The LANSCE FPGA Embedded Signal Processing Framework

Jeff Hill

LANSCE

Scope

- Replacing legacy RICE electronics
  - Mitigating risk
    - Component obsolescence, maintenance burden, Micro VAX II backplane, programmer portability …
  - Obsolesce
    - Old designs multiplexing one type of ADC for all signal types
- Signals requiring
  - Synchronization with the timing system
  - Waveform capture
  - Signal processing
- Systems
  - Current monitors, fast radiation-based loss monitors, beam position monitors
  - Embedded modular CPU hardware design, RTEMS RTOS, and EPICS software
    - Also utilized in RF control boards

Requirements – Timing

- Synchronization
  - With the hardware timing system
  - With data structures determining modal behaviors of LANSCE
    - This could be accomplished in hardware or without RTOS
      - But a software based solution based on a RTOS is the more flexible, cost-effective approach

Requirements – Machine Protection

- Machine protection legacy analog systems
  - Are involved in *essential* functions, are highly redundant
  - Some of the components suffer from obsolescence
- We will not preclude creation of advanced digital-based machine protection prototypes
  - Modern electronics systems offer potential for increased integration
  - Initially redundant with our analog-based systems

Requirements – Subscription Updates

- Sufficient control system bandwidth
  - 10 surface plots of 625 by 48 points
    - For future operations with longer beam pulses
      - Accommodate 1,600 by 48 points
  - Updating at a rate of 4 Hz or better
  - On several workstations at once

Preferences – Overarching

- Framework of modular components
  - Shared spares
  - Shared chassis space
  - Shared developer effort
  - Shared developer expertise
- Industry standard modular interfaces
  - Long lifespan platforms
  - Multiple vendors for purchased components
  - Prefer industry standard development languages
- In a long lifespan facility
  - All electronics will eventually become obsolete
    - With a frame work, based on industry standard interfaces
      - There is an incremental upgrade path

Hardware Design – Electronics Trends

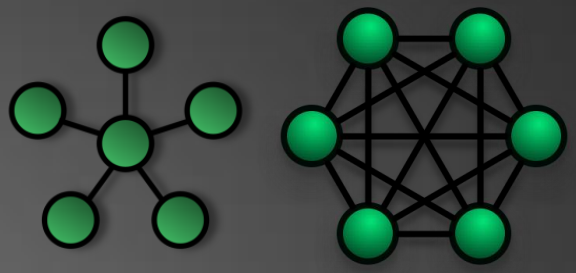
- Increased integration
  - Density
    - Is 3u the new 6u?
  - Mezzanine boards
    - Flexibility
    - Reduced cost due to economy-of-scale
  - Ethernet and processor ubiquitous
- Digital-based signal-processing algorithms
  - Design malleability
  - Component based design, collaborative or COTS
  - Instantiation of hardware from high level languages

Hardware Design – FMC Industry Standard

- FMC – FPGA interfaced mezzanine cards
  - An industry standard, VITA-57
  - A standard tailor-made for a signal processing framework!
  - Economies of scale
    - Vendor's carrier and or mezzanine
      - Can be leveraged into more applications
      - Open competitive environment
      - Hopefully a healthy vendor ecosystem, and a long lifespan
  - Facilitates purchasing off the shelf for specialized FPGA applications
  - The modularity of FMC form factor
    - Enabling technology for integrating
      - ADC, digitizer, signal processing EPICS IOC all into one board
      - Simplified spares cabinet

Hardware Design – FMC Industry Standard

Hardware Design – Backplane Architecture



- Centrally switched versus meshed fabrics for packet networks
  - Considering design evolution inherent to Experimental Physics
    - Centrally-switched, increased likelihood slot agnostic
    - Meshed, increased likelihood slot personalities
  - In Experimental Physics perhaps the practical configuration is to prefer
    - Modules will function in any slot
    - Software doesn't need to learn the physical positions of other modules
    - Relax requirements to know a-prior per-slot throughput demands on the fabric
- Packet Switching Protocols
  - PCI Express
    - Is a network protocol
      - But is transparent to root complex oriented PCI driver software
    - Board-to-board DMA communication
    - The ubiquitous choice in modern electronics
      - Our gateway to MRF timing and industrial IO via PMC/XMC/Compact PCI monarch
  - Gigabit Ethernet
    - Reduced cabling when there are many CPUs in the chassis

Hardware Design –
Backplane Industry Standard

	Stds Org	Generic IO	Activity	EVR Timing	FMC	Rear Transition IO	Packet Switching Backplane
VME	VITA	+	+	+	rare	+	VXS, rare
Compact PCI	PCMG	+	+	+	rare	+	Various, fragmented
µTCA	PCMG	-		PMC*	+	*	+
µTCA.4	PCMG		-	PMC*	-	+	+
VPX	VITA	--	+	PMC	+	+	+

* rear transition IO, particularly PMC RTIO, hasn't been standardized in µTCA prior to dot 4

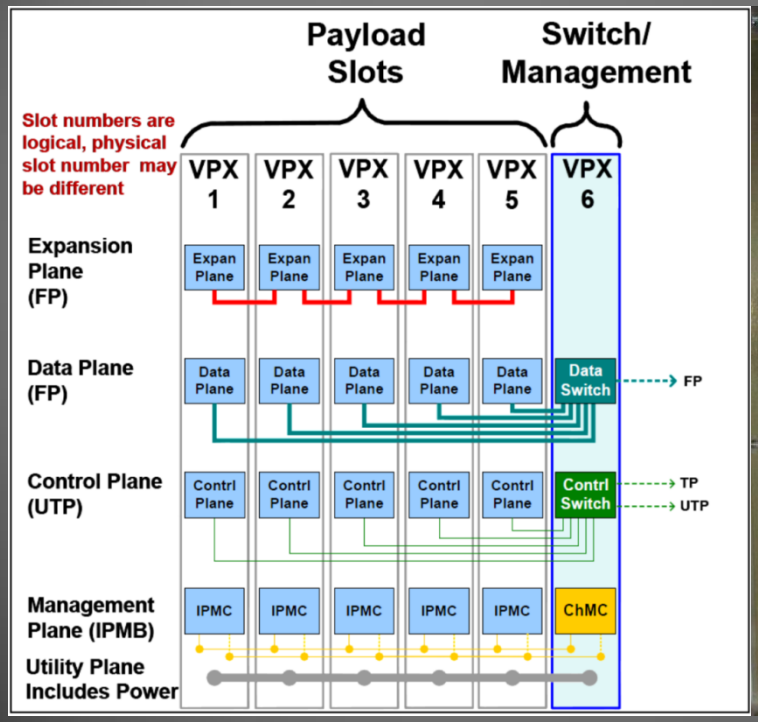
-- cost of proprietary VPX connectors encumbers its industry activity level but standardized PMC mezzanine rear transition IO diminishes this problem

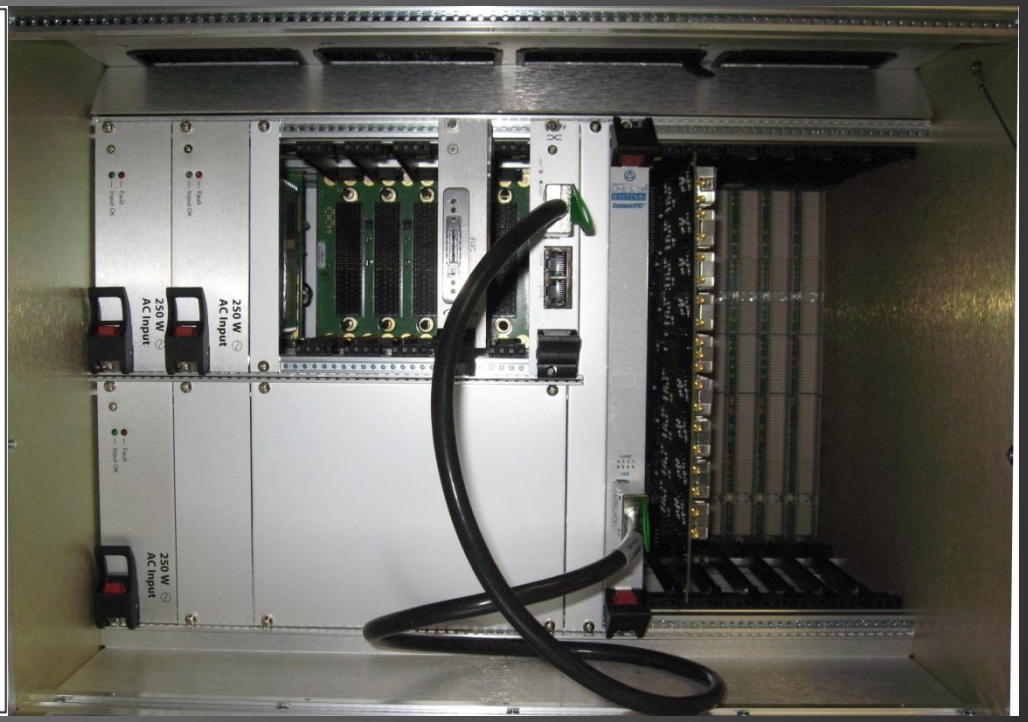
Hardware Design – We selected VPX

- The positives of VPX
  - High levels of vendor signal processing activity
  - Well standardized high-density rear-transition IO
    - Standardized PMC rear-transition IO for MRF timing
  - VPX 3U is a good match for our signal density requirements
  - Modern VPX backplane protocols are appropriate
    - For a green-field signal processing framework
- The negatives of VPX
  - Proprietary impedance-matched high-density *module* connectors
    - Expensive compared to costs of commodity IO modules

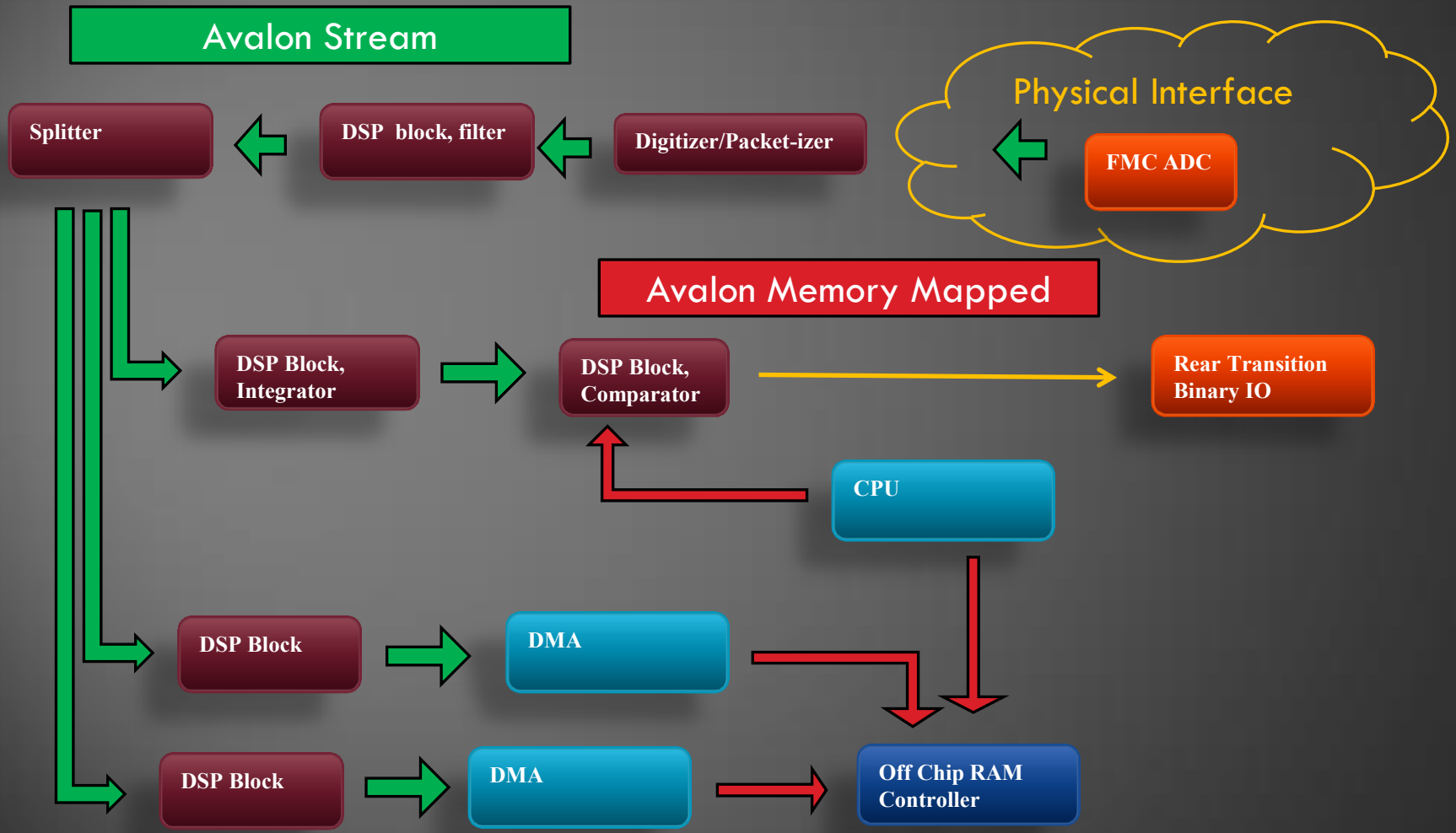
Hardware Design – Open VPX Centrally Switched 3U Backplane

- Open VPX, VITA 65, BKP3-CEN06-15.2.2-2





Hardware Design – Component-Based Signal Processing



Hardware Design – Component-Based Signal Processing

- Signal Processing Blocks Interconnected with Avalon Streams
  - We stay compatible with our RF group
    - An Altera workshop
- Signal Processing Blocks, implemented multiple ways
  - VHDL or Verilog
  - MATLAB/SIMULINK
  - IP Modules, off-the-shelf or open-source
- Softcore CPU, DMA, DDR Ram Controller, SPI Controller
  - Implemented with COTS IP

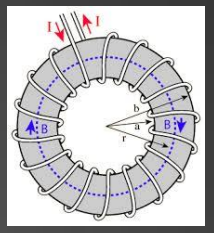
Hardware Design – Altera Build System

- Pin-assignment script detritus accumulates in the Altera project database
  - We need a clean rebuild
    - Only dependent on checkout from source code control
- Hardware builds can take a long time
  - We need an unattended build outside of the Altera Quartus GUI
  - Including automation of pin-assignment scripts for Altera IP
- Support also for
  - Signal tap logic analyzer integration
  - Partitioned builds

Signal Processing Algorithm – Kalman Filter

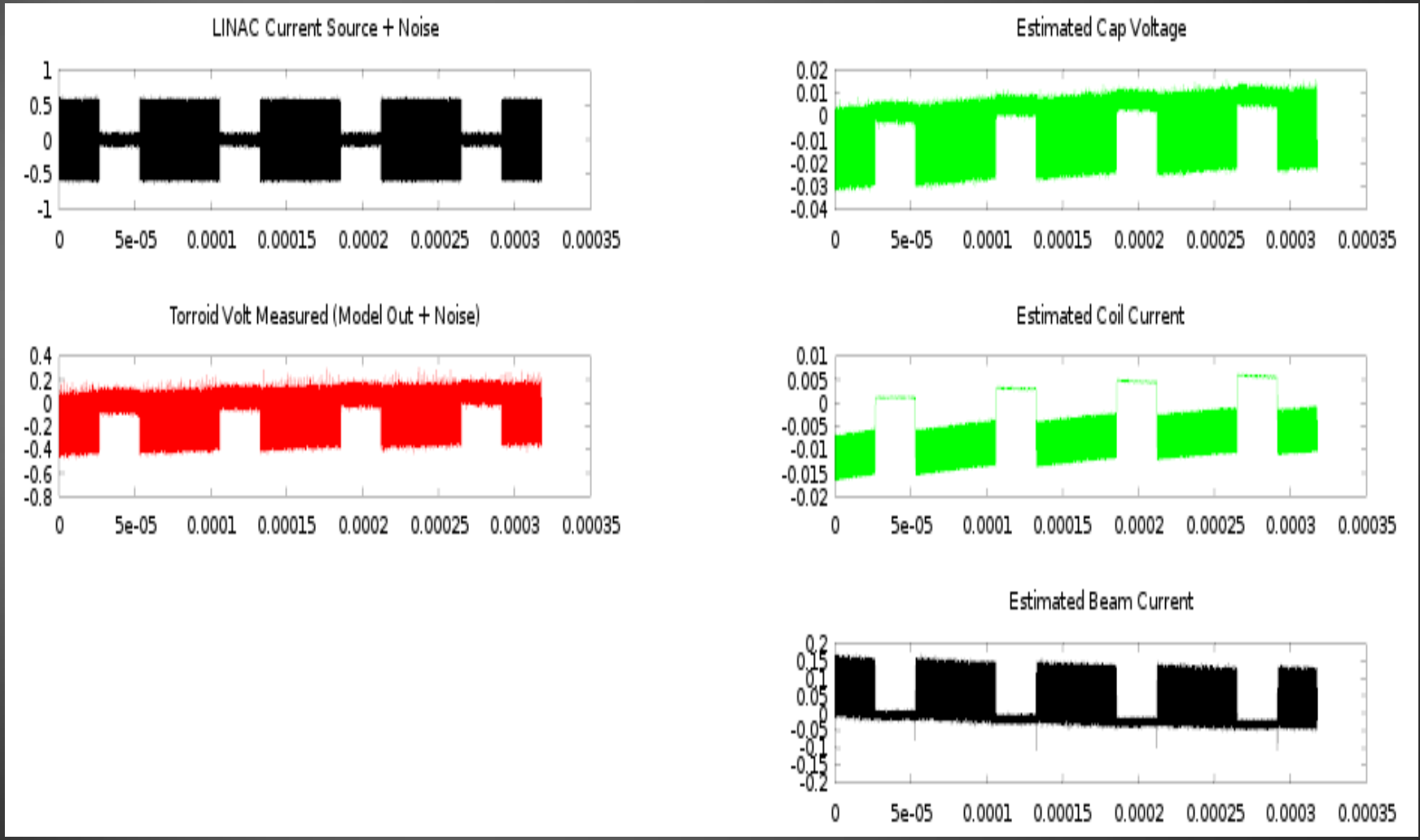
- Its possible to enhance the dynamic range of the system in the presence of noise
  - Oversample, average samples together
- The optimal way, use a Kalman filter
  - Average together
    - Model predicted state of the system based on previous estimate of the system's state
    - Together with a new measurement
  - Using a mathematically optimal weighted average
  - Forming a new estimate of the systems state
- A generalized approach suitable for use in a framework

Signal Processing Algorithm – Kalman Filter



- For example, a current monitor toroid sensor
  - Two dynamical states in this type of system
    - The current flowing in the toroid sensor
    - The voltage across the capacitance seen by the toroid's primary
  - Sensor noise is often a challenge
- The typical goal of legacy analog processing is to produce a reasonable facsimile of beam current in the time domain
  - But often at the expense of lost high frequency information
    - Example, legacy trans-impedance amplifier
- Perhaps a modern approach, with a simplified analog front end
  - Only matching signal amplitude and impedance to the ADC
  - FPGA estimates the states of the system from its noisy outputs via Kalman filter
    - Time domain facsimile of beam current is simple linear multiple of these states
  - Challenges remain
    - Proper toroid sensor *system characterization*
    - Algorithm sensitive to transfer function *pole placement*

Beam Current Monitor Sensor State Estimator Simulation in MATLAB



Software Design

- We have embedded the EPICS IOC within the FPGA Nios2 soft-core processor
  - Our perspective, direct connections from client to server are easier to diagnose on-call compared to system with proxy go–between gateway computers
  - This is very beneficial for instrumenting, debugging, and deploying the FPGA algorithm design
- Local enhancements to RTEMS support for Nios2 soft-core
  - Virtual RTEMS RTOS BSP
    - Implement once, deploy on multiple hardware designs
      - RF and diagnostics hardware designs use the same auto generated BSP
    - Automated system generation
      - First generate Altera BSP from Altera QSYS outputs
      - Extract configuration information
        - Base address, span, interrupt vectors, …
      - Generate source files for consumption by RTEMS
      - All automated in the GNU-based RTEMS build
  - Drivers
    - Ethernet MAC, Scatter-gather DMA, SPI Controller
  - Prioritized interrupt dispatch
    - *Essential* for guaranteeing synchronization with timing hardware
  - Task-level network daemon GNU debugger stub
    - *Essential* for converging on quality with deployed system

Cost Versus Benefit

- Opposing limits of the spectrum of options, creating new system
  - We can buy turn-key integrated system off-the-shelf
  - We can assemble a system from parts, obtained from multiple vendors
- The system assembled from parts certainly cost more to develop
  - Hopefully we realize some benefits
    - Better protected from component/vendor obsolesce
      - Industry standard interfaces, healthy competitive ecosystem
        - Incremental upgrade avoids a clean slate upgrade
    - Better protected from design obsolesce
      - We maintain expertise in-house
        - Can evolve our systems to meet future requirements at LANSCE
        - Can reprogram our algorithms if we don't meet original design specifications
  - Nevertheless
    - Success requires leveraging the framework
      - Over multiple projects and facilities

Conclusions

- We are compelled to replace the RICE systems at LANSCE
- We have chosen the FMC and VPX industry standards
  - A solid vendor neutral technical basis for a green-field system
  - Appropriate for a framework
- We are implementing component-based signal processing
  - Appropriate for a framework
- We have embedded the EPICS IOC at the lowest level
  - Direct client-server communication is easier to diagnose on-call
  - Internals of our algorithms can be directly controlled and monitored
- This locally integrated system costs more to develop
  - Hopefully we are well protected against component and design obsolesce
  - Success requires leveraging the framework into multiple projects