USE OF A GENERAL-PURPOSE TIME-SHARED COMPUTER IN ACCELERATOR CONTROL

M.Q. Barton, B.B. Culwick, J.A. Curtiss, J.J. Dabrowski
R.S. Frankel, W.E. Harrison, F.R. Martin, J.D. Smith,
R.J. Warkentien, I. Weitman

Brookhaven National Laboratory
Associated Universities, Inc.
Upton, L.I., N.Y.

Summary

A general purpose real-time data acquisition and control system implemented on a PDP-10 time shared computer and interfaced to devices via PDP-8 peripheral processors and a serial transmission scheme has been constructed. The design objectives of Fortran coded time-shared access to the accelerator have been achieved under the manufacturer's monitor with the addition only of shareable device handlers. The time-sharing monitor has allowed the development and operation of alternative control systems in parallel with continued use of an earlier system. Program development has been facilitated by the convenient access provided by the operating system and the multiple terminals available for programming and testing.

Objectives

Computer control and monitoring of the Brookhaven AGS dates from 1965. A system of control terminals interfaced to dual PDP-8's with disc storage was developed and served usefully in accelerator operation and physics studies of the machine. With the completion of the AGS conversion, it became apparent that exploitation of the machine could be greatly aided by suitable computerization of instrumentation and control of the accelerator. At this time it was also recognized that such computerization requires very significant efforts in manpower if effective results are to be achieved in a reasonable period of time. Two ways in which this burden could be alleviated were proposed 1) implementation of the control and monitoring programs to a great degree in Fortran, 2) implementation of the system in a time-sharing environment.

The first of these points is not seriously disputed at this time. No matter how efficient the individual, experienced analyst, the range of talent which can usefully be brought to bear on a problem and the speed of implementation are both greatly increased by use of the high level and universally spoken language. The second point is only slightly more subtle. The many and varied activities around the accelerator are obviously facilitated by independent access to computational power but the total integration of the system and interchange of information are simplified in a single machine. Both possibilities are provided by a medium scale general-purpose time-shared computer. This paper will discuss the implementation of an accelerator control system based on such a machine.

Configuration

The time-shared computer selected as the basis for the operating system is a Digital Equipment Corp. PDP-10. Among other reasons for the selection of this machine were principally the amply demonstrated capability of the time-shared operating system and the prior existence of two machines of this class in the Accelerator Department. A more difficult decision, given the objectives which included control of the accelerator complex from a single location, was the selection of direct device communication from the PDP-10 or interfacing to the hardware devices via a small computer. It was decided to interpose the small computer for the following reasons:

1. The PDP-10 manufacturer's time-shared operating system is designed around time responses of the order of terminal needs. To impose on this monitor the real-time needs of the AGS, while not impossible, would nullify many of the time sharing advantages. By interposing a small computer or computers operating synchronously with the AGS it would be possible to allow the PDP-10 to operate asynchronously under its normal scheduling algorithm.

2. The totality of the PDP-10 monitor is impressive in both bulk and complexity. By divorcing the real-time code procedures from the PDP-10 it was possible to develop these independently of the monitor. The alternative of "user mode" I/O development or actual incorporation of the necessarily complex service routine into monitor development versions would be both laborious and hazardous to the system. There would be no time-sharing for the system developers! Definition of a reasonably simple service routine for data communication with the small computer allowed development of the small machine code without "bombing" the PDP-10 monitor.

3. Demanding real-time requirements in one area of the AGS would preclude the provision of such responses to other areas if operation were from a single processor. Such conflicts are rather unpredictable in some areas of accelerator operation and at some periods in the AGS cycle. This leads to the desirability of increased processor power which is conveniently allowed by multiple small computers assigned to specific hardware groupings.

4. An extension of the arguments of 3) is the dedicated processor assigned to function generation or fast switching as a functional part of accelerator hardware as distinct from monitoring or supervisory control. The possibility of implementation of dedicated processors controlled by the PDP-10 is included by adopting similar approaches for all input-output.

5. As will be described later, hardware facilities for communication within the system are by means of relatively slow, serial pulse trains. Any device requiring high speed computer response would require an alternative parallel interface. For consistency with the above arguments this should not be directly on the PDP-10 but requires an intermediate processor which could then be located near to the data source or sink.

6. Some consideration has been given to hardware redundancy. Spare or redundant small processors can be provided but this is hardly an advantage for a system which includes them over one without them! Operational systems to continue during a PDP-10 failure are possible using the small processors but must be at a much reduced scale. Also, a significant effort is required to produce this system for what is likely to be a small reward. Thus although this offers some advantage in principle we believe this to be slight in practice.

The sum of the above arguments has persuaded us that the "peripheral processor" approach has sufficient merit to justify its selection over a directly interfaced design. The peripheral processor selected is the PDP-8E. This decision was based to some extent on historical precedent but has justification in the price, modern physical design and local expertise in coding, cross assemblers, etc.

The remainder of the selected configuration consists of a serial transmission scheme using a single co-axial cable connection to distributed locations for control of hardware devices. The principle is that electronics be located as close to the physical control point as allowed by the radiation environment but long traditions of multiple parallel cables have only been overcome to a modest degree.

The overall configuration is shown in Fig. 1.

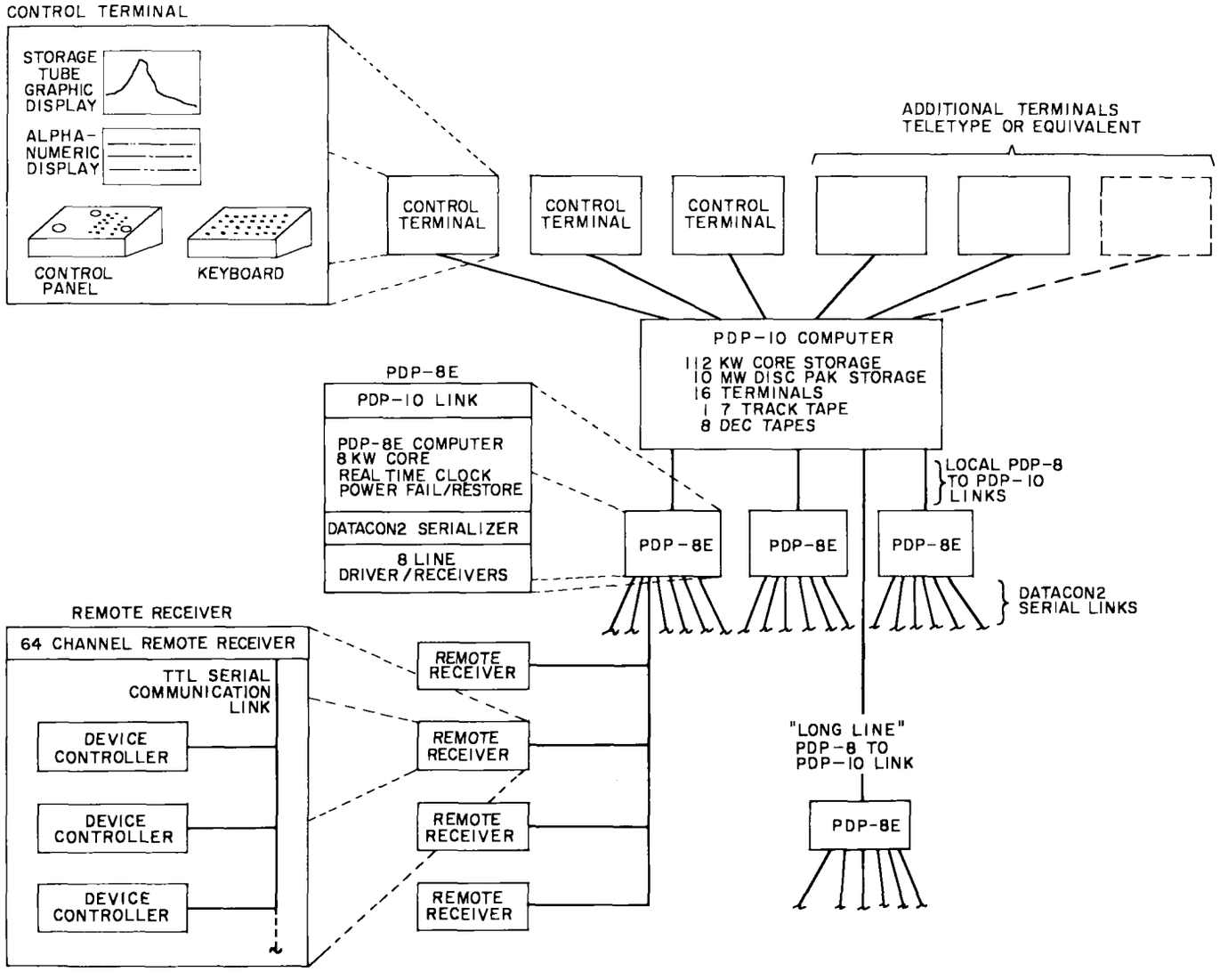

FIGURE 1 - OVERALL CONFIGURATION AGS COMPUTER CONTROL SYSTEM

PDP-8E Monitor

In order to implement a system with the desired properties in the configuration described above, it was necessary to impose severe restrictions upon the processing allowed to be performed within the PDP-8. To preserve the programmer's flexibility it was necessary that all logical decisions as well as all arithmetic calculation be made in the PDP-10. Thus the PDP-8's function only as sophisticated input/output devices rather than processors. Nevertheless, to make these devices useful in a dynamic environment required that input and output buffers and I/O sequences be completely dynamic. An attempt has been made to generalize the input/output operation in the only dimension by which real-time differs from batch or time-share ie. time. Input/Output operations are characterized by a device address, a list, a time and a priority. The time is specified in milliseconds after the start of an AGS pulse. The priority is one of four values representing the real time criticality of the operation and is used by the monitor to resolve conflicts in I/O requests. In particular the lowest priority causes the monitor to ignore the time and perform the operations as soon as higher priority requests permit.

One further complexity was introduced to improve the efficiency of repeated operations. A request to the PDP-8 is divided into two operations. The first step in an I/O operation is to define a COMMAND LIST which is transmitted to the PDP-8. This list is stored until overwritten or explicitly deleted. The commands within a list may then be performed by transmitting an EXECUTE LIST specifying a command list and a time and priority at which to execute it. More than one command list may be executed as defined in one execute list or the same command list may be repeated at two or more times. On completion of the operations specified in an execute list, the PDP-10 is called and the service routine within the PDP-10 schedules the calling job to run again. Multiple (currently 4) user access to the PDP-8's is provided to allow flexibility in the assigning of devices to PDP-8 channels.

Higher Level Organization

Presently a number of Fortran time sharing programs have been written which access machine data via the system described. The Fortran interface to machine data is quite convenient and allows programs to be written without concern for the real-time aspects of data acquisition. When the execute list has been transmitted to the PDP-8, the program can issue an input request and pass into an I/O wait state, or preferably suspend until awakened by the completion of the requested operations.

Operations within the PDP-8 are currently restricted to a single AGS pulse with longer time scale operations integrated in the PDP-10. Design is now proceeding on a system to provide file organized data and operations control for major areas of the machine from a single operations console. It is planned that an operator will maintain general control of machine operations with additional diagnostic activities and studies conducted from other terminals.

System Extension

The present system allows operations initiated by the PDP-10 and completed in the peripheral processor during one AGS pulse. It is envisaged that some repetitive functions in the machine will be allowed by repeating operations within the PDP-8 from pulse to pulse. Such tasks might include function generation, elementary monitoring functions, etc. They would be initiated by a program in the PDP-10 and would notify the program of problems beyond the elementary level. They could be coded to continue if the PDP-10 stopped. It is not planned to extend the PDP-8 capabilities appreciably beyong this level to avoid placing restrictions on the access and decision making functions. However, some extension of the list processing facilities will be required to interface the PDP-8 monitor to some non-standard devices in existing machine equipment. This will be done by defining additional list formats.

Hardware

The control system equipment which connects the PDP-8E's to controlled devices is called collectively DATACON2. The basic data transmission scheme is a transformer coupled phase encoded bipolar signal on a single co-axial line. Each transmission consists of a frame pulse followed by a key bit, 32 data bits and a parity check bit sent to devices. The data is self-clocked. An addressed device responds, after some delay allowed for internal processing, with a similar pattern but phase reversed to distinguish a reply from a message. Remote devices contain high impedance receivers of which 100 may be bridged on the line. Repeaters are required every 2000 feet and allow the number of receivers to be increased to the addressing limit of 256 stations per line. Signal details are given in Fig. 2.

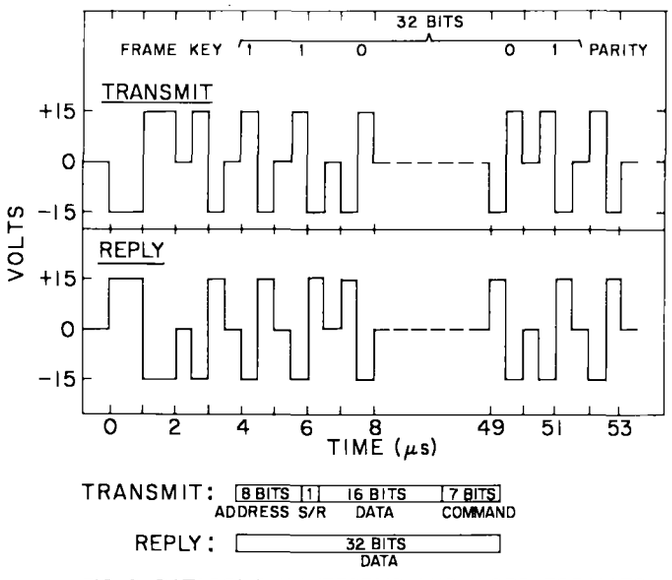

FIG. 2–DATACON2 WAVEFORMS AND WORD FORMATS

The single line technique minimizes hardware and cabling while giving a data rate of 600K bits/s which is a good match to an interrupt driven I/O service routine in the PDP-8E. The frame pulse serves as a "clear and initialize" to all receivers. The key bit which is always logical one indicates completion of a transmission when it reaches the end of a shift register. The parity bit is used for conventional data checking. The system provides for a suitable signalling rate with high noise immunity and excellent isolation.

The same transmission technique has been adapted to couple the PDP-10 to PDP-8 computers. Two bi-directional lines are used to provide full duplex transmissions with interrupt response control. TTL pulse trains are used for distances up to a few feet. Beyond this distance the phase encoded format is adopted using the drivers and receivers developed for the device control system.

The general purpose remote receiver is capable of communicating with up to 64 device controllers. As distinct from other systems using fast parallel data transmission at remote locations and serial transmission between locations, the DATACON2 system uses serial transmission throughout. The remote receiver functions to perform some system overhead of address decoding, parity checking, etc., to convert the phase encoded signal to TTL levels and to generate a clock signal. The data is then transmitted, still as a serial train, to the device control cards where it is clocked into shift registers. After a few microseconds for settling an ACCEPT pulse is generated. The addressed device, which lies at the intersection of two lines generated by two 3 bit to 8 line decoders in the remote receiver, processes data and returns a REPLY pulse. The "crate controller" then generates a REPLY CLOCK and strobes data from the device cards, converts it to phase encoded form and transmits it on the line. This approach, while not fully bussed within a remote crate, minimizes the intercard wiring and is almost free of cross talk problems. A non-trivial checkout

advantage of the serial scheme is that very few lines need be checked to identify a problem.

This system is used as a general purpose means of device control and monitoring. For some large power supplies where serious noise problems and isolation requirements were encountered, a dedicated receiver approach in which the only external connection is via the coupling transformer has been developed. Excellent isolation is achieved allowing control voltages and monitored signals to be referenced to the local ground.

About a dozen standard device control modules have been developed for the common applications of self-contained device control and subsystems. These include power supply controllers, predetermined timers, scalers, and stepping motor controllers. The standard devices also include so-called system components, A/D converter and multiplexers, which permit monitoring of analog signals either associated with controlled equipment or independently generated.

All monitoring within the system is performed on a polling basis. Extension to interrupt operation would be straightforward utilizing a second co-axial line but has not been found useful at this point.

Acknowledgements

Significant contributions to the hardward system have been made by D. Easler, V. Kovarik, R. Scheetz and G. Smith.

DISCUSSION

Is the Datacom transmission system a commercial system?

Barton: No, it was developed at Brookhaven. I believe the specifications are in a paper presented in a previous National Conference. I'm not sure.

James Halbig (LASL): Recently we have run into an interesting problem where we were sending out a command and asking for data in the same area, and since they were running into each other too fast, it bombed the command and we were having very weird things happening. I notice with only three PDP-8's, have you run into this problem yet or are your command ingestion processes fast enough so that you haven't run into these situations?

Barton (rephrasing the question): Do we have a problem with overloading of the Datacom rates and overlapping of commands? No, the system is free of that. It can't happen.

Halbig: Is it protected by the system, you say?

Barton: No transmission can go out until you've got the reply from the previous one. The minicomputer software is run on the interrupt system so that it can't try to send another one until it's got the previous one back. There's also a time-out so that if you fail to get a reply at all, it won't hang up; it will give you an error bit in the system and for that type of problem, it's worked out very well.