INTERFACE HARDWARE FOR THE CESR CONTROL SYSTEM*

R.G. Helmke, D.H. Rice, and S.E. Ball**

INTRODUCTION

The heart of the CESR control system is a network of three PDP-11/34 computers directly controlling the linac, synchrotron, and storage ring respectively, with coordinated functions performed by a central DECsystem-1070. In these proceedings. All console controls and indicators and all machine components accessible to the control system are interfaced via distributed data buses, called X-Buses. There are a total of five X-Buses connected to the PDP-11/34 computers, but this number could be expanded if needed in the future.

Each X-Bus is driven by a micro- programmed controller in a byte serial mode on differentially driven twisted-pair cables. It can address 16 crates with 16 interface cards each, and allows computer access to any machine variable in less than 30us. Interface cards are provided for TTL, 24V conventional control, and analog signals. There are cards for conventional system-ready chains, for digital control of magnet power supplies, and specialized microprocessor-based interfaces.

Equipment is linked to the interface cards via easily understood connections which permit substitution of local controls and fault localization by personnel unfamiliar with the X-Bus system itself. Extensive error checking and diagnostic capability has been built into the system, and operation has been reliable even in high noise environments.

The CESR control system as a whole is described in a companion paper[1]

DESIGN GOALS

There were several goals which greatly influenced the design of the hardware interface system for CESR.

Economy The entire CESR project is based on a tradition of building accelerators with a minimal budget. The interface system was implemented with this in mind.

Noise Immunity The control system has to operate in a high pulse electrical noise environment, so noise immunity has to be built in, as well as error detection capability.

Speed For simplicity of the control software, we want to have each machine variable as immediately accessible to the computer as possible. The interface hardware should allow us to access single variables with minimal delay.

Maintainability There should be sufficient self diagnostic capability to localize a fault to a particular card or section of cable, hopefully without having to leave the control room. Connection between the equipment and the interface system should be straightforward enough for people unfamiliar with the interface system to easily service it.

To meet these goals, we had to decide whether to design an interface system from scratch or to use an existing commercially available system such as CAMAC. We finally decided to develop our own system for the following reasons:

COST We could build our own system for aproximately half the cost of a CAMAC system.

NOISE IMMUNITY CAMAC crates have inadequate RFI shielding for much of our environment whereas we could incorporate proper noise immunity from the beginning.

INTERCONNECTION CAMAC connectors are too small for practical cabling to equipment modules.

BOARD SIZE The small CAMAC boards make special purpose circuits difficult to design and the resulting power density reduces reliability.

X-BUS DRIVER

The X-Bus driver is the microprogrammed hardware interface between a PDP-11 and the CESR control system. To the control software, the driver appears as 16 sequential control registers in the PDP-11 I/O page. Addressing a particular register causes the desired operations to occur. If an attempt is made to read or write a register while a function is in progress, the interface does not respond until it has finished.

The initial X-Bus driver was hard-wired logic, consisting of TTL SSI and MSI. It fully occupied two eight by ten inch circuit boards, leaving little room for modification. The current design employs a high speed eight-bit bipolar microprocessor, and uses about 2/3 of an 8 by 15 inch board. Although the hard-wired version was somewhat faster, ease of modification was felt to be more important.

Even though the driver uses about 90 integrated circuits and component carriers, the implementation was relatively straight forward, the hardware design taking about a month. The initial micro-code was ready in about two months and debugging took another two.

The micro-processor used is the Signetics 8X300. It can execute a read/modify/write cycle in 250 nanoseconds. It is designed for digital controller functions and can rotate, add, mask and shift in a single instruction. The Unibus protocol is completely implemented in micro-code, with only AM2907 open-collector registers between the processor and the PDP-11 bus.

Part of the design project was the generation of a micro-assembler. A macro definition file is used in conjunction with the PDP-11 assembler to implement the necessary functions. Octal numbers representing the 8X300 machine code are generated in the listing file. This is processed by a program which controls a PROM burner interfaced to the X-Bus.

Because most of the functions of the driver are implemented in micro-code, a Unibus/X-Bus conversation is relatively slow. If a nearby crate is accessed, the result is ready in 13 microseconds. To prevent locking up the bus or causing non-existant memory traps, the software includes several no-ops between initiating a function and asking for the result.

X-BUS

The X-Bus cable has 19 twisted pairs. It was designed to work up to 500 meters with good noise immunity and can function to at least twice that. All pairs are driven and received differentially, using DS8832 and AM27LS31 drivers, and SN75115 or AM26LS32 receivers. The data/address lines and certain control lines are used in a 3-state party-line mode. Sixteen bits of data and sixteen of address are byte multiplexed on 8 bidirectional lines plus parity. To control address and data transfer, there are six control lines: XSYNC synchronizes each transfer and resets all circuitry between transfers; XSTRB synchronizes each address or data byte, using both rising and flalling edges; XRPLY is the echo of XSTRB and is used to synchronize returned data bytes; XODD distinguishes odd and even address and data bytes; XDIN requests data to be input to the computer; and XDOUT signals an output operation.

There are also three lines to provide for an interrupt protocol. Although the circuitry is in the crates, interrupts are not supported in the current X-Bus driver microcode or the control system software. The 19th line in the X-Bus is used to transmit the magnet power-supply synchronization clock in the X-Buses that extend around the ring. It is a spare line in the others.

The standard X-Bus protocol uses two address bytes followed by two data bytes, waiting for XRPLY for the address bytes before transmitting the data. Most interface cards reply immediately, but microprocessor interface cards may take a while to respond to a DMA request. The protocol could easily be extended to include multi-word block transfers or single byte transfers, but there has been no need for either of these operations yet.

INTERFACE CRATES

Each interface crate holds 16 interface cards plus a crate controller card. Each card has a dual 22 position card edge connector along the upper back edge which plugs into the crate bus, or C-Bus. External connection to te interface cards is via a dual 36 position card edge connector along the lower back edge of the card. The interface cards can easily be removed and replaced without having to turn off the crate power.

The crate controller serves as an interface between the X-Bus and the C-Bus. It isolates errors between the two buses, halting immediately when it detects an error and saving its status in an error latch. It is connected to the X-Bus in such a way that a crate can be powered up or down or the crate controller removed without disrupting the X-Bus operation.

The C-Bus signal lines are, for the most part, simple 3-state counterparts to the differential X-Bus signal lines. Addressing of interface cards is by position in the crate, so the high order address byte of the X-Bus protocol is suppressed in the C-Bus protocol. The C-Bus strobe signal (CSTRB) is a pulse instead of a level change, but otherwise the X-Bus and C-Bus protocols are the same.

INTERFACE CARDS

There are both standard and special function interface cards located in the crates on the X-Bus. Six general purpose cards provide input and output

capability for analog, TTL, or 24 volt control signals. Special function interface cards handle specific tasks where appropriate. Some of these interface equipment "ready chains," providing on/off, reset, and computer enable functions, as well as readout of equipment status bits. See table A. There are also two kinds of microprocessor based interface cards. One provides control knob scanning, another operates motor controlled devices such as phase shifters and variacs. The microprocessor cards are described in a companion paper in these proceedings.[2]

Signal type	in/out	circuits/card	interface connector	cost each
TTL	input	2 words (16 bits ea)	20 pin	$125
TTL	output	2 words (16 bits ea)	20 pin	$100
Analog	input	30 diff. ch. (12 bits resl)	BNC	$400
24 Volt	input	2 words (16 bits ea)	barrier	$200
24 Volt	output	32 circuits	barrier	$200
24 Volt ready chain	in/out	2 units	56 pin	$250

Table A
Sample Interface Cards

Because of the straight forward byte oriented structure of the bus within the crates (C-Bus), which is derived from the X-Bus, building a specialized interface card requires minimal design overhead. There is a standard "foundation logic" involving only 12 integrated circuits which provides the necessary interfacing to the C-Bus. Several one-of-a-kind cards have been designed. One is the master clock for control of the CESR magnets[3]; another interfaces the old synchrotron multiplexed monitoring and correction coil control system.

The connector hardware at each interface crate is designed to provide a clear-cut and understandable boundary between the X-Bus system and the "real world." All signals pass through conventional connectors (BNC'S, barrier terminal strips, and certain standard multiple pin connectors). Indexing and labeling follows long estpblished procedures which are familiar to our machine technicians. At this interface panel, signal levels are easily measured with a VOM. Equipment may be disconnected and run with a test box so that it is easy to determine if a problem is on the hardware or computer interface side of the panel. This feature has been very effective these last few months during the final phases of construction of CESR.

In the CESR tunnel, the X-Bus system is enclosed in an electromagnetically shielded environment. Plated steel crates connected by soldered copper conduit contain all the printed circuit cards and power supplies, and all signals to the outside use feedthrough bypass capacitrs with ferrite inductoors added outside the crate.

Since the density in the tunnel of any one type of signal is generally too low to justify use of the standard interface cards described above, the 24 volt and analog input signals are accomodated by a "distributed signal" interface system. Four printed circuit cards in separate controller crates contain level translators and other logic functions particular to each element being controlled or monitored. Multiplexors are wired to a card in the associated interface crate. This card has an analog-to-digital

converter and logic for data transfer to and from the C-Bus. Only the interface crates are connected directly to the X-Bus. Most of the electronics in the tunnel is associated with control of the CESR magnet power supplies and is described in a separate paper[3].

NOISE REJECTION

Various components of CESR are expected to be troublesome sources of electrical noise. The pulse forming networks in the linac and transfer line kicker magnets produce severe electrical spikes, while the bunched electron and positron beams are likely to produce a continuous stream of pulses with high frequency harmonics.

In order to minimize interference, the interface system in the CESR tunnel is electromagnetically shielded. Each signal goes through an RF filter. On the 3-state C-Bus, all signal receivers are schmitt triggers to increase the noise margin. On output operations, for which errors would be most harmful, the strobe signal and the "Data Output" signal have opposite polarity, minimizing the chance of latching erroneous data.

The X-Bus uses differential signals since most noise is induced equally on all lines. However, because the data/address and XRPLY lines are bidirectional, they have 3-state drivers at each crate which will only operate in a -1.0 to +6 volt range. To deal with this, we use "catch" diodes on each line so that if a large common mode voltage current spike occurs, every pair shares the resulting current. Because of the common mode inductance of the X-Bus cable (which could be enhanced by ferrite cores), the characteristically short noise pulses do not produce large currents, and do not couple strongly to the differential signals.

During the implementaiton of the X-Bus, it became apparent that the reply signal (XRPLY) was the most susceptible to noise. It is biased to a logical zero to minimize spurious reply signals while the drivers are in a high impedance state between operations. If the bias is too small, there is inadequate noise margin in the high impedance state, but if the bias is too great, the line drivers cannot fully overcome it, causing reduced noise margin in the asserted state. A compromise bias level had to be chosen.

ERROR DETECTION

Error detection is used primarily as a diagnostic aid rather than as a means of increasing reliability through error correction or retrys. We felt that if the noise level were so high that it defeated the inherent noise rejection features of the system, then it would likely be high enough to cause internal errors in the error detection logic. The simple parity scheme chosen, though not reliable in detecting burst errors, is adequate to monitor the error rate and determine when corrective maintenance is needed.

Other error detection features are intended to localize the failure as closely as possible. Each crate controller has built-in test patterns that can verify the functioning of the X-Bus up to that crate. There is also a diagnostic latch located in each crate controller so that data transfers to it follow the same paths as data going to or coming from a C-Bus interface card. Although this can't detect an open circuit in the C-Bus, it will detect a shorted C-Bus driver.

Special care was taken to minimize the chance of erroneous data being written into a machine variable. The crate controller will stop its action immediately upon detecting an error so that the machine variable latch will not be strobed. The X-Bus driver will then timeout waiting for the required reply edges. The operation can be retried if appropriate.

SUMMARY

In support of the CESR project, we have implemented a computer controlled interface system called the X-Bus. The development required the equivalent of three people working full time for a year. It has proven to be both economical and reliable. The X-Bus system has demonstrated its noise immunity for over a year in control of the linac, whose pulse forming networks are some of the worst noise sources in the laboratory.

----

[1] *The CESR Control System*, R.M. Littauer, S.E. Ball, R.G. Helmke, S.B. Peck, and D.H. Rice, Paper E-3, this conference.

[2] *Applications of Distributed Microprocessors in the CESR Control System*, D.H. Reaves, F.W. Dain, R.G. Helmke, paper D-45, this conference.

[3] *The CESR Magnet Power Supply System*, D.L. Hartill and D.H. Rice, paper J-22, this conference.