THE LANSCE FPGA EMBEDDED SIGNAL PROCESSING FRAMEWORK*

J. Hill, LANL, Los Alamos, NM 87544, USA

Abstract

During our replacement of some LANSCE LINAC instrumentation systems we have developed a common architecture for timing system synchronized embedded signal processing systems. Our design follows trends of increasing levels of electronics system integration; a single commercial-off-the-shelf (COTS) board assumes the roles of analogue-to-digital conversion and advanced signal processing while also providing the LAN attached EPICS IOC functionality. These systems are based on agile FPGA-based COTS VITA VPX boards integrating a VITA-57 FMC mezzanine site. Our signal processing is primarily developed at a high level specifying numeric algorithms in software source code to be integrated together with COTS signal processing intellectual property components for synthesis of hardware implementations. We will discuss our requirements, our decision point selecting the VPX together with the FMC industry standards, the benefits along with costs of system integrating multi-vendor COTS components, the design of some our signal processing algorithms, and the benefits along with costs of embedding the EPICS IOC within an FPGA.

SCOPE

Legacy VAX backplane integrated RICE control and instrumentation hardware systems are in service now at LANSCE for forty years. Our maintenance costs only increase and legacy schemes applying a single type of multiplexed ADC for all input signals are now obsolete. We are therefore compelled to replace them.

This paper describes electronics upgrade for a subset of RICE signals requiring real-time synchronization with a timing-system maintained data structures determining the modal operation of the facility. For example, we must flavour-stamp the incoming data streams from the sensors for correlation with the beam species being produced. The system described herein will be initially deployed within replacement loss-monitor, current-monitor, and beam-position-monitor systems at LANSCE. The embedded processor and support software described herein are utilized also within the upgraded RF control systems.

OVERARCHING REQUIREMENTS

The design is based on a reusable framework of modular off-the-shelf components. This approach offers benefits related to a shared spares, chassis-space, source-code, developer-effort, and developer-expertise. The design prefers components based on modular industry-standard long-life-span interfaces, prefers industry-standard development languages, and multiple sources for each vendor purchased modular function.

Furthermore, in a long lifespan facility, we recognize that even newly replaced systems will eventually become obsolete. With electronics components the only certainty is change. However, with a reusable framework, based on industry standard interfaces, our design can accommodate a path forward. Likewise, when one project comes to the end, if we develop our skills within a framework then hopefully some of our efforts can be preserved for use in the next project.

REQUIREMENTS

Digital Signal-Processing

Within a framework, the algorithmic requirements of individual diagnostic systems must be realizable on a shared hardware platform. In a modern system, typically an FPGA (Field Programmable Gate Array) is performing most of the signal processing tasks. Within a framework, the FPGA capacity must be sufficient to meet the needs of a diverse set of systems. Likewise, with a framework we must accommodate the signal capture requirements of many different signal types.

At LANSCE machine protection functions are currently implemented in analogue circuitry responding to inhibit the beam within a maximum overall system response of 6 μS. The signal processing framework shall be capable of responding fast enough so as to not preclude implementation of advanced machine protection functions initially supplying redundant inputs within the pre-existing machine protection system.

Real-Time Software

At LANSCE, it is necessary to beam-species flavour-stamp the incoming data streams. Due to the complexities of the modal behaviours of LANSCE, this task can't be easily accomplished in hardware, and we must therefore deploy software, running on a real-time operating system, in close proximity to the digital signal processing stream. This software must receive time deterministic interrupt response from both the timing system's event receiver module and the digital signal processing system. Certainly, this is also possible sans operating system, but an operating system is specified to improve productivity and reduce maintenance costs.

HARDWARE DESIGN

We are aware of some trends. New products offer increased density and integration; what was once possible within the confines of 6U modules is increasingly possible within the confines of 3U modules. Increasingly, a processor and Ethernet interface are cost effective for integration at the module level. We can purchase off-the-

* Supported by US Depart of Energy contract DE-AC52-06NA25396.

ISBN 978-3-95450-148-9

shelf hardware intellectual property modules for user integration into the FPGA. Increasingly, we can forgo register-level hardware languages for implementing signal processing algorithms in favour of numerical languages originally used only for software development.

VITA-57 FMC - FPGA Mezzanine Card

The VITA-57 FMC industry standard, for **F**PGA interfaced **m**ezzanine **c**ards, appears to be almost tailor made for a signal processing framework. With FMC a common FPGA development module, including an FMC site, can be leveraged over multiple application-specific analogue front-ends. We can identify a range of FMC products fulfilling a range of analogue requirements, offering previously unavailable signal density, all within a modular replaceable nine by seven cm form-factor. We are optimistic for an optimized cost to performance ratio leveraged over a reasonable lifespan, based on a healthy competitive vendor ecosystem; see Fig. 3.

Furthermore, the flexibility of FMC form factor is an enabling technology for increased integration; we can purchase off-the-shelf components for integrating a waveform digitizer, signal processing, EPICS IOC, outputs to the machine protection system, and the timing system's event receiver software all integrated into a single FMC carrier module.

Chassis Backplane

With modern backplanes we have options to choose between centrally switched (modules-to-switch) fabrics, and meshed (modules-to-modules) fabrics. Considering the natural tendency for design evolution inherent to experimental controls, perhaps centrally switched architectures are favoured over mesh fabrics for their flexibility. We can prefer slot agnostic signal processing modules, and also to accommodate evolving module throughput demands on the fabric. Also, centrally packet-switched backplanes offer reduced impact of communication topologies on software in contrast to mesh fabrics and legacy commander-slave backplanes. See Fig. 1.

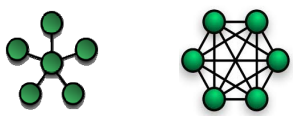

Figure 1: Central Switch (left) vs Mesh (right) Topology

The increased throughput available in modern backplane protocols is a secondary benefit; demand for higher levels of throughput occur when signal processing algorithms are pipelined together through the backplane, and when the software implementations are centralized. Also, with a real-time framework, additional contingency capacity is needed. Our timing Event Receiver Module, and industrial IO modules, require the PCI Express variant of packet switching. We observe also that PCI drivers typically require only small modifications before they are reused with PCI Express. Ethernet on the backplane is also desirable when there are increasing numbers of CPUs within a chassis, to simplify cabling. Proper use of backplanes with Rear Transition IO (RTIO) can result in simplified maintenance, improved cabling modularity, and reduced cable entropy.

In Fig. 2 is a brief summary of our backplane standard selection considerations. It can be observed that FMC carrier options are limited in VME, Compact PCI, and µTCA dot four. Both VPX and µTCA suffer from limited availability of generic IO modules; this issue is typically addressed via PMC or XMC mezzanine carrier modules, but closer inspection reveals that RTIO for such carriers isn't well standardized in µTCA. This was a particularly strong consideration due to our event receiver timing module only being available in these backplanes in PMC. RTIO *is* standardized under µTCA dot four, but vendor activity is scarce, particularly for PMC carriers that route signals to the back of the chassis.

	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	+	+	+

Figure 2: Backplane Decision Summary.

We selected VPX; among its positives were high levels of signal processing vendor activity, well standardized high-density rear-transition IO, the VPX 3U form factor's match to our signal density requirements, and a modern backplane architecture appropriate for green-field signal processing systems. The primary negative with VPX is the expense of its proprietary high-density impedance-controlled module connectors. We also observe that, currently, there is diminished signal-processing vendor activity in 6U compared to 3U VPX.

Backplane and Chassis Architecture

We selected a centrally-switched 3U Open VPX backplane, see Fig. 4. This backplane supplies a single hybrid, PCI Express and Gb Ethernet, switch slot. Due to higher per-slot cost with this backplane, we opted also for a chassis design incorporating a 2nd auxiliary 6U Compact PCI backplane transparently integrated by the VPX PCIe switch, and also modular N+1 redundant 3U Compact PCI power supplies. A front-to-rear internal cable raceway at the top of the chassis was also specified.

Our VPX physical slot-keying design designates only two slot-profiles. Key position two prevents installation of switches into signal-processing slots, and the converse. Typically, module key-blocks for position one will be installed making no distinction among signal-processing slots, with selective keys required only by applications daisy-chaining multi-stage DSP algorithms between modules on the Open-VPX expansion-plane due to insufficient FPGA or FMC capacity; see Fig. 3.

ISBN 978-3-95450-148-9

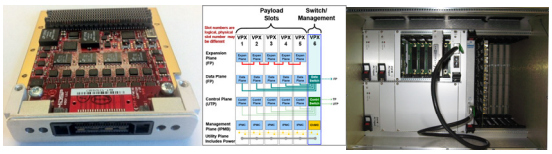

Figure 3: FMC module 125Msps, 14 bits, 16 Channels, 9 cm x 7 cm; Open VPX Backplane Choice; BKP3-CEN06-15.2.2-2, Chassis

Signal Processing Framework

Our design is based on the FPGA vendor's Avalon Streaming Interface allowing component-based modular stages of the signal processing algorithm to be connected together using the FPGA vendor's system integration tool. We have freedom to create algorithm components from high level languages such as MATLAB, create components from register level hardware description languages, or to acquire production ready components. We will also provide facilities to remotely upgrade the algorithms running in our production chassis, over the network. The analogue input is accessed through the same Avalon Streaming Interface. This facilitates algorithm reuse with more than one physical FMC board. See Fig. 4.

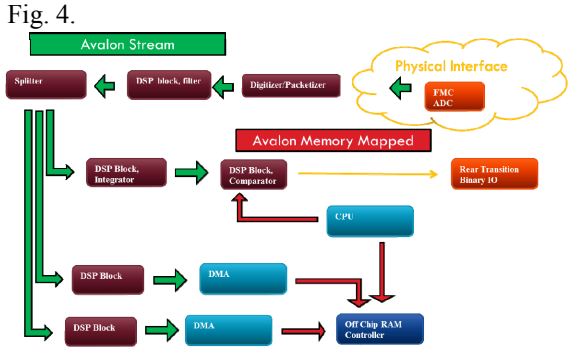

Figure 4: Component-Based Signal Processing

Generalized Filtering Algorithm

The dynamic range of our measurements in the presence of noise can be extended by averaging elements together in an oversampled input waveform; a mathematically optimal way to do this is to employ a Kalman Filter[1]. With a Kalman Filter we combine the model predicted state of the system together with a new measurement using a mathematically optimum weighted average. For example, with a current monitor toroid sensor we have at least two internal states for the system which can't be observed perfectly in the sensor data due to the presence of noise, and also due to imperfections in our estimates for component values of the sensor. These state variables are the voltage across the capacitance seen by the toroid transformer's primary, and the toroid's current. Mathematically optimum weights for these averages can be computed based on the statistical covariance of the system, where the covariance is a measure of the uncertainty of model predicted values of the system's state, sensor noise, and the input noise. Optimum behaviour of the algorithm is contingent on the noise, and other imperfections in our observation of the system varying with a Gaussian random-variable statistical distribution. Fortunately, Gaussian distributions are common with physical systems. The Kalman filter is a general approach, appropriate for a framework. Figure 5 shows some output from the current state of the MATLAB modelling for our filtering algorithm.

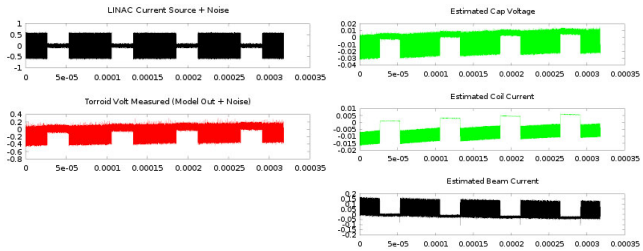

Figure 5: Current Monitor Estimator Simulation.

Furthermore, the framework is optimized to facilitate development of advanced machine protection signal–processing functionality. Currently, these functions are implemented within analogue systems, but this hardware is aging and in select situations becoming obsolete in terms of man-power maintenance costs. The legacy analogue-based radiation detector loss-monitor system compares sensor data against a fault threshold to determine the state of a fast-protect input, and the beam current transmission monitor system fast-protect input trips when an integrated beam current difference for two physically adjacent current sensors exceeds a threshold. We can observe that overall system reliability, in terms of detecting faults, might only improve when providing additional redundant fast-protect inputs from flexibly developed, and therefore potentially more advanced, digital-based algorithms adapting well to our needs, as the capabilities of LANSCE evolves.

Processor

Recently there are new options to integrate soft-core CPU intellectual property modules into the FPGA. Certainly, ASIC CPUs run at higher clock rates, but they are less agile. With a soft-core CPU we can directly connect our processor to the incoming signal-processing stream within the FPGA. Figure 6 shows our processor design including snap-in interfaces for FPGA boundary specific interface components. This approach allows for a reusable design which can be adapted to different boards and signal-processing applications. This type of memory-mapped master-slave integrated architecture is fairly easy to produce using the vendor's system builder tools. Impact on FPGA compile delays can be reduced when the CPU is isolated in an independent design partition from the off-chip interfacing FPGA intellectual property modules (IP), and or the signal-processing application.

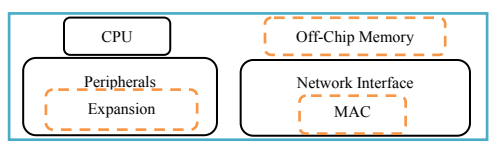

Figure 6: Processor Design.

ISBN 978-3-95450-148-9

FPGA Build System

We have created a gnu-make-based build system for our FPGA development environment. This approach has some benefits compared to working solely in the vendor's graphical-user-interface based tools; we separate out the source code form the large number of generated files produced during a build, we can easily perform a clean rebuild from source so that pin assignment script detritus does not accumulate in the vendor's project database, and we can easily orchestrate an unattended rebuild including automation of any pin-assignment scripts that the vendor's intellectual property (IP) modules might require. A build system also results in a more consistent installation of the signal processing applications, and this hopefully facilitates developer portability between projects.

SOFTWARE DESIGN

Network Architecture

The LANSCE system is based on EPICS, and we can simplify our network architecture when embedding the EPICS IOC directly within our diagnostics systems. Our perspective is that systems based on protocol-converting-gateways are more difficult for on-call personnel to diagnose. This is especially true if they are closed software systems with no mechanisms to remotely observe their internal health within the control room. In contrast, when the IOC is embedded we have new opportunities to observe the internal state of our algorithms, and adjust parameters while the systems are running. Operational experience is proving that these capabilities may be essential for converging to the highest levels of quality. The maximum Ethernet throughput for the FMC carrier based IOC is currently at around 120 Mbps. Considering the number of this type of system deployed times this figure, we conclude that it is essential to govern the rate of EPICS subscription updates for each species beam, synchronized by the timing system.

Real-Time Software

We use the RTEMS real time operating system with some local enhancements to its support for the Altera Nios II Soft-Core Processor. From our perspective, a primary benefit of RTEMS is capabilities for a minimalist installation.

Board-Support Software

With a soft-core CPU we have options to develop a virtual board support package (BSP). With a conventional BSP custom code is required for each hardware design. In contrast, with a virtual BSP more of the BSP is in a reusable library of driver components consistent with the library of IP used to instantiate the hardware. If the processor is instantiated into a new board, the BSP software auto-configures itself off of the outputs from the vendor's system builder tools. For example, with RTEMS we first generate the vendor's BSP, extract the vendor's configuration information, and then auto-generate the necessary source files used to configure the RTEMS BSP. All of these steps are automated in the gnu-autotools-based build for the NIOS II RTEMS BSP. This approach has enable instantiation of the same RTEMS BSP into the processor design for the signal processing framework, and also a different processor design currently being deployed into the low level RF control boards. We have implemented an RTEMS task-level network-accessible GNU Debugger stub, which is perceived to be essential for properly converging to the highest levels of quality with this type of system.

COST VERSUS BENEFITS

When creating a new system there can be two opposite ends of the spectrum approaches; we can buy an integrated system from a single vendor, or at the other end of the spectrum we can assemble a system from parts, obtained from multiple vendors. The system assembled from parts certainly costs more in terms of in-house development time, but we are hopefully realizing some benefits when we are better protected from component obsolesce, and lack of design malleability. Hopefully our components are easily upgraded because they are based on industry standards, and healthy competitive vendor ecosystems. Since we have expertise in-house, then we can reprogram components that do not meet original design specification or that need flexibility to meet future requirements at LANSCE. However, for this type of system to be successful we must amortize its cost over multiple designs and installations. This requires a framework so that we can leverage our skills and components.

CONCLUSIONS

We are compelled to replace certain aging electronics systems at LANSCE. We have chosen the FMC and VPX industry standards as a solid technical basis for a healthy multi-vendor ecosystem, appropriate for a signal processing framework. We have implemented a component-based signal processing architecture, appropriate for a signal processing framework. We have embedded the EPICS IOC at the lowest level permitting the internals of our algorithms to be directly monitored and adjusted while they are in production use. It costs more, due to additional development time, to assemble this type of system from parts, but the benefits are hopefully visible when the system is malleable to our future needs while also reusable over a range of systems, projects, and facilities within a framework.

REFERENCES

[1] K. Shanmugan, A. Breipohl, Random Signals; Detection, Estimation, and Data Analysis, 1988

ISBN 978-3-95450-148-9