Don't Think You Need an FPGA? Think Again!

NATIONAL INSTRUMENTS

Don't Think You Need an FPGA? Think Again!

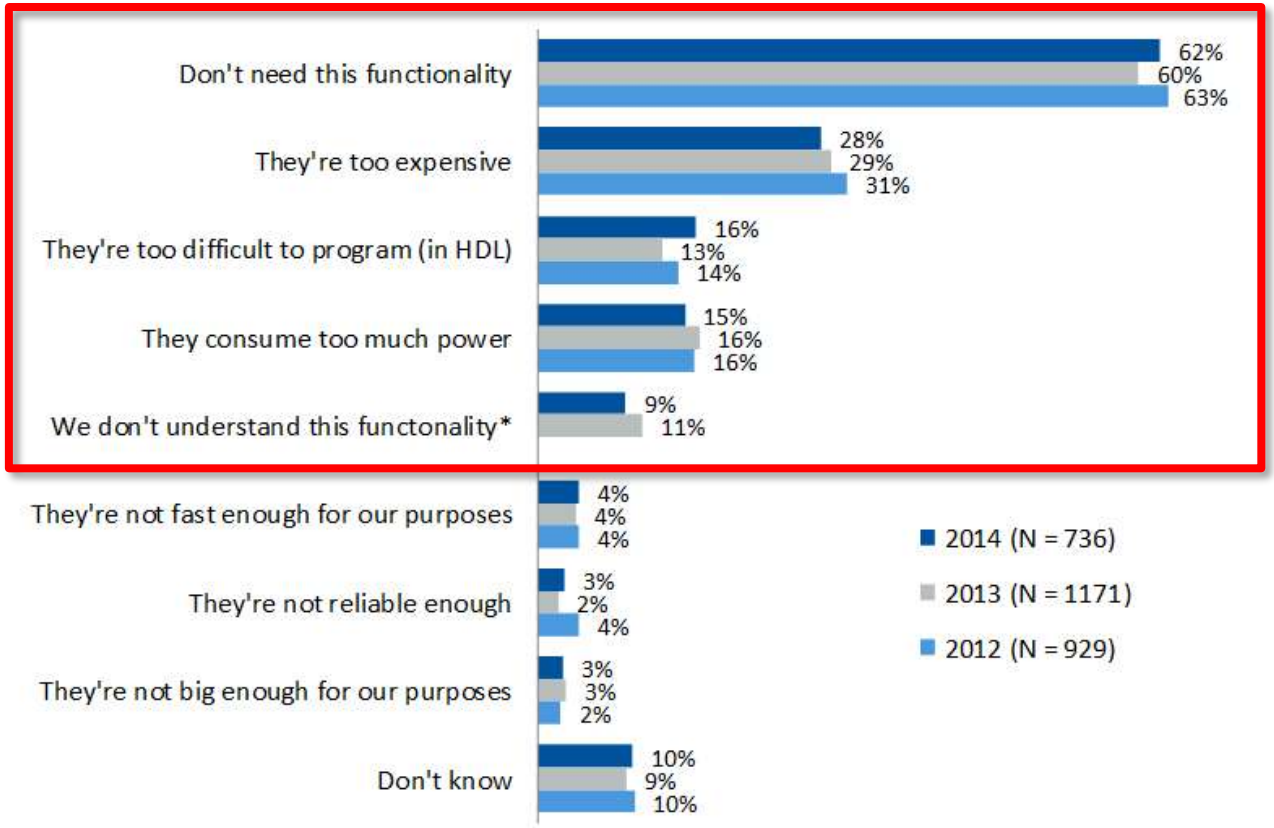
Goals for Today

- Define and explain FPGAs
- Address common misconceptions about FPGAs
- Understand the best and most intuitive way to program FPGAs
- **Understand when and how FPGAs can add value to your applications**

Audience Poll

- Are you using FPGAs in designs today?
- What are some reasons people DON'T use FPGAs?

Why won't your next project include customizable chips?



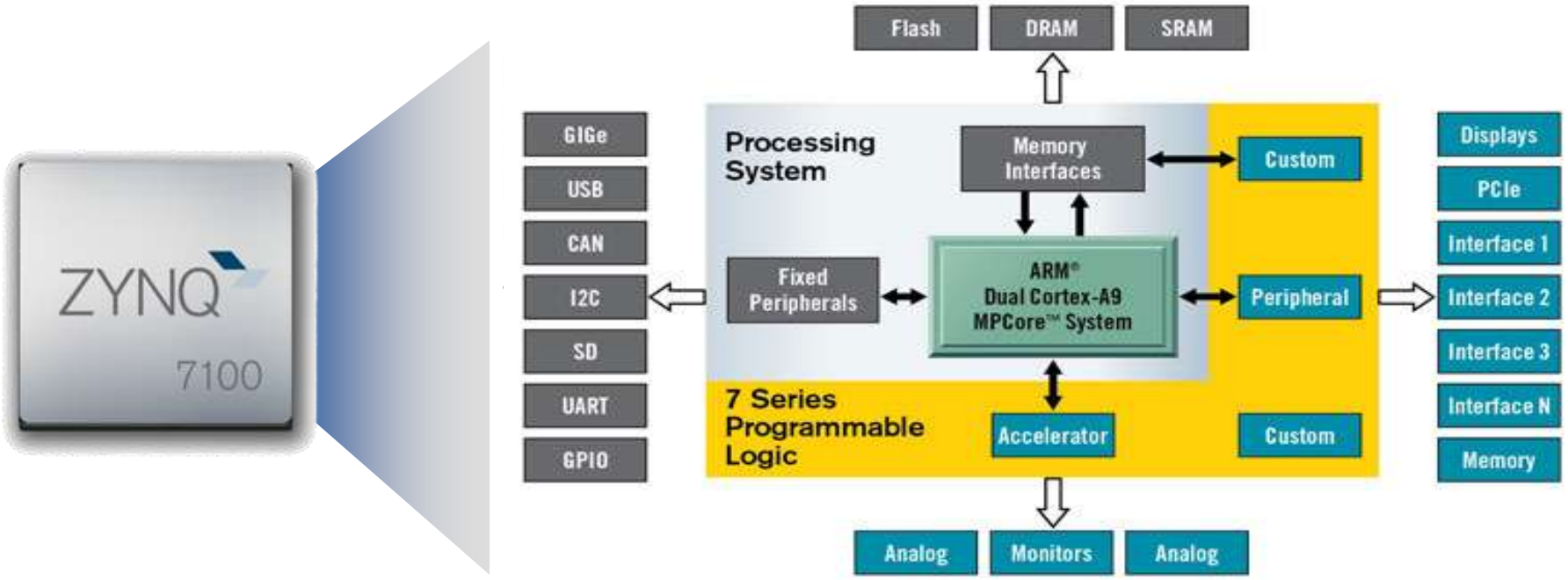
- Added in 2013

UBM Tech

Copyright © 2014 by UBM Tech. All rights reserved

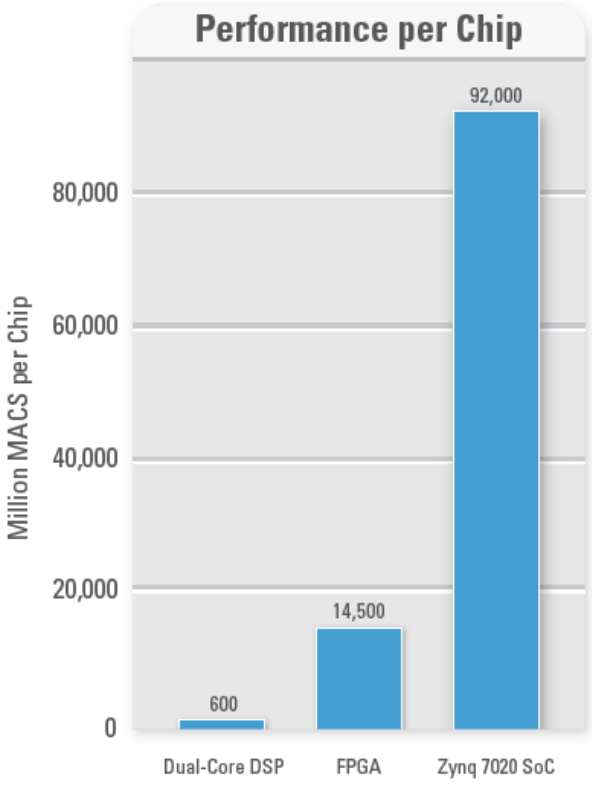
What is an FPGA?

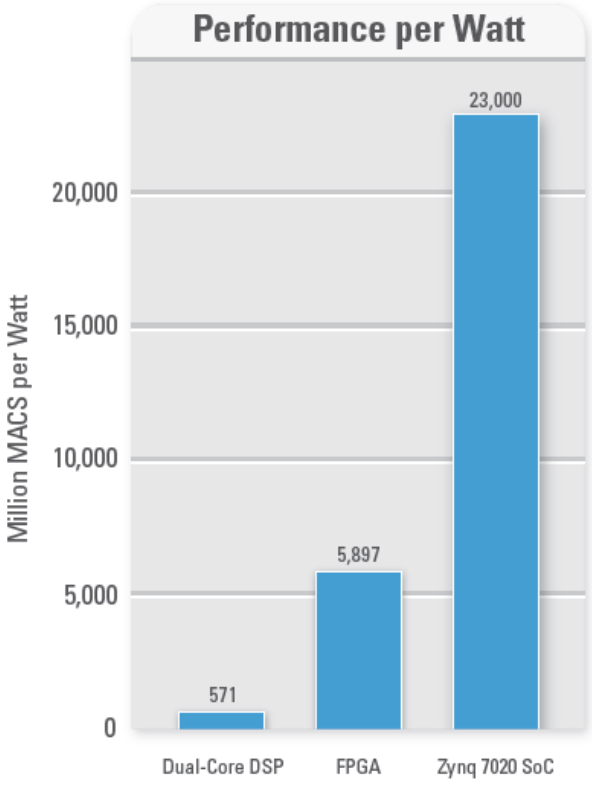
The Effect of Moore's Law on FPGAs: Heterogeneous, Massively Parallel SOCs

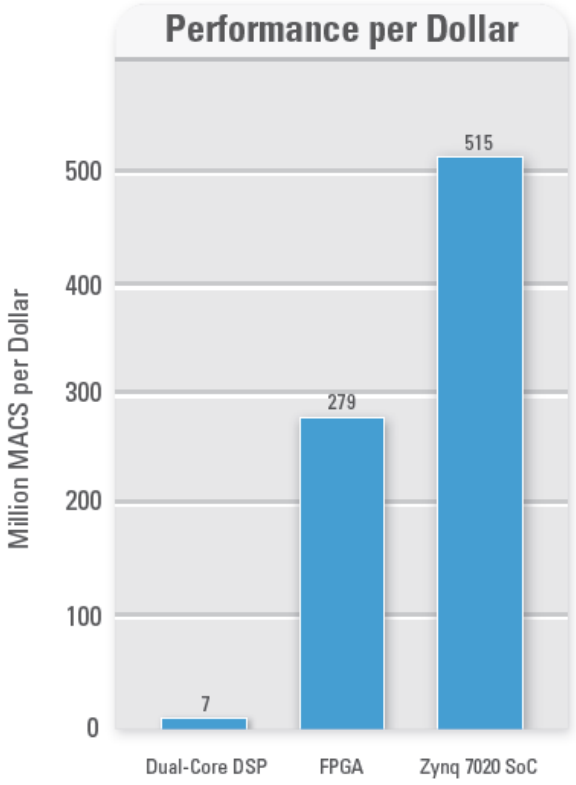


- Reduced power consumption
- Improved reconfigurability
- Lower cost

Advantages Over DSPs or ASICs







	Dual-Core DSP	Spartan-6 LX45 HFPGA	Zynq 7020 SoC	Performance Ratio (FPGA/DSP)	Performance Ratio (Zynq-7020/DSP)
Million MACS per Chip	600	14,500	92,000	24	153
Million MACS per Watt	571	5,897	23,000	10	40
Million MACS per Dollar	7	279	515	40	74

MACS = Multiply-accumulate operations per second
(measure of DSP performance)

Evolving Design Approaches Along With Technology

Traditional

Improved

Don't Think You Need an FPGA? **Think Again!**

– 3 *Reasons to Augment Your Application With an FPGA* –

Future-Proof Your Design

Adapt to changing requirements, evolution of projects

Maximize Reliability and Determinism

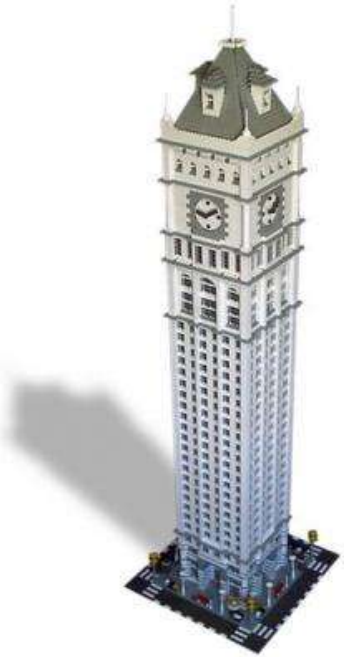
For time-critical, safety-critical, and deployed systems

Enhance Performance and Improve Functionality

Offload processing, ultra-fast control, custom timing…

1. Future-Proof Your Designs

- Adapt to changing requirements
- Change, enhance, add (or remove!) functionality over time
- Reuse hardware (and code) from project to project



Case Study: Innosiv Engineering

Complex Control System

Results

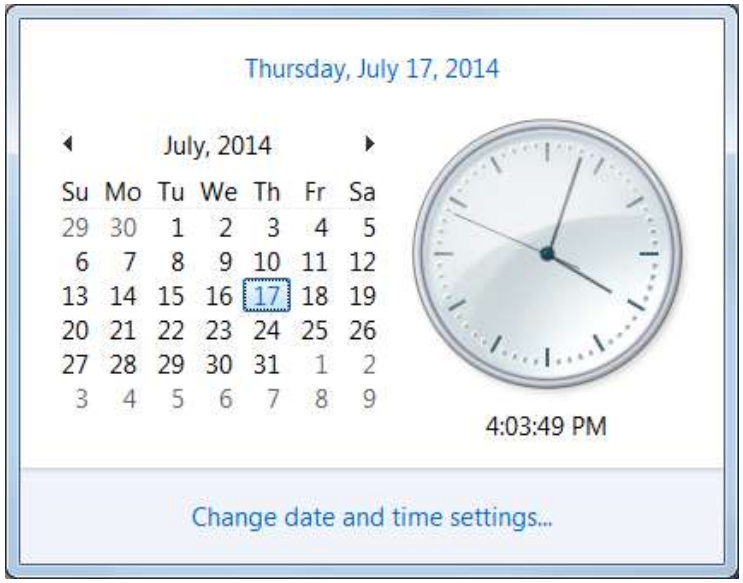
- Reproduced the coast-down timing within 0.10 s
- Force error smaller than 10 N
- Slippage between front and rear wheel kept to a minimum of 0.07 ppm
- The system can run at 140 km/h with the speed differences as small as 0.01 m/s

"The flexibility of the system based on CompactRIO and LabVIEW was more impressive than the end results. For all measurement and control challenges discovered during the implementation, we quickly found a satisfying and innovative solution."

-- Shahzad Sarwar, Innosiv Engineering

2. Maximize Reliability and Determinism

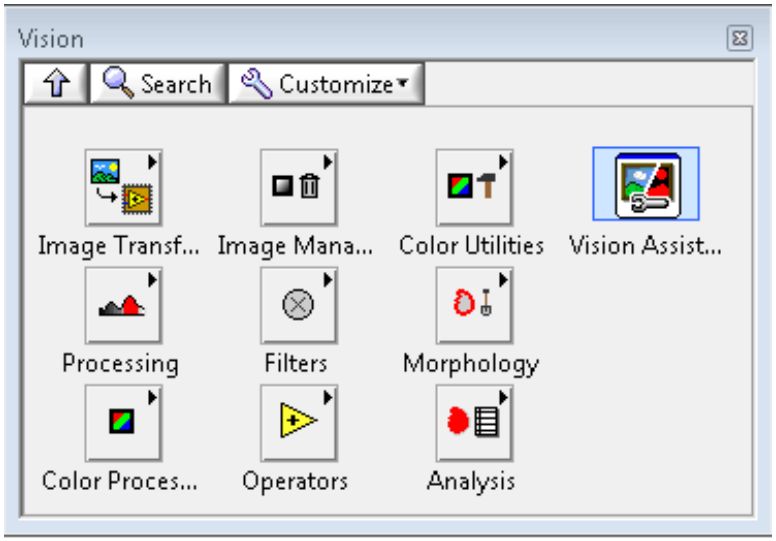
- Application executes in *hardware*
  - Incredible reliability and safety (maximize uptime!)
  - Extreme determinism and clock cycle control
    - Practically zero jitter; Loop rates up to ~200 MHz



3. Enhance Performance and Functionality

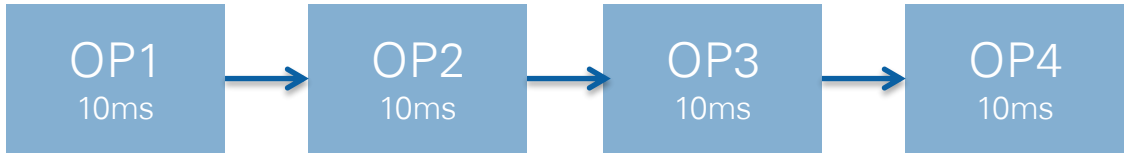
- Extremely fast closed loop control or stimulus/response test
- Inline processing (i.e. Vision or Test applications)

Total time = 40 ms

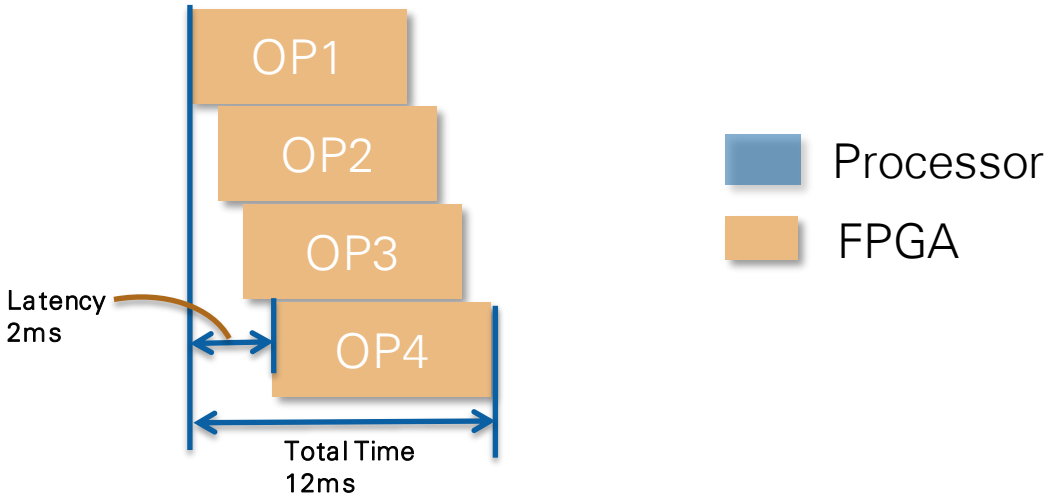


3. Enhance Performance and Functionality

- Extremely fast closed loop control or stimulus/response test
- Inline processing (i.e. Vision or Test applications)



Total time = 40 ms

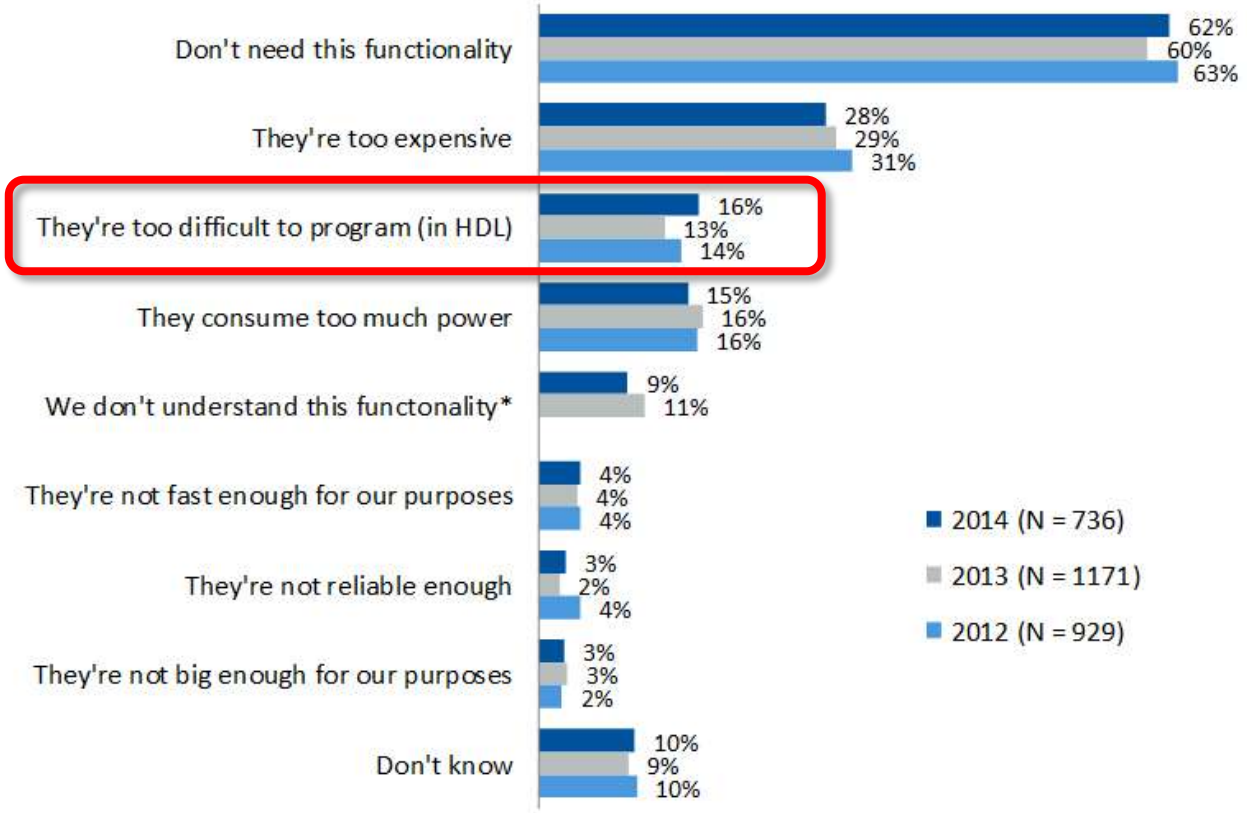


3. Enhance Performance and Functionality

- Extremely fast closed loop control or stimulus/response test
- Inline processing (i.e. Vision or Test applications)
- Ability to extend your DAQ or PLC application
  - Custom timing, triggering, high speed Analog I/O
- Custom digital sensor integration

http://sine.ni.com/cs/app/doc/p/id/cs-678

Why won't your next project include customizable chips?



• Added in 2013

UBM Tech

Copyright © 2014 by UBM Tech. All rights reserved

If FPGAs are so great,
Then why don't more system designers consider them?

Traditional Approach = Very Costly

Traditional Approaches Are Worlds Apart

Processors

What they use
C, C++, C#, Java

What they know
Higher Level APIs

How they develop
Compile early, often

Debug with probes, single stepping

FPGAs

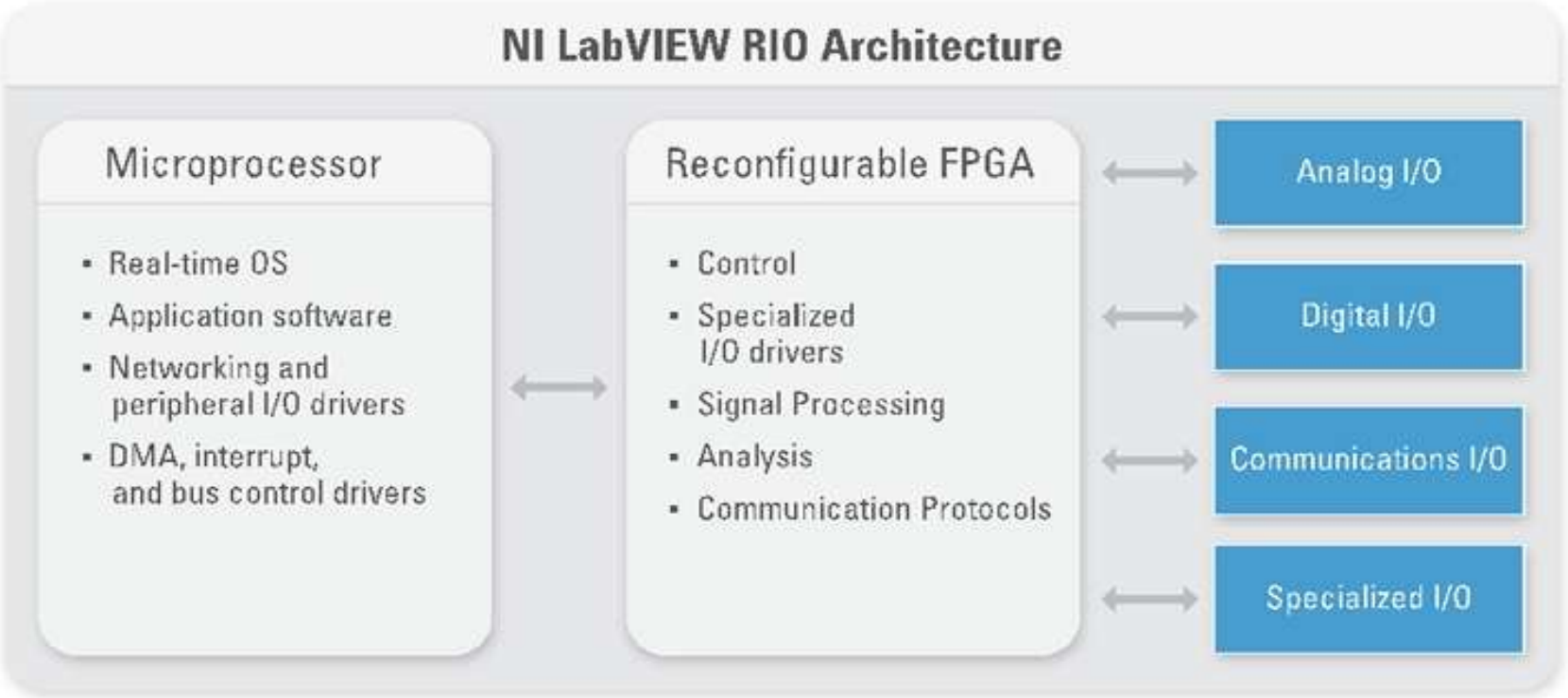
What they use
VHDL, Verilog

What they know
RTL programming

How they develop
Model-based design

Debug with JTAG, ChipScope

The LabVIEW RIO Architecture:
The Foundation for Innovation



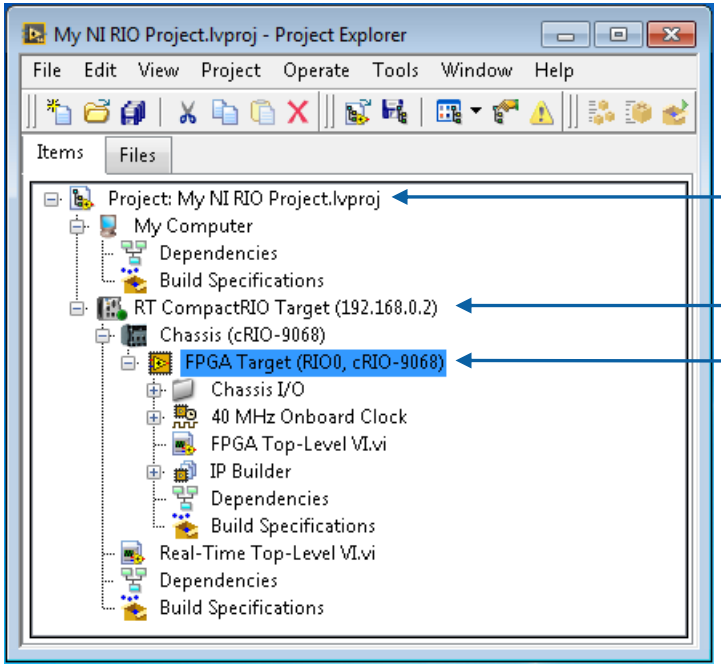
Don't Just Take Our Word for it…

"You need a high-level language to bring FPGA to the masses, people who are not essentially hardware programmers. We think LabVIEW is a very good tool that enables the domain experts to program FPGAs…"
– Ivo Bolsens, CTO, Xilinx

One Tool for Everything

In LabVIEW, you can…

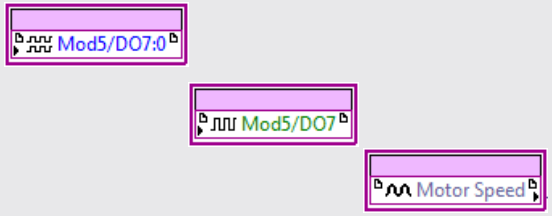
- Manage FPGA resources
- Develop, compile, and deploy application
- Simulate and debug your application



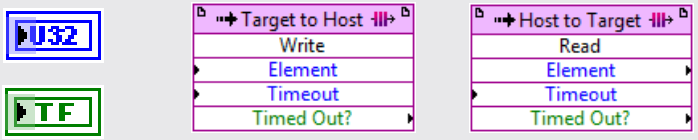
FPGA-Specific Language Elements

LabVIEW FPGA Elements

IO Interface



Data Communication



Timing

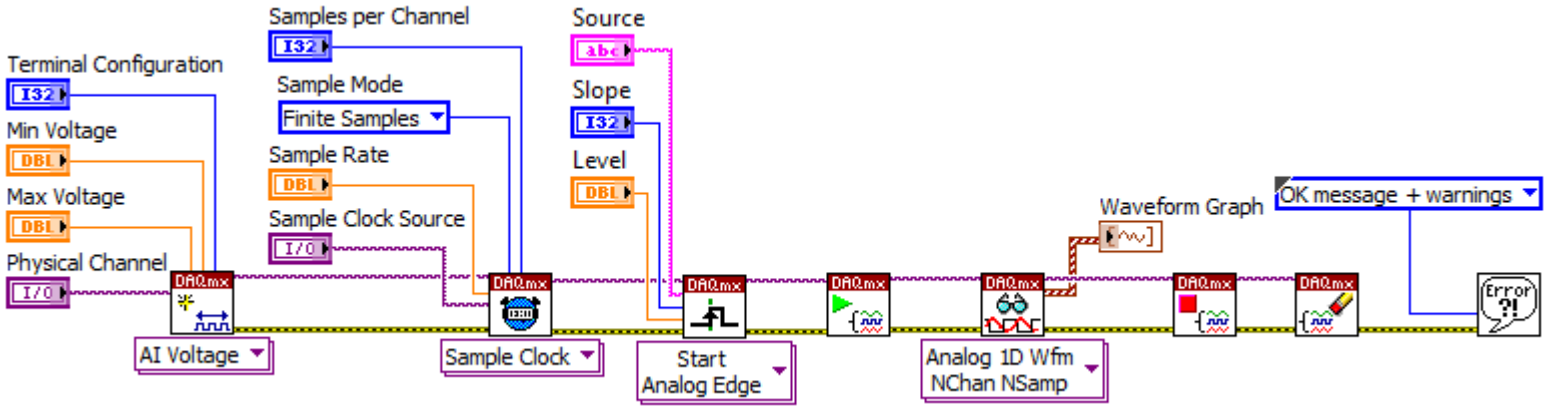
Control

New Rules for Software Development

- Your code actually configures hardware circuits
  - Compile times are longer compared to desktop
  - Signals on a wire--harder to debug
- How can you stay productive?
  - Multiple simulation methods
    - Compile fewer times
    - Verify functionality and timing in simulation
  - Cloud compilation for everyone
    - Compile faster
    - Compile offline
  - IP Builder for automatic optimization

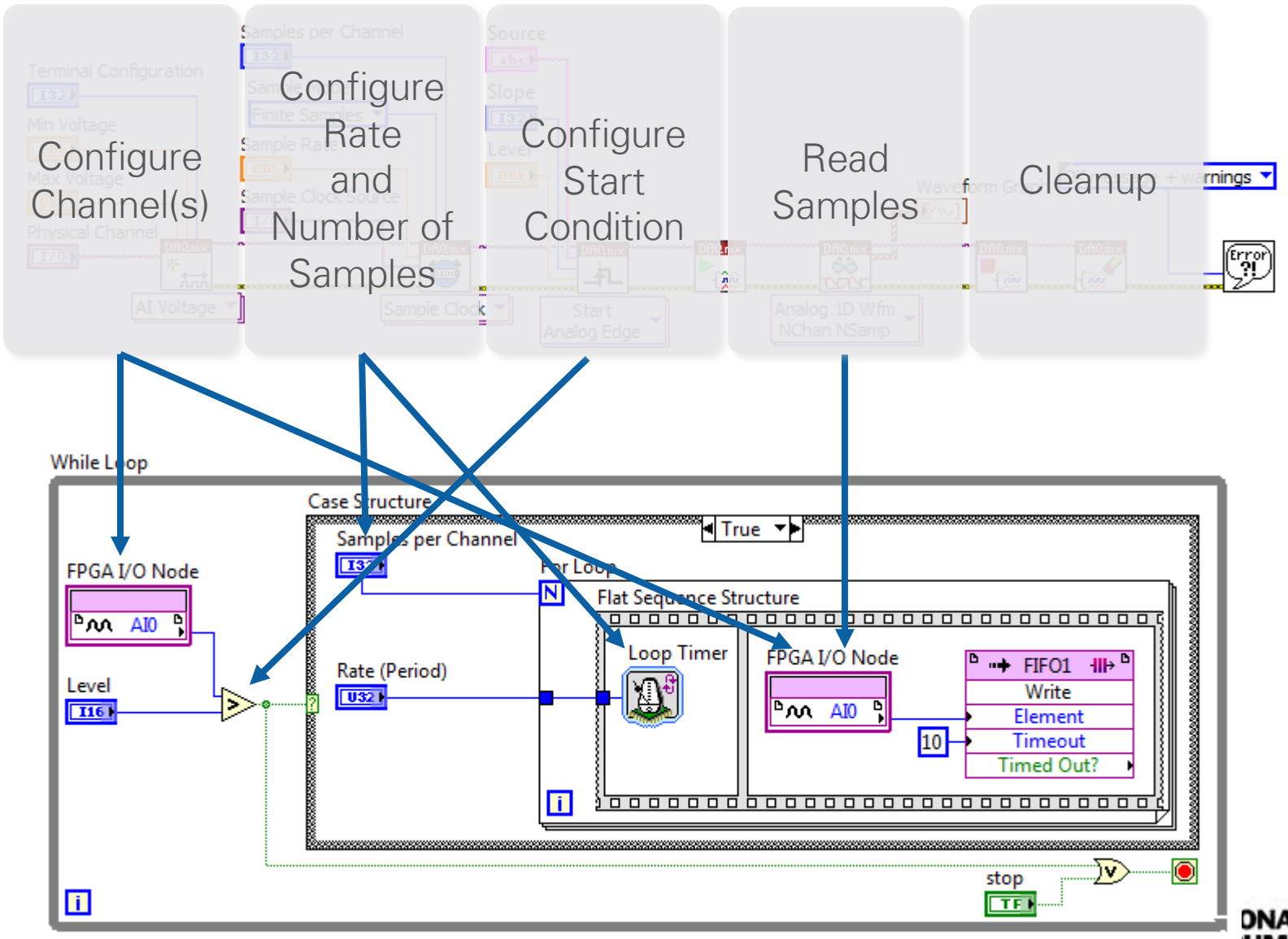
LabVIEW FPGA Demonstration

Porting Your DAQ Application



NATIONAL INSTRUMENTS

Porting Your DAQ Application



Right Tool for the Job?

NATIONAL INSTRUMENTS

Don't Port Your Entire Application

- Pick the right place for your task: processor or FPGA
- Certain operations might be way simpler on the processor

Identify parts that map the best to the FPGA

Are there any fast control loops?
Highly deterministic tasks?

Don't Port Your Entire Application

- Pick the right place for your task: processor or FPGA
- Certain operations might be way simpler on the processor

Identify parts that map the best to the FPGA

Which components need to be highly reliable?

Don't Port Your Entire Application

- Pick the right place for your task: processor or FPGA
- Certain operations might be way simpler on the processor

Identify parts that map the best to the FPGA

What tasks are taxing my processor?

Case Study: Innosiv Engineering

LabVIEW Windows Loop Rate = 20 Hz	LabVIEW Real Time Loop Rate = 250 Hz	LabVIEW FPGA Loop Rate = 1 MHz

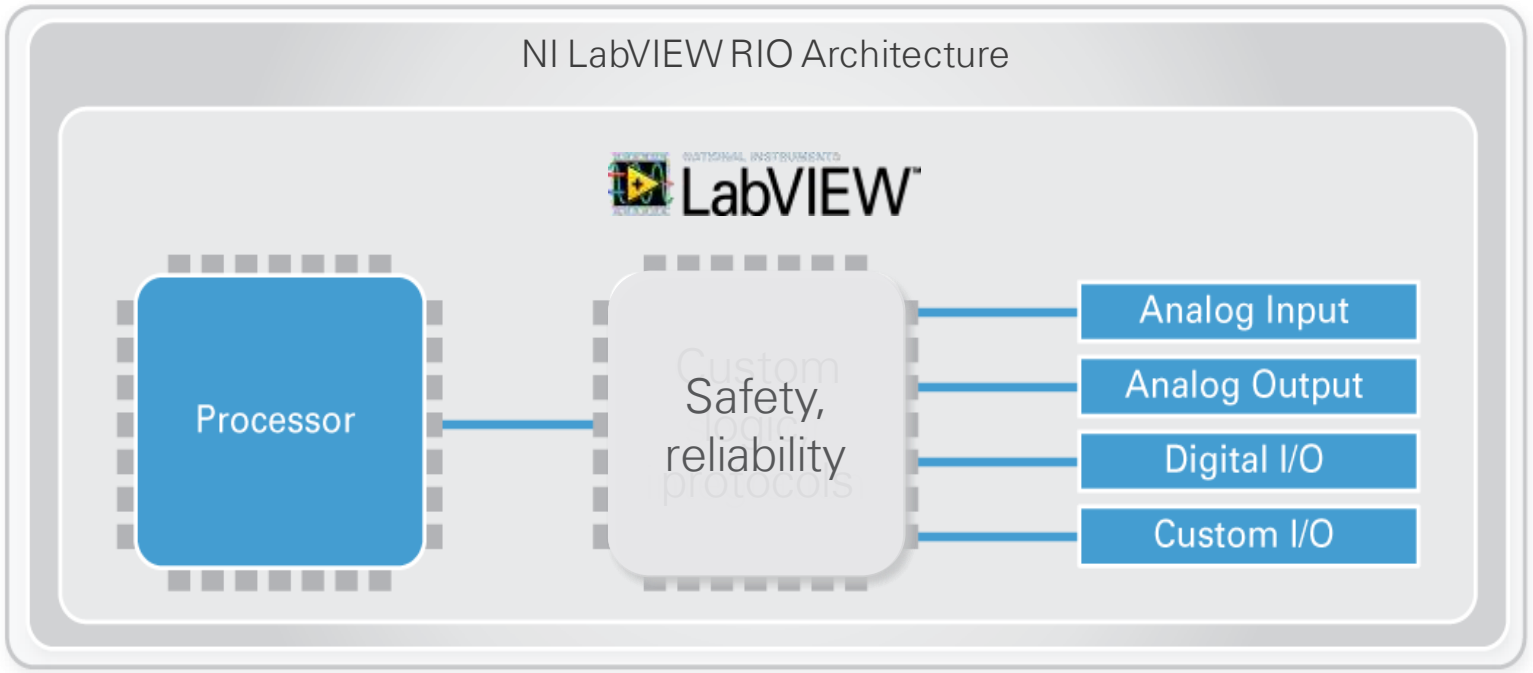
• User Interface
• Data Logging
• Vehicle Database
• Calibration Tasks
  • Torque Loss
  • Base Inertia
• Coast Down
	• PID Control
	  • Torque
	  • Speed
	  • All Wheel Drive
	• Simulation
	• Interlocks
	• Remote Interface
		• Measurements
		  • Torque
		  • Distance
		  • Speed
		  • Acceleration
		• Digital I/O
		• Safety Checks

http://sine.ni.com/cs/app/doc/p/id/cs-12996

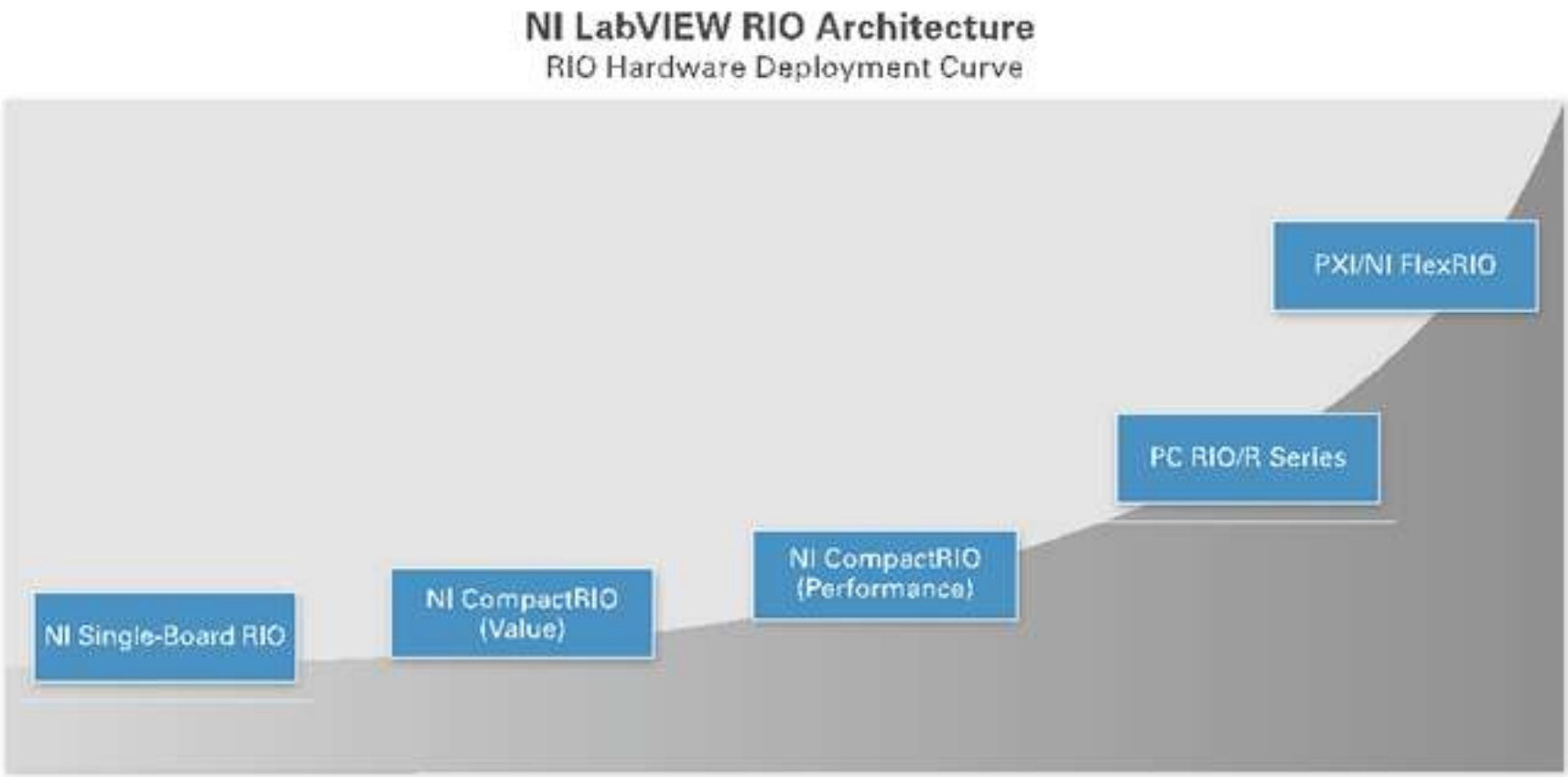
Still think you don't need FPGAs?

Maybe not today, but you might tomorrow!

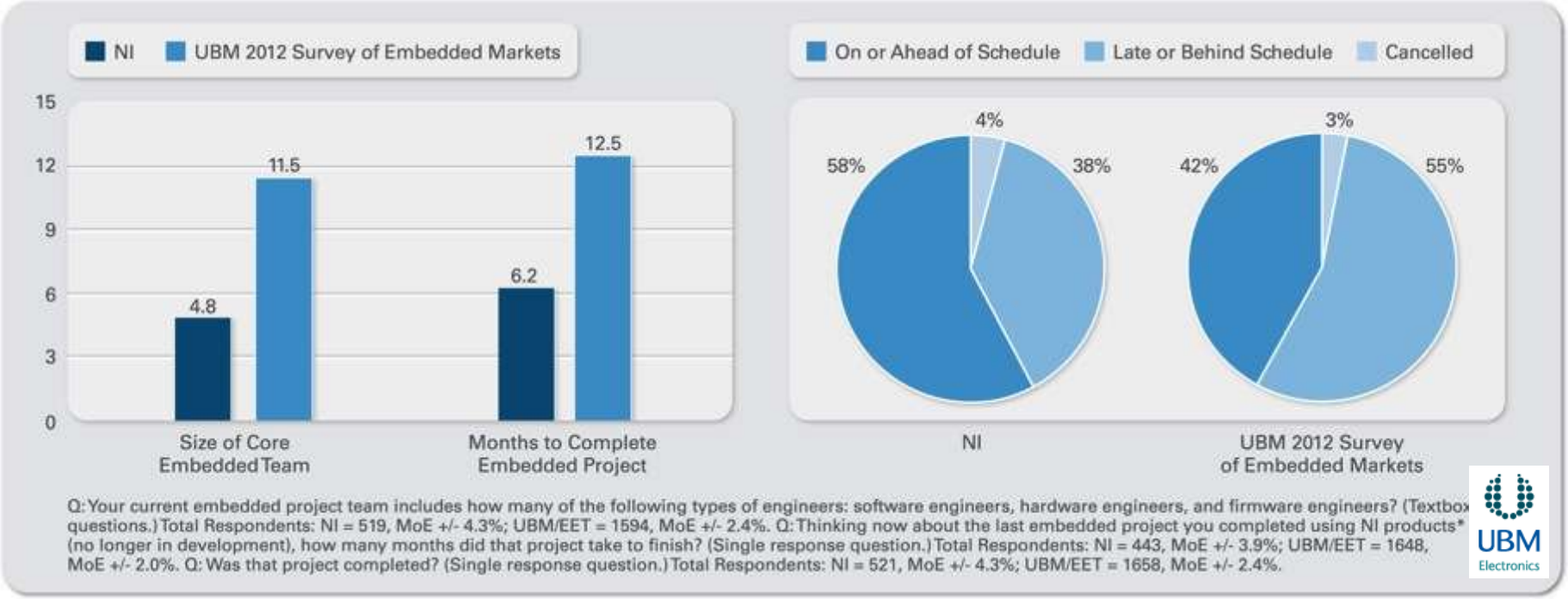
The Foundation for Innovation



A Platform That Scales



Smaller Teams Get to Market Faster With NI Tools



	NI	UBM
Size of core embedded team (average # of SW/HW/firmware engineers)	4.8	11.5
Time to complete projects (months)	6.2	12.5
On/ahead of schedule	58%	42%
Behind schedule/late	38%	55%