SLAC – PUB – 3377
July 1984
(E/I)

A REVIEW OF TRIGGER AND ON–LINE PROCESSORS AT SLAC*

A. J. LANKFORD

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

1. INTRODUCTION

The role of trigger and on–line processors in reducing data rates to manageable proportions in e^+e^- physics experiments is defined not by high physics or background rates, but by the large event sizes of the general-purpose detectors employed. The rate of e^+e^- annihilation is low, and backgrounds are not high; yet the number of physics processes which can be studied is vast and varied.

This paper begins in Section 2 by briefly describing the role of trigger processors in the e^+e^- context. The usual flow of the trigger decision process is illustrated with selected examples of SLAC trigger processing. The examples discussed are the energy trigger of the ASP detector and the charged particle triggers of the Mark III and Mark II detectors. Section 3 sketches the features of triggering at the SLC and the trigger processing plans of the two SLC detectors: The Mark II and the SLD. In Section 4, the most common on-line processors at SLAC, the BADC, the SLAC Scanner Processor, the SLAC FASTBUS Controller, and the VAX CAMAC Channel, are discussed. Uses of the 168/E, 3081/E, and FASTBUS VAX processors are mentioned. The manner in which these processors are interfaced and the function they serve on line are described. Finally, Section 5 outlines the accelerator control system for the SLC. This paper is a survey in nature, and hence, relies heavily upon references to previous publications for detailed description of work mentioned here. In addition, apologies are deserved by all the experimenters whose work is overlooked or improperly acknowledged by this overview.

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

Invited Talk Presented at the Symposium on Recent Developments in
Computing, Processor, and Software Research for High Energy Physics,
Guanajuato, Mexico, May 8 - 11, 1984.

2. TRIGGER PROCESSING FOR e^+e^- ANNIHILATIONS

e^+e^- collisions are provided by bunched beams of electrons and positrons which cross with frequency set by the physical scale of the accelerator. Trigger decisions can be made without deadtime during the period between crossings. This period is shown for the accelerators SPEAR, PEP, and SLC in Table I. The time between crossings at SPEAR and at PEP is sufficiently short that trigger decisions are normally made by multi-level hardware processors. The first-level trigger, made between crossings, reduces the beam-crossing frequency (1.3 MHz at SPEAR and 420 KHz at PEP) to a rate set by its allowed deadtime fraction. First-level trigger rates are typically about a kilo-Hertz. The second-level trigger rate is set by the readout time until an event is buffered. Rates are typically 2 - 5 Hz and deadtimes less than 10% . Some experiments also employ third-level triggers for long decision times on small numbers of events. The MAC experiment at PEP, for instance, reduces its trigger rate by about a factor of two with a software trigger decision requiring 10 to 20 msec on its VAX.

Table I. Beam crossing and interaction rates at SLAC e^+e^- accelerators

Accelerator	Period between beam crossings	Rate ($\sigma_{\rm pt}$)	Rate ($\sigma_{\rm tot}$)
SPEAR	780 $nsec$	$\sim 2\times 10^{-2}\ Hz$	$\lesssim 1\ Hz$
PEP	2.3 μsec	$\sim 2\times 10^{-3}\ Hz$	$\sim 0.01\ Hz$
SLC	5.5 $msec$	$\sim 7\times 10^{-5}\ Hz$	$\lesssim 0.25\ Hz$

The rate of e^+e^- annihilations is parameterized as:

$$\text{rate} = R_{tot}\times\sigma_{pt}\times\mathcal{L}\ , \qquad \text{where} \quad \sigma_{pt} = \frac{87nb}{(2\ E_{\rm beam})^2} \tag{1}$$

is the point-like cross-section for $e^+e^- \rightarrow \mu^+\mu^-$. R_{tot} is the ratio of the total annihilation cross-section to σ_{pt}. R_{tot} is in the range 4 to 8 in continuum regions, and is itself an indication of the onset of new physics processes. On resonances, R_{tot} can be much greater. R_{tot} is $\sim$ 2500 and $\sim$ 25 on the $\psi(3100)$ and $\Upsilon(9460)$ respectively, and is expected to be about 4000 on the Z° resonance. $\mathcal{L}$ is the luminosity, and is characteristically within about a factor of two of $10^{31}\ cm^{-2}\ sec^{-1}$. Typical physics rates are shown in Table I for SPEAR, PEP, and SLC. The total physics rates are quite manageable; consequently, most detectors for e^+e^- physics are general purpose in nature and try to record all physics events. The detectors generally consist of cylindrical drift chambers in solenoidal magnet fields surrounded by calorimetry. Typical event sizes range from a couple of kBytes to about a hundred kBytes. Similar detector geometries result in similar trigger problems and generic solutions.

The general purpose nature of e^+e^- experiments require that triggers include a broad range of physics topologies, from multiparticle hadronic events to two-track leptonic events, single electron events, two-photon or one-photon final states, and decays of possible long-lived particles. These events generally fall into two broad categories: events with two or more charged tracks and events with one or more energetic showers. Trigger designs generally consist of parallel logic to identify these event types. Charged particle triggers define a track as a set of tracking chamber hits in a trigger road. The numbers of planes of tracking available to and required by the trigger processor vary among experiments, as do the precision with which roads are defined and the momentum range covered. Neutral energy triggers discriminate on local or global sums of energy from calorimeter channels. The thresholds accessible and chosen depend on the type of calorimeter and the way in which the sums are formed. Information from charged particle and neutral energy triggers may be combined in other triggers. Additional parallelism at all decision levels provides redundancy helpful for determining trigger efficiency.

The backgrounds to the physics triggers arise from cosmic rays, beam-gas collisions, beam-pipe collisions of off-energy electrons, synchrotron radiation, and electronic pickup. The sources of background can generally be reduced to quite manageable magnitudes by careful masking and shielding. Residual background is reduced by limiting acceptance in the radial and longitudinal position of and in the time of the interaction. Ability to reject random hits can also be important. Most trigger processors require that the projection of candidate tracks in the plane perpendicular to the beam axis ($r-\phi$ plane) pass through a fiducial area surrounding the beam-crossing point ($r = 0, z = 0$). In addition, the track must be in time with the beam-crossing. Few experiments are able to project track candidates longitudinally along the beam-direction (z); so experiments commonly have inner trigger chambers or vertex detectors which restrict the acceptance in this dimension. The TPC detector[1] at PEP, however, uses its drift time measurement along the beam direction to project tracks toward the beam-crossing point in $r - z$ planes instead of the $r - \phi$ plane.

The following sections describe examples of SLAC trigger processors which illustrate solutions to the problems of trigger efficiency and background rejection. The examples discussed are the energy trigger of the ASP detector and the charged particle triggers of the Mark III and Mark II detectors.

2.1 The ASP Energy Trigger

The trigger logic of the ASP Detector[2] provides an example of an energy trigger accomplished between PEP beam crossings so as to have no deadtime. This new detector, designed for the detection of anomalous single photons, consists of four walls of lead-glass shower counters, called quadrants, which surround the interaction point. Each quadrant consists of five layers of glass separated by proportional chambers. The ASP trigger, with several thresholds and programmable logic, provides simple and general logic for triggering on both localized and overall energy deposit.

The flow of the trigger decision is shown schematically in Fig. 1. The 632 photomultiplier signals from the lead glass are each split to a digitizing system and to the trigger. The trigger signals are summed to eighty sums of eight and then summed again to twenty sums, each corresponding to a layer. These twenty layer sums (five per quadrant) each go to integration circuits and then are discriminated to define hit layers.

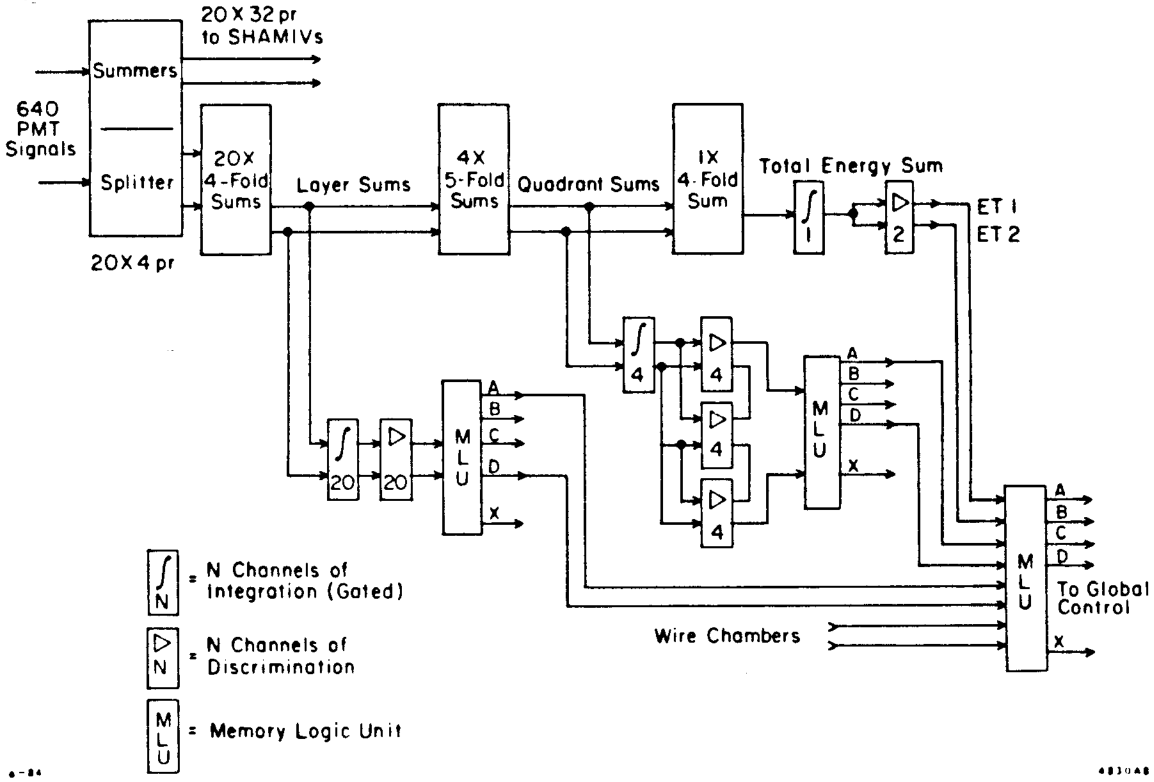

Fig. 1. ASP Energy Trigger block diagram.

The hit layers address a memory look-up which in turn defines allowed combinations of hits; for example, the first two layers of a quadrant but not the last two. The layer sums are also summed in turn into four quadrant sums, which are integrated and each discriminated against three levels defining three energy thresholds for deposit within a quadrant. The resulting twelve signals address a memory look-up that counts and defines combinations of quadrant hits. The quadrant sums are also summed to form a total energy sum, which is also integrated and discriminated against two thresholds. The resulting two total energy sum signals, four-bit combinations of quadrant sums, and four-bit combinations of layer sums, along with signals from PWC's and low-angle shower counters, address a final memory look-up which forms a four-bit output used by the global control module, which issues the trigger interrupt to the data acquisition computer and controls the overall system timing.

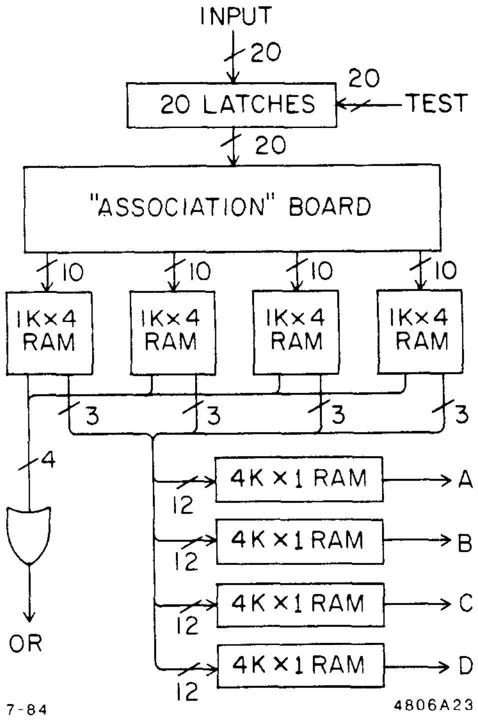

Fig. 2. Memory Logic Unit (MLU) block diagram.

The memory look-up is performed by Memory Logic Units (MLU's) with twenty inputs which address RAM's to provide five outputs, as shown in Fig. 2. To reduce the size of the RAM needed, the latched information from the twenty inputs is passed to an "association" board with forty outputs. This association board is simply a plug-in piggyback card which allows the twenty inputs to be patched in arbitrary fashion to four sets of ten outputs which address four $1K \times 4$-bit RAM's on the main board. One output from each of these RAM's is ORed with a bit from each of the others to produce an OR output from the MLU. The remaining three bits from all four RAM's are combined to address each of four $4K \times 1$-bit RAM's in parallel. These RAM's provide four additional MLU outputs. The RAM's are all read/write, and test inputs to the MLU exist for diagnostics.

2.2 THE MARK III CHARGED PARTICLE TRIGGER

The Mark III charged particle trigger is another example of a fast trigger decision that could be made between beam crossings. In fact, since the Mark III drift chamber has maximum drift times longer than the available decision time at SPEAR, the Mark III trigger has multilevels. The first-level decision is made between crossings, the two second-level decisions are made before the second subsequent crossing, and the third-level is made in the following 100 *μsec*. Only levels 1 and 2b are described here. All levels of the charged particle trigger are described in Ref. 3.

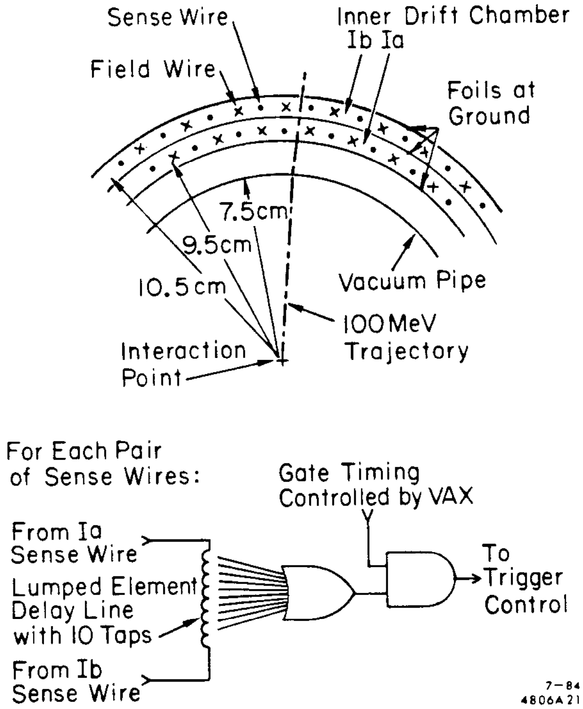

Fig. 3. Mark III Inner Drift Chamber, showing delay line logic for Level 1 trigger.

The Level 1 trigger utilizes the inner drift chamber of the Mark III. This one-meter-long chamber, located 10 *cm* from the interaction point, has two overlapping layers with 1-*cm* drift, as shown in Fig. 3. The Level 1 decision is based on the fact that the sum of the drift times in the two overlapping layers is a constant for tracks originating at the interaction point. It uses a chronotron composed of a lumped element delay line with 10 taps to define a hit. Level 1 reduces the rate to about 300 Hz on the ψ resonance, by restricting the longitudinal acceptance to the length of the chamber and by restricting the radial acceptance and time acceptance using the chronotron. No scintillation counters are used; however, time-of-flight scintillators can also be incorporated if desired. A similar technique is used by the TPC experiment[4] as a first-level trigger.

The Level 2b trigger is based upon fast circle finding using programmed logic arrays. Three drift chamber layers, with radii at 10, 40, and 65 cm, are used as shown in Fig. 4. Since each of the outer layers consists of three sense wires among which only two hits are required, the track-finding efficiency remains at about 95% . The logic searches in parallel for tracks through any of the eighty cells in the outer layer. Through any one outer cell and the interaction point, there exist less than sixteen possible trajectories with momentum greater than 50 MeV. A PAL is associated with each outer cell. Its inputs are that cell, the cells which lie on possible trajectories through the other two layers, and control lines to select a momentum cutoff. The PAL is programmed as an OR of up to 16 AND's such that satisfying any possible trajectory identifies a track. Demanding two or more tracks reduces the trigger rate on the ψ resonance to 3.5 Hz, composed roughly equally of physics, cosmics, and beam-gas. Level 2b decision time requires 25 nsec after the maximum drift time. Searching 80 sets of conditions in parallel provides this speed. Having only 80 sets of conditions by using only three layers to define a trajectory allows the parallel search for tracks. Although this approach has worked effectively, it is potentially susceptible to inefficiency or noise should a chamber layer not operate correctly.

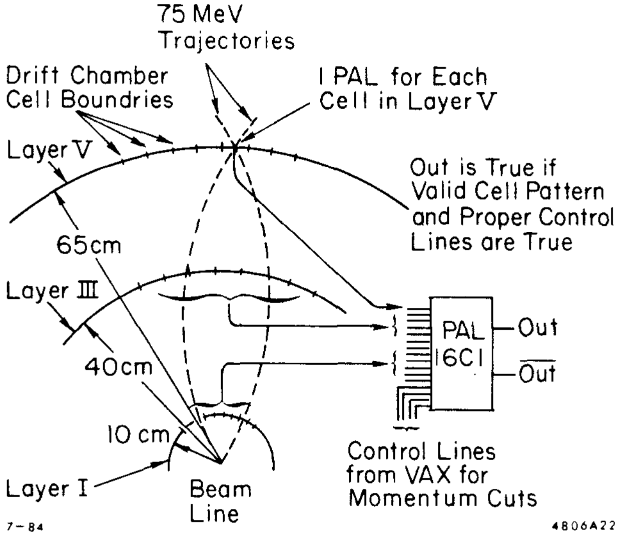

Fig. 4. Mark III Drift Chamber Layers I, III, and V, showing circle combinations and PAL logic for Level 2b trigger.

2.3 The Mark II Charged Particle Trigger

The Mark II charged particle secondary trigger, described in detail in Ref. 5, is an example of a second-level trigger processor which utilizes more complete drift chamber information in a flexible manner. The HRS Curvature Processor[6] is essentially identical, with some minor extensions. The DELCO secondary trigger[7] is also somewhat similar except that only one road pattern is defined. The Mark II processor reads out the cells serially in each of twelve drift chamber layers. This serial readout, which is via shift registers, translates the polar angle of a hit wire into a time t. By shifting twelve layers in parallel at a constant angular velocity, a straight track at an angle ϕ_i gives a coincidence among layers at readout time t_i. Curved tracks are found by appropriately varying the relative delay of the readout of the various layers. This process effectively rotates a set of curved masks through azimuth (see Fig. 5), searching for tracks which match a mask.

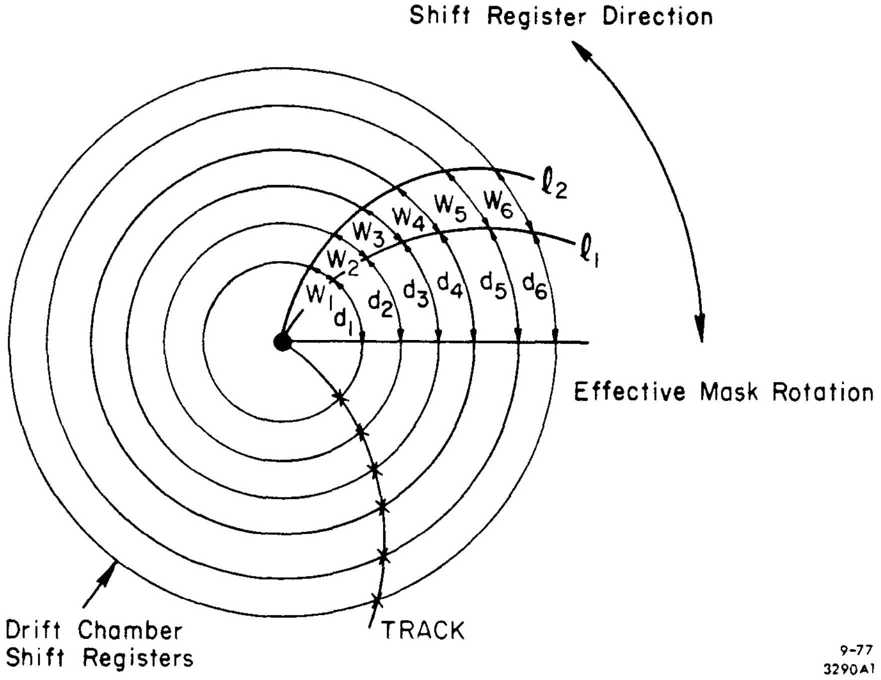

Fig. 5. Mark II Charged Track Finding Principle.

The trigger processor is shown schematically in Fig. 6. Hits in the drift chamber cells are recorded by time-to-amplitude or time-to-digital converters and registered in shift registers in those modules. The shift registers of twelve layers of cells are simultaneously shifted under the control of the Master Clock through a Test and Pickoff module, which places the shift register output onto the auxiliary bus of three CAMAC crates. In these crates, 24 Curvature Modules, operating in parallel, search the data on the bus for tracks in 24 different curvature ranges. A complete set of curvature "masks" is shown in Fig. 7. Identified tracks of three types are signalled to three Track Counter modules which count tracks and record their angles and curvatures. At the end of the process, the track counts are sent to the master interrupt controller where the trigger decision is made.

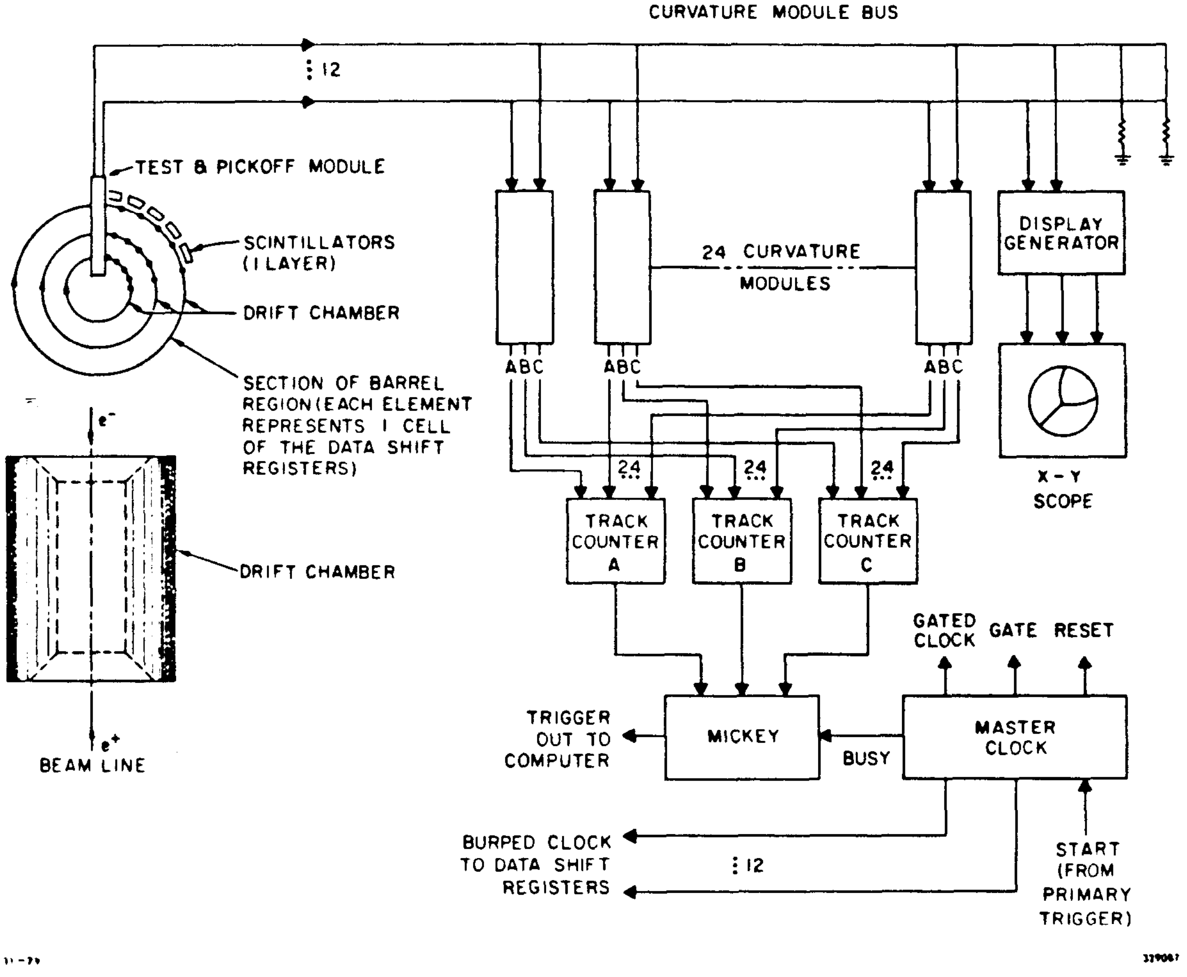

Fig. 6. Mark II Charged Particle Trigger Processor block diagram.

The Test and Pickoff module shifts circularly through 490° in order to find tracks of all curvatures and of both signs at the $\phi = 0$ boundary. It also allows the injection of patterns to test the integrity of the shift registers and of the rest of the logic. The Curvature Modules contain programmable logic to define a momentum bite (curvature) and a road width. Hits in the twelve layers address a $4K \times 2$-bit RAM which is programmed to identify three exclusive track types (seven types in the HRS logic). The programmed parameters are chosen for optimal efficiency using off-line simulation of the hardware interfaced to a physics Monte Carlo. Efficiency is measured using real tracks. The Master Clock module provides twelve separate "burped" clocks for the twelve layers. All clocks are based on the same frequency of 10 MHz; however, clock transitions are periodically skipped for each layer in order to shift all layers at a constant angular velocity. Twelve such clocks are necessary because each drift chamber layer has a different number of cells. The Master Clock is programmable to allow choice of the twelve layers used in the trigger. The Track Counter modules contain logic to avoid double counting of tracks from the same or different Curvature Modules. They also permit CAMAC readout of information about tracks found. This information is used for diagnostics and to aid track reconstruction. In addition, the trigger processor includes a Display Generator, which serves as an invaluable diagnostic tool, and a Colinear Track Finder, which identifies back-to-back track combinations such as small-angle Bhabha events. The master interrupt controller MICKEY manages the multi-level decision process, including gates, resets, and interrupts. It also has memory look-up to trigger on programmable combinations of signals from various trigger processors, including the count of each track type, a count of hit calorimeter modules, and an overall energy threshold. A complete package of diagnostics perform routine tests of the entire processor and identify failing modules.

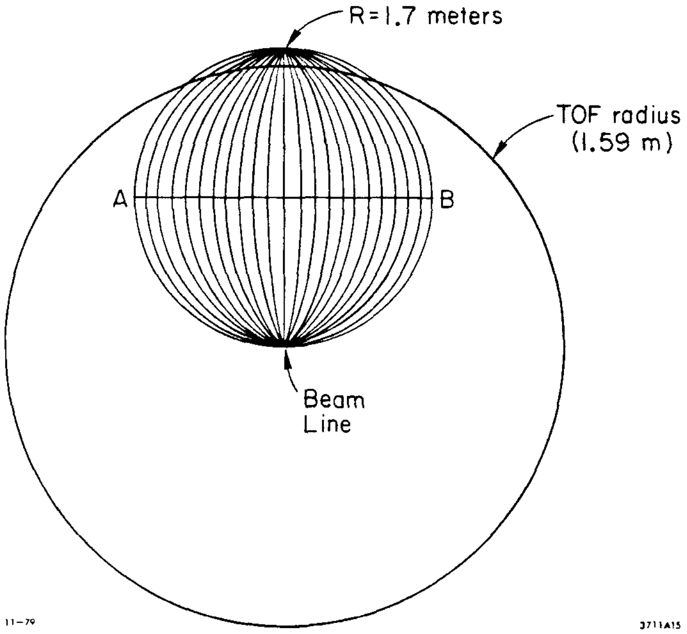

Fig. 7. Mark II Curvature Masks for a given azimuth

The charged-particle trigger decision using this processor requires 35 *μsec*. A compromise between serial and parallel processing is achieved which economizes on the number of connections and on speed. Much drift chamber information is available to the processor, permitting tight roads pointing to the origin which are effective at rejecting background but efficient for finding tracks. Programmability has enabled the processor to function effectively at SPEAR and at PEP with various inner drift chambers and despite the inefficiencies of an aging central drift chamber. This processor will also be used with a new drift chamber at PEP and at SLC.

3. TRIGGERING AT THE SLC

The unique features of the SLC with respect to trigger processing are the long, 5.5-*msec* interval between beam crossings, the high luminosity without high beam current, and the high event rate provided by the Z° resonance. The long interval between beam crossings allows complex, hierarchal trigger processing, including software processing. Such triggers would allow maximum flexibility through programmability, use existing data paths, and allow uniform treatment of all detector subsystems. The lower beam-crossing rate also limits background. Beam-gas rates, with similar vacuums and numbers of electrons per bunch at PEP and SLC, are reduced by more than 10^3 at SLC by the lower crossing rate. Cosmic rates are similarly reduced. Synchrotron radiation, beamstrahlung, and radiation from the beam dump can all be masked, except very near the beam-pipe; however, synchrotron radiation could be a surprise. It could slow trigger processing times and data acquisition times, as well as increasing the background rate. Ability to recognize and reject synchrotron radiation hits could prove to be important.

3.1 THE MARK II TRIGGER AT SLC

Trigger processing for the Mark II detector at the SLC is shaped also by the need to trigger the upgraded detector during checkout at PEP. Consequently, the existing trigger processor (see Section 2.3) will continue to be used. Information from the new central drift chamber will be processed first by trigger cards which interface FASTBUS TDC modules[8] to the existing trigger. These trigger cards connect and are addressed through the FASTBUS auxiliary connector. They latch hits on and do majority logic on the six wires in each drift chamber cell. The results of the majority logic, performed by addressing programmable RAM for all 972 cells in parallel, are then shifted through the existing logic. The ability to do majority logic at the cell level will allow a tighter majority requirement at the road level, which could help reject random background hits. Other new detector elements will be incorporated into the trigger in the fashion of the elements which they replace.

The Mark II at SLC will also be prepared to do software trigger processing if required by severe backgrounds. Synchrotron radiation hits could be rejected, based on pulse height, before shifting drift chamber hits through the existing logic. Beam-gas tracks could be rejected by determining the event vertex position by fitting tracks found by the hardware trigger processor. Such processing would be performed by the SLAC Scanner Processors (SSP's; see Section 4.2) and 3081/E processors (see Section 4.4) which are part of the data acquisition system. The 3081/E's will have available to them the entire event record for trigger considerations.

3.2 THE SLD TRIGGER

Trigger processing for the SLD detector[10] will use data paths similar to the event acquisition paths. Input data for the trigger will be derived in the data acquisition modules. Chamber data, acquired in Waveform Sampling Modules, will be reduced to one bit per wire. Calorimeter data, acquired in Calorimetry Data Modules, will be the digitized energies. This trigger input data will be read out by the same crate-level SLAC Scanner Processors (SSP's; see Section 4.2) used in event acquisition. At the crate level the trigger data will be tested for evidence of tracks and further compressed. For instance, five out of eight drift chamber wires hit in a cell will define a hit cell, and one bit per cell will indicate whether a cell was hit. For calorimeters, the digitizations within a tower will be checked for consistency and compressed to one bit per tower and to a total energy per tower. The crate-level SSP's will send the compressed trigger data to dedicated Trigger Processors. All Trigger Processors and their programs will be identical. The program will perform pattern recognition by table look-up and will result in simple numeric descriptions of recognized patterns. Pattern recognition will be done by separate Trigger Processors for each of the three stereo views of the drift chamber and for each of four calorimeter units. SSP's may serve as these processors. Finally, the recognized patterns will be read by a Trigger Master which makes the final trigger decision. The Trigger Master will also be the event acquisition master, controlling detector resets, waits, etc., data flow, and the running environment. It may be an SSP or a FASTBUS VAX (see Section 4.4). The trigger decision will be complete in time to reset the detector before the next beam crossing.

4. ON-LINE PROCESSING AT SLAC

Large, general-purpose e^+e^- detectors require on-line processors for data acquisition, calibration, monitor, and diagnostic tasks. In order to reduce readout times and to simplify downstream processing, event and calibration data is typically compressed by the data acquisition modules or by crate-level processors such as the BADC and the SSP. Some monitoring and control may be provided by microprocessors, such as the SFC. Thorough monitoring and diagnostics, which frequently involve studying sampled events, sometimes after complete event reconstruction, is usually done by host computers, which at SLAC are VAX's. The host generally writes the data to tape for transfer to off-line processing; however, the MAC experiment at PEP transfers data directly from disk on their VAX to disk on the SLAC IBM facility via a Long Line Adapter. For the complex SLC detectors, processors such as the 3081/E or the FASTBUS VAX will supplement the hosts at the system level in monitor and diagnostic tasks. Microcomputers will manage the flow of data among the on-line processors. The on-line processing systems for the Mark II and SLD detectors at SLC are described in Refs. 9 and 10, respectively. The following sections describe some of the on-line processors commonly used at SLAC.

4.1 THE BADC

The BADC[11] is a microprocessor-based semi-autonomous controller for CAMAC. It was designed to control readout of a CAMAC crate of data acquisition modules, such as sample-and-hold and time-to-amplitude converters. It controls the multiplexing of data onto an analog bus, digitizes the analog data, compresses and processes the data, and buffers results until CAMAC readout. The algorithm most often used for processing of event data discards data below some threshold, corrects data by a quadratic polynomial, and relabels data by function, such as by drift chamber layer number and wire number. Threshold and correction constants for each channel and labeling information are stored in local RAM. This algorithm requires about 3 *μsec* per channel for data below threshold and 10 *μsec* per channel for data above threshold. Another algorithm streamlines calibration procedures by calculating an updated mean and variance for each channel after each event and thereby reducing the amount of CAMAC readout and of host computation. Certain diagnostic algorithms are also implemented.

The BADC economizes on cost, by amortizing the cost of digitizing hardware over hundreds of channels and by allowing higher channel densities. It economizes on readout time, by allowing several crates of electronics to be sparse scanned in parallel followed by block transfer from a small number of BADC's. It economizes on host processing time by correcting and relabeling data at high execution speeds and in parallel, and it simplifies host program structure by handling many thousands of constants.

The architecture of the BADC is described in Ref. 11. Its features are only highlighted here in order to illuminate how it fits into a data acquisition system. The BADC is a triple-width CAMAC module with CPU, RAM, and ADC boards. The ALU is four AMD 2901 four-bit slices controlled by an AMD 2909 microprogram sequencer. Program memory is two 256-word pages of 48-bit PROM. In practice, the second page has never been used in an application. Memory is either $4K$ or $12K$ 16-bit RAM with 220-*nsec* read access. The RAM is used to contain control tables, correction constants, and data buffer. Three clock cycles are defined; short (200 *nsec*) for most operations, long (360 *nsec*) for conditional branches, and pause for operations such as CAMAC I/O which require acknowledgement from another execution unit of the BADC. Interrupts are not implemented. CAMAC commands to the BADC, as well as front-panel signals, force branches to predefined locations in PROM.

The BADC addresses modules within a crate by transmitting an encoded station number N to a SLAC type-U crate controller. The BADC directly accesses the F, A, S1, and S2 lines. Handling of contention on the CAMAC lines is limited. If contention occurs during BADC execution, an error breakpoint is set. A software timer allows a CAMAC cycle which starts BADC execution to complete before the BADC commences CAMAC operation. Control of CAMAC scans is provided by the ADC board which receives the analog data into a three-step pipeline, consisting of address data module, sample-and-hold, and digitize. All CAMAC data is routed through the RAM board, where the D and Y busses of the ALU are interfaced by buffers to the CAMAC W and R Lines. The output buffer from Y to R is gated by a READ from CAMAC. The input buffer from W to D is loaded by a WRITE from CAMAC and gated onto the D bus by microcode which moves data from the buffer, onto the D bus, through the ALU, onto the Y bus, and into RAM in a single CAMAC cycle. Interconnection of the RAM and ADC boards to the CPU and CAMAC are shown in Fig. 8. The modular construction of the BADC has allowed easy modification of the basic design, for instance for 12K RAM boards and for replacement of ADC boards by interfaces for alternate hardware.

Microcode for the BADC is now generated using a machine-independent meta-assembler written in FORTRAN named MAMIC. MAMIC consists of two passes: the first defining the machine and instructions in terms of microcode fields and operations, and the second assembling the user code. It was also used for the SLAC Scanner Processor microcode PROM's and microcode for various other devices such as the VAX CAMAC Channel. In the case of the BADC, the generated microcode can be tested using a debugging RAM in a separate CAMAC module before burning PROM's.

Since its first use in 1977 with the Mark II detector at SPEAR, the BADC has been a central element in the data acquisition systems of many SLAC experiments. Seven different SLAC detectors[12] have used more than fifty of these units (which are now commercially available[13]) for a variety of detector types: MWPC, drift, lead glass, and liquid argon shower counters, drift chambers, MWPC cathode and charge division readouts, time-of-flight counters, muon detectors, and luminosity monitors. Associated data acquisition modules developed for use with BADC's include: four types of sample-and-hold circuits, single-hit and multi-hit time-to-analog converters, and a time-of-flight module containing time and pulse-height measurements. In addition, BADC units have been adapted to alternate hardware configurations, including readout of a second crate, a remote crate, and separate hardware. In the Mark II experiment, BADC's have performed the readout of every detector component except the proportional tubes of the muon system. At SLC, the Mark II will use twenty-two BADC's; however, SLAC Scanner Processors will be used for readout of FASTBUS electronics for the new drift chambers.

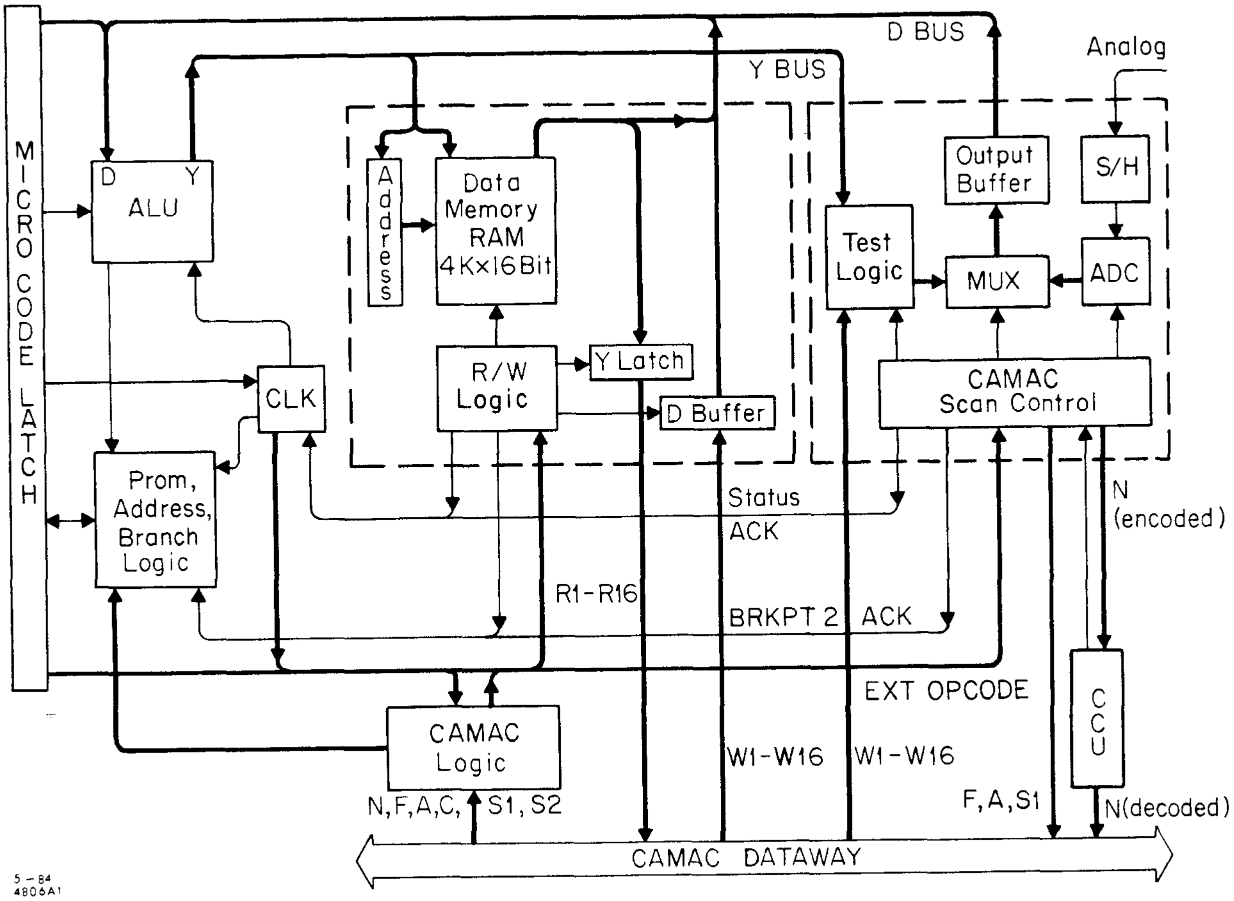

Fig. 8. BADC block diagram.

4.2 THE SLAC SCANNER PROCESSOR

The SLAC Scanner Processor (or SSP), which is described more completely in Ref. 14, is a general-purpose, high-speed, programmable FASTBUS module. It has been designed for the Mark II Upgrade to provide crate-level processing of data from FASTBUS modules similar to that provided in CAMAC by BADC's. The SSP, however, provides a more general, and, hence, more powerful means of moving and processing data in a FASTBUS system. Consequently, SSP's can also be used for various system-level tasks.

From the FASTBUS point of view the SSP can be attached as either a master or a slave to either a crate segment or a cable segment, while being physically connected to both. As a slave, SSP memories can be read and written. The program memory of the SSP maps into CSR space, and data memory maps both into data space and CSR space. Host control of SSP operation as a master is exercised through CSR# 0.

As a slave, the SSP operates without CPU involvement. DS-to-DK response times are about 100 *nsec.*

As a master, the SSP implements a large number of FASTBUS operations. FASTBUS primitives, such as primary address cycle, are implemented as single SSP instructions which include a pointer to an argument list. Other FASTBUS operations, such a read data block, are implemented as single SSP instructions which execute a sequence of FASTBUS primitives at the microcode level. Finally, other FASTBUS operations, such as write random data with pattern select, are implemented as macros of SSP instructions at Assembler level. Instructions which terminate a block transfer normally upon an end-of-block response (SS=2) from a slave and instructions which terminate block transfers upon exhausting a word count are both implemented. Block transfers involve a DK-to-DS delay of about 100 *nsec.* All IBM integer and many byte instructions are implemented for control and data processing.

The SSP consists of two FASTBUS boards, control and processor. The control board contains the interface and I/O logic, which are symmetric with respect to the crate and cable segments, instruction logic, address calculation logic, branch logic, and word count logic. The instruction logic consists of $4K$ words of program memory, PROM address logic, and microcode PROM's in a three-level instruction pipeline. Instructions have IBM format. They are decoded in the microcode PROM's to 56 bits which control the instruction logic and I/O logic and to 48 bits which control the processor. As many as 256 instructions can be defined in the microcode proms. The processor board contains a 32-bit CPU (eight AMD 2901C bit-slices), dedicated input and output shifters to expedite multiple shifts, $32K \times 32$-bit words of data memory, two registers, and condition code logic. The data memory is byte-addressed. It is composed of $16K$ static RAM's and will be easily expandable when $64K$ RAM's become available.

The SSP is designed to be easily programmed. Source code can be written in IBM Assembler or in FORTRAN. Code is then assembled or compiled, translated into separate program and data, optimized, and linked into an image by a translator/linker program on a host computer. This program produces a single file comprised of program and data memory images and of a header which contains information on program size, locations of COMMON blocks, etc. The program and data memories are down-loaded to the SSP over FASTBUS. The initial program status word is loaded into data memory location 0, and execution is started via CSR# 0 or by a hardware signal. Completion of execution can be signalled by a service request or by a front panel output. Execution can be interrupted between instructions if the SSP is so enabled. Instructions can be executed in single-step mode, and a cross-debugger can be implemented via FASTBUS.

The initial application of the SSP is in the drift chamber readout system for the Mark II, where SSP's will scan, process, and buffer data from commercial TDC modules[8] and from SLAC-designed FADC modules. At the crate level, these SSP's will also process calibration data and perform crate initialization, data acquisition module testing, self-testing, and crate segment verification. In addition, SSP's will serve at the system level as cable segment masters and buffers for the cable segments linking the TDC and FADC crates and as data block movers and buffers to and from on-line 3081/E processors. More than 25 SSP's will be used by the Mark II. The SLD detector also plans to use SSP's at the crate level for all detector subsystems ($\sim$ 80 units) to readout and process data from Waveform Sampling Modules and Calorimeter Digitization Modules. The crate-level SSP's also preprocess data for the event trigger, and additional SSP's act as Trigger Processors for each subsystem and as a Trigger Master to complete the trigger decision.

4.3 THE SLAC FASTBUS CONTROLLER

The SLAC FASTBUS Controller (or SFC), which is described more completely in Ref. 15, is a single-card FASTBUS microcomputer suitable for real-time monitoring and control applications. In fact, an SFC can transfer blocks of data at rates of about 5 μsec per 32-bit word (800 KB/sec), which is faster than effective UNIBUS rates (500 KB/sec) and within a factor of about two of fastest CAMAC speeds. It can also implement all possible FASTBUS operations as both a master and a slave and can serve as a host for a FASTBUS system. Moreover, it can be programmed in higher-level languages such as FORTRAN.

The SFC consists of an interface between FASTBUS and MULTIBUS and of any MULTIBUS single-board computer, such as Motorola 68000, Intel 8086, or National 16032, which mounts in the FASTBUS module. The interface connects the 8- or 16-bit MULTIBUS data bus to the FASTBUS AD lines and maps an 8-bit MULTIBUS I/O address onto FASTBUS AS, DS, RD, MS lines. As a master, the SFC executes a FASTBUS cycle by performing a read/write to MULTIBUS I/O space. Processors, such as the 68000 and 16032, which can move a 32-bit longword on the data bus with one instruction are able to execute a FASTBUS cycle, either address or data, in a single instruction. The interface hardware is designed to reduce software overhead by arbitrating for the FASTBUS and by issuing the strobes AS and DS and waiting for acknowledge signals AK and DK while managing a timeout counter, checking parity, and checking for non-zero SS responses. The interface signals cycle completion with the XACK* signal and indicates a FASTBUS error with BERR, thus eliminating software status checking. Interrupts from FASTBUS, such as incoming service request, mastership granted, or selected as slave, are signalled to the processor via GINTR. Three different modes of mastership are supported to enhance execution speed. All FASTBUS operations are supported.

As a slave, the SFC responds in hardware to all types of address cycles and emulates data cycles in software. When the SFC is attached as a slave, the processor is interrupted by GINTR. Then it polls I/O space for DS, reads command bits (RD, MS), and branches to the appropriate command handler which responds to the command and causes the interface to produce DK and SS responses. During the DS-DK interval, the interface normally asserts WT to avoid timeouts. After response is complete, the processor returns to polling DS until the AS-AK lock to the SFC is removed.

As implemented for monitor and control of the Mark II cryogenic system, the SFC is equipped with an Intel iSBC 86/30[16] (8086, 8087 Numeric Data Processor, 256K RAM, and 64K EPROM). FASTBUS standard routines are implemented in ROM, along with a boot program which allows the SFC to be addressed via FASTBUS. Since this SFC uses the same SBC as used in the SLC control system (see Section 5), it can also use the sophisticated software tools developed for the SLC. Applications code can be written in FORTRAN and cross-compiled and linked on a VAX. Code can be downloaded and started via MULTIBUS or FASTBUS; however, the symbolic cross-debugger currently runs only over SLCNET-MULTIBUS. An SFC with a 68000-based SBC has also been implemented at SLAC, and several units have been delivered for implementation outside SLAC.

The SFC is capable of serving as host for a FASTBUS system since it is capable of loading its own logical address and arbitration level registers and of asserting RB and GK to preempt a segment. Additional features include self-diagnostic capability and board area available for wire wrap, for instance, of a sequencer. MULTIBUS is available on the FASTBUS auxiliary connector and to plug-in modules for some expansion and peripherals.

4.4 The 3081/E Processors and FASTBUS VAXes

The 3081/E and FASTBUS VAXes have been described in Refs. 17 and 18 and discussed at this conference.[19,20]

Implemented on line, the 3081/E is a powerful (4 to 5 VAX 11/780 equivalents) processor for applications such as data preprocessing, software triggering, and event reconstruction. It is capable of executing on line the same code used off line for event reconstruction on IBM mainframes. Mark II plans to use 3081/E processors interfaced to FASTBUS through a quasi-dual-ported FASTBUS slave interface. These processors will preprocess drift chamber data from TDC and FADC systems and fully reconstruct events for on-line processing. They will also be used for an on-line software trigger if necessary.

The FASTBUS VAX, equivalent to about 90% of a VAX 11/780 in CPU, offers the easy programmability of VMS, software compatibility with the on-line host computer, and in situ debugging. The SLD plans to use these microcomputers as a master trigger processor and to separately tag physics and background events with a fast filtering algorithm. In addition, they will be used as stand-alone microcomputers for development work.

4.5 The 168/E Processor

The 168/E processor has been described and discussed comprehensively in Ref. 21. These processors have been used at SLAC for off-line event reconstruction by LASS[22] and DELCO and for lattice gauge theory calculations.[23] On line at SLAC they have been used by the SLAC Hybrid Facility (SHF) to provide camera triggers and for monitoring.[24] At SHF, the processors could obtain their data by listening passively to data transfers on a CAMAC backplane.[25] In addition, circuits whose results were sufficiently time critical for triggering were built into a 168/E, where they interacted with the processor as locations in data memory. Data input from PWC digitizers and space point finders were implemented as read-only memories. Camera triggers were provided by accessing memory addresses on other 168/E boards.

4.6 The VAX CAMAC Channel

The VAX CAMAC Channel (or VCC), described in Ref. 26, is a UNIBUS to CAMAC interface utilizing a bipolar microprocessor to supervise the CAMAC system and minimize the host computer's work. The high speed 16-bit CPU is composed of AMD 2900 bit-slice bipolar microprocessor components. The interfaces to a DEC UNIBUS and to CAMAC are peripherals to the CPU linked by input and output data busses, a status bus, and a microcode bus for interface control. The CAMAC interface connects, via SLAC CAMAC protocol, to a set of system crates which house Branch Drivers for parallel or serial branches or for other data links. A VCC operation is initiated by a software device driver which passes the VCC the addresses of control and data buffers. No further VAX CPU activity is required until the VCC has completed the list of data transfers implied by the control "channel program" buffer. The VCC fetches commands from VAX main memory, sets up the Branch Drivers and transfers data between the CAMAC modules and VAX main memory at rates limited by the UNIBUS. Address scanning for block transfers is handled by the Branch Drivers. Data acceptance, conditioned by X and Q responses, and data packing into 16-bit words are performed by the VCC. The VCC also serves as a channel to the VAX for hardware interrupts. A coherent software package integrates high-level programs with system driver-level programs and microcode control of the system.

5. SLC CONTROL SYSTEM

The SLC control system[27] will be an order of magnitude more complex than SLAC's other accelerator control systems. In addition to modernizing and streamlining the operation of the present linac/beam switchyard system, the SLC control system must provide machine modeling to support the extensive accelerator development required to meet the tight SLC beam requirements. In its final configuration, the SLC control system (see Fig. 9) will provide a combination of two VAX 11/780 central processors networked to 70-100 powerful microprocessor clusters which interface through CAMAC with the equipment to be monitored and controlled.

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

The distributed micro-processor clusters are 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 clusters are based on the Intel Multibus architecture, which provides support for an arbitrary number of single-board computers which communicate with each other through the use of shared memory and interrupts. The micro-processor clusters contain an Intel iSBC 86/30[16] with 8087 coprocessor, 786 Kilobytes of RAM, and 8 kilobytes of EPROM, providing about 1/7 of the processing power of the VAX 11/780. The microprocessor clusters interface to a 5 MHz serial CAMAC branch through a DMA channel device.

Intelligence is also distributed into the CAMAC crates via the use of dedicated microprocessors in some of the data acquisition modules. The Smart Analog Monitor (SAM) is a Zilog Z80-based CAMAC module that continuously scans 32 analog channels, auto-calibrates, auto-ranges, digitally filters, and provides floating point voltage values in either VMS or IEEE formats. The Parallel I/O Processor CAMAC module (PIOP) is a general-purpose processor based on the Intel 8088. It interfaces to CAMAC and to a front panel port which is a differential transmitter/receiver version of the micro-processor's bus structure. This port provides a standardized method of interfacing to specific devices or processes. For instance, PIOP's monitor the phase and amplitude of the 240 linac klystrons, as well as provide general klystron monitor and control. Code for the PIOP is cross-compiled or cross-assembled on the VAX, then downloaded via CAMAC to the PIOP or "burned" into EPROM.

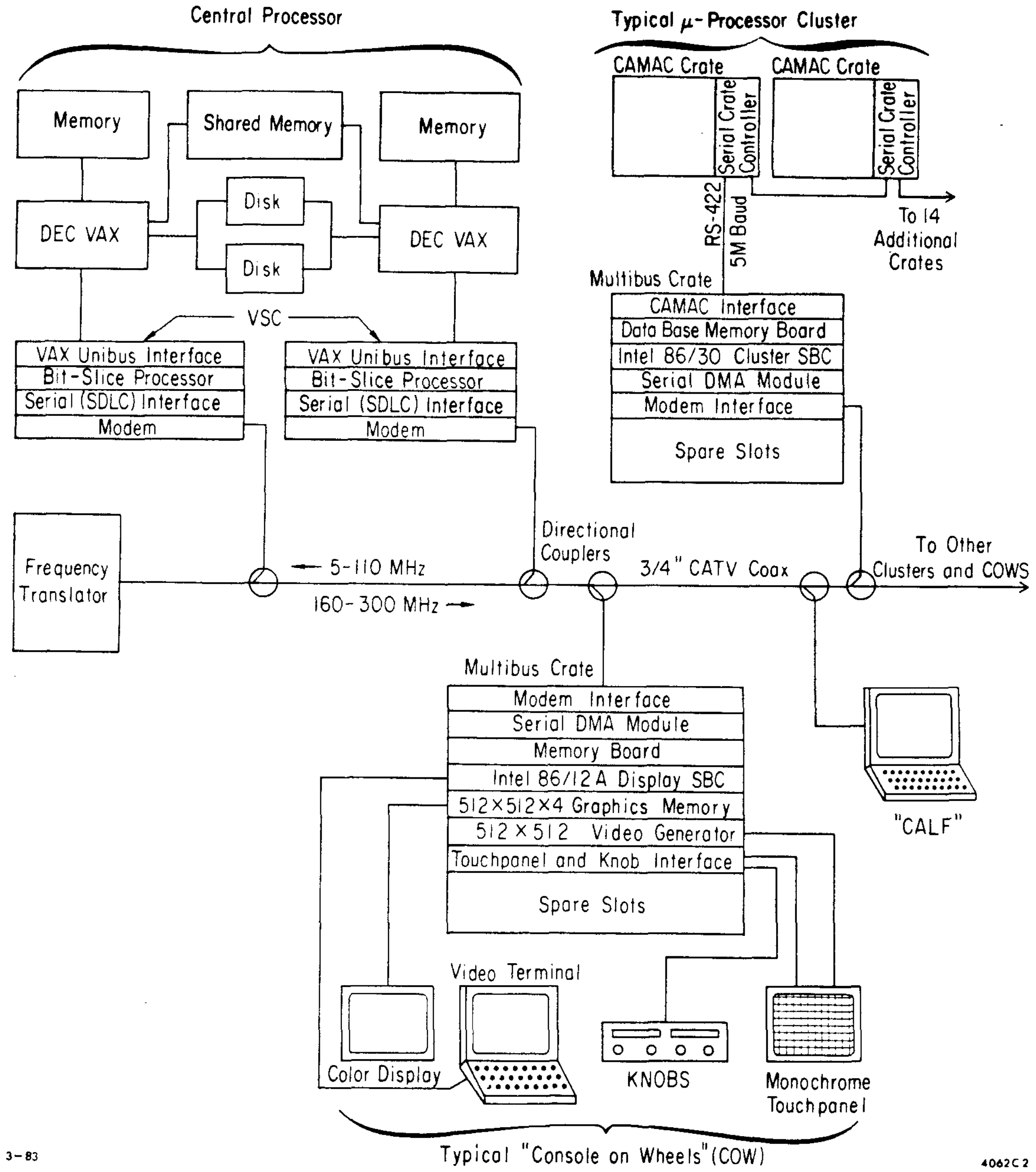

Fig. 9. SLC Control System block diagram.

There are two types of operator consoles in the system. The Console-On-Wheels (COW) consists of a micro-processor cluster, high-resolution color graphics display, touch panel, general-purpose knobs, computer terminal, video monitor, and audio intercom; yet the COW is a fully portable unit which can be connected at any point along the system's Communication Line. The Curser-Addressed Limited Facility (CALF) emulates a subset of COW functions and can be also connected at any point along the

Communications Line. The number of COW's and CALF's that can be supported is limited only by the system's processing power.

The Communications Line for the system consists of a broadband (5 - 300 MHz) cable television (CATV) system. Communications are organized in an inverted-tree topology with mid-split repeaters. Low frequencies feed from the source to the up-converters, and high frequencies feed to all receivers. Several sub-systems use the cable for communications. The micro-processor clusters and the VAX's are interconnected by a one Megabaud polled network. A bit-slice processor directs sequential polling at a rate of about 1000 polls/second and serves as a DMA channel to the VAX. The logical topology is a star network with communication only between the host and the microclusters.

A significant effort has been expended to create an efficient and user-friendly environment for the development of micro-processor software. All SLC software development is performed on the VAX. Wherever possible, which has been in about 95% of cases, FORTRAN 77 is used for applications programming of the micro-clusters. (VMS FORTRAN is used for the VAX's.) In collaboration with Intel, 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 cross-debugger has been developed to allow the remote debugging of micro-processor programs running under iRMX. In this complex system, on-line debugging and diagnostic aids have been essential in order to trace problems efficiently in the system's operating environment.

6. SUMMARY

Trigger and on-line processors share the role of reducing data rates to manageable proportions in large, general purpose e^+e^- physics experiments. This paper has attempted to review a vast amount of work on trigger and on-line processors at SLAC.

Generic solutions to trigger processing have evolved at SPEAR and at PEP, although special-purpose hardware trigger processors have been developed for each experiment. Some of these processors, on the other hand, have a great deal of flexibility and programmability. The environment at the SLC will offer the possibility of sophisticated, software triggers.

On-line processor development at SLAC has focused on data reduction and preprocessing at the data acquisition crate level. The BADC and the SLAC Scanner Processor are examples of such processors. At the SLC, more powerful processors, such as the 3081/E and the FASTBUS VAX, will be used for more sophisticated preprocessing, software triggering, and event sampling and flagging. The SHF use of the 168/E pioneered on-line use of such powerful processors. In addition, general-purpose microprocessors, such as the SLAC FASTBUS Controller and the Parallel I/O Processor, are assuming roles in experiment and accelerator control. At all levels, a high premium has been placed upon uniformity of approach, for both hardware and software, leading to the development of a small number of flexible yet powerful processors and sophisticated software tools.

REFERENCES

1. G. M. Ronan, et.al., "Triggering the LBL Time Projection Chamber," IEEE Trans. Nucl. Sci., Vol. NS-29, No. 1 (1982).
2. R. Hollebeek, 3rd International Conference on Instrumentation for Colliding Beam Physics, SLAC-PUB-3347.
3. J. J. Thaler, et.al., "Event Trigger for the Mark-III Detector at SPEAR," IEEE Trans. Nucl. Sci., Vol. NS-30, No. 1 (1983).
4. H. Ailiara, et.al., "Performance of a Drift Chamber System for the Time Projection Chamber Detector Facility at PEP," IEEE Trans. Nucl. Sci., Vol. NS-30, No. 1 (1983).
5. H. Brafman, et.al., "Fast Track-Finding Trigger Processor for the SLC/LBL Mark II Detector," IEEE Trans. Nucl. Sci., Vol. NS-25, No. 1 (1978).
6. Robert J. Wilson, "A Search for Scalar Electrons at PEP," Thesis, August 1983.
7. Dale Quimette, et.al., " A Versatile Secondary Trigger for a Multi-Detector System," IEEE Trans. Nucl. Sci., Vol. NS-30, No. 1 (1983).
8. LeCroy Model 1879, LeCroy Research Systems Corp., Spring Valley, New York.
9. A. J. Lankford and T. Glanzman, "Data Acquisition and FASTBUS for the Mark II Detector," IEEE Trans. Nucl. Sci., Vol. NS-31, No. 1 (1984).
10. SLD Design Report (Preliminary Edition), April 9, 1984.
11. M. Breidenbach, et.al., "Semi-Autonomous Controller for Data Acquisition, The Brilliant ADC," IEEE Trans. Nucl. Sci., Vol. NS-25, No. 1 (1978).
12. BADC's have been used by the following SLAC experiments (numbers of units shown in parentheses): LASS (6), Hybrid Facility (1), Free Quark Search (1), MAC (7), ASP (5), Mark III (15), and Mark II (20).
13. TRANSIAC Model 7001, TRANSIAC Corp., Mountain View, CA.
14. H. Brafman, et.al., "The SLAC Scanner Processor," submitted to IEEE 1984 Nuclear Science Symposium, Orlando, Florida, Oct. 31 - Nov. 2, 1984.
15. S. R. Deiss, "A Fastbus Controller Module Using a Multibus MPU," IEEE Trans. Nucl. Sci., Vol. NS-30, No. 1 (1983).
16. Intel Corp., Santa Clara, CA.

17. Paul F. Kunz, et.al., "The 3081/E Processor," published in the *Proc. of the Three Day In-Depth Review on the Impact of Specialized Processors in Elementary Particle Physics*, Padova, Italy, March 23-25, Naz. Fis. Nucl. (1983).

18. E. J. Siskind, NYCB Real Time Computing, "Development of 32 Bit Single Board Computer Systems in Fastbus Packaging," DOE Small Business Innovation Research Program (Private Communication).

19. P. K. Kunz, et.al.,"The 3081/E Processor," published in these Proceedings.

20. E. J. Suskind, "VAX FASTBUS Module," published in these proceedings.

21. P. K. Kunz, "The LASS Hardware Processor," Nucl. Instrum. Methods 135, 435 (1976). P. K. Kunz, et.al., "The LASS Hardware Processor," 11th Annual Microprogramming Workshop, Pacific Grove, CA, Nov. 19-22, (1978). SIGMICRO Newsletter 9, 25 (1978). P. K. Kunz, "Use of Emulating Processors in High Energy Physics," *Proceedings of the International Conference on Experimentation at LEP*, Phys. Scr. 23, 492 (1981).

22. P. K. Kunz, et.al.,"Experience Using the 168/E Microprocessor for Off-Line Data Analysis, IEEE Trans. NS-27, 582 (1980).

23. J. E. Hirsch, et.al., "Monte Carlo Simulations of One-Dimensional Fermion Systems," NSF-ITP-82-44.

24. J. T. Carroll, et.al., "On-Line Experience with the 168/E," published in the *Proc. of the Topical Conference on the Application of Microprocessors High-Energy Physics Experiments*, CERN, Geneva, Switzerland, 4-6 May 1981.

25. D. Bernstein, et.al., "Snoop Module CAMAC Interface to the 168/E Microprocessor," IEEE Trans. Nucl. Sci., Vol. NS-27, No. 1 (1980).

26. D. J. Nelson, et.al., "The VAX CAMAC Channel," IEEE Trans. Nucl. Sci., Vol. NS-28, No. 1 (1981).

27. R. Melen, "A New Generation Control System at SLAC," IEEE Trans. Nucl. Sci., Vol. NS-28, No. 3 (1981). J. D. Fox, et.al., "Application of Local Area Networks to Accelerator Control Systems at the Stanford Linear Accelerator," IEEE Trans. Nucl. Sci., Vol. NS-30, No. 4 (1983). R. Melen, "Centralized Digital Control of Accelerators," invited talk to the Europhysics Conference, Computing in Accelerator Design and Operation, Berlin, West Germany, 20-23 September 1983.