

CENTRALIZED DIGITAL CONTROL OF ACCELERATORS

R. E. MELEN*

*Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305***Introduction**

In contrasting the title of this paper with a second paper to be presented at this conference entitled "Distributed Digital Control of Accelerators,"¹ a potential reader might be led to believe that this paper will focus on systems whose computing intelligence is "centered" in one or more computers in a centralized location. Instead, this paper will describe the architectural evolution of SLAC's computer based accelerator control systems with respect to the "distribution" of their intelligence. However, the use of the word "centralized" in the title is appropriate because these systems are based on the use of centralized large and computationally powerful processors that are typically supported by networks of smaller distributed processors.

Linac and Beam Switchyard Computer Control System

Computers were first introduced into SLAC's accelerator control systems approximately 17 years ago when a SDS 925 was used in the beam switchyard area to provide monitoring and control for less than 50 power supplies and 1000 digital status bits.^{2,3,4} In those times, the high cost of even simple computers combined with relatively primitive operations requirements made the use of computers in control systems a hotly debated subject. Several factions of SLAC felt that the advantages (or disadvantages) of computers did not justify their expense. Hence, several years elapsed before a Digital Equipment Corp. PDP 9 was installed in the main linac control system.^{5,6}

The primary motive for using a computer in this system was for status monitoring. Another motivation was to provide automated reconfiguration of the linac after taking one of its 240 klystrons was taken off-line because of a failure and replaced with a spare unit. Ironically, this very complex problem remains today as one of the most demanding procedures that must be implemented in the new SLC control system.

Since the linac control system was originally designed as a manually operated system, the PDP 9 was essentially placed in series between the existing manual human interfaces and operators. The computer's intelligence was used to provide a more flexible man-machine interface and to extend the scope of the operators' observation abilities by providing extensive automated monitoring functions.

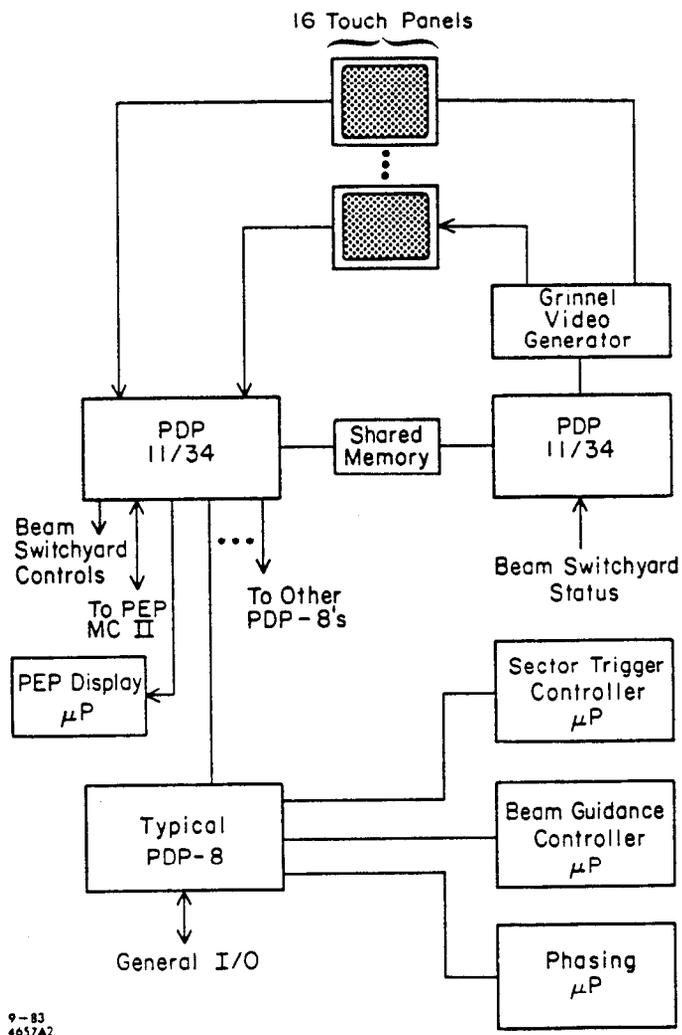
Before the first implementation of the PDP 9 computer system was complete, it became evident that a two control room approach was expensive and irrational and the switchyard and linac control rooms were merged into the switchyard control room, now known as Main Control Center (MCC).^{7,8} Further, it soon became evident that the flow of information to and from the linac through the original manual interfaces was intolerably slow. Therefore, eight PDP 8s were placed along the 2-mile klystron gallery and used as intelligent data acquisition and distribution processors.^{9,10}

The system was later expanded by the use of fixed program micro-processor controllers interfaced to the PDP 8s via serial links to provide specialized dedicated controllers to implement beam guidance,¹¹ phasing,¹² and triggering¹³ functions.

With the exception of updating the system by replacing the PDP 9 and SDS 925 computers with PDP 11 computers, the basic concepts and architecture of the system remain relatively unchanged today.¹⁴ It is serving the intended purpose as a very efficient "look-and-adjust" system.^{15,16,17}

*Work supported by the Department of Energy, contract number DE-AC03-76SF00515.

Fig. 1. The linac/switchyard computer control system.



9-83
4657A2

SPEAR

Two considerations dominated the choice of computer architecture for the SPEAR computer control system¹⁸ (Figure 2), which was implemented in 1970-72. First, a very tight budget and short construction cycle dictated a minimum cost and manpower effort. Secondly, there was a strong desire to provide a system that would provide extensive support of machine physics efforts via the use of real-time machine modeling. This concept would allow the machine operators to specify desired operating conditions such as tunes, beta values, dispersion, and energy and have the computer automatically calculated and set up the appropriate magnet and RF settings.

The second requirement exceeded the capabilities of the then current mini-computers. So it was decided that all of the objectives could be met by combining the control system computational needs with the needs of the two experimental areas, and a XDS Sigma-5 "timesharing" system was purchased.

Although the original computer configuration was woefully inadequate in terms of memory and disk space, it was expanded and both the control system and one experiment coexisted peacefully for several years. This sharing worked successfully primarily because the storage ring required very little "tuning" when the experimenters were taking data and vice versa.

The use of a large and computationally powerful processor in the centralized SPEAR control system had many advantages. The sophisticated CP-5 operating system, when combined with an advanced FORTRAN compiler, provided an excellent software development environment. Secondly, the resources available through the use of the Sigma-5 eliminated the need for a distributed system to acquire and process data from the physically small, 720 foot circumference

facility, which greatly simplified the software task. These advantages allowed the applications software to be developed quickly with a small staff.

Probably the most significant weakness of this architecture was the fact that the CP-5 operating system had a relatively slow time-shared response-time. However, this weakness never really impacted the system performance since a storage ring has little need for faster than human response times (1-2 seconds). Further, the few needs for a fast response time were accomplished by providing carefully written "real-time" foreground tasks.

Through the succeeding years after its initial implementation, there has been very little need or interest in expanding the architecture of the system. New additions or changes have been easily implemented within the existing structure. However, maintenance support concerns for both the outdated hardware and the long neglected control programs finally forced a decision to replace the Sigma-5 and its in-house designed interfaces with a dedicated VAX 11/750 and CAMAC based data acquisition hardware. The new software system and human interfaces are based on the PEP control system while the CAMAC hardware uses elements from PEP, SLC, and SLAC experimental systems. This new system will go on-line within the next ten days.

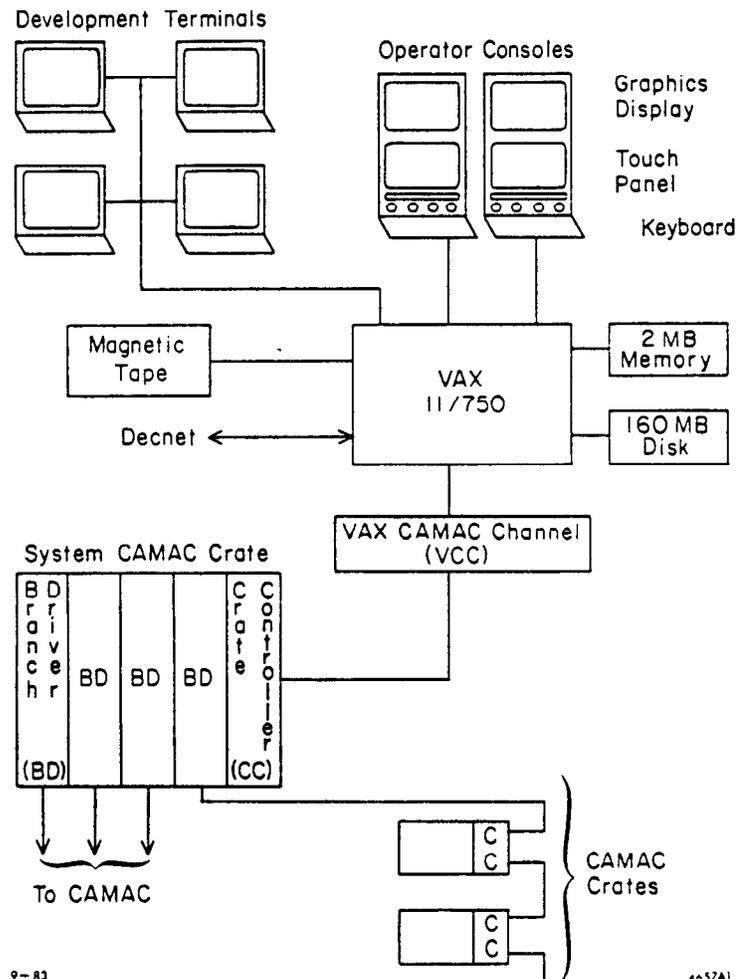


Fig. 2. The SPEAR computer control system.

PEP

The large physical size together with the large number of I/O points associated with PEP necessitated the use of a distributed network of computers to serve as intelligent data acquisition and distribution processors for a central computing complex.^{19,20} As shown in Figure 3, the network contains ten ModComp computers and one VAX 11/780. The central MCIV computer is attached to a single operator console, and is connected via 500 kilobaud serial links to nine MCII remote computers which are in turn interfaced to approximately 50 CAMAC crates via one megabaud serial SDLC links.^{21,22,23} The MVIV central control computer is interfaced to the VAX via a similar link.

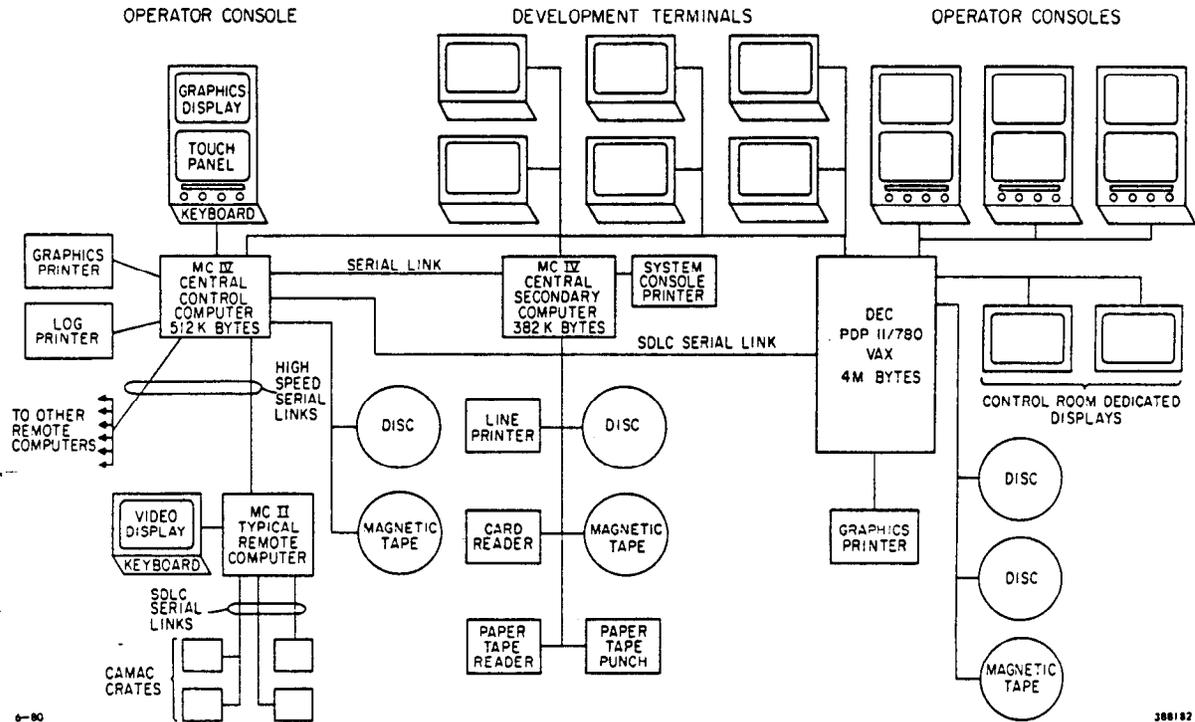


Fig. 3. The PEP computer control system.

In normal operation, the remote MCII computers continuously collect data from their respective CAMAC crates and then forward the refreshed data to the central MCIV computer at a rate of approximately seven refreshes/second. The central MCIV computer receives refreshed data from all of the remote processors and maintains a copy of the latest data for each signal in its RAM. This refreshed data is then sent as a block to the VAX at a rate of three blocks/second. Thus, continuously refreshed data can be accessed by application programs either in the MCIV or VAX. The only input data not read by this process is data collected by a slow digital volt-meter attached to each MCII. This data is delivered upon the request of a program in either processor. Output CAMAC data commands may originate in either processor. They are sent to the CAMAC crates via the MCIV and the relevant MCII.

In order to keep the data flowing at an acceptable rate, only the raw data, consisting of approximately 1200 points per MCII, is refreshed. Conversion to engineering units and other signal specific processing is performed only at the application task level on an "as needed" basis. Similarly, the output routines only process raw data.

The only "applications dependent" processing that occurs in the MCII's relate to the position monitor system which requires a complex readout procedure requiring several time delays. All other significant applications dependent programs, including limit checking and alarms, reside in either the central MCIV or the VAX.

The VAX executes nearly all of the higher level applications and modeling programs.^{24,25,26} However, the MCIV does execute some display programs requiring fast refresh rates as well as a minimal set of programs for control and monitoring functions that can be employed in the event of a VAX hardware failure.

The addition of the VAX to the system architecture was a deviation from the original system design. This addition was necessitated by a general concern that although the hardware was adequate, weaknesses in the MCIV software prevented it from serving effectively as a stand-alone central processor in the system. The software weaknesses existed in three areas.

First, the general quality of the operating system in terms of software development support left a lot to be desired. The general level and quality of support features were substantially below those provided by the Sigma-5 system at SPEAR and eventually by the VAX. This weakness had a significant impact upon the time required to develop and maintain applications software.

Secondly, the operating system lacked "robustness" in terms of its ability to continue running without crashing when it was asked to execute more programs than it could store in its RAM. This weakness was a significant problem because many of the modeling tasks required large blocks of memory.

Thirdly, the operating system was not "bullet-proof" in that relatively insignificant applications programming errors could cause a system crash or could degrade the system response to uselessness.

In contrast The VAX VMS operating system provides a stable, user-friendly, relatively bullet-proof system on which applications programs may be developed and run. The symbolic debugger is an especially valuable tool for debugging and maintaining real-time software. The relatively slow response time of the VAX is overcome by the use of the MCIV which serves as a dedicated front-end processor and generally handles procedures requiring faster than human response times. Further, acceptable system response times are achieved because PEP is a single purpose accelerator, and hence typically must support only three-four operating stations controlled by one or two operators.

SLC

In its final state, the SLC computer control system²⁷ will be an order of magnitude larger and more complex than any of SLAC's other accelerator control systems. In addition to modernizing and streamlining the operation of the present linac/beam switchyard system, the SLC system must provide a system based on machine modeling^{28,34} to support the extensive accelerator development efforts required to develop an accelerating system meeting the tight SLC beam requirements.

In its final configuration, the SLC computer system will provide a combination of two VAX 11/780 central processors networked to 70-100 powerful micro-processor clusters, as shown in the block diagram in Figure 4. The micro-processor clusters interface with the equipment to be monitored and controlled through the use of CAMAC. These clusters will be located in each of the 30 linac sector alcoves and near the damping ring, electron and positron sources, and the SLC arcs and final focus.

The dual-VAX complex will serve to provide a centralized human interface for the machine operators and will be used to provide the on-line execution of the large modeling programs. In addition, these computers will serve to provide an environment for fast, efficient program development and maintenance for both the VAX and micro-processor clusters.

The distributed micro-processor clusters are based on the Intel Multibus architecture.²⁹ This architecture provides support for an arbitrary number of single-board computers (SBC) which communicate with each other through the use of shared memory and interrupts. The micro-processor clusters contain an Intel 86/30 SBC, 768 kilobytes of RAM and 8 kilobytes of EPROM. Various benchmark tests have shown that each micro-processor cluster has somewhere between 1/10 and 1/7 the processing power of the VAX 11/780. The micro-processor clusters are interfaced to CAMAC through a high-speed direct memory access (DMA) device based on the use of a bit-sliced micro-processor.

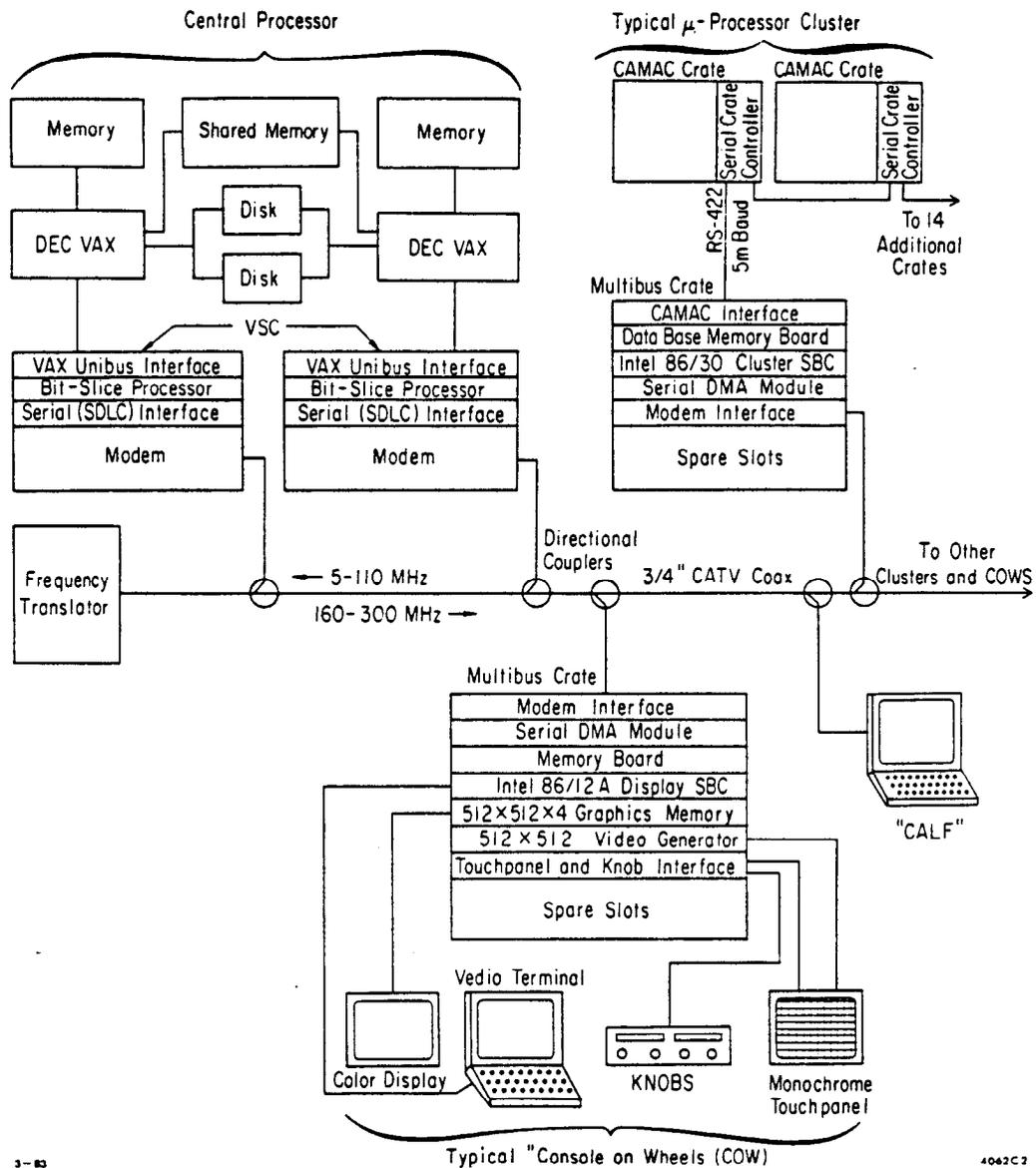


Fig. 4. The SLC computer control system.

Intelligence for the system is also distributed into the CAMAC crates via the use of dedicated controllers. One of these devices, the Smart Analog Monitor (SAM), is a Zilog Z80 based CAMAC module that continuously scans 32 analog channels and provides their floating point voltage values in either VAX or 8086 formats.

A second device, the Parallel I/O Processor (PIOP) CAMAC module, is a general purpose processor based on the Intel 8088 micro-processor chip and presents a standardized interface to the CAMAC data highway. This module provides a front panel port which is a differential transmitter/receiver version of the micro-processor's bus structure. This port provides a simple and straight forward method for interfacing special purpose "heads" that interface to specific devices or processes. So far this module has found use in the monitoring of the 270 linac klystrons' phase and amplitude³⁰ and also for their general monitoring and control.³¹

Programs for the PIOP may be developed by use of a cross-compiler or cross-assembler on the VAX. The compiled or assembled code may then be downloaded to the PIOP or it can be "burned" into EPROM's to provide a non-volatile program source.

There are two types of operator consoles in the system. The primary type has been given the name of console-on-wheels (COW) because it is a fully portable unit which may be connected at any point of the system's communications backbone. The second type is called a CALF and consists of an Ann Arbor Ambassador terminal with a modem which also allows it to be plugged into the communications backbone. Both the COW and the CALF communicate directly with the VAX. Software in the VAX allows the CALF to emulate a subset of the COW functions. The number of COW's and CALF's that can be supported by the system is limited only by the system's processing power.

The communications backbone for the system consists of a broadband (5-300 MHz) Cable Television (CATV) system that has the capability to support several hundreds of frequency-modulated signals on a single cable.³²

Several sub-systems currently use the cable for communication. A high-speed, one Megabaud, polled network has been developed at SLAC to interconnect the micro-processors with the VAX's. A bit-sliced micro-processor is used to direct the sequential polling on the system and serve as a DMA channel to the VAX. This unit provides a maximum poll rate of 1000 polls/second. The use of this network structure is a departure from our earlier plans. We originally contracted with a commercial firm to provide a Carrier-Sense Multiple-Access Collision-Detect (CSMA/CD) network similar to that specified by Ethernet.³³ However, the development effort for this system turned out not to match our schedule needs, so an in-house solution was developed.

The CATV cable also supports terminal/VAX communications with equipment using protocols similar to Ethernet. The cable has a capacity for several hundred terminals. The same cable is also used to support television channels, voice channels, and point-to-point two megabaud data channels.

As previously mentioned, essentially all of the SLC software development is performed through the use of the VAX. Wherever possible, FORTRAN 77 is used for applications programming in both the VAX and the micro-processor clusters. FORTRAN 77 was chosen as the standard language because of its extensive support in the VAX, and because it is the most universally understood language. Although alternative languages could be used for the micro-processors, the consistency provided by standardizing on FORTRAN 77 has been a great advantage for both the development and support of applications programs.

A significant effort has been expended by SLAC to create an efficient and user-friendly environment for the development of micro-processor software. In collaboration with the Intel Corp., FORTRAN 77 and PLM 86 cross-compilers, a cross-assembler, and a cross-linker have been developed to support the 8086/8088 series of micro-processors. Further, a symbolic debugger has been developed to allow the remote debugging of micro-processor based programs.

Applications tasks executed in the VAX are written as structured subroutines which are attached to a VAX process that provides interface routines to the operator console, and to a structured database. This process also provides a scheduling service for the subroutines.

The micro-processor clusters provide local control algorithms for the operation of the technical equipment. In general, the micro-processors receive an operational configuration in engineering units for their equipment from the VAX. The micro-processors then insure that the equipment is set to that configuration and will only report back to the VAX when it is unable to achieve or maintain the desired configuration. The micro-processor clusters also provide monitoring information in engineering units to the VAX upon request. They also support a "pass-thru" mode for I/O commands from the VAX. The commands may originate from a VAX applications process or from a system user via individual, or a file of, interpretive commands. Micro-processor systems will be used in the future to implement time-sensitive digital control loops wherever required.

The first phase of the SLC control system implementation was brought on-line beginning August 1. This phase provides a single VAX 11/780 computer and micro-processor clusters in the first 10 sectors of the linac, the injector, and the Damping Ring. Several of the sub-systems are now operational and we are currently working hard to bring additional subsystems on-line.

It is much too early in the shakedown process to make any generalizations regarding the eventual performance of the system. However, two observations seem to be in order. First, the importance of on-line debugging and diagnostic aids cannot be overemphasized. In a complex system, it is extremely important to be able to trace problems efficiently in the system's operating environment. Although an extensive effort has been applied to providing these tools, we will continue to direct significant resources in this area. Secondly, the number of simultaneous users of the system has been overwhelming at times. It is not unusual to have 5 COW's, 5-7 CALF's, and 5-7 program development terminals simultaneously active. This concern has been partly alleviated by shifting some of the program development efforts to a second VAX. Though current response-times for the system may be tolerable even under heavily loaded conditions, with our current configuration it is evident that we will have to do battle with a response-time problem as the system expands to its fully-implemented state.

Summary

Upon careful examination of the architecture of SLAC's computer control systems described above, it becomes evident that the distribution of the systems' intelligence generally falls into three tree-like layers.

The first layer typically consists of a central computer complex incorporating one or more relatively large and powerful processors. The more modern systems use state-of-the-art 32-bit processors with several megabytes of RAM and several hundreds of megabytes of disk memory. Further, they support extensive user-friendly operating systems and program development facilities.

The second layer typically consists of several smaller processors which are downloaded from the central complex and whose primary task is to provide data acquisition and distribution. The more modern systems are 16-bit processors with several hundred kilobytes of RAM and no disk memory.

The third layer typically consists of several tens or hundreds of micro-processors, each dedicated to a single device. The micro-processors for these "dedicated intelligent controllers" are small and inexpensive and typically require less than 32 kilobytes of RAM or EPROM memory. Their hardware may be general purpose in nature or may be built into the architecture of the device itself. Figure 5 illustrates several of the relevant features of each of these layers.

This paper serves to illustrate that "for better or for worse," SLAC is committed to the centralized digital control of its accelerators.

Acknowledgements

The list of key people contributing to SLAC's control systems over the last 15-20 years is too extensive to single out individuals. Instead we have tried to reference publications that reflect the many hundreds of man-years of effort that has been applied to the design, implementation and support of these systems.

Central Computer Complex	Remote Data Acquisitions and Distribution Processors	Dedicated Intelligent Controllers
<ul style="list-style-type: none"> • Provides centralized human interface for operating personnel. 	<ul style="list-style-type: none"> • Collects and distributes I/O data for the central computer complex. 	<ul style="list-style-type: none"> • Dedicated to a fixed well-defined task for a single device.
<ul style="list-style-type: none"> • Is sufficiently powerful and contains sufficient resources to support the real-time execution of computationally demanding and physically large machine modeling programs. 	<ul style="list-style-type: none"> • Performs tasks requiring greater I/O rates or faster response times than can be provided by the central computer complex. 	<ul style="list-style-type: none"> • Provides high I/O rates and fast response times.
<ul style="list-style-type: none"> • Maintains both volatile real-time and non-volatile disk-base centralized data bases. 	<ul style="list-style-type: none"> • Monitors device stations and reports important changes to the central complex. 	<ul style="list-style-type: none"> • Provides a simple software interface to the device.
<ul style="list-style-type: none"> • Serves to synchronize the over-all system operation. 	<ul style="list-style-type: none"> • Initializes front-end devices and controllers. 	<ul style="list-style-type: none"> • May be downloaded from remote processor or may use EPROM memory.
<ul style="list-style-type: none"> • Provides flexible and efficient operating systems and program development facilities. 	<ul style="list-style-type: none"> • Maintains a local general-use database. 	<ul style="list-style-type: none"> • Does not use an operating system.
<ul style="list-style-type: none"> • Provides cross-compilers, cross-assemblers and remote debugging facilities for development of remote processor programs. 	<ul style="list-style-type: none"> • Supports multi-tasking real-time operating system. 	<ul style="list-style-type: none"> • Does not support extensive on-line debugging capabilities.
<ul style="list-style-type: none"> • Maintains current program images for all processors in the system. 	<ul style="list-style-type: none"> • Supports remote debugging features to allow efficient on-line checkout through the central computer complex. 	
<ul style="list-style-type: none"> • Provides diagnostic support for all processors in the system. 	<ul style="list-style-type: none"> • Does not support mass storage. 	

Fig. 5. Comparison of typical features for the three layers of intelligence.

References

1. M. Crowley-Milling, "Distributed Digital Control of Accelerators", paper to be presented at lecture session LS-B1 of this conference.
2. S. K. Howry, R. Scholl, E. J. Seppi, M. Hu, D. Neet, "The SLAC beam switchyard control computer," *IEEE Trans. on Nuclear Science*, NS-14,3, 1066 (1967).
3. S. K. Howry, SLAC Report CGTM 10, "BSY Control Computer System Language," (1966).
4. S. K. Howry, SLAC PUB-248, "A Concise On-Line Control System," (1966).
5. K. B. Mallory, "The Control System for the Stanford Linear Accelerator Center," *IEEE Trans. on Nucl. Sci.*, 1022-1029, (1967).
6. K. B. Mallory, "Some Effects of (Not Having) Computer Control for the Stanford Linear Accelerator Center," *IEEE Trans. Nucl. Sci.*, NS-20 (1973).
7. K. Breymeyer, et al., "SLAC Control Room Consolidation Using Linked Computers," *IEEE Trans. Nucl. Sci.*, NS-18 (1971).
8. S. Howry, R. Johnson, J. Piccioni, and V. Waithman, "SLAC Control Room Consolidation-Software Aspects," *IEEE Trans. Nucl. Sci.*, NS-18, 403-303 (1971)
9. K. B. Mallory, "Initial Experience with a Multi-Processor Control System," *Proceedings 9th International Accelerator Conference on High Energy Accelerators*, (1975).
10. K. B. Mallory, "Control Through a System of Small Computers," *IEEE Trans. Nucl. Sci.*, NS-22, 1086-1087 (1975).
11. W. C. Struven, K. B. Mallory, "Two Micro-computer Controller Applications at SLAC," *IEEE Trans. Nucl. Sci.*, NS-24, (1977).
12. S. K. Howry, A. Wilmunder, "A Micro-processor Controller for Phasing the Accelerator," *IEEE Trans. Nucl. Sci.*, NS-24, 1804-1806 (1977).
13. S. K. Howry, "Trigger Pattern Generation by Computer," SLAC TN-75-5 (1975).
14. V. Davidson and R. Johnson, "Present SLAC Accelerator Computer Control System Features," *IEEE Trans. Nucl. Sci.*, NS-28 (1981).
15. D. Fryberger and R. Johnson, "An Innovation in Control Panels for Large Computer Control Systems," *IEEE Trans. Nucl. Sci.*, NS-18, (1971).
16. K. Crook, "CRT Touch Panels Provide Maximum Flexibility in Computer Interaction," *Control Engineering Magazine*, 33-34, July 1976.
17. K. Crook and R. Johnson, "A Touch Panel System for Control Applications," SLAC PUB-1861 (1976).
18. A. M. Boyarski, A. S. King, M. J. Lee, J. R. Rees, and N. Spencer, "Automatic Control Program for SPEAR," *IEEE Trans. Nucl. Sci.*, NS-20, 580-583, (1973).
19. A. Chao et al., "PEP Computer Control System," *IEEE Trans. Nucl. Sci.*, NS-26, 3268-3271 (1979).
20. R. Melen, "The PEP Instrumentation and Control System," *Proceedings of the 11th International Conference on High-Energy Accelerators*, pp. 408-420 (1980).
21. J. D. Fox, E. Linstadt, and R. Melen, "Applications of Local Area Networks to Accelerator Control Systems at the Stanford Linear Accelerator Center," *IEEE Trans. Nucl. Sci.*, NS-30 (1983).
22. J. R. Kersey, "Synchronous Data Link Control", *Data Communications*, McGraw-Hill Publications, 49-60 (May/June 1974).
23. A. Altmann, R. Belshe, R. Dwinell, J. Fox and N. Spencer, "CAMAC Micro-processor Crate Controller, Revision J," PEP internal document.

24. M. H. R. Donald, P. L. Morton and H. Wiedemann, "Chromaticity Correction in Large Storage Rings," IEEE Trans. Nucl. Sci., NS-24, 1200-1202 (1977).
25. E. Close, M. Cornacchia, A. S. King and M. J. Lee, "A Proposed Orbit and Vertical Correction System for PEP," IEEE Trans. Nucl. Sci., NS-26, 3502-3504 (1979).
26. M. Donald et al., "Some Schemes for On-Line Correction of the Closed Orbit, Dispersion and Beta Function Errors in PEP," IEEE Trans. Nucl. Sci., NS-28, (1981).
27. R. Melen, "A New Generation Control System at SLAC," IEEE Trans. Nucl. Sci., NS-28 (1981).
28. M. Lee et al., "Mathematical Models for the Control Program of SLAC Linear Collider," IEEE Trans. Nucl. Sci., NS-28, (1981).
29. Intel Multibus Specification Manual 98006832-02, Intel Corporation (1979).
30. J. D. Fox and H. D. Schwarz, "Phase and Amplitude Detector System for the Stanford Linear Accelerator Center," IEEE Trans. Nucl. Sci., NS-30, (1983).
31. R. Keith Jobe, "A New Control System for the SLAC Accelerator Klystrons for SLC," IEEE Trans. Nucl. Sci., NS-30 (1983).
32. W. Struven, "Wide-Band Cable System at SLAC," IEEE Trans. Nucl. Sci., NS-30 (1983).
33. The Ethernet, Digital Equipment Corporation, Intel Corporation, and Xerox Corporation (1980).
34. M. J. Lee et al., "Models and Simulation," paper to be presented at lecture session LS-C2, this conference.