PC BASED INSTRUMENTATION, CURRENT EFFORTS AND FUTURE TECHNOLOGY*

M. Stettler, L. Day, J. Power
Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract

Instrumentation based on the PC architecture has been fielded at many accelerator sites. The wide variety, high performance, and reasonable cost of both commercial hardware and software make this platform an attractive choice for instrumentation. Current efforts on the SNS accelerator include beam position monitors, wire scanners, and beam current monitors. The design issues related to PCI interfacing and mixed signal isolation in the SNS diagnostics are discussed. The detail of the interface between custom hardware and commercial software is also presented. Design issues regarding synchronous acquisition and processing of data in non real time systems are addressed. Current and future trends in PC hardware peripheral interconnects, including USB2, 1394, and PCI express are expanding the options available to instrument designers. The relative strengths of these interconnects for instrumentation and the state of commercial software support is presented.

OVERVIEW

Instrumentation based on commodity computing hardware, such as the PC, is becoming a common occurrence. The high performance and low cost of this platform usually compares favorably to custom and industrial bus based solutions. The many variants of the PC architecture cover a vast array of performance and environmental requirements. In addition to the hardware advantages, a very large selection of off the shelf software exists for this platform. In addition to the obvious choices of operating systems, a variety of commercial application development environments (ADE), control system toolkits, and system utilities are available. These resources facilitate efficient development of intelligent instrumentation.

Instrumentation Issues

When packaging an instrument in the PC form factor, several issues need to be addressed. Current PCs utilize the PCI bus as the standard in-box I/O standard, and since the PC was not designed as an instrumentation platform, electrical noise and power supply regulation is often problematic.

The PCI bus provides a capable, high speed interconnect for instrumentation. The automatic configuration functions of OS or motherboard BIOS allow substantial flexibility in obtaining system resources without resorting to custom configuration software. In addition, the ability to "bridge", or create a local PCI bus on a PCI card, allows complex, high performance subsystems to be easily supported.

Since PC enclosures are designed for computers, several analog isolation issues exist. In general the power supply regulation is inadequate for precision analog measurements, requiring local DC/DC converters and filtering. In addition, the typical PC power supply does not have more than a few watts of power available at -12V, and -5V.

Perhaps the most limiting feature of the PC form factor is the limited front panel space offered by the standard PCI card. In some cases, blank panels in adjacent slots can be utilized, but often custom modifications need to be made in the enclosure. This is often the case when 1 or 2U rack mount enclosures are used.

Software Implementation

The basic software structure of an intelligent instrument consists of the basic operating system, driver software to support the acquisition hardware, any custom utilities required to provide a stable application programming interface (API), application level software which implements the instrument's functionality, and communication software (generally known as middleware) which allows for remote operation.

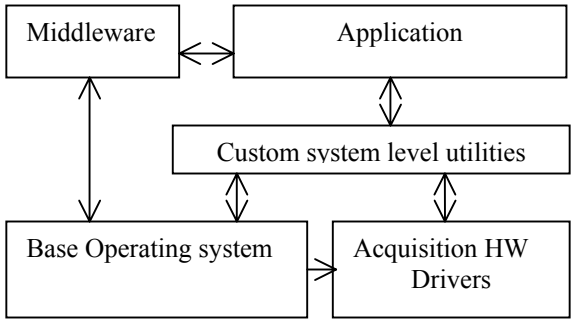


Figure 1: Basic software structure.

The OS selection for a PC based instrument hinges on both system requirements and the desire to include commercial software products in the final product. Commodity operating systems such as Microsoft Windows or one of the versions of LINUX provide access to numerous high quality commercial products at the expense of real time performance, although real time extensions are available for both. Dedicated real time operating systems such as Wind River vxWorks provide excellent real time performance at the expense of commercial tools.

* Work supported by the office of Basic Energy Science, Office of Science of the US Dept. of Energy, and Oak Ridge National Lab.

Unless the acquisition hardware is a commercial product, some form of driver software is required to interface the hardware to the OS. When using commodity operating systems, several generic drivers are available commercially, relieving developers of the task of driver development. Usually some additional utility routines are required to implement a hardware API for the application.

When utilizing a commodity OS, the issue of real time data acquisition must be addressed. While hardware can be triggered by external sources, a method must be implemented to buffer data when the application cannot react in a timely manner. This is typically accomplished at the driver level, which even in non real time systems provides microsecond response. This requirement often leads to the development of "poorly behaved" drivers to support real time acquisition, but is not a problem for a single acquisition card type in a given system.

The Application is the core functionality of the instrument, and is implemented using either low level programming languages (such as C++), or an application development environment (ADE). Modern ADEs provide powerful tools for the manipulation and analysis of data, as well as intuitive tools for creating local user interfaces. The raw performance of applications developed with many of these tools approaches that of low level programming languages.

In order to allow remote control and data acquisition, some form of communication middleware is required. The choice here is dictated by the central control system the instrument is to be used with. Unfortunately, there is no real standardization of the APIs for middleware products, so switching is currently a labor intensive process

SNS BPM EXAMPLE

A typical example of a PC based instrument is the beam phase and position monitor designed for the SNS LINAC. The details of this device are documented in a related paper [1]. This device consists of a custom modular custom PCI card installed in a rack mount server chassis. The software functionality is implemented with a combination of commercial and custom modules.

Hardware design

The PCI based acquisition hardware is implemented in four modules, the PCI acquisition motherboard, an analog front end, a digital front end, and a clock multiplier. The modularity allows significant re-use of parts of the design on other instrumentation, as well as independent development and test.

The PCI acquisition motherboard provides basic acquisition and bus interface functions, as well as some isolated power supplies for analog functions. Eight 16 bit 80MHz acquisition FIFOs are provided, controlled by a 10 channel timing sequencer with 25nS resolution. The bus interface support includes a programmable DMA controller to efficiently transfer the contents of the acquisition FIFOs to main PC memory.

The analog front end is a commercial product designed and built by Bergoz Instrumentation [3], interfacing directly to the RF probes (4 BPM lobes). The AFE provides four channels of down conversion, a local oscillator distribution chain, and calibration. The individual probe signals are either 402.5 MHz or 805 MHz, and are down converted to a 50MHz IF.

The digital front end provides four channels of 14 bit ADCs clocked at 40MHz, and initial digital processing. The digital processing de-convolves the multiplexed quadrature stream created by under sampling the IF at 0.8 times the input stream. The resultant 8 arrays of quadrature phase data are sent to the PCI acquisition FIFOs. In addition, raw data from each ADC, or preset ramp data can be acquired for test purposes.

The clock multiplier module provides the precision clocks necessary for accurate phase measurements. The

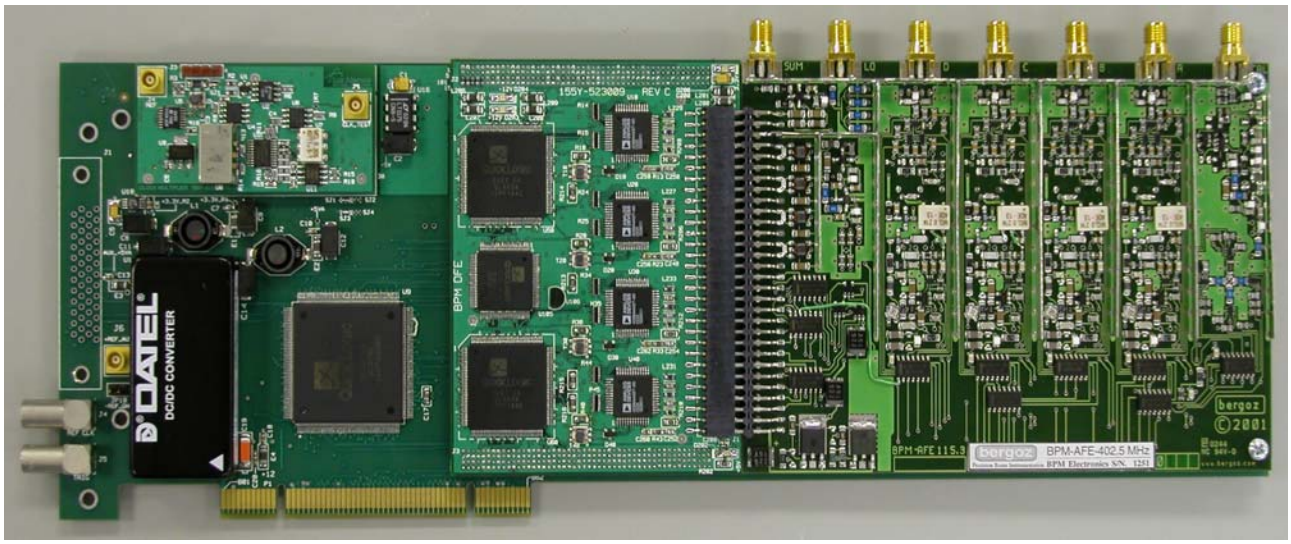


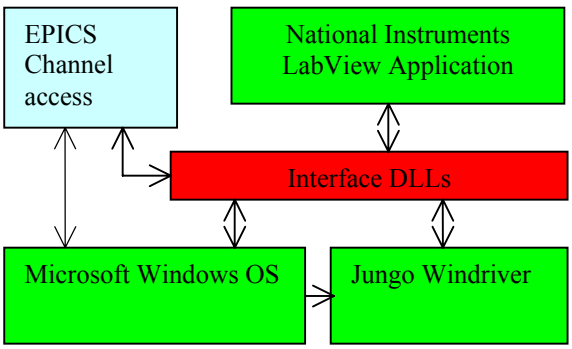
Figure 2: The BPM system PCI card showing the clock multiplier (top left), DFE (center) and AFE (right).

device currently accepts a 2.5 MHz reference and generates the required 40MHz sampling clocks. The device is serially programmed to accommodate a range of reference inputs.

Software design

The software is implemented as a combination of commercial and custom modules. The software design is presented in detail in a related paper [2]. The intent of the design was to utilize commercial products to as great a degree as possible. Figure 3 shows the software structure, which is similar to that of figure 1, with the substitution of the actual products utilized.

The OS selected was Microsoft Windows, which supports an unmatched variety of commercial tools and ADEs. Additional advantages include self hosted development and the ability to utilize many desktop systems for development and test purposes.



Color code:
Green – Commercial, Blue – Toolkit, Red – Custom

Figure 3: Software architecture

The driver and application layers were implemented with commercial products, providing the advantage of a stable, high quality development environment. The Jungo [4] Windriver provided immediate access to the custom PCI card, allowing hardware development to proceed in parallel with software development. Likewise, National Instruments [5] Labview provided a means for Instrument application development independent of other efforts.

The only custom software in the device is two interface DLLs which interfaced the commercial and toolkit modules, whose APIs were not directly compatible. The primary DLL provided an interface between Labview and the PCI acquisition hardware utilizing Windriver. In addition, it automatically substituted a hardware simulator in the absence of a PCI card. This allowed application development on virtually any PC. A second DLL was required to communicate with a toolkit based middleware product, EPICS (Experimental Physics and Industrial Control System) Channel Access. In this case, the typical use of Channel Access required the sending of very small data packets at a high rate. Since the OS provided communication mechanisms were optimized for block transfers, a custom interface provided a large performance boost.

Figure 4: SNS BPM system in 1U, rack-mount PC

Issues

The SNS BPM system shows both the strengths and challenges of packaging instrumentation in standard PCs. While the use of industry standard hardware eases development significantly, the constraints imposed by standard enclosures are easily apparent in Figure 4. The necessity to modify commercial enclosures diminishes one of the main advantages of using the PC architecture, the ability to upgrade freely without hardware or software impact.

PLATFORM EVOLUTION

As a commodity computing platform, the PC is undergoing constant evolution. New base functionality, bus standards, and form factors are appearing regularly. When using the PC as an instrumentation platform, this situation is both an advantage and a challenge, especially to system design.

Base Functionality

Aside from the never ending quest for greater general system performance, the current focus on streaming technologies is of major interest to instrumentation. While it would be nice if these developments were aimed at making the PC a better data acquisition platform, the industry is aiming at vastly improved streaming video performance. Two related technologies, isochronous I/O and hyper threading, are current efforts in this direction.

Unlike traditional computer I/O, isochronous I/O guarantees a fixed latency on a given channel. Unlike TDM (time division multiplexing) systems, modern isochronous standards such as USB or PCI express include hardware and software to implement and manage both fixed and variable latency requirements. With high performance fixed latency I/O included as part of the basic capabilities of the platform, high performance instrumentation will undoubtedly become easier to implement.

While hyper threading is Intel's term for hardware supported thread context switching, the concept of increasing the number of virtual processors is key to the

software support of isochronous I/O hardware. While it is presently unclear what exact method will be used in the future, highly efficient context switching will be required to support multiple fixed latency processing streams.

I/O busses

It should come as no surprise that new I/O standards are being developed to support new higher performance processors. In recent years, there has been much talk regarding high speed serial interconnects.

In the PC, an in-box version of Infiniband, known as PCI express, has been chosen as the next high performance interconnect technology. This standard, controlled by the PCI special interest group [6], presents an interface to software very similar to current PCI while providing support for isochronous operation and better abstraction of the physical hardware interface. Since the hardware implementation is a transaction based, high speed serial channel (after all, it's based on Infiniband), external PCI express channels are certainly feasible.

For medium performance applications, reasonable high speed serial solutions already exist in USB2 and IEEE 1394. Instrumentation systems that can tolerate data transfer rates of 10MB/s or less can easily be implemented in these technologies today. It is interesting to note that USB has always supported isochronous operation, and is specified tightly enough to have commercial generic drivers available. In addition, special bridging devices [7] are currently available supporting USB2/PCI interfacing. These devices greatly simplify the task of implementing remote PCI based instrumentation. While IEEE 1394 is more of a hardware only specification, the ability to have multiple masters outweighs the disadvantage of more complex support software in some applications.

Software Implications

It is no surprise that there are even more changes planned in software than in hardware. Most of these are far beyond the scope of this paper, only some of the most obvious enhancements required to support the new hardware described previously will be discussed.

Presently, only the most primitive support exists for isochronous I/O. With the advent of PCI express, more sophisticated OS support is currently in development (at least for windows systems). This is requiring changes in the way new drivers are written to take advantage of new functionality and maintain system performance [8].

In addition, the platform is moving from a memory map model for I/O to one based on messaging. In fact, the memory map model for PCI is virtual on most modern motherboards, with several pipeline stages for read and write operations. Frequent servicing of interrupts or spinning on a PCI address cause repeated pipeline flushing, seriously degrading overall system performance. In addition, the more sophisticated hardware operation requires significant system software support, making custom solutions less attractive. Utilizing an OS which natively supports the new I/O models is already the obvious solution.

CONCLUSIONS

The PC is an attractive platform for instrumentation development, and will become more so in the near future. The current challenges in hardware and software are being met by technological advances, making future development a more straightforward task than today.

The electrical and mechanical constraints of the standard PC chassis can be overcome for medium speed instrumentation by using USB2 or IEEE 1394 to connect to an external chassis. With the aid of bridging devices, new or existing instrumentation can be developed using PCI, tested in desktop PCs, and deployed in separate chassis. This separation provides not only electrical noise and isolation benefits, but also allows for easy upgrade and maintenance of the PC host in the field.

Effective software support of streaming data will simplify the task of manipulating real time data. In addition to base OS support, it is reasonable to expect that class drivers (or at least generic commercial drivers) will become available to support a high level API. Presently these drivers are available for devices such as disk drives, simplifying the software support for any hardware that can be designed to mimic a standard disk's operation.

As an example of how things have changed in the last two years, the SNS LINAC BPM design is being upgraded to utilize a USB2/PCI bridge, solving the current packaging difficulties while preserving the design investment. The software impact is minor – a different generic commercial driver is available (also from Jungo), which implements a nearly identical API, code changes are confined to the interface DLL. No application level changes are foreseen.

REFERENCES

[1] Beam Position Monitor Systems for the SNS LINAC, J. Power et al., Particle Accelerator Conference, 2003
[2] A Modular Interface between Custom PCI Instrumentation and Commercial Software, L.Day et al, this conference
[3] Bergoz Instrumentation, http://www.bergoz.com
[4] Jungo Inc, http://www.jungo.com
[5] National Instruments Inc, http://www.ni.com
[6] PCI SIG, http://www.pcisig.com
[7] NetChip Technology, http://www.netchip.com, reference device 2280.
[8] Designing PCI Express Architecture Isochronous Devices and Drivers, Session code TPA415, WINHEC 2003. Also session codes TPA416, TPA417, TPA418, TPA419, WINHEC 2003