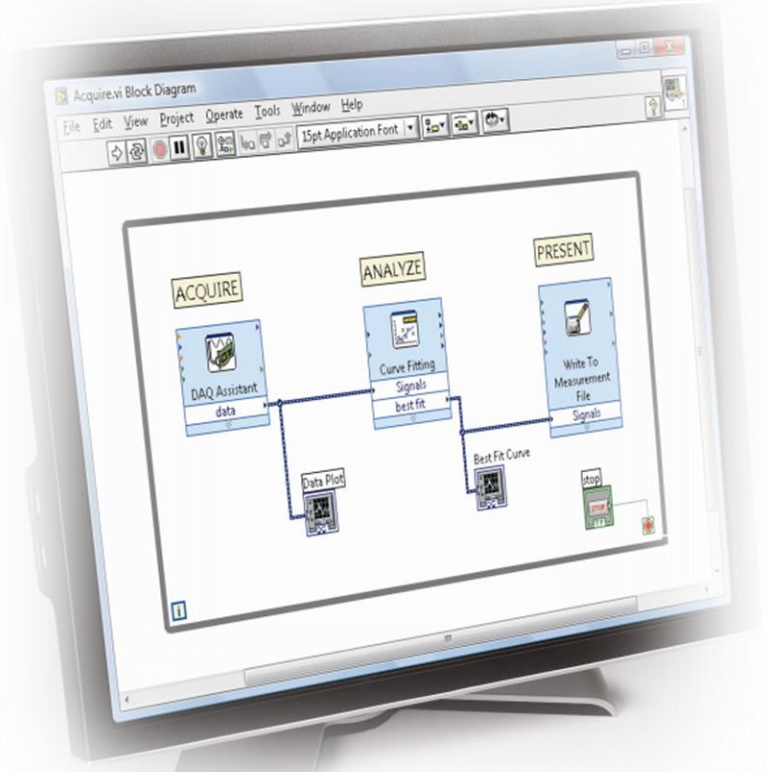
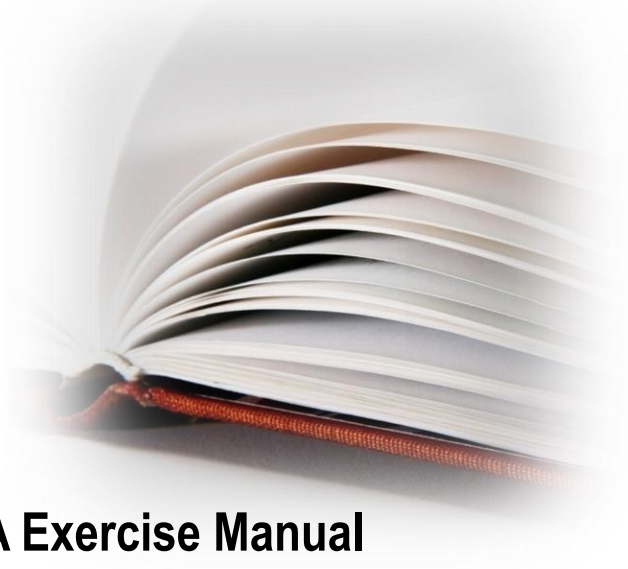
NATIONAL INSTRUMENTS

LabVIEW FPGA

What You Need To Get Started





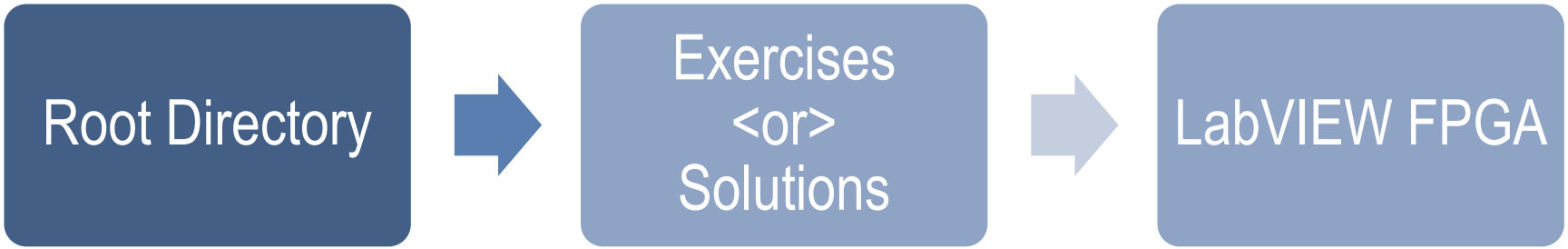
- LabVIEW FPGA Exercise Manual
- LabVIEW FPGA Course CD
  - LabVIEW FPGA Course Manual
- NI Sound and Vibration Signal Simulator
- Two J-type thermocouples
- NI 9074 Integrated Controller or equivalent
- NI 9211, 9233, and 9263 Modules

Computer running Windows 7/Vista/XP with the following software installed:

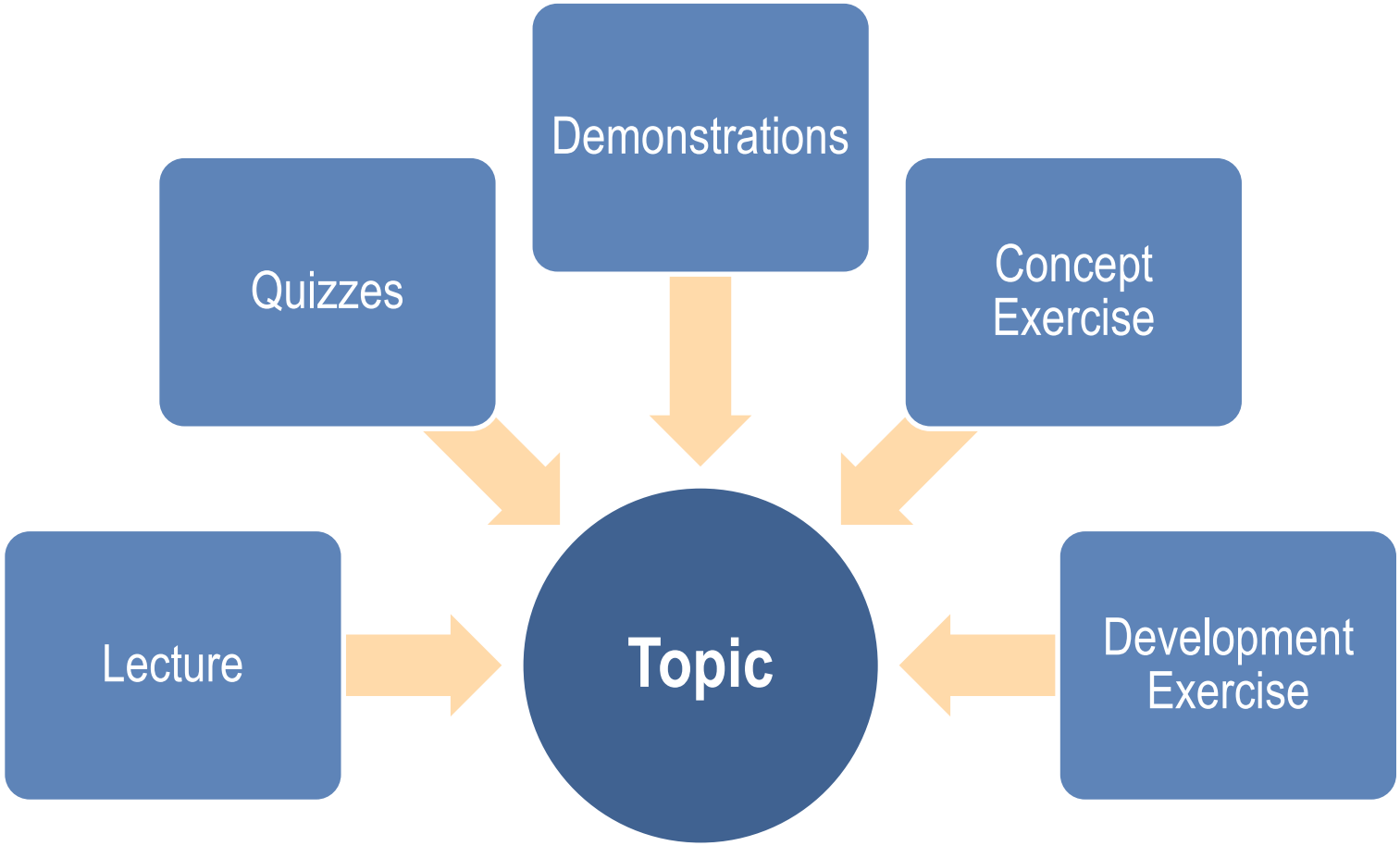
- LabVIEW 2010 or later
- LabVIEW Real-Time
- LabVIEW FPGA
- NI-RIO 3.5.0 or later

File Locations

The course installer places the course files in the following location:

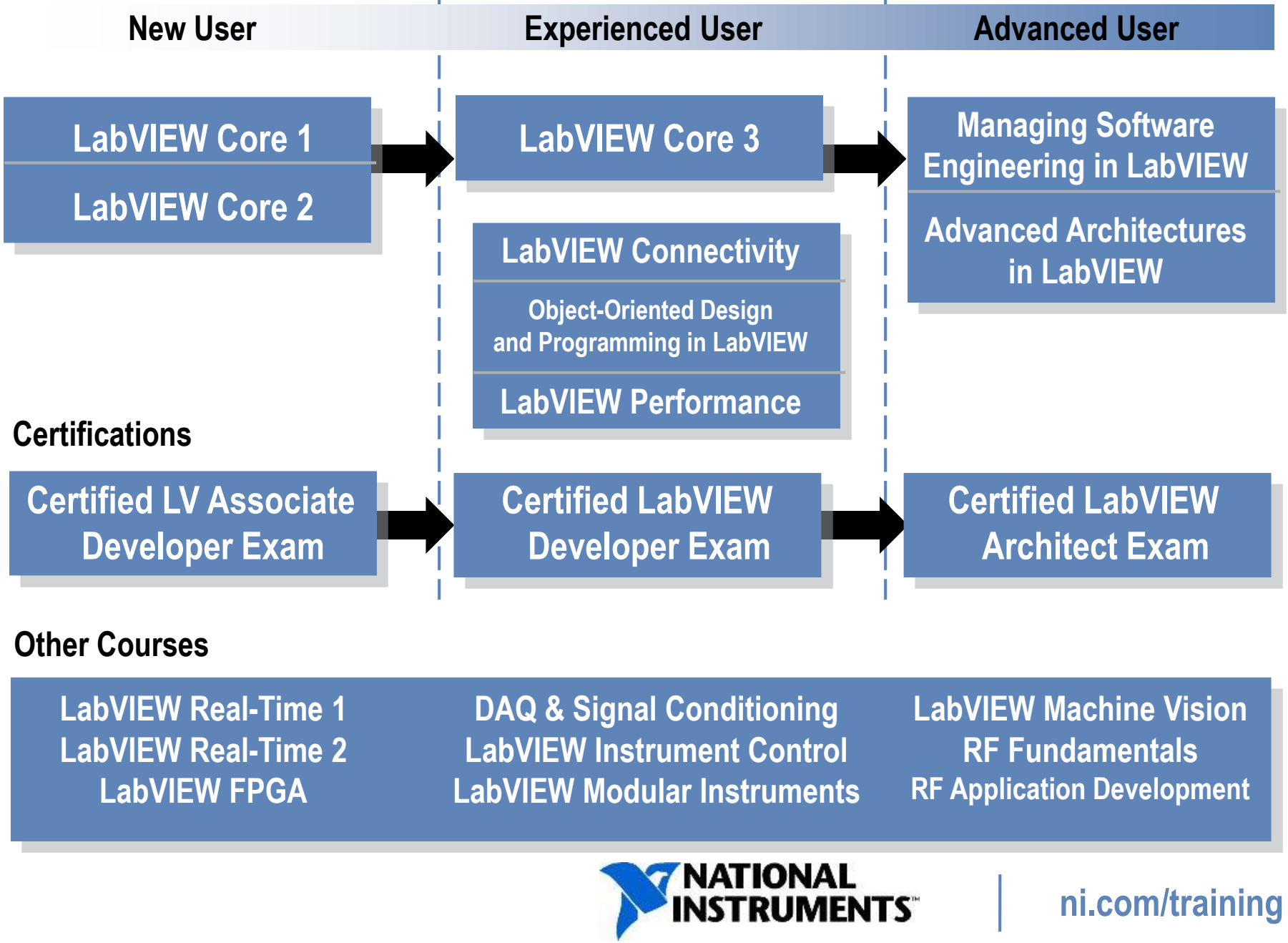


Instructional Methods



Getting The Most Out Of This Course

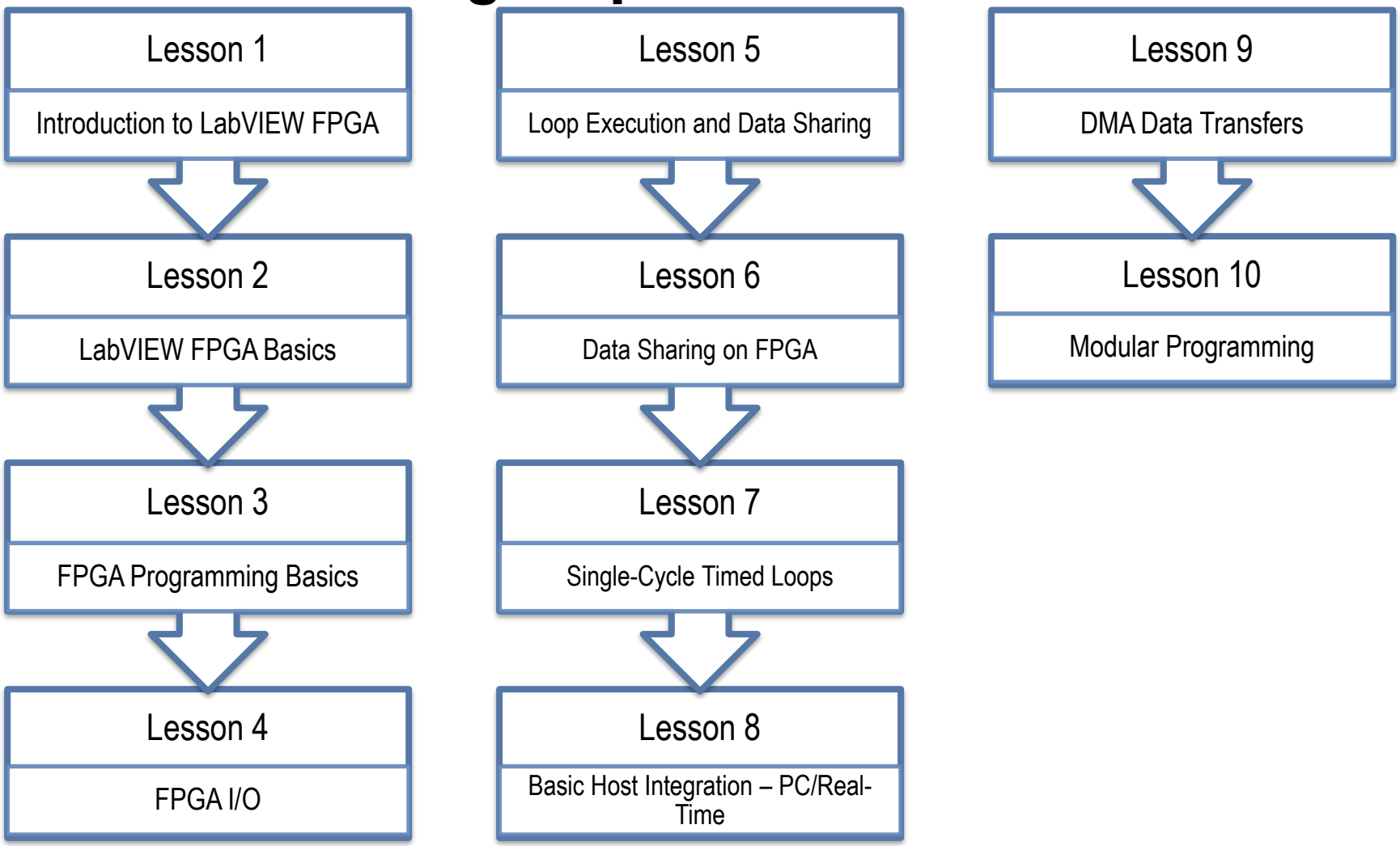
- Ask questions!
- Experiment with hands-on exercises to understand the methods used
- Explore solutions
- Implementations explore a possible solution—you may find a better one



NATIONAL INSTRUMENTS

ni.com/training

Course Learning Map



Course Goals

This course prepares you to:

- Select and configure NI Reconfigurable I/O (RIO) hardware
- Create, compile, download, and execute a LabVIEW FPGA VI and use NI RIO hardware
- Perform measurements using analog and digital input and output channels
- Create host computer programs that interact with FPGA VIs
- Understand and control timing of operations on the FPGA target
- Design and implement applications using the LabVIEW FPGA module

Configuring Your LabVIEW Environment

- Options Dialog Box
  - Controls/Functions Palettes page
    - Select **Load palettes during launch** to make Search Palettes immediately usable after launch
    - Set Palette to **Category (Icons and Text)**
  - Block Diagram page
    - Uncheck **Place front panel terminals as icons** to place control and indicator terminals in a compact format
    - Configure **Block Diagram Cleanup** to customize your block diagram

Configuring Your LabVIEW Environment

- Functions Palette
  - Tack the Functions palette and select **View»Change Visible Categories** then click **Select All**
- Controls Palette
  - Tack the Controls palette and select **View»Change Visible Categories** then click **Select All**

Lesson 1
Introduction to LabVIEW FPGA

TOPICS

A. Introduction to FPGA Technology
B. LabVIEW FPGA System
C. Comparison with a DAQmx System
D. LabVIEW FPGA Applications

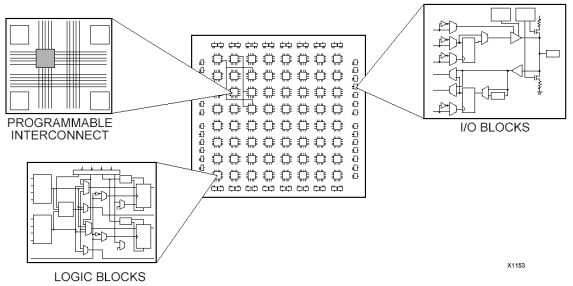
A. Introduction to FPGA Technology

What is an FPGA?

- Field programmable gate array (FPGA)
- A silicon chip with unconnected gates and other hardware resources
- Enables user to define and re-define functionality

How does an FPGA work?

- Circuit behavior is defined using software
- Circuit specification (gate connection, etc.) is loaded into the hardware
- No OS is needed for execution of logic



When is an FPGA used?

- Custom hardware, where it doesn't make sense to pay the high price of developing an ASIC
- Reconfiguration required after deployment

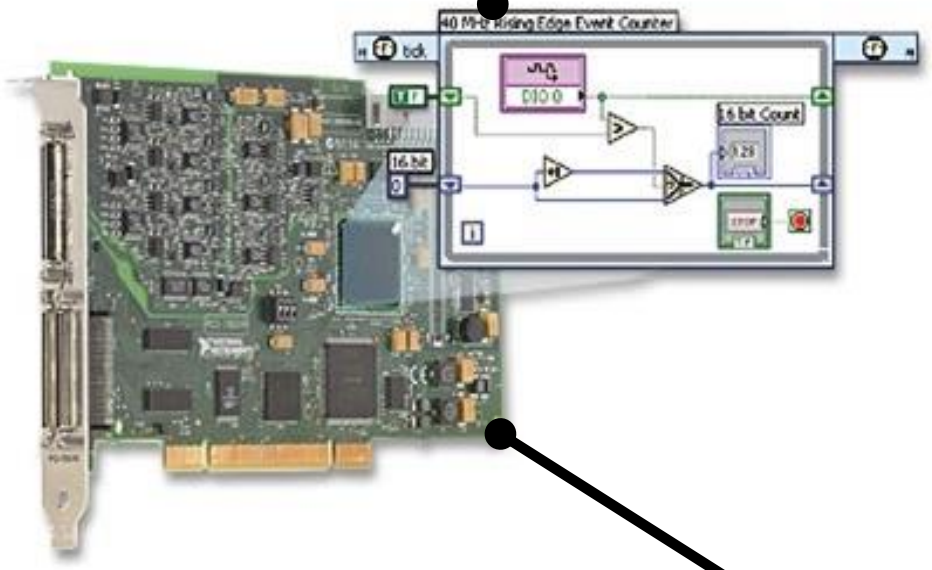
Benefits of FPGA

- Flexibility
  - Reconfigurable through software
- True parallel processing
  - Simultaneous parallel circuits
  - No CPU time sharing
- High Performance
- Reliability
- Offload processing
- Cost

B. LabVIEW FPGA System

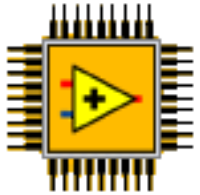
LabVIEW FPGA Module

NI-RIO Driver



Reconfigurable I/O (RIO) Hardware

LabVIEW FPGA Module



- Add-on module for LabVIEW
- Develop VIs for FPGA target
- Develop VIs for host PC or Real-Time interaction with FPGA

LabVIEW FPGA Targets

CompactRIO

PXI R Series Multifunction RIO

PCI R Series Multifunction RIO

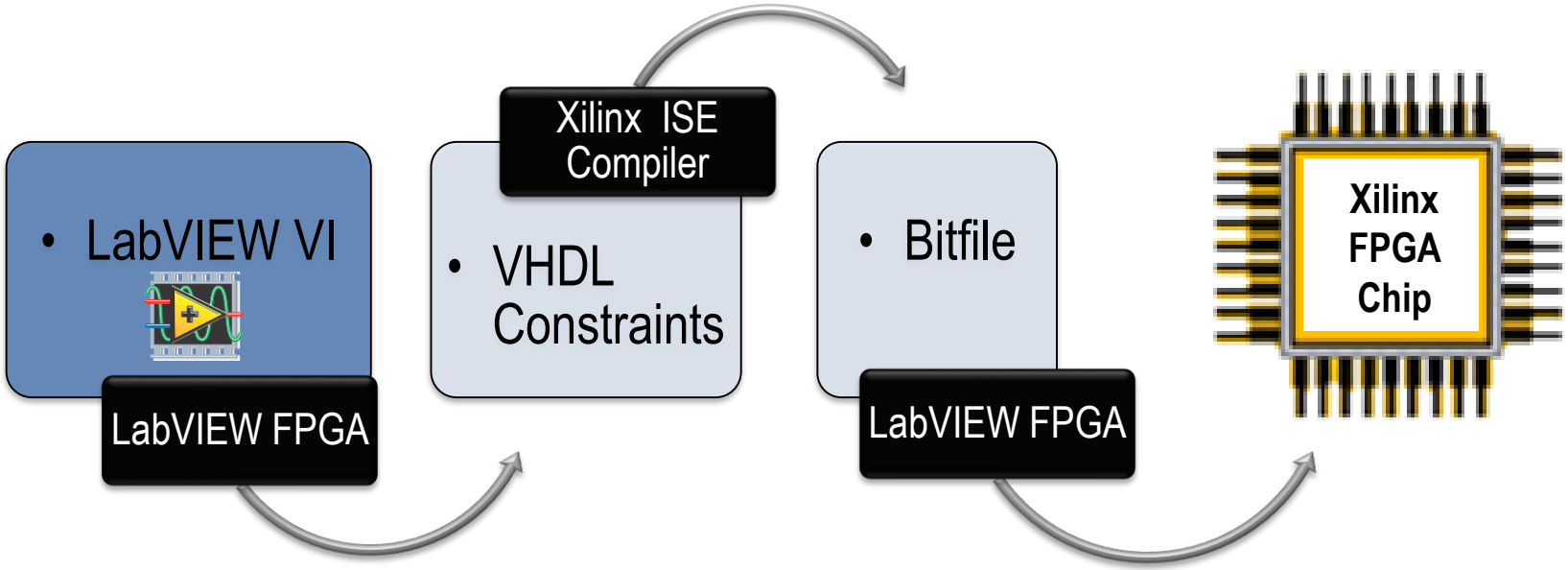
Compact Vision System

Single-Board RIO

FlexRIO

RIO Instruments

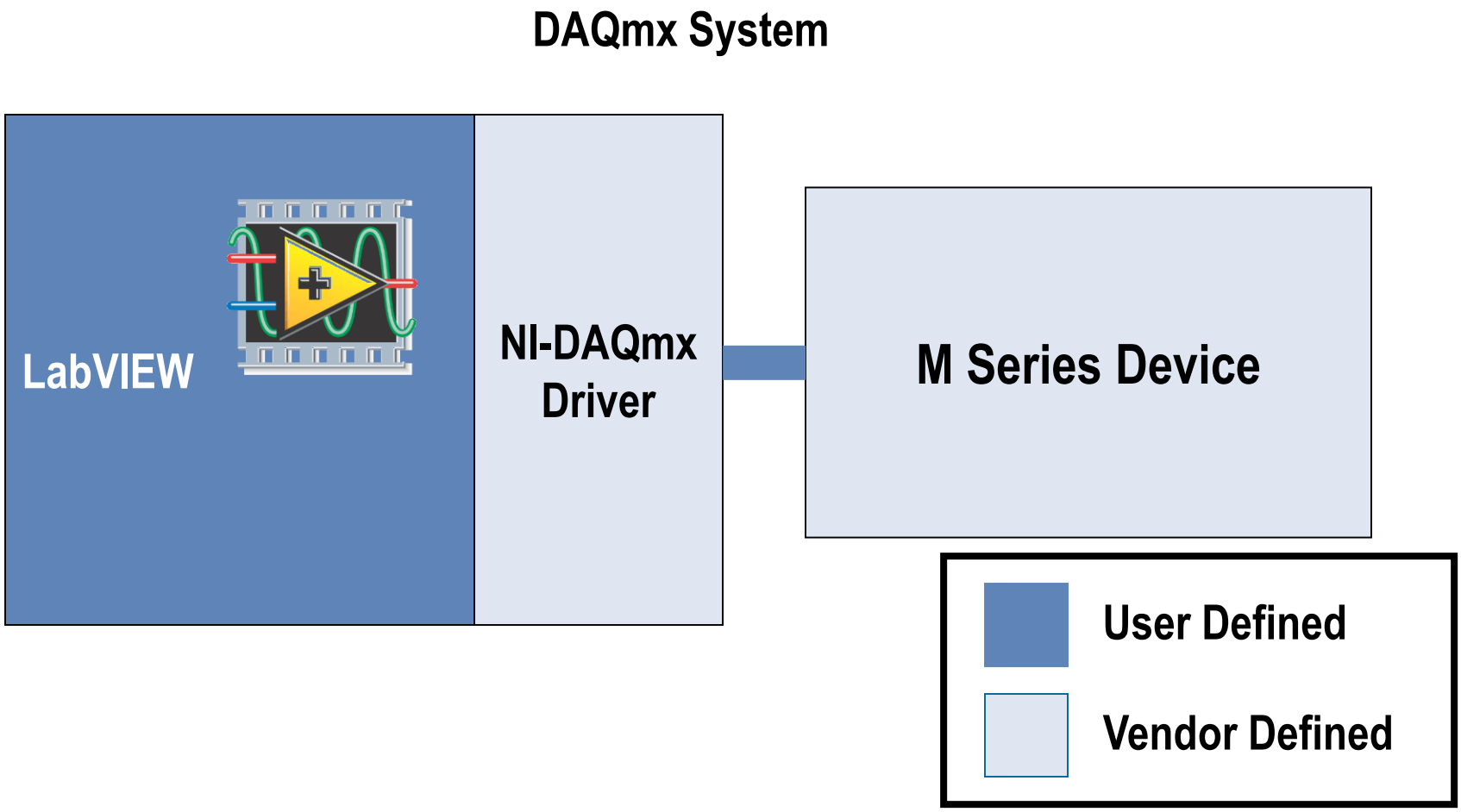
LabVIEW FPGA: How does it work?



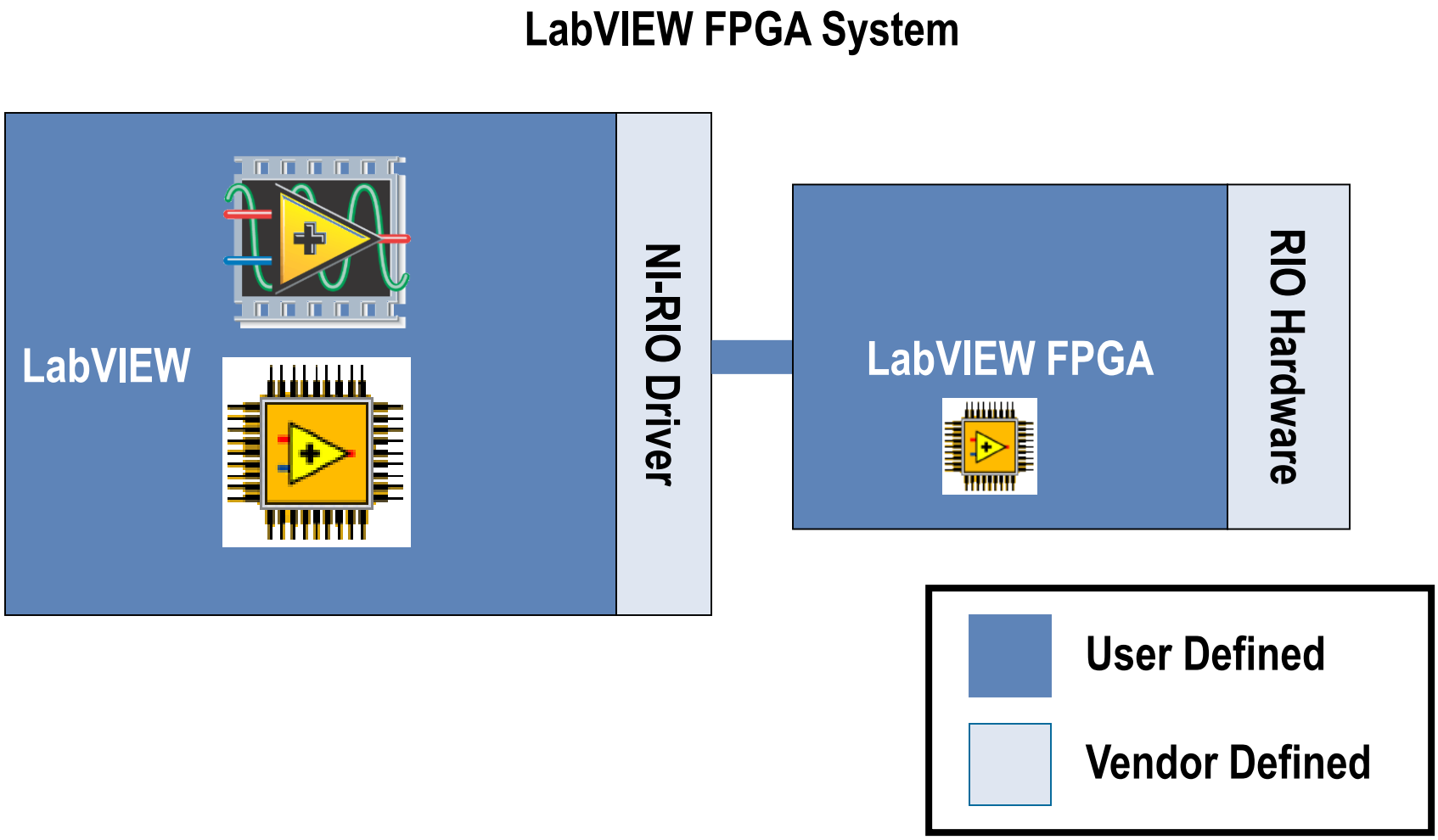
Benefits of LabVIEW FPGA System

- FPGA technology
- Create custom hardware using LabVIEW
  - No Verilog/VHDL coding or board design
- Direct access to hardware terminals
- Extensive library of built-in functions
- Integration with 3rd party IP
- Tools to communicate, monitor and control the FPGA from Windows PC or Real-Time controller

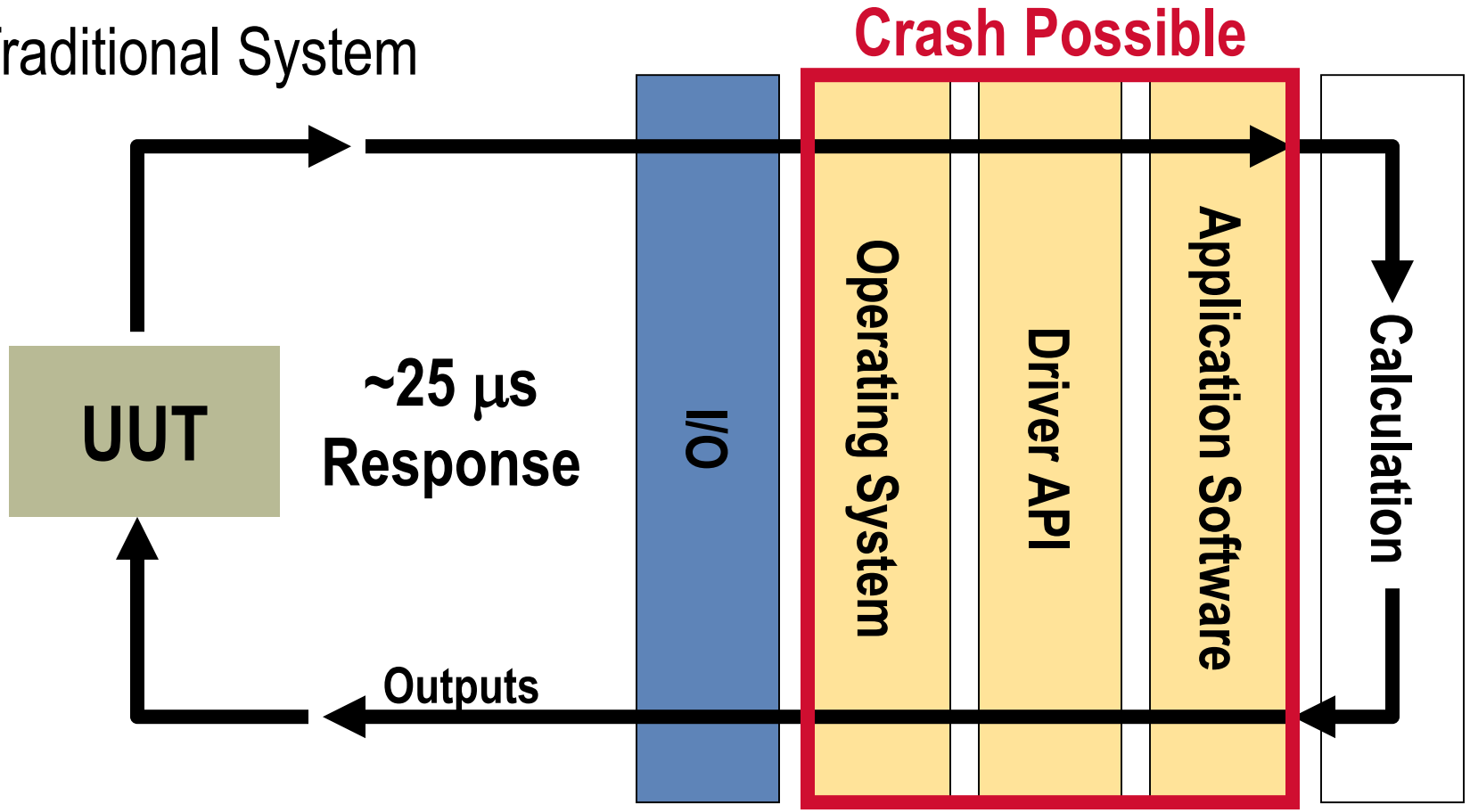
C. Comparison with NI-DAQmx System



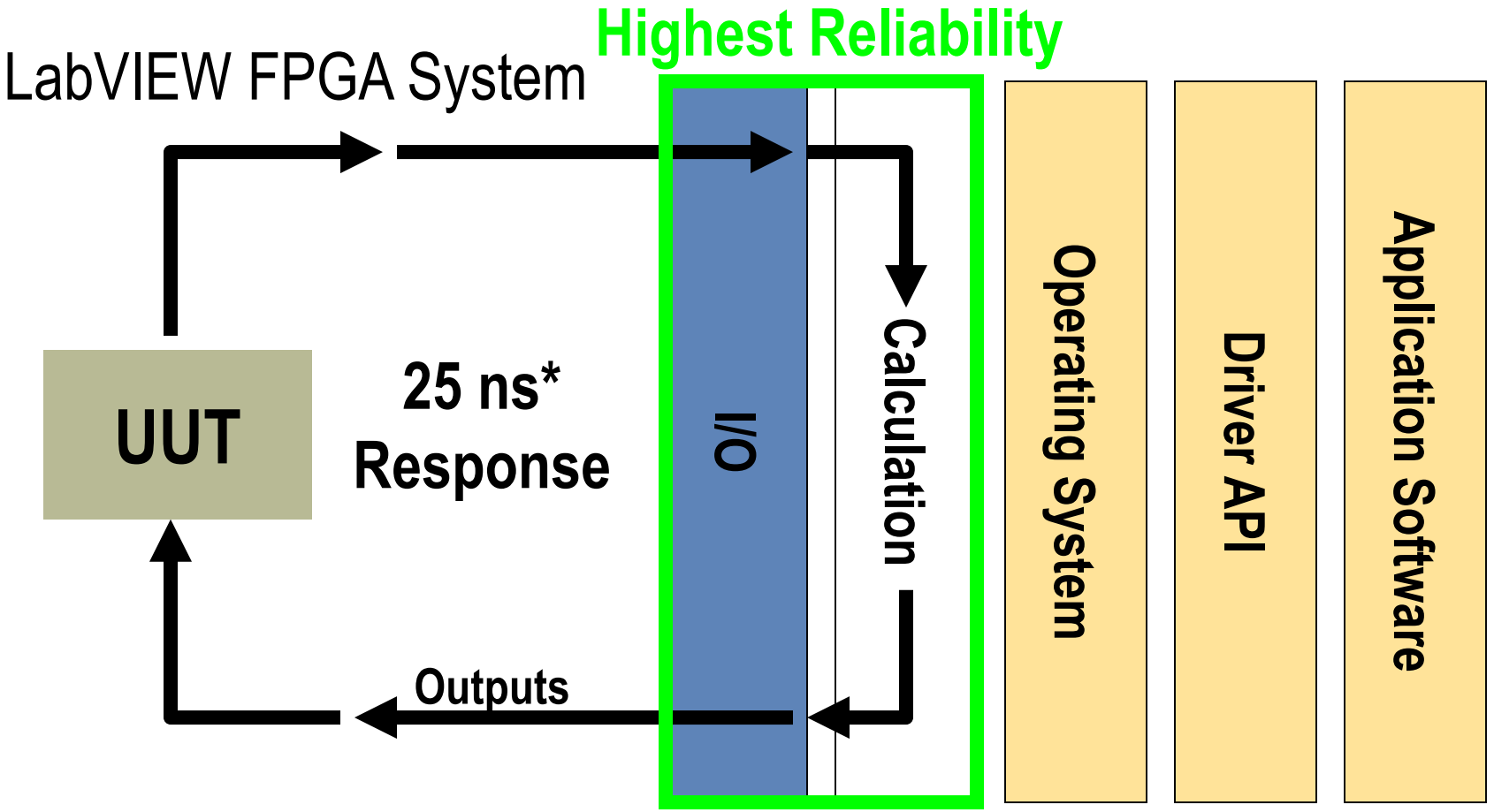
LabVIEW FPGA System



Decision Making in Software



Decision Making in Hardware



* Faster response for 80 and 120 MHz clocks

D. LabVIEW FPGA Applications

Typical LabVIEW FPGA applications

- Intelligent DAQ
  - Custom Timing and Synchronization
  - Custom Counters
  - Multiple Scan Rates
- Ultra-high speed control
- Specialized communication protocols
- Off-load CPU – Processing
- Complex timing and synchronization
- Hardware-in-the-Loop (HIL) testing

Summary—Quiz

1. Which of the following are required software components of a LabVIEW FPGA system?
   a. NI-RIO driver
   b. DAQmx driver
   c. LabVIEW Development System
   d. LabVIEW Real-Time Module
   e. LabVIEW FPGA Module

Summary—Quiz Answer

1. Which of the following are required software components of a LabVIEW FPGA system?
   a. **NI-RIO driver**
   b. DAQmx driver
   c. **LabVIEW Development System**
   d. LabVIEW Real-Time Module
   e. **LabVIEW FPGA Module**

Summary—Quiz

2. What OS runs on the FPGA?
   a. LabVIEW FPGA
   b. MicroLinux
   c. Unix
   d. PharLap
   e. None of the above

Summary—Quiz Answer

2. What OS runs on the FPGA?
   a. LabVIEW FPGA
   b. MicroLinux
   c. Unix
   d. PharLap
   e. None of the above

Summary—Quiz

3. LabVIEW FPGA allows you to program an FPGA without knowing VHDL or HDL.
   a. True
   b. False

Summary—Quiz Answer

3. LabVIEW FPGA allows you to program an FPGA without knowing VHDL or HDL.
   a. **True**
   b. False

Summary—Quiz

4. Which of the following might be typical reasons for using R Series Multifunction RIO device over a DAQ board programmed with DAQmx?
   a. Need more counters than available on a DAQ board
   b. Need custom trigger logic
   c. Need more analog input channels than available on a DAQ board
   d. Need to simultaneously acquire data on different channels at different rates
   e. Need a high-level/easy-to-use driver

Summary—Quiz Answers

4. Which of the following might be typical reasons for using R Series Multifunction RIO device over a DAQ board programmed with DAQmx?
   a. **Need more counters than available on a DAQ board**
   b. **Need custom trigger logic**
   c. Need more analog input channels than available on a DAQ board
   d. **Need to simultaneously acquire data on different channels at different rates**
   e. Need a high-level/easy-to-use driver