Executes compilation jobs and sends back a bitfile

Development Machine
- Sends intermediate files generated by LabVIEW FPGA to the compile server using web services

Compile Server and Compile Workers are connected through Ethernet.
LabVIEW FPGA Compile System

LabVIEW FPGA Compile Cloud Service

Compile Server
Farms out compilation jobs to available workers

Compile Workers
Executes compilation jobs and sends back a bitfile

Development Machine
Sends intermediate files generated by LabVIEW FPGA to the compile server using web services

Ethernet
DEMO
CREATING A TESTBENCH
Host Synchronization

- The Read/Write Controls method can be used for communicating current value data
Host Synchronization

- Direct Memory Access (DMA) FIFOs are an efficient mechanism for streaming data from the FPGA to the host processor
- Does not involve processor resources
DEMO
HOST SYNCHRONIZATION
Unrivaled Integration with the Latest Technology

Software Designed Oscilloscope

System on a Module (SoM)

Performance CompactRIO

USB3 CVS

LabVIEW™ 2014
New FPGA Hardware Targets
Be More Productive with LabVIEW FPGA 2014

Design Faster

**Design High-Performance Algorithms**

Use **LabVIEW FPGA IP Builder** to design optimized, high-performance algorithms using high-level programming constructs.

**Design PID Controllers**

Use the **PID Control VI** to quickly prototype high-speed or high-determinism control algorithms.

**Design Image Processing Applications**

Offload over **50 image processing functions** to the FPGA for maximum performance with the NI Vision Development Module 2014.
Be More Productive with LabVIEW FPGA 2014

Verify Faster

**Verify Code using Simulated I/O**
Use the **Desktop Execution Node** to verify code by developing test benches using simulated or file generated I/O.

**Verify Signal Timing with Waveform Probe**
Use the **Digital Waveform Probe** to probe your signals relative to one another and view history.
Be More Productive with LabVIEW FPGA 2014

Compile Faster

Send Your Compiles to the Cloud
Use the LabVIEW FPGA Compile Cloud Service (free with SSP) to reduce your compile times up to 60%.

Manage FPGA Compilations On-Site
Use the LabVIEW FPGA Compile Farm Toolkit to create an on-site server to manage FPGA compilations.

Increase Compilation Performance with Vivado
Use Xilinx Vivado included with LabVIEW FPGA 2014 to compile faster and more reliably for Kintex-7 FPGAs and Zynq SoCs.
The Benefits of a Platform Based Approach

- High-Level Software
- Flexible Hardware
- Integrated Hardware and Software Platform
ni.com/fpga

- FPGA Fundamentals
- Benefits of FPGAs
- NI FPGA-based Case Studies
- Learn more about the tools
  - LabVIEW FPGA
  - FPGA-based RIO hardware
NI Instructor Led Training
Classroom, Virtual, or Online

Week 1
- Core LabVIEW
  - LabVIEW Core 1
  - LabVIEW Core 2

Week 2
- Core LabVIEW
  - LabVIEW Core 3
- Managing Software Engineering
  - LabVIEW FPGA
  - LabVIEW RT
  - Architecting Embedded Systems

- Embedded Control & Monitoring
  - LabVIEW FPGA

- High-Throughput Test
  - LabVIEW FPGA for HT Test

ni.com
LabVIEW for CompactRIO Developer’s Guide

- Best practices for designing embedded control and monitoring systems with LabVIEW
- Recommended architectures and frameworks
- Downloadable example code throughout

ni.com/compactriodevguide
LabVIEW for CompactRIO Sample Projects

- Recommended starting points designed to ensure the quality and scalability of a system

- LabVIEW FPGA Control & Monitoring (above)
- LabVIEW FPGA Control with Sequencer Engine
- LabVIEW Real-Time Control & Monitoring
- LabVIEW FPGA Waveform Acquisition and Logging