CONF-801126--1

CONF-8011126--1

DE82 004095

A LARGE CAPACITY, HIGH-SPEED MULTIPARAMETER MULTICHANNEL ANALYSIS SYSTEM*

R. W. Hendricks,† P. A. Seeger,‡ J. W. Scheer‡ and S. Suehiro†,§

Preprint of a paper presented at the Workshop on
"X-Ray Position-Sensitive Detectors and
Energy Discriminating Detectors"
Hamburg, West Germany
17–21 November 1980

(Submitted to Nuclear Instrumentation Methods)

*Research sponsored by the Division of Materials Sciences, U.S. Department of Energy, under contracts W-7405-eng-26 with the Union Carbide Corporation and W-7405-eng-36 with the University of California.

†Metals and Ceramics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA

‡Group P-8, Los Alamos Scientific Laboratory, Los Alamos, NM 87545 USA

§Permanent Address: Department of Polymer Chemistry, Faculty of Engineering, Kyoto University, Kyoto 606, Japan

MFW

A LARGE CAPACITY, HIGH-SPEED MULTIPARAMETER MULTICHANNEL ANALYSIS SYSTEM*

R. W. Hendricks,† P. A. Seeger,‡ J. W. Scheer‡ and S. Suehiro†,§

ABSTRACT

A data acquisition system for recording multiparameter digital data into a large memory array at over 2.5 MHz is described. The system consists of a MOSTEK MK8600 2048K x 24-bit memory system, I/O ports to various external devices including the CAMAC dataway, a memory incrementer/adder and a daisy-chain of experiment-specific modules which calculate the memory address which is to be incremented. The design of the daisy-chain permits multiple modules and provides for easy modification as experimental needs change. The system has been designed for use in multiparameter, multichannel analysis of high-speed data gathered by position-sensitive detectors at conventional and synchrotron x-ray sources as well as for fixed energy and time-of-flight diffraction at continuous and pulsed neutron sources. Modules which have been developed to date include a buffer for two-dimensional position-sensitive detectors, a mappe. for high-speed coordinate transformations, a buffered time-of-flight clock, a time-correlator for synchronized diffraction experiments, and a display unit for data bus diagnostics.

---

*Research sponsored by the Division of Materials Sciences, U.S. Department of Energy, under contracts W-7405-eng-26 with the Union Carbide Corporation and W-7405-eng-36 with the University of California.

†Metals and Ceramics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA

‡Group P-8, Los Alamos Scientific Laboratory, Los Alamos, NM 87545 USA

§Permanent Address: Department of Polymer Chemistry, Faculty of Engineering, Kyoto University, Kyoto 606, Japan

INTRODUCTION

This paper describes a high-speed digital data acquisition system which has been designed for use at the Los Alamos WNR pulsed neutron source (Russell et al., 1978), at the Oak Ridge 10-m Small-Angle X-Ray Scattering Camera (Hendricks, 1978) and at the ORNL beam line now under design for the National Synchrotron Light Source at Brookhaven National Laboratory. Applications to other areas of research will be pointed out.

Very large quantities of data can be generated by two-dimensional position-sensitive detectors, especially when a third parameter must be monitored. At LASL, the neutron time-of-flight must be recorded, while at ORNL high-speed time switching experiments are being performed in synchronization with an applied stress on the sample (Hendricks, 1979). The ORNL synchrotron beam line will use an array of linear PSD's at first (an area detector is planned later); but the data rates will be very high, perhaps in excess of 10^6 events/s. These experimental conditions are not very different from those at many major radiation scattering centers around the world. To date, each facility has developed its own approach to these data acquisition problems with varying degrees of sophistication. Many of the problems were discussed at a recent LASL-ANL meeting (Kitchins, 1979) where it became clear that a joint effort might develop an excellent, general purpose data acquisition system which could be suitable for a wide range of experimental needs. This paper describes one possible implementation of such a system.

DESIGN CONSIDERATIONS

When data rates over 1 MHz are anticipated, it is clear that the internationally accepted CAMAC data acquisition interface system (U.S. NIM Committee, 1976) with its 1 μs dataway cycle time is impractical, especially if data must be processed by multiple modules. On the other hand, data processing with TTL logic and with readily available inexpensive commercial memory systems at rates well in excess of 3 MHz is easily achievable. Following ideas presented by Machen (1979) and others, an obvious solution is to perform data acquisition on a separate, faster bus than the CAMAC dataway, but to continue to use CAMAC for lower speed module control. Further, in most data acquisition systems, the flow of data from the experiment to the recording system is unidirectional. However, it is usually essential that external devices (e.g., a computer) have bidirectional access to the memory but at a lower priority than the experiment.

To achieve maximum throughput and flexibility, we have concluded that the experiment (which determines the memory location to be incremented) and any other memory accessing devices should be on separate buses. A schematic configuration is shown in Fig. 1. It is seen that we propose a system which consists of a large capacity 24-bit memory system for which we have developed a four-port interface which contains three general-purpose I/O ports and a high-speed adder/incrementer port. Various interfaces to other computer I/O systems are handled through the general-purpose ports while the interface to the experiment is via the adder/incrementer port and a series of address manipulating CAMAC modules. A photograph of our completed system is shown in Fig. 2.

In the following paragraphs, we describe our specifications for a general-purpose daisy-chain bus for the experimental modules, our selection of the memory system and its bus, and the design of our four-port interface to the memory bus which allows access to the CAMAC dataway, the minicomputer I/O bus, the LASL Bulk Memory Processor (BMP) and other devices such as on-line microprocessors. Complete design details of each of these components, including electronic schematic diagrams, are given in a separate report (Hendricks, Seeger, Scheer, and Suehiro 1981).

MEMORY SYSTEM

Introduction

There is a wide variety of memory systems with well-established bus structures available from numerous manufacturers. For our application, we require a system capable of directly addressing at least 2 Megawords of 24-bit memory. Parity and error checking and correction (ECC) were considered as nonessential luxuries. The large word size (24 bits) was considered essential in order to minimize overflow checking in very high data-rate applications such as the synchrotron data acquisition system. A read-modify-write split cycle with a cycle time (not including modify) of <600 ns is required in order to meet our goal of 1 MHz data rates on the data bus. The system must consist of at least: (1) a chassis and power supply to contain and power the memory cards, (2) automatic refresh (if the memory is dynamic as anticipated), (3) an easily accessible and well-defined bus, and (4) user-adaptable interface cards.

Memory System and Bus

Following McMillan et al. (1979), who are designing a 4 Megaword semiconductor Bulk Memory bit-slice microprocessor system (BMP) and who have thoroughly investigated the cost-effectiveness of various commercially

available memory systems, we have selected the MOSTEK MK8600 Memory System (MOSTEK Inc., 1979) as the memory system of choice for this project. This system has a capacity per chassis of 16 cards each containing 128 K by 24-bit words for a total of 2048 K words. The access time is 250 ns and the cycle time is 450 ns. Further information, including detailed descriptions of the signals on the memory bus, can be found in the MOSTEK MK8000 MOS RAM technical manual. The system defines a bus with 24 bits of input data, 24 bits of output data, 21 bits of memory address (per chassis), and a number of control lines. In developing the interface described below, we have also implemented a 16-way interleaving scheme which reduces the effective cycle time for sequential or random accesses to the memory to about 275 ns.

Memory I/O Port

External devices are interfaced to the memory through one of three identical I/O ports or through the increment port described in the following paragraph. The device-dependent portion of the interface between a peripheral device and the memory system may be mounted on a board near or associated with the device while the three general memory I/O ports and the memory-increment port are constructed on a MOSTEK MK8500 I/O card. The bus connecting the two sections is defined in Fig. 3. A device sets up a 22-bit memory address and a R/$\overline{W}$ signal and may strobe MREQN whenever the I/O port is not BSY. The port will then assert BSYN and hold it until the requested operation is complete. All writes are done in a read-modify-write cycle, so that the external device may operate on the data. Whenever data are ready for writing (which may be at the beginning of the cycle), the device issues the write ready strobe (WRDYN).

The address and control signals are TTL levels; the control signals use a low-true convention. The data bus is a bidirectional tri-state bus; if the device is going to read data from memory it must assert the READN line to allow the port to place data on the bus. READN should be removed no later than the beginning of data set-up for a write.

A timing diagram for the I/O port is shown in Fig. 4. The times schematically indicated there are given in Table I. It is seen that the address (and data on write) must be valid 20 ns prior to MREQ↓ (or WRDYN) and must be held 60 ns following the ↓ transition. The data (on read) will be set up 20 ns prior to the ↑ transition on BSYN and are guaranteed for 60 ns following the transition.

With an I/O port structure of the design described here, it is easy to construct device-dependent interfaces which will make the MOSTEK memory available to a wide variety of devices. One such interface (the first to be considered for this project) is that to the CAMAC dataway and is described in the CAMAC Dataway I/O Port section of this paper. Also, the LASL BMP will interface directly as the same signals and connectors are used. Interfaces planned for future development include a DMA access to the ModComp I/O bus and an interface to at least one industry-standard microcompter bus such as Intel's MULTIBUS (Barthmaier, 1980) or Motorola's VERSABUS (Kister and Robinson, 1980).

Memory Increment Port

In the experiments of interest to us, various kinds of x-ray and neutron detectors record the presence of single particles (Hendricks, 1976). The data acquisition modules described in the next section compute the address in MOSTEK memory which represents the event. That

address is passed to a unique fourth port in the memory, the Memory Increment Port. The purpose of this port is to fetch the contents of the address presented, increment it (or add data to it) and restore the new value.

A block diagram of the auto-increment circuit is shown in Fig. 5. In response to a Request In (REQI) strobe, which is only allowed when Busy Out (BSYO) is not true, the address to be incremented is latched and BSYO is asserted back to the data acquisition bus. Whenever the memory bus is available, a split cycle (read-modify-write) is initiated; this port has higher priority than the general ports. The data out from memory are loaded into a 24-bit full adder, and the "Data Available" signal from the memory (after a suitable delay) asserts "Cycle Continue" which loads the modified value back into the same memory address. The measured cycle time for the adder/incrementer port, as determined from the leading edge of REQI to the trailing edge of BSYO, is 385 ns. As will be seen, this is the slowest time in the data acquisition daisy chain, and thus limits the date rate of our system to just over 2.5 MHz.

If the 24-bit adder overflows, the address may be made available to an external port along with an OVF strobe. This address may be latched into a standard CAMAC 24-bit input register and a LAM may be initiated. At our design rate of 1 MHz, when incrementing by one, a 24-bit memory will overflow every 16.8 s if every event is directed at a single memory location. Thus, overflow handling even under the worst conditions is a very low data-rate problem.

Although the detectors which are of immediate application in LASL and ORNL research are of the single-event gas-filled proportional type, the merits of TV-based position-sensitive detector systems (Hendricks, 1976), especially for synchrotron radiation research, should not be overlooked. In such devices, data at a given (x,y) coordinate pair are integrated for a fixed time, T, in the video tube. The contents are

read out by the scanning electron beam, the intensity of which represents the number of photons stored during T. After digitization, the number of photons stored at each location in the vidicon tube will then be added to the contents of the corresponding memory location. In our interface, the speed of this addition is the same as the single-event increment mode: 2.5 MHz. The auto-increment port has been constructed on the same MOSTEK MK8500 I/O card as the three general-purpose I/O ports, as shown in Fig. 6.

Priority Arbitration

Competition among the three I/O ports and memory incrementer for priority on the MOSTEK bus must be arbitrated. In the present application, the memory incrementer clearly has priority; priority among the I/O ports is at the discretion of the experimenter, merely by choosing which connector he uses for each device. For these purposes, a very simple daisy-chain request/grant type arbitration scheme can be used (as is done with several well-known mini- and micro-computer systems). No port may demand the bus for more than a single cycle. At the time a port is ready to request a bus cycle (1) it disables the grant-out line to lower priority ports, (2) it affirms the request line, and (3) it waits for grant-in to come down from higher priority ports. The highest priority device (in our case, the incrementer) has its grant-in true and its grant-out false until a request comes in from a lower priority port. The maximum time for priority arbitration is normally 75 ns. In case of coincidence (within about 3 ns), the fail-safe solution implemented is not to grant priority to either port, but to start another arbitration cycle.

CAMAC DATAWAY — I/O PORT INTERFACE

In order to make the data acquisition system described in this paper available as rapidly as possible for several different experiments, some of which use different computers, we have chosen to develop a computer-independent CAMAC I/O port first. Computer-specific interfaces will follow at a later time.

The CAMAC interface must realize several goals: (1) data must be buffered on both memory read and write requests because there is no way to synchronize the CAMAC dataway and the memory bus and because the memory bus may not be available at the time the CAMAC dataway requires access to it; (2) block transfers of multiple words of data must be possible; and (3) the memory locations accessed in a block transfer need not be sequential. The last requirement will allow the block transfer of various combinations of two orthogonal parameters regardless of the mapping scheme. This capability is very important for on-line graphical display of time-varying data.

A block diagram of our implementation of the CAMAC interface is shown in Fig. 7 and the function codes to which it responds are given in Table II. A photograph of the completed module is shown in Fig. 8. It is seen that there are three on-board registers; the memory address register (AR), the memory address increment register (AIR), and the transfer count register (TC). There is also a data buffer which is filled in response to memory read requests. The data buffer for memory write requests resides in the I/O port interface. Two types of CAMAC operations are decoded in the interface; those which actually initiate an I/O request to the memory I/O port, and those which operate on information contained exclusively in the CAMAC module. As can be seen in the

timing diagram of Fig. 9, when an I/O request is initiated the I/O port immediately (within 30 ns) asserts BSY. If there is an attempt to execute a CAMAC command which initiates an I/O request while the I/O port is busy, the CAMAC system may take one of two actions depending on the design of the crate controller. The module described here, on decoding A(0)[F(4)+F(16)+F(22)]+A(1)[F(20)+F(22)], immediately sets the crate controller line P2 low if BSY is asserted by the I/O port (see Fig. 9). If the crate controller has the P2 "stretch cycle" feature implemented [as in the case for the crate controllers developed by Seeger (1976) and which are in use in Oak Ridge and at WNR], then if P2 goes low before the S1 strobe, the dataway cycle is held and does not resume until P2 returns high. Thus, dataway functions which initiate I/O port activity are forced to wait for completion of a prior request. The module generates a Q = 1 response to indicate successful completion of the requested function. If the crate controller does not respond to P2, the dataway cycle continues without pause. In this case, our module (1) makes no data transfer, (2) does not initiate any I/O port request, and (3) generates Q = 0 for that dataway operation. Because the timing and logic are slightly different in each of these cases, we have implemented a jumper selectable logic for both. Clearly, it is the programmer's responsibility always to check for a Q = 1 response to be sure the I/O request was actually performed.

Single-word data transfers are performed quite simply. The programmer first loads the memory address register and then reads or writes the data buffer. The CAMAC functions are:

READ:	A(1)F(20)	Load AR; initiate REQ
	A(0)F(0)	Read buffer
WRITE:	A(1)F(16)	Load AR
	A(0)F(16)	Write buffer; initiate REQ

In each case, any I/O request following the second function should be checked for a Q = 1 response to ensure that the I/O request was performed. If Q = 0, the operation can be reinitiated without loading AR again. It is also important for the programmer to be sure that any previously issued I/O requests have been completed prior to changing AR, AIR or TC.

The above example indicates that two dataway cycles are required for each word of data which is to be moved between the computer and the memory. The overhead of loading the memory address register for every word can be reduced significantly by implementing a means of automatically changing the contents of the memory address register following each successful I/O transfer. In addition, if the crate controller design has implemented a means for automatically re-executing a CAMAC function (Seeger, 1976), the operation can be performed multiply until a recognized STOP event occurs. The module described here implements two of the most common STOP modes for block transfers; Q = 0 (Q-STOP) and LAM. Here, in addition to loading the memory address register, one also loads a memory address increment register and a transfer count. The following CAMAC operations will then transfer a block of data:

```
READ:   A(2)F(16)   Load TC
        A(3)F(16)   Load AIR
        A(1)F(22)   Load AR; initiate REQ; AR = AR + AIR; TC = TC — 1
        A(0)F(4)    Read; initiate REQ; AR = AR + AIR; TC = TC — 1
        A(0)F(4)    .
        .           .
        .           .
        .           .
        A(0)F(4)    Repeat until Q = 0 or LAM

WRITE:  A(1)F(16)   Load AR
        A(2)F(16)   Load TC
        A(3)F(16)   Load AIR
        A(0)F(22)   Write; initiate REQ; AR = AR + AIR; TC = TC — 1
        A(0)F(22)   .
        .           .
        .           .
        .           .
        A(0)F(22)   Repeat until Q = 0 or LAM
```

Note that if the P2 "stretch" is not implemented, a Q = 0 response may be generated even though TC > 0 if any I/O operation occurs while BSY is asserted. Thus, when the block transfer terminates, the programmer should read the TC register to verify that all the requested data were transferred.

If hardware direct memory processor (DMP) data transfers have been implemented in the CAMAC crate controller, the block I/O transfers described above can be very efficient. Even without hardware DMP mode transfers they reduce the amount of register-mode computer I/O data transfers by a factor of 2.

DATA ACQUISITION SYSTEM

Introduction

The various experiments for which this data acquisition system was designed vary significantly in their needs for digital pre-processing prior to feeding a memory address to the auto-increment port. However, all experiments need some form of interface to the detector analog position-encoding electronics, an address mapper or calculator, and perhaps some time information either in the form of time-of-flight data or time-synchronization with external stimulae on the sample. In addition, a first-in/first-out (FIFO) buffer is desirable to handle data bursts or situations in which the data acquisition chain does not have priority access to the memory. A typical experimental setup is shown in Fig. 10.

It is especially important to note that for the experiments in which we are interested data flow is unidirectional from the experiment

to the memory. Thus, in order to achieve as much flexibility and modularity as possible, we have designed a unidirectional 24-bit data bus with two control lines. This bus is specified in the next section, and then in succeeding sections we describe some of the modules which have been designed to do time-of-flight neutron scattering with an area detector and x-ray and neutron small-angle scattering with an area detector during time-synchronized mechanical deformation of the sample.

Specification of Data Bus

The MOSTEK MK8600 Memory System selected in the Memory System and Bus section is limited (in one box) to 2048 K words; any memory location may be directly addressed with 21 bits. (Note however that the BMP will support 22-bit addresses.) Thus, our intermodule data bus is up to 22-bits wide. These are in a single cable with Cannon double density 52-pin connectors. As suggested in the IEEE CAMAC standard (1976), the chassis-mounted connectors are pin blocks and the cables have sockets. The bus is controlled by two lines; request (REQ) and busy (BSY). These lines are separate from the address bus and are individual cables with LEMO connectors. In this manner, various modules can be wired in the daisy chain to exert control as dictated by experimental conditions.

The data may be either low or high true while the control lines (REQI, REQO, BSYI, BSYO) are high true. Differences between low or high true addresses are accommodated by replacing four chips in the incrementer port; 74LS366 are used for low true or 74LS367 for high true addresses. A module may assert Request Out (REQO) any time that Busy In (BSYI) is not true. It is the responsibility of the next module downstream to latch the input data following a REQI and then immediately to assert BSYO. This BSYO will normally be connected back to BSYI in the

upstream module, where the positive transition (↑) of BSYI indicates acceptance of the data by the downstream module. Data output from a module must be set up at least 20 ns prior to asserting REQO. REQO may be a pulse (at least 30 ns wide), but in general BSYI should be used to inhibit further REQO's. REQO must be dropped before the trailing edge of BSYI. The timing diagram for these specifications is shown in Fig. 11. and the times are specified in Table III.

Experiment Buffer

As a typical experiment buffer, we show the module developed to synchronize the output from two Wilkinson-ramp ADCs. These ADCs are used to encode the (x,y) coordinates of an event in a two-dimensional position-sensitive x-ray or neutron counter. Because the conversion time in this type of ADC depends on the address converted, the time the x- and y-axis addresses become available may be hundreds of nanoseconds (or even several microseconds) apart. Further processing requires these data to be available simultaneously. The circuit shown in Fig. 12 performs this task. CAMAC control functions include RUN[A(0)F(26)] and HALT [A(0)F(24)]. A photograph of the completed module is shown in Fig. 13 (a).

Mapper

Once a pair of (x,y) coordinates have been latched and synchronized in the experiment buffer, it is necessary to map the data from a two-dimensional matrix notation to a linear vector in MOSTEK memory. In an earlier system, an 8X300 microcontroller was used to perform this task (Hendricks, 1978; Turner and Hendricks, 1979, 1980). The software cycle

time was about 5 μs. Such times are unacceptable if our goal of 1 MHz is to be achieved. In this section, we describe a hardware concept in which data rates over 2.5 MHz are readily achieved. The concept is outlined in Fig. 14. Here, 14 lines of input data (representing an area detector with a resolution of 128 x 128) are used as the address for a 14 line-in, 16 line-out data selector constructed from static RAMs. The information stored in RAM under CAMAC control consists of the 14-bit (or less) mapped address corresponding to the two 7-bit input addresses, an internally generated parity bit, and a transfer control bit. If the transfer control bit is 1, the event will generate a REQO; if the bit is 0, the event will be ignored. A LAM is generated if there is a parity error. Four additional data lines are passed without comment. A front panel switch is provided to prevent unauthorized access to the RAM. All control communication with the module is via the CAMAC dataway. The functions to which the module responds are given in Table IV. The module is built on a standard LASL model WW-3A wire-wrap card. A photograph of the completed module is shown in Fig. 13 (b). The propagation delay through the module, as determined from the leading edge of REQI to the leading edge of REQO, was measured to be 380 ns.

The mapper module described here presents several exciting features. First, even though it is the slowest module of those developed to date, the throughput is extremely high — 2.6 MHz. Second, non-rectangular coordinate transformations are easily handled. Thus, the device is ideal for use with the LASL cylindrical-coordinate area detector in use on the small-angle scattering beam line. It could also perform on the fly the rectangular to spherical coordinate transformations required for spherical drift chamber detectors now under development for crystallographic

research in the U.S. and Europe. The corrections for imperfect electronic transformation of coordinate systems could also be included.

Finally, with appropriate use of the transfer control bit considerable preprocessing of the input data is possible. Consider a crystallographic example in which an area detector is used to record many Bragg reflections simultaneously. The situation is depicted in Fig. 15. For initial crystal alignment, the mapper is loaded to map a 128 x 128 array to memory. Following preliminary graphic analysis the various reflections can be indexed. The mapper is then loaded to map all the intensity in a 4 x 4 box surrounding a given (hkℓ) reflection into a block of 16 locations in memory. The memory array is indexed to contain the data sequentially for each (hkℓ) reflection in ascending order. All data falling outside the boxes around the various reflections are discarded. More elegant schemes in which a second, larger box is used to surround each reflection in order to measure the local background can be envisioned. Clearly, the resolution of the box surrounding each reflection may be selected depending on the needs of the experiment.

It is important to note that, unlike the microprocessor mapping system developed earlier, the mapping time is a constant 380 ns, independent of the complexity of the coordinate transformation scheme in use. The only limitation is that only 14 bits can be mapped.

Time Correlation

LASL Time-of-Flight Clock

The neutron source at LASL is the Weapons Neutron Research (WNR) facility, which is a spallation source associated with the LAMPF linear protron accelerator (Russel et al., 1979). Instead of monochromatizing neutrons, the entire broad-band spectrum is used; the time of each

neutron event, measured from the time the proton pulse hits the spallation target, is linearly proportional to the neutron wavelength. Thus, measurement of time-of-flight (TOF) is of fundamental importance.

The Model 5 TOF Clock is a revision of previous LASL modules in order to accommodate the new data bus and handshaking signals required by the present data acquisition system. This unit allows simultaneous storage of up to 14 bits of routing information. To accommodate all these bits within the 24-bit CAMAC format, the amount of time information produced by the clock is fixed at 10 bits, or 1024 time channels. A compression scheme may be program selected to give adequate resolution at small wavelengths and yet to extend to long wavelengths: after each 256 channels, the channel width doubles. For instance, assuming a flight path (source to sample to detector) of 6.288 m, we may have a recording channel width of 0.001 Å up to $\lambda = 0.256$ Å, of 0.002 Å for $0.256 \text{ Å} < \lambda \leq 0.768$ Å, of 0.004 Å for $0.768 \text{ Å} < \lambda \leq 1.792$ Å, and of 0.008 Å for $1.792 \text{ Å} < \lambda \leq 3.840$ Å. In terms of time-of-flight, the resolution varies from 1.6 to 12.8 μs and the total counting time is 6.144 ms. The resolution may be set to any power-of-two value between 0.2 and 25.6 μs per channel, and the scale may be set to linear or to the compressed form described above. There are always 10 bits of time information.

A block diagram of the clock is shown in Fig. 16, and the CAMAC instruction set is given in Table V. A photograph of the completed module is shown in Fig. 13 (c). Basically, a scaler chain begins counting a free-running 20 MHz crystal oscillator when a "start" pulse arrives from the proton beam line monitor, and whenever an "Event" pulse occurs the contents of the scaler are latched on the fly into a fast

memory, along with the corresponding tag bits (e.g., from the Mapper output). A burst of up to 16 events can be stored with a dead time of only 40 ns per event (!). Whenever one or more events have been stored and BSYI is not set, a REQO is asserted and readout proceeds asynchronously via a front panel connector to the data bus. The time-average throughput is limited by the speed of the data bus and the response time of the MOSTEK memory to the adder/incrementer port. This has been tested at >1.2 MHz.

ORNL Time Correlator

At ORNL, we are developing an apparatus which will allow the recording of small-angle x-ray and neutron scattering data with a two-dimensional position-sensitive detector in synchronization with a periodic stress applied to the sample (Hendricks, 1979). A servo-controlled oil-driven piston is used to apply a time-varying load to a polymeric sample. The time dependence of the piston displacement is determined by a CAMAC programmable function generator (Turner et al., 1980). At various preprogrammed points in the function generator output, a logic level transition is given to indicate that the diffraction pattern should be recorded in a different memory array. We anticipate recording 32 to 128, 64 x 64 data arrays and repetitively switching between them during many (10^2–10^4) deformation cycles.

A time correlation module which will change the origin of the two-dimensional data array on command from our function generator is shown in Fig. 17. The CAMAC functions to which the module responds are given in Table VI and a photograph of the completed module is shown in Fig. 13 (d). There are two registers; a register to store the base address of data

acquisition array (TR) and a register to store the amount by which the base address is incremented at each command (TIR). On receipt of a C = M pulse, the contents of TIR are added to TR. The current value of TR is added to all input from the mapper until the next C = M signal again increments TR. At the end of a cycle, the function generator issues PC = 0 which zeros TR and the process begins again. The propagation delay through the module, as determined from the leading edge of REQI to the leading edge of REQO, was measured to be 170 ns.

Data Bus Display

The data manipulations provided by the various modules described in previous sections can often become quite complex and it is desirable to have some method for performing diagnostic checks on their correct operation. We have developed a buffered data bus display (DBD) module for this purpose. This module consists of a 24-bit input latch, a CAMAC-controlled steering register which allows the input source to be either an upstream data module or the CAMAC dataway, a 6-digit hexadecimal LED display, a CAMAC-accessible output register, and manual/automatic control circuitry. The output can be routed either to the next downstream module on the data bus, or to the CAMAC dataway. A LAM may be generated for each event. A block diagram is shown in Fig. 18 while the CAMAC control functions are given in Table VII. A photograph of the completed module is shown in Fig. 13 (e).

The Control Register (CR) determines the mode of operation of the module as indicated in Table VIII. If the dataway is selected, the input register may be filled with a 24-bit CAMAC write A(0)F(16). The contents of the latch may be read with A(0)F(0) regardless of the source data. The data in the register may be automatically passed to the next

downstream module as soon as $\overline{\text{BSYI}}$ is asserted, or may be held until manually released by a front panel pushbutton (ADV SW). The contents of the register are displayed by six TIL311 hexadecimal LED displays on the front panel. For operation in Mode 1, the display is held for a period of 0.1×2^n s where n is loaded into the display control register. During this display hold time, all data received by the module are passed without display. The propagation delay for the module operating in Mode 1, as determined from the leading edge of *REQI* to the leading edge of REQO, was measured to be <20 ns.

In normal operation two data bus displays are used — one immediately follows the ADC buffer and one is used as the last unit before the adder/incrementer port. In this manner it is possible to completely test the correct operation of all address manipulation modules. A package of FORTRAN diagnostic subroutines is currently being developed which loads the mapper and time correlator (and any other modules in the system) with known memory mapping schemes and then transmits known random bit patterns through the first data bus display via the CAMAC dataway. The resultant mapped pattern is read back from the second display onto the CAMAC dataway and is compared with the expected transformed pattern. When not in use for such diagnostics, the data bus displays are left in the system (with no change in cabling). With the units programmed to operate in Mode 1, the experimental data are passed through without modification and with only a 40 ns total delay (20 ns per DBD). Events are periodically sampled and displayed, as described above, to give a visual indication of the correct operation of the data bus.

It should be noted that if the data rates are relatively low and if the memory requirements are not too large, a very simple and inexpensive

CAMAC data acquisition system can be built by using any of the address manipulation modules described above and a data bus display module set to operate in Mode 2 with the LAM enabled. The data acquisition array would then reside in the control computer memory (rather than the MOSTEK memory, which is not used in this scheme) and the mapped memory location would be incremented by a LAM-driven software interrupt routine.

Other Modules

It is clear from the previous discussion that the various modules on the experimental bus may be daisy-chained together in any desired manner. Further modules can be developed as experimental needs dictate. Two potentially valuable modules for use with position-sensitive detectors are a direct time digitizer (TD) which would replace the time-to-amplitude converters (TAC) and analog-to-digital converters (ADC) currently in use, and a first-in/first-out (FIFO) buffer which could be placed immediately in front of the memory increment stage.

The signals generated in PSDs are time-related. It thus appears that direct time digitization is philosophically more appealing than the present TAC-ADC approach. However, at high data rates, there are some difficult gating problems associated with direct time digitization. Solutions to these problems are now available. Lynch (1980) and LeCroy (1980) have developed fast CAMAC TDs specifically for use with position-sensitive detectors. An interface for the LeCroy TDs to the data bus described in this paper is now being developed.

In experiments where the data rates are quite high and where considerable activity on the MOSTEK bus is also anticipated, the dead-time of the data bus could be compensated if a FIFO buffer were placed as the last module on the data bus. In addition, such a FIFO would allow for

bursts of data from the detector system, provided the mean data rate did not exceed the incrementer port throughput of 2.5 MHz. The actual burst rate which could be accommodated depends on the specific modules installed in the system, but with the modules developed to date could be from 2.6 to >10 MHz. Such a module is planned for the system proposed here.

FUTURE EXPANSION

The data acquisition system described here is very flexible because of its modular design and has considerable potential for future expansion. The most obvious possibilities are (1) direct minicomputer — I/O port interfaces, (2) multiprocessor — I/O port interfaces, and (3) dedicated custom microprocessor data processing. These options are outlined in Fig. 1.

It is clear that moving large blocks of data between the control computer and the memory system via the CAMAC dataway is really quite inefficient. One obvious solution is the construction of a custom I/O device for the computer which interfaces directly to the I/O port bus. Such a module is easily constructed for the ModComp computers in use at both LASL and ORNL. It will be implemented on a 4805 general purpose controller and will contain much of the logic developed for the fast-response crate controller (Seeger, 1976).

The memory system described here could be used in place of the high-speed swapping disc necessary for the time-sharing operating systems available on many minicomputers. For 16-bit minicomputers, the 24-bit memory might appear unnecessary. However, with readily available integrated circuits it is reasonable to use six bits for error checking and correcting (ECC) algorithms. The remaining two bits can be used for

protection codes and/or end-of-file bits needed to simulate the operation of many standard disc drives. We propose to implement this hardware in such a manner that various blocks of memory within the same system can be used as either 24-bit data acquisition arrays or as 16-bit ECC memory storage arrays.

The general nature of the four-port memory interface described in the Memory System section allows the memory system to be interfaced to a completely independent computer I/O bus. Thus, the memory can be shared by multiple computers which need not be of either the same manufacture or the same word size. An obvious possibility would be the implementation of an interface to a widely accepted microcomputer I/O bus such as the Intel MULTIBUS as described by Barthmaier (1980) or the Motorola VERSABUS as descibed by Kister and Robinson (1980). Such an interface would provide inexpensive distributed data processing for a variety of dedicated functions (such as for video displays etc.).

Finally, one of the I/O ports could be used to implement specialized custom microcomputer data processors. An obvious and very attractive possibility is the Bulk Memory Processor (BMP) now under development at LASL by McMillan et al. (1979), which is totally compatible with the memory system described here because we have implemented fully their interface. The BMP might convert this system into an extremely powerful fast integer microcoded data processing unit. With its 80-bit instruction word and multiple parallel processing capability, the BMP is considerably faster than almost any minicomputer with which this data acquisition system is likely to be operated, and is ideally suited to complex multi-dimensional fast Fourier tranforms.

ACKNOWLEDGMENTS

The authors are indebted to Drs. R. O. Nelson, R. N. Silver, C. J. Sparks, P. J. Vergamini, and H. L. Yakel for their continued encouragement and interest. Without their support, this project could not have proceeded. We are also grateful to H. A. Bishop, H. E. Bowen, Ms. M. A. Brock, B. A. Denning, Jr., M. E. Fuehrer, N. B. Hickman, R. W. Jones, and J. E. McCarter for their care and skill in drawing the circuit diagrams and wirewrapping and assembling the prototype modules. This project was conceived and the prototypes were designed, built, and debugged in a total of five months. The success of this project would not have been possible without the interest and enthusiasm of these individuals.

REFERENCES

Barthmaier, J. P., *Computer Design* 19(2): 137 (1980); *Intel Multibus Specification*, Manual 9800683, Intel Corp., Santa Clara, CA.

Hendricks, R. W., *Trans. Am. Cryst.* 12: 103 (1976).

Hendricks, R. W., *J. Appl. Cryst.* 11: 15–30 (1978).

Hendricks, R. W., "Design and Construction of a Specimen Holder and Control System for Dynamic X-Ray and Neutron Small-Angle Scattering Studies of the Plastic Deformation of Polymers," proposal accepted for funding by the National Science Foundation (February 1979).

Hendricks, R. W., Seeger, P. A., Scheer, J. W., and Suehiro, S., The LASL-ORNL Fast Digital Data Acquisition System, Oak Ridge National Laboratory Report ORNL/TM-7325 (Rev. 1), 1981.

IEEE, 345 E. 47th Street, New York, NY 10017, *CAMAC Instrumentation and Interface Standards*.

Kister, J., and Robinson, I., *Electronics* 53(3): 81 (1980).

Kitchens, T. A., *ANL-LASL Workshop on Advanced Neutron Detection Systems*, LASL-7890-C (April 1979).

LeCroy Research Systems Corp., (1980), Specification Sheets for Model 4201 Fast Time Encoder.

Lynch, F. J., *IEEE Tr. Nucl. Sci.* NS27: 327 (1980).

Machen, D. R., *Design and Operation of the LAMPF Auxiliary Controller*, LASL-7683-MS (February 1979).

McMillan, D. E., Nelson, R. O., Poore, R. V., Ross, J. J. and Sunier, J. W. "An 8-Megaword Semiconductor Bulk Memory Bit-Slice Microprocessor System," *IEEE Conference on Data Acquisition Systems,* Santa Fe, NM (May 1979).

MOSTEK Memory Systems, 1215 W. Crosby Road, Carrollton, TX 75006, *MOSTEK MK8000 Technical Manual* (1979).

Russel, G. J., Lisowski, P. W., and King, N.S.P., *Int. Conf. on Neutron Physics and Nuclear Data for Reactors and Other Applied Purposes,* UKAERE, Harwell, England, September 25–29, 1978.

Seeger, P. A., *A Fast-Response CAMAC Crate Controller*, Los Alamos Scientific Laboratory Report LA-6605-M (December 1976).

Turner, G. W., and Hendricks, R. W., An Intelligent CAMAC I/O Module Based on the Signetics 8X300 Microcontroller, Oak Ridge National Laboratory Report ORNL/TM-6922 (September 1979).

Turner, G. W., and Hendricks, R. W., *Nucl. Instr. Methods* 169, 373–380 (1980).

Turner, G. W., Suehiro, S., and Hendricks, R. W., *A Programmable CAMAC Function Generator*, Oak Ridge National Laboratory Report ORNL/TM-7363

Table I. Four Port Timing Specifications

Symbol	Function	Minimum Time (ns)
t_s	Signal or Data Set-Up	20
t_p	BSYN Propagation	30
t_h	Signal or Data Hold	60
t_w	WRDYN Pulse Width	30

Table II. CAMAC Function Codes for MOSTEK I/O Port Interface.

Function	Operation
A(0)F(0)	Read data buffer.
A(0)F(4)	Read data buffer; AR = AR + AIR; TC = TC − 1; initiate memory read request. IF TC < 0, AR overflow, or BSY then Q = 0. Otherwise, Q = 1. LAM generated if enabled.
A(0)F(16)	Write data buffer; initiate memory write request. If $\overline{\text{BSY}}$, Q = 1.
A(0)F(22)	Write data buffer; initiate memory write request; AR = AR + AIR; TC = TC − 1. If TC < 0, AR overflow, or BSY, then, Q = 0. Otherwise, Q = 1. LAM generated if enabled.
A(1)F(0)	Read address register (AR).
A(1)F(16)	Load AR.
A(1)F(20)	Load AR; initiate memory read request. If $\overline{\text{BSY}}$, Q = 1.
A(1)F(22)	Load AR; initiate memory read request; AR = AR + AIR; TC = TC − 1. If $\overline{\text{BSY}}$, Q = 1.
A(2)F(0)	Read transfer count register (TC).
A(2)F(16)	Load TC.
A(3)F(0)	Read address increment register (AIR).
A(3)F(16)	Load AIR.
A(0)F(8)	Test LAM; If present, Q = 1 (LAM generated on TC < 0 or AR overflow).
A(0)F(9)	Initialize; AR = 0, AIR = 1, TC = 0.
A(0)F(10)	Clear LAM.
A(0)F(24)	Disable LAM.
A(0)F(26)	Enable LAM.

Table III. Data Bus Timing Specifications

Symbol	Function	Minimum Time (ns)
t_s	Data Set-Up	20
t_w	Request Pulse Width	30
t_h	Data Hold	40
t_r	Request Deadtime	30

Table IV. Mapper Module CAMAC Function Codes

Function	Operation
A(0)F(0)	Read current address register. Q = 1.
A(0)F(8)	Test for LAM present. Returns Q = 1 if LAM is enabled and a parity error has occurred; Q = 0 otherwise.
A(0)F(10)	Clear LAM. Q-response as for A(0)F(8).
A(0)F(16)	Write the 14-bit address register from lines W1 thru W14 of the CAMAC dataway. Q = 1.
A(0)F(24)	Disable LAM. Q = 1.
A(0)F(26)	Enable LAM. Q = 1. A LAM will be generated if a parity error occurs in response to a data bus input.
A(1)F(0)	Read from memory at location in address register, to dataway lines R1 thru R16. Returns Q = 1 if no parity error, Q = 0 if parity error. Increments address register after reading.
A(1)F(16)	Write into memory at location in address register, from dataway lines W1 thru W15. (The lowest 14 bits will be the DATA OUT output when this location is addressed from the DATA IN input. If bit 15 is on, a REQO strobe will also be generated in response to REQI.) Parity is generated and stored as bit 16. The address register is incremented after writing. Q = 1.
Z	Clears LAM. Does not affect LAM Enable, or have any effect on contents of memory.

Table V. T-O-F Clock (Model 5) CAMAC Function Codes

Function	Operation
A(0)F(10)	Stop the clock; reset time to zero.
A(0)F(24)	Disable the clock. Finish cycle if on, but do not allow any more Start's.
A(0)F(25)	Start the clock if enabled.
A(0)F(26)	Enable the clock to recognize Start pulses.
A(0)F(27)	Test if clock enabled; Q = 1 if enabled.
A(1)F(0)	Read the Resolution Control Register to lines R1-R4.
A(1)F(8)	Test LAM; Q = 1 if LAM is enabled and FIFO is full.
A(1)F(10)	(Re)initialize FIFO buffer pointers; discard any data.
A(1)F(16)	Write the Resolution Control Register from lines W1-W4. If W4 is on, scale will be linear, and if off, scale will be compressed. If the binary number in W1-W3 is n, $0 \leq n \leq 7$, then the resolution is $0.2\ \mu s * 2^n$.
A(1)F(24)	Disable the LAM.
A(1)F(25)	Generate an "Event"; store current time in FIFO if clock is running.
A(1)F(26)	Enable the LAM for FIFO full.
A(1)F(27)	Test if LAM enabled: Q = 1 if enabled.
Z	Disable Clock and LAM; empty FIFO; clear Resolution Control Register.
I	Inhibit Starts.

Table VI. Time Correlator CAMAC Function Codes

Function	Operation
A(0)F(0)	Read TR (Time Axis Register)
A(1)F(0)	Read TIR [Time Axis Increment (= Block Size) Register]
A(0)F(9)	Clear TR and TIR
A(0)F(16)	Load TR
A(1)F(16)	Load TIR
A(0)F(25)	TR = TR + TIR
A(0)F(26)	Enable Data Acquisition
A(0)F(24)	Disable Data Acquisition
A(1)F(26)	Enable Data Acquisition Synchronized with Next [C = M]
A(1)F(24)	Disable Data Acquisition Synchronized with Next [C = M]
A(2)F(26)	Enable External Controls (PC = 0; C = M)
A(2)F(24)	Disable External Controls (PC = 0; C = M)
A(3)F(8)	Test LAM
A(3)F(10)	Clear LAM
A(3)F(26)	Enable LAM
A(3)F(24)	Disable LAM

Table VII. Data Bus Display CAMAC Function Codes

Function		Operation
A(0)F(0)		Read DATA latch
A(0)F(4)	Mode 3:	Read DATA latch and enable REQI. Generate LAM on next data latch
	Mode 4:	Read DATA latch and enable REQI. Generate LAM on next data latch. Generate REQO.
A(0)F(16)		Load DATA latch
A(0)F(20)		Load DATA Latch; if $\overline{\text{BSYI}}$, Q = 1 and REQO is generated; if BSYI, Q = 0 and no REQO is generated
A(1)F(0)		Read CONTROL register
A(1)F(16)		Load CONTROL register
A(2)F(0)		Read DISPLAY CONTROL register
A(2)F(16)		Load DISPLAY CONTROL register. Display hold time is 0.1×2^n s where n is number in register.
A(0)F(25)	Mode 2:	If $\overline{\text{BSYI}}$, generate REQO; Q = 1. If BSYI, do not generate REQO; Q = 0.
	Mode 3:	Enable REQI. Generate LAM on next data latch.
	Mode 4:	Enable REQI. Generate LAM on next data latch. Generate REQO.
A(0)F(8)		Test LAM; Q = 1 if present.
A(0)F(10)		Clear LAM
A(0)F(24)		Disable LAM
A(0)F(26)		Enable LAM
Z, C		Initiallize module; clear registers and data latch.
I		No response.

Table VIII. Data Bus Display Control Register Modes

Mode	Control Reg. (DO/DI/ Manual)	Manual LED	Who Generates REQO	Who Generates BSYO	DO	DI	Appropriate CAMAC Functions	Comments
1	X X 0	Off	REQI	BSYI	DAISY	DAISY	A(0)F(0)	Display hold time is controlled.
2	0 1 1	On	CAMAC	kept BSY	DAISY	CAMAC	A(0)F(16) A(0)F(20) A(0)F(25)	ADV does the same thing as A(0)F(25).
3	1 0 1	On	None	kept BSY	CAMAC	DAISY	A(0)F(0) A(0)F(4) A(0)F(25)	ADV gets next data and generates LAM when enabled.
4	0 0 1	On	ADV SW or CAMAC	BSYI	DAISY (CAMAC)	DAISY	A(0)F(0) A(0)F(4) A(0)F(25)	Ignores data until ADV is pushed. When pushed, latches next data, issues REQO and generates LAM when enabled.

1 1 1 — Illegal

LIST OF FIGURES

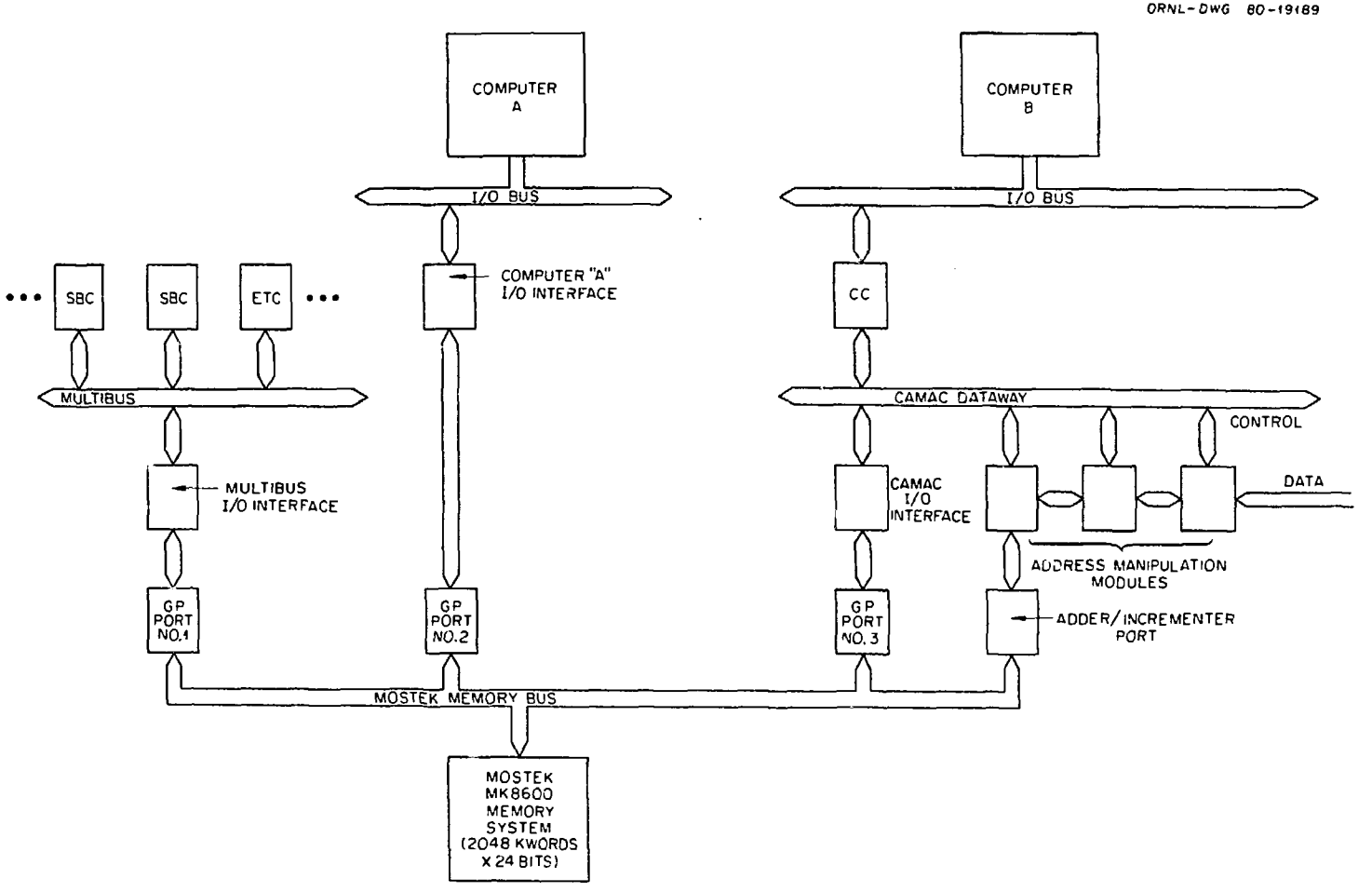
ORNL-DWG 80-19189
COMPUTER A
COMPUTER B
I/O BUS
I/O BUS
COMPUTER "A" I/O INTERFACE
CC
SBC
SBC
ETC
MULTIBUS
CAMAC DATAWAY
CONTROL
DATA
MULTIBUS I/O INTERFACE
CAMAC I/O INTERFACE
ADDRESS MANIPULATION MODULES
GP PORT NO.1
GP PORT NO.2
GP PORT NO.3
ADDER/INCREMENTER PORT
MOSTEK MEMORY BUS
MOSTEK MK8600 MEMORY SYSTEM (2048 KWORDS X 24 BITS)

Fig. 1

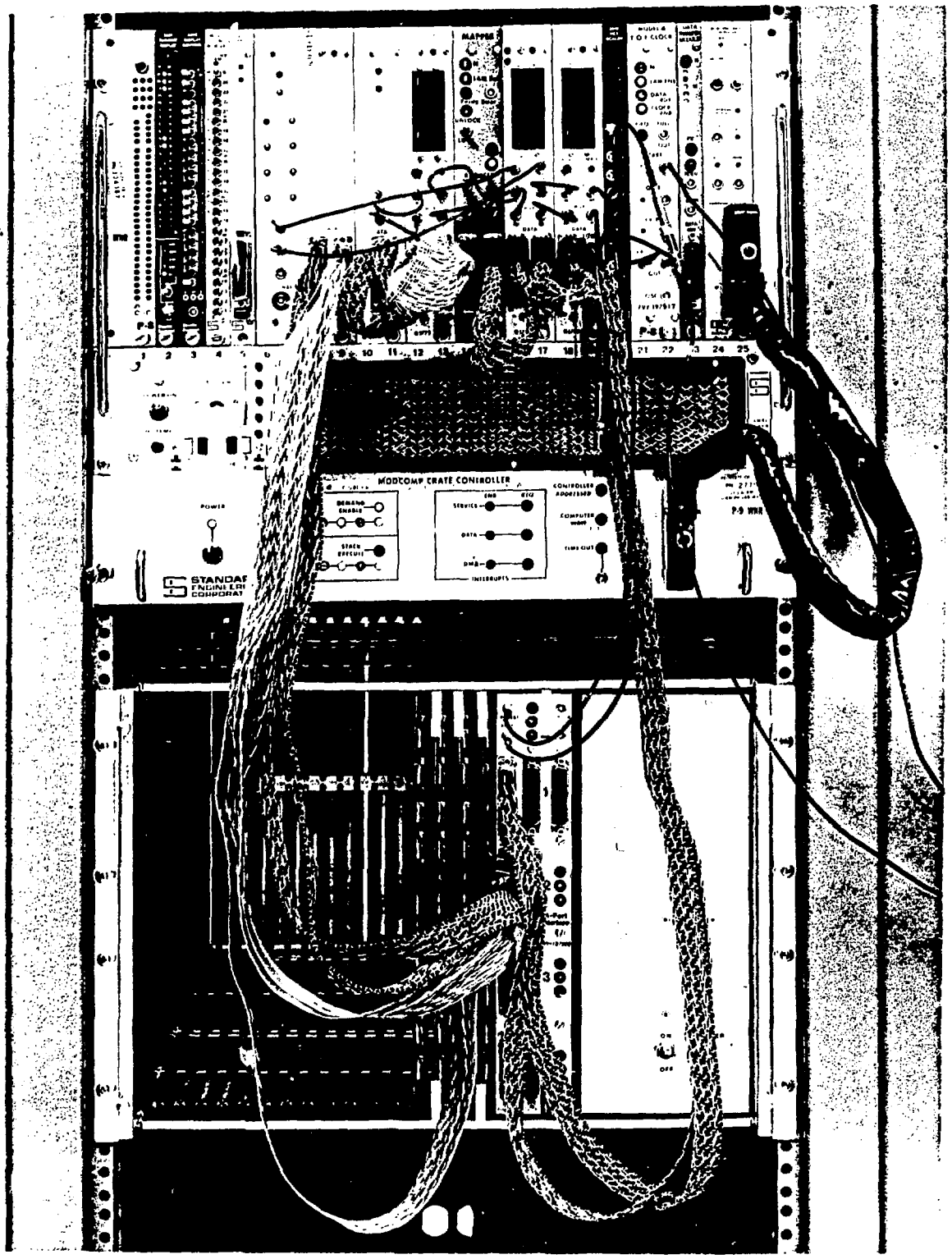

Fig. 2

ORNL-DWG 80-9529

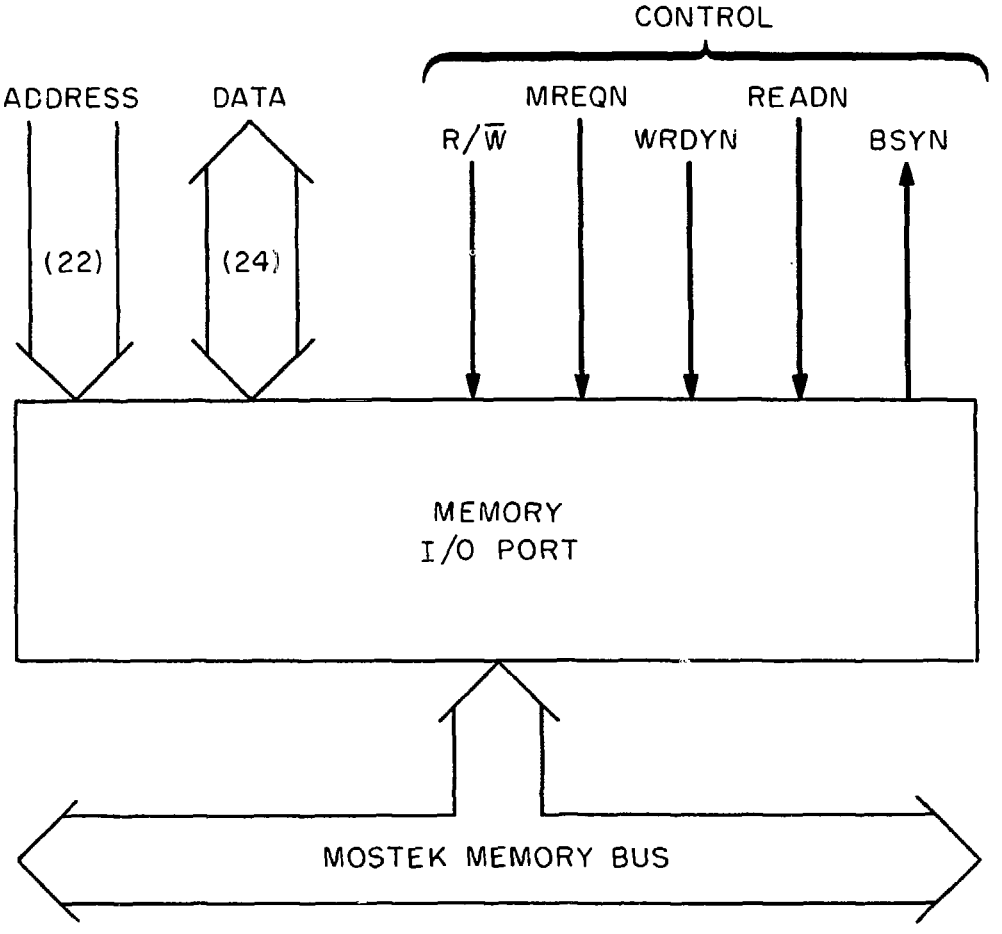

Fig. 3

ORNL-DWG 80-9530

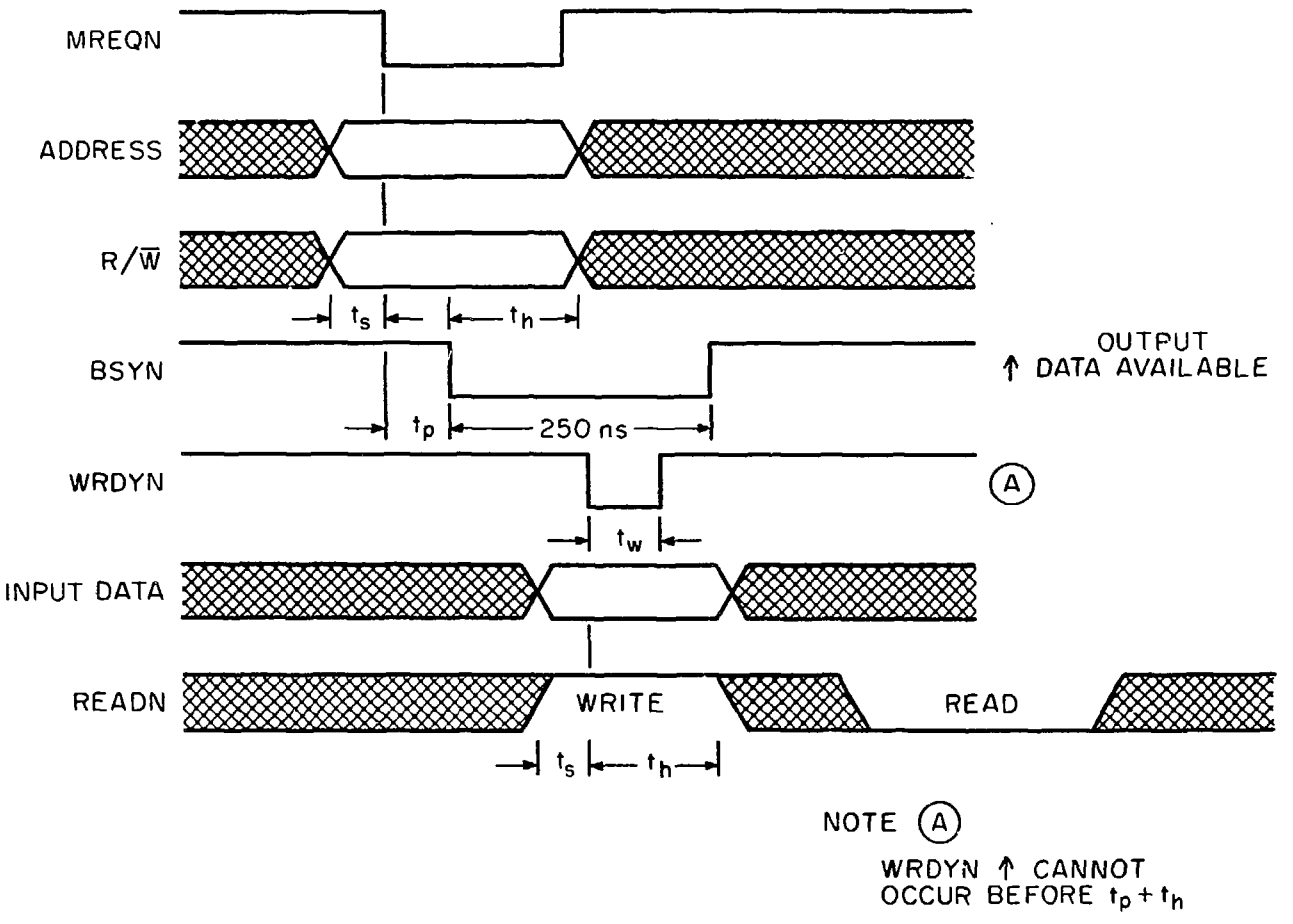

Fig. 4

ORNL-DWG 80-9531

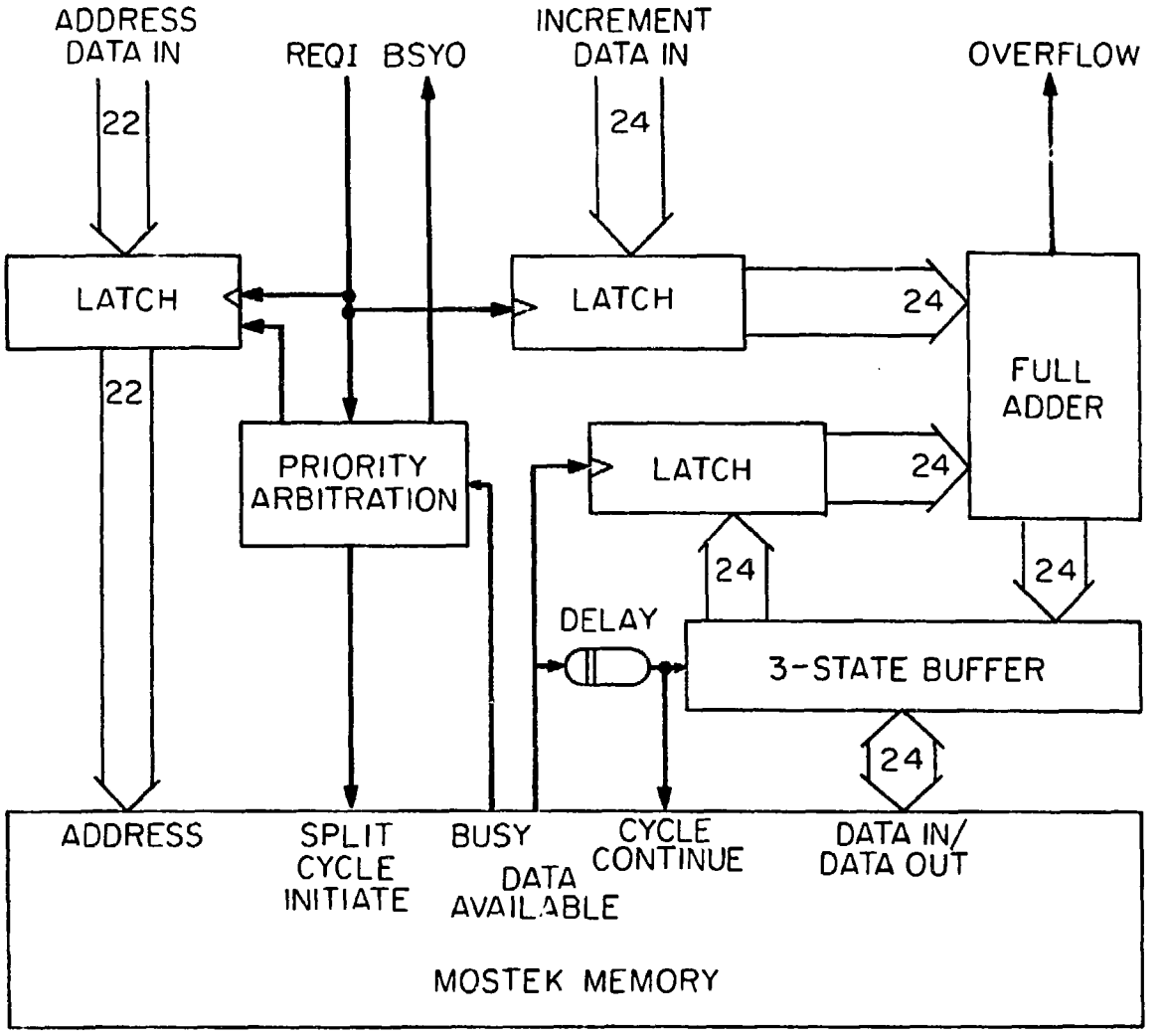

Fig. 5

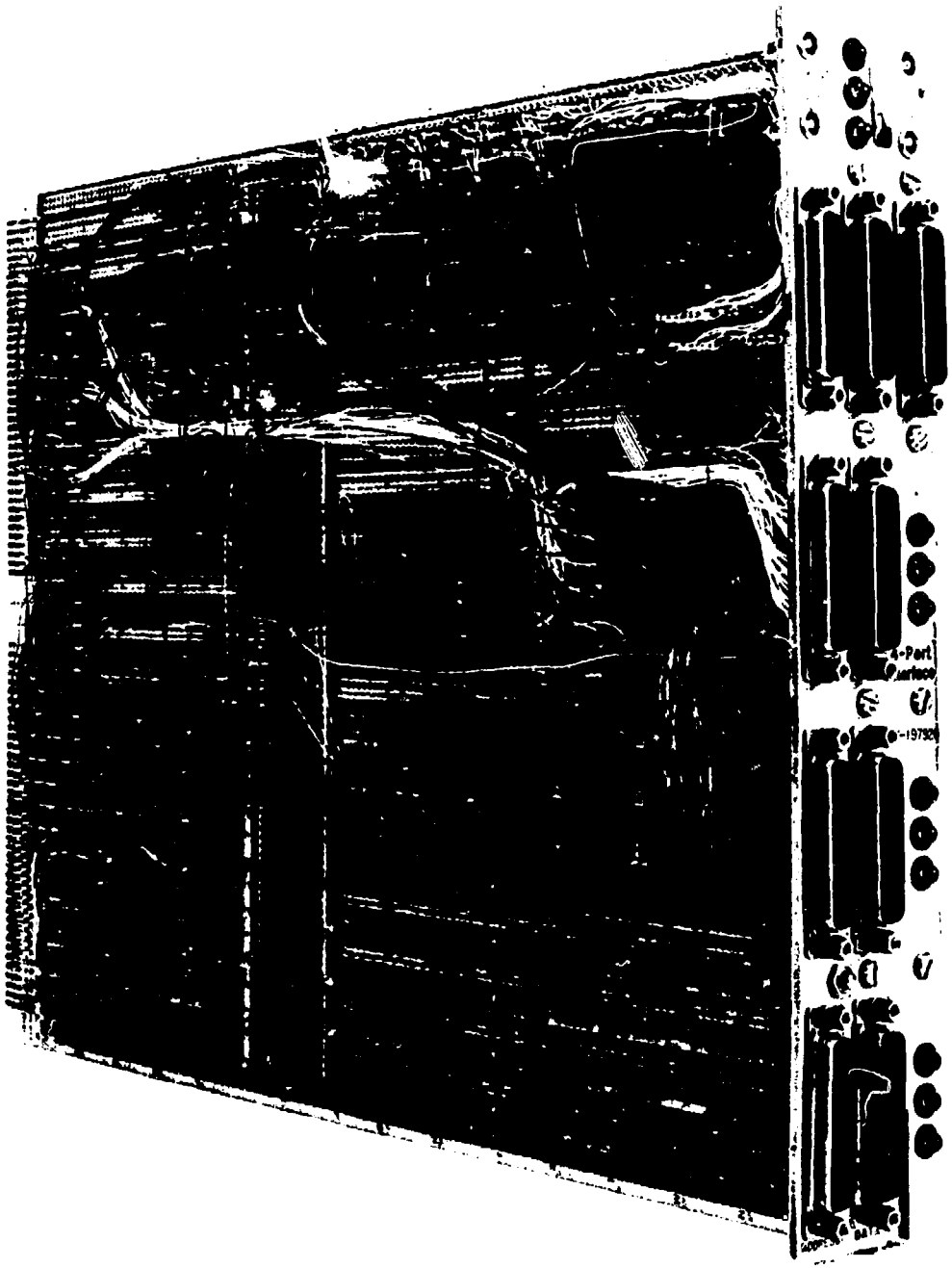

Fig. 6(a)

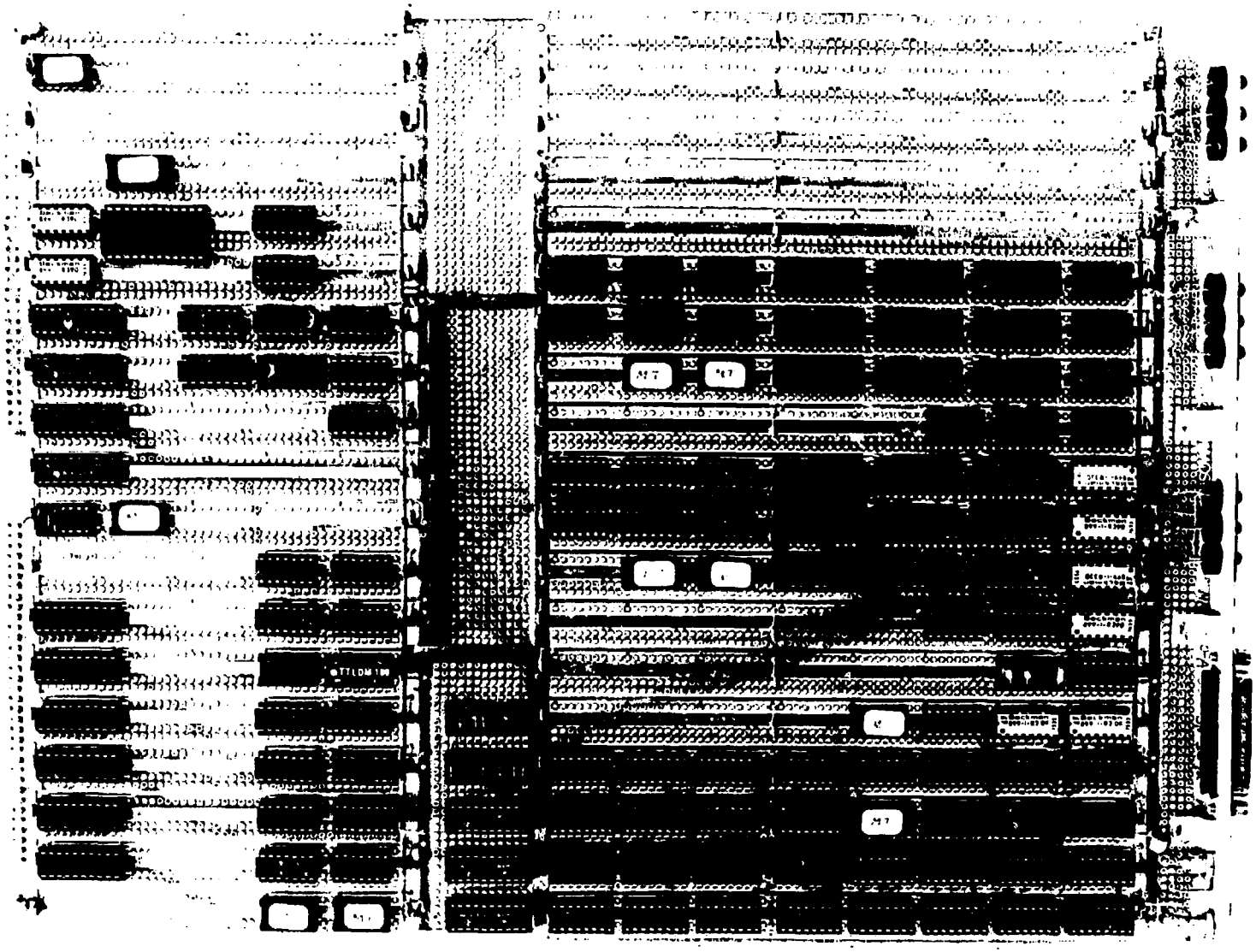

Fig. 6(b)

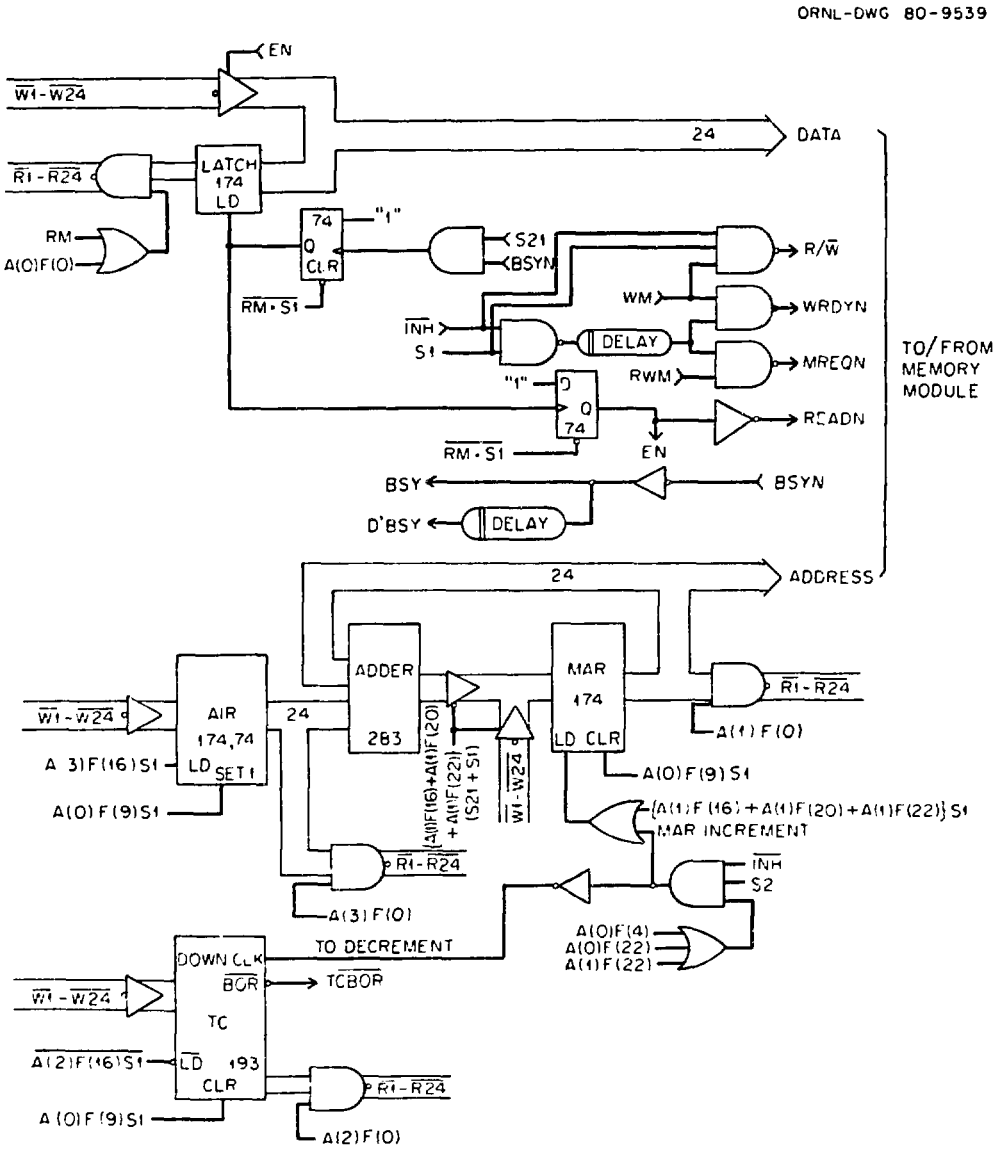
ORNL-DWG 80-9539
EN
W1-W24
24
DATA
R1-R24
LATCH
174
LD
RM
A(0)F(0)
74
Q
CLR
"1"
S21
BSYN
R/W
WM
WRDYN
INH
S1
DELAY
MREQN
RWM
RDADN
RM·S1
EN
TO/FROM
MEMORY
MODULE
BSY
BSYN
D'BSY
DELAY
24
ADDRESS
ADDER
283
MAR
174
LD CLR
AIR
174,74
LD SET
A 3)F(16)S1
A(0)F(9)S1
24
A(1)F(20)+A(1)F(22)
(S21+S1)
W1-W24
R1-R24
A(1)F(0)
A(0)F(9)S1
{A(1)F(16)+A(1)F(20)+A(1)F(22)}S1
MAR INCREMENT
R1-R24
A(3)F(0)
INH
S2
TO DECREMENT
A(0)F(4)
A(0)F(22)
A(1)F(22)
DOWN CLK
BOR
TCBOR
TC
W1-W24
A(2)F(16)S1
LD
193
CLR
R1-R24
A(0)F(9)S1
A(2)F(0)

Fig. 7(a)

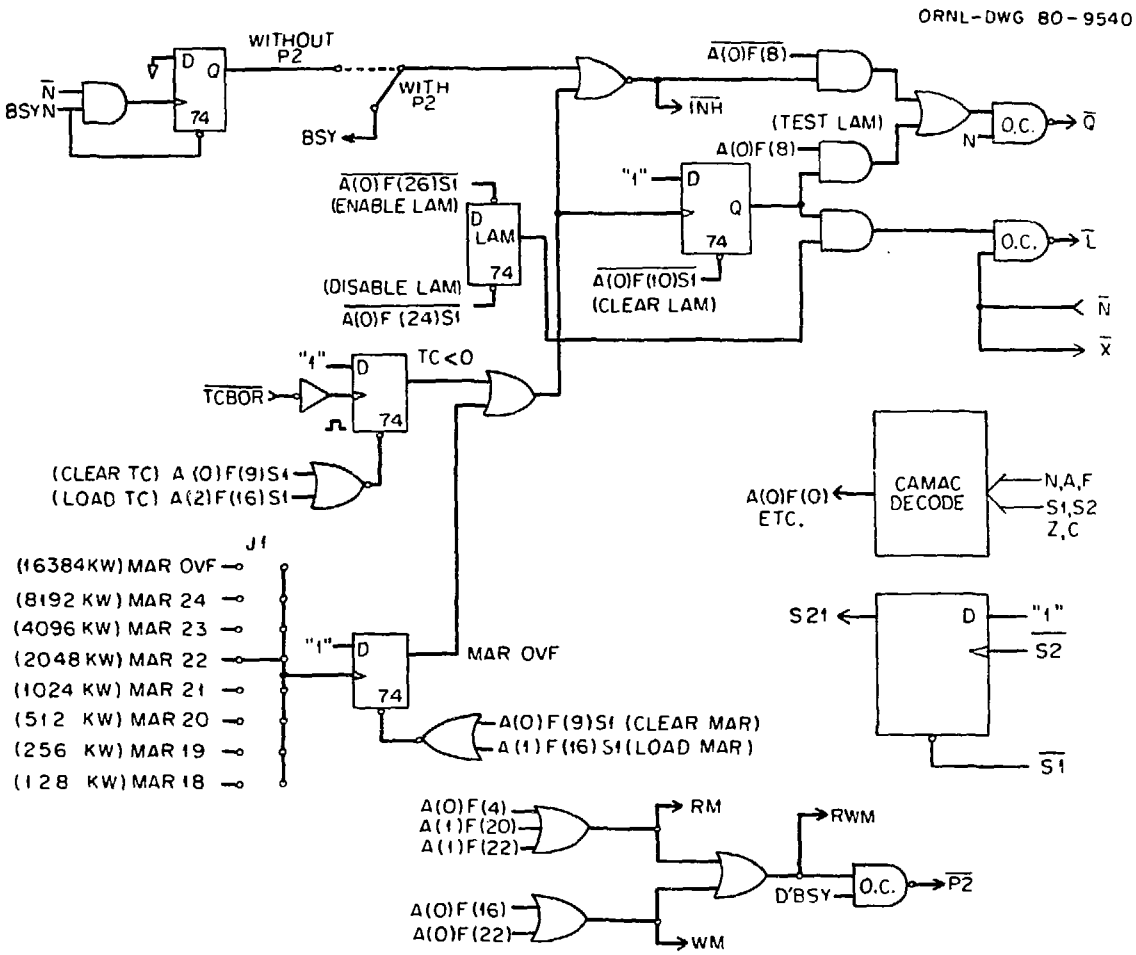
ORNL-DWG 80-9540
WITHOUT P2
WITH P2
BSY
N BSYN
D Q 74
A(0)F(8)
INH
(TEST LAM)
A(0)F(8)
N
O.C.
Q
"1"
D Q 74
A(0)F(26)S1 (ENABLE LAM)
D LAM 74
(DISABLE LAM) A(0)F(24)S1
A(0)F(10)S1 (CLEAR LAM)
O.C.
L
N
X
"1"
D 74
TCBOR
TC<0
(CLEAR TC) A(0)F(9)S1
(LOAD TC) A(2)F(16)S1
A(0)F(0) ETC.
CAMAC DECODE
N,A,F
S1,S2
Z,C
J1
(16384KW) MAR OVF
(8192 KW) MAR 24
(4096 KW) MAR 23
(2048 KW) MAR 22
(1024 KW) MAR 21
(512 KW) MAR 20
(256 KW) MAR 19
(128 KW) MAR 18
"1"
D 74
MAR OVF
A(0)F(9)S1 (CLEAR MAR)
A(1)F(16)S1 (LOAD MAR)
S21
D
"1"
S2
S1
A(0)F(4)
A(1)F(20)
A(1)F(22)
RM
RWM
D'BSY
O.C.
P2
A(0)F(16)
A(0)F(22)
WM

Fig. 7(b)

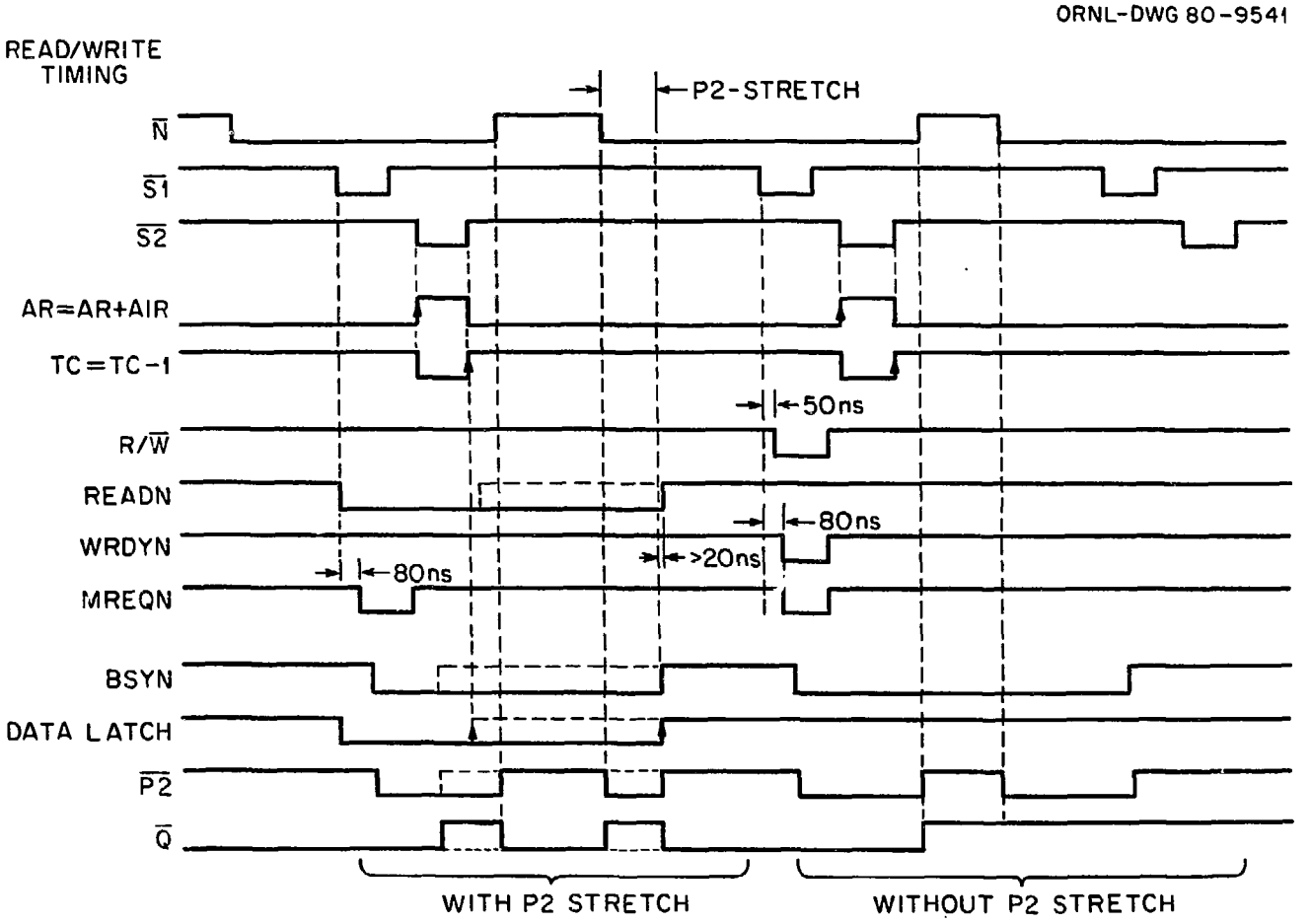
ORNL-DWG 80-9541
READ/WRITE TIMING
P2-STRETCH
N
S1
S2
AR=AR+AIR
TC=TC-1
50ns
R/W
READN
80ns
WRDYN
>20ns
80ns
MREQN
BSYN
DATA LATCH
P2
Q
WITH P2 STRETCH
WITHOUT P2 STRETCH

Fig. 9

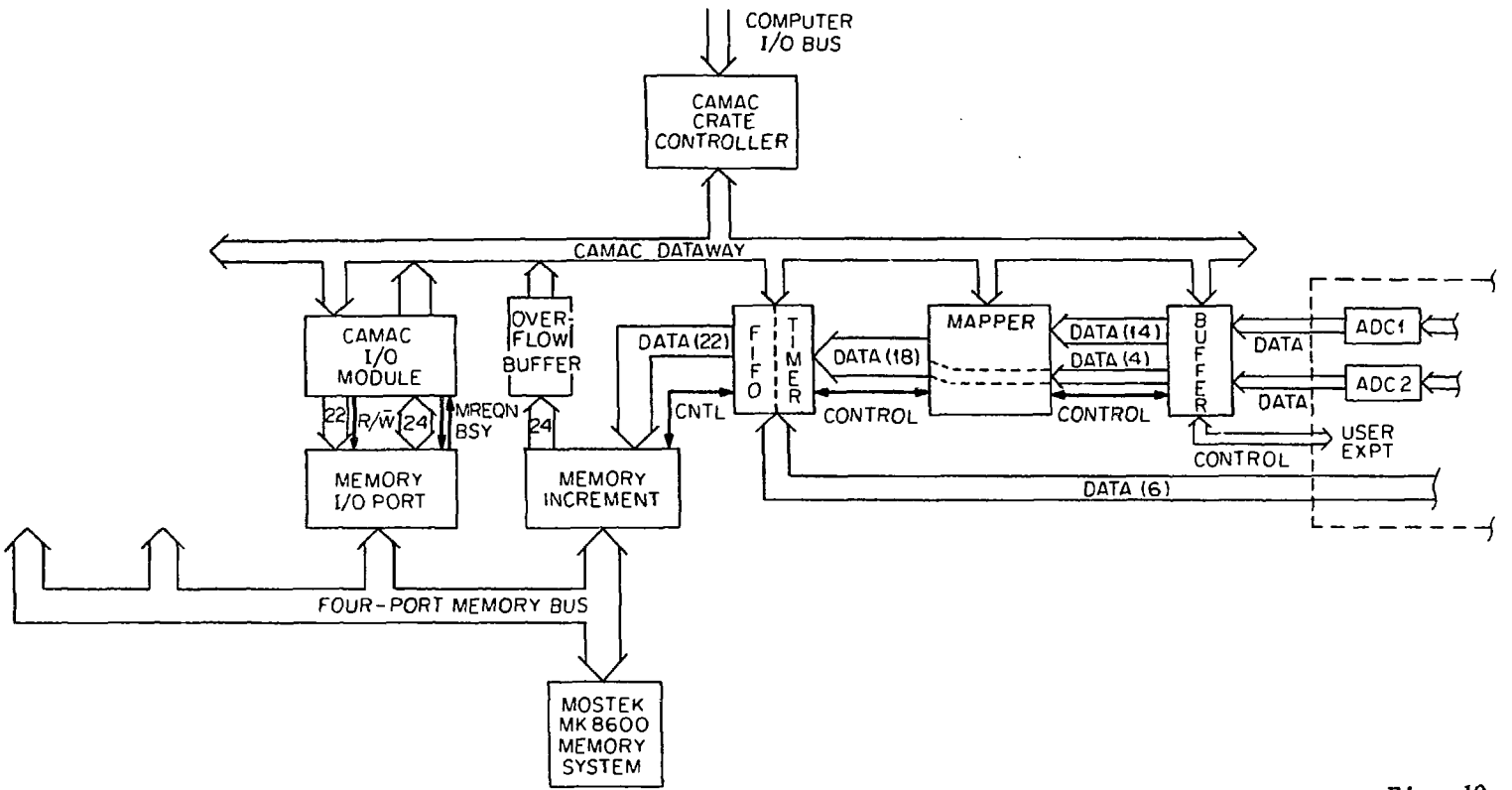
ORNL-DWG 80-9532
COMPUTER I/O BUS
CAMAC CRATE CONTROLLER
CAMAC DATAWAY
CAMAC I/O MODULE
OVER-FLOW BUFFER
22
R/W
24
MREQN
BSY
24
MEMORY I/O PORT
MEMORY INCREMENT
DATA (22)
CNTL
F I F O
T I M E R
DATA (18)
CONTROL
MAPPER
DATA (14)
DATA (4)
CONTROL
B U F F E R
DATA
DATA
ADC1
ADC 2
CONTROL
USER EXPT
DATA (6)
FOUR-PORT MEMORY BUS
MOSTEK MK 8600 MEMORY SYSTEM

Fig. 10

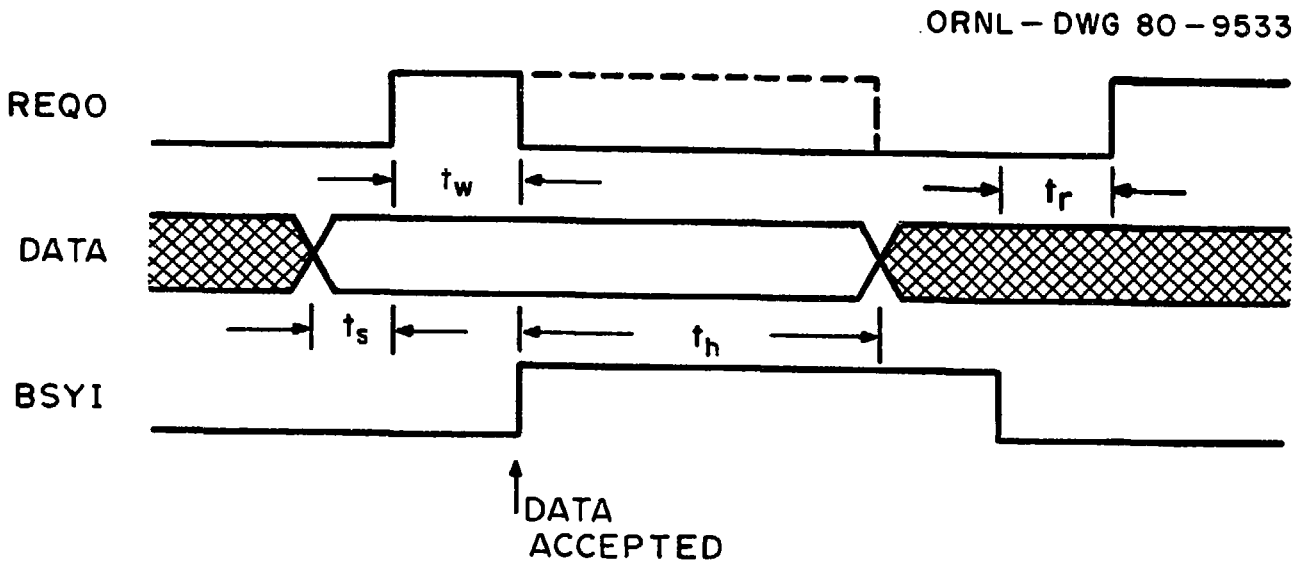

Fig. 11

ORNL-DWG 80-9779

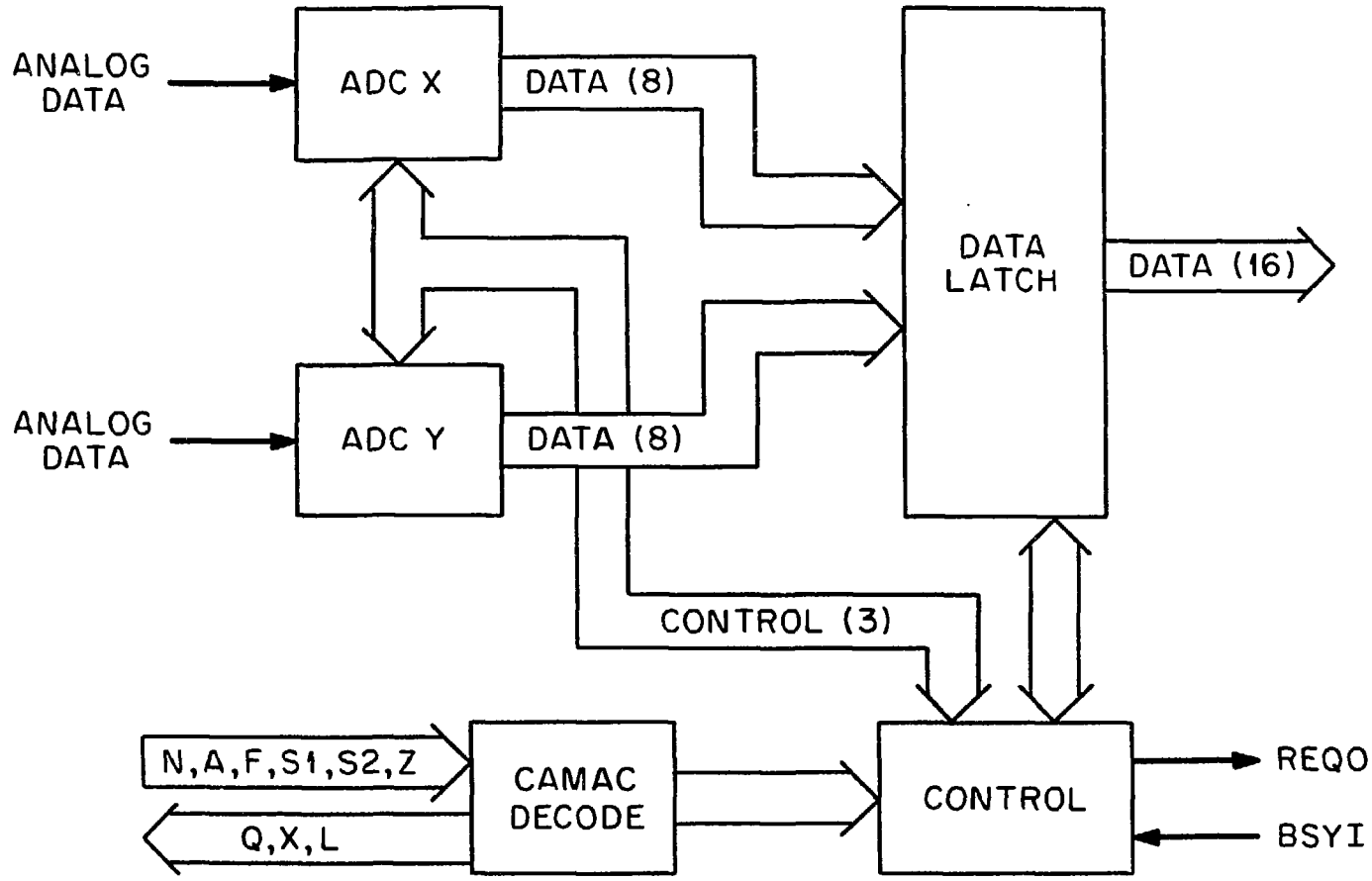

Fig. 12

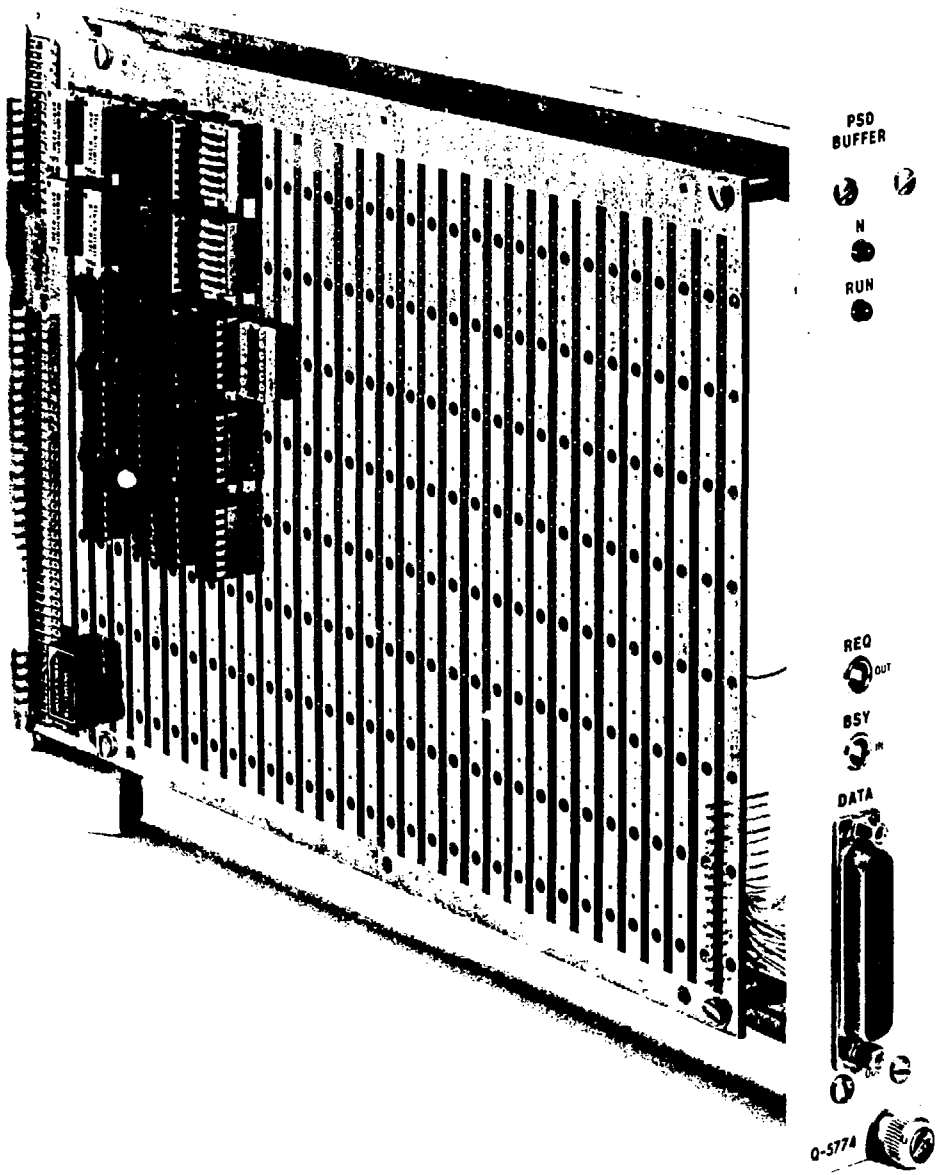

Fig. 13(a)

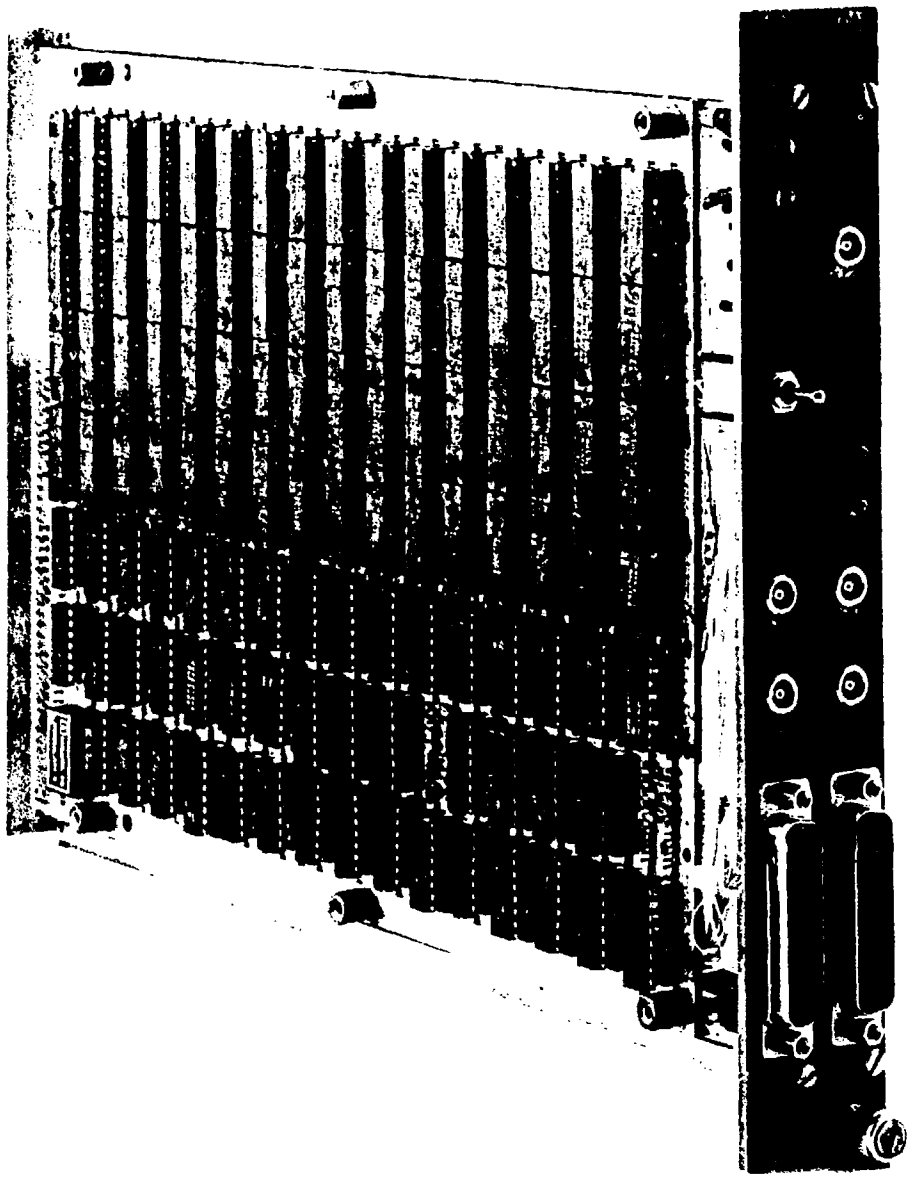

Fig. 13(b)

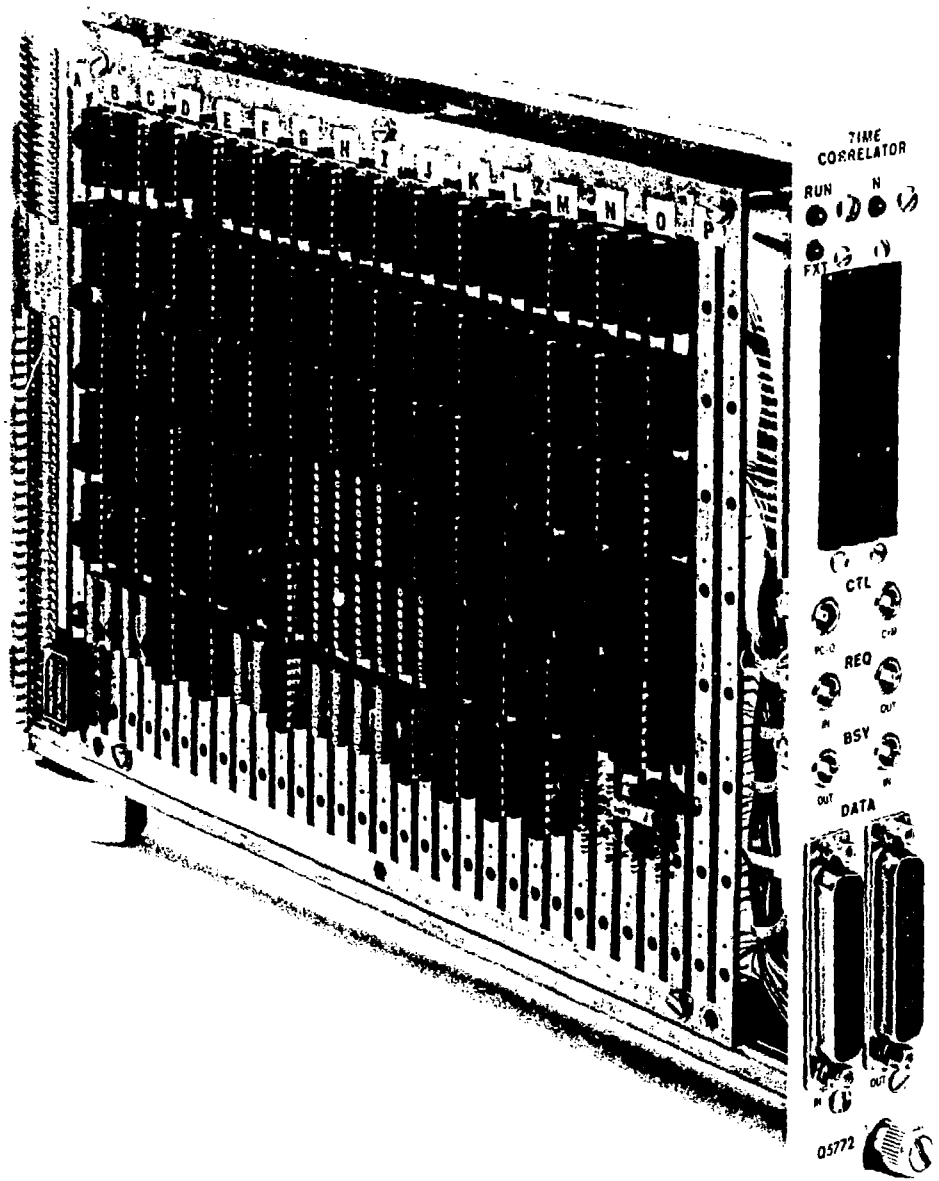
A B C D E F G H I J K L M N O P
TIME
CORRELATOR
RUN
N
FXT
CTL
REQ
BSY
DATA
05772

Fig. 13(d)

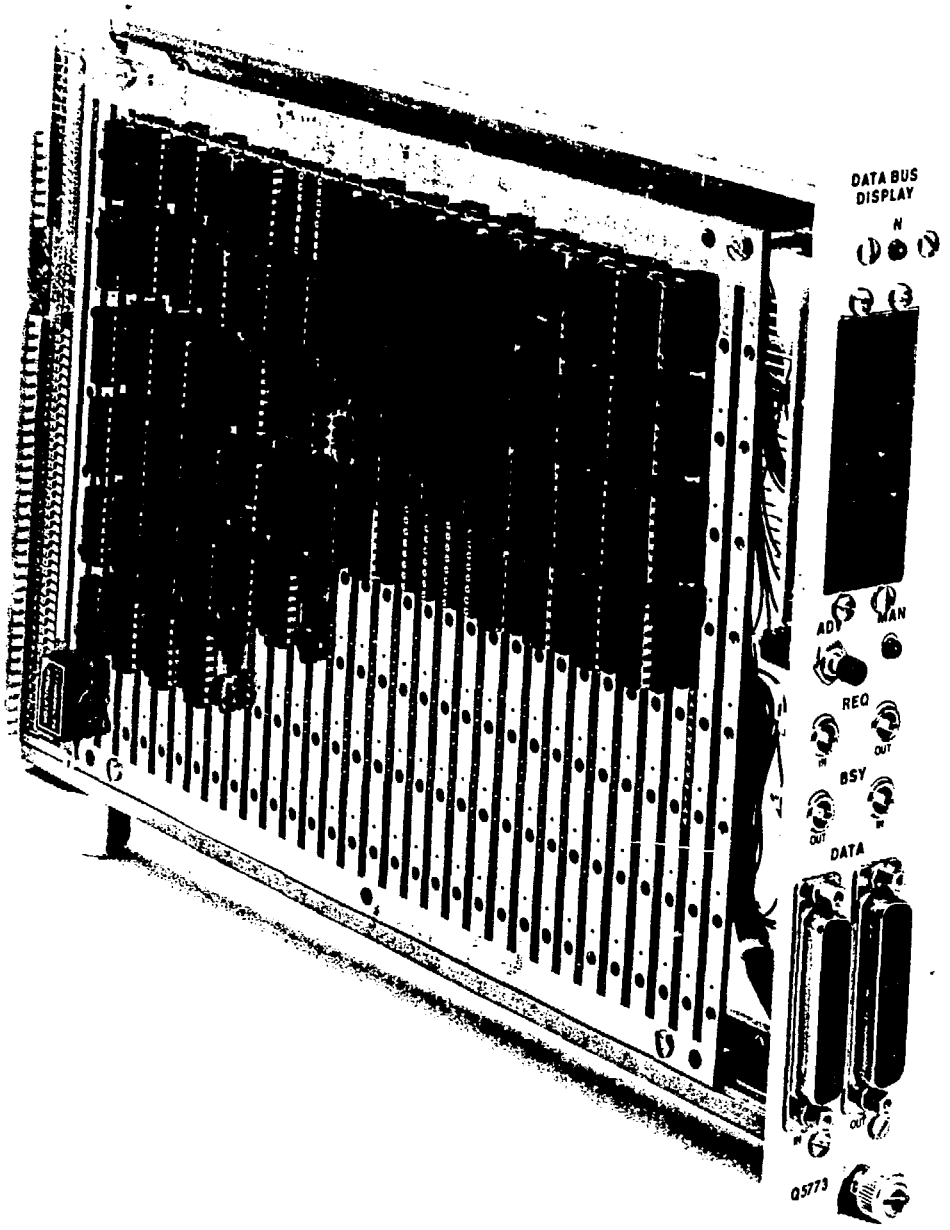
DATA BUS
DISPLAY
N
ADD
MAN
REQ
IN
OUT
BSY
OUT
IN
DATA
IN
OUT
Q5773

Fig. 13(e)

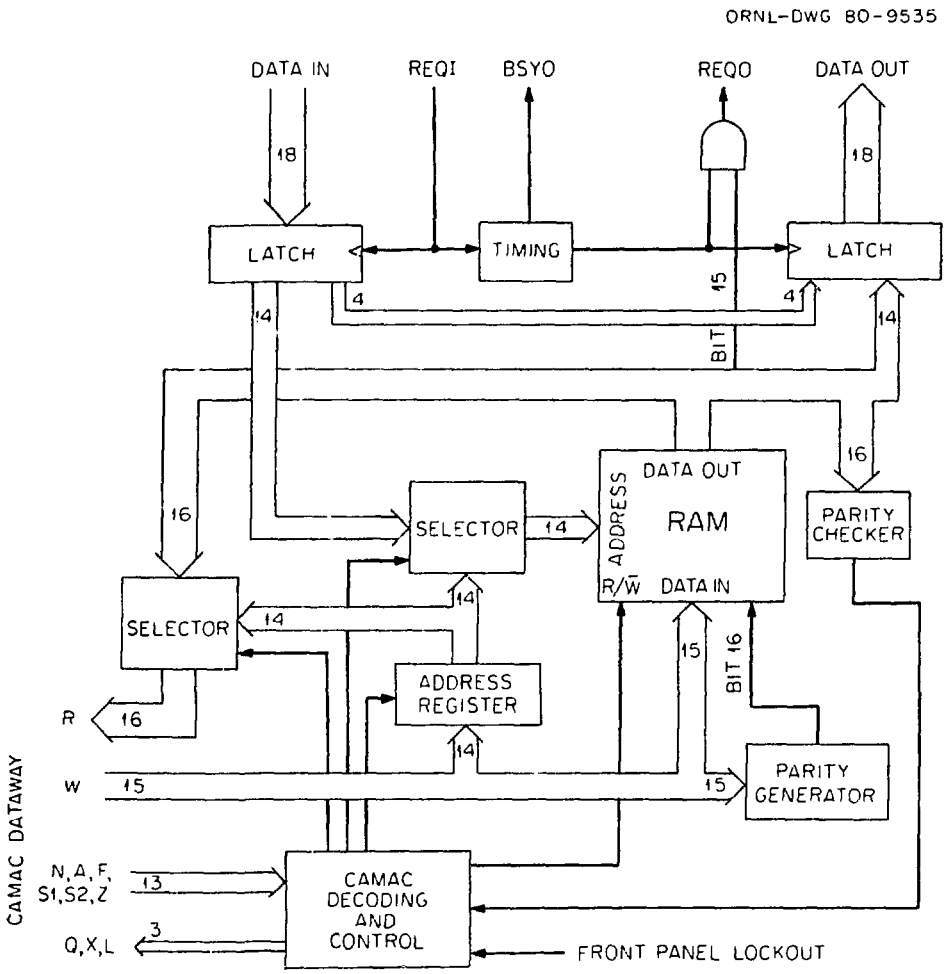
ORNL-DWG 80-9535
DATA IN
REQI
BSYO
REQO
DATA OUT
18
18
LATCH
TIMING
LATCH
15
4
4
14
14
BIT
16
16
DATA OUT
ADDRESS
RAM
R/W
DATA IN
SELECTOR
14
PARITY CHECKER
SELECTOR
14
14
15
BIT 16
ADDRESS REGISTER
R
16
14
W
15
15
PARITY GENERATOR
CAMAC DATAWAY
N, A, F,
S1, S2, Z
13
CAMAC DECODING AND CONTROL
Q, X, L
3
FRONT PANEL LOCKOUT

Fig. 14

ORNL-DWG 80-9536

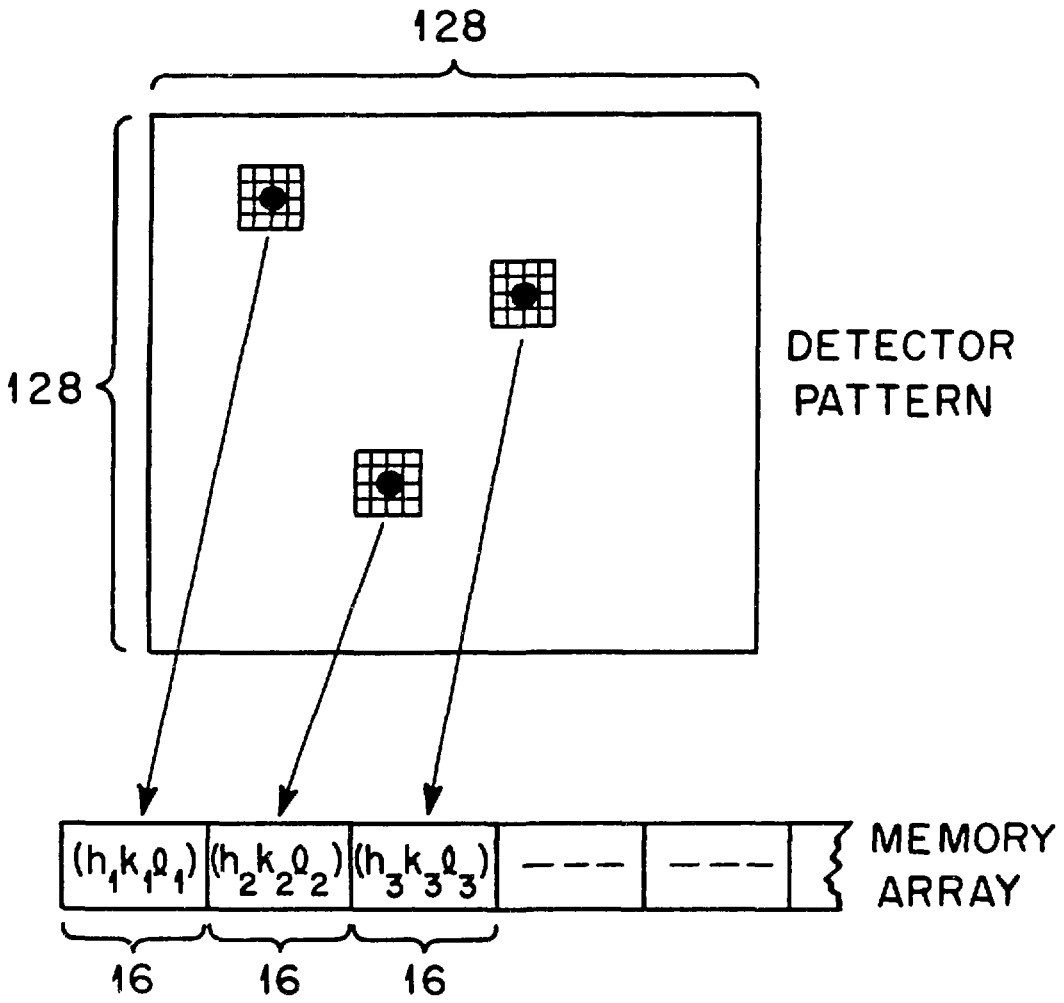

Fig. 15

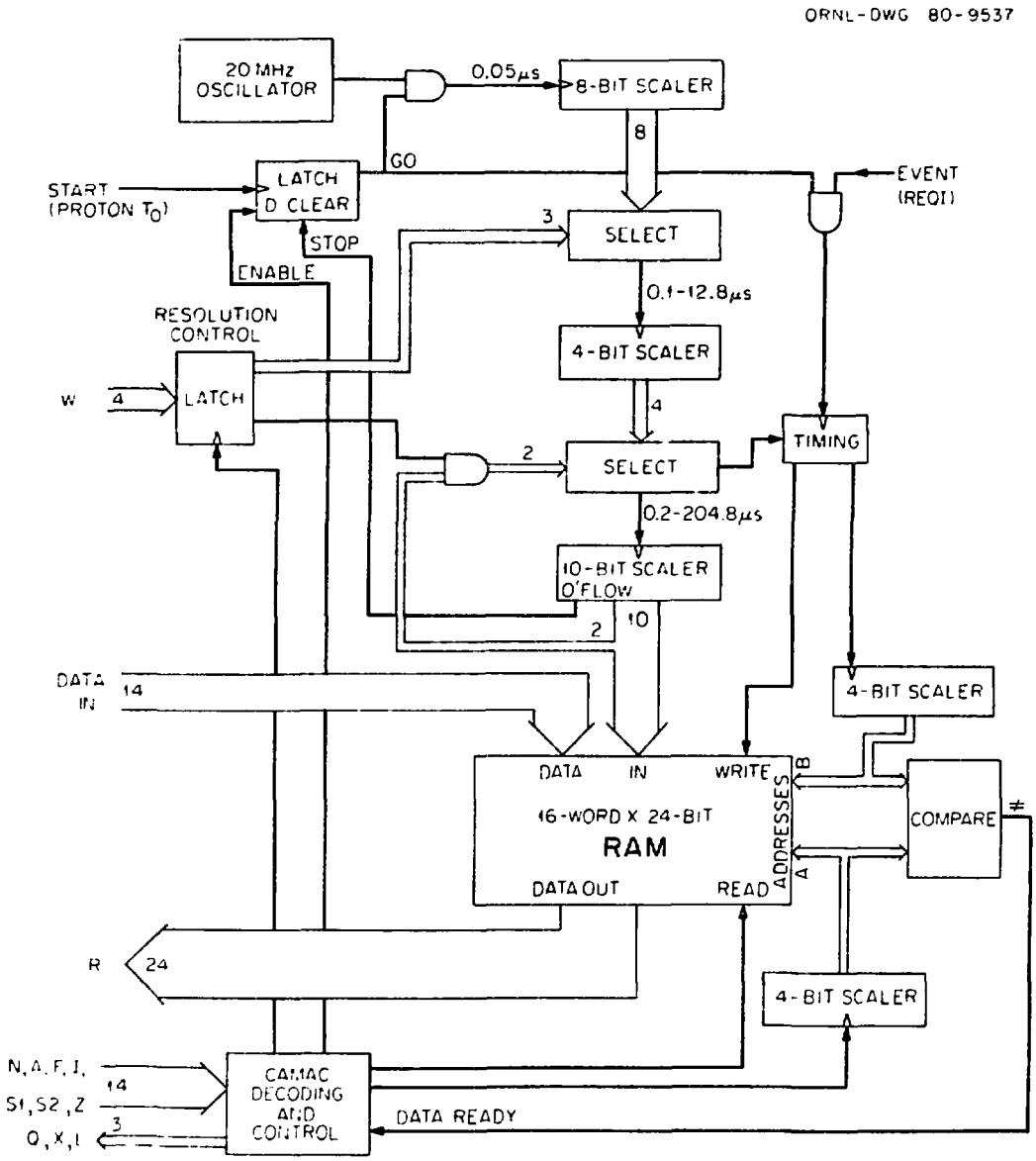
ORNL-DWG 80-9537
20 MHz OSCILLATOR
0.05μs
8-BIT SCALER
8
GO
START (PROTON T0)
LATCH D CLEAR
STOP
ENABLE
EVENT (REQ I)
3
SELECT
0.1-12.8μs
4-BIT SCALER
4
RESOLUTION CONTROL
W
4
LATCH
2
SELECT
TIMING
0.2-204.8μs
10-BIT SCALER O'FLOW
2
10
DATA IN
14
4-BIT SCALER
DATA IN
WRITE
16-WORD X 24-BIT
RAM
DATA OUT
READ
ADDRESSES
B
A
COMPARE
≠
R
24
4-BIT SCALER
N, A, F, I,
S1, S2, Z
O, X, L
14
3
CAMAC DECODING AND CONTROL
DATA READY

Fig. 16

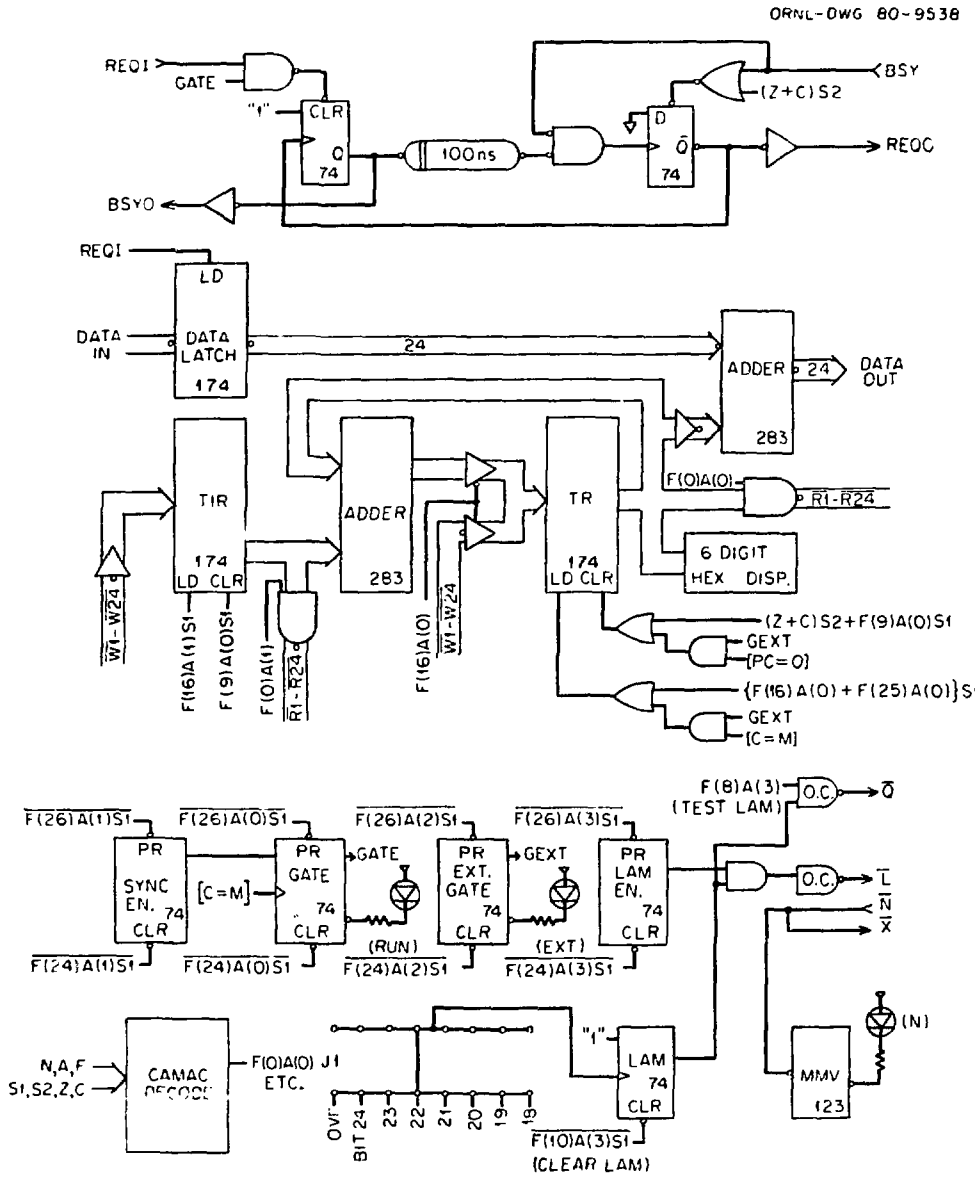
ORNL-DWG 80-9538
REQI
GATE
"1"
CLR
Q
74
100ns
D
Q̄
74
BSY
(Z+C)S2
REQO
BSYO
REQI
LD
DATA IN
DATA LATCH
174
24
ADDER
24
DATA OUT
283
TIR
174
LD CLR
ADDER
283
TR
174
LD CLR
F(0)A(0)
R1-R24
6 DIGIT HEX DISP.
W1-W24
F(16)A(1)S1
F(9)A(0)S1
F(0)A(1)
R1-R24
F(16)A(0)
W1-W24
(Z+C)S2+F(9)A(0)S1
GEXT
(PC=0)
{F(16)A(0)+F(25)A(0)}S1
GEXT
(C=M)
F(26)A(1)S1
F(26)A(0)S1
F(26)A(2)S1
F(26)A(3)S1
PR SYNC EN. 74 CLR
[C=M]
PR GATE 74 CLR
GATE
PR EXT. GATE 74 CLR
GEXT
PR LAM EN. 74 CLR
F(8)A(3)
(TEST LAM)
O.C.
Q̄
O.C.
L̄
N̄
X̄
F(24)A(1)S1
F(24)A(0)S1
(RUN)
F(24)A(2)S1
(EXT)
F(24)A(3)S1
N,A,F
S1,S2,Z,C
CAMAC DECODE
F(0)A(0) J1 ETC.
OVF
BIT 24
23
22
21
20
19
18
"1"
LAM 74 CLR
F(10)A(3)S1
(CLEAR LAM)
MMV 123
(N)

Fig. 17

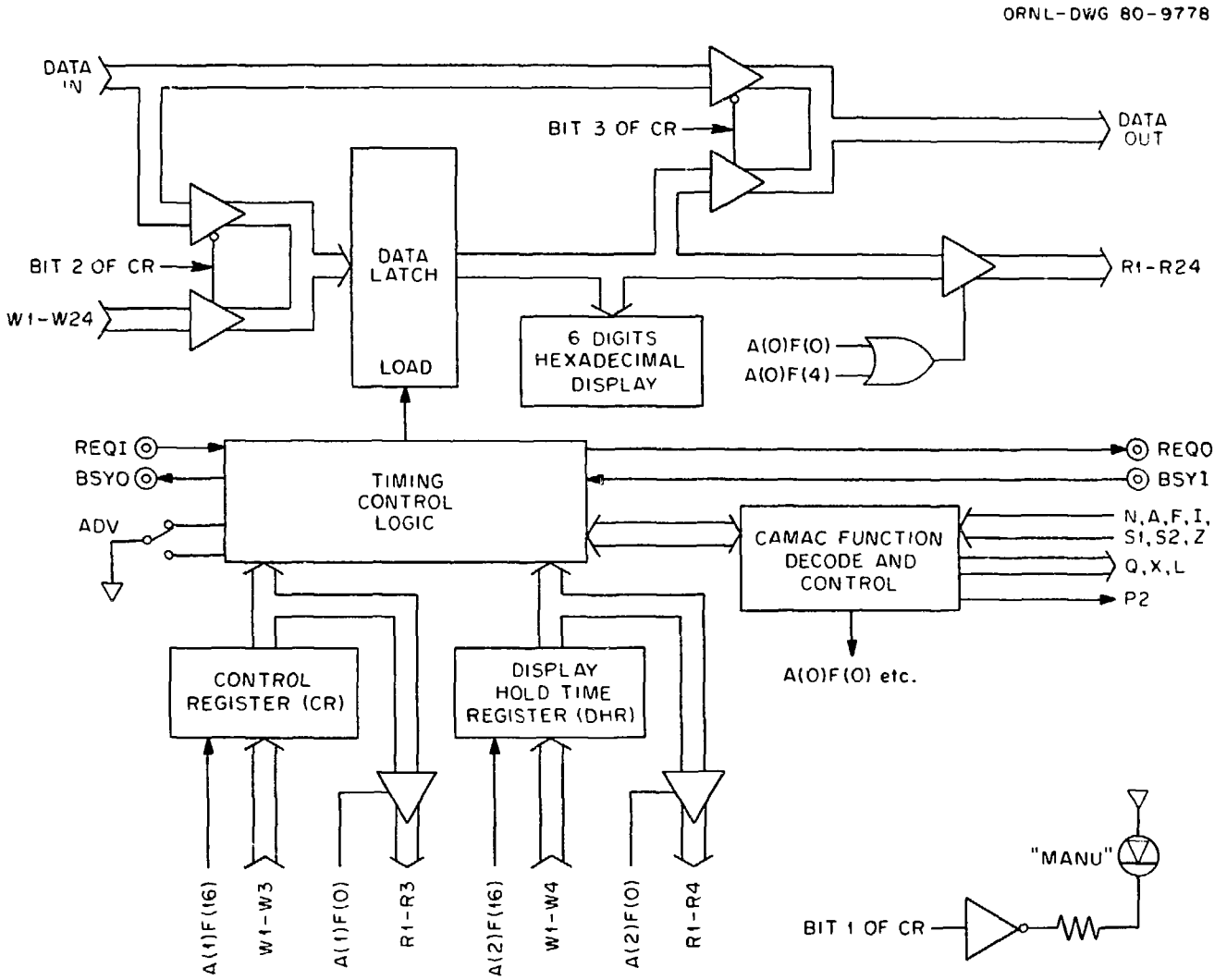
ORNL-DWG 80-9778
DATA IN
BIT 3 OF CR
DATA OUT
BIT 2 OF CR
W1-W24
DATA LATCH
LOAD
6 DIGITS HEXADECIMAL DISPLAY
A(0)F(0)
A(0)F(4)
R1-R24
REQI
BSYO
ADV
TIMING CONTROL LOGIC
REQO
BSY1
CAMAC FUNCTION DECODE AND CONTROL
N,A,F,I,
S1,S2,Z
Q,X,L
P2
A(0)F(0) etc.
CONTROL REGISTER (CR)
DISPLAY HOLD TIME REGISTER (DHR)
A(1)F(16)
W1-W3
A(1)F(0)
R1-R3
A(2)F(16)
W1-W4
A(2)F(0)
R1-R4
"MANU"
BIT 1 OF CR

Fig. 18