

Table 1. Comparison of fixed point DSP basic parameters

Parameters	ADSP2100	TMS320C50	DSP56000
-Instruction execution	50ns	35/50ns	50ns(40MHz)
- MIPS	20	30	30
- RAM, kB	2/P,1/D	10	50
- Com. Port	2	1	1

Table 2. Comparison of floating point DSPs.

Parameters	ADSP-21060	TMS32C40/C32	C80	DSP96002	i860
-Number of μ P	1	1	4DSP+	1	1
-Instr.exec. time,ns	20	60/50	+1RISC	50	20
- MIPS	40	25/20	2BOPS	20	30
- MFLOPS	120	50/60	-	60	100
- RAM (32 bit), K	128	2	4*2K-DSP 4K-RISC	2	8
- Number of reg.	32(16)	8	-	8	30
- Com. Port	2	4/2	-	0	0
-DMA bandwidth MB/s	240	100	400	80	0
- FFT complex,ns	0.46	1.54	-	1.05	0.75
- Divide 32 bit,ns	150	360	-	350	440
- float.point,cycles	6	9	-	7	22

The Adaptation of Control System Software According to Experimental Characteristic Features at the Pulsed Thermonuclear Installation ANGARA-5.

V.I. Zaitsev, A.V. Kartashov, Yu.N. Lusin, A.I. Nebogin,
Yu.V. Papazyan, A.N. Savochkin

Troitsk Institute for Innovation and Fusion Research, Russia.

Abstract.

High pulsed power installations such as Angara-5 are used for different kinds of experiments, e.g. nuclear fusion, high current Z-pinch and X-ray generation. These various experiments have differences including timing, measurement circuits, sensor types and other parameters of the measurement and control system (MCS). Software of such a system must be flexible enough to be quickly tuned for all kinds of experiments, and special software satisfying this requirement has been developed.

All the MCS parameters are stored in appropriate database files (DBF). A special utility similar to usual database management programs provides presentation and editing of these parameters. The interaction between DBF and local system software is performed by means of a special dispatcher program. Such a set of tools allows one to reconfigure the system immediately before the experiment. Both MCS parameters and data obtained are stored in the main DBF and can be used during data processing. The present paper describes the software structure and configuration tools.

1. Introduction.

The features of the Angara-5 measuring and control systems were discussed at ICALEPCS'91 [1]. The program of the installation includes different types of experiments. Each type usually consists of a few tens of real cycles (shots), of which the installation construction allows several per day. In the course of each experiment, and especially when a new experiment is being prepared there is a need to correct the measuring environment. Examples are to change the set of detectors and MCS equipment modifications. In connection with any hardware change it is necessary to correct MCS software as well, and it is desirable to produce this correction without a delay in the experimental program.

The Angara-5 MCS software structure and special tools that allow fast and easy modification of the measurement environment according to the experimental requirements are presented in this paper.

2. MCS information structure.

Information streams in the MCS structure are shown in Fig. 1. Signals from sensors and detectors are transmitted in analog form by coaxial cables to local systems where analog-to-digital conversion is performed. There are several local systems, each dealing with a certain set of signals. Data are sent by means of a computer network to a supervisor computer for later processing, presentation and storage in a database. The main database is accessible for several applications, chiefly ones doing mathematical and statistical analysis and process modeling. A computer network gives one the possibility of reading data on his personal workstation. There is also a reverse information stream from the supervisor to control the local systems and the installation as a whole.

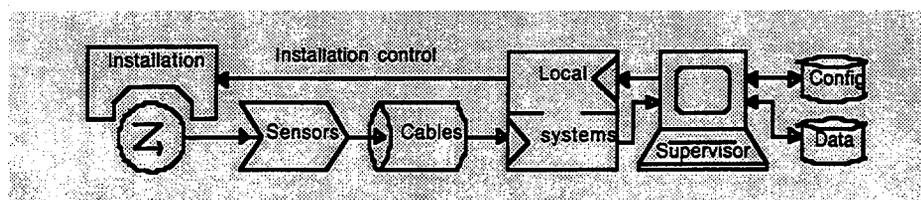


Fig. 1. Information streams

There are three general stages determining the information characteristics - detectors, cables and local systems. There are a few hundred distinct detectors in the Angara-5 installation. Each is described by its own purpose, sensitivity and frequency band. Signals from detectors are transmitted by coaxial cables to data acquisition systems which are located in a special shielded enclosure. The cable length may be up to 80m, and because a general process of the installation has typical time 10^{-8} seconds it is necessary to take account of time delay and high

frequency signal damping during cable passage. For this purpose each cable line has two parameters - time/length and bandpass. These parameters are sufficient for reconstruction of signals [2].

MCS measurement equipment includes a digital waveform recorder, time and amplitude measurement devices etc. Each type of device is described by means of a parameter set, i.e. sensitivity, time and amplitude coefficients, digitizer least count etc. Angara-5 MCS uses about 200 gauges of different types, each one having a specific parameter set.

The full description of the measurement equipment used in a particular experiment, combined with data from the installation, is stored in the main database for use in data processing. However it is a difficult job to make up such a description for each experiment type. Moreover there are often changes in experiments and no time to carefully correct this description. For this purpose a special tool, the measuring environment description program (DSR) was designed. DSR combines all gauge characteristics in a special database and allows one to check and correct the parameters just before the shot.

3. MCS description database management.

Fig. 2 illustrates schematically the structure of the measurement environment database and its usage during an experiment. All parameters of detectors, cables and gauges are contained in corresponding sections of the database. Also stored are typical signal parameters - amplitude, waveform, etc., for the information of the operators.

The DSR program is a database manager which performs the following:

- enter and to edit the MCS component parameters;
- create the description file (experiment scheme) of a certain experiment's measurement environment;
- output the experiment scheme as a document.

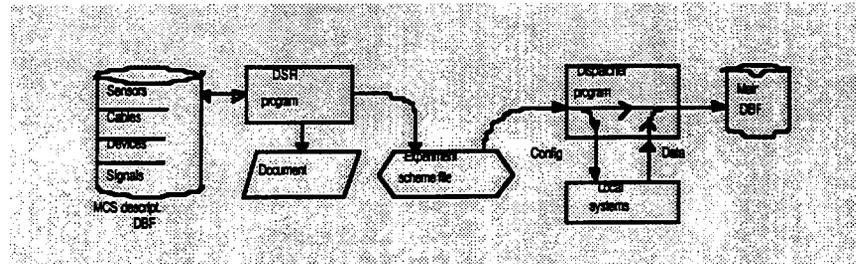


Fig. 2. MCS overview

The component parameters are entered by means of the special tables and sorted in accordance with their functions and locations inside the system. A program can indicate a signal name for reference if the component is used in the current configuration; thus it is possible to view signal parameters at once. Such tables were prepared for each component type and are in spreadsheet form.

The experiment scheme composition is a most interesting and critical part of DSR. Fig. 3 illustrates the process, where the main point is to bind the signal to measurements and MCS components in the chain. The concept is realized made by means of a four column table, as indicated in the figure, which contains names and codes of signals used in the current experimental session. The DSR automatically takes all component parameters from the database and saves them in a special file, called the 'experiment scheme.' This file contains a full description of the measurement environment and is used in the main control program (dispatcher) at experiment time. There are several advantages to such a procedure:

- all the MCS component parameters are located in the same record which one can easily edit if required;
- an experiment scheme has no surplus information;
- MCS preparation for a new experiment type does not require much time; moreover some changes can be made just before a shot;
- the main database record contains the minimum of information, including only experimental data and the description of gauges used.

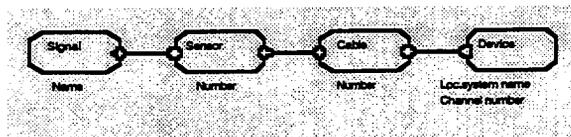


Fig. 3 Experiment scheme composition

The DSR program creates the experiment scheme document which reflects the current MCS state. The parameters of all detectors, cables and devices are contained in additional forms also provided by DSR.

The main control program on the supervisor computer uses the computer network Ethernet (Decnet-2.01 software) for distributed system control according to the appropriate experiment scheme. The user protocol is designed for interaction between the supervisor and local systems. Its main functions are:

- to control the communication channel (open, close and obtain status);
- to send the experiment scheme to the local systems;
- to send the current shot number and prepare all systems;
- to acquire data from local systems.

The local system programs receive the experiment scheme and set devices to their required states. After a shot one performs data acquisition, storage and transfer to the supervisor, where the data are stored in the main database with the measurement environment description (experiment scheme file).

4. Conclusion.

The experience gained from DSR usage has shown the following:

- the time of MCS preparation for an experiment is reduced considerably;
- there were no errors caused by operator mistakes in parameter entry;
- the DSR software, together with utilizing adjustable configurations of local systems [3], provides flexibility and convenience during the control of an experimental session.

The software structure presented here allows one to include additional local measurement systems into Angara-5 MCS with minimum effort.

Acknowledgments.

The authors would like to thank Dr. G.M. Oleinik and Dr. V.O. Mishensky for fruitful discussion.

References.

- [1] V.V.Bulan et al., Proc. Int. Conf. Accelerator and Large Experimental Physics Control Systems, KEK, Tsukuba, Japan, 1991, KEK Proc., p.p. 235-238.
- [2] V.I.Zaitsev, A.V.Kartashov, J.M. Oleinik., JET, v.36, No.3, 1993, p.p. 395-401.
- [3] V.I.Zaitsev, Yu.V. Papazyan, V.N. Savochkin., Nuclear Instrument and Methods in Physics Research, A 352, 1994, North Holland, p.p. 418-420.

Microcontrollers applications for IHEP accelerator control.

Yu. Bardik, E. Kallistratov, A. Makhnachev,
A. Matiouchine, G. Obukhov

1. Introduction

About 14 years ago Intel introduced the MCS 51 family of 8-bit microcontrollers. They are still on the market, keeping compatibility with their successors, which have expanded in functionality and improved in speed. This family is suitable for use in the most popular embedded control applications. The last members of the MCS 51 family provide very high performance and superior code density.

Dallas Semiconductor designed a few 80C32-compatible chips based on the Dallas High Speed core. As a result, every 8051 instruction is executed between 1.5 and 3 times faster than the original one for the same clock rate. The DS80C520 is the fastest of them. It offers a maximum clock rate of 33 MHz, yielding a single-cycle instruction execution time of 121 ns (8.25 MIPs).

As was mentioned above, the 8-bit 8051 micro is now 14 years old, and characteristic of many teenagers, is expected to experience a quick surge in growth. Two 16-bit derivatives soon will be widely available. The 3-stage CPU architecture of the new MCS 251 microcontrollers (Intel) boosts 8-bit performance five times by just swapping chips and up to 15 times by incorporating new instructions and addressing modes. Philips Semiconductors has announced the sampling of the XA-G3 microcontroller, the first member of Philips' extended architecture family. The G3 is a powerful, general-purpose microcontroller that is code compatible with the 80C51. With a 30 MHz clock, the XA-G3 has a typical instruction execution time of 100 ns.

It is very important that all mentioned microcontrollers are fully code- and pin- compatible with the MCS 51 microcontroller family, so one can increase significantly performance without reworking the old design. On the other hand the speed and efficiency of these micros allow easy handling of a broad range of highly-demanding embedded requirements in the control, timing and computation functions.

2. Field-bus controller VBC

The field-bus controller VBC designed in IHEP is a typical example of latest generation microcontroller applications for accelerator control. The VBC supports the MIL-1553-STD-B communication, which has been extended in CERN to meet the specific requirements of accelerator control. The VBC is a one-board VME module based on the 8-bit microcontroller DS80C320 from Dallas Semiconductor. The VBC simulates all functions of the NBC [1] and may replace the latter without any hardware or software adjustments.

The DS80C320 micro is pin-compatible with the standard 80C32 and offers the same timer/counters, serial port, and I/O ports. The DS80C320 provides several extras in addition to greater speed. These include a second full hardware port, seven additional interrupts, programmable watchdog timer, dual data pointers, power-fail interrupt and reset. The DS80C320 is extremely familiar to 8051 users but with a maximum clock rate of 25 Mhz provides the speed of a 16-bit processor.

The VBC micro makes a link between the 1553 bus protocol and data exchange with the VME CPU. These data pass through a 16 Kb dual port memory (DPM) and are synchronized with an interrupt to the host computer. A control and status register (CSR), accessed either from the host CPU or the microcontroller, allows to identify each interrupt. The DS80C320 performs three real-time tasks: receive/transmit, slow polling and fast polling [2]. It allows the avoidance of wasting host CPU time. The fast serial communication (1Mb/s) is performed under full control of the microcontroller.

Although the NBC is based on a conventional 16-bit microprocessor, the performance measurements show similar timing responses for the VBC and NBC, which are mostly defined by the serial protocol itself. Nevertheless this experience has shown that high grade 8-bit micros may force out 16-bit microprocessors today in many control applications, providing simplicity of design, low cost and high performance.

The high speed microcontrollers can replace conventional logic-based state machines. This idea has been implemented in the development of different I/O modules for U-70 controls.

3. General structure of microcontroller-based modules

The general structure of microcontroller-based VME modules consists of a DPM, a CSR, an MCS 51 family microcontroller and application dependent peripherals. All modules provide some logical compatibility; that means a uniform scheme of register field and DPM allocation within a module's address space, plus a mandatory defined format for the device control and status register (CSR).

Each module allocates the 32 Kb of VME address space that are shared between the module's register field (the lowest 256 bytes) and the DPM. The dual port memory is divided into 4 areas: the Command Area, the Reading Buffer, the Polling Area and the Error Buffer.

The Command Area is used by the host to write a sequence of actions (commands) followed by data. Tables of vectors for function generators, scan-lists for multichannel ADCs and settings for timers are placed in this area.

The Reading Buffer contains some data captured by the microcontroller during command execution and prepared for the host. They are the results of analog to digital conversions, contents of counters or any data requested by the host.

The Polling Area is a special command area that is filled by the host if a microcontroller performs periodic input signals polling and then compares results with some predefined conditions placed in the same area. The polling is a typical control task for modules having a number of analog or digital input channels.

The Error Buffer is used by a microcontroller to report about any faults that occur during command area execution.

These areas have different sizes in different I/O modules but accesses to each of these areas are performed under software control using the same CSR flags.

4. Module register assignment

Within the module's registers, the lowest 8 are reserved for functions that apply to the whole module. These are: the Interrupt Vector Register (IVR), the Control and Status Register (CSR), the Module Reset Register (MRR) and the Module Identification Register (MIR).

The 16-bit IVR allows the assignment of the module interrupt to one of seven interrupt levels and to program the vector number, supplied with the interrupt acknowledge VME cycle. This register could only be accessed from the VME side in read or write mode.

A host writing action in the MRR results in a complete reset of the module including the microcontroller.

The 16-bit read-only MIR contains very useful information for the host: crate number(C), module number(M), and identification number(I) -- the unique bit pattern assigned to each type of modules.

The CSR is a dual port register accessed either from the VME bus or the internal microcontroller bus. It is used to synchronize the data exchange through the DPM between host and microcontroller. The content of the CSR reflects the DPM organization. Setting POLL by the host CPU starts the polling action. Detecting any changes by microcontroller activates the CHG flag together with INT. Setting GO by the host causes the microcontroller to execute the sequence of actions written in the Command Area. The ERR flag shows that the Error Buffer is filled with information concerned the last detected fault. The active DFH (Data for Host) flag means that data in the Reading Buffer is ready for the host.

The host sets two additional flags INE (Interrupt Enable) and TST (Test), enabling a VME interrupt or activating an internal test. The microcontroller status is shown by HALT and module status by RDY.

Other bits of the CSR are reserved for future assignment.

5. Technical parameters of the microcontroller based I/O modules

The logical compatibility of I/O modules leads to the standard hardware implementation of the essential part of the module, including VME interface, DPM, CSR and microcontroller with support. This provides the facility whereby I/O functional modules covering a different range of tasks may be largely developed around a common structure, with the resulting benefits in development time, documentation, training overheads, etc.

The set of VME microcontroller-based I/O modules includes the scanning ADC (VSA), the two-channel function-generator (VFG) and the timing generator (VTG). In the following are some technical parameters of these modules emphasizing the microcontroller application.

The VSA is the 12-bit, +/- 10V, 16-channel differential/32 single-ended ADC with 24 Kb of data store buffer. Everything from the selection of sequences of channel sampling, sampling rate and operating mode, to the size of data stored are software programmable. The high speed microcontroller DS80C320 allows a maximum sampling rate of 5µs/channel.

The VFG is a 16-bit, +/- 10V, two-channel function-generator intended for power supply control. The 8-bit DS80C320 provides a maximum rate of 50 µs/point, as has been requested for this application. The microcontroller spends most of this time computing the next point. An estimation shows that a compatible microcontroller with 16-bit internal structure, such as Intel's 80C251SB, can significantly improve the maximum rate up to 10 µs/point using the same algorithm.

The VTG is a timing counter board providing the user with 8 general purpose, 16-bit timer/counters with a maximum rate of 4 MHz and 16 channels with 1 ms resolution and 1 µs accuracy. Each counter is fully programmable from a user-defined start value up to that selected from one of the range of sources. The contents of each counter may be read and stored at any time without disturbing the counting process. The board can also work as an 8-channel pulse-width modulator (PWM) or in a combination of PWM and timing channels. The 8-bit DS80C320 simulates the 16 slow channels in software, simplifies the setting of the fast channels and controls the PWM outputs. This module may generate, in one of the applications, a setting value (PWM output) and two timing pulses for 8 power supplies.

Conclusion

The microcontrollers of the last generation can be successfully used for the development of high performance VME modules as has been shown here. The present approach gives to designers some advantages, the most important of which are:

1. Much of the hardware development portion of the project may be reduced to the design only of function-dependent parts. This obviously results in big savings of development time and money and reduced technical risk.
2. More of the development budget is applied to product-specific software development.
3. The same basic product design could be fitted with different types of compatible microcontrollers allowing faster product operation with similar features or enhancing features while maintaining acceptable performance.

Acknowledgments

We would like to thank our colleagues from CERN D.Francart, C.Pignard, G.Surback and O. Van der Vossen for a sharing their knowledge about NBC and testing the VBC module, and our colleagues from IHEP V.Komarov and A.Sytin for the constant support of this project.

References

- [1] D.Francart, C.Pignard, G.Surback, NBC, CERN SL Note 92-54 (CO),
- [2] C.Pignard, G.Surback, NBC, CERN SL Note 93-92 (CO).

DISTRIBUTED SIGNAL MULTIPLEXOR at FERMILAB

Charlie Briegel, Dean Still
Fermilab
Accelerator Division Controls Department
MS 347
P.O. Box 500
Batavia, IL. 60510 USA

A set of 1 GHz multiplexors is distributed at remote locations which enable signals to be routed to a variety of instrumentation equipment. The commercial multiplexor provides up to 20 1X4 switches per location and can be controlled via GPIB.

Control of the multiplexors utilizes MOOC (Minimal Object Oriented Communications) which provides the skeleton software for communication to an application program. The control of a route across distributed multiplexors can be effected by turning ON a device. The route has various mechanisms for reserving or usurping the ownership of the multiplexors which make up the signal's route. The application and front-end software implementation is described.

The hardware consists of a VME crate, a 68040 processor running vxWorks OS, and GPIB interfaces including transparent support of a remote GPIB connected via Ethernet to the front-end computer.

1. OVERVIEW

A reoccurring goal in the control room is to measure a signal on the appropriate scope for a test. A distribution panel with a maze of wires is commonly used to acquire this signal. The cable is manually connected to the scope from the source which may span multiple distribution panels from remote locations. While this method works for stable connections or short-term tests, it does not adequately address today's dynamic needs for acquiring various signals to a shared set of instrumentation. In this case, depending on the sequence of events, a different set of signals need to be piped to a set of instruments as the sequence evolves and then data is acquired. A reliable and efficient method for distributing these signals is required.

A commercial multiplexor was purchased which provided a variety of options. Fermilab purchased only one type of multiplexor card; a 1 GHz 1X4 multiplexor with 4 sets of multiplexors per card, 5 card slots per box, a total of 20 1X4 1 GHz multiplexors per box. The control system interface is via a GPIB connection and utilized existing software and hardware infrastructure to manipulate the switches. Four boxes were acquired; two were located in the Main Control Room (MCR), one was located in Booster in the RF area, and one was located in the Main Ring in the RF area.

Minimal Object Oriented Communications (MOOC) is the underlying software used to implement support for GPIB communications. MOOC hides the layers of software required for communications and invokes methods to fulfill data acquisition requests and settings from the control system.

2. MOOC

This infrastructure enables classes to be individually loaded and initialized so a system can be individually tailored. Although the classes start independently, the GPIB class is inherited, so backward references will be made to this base class. Classes which inherit the GPIB class can provide hidden functionality to the user, or derived calculations from GPIB devices. MOOC and the pre-existing GPIB class minimizes the effort needed to create the MULTIPLEXOR class and enables the user to concentrate on the problems of interfacing to a particular GPIB device. The following is a portion of the startup script for the vxWorks node.

```

shellPromptSet ("GPiB->")
# a bunch of loads were deleted for this document...
# specify the controller type per 0-31 gpib address range
# and IP host for ENET boards
GPiB_obj("NI_1014", "")
GPiB_obj("ENET_GPiB", "host_name")
#Create the gpib class
GPiB_obj_init()
# specify the gpib address of the MUXes
MULTIPLEXOR_obj(6)
MULTIPLEXOR_obj(7)
MULTIPLEXOR_obj(44)
#Create the multiplexor class
MULTIPLEXOR_obj_init()
# Slam: wake up VAX task for download of MUX settings
MiscBoot()

```

3. GPiB Class

A class was written to support up to eight separate GPiB controllers from a single VME processor. The supported controllers reside on the VME bus, an Industry Pack (IP) interface, or Ethernet through an Ethernet to GPiB interface. The Ethernet to GPiB controller enables distributed connectivity from a central location via a TCP/IP connection. This connection is treated as another GPiB bus controller and the network connection is maintained inside the driver interface.

The GPiB class supports transparent access to the device by enabling the console user to send and receive strings of data which is normally ASCII text. The class provides semaphore protection so only one user can access a GPiB controller at a given time. The semaphore provides hardware encapsulation of a single command. The sequence of commands to the device are not protected by the class and no logical ownership of the device is provided. When protection is required, the application is set to single-user execution.

The GPiB class provides control so an application can reset the controller, clear the GPiB device, trigger the device, set the REN line for the GPiB bus, or put the device in LOCAL. Digital status is returned to reflect the serial poll of the device.

4. MUTIPLEXOR Class

The multiplexor class effectively inherits the GPiB class and adds functionality. While the GPiB class is still available to the user, this new class and its corresponding devices provide a higher level of functionality. The user is unaware of the GPiB commands required for reading, setting, control, or status of the device.

There are three types of devices associated with this class.

1. The 1X4 multiplexor contains 4 devices which can be closed (ON) or opened((OFF). The multiplexor provides BBM (break before make), so closing a device will automatically open all the other three devices associated with the multiplexor. The status will return the state of the device. Currently, these devices can be manipulated without any constraints. All changes to the device are saved internal to the box and are recalled at initialization time of the class. This device type is intended for diagnostics.

2. The status of all the switches can be read by a single device which uses a nibble to represent the status of the four switches. So five 16-bit words are returned, one for each card in the box. This device cannot be set and is intended for status display of the box. The device can be reset, which places all the switches into a previously saved state.

3. The set of multiplexors can be manipulated by setting a device to ON. This device has a data structure associated with it which specifies a route for a signal to get to an instrument. The route can be read or set as an array of four byte structures. There can be up to 512 such devices, consisting of up to 20 multiplexors per route specification. Further, the device can return the status and ownership of each multiplexor specified in the route in an array of structures.

When a device is turned ON, the route is searched to determine that no other device of this type is currently using any of the multiplexors. If all the multiplexors are available, then the path is created by sequentially closing the multiplexors which make up the route. At the same time, the user can request ownership of the device for 5 minutes, 15 minutes, 1 hour, 8 hours, forever, or go to war with other users (LIFO). If the device is not available, the application can raise its priority by sending a request to turn ON the device as a super-user and over-ride any previous ownership. When the device is turned OFF, the path remains closed until someone requests one of the multiplexors. The OFF command simply implies the multiplexors in this device's route are available. A reset to the device opens all multiplexors along the route.

This class provides ownership and easy manipulation of a set of multiplexors. A software program such as a sequencer can simply turn ON this type of device, collect the data for the signal on the instrument, then move on to the next measurement.

5. USER INTERFACE

A console user interface layer provides access to the GPIB class device by specifying only the node and address, where the address specifies both the controller (0-7) and the GPIB device address (0-30). The console user interface finds the corresponding device name and utilizes generic communication mechanisms to manipulate the device. The user needs to know where the device is located. Existing applications for GPIB devices allow commands to pass to the device without any programming and can easily be adapted to provide specific support for a device.

The MULTIPLEXOR class supports generic data acquisition services in the Fermilab control system. This enables these devices to appear as generic parameters for pre-existing applications. The underlying GPIB commands to manipulate the multiplexor are hidden from the user.

A specific application was written to specify and display the multiplexors associated with the route for a given device. The device name for each multiplexor was used to specify the route. When the device's route is set, the information is automatically forwarded by MOOC to the central database. When the processor is booted, the data base is downloaded automatically. The following shows the application displaying a route specification.

```

24 MCR PACAL-DANA Mux Controller 29-OCT-95 17:51:18 Pgm_T
Select MUX Files Commands
-----
Mux Route Control
  ◆ Signal Route Directory
  ◆ Create Signal Route
  ◆ Display Signal Route
  ◆ Close Signal Route
  ◆ Open Signal Route
Individual Mux Switch Control
Mux switch device [
Switch Status:
ROUTE--- - loss switch
X:SIGSCP p pen switch
X:PATH01
X:PATH02
X:PATH03
X:PATH04
X:PATH05
X:PATH06
X:PATH07 n
+
-----
Messages
Program initialization complete.

```

6. COMMENTS

The implementation currently consists of approximately 40 signals and approximately 12 scopes. The Main Ring RF box is not connected to the control room, due to a limited number of wires between RF and MCR. The system should expand when more operational experience is acquired.

The physical routing of signals to scopes and maintaining these paths is not an easy task. The maze of wires is not significantly reduced and the wires must still be connected appropriately to the multiplexors. This multiplexor does not provide any testing mechanism to understand the connectivity. The visualization of the connections will be enhanced by using a graphical interface for documenting the routes.

The Ethernet to GPIB controller enables distributed boxes to be centrally controlled. This is an essential part in providing ownership and enabling the routes to be accurately controlled.

The protection of the system could be improved. The multiplexors can be manipulated directly without any protection through the GPIB class or the first device type of the MULTIPLEXOR class. The intended method to manipulate the multiplexors is through the last device type of the MULTIPLEXOR class which is protected. If necessary, all other mechanisms could be blocked or provide feedback to the existing protection mechanism.

There is no mechanism to prevent an inappropriate signal (i.e. high voltage) to be routed to an instrument. The ease by which the signal can be routed increases the possibility of causing damage to an instrument. A digital volt meter (DVM) accessible within the multiplexor could characterize the signal and perhaps prevent inappropriate connections. It might be interesting to use the multiplexor to multiplex a DVM to test signals before they are switched onto equipment.

References

- [1] RACAL-DANA Series 1250 Universal Switch Controller, May 1993, RACAL-DANA Publication No. 980609.
- [2] ESP-488 Software Reference Manual for the GPIB-ENET, Feb., 1995, National Instruments Part Number 320910A-01.
- [3] User Manual IP-488, Green Spring Manual Revision ii.
- [4] Boris Lublinsky, MOOC, Fermilab internal document.
- [4] Jeff Utterback, Making Data Accessible to ACNET from a VxWorks Front End, Fermilab internal document.
- [5] B. Lublinsky, Nuclear Instruments & Methods in Physics Research A 352 (1994) 403-406 North-Holland.

Application of a TMS320C31 chip for DSP/Embedded System

Shuchen Kate Feng, Rolf Olsen, Phil Pietraski, and D. Peter Siddon

National Synchrotron Light Source, Brookhaven National Laboratory

Upton, NY 11973, USA

The TMS320C31 chip from Texas Instruments is ideal for DSP and Embedded System Applications. Its powerful instruction set, high speed and innovative architecture enhances the performance of DSP applications. Its JTAG IEEE 1149.1 emulation port and microcomputer/boot-loader function facilitate computer simulation, debugging and embedded-system development. To date, the speed of this low-cost chip ranges from 27MHz to 50MHz.

At NSLS, we have incorporated a TMS320C31 chip and an SN74ACT8990 test bus-controller with an open-target-interfaced emulation porting kit for VME-bus-hosted emulation and AT-bus-hosted emulation. Using the AT-bus-hosted emulator for development, we have designed and implemented a fifteen-channel-counting device for NSLS beam line data acquisition. The implemented device is a stand-alone counter which can be controlled via front panel keypads with display or by a host computer over an IEEE-488 interface. Some technical hardware and software details in designing the emulation as well as the counter will be described. Other commercial scan-path emulators for the TMS320C31 will be listed as alternatives which could be considered depending on a project time frame, financial plan and design needs.

1. Introduction

At the beginning of this project, we decided to update the processor used for the embedded systems developed by the NSLS Beam Line Group from a 2MHZ 6502 microprocessor we used in our in-house-designed motor controller [1] to a faster microprocessor. Making good use of the existing resources in NSLS was a sensible approach to the upgrade. NSLS had a site-wide-licensed emulation porting software kit [2] for the Texas Instruments TMS320 DSP family. The kit emulated the TMS320-based device through the JTAG IEEE 1149.1 emulation port in conjunction with an SN74ACT8990 test bus controller. The kit contains source codes, which allows users to apply emulation technology (i.e. scan and debug) directly to their product in diverse hardware and software platforms. In our application, we designed an AT-bus-hosted emulation board installed on a PC to interface the AT-bus with the JTAG bus. As to the emulation software, we compiled the emulation porting software on a 486 PC MS-DOS platform. We adapted the TMS320C31 chip to utilize its microprocessor mode during development for on-line emulation and its boot-loader mode for a stand-alone, embedded device. Using that emulation scheme, we developed a stand-alone, fifteen-channel-counting device for the NSLS beam line data acquisition.

2. The emulation hardware and software architecture

The emulation porting kit written in C language can be compiled easily under MS-DOS/Windows, OS2, SUN-OS, SUN-solaris, HP -UX, etc, by changing the compiling option in a makefile. The emulation hardware platform can be based on an AT bus, SBus, VME bus, and so on. A few definitions in header files can be tailored easily to fit different emulation hardware platforms. The TMS320C31 chip allows complete emulation via a serial scan path to the SN74ACT8990 chip through a 12-pin header. One needs to design interface hardware to communicate between the emulation hardware platform and the JTAG IEEE 1149.1 emulation port using a 44-pin surface-mounted SN74ACT8990 test bus controller[2]. The block diagram in Fig. 1 depicts the design concept of our TMS320C31 emulation hardware hosted on an AT-bus.

This in-house-designed emulation system is inexpensive and versatile, particularly in the situation when developers need to implement the application on diverse emulation hardware and software platform. However, there are other commercially-built emulators for the TMS320C31 on certain emulation hardware platforms. Texas Instruments markets DSP development hardware and software for PC and SCSI-hosted workstations. White Mountain DSP [3] markets emulators for PC and SBus-hosted Sun workstations. Those commercially built emulators provide convenience and time-saving for development on a fixed platform.

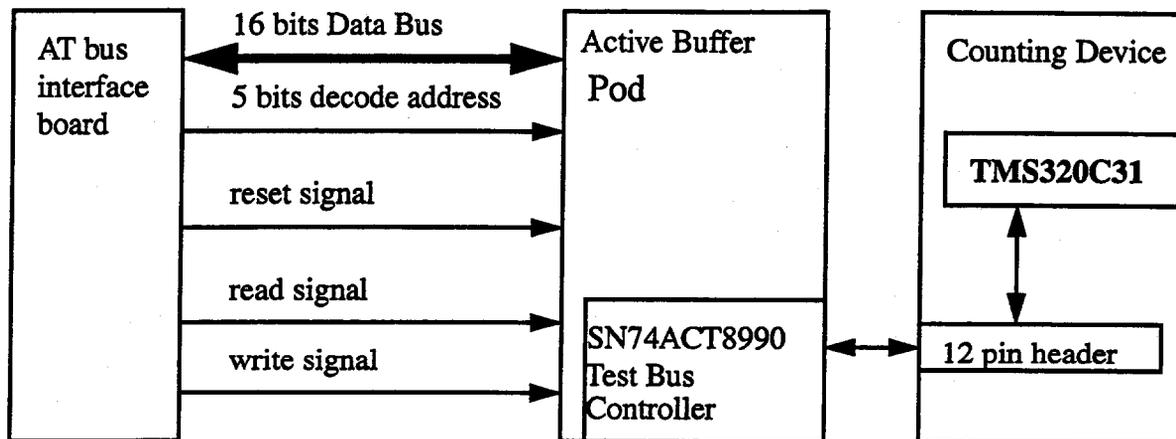


Figure 1: AT-bus-hosted emulation hardware for a TMS320C31 based device

All the emulation systems, custom-built or commercially-built, have similar graphical-user interfaces that help users to develop, test and refine C programs and assembly language programs. It provides effective and flexible functions such as breakpoints, single-step, user halt and memory display.

3. Features of the developed counting device

The developed fifteen-channel-counting device can be configured as one group of fifteen-channel counters, or three independent groups of five-channel counters. Each group can count a specified amount of time based on the internal clocks, or a specified amount of external events from any channel in the same group. The counters can be controlled via a front panel keyboard with display, or by a host computer over an IEEE-488 interface. The following is a list of the additional features:

- Flexible choice of a preset channel in any group
- Clocking time based in milliseconds from 20 to 1000000 (e.g. 10^6), or in seconds (default) from 1 to 10000000 (e.g. 10^7)
- Maximum external event counts is 0xffffffff (32 bits resolution $\sim 4 \times 10^9$)
- Readout of counts during counting
- GPIB data transfer interrupt rate is capable of up to 1.2 MHz
- Overflow flag displayed on a front panel and readable through GPIB
- Maximum counting frequency for input is 4MHz

4. The design of the developed counting device

Combining the advantages of hardware, firmware, and software designs, a GPIB data transfer interrupt rate for the counting device of up to 1.2 MHz has been achieved. Figure 2 shows the prototype unit now in use on beam line X16B at the NSLS.

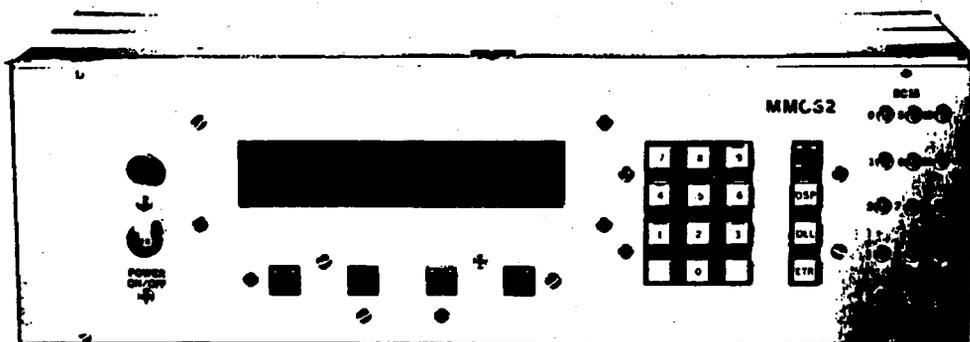


Figure 2: The prototype counter now in use on beam line X16B at NSLS

4.1 Hardware Design

We chose the 33MHz TMS320C31 chip as the system microprocessor not only because of its low-cost but also because of its performance. It is an ideal chip for DSP applications as well as general-purpose microprocessor-based applications. Its powerful instruction sets reduce the software overhead drastically by a factor of fifty as compared with the previous 2MHz microprocessor we used.

We use six surface-mounted AM9513A System Timing Controllers [4] to master the fifteen-channel counting. In addition, the system has several programmable logic devices[5] to program the counting synchronization, timing, overflow latching, device decoding and a wait state mechanism. The propagation delays of the programmable logic devices are all under 35 nanoseconds. Using PLDs with direct I/O, propagation delays are more favorable in this high-speed application. The wait-state mechanism was designed to handle slow timing peripherals such as the display unit and keypads which are used for local display and control.

A surface-mounted National Instrument GPIB-TNT4882 chip [6] on the board communicates between the counting device and a host computer via a GPIB connection. Its 8-bit 16-deep FIFO buffers data between the GPIB and the CPU, which increases the GPIB data transfer throughput.

All the hardware components are mounted on a 6.35 inch x 9.25 inch, six-layered board. The ground and 5-volt signal each occupy an individual plane. Figure 3 shows the hardware structure of the counter board.

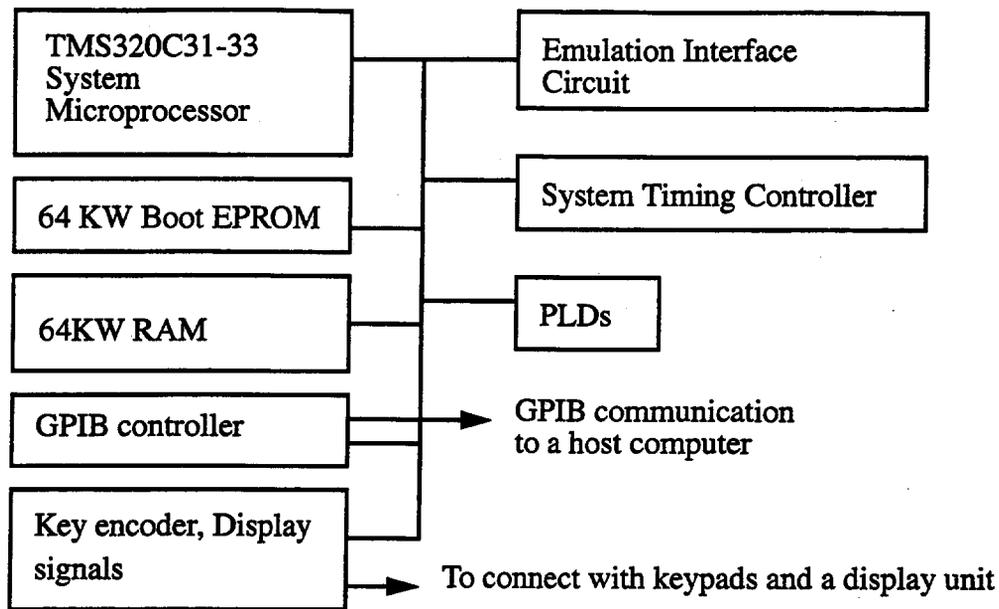


Figure 3: Hardware structure of the counter board

4.2 Firmware Design

There are four 64Kx8 27512 EPROMs and one 64Kx32 RAM module[7] with 20ns of access time. During the emulation/development stage the software is downloaded to the fast RAM directly from the PC host for on-line simulation and debugging. After development, the finished code and a bootloader were programmed into the inexpensive, slow, erasable EPROM. On power-up, the TMS320C31 downloads the codes resident on the EPROM to the fast RAM for real-time execution. The boot loading time for a 64K 32-bit word code is only 60 milliseconds.

4.3 Software Design

SPOX [8] is a real-time DSP operating system for TMS320 DSP chips. We decided not to use the SPOX OS because of its large memory requirements (43K 32-bit words) and the possibility of increased software overhead from features we did not need. Rather, a primitive microkernel is sufficient for our counter application. We wrote our own real-time kernel in assembly language in the time-critical lower layers (e.g. I/O drivers and interrupt handlers) to maximize efficiency. The use of C for the uppermost layers facilitated sophisticated function development without significant software overhead. The C crosscompiler from Texas Instruments includes mathematical functions, type-conversion functions and other general utilities in its library. However, it did not contain functions for stream I/O interfaces (e.g. scanf, and printf). One could get the source code from public domain sites or purchase the C compiler for TMS320C31 from Tartan [9]. Including kernels, stack pointers and dynamic variables, the complete codes for the counting device take 12.5K 32-bit words, which leaves plenty of room in the RAM for further development. Figure 4 shows the software development flow.

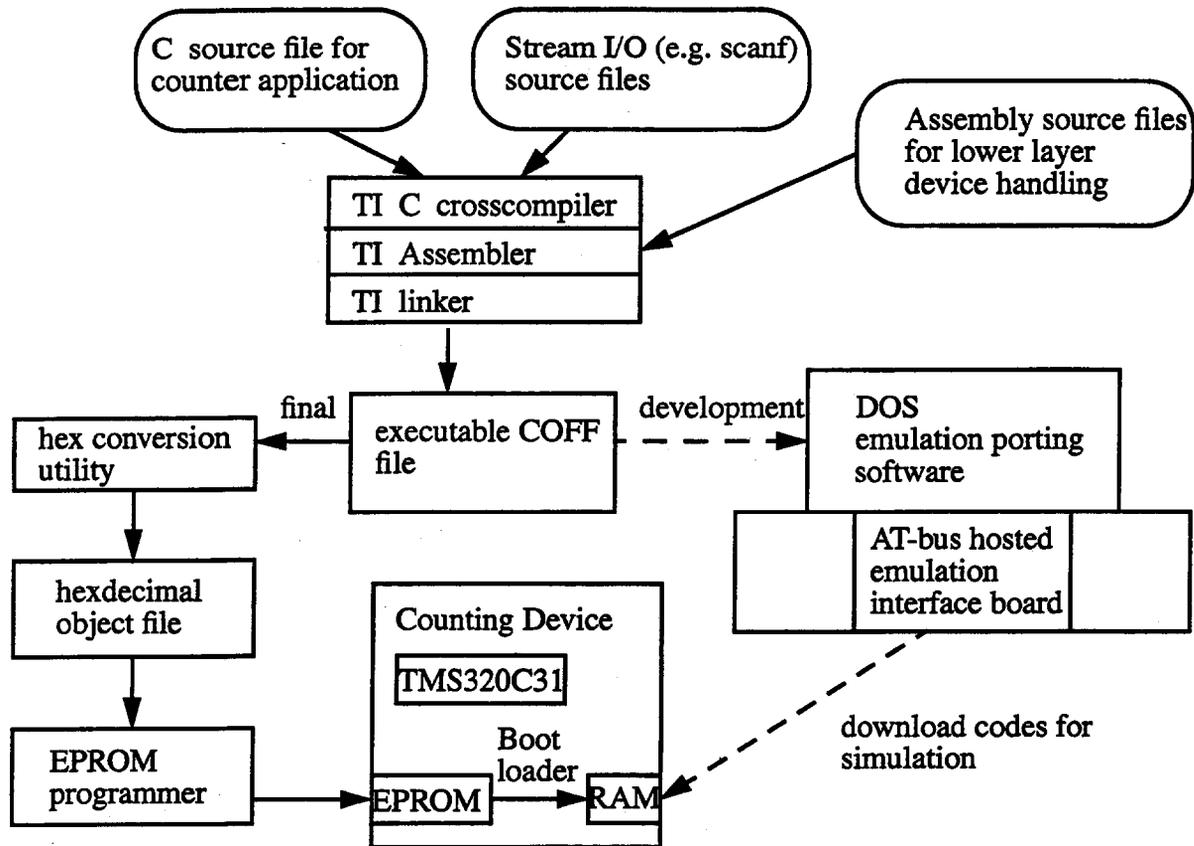


Figure 4: Software development flow of the counting device

5. Discussion

5.1 Emulation Scheme

The current emulation system has a restriction of 27K 32-bit words of memory for emulation due to the 640K memory limitation of DOS. If one uses Microsoft Windows, the amount of memory one can configure in the memory map corresponds directly to the amount of memory in the PC. However, one can easily migrate the AT-bus-hosted emulation board from a PC to an HP workstation equipped with an EISA slot. As soon as one recompiles the emulation porting kit under HP-UX, the application software for the counting device can be ported to HP-UX without many changes because of the portability of C. To do so, one needs to get a C crosscompiler for TMS320C31 running under HP-UX from TI. Migrating the emulation system to an HP workstation not only overcomes memory restrictions but also allows multi-tasking, which facilitates development and testing. The use of a NSLS PC-based UNIX system (e.g. LINUX [10]) would require TI to port their crosscompiler, which so far has not been done.

5.2 Expansion of the counting device

One can expand the counting device by adding our previously in-house-designed motor controller boards into the same chassis. An optimal solution is to design additional interface circuits to the motor boards on the counter board without modifying the motor boards. That will result in a device with a maximum configuration consisting of a fifteen-channel counter and a thirty-two channel motor controller.

Acknowledgments

We greatly appreciate Professor John Murray of Electrical Engineering Department of SUNY at Stony Brook for discussions about the emulation porting software. This work was supported by the U.S. Department of Energy under Contract DE-AC02-76CH00016 with Associated Universities, Inc.

References

- [1] S. K. Feng, D. P. Siddons, "A Versatile Stepping Motor Controller For Systems With Many Motors", Nuclear Instrument and Methods in Physics Research A291, 1990, 422-426.
- [2] Texas Instruments, U.S.A., (214)644-5580.
- [3] White Mountain DSP, U.S.A., (603)883-2430.
- [4] Advanced Micro Devices, (800)862-0020.
- [5] Philips Semiconductors, U.S.A., (619)560-0242.
- [6] National Instruments, U.S.A., (800)433-3488.
- [7] Cypress Semiconductor, U.S.A., (408)943-2600.
- [8] Spectron Microsystems Inc., U.S.A., (805)968-5100
- [9] Tartan Compilers, available from Tartan Inc., (412)856-3600.
- [10] S.K. Feng et al., "NSLS Beamline Control and Data Acquisition Computer-System Upgrade", these proceedings.

BEAM DYNAMIC INTERLOCK AND BEAM ABORT TRIGGERING SYSTEMS OF THE IHEP ACCELERATOR COMPLEX

V.V. Gotsev, V.V. Komarov, I.V. Lobov, A.A. Matyushin,
A.V. Popov, Yu.S. Tchernousko, V.I. Terekhov*
OEA Department, IHEP, Protvino, Russia.
*Corresponding author, E-mail: terekhov@tx.oea.ihep.su

Abstract

The presented systems are aimed at increasing the accelerator radiation safety. They are built as standalone hardware distributed systems. The Beam Dynamic Interlock System prevents the beam transfer to the next stage of the accelerator complex if that stage is not ready for the operation (e.g., magnetic field is out of range, bad vacuum, etc.). The Beam Abort Triggering System reacts immediately to an emergency situation at the accelerator by triggering the kicker magnets to dump the beam. Conventional computer control means are only used for post-mortem diagnosis and self testing during beam pauses. This paper describes the conception and components of UNK Beam Abort Triggering System under development and Beam Dynamic Interlock System of U-70/UNK Transfer Line, which was successfully launched in 1994.

I. INTRODUCTION

The IHEP Fast Ejection System (FES) extracts beams to external targets of channels 8,21,22,23 [1] and UNK-I, via the Beam Transfer Line (BTL) [2]. Fig.1 gives a schematic layout of the beam ejection area. A finite probability of pulsed transfer line malfunctions can cause appreciable radiation losses, exceeding tolerable levels. To avoid such a situation permanent monitoring of the magnet excitation currents is used. If operating variables are out of preset range the fast ejection must be interlocked for the corresponding line. The problem at hand is solved by the Beam Dynamic Interlock System (BDIS).

A second aspect of accelerator radiation safety is concerned with the necessity of a single-turn ejection (emergency ejection) of the beam from the UNK-I to the beam absorber when the danger of radiation losses arises. The so-called beam abort is executed by kicker-magnet 10KM1 within the energy range of 70 GeV to 600 GeV [3,4]. The triggering pulse for 10KM1 is generated by the Beam Abort Triggering System (BATS) which acquires and processes the emergency signals from threshold devices distributed along the ring. The radiation monitors are mainly used due to their fast reaction to emergency and preemergency situations. In the framework of the given paper main components of both BDIS and BATS are outlined.

II. BDIS

The main components and structure of the BDIS are shown in Fig.2. This system is distributed, as dynamic permission signals are created at the Ejection Building (EB) and at 3 Surface Buildings along the BTL. These signals are input parameters of the BDIS. They are yielded by permanently monitoring in hardware the excitation currents of the pulsed magnets (window monitoring). The output of the BDIS is the dynamic permission signal for the KM14 and KM16 triggering.

In addition, a fraction of the BDIS inputs is used for monitoring the status of vacuum valves and the pressure at the beam pipes. At the EB some inputs are connected to the radiation monitoring system of the experimental area.

At the heart of the BDIS is a module called Concentrator and Analyzer of Dynamic Permissions (CADP). It has 8 inputs with Diode Optoelectronic Couplers (DOC) for galvanic decoupling that is important for such a distributed system. A built-in time-delay analyzer permits time monitoring of input signals with reference to ejection time with an accuracy of up to 2 ms. A mask register gives an opportunity to disable any input on operator command. That is very useful while modes are changing or during commissioning. A Multibus-I interface allows one to identify inhibition sources and to test the BDIS itself. An emergency switch for the operator is provided.

III. BATS

The system monitors and acquires all local alarm signals (LAS). Their sources are distributed over 16 Auxiliary Buildings located mainly along the UNK ring at typical distances of about 1.8 km from each other. The following original signals are used to generate the LAS:

- the RM signals;
- beam position monitor signals;
- signals from pick-ups of dampers of coherent beam transverse and longitudinal oscillations;
- signals from pressure monitors of the Vacuum System;
- signals "Failure" of the Timing System, etc.

The LASs are generated by the threshold devices associated with the corresponding systems and are sent to a Concentrator - Repeater (CR). The inputs of the CRs are optically isolated from all external devices. All CRs are interconnected by means of an independent loop (coaxial cable). Activity of the BATS takes place during the period between beam injection and beam transfer to the 2nd UNK stage. A Driver Generator (DG), located in the Main Control Room, feeds a 1 MHz pulse train into the loop and receives it. The presence of any LAS at the input of any CR interrupts the train propagation. Having detected train disappearance, the CR of the Abort System building promptly outputs a kicker trigger pulse.

Such an algorithm ensures a time lag between firing of an LAS and the trigger pulse of less than 100 ms including the time propagation delays. Thus the kicker trigger pulse is generated only by hardware. All the CRs are linked to a Multibus-I backplane for post-mortem diagnosis, selftesting in no-beam states and masking any CR input on operator command. It is important to notice that the BATS provides for injection inhibition after an ejection trigger pulse has been generated.

As it was mentioned above, the fastest response is ensured by the RM signals. That is why the associated threshold device was given special attention. The simplified block diagram of the Alarm Level Discriminator (ALD) of radiation losses is given in Fig.3. The signals from the RMs, located near septum magnets, scrapers and other so-called "narrow places", are individually input to Current - to - Voltage Converters (CVC). Their rise time is about 05 ms over an input current range of 15 nA - 15 mA. The output voltage of the CVC is compared with a predetermined reference. When that reference is exceeded, that comparator creates the LAS. A counter registers the event time for post-mortem processing. The reference levels are changed during the period of beam circulation, taking into account the rising risk of radiation as the beam energy increases. That energy is provided by using timing pulses at the major points of the UNK-I magnet cycle.

IV. CONCLUSION

The created BDIS was successfully used during the launching of the BTL in March 1994. The main principles of the BDIS have been confirmed by long-term operation of dynamic interlocking at the FES of U-70. Prototypes of the modules for the BATS have been tested. The electronics modules of the both systems are packed in Euromechanics. The systems are expected to be extended to UNK-II as well.

References

- [1] Yu.M. Ado et al. *Particle Beam Channels Development at IHEP Accelerator*. Preprint IHEP 85-182, Serpukhov, 1985 (in Russian).
- [2] V.T. Baranov et al. *UNK Injection System*. Preprint IHEP 92-118, Protvino, 1992 (in Russian).
- [3] *Design Study on the UNK-I Control System*. IHEP Internal Report, 1988.
- [4] V.N. Alferov et al. *The UNK Control System, Proceedings of the 1991 IEEE Particle Accelerator Conference*. San Francisco, May 1991, pp.1356-1358.

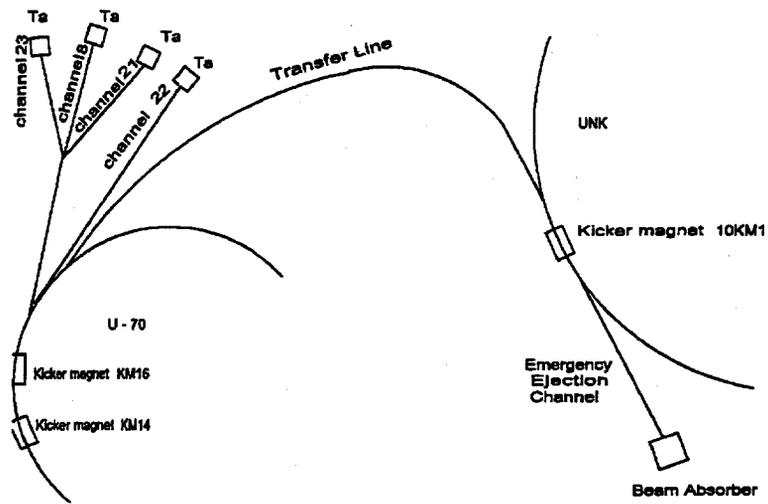


Figure 1. Schematic Layout Of The Beam Area Of The IHEP Accelerator Complex

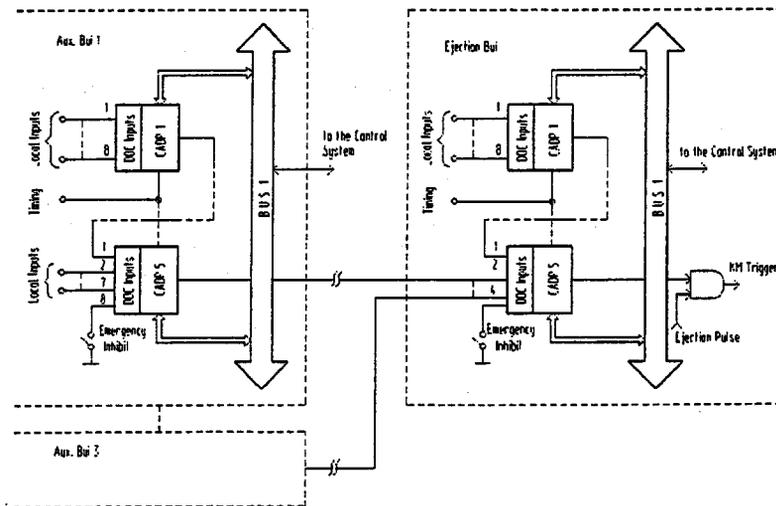
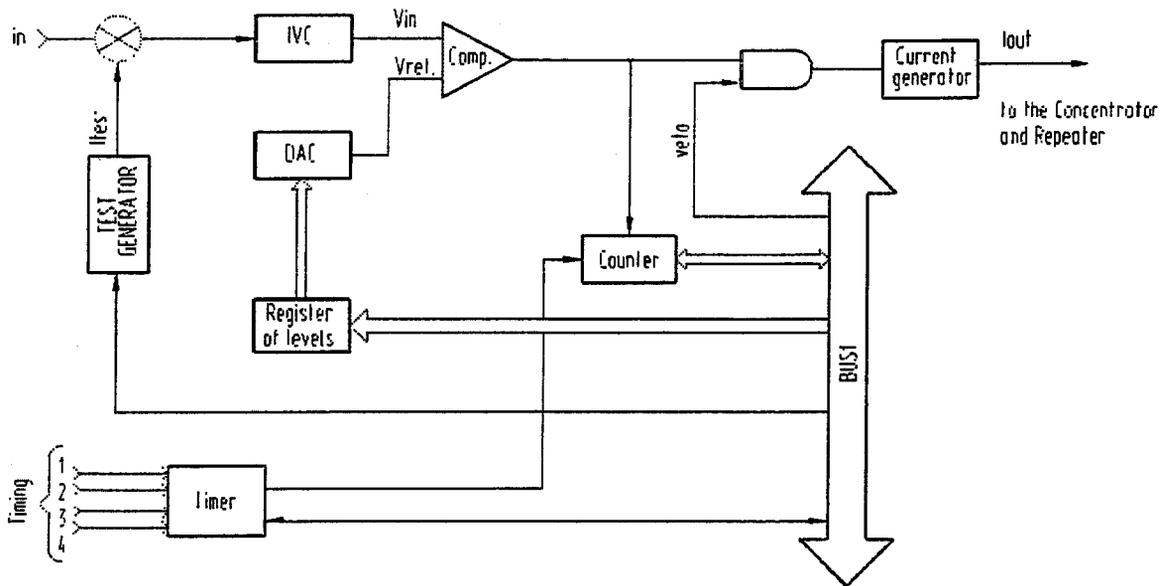


Fig.2. Block Diagram of the Beam Dynamic Interlock System

Figure 2. Block Diagram Of The Beam Dynamic Interlock System



- Timing pulses.
1. Start
 2. Acceleration
 3. FlatTop
 4. Inversion

Fig.3 Block Diagram of the ALD of Radiation Losses

Figure. 3. Block diagram of the Alarm Level Discriminator (ALD)

A NEW LAN CONCEPT FOR LEP MACHINE NETWORKS

A STEP TOWARDS LHC

Louis GUERRERO, Patrick LIENARD, CERN, Geneva, Switzerland.

Abstract

The LEP networks, implemented in 1987, are based on two Token-ring backbones using TDM as the transmission medium. The general topology is based on routers and on a distributed backbone. To avoid the instabilities introduced by the TDM and all the conversion layers it has been decided to upgrade the LEP machine network and to evaluate a new concept for the overall network topology. The new concept will also fulfill the basic requirements for the future LHC network. The new approach relies on a large infrastructure which connects all the eight underground pits of LEP to the Preveessin control room (PCR) with single-mode fibres. From the bottom of the pits, the two adjacent alcoves will be cabled with multi-mode fibres. FDDI has been selected as the protocol. This new concept is based on switching and routing between the PCR and the eight pits. In each pit a hub will switch between the FDDI LMA backbone and the local Ethernet segments. Two of these segments will reach the alcoves by means of a 10Base-F link. In a second phase of the implementation, this scheme will provide for workgroup organisation and bandwidth allocation. The technological choices make a future evolution towards ATM and 100Base Ethernet possible and allow us to preserve a large part of the investment. This paper describes the implementation of this scheme.

1. INTRODUCTION

Since 1974, starting with the SPS machine, networking has always been central to the CERN SL division controls system. When the LEP machine studies were made it was decided to adopt standard protocols. LAN evolution was not clear at that time. To cope with requirements for safety as well as machine performance and operation, delicate decisions had to be made, leading to the use of separate networks for the services (water, cooling, ventilation etc.) and for the machine operation and beam observation. These two networks are known as LEP SERVICES (LSV) and LEP MACHINE (LMA). It was decided to use the IBM Token-Ring protocol and the CCITT Time Division Multiplexing (TDM) as the transmission medium. During the past years these networks have undergone many changes [1] to reach the higher performances always required by users. For many reasons, these basic decisions were never revised. Preparation for the projects LEP 200 and LHC have led to fundamental changes.

This paper outlines these new ideas and concepts and gives a description of the present state of the LMA network after the first phase of improvements. The steps necessary to complete the project are outlined.

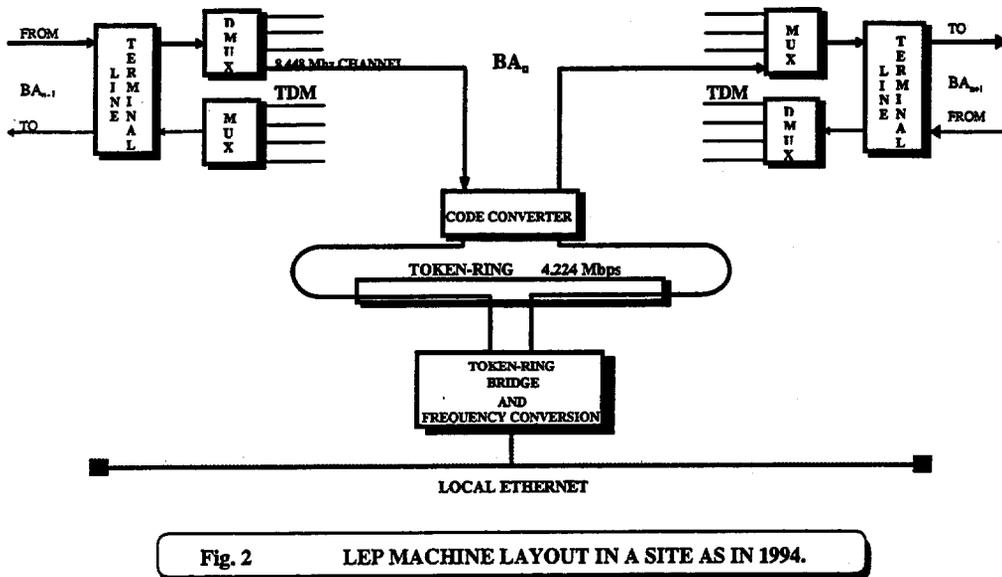
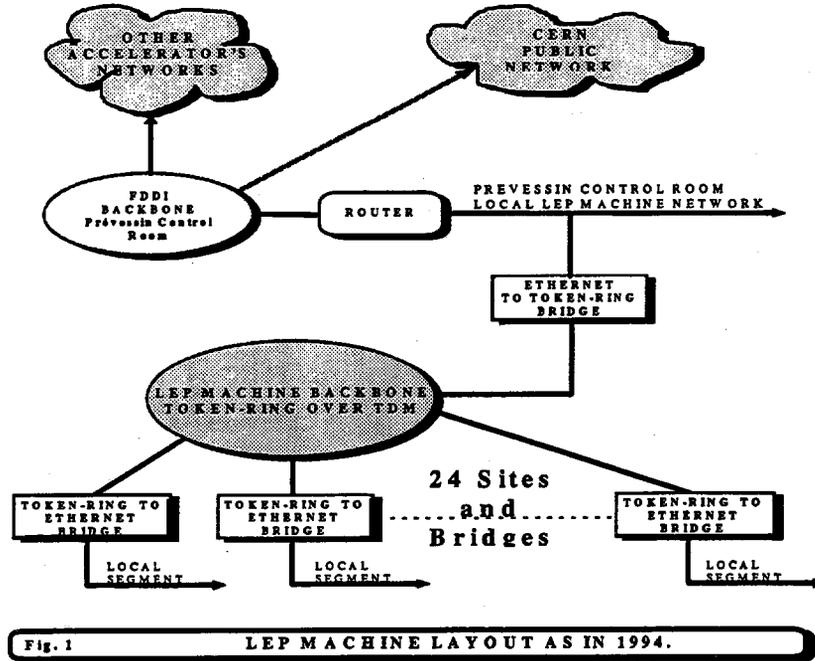
2. THE LEP MACHINE NETWORK IN 1994

2.1 LAYOUT AND CONCEPTS.

The LMA network is interconnected with the Preveessin control room (PCR) and the "rest of the world" by means of a router running the TCP/IP suite of protocols as well as the Apollo Domain one. As shown in figure 1, the LMA backbone was made of a Token-ring running over the TDM and interconnected with local Ethernet segments in alcoves by means of translation bridges. Although this worked satisfactorily for many years it bore some weaknesses which became critical with the increasing network load and the requirements for higher performance.

2.2 RELATED PROBLEMS.

Figure 2 represents the relationship between the Token-ring and the TDM at a site level. It shows the boundaries of the two systems where the problems mentioned below can be identified.



• INCOMPATIBILITY BETWEEN TDM AND THE TOKEN-RING.

Levels of incompatibility can be identified as :

1) We use an 8.448 Mb/s channel of the TDM to sample the Token-ring which has a 4 Mb/s bit rate. These rates not being multiples of each other, some buffering is needed to enable the frequency conversion. We decided to make this conversion at the Token-ring to Ethernet bridge level. It has worked perfectly up to now but it is incompatible with industrial products.

2) The TDM transmission system uses the HDB3¹ code. The IBM Token-ring uses a biphasic technique called Differential Manchester. The necessary conversion between these two codes was made by a home-made code converter [2]. In fact, this was not a true conversion since it only considered the positive part of the electrical signals, but it allowed a simplified design. Some very exceptional patterns are not correctly translated. This introduces some frame errors and penalties on the bandwidth by creating unnecessary re-transmissions. Reliability of the whole system is decreased by the failure potential of this additional interface.

3) During the insertion phase of a Token-ring station the insertion relay of the concentrator bounces. This introduces synchronization errors on the TDM. As it is almost impossible to discriminate between these and real losses it becomes difficult to manage the TDM synchronization signals.

- ***BANDWIDTH WASTING AND LIMITATIONS.***

Bandwidth is always a great concern in transmission systems and using an 8.448 Mb/s channel to transmit a 4 Mb/s protocol appears wasteful. It was imposed by the necessity to sample the Token-ring twice per bit, due to the Manchester encoding. It also means that to increase the transmission rate of the Token-ring to 16 Mb/s would oblige us to raise the TDM rate to 140 Mp/s (a sampling rate of 34 Mb/s times four levels of multiplexing). This represents a real technical and financial limitation which eliminated the use of the Token-ring as a means of improving performance.

- ***TRANSLATION BRIDGES.***

In the 1994 topology, as a result of the historical evolution of the LMA network, a packet traveling from the PCR network to a computer attached to a local segment in an alcove had to be translated three times, introducing longer transmission delays and some penalty on the bandwidth. The traffic increase sometimes resulted in critical bridge congestion, often made worse because broadcast propagation could not be limited. In addition, the introduction of a great number (25) of bridges increased the potential of failure.

- ***FAULT DEPENDENCY.***

Using the TDM as a medium for the Token-ring meant that a fault on the TDM, either at the line terminal or at the multiplexer level, broke the ring integrity. An incident on the Token-ring has also severe repercussions on the TDM transmission. This interdependency made recovery from a fault and the management of both systems very difficult.

3. THE NEW CONCEPT

3.1 INTRODUCTION.

The introduction of diskless local computers, the generalised use of X terminals and of the SUN Network File System (NFS) protocol rendered the system described above inadequate by the end of 1993. Fortunately this situation had been anticipated [1] and the infrastructure needed for the future was already well established. The new network had to eliminate all the above-mentioned problems, i.e., it had to be independent of the TDM. The decision to use an optical fibre-based network was made even if the protocol to be run on it was not firmly decided: technology was again at a decision point and we were too early for the market.

3.2 THE FIBRE LAYOUT.

The installation of optical fibres started in 1985 to implement the TDM transmission over the long distances of LEP. At that time, the equipment available was essentially multi-mode so the cables were composed mainly of multi-mode (MM) fibres with only a few single-mode (SM) ones. We reinforced the single-mode infrastructure for LEP 200 in 1992. Thereafter, we had a comfortable number of SM fibres linking the PCR to each pit. In early 1994, we started the installation of SM fibres from the surface down to the equipment alcoves round the ring pits. The major problem to be

¹ High Density Binary code of the 3rd order, CCITT G703 recommendation.

solved in using fibres in the accelerator environment was to protect them from radiation degradation. Earlier tests had been made by putting some fibres in the central drain under the floor of the LEP tunnel [1] and they were found to be virtually unaffected after one and half years. The decision was made to apply the method to the whole network. Insertion, pulling and extraction of the fibre cables involved considerable engineering work around the drain-pipe. The fibres run in surface trenches to each pit and then go down to the underground site where they are connected to the electronics and then run from there through the drain-pipe to connect to both of the adjacent alcoves. In the drain, the cables are composed of 6 SM and 6 MM fibres. Elsewhere, the cable compositions vary according to the sites but are never less than these figures. The types of fibre in the cables are 10/125 μm for the SM and 50/125 μm for the MM. The surface layout is shown in Fig. 3.

3.3 THE ARCHITECTURE.

THE CHOICE OF THE PROTOCOL.

In 1994, The ATM² implementations were proprietary solutions and ATM standards specifications had not yet reached stable levels. We decided to use FDDI³ which was offering, and still offers today, a more stable protocol for packet transfer. Nevertheless we had to choose a solution that prepared for the future and protected the current investment. This required us to anticipate a migration towards ATM in the future.

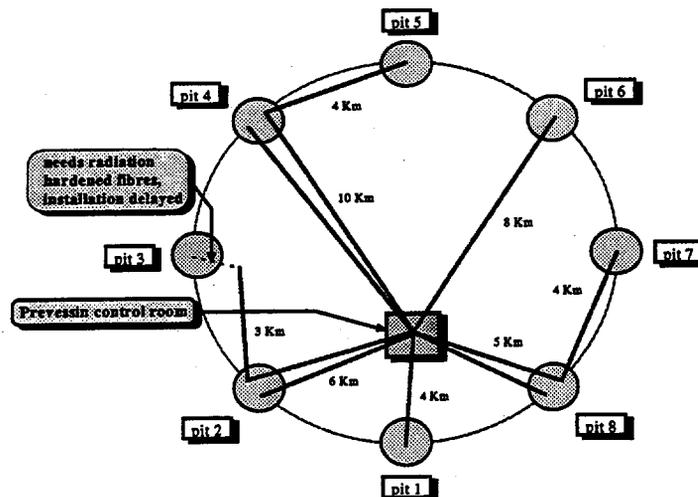


Fig. 3 LEP SINGLE MODE FIBRE LAYOUT.

THE LAYOUT.

Our first approach towards the new network architecture was again based on routing. It soon became evident that we should take advantage of the latest switching technology which provides an increase of bandwidth, bandwidth allocation and virtual LAN (VLAN) facilities. Preserving the investment should also be easier. This led to the topology of figure 4 which shows a central hub in the PCR, switching FDDI links and eight peripheral hubs switching Ethernet links from each FDDI link. For reasons of economy we decided on a Single Attachment (SA) type of FDDI connection. Thus, two pits could connect to the same Double Attachment (DA) port of the central hub and share the same logical ring (see figure 5). Although this solution is less reliable than one based on the dual (counter-rotating) ring and DA type of connections, it saved a large amount of money (about 30%) for the electronics and the SM fibres. Another advantage of this layout will be the possibility of equipping the alcoves with FDDI instead of 10Base-F Ethernet. Finally, installation of dedicated servers on the LMA backbone was also foreseen at the PCR location.

² ATM : Asynchronous Transfer Mode retained by CCITT and supported by ATM Forum.

³ FDDI : Fiber Distributed Data Interface, ANSI X3T9 or ISO 9314 standards.

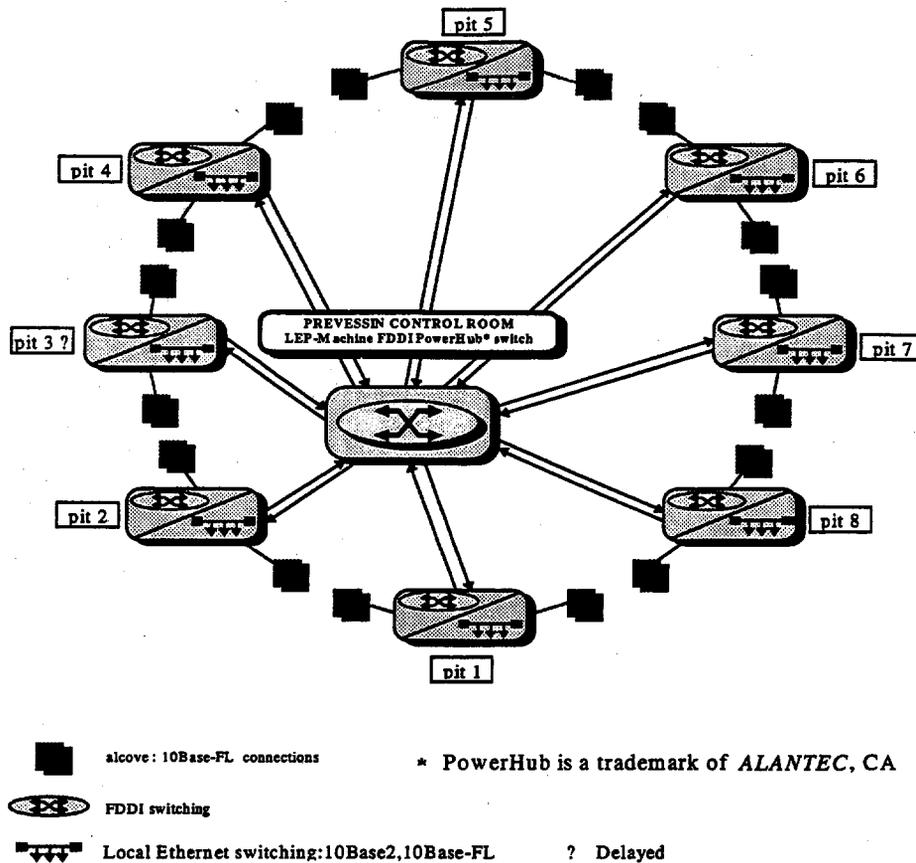


Fig. 4 NETWORK CONCEPTION BASED ON POWERHUBS* FDDI AND ETHERNET SWITCHING

THE ELECTRONICS.

The specification for the new network equipment called for:

- switch and bridge FDDI
- switch and bridge Ethernet from FDDI
- route multi-protocols, essentially the IP protocol suite and to be transparent to Apollo Domain
- have multi-mode and single-mode types of connection for FDDI
- deliver full bandwidth on each port
- allow bandwidth allocation
- feature VLAN facilities
- offer modules equipped with 10Base-2 and 10-F Ethernet ports
- are modular and scalable
- are compliant with SNMP
- are committed to ATM

It also included several options. Among them were:

- full-duplex Ethernet
- TPPMD⁴ FDDI
- 100 Mb/s Ethernet with a preference for 100VG-AnyLAN⁵, our workstations being HP ones

⁴ TPPMD : Twisted Pair Physical Media Dependent.

⁵ 100VG-AnyLAN is defined by IEEE 802.12 standard.

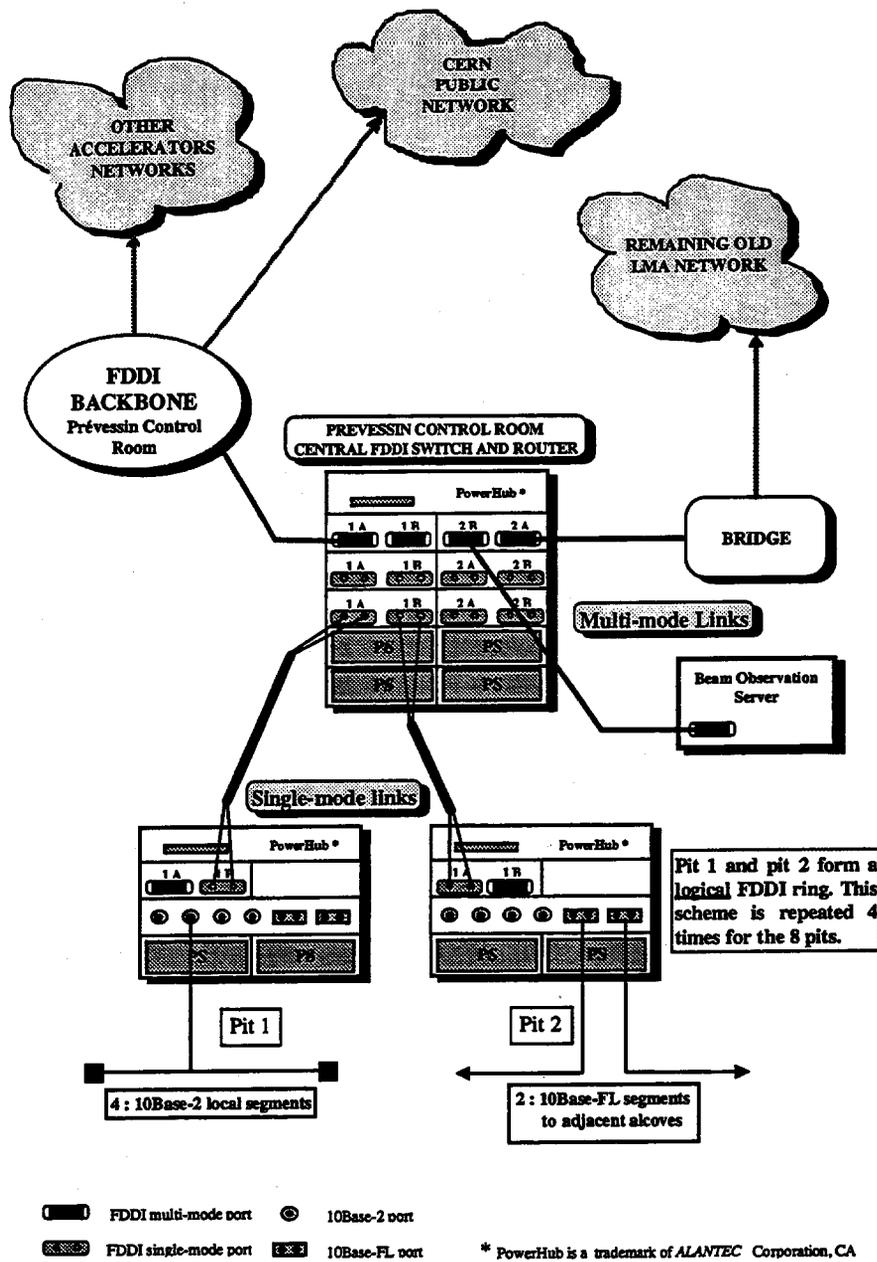


Fig. 5 LEP MACHINE LAYOUT AS FROM MARCH 1995.

All these features together led us to think that we had no chance of finding such a product on the market, but we knew several firms were ready to make some announcements in the year to come. We issued a call for tender and surprisingly found two firms with products closely following the specification.

PowerHub 7000 (PH) from ALANTEC was selected because it completely fulfilled the compulsory requirements and the optional features were also available, apart from the 100VG requirement.

The PH is expandable and scalable from 1.6 to 3.2 Gb/s total channel bandwidth. Its multiple-port shared memory provides and shares bandwidth at the chip level. This means that the PH software can forward packets using any bridging, routing or filtering method which can be expressed by a programmed algorithm. It also provides many network management features such as statistics gathering, security filtering and port monitoring. Furthermore, new features can be added at will, at any time and *adaptation to new standards* is easy.

NETWORK MANAGEMENT.

The LEP network, like LEP Services and the SPS accelerator networks, uses SNMP⁶ and RMON⁷ protocols, via the HP OpenView Interconnect management product and the HP Metrics statistics and analyser tool, to control all the network equipment. These were all retained for the new project. In addition, we needed and obtained an easy but powerful configuration tool for VLAN implementation. HP OpenView is a very flexible product which allows easy integration of other vendors' products. ALANTEC offers Power Sight, a network management utility, which is totally compatible with OpenView.

4. THE STAGE OF IMPLEMENTATION IN 1995

4.1 PRESENT STAGE.

LAYOUT.

At the end of the LEP annual shutdown (end of March 1995) fibres had been installed down all the pits except pit 3. For the latter we had either to find radiation-resistant fibres or a special path to pull normal ones. Alcoves were linked to the pits with fibres in the drain-pipe, except between pits 1 and 4 where special engineering was needed to clear the drain.

ELECTRONICS.

PowerHubs were installed down all the pits, except pit 3, with the following configuration:

- one chassis with one power supply
- one packet engine
- one universal FDDI module equipped with one A or B type SM port (two sites sharing the same logical ring) and one B or A type MM port allowing local FDDI ring extension
- one universal Ethernet module equipped with four 10Base-2 ports and two 10Base-F ports

The PowerHub in the was installed with the following configuration :

- one chassis with three power supplies
- one packet engine
- one dual FDDI MM module connecting the PCR backbone and offering an attachment for LMA local servers
- two dual FDDI SM modules connecting (switching) all the pits to the PCR backbone

The alcoves from pits 1 to 4 via 8,7... were all equipped with a 10Base-F link originating in the pit PH and using optical to copper converters to create the local Ethernet segment.

CONFIGURATION.

One of the MM ports on the PCR PowerHub was configured as the router port between the PCR and LMA subnets. All other ports of the PH are members of the LMA subnet and as such have received the same IP address. All ports are bridged. All the ports of a PH in a pit are bridged and have the same IP address per hub. Thus, only bridging is implemented nowadays without using VLAN, unless one considers the whole LMA network as a workgroup.

FIRST IMPRESSIONS.

To date, we have accumulated six months of experience and had to face several small problems :

- the power supplies did not comply with European standards
- some rare losses of a FDDI connection disturbed the PCR backbone
- some rare IP routing lock-out on the PCR port isolated the LMA subnet

⁶ SNMP : Simple Network Management Protocol.

⁷ RMON : Remote MONitoring protocol.

We have been testing some new power supplies proposed by ALANTEC and they have proved satisfactory, so we are now ready to install them. Upgrading the FDDI and OS software versions seems to have solved the problem of losses. We have now been running the latest version for over a month without any trouble.

At this stage, after a six months' operation, we have gained appreciably in reliability and performance. Time was too short to make measurements on the network but in normal operation of LEP we have gained about a factor of three on the response time of a standard 64-byte ICMP⁸ echo message crossing all the network. Statistics show extremely low error rates over long periods, for example:

PCR PH :

FDDI : 1 800 million packets over 605 hours with 45% peak utilisation showed zero errors on Xmit/Rcv buffers and zero packets dropped.

US25 PH :

FDDI : 69 million packets over 410 hours and 43% peak utilisation, zero dropped .
Ethernet port 13 : 7 million packets with 21% peak utilisation showed 21 Xmit buffer errors, 1 Frame Check Sequence error, 1 Frame Alignment error and 262 collisions. All other ports have 0 error counters.

US25 is the worst case today of all the PHs installed.

4.2 COMPLETION OF THE FIRST PHASE

During the 1995-96 shutdown the drain-pipe preparation will be completed between pits 1 and 4 via pits 2 and 3. Then we will be able to pull the fibres in the drain. We now have a solution for the connection of pit 3 with normal fibres. The first phase will be completed by the end of the shutdown as we already have the electronics to equip these remaining sites.

5. FUTURE STEPS

The second phase will involve implementing VLAN over the network. Before doing this we need to gain confidence in the whole first phase, so it will certainly not take place before 1997. Studies could start in 1996.

During the 1996 shutdown we also plan to implement the same topology for the LEP Services network. Studies have already been undertaken for the SPS network.

6. CONCLUSION

Due to a lack of Token-ring components on the market and to a non-standard network implementation (the TDM being used as a communication media) the LEP machine network was inadequate for the requirements foreseen for LEP 200 and LHC. A new network conception had to be evaluated, based on today's standards and on an optical fibre infrastructure. It has been (and will be) a difficult job to install the fibre infrastructure but the effort was worthwhile as we have gained a solid base for many years. The decision to implement switching over FDDI and Ethernet appears to have been confirmed by market trends. Choosing the PowerHub brought, very quickly, more than the expected improvement in performance and reliability. The choice of ALANTEC offers us VLAN techniques today, good management tools and confirms that our decision on the HP network management product was also correct. Not only can we meet LEP machine users' requirements for many years but also we have prepared for future migration towards ATM. Associated with the category 5 cabling we had already installed in the PCR, switching will also provide us with a solid base for multimedia communications.

⁸ ICMP : Internet Control Message Protocol.

7. ACKNOWLEDGMENTS

The project has always been encouraged by L. Evans, director of the LHC project, P.G. Innocenti, head of CERN's Telecom board, K.-H. Kissler, head of SL division, and H. Isch, SL division planning officer. We also received a strong support from our group, particularly R. Lauckner, group leader, and R. Rausch, deputy group leader. We express here our gratitude.

It is a pleasure to acknowledge the contributions of :

B. Amacker, K. Kohler, O. Olsen and D. Schamme for their tremendous work on the fibre installation, and A. Francano and J.-L. Vo-Duy for the electronics installation

8. REFERENCES

- [1] P. Lienard, "Evolution of the SPS and LEP communication control network for the SPS and LEP accelerators", ICALEPCS 1993. NIM Vol. A. 352, Nos. 1,2.
- [2] B. Hall and D. Swoboda, The ACCI code conversion interface for the LEP Controls token ring, LEP Controls note 84, CERN, 25 February 1988.

CANDIDE

A fieldbus system to control detector front-end electronics.

W.P.J.Heubers, R.G.K.Hart, H.Boterenbrood, P.Rewiersma, I.Weverling
NIKHEF, National Institute for Nuclear and High-Energy Physics
P.O.Box 41882, 1009 DB Amsterdam, The Netherlands

The CANDIDE system is developed to control and monitor the front-end electronics of a subdetector prototype for the ATLAS inner tracker. CANDIDE was based on the Controller Area Network (CAN), which is developed by Bosch from Germany for the automotive industry. Low cost, reliability and real-time capabilities were important issues in the selection of this fieldbus for the control of the front-end electronics. Prototype CAN nodes, based on the Phillips 80C592 micro controller, have been built to control functions such as setting discriminator thresholds, masking noisy channels and monitoring temperature sensors of the channels in the ATLAS subdetector. This paper describes the implementation of a network of these prototype CAN nodes. Additionally, the integration of a generic fieldbus controller into the Cortex distributed control system (CICERO project) is discussed.

1. Introduction.

ATLAS is a high-energy physics detector to be used in the Large Hadron Collider at CERN and is planned to be operational around the year 2004. At present the detector control system for the ATLAS project is in the phase of determining the user requirements. Prototypes are built and tested in the laboratories and in test beam runs at CERN.

Section 2 gives a brief outline of the organization of a detector control system and in some more detail the subsystem for the front-end controls. In section 3 a description is given of the Controller Area Network (CAN) and section 4 presents the CANDIDE system, based on this fieldbus. Section 5 outlines a contribution to an R&D project initiated by CERN [1], concerning the definition of a generic fieldbus controller that has to be integrated into the Cortex (object-oriented) control system.

2. Detector control.

From a control system point of view the ATLAS detector is a large scale system, that can be described as a set of devices, such as the inner tracker detector, the calorimeter, the muon detector and systems like the data acquisition system and the gas system. Each of these devices and systems can be decomposed into subsystems with their own functionality and degree of autonomy. This hierarchical organization of the detector control system is illustrated in figure 1.

For each subdetector (device) the control and monitoring of the front-end electronics can be considered as a subsystem of the overall detector control system. In case of malfunction of the front-end electronics, an alarm generated by this subsystem must be reported to the detector control system. Typically for each channel (or set of channels) in the detector, slow control functions, such as setting a discriminator threshold or masking noisy channels, could be implemented. Besides these control commands, it is necessary to monitor continuously the status of certain parameters in the front-end system (e.g. voltage trip levels and temperature), which generates possible alarms. The number of detector channels in a high-energy physics detector like ATLAS can be very large. Depending on the type of (sub)detector several thousand (sets of) channels, geographically distributed in the detector, have to be controlled and monitored.

The environment in which the front-end electronics operates puts special requirements to the system. The detector is only accessible during maintenance periods. This puts demands on the reliability and fault-tolerance of the control system. The control path must be redundant and independent from the functioning of the data acquisition system itself. Another important constraint concerns the radiation. This is especially a problem in the inner part of the detector where all electronic components must be radiation hard.

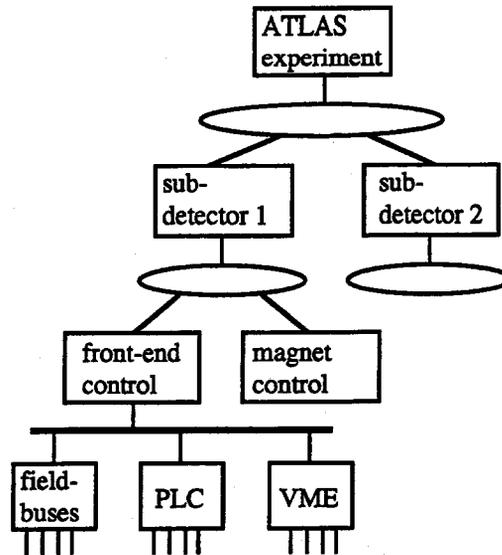


Fig. 1. An example of the hierarchical organization of a detector control system.

3. The CAN fieldbus.

A fieldbus or serial bus is in principle an industrial communication network to which a number of geographically distributed nodes are connected. Fieldbuses are meant for low level equipment control functions like monitoring sensors and controlling actuators. The fieldbus network is able to transport relatively small messages in a fast and reliable way to exchange information between the different nodes. Fieldbus nodes often have some degree of local intelligence, implemented in a micro controller chip, in a PLC system or even in a PC. Low-cost, compactness, reliability and real-time capabilities (requiring response times to be short and deterministic) are important features for selecting a fieldbus system. See reference [2] for a tutorial on fieldbus applications in physics laboratories.

The CAN (Controller Area Network) fieldbus was selected for the controls of a prototype subdetector for ATLAS. The CAN fieldbus was developed by the German company Bosch, initially for the automotive industry, but nowadays CAN is used extensively in industrial fields as well. One major reason for selecting CAN was that the first two layers of the ISO/OSI reference model are available as an open industrial standard as defined in ISO/DIS 11898 [3]. This standard does not cover the transmission medium (twisted pair or optical fiber) and the transmission speed (between 5 Kb/s and 1 Mb/s, depending on distance and the medium). The CAN protocol specifications define multi-master mode, multicasting, a message priority scheme and a high degree of fault-tolerance. Integrated circuit manufacturers such as Philips, Motorola and Siemens offer both stand-alone protocol controller IC's and integrated protocol controllers on a micro controller chip.

The application layer of the ISO/OSI reference model is covered by the CAN Application Layer CAL [4]. The layers between the data-link layer and the application layer are not implemented in fieldbus networks for performance reasons, although management control functions available with CAL do offer services concerning network management, layer management and identifier distribution (figure 2).

The CAL application layer defines a set of services, called CAN-based Message Specification (CMS). With CMS the applications can specify CMS-objects like variables, events and domains and specify services upon these objects such as 'write variable', 'notify event' or 'download segment'. CMS is derived from the ISO/IEC 9506 standard MMS, an application layer standard designed for the remote control and monitoring of industrial devices (such as PLC's).

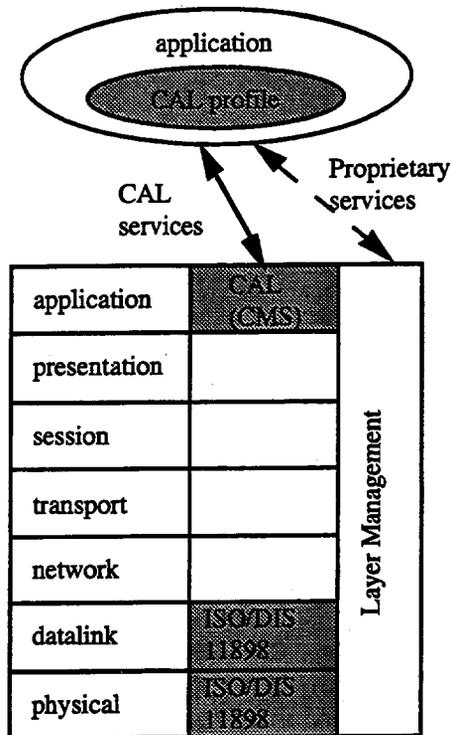


Fig. 2 The CAL reference model

4. CANDIDE.

CANDIDE (a CAN Diverse Interface for Detector Electronics) is developed to control and monitor the front-end electronics of prototype micro-strip gas counters (MSGC). It was foreseen that the forward trackers of the ATLAS detector would be partly implemented with MSGCs¹.

A general-purpose CAN node GPCAN [5] has been designed around the P87C592 micro controller from Philips and has been implemented as a piggyback board. This micro controller includes EPROM and RAM memory, a CAN protocol controller, a 10-bit ADC with 8 multiplexed outputs, two 8-bits PWM outputs and five 8-bit I/O ports. The piggyback card is placed on a GPCAN motherboard, that contains additional functionality such as a serial DAC and a CAN bus connector. A PC with a CAN controller from Softing GmbH was used for the development of the system. The configuration to be used in the test beam runs was hosted by a Sun workstation interfaced by a CV002 VME module from MicroSys Electronics GmbH. The VME module is connected to the actual CAN network by an optical link as is shown in figure 3.

The application software is not implemented with the CAL standard, but instead a custom-made message format was defined, with a maximum frame length of 130 bits. A transmission speed of 125 Kb/s on the bus resulted in a maximum transfer time of 1.04 ms per message.

¹Recently the ATLAS collaboration took the decision to use silicon strip detectors instead of MSGC's.

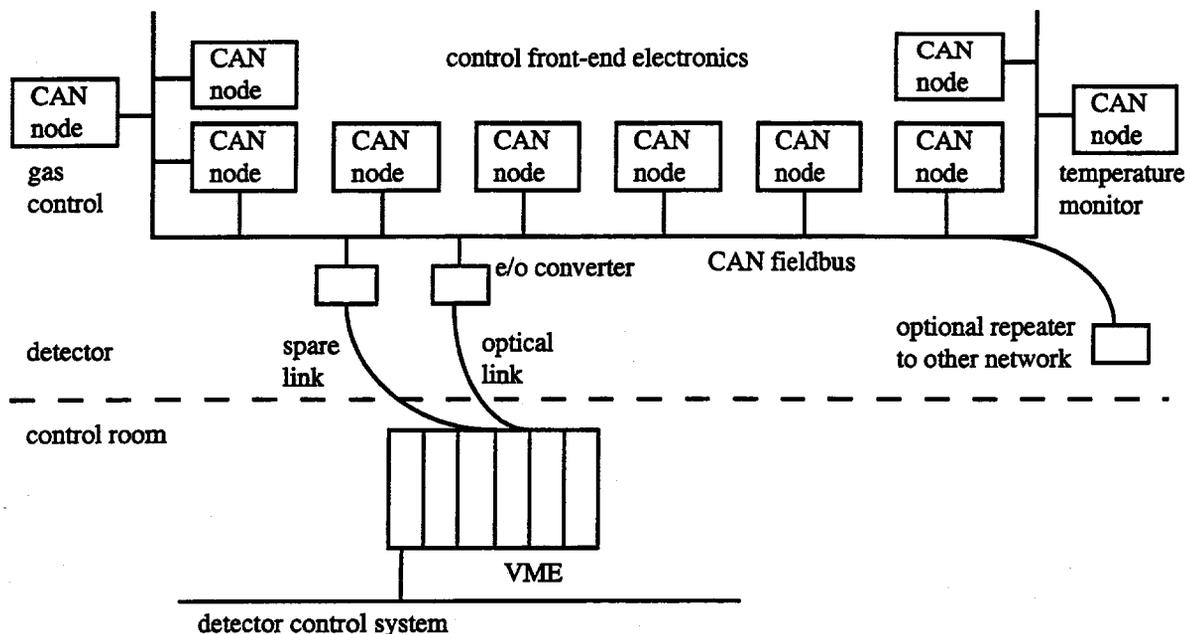


Figure 3 Detector front-end control with a CANDIDE fieldbus system

5. Integration in the control system.

As outlined in figure 1, the fieldbus networks are used for low-level, real-time control of the detector front-end electronics. In a large detector control system as needed for the ATLAS detector, it is foreseen that both fieldbuses and PLC systems will play an important role in the low-level control of other parts of the detector system as well (e.g. vacuum and magnet systems). Although the aim is to apply within the scope of this project 'standardized systems' as much as possible, it is hard to avoid using different PLC and fieldbus systems for different subsystems, because of complexity and time scale constraints. The control system must be able to integrate these heterogeneous subsystems into the overall detector control system.

CICERO is a research project at CERN [1] that will outline and provide the main building blocks of a generic control information system in order to reduce costs and maintenance efforts of future experiments there. The CICERO group has designed and partially implemented an integration framework ('software bus') called Cortex [6]. The Cortex system has two parts: an off-line part in which the configuration of the control system is described and an on-line part where information and commands must be distributed to the components of the system. Cortex has to be flexible enough to support the addition or removal of components without deteriorating the operation of the system. For this purpose Cortex maintains a repository to handle the logical description of the architecture and to describe the exchange of information between the different components.

Industrial fieldbus and PLC systems often come with their proprietary configuration tools. Most of these tools use a graphical editor to design the fieldbus or PLC configuration and to generate files that contain tables with the configuration information. At the moment there is no standard available for the configuration formats of fieldbus and PLC systems, as there is no generally accepted standard available which specifies the application interface, although in some cases suppliers publish their interface specifications and make them available as an open standard to other suppliers. Examples of these are the ISO/DIS 119898 standard that defines the CAN Application Layer for the CAN fieldbus and the ISO/IEC 9506 standard that specifies the Manufacturing Message Specification for PLC systems.

Work has started to design a generic fieldbus controller for the object-oriented Cortex system as it is defined by CICERO. From this generic or virtual controller, other controllers can be derived for specific fieldbus implementations. These derived controllers will inherit attributes and methods as they are defined by the generic fieldbus controller. One interesting topic in this respect is how to convert the proprietary configuration formats into the Cortex repository. As far as the application layer is concerned, the Can Application Layer (CAL) is a good candidate for the implementation of a prototype Cortex fieldbus controller.

6. Future directions.

Although the initial application for the CANDIDE system, the micro-strip gas counter detector for the inner detector of ATLAS, is canceled, nonetheless the project will continue. A modified version of CANDIDE will be built to control the electronics of a prototype muon detector that is part of the outer detector of ATLAS. For this purpose a CAN node is being developed that can control up to 16 high-voltage channels in the front-end system. The intention is to interface the application software for this prototype using the CAN-based Message Protocol CMS as defined in the standardized CAL.

A study on the integration of these fieldbus applications into the higher-level layers of the control system will continue within the framework of the CICERO research project.

7. Conclusions.

The use of an industrial fieldbus to control the front-end electronic system of a large detector like ATLAS seems to be a good choice because of its reliability, compactness and low-cost and because of the industrial support. The design constraint to separate the data path from the control path can be easily achieved by implementing a fieldbus network. The CAN fieldbus is an excellent choice, not only for its reliability and its real-time capabilities, but also because its availability as an open standard. The CAN application layer provides a complete and well-defined interface to access and manage the fieldbus network configuration. One point not yet resolved, concerns the radiation hardness of the industrially available micro controllers. Radiation hardness is a major issue, especially in the inner parts of ATLAS .

References.

- [1] J.M.Le Goff et.al. CICERO: Control Information system Concepts based on Encapsulated Real-time Objects. 1994 Status Report. CERN/LHCC 95-15 LCRB status report / RD 38
- [2] R.Rausch, CERN. Tutorial on standard fieldbuses applications in physics laboratories. Invited tutorial paper ICALEPS '93
- [3] Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication. International Standard Organization. Reference number ISO 11898:1993(E)
- [4] Konrad Etschberger CAN, Controller Area Network Grundlagen, Protokolle, Bausteine, Anwendungen ISBN 3-446-17596-2
- [5] ATLAS MSGC Support Document for the Micro Strip Gas Counter Sub-system. ATLAS internal note, INDET-NO-076, 15 December 1994
- [6] R.Barillere et.al. The Cortex project. A quasi real-time information system to build control systems for high-energy physics experiments. Proceedings ICALEPS'93

The Timing System of the IHEP Accelerator Complex

V.V. Komarov, I.V. Lobov, A.V. Popov, Yu.S. Tchernousko

1. INTRODUCTION

The main purpose of the timing system is generation of timing information, its transmission to technical systems and its reception and transformation into a form convenient for users. The timing information includes a 1-ms clock, main events corresponding to special points in the magnetic cycle, PLS-telegrams, values defining types and numbers of magnetic cycles, astronomical time and data carrying the current parameters of the accelerator (beam intensity and magnetic field).

The main parts of this system are (Fig. 1):

- the Timing Message Generator (TMG),
- the Repeaters (REP),
- the Timing Message Receivers (TMR),
- the Multichannel Timers (MT).

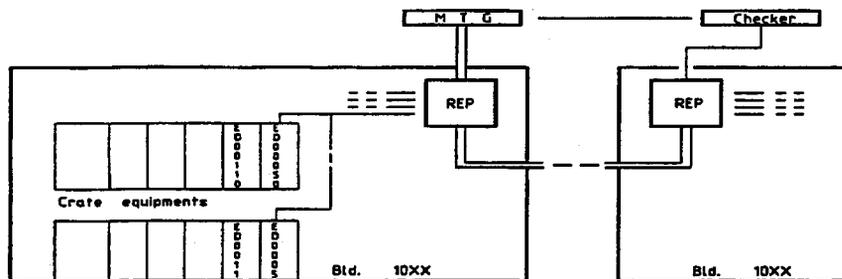


Fig.1. Timing structure

2. DISTRIBUTION.

Timing information is transferred as two-word messages through a network which has a multistar ring topology. The transfer standard is MIL1553. The nodes of the timing network are repeaters. They are placed in every technical building and divide the transmission line into more than 20 segments, including beam transfer lines, of 1.6 km each. Vitality of the system depends on the quality of a transmission medium, which is based on 9 mm 75 Ohm coaxial cables and methods of their connection with the repeaters in each segment.

Three versions of segment organization were studied, Fig.2:

- a single cable,
- two cables in parallel,
- two cables combined by a transformer in a special way.

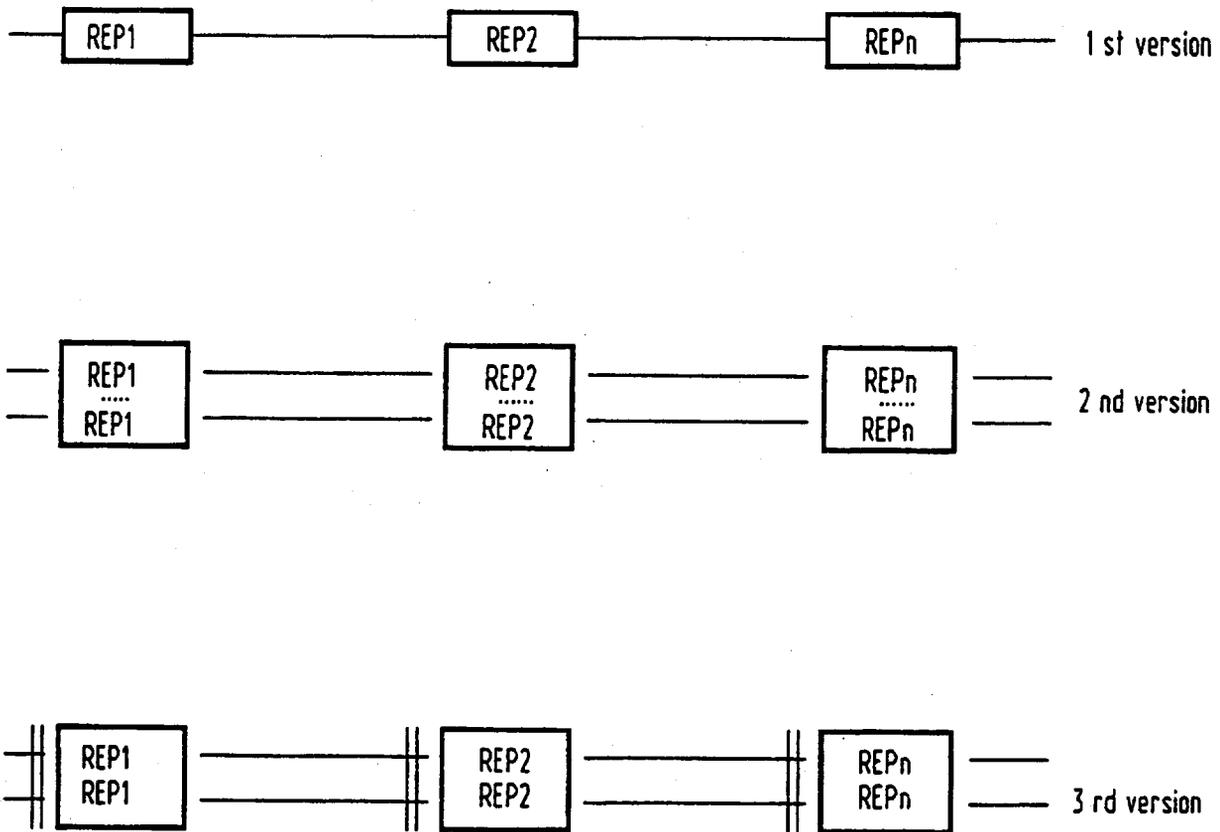


Fig.2. Transmission line structure

Probabilities of the faults for each scheme in the case of cable damage correspond to

$$P_1 = 1 - (1-p)^n, \quad P_2 = [1 - (1-p)^n]^2, \quad P_3 = 1 - (1-p^2)^n,$$

where: n - the number of segments,

p - the probability of one cable damage in any segment.

For $p \ll 1$ these equations become:

$$P_1 = np, \quad P_2 = n^2 p^2, \quad P_3 = np^2$$

The third scheme of communication was chosen because its probability of failure is lower than in the other two cases by factors of p and $1/n$ respectively. Multidrop buses corresponding to MIL 1553 standards are used in buildings.

3. SYNCHROTRAIN

The stream of timing messages forms the UNK Global Synchrotrain (GST), that has a periodic character. The basic time unit of this train is a frame of 1 ms duration. Each frame is divided into 10 time slots which are spaces for messages, Fig. 3. The message in slot number 0 is available to all users (broadcast mode of transmission) Messages in the slots numbered 1 - 9 belong to individual users or fulfill specific functions (address mode of transmission).

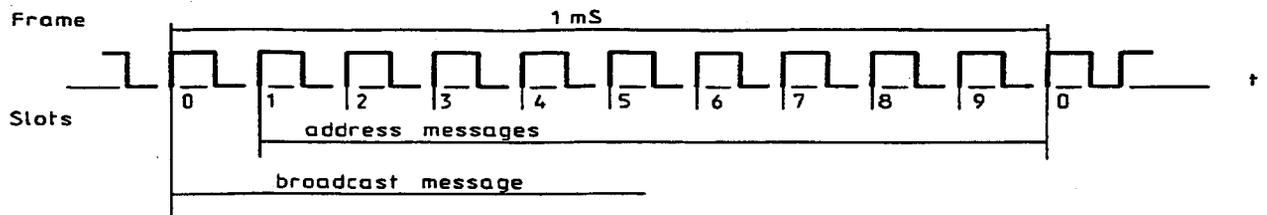


Fig.3 UNK GST frame

These timing messages follow the MIL 1553 standard and consist of two words - a command word and a data word in MIL 1553 terminology. The first byte of a message is for control; it defines addresses of accelerating stages to which the transferred information belongs and the type of information to follow. The next three bytes contain data itself. The entire data format is spelled out in Table 1.

4. COMPONENTS.

Most of timing modules were designed to the MULTIBUS-1 standard.

4.1 Timing Message Receiver.

The TMR (serial number ED00050) is a part of all equipment controllers. It receives messages and processes them in two ways:

- hardware decoding of the 1-ms clock and up to 8 events, output to users being via external connectors,
- data storage of "beam intensity", "magnetic field", "astronomical time" and "events" in special registers, readable via the bus.

Two levels of interrupts initialize crate controller programs. The first level informs the processor of an absence of the GST, the second one combines the signals of decoded events, astronomical time and faults in the received information.

A built-in message generator permits checking of the functioning of all the nodes of the receiver.

Output-event pulses are synchronized by the 1-ms clock. The set of selected events is held in PROM (PAL16L8).

4.2 Multichannel Timer (MT)

This module (serial number ED00110) is used in equipment controllers to form additional pulses delayed from start event pulses and/or resynchronized by other clocks. It consists of 6 independent preloaded timer channels; each of them has a 16 bit capacity and is able to count up to a 2 MHz clock rate. It is possible to use any of the 7 clocks: three of them are produced by a generator (1MHz, 100 kHz and 10kHz), the other four are external. Cross connection of the timers' inputs and outputs in the connector in a special way allows the formation of any of the three orders of pulse generation (Fig. 4). The pulse signals produced are used to start controlled equipment via external connectors and/or to initialize crate controller programs via the interrupt mechanism. A built-in diagnostic scheme allows the testing of any timer channel of the module.

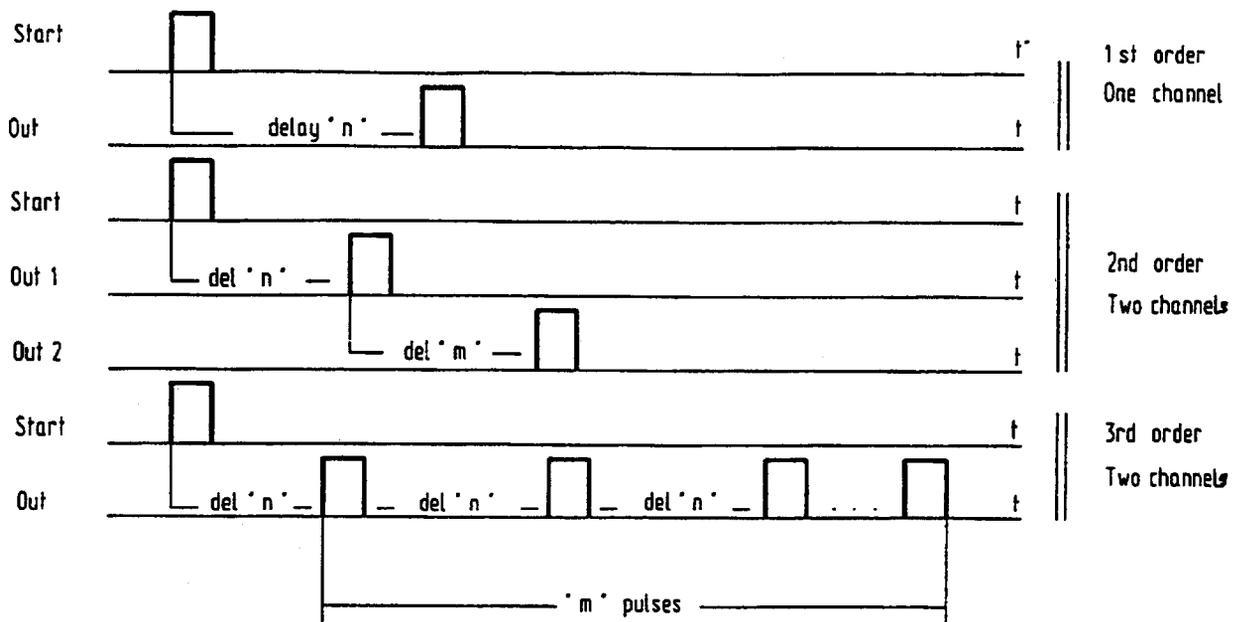


Fig.4. MT pulse generation

The serial number of the REP is ED00140. This is a part of a UNK retransmission station and it works under the control of a crate controller. The main function of the module is shape restoration and distribution of the GST signal to users. Another is to supervise the availability of the signal in both the cables of the transmission line and to receive test messages. The crate controller changes the addresses of the MIL 1553 decoder and checks a predefined sequence of test messages. The module also receives astronomical time messages and registers hardware faults. The crate controller programs are initiated via two levels of interrupts.

4.4 Timing Message Generator (TMG)

- This device consists of four types of module:
- crate controller (PC-16, based on the Int8086),
 - 0.5 Mbyte RAM (DM-16),
 - multiplexer (ED00150) and
 - timing message transmitter (ED00100).

The DM-16 keeps the working table of events for the UNK supercycle. The timing message transmitter coordinates the running of the crate controller program via four interrupt levels, receives table data and data from the two multiplexers (4 channels, 16 bit words), forms timing messages according to defined requirements and transmits them. The multiplexers take technical data (beam intensity, magnetic field, astronomical time) from the corresponding measuring systems in parallel.

5. DIAGNOSTICS.

An important requirement in the design of the UNK timing is the ability to find out when and where something goes wrong within the system. Special devices and built-in means allow one to inspect streams of data in the timing network and to check the functioning of the timing modules.

There are three levels of timing message control:

On the global level a special device (Checker) compares the messages at the input with the same messages at the output of the ring transmission line.

At the level of the repeaters, the availability of the GST signal in both the cables is monitored and special test messages are checked.

At the level of the timing message receivers, checks are made of the availability of the GST signal in the multidrop lines and the correctness of the message decoding. In addition the operation of the timing modules is reflected in status registers that can be read, and most of their working functions can be simulated.

6. DELAYS.

The time delay of the GST signal along the ring transmission line depends on the length of the cable and number of repeaters used:

$$T_{del} = t_{cabl} * L + t_{rep} * n,$$

where: $t_{cabl} = 5$ ms per km,

$t_{rep} = 0.25$ ms per unit, defined by the process of GST signal shape restoration.

For the whole of the ring transmission

$$T_{del} = 5 * 20 + 0.25 * 16 = 104 \text{ (ms)}.$$

The influence of the seasonal change of temperature on the cable is the major cause of time delay instability. Given this it will be 1.2 ms, if the changes of temperature are no larger than 60 C.

The process of GST signal decoding in the TMRs causes a time jitter of both the 1-mS clock- and the event-pulses relative to the corresponding frames. This jitter can amount to 80 nS.

Controller Area Network (CAN) — a Field Bus Gives Access to the Bulk of BESSY II Devices

J. Bergl, B. Kuner, R. Lange, I. Müller, R. Müller,
G. Pfeiffer, J. Rahn, H. Rüdiger

Berliner Elektronenspeicherring-Gesellschaft für Synchrotronstrahlung m.b.H.
(BESSY), Lentzeallee 100, D-14195 Berlin, FRG *

Abstract

The superior properties of the CAN fieldbus will be widely utilized for BESSY II. Literally any power converter, the PLC of the radio frequency, vacuum valves and gauges etc. will be interfaced by embedded controllers attached to a CAN segment. Furthermore it is envisaged to set up a feed forward insertion device compensation scheme that is based on a specific CAN installation. Today a unified data link layer (called SCI: Simple CAN Interface) simplifies the development of communication software between CAN chips of different manufacturers on the embedded controllers or the VME master. The usefulness of the specific subset CAN Message Specification (CMS) of the CAN Application Layer (CAL: proposed standard) protocol is under study.

Introduction

The control system at BESSY I has used the CAN field bus for controlling many power supplies for two years. The installation has proven to be robust and reliable in a very noisy environment [2]. This excellent experience has resulted in the extended use of CAN as the field bus in the BESSY II control system. DESY also uses the CAN bus for device control, so it is planned to combine the efforts of the two laboratories in this field. BESSY II is designed to use insertion devices to provide high brilliance synchrotron light. This leads to some special control system demands. One is to implement an insertion device (ID) compensation scheme that guarantees a correction of the machine orbit every 100 ms.

I. A Field-Bus-Based Control System Solution

The BESSY II control system will be distributed over the storage ring, the synchrotron and the transfer line. At the primary IO level, VMEbus computers attached to an ethernet with EPICS as the control system software will be placed at strategic places around the storage ring and the synchrotron.

A first goal is to incorporate intelligent actuators and provide a high flexibility in connecting distributed hardware to the primary IO level. Therefore a secondary IO level based on the CAN field bus will be installed. Only a small amount of equipment will be directly interfaced by VME cards.

A. *Embedded Controller Concept*

The main power supplies require analog signals with 16-bit resolution and low thermal drift. Therefore the DAC has to be placed as close as possible to the power supply. Each power supply at BESSY II will be equipped with an interface card that consists of the DAC, an ADC and digital IO. The interface card carries an embedded controller board based on the Intel i80386EX processor. This ensemble fits directly into a slot in the power supply. Furthermore the embedded controller provides the CAN interface using the Intel 82527 full CAN chip. This solution leads to power supplies fully controllable via the CAN bus with short distances between the interface and power supply electronics.

For software development the C programming language has been chosen. The embedded controller is fully hardware compatible with an 80x86-PC. Therefore the software can be used on both platforms with minimal modifications. A CAN interface card for 80x86-PCs using the i82527 CAN chip is also available.

B. *ID Compensation*

*eMail: bergl@bii.bessy.de, kuner@bii.bessy.de, lange@bessy.de, muelleri@bii.bessy.de, mueller@acc.bessy.de, pfeiffer@bii.bessy.de, rahn@acc.bessy.de, ruediger@bii.bessy.de

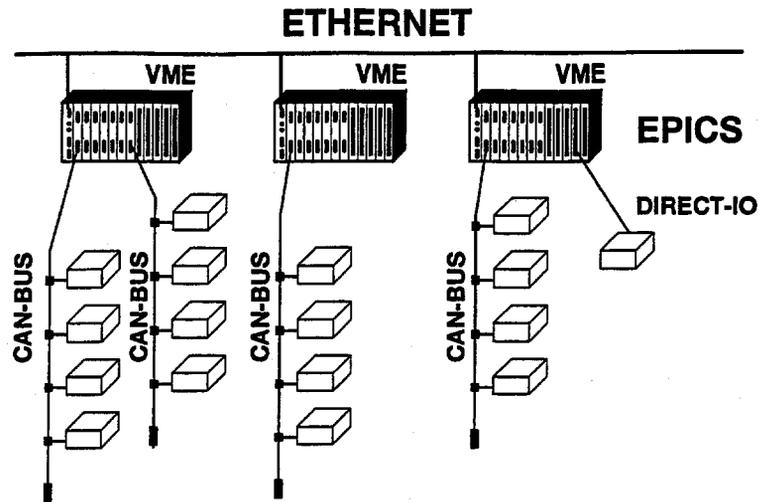


Figure. 1. IO Level Overview

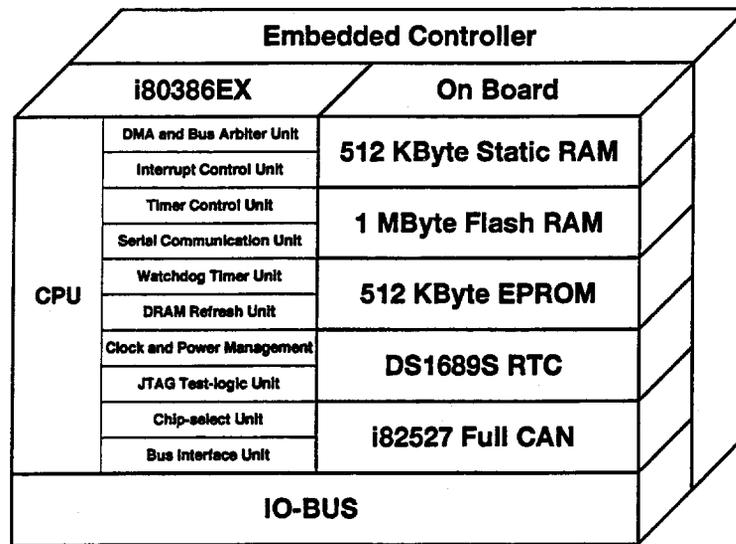


Figure. 2. Embedded Controller Schematic Diagram

To compensate the influence of the IDs on the beam in the storage ring a feed forward correction scheme has to be implemented. Every 100 ms it reads the gap information of all IDs and distributes the information to all power supplies (about 200) involved in the correction scheme. In this approach the ID compensation master takes the gap information of the IDs connected by separate 100 Kbit CAN segments. An additional 100 Kbit CAN segment will be used to distribute the gap information to the IOCs controlling the power supplies involved.

The chosen CAN interfaces at the VME level control two CAN segments and are capable of mapping automatically a specific CAN read object to a CAN write object on the other bus segment. Using that feature the CAN interface forwards objects from the 100 Kbit CAN segments to the 1 Mbit segments without wasting CPU resources on the IOCs. The embedded controller for each power supply holds a experimentally-determined correction table to alter the setpoint with respect to the GAP information. The deterministic message transfer time combined with the multicast capability of the CAN bus allows an implementation where the time needed for distributing the gap information does not depend on the number of power supplies involved.

II. CAN Communication

Within the BESSY II control system the CAN field bus will serve a variety of tasks: In addition to the 'normal' device control it will interface underlying intelligent subsystems (like measurement PCs, PLCs), can be used for downloading configuration data and possibly object code to the embedded controllers, carries the ID compensation data and can be

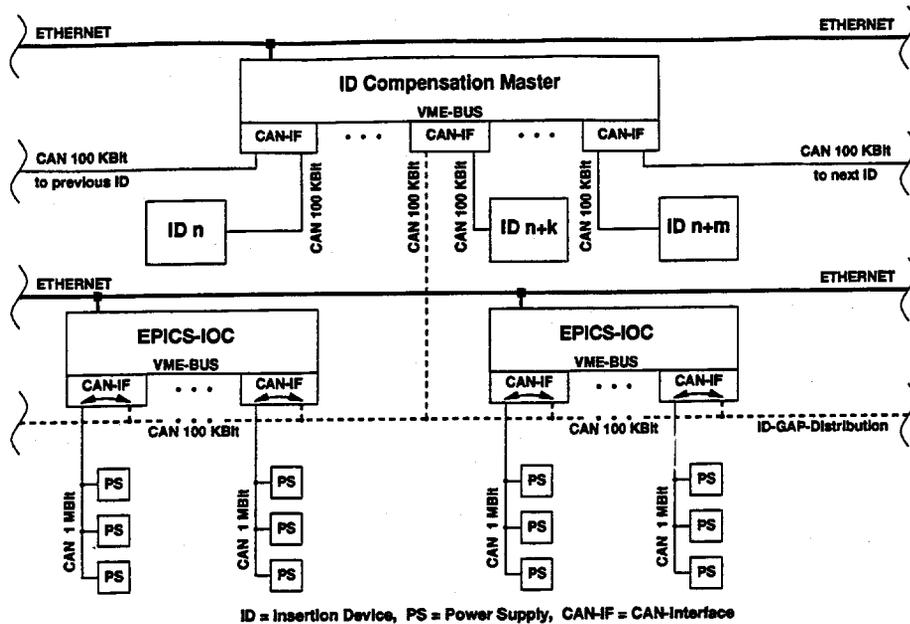


Figure 3. ID Compensation Hardware

used for synchronization of IOC tasks. These different communication tasks require different protocols to be used in parallel on one CAN segment.

The CAN interface has to be implemented on three platforms: The IOC (Motorola MVME with VxWorks/EPICS and commercial 68000-based CAN cards), the embedded controller (i80386EX without OS with a single on-board CAN chip – see fig. 2) and a PC used for measurement and debugging (80x86 using DOS and an AT bus CAN card).

A. SCI - a Protocol Independent Data Link

SCI, the *Simple CAN Interface*, is the specification of a library that embodies a standard for accessing CAN interface cards independent of the actual hardware. Programs that use this library to access the CAN field bus will be portable between different CAN cards and even different operating systems. SCI is intentionally kept simple and it largely represents the data-link layer of the OSI model.

SCI is prepared to handle several CAN ports which are distinguished by port numbers. The ports may be realized on one or more interface cards, not necessarily from the same vendor. Different interface cards may have to be accessed through different drivers, but these differences are hidden by SCI.

SCI is capable of multithreading. In order to facilitate this function, each thread that opens the library is provided with a unique pointer. This pointer, which is a parameter for all SCI functions, is used to distinguish different threads. The properties of a COB¹ – identifier, data length, timeout and type, where type can be one of read, write, remote-read and remote-write – are defined when it is initialized. SCI then returns a pointer to an internal structure that is allocated and initialized per object. All other functions use this `sci_object` pointer as a handle to a certain object. Read and write functions exist in order to transfer data to and from CAN objects.

Reading can be done in three basically different ways: *Immediate read* gets whatever data is stored on the CAN hardware for a certain object. *Read with timeout* gets new data or – if there is no new data – waits for the COB to arrive. *Install a callback function* that is called when new data arrives for an object is the asynchronous mode to retrieve data. On multitasking operating systems this callback is called from within a signal handler.

¹Communication Object, message with a unique tag.

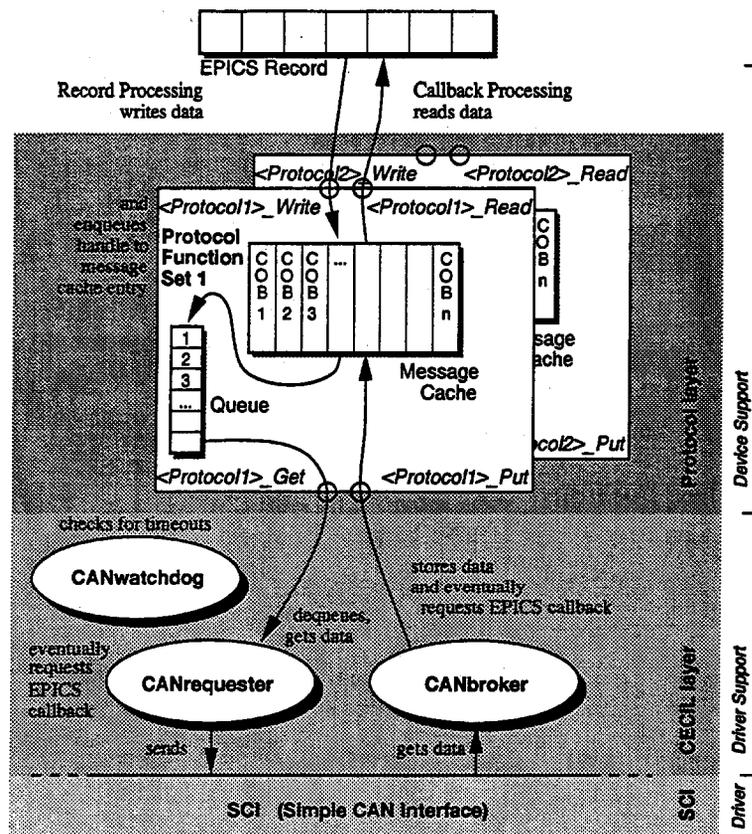


Figure. 4. CAN Interface Structure

B. CAN Integration into EPICS

B.1 Concept

The requirements of portability and reusability are fulfilled best by a layered interface structure. This allows the parallel use of different protocols through well-defined interfaces into the protocol layer and collects the platform dependencies in another layer. Thus the support for a protocol can be implemented fully independent from hardware and OS properties. Fig. 4 shows the task level structure of the CAN interface in front of its layer structure.

Above the low level CAN library SCI is the OS/hardware dependent CECIL (*Common EPICS CAN Interface Layer*). The tasks and functions in this layer set up the data transfer between the CAN segments (through SCI) and a COB oriented cache structure which is an interface to the next layer handling the different protocols. This layer consists of a small set of functions for each protocol to be run over the CAN bus. These functions include initialization, reading and writing data in protocol specific format (the upper interface) as well as reading and writing data in SCI conformal format (the lower interface).

EPICS device support writes and reads data to/from the cache using the upper interface functions of the appropriate Protocol Function Set. For output records the write request is fed into a queue. The CANrequester dequeues it, gets the data through the lower interface of the protocol layer and calls SCI to send the message. For asynchronous record processing an EPICS callback may be requested. On an incoming message SCI calls the CANbroker task. The CANbroker gets the COB, stores it through the appropriate protocol function and eventually requests an EPICS callback, which gets the data out of the protocol layer and moves it to the EPICS record. To avoid any mutual influences between field bus links there is one three task ensemble for each CAN segment connected to the VME.

B.2 Protocols

It is intended first to implement the following three protocols:

- *flatCAN*, a very basic protocol for reading and writing raw messages over the CAN bus, will be used to interface commercial CAN nodes that do not support our message transfer standard.
- *lowCAL* is a subset of the CiA² CAL (*CAN Application Layer*) [1]. CAL is a huge protocol suite defining an open CAN environment, including services for message transfer, network management, on-line address distribution as well as remote configuration of certain layer parameters. Our lightweight *lowCAL* subset defines only the message specification as well as the encoding rules for the variable types used in BESSY II device control. This will be the protocol used for communication with the embedded controllers.
- *CANal (CAN arbitrary length)* specifies two protocol variants for sending data of arbitrary length such as tables, configuration data or object code: A multicast transfer which is not confirmed and a point-to-point transfer which is confirmed and allows partial re-transfer of data.

C. Status

SCI has been implemented on an HP9000/743 (HP-RT) and on an MVME162 (VxWorks 5.2). It is being implemented at the moment on the i80386EX Embedded Controller and the 80x86 (DOS).

The CAN-EPICS interface has been designed with the protocols, layer interfaces and data structures being defined already. We are beginning to implement it under VxWorks and in parallel to develop the embedded controller software.

References

- [1] CAN in Automation International Users and Manufacturers Group e.V.: *CAN Application Layer for Industrial Applications (CAL)*, CiA e.V., Nürnberg, 1995.
- [2] G. v. Egan, T. Stein, J. Rahn: *Object Oriented Device Control Using the CAN Bus*, presented at ICALEPCS '93, Berlin, 1993.

²CAN in Automation International Users and Manufacturers Group e.V.

A Cost-Effective Way to Operate Instrumentation Using the Motorola MVME162 Industry Pack Bus and HiDEOS

J.B.Kowalkowski
Advanced Photon Source
Argonne National Laboratory

ABSTRACT

The HiDEOS software package has been implemented at the Advanced Photon Source to run the Motorola MVME162 embedded controller together with a variety of industry pack modules. The HiDEOS-based MVME162 controller acts as an intelligent second slave I/O processor in a VME crate. The implementation consists of vxWorks running the EPICS control system package with HiDEOS running the MVME162 in stand-alone mode with no additional operating system. Use of Industry Pack modules provides an inexpensive way to control a large number of serial ports, add a GPIB controller, or add a large number of analog and digital I/O modules at a reasonable cost. HiDEOS only utilizes software available in the public domain, allowing users to operate the MVME162 and supported Industry Pack hardware at no additional software cost. The HiDEOS package has been interfaced and integrated into the EPICS control system toolkit and handles the backplane communication with tasks under vxWorks and adds an object-oriented model to EPICS device support. Being object oriented, HiDEOS contains Industry Pack control classes that make it easy to add support for new Industry Pack modules.

INTRODUCTION

The software package HiDEOS [1] is currently being used at the Advanced Photon Source for GPIB, digital/analog input/output and serial communication to instruments. A library was added to create a standard interface to the EPICS control system toolkit [2]. The primary target is the Industry Pack (IP) Bus available on the Motorola MVME162 because of the low-cost IP modules available to the user. HiDEOS was chosen as a development platform because of its ability to hide complex issues, such as the IP Bus Controller and interprocess communications from the device driver creator, and to distribute processing among several processors.

One of the objectives of this development effort was to create generic support routines, accessible through the higher-level EPICS control system, that perform transactions with an instrument on a serial or GPIB link. With this set of tools, users can add new instruments and control them through EPICS without writing any embedded application programs. Existing scripting language interfaces to EPICS can interact with the existing embedded system.

A VME-embedded system main processor can be under substantial load in a large accelerator application. Adding a number of serial-linked instruments to the system can create a large additional load on the main CPU, especially if the instrument requires a complex protocol. A second goal of this development effort is to allow additional general-purpose, low-cost processors to perform the actual transactions to the GPIB and serial links. By allowing this type of system to be assembled, the main processor load, which is implementing most of the actual control algorithms, can be reduced.

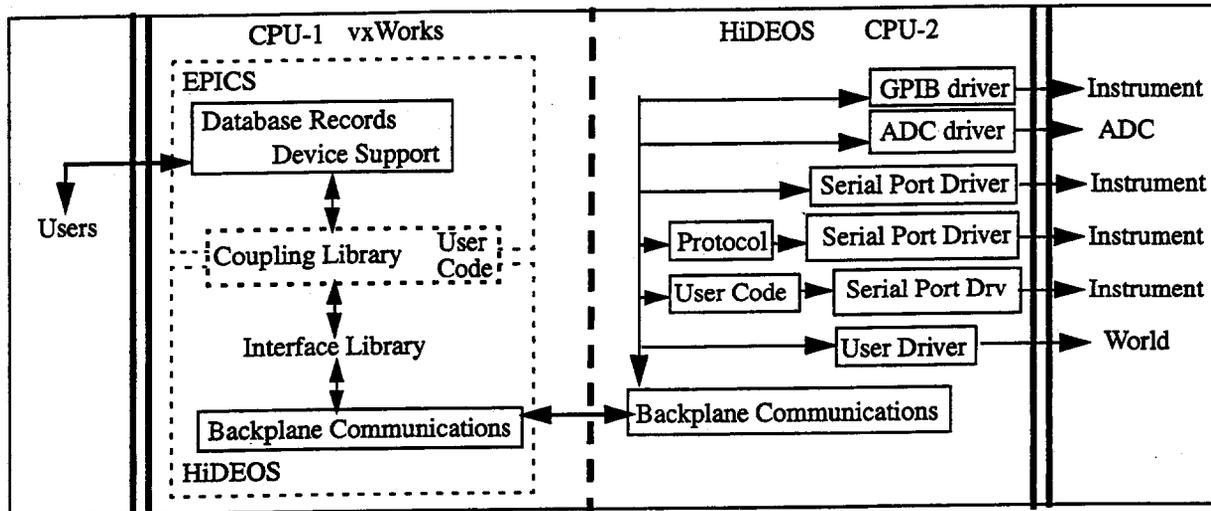
A third goal was to keep the cost of purchasing and running serial ports and GPIB connections low. The Industry Pack modules have helped fulfill this goal. The cost of serial ports on IP modules can be less than \$50 per serial port. A GPIB interface module can be added for \$200. By using HiDEOS, drivers can be provided to users with no licensing fees. A site running EPICS and owning a vxWorks license can easily add the HiDEOS subsystem for basically the cost of hardware plus setup time.

ARCHITECTURE OVERVIEW

The control system software includes three major pieces: EPICS, vxWorks, and HiDEOS. With the implementation discussed here, EPICS and vxWorks reside and operate on the main embedded system processor and the main portion of HiDEOS resides and runs on a second slave processor in a VME crate. The main control procedures are written in

EPICS which runs under vxWorks. EPICS handles the interaction with remote users and the running of high-level control algorithms. The job of HiDEOS is to operate the IP modules, run protocols or user algorithms and operate instruments connected to the IP modules. It also has the responsibility of running the backplane communications between processors. A portion of HiDEOS runs on the main processor under vxWorks and works together with EPICS to communicate with the slave HiDEOS board.

FIGURE 1. General Architecture



Users interact with EPICS components without any knowledge of the HiDEOS portion and they usually sit on an ethernet LAN away from the VME control crate. As shown in Figure 1, the crate contains two processors: the main controller and the HiDEOS slave MVME162 board. All communications between the two CPUs is made through the HiDEOS package. EPICS allows device support routines to be attached to control system primitives called records. A record can contain an arbitrary number of device support routines of which one is active at a time. The EPICS database designer generally chooses the device from which the primitive is to get data. The user is generally free to implement device support in any fashion; there are really no hard rules on implementing device drivers. HiDEOS has a fairly rigid method defined for writing drivers or software modules which communicate using messages. The coupling library sits between the two systems and serves as a simple place for users to add code to the system. A simple generic EPICS device support routine communicates with the coupling library. The coupling library uses the HiDEOS interface library to push messages into HiDEOS and get results in the form of messages. HiDEOS backplane communication tasks reside on both CPUs and exchange messages. CPU-2 receives commands in the form of messages and distributes them to the running HiDEOS device drivers. This is a very flexible architecture, allowing users to choose the location from which an instrument will be operated. User code can exist as part of the coupling library, where the main processor executes it under EPICS, or as HiDEOS software modules on the slave processor. In addition, users can attach to existing HiDEOS drivers and utilize their services. Figure 1 shows a series of drivers that could be running in a system.

INDUSTRY PACK MODULE SUPPORT

The primary supplier of IP modules for the APS has been Green Spring Computers. A few have come from Systran Corporation. An MVME162 can accommodate up to four IP modules on the IP bus. The HiDEOS package comes with drivers for the following modules:

- IP-Serial - 2 serial ports (RS232/RS422/RS485)
- IP-488 - GPIB
- IP-OctalSerial - 8 serial ports, (RS232/RS422/RS485)
- IP-QuadSerial - 4 serial ports
- IP-PrecisionADC - 12 bit 16 channel ADC

- Systran-DAC128V - 12-bit DAC
- IP-16DAC - 16 channel, 16-bit DAC (under construction)
- IP-16ADC - 16 channel, 16-bit ADC (under construction)
- IP-ADIO - 48-bit digital I/O, 13-bit ADC, counter/timers (under construction)
- Systran-DIO - 48-bit digital I/O (under construction)

The IP bus standard defines an ID memory space which contains manufacturer and model codes. HiDEOS contains a database of all supported IP module make and model codes and provides a way to create a driver for each of them. During the boot process, HiDEOS probes the ID space of each IP bus slot on the MVME162. For each module that is found, the make and model codes are compared to those in the database. If a match is made with a database entry, then a HiDEOS driver is created and given a name that includes information about the slot where it exists. This discovery process allows HiDEOS to be self-configuring for IP modules; a driver will be started for each of the services that needs to be available on the IP bus.

All IP drivers are written under HiDEOS using C++ and the HiDEOS methodology of task writing. Application developers required to create IP device drivers are given several tools through HiDEOS to simplify their effort. The driver creator must conform to the HiDEOS IP module manager database rules in order to have the driver automatically started during the boot process. HiDEOS provides a tool for slot management which allows the user to easily manipulate the IP bus. Some common features are the ability to enable and disable interrupts from an IP slot and find the memory addresses of the different IP memory spaces.

EPICS INTERFACE

All HiDEOS drivers are named with a character string. The character string in the case of an IP module driver is prefixed with a single character indicating the slot in which the module exists. A typical example is the GPIB module. If the user installs a GPIB IP module in slot B of the MVME162, then the string "b-GPIB" will refer to the driver that operates the module. Many of the IP modules have several channels or ports, such as the quad-serial. In this case the string must include the port or channel using the C language style array index operator. If the user wants to communicate with port 3 of the quad-serial IP module installed in slot C, then the name string will be "c-Serial[2]" (indexing starts at zero). The current implementation of HiDEOS requires the user to know the card in which the HiDEOS driver resides. The main processor is zero, the first slave processor is board one. The reason for this is that a user must know exactly where an instrument is attached. To summarize, the HiDEOS IP naming mechanism requires the user to know up to four simple pieces of information: the card in which the driver exists, the slot in which the module exists, the name of the module and the port or channel of the module (if needed).

The naming of a user-supplied HiDEOS process or protocol driver differs slightly from the above scheme. This type is not associated with an IP module, so no slot character is required. Generally these processes do not have multiple ports or channels, so that portion is also not required. Usually only two pieces of information are required: the card where the HiDEOS process is running and the name of the process.

When EPICS database records are created for use with a HiDEOS driver, the database designer fills in the card number where the HiDEOS driver exists and the above name information in the optional parameter field. The coupling library will parse the parameter field and verify the existence of the HiDEOS driver during EPICS boot time.

EPICS is written in C and HiDEOS requires C++. A problem with vxWorks 5.2 is that it does not supply a loader capable of loading C++ object modules correctly. HiDEOS provides a wrapper on top of the vxWorks ld() function called ldpp() which correctly loads C++ modules. The functionality added to ld() is the ability to run global constructors and supply the default new and delete operators.

The coupling library discussed earlier hides many of the ugly details of EPICS device support and HiDEOS driver support from a user in a base class. The user is required to create three functions: a start I/O, a complete I/O and an initialization function. The start I/O is invoked by EPICS when a record requires an I/O transaction to be started on a HiDEOS driver. The complete I/O function is invoked by EPICS when the HiDEOS driver completes the transaction started earlier. The initialization function is invoked when the EPICS record is being initialized. The HiDEOS/EPICS

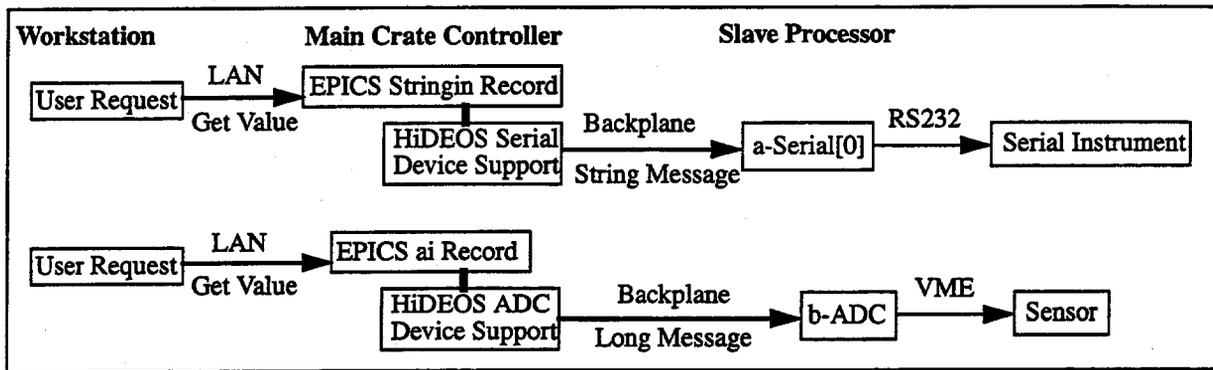
interface package includes the coupling library, along with code which does string transactions, ADC/DAC I/O, and GPIB transactions using various EPICS records.

APPLICATIONS

User applications developed so far fit into three categories. The first category includes applications which use existing EPICS and HiDEOS support software. All that is required is to create EPICS database records and specify the parameters correctly. The second category includes applications requiring code to be added to the EPICS/HiDEOS coupling library. This type of application will usually communicate with existing lower-level HiDEOS drivers. The third category requires a HiDEOS driver to be written and run on the second processor. For the purpose of this paper, all applications will require EPICS database records. Some of the more complex applications fit into categories two and three.

Many instruments and sensors can be operated using the first application category. These are important because the user does not need to write any code. An example application of this type is reading from a supported ADC using an EPICS analog input record or writing to a DAC using the analog out record. Another example is writing and reading strings directly to a serial link through EPICS string out/in records. Generally only the simplest of serial-controlled instruments will be able to make use of this method. GPIB devices can be operated in this fashion using the EPICS GPIB library. Figure 2 illustrates two simple example applications which fall into the first category.

FIGURE 2. Simple Applications



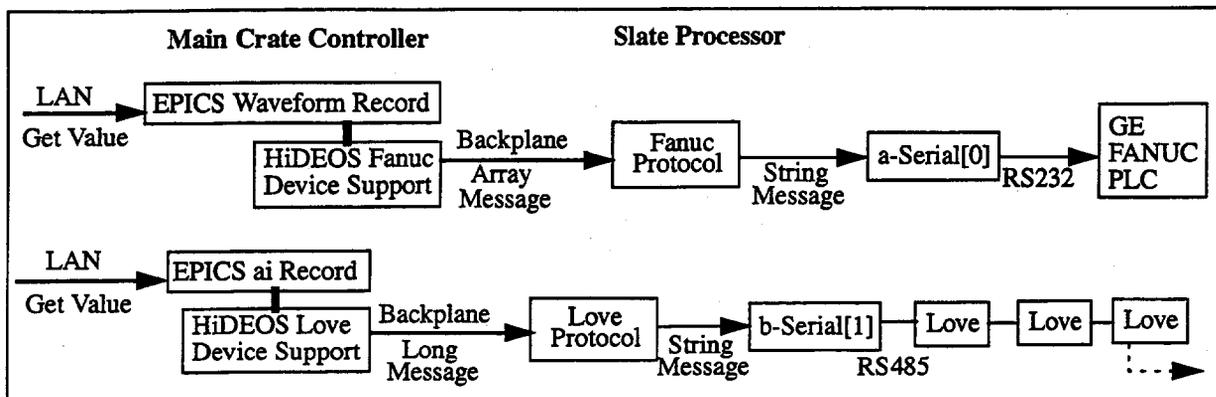
Instruments or IP modules which require simple transactions to operate and return status information fit into the second category. Here a simple piece of code must be added using the coupling library which will in turn be added to EPICS device support. Simple serial-controlled instruments will usually require delimiters to be added to outgoing transactions and require a time-out; incoming transactions will require delimiters to be stripped and data interpreted. This category usually requires a small, simple program to be developed and added to the EPICS/HiDEOS support library which exists under the coupling library of Figure 1.

Multi-function or multi-state instruments such as digital voltmeters, motor controllers, temperature controllers, and PLC usually require a complex protocol to operate. These instruments can be attached to IP modules with existing drivers such as the serial highway or GPIB. A new HiDEOS program will sit between the EPICS/HiDEOS coupling interface and the IP driver to actually operate the instrument. This type of configuration typifies the third application category and requires the largest code development effort. Examples of such a setup are the GE FANUC [3] interface and the Love temperature controller developed for the Experimental Facilities Division of the APS.

In Figure 3, the GE FANUC sends a string of characters representing the current state of the PLC. The existing HiDEOS IP serial driver accepts the string and passes it to a GE-FANUC protocol driver which deciphers it and decides whether or not to inform the EPICS control system of any state changes. The EPICS/HiDEOS device support procedure converts the data to EPICS-acceptable format and notifies other EPICS database records of change-of-state information. The GE FANUC provides data to the HiDEOS processor continuously at 19,200Kbps which would put a large load on the main processor if it were not configured in the two processor fashion.

The Love controllers operate on a RS485 multi-drop network. The HiDEOS Love controller protocol driver polls the controllers for temperature and status reading using the Love protocol. The data is converted to a simple transaction that can be sent to the EPICS/HiDEOS support module where EPICS database records are updated.

FIGURE 3. Bigger Applications



CONCLUSION

This development effort has demonstrated that HiDEOS can be integrated into an existing system. Interacting with HiDEOS using its interface library is straightforward providing the user has an understanding of the HiDEOS system. Addition of the coupling library was an important part of the development, as it allows easy access to commonly used pieces of both systems. The current complete system has a considerable number of important IP modules and sample applications developed, with more to be added. HiDEOS is explained in more depth in [4].

The learning curve for understanding the HiDEOS system and using its facilities can be high for a user not familiar with object-oriented programming and design in C++. Understanding each component which must be modified or added to EPICS and vxWorks start-up scripts is difficult. Like EPICS, the HiDEOS package cannot just be picked up and used without first understanding a set of fundamental principles.

ACKNOWLEDGMENTS

Thanks to Dave Reid, Josh Stein, Tim Mooney, and Greg Nawrocki of the APS for providing applications, hardware, and time for testing the initial development of the HiDEOS system and the EPICS interface library. This work was supported by the U.S. Department of Energy, Office of Basic Energy Sciences, under Contract No. W-31-109-ENG-38.

REFERENCES

- [1] J.Kowalkowski; "Home Of HiDEOS," World Wide Web URL: <http://www.aps.anl.gov/asd/controls/hideos/intro.html>.
- [2] W.McDowell; "Experimental Physics Industrial Control System (EPICS)," World Wide Web URL: http://www.aps.anl.gov/asd/controls/epics_home.html.
- [3] J.Stein; "Personnel Safety System Status Reporting Via EPICS at the Advanced Photon Source," these proceedings.
- [4] J.Kowalkowski; "An Object-Oriented Approach to Low-Level Instrumentation Control and Support," these proceedings.

Optic Data Link for Tevatron Synchronization System

A. Mason and V. Rytchenkov *

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510

Abstract

A fiber optic data link for the synchronization systems of the Fermilab accelerator has been designed, produced and tested. The link uses one multimode fiber in a previously installed cable. This link is capable of simultaneously delivering a 7.5 MHz biphase encoded clock signal and a 53 MHz RF signal to 30 service buildings distributed around the 6.28 km circumference accelerator ring. An accumulated time jitter at the far end of the link is less than ± 2 ns while a variation of the carrier or RF is about 300 kHz.

I. INTRODUCTION

The Fermilab Accelerator Timing and Synchronization System is comprised of four clock signals. Three of these are separate beam synchronous clocks (BS CLK) which are synchronized to the actual beam revolution frequencies (or equivalently, to the accelerating RF) of the Main Ring, Tevatron protons, and Tevatron antiprotons. The serial clock signal can carry timing markers, or "events", encoded in a biphase (or modified Manchester) code onto the 7.5 MHz (53 MHz RF divided by 7) carrier.

A serial data repeater system [1] has been used to accommodate distribution of each BS CLK. The repeaters are positioned at each of the thirty service buildings located throughout the 6.28 km Tevatron ring accelerator. A coaxial cable is used as a communication medium between the repeaters. The original coaxial cabling supporting the accelerator controls has existed for a long time. Recently there was an upgrade of the Control System, and fiber optic communication links have been implemented [2]. This allowed adding new links for the upgraded system and avoiding the disadvantages of a gradual degradation of the coaxial cable's characteristics.

A twenty-four fiber trunk cable with multimode graded index fibers (62.5/125 μm) was installed throughout the accelerator ring. Its full length is nearly nine kilometers, a mean length between two service buildings is about 300 meters. An optic data link for 10 Mbps biphase encoded data was designed and installed to replace an original serial CAMAC interface [4]. That link is split, each leg connecting half of the ring length using 15 repeaters. The repeaters used inexpensive optic transmitters and receivers at 820 nm. With an edge skew less than one nanosecond per repeater, the link containing 15 repeaters operated without error at the far end.

While this amount of jitter is acceptable in these cases, for timing synchronization purposes with new requirements of accuracy it is not tolerable. Besides, delivering of low level RF signals is also desirable. A new optic data link (ODL) has been designed which is capable of delivering a 7.5 MHz biphase encoded BS clock signal and the 53 MHz RF signal simultaneously on one fiber.

II. REALIZATION

To achieve time error of ± 2 ns, conventional optic data links should use a single mode fiber, expensive laser transmitters, high speed optic receivers, high speed retiming circuits in repeaters, and a stable carrier with frequency deviation less than 0.1%. Our solution is based on using of the existing multimode fiber cable and takes into account:

- large quantity of repeaters (30) and a relatively short distance between repeaters
- a relatively low data rate (7.5 Mbps biphase data with 15 MHz clock)
- a very small jitter (± 2 ns, relative to the 132 ns period of BS CLK)
- a large variation of carrier (up to 300 kHz variation of 53 MHz RF)
- use of a single fiber for both BS CLK and RF signal
- unipolar +5 V power supply.

To decrease accumulated time error, including slow pulse width distortion and fast jitter due to fiber and components of repeaters, retiming of the distorted signal is used in a number of the ODL repeaters. A clock recovery and data retiming circuit (CRDR) is utilized. Our choice is the Analog Device AD800-52 integrated circuit for 52 Mbps NRZ data. Output jitter of this circuit is about twenty degrees of the nominal data rate period (data pattern 1010...) and increases in any other patterns in accordance to decreasing of the number of transitions in the signal.

* Work supported by the U.S. Department of Energy under Contract No. DE - AC02 - 76CH03000.

Since this ODL is intended to deliver both BS CLK and RF CLK simultaneously we have to encode these signals with a code which can provide a minimal jitter while providing data retiming and RF CLK recovering in each repeater. This 53 Mbps NRZ encoded signal is obtained when the 53/2 MHz square wave stream (RF divided by 2) is modulated by the 7.5 MHz biphasic BS CLK signal. Each time slot of biphasic BS CLK signal is marked on the pulse stream as missing a level change during one RF period. A zero time slot is marked by missing two level changes. Such a signal has 50% duty factor and the maximum number of transitions that will provide minimal pulse width distortion and jitter.

The enco module which converts BS CLK biphasic data to 53 Mbps encoded data is installed ahead of the ODL at the F0 service building as shown in Fig. 1.

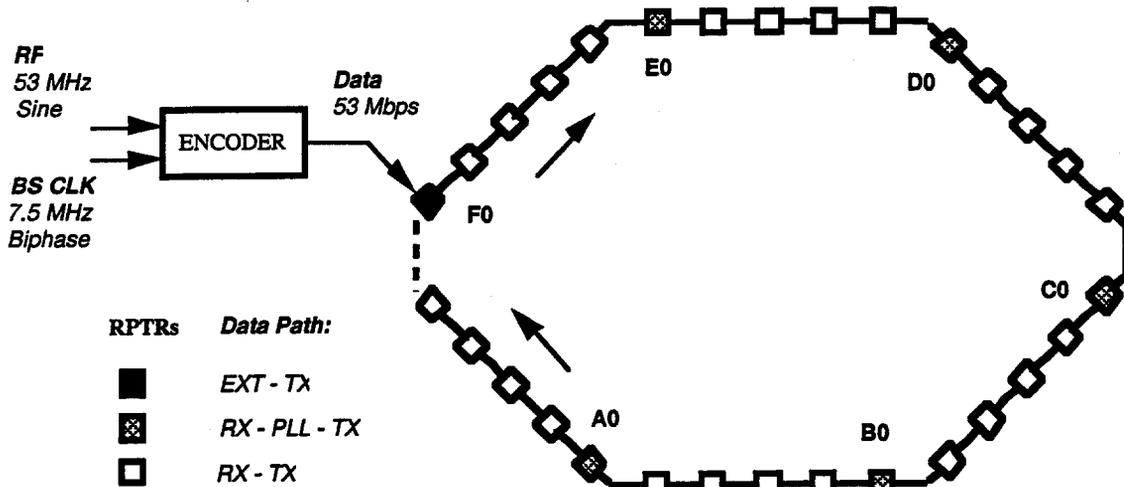


Fig. 1. Diagram of the optic data link.

An electrical 53 Mbps signal is applied to the first fiber optic repeater. It transmits the 53 Mbps optic signal at 1300 nm wavelength via the 62.5/125 μm multimode graded index fiber to other repeaters located in 30 service buildings. Each repeater provides regeneration of the optic signal, decodes the 53 Mbps signal to biphasic BS CLK and 53 MHz RF CLK, and drives electrical signals to the control system hardware. A few of the repeaters (zero-buildings in Fig. 1) provide retiming of the signal transmitted through the ODL to decrease time error. The number of such repeaters depends on properties of the fiber and the repeaters, and could be reduced.

A diagram of the repeater is shown in Fig. 2. Optic receiver RX and transmitter TX are AT&T ODL series II 125 Mbps 1361RX and 1261TX respectively [3]. They operate at 1300 nm wavelength and are compatible

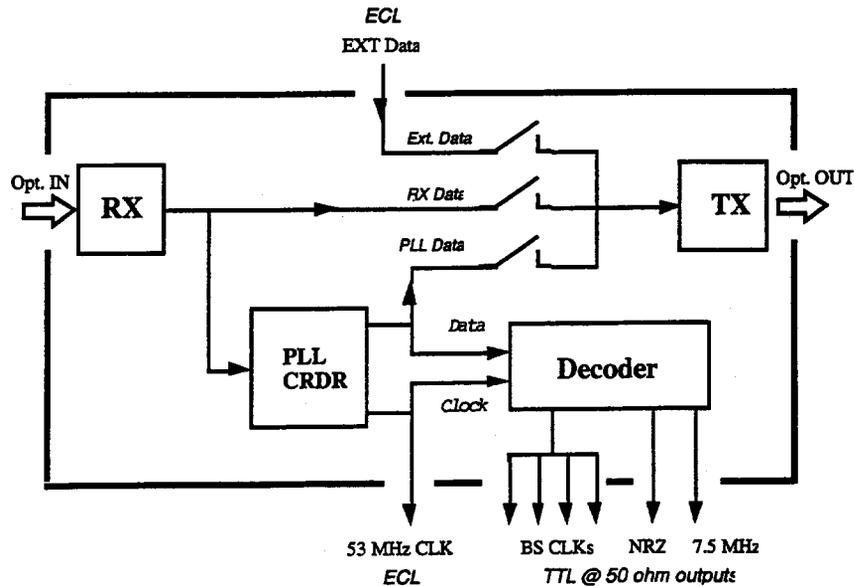


Fig.2. Diagram of the repeater.

with PECL 10HK logic. The repeater can transmit either external electrical 53 Mbps data, received data from RX, or retimed data from the PLL clock recovery and data retiming circuit (Analog Devices AD800-52) [4]. The recovered clock goes to a repeater output as the 53 MHz RF CLK. The BS CLK output signals come from the decoder circuit which is comprised of a programmable chip, ALTERA EPM 5032-16 (85 MHz toggle speed).

Additionally, the decoder supplies NRZ data obtained from BS CLK, and 7.5 MHz CLK signals for possible use in future control system hardware. All decoded output signals are compatible with existing hardware. The repeater uses a single +5 V power supply voltage, includes ECL 10HK logic and ECL to TTL translators. The printed circuit board of the repeater is compatible with the fiber optic repeater chassis used for existing optic links allowing simplicity of installation of the new ODL.

III. TEST RESULTS

The optic data link including 30 repeaters was produced, installed at the Tevatron accelerator and tested. Without PLL retiming the total time error including jitter and pulse width distortion is ± 6 ns. When PLL retiming was on in repeaters at the zero locations only (five repeaters) this error was 1.8 ns (peak to peak) for RF CLK output and 2.9 ns for BS CLK output. These results depend upon the frequency variation of the RF carrier which is about 300 kHz for the 53 MHz Main Ring RF. To obtain these results, the maximum bandwidth AD800-52 data retiming circuit should be selected.

Long term stability was measured during two months when the ODL delivered the real Tevatron BS CLK and RF CLK for D0 experimental facility. Data obtained by experimenters show that the drift of position of the RF CLK and BS CLK relative to accelerator beam was 2 ns and 3.2 ns respectively, i.e. less than for the same signals transmitted on the copper cable repeater system.

IV. CONCLUSIONS

The optic data link for the beam synchronization clocks and accelerator RF low level signals has been designed, produced and tested. It provides for delivering of 7.5 MHz biphas encoded BS CLK and 53 MHz RF to 30 service buildings throughout the Tevatron accelerator with time jitter less than ± 2 ns. This link uses existing multimode fiber cable installed at Tevatron. Such ODL can be used for three synchronization systems - Main Ring (MRBS), Tevatron (TVBS), and Anti-Proton (APTVBS).

REFERENCES

- [1] R.Ducar, "Tevatron Serial Data Repeater System", *IEEE Transactions on Nuclear Science*, vol. NS-28, No.3, June 1981, pp. 2301-2302
- [2] R. Ducar, "Fiber Optic Communications Links for the Main Ring Control System Upgrade", pp. (3)
- [3] "ODL 125 Series II Lightwave Data Link", *AT&T Data Sheet*, August 1992.
- [4] "AD800/802 Clock Recovery and Data Retiming Phase-Locked Loop", *Analog Devices Data Sheet*, rev. 0, 1994.

THE INTEGRATED HIGH SPEED COMMUNICATION SYSTEM FOR IHEP ACCELERATING AND EXPERIMENTAL COMPLEX

B.V.Prozin, D.U.Kisorzhevsky, V.N.Liamin,
V.P.Panov, O.V.Romanov, V.A.Ushakov
Institute for High Energy Physics
Protvino, Moscow Region 142284, Russia

Abstract

The combined system for collection, conversion and transfer of information based on high speed medium data transfer in IHEP is described. The feature of that approach is the usage of long and short distance radio-relay links for connecting computer networks to remote buildings and short general purpose locations.

I. Introduction

IHEP has rather extensive and growing computing base. However the rigorous realization of possibilities for IHEP was limited essentially by a lack of high-speed channels of computer connection with centers of science. It was determined to be essential to connect IHEP with basic points of access to global computer networks. The nearest points are in Moscow at a distance about 100 km. Possible modes of access are obvious: satellite channels, fiber-optical lines and radio-relay. For many reasons, both technical and economic, a radio-relay (RRL) connection with Moscow was selected.

II. Configuration

Owing to distance and low elevation, IHEP does not have direct visibility with any point in Moscow. (Direct line-of-sight communication would require 200 m antenna altitudes at both ends, impractical in this case.) Research showed that three retransmissions would be necessary. This approach is undesirable for many reasons, mainly due to organizational problems of equipment exploitation.

A successful solution was found by a careful choice of an installation site requiring only one re-translator between Moscow and Protvino (Fig.1). The choice was made to use the unique 300 m meteorological mast in the city of Obninsk. This location is 47 km from Protvino and 100 km from Moscow. From this mast direct visibility with any high building on territory of IHEP and with two points in Moscow is realized: a television tower "Ostankino" and a building of MSU (Moscow State University) (Fig.2). Both these locations in Moscow are large communication nodes for connection to global webs.

III. Engineering Realization

Researches were made as to the choice of parameters for a radio-relay station providing a reliable connection on the rather extended (100 km) segment, given the complicated conditions and the rules on frequency and power of transmitters in the Moscow region. Finally the project was realized using specialized (television) radio-relay stations, working at the frequency 7.5 GHz and at an emitted power of 0.5 W. Presence of a low noise receiver has allowed use of parabolic antennas of diameter 0.65 m; this small size has eliminated potential problems related to wind forces. The frequency is sufficiently low to avoid signal attenuation due to precipitation but is high enough to obtain a narrow rf beam with a small antenna. The potential channel capacity of this radio-relay station reaches 34 Mbits/sec (T3/E3).

For two years the RRL was operated at an experimental data rate of 2.048 Mbits/s (E1). For more than 95% of a one year operating period the daily average error level using the HDB3 code did not exceed 10^{-8} . At this time two channels of 2.048 Mbits/s are realized with a connection at one point with satellite communication channels to European networks. Through a node at MSU we have reached DESY, Hamburg via the "Raduga" satellite.

The radio-relay communications have been integrated with the IHEP computer network (Fig. 3). Multifiber optical cables connect the main buildings on the IHEP site. Thick ethernet is used for intra-building connections.

IV. New Applications

The operation of this communication circuit has pointed out the possibility of using radio-relay communication circuits not only for long distances, but also for short ones, comparable with the size of our accelerator. It has been observed that remote locations of the institute can be given high-speed communications to our computer center using RRL (Fig.4). The top balcony of the heating plant's chimney provides line of sight visibility from the retransmitter to all buildings in the UNK complex. Among these are experimental labs, with huge streams of an information, and monitoring systems and controls of the accelerator. Therefore we researched transmission of digital information at a frequency of 39 GHz for a

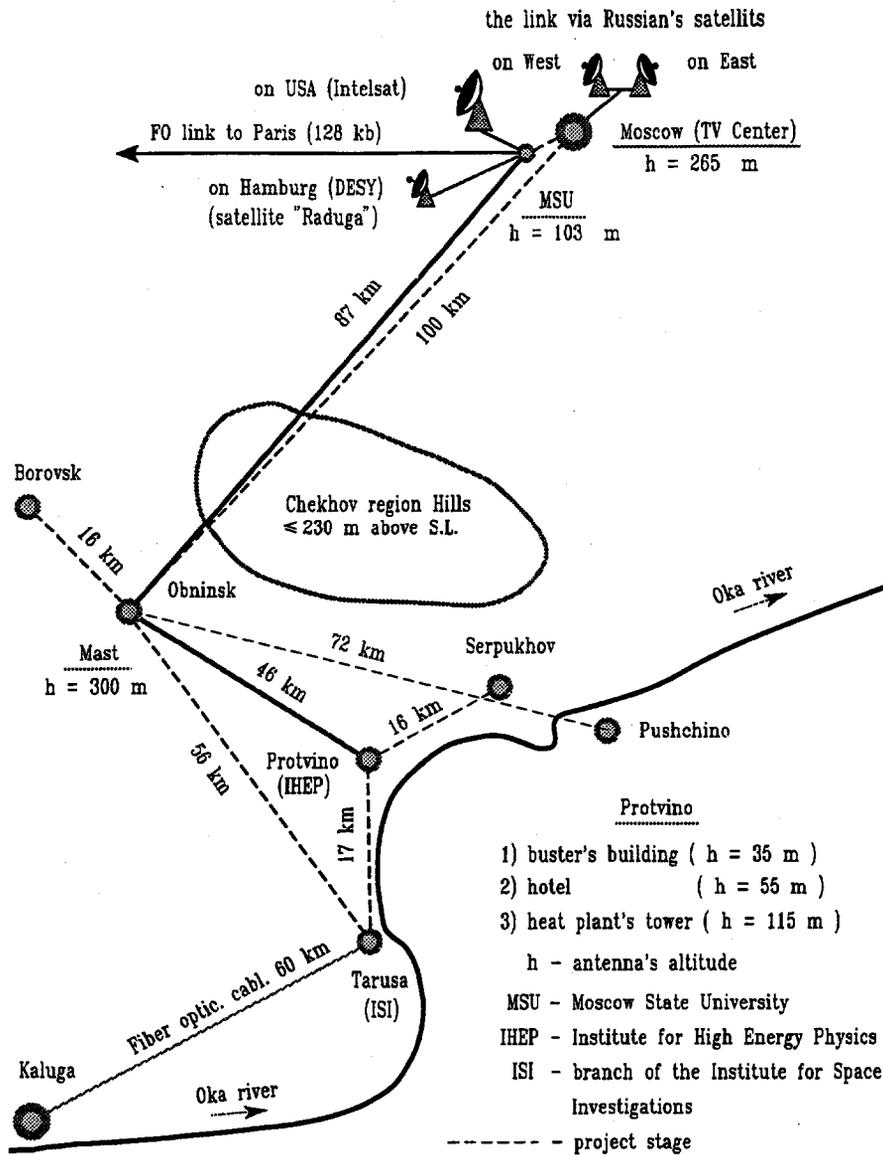


Fig.1 Radio relay network for Protvino (IHEP)

distance of 5 km, and a data rate of up to 140 Mbits/s was attained. It is important to note that the simplicity and speed of realization of the installation of RRL are attractive compared with other connection modes. We suggest this approach as a possible alternative to more conventional ones.

V. Development

We will continuously carry out work on growth of channel capacity of the high-speed channel connecting Protvino - Obninsk - Moscow (MSU) from 2 up to 8 and eventually to 16 channels at 2.048 Mbits/s. The application of radio-relay methods of communication for the purposes of a data gathering and control on the accelerator complex is in a stage of active research.

VI. Conclusion

Our experience here allows us to state that RRL-based methods provide a useful supplement to other communication techniques. This combination permits an inexpensive and fast realization of networking projects for extended plants such as accelerator complexes.

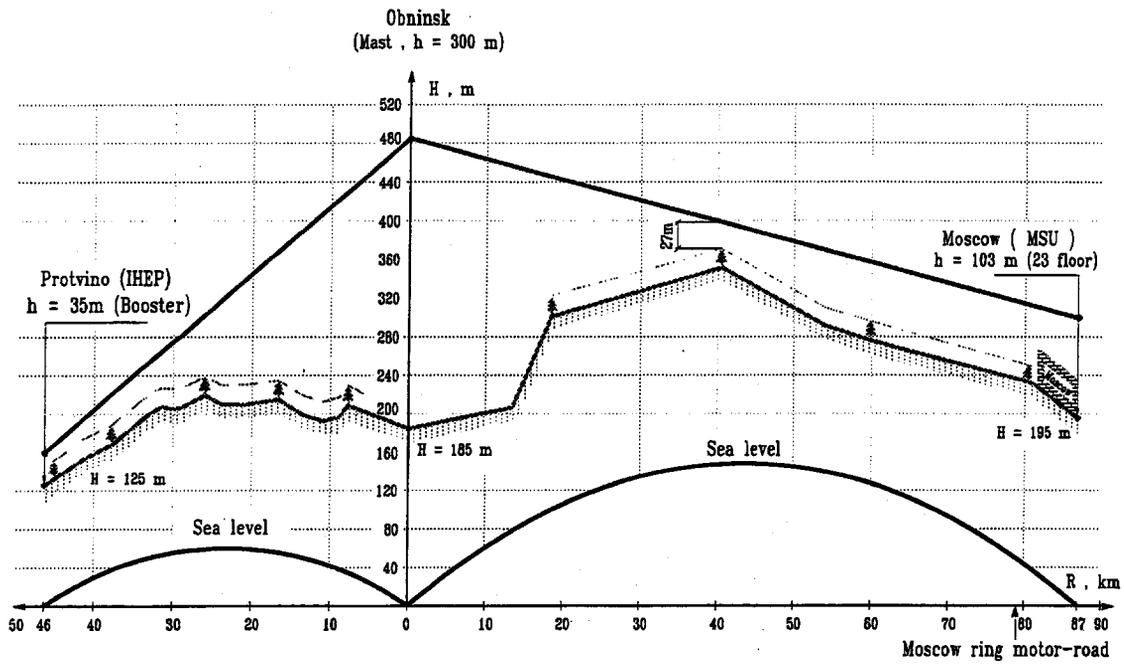


Fig. 2 The profile of trace Protvino (IHEP) - Moscow (MSU)

VII. Acknowledgments

We would like to thank our colleagues V.P. Sakharov and P.N. Kazakov for their help with the preparation of this report.

VIII. References

1. "Manual on Radio-relay Connection", Edited by N. A. Borodich 1981
2. "Digital Satellite Connection" J. Spilker 1984
3. "Application of PCM to Broadcast Quality Video Signals" V. G. Devereux 1974
4. "Problems of an Electromagnetic Compatibility of Services of Television Broadcast" V. A. Borovkov and N. G. Lokshin 1988

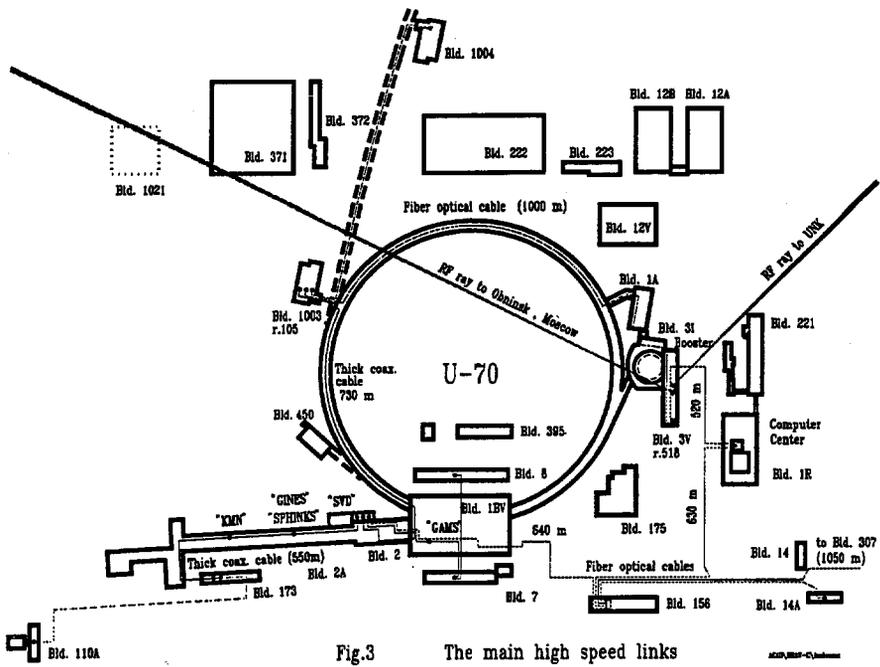


Fig.3 The main high speed links

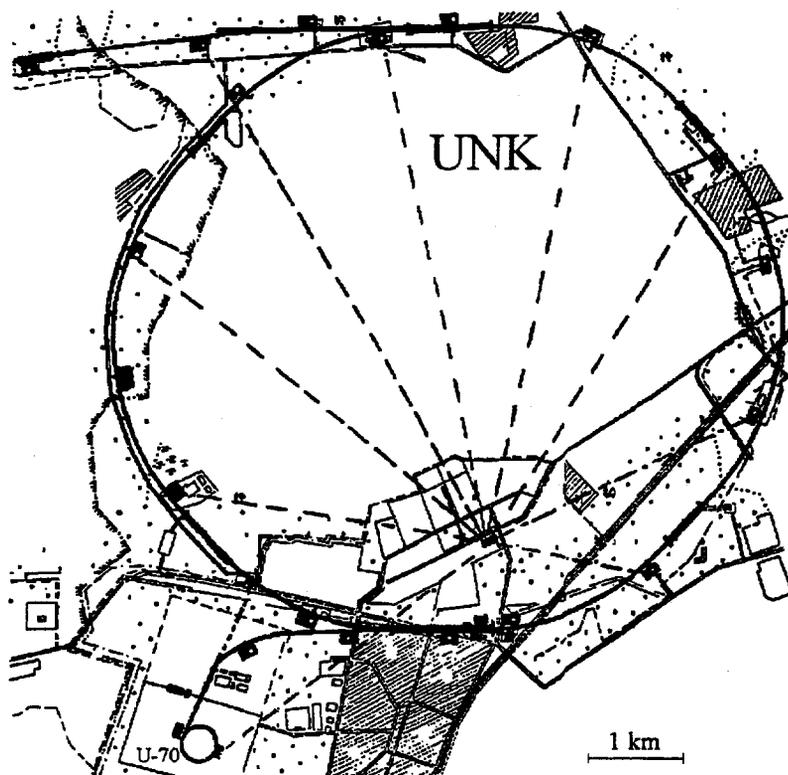


Fig.4 Layout of UNK with RF traces (dashed lines)

Figure. 4. Layout of UNK with Rf traces (dashed lines)

HIGH PERFORMANCE DSP DATA ACQUISITION ENGINE FOR ACCELERATOR INSTRUMENTATION AND CONTROL

K. Woodbury, J. Smolucha, C. McClure, K. Fullett, B. Chase
 Fermi National Accelerator Laboratory
 Batavia, Illinois 60510 USA

ABSTRACT

A low cost, high performance data acquisition and processing module is being developed incorporating the Super Harvard Architecture Computer (SHARC) DSP made by Analog Devices. The module provides independent ADC channels and high-speed, real-time data processing with pre- and post-trigger data collection. Using these modules, both the processing capabilities and number of ADC channels in the Fermilab Tevatron and Main Ring accelerator systems will be expanded. There are also potential applications in other accelerator instrumentation and control projects.

1. INTRODUCTION

The current Main Ring and Tevatron Accelerator Data Acquisition Systems provide over 3,000 multiplexed analog-to-digital conversion channels. The hardware used for this data collection network consists of 64-channel, multiplexed analog-to-digital conversion chassis, MADC-I (12 bit) and MADC-II (14 bit) [1],[2]. A CAMAC module [3] with embedded micro-processor is used to manage data collection, timing and data transfer to the ACNET control system [4].

This extensive network of channels provides a very effective mechanism for monitoring at both high and low frequencies. Low frequency monitoring (15Hz) can be provided essentially for all channels simultaneously. Higher frequency monitoring (66 - 100 KHz for MADC systems) can only be provided for one channel at a time. If multiple channels of high speed data collection are required this sampling rate must be divided between the desired channels.

The physics and accelerator operations staff expressed the need for more high frequency data collection channels in a single location and expanded capabilities for pre- and post-trigger data collection. In addition, continuous data processing such as FFTs and digital filtering was also desired.

Addressing these needs an R&D effort was started to design a new data acquisition system which could provide these added capabilities and complement the existing MADC system. This new system has been named the Data Acquisition System, version one (DAS I).

2. THE DAS SYSTEM

The essential components of this planned VMEbus-based system are shown in figure 1. All modules are 6U x 160 mm.

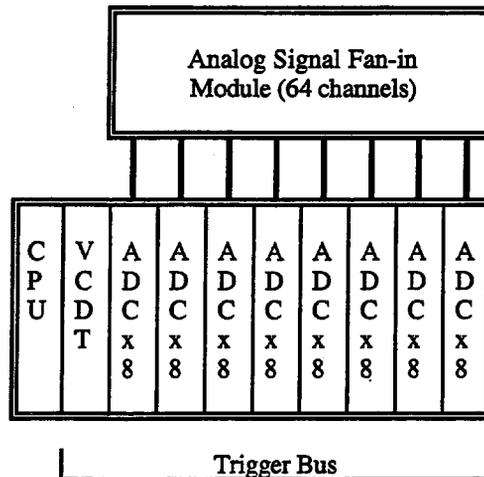


Figure 1. DAS VMEbus Chassis and Fan-in Module

- *VMEbus crate.* This VMEbus chassis is specifically designed with a low switching noise power supply and 6U x 160 mm transition modules in the rear. These modules provide connections to the Analog Fan-In Box (shown above), and the capability of bussing between P2 connectors.
- *VMEbus Clock Decoder and Timer (VCDT).* This module is a modified VMEbus Universal Clock Decoder (VUCD), which provides eight channels of delay timers combined with an interface with accelerator timing systems. For a more detailed description of the VUCD module see references [5],[6].
- *Trigger Bus.* This trigger bus provides a connection from the VCDT to all of the ADC converter boards, synchronizing data collection with the accelerator timing. This provides an equivalent connectivity to that provided by VXIbus TTL trigger bus specification. This can be implemented using the rear transition modules.
- *CPU / System Processor.* The current system processor of choice is a Motorola MVME162 - 68040 based processor running the VxWorks operating system. This processor has an on-board Ethernet adapter providing a network connection to the control system.
- *ADC module (under development).* Each of these ADC modules will have an on-board DSP and eight independent input channels with a minimum effective resolution of 14 bits. The planned conversion rate is 1-2 MHz. This frequency range was chosen as 640 kHz would be the minimum frequency to provide turn-by-turn data collection in our Booster and Anti-proton Source accelerators. Differential input is desirable but pre-trimmed differential amplifiers have not been found with the 10-20 MHz input bandwidths desired for this application. Anti-aliasing filters are also desired.
- *Analog Fan-in Module (under development).* This external chassis will allow for numerous user inputs which are then cabled to the J2 analog inputs.

The architecture of the DAS ADC module is described below.

3. DAS ADC ARCHITECTURE

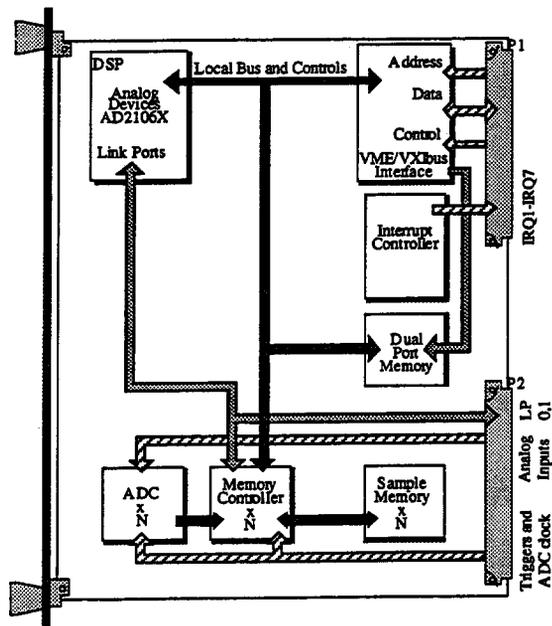


Figure 2. Simplified DAS ADC Block Diagram

The main components of the DAS ADC board are the following:

- Digital Signal Processor (DSP)
- Memory Controller and Sample Memory
- ADC Converters and Signal Conditioning
- VME/VXIbus Interface
- Event Trigger Inputs
- Dual-Port Memory

3.1 Digital Signal Processor

The primary use of the DSP is for real time data processing. Examples include gain and offset compensation, digital filtering and FFT computation. Additional functions provided by the DSP include controlling DMA transfers over the VME/VXIbus backplane and providing on-board process control as well. The DSP that was chosen is the Super Harvard Architecture Computer (SHARC) DSP developed by Analog Devices.

The SHARC DSP has a number of unique features. It has a maximum processing rate of 40 MIPS and 2-4 Mbits of internal program and data SRAM. The DSP was developed for high-speed radar data processing and was specifically designed to operate in a mesh architecture, where each processor can have a dedicated high speed communication link (Link Port) with up to six other processors. The Link Ports connect directly to an internal DMA controller as well. These Link Ports are one of the features which make the SHARC DSP well suited for the DAS ADC as described below.

3.1.1 Real Time Data Transfers

Real time data transfer to the DSP can be done using the Link Ports. These Ports can be used to transfer four bits of data at twice the clock rate of the DSP processor. This translates to 33 or 40 Megabytes per second depending on the speed of the processor. These Link Ports were originally designed for high speed processor-to-processor data transfer, but for our application they are ideal for real time data transfer from the ADC converter. Data transferred to the Link Ports feed directly through to the on-chip DMA controller which can be programmed to interrupt the DSP when sampling is completed. This has a direct application for data acquisition tasks requiring data collection and post-processing.

It should be noted that Link Ports 2-5 will be used for transferring ADC data, and Link Ports 0 and 1 are reserved for multi-processor configurations where the ADC board is used in conjunction with another DSP module, for example, in closed loop control applications.

3.2 Memory Controller

The core of the ADC module design is the Memory Controllers. These allow for various sampling and data storage modes and thereby allows the DSP processor to be dedicated for data manipulation and processing. The Memory Controller will also contain the ADC to Link Port interface which provides the real time data transfer path described above. Both the real time data transfer path and transfer to memory can operate simultaneously.

3.2.1 Memory Controller Sampling Modes

The Memory Controller is designed with an on-board sample counter which can operate in several different modes.

- *N Sample Mode.* A pre-determined number of samples are collected and transferred to memory. The Memory Controller operates in a count down mode where the number of samples is programmed into the counter register, and sampling is terminated on under-flow of the counter.
- *Continuous Sampling Mode.* In this mode, the ADC converter is sampling and data is transferred into memory continuously. The Memory Controller operates in a circular mode where the incoming data wraps for the depth of the memory, overwriting the oldest data.
- *Gated Sampling Mode.* In this mode, ADC converter data is transferred into memory only when an input gate signal is valid.

Data collection is started or stopped using trigger inputs or by access from the DSP or host processor.

A pre-scalar counter is also provided, which allows the Memory Controller to effectively reduce the sampling frequency by an integer modulus. The pre-scalar counter is eight bits in length.

Immediate data is always available from a sample register within the Memory Controller. This is especially useful for applications where slow monitoring is required.

After sampling is complete, data is transferred through the Memory Controller to either the DSP or host processor.

3.3 ADC Converters and Signal Conditioning

The ADCs have yet to be selected, however, a converter with a minimum 14 bit effective resolution and a minimum 1-2 MHz conversion rate is desired. The maximum ADC data width in the current design is 16 bits. The maximum ADC conversion rate is determined by the speed of the Memory Controller. Simulations up to 35 MHz have been made of the Memory Controllers.

It is hoped that at least eight ADC converters will fit on each module. However, due to the small size of the module (6U x 160 mm), four channels may be all that is possible.

Signal conditioning, as described in section 2, would include differential input amplifiers and anti-aliasing filters and input voltage protection.

3.4 VME/VXIbus Interface

This module will be a D16/D32 module which is compatible with both VME and VXIbus specifications. VXIbus Configuration and Control is not required for this application.

3.5 Event Trigger Inputs

The event trigger inputs, as described in section 2, provide a mechanism for data collection to be synchronized to accelerator activity.

3.5.6 Dual-Port Memory

The dual-port memory provides a standardized communication mechanism between the host processor and the DSP. It is used to transfer data and commands between the two processors and for issuing interrupts between them.

4. PROTOTYPE DEVELOPMENT

The current prototype development is being aimed at a high priority accelerator improvement project, the Anti-proton Source Beam Position Monitor upgrade project (PBPM) [7]. This project requires some of the exact features that can be provided by the DAS ADC board. Specifically, this project requires:

- 8 ADC channels per module.
- on-board Memory Controllers with sampling modes as described above (3.2.1) and
- on-board DSP processor.

This system does require some particular features that were not initially specified for the DAS ADC.

- *High Speed ADCs*, specifically 12 bits at a 31 MHz maximum sampling rate. This high rate is due to the super Nyquist sampling mode that is desired in this application.
- *Front Panel Analog Inputs*. Front panel analog inputs are required, as VXIbus specifications do not allow for analog signals to be fed through the P2 connector. Front panel I/O allows for a more optimized analog layout for the module.
- *Front Panel Triggering*. Front panel trigger inputs are included, in addition to the triggers provided from the VXI trigger bus (P2).
- *Front Panel Conversion Clock Input*. Front panel conversion clocks will be provided for the precise phasing of the ADCs as required by the system.

- *VXIbus Configuration Management Support.* As this module will reside in a VXIbus environment, it was preferred that the module should provide configuration management support.

Some of these features may be incorporated into the production DAS ADC as well.

5. ADDITIONAL FEATURES

Additional features that are being considered for the production PBPM ADC board include the following:

- *Link Port Support.* Although data collection rates from the ADC are too high for full, real time data transfer to the DSP., this communication link may still be useful.
- *NVRAM for DSP Boot Memory.* The current design requires the host processor to download the DSP code at boot time. On-board NVRAM provides a mechanism to download code only when changes are needed, speeding up the system boot process. This same memory could be used for non-volatile parameter storage as well, such as channel gains and offsets.

6. CONCLUSION

The development of these modules will provide a first step towards an overall enhancement of the general data collection capabilities for the accelerator control system. In addition, high speed versions of this board will have immediate application for current VME and VXI instrumentation projects.

7. ACKNOWLEDGMENTS

Our thanks to everyone who helped us to review and develop the DAS ADC specification, including the AD/Controls Engineering Staff. And special thanks to the Anti-proton Source Beam Position Monitor Upgrade team, for helping us to develop the prototype.

8. REFERENCES

- [1] K.C. Seino, "New General Purpose MADC", Controls Hardware Release no. 11.0, 1981.
- [2] K.C. Seino, "Multiplexed ADC Second Generation", Controls Hardware Release no. 72.1, 1990. Fermi National Accelerator Laboratory document.
- [3] J. Smedinghoff, "ACNET Fast Time Plot System", ACNET Design Notes no. 49.4, 1988. Fermi National Accelerator Laboratory document.
- [4] A. Thomas, et. al., "CAMAC 190 Module Multimode Buffered MADC Controller", Controls Hardware Release no. 26.4, 1988. Fermi National Accelerator Laboratory document.
- [5] K. Woodbury and C. McClure, "Integrated Finite State Machine and RF Timing Modules for VMEbus and VXIbus Instrumentation", proceedings of the 1995 ICALEPCS conference, *these Proceedings*.
- [6] C. McClure, K. Woodbury, et. al., "VMEbus Universal Clock Decoder (VUCD) User's Manual", controls hardware release, 90.01, 1994. Fermi National Accelerator Laboratory document.
- [7] K. Fullet, et. al., Anti-proton Source Beam Position Monitor Upgrade Project, DOE submission for an Accelerator Improvement Project, 1995.

INTEGRATED FINITE STATE MACHINE AND RF TIMING MODULES FOR VMEBUS AND VXIBUS INSTRUMENTATION

K. Woodbury, C. McClure
Fermi National Accelerator Laboratory*
Batavia, Illinois 60510 USA

ABSTRACT

A set of control and timing modules that provide a combination of a finite state machine (FSM) interface to the accelerator clock systems and RF resolution timing have been developed. These modules provide external process control and synchronization with accelerator events. Designed for both VMEbus and VXibus platforms, these devices provide an integrated timing resource that has been utilized by various distributed control systems at Fermilab.

1. INTRODUCTION

Much of the timing and synchronization of the accelerators at the Fermi National Accelerator Laboratory complex is done through the use of global clock-timing systems; primarily the Tevatron Clock (TCLK) and Beam Synchronous Clocks (BSCLK). These clock signals carried on serial data links consist of a carrier (TCLK = 10MHz, BSCLKs = approximately 7.5 MHz, based on the RF frequency), onto which are encoded 8 bit events. Bi-phase Mark (modified Manchester) encoding is used. Events are decoded off the data stream and delays timed using the carrier. Beam synchronous clocks exist for almost all of the major accelerators at Fermilab.

In addition, primary machine parameters, such as the Main Ring and Tevatron accelerator dipole magnet currents, are transmitted on another serial link called the Machine Data (MDAT) link. MDAT data frames are also encoded using Bi-phase Mark, however, no continuous carrier is provided. Data frames on the MDAT link are 24 bits in length, including an 8 bit data frame type identifier and 16 bit data value.

These clocks are distributed throughout the accelerator complex via a system of repeaters and fan-out modules, and are used by a wide variety of control and timing modules and systems.

2. UNIVERSAL DECODING RESOURCE

The initial motivation for this project was the pending retirement of the Unibus Clock Decoder [1]. This module, which resided in a PDP-11 computer, was responsible for receiving and storing Tevatron Clock event data. Data received was then distributed over Ethernet for transfer to the accelerator control consoles. For compatibility with previous system development efforts, a VMEbus replacement for this module was desired.

For the VME/VXibus system designer there was also the desire to use a single module to receive and decode the various clocks described above. This would, of course, reduce the amount of crate space used, as well as eliminating the need for managing module inter-activity.

In response to both of these requirements, a universal decoding module was designed. This first module was most aptly named, the VMEbus Universal Clock Decoder (VUCD) [2].

3. SYSTEM ARCHITECTURE

The VUCD can decode and process the TCLK, a BSCLK, and MDAT simultaneously. The overall module architecture is shown in figure 1 below.

* Work supported by the U.S. Department of Energy, contract No. DE-AC02-76CH03000.

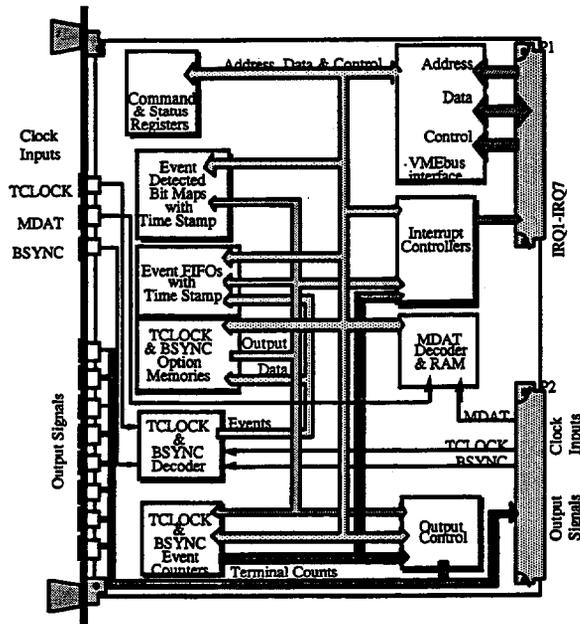


Figure 1. VUCD Block Diagram

The major components of the module can be divided into two functional subgroups; *Event Reporting*, with features similar to those found on the Unibus Clock Decoder and *Process Control*, which is primarily used for control system and instrumentation applications.

Event reporting features include:

- Event FIFO memory and free-running time stamp counter with programmable time-base (1 KHz - 1 MHz).
- Two commercial interrupt controllers (MX68C153) used for immediate event interrupts.
- One custom, FIFO-based, interrupt controller, which is preceded by event scalars. This controller is used for generating interrupts after a programmed number of event occurrences.
- MDAT memory, which hold the current machine parameter values.
- Two independent event bit-map memories, one for TCLK and one for a BSCLK. These are 16 bit x 16 bit matrices which record all incoming event occurrences. Automatic clear on read and time stamp support are also provided.

Process control features include:

- Two independent RAM-based finite state machines (FSM), one for the TCLK and one for a BSCLK.
- Output control logic for processing FSM outputs.

3.1 I/O Configuration

The clock inputs are taken either from the front panel or the rear P2 connector of the card. Outputs generated from the TCLK or the BSCLK (shown as BSYNC), can also be presented to the front panel or to the P2 connector. Outputs are predominantly used for external instrument timing or triggering.

All outputs on the P2 match VXibus TTLTRGn* assignments and only require output driver replacement for specification compatibility.

3.2 RAM-Based Finite State Machines

The core components of the module are the two RAM based finite state machines. Figure 2 shows the basic feedback mechanism which is used for state machine operation.

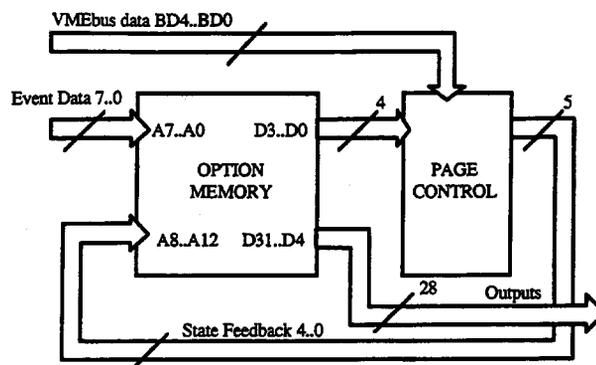


Figure 2. RAM-based Finite State Machine

Each FSM consists of two elements: the Option Memory (FSM RAM) which holds a list of immediate actions to be completed when a particular clock event is decoded and the Page Control Logic which contains the state register. When a particular event is decoded and all programmed actions are completed, the new state, defined by the lower four bits of the active memory location, is placed in the state register. A change in the state register value forces the Page Control Logic to point to a different section of memory and therefore an entirely different response configuration for the incoming clock events. The FSMs for the TCLK and BSCLK operate independently from one another.

As shown in Figure 2, the VME/VXIbus can force the FSM to a particular state in between clock events. This capability can be used to configure a FSM with a hold state and having the exit from the hold state under program control. Additional control of the FSM is also provided by allowing the VME/VXIbus to suspend event input into the FSM. VME/VXIbus actions are arbitrated to avoid collisions with clock events.

Note that there is an extra line coming from the VME/VXIbus interface into the page control. This allows for programmed transition into another state machine stored in a different bank of memory. This effectively doubles the number of possible states from 16 to 32.

These FSMs can operate very rapidly as TCLK event spacing can be as little as 1.2 μ s.

For a more detailed example of a RAM Based FSM see reference [3].

3.3 Event Triggered Actions

Event triggered actions can either occur automatically or are programmed into the FSM RAM.

Automatically occurring actions include the following:

- Marking the event in the event bit map.
- Updating system status.

Programmed actions include the following:

- Writing the event and current time stamp into the FIFO memory.
- Resetting the time stamp counter.
- Counting the event and automatically generating an interrupt on terminal count (often used for process scheduling).
- Generating an immediate interrupt.
- Generating an output pulse.

- Setting or re-setting an output level.
- Changing FSM state as outlined above.

3.4 MDAT data storage

All current machine parameter values received from MDAT are stored in a dual port memory for immediate access.

3.5 Combined FSM Output

As described above, the two state machines, operate independently from one another. The FSM output control bits, however, feed a common output logic block. This provides a mechanism for providing RF synchronous outputs following a pre-determined TCLK event sequence.

The BSCLK resolution is 132 ns (RF frequency / 7). To provide RF resolution timing two more modules were developed as described below.

4. RF RESOLUTION TIMING SUPPORT

4.1 VXI-UCD

The first module to be built with RF resolution timing was a "C sized" VXIbus card called the VXI Universal Clock Decoder (VXI-UCD) [4]. The core of VXI-UCD is identical to the VUCD, however, there have been some significant features added to the board. Among them are:

- 8 channels of independent event delay timers (time-base options: TCLK, BSCLK, VXIbus CLK10, External to 15 MHz).
- Added interrupt based on programmed MDAT frame type arrival and
- RF based Pulse Pattern Generator (PPG).

4.1 RF Pulse Pattern Generator

The PPG consists of two elements, a 64K x 8 pattern memory and a programmable address counter which is used to index into the memory. Each bit position in the memory can be thought of as a column in a table which is either a "0" or a "1". Each of these columns can be directly mapped to one of the outputs. Pattern generation is initiated by programming a start address in the address counter and an end address in the address counter. Once a trigger is received by the address counter the pattern that is programmed in the memory is then presented to the output(s).

The address counter can be programmed to play a selected number of pattern cycles, or can free-run until triggered or programmed to stop.

The frequency that this PPG operates in is the range 35 - 53 MHz which corresponds to the frequencies of operation for the Fermilab accelerators.

4.2 VMEbus RF Timer

In order to provide the same functionality as the VXI-UCD in a 6U VMEbus form factor, a third card was made, the VMEbus RF Timer (VRFT) [5]. The VRFT combined with a VUCD provides the same essential capabilities as the VXI-UCD.

4.3 Example usage

An excellent example of the use of the VXI-UCD is described below (excerpt from reference [6]).

The upgrade of the Fermilab Linac from 200 MeV to 400 MeV has reduced the losses in the Booster due to space charge effects, but the increased beam current causes greater coupled bunch mode instabilities. The challenges associated with designing a coupled bunch mode damper for the Booster are a large dynamic range, a

fast sweeping RF system, and a large spread in tunes through the cycle. A digital system is ideal for handling these problems; therefore, digital bunched beam dampers were designed. The damper configuration is shown in Figure 3. It consists of a common mode rejection front-end, digitizing units, fast memory, a D/A unit, and power amplifiers. All of the components, except for the power amplifiers, are VXI compatible and can be controlled with a personal computer or any other VXI control system.

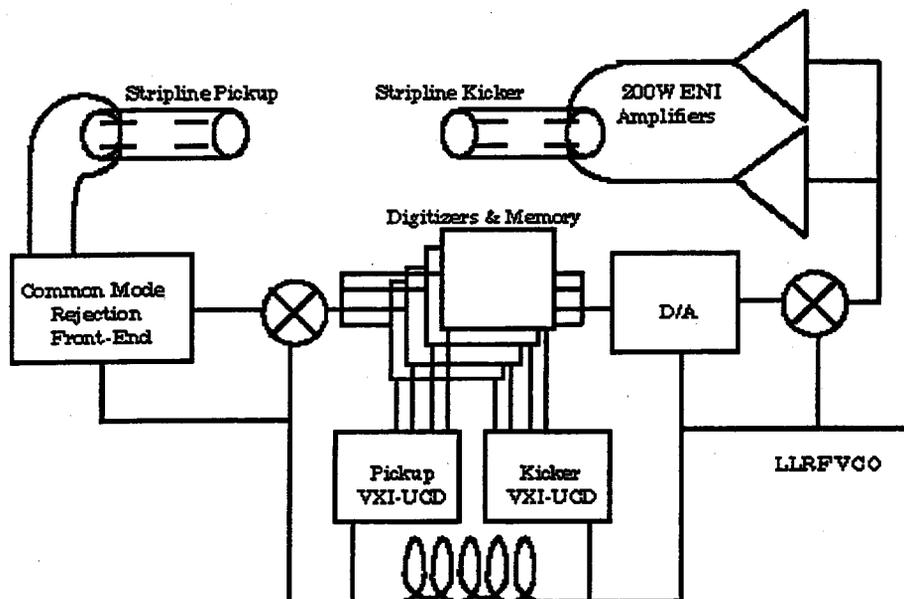


Figure 3: Block diagram of Fermilab Booster transverse digital damper system.

The RF accelerating voltage in the Booster must ramp from a frequency of 37 MHz to 53 MHz in a cycle time of 33 ms, and the non-linear frequency ramp has a peak slope of 1 GHz/s near the beginning of the cycle. The revolution period varies from 2.8 μ s to 1.59 μ s. To maintain feedback on the proper bucket, the processing system must handle 1.21 μ s of delay change quickly.

The VXI-UCD cards maintain the proper delay by remaining locked to the beam. The waveform generator memory is programmed with an interleaving pattern and clocked with an external reference which is the beam reference oscillator. This pattern triggers the digitizers to sample the beam and also triggers the digitizers to send signal to the kicker. The pattern can also be fine tuned to allow for single bucket precision in adjusting the delay.

VXI-UCD cards also act as interpreters of the laboratories global trigger system. These triggers are used to control the start and stop times of the system as well as external gates which turn off the signal during delicate times of the acceleration cycle.

5. POSSIBLE IMPROVEMENTS

Possible improvements for these modules include:

5.1 State Tracing

State tracing capabilities can be especially useful for process control and interfacing. This feature requires storing not only event and time stamp information, but state and possibly state transition information as well.

5.2 Enhanced Multi-processor Support

To improve support for multi-processor systems, an effective method for providing several message queues could be employed. This could include state tracing information as described above.

5.3 Improved PPG output phase adjustment

Improved linearity of the PPG output phase adjustment circuitry would be beneficial, as the current circuitry shows significant phase variation over the full frequency range of operation.

6. CONCLUSION

These modules have proven to be an effective tool for the Fermilab VME/VXIbus system designer. They provide a seamless interface to the accelerator timing systems and provide RF resolution synchronization capabilities as well.

7. ACKNOWLEDGMENTS

We wish to acknowledge David DuPuis who did the layout and prototype assembly of all of these modules and Jay Ticku who designed several of the timer chips used on the boards. We also wish to acknowledge all of the users, whose support and encouragement helped make these modules a reality. Special thanks to Jim Steimel for providing the module usage example (4.3) for this manuscript.

8. REFERENCES

- [1] W. Knopf, "Unibus Clock Decoder", Controls Hardware Release 30.0, 1983. Fermi National Accelerator Laboratory document.
- [2] C. McClure, Kerry Woodbury, et. al., "VUCD User's Manual", Controls Hardware Release 90.01, 1994. Fermi National Accelerator Laboratory document.
- [3] D. Beechy, "A Programmable Finite State Module for use with the Fermilab Tevatron Clock", Fermi National Accelerator Laboratory document.
- [4] K. Woodbury, Craig McClure, et. al., "VXI-UCD User's Manual", Controls Hardware Release 91.0, 1994. Fermi National Accelerator Laboratory document.
- [5] K. Woodbury, "VRFT-beta User's Manual", Controls Hardware Release 93.0, 1995. Fermi National Accelerator Laboratory document.
- [6] J. Steimel, "Fast Digital Dampers for the Fermilab Booster", Proceedings of the 1995 Particle Accelerator Conference, *publication pending*.

Control System Data Management from an Operational Point of View¹

B. Dunham, Continuous Electron Beam Accelerator Facility,
12000 Jefferson Ave., Newport News, VA 23606

Abstract

Day-to-day operation of the Continuous Electron Beam Accelerator Facility (CEBAF) requires the management of over 25,000 signals on the EPICS control system. Operators need to carry out tasks such as saving the present state of the machine, restoring old machine states and rebooting IOCs. Several tools for performing data management tasks are described in this paper. The saving and restoring of signals from a user friendly GUI using the Tcl/Tk programming language is presented, along with a method for recovering from IOC reboots, both intentional and unintentional.

Introduction

The management of the EPICS control system signals for the accelerator at CEBAF is an important issue from the standpoint of machine reliability and reproducibility. Machine uptime for the nuclear physics program is of critical importance and downtime due to poor data management techniques cannot be tolerated. Thus, one must be able to take a snapshot of the state of the machine at a particular time in such a way that the signals are easily accessible in the future for restoring or studying, with the knowledge that all signals necessary for these operations have been saved.

There are two main data management tasks of particular interest to the operations group at CEBAF: saving and restoring operational parameters, and living through IOC reboots (EPICS runs on computers known as Input-Output Controllers, or IOCs). The first section of this paper describes a save and restore tool that has been developed using the Tcl/Tk programming language[1]. It provides a simple graphical user interface to allow operators to save, restore, view and compare control system signals.

The second section of this paper describes a method for recovering from IOC reboots, both intentional and unintentional. Operationally, the main requirement when performing a reboot is to return the machine to its previous state with little or no effort. The methods described here provide a framework for meeting this goal.

I. SAVINGS AND RESTORING OPERATIONAL SIGNALS

Requirements

The save and restore program is required to manage all of the signals necessary for daily operation of the accelerator. These signals include magnet setpoints, parameters for the RF system and for diagnostic systems. The signals must be accessible by area and by type within an area, where typical areas are the injector, linacs, arcs and experimental halls, and possible types are quadrupoles, dipoles, sextupoles, correctors and RF cavities.

The program must perform several tasks. The first task is to save all of the operational signals for the entire machine (called an *ALL save*). These signals, defined by various groups (i.e. magnets, rf or diagnostics) are what is required to run the accelerator; for example, the fields in magnets and cavity gradients and phases. One should also be able to save a portion of the machine for troubleshooting or a quick backup during machine setup (called a *Partial save*).

The next task is to restore signals, either from an *ALL save*, a *Partial save*, or a special file. The user can select which data set to restore, as well as which area and device type within that data set. The special files are to be user generated files for downloading signals for particular machine settings, such as different optical lattices.

The final important task is the ability to display the saved settings and to compare them with the present settings on the machine. The compared signals should be displayed side by side, and the signals differing by more than a given tolerance should be highlighted for ease of visual comparison.

Methods

The BackUp and Restore Tool (BURT) written by N. Karonis[2] of Argonne National Laboratory is the program used for saving and restoring EPICS database signals. The present version of BURT has been tested extensively at CEBAF and has proved to be very reliable. To save signals, the basic command is

1. Supported by U.S. DOE Contract DE-AC05-84-ER40150

```

burtrb -f reqfile -o snapfile
while

```

```

  burtwb -f snapfile

```

will restore signals to the control system. The term *reqfile* refers to a file containing a list of signals that one requests to save or backup, and *snapfile* refers to the file generated by BURT that holds the saved signals and their values. Error messages are generated if a signal cannot be connected to for some reason (wrong name, IOC down, network traffic, etc.).

As BURT saves all of the signal values into one *snapfile*, it is not possible to index the values in a way so that they can be retrieved by area and device type as is required. To implement this desired feature, individual *reqfiles* were generated for each area and device type, and each file has a descriptive name. For example, the rf signals for the north linac are stored in a file *NLrf.req* and their values are stored in a file *NLrf.snap*. Presently, 68 such files are needed to cover the entire accelerator.

Whenever a save is performed, all of the *snapfiles* are put in a directory tagged by the date and time of the save. This allows a large number of files to be managed by keeping track only of the date and time of the save and the areas of the machine that were saved.

File System Organization

The file system resides on the operations cluster at CEBAF (see figure 1). The *burt* subdirectory contains the directories that hold the *reqfiles* and the *snapfiles* (from ALL saves, partial saves and special download files). The *tcl_apps/burt* subdirectory contains all of the source code as well as scripts for automatically generating all of the *reqfiles*. These scripts (for rf, bperms and magnets) are located in separate subdirectories.

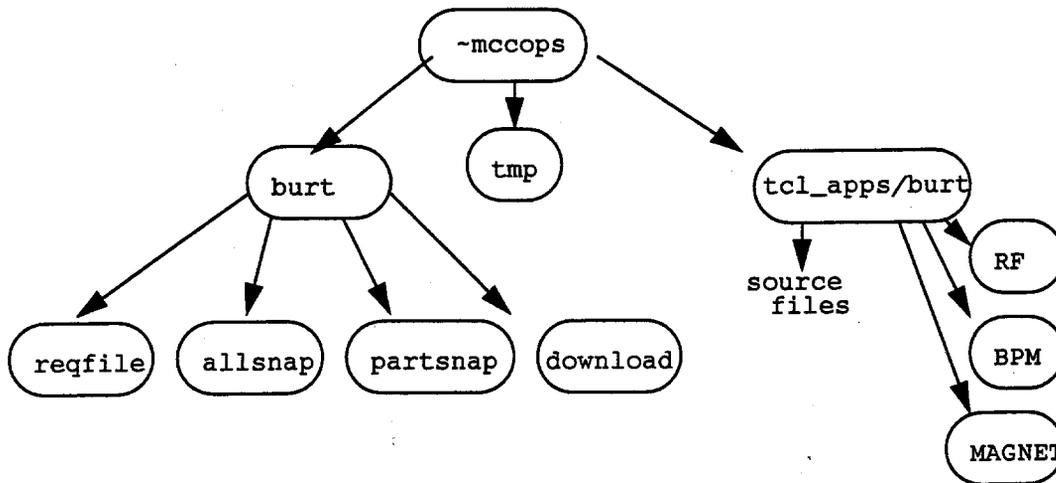


Figure 1. The file system hierarchy for saving and restoring signals

User Interface

A graphical user interface using the Tcl/Tk windowing language controls the save and restore program (see [3] for complete details). Tcl/Tk is extensively used at CEBAF as it provides a way to quickly develop complex programs in a windowing environment. The main menu allows the user to execute various save and restore commands, select the areas of the machine to be saved or restored, select options, or display a help screen. Figure 2 shows the main menu and the commands under the *Execute* menu button, each of which are described below.

The first task is to save a set of signals and it is initiated by selecting *Perform A Save of the Selected Areas*. Before saving the signals, a comment must be entered in the display that appears and an area(s) must be selected from the *Area-Selection* menu (see figure 3). The save can then begin, with the name of each *reqfile* showing at the top of the display as it is processed. Each corresponding *snapfile* is sent to a directory whose name contains the date and time that the save was started (a subdirectory of *allsnap* or *partsnap*), for example *08-20-95_12:05:27*. At the conclusion of the save, the vital information (user, date, time, comment, and selected area) is appended onto the file *master.save*, which is used as a pointer to the location of the saved signals. If there were any errors or signals not connected to during the save, they will be displayed as well as appended to a log file.

The *ALL* selection will obviously select all of the signals to be saved. This area selection is different in that the signal values will be directed to files in the *allsnap* directory (see the section *File System Organization*), whereas selecting an individual area(s) will direct the signal values to the *partsnap* directory. Since saving all of the signals can take several minutes, this feature was implemented to allow a quick save of a small portion of the machine.

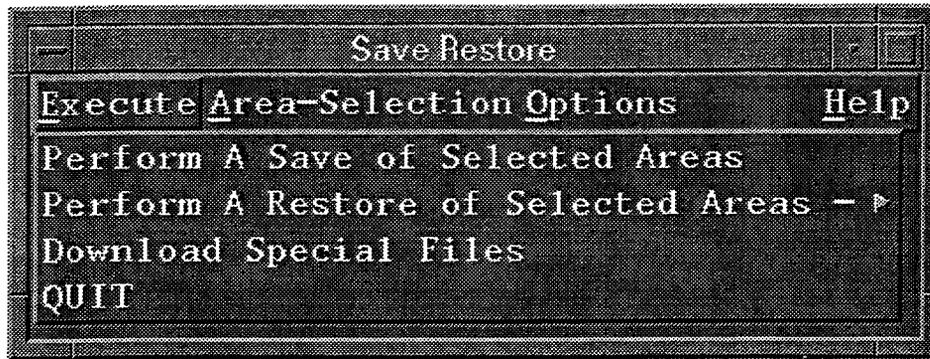


Figure 2. The *Execute* Menu Item

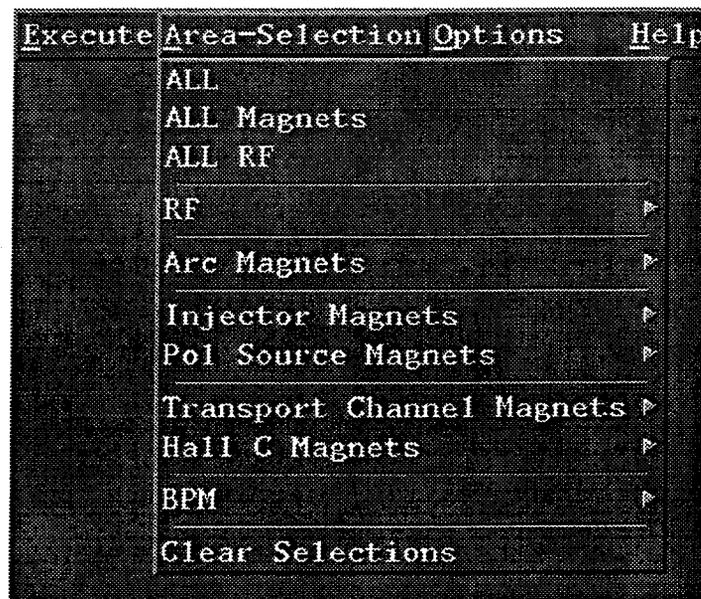


Figure 3. The Area Selection Menu Item

The next major task is restoring signals. Selecting *Perform a Restore of Selected Areas* under the *Execute* button on the main menu will generate the window shown in figure 4. Again, an area and device type must be selected before attempting to restore. The desired data set is chosen by clicking on the line showing the date and time or on the corresponding line showing the comments. Choosing *Restore* will initiate the restore, but a window will first pop up asking for verification before actually loading the files. The *snapfile* names will be displayed on the top of the window as they are being restored. When the restore has concluded, any signals that were not restored will be displayed and appended to a log file.

One of the most useful (and most used, over 100 times on some days) features is the ability to take a set of data, look at all or part of it, and compare it to what is presently on the machine. After selecting a data set, clicking on the *Filter/View/Compare* button will bring up the window shown in figure 5. All of the files corresponding to the selected areas are then concatenated together and displayed in the listbox.

A number of options are available on the *Filter/View/Compare* window. It is often desirable to be able to filter the signal list to examine only a portion of them. For example, to look at only the rf phase setpoints, typing *PSET* in the filter box and hitting either return or the *Filter* button will select only signals that have the symbols *PSET* in them. The *Print* button will dump the signals names and their values to the printer (the printer destination can be set under *Options* on the main menu bar).

Pressing the *View Current Values* button retrieves the present settings for the signals from the control system and displays them in a listbox next to the old settings. If the actual settings differ from the saved settings by more than a given tolerance, the value is highlighted. The *Print* button will dump the differences to the printer.

Execute Area-Selection Options			
Selected Areas	#	USER	DATE
NLrf	360	lehmann	08-30-95
	361	law	08-30-95
	362	lehmann	08-31-95
	363	law	08-31-95

Figure 4. The window for restoring signals

II. IOC REBOOTS

Introduction

At CEBAF, the EPICS control system presently oversees 37 IOCs which handle all of the real-time database processing for the machine. Each IOC controls specific devices at various locations around the accelerator, some handling mostly RF systems, while others handle magnets or a wide range of diagnostics. They typically run continuously, with occasional planned reboots for software updates and less occasionally, unplanned crashes.

For both planned and unplanned reboots, it is necessary to save all of the signals on the IOC as close as possible to the time before the reboot occurs, and then to restore the signal after the completion of the reboot. The signals to be saved are determined by the various software and hardware groups, and include all of the parameters needed to restore the machine to its pre-reboot state. If the reboot is planned, one can save the signals immediately beforehand, but if the IOC crashes, this is not possible. A crash can disrupt accelerator operations if any of the signals on that IOC have been changed since the last time a save was performed. To minimize such problems, a system has been established to save the signals on each IOC once every hour, thus the state of the machine no more than an hour old can be recovered. The methods for managing the data for both planned and unplanned reboots are similar and are described below.

Methods

A number of shell scripts oversee the saving and restoration of signals on the IOCs for reboots (see [4] for complete details). The script *ioc-save.csh* *<iocname>* *<type>* (the brackets indicate the arguments to the script) contains a list of the subsystems running on each IOC and it directs each of these subsystems to execute its own script which determines the appropriate *reqfiles* to save. The argument *<type>* can be either the word *normal* or *crash*. *Normal* indicates that a save for a planned reboot should be initiated and directs the *snapfiles* to a directory named *normal*. For planned reboots, the database signals are saved immediately before and restored immediately after a reboot, thus reducing the possibility that any signals will change just before the reboot. If the argument is *crash*, the *snapfiles* are sent to a directory with a two digit number (00-23) indicating the hour during which the save is performed. In both cases, the set of signals to save are identical, only the destination for their values is different. Any errors are re-directed to a log file for later inspection.

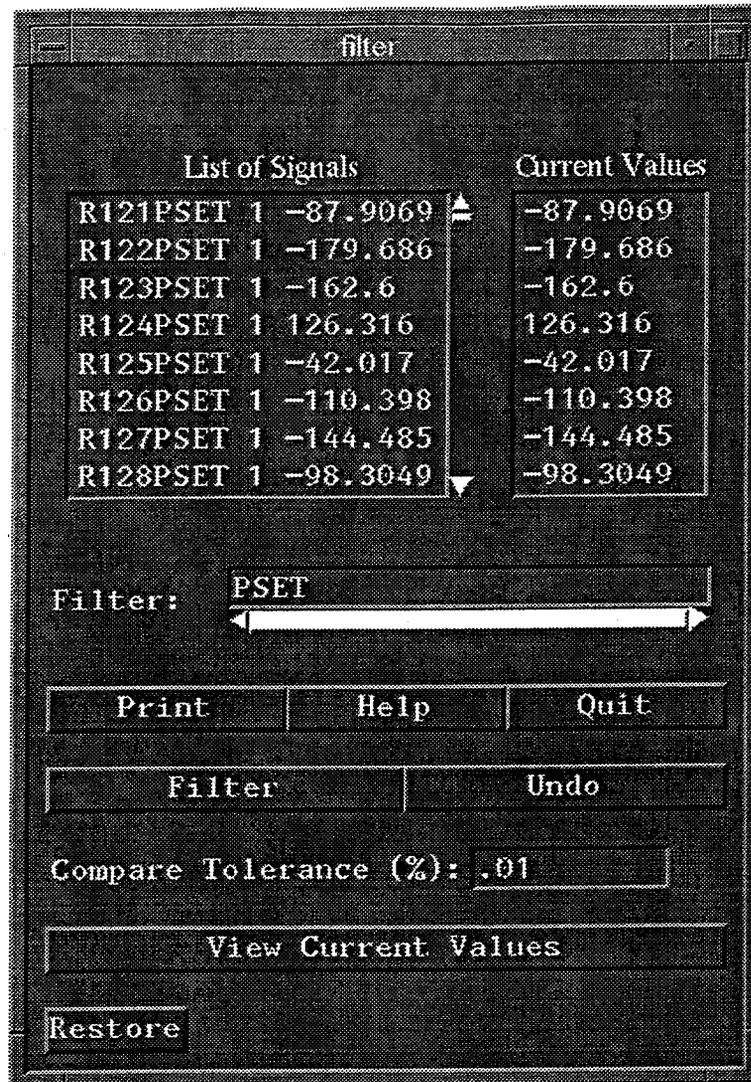


Figure 5. Comparing saved signals to the present machine state

The script *ioc-restore.csh* *<iocname>* *<location>* manages the restoration of signals. The *location* can either be *normal*, again referring to a planned reboot, or a number indicating the 'hour' from which to restore, *00, 01, . . . , 23* for crash recovery. This script also directs each subsystem to execute its own script to determine the *snapfiles* to restore for a given IOC. Error messages are echoed to the screen for the operators to see.

The Unix clock daemon *cron* controls the periodic execution of scripts on each IOC. At present, 37 IOCs are used at CEBAF, thus to backup each IOC once an hour requires a save approximately every 1.5 minutes. This distribution, rather than saving all at once, keeps the additional network load down so that it is not noticeable to the operators.

The entire system is designed so that adding additional subsystems and signals is simple. First, a subdirectory for the new subsystem must be added, along with the necessary signals to save and scripts for saving and restoring those signals in the appropriate order. Then the subsystem name must be added to the list for each IOC that it runs on in the scripts *ioc-save.csh* and *ioc-restore.csh*. More frequent backups may also be implemented by changing the crontab file and adding additional subdirectories for each new time.

The procedure for a reboot differs slightly for planned and unplanned reboots. For planned reboots, one must hit a button to start the save, perform the reboot and then hit another button to restore the signals. For an unplanned reboot, the process is more complicated. On a particular IOC, the restore scripts will show the the time, the total file size, the expected total file size and the log file size for each of the hourly saves. A zero log file size indicates that the save was completed successfully. The script instructs the operator to restore the signals from the last time which had a zero log file size before the crash using *ioc-restore.csh <iocname> crash <00-23>*.

File System Organization

The file system resides on the operations cluster under the *REBOOT* directory (see figure 6). There is a subdirectory for each accelerator system: beam position monitors, rf, magnets, fast shutdown system (fsd), beam loss monitors, viewers, vacuum and various high-level applications; a subdirectory for shell scripts to control execution; and a subdirectory *snapfiles* that contains the saved signals for each IOC.

For each accelerator system, there is a list of signals that need to be saved in a subdirectory *req*, and a shell script for saving and restoring those signals. The *snapfiles* directory contains all of the saved signals for the IOCs. There is a subdirectory for each IOC and under each IOC there are 25 subdirectories: one for each hour (00 through 23) and one for normal reboots. The *scripts* directory contains a number of shell scripts for controlling the saving and restoring of signals and the *cron* directory contains the file used for starting up the Unix cron clock daemon.

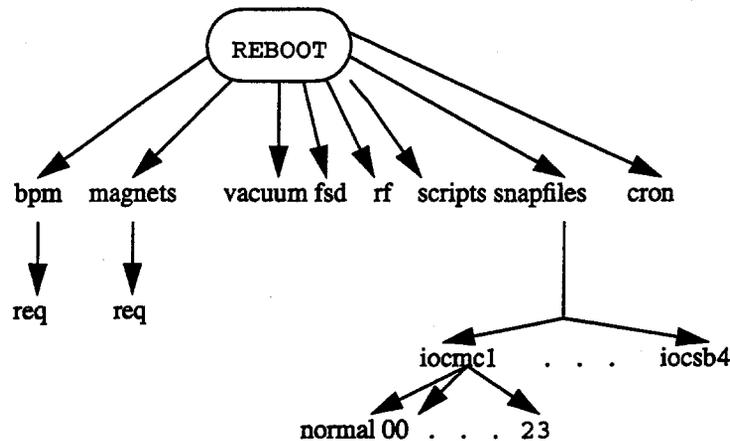


Figure 6. The file system hierarchy for IOC reboots

III. SUMMARY

Managing all of the signals necessary for accelerator operations at CEBAF is an important job in terms of machine reliability and restorability. The save and restore tool has provided a way to save, restore, view and compare the control system signals necessary for operations, and the reboot methods have provided a framework for a smooth transition through an IOC reboot. The tools described here have performed well in over a year of intensive use and have significantly improved the operation of the CEBAF accelerator.

IV. REFERENCES

- [1] John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1992.
- [2] N. Karonis, *The Backup and Restore Tool: Theory and Use*, EPICS User Manual, Ch. 17, 1992, Los Alamos National Laboratory
- [3] B. Dunham, *Recovery from IOC Crashes on the EPICS Control System at CEBAF*, CEBAF technote CEBAF-TN-95-048
- [4] B. Dunham, *A Tool for Saving and Restoring Control System Signals*, CEBAF technote CEBAF-TN-95-057

Improvements to Realize a Higher Reliability of the KEK Linac Control System

Norihiko Kamikubota, Kazuro Furukawa, Kazuo Nakahara, Isamu Abe and Akihiro Shirakawa
National Laboratory for High Energy Physics (KEK)
1-1 Oho, Tsukuba, Ibaraki 305, Japan

Abstract

A new control system for the KEK 2.5-GeV electron/positron linac, which comprises VME stations with the OS-9 operating system and UNIX-based workstations, started operation in October 1993. During the 10000 hours of operation in the first two years, the failure rate of the control system was less than 1%. However, two-year operation has provided several suggestions as to improving the reliability of the control system. On the basis of our current experience several kinds of improvements have been carried out or are in progress.

I. Introduction

The new control system for the KEK 2.5-GeV electron/positron linac started operating in October 1993 [1], [2]. It comprises three components:

- (a) UNIX-based workstations as resource servers,
- (b) the TCP/IP network as a main communication path, and
- (c) VME-bus computers with the OS-9 operating system as front-end interfaces.

During the 10000 hours of operation in the first two years, the failure rate of the control system was less than 1%¹.

However, we experienced a few problems which forced us to stop the control system for more than one hour. The most serious one was due to a disk crash. At the first occurrence it forced us to stop the control system for ten hours. In order to realize a higher reliability of our control system several kinds of improvements have been carried out or are in progress.

In this report, the improvements concerning hardware redundancy are described in Section II. The developments in the control software are discussed in Section III.

II. Hardware redundancy

A. Workstations to distribute functions

When the new system started in October 1993 we had only two UNIX workstations as resource servers.² One was used for the operation (hostname *peach*), and the other was for software development (*maple*). At that time all of the functions needed for linac operation were concentrated on *peach*.

We experienced two problems: One was that the resources of *peach*, such as the CPU power and the memory, had become inadequate for the increasing demand. The other was that a problem concerning *peach* inevitably resulted in a stoppage of the linac operation. It was apparent that a backup scheme to cover the fault of *peach* was desirable. We thus decided to introduce a new workstation with higher performance as a secondary machine for linac operation.

Since our inter-process communication protocol has availability among workstations of different operating systems [3], we were able to choose a workstation from among a wide variety of candidates having various operating systems. We selected the DEC3000AXP, since it seemed to have the best cost-performance at the time. The new workstation, called *grape*, is roughly 3–5 times as powerful as the previous one, *peach*. A list of the workstations in our control system is given in Table I.

¹The total operation hours in fiscal year 1994 was 5040 hours, while the failure time due to the problems concerning the control system was 13.4 hours.

²We also had a disk-less UNIX machine, *lime*, as well as a few old-fashioned UNIX-based machines. However, they were not capable of acting as resource servers.

In order to proceed with a further distribution of the functions, we will introduce two additional workstations by the end of fiscal year 1995. One will be used as a secondary machine for software development with the same operating system as *grape*; the other will be used as the third operation server.³ Both are also given in Table I.

Table I
List of the UNIX workstations in the KEK linac control system.

hostname	machine(operating system)	introduced in	main function
<i>peach</i>	DECstation5000(Ultrix v4.4)	Nov.1990	operation 1
<i>lime</i>	DECstation5000(Ultrix v4.4)	Oct.1991	status display
<i>maple</i>	DECstation5000(Ultrix v4.4)	Oct.1992	development 1
<i>grape</i>	DEC3000AXP(Digital UNIX v3.2)	Mar.1994	operation 2
-	DEC3000AXP(Digital UNIX)	Feb.1996 (plan)	development 2
-	DEC3000AXP(Digital UNIX)	Feb.1996 (plan)	operation 3

B. Disk-array unit

During the years before 1994 we experienced a few problems due to disk-crashes. Fortunately these problems were not serious, since the disks did not contain any important files necessary for linac operation.

In order not to suffer from such problems we introduced a disk-array unit in July, 1994. The unit contains seven disks of 2 GB and works as a single disk with a size of 12 GB. Any fault of one disk does not affect to the disk-array operation at all.⁴ We now keep any important files in the disk-array unit. It is worth noting that the files for the VME stations are also kept there.

C. Network improvements

We started a new control system with a few network segments. In October 1993 we used the popular Ethernet (coaxial cables called 10Base5), linked together with a network repeater. Each VME station in the five sub-control rooms is located at 80-meter intervals, and is connected by a long coaxial cable. Since there were several high-power klystrons along the cable, we were afraid of some influence of pulsed noise from them.

The first improvement was carried out in July 1994. We introduced fiber-optic cables between the center room and the sub-control rooms, which are used instead of the coaxial cable. In addition a star-topology is preferable, since any problems at a sub-control room do not affect the other rooms. Each of the coaxial cables and the fiber-optic network has a capability of 10 Mbps throughput.

Since a natural increase of the network traffic was expected, a network hub, having a backbone capable of 100 Mbps communication throughput, was introduced in March 1995. All of the network segments were connected with the hub, and each segment was separated by a network bridge. In addition, in order to separate the network traffic concerning only the workstations in the center room, a FDDI link between the major workstations was introduced in June 1995.

D. Other notes

Since the VME station called *kannaduki* was in charge of both the injector and the center parts, a new VME station, *hatsuhi*, was introduced in September 1994 to accept the functions for the center part.

We sometimes experience a momentary power outage caused by lightning during the summer. In order to avoid such problems a couple of UPS systems (Uninterruptible Power Source) were introduced in January 1995. They cover the power supplies at two workstations for linac operation (*peach* and *grape*).

³See also the descriptions in III-A.

⁴During the recent 1 year, the disk-array unit unfortunately stopped twice due to unknown errors. We will replace it with a new one to be introduced with the additional workstations in February 1996 (see Table I).

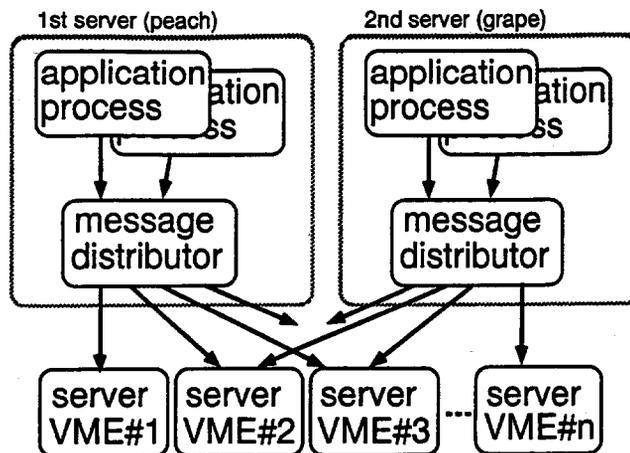


Figure. 1. Relation between the two servers and the VME stations.

III. Software techniques

A. Two servers for the linac operation

As described in II-A, a new workstation, *grape*, was introduced as a secondary server for the linac operation in March 1994. The relation between the two servers and the VME stations is shown in Figure 1. The great advantage of this scheme is that when one of the two servers stops due either to a problem or to a maintenance shutdown, operators are able to continue linac operation with the remainder server.

Since the operating system of the secondary server (Digital UNIX) is different from that of the first server (Ultrix), we have prepared the server processes and the application processes for the secondary server by re-compiling the sources written in C language. Each of the control services and associated processes, such as the digital-output and field networks (LOOP-II or LOOP-III), has been prepared one by one, as given in Table II.

Table II
Schedule of the control services to arrange for the secondary server, *grape*.

service name	description	completed in
out16	16bit digital output	Dec.1994
in16	16bit digital input	Feb.1995
dac12	12bit analog output	Feb.1995
adc12	12bit analog input	Feb.1995
loop3	48kbps field network	Jun.1995
loop2	500kbps field network	Dec.1995(plan)
gpib	for GPIB instruments	Feb.1996(plan)
rs232c	serial line	Mar.1996(plan)

As shown in Table I, we will introduce a third server with the operating system Digital UNIX. In order to decrease maintenance work involving the two operating systems, the first server, *peach*, will be replaced by it.

B. Use of the UDP protocol instead of the TCP protocol

The inter-process communication for our control system has relied on a stream socket based on the TCP protocol [3]. Since the TCP protocol contains an automatic error-retry scheme, we always wait for roughly one minute to receive a time-out signal. This is too long of a duration for typical linac operators; however the TCP protocol does not provide a method to change the time-out period. We have thus decided to use the UDP protocol instead of the TCP protocol.⁵

⁵We use the UDP protocol for the basic services given in Table II. For other services, we continue to use the TCP protocol, since most of the communications are made between two processes in one workstation; that is, no time-out errors are expected.

We also developed a basic inter-process communication procedure with the UDP protocol in August 1994. The procedure includes three retries with a 2-second time-out interval (8-second time-out in total). The modifications to switch over the control services to use the UDP protocol have been carried out one by one, together with the work involving the secondary operation server. The present modification is also effective for reducing the overall round-trip time of a control message. The round-trip time between processes at *peach* and at a VME station, including overhead with the control system, is about 40 ms with the TCP protocol, while it is 14 ms with the UDP protocol. The round-trip between *grape* and a VME station is 8 ms with the UDP protocol.

C. Disk-less VME stations

Each VME station has a local disk which is used for bootstrap loading. Since such a local disk has a greater possibility to cause problems than the other components in a VME-bus system, studies to realize a disk-less environment were made during the summer shutdown of 1995. We changed each VME station so as to use a network bootstrap protocol (BOOT-P), which enables us to boot from remote files. Since August 1995, the files needed for the bootstrap have been kept on the disks of both the two operation servers (*peach* and *grape*). Since they are duplicated, a bootstrap of any of the VME stations is possible even when one of two servers stops.

D. Other notes

Since there are several workstations, VME stations, and PC's, it is important to synchronize the internal clocks. The NTP (Network Time Protocol) is utilized for this purpose. It seems to keep the internal clocks of the UNIX workstations within one second (or better) of accuracy.⁶ The present control system produces several kinds of operation logs and history files for various devices. The total amount of such logs and history files is 5–10 MB per one day. In the case of a problem, they are very useful, since we can extract any device history of interest. The details will be described elsewhere in the future.

IV. Acknowledgement

We wish to thank Professor I. Sato for his kind encouragement during the present work. We also express our best thanks to the Linac operators for discussions and various kinds of help.

References

- [1] N. Kamikubota, K. Furukawa, K. Nakahara and I. Abe, Nucl. Instr. Meth. A352(1994)131
- [2] N. Kamikubota, K. Furukawa, K. Nakahara and I. Abe, Proc. 1994 International Linac Conference (LINAC94), Tsukuba Japan, August 1994, p.822; KEK-Preprint 94-73
- [3] N. Kamikubota, K. Furukawa, K. Nakahara and I. Abe, Proc. Int'l Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'91), Tsukuba Japan, November 1991, p.318

⁶We currently use the OS-9 of version 2.4 (ISP v1.4), with which the NTP is not available. The next version is already available, and the NTP might be available with it.

EXPERIENCE IN THE EXPLOITATION OF A LARGE CONTROL SYSTEM

G.Daems, F.Perriollat, Ch.Serre
PS Division, CERN, CH-1211 Geneva 23, Switzerland

Abstract

Experience of a four-year exploitation of the large control system of the CERN PS accelerator complex is presented with special emphasis on the parameters which are very sensitive to the exploitation team productivity. The software tools suite used in this daily maintenance is described and a particular analysis of the power and benefits of the advanced software technology used for the architecture of this suite is explained. The integration of this suite into the control system is presented, as well as its use in the control system development phase. Some considerations of the potential benefit of an Object Oriented equipment access are outlined.

1. INTRODUCTION

The latest version of the CPS (CERN Proton Synchrotron) control system has now been in use for four years (1992- 5) on six of the nine accelerators forming the CPS complex [Fig.1]. The system [Fig.2 & 3], [1] is based on the "Standard" model with 3 levels: interaction, front-end computing and equipment control. An Ethernet network running TCP/IP links the user interface, the central services and the real-time equipment processing, while field-buses (CAMAC and 1553) interlink the equipment to the real-time equipment processing. The total number of items controlled is around 10000, not including the instrumentation.

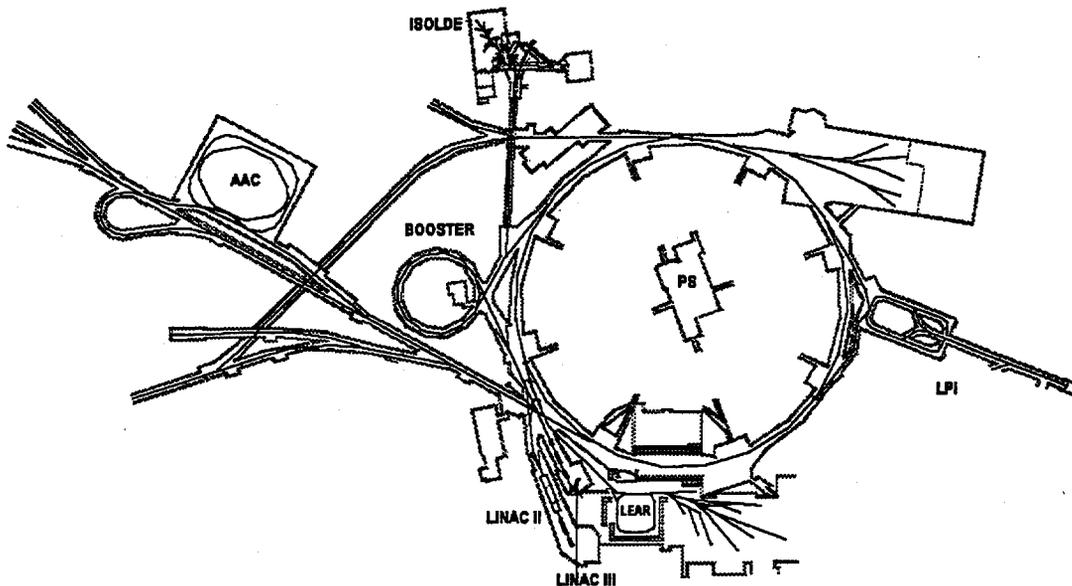


Fig. 1: Layout of the PS Complex

During these four years, the exploitation of the control system has been carried out in parallel with the old one to keep the complex in running order. By exploitation we mean the maintenance, problem solving and improvements to the control system. The people charged with this exploitation have had to adapt their knowledge and tools to the new system in which they also played an active part during its development.

The experience gained in the exploitation of such a large control system is presented, with a special emphasis on the integration and the expected benefits of modern software technologies such as the Object Oriented Programming and Development and the Knowledge Based Techniques.

2. WHAT THE CPS CONTROL SYSTEM EXPLOITATION INVOLVES

2.1 The environment

The CPS Accelerator Complex operation runs round the clock, 24 hours a day, ten months a year, with two short intermediate stops of 2 to 3 days. This means that new installations, modifications and maintenance are not easy. Nevertheless, because the CPS complex is the source of all the particles beams at CERN, its operation evolves continuously. There are several machine development periods interlaced with the normal running of the machines which result in constant modifications and updates to the control system and may require software changes in the operational environment. In addition two points should be noted; the software developers often are only transitory people such as fellows or students, with all the follow-up problems and the exploitation team have to deal with a historical inheritance, translated into a great diversity of interfaces and equipments, some of them only partially renovated due to budget constraints.

The CPS operation requires frequent daily operational changes, for example for antiproton transfers. The present staff policy results in a continuous rotation of operators with short-term contracts which leads to less experienced staff. As the operation of the machines is seen by the operators through a thick control layer, it is necessary for controls people to have some knowledge of how to operate in order to be able to help the operation crew in its daily job. This sometimes allows the exploitation people to identify as operation faults, things that were initially reported as control faults.

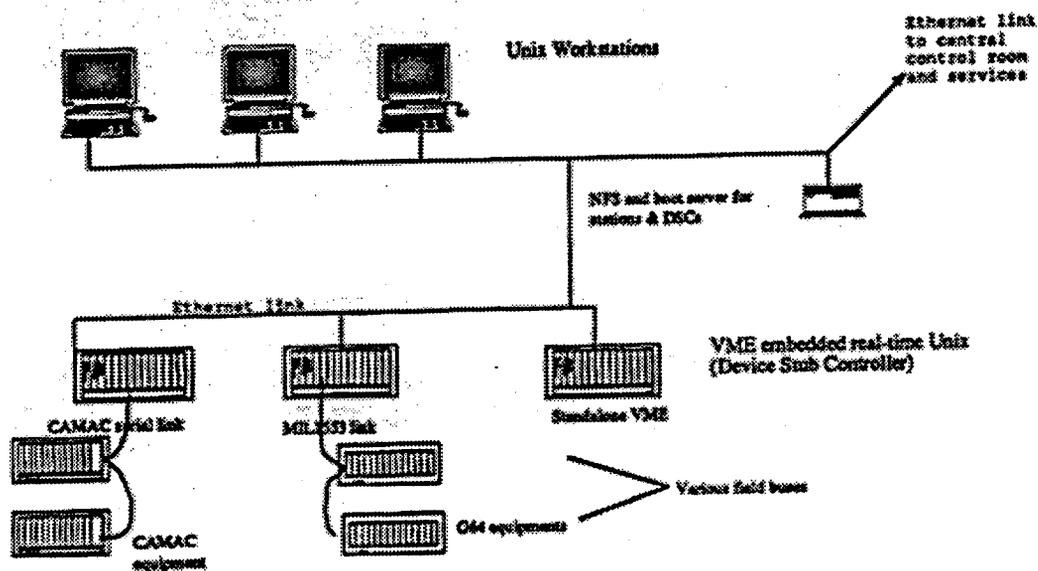


Fig. 2: The standard architecture applied to the CPS complex

2.2 On-line and Off-Line exploitation

The control system exploitation is divided in two types of work, the on-line exploitation, which is the immediate answer to faults or wrong conditions in the control system elements (hardware or software) and the off-line exploitation, which can also be called the follow-up and the solving of problems encountered during the operation of the machines.

The on-line exploitation is the field of a small and well experienced team which take charge of all problems and bad behaviour labelled by the operators as "control faults". People of this team trace the faults, analyze and

correct the faulty control elements (hardware side), check and restart the faulty software components and finally support the operation team to recover and return to the required operational state.

The off-line exploitation involves practically all the people of the Controls Group in order to improve the reliability and the robustness of the control system (hardware as well as software). A regular weekly exploitation meeting enforces a strict follow-up of the different problems and changes. Agreed upgrades, extensions or enhancements are mainly coordinated by the exploitation team. It also proposes and develops the necessary diagnosis and exploitation tools.

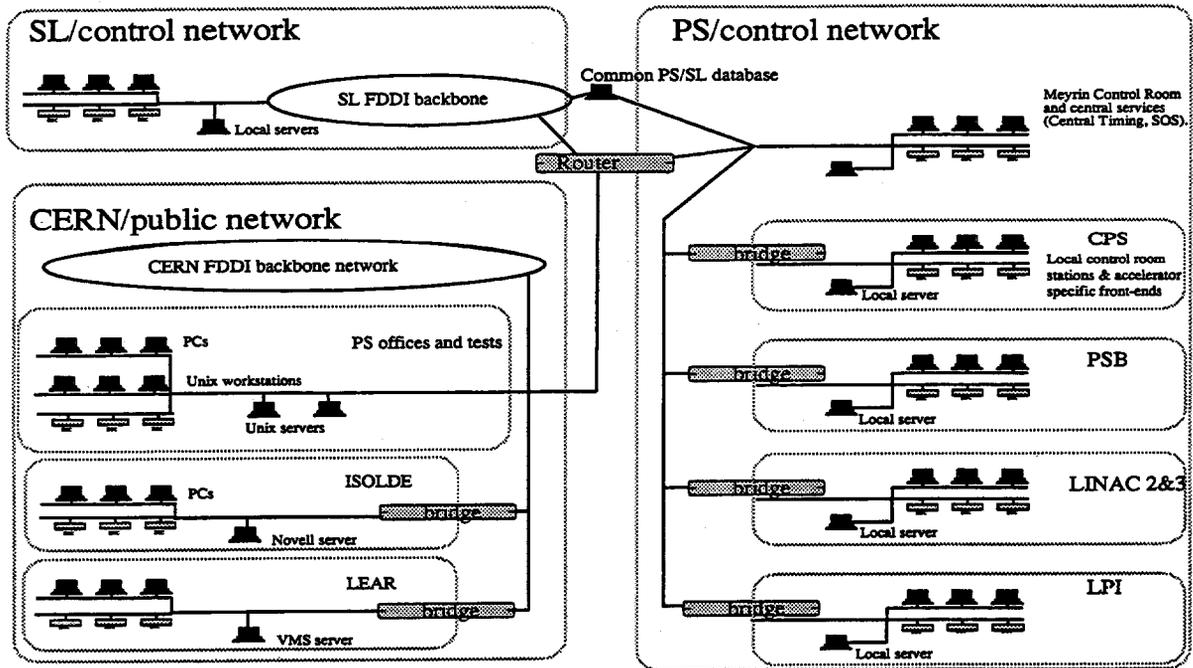


Fig. 3: The CPS Network

2.3 The Exploitation Team ?

The exploitation team is a small team with good experience, an overview of the system and general purpose knowledge and its first priority is to keep the control system running correctly, whatever the time and the work required. A "piquet" service (5 persons) is in charge of the on-line services; one person is assigned for one week (24 hours a day) to answer as fast as possible the calls from the operation crew if there is any breakdown of the control system components.

This exploitation team provides a single entry point (on-line and off-line) for all the requests concerning the problems, improvements and modifications to the running control system, which enforces directly the homogeneity of the hardware/software solutions. The persons of this team are attached to the other sections of the control group to develop diagnosis and exploitation tools. This is mainly software for interface and equipment access and for the overall survey system.

3. EXPLOITATION TOOLS

The performance of this team depends on the continuous education of the persons, the documentation of the hardware and software components of the control system, the communication between the developers and the members of this team, but also on the tools provided to this team to be able to diagnose, recover, restart and/or set up the components of the control system. Data logging of the element values, easy-to-use links with the data base (ORACLE), automatic configuration of front-end processor devices from the data base, tracing and log of faults with post mortem analysis capabilities [2], testing and setup facilities are the main tools needed by the members of the exploitation team to fulfil correctly their first priority job.

The status of the different controlled elements of the CPS complex are permanently monitored on the screens of the Control Room workstations, in a window generated by the Alarm process [3]. This program, used by both the operation and exploitation crews, scans the status of the controlled equipments every 30 seconds, via the uniform equipment access level (Equipment Modules [4]). In addition, it surveys the VME and CAMAC crates, both for overall hardware and software status.

This Alarm process is the main entry point for diagnosis tools and information. It allows the presentation of :

- the details of faulty equipments
- the layout of input/output crates (VME and CAMAC), the modules in the crates and the equipments controlled
- the front-end processor status : accessibility, interrupts and errors
- the fault log, the pulsed status and the interrupt layout of the timing system elements.

From this information one can reset a faulty equipment (such as a power supply, a vacuum pump, or a RF cavity) or detect a fault at the element level. A front-end processor not accessible through the control system can be remotely rebooted after verification of its status. The history of the reboots and the errors are logged and can be displayed on request.

Most of the exploitation tools can be called through this Alarm process, either directly or through two important general programs :

The first program, **Equipment Info**, gives the possibility of selecting dedicated programs to diagnose, test and initialize an equipment or a control interface; this is especially the case for the different instrumentation and field-buses (CAMAC and 1553). One can also test a particular equipment access and initialize it if necessary. Special debug or repair commands, documentation concerning the Equipment Module and the layout of subsystems (timing systems for example) are also available from this general panel of services. From an other special panel, an equipment can be accessed through the Equipment Module either by its name or its number or its address. The selected equipment can be tested or controlled up to the last bit.

The second program, **Setup**, is created to reinitialize automatically an equipment or a whole system (a CAMAC crate for example). It is a very important and general purpose tool which is described below.

A few other important tools exist, which can be called for a specific purpose. These are:

- for the verification of the description and synchronization of the different operations of the CPS accelerator complex, where a rule-based consultant can be called to verify if the required schedule is acceptable. This is the Beam Card Desk Checker [5]
- for the automatic configuration of the software of the front-end processors directly from the data kept in the ORACLE data base [Ref.6]
- for the Data Browser/Editor which gives the possibility to read, compare and modify the operational values stored in the data table of the equipment access modules (Equipment Module); with this tool, the exploitation team can work on the actual operational data, the regularly-saved values and the values called references.

4. SETUP : KNOWLEDGE-BASED TECHNOLOGY

The development of the Setup program is based on an Object Oriented programming, together with a procedural knowledge representation [7]. Setup provides the means of initialization and of non-destructive testing for the accelerator equipment based on CAMAC and VME control interfaces. It is a rule-based process which can cope with any unknown initial state of the equipment.

4.1 Utilization

This Setup program is used both by the exploitation and the operation Teams. The primary uses are after a power failure, a shutdown or the replacement of a faulty hardware module or crate; Setup allows one to initialize

the hardware with the correct procedures and the correct operational values (last saved values). Setup is also used to reset a faulty equipment (power supply for example) to come back to the normal state. When an instrumentation gives wrong information or does not run correctly, the best solution is to call Setup. Finally Setup can also test an equipment without disturbing or with the minimum disturbance of its normal functioning.

4.2 Main points and Realization [7,8]

The Setup system can easily be adapted to the control system environment. It is flexible and evolutive and adapted to a huge variety of hardware/software modules, including the treatment of hardware-coupled equipments. The description authorizes different sets of objects, so that the Setup actions can be process- or hardware-oriented. The access to the control system elements is made using the uniform equipment access method (Equipment Modules).

During the running of the program, the user is informed about the current state of the equipment and about the diagnosis of the procedure in real-time; if an action is not successful, the diagnosis of the faults is given. Post mortem analysis exists for the Setup protocols that have taken place during the test or the reset of an equipment or a process.

The realization of the program is based on the development of an object model which reflects the structure of the Control system equipment access: front-end processor, CAMAC loop, CAMAC crate, CAMAC module and Equipment. The knowledge-based representation gives the description of object classes, control rules and operation algorithms. The concrete object lists are automatically "*instanciated*" from the Real-Time Data Base. The realization of the Setup program is based on an expert system shell (PROSC: Procedural Reasoning Object System for Control) which allows object oriented knowledge description, procedural reasoning techniques, real-time features and direct queries to the ORACLE data base.

4.3 Benefits for the exploitation of the control system

The user interface of Setup is done through X-windows/Motif panels which are called by the Alarms process, the exploitation programs or the hardware surveillance at the level of the faulty object. The Setup facility is also accessible from terminals using a certain number of commands (for tele-diagnosis capability).

As stated in the previous paragraph, the Setup rules and algorithms are easy to describe and to change for the different classes of the control system. This results in a well-adapted, up-to-date and reliable tool that executes the necessary actions demanded by the operator, taking into account real-time events and hardware-coupled equipment, which are otherwise invisible on the interaction level.

The automatic generation of the concrete object lists related to a "*super*" object to initialize is one of the essential benefits of this setup. The Setup software executes automatically all necessary actions, either when one has to replace one hardware module or to initialize one equipment or a whole CAMAC loop. The setup can be called either from a hardware description layout or from a software equipment level. The Setup operates today on:

- 100 Front end processor VME crates (DSC)
- 130 CAMAC crates
- 3000 equipments
- with more than 100 object classes and 200 control procedures.

It is a necessary and powerful tool for the daily operation of the six machines of the CPS complex controlled by an example of the standard model of control system for accelerators. The technology in use and the integration in the control system provide the capability to evolve with the system without difficulty.

5. CONSIDERATIONS ON OBJECT ORIENTED TECHNOLOGY WITH BENEFITS FOR THE EXPLOITATION

The equipment access software structure of the CPS control system has been Object Oriented since 1988 [4]. The previously realized Equipment Module model has proved to be very useful for the exploitation and developers because it is a standard generic facility which gives access to all the process equipment on a standard way and allows the use of powerful and generic tools.

We already found that Object Oriented Techniques facilitate the tasks of the exploitation Team by providing:

- easier maintenance of software programs (produced by temporary or external people)
- faster and safer modifications of the software products

- easy integration of jobs developed outside the site (to benefit from external expertise)
- good integration with an ORACLE Data Base description
- independence of the control system architecture, to be able to follow the technological evolution of the different control system components.

The fields in which we are less confident, due to our short experience, are on the benefits to be gained from object languages. Especially, we have some doubts on:

- the amount of time necessary for a large community of software developers to be fluent in C++ and in the object oriented technology environment
- the difficulty to define and setup the object libraries for the different usage
- the complications in building a coherent set of abstract object models for the various control process provided
- the facility for the casual software developers to use efficiently the object oriented development tools to increase substantially their productivity
- the impact of Object Oriented Databases and their level of standardization
- and, last but not least, the stability of the emerging standardization, especially on standard C++ and the basic object library (both on real-time applications and on object broker technology in a distributed environment)

All these points can have a direct impact both on the long term exploitation (and re-use) of the huge amount of control software, and on the easy integration of third party components.

6. CONCLUSION

One of the main tools used by the exploitation team on the large CPS control system is the Alarms process, which acts both as a surveillance program running continuously on the operational work stations and as a switchboard giving the possibility of calling the other exploitation tools described in this paper.

The other important tool is Setup, which uses the object oriented and the knowledge based technologies to offer a basis for the test, setting up and initialization of the different control system elements, both individually at the equipment level and at the subprocess level. These technologies could be used for other applications such as the timing survey, synchronization generation and the interpretation of complex results. These technologies have shown their capability in solving specific problems and they are able to be integrated in a classical system in a very cost effective way.

Our long experience in use of the Object Oriented Technology gives us full confidence in the future benefits of large scale implementation. However, to be in a position to carry out efficiently the large scale integration of this type of software, we believe that more time is needed and that wider open collaboration and exchange of experience is essential. These points should receive the full support of the management of the major projects in our field of activity.

7. ACKNOWLEDGMENTS

We give special thanks to the members of the exploitation team whose efforts kept the control system running with a high level of availability (< 1% of machine down-time had been due to control problems). We are also grateful to the Control Group members who continuously try to provide excellent exploitation tools using the most suitable and efficient software technologies.

REFERENCES:

- [1] F.Perriollat, Ch.Serre, "The new CERN PS control system, Overview and status", ICALEPCS, Berlin, Germany, October 18-23, 1993, Nucl. Instr. And Meth. A352 (1994) 86.
- [2] M.Beharrell, G.Benincasa, J-M.Bouche, J.Cuperus, M.Lelaizant, L.Merard, "Model based, detailed fault analysis in the CERN PS Complex equipment", ICALEPCS 95, in these Proceedings.
- [3] J-M.Bouche, J.Cuperus, M.Lelaizant, "The data driven alarm system for the CERN PS accelerator complex", ICALEPCS, Berlin, Germany, October 18-23, 1993, Nucl. Instr. And Meth. A352 (1994) 196
J-M.Bouche, "Utilisation du programme Alarm", 6.03.95, PS/CO/Internal Note 95-17.
- [4] L.Casalegno, J.Cuperus, Cl-H.Sicard, "Process equipment data organization in CERN PS controls", ICALEPCS, Vancouver, Canada, October 30-November 3, 1989, Nucl. Instr. And Meth. A293 (1990) 412.
- [5] J.Lewis, P.Skarek, L.Varga, "A rule-based consultant for accelerator beam scheduling used in the CERN PS complex", ICALEPCS' 95, in these Proceedings.
- [6] J.Cuperus, A.Gagnaire, "Automatic Generation of Configuration files for a distributed Control system", ICALEPCS 95, in these Proceedings.
- [7] J-M.Bouche, G.Daems, V.Filimonov, V.Khomoutnikov, F.Perriollat, Yu.Ryabov, "Representation and Usage of knowledge for Initialization of accelerator control equipment", presented to RT-95, East Lansing, MI USA, May 23-26, 1995.
J-M.Bouche, G.Daems, V.Filimonov, V.Khomoutnikov, Yu.Ryabov, "Knowledge based Setup facility", 22.11.94, PS/CO/Internal Note 94-81.
- [8] G.Daems, V.Filimonov, V.Khomoutnikov, F.Perriollat, Yu.Ryabov, P.Skarek "A knowledge based control method : application to accelerator equipment Setup", ICALEPCS, Berlin, Germany, October 18-23, 1993, Nucl. Instr. And Meth. A352 (1994) 325.

OPERATIONAL EXPERIENCE WITH THE CEBAF CONTROL SYSTEM*

K. White, D. Kehne, Continuous Electron Beam Accelerator Facility, Newport News, VA 23606 USA

Abstract

This paper describes our operational experience with the CEBAF control system. CEBAF is operated using a variety of software packages and custom applications. In the spirit of collaboration and software sharing, the current system utilizes code developed at four major laboratories as well as commercial packages and custom applications. The basis of the CEBAF control system is EPICS (Experimental Physics and Industrial Control System), originally developed at LANL and ANL. The CEBAF accelerator is operated primarily using the common EPICS tools such as the archiver, alarm handler and display manager. In order to supplement the features of EPICS, custom applications have been developed and incorporated into the control system using C, C++ and TCL. Additionally, the CEBAF helium plant is controlled using the CEBAF TACL system which requires communications between the two control systems. CATER, a problem tracking program developed at SLAC, has also been integrated into the operational software. Current plans call for the integration of an on-line logbook developed at APS and the internal development of a down-time logger program. We discuss how these various tools and applications are used operationally, the advantages and disadvantages of the systems and challenges related to integrating this diverse array of software.

Introduction

CEBAF is a 4 GeV CW, high luminosity electron accelerator in Newport News, Virginia. The accelerator consists of two 0.4 GeV superconducting RF linacs connected by two 180° arcs. Each linac consists of 20 liquid helium vessels (cryomodules), each containing 8 cavities. The 45 MeV injector contains an additional 18 superconducting cavities. The beam is recirculated through the linacs up to 5 times yielding an energy of 4 GeV. After the 5th pass, the beam will be split and delivered to 3 halls simultaneously. The 3 experimental halls house a variety of complex detectors. The control systems for both the accelerator and the experimental facilities have recently been migrated from in-house developed packages to systems based upon EPICS (Experimental Physics and Industrial Control System). This paper describes the operational experience with the new control system and the additional tools which have been added to supplement the basic EPICS package.

Background

In the summer of 1993, in the midst of accelerator commissioning, it was decided to replace the CEBAF in-house control system TACL (Thaumaturgic Automated Control Logic) with EPICS, the product of a multi-lab collaboration. The integration of EPICS was accomplished using a phased approach which allowed TACL and EPICS to coexist in the same operational environment.[1] This allowed commissioning to continue with a minimum of interruption from control system changes. Due to the untimely nature of this change, many of the operational aspects of the control system were not fully developed in time for complete machine operation. While a number of EPICS tools have been successfully used, both for commissioning and operations, other programs have been needed to supplement the EPICS capabilities. Additional programs have been developed both in-house by programmers, operators and scientists and adapted from other laboratories.

I. THE EPICS TOOLS

EPICS includes a number of utility programs which run under Unix and communicate with IOCs (Input-Output Controller) over Ethernet using channel access software. The following describes each tool, how CEBAF has used them and where improvements would improve machine operability.

Graphical User Interface

The EPICS tool MEDM (Motif-based Editor and Display Manager) is used in producing and animating graphical user interfaces. CEBAF has developed over 800 operational display pages using this tool. These pages form the biggest component of CEBAF's operational user interface. MEDM provides the ability to quickly develop and modify pages to meet changing requirements. The editor tool has a menu driven, point and click, WYSIWYG type of interface which requires no programming, so operators and hardware support groups are able to develop or customize display pages for their own individual needs. MEDM is cumbersome to use when developing screens with a large number of display elements, a problem that has been addressed by an engineer who has developed a library of subroutines which allow screens to be generated programmatically.[2] CEBAF has added several features to MEDM, such as bit displays, remapping of visible windows and zero centering bar graphs to customize this tool for our needs. EPICS provides two independent tools and work has begun at LANL to merge

* Supported by U.S. DOE Contract DE-AC05-84-ER40150

these to produce a tool with the features of both. CEBAF will collaborate in this effort.

Data Archival and Archive Data Viewing

The EPICS data archiver tool, AR, is used at CEBAF to log values continuously, or to log values upon some condition. The EPICS archive viewer tool, ARR provides some basic display capabilities for a very limited amount of data. Both the archiver and archive viewer in their current form need a great deal of improvement to be useful and reliable for operations. The callable interface to the archived data needs improvement to make the data more accessible to other analysis programs. The EPICS archive viewer does not have enough capacity to be useful for operations. CEBAF has already written an X-based archive viewer, XARR, that provides basic viewing capabilities for EPICS archived data. This tool needs to be improved and more features added. The EPICS collaboration plans to replace AR and ARR, and CEBAF will participate in this effort. Despite problems with these tools, about 42 megabytes of data are logged each day by operations. Some of this data is used to characterize the machine and some is used by support groups to analyze problems and study their systems.

Alarms

ALH, the EPICS Alarm Handler, is used to notify the operations staff of current or potential problems with machine hardware and software. The magnet system, with over 2000 elements, is the largest system monitored by ALH. For each magnet, two different parameters are monitored. Other systems incorporated into the alarm handler include Viewers, Vacuum and Valves, and Controls. The alarm handler also includes the ability to log alarm data and this information is used by support groups to determine which components of the system may potentially fail, or are inherently unreliable. The most frequently used alarm handler is the magnet page. This page allows rapid detection of magnet errors that can cause critical, though subtle, changes in the electron beam optics. These errors can be disguised by the orbit feedback loops that are used to correct the orbit through each arc.

Save and Restore

BURT, the EPICS Backup and Restore Tool is used to save and restore operational machine parameters. In general, this tool has worked well and proven to be reliable at saving and restoring data. Many thousands of data files have been saved for various parts of the accelerator and to reflect different configurations for operations. While the tool itself works well, it lacks any useful user interface. In order to solve this problem, operations personnel developed an interface to BURT using TCL. This interface makes it trivial for an operator to save or restore predefined sets of parameters from any combination of geographical regions of the accelerator. It also allows saved files to be compared for differences.

II. THE TACL TO EPICS INTERFACE

In order to facilitate a smooth transition to EPICS, an interface which allowed TACL displays to access EPICS data was developed. This allowed the new systems to be tested using existing interfaces and allowed EPICS databases to be installed and operational before time-consuming operator interfaces were completed. The CEBAF cryogenic system also remains under TACL control, so this interface allows communication between the two control systems. While this approach allowed a smooth transition and installation of new software during machine commissioning, it added to the work of the operations staff. Firstly, having two control systems meant training all new operators to use both systems. Furthermore, the two systems run on two physically distinct consoles meaning the operator has to use two different keyboards, screens and mice. As more of the control system has been converted to EPICS, expertise with the TACL systems is decreasing, making troubleshooting difficult and more time consuming.

III. SOFTWARE DEVELOPMENT BY OPERATIONS AND ACCELERATOR PHYSICS STAFF

Due to the overwhelming amount of work involved in changing control systems, many of the high-level applications software programs are prototypes developed by Operations and Accelerator Physics personnel. Otherwise, these applications could not have been made available in time for commissioning. One important result was a closer working relationship between the software and accelerator operations groups. This situation brought on the additional advantage of allowing rapid feedback on algorithm errors found during use of the application. In general, an application was developed when it became necessary or would greatly enhance commissioning. The major applications developed in this way are for save/restore (BURT GUI), orbit and energy feedback locks, automatic and noninvasive cresting of the linac cavity phases (KREST program), Linac Energy Management (LEM) for setting energy and optics in the linacs, automatic orbit steering of the linac and arcs, comment data base for RF cavities and beam position monitors and on-line procedures. Each of these is discussed in more detail in the following sections.

Display Generation Library

As discussed in the section on EPICS Tools, the use of MEDM is inefficient for building complex screens with a large number of signals. In order to address this problem, a library of C functions was developed by an Operations crew chief. These functions can be used to generate display pages. This allows the quick development of complex screens and allows similar screens to be developed quickly, simply by loading a new set of signal names and executing the program again. Not only have these functions saved many hours in generating display pages, they have also created more uniform and effective displays than could have been created by hand. In general, the pages created by using these programs contain far more information content than comparable hand drawn displays.

Slow Feedback Locks

The orbit and energy feedback locks were developed in TCL by a single member of the Accelerator Physics Group[3]. The locks were found to be necessary when slow drifts of the RF over the span of minutes to hours made the setup of the optics extremely tedious. Status of the RF had to be monitored and adjusted every few hours. The application had immediate impact on improving the efficiency of commissioning. Launch angles out of the injector, linacs and arcs can be reproduced. This allows the energy to be set and accurately locked to all beam position monitors in the arcs. One major drawback is that it can only be effectively maintained by the person who wrote it. Moreover, since it was developed rapidly, the program does not have mature error handling, thus increasing the number of faults the program experiences.

On-line Maintenance Comments

CEBAF has a total of 338 superconducting cavities, only two of which are individually critical to beam operation. As cavities fail, they are taken off-line and then repaired during scheduled machine maintenance periods. Though a summary of the status of every cavity in the accelerator is shown on an MEDM screen, the most effective method to access a more detailed history of individual cavities was to establish a database that could be easily modified by the person shutting down or repairing the given cavity. In order to easily access and update the database, a TCL interface was written using key buttons and comment fields. In addition, the name of the person entering the comment as well as the date are automatically recorded. Installation of this database has proven to be an effective method of communicating cavity status and problems to the RF maintenance group, as well as to operations. It has also been implemented to track detailed status for beam position monitors.

Automated Linac Phasing

The KREST application arose when drifting phases in the RF, due to diurnal temperature changes, caused unacceptable energy spread in the beam.[4] Depending on the request file, KREST will change the phase of individual cavities, cryomodules or an entire linac. After changing the phase, KREST monitors the cavity gradients used by energy lock to see how the cavities respond. By changing the phase, the KREST program can determine where the peak of the RF phase is located. Since each cavity phase need only be changed by small amounts to achieve the desired precision, the total energy in the accelerator is not affected significantly. Therefore, assuming the energy locks are running, KREST can usually be run in the background during commissioning. If a cavity gradient becomes unstable or beam energy is deliberately changed while KREST is running, a cavity can be left substantially off-crest. Often this condition is not discovered unless one specifically pulls up the KREST history file. KREST takes approximately 2-3 hours to crest all cavities in a linac.

Save/Restore Interface

In order to provide a usable interface to the EPICS save and restore program, a physicist develop the BURT Graphical User Interface (BURT GUI).[5] This Tcl/Tk application displays a list of files available to restore, along with the date each file was updated and the name of the operator who saved the file. A comments field is also displayed. This interface also allows two BURT files to be compared for differences. This has proven to be a valuable diagnostic tool.

Linac Energy Management

During early commissioning, it was soon discovered that cavities were being lost at a rate of 3-6 per day. When a cavity is lost, the energy must be compensated for elsewhere in the linac and the optics, set for 120° phase advance per period, would usually have to be reset. Resetting the linac optics manually, after much practice, could be done in one half hour. The Linac Energy Management (LEM) program was written to automatically redistribute the gradients and recalculate and load the optics using the optics code DIMAD. As with the slow feedback locks, the program is maintained by one individual and is not robustly coded.

Beam Steering

The AUTOSTEER program serves several purposes. It is used primarily to minimize the orbit through certain sections of the

machine, such as an arc or linac. In the linac, autosteer simply flattens the orbit of the first pass. In the arcs, energy lock not only flattens the orbit but can also be used to measure changes in beam energy to 10^{-4} . Absolute energy measurement accuracy is $\sim 10^{-3}$. Effects of the earth's field can be included in the calculation of the orbit and energy but this causes an error in the measurement of the absolute energy.

On-line Procedures

Instead of the traditional method of having a notebook of procedures on paper, the CEBAF operational procedures are written using FrameMaker and utilize hypertext links for document organization. This allows the procedures to be displayed on the X-terminal alongside other control system displays. The operator can always find the procedures and modified versions can be stored on-line as a procedure is updated. Operational experience has shown that it is more straightforward to have hard copies of procedures to make comments on rather than a framemaker file on screen. Therefore, there is also the option to simply print out the procedure and marked up hardcopies are then directed to the authors.

IV. APPLICATIONS ADAPTED FROM OTHER LABS

In addition to joining the EPICS collaboration, CEBAF strives to take advantage of common needs and adapt programs from other labs to reduce software development time and expenses. We have successfully integrated a trouble reporting program and an optics diagnostic tool from SLAC (Stanford Linear Accelerator). We have plans to adopt the electronic logbook used at the Advanced Photon Source (APS).

Trouble Reporting

Once commissioning was well underway, the Hardware Reliability Team recognized a need for a database-driven problem tracking program. With a small amount of software work (as compared to a full development) to add custom menu items and help, a program written at SLAC was adapted. The CATER (Computer Aided Trouble and Error Reporting) program has now been in use for two years at CEBAF. During this time, nearly 4000 trouble reports have been entered and routed to the appropriate groups for action. This program has provided a consistent method for communication between operations and support groups and allows on-demand tracking of problems. Since the information is stored in a database, it is available for trending and failure analysis. The main difficulty with this system is the lack of a modern user interface and analysis tools, thus making it difficult to perform some functions. In addition, CATER is a VMS-based application, making any future integration with UNIX-based control system tools difficult and prohibitively expensive.

Optics Diagnostics Tool

For on-line analysis of beam orbit deviations, the RESOLVE program developed at SLAC has been imported to CEBAF. By experimentally producing orbit and energy changes at certain points in the accelerator, deviations of the machine orbit compared with the orbit analytically calculated by RESOLVE can quickly indicate a faulty element. Like CATER, the version of RESOLVE currently used at CEBAF operates on a VMS machine. Therefore, data must be transferred to that computer, substantially slowing the analysis process. In addition, the current program configuration allows only approximately half a pass of optics to be analyzed at a given time. Significant amounts of data manipulation must be done to observe orbit changes through multiple passes, thus making the analysis turnaround too lengthy. Both of these problems are being addressed.

Electronic Logbook

CEBAF is currently planning to adopt the an on-line logbook similar to the e-log used by the APS at Argonne. The system uses FrameMaker and World Wide Web to enter and display control room logbook information. The use of these tools coupled with a scanner will allow paper and computer documents as well as on screen graphics to be quickly incorporated into the logbook. The hypertext capability will allow better information organization. Other advantages of having the logbook on-line are the ability to use computer search functions and the ability to display the information to multiple users simultaneously.

V. CONCLUSION

In addition to the EPICS toolkit developed by a collaboration of laboratories, CEBAF has used software from a variety of

sources to commission and operate the accelerator. The use of X-Windows allows many different applications to be used in a common environment. Software has been written by programmers, operators, physicists, and students. Scientists and support groups work closely with software developers to develop requirements for new projects. Operations personnel have taken an active role in developing prototypes of programs immediately needed for operational use. This method has allowed operations and commissioning to proceed while more robust and full featured applications are developed by the Controls Software Department. Several programs have also been adapted from other laboratories for use at CEBAF. Only by taking this team approach to integrate applications from many sources has the controls software reached a level sufficient for machine operations after a mid-commissioning control system change.

VI. REFERENCES

- [1] Karen S. White, et. al., "The Migration of the CEBAF Accelerator Control System from TACL to EPICS", CEBAF Control System Review (1994).
- [2] Jeff Karn, "Code Driven Construction of MEDM Screens", CEBAF-TN-95-038 (1995).
- [3] Johannes Van Zeijts, "Rapid Application Development Using the Tcl/Tk Language", (PAC 1995).
- [4] M. Tiefenback, K. Brown, CEBAF Internal Documentation.
- [5] Bruce Dunham, "Control System Data from an Operational Point of View", (Proceeding of this conference).

Development of Fast Output VME Modules with External Clocks

E.Takada, N.Araki, A.Itano* , M.Kanazawa, M.Kumada, K.Noda, S.Sato
Accelerator Physics & Engineering Division, National Institute of Radiological Sciences, Chiba 263, Japan
* Public Health & Environment Department, Hyogo Prefectural Government, Kobe 650, Japan

E.Hishitani and Y.Yamamoto
Electronic & Control Systems Department, Hitachi Zosen Corporation, Osaka 554, Japan

ABSTRACT

Heavy ion acceleration in a synchrotron requires dynamic control of lattice magnets and the rf system to cover a wide variety of species and energies of ions. For example, spill-by-spill energy change is necessary for three dimensional scanning in cancer radiotherapy. The excitation pattern of lattice magnets must be switched without interruption to beam, while the rf system should follow the frequency shift caused acceleration. Since both magnet power supplies and the rf system (acceleration voltage, frequency, ferrite bias current, etc.) are driven by digital settings from VME modules, although with different clock systems, in the HIMAC synchrotron, development of a new VME module that can deal more dynamically with both magnets and rf in a unified manner is initiated as a logical extension of the present system.

Features include:

- (1) 20bit (max.) pattern output with 125kHz response to external bi-directional clock triggers
- (2) Versatile pattern switching with Look-Up-Table and selection command input
- (3) Feed-back loop capability through VSB bus, with high performance RISC microprocessor

System construction and firmware concepts will be discussed also.

1. INTRODUCTION

Operation of synchrotron magnets and rf system are characterized by repetitive patterns which consist of injection, acceleration, extraction, deceleration and reset of beam. In order to accelerate the beam without losses, each field of the lattice magnets, Bending Magnets (BM) and Quadrupoles (QF/QD), should be excited in a matched strength, which is called "tracking". Power supplies of lattice magnets, accordingly, should be precisely controlled in synchronization with the thyristor trigger timing pulse. Therefore it is essential for the tracking be synchronized by means of the external clock supplied from a phase locked loop. HIMAC has already made use of Fast Digital Input/Output (FDI/FDO) modules which were developed for this purpose [1]. The rf system should also be operated in coordination with the main magnet excitation pattern, but the clock system differs from that of main magnet power supply. A bi-directional B clock of the rf system drives several patterns of frequency and voltage together with beam feed-back loops to stay with fluctuations of magnetic field for both acceleration and deceleration of beam [2]. The present control system of magnet power supplies and timing system works well. However, an advance in heavy ion therapy requires more flexible and dynamic control, as the beam delivery with pulse-to-pulse energy shift is envisaged for cancer treatment irradiation of three dimensional scanning for HIMAC and Particle Therapy Project in Hyogo prefecture [3]. A dynamic pattern operation is required to enable raster scanning, as the beam has to be swept in a Zig-zag line by controlling scanner magnets. Dynamic switching of these patterns (both magnets and rf) is demanded in various aspects of beam operations. For example, it is required to switch an arbitrary combination of magnets simultaneously during beam tuning or to use a set of excitation patterns as mentioned above. Usage is also expected to measure beam parameters, e.g. chromaticity, to initialize magnets, and to achieve digital feed-back control.

In order to select various excitation patterns of main magnets and rf system dynamically, Dynamic Pattern Input/Output (DPI/DPO) modules are now under development. This paper describes the DPO module with possible system construction.

2. SYSTEM CONSTRUCTION

DPI/DPO modules will be installed in subsystems where VME is used as a major equipment controller. An advanced facility such as Hyogo Prefecture's Project is expected to utilize these modules. Another example is the present HIMAC synchrotron control system. It is hierarchically structured by a network consisting of a main computer (CS) with two 20" & 14" displays, and equipment group controllers of 14 VME crates, an rf control computer (RC), a beam transport control computer and Programable Logic Controller. VME crates include timing system (TS) and Magnet Power-supply Controllers (PC), whose software is downloaded from a program server workstation (WS) [1].

The CS serves man-machine interfacing of the presentation layer. Conditions of exciting patterns of magnets etc. are set on the console displays after a decision as to the mode of operation, such as initialization of magnets, normal repetitive operation, spill-by-spill energy shift mode, adjustment of parameters or measurement of beam. These conditions will be encoded and sent to DPO as external control signals.

The TS is a system for which the main functions are distribution of base and master clocks and generation of various event signals response to CS commands. Event signals can be generated and controlled by DPO modules. Clock and event signals will then be fed to the DPI/DPOs that control magnets and rf system. DPI/DPO will be the main ingredients of PC, TS, or RC when they are constructed as VMEbus systems.

3. DESIGN BACKGROUND AND SPECIFICATION

The repetitive operation of lattice magnets and rf cavities in our synchrotron should be carried out independently in closed cycles in the equipment control layer, unless any commands or interrupts occur. The tracking should also be kept in synchronization with the basic clocks. Therefore DPI/DPO modules play essential parts in this layer. They should be able to communicate in real time with the host CPU of the IOC in the VMEbus system, to prepare many files of binary excitation pattern data with necessary processing on the basic data, and to respond to various external operation commands. In addition, they should have flexibility to utilize various application programs, which are downloaded from the program server on the workstation via IOC on VMEbus systems when booted.

Basic functions of DPI/DPO modules follow.

3-1. External clock

DPI/DPO are supplied two kinds of basic clocks. One of these functions at 1200Hz, and supplies a trigger timing pulse for 24 phases in a 50Hz thyrister converter at the lattice magnet power supply. The other is T-clock/B-clock which are used to control the rf system. T-clock is at a constant frequency of 50kHz while B-clock is variable up to 120kHz, in order to cover the BM excitation rate of 2.4T/sec at 0.2gauss/pulse, for both increasing and decreasing fields.

3-2. Pattern memory

DPO must be able to expand the excitation pattern data from a basic trapezoid and to store the data. The necessary precision of BM data is 18 bits, while that of the rf system is 20bits, to cover an 8Mhz span in steps. To provide sufficient capacity 16MB capacity is assumed for pattern memory.

3-3. External control signal

It should be able to select dynamically a required one from within several tens to hundreds of excitation patterns. For this purpose an encoded 12bit external control signal is prepared. The decoding method is effective in expanding the range of selection, although a surplus strobe signal is necessary. The module is capable of selecting not only any single excitation pattern but also a set of patterns, and to operate grouped magnets by means of masking bits, which are applied to Start or Stop events when the initialization of a group of magnets is required.

3-4. Look-Up-Table

It provides a Look-Up-Table (LUT) to follow the incremental or decremental B clock of the rf system. LUT is a memory which rapidly converts any data placed on the address line to other values. This feature is useful in converting a nominal value of B clock to the rf frequency, which is not always linear with the B-field. Typically 80k table entries are required when BM is excited up to 1.6T with a B clock of 0.2gauss/tick.

3-5. VSBbus

VSBbus should make it possible to form a closed loop of digital feedback control by a direct connection between DPI and DPO. This feature is necessary to achieve "Iterative control" more quickly. Iterative control is effective in controlling the driving voltage of a power supply so that the current deviation can be reduced to a tolerable value by means of a digital filter. In the present HIMAC system, the host cpu module of the VME crate performs the iteration together with the FDI and FDO [4].

4. FEATURES OF THE DPO HARDWARE

A number of high performance RISC microprocessors for embedded systems have flourished in a wide range of electronic appliances. After our investigation of ability to handle external interrupts and to control timers/counters for various functions, we have made a choice of the Hitachi Super-H (SH) series CPU, which uses a 32-bit RISC core optimized for high speed and low power consumption. Another advantage of the SH-1 is that it contains several on-chip peripherals to eliminate a number of additional devices.

Table 1 shows the outline specification of DPO, for which the block diagram is shown in Fig.1. A multiple CPU system is provided on this module. The SH-1 CPU serves to control communications with VMEbus, VSBbus and all input and output ports. SH-1 also manages the SH-2 CPU, which is provided for dedicated calculation. To avoid bus contention of multiple CPUs, a separate local bus is provided for each. Pattern memory of 16MB will be used in a 3 byte/word format to cover the system requirements for rf and BM operation.

User supplied programs can be loaded into the DRAM of SH-1 or the dedicated SRAM of SH-2 and executed locally. Each CPU responds to commands and environment parameters placed in each 32kB dual-ported SRAM, is accessed from both the VMEbus and the local bus through synchronous arbitration logic.

SH-2 resides on a mezzanine board to be upgradable to an SH-3 CPU in the future. Input/output ports are also provided on the other mezzanine boards to change RS485 to TTL etc.

Table 2 shows 4 external clocks and control signals. A 12bit control signal is decoded into various 'events'. These events are executed after the next master clock pulse except for Change signals, which are executed immediately. The LUT consists of dual 80k x 20-bit memories, of which one serves to form data output while the other receives the next input.

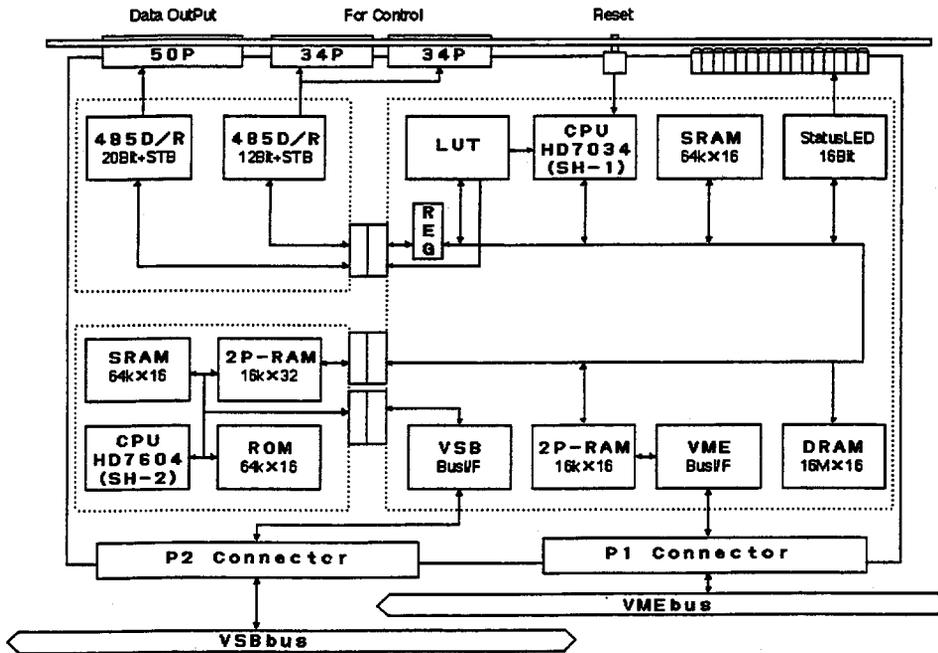


Fig1. Block Diagram of DPO

Table 1. Outline Specification of DPO

	Main Board	Mezzanine Board for calculation	Mezzanine Board for I/O adaptation
CPU	SH7034(SH-1) 4kB On-chip RAM 64kB On-chip ROM Hardware multiplier 9ch. Ext. interrupt handler 4ch. DMA controller 5ch. Timer/Counter 16bit Timing pattern gen. Watch-dog timer 2ch. Serial comm. port Universal I/O ports	SH7604(SH-2) 4kB cache memory Hardware multiplier Hardware divider 15ch. Ext. interrupt handler 2ch. DMA controller 16bit Counter Watch-dog timer 1ch. Serial comm. port	Bi-directional two ports of 4bit clock and 12bit control signal with strobe (Opto-isolated RS485 interface) 20bit+Strobe output port (Opto-isolated RS485 interface)
Clock	20MHz (16MIPS)	28.7MHz (25MIPS)	
SRAM	128kB with zero wait state	128kB with zero wait state	
DRAM	16MB with zero wait state	128kB with zero wait state	
Dual-ported SRAM	32kB with one clock wait state	32kB with one clock wait state	
LUT SRAM	384kB with 20bit U/D counter		
Front panel functions	16bit Status LED Reset switch		
Bus spec.	VMEbus IEC821 compatible slave interface (A24,D16) VSBbus IEC821 compatible master/slave interface		
Operating condition	Power source: 5V±5%, 3A 5V±10%, typ.1.5A (I/O isolation p/s supplied from front connector) Temperature : 0°C~50°C Humidity : 30%~90% (Non-condensing)		

5. FEATURES OF DPO SOFTWARE

Most of the intelligence of DPO resides in the SH-1CPU, which handles numerous internal/external interrupts and controls many peripherals. In order to reduce any overhead time, the firmware has no OS except for a minimum context switching kernel which arbitrates task priorities when interrupts occur.

When the host CPU on VMEbus receives a command from the CS via TCP/IP, it interrupts the DPO through the dual-ported memory, e.g. dpo-set, dpo-read, dpo-write, dpo-init, dpo-clear, etc. DPO responds to the host on any interrupt. Base and master clocks, similar to T and B clocks from the RC, and all of event signals from the TS are also recognized as interrupts by DPO; Table 3 lists the software functions.

DPO firmware has some optional functions to change smoothly the rf frequency from T clock region to B clock region, to excite current of a magnet from one flat-top region to another and to turn on and off the magnet power supply.

Table 2 External clocks & control signals

Code	Name	Contents
-	Master clock	Pattern start signal. It corresponds to repetitive cycle of synchrotron operation, typically ~1Hz. This is also used as rf capture.
-	Base clock	1,200Hz is fundamental clock distributed from PLL.50Hz is also used to 24 phase thyristor trigger timing with zero cross on U phase.
-	B+ clock	Incremental B clock for rf.
-	B- clock	Decremental B clock for rf.
001	Start	Command to start pattern output/input to devices altogether.
002	Stop	Command to stop pattern output/input to devices altogether.
004	Pause	Command to break temporarily pattern output/input.
008	Rerun	Command to rerun the pause devices.
010	T - B	Command to change signal from T clock to B clock.
020	B - T	Command to change signal from B clock to T clock.
801~	Select	Command to select any excitation pattern.

Table 3 Software functions of DPO

SH-1 CPU for control	SH-2 CPU for calculation
Communication with Host CPU of IOC on VMEbus.	Communication with SH-1 CPU for control.
Communication with SH-2 CPU for calculation.	Calculation of feed-back control or data filtering
Excution of event signal.	Smoothing calculation of pattern.
VMEbus and VSBbus control.	
Initialization and placing data on LUT.	
Smoothing control of pattern.	
Data output synchronized with external clock.	

6. DEVELOPMENT SCHEDULE

DPO with basic firmware will be prepared for a performance test by the end of this year. Measurements of response times needed for critical applications e.g. a smooth change from B and T clock of rf, realtime operations of feedback control, etc., will be carried out. The fine adjustment of event control timing in microseconds will have to be checked to use DPO for the TS. System considerations are necessary to specify further details and to establish the best means of utilization. Especially, an optimization of the division between hardware and software is essential, as they are usually contradictory to performance and diversification.

High-speed digital feed-back control with high accuracy is attractive to challenge many technical matters of instabilities in various aspects of synchrotron operation. We expect that the combination of DPI/DPO will be useful to reduce beam ripple or to stabilize beam position by means of high-speed data acquisition and processing.

ACKNOWLEDGMENTS

We would like to acknowledge those who developed the present control system of HIMAC and enlightened us through discussions. Especially, Mr.N.Tsuzuki of Fuchu Works, Toshiba Corp. (rf control), Mr.S.Sakamoto of mtt Instr. Inc. (FDI/FDO hardware), Mr.T.Nakayama of Hitachi Inf. & Cont. Systems Inc. (FDI/FDO system) and Prof.K.Sato of Osaka University are acknowledged with sincere gratitude.

We would like to thank Dr.K.Kawachi, Dr.S.Yamada and colleagues of the Accelerator Physics & Engineering Division of NIRS for discussion.

REFERENCES

1. E.Takada et al., "The HIMAC Synchrotron Control System", Proc ICALEPCS'93. Berlin, Germany, October 1993, pp.40-42, (Nucl. Instr. & Meth A352, ('94)) The article has a typographical error on the name of realtime OS. It should read PDOS, instead of PSOS.
2. E.Takada et al., "Successful Application of Digitally Controlled RF Accelerating System for HIMAC Synchrotron HEACC'92, Hamburg, Germany, July 1992, pp.763-765.
3. E.Takada et al., "Synchrotron Control System of HIMAC", Proc. 9th Symp. Acc. Sci. Tech., KEK, Japan, November 1993, pp.73-75.
4. A.Itano et al., "Heavy Ion Medical Accelerator Project by Hyogo Prefectural Government", Proc. 10th SAST, Hitachi-Naka, Japan, October 1995, pp.398-400.
5. E.Takada, T.Nakayama, E.Hishitani, "VME Control System for Synchrotrons of Heavy Ion Medical Accelerator in Chiba" (text in Japanese), HITACHI ZOSEN TECHNICAL REVIEW 53, December 1992, pp.317-323.

Status Report of the HIMAC Control System

--Reliable beam supply for cancer radiotherapy with carbon beam--

E. Takada, N. Araki, T. Kohno[#], M. Torikoshi, S. Yamada, and Accelerator Group.
Accelerator Physics & Engineering Division,
National Institute of Radiological Sciences, Inage, Chiba 263, Japan

Abstract

HIMAC has been in operation for nearly two years. During this period, the accelerator complex was commissioned, clinical trial started June 1994, and more than fifty patients treated with a 290 - 400 MeV/u carbon beam, which has been delivered on schedule and with greater than 99% reliability.

Introduction

HIMAC, the Heavy Ion Medical Accelerator, in Chiba was commissioned during November 1993 and February 1994[1]. After the pre-clinical physics and biology experiments, a clinical trial of ion beam therapy started in June 1994, when a 290 MeV/u carbon beam was delivered against tumors. Since October '94, accelerator operation has been extended from 12 hour/sday to 24 hours/day, Monday through Saturday, and basic research experiments in biology and physics have been carried out during night and week-end shifts. Various beams have been supplied for experiments, ranging from He to Ar.

In the first year of operation 55 patients were treated, and preliminary results show the effectiveness of ion beams. This paper presents operational experience of the HIMAC accelerator complex during this period.

Control system of HIMAC

The control system of HIMAC comprises a supervising computer and three "sub-system" controls: Injector[2], Synchrotron[3], and High Energy Beam Transport system[4]. The irradiation control, including beam shaping and dosimetric functions for clinical trials, is carried out at a therapy control area.

The design principle of the control system, concerning operational aspects, is such that: 1) all the equipment is controlled at a "sub-system" console, 2) the operational procedures or sequences are unified to touch panel operations and rotary dials and are common to all subsystems, and 3) parameters such as current settings of magnets are saved as files and can be reloaded. The design was to assure operators easy access to control of the machine. This principle was followed for the most part, but response time, for example, differs somewhat due to different platforms and system constructions among sub-systems.

The injector control system was designed and built earlier than the other subsystems, and has been used for longer time. The CPU for System Control Unit and Group Control Units were upgraded during a scheduled maintenance period in March 1995, and improved system performance substantially. Typically, loading time for the entire injector system from cold start is reduced to 5 minutes from 40 minutes. Application enhancement for various operational needs is also being made; details will be described elsewhere.[5]

Synchrotron controls perform timing control for the overall system and control of repetitive patterns of magnets and rf. As for tracking of main dipole and QF/QD magnets, although we had prepared a tool to form the current waveform from the desired values of magnetic field, it was found that simple scaling of the current pattern between dipole and QF/QD would suffice for daily operation, provided that an iterative adjustment for the driving voltage waveform was carried out. However, more studies are needed to understand the system.[6]

High Energy Beam Transport control has to deal with requests from Irradiation Control for beam control and switching. Since HIMAC comprises two synchrotron rings and four treatment beam ports, as well as two physics ports and a biology port, HEBT control must be capable of exciting proper switching magnets and de-exciting unnecessary ones. Using an operation parameter file that has descriptions of source ring(s) and destination port(s), it can set up the HEBT system for the requested delivery.

The overall controls were designed so that one can set up each subsystem from the supervisor system using a system standard file, while at present daily operation is carried out from each subsystem.

[#] Present Address: Energy Science Dept., Tokyo Inst. of Tech., Yokohama 226, Japan

Operations Framework

Daily operations of the machine are carried out by the crew of operators who belong to a company established for accelerator operation, maintenance and development by manufacturing firms. The operators' prior experience on accelerators varies from null, some even without a college physics degree, to years at other facilities. The man-machine interface of the control system provides tools for operators. Twenty-four hour operation started in October 94, with two shifts of six operators.

The weekly operational schedule is as follows: Monday 9 a.m. to 3 p.m. for preventive maintenance, then the cold start of machine, from 7 p.m. delivering beam to experiments until Tuesday 7 a.m. After tuning the machine for treatment beam, 9 a.m. to 7 p.m. are exclusively for medical treatment and relevant dosimetry. Again with re-tuning, beam is delivered to other experiments until 7 a.m. next day. From Friday evening the time is for experiments usually until Saturday 8 p.m. and sometimes until Sunday morning.

Beam Delivery

In March 1994, we accelerated carbon beam to 290 MeV/u, which has a most reasonable characteristic for initial clinical trial, in both rings. Typical intensity for treatment is 3×10^8 particles per second. In clinical irradiation, beam is used after being modulated by wobbler magnets and scatterers to cover the tumor volume with uniform intensity for treatment. The patient must be immobilized in a pre-determined position so that beam irradiates cancerous parts without damaging healthy organs and tissue. It usually takes 20 - 40 minutes to position the target in a patient, then irradiation needs less than 5 minutes, which takes place typically 18 times per patient. This means that stable and reproducible beam delivery is necessary during a day and also for an extended period of several months. HIMAC has been up to this demand since the first clinical trial. Exceptions are one call-off for the day due to rf trouble in the linac and a few re-schedulings within the day due to various minor troubles such as malfunction of waterflow monitors.

We also accelerated Ne, Si, and Ar up to 600, 800, and 650 MeV/u, respectively. In accelerating these ions the synchrotron main magnets are expected to show field saturation; nevertheless, we could accelerate them within a couple of hours thanks to the existing excitation pattern and scaling function of the control system. Notably, for beams with charge-to-mass ratio of 1/2, parameters can be essentially common as long as the acceleration energy is the same.

Since May 1995 clinical trials have been carried out also with 350 MeV/u for vertical irradiation and 400 MeV/u for horizontal ports, in addition to 290 MeV/u. This represents another challenge in operation: We have to deliver beam of two different energies within an hour or less, and they must both be stable and reproducible. It has been met after establishing parameter files and simulating the situation.

Still another requirement is that beam must be delivered to different irradiation ports without adjusting beam position, etc. at the final irradiation point. This has also been examined and is ready for actual practice.

Most of the tuning effort is to adjust ion source parameters and transport to linacs in the case of the injector system, injection angle, position and timing for synchrotron rings, and centering the extracted beam for both synchrotron and HEBT systems. By analyzing these activities, necessary tuning time for treatment beams is reduced and stability and reproducibility of the beam are also improved.

Development Activities

For improving irradiation characteristics further, several efforts are being made. One is to gate beam extraction by signals from the patient's respiratory motion. This can be realized by applying transverse electric fields to the beam at the flat top.[7] The respiration motion sensor and interlock system is scheduled to begin operation in summer 1996.

A VME module for dynamic switching of the excitation pattern is also under development, and is reported in this conference.[8]

In the injector, it is planned to expand pulse operation to the entire system in order to accelerate different ion species for respective rings in a pulse-by-pulse mode. This will also enable more independent use of the 'medium energy' beam port, which utilizes the 6 MeV/u output beam from the linac. An 18 GHz ECR ion source is also under development for heavier ions such as Fe.

Major effort is also now expended to design and to construct Secondary Beam irradiation ports. With use of positron emitters beam such as ^{11}C , one can expect to monitor the beam irradiation volume by PET technology. It will also provide for experimental research in physics and biology.

Acknowledgments

We would like to acknowledge the crew members of Accelerator Engineering Corporation for continuous efforts on operations, maintenance, and developments. Special thanks are due to Messers C. Kobayashi, Y. Sano, T. Togashi, F. Sasaki, and K. Ueda. We are grateful to Dr. K. Kawachi and colleagues of Accelerator Physics & Engineering Division of NIRS for encouragement and discussion.

References

- [1] K. Sato et al., "Status Report on HIMAC", Nucl. Phys. A588, 299 (1995). (Proc. EPAC94, London)
- [2] T.Kohno et al., "Control system for HIMAC injector", Proc. 7th Symp. Acc. Sci. Tech., Osaka, Japan, 1989, pp.246-248.
- [3] E. Takada et al., "The HIMAC synchrotron control system", Nucl. Instr. & Meth. in Nucl. Res. A352, 40 (1994). (Proc. ICALEPCS'93, Berlin, Germany, October 1993)
- [4] M. Torikoshi et al., "Control system for a high energy beam transport system of HIMAC", Proc. 10th Symp. Acc. Sci. Tech., Hitachi-Naka, Japan, October '95, pp.306-308.
- [5] S. Yamada et al., to be published.
- [6] E. Takada et al., "Present Status of HIMAC Synchrotron Control System", Proc. 10th Symp. Acc. Sci. Tech., Hitachi-Naka, Japan, October '95, pp.398-400.
- [7] K. Noda et al., "A treatment beam control system for irradiation gated by respiration of a patient", Proc. 10th Symp. Acc. Sci. Tech., Hitachi-Naka, Japan, October '95, pp.306-308. See also Proc. EPAC94, pp.982-984.
- [8] E. Takada et al., "Development of Fast Output VME Modules with External Clocks", in these proceedings.

Author Index

I. Abe	863	926	1052	645	713
J. S. Adhikari	197				
S. Ahn	469	176			
N. I. Aizatsky	821				
Yu. I. Akchurin	821				
W. Akers	97	607			
A. Akiyama	899	423	593		
S. Albrand	791				
A. Aleshaev	799				
V. Alferov	464				
A. S. Alimov	501				
D. Anicic	814				
J. Annala	328				
D. Arakawa	963				
N. Araki	1068	1074			
S. Araki	899				
L. Arnaudon	508				
R. Bacher	805				
R. Bailey	1	333			
N. Baker	278				
A. Balestra	808				
B. Balhan	589				
P. Barale	922				
L. Barbina	823				
Yu. Bardik	980				
R. Barillere	20	278			
D. Barr	263				
F. Bartlett	469				
J. F. Bartlett	176				
A. Batrakov	799	258			
C. C. Baum	249				
M. Beharrell	474				
S. Belov	799				
G. Benincasa	474				
J. Bergl	1017				

E. Bernard	20	278
Ch. Bertuzzi	405	
V. C. Bezuglov	534	
M. Bickley	435	
F. Bieser	922	
R. Billen	440	
T. Birke	648	
D. Bishop	837	
E. A. Bjorklund	217	
D. Blanc	770	
A. Bland	74	
W. Blokland	226	
T. Blumer	814	
J. P. Bobis	547	
C. J. Bocchetta	949	376
V. N. Boriskin	821	
M. Borland	382	653
C. Bortolotto	823	
H. Boterenbrood	1007	
J. M. Bouché	474	
J. Bovier	57	335
B. Bowling	435	481
P. S. Bowling	249	
D. Brahy	398	
J. C. L. Brazier	351	
C. Briegel	983	
S. K. Brown	249	217
K. L. Buescher	249	
D. Bulfone	823	
R. Burke	922	
A. J. Burns	524	
A. Burton	589	
K. Cahill	420	684
N. T. Callaway	217	
E. Carlier	589	485
G. P. Carr	217	
T. J. Carroll	633	
J. A. Carwardine	489	

J. Casas-Cubillos	398	
J. Chan	97	
P. Charrue	828	74
B. Chase	1035	
H. Cheng	497	
A. S. Chepurnov	501	209
M. Cherney	922	
F. Chevrier	504	
G. Chiozzi	121	
M. P. Chowdhary	429	
J. Chrin	922	
H. P. Christiansen	50	
Y. Chung	263	547
S. Churcher	91	
E. Ciapala	508	
D. J. Ciarlette	322	
M. Clausen	360	397
M. J. Clayton	828	
T. Clifford	718	142
P. N. Clout	223	
P. Colarco	922	
R. J. Colchester	514	
R. Cole	937	
G. Crockford	504	
M. Crowley-Milling	iii	
G. Csuka	397	
J. Cuperus	474	148
G. Daems	1056	
D. Dale	837	
L. R. Dalesio	360	217
L. David	763	
G. Decker	263	
N. de Metz-Noblat	543	663
N. V. Demidov	821	
B. Denis	244	
G. de Rijk	504	
V. A. Deryuga	954	
J. C. deVries	524	

F. DiMaio	105	543		
J. H. Dieperink	589			
A. H. Dinius	351			
A. A. Dorochin	209			
A. N. Dovbnya	821			
A. Dubrovin	799			
P. Duneau	763			
B. Dunham	1046			
D. B. Duran	217			
P. Duval	843	805		
Yu. Eidelman	799			
D. E. Eisert	851	667		
L. Emery	382	653		
F. Epaud	185			
C. Escrihuela	20	278		
K. Evans	756			
T. Ewert	837			
J. A. Faucett	217			
V. D. Fedorov	534			
S. K. Feng	857	988		
J. Feres	514			
J. Fontaine	405			
J. D. Fox	248			
J. C. Frances	405			
A. Franck	558			
V. Fransev	910			
S. Fuess	469	602	176	
S. Fujiwara	584	672	201	
T. Fukui	584	625	672	201
H. Fukuma	423	593		
K. Fullett	1035			
K. Furukawa	863	1052		
A. Gagnaire	148			
I. V. Galachov	534			
J. Galayda	263			
R. Garoby	543			
R. Garrett	867			
A. N. Gascheev	534			

R. Gavaggio	538		
M. D. Geib	299	676	237
X. Geng	874	751	
V. P. Gerasimov	534		
R. Gerig	322		
F. Ghinet	74		
P. Gillette	561		
J. Gilot	57		
S. Goloborodko	877		
V. Goluzov	910		
V. V. Gotsev	994		
F. Gougnaud	881		
M. Gourber-Pace	543		
J. F. Gournay	881		
J. J. Gras	514		
M. Grippeling	398		
J. Gross	922		
K. A. Gudkov	209		
L. Guerrero	998		
A. Guiard-Marigny	405		
D. P. Gurd	217		
V. A. Gurin	821		
A. M. Gusev	534		
A. Götz	105		
V. M. Hanagandi	249		
S. Hancock	543		
M. W. Hardy	217		
M. P. Harrington	217		
W. Harris	20		
D. Harrison	837		
R. G. K. Hart	1007		
E. Hatziangeli	440		
B. Hendricks	684	689	
S. Herb	887	805	
D. A. Herrup	572		
W. P. J. Heubers	1007		
G. Heyes	97		
A. Hilaire	504		

J. O. Hill	136	217	
R. F. Hinde	249		
H. Hindi	248		
E. Hishitani	1068		
L. Hoff	142		
L. T. Hoff	680	718	
J. A. Holt	413		
T. Hori	639	726	
S. Howry	867		
T. Huang	890		
J. Hunter	922		
F. Iazzourene	949	376	
B. S. Ishkhanov	501		
A. Itano	1068		
Y. Itoh	639		
I. Ivanov	684		
S. Jacobsen	922		
J. P. Jensen	397		
I. Jirousek	814		
R. D. Jones	249		
F. Josa	405		
R. Jung	514		
A. Kadnikov	910		
A. I. Kalinichenko	954		
A. Kalinin	799		
E. Kallistratov	980		
S. Kamada	423		
N. Kamikubota	863	894	1052
M. Kanazawa	1068		
N. Kanda	894		
E. Karantzoulis	949	376	
S. Karnaev	799	786	
A. V. Kartashov	977		
T. Katoh	899		
T. Kawamoto	899	423	
T. Kean	91		
D. Kehne	1063		
D. M. Kerstiens	217		

A. Kholodnyi	877		
M. Kikuchi	423		
J. H. Kim	903		
J. M. Kim	903		
S. C. Kim	903		
J. A. Kirchman	547		
D. U. Kisorzhevsky	1030		
W. Kissel	558		
S. Klein	922		
W. B. Klein	746		
M. Knott	370		
I. S. Ko	903		
M. Kodera	639		
T. Kohno	1074		
M. Kollegov	799		
I. M. Koltsov	908		
K. Komada	899		
A. A. Komarov	908		
V. Komarov	930	994	1012
J. Koopman	514		
L. Kopylov	689	693	
V. Kovaltsov	915		
J. B. Kowalkowski	1022	113	598
V. Kozak	799		
A. J. Kozuba	217		
G. A. Krafft	429		
M. R. Kraimer	360	620	
U. Krause	14		
O. Krebs	397		
Yu. A. Kresnin	954		
Y. Krylov	910		
S. Krzywdzinski	469	602	176
A. Kuba	639	726	
K. Kudo	899		
M. Kumada	1068		
K. Kumagai	625		
B. Kuner	1017		
E. Kuper	799	786	

S. Kuroda	423			
S. Kurokawa	899			
V. A. Kushnir	821			
S. Kuznetsov	910	258		698
M. N. Lack	633			
H. Laeger	775	780		154
R. Laird	345	567		
J. Lam	837			
M. Lamont	316			
R. Lange	1017	648		
R. J. Lauckner	457			
I. Laugier	440			
D. Lavielle	398			
J-M. LeGoff	20	278		
M. LeRoss	837			
C. Leboucher	561			
S. Lechner	780	154		
E. Lécorché	561	763		
A. Ledenev	799	786		
G. S. Lee	903			
M. Lelaizant	474			
F. Lenkszus	263	345		567
G. Leo	398			
P. Lermine	561	763		
B. Levichev	799	786		
E. Levichev	258			
J. Lewis	915	703		
V. N. Liamin	1030			
P. Lienard	998			
W. Liu	874			
I. V. Lobov	994	1012		
N. S. Lockyer	181			
M. Lonza	823			
B. Lopez	514			
B. Lublinsky	558	572	576	580
P. Lucas	iii			
K. Lüddecke	937			
G. F. Luger	746			

A. Lumpkin	263			
H. Luo	874	751		
Yu. N. Lusin	977			
Y. Lussignol	881			
B. MacKinnon	469			
A. Machnachev	980			
L. Madaro	398			
R. Marizza	823			
R. Martini	775	154		
V. Martz	707			
K. Marutsuka	963			
A. Mason	1027			
T. Masuda	584	625	672	201
A. Matiouchine	980			
S. Matsumoto	423			
A. A. Matyushin	994			
C. Maugeais	561	763		
Y. Maumary	504			
R. McClatchey	20	278		
C. McClure	1035	1040		
E. McCrory	8	128		
W. McDowell	68	73		
R. J. McGonegal	284	167		
C. McParland	922			
T. McShane	922			
J. Meier	922			
I. Mejuev	713			
L. Merard	474			
V. Mertens	589	485		
K. H. Mess	805			
J. Meyer	105			
A. Mezger	814			
P. Michelini	823			
M. Mikheev	689	693		
T. Mimashi	899	423	593	
Sv. Mishnev	799			
S. Mishra	8			
V. V. Mitrochenko	821			

A. Mizuno	639			
E. Mizuno	894			
V. E. Mnuskin	209			
F. Momal	62			
M. Moscatello	561			
I. V. Mozin	534			
R. Muller	648			
V. M. Murugov	534			
M. Mutoh	926			
I. Müller	1017			
R. Müller	1017			
R. Nagaoka	949	-376		
T. Naitoh	899			
F. J. Naivar	217			
K. Nakahara	863	1052	645	713
T. T. Nakamura	899			
W. Namkung	903			
A. Naumenkov	799			
G. J. Nawrocki	598	620		
A. I. Nebogin	977			
M. C. Needes	217			
G. Neu	937			
P. Ninin	775	154		
C. Niquille	504			
K. Noda	1068			
G. Obukhov	980			
M. Ohashi	894			
J. Ohnishi	625			
T. Oide	423			
R. Olsen	988			
R. H. Olsen	718			
M. A. Oothoudt	943	217		
J-F. Ostiguy	8			
D. J. Ostrem	217			
E. Oustinov	877			
P. Pain	561			
V. P. Panov	1030			
Yu. V. Papazyan	977			

V. Paris	504		
W. J. Parkinson	249		
F. Pasian	808		
L. Paterno	469	602	176
M. Patz	86		
F. Perriollat	1056	367	
J. G. Pett	351		
G. Pfeiffer	1017		
E. Pfirsch	775		
P. Pietraski	988		
C. Pinto-Pereira	62		
G. Piskunov	799		
M. Plesko	949	376	
R. Poboni	269		
J. Poole	446		
A. V. Popov	994	1012	
G. F. Popov	954		
F. Potepan	823		
J. M. Potter	217		
B. V. Prosin	1030		
H. B. Prosper	602		
I. Protopopov	799	786	
M. Pucillo	808		
R. Pugliese	269		
B. A. Quintana	217		
M. Rabany	392	191	
F. Radovcic	823		
G. Raffi	162		
J. Rahn	1017		
L. Rasmussen	176		
G. Raupp	937		
R. Rausch	457		
I. Reguerro	440		
O. Reisacher	50		
P. Rewiersma	1007		
P. Ribeiro	50	74	504
A. Rijllart	398		
M. Rivers	184		

O. V. Romanov	1030			
R. Rothrock	707	223		
W. von Rüden	466			
H. Rüdiger	1017			
V. M. Rybin	908	722		
G. V. Rybina	722			
V. Rytchenkov	1027			
R. Saban	398			
V. Sajaev	258			
H. Sakaki	639	726		
P. Santin	808			
D. Sarjantson	154			
S. Sathe	142			
H. Sato	963			
S. Sato	1068			
A. N. Savchenko	821			
J. J. Savioz	524			
A. N. Savochkin	977			
C. Scafuri	823			
S. Schaffner	607			
S. C. Schaller	943	217		
P. Schoessow	iii			
W. Schütte	958			
C. L. Seaver	598	620		
N. Segura	440			
L. Sekolec	814			
N. Sereno	382			
Ch. Serre	1056			
Y. Seth	197			
D. Shatilov	799			
T. J. Shea	80			
F. E. Shelley	217			
Y. Shibasaki	926			
V. Shilo	258			
M. Shirakata	963			
A. Shirakawa	863	1052		
H. Shoae	414	429	481	607
A. V. Shumakov	209			

V. I. Shvedunov	501			
C. H. Sicard	305			
D. P. Siddons	857	988		
K. V. Sidorowicz	68			
E. Simonov	799			
P. Skarek	703			
M. Skiadelli	398			
J. Skinner	857			
V. Smaljuk	799			
A. Smirnov	799			
G. Smith	504			
J. Smolucha	1035			
P. Sollander	770	775	154	
M. Stanek	312			
M. Steffensen	612	693		
S. J. Stein	615	620		
R. Steiner	14			
N. G. Stervoedov	954			
M. W. Stettler	217			
R. Stevens	29			
D. Still	983			
P. M. Strubin	446	538	612	693
S. Suzuki	639	726		
R. M. Sweet	857			
A. Swift	770	154		
A. Sytin	877	930		
E. Takada	1068	1074		
R. Takahashi	894			
T. Takashima	899			
H. Takebe	625			
A. Taketani	584	625	672	201
K. Tamezane	639			
R. Tanaka	584	625	672	201
J. Tang	414			
T. Taniuchi	639	726		
S. Tararishkin	799			
E. Taurel	450			
Yu. S. Tchernousko	994	1012		

P. Teeter	922			
V. I. Terekhov	994			
M. E. Thuot	217			
P. L. Tkatchev	501			
M. Tobiyama	593			
M. Torikoshi	1074			
L. Tosi	949	376		
T. Toyama	963			
L. Trasatti	234			
W. Treutterer	937			
N. Trofimov	693			
E. Trojanov	930			
Yu. D. Tur	821			
M. Ulrich	763			
V. A. Ushakov	1030			
J. Utterback	628			
V. L. Uvarov	821			
M. Vadon	405			
A. Vaguine	464			
B. J. Vaidya	197			
A. Valentinov	910			
J. Van Zeijts	481			
M. Vanden Eynden	74	733		
L. Varga	703			
G. D. Vaughn	217			
H. Verhagen	485			
J. Vila	763			
M. F. Vineyard	633			
V. I. Vinogradov	970			
V. Voevodin	930			
G. Vogel	628			
A. J. Votaw	263			
J. M. Vouillot	514			
C. Vuerli	808			
T. Wada	584	625	672	201
A. M. Waller	293	739		
O. Walter	305			
S. B. Wampler	284	167		

B. Wang	751			
C. Wang	751			
J. Wang	572			
D. S. Warren	217			
W. Watson	360			
W. A. Watson	97	429	607	
R. E. Weiss	217			
C. Wermelskirchen	867			
R. T. Westervelt	746	223		
D. Wetherholt	607			
I. Weverling	1007			
K. White	1063			
R. R. Whitney	132			
B. Wilkie	91			
S. Witherspoon	435	481		
K. Woodbury	1035	1040		
F. Worm	335			
R. M. Wright	217			
D. Wu	97	607		
H. G. Wu	887	805		
J. Xu	874	751		
W. Xu	584	625	672	201
S. Yamada	1074			
N. Yamamoto	899	423		
Y. Yamamoto	1068			
A. Yamashita	584	625	672	201
T. Yamazaki	894			
Y. Yan	874			
K. Yanagida	639			
J. Yang	867			
H. Yokomizo	639	726		
S. Yoshida	423	593		
H. Yoshikawa	639	726		
Y. Ypinov	910			
Y. Yu	874			
V. Yurpalov	930			
V. I. Zaitsev	977			
Io. Zaroudnev	799	786		

D. Zasche	937	
T. Zehetbauer	937	
A. A. Zelezin	908	
T. Zhai	497	
M. Zhan	497	
J. Zhao	874	751
M. V. Zumbro	217	