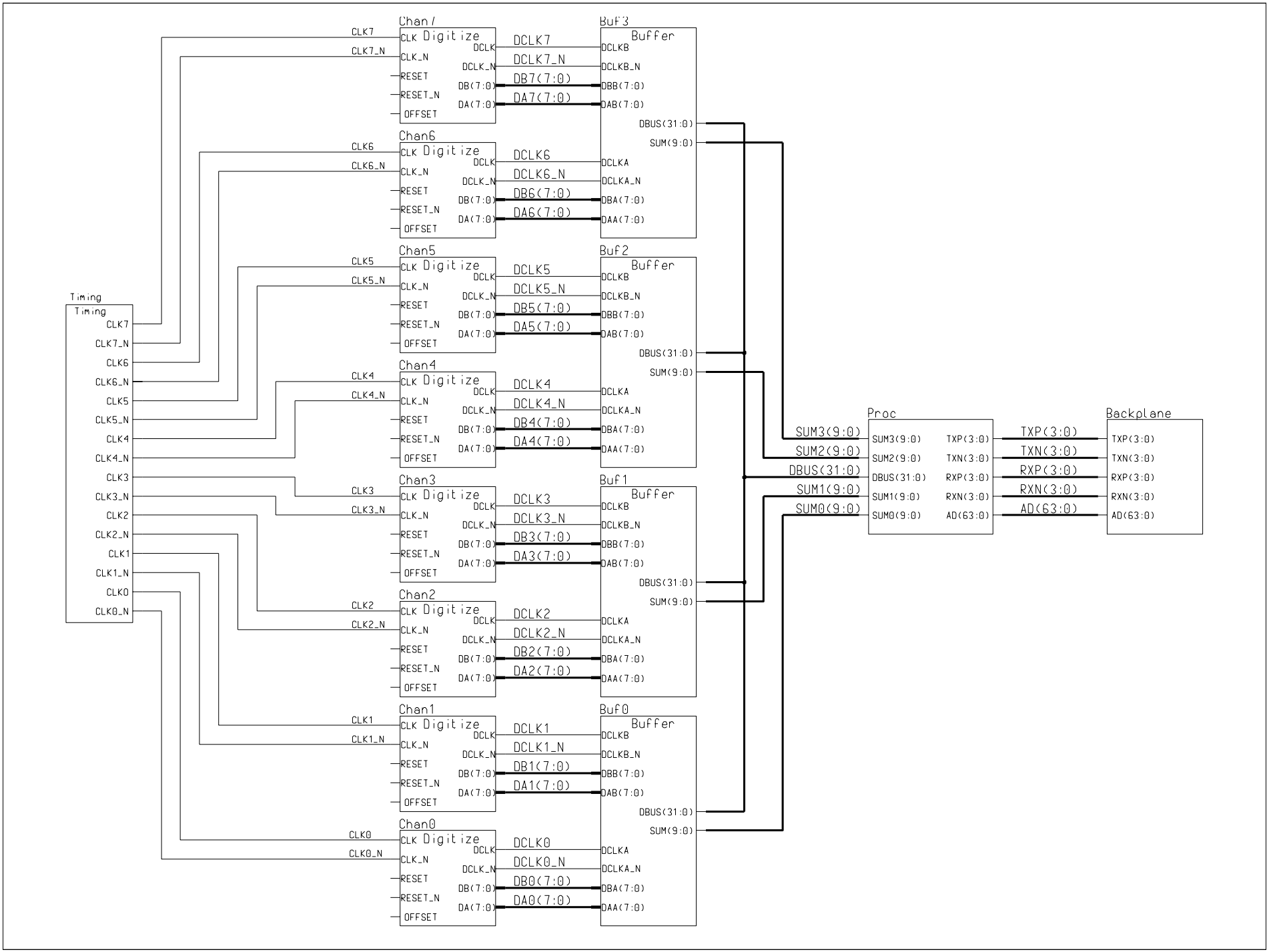
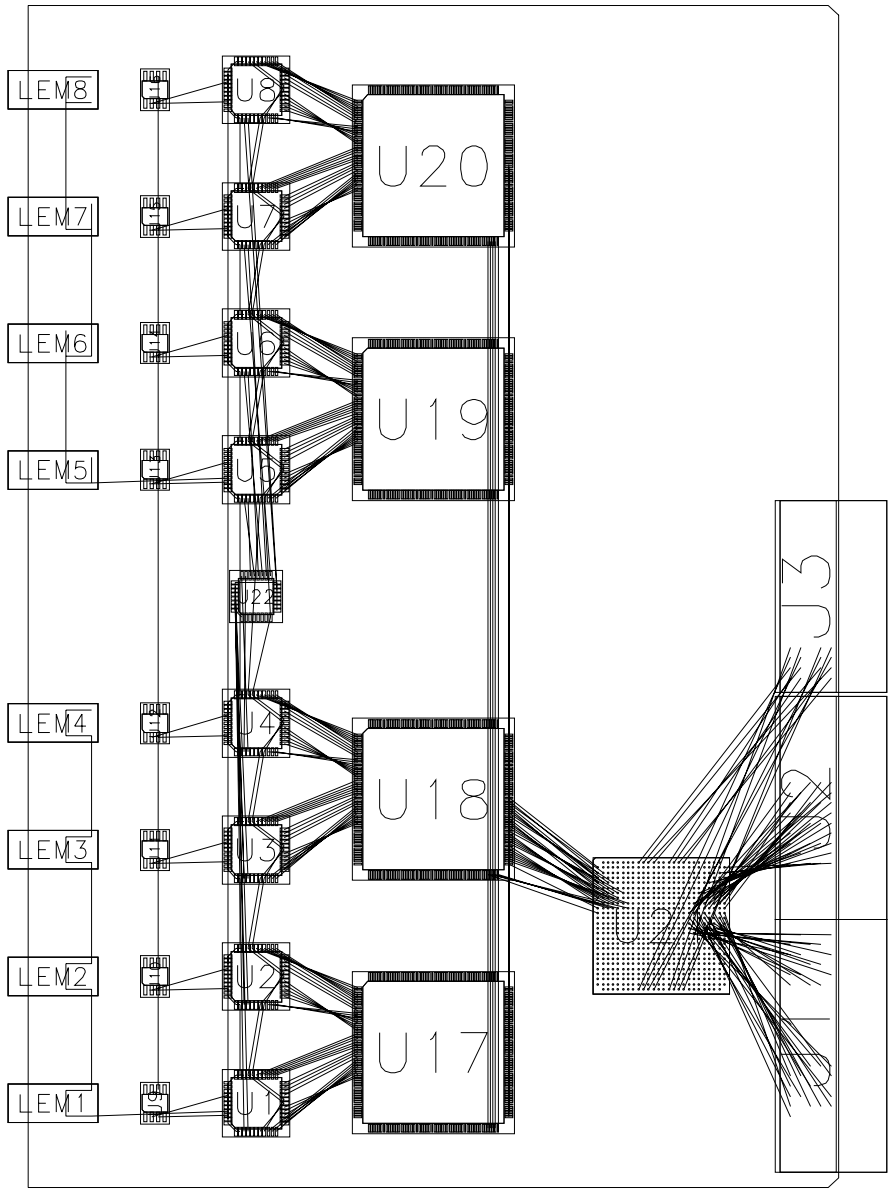

Timing
Timing
CLK7
CLK7_N
CLK6
CLK6_N
CLK5
CLK5_N
CLK4
CLK4_N
CLK3
CLK3_N
CLK2
CLK2_N
CLK1
CLK1_N
CLK0
CLK0_N
Chan7
Clk Digitize
Chan6
Clk Digitize
Chan5
Clk Digitize
Chan4
Clk Digitize
Chan3
Clk Digitize
Chan2
Clk Digitize
Chan1
Clk Digitize
Chan0
Clk Digitize
CLK_N
RESET
RESET_N
OFFSET
DCLK
DCLK_N
DB(7:0)
DA(7:0)
DCLK7
DCLK7_N
DB7(7:0)
DA7(7:0)
DCLK6
DCLK6_N
DB6(7:0)
DA6(7:0)
DCLK5
DCLK5_N
DB5(7:0)
DA5(7:0)
DCLK4
DCLK4_N
DB4(7:0)
DA4(7:0)
DCLK3
DCLK3_N
DB3(7:0)
DA3(7:0)
DCLK2
DCLK2_N
DB2(7:0)
DA2(7:0)
DCLK1
DCLK1_N
DB1(7:0)
DA1(7:0)
DCLK0
DCLK0_N
DB0(7:0)
DA0(7:0)
Buf3
Buffer
Buf2
Buffer
Buf1
Buffer
Buf0
Buffer
DCLKB
DCLKB_N
DCLKA
DCLKA_N
DBB(7:0)
DBA(7:0)
DAB(7:0)
DAA(7:0)
DBUS(31:0)
SUM(9:0)
Proc
SUM3(9:0)
SUM2(9:0)
DBUS(31:0)
SUM1(9:0)
SUM0(9:0)
TXP(3:0)
TXN(3:0)
RXP(3:0)
RXN(3:0)
AD(63:0)
Backplane



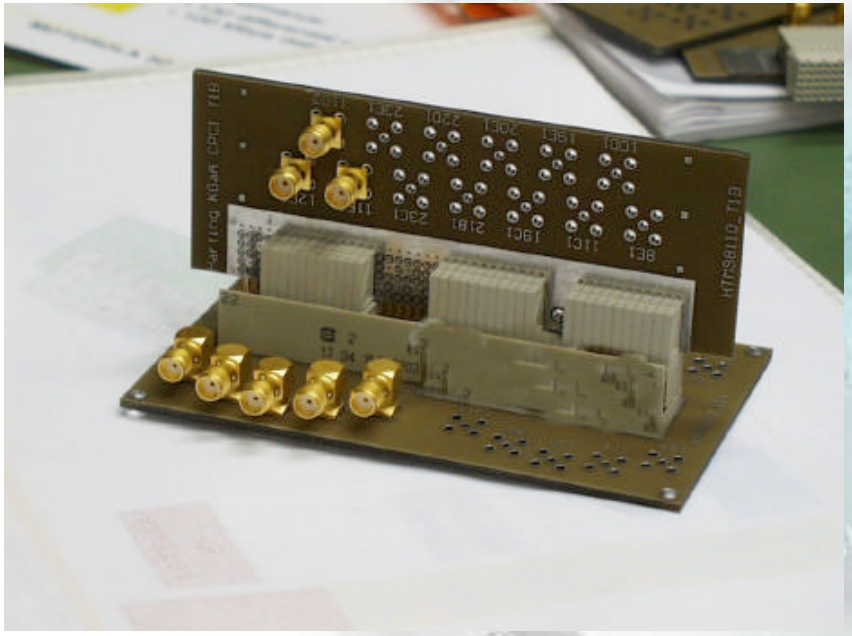
XC3S50 $15

XC2VP4 $130

Test setups and devices under tests

VITA-meeting in july:

har-bus® HM Board (with/without integrated shielding):



- 50Ω stripline-traces resulting in 100Ω Ω differential-impedance
- identical trace-lengths
- 3 identical boards: 2 boards with integrated shielding plates (with/without grounding row C); 1 without.

Goal: Informations about row/column influence, crosstalk-effects, pin-pattern and row-position on the signal-integrity. Basis are multilayer-boards and commercial bus-transceivers

People
Power
Partnership

Summary

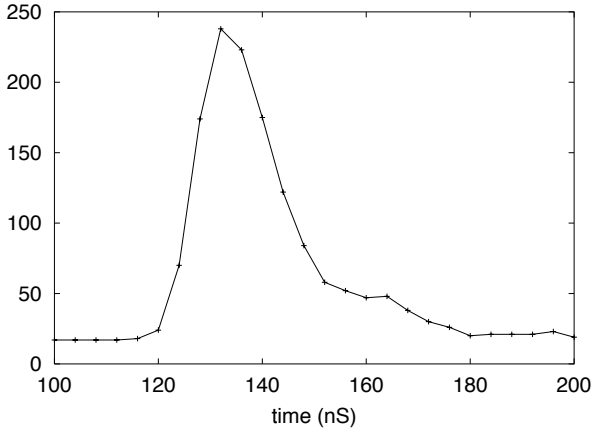
The measurements performed so far on differential routed edge-coupled coplanar microstrips with ground-plane indicate that:

- **the bare connector influence is marginal even at extreme datarates like 5 Gbps**
- **no significant effects due to counterboring (i.e. backdrilling) could be measured**
- **a nearly perfect differential signal-source like the used differential TDT is the necessary basis for a proper "de-embedding" of the connector**
- **the tests are in line with the former tests (VITA-july: multilayer-boards, TLK-drivers, PTH-SMAs): row A is the worst**

The *har-bus*® HM connector shows good performance in applications, where the differential data-rate approaches the 2.5 Gbps-region and higher !

Calorimeter fADC processing

GlueX-docs 424 - 427



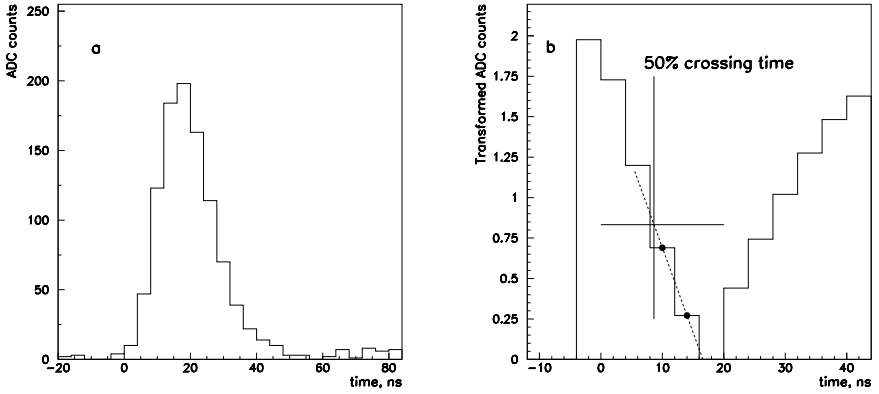


Figure 2: A) The simulated response of an 8-bit, 250 MHz FADC generated from a typical pulse by the method described in the text. B) The FADC response transformed by eqn. 2 and an illustration of the method used to find the 50% crossing time.

Per channel:

extract time slice
sum samples
find peak(s)
sparsify?

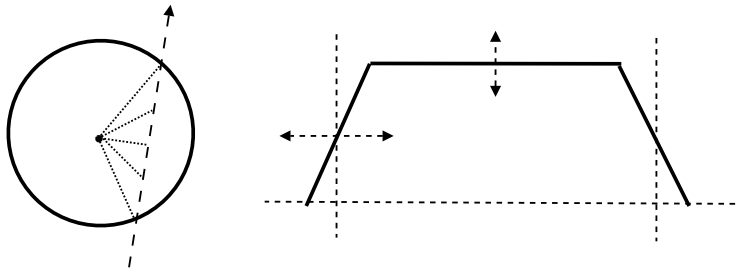
Per board:

sparsify?
look up time (64Kx8)
format/block/packet
send?

CDC processing:

dE/dX ∝ average pulse height?

leading edge time
trailing edge time!



FDC cathode processing:

charge only?
position algorithm?

non-calorimeter PMT processing:

charge only?
timing?

8th International Conference on Accelerator & Large Experimental Physics Control Systems, 2001, San Jose, California

WEAP023
cs.DC/0111033

MODERNISING THE ESRF CONTROL SYSTEM WITH GNU/LINUX

A.Götz, A.Homs, B.Regad, M.Perez, P.Mäkijärvi, W-D.Klotz
ESRF, 6 rue Jules Horowitz, Grenoble 38043, FRANCE

Abstract

The ESRF control system is in the process of being modernised. The present control system is based on VME, 10 MHz Ethernet, OS9, Solaris, HP-UX, NFS/RPC, Motif and C. The new control system will be based on compact PCI, 100 MHz Ethernet, Linux, Windows, Solaris, CORBA/IIOP, C++, Java and Python. The main frontend operating system will be GNU/Linux running on Intel/x86 and Motorola/68k. Linux will also be used on handheld devices for mobile control. This poster describes how GNU/Linux is being used to modernise the control system and what problems have been encountered so far[1].

1 INTRODUCTION

The ESRF control systems control 3 accelerators and 32 beamlines. They have been built using the same technology and are completely compatible. They were built 10 years ago based on the state-of-the-art technology ten years ago. This included VME, 10 MHz Ethernet, OS-9, Solaris, HP-UX, NFS/RPC, Motif and C. Most of these technologies have not evolved over the last few years. In our search for better tools, support, ease of programming, and overall stability and quality we have put all our old technologies to the test. Our main criterium was which technology or tool will allow us to offer users a better control system. A better control system means one which offers more features to users without losing any of the present good features.

The result of this technology survey was 100 MHz Ethernet, VME (for the existing hardware), CompactPCI (cPCI) and PCI for new hardware, Linux as the main frontend operating system, Windows for commercially supported hardware and software, Solaris and GNU/Linux as the main desktop operating systems, CORBA/IIOP as the new network protocol, C++, Java and Python as the main programming languages.

2 WHY GNU/LINUX?

What does GNU/Linux offer that other systems don't?

1. FREEDOM! Freedom in this context means access to all the source code so that it can be compiled, understood and improved. An additonal freedom is the freedom from supplier pressure and fees.

[1] work supported by J.Klora, J.M.Chaize and P.Fajardo

3 LINUX/M68K + VME

The ESRF has over 200 VME crates installed. This represents an investment of millions of Euros as well as many tens of years of work in hardware and software development. Any modernization project must take this investment into account. The modernization foresees two ways to do this: using the Motorola CPU's (MVME-162) to run GNU/Linux directly, or replacing the CPU with a bus extender which allows the VME bus to be controlled from PC running Linux/x86. This section describes the first option. The bus extender solution is discussed in the next section.

4 LINUX/X86 + BUS EXTENDERS

The modernization project of the instrument control at the ESRF using GNU/Linux supports two main hardware platforms: PCI/cPCI and VME. The former provides access to the most recent interface boards developed for a highly demanding market, and hence, with better performance/price ratios. The latter is needed for a gradual transition between the current VME instrumentation and the PCI technology. VME boards can be controlled from a Motorola MVME CPU or from a PC through a PCI/VME bus extender, both running GNU/Linux as OS.

Setups based on the PCI architecture have been mounted using both a desktop PC and an industrial PC that implements the PICMG standard. Remote VME crates are controlled through SBS Technologies PCI/VME bus extenders, and cPCI crates are directly linked to the main PCI bus by means of National Instruments MXI-3 PCI/cPCI/PXI bus adapters. These adapters expand by a large factor the amount of hardware that can be managed by a single host. Furthermore, both MVME and PCI GNU/Linux can independently control boards in the same crate, providing even more possibilities for the VME - PCI transition.

9 REALTIME

"Linux isn't realtime". This is true therefore we do not claim to do any realtime with GNU/Linux. However we have found that for our applications we need little or no realtime. Most realtime needs are delegated to hardware or DSP's. Where we do need soft realtime (i.e. 99% guarantee) we use interrupt routines in device drivers. We measured an interrupt response time of $< 50\mu s$ for Linux on 68k and x86. Using a driver interrupt routine we achieve a soft realtime response of $500\mu s$ for a function generator (providing we do not recompile the kernel at the same time!). For the rest of our applications *"as-fast-as-possible"* is good enough. And for that GNU/Linux on commodity hardware is surprisingly good.

10 PROBLEMS

GNU/Linux is not without problems. The main problems we have identified so far are:

- the standard GNU/Linux distributions are not easily adapted to running on diskless systems
- most commercial hardware does not have GNU/Linux drivers but Windows drivers

11 CONCLUSION

There is a viable alternative to Windows 95/98/ME, NT/2000/CE/XP for building control systems and it is called GNU/Linux! Linux is sufficiently mature for the task and even offers some advantages i.e. it is easier to program, is better adapted to distributed control and is free of commercial pressure.

Abstract

The inexorable march of Moore's Law has given engineers the capability to produce front end equipment with capabilities and complexity unimaginable only a few years ago. The traditional standardized crate, populated with off-the-shelf general-purpose cards, is ill suited to the next level of integration and miniaturization. We have reached the stage where the network protocol engine and digital signal processing can, and should, directly adjoin the analog/digital converters and the hardware that they monitor and control.

The current generation of Field Programmable Gate Arrays (FPGAs) is an enabling technology, providing flexible and customizable hard-real-time interfacing at the downloadable firmware level, instead of the connector level. By moving in the direction of a system-on-a-chip, improvements are seen in parts counts, reliability, power dissipation, and latency.

This paper will discuss the current state-of-the-art in embedded, networked front end controllers, and gauge the direction of and prospects for future development.

1 THE CRATE AGE

For decades, CAMAC[1] and VME[2] crates have formed the basis for new designs of accelerator front end equipment. These designs still make a certain amount of sense when 100% of the desired functionality can be assembled using off-the-shelf boards.

Crates have their origin in the times when no single board had, or could have, enough interface gear to run a piece of equipment. It was reasonable to line cards up in a crate to get enough digital inputs, digital outputs, analog inputs, and analog outputs to meet the needs of a system.

As a natural consequence of Moore's Law[3], the amount of functionality available on a board has risen progressively. Last year's system fits in today's crate, last year's crate fits on today's circuit board, and last year's circuit board fits on today's chip.

A side effect of this progression is that, for the fixed form factor of a crate, the number of connection points to a board is larger, so more wires are involved. To justify having the large cost overhead of a crate, people are motivated to "fill

* This work is supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. The SNS project is being carried out by a collaboration of six US Laboratories: Argonne National Laboratory (ANL), Brookhaven National Laboratory (BNL), Thomas Jefferson National Accelerator Facility (TJNAF), Los Alamos National Laboratory (LANL), E. O. Lawrence Berkeley National Laboratory (LBNL), and Oak Ridge National Laboratory (ORNL). SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy.

http://recycle.lbl.gov/~ldoolitt/icalepcs2003/WE601.pdf

EMBEDDED NETWORKED FRONT ENDS - BEYOND THE CRATE*

Lawrence R. Doolittle, LBNL, Berkeley, CA 94720, USA

The flexibility and end-to-end integration of an FPGA-based SOC make it plausible to use Ethernet with a hard real time mind-set that is inconceivable using a CPU and a conventional MAC. Frame preamble and header information can be sent down the wire while results are still being acquired from the hardware.

6 NETWORK CHOICES

At some point in the chain from hardware to operator, standards (as published by sanctioned standards bodies) are essential for communication between hardware built by different people at different times.

There are many historical standards for parallel bus attachments of peripherals to computers: CAMAC (IEEE-583), VME (IEEE-1014), VXI, GPIB (IEEE-488), SBUS (IEEE-796), ISA/AT, ATAPI (ANSI NCITS 317-1998 and later), PCI/cPCI. At the time of this writing, all are considered obsolete or dying, in many cases explicitly replaced with a serial equivalent. PCI sees extremely wide use, but is also very political, and many commercial interests appear eager to upgrade or replace it soon. Modern serial buses include Ethernet (IEEE-802), Firewire (IEEE-1394), Fibre Channel, USB, CAN (ISO 11898), SATA, and ATM.

Ethernet is both the oldest and most vibrant. It is in the heart of the wireless storm. Power Over Ethernet[22], which provides up to 13 W for peripherals over the same CAT5 cable as the network, is just taking off. Fiber and twisted pair transmission speeds are set for another jump in speed and/or availability. It's very hard to imagine any difficulty connecting Ethernet-based gear to the Internet anytime in the next two decades. The same cannot be said about *any* of the other listed protocols.

With ubiquitous CAT5 cable, 100BaseTX and 1000BaseT Ethernet will reach 100 m. On a fiber physical layer, 100BaseFX in full-duplex mode will reach 2000 m, and 1000Base-LX on a single mode fiber will reach 3000 m[23].

While not normally thought of as a hard-real-time link, point-to-point Ethernet does have deterministic latency. Direct links between networked front ends could take advantage of that to implement wide-area feedback and interlocks.

8 CONCLUSIONS

Networked front end hardware has tremendous opportunity to make accelerator electronics simpler, cheaper, more featureful, better understood, and more reliable. By distributing the hardware closer to the gear it controls, field wiring becomes quieter and more maintainable. Standardized high speed network communications between front end modules and the global control system maximizes short and long term flexibility, and minimizes installation costs.

Since so much of the intellectual content of the devices will reside in its programming, it is appropriate to suggest widespread Internet-based collaboration within the community, as exists now in the SNS project and the EPICS collaboration. This "many eyeballs" approach can drive up quality and drive down costs compared with the "lock it in the desk drawer" approach.

Many more changes are on the horizon. Even with the demise of Moore's Law looming in the next few years, imaginative applications of programmable digital circuitry will continue to enhance the performance and capabilities of front end hardware.