Tailoring The Hardware To Your Control System

Eric Björklund
Scott Baily
Los Alamos Neutron Science Center
Part I

Our Journey Into Adaptation Hell
How We Got Here:

1960’s

Control System

RICE
How We Got Here:

1960’s

Control System

RICE

(Remote Instrumentation & Control Equipment)
How We Got Here:

1970’s – 1980’s

Control System

RICE

CAMAC
How We Got Here:

1990’s – 2000’s

- Legacy Control System
- EPICS
- CAMAC
- VME

Legacy to EPICS
EPICS to Legacy
Part II

Tailoring The Controls Hardware To The Accelerator Equipment
Types Of Programmable Controllers

PLC
(Programmable Logic Controller)
- Long-time standard for Industrial I/O applications.
- I/O connects to proprietary bus.
- Typically programmed in Ladder Logic.
- Rugged & reliable, not particularly fast. (mSec response)
- Some are safety certified.

PAC
(Programmable Automation Controller)
- Recently becoming available for Industrial I/O applications.
- I/O connects to FPGA.
- Typically programmed in VHDL/AHDL.
- Becoming rugged & reliable. Typically faster than PLCs. (μSec response)
- Unaware of any safety certified commercial PAC products.
- FPGA runs independently of processor.
RICE Hardware Features

- **Binary Output:**
  - Four Protocols
    - Command Only, Latchback, Momentary Open, Momentary Closed
  - Protocol Selected by Jumper on Card

- **Analog Input**
  - Three ADC Ranges
    - 10 Volt, 1 Volt, 100 mVolt
  - ADC’s triggered to avoid RF-induced noise.

- **Analog Output**
  - Mostly stepper motors
  - Primary operator interface is assignable control knobs
    - Custom software interface required to make EPICS work well with control knobs.
New LANSCE Industrial I/O System

- Initially replaced the Industrial I/O functions of one RICE module with a PLC.
  - Not quite fast enough for some of our applications.
  - 3rd party stepper motor controller was a little unreliable.

- Second iteration replaced PLC with National Instruments Compact RIO.
  - I/O Interfaces to FPGA.
  - FPGA interfaces to processor over PCI bus.
  - Programmed a binary output module to be a stepper motor controller.
    - Exact pulse width, speed, and ramp up we wanted.
    - Command overrides programmed in.
New LANSCE Industrial I/O System

- Binary Input (64 Channels)
- Sinking Binary Output (32 Channels)
- Stepper Motors (4 Channels)
- Processor
- Differential Analog Input (32 Channels)
- Relay Binary Output (8 Channels)
- Analog Output (16 Channels)
Part III

Reconfiguring The Hardware
Reconfiguration With RICE

RICE Binary Output Channels

- Binary Output Protocol Changed With Jumpers
- Only Disrupts The Channel You Are Modifying
Reconfiguration With PLC/PAC

**Ladder Logic (PLC)**
- Modify Ladder Logic
- Take Controller Off-Line
- Load New Ladder Logic
- Take Controller On-Line

**FPGA (PAC)**
- Modify FPGA Code
- Compile FPGA Code Into Bitmap (lengthy)
- Take Controller Off-Line
- Flash New Bitmap
- Take Controller On-Line
Reconfiguration With Stem Cells

Binary Output Channels
(Stem Cells)

Readback Channels
Reconfiguration With Stem Cells

Configure Binary Output Channels

Configure

Readback Channels
Reconfiguration With Stem Cells

Binary Output Channels
(Mature)

Latchback
Hold Time = 10

Command

Momentary
Hold Time = 70

Latchback
Hold Time = 100

Readback Channels
Reconfiguration With Stem Cells

Configure

Binary Output Channels

Latchback
Hold Time = 10

Latchback
Hold Time = 70

Latchback
Hold Time = 100

Readback Channels

Configure

Binary Output Channels

Latchback
Hold Time = 10

Latchback
Hold Time = 70

Latchback
Hold Time = 100

Readback Channels
Reconfiguration With Stem Cells

Binary Output Channels

- Latchback
  - Hold Time = 10
- Latchback
  - Hold Time = 5
- Momentary
  - Hold Time = 70
- Latchback
  - Hold Time = 100

Readback Channels
Other Stem Cell Types

- Analog Inputs
  - ADC Range
  - Trigger/No-Trigger
  - Trigger Delay

- Stepper Motors
  - Pulse Rate
  - Ramp Speed

- Counters
  - Integration Time
Reconfiguration With Stem Cells

**Advantages**

- Reconfiguration time as fast or faster than hardware.
- Reconfiguration does not interrupt service.
- Can use the same bitmap for all controllers.

**Disadvantages**

- Uses More FPGA Real-Estate
  - Virtex 2 only had room for 11 binary output stem cells
  - Virtex 5 had room for more than 40 binary output stem cells
Part IV

Tailoring The Hardware To The Software
Tailoring The Hardware To The Software

- Perhaps the ultimate way to tailor the hardware to your control system is to actually embed the control system within the hardware.

- Many commercial products have embedded processors with network access and a real-time (or soft real-time) operating system.

- Some can allowing your control system (or at least the front-end) to run on the processor and interact with the vendor’s code.
  - Vendor may do the embedding.
  - Vendor may supply interface for embedding.

- Power of Collaborations
Tailoring The Hardware To The Software

Some Vendors That Have Already Supported Embedding

- Instrumentation Technologies
- Moxa
- National Instruments
- Yokogawa
- ZTEC Instruments
Embedded Control Systems

Controller With Embedded Control System

- One Network Trip
Embedded Control Systems

Controller With Embedded Control System

- One Network Trip
- Access To Standard CS Utilities
  - Access Control
  - Archiving
  - Performance Monitors
  - Diagnostics
Embedded Control Systems

Controller With Embedded Control System

- One Network Trip
- Access To Standard CS Utilities
  - Access Control
  - Archiving
  - Performance Monitors
  - Diagnostics
  - Our Custom Knob Software
Part V

Conclusions
Conclusions:

- Programmable controllers can simplify interfacing to the equipment.
- Embedding the control system in the controller can simplify interfacing to the software.
- “Stem Cells” can simplify hardware reconfiguration.
- Our standard Compact RIO IIO system is working well as a replacement for the industrial I/O functions of our old RICE system.
- With a few tweaks, it has also worked well as a replacement for our old CAMAC equipment.
- To date we have replaced two RICE modules and two CAMAC crates with IIO systems.
Where We Were:

- Legacy Control System
- RICE
- CAMAC
- EPICS
- VME

Legacy to EPICS

EPICS to Legacy
Where We Are Today:

- Legacy Control System
- RICE
- CAMAC
- EPICS
- VME
- PAC

Legacy to EPICS
EPICS to Legacy
Where We Are Headed:

- EPICS
- VME/cPCI
- PAC/PLC