21
10-15-79

24 NTIS

ORNL/TM-6922

An Intelligent CAMAC I/O Module Based on the Signetics 8X300 Microcontroller

G. W. Turner
R. W. Hendricks

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION · FOR THE DEPARTMENT OF ENERGY

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A04 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use or the results of such use of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

ORNL/TM-6922

Contract No. W-7405-eng-26

Solid State Division

AN INTELLIGENT CAMAC I/O MODULE BASED ON THE SIGNETICS 8X300 MICROCONTROLLER*

G. W. Turner
Instrumentation and Controls Division

and

R. W. Hendricks
Metals and Ceramics Division

National Center for Small-Angle Scattering Research
Oak Ridge National Laboratory**
Oak Ridge, Tennessee 37830

Date Published: September 1979

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

*Research sponsored by National Science Foundation, Grant No. DMR-77-244-58 under Interagency Agreement No. 40-637-77 with the Department of Energy.

**Operated by Union Carbide Corporation under contract W-7405-eng-26 with with the U. S. Department of Energy.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

CONTENTS

AN INTELLIGENT CAMAC I/O MODULE BASED ON THE SIGNETICS 8X300 MICROCONTROLLER

G. W. Turner[*] and R. W. Hendricks[**]
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830

ABSTRACT

An intelligent CAMAC I/O module based on the Signetics 8X300 microcontroller has been developed. Sixteen 8-bit I/O ports have been utilized; eight are dedicated for data transfers with external devices and/or processes and eight are dedicated to communication with the CAMAC dataway. Separate status and data registers are provided. The input status port (SIN) can receive up to seven individual signals from external devices or the host computer while the output status port (SOUT) can be used to provide up to seven internally graded LAMs and one bit can be used to generate a Q-response for termination of block transfers. Diagnostic software has been developed to operate on the host computer which fully tests all implemented instructions. In our application the device is used in a high-speed memory mapping scheme for data acquisition with a two-dimensional position-sensitive detector system.

[*] Instrumentation and Controls Division

[**] Metals and Ceramics Division

I. INTRODUCTION

The past decade has brought rapid and profound changes to digital electronics and as a result to the procedures and techniques of data acquisition and manipulation in experimental research. The instrument described in this report, an intelligent CAMAC I/O module based on the Signetics 8X300 microcontroller, was developed for use in mapping two-parameter data gathered with a two-dimensional position-sensitive neutron proportional counter into the memory space of a general purpose 16-bit minicomputer. However, we have attempted to keep the design of the module completely general in order that it be able to serve as wide a range of applications as possible.

The Signetics[1] 8X300 microcontroller is the only 8-bit microprocessor built from bipolar technology, and allows 16-bit instructions to be fetched, decoded and executed in 250 ns. It has a repertoire of 8 instructions which are designed for testing status lines, setting or resetting control lines and performing high-speed I/O. With some of its 8-bit I/O ports dedicated to external use and some utilized for interfacing the CAMAC dataway, this unit is ideally suited for high speed, programmable bit-oriented data manipulation.

Our initial experience with the 8X300 was in the development of the ORNL 10-m small-angle x-ray scattering camera as described by Hendricks.[2,3] Here the microcontroller, as implemented by Scientific MicroSystems[4] in their SMS 300 series microcontrollers, was interfaced directly to the I/O bus of a ModComp II 16-bit general purpose minicomputer using standard model 4801 general purpose controllers. Details of the interface are given elsewhere.[5,6] In this application, the 8X300 microcontroller serves several functions. First, it assumes control of all data flow from the ADC's of a two-dimensional position-sensitive detector system to the computer memory by monitoring a variety of flags such as ADC DATA READY, MODCOMP BUSY, HALT DATA AQUISITION, etc. Second, it performs the necessary manipulations to take various bit patterns presented by the ADC's, map them into a linear memory space, and add an origin offset to account for the fact that the origin of the data acquisition array may be at various absolute memory locations of the host computer depending on experimental conditions. With such an arrangement, it is easy to code various software routines in the 8X300 to map different data acquisition arrays (e.g., 64x64, 128x128, 128x32, 256x16, etc.). All handshaking between the MODCOMP and the 8X300 is performed interactively via a 4801 controller. High speed transfers of the assembled address to the ModComp for memory increment are handled via a second 4801 controller which is coupled to a custom memory increment instruction loop which has been installed in the ModComp. [5,6]

The present application, which arose during the development of the NSF-ORNL 30-meter small-angle neutron scattering facility, is almost identical to that of the 10-m x-ray facility. However, with the availability of a fast-response CAMAC crate controller for the ModComp computer[7] it was desirable to interface the 8X300 microcontroller status and control functions to the host computer via the CAMAC dataway instead of through a general purpose controller as was done earlier. The problem was thus to develop hardware which would serve as the interface between the standardized CAMAC bus (dataway) and the input/output ports of the 8x300. The dataway contains a 24 bit read bus and a 24 bit write bus as well as various control lines. A means of accessing all of these lines by the 8X300 was needed, both for input as well as output. As will be seen, we have dedicated eight 8-bit 8X300 I/O ports to this task; three for data input on the CAMAC write lines (24 bits); three for data output on the CAMAC read lines (24 bits); one for status and control input (8 bits); and one for status and control output (8 bits). The status in (SIN) and status out (SOUT) control allow a wide variety of connections to the dataway look-at-me (LAM), Q-response, and I/O lines as well as to external devices such as ADC's and ModComp transfer initiate lines. Finally, packaging had to be considered. The 8X300 is a very high-speed (250 nsec cycle time) processor and thus cable lengths had to be minimized. The following paragraphs described our development of a general purpose CAMAC module which meets these needs.

II. THE 8X300 MICROCONTROLLER

The Signetics 8X300 microcontroller is an 8-bit microprocessor built from bipolar technology. The device has separate instruction address, instruction, and I/O buses. Sixteen bit instructions can be addressed in up to 8192 words of program storage (13 bits), fetched, decoded and executed in only 250 nsec. A block diagram of a typical system is shown in Figure 1. The device used in our application, an SMS 300-140 microcontroller, has 4K of program storage, 1K of read/write data storage, and 16 I/O ports or Interface Vector (IV) bytes.

The 8X300 processor chip has eight internal 8-bit registers and two address registers. The theory applied which makes such a fast instruction cycle time possible is to access program memory and data on the same machine cycle. Normally an instruction is decoded, data read, manipulated and stored

ORNL-DWG 79-12933

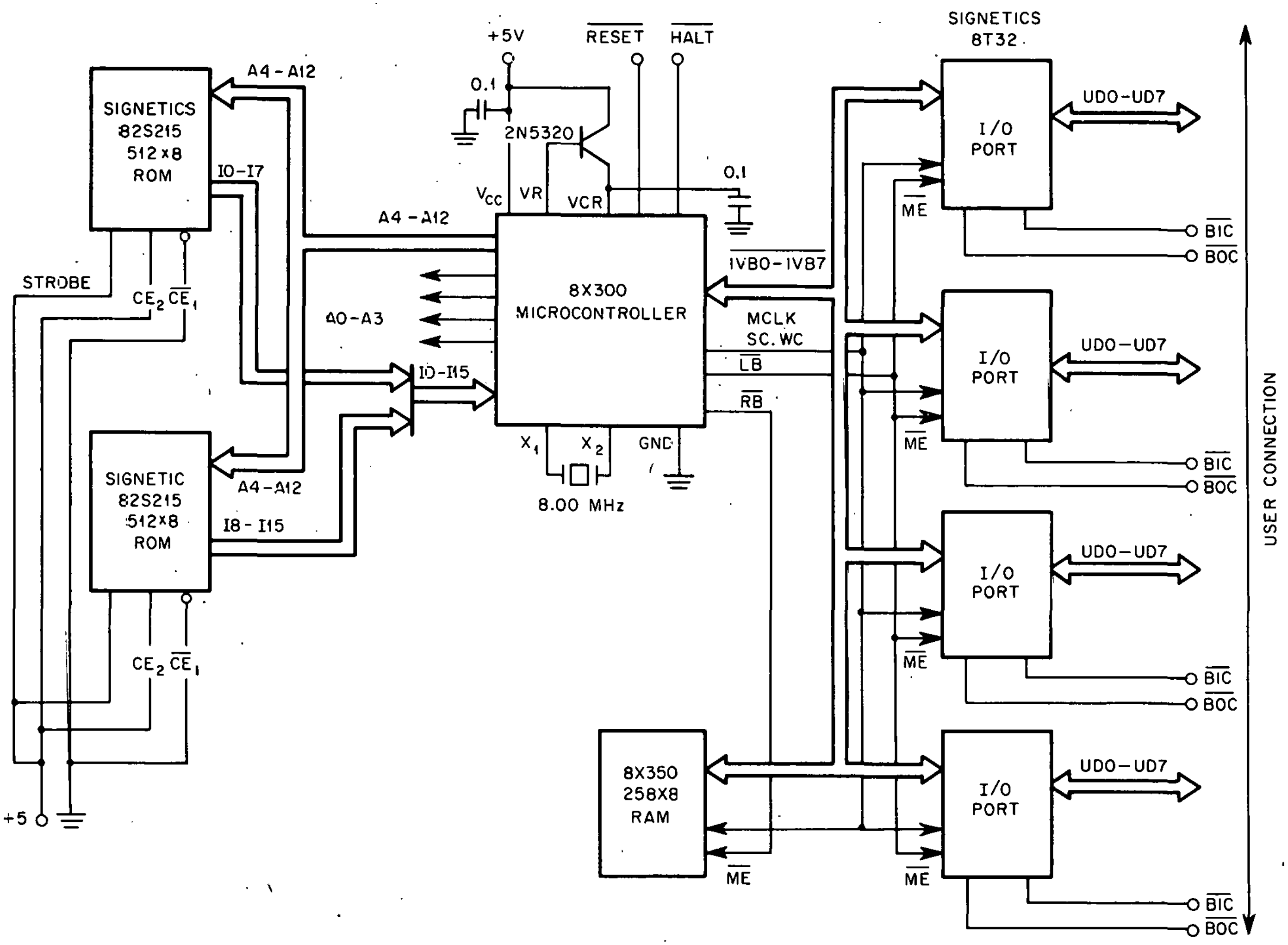


Figure 1. Block diagram of the data flow in the 8X300 microcontroller system (by permission from Ref. 9).

and the next instruction address generated in one machine cycle. The 16 bit instruction word contains the operation code, source, destination and bit field length or rotation of data.

The 8X300 has a repertoire of eight instructions. These include bit manipulation (MOVE, ADD, AND, XOR), control (XEC, NZT, JMP) and data transfer (XMIT). A summary of each instruction is given in Table I. A unique feature is that data on the various IV bytes are treated the same as data in the internal registers.

Data I/O is handled over the Interface Vector (IV) bus. These IV bytes are organized as 8-bit wide bidirectional ports with latch tristate outputs. (Signetics 8T32 I/O ports). Each I/O port has an on-board 8 bit address burned in and a byte input and byte output control (BIC and BOC). The select control/write control (SC/WC) lines are used to determine whether the data on the I/O bus are an I/O port select address (SC high; WC low) or are data address to the last selected port (SC low; WC high). In the I/O port select mode, the port whose internal address matches the address on the I/O bus is enabled and remains enabled until the next I/O select command. All other ports are disabled. The I/O ports have the option of being bidirectional if BIC and BOC are under program control from a different output IV byte. However, in our application we have chosen to dedicate certain I/O ports to input and others to output by hard-wiring BIC and BOC. This procedure eliminates instructions and thus reduces the execution time of certain very fast loops which are necessary for our data handling software.

For further details concerning the 8X300 the reader is referred to the review by Lau,[9] the 8X300 Reference[10] and Programming[11] Manuals, the

Table I

SUMMARY

8X300 INSTRUCTION SET[8]

INSTRUCTION	OP CODE	DESCRIPTION
MOVE	0	Data from the source register or IV bus is moved to the destination register or IV bus. The data may be rotated any number of places and/or masked to any length during the MOVE operation. The source data field remains unchanged after the operation.
ADD	1	Data from the source register or IV bus is added to the contents of the AUX register in ALU and the result is placed in the destination register or IV bus. The data may be rotated and/or masked during the operation. The source data field and the AUX register remain unchanged unless one is also the destination.
AND	2	Data from the source register or IV bus is ANDed with the contents of the AUX register and the result is placed in the destination register or IV bus. The data may be rotated and/or masked during the operation. The source data field and AUX register remain unchanged unless one of those is also the destination.
XOR	3	Data from the source register or IV bus undergoes an EXCLUSIVE OR comparison with the contents of the AUX register. The result is placed in the destination register or IV bus. Data may be rotated and/or masked during the operation. The source data field and AUX register remain unchanged unless one of those is also the destination.
XEC	4	Causes execution of the instruction at the address formed by replacing the least significant bits of the last address with the sum of the I field and the data in the source register or IV byte. After execution of the instruction at the specified address, instruction execution continues at the address following the XEC instruction, unless the executed instruction caused a jump.
NZT	5	The least significant bits of the instruction address are replaced by the I field data if the register or IV bus specified by the source field has nonzero contents. The tested data field remains unchanged.
XMIT	6	The data in the I field is placed in the register or IV bus specified as the destination.
JMP	7	The address of the next instruction to be executed is changed to that specified by the 13-bit A field of the instruction.

SMS 300 data sheets[12] and the Microcontroller Application Guide [13] and System Description.[14] To assist the user with 8X300 applications there are also available an SMS microcontroller monitor[15] which provides display and control of the SMS 300 implementations of the 8X300, and an SMS microcontroller simulator[16] which provides capability for *in situ* development and testing of microcontroller software before the instruction ROMs are burned. We have found the latter device to be indispensible in our prototype work. Finally, there is a FORTRAN IV cross assembler program[17] which executes on a large computer and which generates symbolic listings and object modules. The resulting object modules can then be loaded and tested with the simulator before committing the program to ROM coder.

III. INTRODUCTION TO CAMAC

CAMAC (Computer Automated Measurement and Control) is an internationally defined system by which various modules, manufactured by different firms and designed to perform a wide variety of data acquisition and control functions, can be interfaced to general-purpose digital computers. An excellent overview of the CAMAC system is given in Reference 18, while the actual standards are given in Reference 19. The standard includes the definition of a crate, which is the box into which all the modules fit, and a backplane dataway which contains all power, control, and data lines. A crate contains 25 physically identical slots or stations. However only 24 can be used for general purpose modules; the twenty-fifth is the master or crate controller position. Modules may occupy any integral member of stations. The module described in this report occupies four stations. Each station has identical access to the dataway whose signals are described below. Two lines in the dataway, the address (N) line and the look-at-me (LAM) or interrupt line, are unique to each station and are connected from a given station directly to the master control station (N = 25). All other dataway signals are available to every station. Thus modules can be uniquely addressed by slot and generate interrupts to the crate controller independent of the type of module in the slot, and there is no need for on-board address decoding or source identification generation within the modules.

Control and status signals are generated by the crate controller to provide the necessary communications between it and a given station. These include N (station number), A_0-A_{15} (subaddress within a module), F_0-F_{31} (function code), B (dataway busy), Z (initialize, set all modules to a defined state), C (clear all modules), and I (inhibit operation). Module responses to various dataway operations are Q (indicates module status and depends on NAF command) and X (instruction accepted). Of the 32 function (F) codes, 18 are used for defined operations (read, write, test LAM, clear, etc.). An additional 6 are reserved for future expansion of the system and the remainder are nonstandard for the designer to implement his own functions. All data communication is via a 24 bit read bus and a separate 24 bit write bus. Two strobe clocks (S1, S2) are generated by the crate controller for synchronization.

IV. OVERVIEW OF THE MODULE

We set three criteria for our implementation of the 8X300 in a CAMAC module: (i) the device must be self-contained and occupy a minimum number of CAMAC stations; (ii) it must meet all of the specifications of the IEEE-583 and following standards,[19] and (iii) it should be as general purpose a module as possible and yet perform our primary mission without compromise. The Scientific MicroSystems SMS 300-140 module contains the 8X300 microprocessor, 4K of program storage, 1K of RAM data storage and 16 IV bytes on a single board which will fit in a single CAMAC slot. Of the 16 I/O ports, it was determined that one each should be dedicated to a status in (SIN) and a status out (SOUT) word. These control words provide the necessary signals for the 8X300 software to handle a variety of handshaking tasks between external devices and the CAMAC dataway. In addition, we decided that the CAMAC read and write lines should be interfaced to the 8X300 through separate IV bytes.* We thus utilize a total of six bytes for data I/O and two more for status and control. The remaining eight IV bytes may be used for connections to external devices.

A block diagram of our implementation is given in Figure 2, and photographs of the completed assembly are shown in Figure 3. The CAMAC functions for which there is a module response are given in Table II. The module produces an X-response to every N which addresses the station in which the interface card is mounted. The addresses of the 8X300 interface vector I/O ports and their corresponding dataway connections are given in Table III.

Our module occupies four CAMAC stations (N through N+3) and has two card connectors to the dataway. One of these (N+1) is used exclusively to

*It is possible to use only three IV bytes for I/O by using an SOUT bit to control the I/O BIC and BOC lines. However, such a procedure forces the programmer to use two instructions to address an IV byte; one for IV byte select and a second for I/O direction. We choose to use an additional three IV bytes to gain speed and flexibility.

ORNL-DWG 79-12932

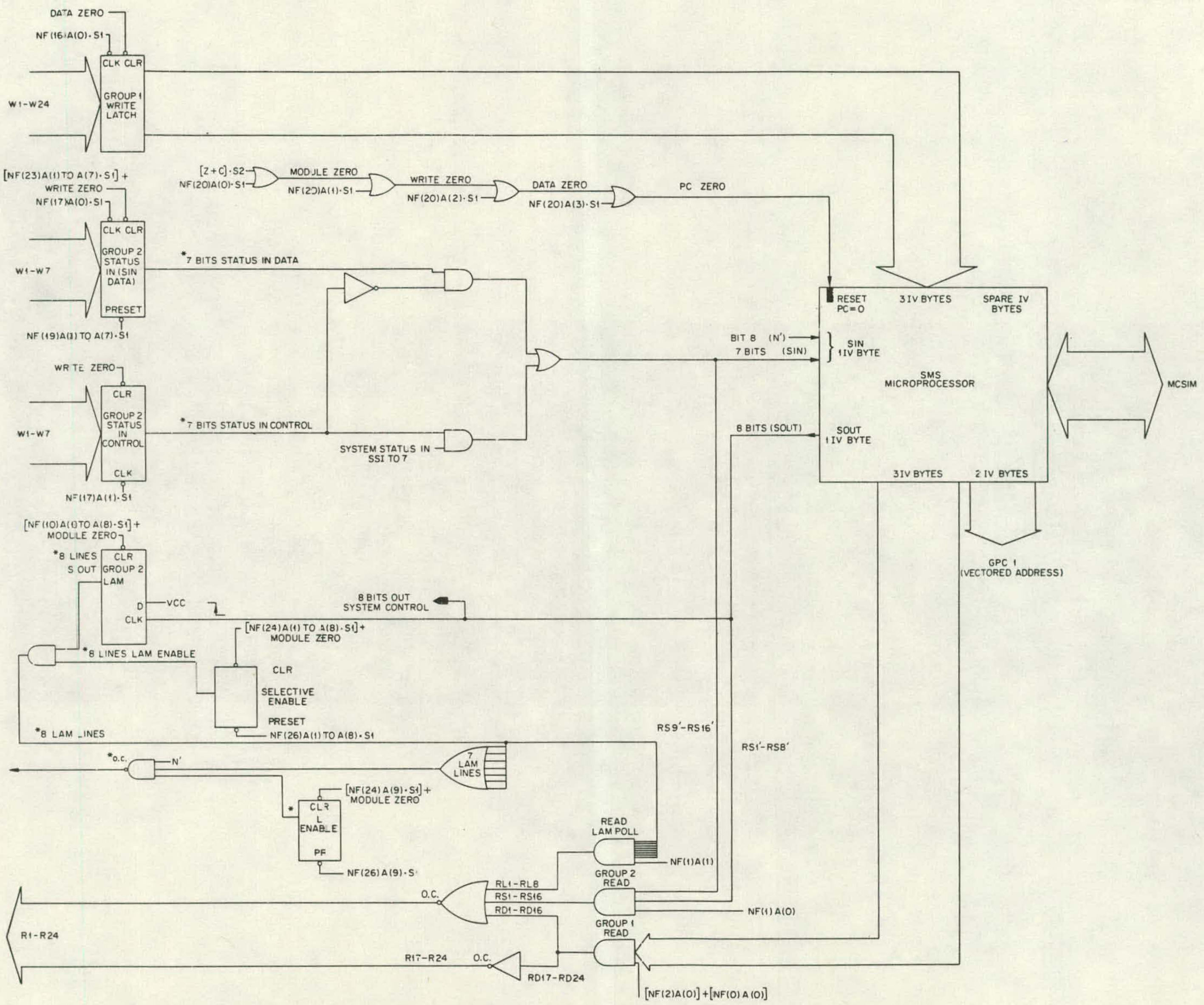


Figure 2. Blɔck diagram of the 8X300 CAMAC interface implementation.

ORNL PHOTO 1576-79

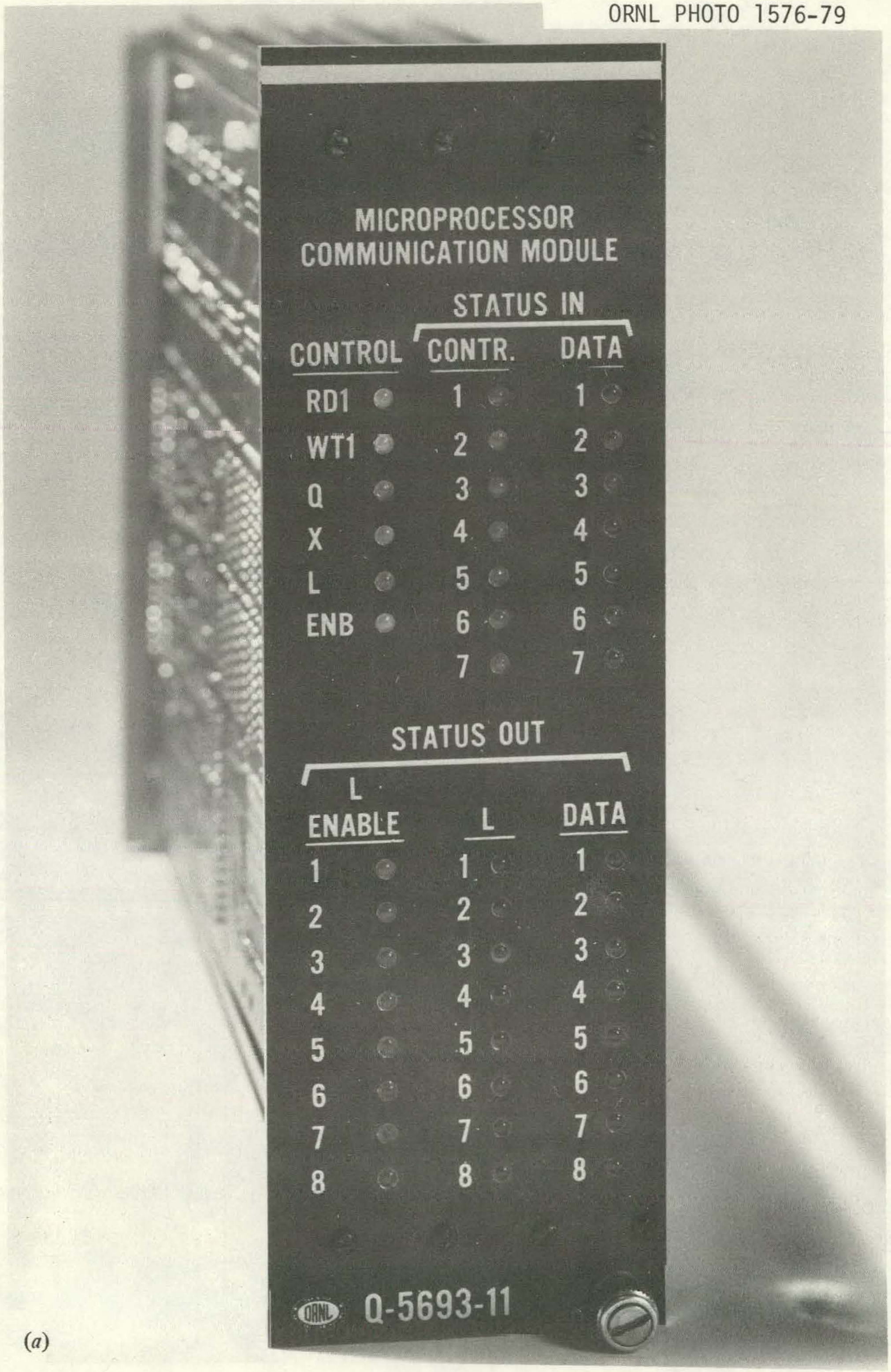
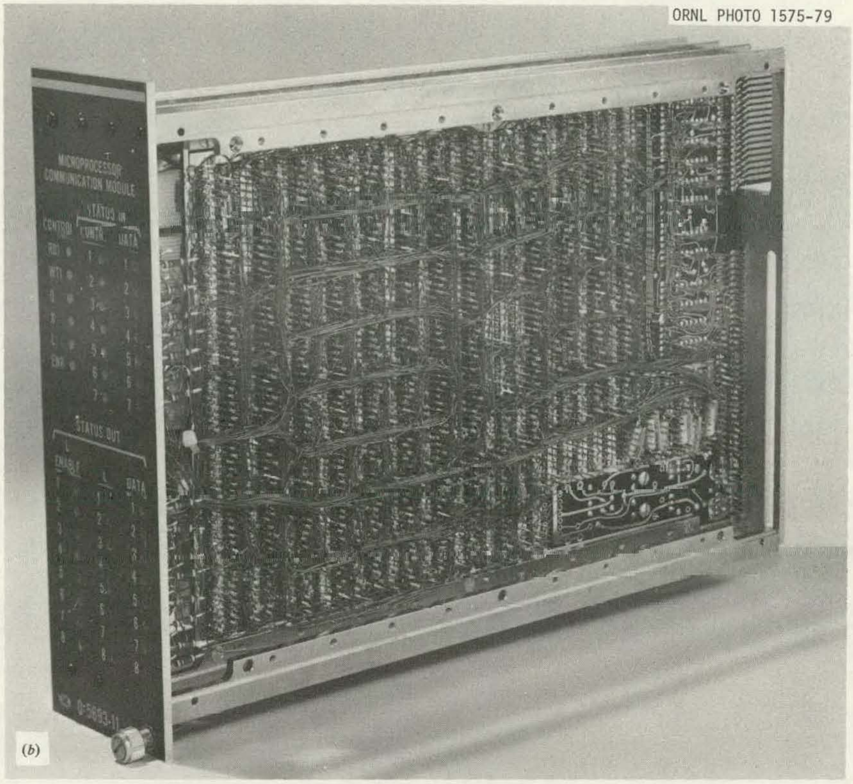


Figure 3. The 8X300 CAMAC module. (a) front panel; (b) interface board wire-wrap side; (c) open module with interface board (left) and SMS300-140 (right) swung open for access.


ORNL PHOTO 1575-79
MICROPROCESSOR
COMMUNICATION MODULE
CONTROL
STATUS IN
CONTR.
DATA
RD1
WT1
STATUS OUT
ENABLE
DATA
Q-5693-11
(b)

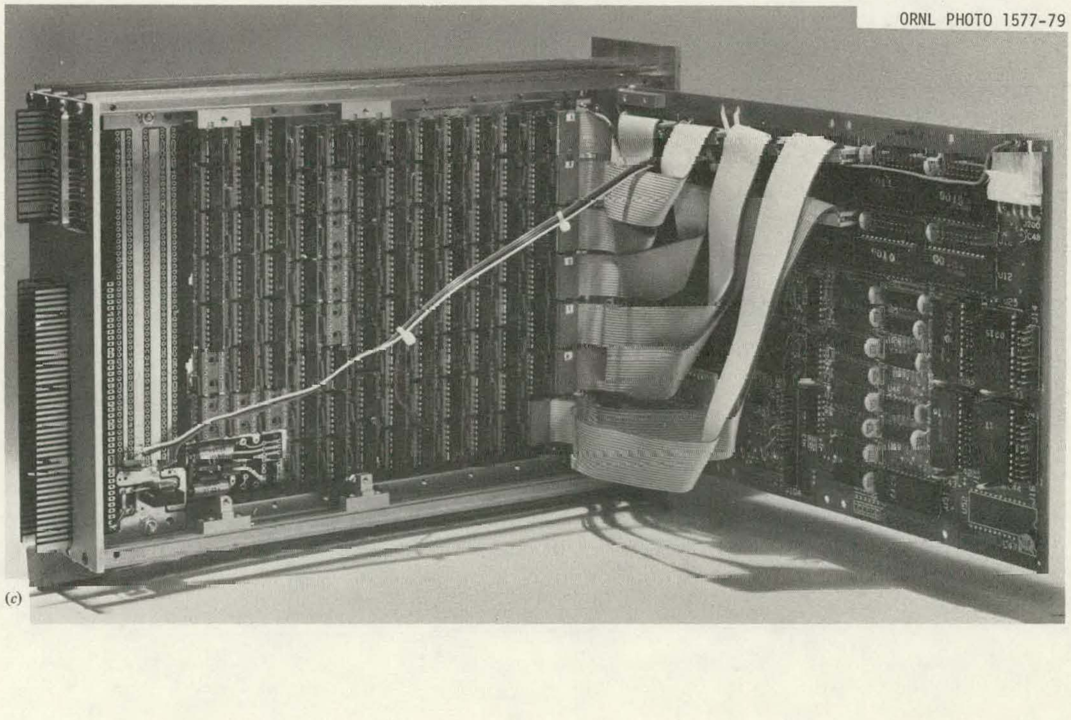

ORNL PHOTO 1577-79
(c)

TABLE II

CAMAC FUNCTION CODES

Command	Function Mnemonic	Q	Action
F(0)•A(0)	RD1	0	Strobe IV2, IV5, and IV4 to R1-R24
F(1)•A(0)	RD2	0	Strobe SOUT (IV6) to R1-R8 and SIN (IV7) to R9-R16
F(1)•A(1)	RD2	0	Strobe LAM status to R1-R8
F(2)•A(0)	RC1	0	Strobe IV2, IV5, and IV4 to R1-R24; clear SOUT(3)
F(8)•A(0)	TLM	1	Test LAM, Q=1 if set
F(8)•A(1)	TLM	1	Test block transfer response SOUT (8); Q=1 if set
F(10)•A(n)	CLM	0	Selective clear SOUT(n); n=1,2...7
F(10)•A(8)	CLM	0	Selective clear SOUT(8) for Q response
F(16)•A(0)	WT1	0	Latch W1-W24 to IV1, IV3, and IV20
F(17)•A(0)	WT2	0	Latch W1-W7 to SIN (IV7)
F(17)•A(1)	WT2	0	Latch W1-W7 to select external device or computer for input to SIN (IV7) register; 0 selects computer, 1 selects external device.
F(19)•A(n)	SS2	0	Selective set SIN(n) latch; n=1,2,...7
F(20)•A(0)	MZ	0	MODULE ZERO. Disable LAM's; clear individual LAM's; disable all SOUT bits for LAM; perform all functions of F(20)•A(1), F(20)•A(2), and F(20)•A(3).
F(20)•A(1)	WZ	0	WRITE ZERO. Clear SIN register and SIN control register; perform all functions of F(20)•A(2) and F(20)•A(3).
F(20)•A(2)	DZ	0	DATA ZERO. Clear input write registers; perform functions of F(20)•A(3).
F(20)•A(3)	PZ	0	PROCESSOR ZERO. Reset microcontroller program counter to zero.
F(23)•A(n)	SC2	0	Selective clear SIN(n); n=1,2,...7
F(24)•A(n)	DIS	0	Selective Disable SOUT(n) for LAM; n=1,2,...7
F(24)•A(8)	DIS	0	Selective disable SOUT(8) for Q-response
F(24)•A(9)	DIS	0	Disable LAM

TABLE II (Cont'd)

Command	Function Mnemonic	Q	Action
F(26)•A(n)	ENB	0	Selective enable SOUT(n) for LAM
F(26)•A(8)	ENB	0	Selective enable SOUT(8) for Q-response
F(26)•A(9)	ENB	0	Enable LAM
Z		0	MODULE ZERO [same as F(20)•A(0)]
I		0	NO ACTION
C		0	MODULE ZERO [same as F(20)•A(0]

TABLE III

8X300 INTERFACE VECTOR CONNECTIONS

INPUT		INPUT	
CAMAC	IV Byte	CAMAC	IV Byte
W1-W8	20_8	R1-R8	1
W9-W16	5	R9-R16	3
W17-W24	4	R17-R24	2
SIN	7	SOUT	6

obtain power for the SMS 300 microcontroller (2.0 A; 5V). All of the CAMAC dataway interface is on a single Standard Engineering[20] wire-wrap kludge card (WW04) which is mountd in slot N+2. For ease of servicing, we have mounted the microcontroller on hinges in such a manner that it can be easily swung open to gain access to the component sides of both boards (Fig. 3c).

Throughout we have used low power Schottky (LS) TTL logic wherever possible in order to minimize power consumption. Still the interface board draws 2.0 amps at 5 volts, thus making the total module power requirement 4.0 amps at 5 volts. A layout of the various integrated circuit chips is given in Appendix A.

Several features of the module should be pointed out. Each SIN bit may be externally selected to be connected either to latches on the dataway or to an external device. The choice is determined by loading a 7 bit word onto W1-W7 with a 0 implying computer control and a 1 external control. Thus, it is possible to have status information available to the microcontroller from both the computer or an external device and to be sure that a given bit is connected to only one device at a time. Note also that this arrangement allows an external device to be connected but be ineffective by setting its control bit to zero. Finally, bit 8 of SIN is connected directly to the dataway N line in order to make that signal available to the 8X300. The mode in which a given SIN bit is utilized is displayed on the front panel as Status in Control (see Fig. 3a); an illuminat LED indicates external source while a dark LED indicates computer source. The state of each SIN bit is displayed in the second column of LED's (Status In Data) and can be read directly on to the dataway R-lines (R9-R16). This gives the host computer access to the external device status lines. There is no LED display for the standard read/write lines as this information is readily available with commercial dataway display modules.

The SOUT IV byte is more complex. First, each bit can be read directly to the dataway R-lines (R1-R8). More importantly, bits 1-7 can be used to generate structured LAM's, with the enable and disable for each bit under selective control. Bit 8 can be used to generate a Q-response. This is essential to implement certain modes of CAMAC block data transfers.[21] The status of the individual LAMs can be determined from the response to a test LAM command. Front panel LED's indicate the state of the SOUT IV byte, whether a given bit is selected for LAM, and whether that LAM is set.

The various sub-LAM's described above are OR'ed to produce a single module LAM which may also be separately enabled or disabled. Front panel LED's indicate whether or not the module LAM is enabled and whether or not it is set. In addition, N, X, and Q are displayed.

V. DETAILED DESCRIPTION

A. Block Diagram

The following description provides details for which the reader should reference the schematic drawings found in Appendix D. In this section reference is made to drawing Q5693-11A Functional Block Diagram Microprocessor Communications Module.

The dataway write lines (W1-W24) are strobed at S1 into a 24 bit latch register with N•A(0)•F(16) and may be accessed by three 8X300 IV bytes (Table 3). This register (called the group 1 write latch) can be cleared with the data zero (DZ), the crate clear (C), or the crate zero (Z) commands.

The status in (SIN) register consists of 7 bits of user controllable information plus the module select (N) which is tied to bit 8. The status information may come either from the CAMAC write lines (W1-W7) or from external inputs. In order that a SIN bit can be used with either source without the need for removing cables, a 7 bit steering register was installed to select one or the other input. This register, the group 2 status input control, is loaded from W1-W7 with an N•A(1)•F(17) command. A zero on write line n selects SIN(n) for CAMAC input; a one selects it for external input. The virtue of this control is that it can be latched and then not reset until it is necessary to change data paths. The group 2 status control register is cleared with a write zero (WZ), C, or Z. Once the status control register has been loaded, the status bits connected to the dataway can be latched from W1-W7 into the group 2 status input data register with an N•A(0)•F(17). In addition, this register has a network of discrete flip-flops which allows any bit to be individually cleared or set with the commands N•A(n)•F(19) or N•A(n)•F(23) respectively. In addition, the WZ, C, and Z commands also clear the group 2 status input data register. The purpose of the individual clear/set operations is to provide the programmer with the capability of operating on a single bit during handshaking operations. (Single bit operations are implemented in the 8X300 instructions as described in Section II.) The status input data, regardless of the data path through which it enters the module, can be read back into the dataway as will be described below.

The final input signal to be described is processor zero (PZ). This signal resets the 8X300 program counter (PC) to zero, and is derived from N•A(3)•F(20).

The data from the microcontroller out to the dataway read lines (R1-R24) are set up in a 24 bit wide bus. Since the 8X300 IV bytes are tristate latched output, it was necessary only to gate these signals with the appropriate CAMAC command and put them through open collector drivers. The data on R1-R24 can represent numerous microcontroller functions: N•A(0)•F(0) or N•A(0)•F(2) (group 1 read) place the data in the three dedicated IV bytes (Table III) on R1-R24; N•A(1)•F(0) is a group 2 status read and places the SOUT IV byte on R1-R8 and the SIN IV byte on R9-R16; and N•A(1)•F(1) is a LAM status poll. These functions will be discussed in detail.

The group 1 read [N•A(0)•F(0)] is a simple read which strobes the data IV bytes onto R1-R24. The group 1 read and clear [N•A(0)•F(2)] performs as N•A(0)•F(0) but in addition automatically clears SOUT(3). This operation is equivalent to N•A(0)•F(0) followed by N•A(3)•F(23), and increases the speed of I/O mode CAMAC-host computer data transfers by eliminating the need for the second instruction.

The LASL fast-response CAMAC crate controller[7] is capable of performing hardware controlled DMP block transfers to the host computer in which the Q-response is used as the block terminator.[5,18] It was therefore decided to implement one status control bit [SOUT(8)] such that the 8X300 could generate a Q-response during high speed data transfers out of its own working storage ROMS. The remaining SOUT bits can be used to generate LAM's or be routed to external devices for control purposes. When setting up the logic surrounding the LAM interrupts, we wanted the ability to selectively enable or disable each bit for LAM (or Q) generation individually as well as the ability to clear the entire register simultaneously in order to have the greatest versatility and dataway contol. Thus, the edge triggered group 2 LAM register is cleared with N•A(1)• F(10) through N•A(8)•F(10). The output of this register is ANDed with the outputs of the selective enable/disable register, an operation which determines if the output of the group 2 LAM register is allowed to advance to the LAM request circuit. Each bit of the selective enable register is enabled with N•A(1)•F(26) through N•A(8)•F(26) and selectively disabled with N•A(1)•F(24) through N•A(8)•F(24). Module Zero (MZ) disables all bits simultaneously. The seven enabled LAM requests are ORed onto a single line which is then gated with the module LAM enable [N•A(9)•F(26)] or disable [N•A(9)•F(24)] signal and N. The eighth bit is connected directly to the Q line and has no additional control. The presence of a LAM may be tested with N•A(0)•F(8) which produces Q=1 (LAM) or Q=0 (no LAM). The Q-response bit can be tested with N•A(1)• F(8) which results in Q=1 if SOUT(8) is both enabled and set. Otherwise Q=0.

B. LAM/Status Logic

This section describes the logic on drawing Q5693-11B. The basic block is the status out registers, the LAM enable registers, the LAM bus, and the logic associated with L, Q, and X.

The status out (SOUT) bits 1-8 are paralleled off the inputs to the 74LS153's 14D, 8E, 14E, and 8F (Group 1/group 2 read and function decode). These inputs are used as the clock input to a bank of eight D flip-flops (10A, 11A, 11B, 11C, 11D) which make up the status out register. The D input is tied to V_{CC}; thus any positive transition will be latched in these registers. The Q outputs of these registers are called SOUT(1-8) and are displayed on the front panel. The signals are run through a hex inverter and a 360Ω current limiting resistor and then to the LED. The status out registers may be cleared as a group or individually. The signal that clears all of them is MZ (module zero).* The registers may be individually cleared with N•A(1)•F(10) through N•A(8)•F(10). This is accomplished by taking the AND of N•F(10) and S1 then ANDing this with the appropriate A signal. For reasons associated with our particular application, we chose SOUT(3) to be the status flag for a SMS-ModComp data transfer request. Because of this, SOUT(3) can be cleared with one additional signal, N•A(0)•F(2). The N•A(0)•F(2) is NOR'ed with the standard clearing signal N•A(3)•F(24) only on bit 3.

The other major register on this sheet is the LAM enable register which is also implemented with D flip-flops. The individual bits of the register are cleared with N•A(1)•F(24) to N•A(8)•F(24). N•F(24) is ANDed with S1 and this signal is ANDed with the appropriate A signal. These are then NORed with MZ which is used to clear all registers. To set a bit in the LAM register, N•A(1)•F(26) to N•A(8)•F(26) are used. The AND of N•F(24) and S1 is NANDed with the appropriate A signal. Thus, when the output of one of these gates (10A, 10B, 10C, 10D, 5B) goes low the corresponding register FF is preset (set Q=1). These functions enable the programmer to choose any combination of SOUT bits to be seen on the CAMAC dataway as LAMs.

The output of these L enable registers is displayed on the front panel through the same type circuitry as the SOUT signals. These two latched outputs are ANDed to give read LAM (RL1-RL8). This eight bit group of signals is then put back into the input of the read selectors (74LS153's on group 1/group 2 read and function decode), and are displayed on the front panel on LED's. The RL signals are also all ORed to one signal at 13D(6).

*This signal also clears all registers in this module.

Thus when any LAM goes high 13D(6) goes high. This signal is gated with a latched L enable flag. L is enabled with N•A(9)•F(26) which is obtained by ANDing N•F(26), A(9) and S1. When this signal goes low we preset the enable flip-flop, priming the gate 13E(12) and enable the signal at 13D(6) to be transmitted to the L line on the dataway. N•A(9)•F(24), gated with S1, is used to clear the enable flip-flop. As can be seen in Fig. 3(a), L and L ENABLE are displayed on the front panel. To produce X we simply turn N around and also display it on the front panel.

The Q response is used to indicate the status of two functions. In testing the L status N•A(0)•F(8) is used to gate L at 13D(6) onto the Q line. Q is, of course, ANDed with N'. The Q response is also used during a block transfer control (BTC) operation. To initialize the BTC N•A(1)F(8) is gated with SOUT(8) which is used as a BTC request. SOUT(8) is derived just as the other SOUT signals except that it cannot generate an L. When a BTC operation is desired SOUT(8) is set to 1, and N•A(1)•F(8) is generated to enable the Q output and then Q is monitored by the crate controller. Since Q is not latched, a one shot 27KΩ/19μF (70 ms) pulse stretcher is used to display it on the front panel.

C. Read Group 1/Group 2 and Function Decode

This section describes the logic referenced to drawing Q5693-11C. First, consider the decode logic at the left side of the drawing. Since the CAMAC dataway is a negative true bus, while positive true logic is used throughout our module, all signals are inverted as they are either buffered or gated onto the interface. Inhibit (I) and busy (B) are brought onto the board but are not used. The two bus strobes S1 and S2 are inverted and put through a 100Ω/100μF filter network. Initialize (Z) and clear (C) are individually ANDed with S2 to give Z S2 and C S2 and are ORed to produce (Z+C)•S2. The module select (N) is used to strobe the function and subaddress inputs into the 74154 4-to-16 line decoder/multiplexer chips. Two chips are used to decode the dataway signals F1, F2, F4, F8 and F16 into N•F(0) to N•F(31), while a third is used to decode A1, A2, A4, and A8 into A(0) to A(15).

The remaining sections of this drawing show the logic for access to the dataway control and read lines (R1-R24) as shown schematically in quadrants G6/G7 of Q5693-11A. RD1-RD24 are the "read data" lines from the three SMS IV data bytes. Each byte has a unique 3M* cable connector to the interface board, the location of which may be found from Appendix A and Table III. The BOC signal is grounded on these bytes to make them output only bytes. The RS1-RS16 lines are the 16-bit "read status" bus. RS1-RS8 are the lines from the SOUT byte while RS9-RS16 are the lines from the SIN byte. In this case the SOUT IV byte has BOC grounded while SIN has BIC grounded. RL1-RL8 are the "read LAM" lines and come from the LAM logic. They are used in addition to the master LAM (L) line to allow the programmer to set up a priority interrupt scheme for servicing of the LAM.

The hardware used to implement these operations is a series of 74LS153 and 74LS157 two- and three-input multiplexers. The control for selecting which enable is decoded is a 74LS148 priority encoder. Follow now the operation of accessing RD1-RD24. The 24 data lines are already latched on the 8X300 IV byte so it is unnecessary to latch them again. RD1-RD8 are made available to the four 74LS153 chips in 14D, 8E, 14E and 8F. These 'LS153's are dual four-to-one line data selectors. RD9-RD16 are available

*3M Company, St. Paul, MN.

at the inputs of the 74LS157's in 14F and 8G. These 'LS157's are quadruple two-to-one data selectors. Since RD17-RD24 are not multiplexed they simply go through a bank of open collector NAND gates to the dataway. The signals are inverted for the negative true CAMAC logic and open collectors are used to prevent loading of the bus. These gates are strobed with [N•F(0)+N•F(2)]•A(0). This signal is also OR'ed with N•A(1)•F(1) and, when low, is used to strobe 14F and 8G. [N•F(0) + N•F(2)]•A(0) is used as the 0 input to a 74LS148 eight-to-three line priority encoder in 14C which is utilized as a three-to-two line encoder. Thus, a low signal on 0 with 1 and 2 high gives a high on both A(0) and A(1). These signals are used as the A and B inputs respectively to the data selectors 14D, 8E, 14E, and 8F and allow the RD inputs (C0) to appear on the Y output when pins 1 and 15 are strobed low. This signal is also used to take select low on 14F and 8G thus allowing the RD input (A) to appear on the Y output. All the Y outputs of the 'LS153's and 'LS157's go through 7405 inverters to get to a negative true state and open collector output before going to the dataway bus.

RS1-RS16 access is handled in the same manner as the RD lines. Here, an N•A(0)•F(1) is generated by ANDing N•F(1) and A(0). This signal is used as the "1" input for the 'LS148. When this input goes low the output on A(0) and A(1) will be 0 and 1 respectively. These data are put on the A and B inputs of the 'LS153's (14D, 8E, 14E, 8F) allowing the RS signals through the multiplexer to the Y outputs. It should be noted that these chips are strobed by any of the functions which are OR'ed at 13G(6) to go low when [N•F(0) + N•F(2)]•A(0) + N•F(1)•A(1) is present. The 'LS157's carrying RS9-RS16 (14F and 8G) are strobed from 4F(13) going low because N•F(1)•A(0) is OR'ed with [N•F(0) + N•F(2)]•A(0). If the select line is high the RS inputs (B input) will appear on the Y outputs.

RL1-RL8, the read LAM lines, are accessed with an N•A(1)•F(1). This signal is derived by ANDing N•F(1) and A(1). When this signal goes low it puts a low on the (2) input of the 'LS148 which gives an output code of 1 and 0 on A(0) and A(1) respectively. This code, when applied to the A and B inputs of the 'LS153's, enables the RS inputs (C2) to appear on the Y outputs.

The circuit in 1G is a pulse stretcher to display the read data on an LED indicator. The 27 KΩ/10μF arrangement on this one-shot produces a 70 ms delay. These pulse stretchers are used everywhere an indicator is desired on a pulsed output. This particular circuit is active with a high on pin 10 from a [N•F(0) + N•F(2)]•A(0).

D. Write Group 1/Group 2 and Status Control Logic

This section describes the logic on drawing Q5693-11D. The main parts of the logic are the group 1 write buffer register, the status input select and control logic, and the module initialization and zero logic.

The write buffer is the logic used to control the input of the 24 CAMAC write lines W1-W24. These lines are negative true and are inverted for our positive true logic. They are then put into the group one write registers (74LS174's in 8B, 8C, 8D and 14B). These chips contain six D flip-flops to a package and have a common clear and common clock. Only the Q output is available. These registers are needed as a buffer between the CAMAC dataway and the 8X300 because W1-W24 are available for only 600 nsec (during one dataway cycle) and because the 8X300 operates asynchronously with the dataway and thus may not be ready to accept the data at the time they are available directly from the dataway. The data from W1-W24 are latched with the command N•A(0)•F(16) and are strobed with S1. When a high, obtained by ANDing N•F(16), A(0), and S1, is asserted on pin Q of the 'LS174's the data present on the inputs will appear on the outputs. This write command (WT1) is also displayed on the front panel via a pulse stretcher. The output data go directly to the 8X300 input IV bytes (see Table 3 and Appendix A for detailed connections). However, the 24 data lines are not displayed (because of power consumption and an already high chip count). To clear the registers, a data zero (DZ) may be generated. This signal, when low and applied on pin 1 of the 'LS174's, will clear the register. How the zero pulses are derived will be discussed at the end of this section.

The status input select and control logic performs two functions. The first is to present the seven bits of status information to the SIN input IV byte. The second is to provide, under program control, the ability to select whether a bit comes from outside the module system or from the dataway.

Consider first the status select operation. The command N•A(1)•F(17), when strobed with S1, gives a high on pin 9 of the 74LS174 control registers in 8A and 14B and their input is transferred to the Q output. A zero on this output will enable the gate at 9C(9), for example, and because of the inverter 7G will disable the gate at 9C(4). Thus, the 8X300 SIN byte sees the input from the CAMAC dataway line W1' at 3E(2). However, we see that the inputs to the status select registers (8A, 14A) are also derived from W1'-W7'. Thus, a 0 on W1'-W7' selects dataway input to SIN while a 1 selects external sources of data SSI1-SSI7. Since the registers in 8A, 14A are latched, the entire status bus is set up with a single N•A(1)•F(17) command in which the choice of input source is determined by the bit pattern on the W1-W7 lines. The front panel LED's for STATUS IN CONTROL are illuminated if the CAMAC dataway is selected for input, while the STATUS IN

DATA LED's are illuminated if the source (either dataway or external source) is high.

The discussion above indicates how the sources of the SIN data are selected in a single CAMAC dataway cycle. However, in the bit manipulation, handshaking kind of software operations for which the 8X300 is so powerful, it is essential to be able to set or clear a given status bit without affecting any of the other status lines. This was accomplished with the registers in 4A, 4B, 5A, 5B. These are 'LS74 RS flip-flops. To set the flip-flop, the commands N•A(1)•F(19) through N•A(7)•F(19) were used. Each "A" is gated to the appropriate logic. In the case of W1, when N•A(1)•F(19)•S1 is asserted 6A(8) goes low presetting the flip-flop and gives a one on 4A(5). Note that this bit has been set to a one without disturbing or checking the other bits. These bits are cleared in the same manner with N•A(1)•F(23) to N•A(7)•F(23). In the case of W1, N•F(23)•A(1)•S1' is ORed with WZ to give a low at 4A(1) to clear the flip-flop. WZ is OR'ed with the discrete clear lines so that the entire register can be cleared with a single command (WZ also clears the status input control group 2 register).

The zero or clear logic is the other function implemented or Q5693-11D. The basic concept is to generate a controlled clearing signal which would (i) respond to crate clear; (ii) clear only the registers and logic on our module; or (iii) clear discrete segments with specified signals. The logic is shown at the bottom of the drawing. A dataway Z or C clear the entire module and resets the 8X300 program counter (PC) to zero. These signals may also be generated exclusively in this module with a module zero (MZ) command N•A(0)•F(20). MZ is OR'ed with Z and C. A less drastic operation is the write zero (WZ) command N•A(1)•F(20) which clears the dataway group 1 write buffers (8B, 8C, 8D, 14B), the group 2 status register (4A, 4B, 5A, 5B) and the group 2 status control register (8A, 14A). In addition, the 8x300 PC is reset. The data zero command N•A(2)•F(20) clears only the group 1 write buffer and the PC, while the program counter zero (PZ) command N•A(3)•F(20) resets only the PC. WZ is obtained by ANDing [N•A(1)•F(20)]•S1 and ORing it with MZ. Thus, at 10H(8) the signal is WZ=MZ+[N•A(1)•F(20)]•S1. Data zero (DZ) is produced with a [N•F(20)•A(2)]•S1. The signal at 9E(11) is DZ=MZ+WZ+ [N•F(20)•A(2)]•S1. Program counter zero (PZ) is derived with a [N•F(20)•A(3)]•S1 to produce at 10H(11) PZ=MZ+WZ+DZ+ [N•F(23)•A(30]•S1. This command sequence provides a very versatile clearing arrangement which allows certain segments or combinations thereof to be asserted without clearing others.

VI. DIAGNOSTIC SOFTWARE

Early in the devlopment of the module it became clear that it would not be possible to manually verify the correct operation of all functions under all conditions because of the very large number of allowable operations. Therefore we have developed diagnostic software for both the host computer and the 8X300 module which test all of the implemented functions under all possible combinations of input data and control functions.

The concept of the diagnostic procedure is simple. A code for the 8X300 was written (Appendix B) which simulates a direct connection of the input data and status IV bytes to their corresponding output IV bytes. The code is written in 8X300 machine language described elsewhere,[11,13] and can be implemented using the SMS3000 microcontroller simulator.[16] The loop contains 17 instructions and executes in 4.25 μsec. This time is sufficiently short compared to the execution time of our register I/O CAMAC subroutine CFSA so that the 8X300 input and output IV bytes can be considered to be hardwired together on a bit by bit basis.

The host computer diagnostic program (Appendix C), written in top-down structured FORTRAN[22] and compiled with the SFORTRAN precompiler,[23] tests for correct operation of the module by setting up appropriate CAMAC functions (NAF) and outputing known data patterns to the various IV bytes. All CAMAC I/O is performed through the subroutine CFSA which has been written in assembly language specifically for our fast-response crate controllers which were obtained from Los Alamos Scientific Laboratory.[7] This routine meets the proposed specifications for CAMAC FORTRAN subroutines.[24] The 8X300 code (Appendix B) transfers the data from the four input IV bytes to their corresponding output IV bite and the resultant data are read back into the host. The expected pattern and the pattern received are compared; disagreement indicates a module fault.

Logically, the implemented CAMAC functions are tested in an order such that each operation which relies on another CAMAC function is tested only after the prerequisite function has been tested. The order of testing is: Read/Write lines; Status In/Status out data lines; LAM enable/disable; and Q-response.

VII. ACKNOWLEDGMENTS

We are indebted to R. T. Roseberry, P. A. Seeger and J. W. Woody, Jr. for valuable discussions in designing and implementing the microprocessor control module, and to Sharon P. King for help in developing the diagnostic software.

VIII. REFERENCES

1. Signetics Corp., 811 East Arques Ave., Sunnyvale, CA 94086
2. Hendricks, R. W., Trans. Am. Crystallogr. Assoc. 12, 103-146 (1976).
3. Hendricks, R. W., J. Appl. Crystallogr. 11, 15-30 (1978).
4. Scientific MicroSystems Inc., 520 Clyde Ave., Mountain View, CA 94043
5. Twichell, J. C. and Hendricks, R. W., (1978) Proceedings of Second Annual ModComp Users' Exchange, held in Ft. Lauderdale, FA. 6-8 December 1976 (Published by Modular Computer Systems, Inc., Ft. Lauderdale, FA).
6. Twichell, J. C. and Hendricks, R. W., (1979) to be published.
7. Seeger, P. A. (1976) "A Fast-Response CAMAC Crate Controller" Los Alamos Scientific Laboratory Report LA-6605-M, Los Alamos, NM 87545.
8. Taken (with permission) from Signetics 8X300 Programming Manual, p. 15.
9. Lau, S. Y. (1979) Electronic Design 4 (Feb. 15), 128-139.
10. Signetics 8X300 Reference Manual, Signetics Corp. Sunnyvale, CA.
11. Signetics 8X300 Programming Manual, Signetics Corp., Sunnyvale, CA.
12. SMS 300 Series Data Sheets, Scientific MicroSystems Inc., Mt. View, CA.
13. The Microcontroller Application Guide, Scientific MicroSystems, Inc., Mt. View, CA.
14. The Microcontroller System Description, Scientific MicroSystems Inc., Mt. View, CA.
15. MS 3300 Microcontroller Monitor, Scientific MicroSystem, Inc., Mt. View, CA.
16. MS 3000 Microcontroller Simulator, Scientific MicroSystems, Inc., Mt. View, CA.
17. Signetics Microcontroller Cross Assembly Program, Signetics Corp., Sunnyvale, CA.

18. U. S. NIM Committee (1976), CAMAC Tutorial Articles, Energy Research and Development Administration report TID-26618, Washington, D.C. 20545.

19. CAMAC Instrumentation and Interface Standards: Modular Instrumentation and Digital Interface System (CAMAC) IEEE Std. 583-1975; Serial Highway Interface System (CAMAC) IEEE Std. 595-1976; Parallel Highway Interface System (CAMAC) IEEE Std. 596-1976; Block Transfers in CAMAC System IEEE Std. 683-1976, The Institute of Electrical and Electronic Engineers, Inc. (New York, NY).

20. Standard Engineering Corp., 44800 Industrial Drive, Freemont, CA 94538.

21. Reference 18, page 85; Reference 19, IEEE Std. 683-1976.

22. Yourdon, E. (1975) Techniques of Program Structure and Design (Prentice Hall: Englewood Cliffs, NJ).

23. SFORTRAN was developed by Lars G. Mossberg of Volvo Flygmotor, AB, Trollhattan, Sweden and is distributed in the United States by Software Counsulting Services, 901 Whittier Dr., Allentown, PA 18103

24. Reference 18, p. 75.

APPENDIX A

DETAILED CHIP COORDINATE LAYOUT FOR INTERFACE BOARD

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ORNL-DWG 79-12934

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A		☒	☒	'LS04	'LS74	'LS74	'LS00	'LS08	'LS174	'LS08	'LS74	'LS74	'LS08	'LS08	'LS174	3-M READ DATA (1)
B		☒	☒	'LS04	'LS74	'LS74	'LS02		'LS174	'LS08	'LS74	'LS74	'LS08	'LS08	'LS174	3-M READ DATA (2)
C		☒	☒	'LS04	'LS04	'LS08	'LS02		'LS174	'LS08	'LS74	'LS74	'LS02	'LS32	'LS148	3-M READ DATA (3)
D		☒	☒	'LS04	'LS04	'LS00	7405		'LS174	'LS32	'LS74	'LS74	'LS02	'LS32	'LS153	3-M WRITE DATA (1)
E		☒	☒	'LS04	'LS04	'LS00	7405		'LS153	'LS32	7404	'LS08	7404	'LS08	'LS153	3-M WRITE DATA (2)
F		☒	☒	STATUS IN/OUT CONNECTOR	'LS02	'LS00	7405	'LS02	'LS153	'LS11	7404	'LS08	7404	7408	'LS157	3-M WRITE DATA (3)
G	74123	74123	74123				7401	'LS04	'LS157	7411	7404	'LS00	7404	7427	3-M STATUS OUT	3-M STATUS IN
H		☒	☒	☒	☒	☒	7401	'LS04		7408	7432	'LS00	7404	7408		

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

APPENDIX B

8X300 MICROCONTROLLER DIAGNOSTIC SOFTWARE

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

APPENDIX B

8X300 DIAGNOSTICS HANDLER

```
PC     CODE

00     607007        SELECT SIN (IV7)
01     027001        (IV7)→ R1
02     607006        SELECT SOUT  (IV6)
03     001027        (R1)→IV6
04     607020        SELECT W1-W8 (IV20)
05     027001        (IV20)→R1
06     607001        SELECT R1-R8 (IV1)
07     001027        (R1)→IV1
10     607005        SELECT W9-W16 (IV5)
11     027001        (IV5)→R1
12     607003        SELECT R9-R16 (IV3)
13     001027        (R1)→IV3
14     607004        SELECT W17-W24 (IV4)
15     027001        (IV4)→R1
16     607002        SELECT R17-R24 (IV2)
17     002027        (R1)→IV2
20     700000        BRU START
```

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

APPENDIX C

HOST COMPUTER DIAGNOSTIC SOFTWARE

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

```
      PROGRAM SMS
C***  VERSION B.00 *** FEB 19,1979
      DIMENSION JDATA(2),KDATA(2),LDATA(2)
      LOGICAL ID,IS,BIT,TESTB,REPEAT,BIT1
      INTEGER CDREGF,BOOL,LAM,ANS,YES
      INTEGER RD1,RD2,RC1,TLM,CLM,WT1,WT2,SS2,F20,SC2,DIS,ENB
      DATA RD1/0/,RD2/1/,RC1/2/,TLM/8/,CLM/10/,WT1/16/,WT2/17/,
     1     SS2/19/,F20/20/,SC2/23/,DIS/24/,ENB/26/
      DATA YES/'YE'/
      DATA LDATA/2*0/
C...
C... INNITIALLIZE CRATE
C...
      ID=.FALSE.
      IS=.FALSE.
      REPEAT=.TRUE.
      IB=0
      WRITE(6,8001)
      READ(5,8002) N
      CALL CMCFNC('Z',ID,IS)
C...
C...  TEST READ/WRITE LINES
C...
      IA=0
      LREG=CDREGF(IB,N,IA)
      WHILE(REPEAT) DO
         IERR=0
         JDATA(1)=0
         JDATA(2)=-32767
         WHILE(JDATA(2).LE.32766) DO
            IF(JDATA(1).GE.256) JDATA(1)=1
            CALL CFSA(WT1,LREG,JDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            CALL CFSA(RD1,LREG,KDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            IF((JDATA(1).NE.KDATA(1)).OR.
     1      (JDATA(2).NE.KDATA(2))) THEN
               WRITE(6,8010) J,JDATA,KDATA
               IERR=IERR+1
            ENDIF
            JDATA(1)=JDATA(1)+1
            JDATA(2)=JDATA(2)+1
         ENDDO
         IF(IERR.EQ.0) THEN
            WRITE(6,8015)
         ELSE
            WRITE(6,8020) IERR
         ENDIF
         WRITE(6,8005)
         READ(5,8006) ANS
         IF(ANS.EQ.YES) THEN
            REPEAT=.TRUE.
         ELSE
            REPEAT=.FALSE.
         ENDIF
      ENDDO
      REPEAT=.TRUE.
C...
C...  TEST STATUS CONTROL FUNCTIONS (PHASE 1)
C...  LOAD SIN REGISTER
```

```
C...
      IA=1
      LREG2=CDREGF(IB,N,IA)
      WHILE(REPEAT) DO
         JDATA(1)=0
         JDATA(2)=0
         CALL CFSA(WT2,LREG2,JDATA,ISTAT,JERR)
         IF(JERR.NE.0) WRITE(6,8000) ISTAT
         IERR=0
         J=1
         WHILE(J.LE.128) DO
            CALL CFSA(WT2,LREG,JDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            CALL CFSA(RD2,LREG,KDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            CALL SETBT(KDATA(2),1)
            CALL SETBT(KDATA(2),9)
            ITRANS=ISHFT(KDATA(2),-8)
            K=2
            WHILE(K.LE.8) DO
               CALL SETBT(KDATA(2),K)
               K=K+1
            ENDDO
            IREC=KDATA(2)
            IF(ITRANS.NE.IREC.OR.IREC.NE.JDATA(2)) THEN
               WRITE(6,8030) JDATA(2),ITRANS,IREC
               IERR=IERR+1
            ENDIF
            JDATA(2)=JDATA(2)+1
            J=J+1
         ENDDO
         IF(IERR.EQ.0) THEN
            WRITE(6,8035)
         ELSE
            WRITE(6,8020) IERR
         ENDIF
         WRITE(6,8005)
         READ(5,8006) ANS
         IF(ANS.EQ.YES) THEN
            REPEAT=.TRUE.
         ELSE
            REPEAT=.FALSE.
         ENDIF
      ENDDO
      REPEAT=.TRUE.
C...
C... TEST STATUS CONTROL FUNCTIONS (PHASE 2)
C...  SELECTIVE INPUT FUNCTIONS
C...
      WHILE(REPEAT) DO
         JDATA(2)=0
         CALL CFSA(WT2,LREG,JDATA,ISTAT,JERR)
         IF(JERR.NE.0) WRITE(6,8000) ISTAT
         IERR=0
         J=1
         WHILE(J.LE.127) DO
            K=1
            WHILE(K.LE.7) DO
               LREG1=CDREGF(IB,N,K)
               KTST=17-K
               BIT=TESTB(J,KTST)
               IF(BIT) THEN
                  CALL CFSA(SC2,LREG1,JDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ELSE
                  CALL CFSA(SS2,LREG1,JDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ENDIF
               K=K+1
            ENDDO
            CALL CFSA(RD2,LREG,KDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            CALL SETBT(KDATA(2),1)
            CALL SETBT(KDATA(2),9)
            ITRANS=ISHFT(KDATA(2),-8)
            L=2
            WHILE(L.LE.8) DO
               CALL SETBT(KDATA(2),L)
               L=L+1
            ENDDO
            IREC=KDATA(2)
            IF(J.NE.IREC.OR.J.NE.ITRANS) THEN
               IERR=IERR+1
               WRITE(6,8040) J,ITRANS,IREC
            ENDIF
            J=J+1
         ENDDO
         IF(IERR.EQ.0) THEN
            WRITE(6,8045)
         ELSE
            WRITE(6,8020) IERR
         ENDIF
         WRITE(6,8005)
         READ(5,8006) ANS
         IF(ANS.EQ.YES) THEN
            REPEAT=.TRUE.
         ELSE
            REPEAT=.FALSE.
         ENDIF
      ENDDO
      REPEAT=.TRUE.
C...
C...  TEST LAMS (PHASE 1)
C...  SELECTIVE LAM ENABLES
C...
      WHILE(REPEAT) DO
         IERR=0
         JDATA(2)=0
         J=0
         WHILE(J.LE.127) DO
            K=1
            WHILE(K.LE.7) DO
               LREG1=CDREGF(IB,N,K)
               KTST=17-K
               BIT=TESTB(J,KTST)
               IF(BIT) THEN
                  CALL CFSA(DIS,LREG1,JDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ELSE
                  CALL CFSA(ENB,LREG1,JDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ENDIF
               K=K+1
            ENDDO
            JDATA(2)=0
            K=0
            WHILE(K.LE.127) DO
               CALL CFSA(WT2,LREG,JDATA,ISTAT,JERR)
               IF(JERR.NE.0) WRITE(6,8000) ISTAT
               CALL CFSA(RD2,LREG2,KDATA,ISTAT,JERR)
               IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ICOMP=BOOL(J,K,8,16)
               IF(ICOMP.NE.KDATA(2)) THEN
                  WRITE(6,8050) J,JDATA(2),KDATA(2),ICOMP
                  IERR=IERR+1
               ENDIF
               CALL CFSA(WT2,LREG,LDATA,ISTAT,JERR)
               IF(JERR.NE.0) WRITE(6,8000) ISTAT
               L=1
               WHILE(L.LE.7) DO
                  LREG1=CDREGF(IB,N,L)
                  CALL CFSA(CLM,LREG1,LDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
                  L=L+1
               ENDDO
               JDATA(2)=JDATA(2)+1
               K=K+1
            ENDDO
            J=J+1
         ENDDO
         IF(IERR.EQ.0) THEN
            WRITE(6,8055)
         ELSE
            WRITE(6,8020) IERR
         ENDIF
         WRITE(6,8005)
         READ(5,8006) ANS
         IF(ANS.EQ.YES) THEN
            REPEAT=.TRUE.
         ELSE
            REPEAT=.FALSE.
         ENDIF
      ENDDO
      REPEAT=.TRUE.
C...
C...  TEST LAMS (PHASE 2)
C...  SELECTIVE CLEAR SOUT
C...
      WHILE(REPEAT) DO
         JDATA(2)=0
         J=1
         WHILE (J.LE.7) DO
            LREG1=CDREGF(IB,N,J)
            CALL CFSA(ENB,LREG1,JDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            J=J+1
         ENDDO
         IERR=0
         J=0
         WHILE(J.LE.127) DO
            JDATA(2)=0
            CALL CFSA(WT2,LREG,JDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            K=1
            WHILE(K.LE.7) DO
               LREG1=CDREGF(IB,N,K)
               CALL CFSA(CLM,LREG1,JDATA,ISTAT,JERR)
               IF(JERR.NE.0) WRITE(6,8000) ISTAT
               K=K+1
            ENDDO
            JDATA(2)-127
            CALL CFSA(WT2,LREG,JDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            K=1
            WHILE(K.LE.7) DO
               LREG1=CDREGF(IB,N,K)
               KTST=17-K
               BIT=TESTB(J,KTST)
               IF(BIT) THEN
                  CALL CFSA(CLM,LREG1,JDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ENDIF
               K=K+1
            ENDDO
            CALL CFSA(RD2,LREG2,KDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            CALL SETBT(KDATA(2),9)
            IF(J.NE.KDATA(2)) THEN
               IERR=IERR+1
               WRITE(6,8060) J,KDATA(2)
            ENDIF
            J=J+1
         ENDDO
         IF(IERR.EQ.0) THEN
            WRITE(6,8065)
         ELSE
            WRITE(6,8020) IERR
         ENDIF
         WRITE(6,8005)
         READ(5,8006) ANS
         IF(ANS.EQ.YES) THEN
            REPEAT=.TRUE.
         ELSE
            REPEAT=.FALSE.
         ENDIF
      ENDDO
      REPEAT=.TRUE.
C...
C... TEST LAMS (PHASE 3)
C...  Q-RESPONSE MODE
C...
      CALL CMCFNC('RI',ID,IS)
      WHILE(REPEAT) DO
         IERR=0
         JDATA(2)=0
         J=0
         WHILE(J.LE.127) DO
            K=1
            WHILE(K.LE.7) DO
               LREG1=CDREGF(IB,N,K)
               KTST=17-K
               BIT=TESTB(J,KTST)
               IF(BIT) THEN
                  CALL CFSA(DIS,LREG1,JDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ELSE
                  CALL CFSA(ENB,LREG1,JDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
               ENDIF
               K=K+1
            ENDDO
            JDATA(2)=0
            K=0
            WHILE(K.LE.127) DO
               CALL CFSA(WT2,LREG,JDATA,ISTAT,JERR)
               IF(JERR.NE.0) WRITE(6,8000) ISTAT
               CALL CFSA(TLM,LREG,KDATA,ISTAT,JERR)
               IF(JERR.NE.0) WRITE(6,8000) ISTAT
               BIT=TESTB(ISTAT,11)
               ICOMP=BOOL(J,K,8,16)
               IF(((ICOMP.EQ.0).AND.BIT).OR.
     1         ((ICOMP.NE.0).AND..NOT.BIT)) THEN
                  WRITE(6,8070) J,K,ISTAT,ICOMP
                  IERR=IERR+1
               ENDIF
               CALL CFSA(WT2,LREG,LDATA,ISTAT,JERR)
               IF(JERR.NE.0) WRITE(6,8000) ISTAT
               L=1
               WHILE(L.LE.7) DO
                  LREG1=CDREGF(IB,N,L)
                  CALL CFSA(CLM,LREG1,LDATA,ISTAT,JERR)
                  IF(JERR.NE.0) WRITE(6,8000) ISTAT
                  L=L+1
               ENDDO
               JDATA(2)=JDATA(2)+1
               K=K+1
            ENDDO
            J=J+1
         ENDDO
         IF(IERR.EQ.0) THEN
            WRITE(6,8075)
         ELSE
            WRITE(6,8020) IERR
         ENDIF
         WRITE(6,8005)
         READ(5,8006) ANS
         IF(ANS.EQ.YES) THEN
            REPEAT=.TRUE.
         ELSE
            REPEAT=.FALSE.
         ENDIF
      ENDDO
      REPEAT=.TRUE.
C...
C...  TEST SOUT Q-RESPONSE
C...
      IA=8
      LREG1=CDREGF(IB,N,IA)
      WHILE(REPEAT) DO
         JDATA(2)=0
         CALL CFSA(ENB,LREG1,JDATA,ISTAT,JERR)
         IF(JERR.NE.0) WRITE(6,8000) ISTAT
         IERR=0
         J=0
         WHILE(J.LE.255) DO
            CALL CFSA(WT2,LREG,JDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            CALL CFSA(TLM,LREG2,JDATA,ISTAT,JERR)
            IF(JERR.NE.0) WRITE(6,8000) ISTAT
            BIT=TESTB(ISTAT,11)
            BIT1=TESTB(J,9)
            IF((BIT.AND.BIT1).OR.(.NOT.BIT.AND..NOT.BIT1)) THEN
               WRITE(6,8080) J,KDATA(2),ISTAT
               IERR=IERR+1
            ENDIF
            JDATA(2)=JDATA(2)+1
            J=J+1
         ENDDO
         IF(IERR.EQ.0) THEN
            WRITE(6,8075)
         ELSE
            WRITE(6,8020) IERR
         ENDIF
         WRITE(6,8005)
         READ(5,8006) ANS
         IF(ANS.EQ.YES) THEN
            REPEAT=.TRUE.
         ELSE
            REPEAT=.FALSE.
         ENDIF
      ENDDO
      REPEAT=.TRUE.
      STOP
8000 FORMAT(1H ,'CAMAC STATUS ERROR'/1H ,2Z4)
8001 FORMAT(1H ,'WHERE IS SMS? (I2)')
8002 FORMAT(I2)
8005 FORMAT(1H ,'REPEAT?')
8006 FORMAT(A2)
8010 FORMAT(1H ,'I/O ERROR'/1H ,3(2Z4,2X))
8015 FORMAT(1H ,'I/O PHASE ERROR FREE')
8020 FORMAT(1H ,I5,2X,'ERRORS DETECTED')
8025 FORMAT(1H ,'SIN CONTROL ERROR',1H ,2Z4)
8030 FORMAT(1H ,'SIN LOAD ERROR'/1H ,3(2Z4,2X))
8035 FORMAT(1H ,'STATUS CONTROL FUNCTIONS (PHASE 1) ERROR FREE')
8040 FORMAT(1H ,'STATUS SELECTIVE SET ERROR'/1H ,3(2Z4,2X))
8045 FORMAT(1H ,'STATUS CONTROL FUNCTIONS (PHASE 2) ERROR-FREE')
8050 FORMAT(1H ,'LAM ERROR (PHASE 1)'/1H ,4(2Z4,2X))
8055 FORMAT(1H ,'LAM TEST (PHASE 1) ERROR-FREE')
8060 FORMAT(1H ,'SOUT SELECTIVE CLEAR ERROR'/1H ,2(2Z4,2X))
8065 FORMAT(1H ,'LAM TEST (PHASE 2) ERROR-FREE')
8070 FORMAT(1H ,'LAM ERROR (PHASE 3)'/1H ,4(2Z4,2X))
8075 FORMAT(1H ,'LAM TEST (PHASE 3) ERROR-FREE')
8080 FORMAT(1H ,'SOUT Q-RESPONSE ERROR'/1H ,3(2Z4,2X))
8085 FORMAT(1H ,'SOUT Q-RESPONSE ERROR-FREE')
      END
```

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

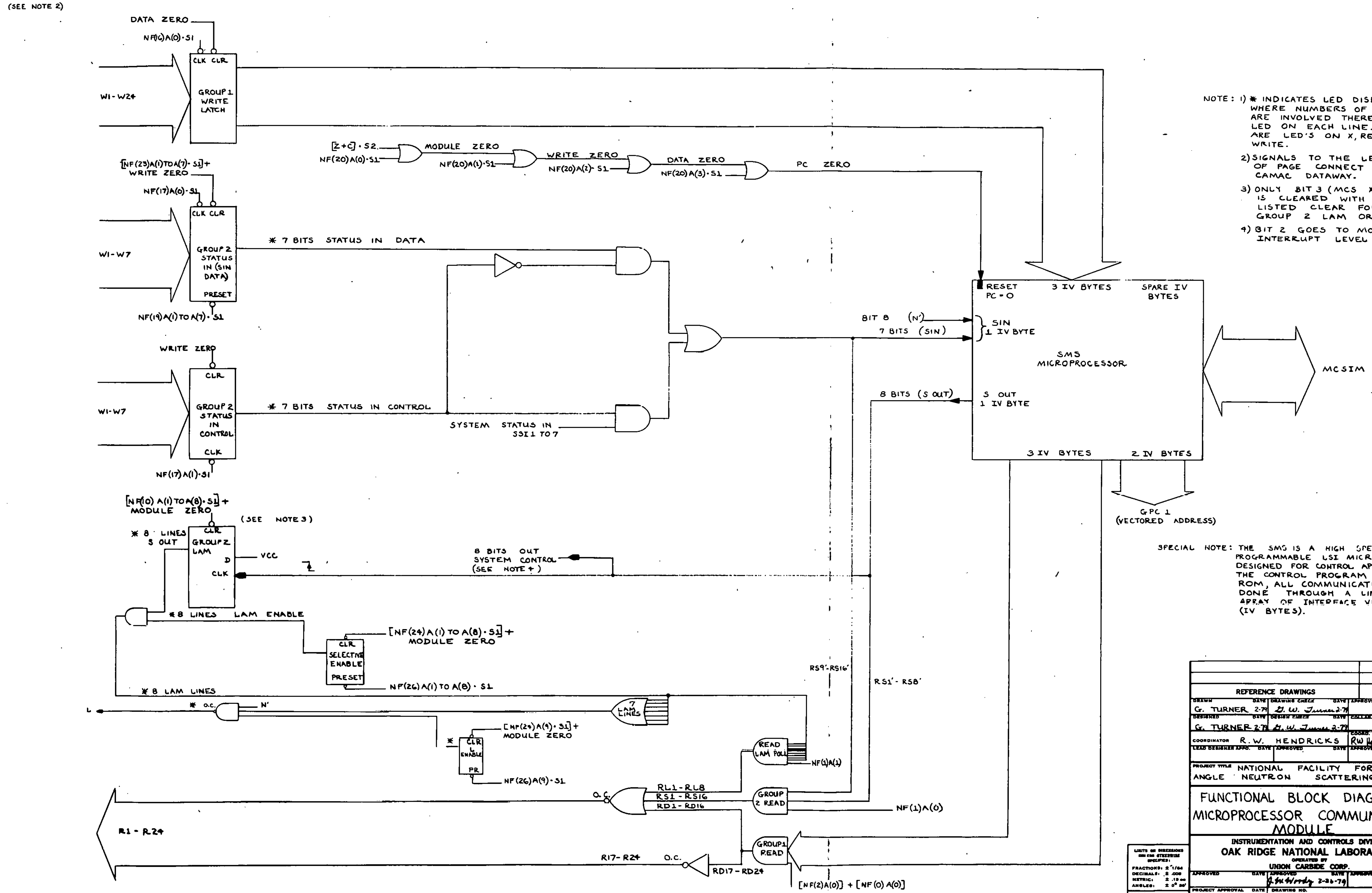
APPENDIX D

CIRCUIT DIAGRAMS

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK



NOTE: 1) * INDICATES LED DISPLAY, WHERE NUMBERS OF LINES ARE INVOLVED THERE IS AN LED ON EACH LINE. THERE ARE LED'S ON X, READ, AND WRITE.

2) SIGNALS TO THE LEFT EDGE OF PAGE CONNECT TO THE CAMAC DATAWAY.

3) ONLY BIT 3 (MCS XFR REQ) IS CLEARED WITH THE LISTED CLEAR FOR THE GROUP 2 LAM OR NF(2) A(0).

4) BIT 2 GOES TO MOD COMP INTERRUPT LEVEL 7.

SPECIAL NOTE: THE SMS IS A HIGH SPEED PROGRAMMABLE LSI MICROPROCESSOR DESIGNED FOR CONTROL APPLICATIONS. THE CONTROL PROGRAM IS ON A ROM, ALL COMMUNICATION IS DONE THROUGH A LINEAR ARRAY OF INTERFACE VECTORS (IV BYTES).

REVISIONS

1 OF 4

REFERENCE DRAWINGS | DWG. NO.

DRAWN G. TURNER DATE 2-79 | DRAWING CHECK G. W. Turner 2-79 | APPROVED DATE

DESIGNED G. TURNER DATE 2-79 | DESIGN CHECK G. W. Turner 2-79 | COLLAB. APPD. DATE

COORDINATOR R. W. HENDRICKS | COORD. APPD. RW Hendricks DATE 2-1-79

LEAD DESIGNER APPD. DATE | APPROVED DATE | APPROVED DATE

PROJECT TITLE NATIONAL FACILITY FOR SMALL ANGLE NEUTRON SCATTERING

FUNCTIONAL BLOCK DIAGRAM MICROPROCESSOR COMMUNICATION MODULE

INSTRUMENTATION AND CONTROLS DIVISION
OAK RIDGE NATIONAL LABORATORY
OPERATED BY
UNION CARBIDE CORP.

LIMITS ON DIMENSIONS UNLESS OTHERWISE SPECIFIED:
FRACTIONS: ± 1/64
DECIMALS: ± .005
METRIC: ± .13 mm
ANGLES: ± 0° 30'
SCALE:

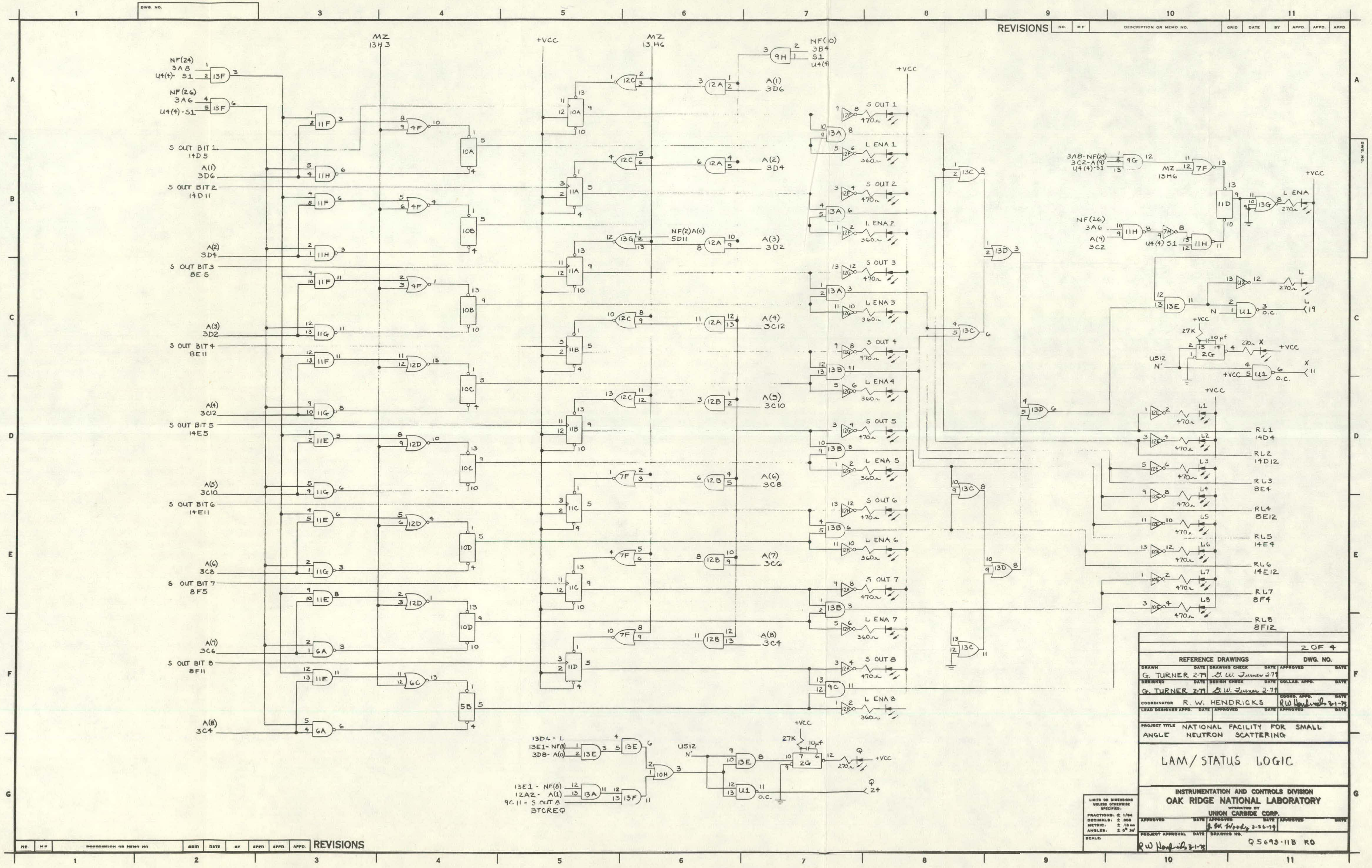
APPROVED DATE | APPROVED J. M. Woody DATE 2-26-79 | APPROVED DATE

PROJECT APPROVAL RW Hendricks DATE 2-79 | DRAWING NO. Q5693-11A R0

THIS PAGE

WAS INTENTIONALLY

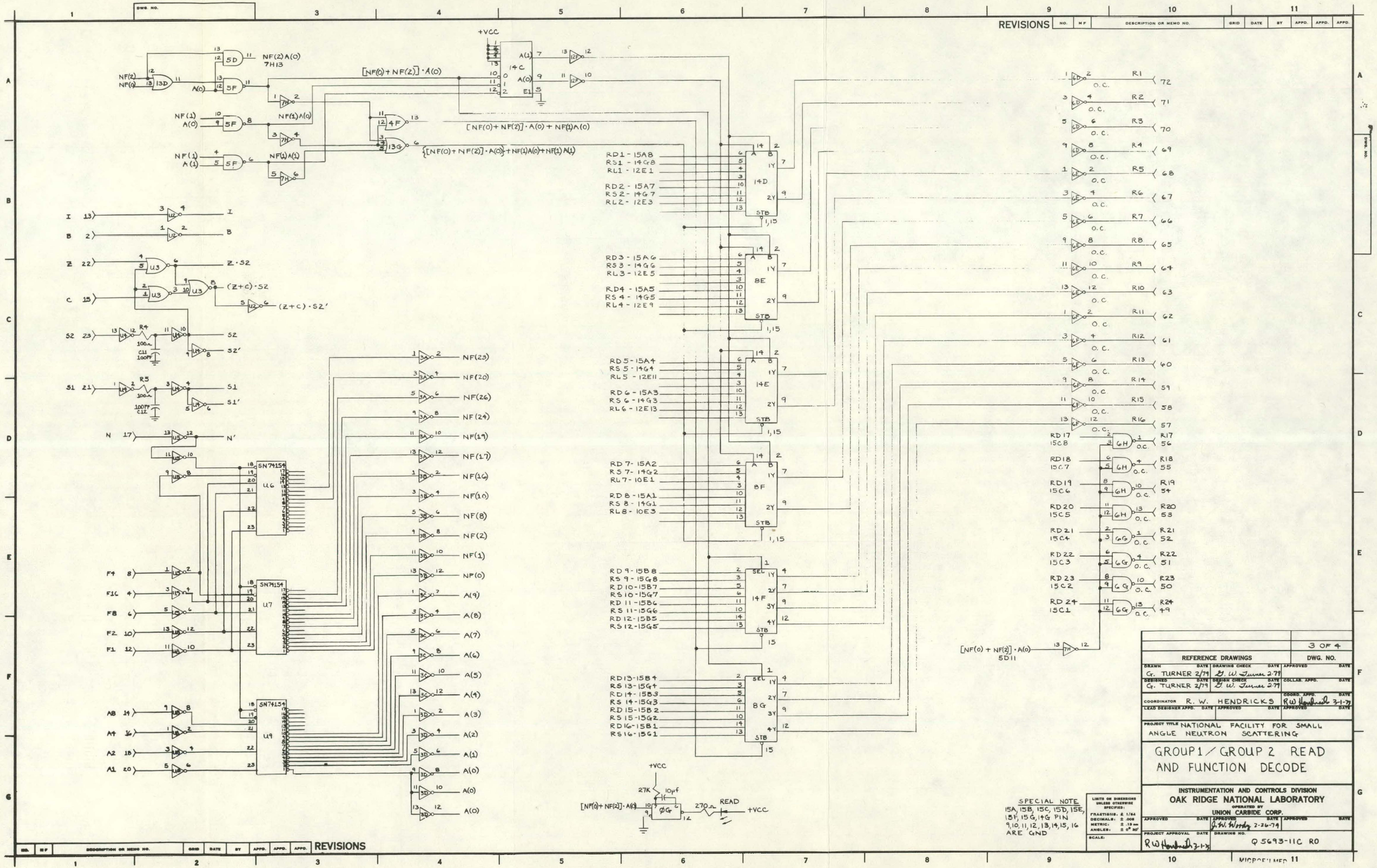
LEFT BLANK


REVISIONS
NF(24) 3A8 U4(4)-S1
NF(26) 3A6 U4(4)-S1
S OUT BIT 1 14D5
A(1) 3D6
S OUT BIT 2 14D11
A(2) 3D4
S OUT BIT 3 8E5
A(3) 3D2
S OUT BIT 4 8E11
A(4) 3C12
S OUT BIT 5 14E5
A(5) 3C10
S OUT BIT 6 14E11
A(6) 3C8
S OUT BIT 7 8F5
A(7) 3C6
S OUT BIT 8 8F11
A(8) 3C4
MZ 13H3
MZ 13H6
+VCC
NF(10) 3B4 S1 U4(4)
NF(2)A(0) 5D11
S OUT 1
L ENA 1
S OUT 2
L ENA 2
S OUT 3
L ENA 3
S OUT 4
L ENA 4
S OUT 5
L ENA 5
S OUT 6
L ENA 6
S OUT 7
L ENA 7
S OUT 8
L ENA 8
470Ω
360Ω
3A8-NF(24) 3C2-A(4) U4(4)-S1
NF(26) 3A6
A(4) 3C2
L ENA
270Ω
L
L 19
X
X 11
27K
10μF
U5 12 N'
RL1 14D4
RL2 14D12
RL3 8E4
RL4 8E12
RL5 14E4
RL6 14E12
RL7 8F4
RL8 8F12
L1
L2
L3
L4
L5
L6
L7
L8
13D4-1
13E1-NF(8)
3D8-A(0)
13E1-NF(8)
12A2-A(1)
9C11-S OUT 8
BTCREQ
Q
Q 24
O.C.
2 OF 4
REFERENCE DRAWINGS
DWG. NO.
DRAWN DATE G. TURNER 2-79
DRAWING CHECK DATE G. W. Turner 2-79
APPROVED DATE
DESIGNED DATE G. TURNER 2-79
DESIGN CHECK DATE G. W. Turner 2-79
COLLAB. APPD. DATE
COORDINATOR R. W. HENDRICKS
COORD. APPD. DATE RW Hendricks 3-1-79
LEAD DESIGNER APPD. DATE
APPROVED DATE
APPROVED DATE
PROJECT TITLE NATIONAL FACILITY FOR SMALL ANGLE NEUTRON SCATTERING
LAM/STATUS LOGIC
INSTRUMENTATION AND CONTROLS DIVISION
OAK RIDGE NATIONAL LABORATORY
OPERATED BY
UNION CARBIDE CORP.
LIMITS ON DIMENSIONS UNLESS OTHERWISE SPECIFIED:
FRACTIONS: ± 1/64
DECIMALS: ± .005
METRIC: ± .13 mm
ANGLES: ± 0° 30'
SCALE:
APPROVED DATE
APPROVED DATE J. W. Woody 2-26-79
APPROVED DATE
PROJECT APPROVAL DATE RW Hendricks 3-1-79
DRAWING NO. Q5693-11B R0
1 2 3 4 5 6 7 8 9 10 11
A B C D E F G

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK



SPECIAL NOTE
15A, 15B, 15C, 15D, 15E, 15F, 15G, 14G PIN 9, 10, 11, 12, 13, 14, 15, 16 ARE GND

LIMITS ON DIMENSIONS UNLESS OTHERWISE SPECIFIED:
FRACTIONS: ± 1/64
DECIMALS: ± .005
METRIC: ± .13 mm
ANGLES: ± 0° 30'
SCALE:

3 OF 4

REFERENCE DRAWINGS	DWG. NO.

DRAWN DATE	DRAWING CHECK DATE	APPROVED DATE
G. TURNER 2/79	G. W. Turner 2-79	
DESIGNED DATE	DESIGN CHECK DATE	COLLAB. APPD. DATE
G. TURNER 2/79	G. W. Turner 2-79	
COORDINATOR R. W. HENDRICKS		COORD. APPD. DATE RW Hendricks 3-1-79
LEAD DESIGNER APPD. DATE	APPROVED DATE	APPROVED DATE

PROJECT TITLE NATIONAL FACILITY FOR SMALL ANGLE NEUTRON SCATTERING

GROUP1 / GROUP 2 READ AND FUNCTION DECODE

INSTRUMENTATION AND CONTROLS DIVISION
OAK RIDGE NATIONAL LABORATORY
OPERATED BY
UNION CARBIDE CORP.

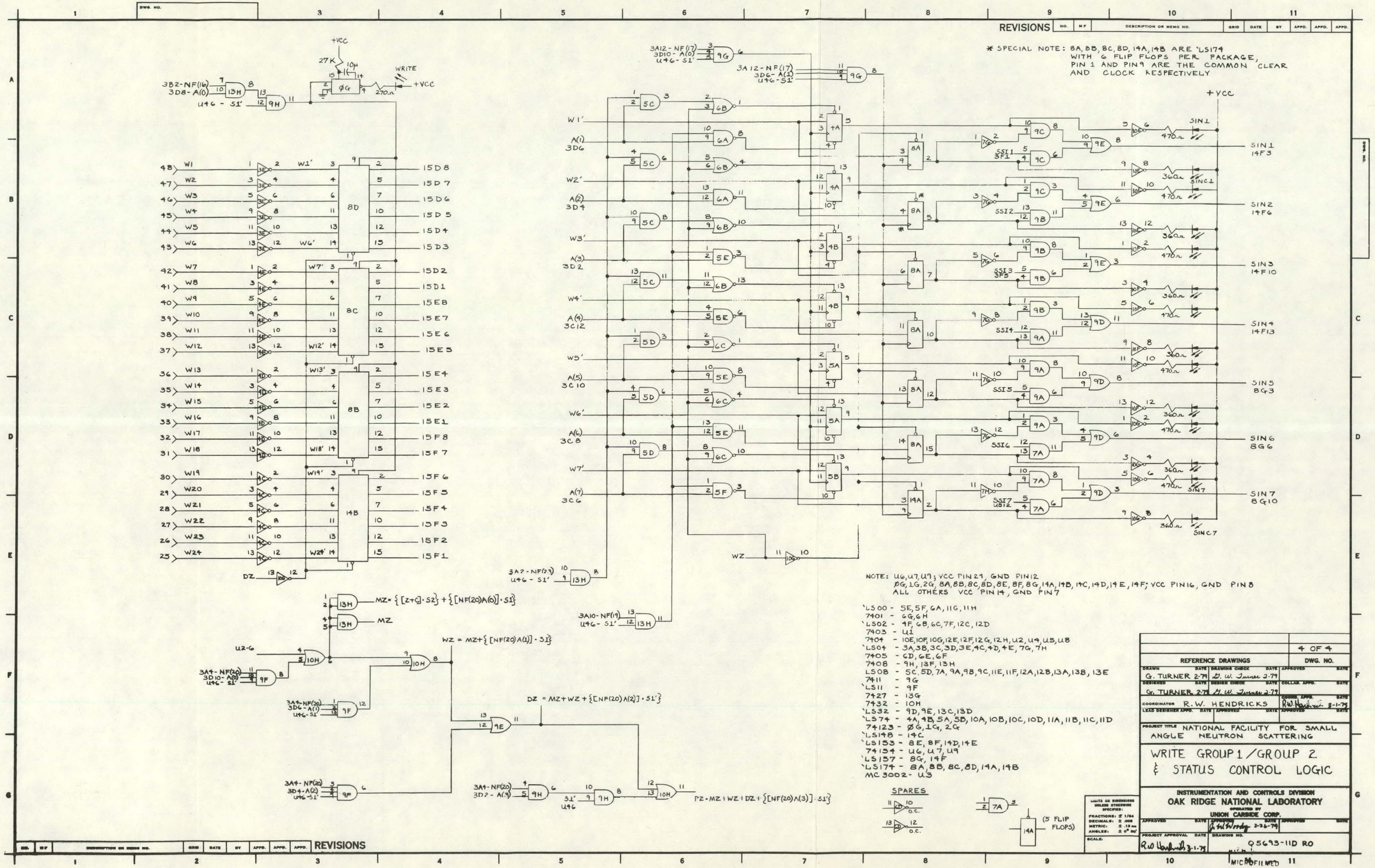
APPROVED DATE	APPROVED DATE	APPROVED DATE
	J. W. Woody 2-26-79	
PROJECT APPROVAL DATE	DRAWING NO.	
RW Hendricks 3-1-79	Q 5693-11C R0	

MICROFILMED

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK



* SPECIAL NOTE: 8A, 8B, 8C, 8D, 14A, 14B ARE 'LS174
WITH 6 FLIP FLOPS PER PACKAGE,
PIN 1 AND PIN9 ARE THE COMMON CLEAR
AND CLOCK RESPECTIVELY

NOTE: U6, U7, U9; VCC PIN24, GND PIN12
ØG, 1G, 2G, 8A, 8B, 8C, 8D, 8E, 8F, 8G, 14A, 14B, 14C, 14D, 14E, 14F; VCC PIN16, GND PIN8
ALL OTHERS VCC PIN14, GND PIN7

'LS00 - 5E, 5F, 6A, 11G, 11H
7401 - 6G, 6H
'LS02 - 4F, 6B, 6C, 7F, 12C, 12D
7403 - U1
7404 - 10E, 10F, 10G, 12E, 12F, 12G, 12H, U2, U4, U5, U8
'LS04 - 3A, 3B, 3C, 3D, 3E, 4C, 4D, 4E, 7G, 7H
7405 - 6D, 6E, 6F
7408 - 9H, 13F, 13H
LS08 - 5C, 5D, 7A, 9A, 9B, 9C, 11E, 11F, 12A, 12B, 13A, 13B, 13E
7411 - 9G
'LS11 - 9F
7427 - 13G
7432 - 10H
'LS32 - 9D, 9E, 13C, 13D
'LS74 - 4A, 4B, 5A, 5B, 10A, 10B, 10C, 10D, 11A, 11B, 11C, 11D
74123 - ØG, 1G, 2G
'LS148 - 14C
'LS153 - 8E, 8F, 14D, 14E
74154 - U6, U7, U9
'LS157 - 8G, 14F
'LS174 - 8A, 8B, 8C, 8D, 14A, 14B
MC3002 - U3

SPARES

(5 FLIP FLOPS)

LIMITS ON DIMENSIONS UNLESS OTHERWISE SPECIFIED:
FRACTIONS: ± 1/64
DECIMALS: ± .005
METRIC: ± .13 mm
ANGLES: ± 0° 30'
SCALE:

	4 OF 4
REFERENCE DRAWINGS	DWG. NO.

DRAWN DATE	DRAWING CHECK DATE	APPROVED DATE
G. TURNER 2-79	G. W. Turner 2-79	
DESIGNED DATE	DESIGN CHECK DATE	COLLAB. APPD. DATE
G. TURNER 2-79	G. W. Turner 2-79	
COORDINATOR R.W. HENDRICKS		COORD. APPD. DATE R.W. Hendricks 3-1-79
LEAD DESIGNER APPD. DATE	APPROVED DATE	APPROVED DATE

PROJECT TITLE NATIONAL FACILITY FOR SMALL ANGLE NEUTRON SCATTERING

WRITE GROUP 1 / GROUP 2
& STATUS CONTROL LOGIC

INSTRUMENTATION AND CONTROLS DIVISION
OAK RIDGE NATIONAL LABORATORY
OPERATED BY
UNION CARBIDE CORP.

APPROVED DATE	APPROVED DATE	APPROVED DATE
	J. W. Woody 2-26-79	
PROJECT APPROVAL DATE	DRAWING NO.	
R.W. Hendricks 3-1-79	Q5693-11D R0	

MICROFILMED

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

INTERNAL DISTRIBUTION

1-2. Central Research Library
3. Document Reference Section
4-5. Laboratory Records Department
6. Laboratory Records, ORNL - RC
7. ORNL Patent Office
8. J. A. Biggerstaff
9. B. S. Borie
10. H. R. Child
11-61. R. W. Hendricks
62. D. Hensley
63. H. N. Hill
64. M. R. Hill
65. J. A. Keathley
66. S. P. King
67. W. C. Koehler
68. M. K. Kopp
69. J. S. Lin
70. J. L. Lovvorn
71. C. J. McHargue
72. E. Madden
73. C. D. Martin
74 H. Postma
75. R. T. Roseberry
76. C. T. Stansberry
77. C. J. Sparks
78. J. O. Stiegler
79. D. B. Trauger
80-90. G. W. Turner
91. R. G. Upton
92. J. R. Weir, Jr.
93. G. D. Wignall
94. M. K. Wilkinson
95. J. W. Woody
96. H. L. Yakel
97 R. E. Zedler
98. A. Zucker

EXTERNAL DISTRIBUTION

99. B. W. Batterman, Division of Applied Physics, Cornell University, Ithaca, NY 14853
100. G. Bauer, Institut für Festkörperforschung der Kernforschungsanlage, 517 Jülich, West Germany
101. H. Brumberger, Department of Chemistry, Syracuse University, Syracuse, NY 13200
102. P. J. Flory, Department of Chemistry, Stanford University, Stanford, CA 11973

103. T. Hashimoto, Department of Polymer Chemistry, Kyoto University, Kyoto 606, Japan
104. L. C. Ianniello, Department of Materials Sciences, Department of Energy, Washington, DC 20545
105. R. Kurz, Institut für Elektronik, Kernforschungsanlage Jülich, 517 Jülich, West Germany
106. S. Y. Lau, Signetics Corp., Sunnyvale, CA 94086
107. M. Liccardo, Scientific MicroSystems Inc., 520 Clyde Ave., Mountain View, CA 94043
108. D. McMillan, P-9, Los Alamos Scientific Laboratory, Los Alamos, NM 87545
109. D. L. Mitchell, National Science Foundation, 1800 G Street, N.W., Washington, DC 20550
110. L. H. Nosanow, National Science Foundation, 1800 G Street, N.W., Washington, DC 20550
111. J. Schelten, Institut für Festkörperforschung der Kernforschungsanlage, Jülich, 517 Jülich, Germany
112. B. P. Schoenborn, Biology Division, Brookhaven National Laboratory, Upton, NY 11973
113. C. G. Shull, Massachusetts Institute of Technology, Cambridge, MA 02139
114. P. A. Seeger, P-8, Los Alamos Scientific Laboratory, Los Alamos, NM 87545
115. D. K. Stevens, Division of Materials Sciences, Department of Energy, Washington, DC 20545
116. J. West, Daresbury Laboratory, Keckwick Lane, Warrington WAY 4AD, England
117-143. Technical Information Center
144. DOE-Assistant Manager, Oak Ridge Operations, Oak Ridge, TN

☆U.S. GOVERNMENT PRINTING OFFICE: 1979-640-079/226