Introduction to NI LabVIEW and Computer-Based Measurements

Augusto Mandelli

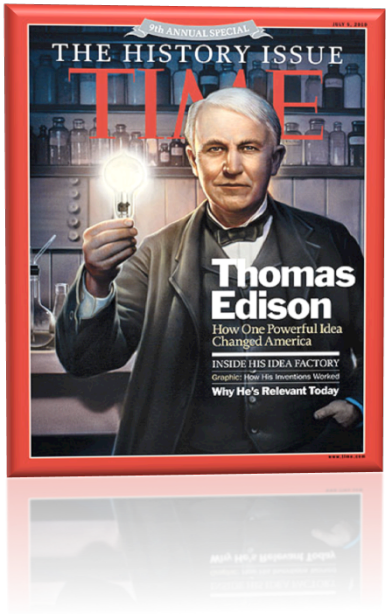
National Instruments

National Instruments—Our Stability

- **Non-GAAP Revenue:** $1.1 B in 2013
- **Global Operations:** Approximately 6,300 employees; operations in more than 40 countries
- **Broad customer base**: More than 35,000 companies served annually
- **Diversity**: No industry >15% of revenue
- **Culture**: Ranked among top 25 companies to work for worldwide by the Great Places to Work Institute
- **Strong Cash Position**: Cash and short-term investments of $377M as of March 31, 2012

*A reconciliation of GAAP to non-GAAP results is available at investor.ni.com

Thomas Edison…born in the (18)40s



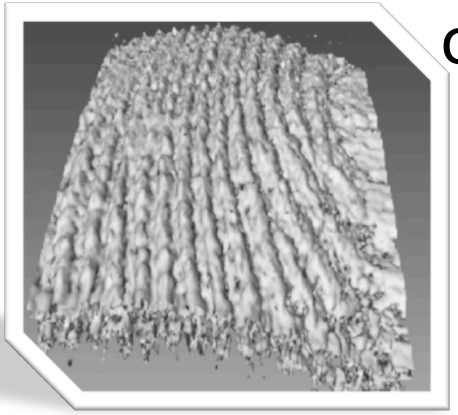
- Very fortunate and exciting 84 years of history….
  - First practical and commercial light bulb
  - Championed distributed electrical power to the masses
  - First practical phonograph

Many inventions "improvements" over previous ideas

The Impact of Great Engineering

Saving time, effort, and money

Improving quality of life

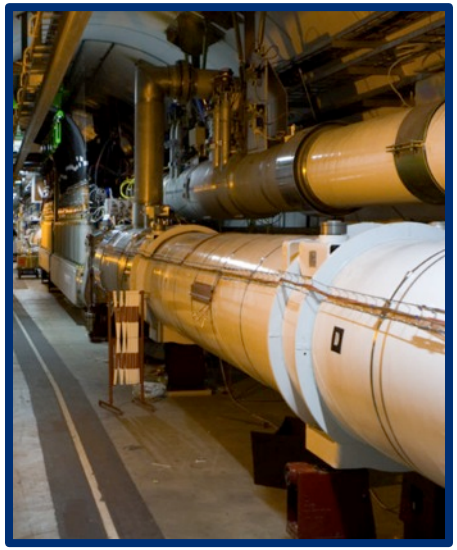
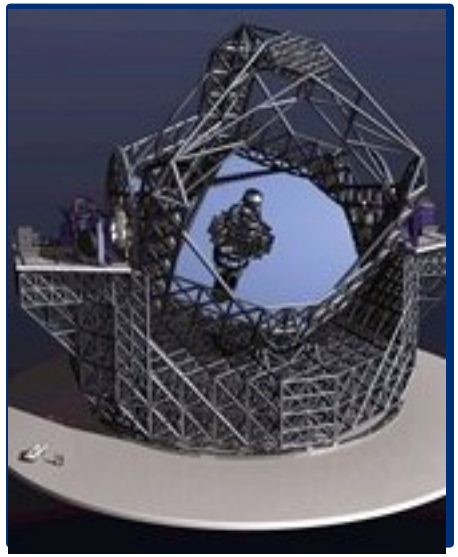


Averting catastrophic damage

Today's Engineering Challenges

- Minimizing power consumption
- Managing global operations
- Getting increasingly complex products to market faster
- Maximizing operational efficiency
- Adapting to evolving application requirements
- Protecting investments
- Doing more with less
- Integrating code and systems



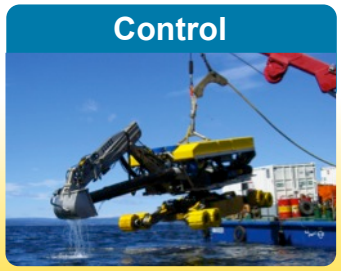
Measurement Challenges Are Compounded By:

- Compressed Timelines
- Fixed Software and Hardware
- Conflicting Programming Approaches
- Inadequate Hardware Performance
- Disparate Driver APIs
- Varying Sensors and Connectivity
- Custom Signal Conditioning
- Advanced Visualization
- Changing Application Requirements
- Complex Analysis Algorithms
- Evolving Technology Trends
- Confusing Data Storage
- Differing Sampling Rates

NATIONAL INSTRUMENTS

National Instruments' Strategy: Graphical System Design

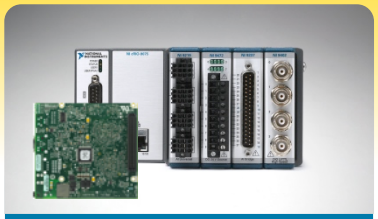
Your Investment in a Platform-Based Approach to Measurements Scales Across…



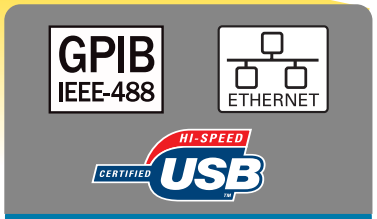
NATIONAL INSTRUMENTS LabVIEW™

Desktops and PC-Based DAQ

PXI and Modular Instruments



NI CompactRIO



Open Connectivity With Third-Party I/O

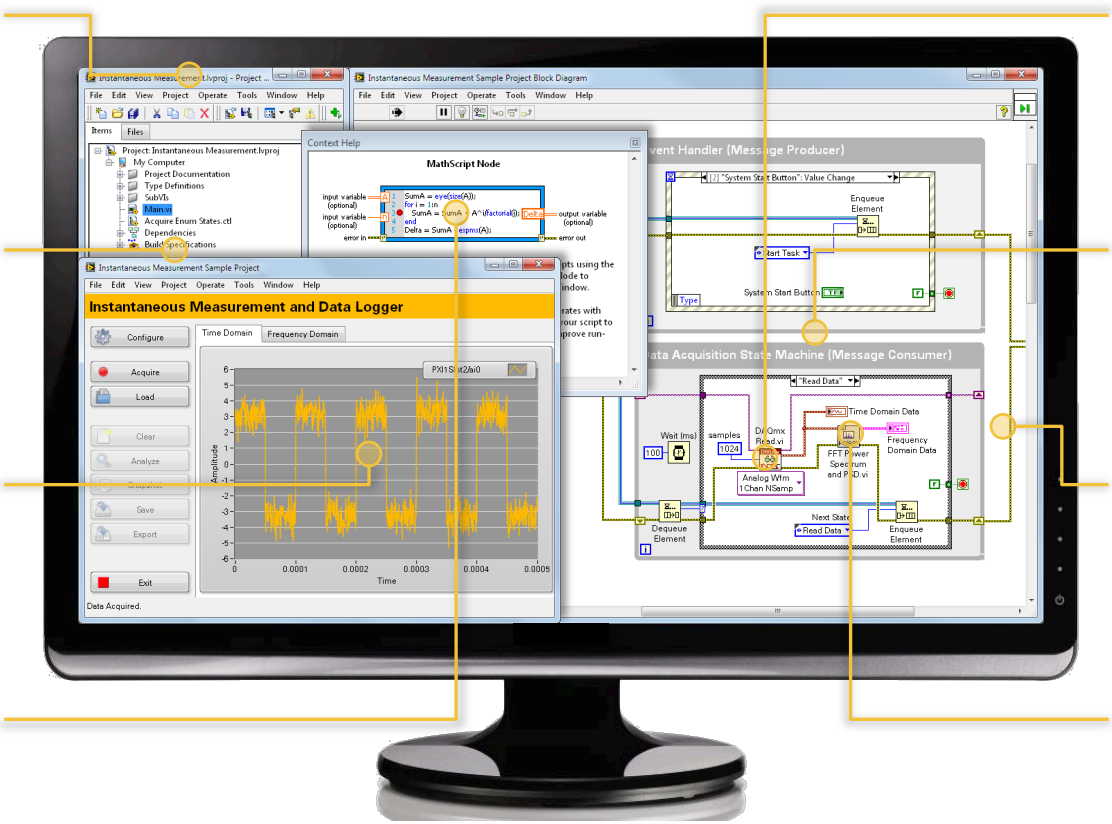
LabVIEW Overview

Unified Software Solution
Manage and organize all system resources in a single software environment.

Deployment Targets
Deploy LabVIEW code to the leading desktop, real-time, and FPGA hardware targets.

Convey With a Clear UI
Create modern user interfaces to display measurements and results.

Integrate Existing Code
Combine and reuse .m files, C code, and HDL with graphical code.



Hardware Connectivity
Bring real-world signals into LabVIEW from any I/O on any instrument.

Parallel Programming
Easily create independent loops that automatically execute in parallel.

Measure in Minutes
Reduce development time with abundant sample projects and templates.

Analysis Libraries
Use built-in high-performance analysis libraries designed for measurement applications.

LabVIEW abstracts low-level complexity and integrates all of the tools engineers and scientists need to build **any** measurement or control system.

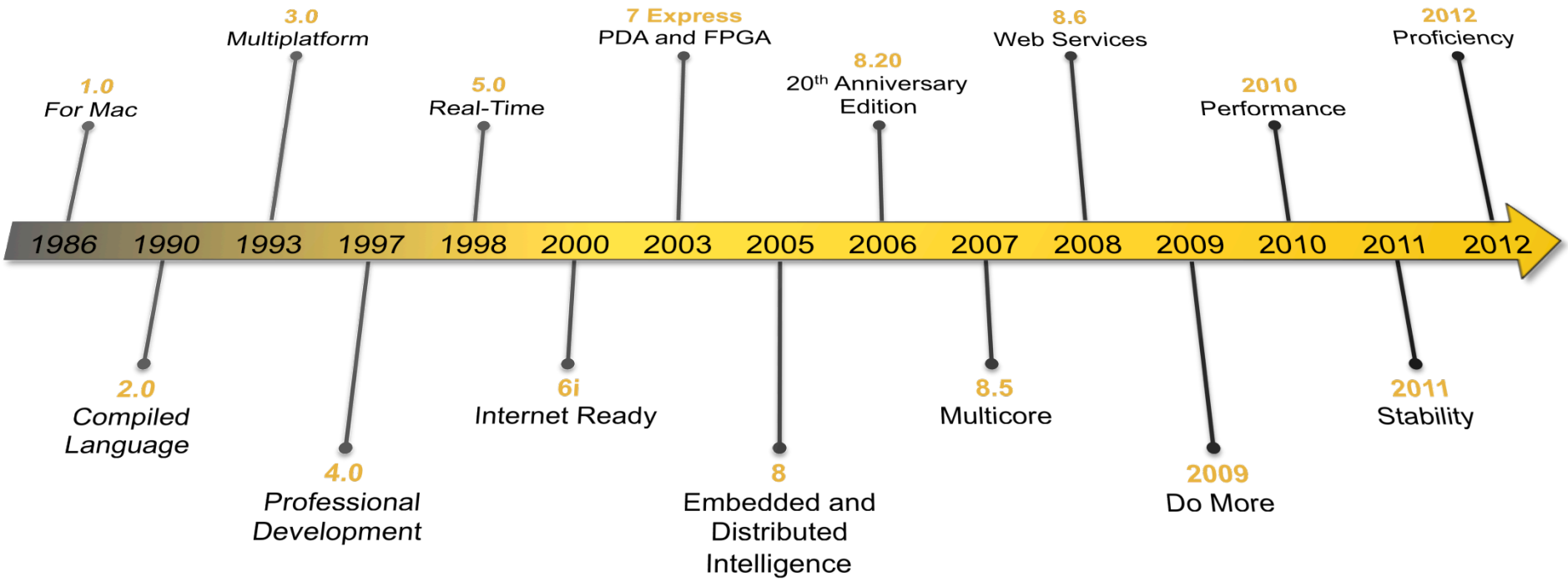
Unrivaled Hardware Integration in a Single Environment

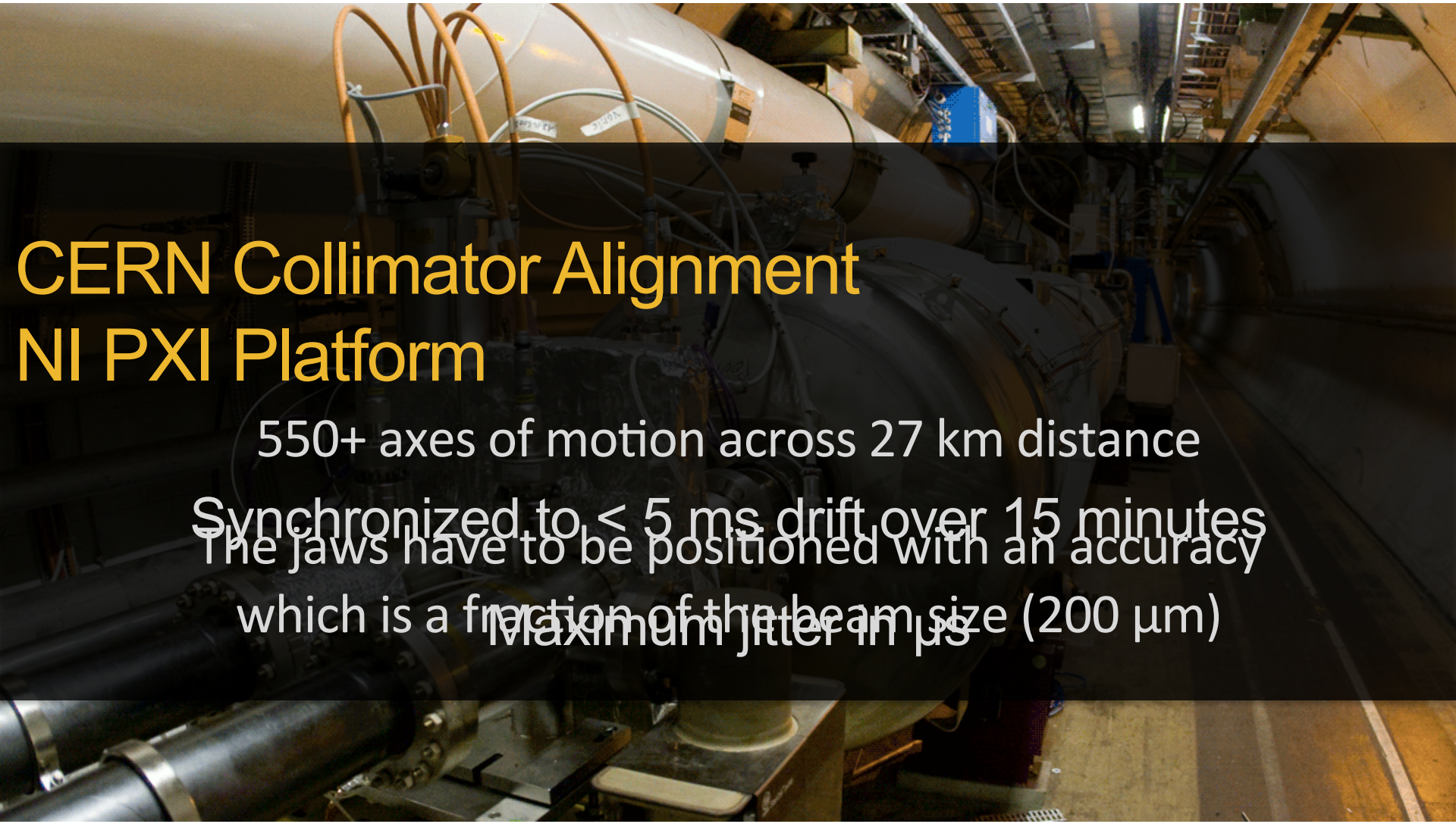
- NI hardware
  - 200+ data acquisition devices
  - 450+ modular instruments
  - Cameras
  - Motion control
- Third-party hardware
  - Instrument Driver Network
    - 10,000+ instrument drivers
    - 350+ instrument vendors
    - 100+ instrument types
  - Communicate over any bus

Because It Has Been Proven Over Nearly 30 Years…

Withstanding the test of time across operating systems, buses, technologies, and more

NATIONAL INSTRUMENTS LabVIEW™

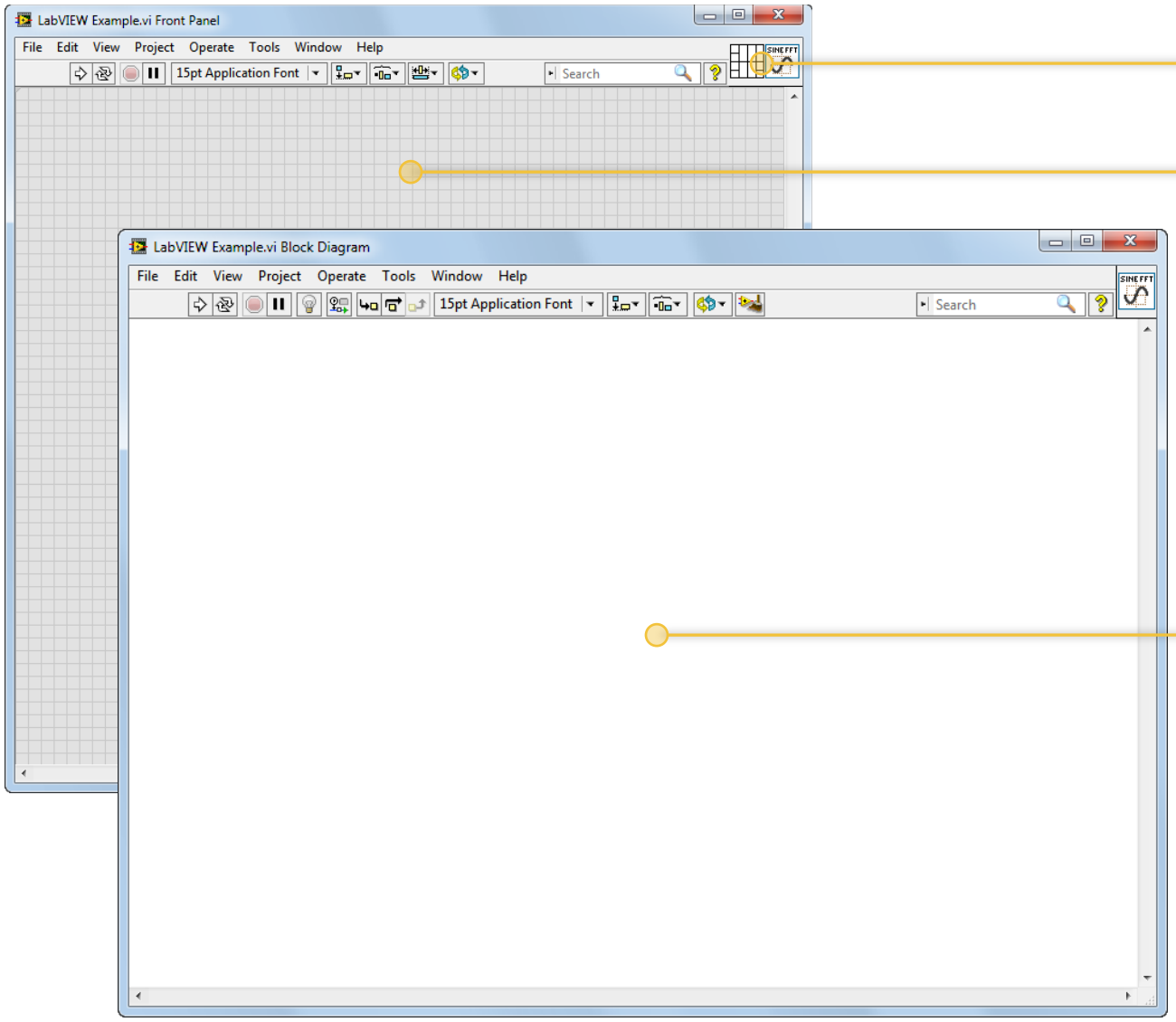



CERN Collimator Alignment
NI PXI Platform
550+ axes of motion across 27 km distance
Synchronized to < 5 ns drift over 15 minutes
The jaws have to be positioned with an accuracy
which is a fraction of the beam size (200 µm)
Maximum jitter in µs

Introduction to LabVIEW

System Design Software for Any Measurement Application

Therefore, LabVIEW Building Blocks Are Called Virtual Instruments (*.VI)



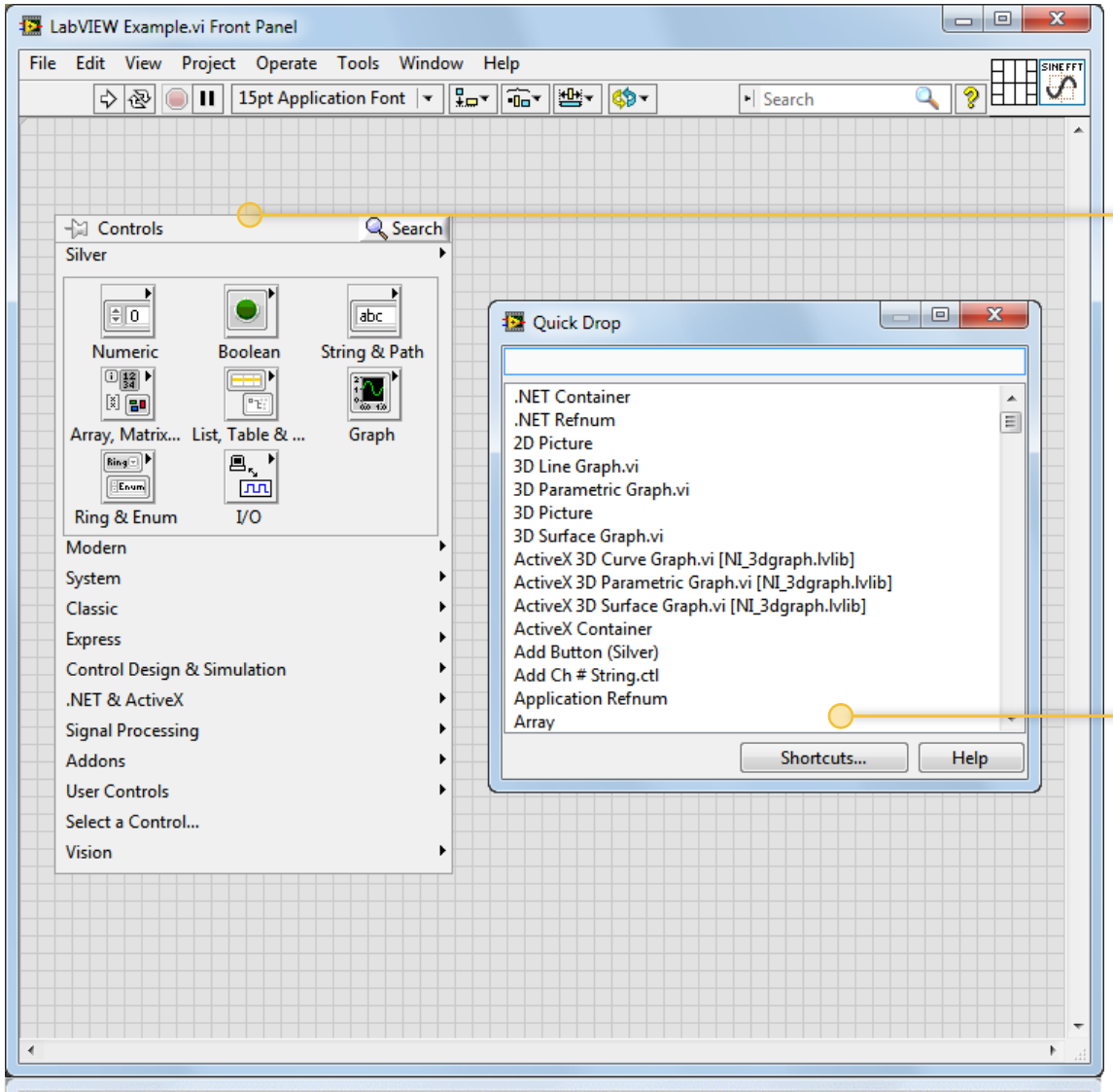
Icon / Connector Pane
Maps inputs and outputs

LabVIEW Front Panel
The user interface of a VI

LabVIEW Block Diagram
The source code of a VI

*Note: A *.vi file encapsulates all three elements*

Creating a LabVIEW Front Panel

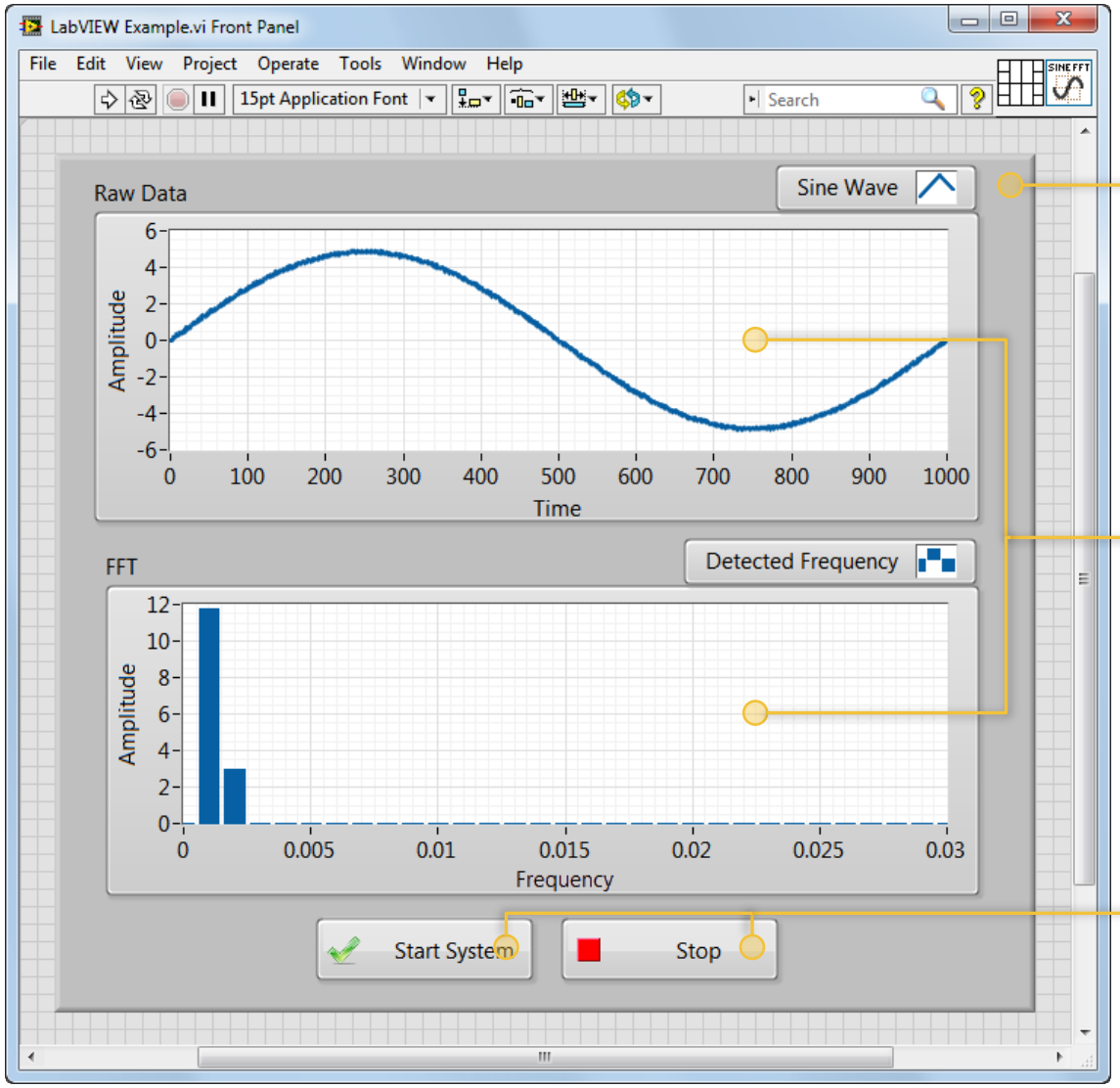


Controls Palette (Right-Click)
Access a hierarchical palette of all front panel elements.

Quick Drop (Ctrl + Space)
Search by object name.

NATIONAL INSTRUMENTS

Front Panel Objects



Decorations

Decorative elements and imagery

- Text
- Arrows
- Callouts
- Lines
- Images
- …and more

Customizable Indicators

Used to convey outputs to a user

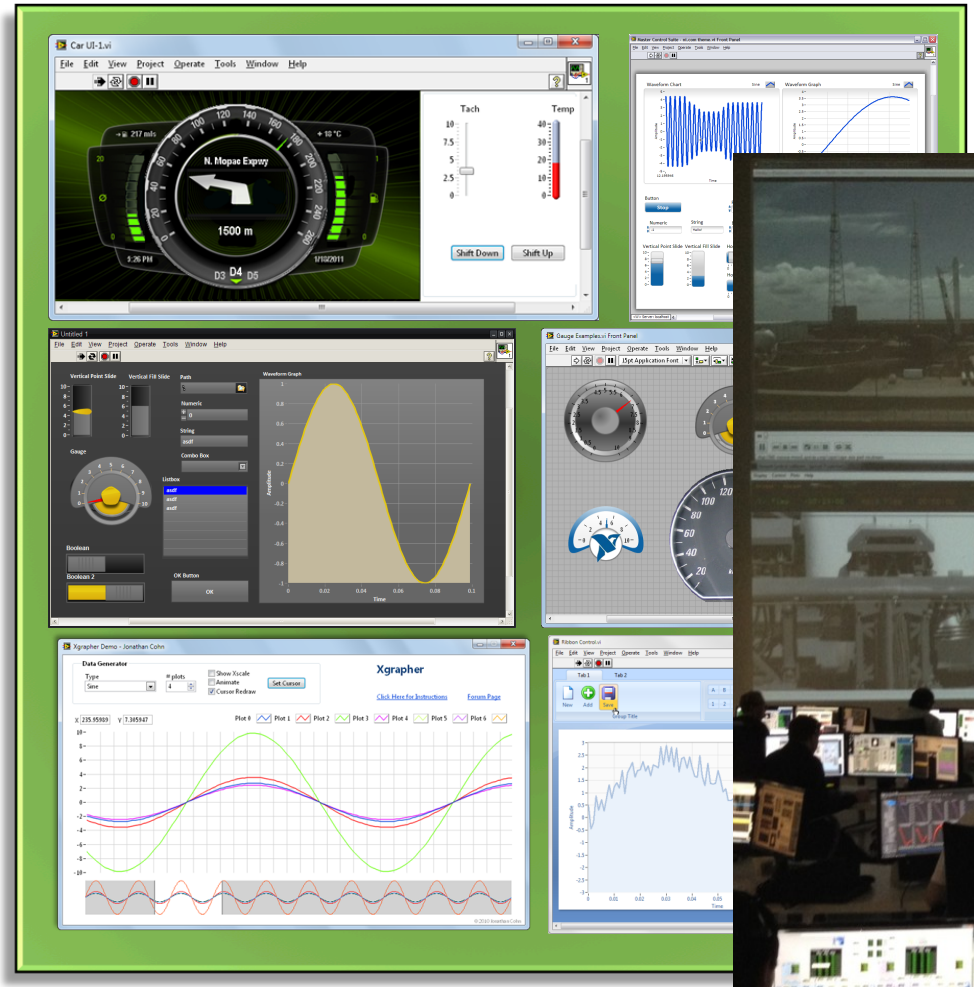
- Graphs and Charts
- Progress Bars
- Gauges and Meters
- LEDs
- Numerics
- Strings and Paths
- …and more

Customizable Controls

Used to receive input from a user

- Knobs and Dials
- Sliders
- Buttons
- Numerics
- Strings and Paths
- …and more

LabVIEW Front Panels in Action

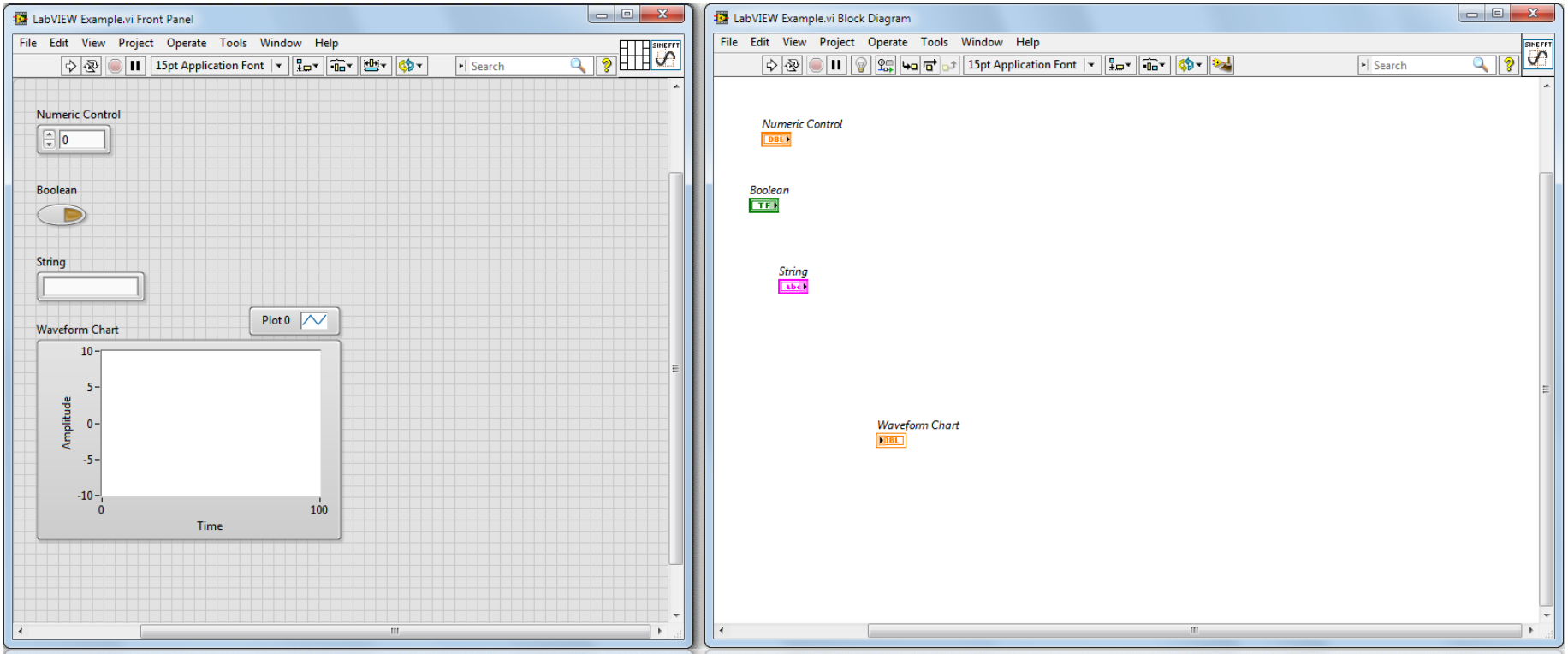


All of the front panels above were contributed for sharing and reuse by members of the global LabVIEW community.

Dozens of LabVIEW front panels at SpaceX Mission Control during successful launch of Dragon
Photo Credit: Elon Musk

All Front Panel Elements Have Block Diagram Terminals

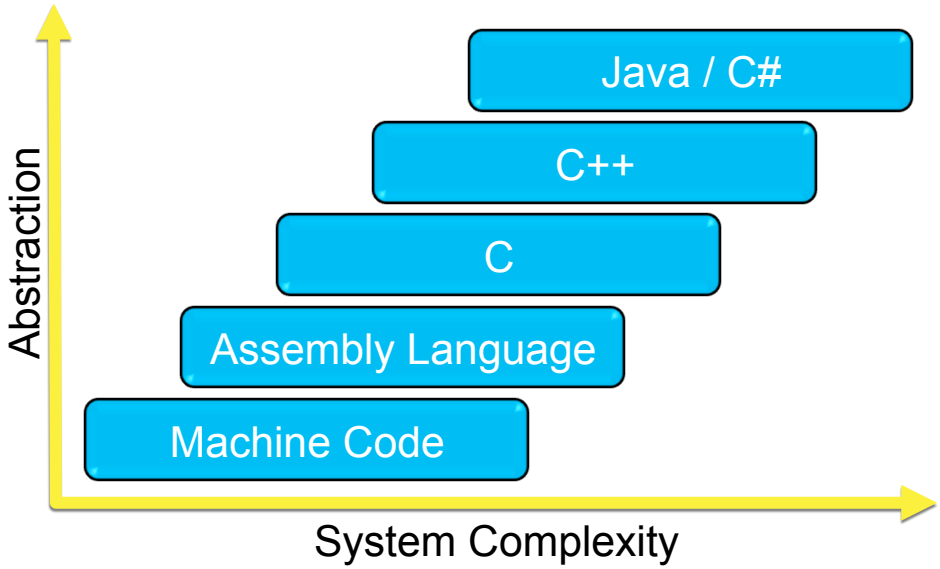
Block diagram terminals provide access to front panel values



NATIONAL INSTRUMENTS

Examining Traditional Source Code

Humans use abstracted languages because machine code is too hard to comprehend



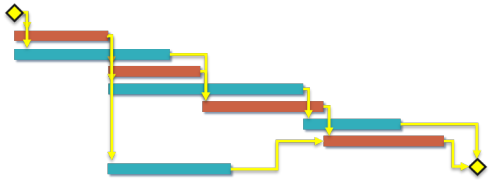

```
Bool32 CanProbeSignal(OHHandle h, ObjID s)
{
   ObjList *lp;
   ObjID tl, t = 0L;
   int16 *tdp=NULL;
   InstrHandle vi;
   if(s)
   {
      tl= SignalPtr(h,s)->termList;
      DAssert(tl, 0);
      lp= ObjListPtr(h, tl);
      t= lp->nObj? lp->obj[0] : 0;
   }
   if(!t)
      return FALSE;
   tdp= TDPtrFromID(h, oGetDataType(h, t));
   vi = HeapVI(h);
   return DefinedType(tdp) && ((*vi)->instrState & viDebugCapable)
         && !((*vi)->flags & parallelVIMask) && (!((*vi)->instrState & viRetrieveMask)
            || !(*(*vi)->retrieve)->wasEditing);
}

int32 SelectPopUpMethod(OHHandle h, ObjID s, Args *ap)
{
   int32 reply= 0L;
   if (reply = SelectSupPopUp(h, s, ap))
      return reply;
   ap->popup.level++;
   if (ap->popup.doWhat == kBuildPopUp)
   {
      Str255 buff;
      int32  i;
      AppendDiagramListMenu(h, s, ap->popup.level, ap->popup.menu, FALSE);
      if (lvConfigSettings.autoReInitSelectFeatureEnabled)
      {
         InsertObjMenuItem(ap->popup.level, ap->popup.menu, NULL, 0, -1);
         GetPopUpString(miSelNodeAutoInit, buff);
         i = InsertObjMenuItem(ap->popup.level, ap->popup.menu, buff, miSelNodeAutoInit, -1);
         MCheckItem(ap->popup.menu, i, (SelNodePtr(h,s)->flags & autoReInitSelMask));
      }
   }
   else if (ap->popup.doWhat == kSelectPopUp)
      reply= SeqSelPopUpSelect(h, s, ap, FALSE);
   return reply;
}
```


NATIONAL INSTRUMENTS

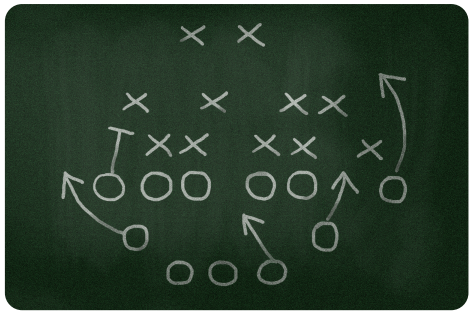
We Live in a Graphical, Parallel World

…But what if everything was represented using sequential, textual syntax?

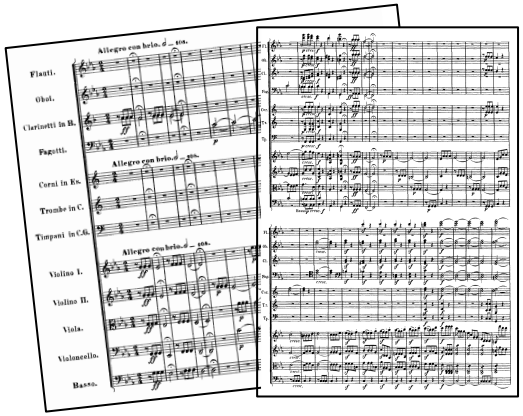
Gantt Chart



Football Play



Musical Score



Our World

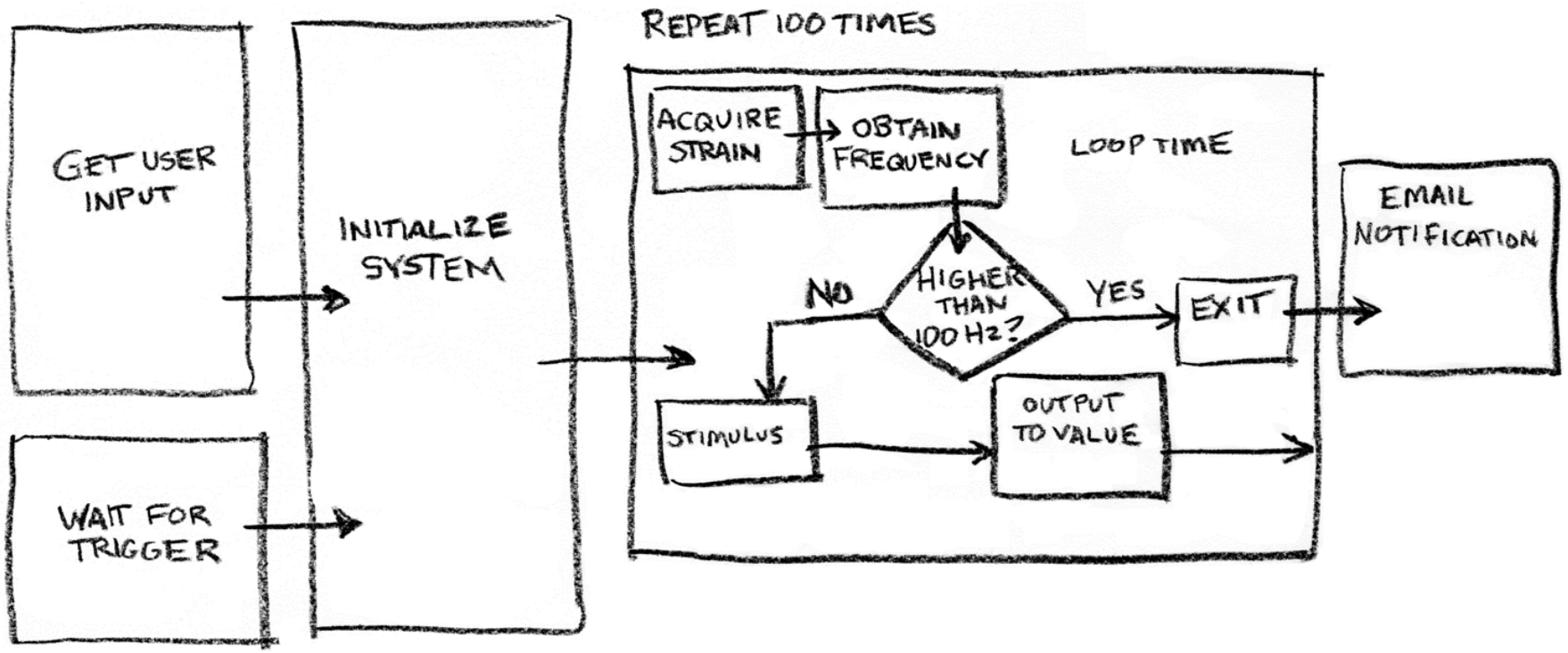
A World Without Graphical

Begin Project
Simultaneously Begin Tasks A and B
When Task A Ends,
Simultaneously Begin Tasks C, D, and H
When Tasks B and C Both End,
Begin Task E
When Task D Ends,
Begin Task F
When Task E Ends, If Task H has Ended,
Begin Task G
When Task F and G End,
Finish Project

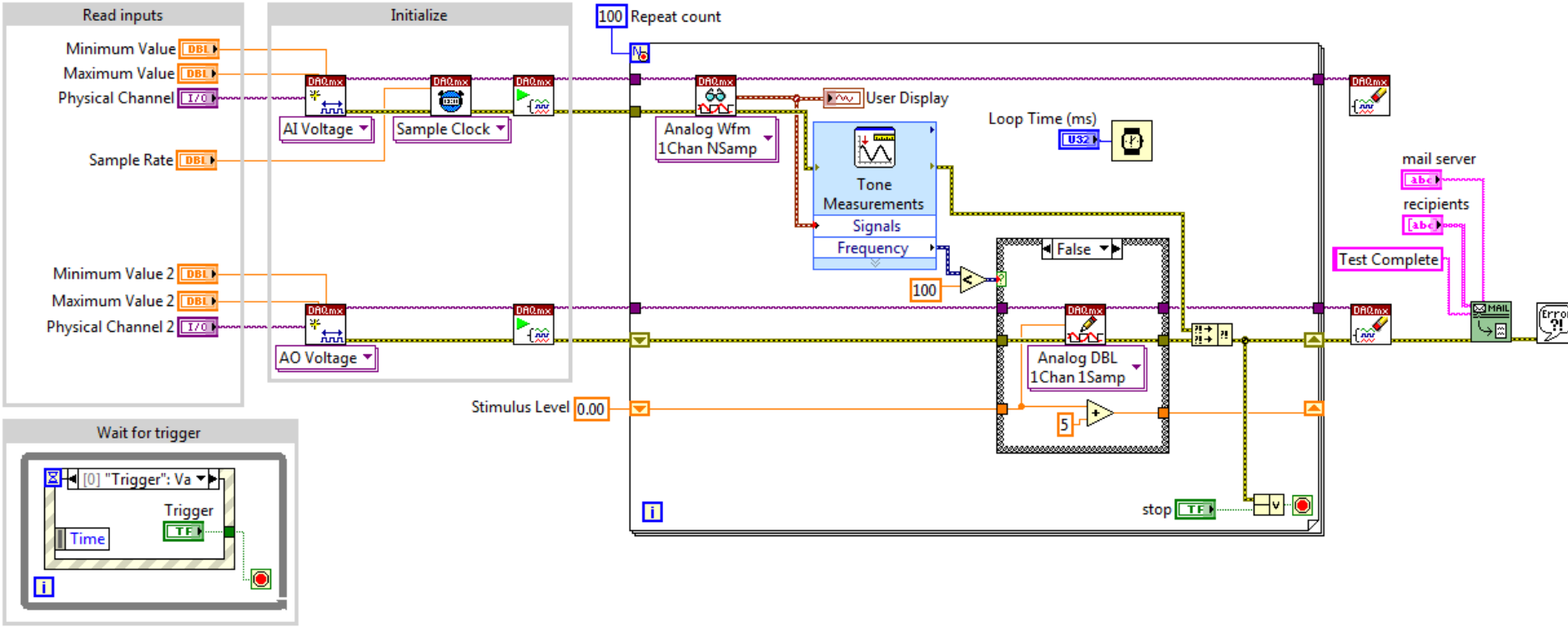
Align in Split-Back Formation
Center Hikes Ball to Quarterback
Simultaneously,
    Center Blocks Defensive Tackle
    Quarterback Hands Ball to Tailback
    Offensive Tackles 1-4 Block Defensive End
    Wide Receiver Right Runs In Route
    Wide Receiver Left Runs Screen Route
    Tight End Blocks Linebacker
    Tailback Runs Through Center Hole
    Fullback Blocks Middle Linebacker
End Play

Begin Song
Rest Two Beats in ¾ Time
While Three Iterations Haven't Been Played,
    Left Hand Plays Low C, G, and Middle C
    And Right Hand E, G, and High C
    Hold for Two Beats
    Pause for One Beat
    Left Hand Plays Low A, D, F
    And Right Hand Plays High F, A, F
    Hold for Three Beats
Repeat
End Song

With LabVIEW, You Can Program the Way You Think



With LabVIEW, You Can Program the Way You Think

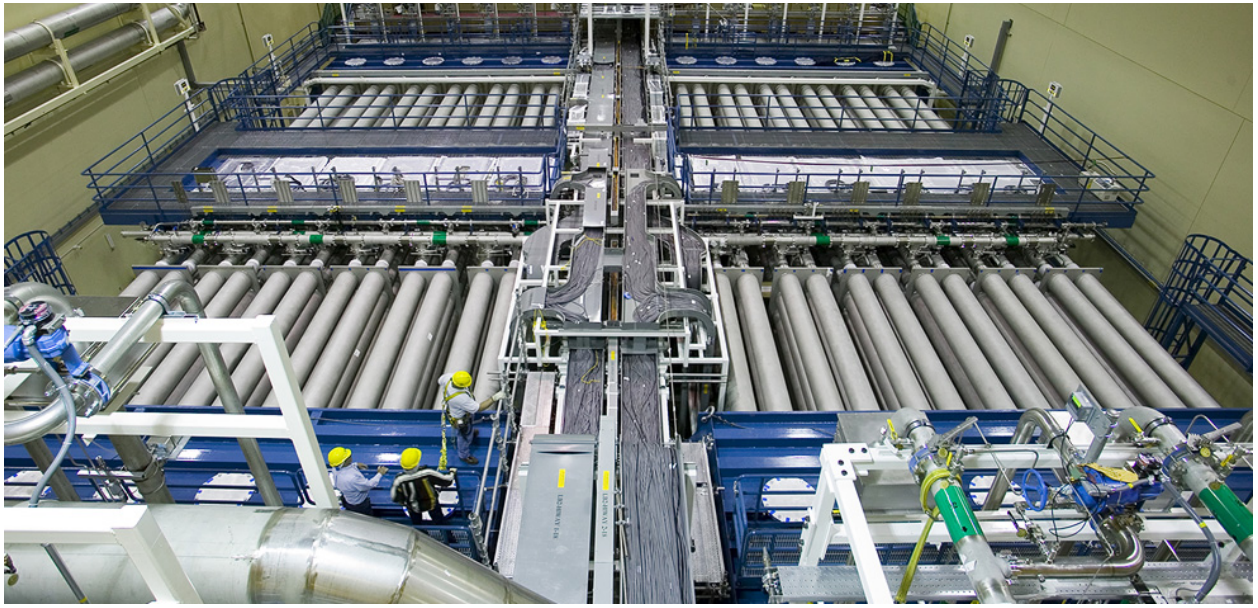


The graphical, dataflow-based G programming language is ideal for programming parallel data acquisition hardware.

Lawrence Livermore National Labs

Developed automated maintenance process for world's largest laser array at the National Ignition Facility using NI LabVIEW and PXI

- LabVIEW increased productivity by 3X over Java and C++
- Developed complex application consisting of over 1,000 VIs



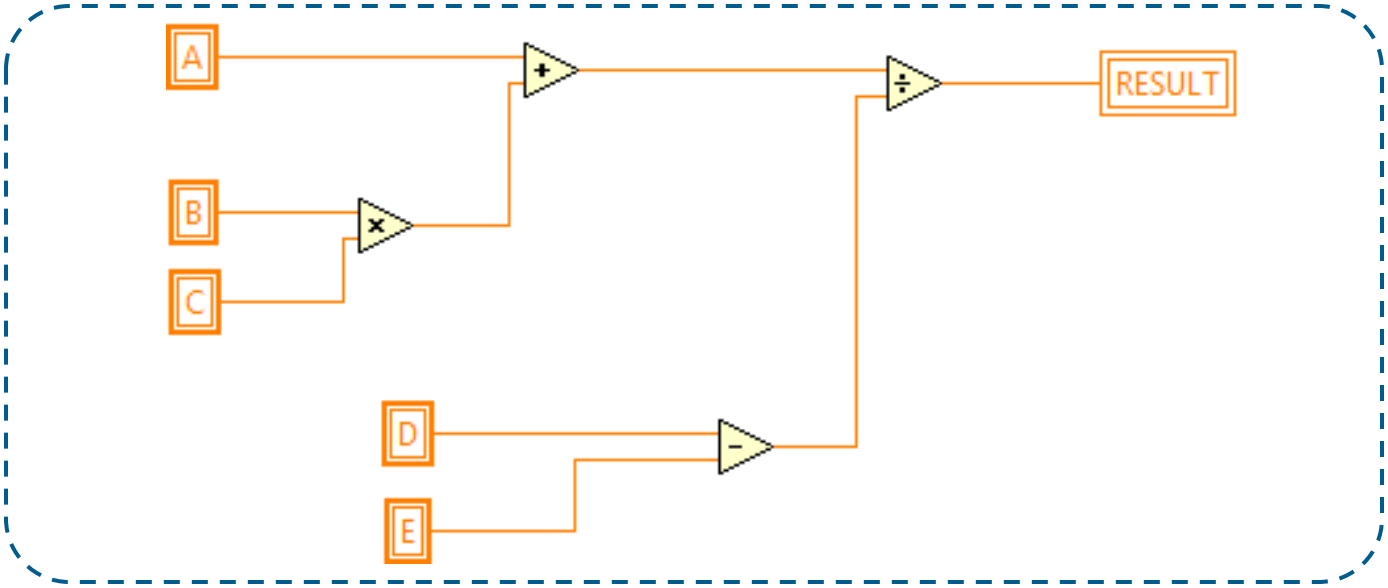
An overhead view of one of the main laser chambers

"The value in using the graphical dataflow language is the speed in which a team can deliver a robust solution while still using proper software engineering practices.

- Glenn Larkin, LLNL

What Is Data Flow?

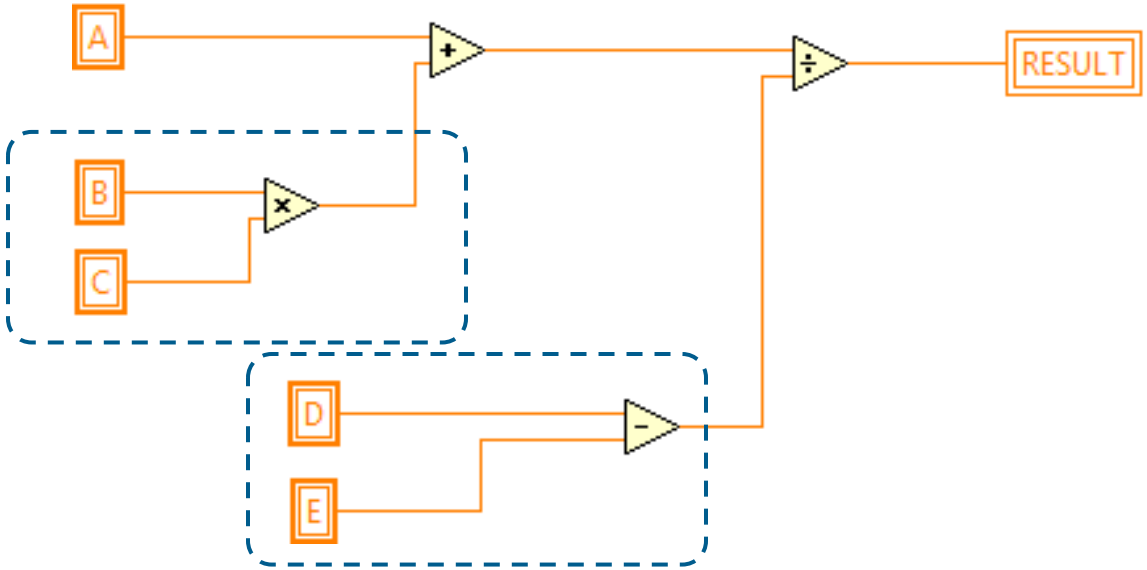
- Each block diagram node executes only when it receives all inputs
- Each node produces output data after execution
- Data flows along a path defined by wires
- The movement of data determines execution order



Formula: Result = (A+B*C) / (D-E)

What Is Data Flow?

- Each block diagram node executes only when it receives all inputs
- Each node produces output data after execution
- Data flows along a path defined by wires
- The movement of data determines execution order

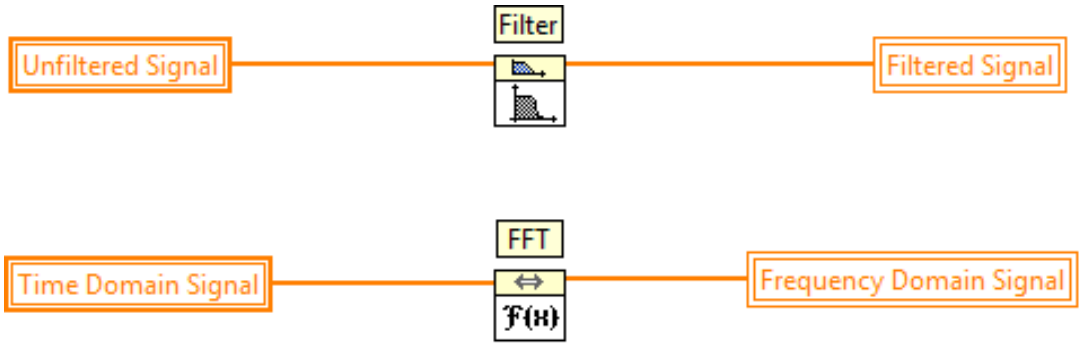


The [Multiply] and [Subtract] operations can execute at the same time since they don't have any data dependencies.

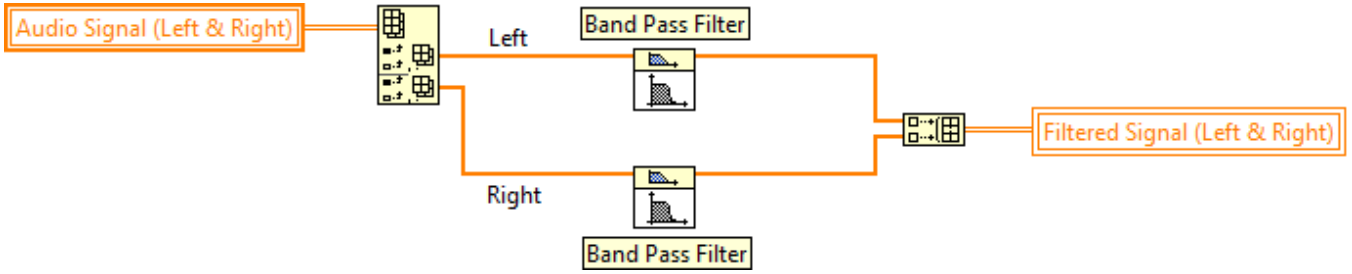
Dataflow Languages Naturally Express Parallelism

The LabVIEW compiler will automatically multithread code expressed in parallel

Task Parallelism



Data Parallelism



IPP Max Planck Institute - Germany

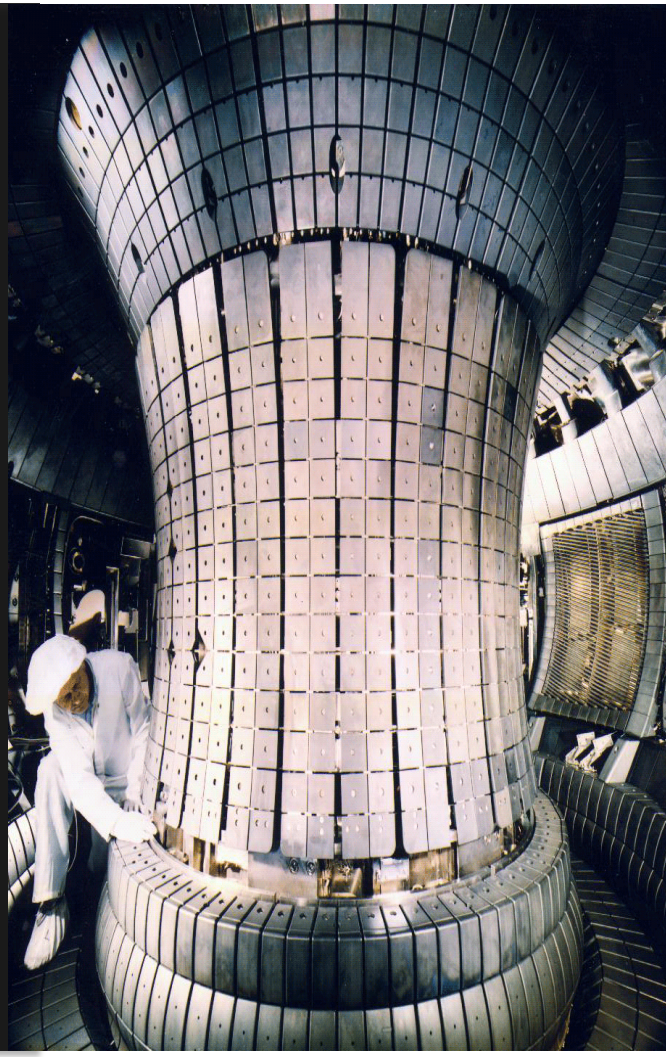
Plasma Diagnostics & Control with NI LabVIEW Real Time

Max Planck Institute

Plasma control in nuclear fusion Tokamak with NI LabVIEW on an eight-core real-time system

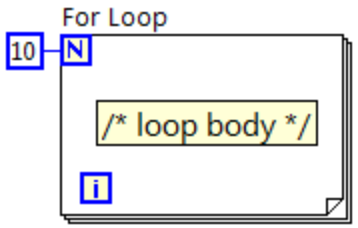
"…with LabVIEW, we obtained a 20X processing speed-up on an octal-core processor machine over a single-core processor…"

Louis Giannone
Lead Project Researcher
Max Planck Institute

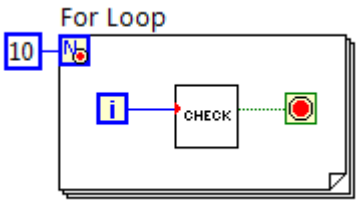


Text Loops and Their LabVIEW Equivalents

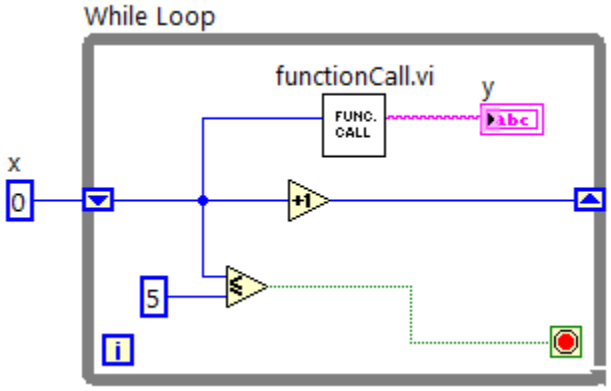
```
for (i = 0; i < 10; i++)
{
    /* loop body */
}
```




```
for (i = 0; i < 10; i++)
{
  if(check(i)) break;
}
```



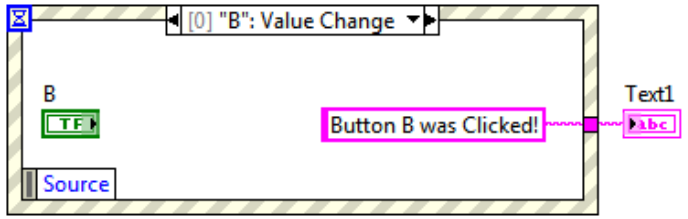

```
int x = 0;
String y;
while (x < 5)
{
    y = functionCall(x);
    printf(y);
    x++;
}
```



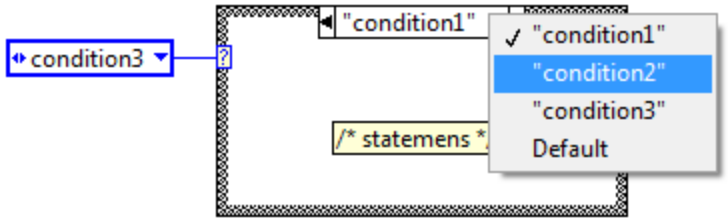
NATIONAL INSTRUMENTS

Text Events, Cases, and Their LabVIEW Equivalents

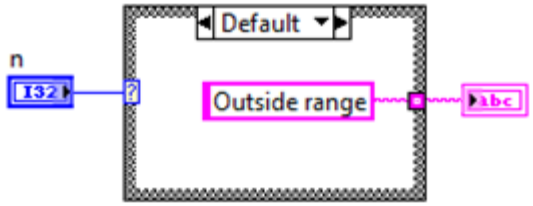
```
Button B = new Button();
B.Click += new RoutedEventHandler(OnBClick);

void OnBClick(object Source)
{
      Text1.Text = "Button B was Clicked!";
}
```




```
if condition1 then
  -- statements;
elseif condition2 then
  -- more statements
elseif condition3 then
  -- more statements;
else
  -- other statements;
end if
```

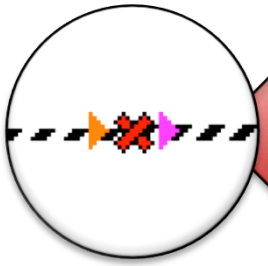



```
switch (n) {
  case 5:
    printf("Small number.");
    break;
  case 100:
    printf("Large number.");
    break;
default:
    printf("Outside range");
    break;
}
```



The Color, Style, and Thickness of Common Wires

Wire Type	Scalar	1D Array	2D Array	Color
Floating Point				Orange
Integer				Blue
Boolean				Green
String				Pink
Error				Yellow

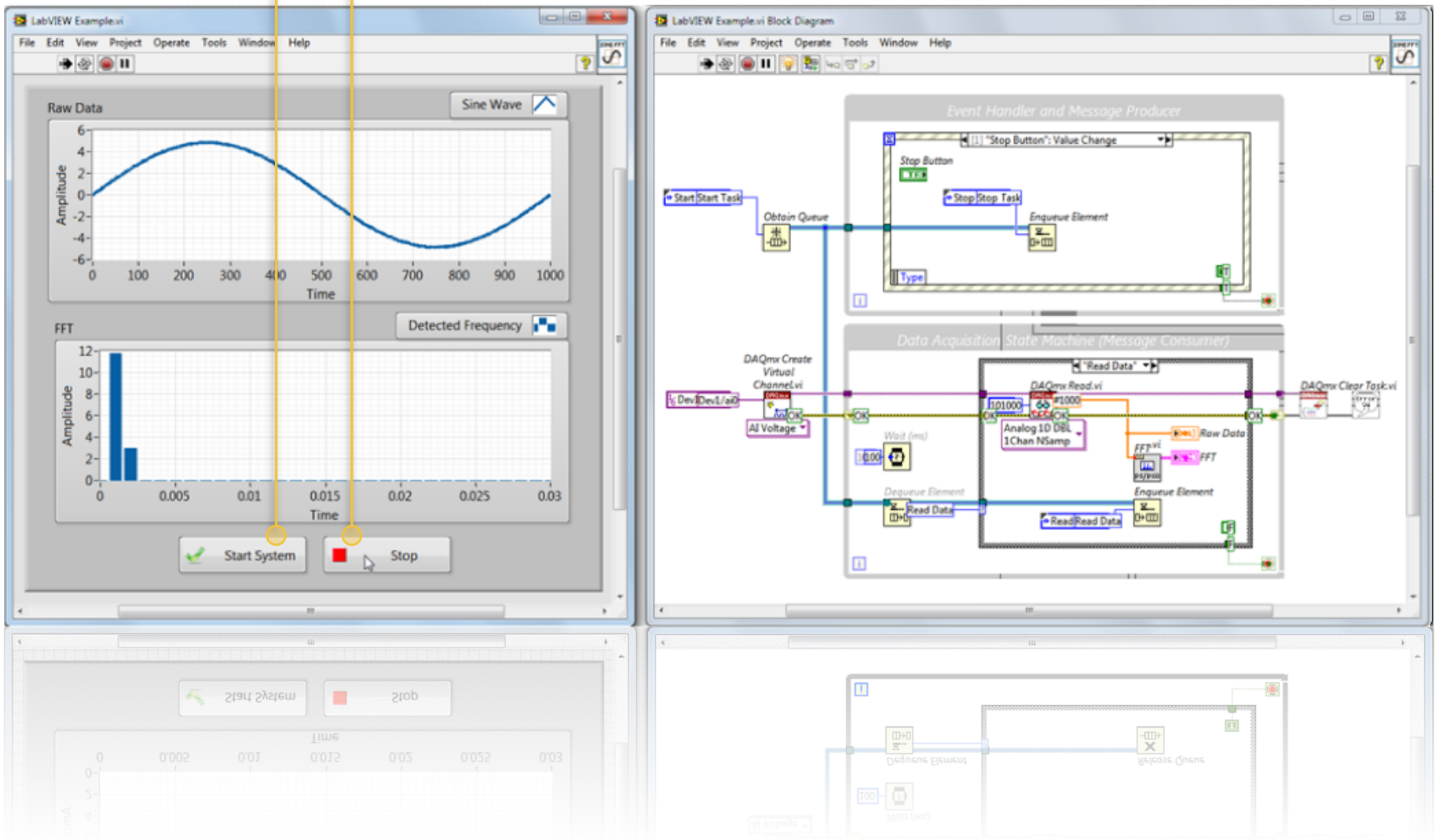


A "broken wire" represents a data type conflict that LabVIEW cannot automatically resolve. Fix it, or your code won't run!

Visualizing Data Flow Along Wires: Highlight Execution

User presses the “Start” button to fire the first event

User presses the “Stop” button to fire the second event

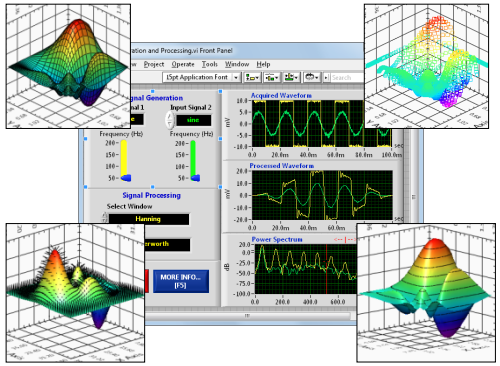


Extending LabVIEW Beyond Data Acquisition

Express VIs
Spectral Measurements
Distortion Measurements
Tone Measurements
Amplitude and Level Measurements
Timing and Transition Measurements
Curve Fitting
Filters
Statistics
Convolution and Correlation
Signal Simulation
Mask and Limit Testing
Histogram

Signal Processing and Math
Digital Filters and Banks
Frequency and Time Domain
JTFA
Sampling, Re-Sampling
Transforms
Wavelets and Windowing
Signal and Waveform Generation
Pulse and Pattern Generation
Basic Numeric Functions
Curve Fitting and Data Modeling
Differential Equations and Linear Algebra
Interpolation and Optimization
Nonlinear Systems
Root Finding
Statistics and Random Processes

Measurements
Amplitude
Distortion
Frequency Domain (Spectral)
Noise
Phase Noise/Jitter Analysis
Power (AC) Measurements
Pulse and Transition
Sampling (Discrete Time) Concerns
Time Domain Analysis
Tone Detection
Transient Signal Analysis



Advanced Analysis

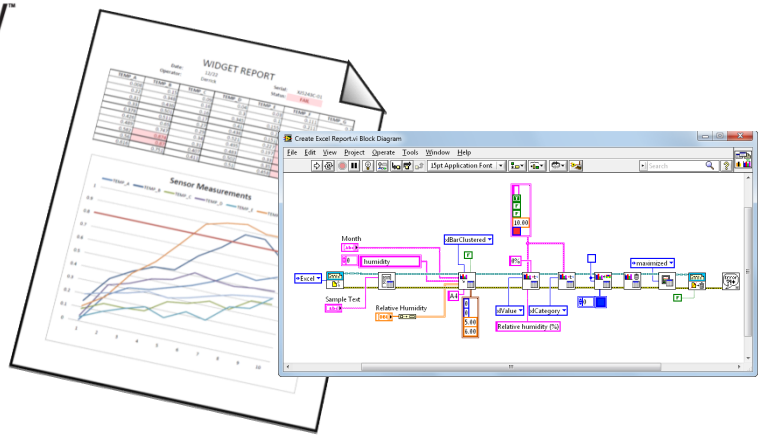
External Code Integration

NATIONAL INSTRUMENTS LabVIEW

Complex Visualization

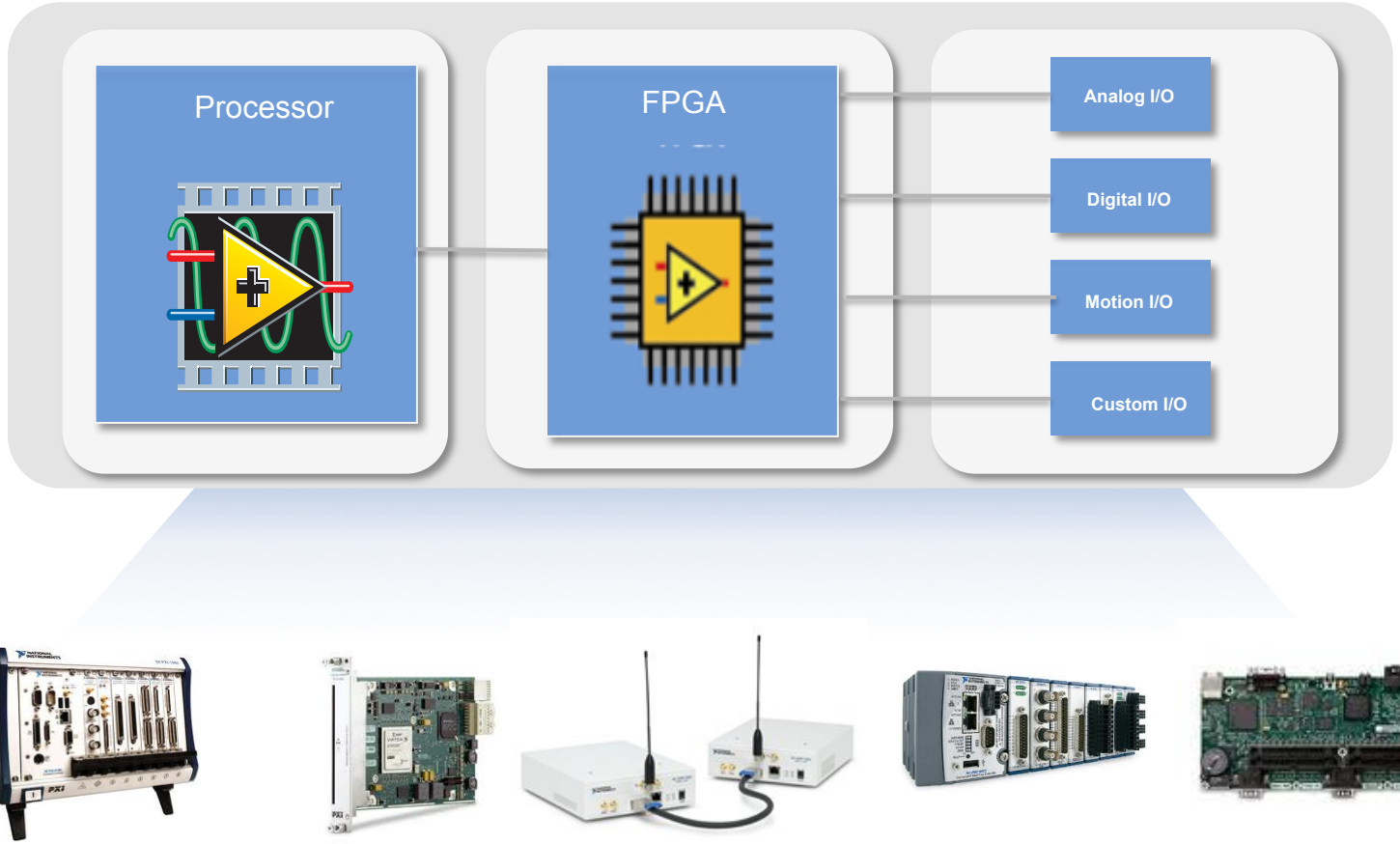
Automated Reporting

.NET Assembly, C DLL, .m File

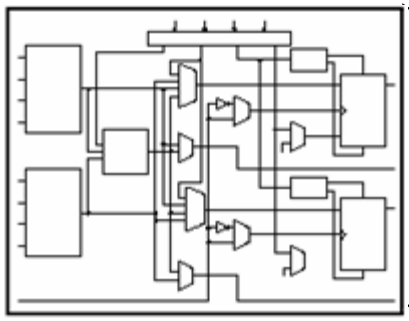


NATIONAL INSTRUMENTS

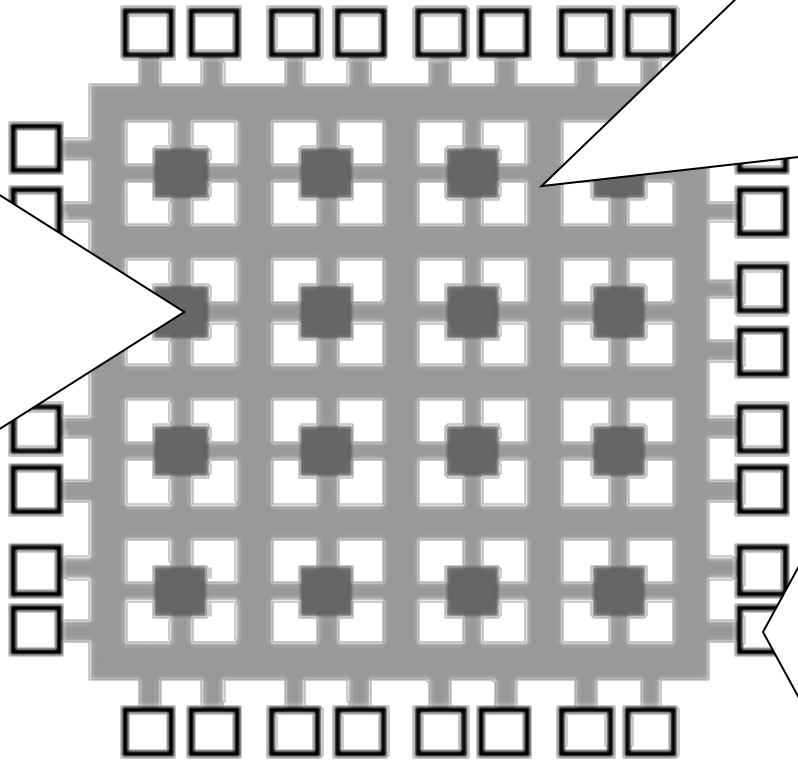
Combining COTS With Your Design: LabVIEW Reconfigurable I/O (RIO) Architecture

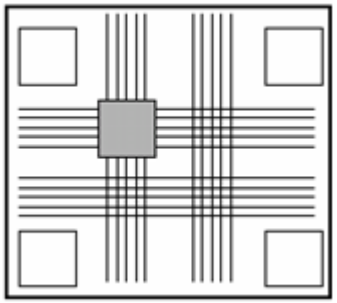


FPGA Technology

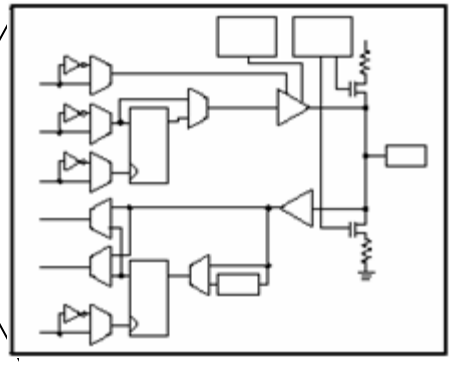


Logic Blocks





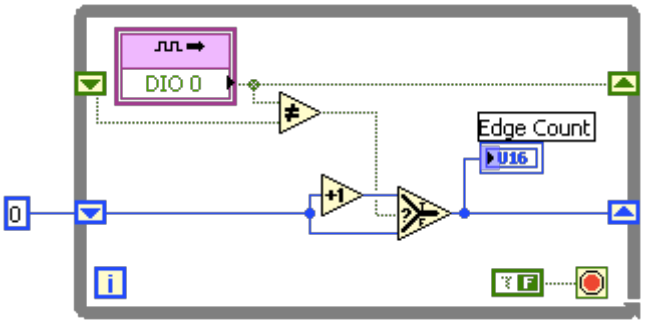
Programmable Interconnects



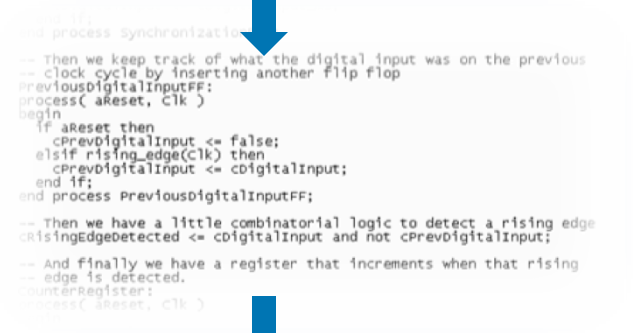
I/O Blocks

How Does LabVIEW FPGA Work?

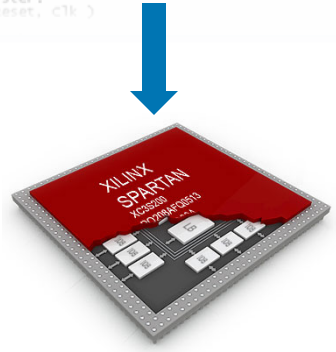
1. Same graphical programming



2. Generate VHDL

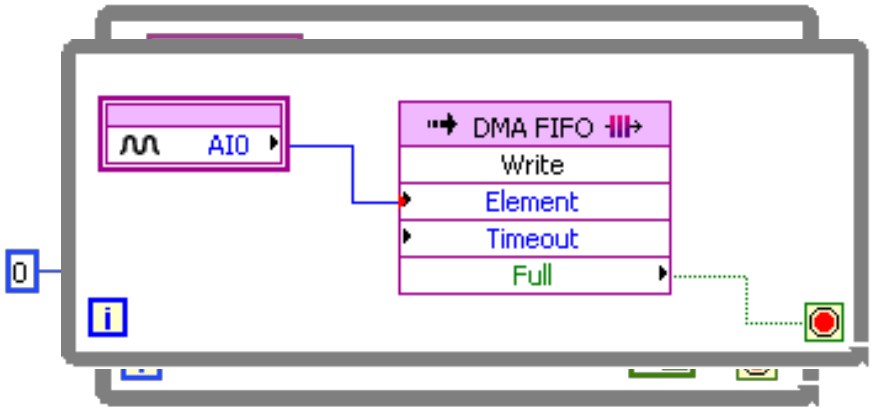


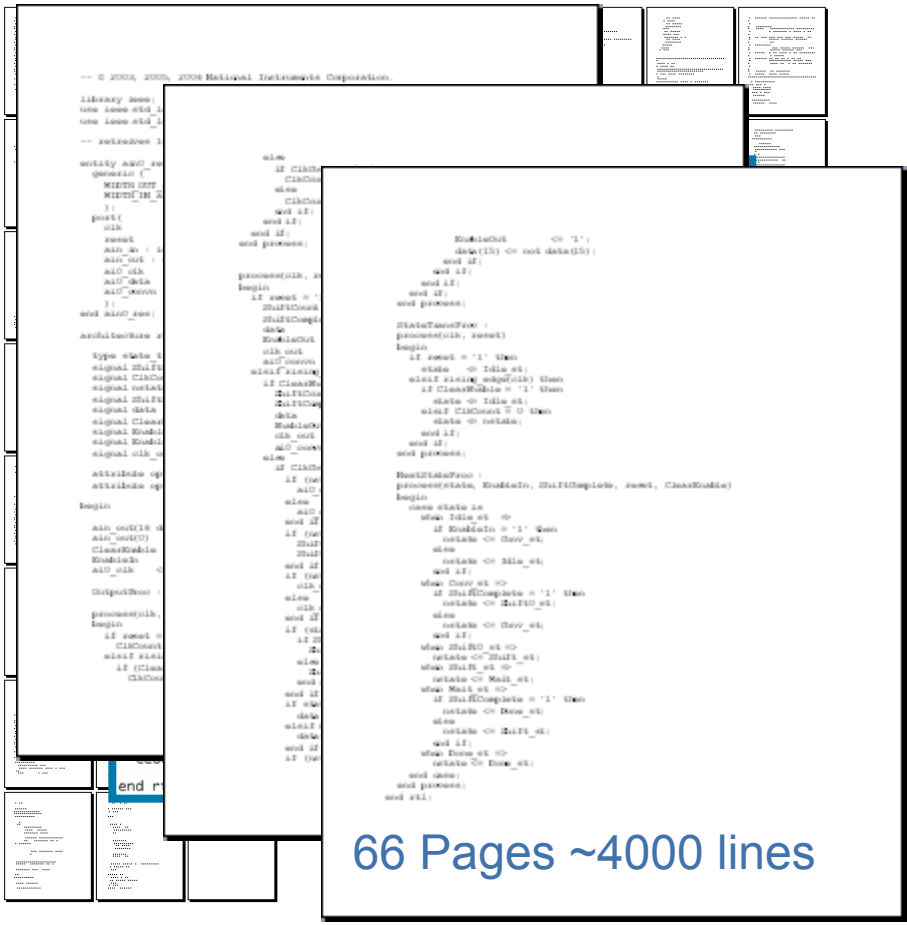
3. Compile VHDL through Xilinx



4. Generate downloadable bit file

LabVIEW FPGA vs. VHDL



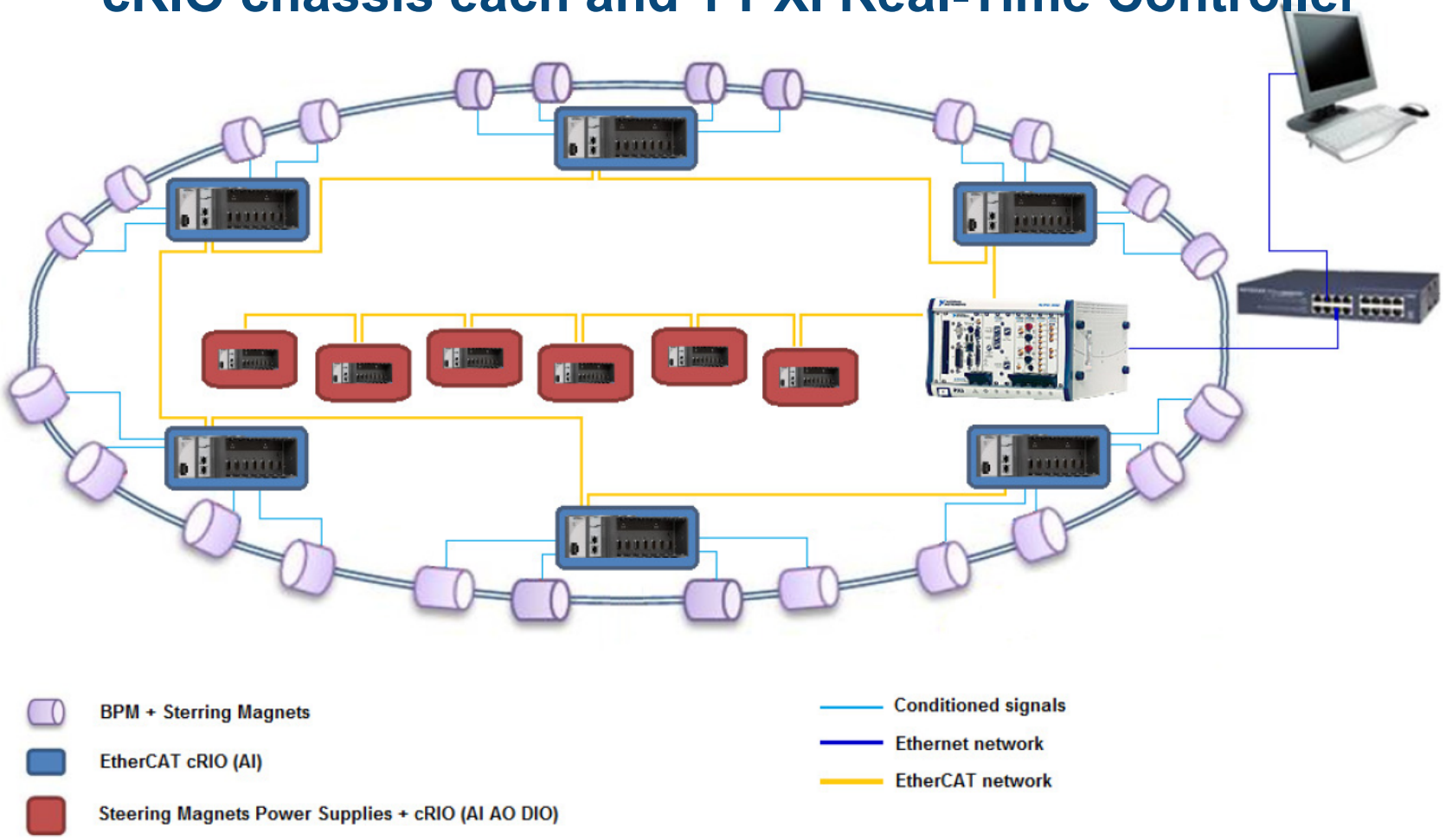


Counter **Analog I/O** **I/O with DMA**

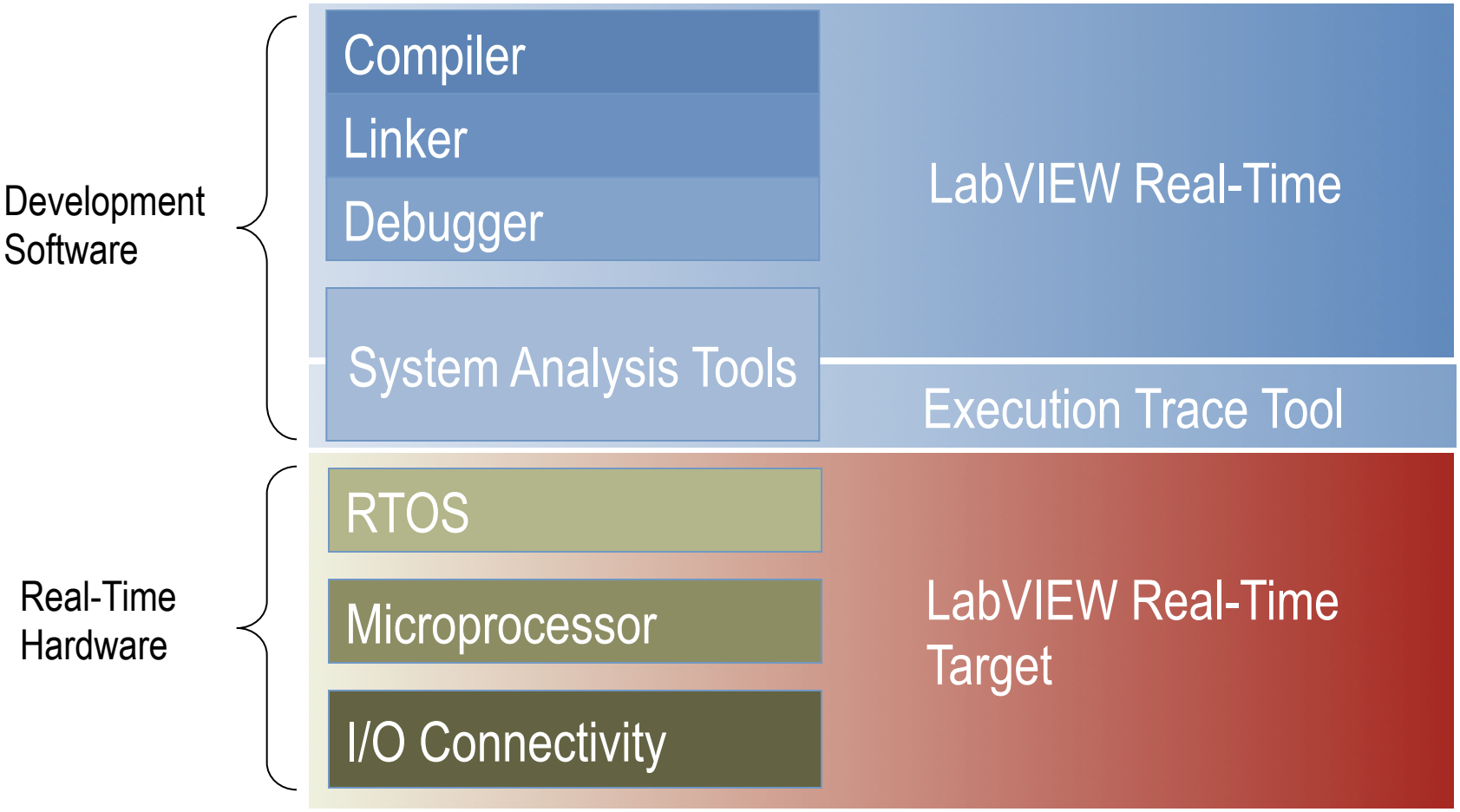
NATIONAL INSTRUMENTS

LNLS – Brazilian Synchrotron
Fast Orbit Feedback Control System

New orbit control system topology: 2 EtherCAT loops with 6 cRIO chassis each and 1 PXI Real-Time Controller



Real-Time Development Tools



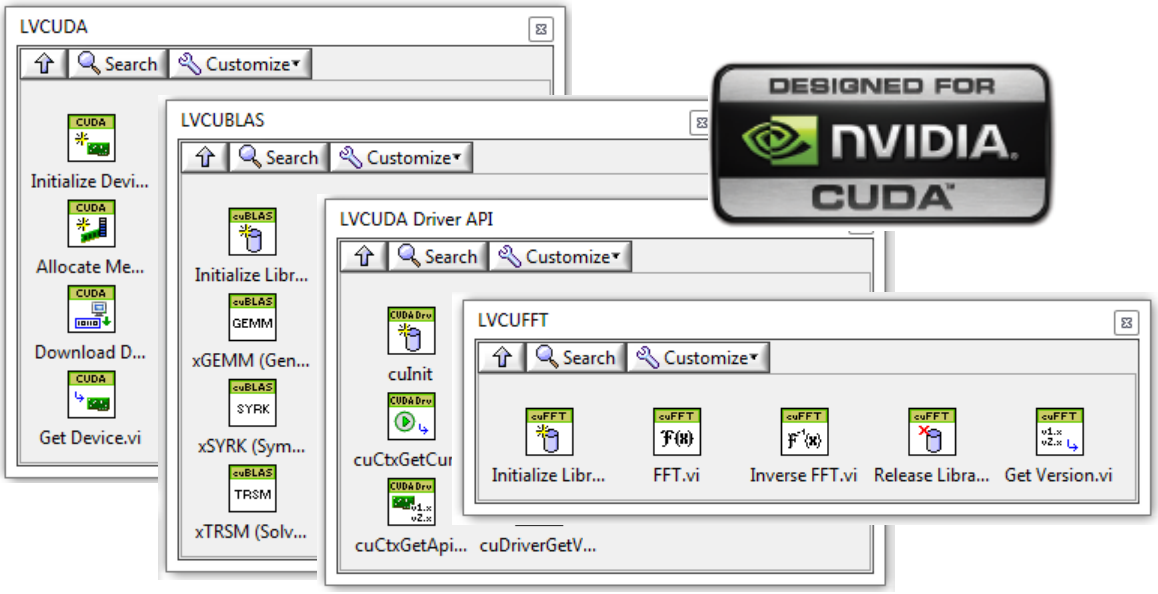
ISIS Synchrotron, Rutherford Appleton Labs

Beam data acquisition and analysis Hardware based on PXI platform

Beam loss monitoring
High speed digitizers
Beam position monitoring
Timing and synchronization
Multichannel profile monitoring

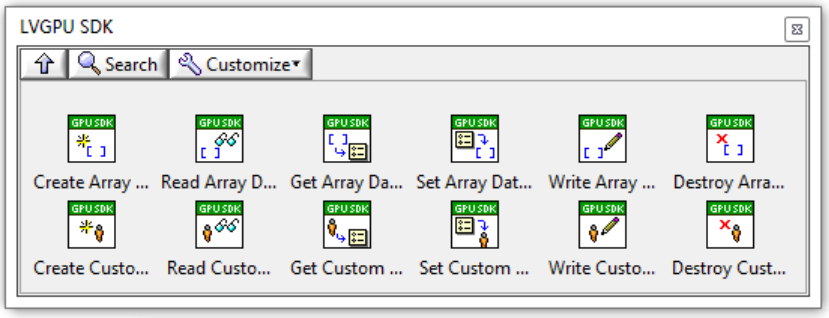
Toolkits for Real-Time Computation

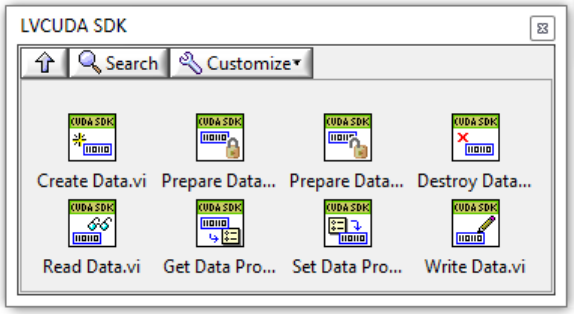
- GPU Analysis Toolkit
  - Set of CUDA™ Function Interfaces



Toolkits for Real-Time Computation

- GPU Analysis Toolkit
  - Set of CUDA™ Function Interfaces
  - SDK for Custom Functions

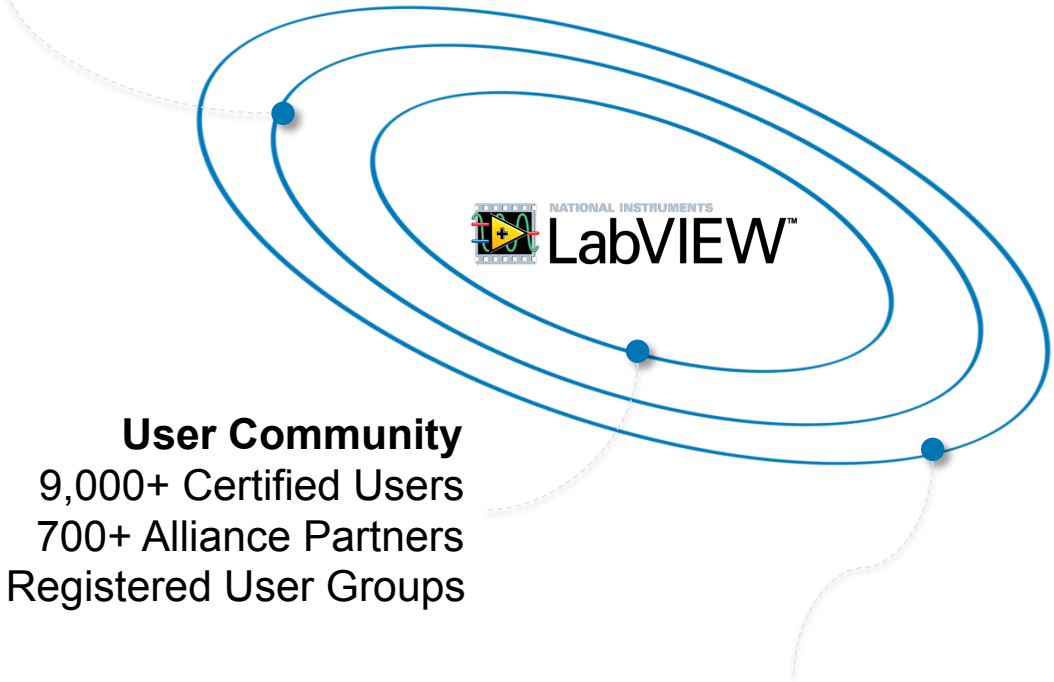




DirectX/DirectCompute

Leveraging the LabVIEW Ecosystem

LabVIEW Tools Network
1,000,000+ Add-Ons Downloaded
26+ Certified Add-Ons
100+ Available Add-Ons



User Community
9,000+ Certified Users
700+ Alliance Partners
60+ Registered User Groups

Modules and Toolkits
40+ Toolkits and Modules Including:

LabVIEW Real-Time Module
LabVIEW FPGA Module
LabVIEW Embedded Module for ARM
LabVIEW Touch Panel Module
LabVIEW Wireless Sensor Network Module
LabVIEW C Code Generator
NI Real-Time Hypervisor
Vision Development Module for LabVIEW
Sound and Vibration Measurement Suite
Sound and Vibration Toolkit
LabVIEW Advanced Signal Processing Toolkit
LabVIEW Adaptive Filter Toolkit
LabVIEW Digital Filter Design Toolkit
LabVIEW MathScript RT Module
Spectral Measurements Toolkit
Modulation Toolkit for LabVIEW
LabVIEW Robotics Module
LabVIEW Biomedical Toolkit
ECU Measurement and Calibration Toolkit
GPS Simulation Toolkit for LabVIEW
Measurement Suite for Fixed WiMAX
WLAN Measurement Suite
Automotive Diagnostic Command Set
LabVIEW GPU Analysis Toolkit
Multicore Analysis and Sparse Matrix Toolkit
LabVIEW PID and Fuzzy Logic Toolkit
LabVIEW Control Design and Simulation Module
LabVIEW System Identification Toolkit
LabVIEW Simulation Interface Toolkit
LabVIEW SoftMotion Module
LabVIEW Datalogging and Supervisory Control Module
LabVIEW Report Generation Toolkit for Microsoft Office
LabVIEW Database Connectivity Toolkit
LabVIEW DataFinder Toolkit
LabVIEW SignalExpress
LabVIEW VI Analyzer Toolkit
LabVIEW Statechart Module
LabVIEW Desktop Execution Trace Toolkit
NI Requirements Gateway
NI Real-Time Execution Trace Toolkit
LabVIEW Unit Test Framework Toolkit
LabVIEW Application Builder for Windows