COTS TECHNOLOGY FOR HIGH ENERGY PHYSICS INSTRUMENTATION

J. Truchard, R. Marawar, A. Veeramani, M. Ravindran
National Instruments, Austin, TX 78759, U.S.A

Abstract

National Instruments (NI) uses Commercial Off-The-Shelf (COTS) semiconductor and computing technology and applies it to measurement, diagnostics and instrumentation needs. NI leverages the rapid technological advancement of the semiconductor and computer industry, while retaining the flexibility and ensuring interoperability between HW & SW. This paper elaborates how general purpose NI products and technology can be applied to specialized instrumentation, measurement and diagnostic needs in High Energy Physics (HEP). In particular we will discuss programming and processing tools, instrumentation platform with custom hardware development using FPGA.

INTRODUCTION

Until recently, most measurement, diagnostic and instrumentation applications in HEP required custom development efforts. Most COTS products did not meet the needs of the community due to high-end specifications, high availability requirements, long term serviceability aspects, closed driver stacks, limited operating system support etc. In recent years NI has led efforts to address these issues by developing state-of-the-art products that meet demanding specifications, making great progress towards reliable and redundant systems, providing long-term replacement and calibration services, support for Linux, providing register maps for open driver development and collaborating on EPICS, Tango and other middleware support. In addition, using COTS products minimizes development costs, reduces unit costs thus focusing limited resources on scientific endeavors.

Reducing Development Time

Many of the HEP applications require development of complex measurement, control and diagnostic systems. In addition, data must be processed real-time to make intelligent decisions about measurement and control parameters. For mathematical computations and simulations it would be desirable to take advantage of COTS technologies such as multicore processors and FPGAs. National Instruments LabVIEW offers scientists and engineers a unique graphical programming environment that combines control and measurement system development along with real-time high performance computing including multicore processors and FPGAs.

LABVIEW GRAPHICAL SYSTEM DESIGN

LabVIEW is a general purpose graphical programming environment used to develop measurement, test, and control systems using intuitive graphical icons and wires that resemble a flowchart. LabVIEW is supported on a variety of platforms including Microsoft Windows (2K, XP, Vista), Linux, and Macintosh. LabVIEW Real-Time, the embedded systems solution for LabVIEW, is supported on VxWorks and Pharlap targets on hardware platforms such as CompactRIO and PXI. LabVIEW offers tight integration with thousands of hardware devices and provides hundreds of built-in libraries for advanced analysis and data visualization. LabVIEW also enables parallel programming targeting Multicore CPUs and FPGAs.

As a COTS product, LabVIEW provides an extensively tested and industry validated product with multiple models of computation, leading data flow architecture and significant R&D investment that is amortized over hundreds of thousands of installations.

LabVIEW on Multicore CPU

To reduce computation time, programmers must divide applications into parallel parts using multithreading. Each of the threads can run in parallel in a multicore system. Applications that take advantage of multithreading have numerous benefits, such as more efficient CPU use, better system reliability, and improved performance. Three techniques to commonly used for reducing computation time include: task parallelism, data parallelism, and pipelining. LabVIEW is an inherently parallel programming language without having to worry about thread management or synchronization between threads. Consider two tasks shown in Fig.1. The two tasks do not have any data dependencies, so the LabVIEW compiler will automatically multi-thread the application.

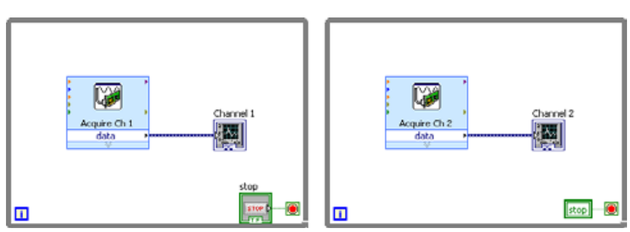

Figure 1: LabVIEW block diagram showing two tasks that will be automatically be executed in parallel.

Data parallelism can be applied to large data sets by splitting up a large array or matrix into subsets, performing the operation, and combining the result. Consider the sequential implementation, where a single core of a given CPU attempts to process the entire data set. Instead if the data set is split four parts, all the cores available will execute the data sets in parallel resulting in significant speed increase.

In real-time high-performance computing (HPC) applications such as control systems, a common and efficient strategy is the parallel execution of matrix-vector multiplications of considerable size. Typically, the matrix is fixed, and the matrix can be decomposed in advance. Measurements gathered by sensors provide the vector on a per-loop basis. For example, the result of the matrix-vector can be used to control actuators. Fig 2 shows block diagram of a matrix-vector multiplication distributed to eight cores.

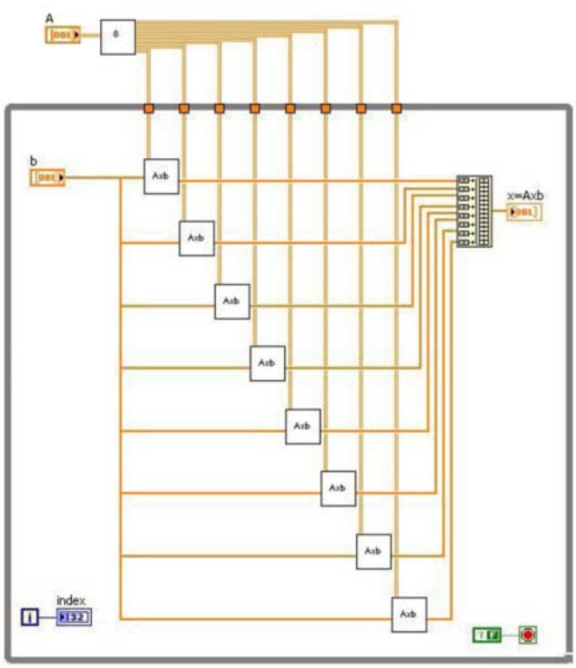

Figure 2: Matrix vector multiplication LabVIEW.

Programming FPGAs with LabVIEW

The NI LabVIEW FPGA Module uses LabVIEW embedded technology to extend LabVIEW graphical development and target field-programmable gate arrays (FPGAs) on NI reconfigurable I/O (RIO) hardware. LabVIEW is distinctly suited for FPGA programming because it clearly represents parallelism and data flow. LabVIEW programming palette is simplified to contain only the functions that are designed to work on FPGAs. The primary programming difference compared to traditional LabVIEW is that FPGA devices use integer math rather than floating-point math. Also, there is no notion of multithreading or priorities because each loop executes in independent dedicated hardware and does not have share resources — in effect, each loop executes in parallel at "time critical" priority.

With the LabVIEW FPGA, you can create custom measurement and control hardware without low-level hardware description languages or board-level design. You can use this custom hardware for unique timing and triggering routines, ultrahigh-speed control, interfacing to digital protocols, digital signal processing (DSP), RF and communications, and many other applications requiring high-speed hardware reliability and tight determinism. Thus this COTS product with hundreds of thousands of man hour R&D investment is available for HEP community to design highly customized measurement and control applications.

INTERFACING EPICS AND LABVIEW

In order to take advantage of commercial off the shelf (COTS) hardware that is available through LabVIEW, the physics community has come up with different ways to bridge LabVIEW and EPICS such as the Shared Memory Interface and Channel Access (CA) Server from Oak Ridge National Lab. National Instruments has also developed EPICS client and CA Server in addition to the EPICS IOC to LabVIEW Real-Time interface that was developed in partnership with Cosylab. Currently the main method at SNS of connecting LabVIEW with EPICS is the EPICS Shared Memory Interface [1].

LabVIEW 2009 introduced an EPICS Server which is really a configuration based CA Server. Similar to the CA Server implemented by SNS, the CA Server in LabVIEW allows LabVIEW enabled hardware to appear as an EPICS node. The processing is all done using the function blocks in LabVIEW. Using the configuration based CA Server, you choose which shared variables in LabVIEW need to be published as PVs over the CA network.

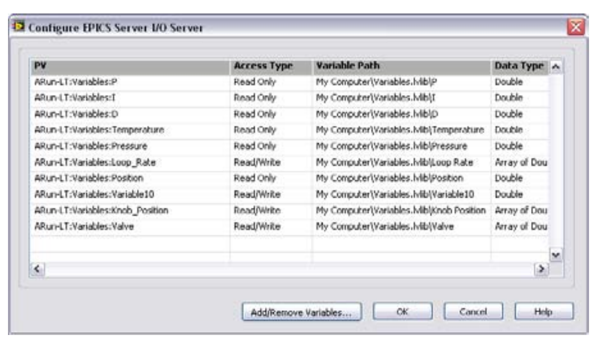

Figure3: Configuration based CA Server in LabVIEW.

The CA Server publishes value (VAL), description (DESC), timestamp (TIME) and the number of elements (NELM) in an array data type. This implementation of the configuration based CA Server can run on Windows and also in VxWorks and Pharlap RTOS. With the ability to run on Pharlap RTOS, PXI based instruments can now be integrated into an EPICS based control and data acquisition system.

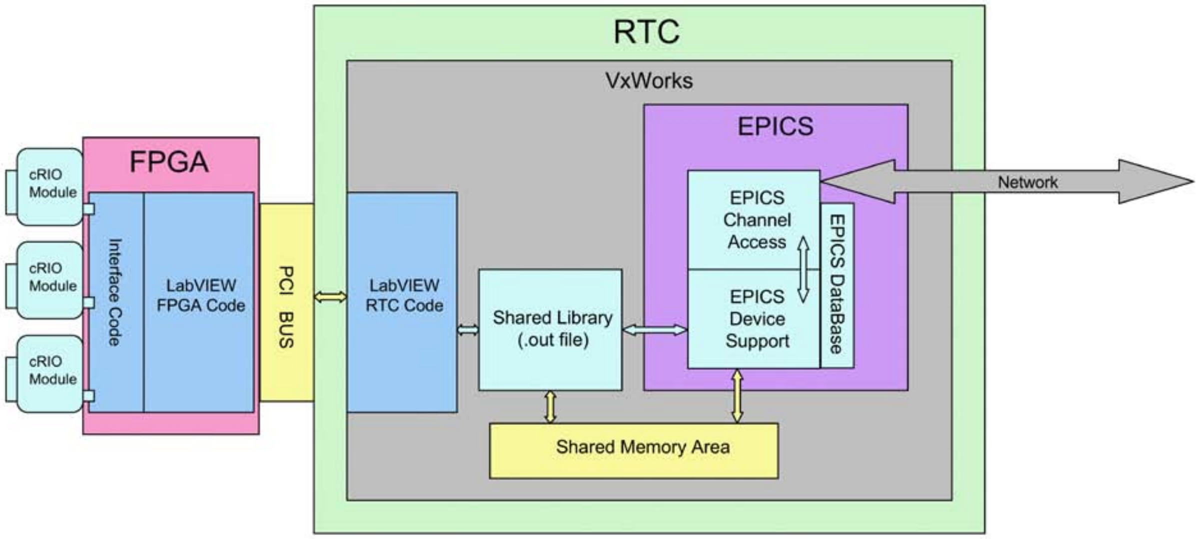

Figure 4: System architecture for running EPICS on the NI CompactRIO platform.

Another option is embed the EPICS IOC in CompactRIO, a "Programmable Automation Controller" (PAC) manufactured by National Instruments that is covered in the later section of this paper. Los Alamos National Laboratory has been working with NI to implement EPICS IOC on CompactRIO to be used in their project of upgrading the control system at the Los Alamos Neutron Science Centre (LANSCE) [2]

LEVERAGING COTS INSTRUMENTATION

PXI, or PCI eXtensions for Instrumentation, leverages COTS components while adding the necessary features such as ease of use, timing and synchronization, and rugged industrial components for enhanced reliability. In addition, PXI is an open platform enabling many industries to create custom modules for their needs.

We recommend PXI as the preferred COTS platform for the CODAC HPN due to PXI's superior offering in the following areas:

- Open, international standard
- High bandwidth PCIe/PCI COTS architecture
- Variety of I/O available, wide industry adoption
- High performance timing and synchronization
- Time network compatibility
- FPGA Custom modules on PXI
- Compatible with industry standard and custom protocols
- Scalable for large instrumentation
- Real time, Linux, Windows OS support

Because PXI is an open industry standard, today nearly 1,500 products are available from the 70+ members of the PXI Systems Alliance. Available device categories include: analog input and output, high speed digitizers,

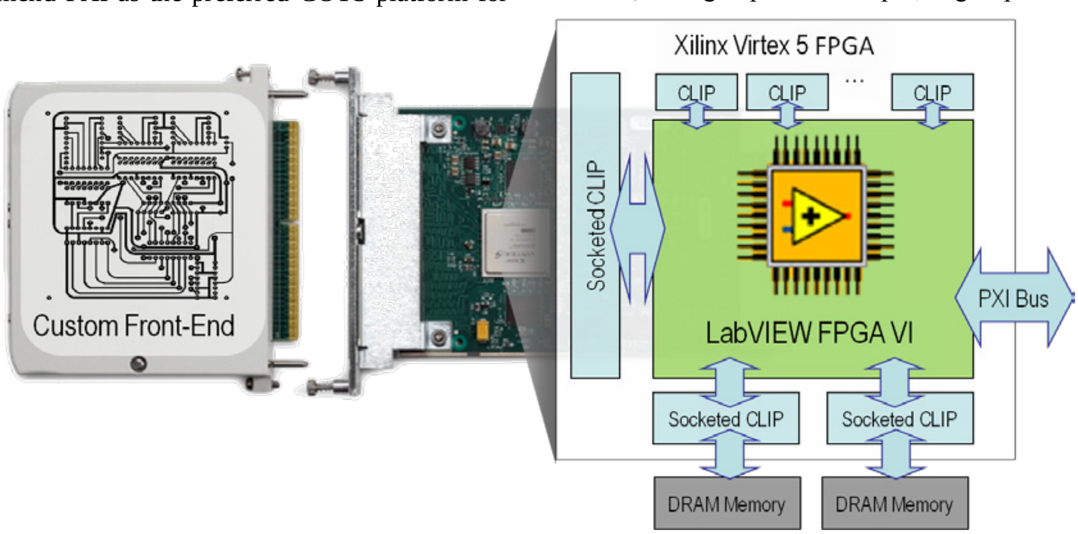

Figure 5: NI FlexRIO with custom front-end and LabVIEW FPGA programmable target.

digital input and output, timing input and output, reconfigurable I/O, power supplies and motion control, to name a few.

NI FlexRIO

The NI FlexRIO product family provides flexible, customizable I/O for NI LabVIEW FPGA. Solutions consist of NI FlexRIO FPGA Modules for PXI and PXI Express, and NI FlexRIO Adapter Modules, which add I/O to the FPGA. Together, they form a high-performance, reconfigurable instrument powered by LabVIEW FPGA. Adapter modules are available from NI and National Instruments Alliance Partners, or you can use the NI FlexRIO Module Development Kit (MDK) to build your own.

NI FlexRIO Adapter Module Development Kit You can build the exact adapter module your application needs with the NI FlexRIO MDK. This requires experience in custom printed circuit board (PCB) design and hardware description language (HDL) code development. The NI FlexRIO Adapter MDK includes electrical and mechanical specifications, example code, and design documentation.

CONTROL, DIAGNOSTICS AND AUTOMATION

National Instruments is uniquely positioned to offer COTS products that span instrumentation to measurements, diagnostics and supervisory control. National Instruments CompactRIO is a small rugged industrial control and acquisition system powered by reconfigurable I/O (RIO) FPGA technology for ultrahigh performance and customization. NI CompactRIO incorporates a real-time processor and reconfigurable FPGA for reliable stand-alone embedded or distributed applications, and hot-swappable industrial I/O modules with built-in signal conditioning for direct connection to sensors and actuators. CompactRIO represents a low-cost architecture with open access to low-level hardware resources.

Figure 6: NI CompactRIO, an FPGA hardware platform.

Each I/O module contains built-in signal conditioning and screw terminal, BNC, or D-Sub connectors. A variety of I/O types is available, including ±80 mV thermocouple inputs, ±10 V simultaneous-sampling analog inputs and outputs, 24 V industrial digital I/O with up to 1 A current drive, differential/TTL digital inputs with 5 V regulated supply output for encoders, and 250 Vrms universal digital inputs. In addition a large number of Third Party vendors develop I/O modules with wide ranging functionality and specifications. For even more flexibility, a custom module development kit is available. NI cRIO-9951 CompactRIO Module Development Kit, users can develop custom modules to meet the unique needs of particular products and applications. The kit includes CompactRIO module development software and the CompactRIO Module Development Kit user manual. Additionally, it features housing module shells and connectors from the following packages: NI cRIO-9952, cRIO-9954, and cRIO-9955

REFERENCES

[1] A.Veeramani, T. Debelle, W. Blokland, R. Dickson, A. Zhukov, "Options for Interfacing EPICS to COTS Hardware Through LabVIEW", THD004, ICALEPCS 2009.

[2] E. Björklund, A. Veeramani, T. Debelle "Using EPICS Enabled Industrial Hardware for Upgrading Control Systems", WEP078, ICALEPCS 2009.