AD-A078 391 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66157 SBIE-AD-E100 313 NL

1 OF 5
AD A078391

Book 1

November 1979

SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM)

for

THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

Burroughs Corporation
Federal and Special Systems Group
Paoli, Pa. 19301

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 66157	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Software Maintenance Manual for the Modular System Control Development Model (MSCDM) Book 1		5. TYPE OF REPORT & PERIOD COVERED FINAL Sep 76 - Nov 79 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DCA100-76-C-0083
9. PERFORMING ORGANIZATION NAME AND ADDRESS Burroughs Corporation Federal and Special Systems Group Paoli, PA 19301		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 33126 T&CCP 3012 - Task 15203
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency Defense Communications Engineering Center 1860 Wiehle Ave., Reston, VA 22090		12. REPORT DATE November 1979 13. NUMBER OF PAGES 391
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Program Listings
Loop Network
Ring Network
Modular Architecture
DCS System Control
Distributed Computer System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Distributed Master Control Program (DMCP) and the technical control application programs of the Modular System Control Development Model (MSCDM) are described. Program listings of each program and subprogram are presented. Programs are written in FORTRAN and MACRO-11 for Digital Equipment Corporation (DEC) LSI-11 microcomputers.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Book 1

(6) SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVELOPMENT MODEL (MSCDM). Book 1.

(9) Final rept. Sep 76 - Nov 79.

for

THE DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

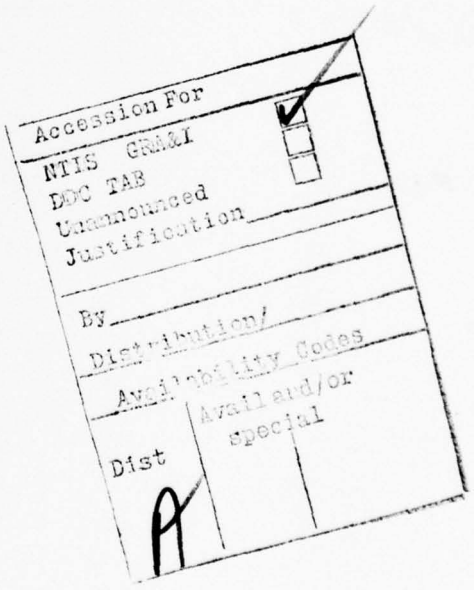
(15) DCA100-76-C-0083

Burroughs Corporation
Federal and Special Systems Group
Paoli, Pa. 19301

4/B

Foreword

This publication is the Software Maintenance Manual for the Modular System Control Development Model (MSCDM). This manual was prepared by the Burroughs Corporation and is submitted in accordance with the requirements of Contract DCA 100-76-C-0083.



MSCDM SOFTWARE MAINTENANCE MANUAL

Book 1		PAGE
	INTRODUCTION	1
1.0	MSCDM DMCP AND APPLICATION PROGRAMS	3
1.1	DESCRIPTION OF DMCP	4
1.2	SSCI NODE 21	18
1.3	VSQC NODE 22	51
1.4	DSQC NODE 23	93
1.5	DBMS NODE 24	133
1.6	OCRI NODE 25	164
1.7	BWBSA NODE 26	198
1.8	FIAC NODE 27	238
1.9	SDCA NODE 28	279
1.10	USER LANGUAGE	320
1.11	SIG / SDCA - 11/40 PROGRAM	371

Book 2

2.0	UTILITY PROGRAMS	
2.1	PAGE PROGRAM	
2.2	FORMAT PROGRAM	
2.3	STATUS PROGRAM	
2.4	FDMLDR PROGRAM	
3.0	MACRO-11 PROGRAMS	
3.1	DIAGNOSTICS LOOP 5	
3.2	IEEE PROGRAMS	
3.3	LOADER PROMS	

MSCDM SOFTWARE MAINTENANCE MANUAL

Book 2 PAGE

4.0 MODIFICATION TO ESMD SOFTWARE

4.1 GAT13 STACK MACHINE

4.2 ESMD USR LANGUAGE

4.3 ESMD LOADER

APPENDIX 1 Systems Drawings

APPENDIX 2 System Diskettes

APPENDIX 3 Startup Procedures

APPENDIX 4 Interface Data

APPENDIX 5 Acronyms Glossary

Book 3 PAGE

5.0 MSCDM Software Flow Charts

INTRODUCTION

The Modular System Control Development Model (MSCDM) consists of nine functional nodes: Station-to-Station Communications Interface (SSCI), Voice Service Quality Control (VSQC), Digital Service Quality Control (DSQC), Data Base Management Service (DBMS), Operator Control and Report Interface (OCRI), Baseband Signal Analysis and Wide Band Signal Analysis (BWBSA), Fault Isolation and Control Coordination (FIAC), Switch Data Collection and Analysis (SDCA), and Simulated Input Generator (SIG). Each of these nodes is implemented using microcomputer hardware and software, and node intercommunication is performed via a Burroughs loop architecture under control of a distributed master control program (DMCP).

Each node can communicate with any other node on the loop; however, the nodal software for the MSCDM application defines the flow of information in the system. For example, the OCRI terminal normally communicates with the DBMS node, which runs the User Language. The other ESM terminals communicate with the User Language via the loop 4-5 gateway Node 21 (SSCI).

A simulated input generator (SIG) generates inputs to the VSQC, DSQC and BWBSA, which communicate faults to the FIAC module. FIAC generates event reports to the OCRI and DBMS. The PDP 11/40 in loop 2 generates inputs to the SDCA which generates fault reports to the OCRI and DBMS. The DBMS, OCRI and FIAC communicate with the other loops via SSCI.

An LA36 DECWRITER is be used as the OCRI hard-copy terminal attached to node 25. A VT52 DECScope is be used as a local CRT terminal connected to the Program Development Unit (PDU).

The Digital Equipment Corporation System's software for the PDP-11/V03 is contained on 9 - Floppy diskettes conforming to DEC's RX01 Floppy drive format.

The MSCDM applications software which runs on the PDP-11/V03 and the LSI-11/2 nodes is contained on 21- floppy diskettes.

References for the FORTRAN and MACRO-11 languages and the PDP-11 /V03 system used include documentation:

1. "RT-11 System Generation Manual"
2. "Introduction to RT11"
3. "RT-11 System User's Guide"
4. "RT-11 System Message Manual"
5. "PDP-11 MACRO Language Manual"
6. "PDP-11 FORTRAN Language Manual"
7. "Advanced Programmer's Guide"

1.0 MSCDM DMCP and APPLICATION Programs

The MSCDM loop contains nine LSI-11 microprocessors, eight of which are configured with loop interface units. The ninth is a LSI-11 which contains three asynchronous interfaces, and is used as the simulated input generator to VSQC, DSQC and BWBSA. The eight loop nodes run a program called the Distributed Master Control Program (DMCP), which performs loop and exodevice interfacing and the queuing of incoming and outgoing messages. Each MSCDM loop microprocessor also runs an application program which is then linked to the DMCP to form a load module for each node. Section 1.1 provides a functional description of the DMCP and the common routines of DMCP shared by the eight loop nodes.

Sections 1.2 to 1.9 contain the description of non-common routines along with the listings of the DMCP and application programs for each node (e.g., Section 1.2 Node 21 (SSCI), Section 1.3 Node 22 (VSQC), etc.). Section 1.10 is the description and listings of the MSCDM User Language which is the application program for Node 24 (DBMS). Section 1.11 is the description and listings of the two simulated input programs used by MSCDM. At the end of each Section (1.2 - 1.11) there are listings of the command files used for compiling and linking the DMCP and application source files.

1.1 THE DISTRIBUTED MASTER CONTROL PROGRAM

The Distributed Master Control Program (DMCP) resides mainly in the microprocessor (LSI-11) nodes which make up the Burroughs Loop Architecture. DMCP modules which reside in the host computer include a Down-Line Load Module which loads remote microprocessors with memory image object files, an Initialization Module which is used at start-up time, and a System Control Resource Allocation Module which is used for dynamic system reconfiguration. In addition, the host computer stores the tables required by the node microprocessors for reconfiguration and functional task assignment.

1.1.1 DMCP Functional Description

The DMCP can be described in terms of the following functional modules: Interrrupt Handlers, Protocol Modules, Loop Manager, Read and Write Modules, Queue Manager, Command Interpreter, Error Module, and I/O Queues. These functional modules are described below.

The DMCP is an Interrupt or Event Driven System. Events result in interrupts being generated by the external hardware (e.g., Loop Interface Unit - LIU) to the LSI-11 microprocessor. The interrupt results in the execution of a vectored interrupt address which passes program control to an I/O Handler (written in MACRO-11 Assembly Language).

A general understanding of the DMCP can be obtained by examining the flow of control resulting from an event. Figure 3 gives the flow control resulting from a packet being received by an LIU from the loop. The LIU generates an interrupt to its microprocessor which results in the execution of the LIU Handler. The LIU Handler examines a 16-bit LIU Status Register. Conditions which the LIU Status Register indicate include Buffer 0 or 1 Full, CRC 0 or 1 OK, Input Buffer 0 or 1 Overflow, Line Switch Primary, and Line Switch Backup. Depending on the LIU Status Register value and the information received, control can be passed to one of three modules: Error Module, Input Q Handler or Command Interpreter. The Error Module is entered if the LIU status register indicates an error condition (e.g., Line Switch Primary). An error report is generated describing the failure. The Loop Protocol Module is then entered and the Packet Header Information will be determined (e.g., sequence number destination and source process ID's). The Loop Address field is also determined for the process ID of the destination task which accepts reports. Finally the packet is written to the loop using the Write to Loop Module.

The second path in Figure 3 is followed if the input from the loop is a normal type packet destined to an Exodevice (e.g., terminal, minicomputer, data comm line). The Input Q Handler is entered to link the packet to the Input to Exodevice Queue. The Exodevice Protocol Module builds the protocol characters for the Exodevice (e.g., SOH, ETX). The Write to Exodevice Module is then called to write the packet onto the loop.

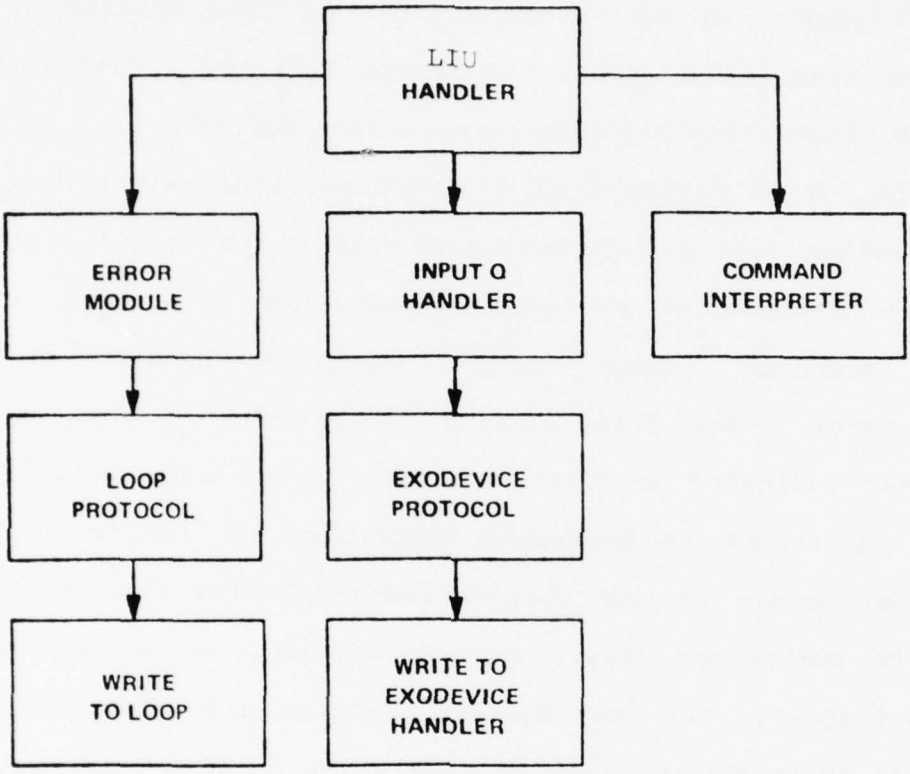


Figure 3. Input from Loop Flow Control

The third path in Figure 3 is the command interpreter function, for handling ACK and NAK messages.

The control for an input from an Exodevice is given in Figure 4. The interrupt results in the execution of the Exodevice Read Handler which can pass control to two modules. A normal loop destined packet results in the Output Q Handler being called. The Loop Protocol Module formats the header of the packet and determines the loop address. The Write to Loop Module writes the packet to the loop.

The second path in Figure 4 results in the Command Interpreter. This module interprets commands originating from the Exodevice. An example would be an ATTACH command which would be used to logically attach a peripheral connected to the loop to a minicomputer.

The DMCP Idle Flow Control is given in Figure 5. The Input Q Handler is entered to examine the current input queue size. The Output Q Handler is entered to examine whether the packet in the output queue has been ACKed or NAKed. The Loop Manager maintains the watchdog timing function for write token timeouts and retransmissions of packets that were not ACKed or NAKed. The System Monitor sends messages to its Exodevice and a System Master to monitor equipment operation.

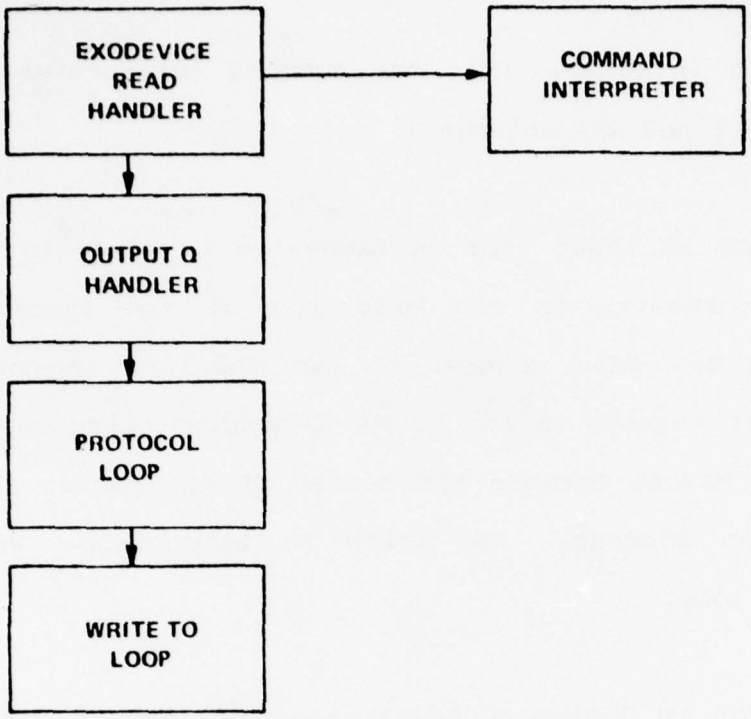


Figure 4. Input from Exodevice Flow Control

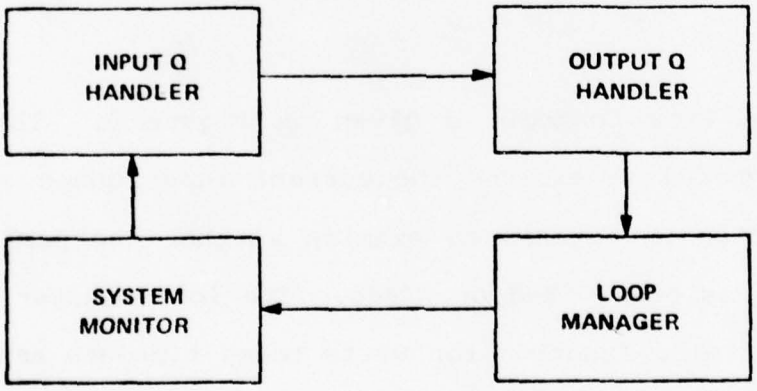


Figure 5. DMCP Idle Flow Control

1.1.1.1 Interrupt Handlers Function

Interrupt Handlers are used to process I/O interrupts. The DMCP in each node contains four basic types. The LIU Handler, examines the LIU Status Register and does a Memory Access (DMA) transfer to move a packet (at least 256 bytes) from the LIU buffers to the microprocessor memory. The Clock Handler processes interrupts from the 60 Hz Line Frequency Clock. Interrupts occur every 16.67 msec.; a clock register in memory is incremented at each interrupt. The LIU and Clock Handlers are identical for each node.

A Read and Write Exodevice is used for each type of Exodevice. Peripheral and data comm handlers receive interrupts on a per byte basis. Interrupt Handlers are written in MACRO-11 Assembly Language.

1.1.1.2 Protocol Modules Function

There are two basic types of Protocol Modules. A Loop Protocol Module and Exodevice Protocol Modules. The Loop Protocol Module is the same at all nodes. Exodevice Protocol Modules are used for each type of exodevice. Protocol Modules format packets, perform code conversion if necessary, and handle communication protocols for the loop, processors, peripherals, and data comm. The Loop Protocol Module determines destination loop addresses based upon its tables for reconfiguration and functional task assignment. Protocol Modules are either separately written in FORTRAN IV or combined with Interrupt Handlers and written in MACRO-11.

1.1.1.3 Loop Manager Function

The Loop Manager is written in FORTRAN and is identical for all nodes. It manages certain control functions associated with the loop. One such function is the regeneration of Write Tokens. A Write Token is a special circulating control packet which the LIU must recognize before it can write on the loop. If a Write Token is not received within a certain amount of time, the Loop Manager creates a new Write Token. Another function is the resending of packets to the loop that have not ACKed or NAKed within a certain amount of time.

1.1.1.4 Read Module Function

Read Modules read data from either the loop or an Exodevice. When combined with an Interrupt Handler, Read Modules are written in MACRO-11 Assembly Language.

1.1.1.5 Write Module Function

Write Modules write data to the loop or an Exodevice. When combined with an Interrupt Handler, Write Modules are written in Assembly Language.

1.1.1.6 Queue Manager Function

The Queue Manager maintains input and output queues. It maintains status on the number in queue and the current top-of-queue pointer. It moves certain control type command messages to top-of-queue or to other I/O buffers (e.g., intermediate ACK's). The Queue Manager monitors the size of queue and generates flow control command messages. The flow control messages direct nodes to stop sending to a node whose queue is nearly fully and resume sending to a node whose queue has emptied. The Queue Manager is used at all nodes. It is written in FORTRAN and interfaces to Interrupt Handlers via COMMON I/O buffers.

1.1.1.7 Command Interpreter Function

The Command Interpreter is used to perform nodal control functions as a result of interpreting command packets (utilizing the header control bytes). Many of these command packets such as ACK-NAK's, line switch, etc. are described above. A function not listed above is Down-Line Loading Command interpretation. A Down-Line Loading command forces the microprocessor to execute a bootstrap loader program stored in PROM memory. The bootstrap loader program loads data coming from the loop into the RAM memory. The data is sent to Absolute Loader Format by the loader host computer (Downline Load Module). The Loader PROM program returns program control to the first executable RAM memory location when an end-of-load command is received. The LSI-11 microprocessor is strapped such that the bootstrap loader PROM is executed at power-up time. The bootstrap loader PROM program is written in Assembly Language.

1.1.1.8 Error Module Function

The Error Module is written in FORTRAN and is used at all nodes. The Error Module formats error reports to describe nodal error conditions. These error reports include error conditions indicated by the LIU Status Register (e.g., Loopback), no response from destination nodes, and queue overflow. The reports are sent to a loop connected peripheral.

1.1.1.9 I/O Queues Function

I/O Queues are maintained by the Queue Managers. The maximum queue sizes are determined by the speed of the Exodevice Interface, the traffic at the node, and the availability memory (64K bytes maximum). Output to the loop queues is effectively of size one since each packet is ACKed before the next one is sent. This guarantees that packets are received in the same order as sent and eliminates the need for packet sequencing software.

1.1.2 DMCP ROUTINES

The DMCP Control Software comprises a set of FORTRAN and MACRO-11 routines which are linked to an application program to form a load module which is then down-line loaded to the LSI-11 nodes.

Since most of the code is common to all nodes, the description of the common routines will be divided and discussed in this section. Unique code or routines will be discussed later under the sections devoted to each node (i.e., Sections 1.2 to 1.9).

1.1.2.1 Program Nodal Description (FORTRAN)

This is the main program of the DMCP and its functions are to examine the queues to see if they are empty or full, move data to empty queues, perform output to the LIU and exodevices, and check for timeouts, for write token detect, and LIU output message acknowledgement. It also calls the application program when there is something in the loop input queue.

1.1.2.2 Subroutine IGETSP (FORTRAN)

The purpose of this subroutine is to fetch a pointer to a free packet array from the free packet list. The free packet list is examined to determine if the list is empty; if it is, the INIT subroutine is called. If not, a pointer to an available packet is fetched.

1.1.2.3 Subroutine ENQUE (FORTRAN)

This subroutine adds an element at the tail of a queue. If the current tail is not at the lower limit of the queue, the new element is appended and the tail pointer adjusted. Otherwise, a

check is made to determine whether or not the ENQUE is full. If the queue is not full, the queue is moved so that the head is at the upper limit and the new element is added. If full, subroutine INIT is called.

1.1.2.4 Subroutine DEQUE (FORTRAN)

This subroutine returns the head of a specified queue element. The element may or may not be removed from the queue, depending upon a parameter setting.

1.1.2.5 Subroutine ACKNAK (FORTRAN)

This subroutine is called when an ACK or NAK message is received. In the case of the ACK message, the output queue is marked empty, the pointer to the packet pointed to by the output queue is returned to the free packet list and the write flags are reset.

1.1.2.6 Subroutine INPTQ (FORTRAN)

This subroutine checks the validity of a non-control message received from the LIU. If the message is valid, an ACK message is generated and placed in the ACK queue. If the message is not valid, a NAK message is generated.

1.1.2.7 Subroutine LPINPT (FORTRAN)

This procedure determines whether a message received from the LIU is a control message, an ACK or NAK message, or a normal message. The appropriate subroutine is called after the determination has been made. For ACK's and NAK's, subroutine ACKNAK is called. For normal messages, subroutine INPTQ is called for normal messages.

1.1.2.8 Subroutine LINLOS (FORTRAN)

This subroutine assembles a message indicating the switching of the primary or backup lines. The message is sent to Node 25 and printed on the LA-36.

1.1.2.9 Subroutine INIT (FORTRAN)

This subroutine initializes the queues, the free arrays, the LID of the node, the read address of the node, the LID/FAD table and the address comparison RAM of the LIU.

1.1.2.10 Subroutine MASTER (MACRO)

This subroutine initializes all interrupt vectors of the devices of the node, lowers the CPU priority to allow interrupts to be processed.

1.1.2.11 Subroutine LIUINT (MACRO)

This subroutine initializes the LIU's, address comparison RAM, both input and output buffers, and the status register.

1.1.2.12 Subroutine LIO (MACRO)

This subroutine in entered by a interrupt from the LIU. It reads the status of the LIU and determines what caused the interrupt. If one of the input buffers is full, it unloads them and calls subroutine LPINPT. If it was a line loss, it calls subroutine LINLOS, and if it was a bad CRC on either of the input buffers, it unloads the buffers and calls subroutine LPINPT.

1.1.2.13 Subroutine ENABLE (MACRO)

This subroutine is passed a variable which either lowers the CPU priority to enable interrpts or raises priority thus disabling interrupts from being acknowleged.

1.1.2.14 Subroutine Switch (MACRO)

This subroutine is passed a variable which either sets or resets the backup or primary line switches.

1.1.2.15 Subroutine Status (MACRO)

This subroutine is passed one argument that specifies which LIU status byte is to be read. That status byte is then passed back as the second argument in the call statement.

1.1.2.16 Subroutine WTOKEN (MACRO)

This subroutine is called to regenerate a write token in the loop.

1.1.2.17 Subroutine TIME (MACRO)

The subroutine is the interrupt routine that is called by the LTC interrupt. It increments a variable and checks for counter overflow.

1.1.2.18 Subroutine RAM (MACRO)

The subroutine is called to read or write a location in the LIU address comparison RAM.

1.1.2.19 Subroutine LPOUT (MACRO)

This subroutine is called to load output buffer 0 with a message, and load output buffer 1 with a write token.

1.1.2.20 Subroutine RDPNT (MACRO)

This subroutine is called to read a buffer pointer of either input buffer 0 or 1 or output buffer 0 or 1.

1.1.2.21 Subroutine RSTART (MACRO)

The subroutine is called to either restart nodal software, halt processor or jump to loader prom.

1.2 NODE 21 (SSCI)

The Station-to-Station Communication Interface (SSCI) serves as a gateway node interface to loop 4 of the ESM. The loop 4 - loop 5 interface implemented is 9600 baud asynchronous. The SSCI is used to simulate communication between different system control sites. The SSCI performs code conversion, intransit queueing and packet routing.

1.2.1 Program Description

1.2.1.1 Refer to Section 1.1 for descriptions of routines - IGETSP, ENQUE, DEQUE, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME, RAM, LPOUT, RDPNT, RSTART.

1.2.1.2 Program NODAL Description (FORTRAN)

This is the main program for the SSCI DMCP and its functions are to examine the queues to see if they are empty or full, move data to empty queues, write the output of all the incoming loop messages to the gateway interface (Node 13, loop 4) and place on loop 5 all messages received from the gateway interface, SSCI does not acknowledge messages received from the loop, ESM employs "end to end" acknowledgement (SSCI functions only as the gateway).

1.2.1.3 Subroutine INPTQ (FORTRAN)

This subroutine does the queueing of incoming packet from the way. No check is made for control packets.

1.2.1.4 Subroutine LPINPT (FORTRAN)

This subroutine only calls INPTQ routine because SSCI does not scan any of the incoming messages from the loop interface. All incoming messages go to the gateway interface, also LPINPT do not generate ACK or NAK to messages.

1.2.1.5 Subroutine GIO (MACRO)

This subroutine is the interrupt handler for the gate-way interface. It reads the message from the interface and then sets a flag containing the byte count of that message.

1.2.1.6 Subroutine GOUT (MACRO)

This subroutine is called to send a message over the gateway interface to node 13, loop 4. It's passed one argument the byte count of the message to be sent.

```
0001        PROGRAM NODAL
0002        INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003        INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004        INTEGER*2 SETPRM,RSTPRM,SETBKP,CC1,CC2
0005        INTEGER*2 RSTBKP,Q1,Q2,RESNLM,OUTQ,DEQUE
0006        LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007        LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0008        INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0009        COMMON /MESS/ MESSEQ
0010        COMMON/DFM/ OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0011        COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
           1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0012        COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
           1             LIDFD(256),IOFLG,LLFLG,IRSEND
0013        COMMON /FRE/ FREE(64),IFR,IFRSZ
0014        COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0015        COMMON       PACK(256,64)
0016        COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0017        COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
           1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0018        CALL MASTER
0019        CALL INIT
0020        CALL ENABLE(1)
0021     5  CONTINUE
0022    25  IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0024        CALL ENABLE(0)
0025        Q2=DEQUE(PINQ,INQ,1)
0026        CALL ENABLE(1)
0027        CALL DESTR(PACK(254,Q2),LEN)
0028        DO 30 I=1,LEN
0029          XOUTBF(I)=PACK(I,Q2)
0030    30  CONTINUE
0031        CALL GOUT(LEN,IST)
0032        IFR=IFR+1
0033        FREE(IFR)=Q2
0034    40  CALL STATB0(IS)
0035        IF(IS .EQ. 1) GOTO 110
0037        IF(IOFLG .EQ. 0) GOTO 81
0039    83  CALL ENABLE(0)
0040        Q2=IGETSP(N)
0041        CALL ENABLE(1)
0042    21  DO 82 I=1,IOFLG
0043        PACK(I,Q2)=XINBF(I)
0044    82  CONTINUE
0045        CALL ENSTR(PACK(254,Q2),IOFLG)
0046        CALL ENABLE(0)
0047        CALL ENQUE(PXINQ,XINQ,Q2)
0048        CALL ENABLE(1)
0049    84  IOFLG=0
0050    81  IF(LLFLG .GT. 0) CALL LINLOS
0052        IF(PXINQ(1) .LT. PXINQ(2)) GOTO 110
0054        CALL ENABLE(0)
0055        Q1=DEQUE(PXINQ,XINQ,1)
0056        CALL ENABLE(1)
```

```
0057          OUTQ=Q1
0058     85   CALL DESTR(PACK(254,OUTQ),Q2)
0059          DO 90 I=1,Q2
0060            OUTBF(I)=PACK(I,OUTQ)
0061     90   CONTINUE
0062          OUTBF(Q2+1)=0
0063          IPT=OUTBF(5)
0064          OUTBF(Q2+2)=LIDFD(IPT)
0065          CALL LPOUT(Q2+2)
0066          IWRTTM=0
0067          IWRT=1
0068    110   OLDTIM=NEWTIM
0069          NEWTIM=LTIME
0070          IF(IWRT .EQ. 0) GOTO 130
0072          IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0073          IF(IWRTTM .LT. TIMLIM) GOTO 130
0075          CALL WTOKEN
0076          IWRTTM=0
0077          IWRT=0
0078    130   CONTINUE
0079          GOTO 5
0080          END
```

```
FORTRAN IV     V02.1-1     Mon 04-Jun-79 17:07:20              PAGE 001

0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(64),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IF(IFR .LT. 1) CALL INIT
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```

```
0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008       10 IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016       20 IF(IQHEAD .GE. IQLNTH) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020       30 B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024       40 CALL INIT
0025      999 RETURN
0026          END
```

```
FORTRAN IV     V02.1-1    Mon 04-Jun-79 17:07:43              PAGE 001

0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016       10 A(1)=IQHEAD-1
0017      999 RETURN
0018          END
```

```
FORTRAN IV      V02.1-1     Mon 04-Jun-79 17:08:05              PAGE 001

0001          SUBROUTINE INPTQ(L)
0002          INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
0003          INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005          LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
0006          INTEGER*2 LTIME,IWRTTM
0007          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IOFLG,LLFLG,IRSEND
0009          COMMON         PACK(256,64)
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          IF(L .LT. 1) GOTO 999
0014          LI=L-2
0015          T1=IGETSP(N)
0016          DO 10 I=1,LI
0017       10 PACK(I,T1)=INBF(I)
0018          CALL ENSTR(PACK(254,T1),LI)
0019          CALL ENQUE(PINQ,INQ,T1)
0020      999 RETURN
0021          END
```

```
FORTRAN IV     V02.1-1    Mon 04-Jun-79 17:08:19              PAGE 001

0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON/DFM/OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008     40   CALL INPTQ(LI)
0009          RETURN
0010          END
```

```
0001          SUBROUTINE INIT
0002          REAL*4 RH,RL,AH,AL,IRSV,VMEAS
0003          REAL*8 VTR
0004          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0005          INTEGER*2 STAT,FLWCNT,OUTFCT
0006          INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0008          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM,IDATA
0009          INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0010          COMMON /DFM/ OUTBF(256),INBF(256),
             1    IWRTTM,IWRT,STAT,LTIME
0011          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1             ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0012          COMMON /MESS/ MESSEQ
0013          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1    IOFLG,LLFLG,IRSEND
0014          COMMON /FRE/  FREE(64),IFR,IFRSZ
0015          COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0016          COMMON        PACK(256,64)
0017          COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0018          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0019          CALL RAM(0,4,4)
0020          CALL RAM(0,1,4)
0021          CALL RAM(0,255,0)
0022          MESSEQ=0
0023          IOFLG=0
0024          IRSEND=0
0025          LLFLG=0
0026          IFRSZ=64
0027          INFLCT=0
0028          ISLID=21
0029          CR='015
0030          LF='012
0031          ETX='003
0032          RESNLM=2
0033          IQLNTH=16
0034          IALTRT=0
0035          FLWCNT=0
0036          IWRTTM=0
0037          TIMLIM=35
0038          SETPRM=4
0039          RSTPRM=32
0040          SETBKP=8
0041          RSTBKP=64
0042          ATIMLM=200
0043          MONTOR=25
0044          IWRT=0
0045          OUTFCT=0
0046          IFULL=0
0047          ISENT=0
0048          DO 40 I=1,20
0049       40 LIDFD(I)=4
0050          LIDFD(21)=1
```

```
0051          LIDFD(22)=3
0052          LIDFD(23)=6
0053          LIDFD(24)=5
0054          LIDFD(25)=7
0055          LIDFD(26)=8
0056          LIDFD(27)=9
0057          LIDFD(28)=2
0058          DO 50 I=29,39
0059       50 LIDFD(I)=0
0060          DO 60 I=40,44
0061       60 LIDFD(I)=1
0062          DO 70 I=45,59
0063       70 LIDFD(I)=0
0064          DO 80 I=60,64
0065       80 LIDFD(I)=2
0066          DO 90 I=65,79
0067       90 LIDFD(I)=0
0068          DO 100 I=80,84
0069      100 LIDFD(I)=4
0070          DO 110 I=85,99
0071      110 LIDFD(I)=0
0072          DO 120 I=100,104
0073      120 LIDFD(I)=5
0074          DO 130 I=105,256
0075      130 LIDFD(I)=0
0076          PINQ(1)=IQLNTH
0077          PINQ(2)=IQLNTH+1
0078          PACKQ(1)=IQLNTH
0079          PACKQ(2)=IQLNTH+1
0080          PXOUTQ(1)=IQLNTH
0081          PXOUTQ(2)=IQLNTH+1
0082          PXINQ(1)=IQLNTH
0083          PXINQ(2)=IQLNTH+1
0084          DO 10 I=1,IFRSZ
0085            PACK(255,I)=0
0086       10 PACK(256,I)=0
0087          DO 20 I=1,IFRSZ
0088       20 FREE(I)=I
0089          DO 30 I=1,256
0090       30 ACKSEQ(I)=256
0091          IFR=IFRSZ
0092          RETURN
0093          END
```

```
0001          SUBROUTINE LINLOS
0002          REAL*8 RM1(5),LIN08,LIN18
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,PINQ
0005          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0007          LOGICAL*1 PACK,M1(40),LIN0(8),LIN1(8)
0008          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0009          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &            IOFLG,LLFLG,IRSEND
0010          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011          COMMON        PACK(256,64)
0012          COMMON /MESS/ MESSEQ
0013          DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0014          DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0015          DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0016          DATA RM1(5)/'DE 21   '/
0017          EQUIVALENCE(LIN0,LIN08)
0018          EQUIVALENCE(LIN1,LIN18)
0019          EQUIVALENCE(M1,RM1)
0020          IS=LLFLG
0021          CALL ENABLE(0)
0022          T1=IGETSP(N)
0023          CALL ENABLE(1)
0024          PACK(1,T1)=0
0025          IF(MESSEQ .EQ. 126) MESSEQ=0
0027          MESSEQ=MESSEQ+1
0028          PACK(2,T1)=MESSEQ
0029          PACK(3,T1)=0
0030          PACK(4,T1)=0
0031          PACK(5,T1)=25
0032          PACK(6,T1)=ISLID
0033          DO 20 I=7,9
0034            PACK(I,T1)=LF
0035       20 CONTINUE
0036          DO 30 I=1,22
0037       30 PACK(I+9,T1)=M1(I)
0038          IF(IS .EQ. 1) GOTO 50
0040          DO 40 I=1,8
0041       40 PACK(I+31,T1)=LIN0(I)
0042          GOTO 70
0043       50 DO 60 I=1,8
0044       60 PACK(I+31,T1)=LIN1(I)
0045       70 DO 80 I=23,40
0046       80 PACK(I+17,T1)=M1(I)
0047          PACK(58,T1)=CR
0048          PACK(59,T1)=LF
0049          PACK(60,T1)=ETX
0050          CALL ENSTR(PACK(254,T1),60)
0051          CALL ENABLE(0)
0052          CALL ENQUE(PXINQ,XINQ,T1)
0053          CALL ENABLE(1)
```

```
0054          LLFLG=0
0055          RETURN
0056          END
```

```
                                .TITLE  FDM.MACRO
                                .SBTTL  NODE21
                                .IDENT  /V3.0/
                                .GLOBL  LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
                                .GLOBL  WTOKEN,RAM,STATUS,STATBO
                                .GLOBL  LPOUT,RSTART,RDENT,DESTR,ENSTR,LPINPT
                                .NLIST  CND
000000                          .PSECT

000000          R0      =%0
000001          R1      =%1
000002          R2      =%2
000003          R3      =%3
000004          R4      =%4
000005          R5      =%5
000006          SP      =%6
000007          PC      =%7

                ;INTERFACE ADDRESS

172410          BAR=    172410
172412          WCR=    172412
172414          CSR=    172414
172416          IOBUF=  172416
172416          OPREG=  172416
177560          HRCSR=  177560
177562          HRBUF=  177562
177564          HXCSR=  177564
177566          HXBUF=  177566

                ;INTERFACE OPTIONS

000001          XCIO=1
000001          XSIO=1
000000          XGIO=0
000000          NDXX=0
000001          ND24=1
000000          NHOST=0
000001          DHOST=1

                ;LOCAL VARIBLES

000002          ARG1=   2
000004          ARG2=   4
000006          ARG3=   6
000010          ARG4=   10

100000          BIT15=  100000
040000          BIT14=  40000
000400          BIT08=  400
000200          BIT07=  200
000100          BIT06=  100
000040          BIT05=  40
000020          BIT04=  20
000010          BIT03=  10
000004          BIT02=  4
```

```
FDM.MACRO       MACRO V03.01 4-JUN-79 17:09:17 PAGE 1-1
NODE21

     66          000002                          BIT01=  2
     67          000001                          BIT00=  1
     68
```

```
                        ;*************************************************************
                        ;**** PROCEDURE MASTER START UP                              *
                        ;*************************************************************

000000  012700  000340          MASTER: MOV     #340,   R0                      ;PRI=7
000004  106400                  MTPS    R0
000006  012700  000124                  MOV     #124,   R0
000012  012720  000436'                 MOV     #LIO,   (R0)+                   ;LIU HANDLER
000016  012710  000340                  MOV     #340,   (R0)                    ;PRI=7
000022  000005                  RESET                                           ;RESET BUS
000024  012700  000060                  MOV     #60,    R0                      ;INTERFACE VECTOR
000030  012720  002474'                 MOV     #GIO,   (R0)+                   ;GATE WAY HANDLER
000034  012720  000340                  MOV     #340,   (R0)+                   ;PRI=7
000040  052737  000100  177560          BIS     #BIT06, @#HKCSR
000046  012700  000100                  MOV     #100,   R0                      ;CLOCK VECTOR
000052  012710  001662'                 MOV     #TIME,  (R0)                    ;CLOCK
000056  042737  040100  172414          BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000064  004767  000014                  JSR     PC,     LIUINT                  ;CLEAR LIU
000070  005037  172414                  CLR     @#CSR                           ;CLEAR BLIUI CSR
000074  012700  000000                  MOV     #000,   R0                      ;PRI=0
000100  106400                  MTPS    R0                                      ;CPU DOWN
000102  000207                  RTS     PC


                        ;*************************************************************
                        ;**** PROCEDURE INITIALIZE                                  *
                        ;*************************************************************

                                ;CLEAR ACRAM

000104  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000112  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000120  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000124  100375                  BPL     .-4                                     ;NO RETRY
000126  105037  172414                  CLRB    @#CSR
000132  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
000140  012700  177400                  MOV     #-256., R0                      ;COUNTER
000144  012737  004407  172416  1$:     MOV     #2311., @#OPREG                 ;WRITE A NULL
000152  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000156  100375                  BPL     .-4                                     ;NO LOOP UNTIL READY
000160  105037  172414                  CLRB    @#CSR
000164  005200                  INC     R0                                      ;COUNT+1
000166  001366                  BNE     1$

                                ;CLEAR INPUT/OUTPUT BUFFERS

000170  012701  010410                  MOV     #10410, R1                      ;RDBUFADR CMD
000174  012700  010440                  MOV     #10440, R0                      ;SEL INBUF0
000200  004767  000106                  JSR     PC,     ZEROBP                  ;SET POINTER=0
000204  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000212  105737  172414                  TSTB    @#CSR                           ;GOOD READ
000216  100375                  BPL     .-4                                     ;NO RETRY
000220  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
000224  012701  010610                  MOV     #10610, R1                      ;RDBUFADR COMMAND
000230  012700  010640                  MOV     #10640, R0                      ;SEL INBUF1 COMMAND
```

```
 98 000234  004767  000052                    JSR     PC,     ZEROBP          ;POINTER=0
 99 000240  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
100 000246  105737  172414                    TSTB    @#CSR                   ;GOOD RD
101 000252  100375                            BPL     .-4                     ;NO RETRY
102 000254  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
103 000260  012701  010510                    MOV     #10510, R1              ;RDBUFADR COMMAND
104 000264  012700  010540                    MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
105 000270  004767  000016                    JSR     PC,     ZEROBP
106 000274  012701  010710                    MOV     #10710, R1              ;RDBUFADR COMMAND
107 000300  012700  010740                    MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
108 000304  004767  000002                    JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
109 000310  000440                            BR      STAT$                   ;GO CLEAR STATUS
110 000312  010137  172416          ZEROBP:   MOV     R1,     @#OPREG         ;RDBUFADR
111 000316  012737  001400  172416            MOV     #1400,  @#OPREG         ;RD
112 000324  105737  172414                    TSTB    @#CSR                   ;GOOD RD
113 000330  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
114 000332  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
115 000336  013702  172416                    MOV     @#IOBUF, R2             ;FETCH POINTER
116 000342  042702  177400                    BIC     #177400, R2             ;CLEAR MST BYTE
117 000346  010037  172416                    MOV     R0,     @#OPREG         ;SEL BUFFER
118 000352  022702  000000          2$:       CMP     #0,     R2              ;POINTER=0?
119 000356  001412                            BEQ     3$                      ;YES RECHECK
120 000360  012737  001400  172416            MOV     #1400,  @#OPREG         ;NO FALSE RD
121 000366  105737  172414                    TSTB    @#CSR                   ;GOOD RD
122 000372  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
123 000374  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
124 000400  005302                            DEC     R2                      ;LOCAL POINTER-1
125 000402  000763                            BR      2$                      ;BR UNTIL 0
126 000404  010037  172416          3$:       MOV     R0,     @#OPREG         ;SEL BUFFER
127 000410  000207                            RTS     PC                      ;RETURN
128
129                                           ;READ AND CLEAR STATUS
130
131 000412  012737  010400  172416  STAT$:   MOV     #4352.,@#OPREG          ;WCR:RS
132 000420  012737  002400  172416            MOV     #1280.,@#OPREG          ;RS(FALSE)
133 000426  012737  002400  172416            MOV     #1280.,@#OPREG          ;RS
134 000434  000207                            RTS     PC
135
136
```

```
                                ;*************************************************************
                                ;**** PROCEDURE LIU-HANDLER                                 *
                                ;*************************************************************


000436  010046                  LIU:    MOV     R0,     -(SP)                   ;SAVE R0
000440  010146                          MOV     R1,     -(SP)                   ;SAVE R1
000442  010246                          MOV     R2,     -(SP)                   ;SAVE R2
000444  010346                          MOV     R3,     -(SP)                   ;SAVE R3
000446  010446                          MOV     R4,     -(SP)                   ;SAVE R4
000450  010546                          MOV     R5,     -(SP)                   ;SAVE R5
000452  005737  172414                  TST     @#CSR                           ;? LIU INTERRUPTED
000456  100153                          BPL     RTI$                            ;NO
000460  042737  040000  172414          BIC     #BIT14, @#CSR                   ;YES/DISABLE LIU
000466  012737  010400  172416          MOV     #4352., @#OPREG                 ;WCR : RS(0)
000474  012737  002400  172416          MOV     #1280., @#OPREG                 ;RD
000502  013767  172416  000502          MOV     @#IOBUF, CAUSE
000510  042767  177400  000474          BIC     #177400, CAUSE                  ;CLEAR UNUSED BITS

                                        ;PREFORM OPERATION DEPENDING ON BITS

000516  132767  000004  000466  B2$:    BITB    #BIT02, CAUSE                   ;OV-FL
000524  001406                          BEQ     B3$
000526  012702  010410                  MOV     #4360., R2                      ;RDBUFADR IN0
000532  012703  010440                  MOV     #4384., R3                      ;SEL INBUF0
000536  004767  000270                  JSR     PC,     EMBF                    ;GO EMPTY
000542  132767  000010  000442  B3$:    BITB    #BIT03, CAUSE                   ;OV-FL
000550  001406                          BEQ     B4$
000552  012702  010610                  MOV     #4488., R2                      ;RDBUFADR IN1
000556  012703  010640                  MOV     #4512., R3                      ;SEL INBUF1
000562  004767  000244                  JSR     PC,     EMBF                    ;GO EMPTY
000566  132767  000020  000416  B4$:    BITB    #BIT04, CAUSE                   ;LINE-LOSS PRIMARY
000574  001407                          BEQ     B5$
000576  132767  000200  000406  B7$:    BITB    #BIT07, CAUSE                   ;LATCH ON
000604  001403                          BEQ     B5$                             ;YES LOOP AROUND
000606  012767  000002  002002'         MOV     #2,     LLFLG                   ;FLAG PRIMARY SW. FAIL
000614  132767  000040  000370  B5$:    BITB    #BIT05, CAUSE                   ;BACKUP LINE LOSS
000622  001403                          BEQ     B6$
000624  012767  000001  002002'         MOV     #1,     LLFLG                   ;FLAG BACKUP SW. FAIL
000632  132767  000100  000352  B6$:    BITB    #BIT06, CAUSE                   ;WRITE TOKEN DETECT
000640  001404                          BEQ     B0$
000642  005067  001000'                  CLR     IWRTTM                          ;CLEAR TOKEN VAR
000646  005067  001002'                  CLR     IWRT

                                        ;BUFFER FULL

000652  132767  000001  000332  B0$:    BITB    #BIT00, CAUSE                   ;INBUF0 FULL
000660  001423                          BEQ     B1$                             ;NO
000662  012701  000400'                  MOV     #INBF,  R1                      ;BUFFER ADDRESS
000666  012702  010410                  MOV     #4360., R2                      ;RDBUFADR COMM
000672  012703  010440                  MOV     #4384., R3                      ;SEL BUF
000676  012704  000001                  MOV     #BIT00, R4                      ;CRC BIT
000702  004767  000124                  JSR     PC,     EMBF                    ;EMPTY BUFFER
000706  012705  001214'                  MOV     #AREA,  R5                      ;DATA LINK AREA
000712  012767  000001  000274          MOV     #1,     AREA                    ;1 VARIBLE
000720  010267  000274                  MOV     R2,     DATA                    ;CRC OR BYTE COUNT
000724  004767  000000G                 JSR     PC,     LPINPT                  ;CALL FORTRAN QUE'ER
```

```
     58 000730  132767  000002  000254  B1$:    BITB    #BIT01, CAUSE           ;INBUF1 FULL
     59 000736  001423                   BEQ     RTI$                    ;NO WERE DONE
     60 000740  012701  000400'                  MOV     #INBF,  R1              ;BUFFER ADDRESS
     61 000744  012702  010610                    MOV     #4488., R2              ;RDBUFADR COMM
     62 000750  012703  010640                    MOV     #4512., R3              ;SEL INBUF1
     63 000754  012704  000002                    MOV     #BIT01, R4              ;CRC BIT
     64 000760  004767  000046                    JSR     PC,     EMBF            ;GO EMPTY BUFFER
     65 000764  012705  001214'                   MOV     #AREA,  R5              ;DATA LINK AREA
     66 000770  012767  000001  000216          MOV     #1,     AREA            ;ONE VARIBLE
     67 000776  010267  000216                    MOV     R2,     DATA            ;CRC OR BYTE COUNT
     68 001002  004767  000006                    JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
     69
     70 001006  012605                  RTI$:    MOV     (SP)+,  R5              ;RESTORE REGISTERS
     71 001010  012604                           MOV     (SP)+,  R4
     72 001012  012603                           MOV     (SP)+,  R3
     73 001014  012602                           MOV     (SP)+,  R2
     74 001016  012601                           MOV     (SP)+,  R1
     75 001020  012600                           MOV     (SP)+,  R0
     76 001022  052737  040000  172414          BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
     77 001030  000002                           RTI                             ;RETURN FROM INTERRUPT
     78
     79 001032  012737  010600  172416  EMBF:    MOV     #4480., @#OPREG         ;READ STATUS 1
     80 001040  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
     81 001046  013700  172416                    MOV     @#IOBUF, R0             ;FETCH STATUS
     82 001052  130400                           BITB    R4,R0                   ;GOOD CRC
     83 001054  001002                           BNE     CRCOK
     84 001056  012704  177777                    MOV     #-1,    R4              ;NO FLAG
     85 001062  010237  172416          CRCOK:   MOV     R2,     @#OPREG         ;RDBUFADR
     86 001066  012737  001400  172416          MOV     #768.,  @#OPREG         ;RD
     87 001074  105737  172414                    TSTB    @#CSR                   ;GOOD RD
     88 001100  100375                           BPL     .-4
     89 001102  013702  172416                    MOV     @#IOBUF, R2             ;REPLACE WITH POINTER
     90 001106  042702  177400                    BIC     #177400, R2             ;CLEAR MST BITS
     91 001112  010200                           MOV     R2,     R0              ;SAVE IT
     92 001114  005400                           NEG     R0                      ;2'S COMP
     93 001116  010037  172412                    MOV     R0,     @#WCR           ;BYTE COUNT NOW
     94 001122  010137  172410                    MOV     R1,     @#BAR           ;ADDRESS IN MEMORY
     95 001126  010337  172416                    MOV     R3,     @#OPREG         ;SEL BUFFER
     96 001132  012737  001400  172416          MOV     #768.,  @#OPREG         ;FALSE RD
     97 001140  105737  172414                    TSTB    @#CSR                   ;DONE ON
     98 001144  100375                           BPL     .-4                     ;NO LOOP
     99 001146  012737  021000  172416          MOV     #8704., @#OPREG         ;FIRE DMA
    100 001154  000240                           NOP                             ;DELAY
    101 001156  105737  172414                    TSTB    @#CSR                   ;GOOD DMA
    102 001162  100401                           BMI     DMAOK
    103 001164  000240                           NOP                             ;ERROR IF HERE
    104 001166  012737  004400  172416  DMAOK:   MOV     #2304., @#OPREG         ;FALSE WD
    105 001174  105737  172414                    TSTB    @#CSR                   ;GOOD WD
    106 001200  100375                           BPL     .-4
    107 001202  005704                           TST     R4                      ;WAS CRC OK
    108 001204  100001                           BPL     END$                    ;YES
    109 001206  010402                           MOV     R4,     R2              ;NO FLAG IT
    110 001210  000207                  END$:    RTS     PC                      ;RETURN
    111 001212  000000                  CAUSE:   .WORD   0                       ;STATUS BYTE 0 HOLDER
    112 001214  000000                  AREA:    .WORD   0
    113 001216  001220'                          .WORD   DATA
    114 001220  000000                  DATA:    .WORD   0
```

```
                                .ENABLE LSB
                        ;*************************************************************
                        ;**** PROCEDURE ENABLE                                      *
                        ;*************************************************************

001222  017501  000002          ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
001226  022701  000001                  CMP     #1,     R1
001232  001404                          BEQ     E1$
001234  042737  040000  172414  E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
001242  000207                          RTS     PC
001244  052737  040000  172414  E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
001252  000207                          RTS     PC

                                ;FORTRAN ENCODE/DECODE MACRO

001254  017575  000004  000002  ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
001262  000207                          RTS     PC
001264  017575  000002  000004  DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECODE BYTE COUNT
001272  000207                          RTS     PC


                        ;*************************************************************
                        ;**** PROCEDURE LINE SWITCH                                 *
                        ;*************************************************************

                                ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                ;
                                ;04=SET PRIMARY LINE
                                ;32=RESET PRIMARY LINR
                                ;08=SET BACKUP LINE
                                ;64=RESET BACKUP LINE

001274  017501  000002          SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
001300  012737  010420  172416          MOV     #4368., @#OPREG                 ;WCR : MODSTAT
001306  062701  004400                  ADD     #2304., R1
001312  010137  172416                  MOV     R1,     @#OPREG                 ;RD
001316  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001322  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001324  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001330  000207                          RTS     PC
```

```
                                ;*************************************************************
                                ;**** PROCEDURE STATUS                                      *
                                ;*************************************************************
                                
                                        ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
                                        ;
                                        ;X=0 STATUS BYTE 0
                                        ;X=1 STATUS BYTE 1
                                
001332  017501  000002        STATUS: MOV     @ARG1(R5), R1           ;WHICH BYTE
001336  022701  000001                CMP     #1,     R1              ;COMPARE
001342  001415                        BEQ     2$
001344  012737  010400  172416  1$:   MOV     #4352., @#OPREG         ;WCR : RS(0)
001352  012737  002400  172416        MOV     #1280., @#OPREG         ;RS
001360  013700  172416                MOV     @#IOBUF, R0             ;FETCH DATA
001364  042700  177400                BIC     #177400, R0             ;CLEAR BITS
001370  010075  000004                MOV     R0,     @ARG2(R5)       ;RTN DATA
001374  000207                        RTS     PC
001376  012737  010600  172416  2$:   MOV     #4480., @#OPREG         ;WCR : RS(1)
001404  012737  002400  172416        MOV     #1280., @#OPREG         ;RS
001412  013700  172416                MOV     @#IOBUF, R0             ;FETCH DATA
001416  042700  177400                BIC     #177400, R0             ;CLEAR BITS
001422  010075  000004                MOV     R0,     @ARG2(R5)       ;RTN DATA
001426  000207                        RTS     PC
001430  012737  010400  172416  STATB0: MOV     #4352., @#OPREG         ;STATUS BYTE 1
001436  012737  002400  172416        MOV     #1280., @#OPREG         ;RS
001444  013701  172416                MOV     @#IOBUF, R1             ;FETCH DATA
001450  042701  177400                BIC     #177400, R1
001454  132701  000010                BITB    #BIT03, R1              ;BUFFER FULL
001460  001404                        BEQ     3$
001462  012775  000001  000002        MOV     #1,     @ARG1(R5)       ;LOAD RETURN
001470  000207                        RTS     PC
001472  012775  000000  000002  3$:   MOV     #0,     @ARG1(R5)       ;LOAD RETURN
001500  000207                        RTS     PC
```

```
                        ;*********************************************************
                        ;**** PROCEDURE WRITE TOKEN (WTOKEN)                    *
                        ;*********************************************************

                                ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP

001502  012737  010600  172416  WTOKEN: MOV     #4480., @#OPREG                 ;READ STATUS(1)
001510  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001516  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001522  132701  000010                  BITB    #BIT03, R1                      ;BUFFER LOADED
001526  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001530  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001536  012737  004423  172416          MOV     #2323., @#OPREG                 ;MARK 0B0F/0B1F/WTCMD
001544  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001550  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001552  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001556  000207                          RTS     PC
001560  012737  010740  172416  WTKG:   MOV     #4576., @#OPREG                 ;SELECT BUFFER 1
001566  012737  001660' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001574  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001600  005402                          NEG     R2                              ;FORMAT IT
001602  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001606  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001614  000240                          NOP                                     ;DELAY
001616  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001622  000240                          NOP                                     ;ERROR IF HERE
001624  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001630  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001636  012737  004422  172416          MOV     #2322., @#OPREG                 ;MARK 0B1F/WTCMD
001644  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001650  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001652  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001656  000207                          RTS     PC                              ;RETURN
001660     000     377          WTD:    .BYTE 0,377                             ;A WTOKEN

                        ;*********************************************************
                        ;**** PROCEDURE TIMER (TIME)                            *
                        ;*********************************************************

001662  005267  001006' TIME:   INC     LTIME                           ;TIME+1
001666  022767  077777  001006'          CMP     #077777, LTIME                  ;LIMIT
001674  001002                          BNE     4$
001676  005067  001006'                  CLR     LTIME
001702  000002                  4$:     RTI
```

```
                        ;*********************************************************
                        ;**** PROCEDURE ACRAM                                    *
                        ;*********************************************************

                                ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
                                ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
                                ;
                                ;0110   =NREAD
                                ;0100   =DREAD
                                ;0111   =NULL
                                ;0000   =WTOKEN


001704  017500  000002         RAM:    MOV     @ARG1(R5), R0                   ; RAM
001710  022700  000000                 CMP     #0,     R0                      ;WHICH OPERATION
001714  001035                        BNE     RDRAM                           ;READ OP
001716  017500  000004         WTRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS
001722  017501  000006                 MOV     @ARG3(R5), R1                   ;WRITE DATA
001726  012737  010402  172416         MOV     #4354., @#OPREG                 ;SEL LDACTR
001734  062700  004400                 ADD     #2304., R0                      ;WD/DATA (ADDR)
001740  010037  172416                 MOV     R0,     @#OPREG                 ;WRITE DATA
001744  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
001750  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
001752  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
001756  012737  010401  172416         MOV     #4353., @#OPREG                 ;SEL ACRAM
001764  062701  004400                 ADD     #2304., R1                      ;WD/DATA (CMD)
001770  010137  172416                 MOV     R1,     @#OPREG                 ;WRITE
001774  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
002000  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
002002  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002006  000207                        RTS     PC
002010  017500  000004         RDRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS TO READ
002014  012737  010402  172416         MOV     #4354., @#OPREG                 ;SEL LDACTR
002022  062700  004400                 ADD     #2304., R0                      ;WD/DATA (ADDR)
002026  010037  172416                 MOV     R0,     @#OPREG                 ;WRITE DATA
002032  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
002036  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
002040  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002044  012737  010401  172416         MOV     #4353., @#OPREG                 ;SEL ACRAM
002052  012737  001400  172416         MOV     #768.,  @#OPREG                 ;READ DATA
002060  105737  172414                 TSTB    @#CSR                           ;VALID READ
002064  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
002066  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002072  013700  172416                 MOV     @#IOBUF, R0                     ;FETCH DATA
002076  042700  177760                 BIC     #177760, R0                     ;CLEAR BITS
002102  010075  000006                 MOV     R0, @ARG3(R5)                   ;RTN DATA
002106  000207                        RTS     PC
```

```
                                        .ENABLE LSB
                                ;*******************************************************
                                ;**** PROCEDURE WRITE LOOP (LPOUT)                     *
                                ;*******************************************************

002110  017502  000002          LPOUT:  MOV     @ARG1(R5), R2                   ;BYTE COUNT
002114  012700  000340                  MOV     #340,   R0                      ;PRI=7
002120  106400                          MTPS    R0
002122  012737  000000' 172410          MOV     #OUTBF, @#BAR                   ;BUS ADDRESS
002130  005402                          NEG     R2                              ; 2'S COMP COUNT
002132  010237  172412                  MOV     R2,     @#WCR                   ;COUNT
002136  012737  010540  172416          MOV     #4448., @#OPREG                 ;OBO COMMAND
002144  012737  024000  172416          MOV     #10240., @#OPREG                ;DMA GO
002152  000240                          NOP                                     ;INTERFACE TIME
002154  105737  172414                  TSTB    @#CSR                           ;DMA OK
002160  000240                          NOP                                     ;ERROR IF HERE
002162  105037  172414                  CLRB    @#CSR
002166  012737  010740  172416          MOV     #4576., @#OPREG                 ;OB1 COMMAND
002174  012737  001660' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
002202  012702  000002                  MOV     #2,     R2                      ;BYTE COUNT
002206  005402                          NEG     R2                              ;FORMAT FOR BLIUI
002210  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
002214  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
002222  000240                          NOP                                     ;DELAY
002224  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
002230  000240                          NOP                                     ;ERROR IF NOT
002232  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
002236  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
002244  012737  004403  172416          MOV     #2307., @#OPREG                 ;BUFFERS FULL
002252  105737  172414                  TSTB    @#CSR                           ;GOOD WRITE
002256  100375                          BPL     .-4                             ;NO LOOP UNTIL
002260  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002264  012700  000000                  MOV     #000,   R0                      ;PRI=0
002270  106400                          MTPS    R0                              ;LOWER CPU
002272  000207                          RTS     PC
```

```
                        ;*************************************************************
                        ;**** PROCEDURE READ BUFFER POINTER                        *
                        ;*************************************************************

002274  017500  000002          RDPNT:  MOV     @ARG1(R5), R0                   ;FETCH COMMAND
002300  022700  000000                  CMP     #0,     R0                      ;INBUF0 ?
002304  001004                          BNE     1$                              ;NO
002306  012700  010410                  MOV     #4360., R0                      ;RDBUFADR IN0
002312  000167  000046                  JMP     4$
002316  022700  000001          1$:     CMP     #1,     R0                      ;INBUF1 ?
002322  001004                          BNE     2$                              ;NO
002324  012700  010610                  MOV     #4488., R0                      ;RDBUFADR IN1
002330  000167  000030                  JMP     4$
002334  022700  000002          2$:     CMP     #2,     R0                      ;OUTBUF0 ?
002340  001004                          BNE     3$                              ;NO
002342  012700  010510                  MOV     #4424., R0                      ;RDBUFADR OUT0
002346  000167  000012                  JMP     4$
002352  022700  000004          3$:     CMP     #4,     R0                      ;OUTBUF1 ?
002356  001020                          BNE     5$                              ;NO RETURN
002360  012700  010710                  MOV     #4552., R0                      ;RDBUFADR OUT1
002364  010037  172416          4$:     MOV     R0,     @#OPREG                 ;RDBUFADR
002370  012737  001400  172416          MOV     #768.,  @#OPREG                 ;RD
002376  105737  172414                  TSTB    @#CSR                           ;READY
002402  100375                          BPL     .-4                             ;NO LOOP UNTIL
002404  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH POINTER
002410  042700  177400                  BIC     #177400, R0                     ;CLEAR MST
002414  010075  000004                  MOV     R0,     @ARG2(R5)               ;RETURN POINTER
002420  000207                  5$:     RTS     PC
```

```
                        ;*************************************************************
                        ;**** PROCEDURE RESTART                                     *
                        ;*************************************************************

                                ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
                                ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
                                ;CALL RSTART(2) -LOAD MODE(173000)

002422  017500  000002  RSTART: MOV     @ARG1(R5), R0                   ;MODE
002426  022700  000000  RS0:    CMP     #0,     R0                      ;LOAD ADDRESS
002432  001002                  BNE     RS1
002434  000000                  HALT
002436  000207                  RTS     PC
002440  022700  000001  RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
002444  001003                  BNE     RS2
002446  013700  000040  G4$:    MOV     @#40,   R0                      ;START ADDRESS
002452  000110                  JMP     (R0)
002454  022700  000002  RS2:    CMP     #2,     R0                      ;LOAD MODE
002460  001003                  BNE     RST
002462  013700  002472'                  MOV     @#ROM,  R0              ;BOOT ADDRESS
002466  000110                  JMP     (R0)
002470  000207  RST:    RTS     PC
002472  173000  ROM:    .WORD   173000
```

```
                                        .ENABLE LSB
                                ;*************************************************************
                                ;****   PROCEDURE GATE WAY INTERFACE                        *
                                ;*************************************************************


002474  010046                GIO::   MOV     R0,     -(SP)                   ;SAVE REGISTERS
002476  010146                        MOV     R1,     -(SP)
002500  010246                        MOV     R2,     -(SP)
002502  042737  000100  177560        BIC     #100,   @#HRCSR                 ;CLEAR INTERRUPTS
002510  005000                1$:     CLR     R0                              ;BYTE COUNTER
002512  012701  000400'              MOV     #XINBF, R1                      ;BUFFER AREA
                                ;       CLR     PXOR                            ;OR BOX
002516  105737  177560          2$:     TSTB    @#HRCSR                         ;BYTE READY
002522  100375                        BPL     .-4                             ;LOOP UNTIL
002524  113702  177562                MOVB    @#HRBUF, R2                     ;FETCH A BYTE
002530  042702  177400                BIC     #177400, R2                     ;CLEAR MST
                                ;       XOR     R2,PXOR                         ;XOR AND SAVE
002534  110221                        MOVB    R2,(R1)+                        ;SAVE BYTE
002536  005200                        INC     R0                              ;AND COUNT
002540  122702  000003          3$:     CMPB    #003,   R2                      ;THE END (EOP)
002544  001364                        BNE     2$                              ;NO FETCH A BYTE
002546  022700  000006                CMP     #6,     R0
002552  003361                        BGT     2$
                                ;       TSTB    @#HRCSR                         ;READY TO RECEIVE B/C
                                ;       BPL     .-4                             ;WAIT UNTIL
                                ;       MOVB    @#HRBUF, R2                     ;FETCH IT
                                ;       BIC     #177400, R2                     ;CLEAR MST
                                ;       CMPB    R2,PXOR                         ;BLOCK CHECK
                                ;       BNE     4$                              ;NO NAK
                                ;       TSTB    @#HXCSR                         ;YES ACK
                                ;       BPL     .-4                             ;LOOP TILL WE SEND
                                ;       MOVB    #006,@#HXBUF                    ;SEND A ACK
002554  010067  002000                MOV     R0,     IOFLG                   ;PASS BYTE COUNT
                                ;       BR      5$                              ;EXIT
                                ;4$:    TSTB    @#HXCSR                         ;INT. READY
                                ;       BPL     .-4                             ;LOOP UNTIL
                                ;       MOVB    #006,@#HXBUF                    ;SEND A NAK
002560  052737  000100  177560  5$:     BIS     #100,   @#HRCSR
002566  012602                        MOV     (SP)+,  R2                      ;RESTORE REGISTERS
002570  012601                        MOV     (SP)+,  R1
002572  012600                        MOV     (SP)+,  R0
002574  000002                        RTI


                                ;SEND OVER GATE WAY

002576  012700  000340          GOUT::  MOV     #340,   R0                      ;RAISE CPU PRI
002602  106400                        MTPS    R0
002604  017500  000002                MOV     @ARG1(R5),R0                    ;SEND X BYTES
002610  012701  000000'              MOV     #XOUTBF,  R1                    ;BUFFER
002614  042737  000100  177560        BIC     #100,   @#HRCSR
002622  105737  177564                TSTB    @#HXCSR
002626  100375                        BPL     .-4
002630  012737  000005  177566        MOV     #005,   @#HXBUF
002636  105737  177560                TSTB    @#HRCSR
002642  100375                        BPL     .-4
```

```
     58 002644  113702  177562                    MOVB    @#HRBUF, R2
     59 002650  005002                            CLR     R2
     60 002652  112102                    6$:     MOVB    (R1)+,  R2              ;FETCH BYTE
     61 002654  042702  177400                    BIC     #177400, R2
     62 002660  105737  177564                    TSTB    @#HXCSR                 ;READY TO SEND
     63 002664  100375                            BPL     .-4                     ;LOOP UNTIL
     64 002666  110237  177566                    MOVB    R2,     @#HXBUF         ;SEND THE BYTE
     65 002672  000240                            NOP
     66 002674  000240                            NOP
     67 002676  077013                            SOB     R0,     6$              ;SEND ALL BYTES
     68 002700  012700  000000                    MOV     #000,   R0              ;LOWER PRI
     69 002704  106400                            MTPS    R0
     70 002706  052737  000100  177560            BIS     #100,   @#HRCSR
     71 002714  000207                            RTS     PC                      ;ALL DONE
     72 002716  000000                    PXOR:   .WORD   0
     73 002720                             DUMP:   .BLKW   50.
    124
```

```
                                ;*************************************************************
                                ;****        COMMON DATA AREA                                 *
                                ;*************************************************************

000000                                  .PSECT  DFM,RW,D,GBL,REL,OVR

000000                          OUTBF:  .BLKB   256.
000400                          INBF:   .BLKB   256.
001000                          IWRTTM: .BLKW
001002                          IWRT:   .BLKW
001004                          STAT:   .BLKW
001006                          LTIME:  .BLKW
                                .EVEN

000000                                  .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                          XOUTBF: .BLKB   256.
000400                          XINBF:  .BLKB   256.
001000                          ACKSEQ: .BLKB   256.
001400                          LIDFD:  .BLKB   256.
002000                          IOFLG:  .BLKW
002002                          LLFLG:  .BLKW
002004                          IRSEND: .BLKW
                                .EVEN


        000001                  .END
```

```
FDM.MACRO       MACRO V03.01 4-JUN-79 17:09:17 PAGE 12-1
SYMBOL TABLE

ACKSEQ  001000R     003 B1$     000730R         E0$     001234R         LPINPT= ****** G        RTI$    001006R
AREA    001214R         B2$     000516R         E1$     001244R         LPOUT   002110RG        STAT    001004R     002
ARG1  = 000002          B3$     000542R         GIO     002474RG        LTIME   001006R     002 STATBO  001430RG
ARG2  = 000004          B4$     000566R         GOUT    002576RG        MASTER  000000RG        STATUS  001332RG
ARG3  = 000006          B5$     000614R         G4$     002446R         NDXX  = 000000            STAT$   000412R
ARG4  = 000010          B6$     000632R         HRBUF = 177562          ND24  = 000001            SWITCH  001274RG
BAR   = 172410          B7$     000576R         HRCSR = 177560          NHOST = 000000            TIME    001662RG
BIT00 = 000001          CAUSE   001212R         HXBUF = 177566          OPREG = 172416            WCR   = 172412
BIT01 = 000002          CRCOK   001062R         HXCSR = 177564          OUTBF   000000R     002 WTD     001660R
BIT02 = 000004          CSR   = 172414          INBF    000400R     002 PXOR    002716R         WTKG    001560R
BIT03 = 000010          DATA    001220R         IOBUF = 172416          RAM     001704RG        WTOKEN  001502RG
BIT04 = 000020          DESTR   001264RG        IOFLG   002000R     003 RDPNT   002274RG        WTRAM   001716R
BIT05 = 000040          DHOST = 000001          IRSEND  002004R     003 RDRAM   002010R         XCIO  = 000001
BIT06 = 000100          DMAOK   001166R         IWRT    001002R     002 ROM     002472R         XGIO  = 000000
BIT07 = 000200          DUMP    002720R         IWRTTM  001000R     002 RST     002470R         XINBF   000400R     003
BIT08 = 000400          EMBF    001032R         LIDFD   001400R     003 RSTART  002422RG        XOUTBF  000000R     003
BIT14 = 040000          ENABLE  001222RG        LIO     000436RG        RS0     002426R         XSIO  = 000001
BIT15 = 100000          END$    001210R         LIUINT  000104RG        RS1     002440R         ZEROBP  000312R
BO$     000652R         ENSTR   001254RG        LLFLG   002002R     003 RS2     002454R

. ABS.  000000      000
        003064      001
DFM     001010      002
BUFS    002006      003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:FDM=DK:FDM
```

```
FOR NODAL/NOLINE
FOR NODAL1/NOLINE
MAC FDM
```

```
ASSIGN DX0: DK:
R LINK
DX1:NODE21.SAV,DX1:MAP=DX1:NODAL/C/I/W
DX1:FDM/C
DX1:NODAL1//
$SIMRT

ASSIGN DX1: DX:
```

1.3 NODE 22 (VSQC)

The Voice Service Quality Control (VSQC) module assesses the performance of voice channels for the purpose of detecting degrading performance and assisting in fault isolation. The MSCDM is capable of monitoring at least 1000 channels. By convention, channels numbered 1-500 are monitored by the VSQC. There are six parameters to be checked per channel.

1. Peak Power (PK)
2. Average Power (AV)
3. Frequency Offset (FO)
4. Phase Jitter (PJ)
5. C-Message Noise (CN)
6. 3 KHz Flat Noise (FN)

Values for these parameters are generated by the Simulated Input Generator (SIG) microcomputer. Thresholding is done on the parameters by comparison with the Red High (RH), Red Low (RL), Amber High (AH), and Amber Low (AL) values in order to determine whether the channel parameters are in the Red, Amber or Green region.

VSQC Parameter Thresholds

Parameter	RH	RL	AH	AL
PK	10	0	9	2
AV	10	0	8	2
FO	20	-100	0	-80
PJ	10	-90	-10	-70
CN	0	-80	-10	-60
FN	20	-100	0	-80

The VSQC also performs trending on the parameter values. Trending begins when the parameter is within a delta value of the Threshold. The critical trending values for the VSQC parameter thresholds are given below.

VSQC Critical Trending Values

Parameter	Delta	RHT	RLT	AHT	ALT
PK	0.5	9.5	0.5	7.5	2.5
AV	0.5	9.5	0.5	7.5	2.5
FO	3	17	-97	-3	-77
PJ	2	8	-88	-8	-68
CN	1	-1	-79	-19	-59
FN	2	18	-98	-2	-78

The threshold and trending values may be modified by modifying the program DATA statement and recompiling the VSQC nodal software.

Event Reports are sent to the FIAC when a parameter value is Amber or Red and the Event Reporting condition parameter is turned ON (via Module Update Mode 3 of the User Language). The Event Report consists of an 8-byte trunk name, 4-byte channel number, one-byte condition, one-byte parameter number that caused the Red or Amber condition, 3-byte monitor point number, and the node designator to which reports should be sent.

The VSQC interprets command from the DBMS generated by Mode 3 of the User Language. The commands include Event Reporting ON or OFF; if ON, the node designator to which the Event Reports are to be sent, i.e., the terminal used for display, is given. The VSQC also interprets the command, received as a packet from the loop, to measure a specific channel which is sent to the SIG; the resulting measurement value is sent to the OCRI terminal requesting the measurement (via Mode 3 of the User Language).

1.3.1 Program Descriptions

1.3.1.1 Refer to Section 1.1 for descriptions of routines NODAL, IGETSP, ENQUE, DEQUE, ACKNAK, INPTQ, LPINPT, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME RAM, LPOUT, RDPNT, RSTART.

1.3.1.2 Subroutine VSQC (FORTRAN)

This subroutine is called from the nodal program when a packet has been received from the SIG interface. It decodes the simulated input values read from the SIG and performs the function described in Section 1.3.

1.3.1.3 Subroutine TREND (FORTRAN)

This subroutine is called by the VSQC routine to perform the trending on the values sent by the SIG. This value is compared with DELTA values; when a channel consistently is in the DELTA region, an event report is generated.

1.3.1.4 Subroutine VSQLP (FORTRAN)

The subroutine is called from program NODAL when a packet is received from the loop. It performs one of three functions: turns reporting on or off, and takes a measurement of a specified channel.

1.3.1.5 Subroutine SIO (MACRO)

The subroutine is called from an interrupt received by the SIG interface; it then reads the message and sets a flag containing the byte count of the message.

1.3.1.6 Subroutine SOUT (MACRO)

This subroutine is called from subroutine VSQC and passes to the SIG processor the channel number of a channel for which the user language requests a measurement.

1.3.1.7 Subroutine READY (MACRO)

This subroutine is called to signal the SIG that VSQC is ready for another measurement.

```
0001          PROGRAM NODAL
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003          INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP
0005          INTEGER*2 RSTBKP,Q1,Q2,RESNLM,OUTQ,DEQUE
0006          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007          LOGICAL*1 IDATA
0008          LOGICAL*1 PACK,ICODE,ETX,CR,LF,MONTOR,ISLID,DUM
0009          INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0010          REAL*4 RH,RL,AH,AL,RHT,RLT,AHT,ALT,DELTA,IRSV,VMEAS
0011          REAL*8 VTR
0012          COMMON /MESS/ MESSEQ
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0014          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0015          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1          LIDFD(256),IOFLG,LLFLG,IRSEND
0016          COMMON /FRE/ FREE(64),IFR,IFRSZ
0017          COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0018          COMMON        PACK(256,64)
0019          COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0020          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0021          COMMON /VCOM/ ISWTCH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSW,
             1     AL(6),AH(6),DELTA(6),ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10)
0022          CALL MASTER
0023          CALL INIT
0024          CALL ENABLE(1)
0025        5 CALL READY
0026       25 IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0028          CALL ENABLE(0)
0029          Q2=DEQUE(PINQ,INQ,1)
0030          CALL ENABLE(1)
0031          DO 30 I=1,10
0032            IDATA(I)=PACK(I+6,Q2)
0033       30 CONTINUE
0034          CALL VSQCLP
0035          IFR=IFR+1
0036          FREE(IFR)=Q2
0037       40 CALL STATBO(IS)
0038          IF(IS .EQ. 1) GOTO 100
0040          IF(IOFLG .GT. 0 .AND. IRSEND .EQ. 0) CALL VSQC
0042          IF(LLFLG .GT. 0) CALL LINLOS
0044       80 IF(OUTFCT .EQ. 1) GOTO 100
0046          IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0048          IF(IRSEND .EQ. 1) GOTO 85
0050          IF(PXINQ(1) .LT. PXINQ(2)) GOTO 100
0052          CALL ENABLE(0)
0053          Q1=DEQUE(PXINQ,XINQ,1)
0054          CALL ENABLE(1)
0055          OUTQ=Q1
0056       85 CALL DESTR(PACK(254,OUTQ),Q2)
0057          DO 90 I=1,Q2
```

```
0058            OUTBF(I)=PACK(I,OUTQ)
0059      90    CONTINUE
0060            OUTBF(Q2+1)=0
0061            IPT=OUTBF(5)
0062            OUTBF(Q2+2)=LIDFD(IPT)
0063            IFULL=1
0064            ISENT=1
0065            CALL LPOUT(Q2+2)
0066            IRSEND=1
0067            ACKTIM=0
0068            IWRTTM=0
0069            IWRT=1
0070    100     INFLCT=0
0071    110     OLDTIM=NEWTIM
0072            NEWTIM=LTIME
0073            IF(IWRT .EQ. 0) GOTO 120
0075            IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0076            IF(IWRTTM .LT. TIMLIM) GOTO 120
0078            CALL WTOKEN
0079            IWRTTM=0
0080            IWRT=0
0081    120     IF(ISENT .EQ. 0) GOTO 130
0083            ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0084            IF(ACKTIM .LT. ATIMLM) GOTO 130
0086            CALL ENABLE(0)
0087            CALL ACKNAK(0)
0088            CALL ENABLE(1)
0089    130     CONTINUE
0090            GOTO 5
0091            END
```

```
0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(64),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IF(IFR .LT. 1) CALL INIT
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```

```
FORTRAN IV     V02.1-1    Mon 04-Jun-79 17:28:15              PAGE 001

0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1         ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008       10 IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016       20 IF(IQHEAD .GE. (IQLNTH)) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020       30 B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024       40 CALL INIT
0025      999 RETURN
0026          END
```

```
FORTRAN IV     V02.1-1    Mon 04-Jun-79 17:28:28              PAGE 001

0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNUM
0005          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1         ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNUM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)          !HEAD OF QUEUE
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016       10 A(1)=IQHEAD-1
0017      999 RETURN
0018          END
```

```
0001          SUBROUTINE ACKNAK(N)
0002          INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ,PXOUTQ
0003          INTEGER*2 ACKQ,PACKQ,PINQ,FREE,OUTQ,XOUTQ
0004          INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006          LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008          INTEGER*2 LTIME,IWRTTM
0009          REAL*8 COMT(5)
0010          EQUIVALENCE(COMT,LCOMT)
0011          DATA COMT(1)/'MESSAGE '/,COMT(2)/'NOT SENT'/,COMT(3)/' FROM    '/,
             1    COMT(4)/'NODE     '/,COMT(5)/' TO NODE'/
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1       IWRTTM,IWRT,STAT,LTIME
0014          COMMON         PACK(256,64)
0015          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1          PACKQ(2),INQ(16),PINQ(2)
0016          COMMON /FRE/  FREE(64),IFR,IFRSZ
0017          COMMON /MESS/ MESSEQ
0018          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1    IDFLG,LLFLG,IRSEND
0019          IF (N .NE. 1) GOTO 10
0021          IFR=IFR+1
0022          DO 5 I=1,256
0023            PACK(I,OUTQ)=0
0024        5 CONTINUE
0025          IFULL=0
0026          ISENT=0
0027          IRSEND=0
0028          FREE(IFR)=OUTQ
0029          GOTO 999
0030       10 T2=PACK(256,OUTQ)
0031          T2=T2+1
0032          IF(T2 .LE. RESNLM) GOTO 120
0034          T3=PACK(3,OUTQ)
0035          T4=MOD(T3,64)
0036          IF(T4 .LT. 32) GOTO 110
0038          T5=IGETSP(N)
0039          IF(MESSEQ .EQ. 126) MESSEQ=0
0041          MESSEQ=MESSEQ+1
0042          PACK(1,T5)=0
0043          PACK(2,T5)=MESSEQ
0044          PACK(3,T5)=0
0045          PACK(4,T5)=0
0046          PACK(5,T5)=25
0047          PACK(6,T5)=ISLID
0048          DO 20 I=7,60
0049            PACK(I,T5)='040
0050       20 CONTINUE
0051          DO 30 I=7,9
0052            PACK(I,T5)=LF
0053       30 CONTINUE
```

```
0054          DO 40 I=1,8
0055            PACK(I+9,T5)=LCOMT(I)
0056     40   CONTINUE
0057          DO 50 I=9,16
0058            PACK(I+9,T5)=LCOMT(I)
0059     50   CONTINUE
0060          DO 60 I=17,22
0061            PACK(I+9,T5)=LCOMT(I)
0062     60   CONTINUE
0063          DO 70 I=25,29
0064            PACK(I+7,T5)=LCOMT(I)
0065     70   CONTINUE
0066          ENCODE(3,80,PACK(38,T5))ISLID
0067     80   FORMAT(I3)
0068          DO 90 I=33,40
0069            PACK(I+11,T5)=LCOMT(I)
0070     90   CONTINUE
0071          ENCODE(3,80,PACK(52,T5))PACK(5,OUTQ)
0072          PACK(55,T5)=LF
0073          PACK(56,T5)=LF
0074          PACK(57,T5)=CR
0075          PACK(58,T5)=ETX
0076          CALL ENSTR(PACK(254,T5),58)
0077          CALL ENQUE(PXINQ,XINQ,T5)
0078          DO 102 I=1,256
0079            PACK(I,OUTQ)=0
0080    102   CONTINUE
0081          ISENT=0
0082          IRSEND=0
0083          IFULL=0
0084          IWRT=0
0085          IFR=IFR+1
0086          FREE(IFR)=OUTQ
0087          GOTO 999
0088     110  PACK(3,OUTQ)=PACK(3,OUTQ)+32
0089          PACK(256,OUTQ)=0
0090          ISENT=0
0091          IFULL=0
0092          IF(IRSEND .EQ. 1) GOTO 999
0094          CALL ENQUE(PXINQ,XINQ,OUTQ)
0095          GOTO 999
0096    120   PACK(256,OUTQ)=T2
0097          IFULL=0
0098          ISENT=0
0099          IF(IRSEND .EQ. 1) GOTO 999
0101          CALL ENQUE(PXINQ,XINQ,OUTQ)
0102    999   RETURN
0103          END
```

```
FORTRAN IV     V02.1-1     Mon 04-Jun-79 17:29:11                PAGE 001

0001          SUBROUTINE INPTQ(L)
0002          INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
0003          INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005          LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
0006          INTEGER*2 LTIME,IWRTTM
0007          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IOFLG,LLFLG,IRSEND
0009          COMMON         PACK(256,64)
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          IF(L .LT. 1) GOTO 30
0014          LI=L-2
0015          II=INBF(6)
0016          IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0018          T1=IGETSP(N)
0019          DO 10 I=1,LI
0020       10 PACK(I,T1)=INBF(I)
0021          CALL ENSTR(PACK(254,T1),LI)
0022          CALL ENQUE(PINQ,INQ,T1)
0023          N=2
0024          GOTO 40
0025       20 N=1
0026          GOTO 40
0027       30 N=0
0028       40 DO 50 I=1,2
0029            OUTBF(I)=INBF(I)
0030       50 CONTINUE
0031          NN=1
0032          IF(N .EQ. 0) NN=128
0034          OUTBF(3)=NN
0035          OUTBF(4)=0
0036          OUTBF(5)=INBF(6)
0037          OUTBF(6)=ISLID
0038          OUTBF(7)=ETX
0039          OUTBF(8)=0
0040          OUTBF(9)=LIDFD(INBF(6))
0041          CALL LPOUT(9)
0042          IF(N .NE. 2) GOTO 999
0044          ACKSEQ(II)=INBF(2)
0045      999 RETURN
0046          END
```

```
FORTRAN IV     V02.1-1     Mon 04-Jun-79 17:29:28                PAGE 001

0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON/DFM/OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          CC1=INBF(3)
0009          CC2=INBF(4)
0010          IF(CC1 .LT. 0) GOTO 25
0012          IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0014          CC3=MOD(CC1,2)
0015          CC4=MOD(CC1,256)
0016          IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0018          IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0020          CC5=MOD(INBF(3),64)
0021          IF(CC5 .GE. 32) CALL INPTQ(LI)
0023          GOTO 99
0024     20   CC7=MOD(CC1,2)
0025          IF(CC7 .GE. 1) GOTO 30
0027     25   CALL ACKNAK(0)
0028          GOTO 99
0029     30   CALL ACKNAK(1)
0030          GOTO 99
0031     40   CALL INPTQ(LI)
0032     99   CONTINUE
0033          RETURN
0034          END
```

```
0001          SUBROUTINE INIT
0002          REAL*4 RH,RL,AH,AL,IRSV,VMEAS
0003          REAL*8 VTR
0004          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0005          INTEGER*2 STAT,FLWCNT,OUTFCT
0006          INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0008          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM,IDATA
0009          INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0010          COMMON /DFM/ OUTBF(256),INBF(256),
             1     IWRTTM,IWRT,STAT,LTIME
0011          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1              ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0012          COMMON /MESS/ MESSEQ
0013          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0014          COMMON /FRE/  FREE(64),IFR,IFRSZ
0015          COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0016          COMMON        PACK(256,64)
0017          COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0018          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0019          COMMON /VCOM/ ISWTCH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSW,
             1     AL(6),AH(6),DELTA(6),ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10)
0020          CALL RAM(0,3,4)
0021          CALL RAM(0,255,0)
0022          MESSEQ=0
0023          IOFLG=0
0024          IRSEND=0
0025          LLFLG=0
0026          IFRSZ=64
0027          INFLCT=0
0028          ISLID=22
0029          CR='015
0030          LF='012
0031          ETX='003
0032          ICHNSW=0
0033          RESNLM=2
0034          IQLNTH=16
0035          IALTRT=0
0036          FLWCNT=0
0037          IWRTTM=0
0038          TIMLIM=50
0039          SETPRM=4
0040          RSTPRM=32
0041          SETBKP=8
0042          RSTBKP=64
0043          ATIMLM=200
0044          MONTOR=27
0045          IWRT=0
0046          ISWTCH=0
0047          OUTFCT=0
0048          IFULL=0  .
0049          ISENT=0
```

```
0050          DO 40 I=1,20
0051     40   LIDFD(I)=4
0052          LIDFD(21)=1
0053          LIDFD(22)=3
0054          LIDFD(23)=6
0055          LIDFD(24)=5
0056          LIDFD(25)=7
0057          LIDFD(26)=8
0058          LIDFD(27)=9
0059          LIDFD(28)=2
0060          DO 50 I=29,39
0061     50   LIDFD(I)=0
0062          DO 60 I=40,44
0063     60   LIDFD(I)=1
0064          DO 70 I=45,59
0065     70   LIDFD(I)=0
0066          DO 80 I=60,64
0067     80   LIDFD(I)=2
0068          DO 90 I=65,79
0069     90   LIDFD(I)=0
0070          DO 100 I=80,84
0071    100   LIDFD(I)=4
0072          DO 110 I=85,99
0073    110   LIDFD(I)=0
0074          DO 120 I=100,104
0075    120   LIDFD(I)=5
0076          DO 130 I=105,256
0077    130   LIDFD(I)=0
0078          PINQ(1)=IQLNTH
0079          PINQ(2)=IQLNTH+1
0080          PACKQ(1)=IQLNTH
0081          PACKQ(2)=IQLNTH+1
0082          PXOUTQ(1)=IQLNTH
0083          PXOUTQ(2)=IQLNTH+1
0084          PXINQ(1)=IQLNTH
0085          PXINQ(2)=IQLNTH+1
0086          DO 10 I=1,IFRSZ
0087            PACK(255,I)=0
0088     10   PACK(256,I)=0
0089          DO 20 I=1,IFRSZ
0090     20   FREE(I)=I
0091          DO 30 I=1,256
0092     30   ACKSEQ(I)=256
0093          IFR=IFRSZ
0094          RETURN
0095          END
```

```
0001          SUBROUTINE LINLOS
0002          REAL*8 RM1(5),LIN08,LIN18
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,PINQ
0005          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0007          LOGICAL*1 PACK,M1(40),LIN0(8),LIN1(8)
0008          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0009          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &            IOFLG,LLFLG,IRSEND
0010          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011          COMMON         PACK(256,64)
0012          COMMON /MESS/ MESSEQ
0013          DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0014          DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0015          DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0016          DATA RM1(5)/'DE 22   '/
0017          EQUIVALENCE(LIN0,LIN08)
0018          EQUIVALENCE(LIN1,LIN18)
0019          EQUIVALENCE(M1,RM1)
0020          IS=LLFLG
0021          CALL ENABLE(0)
0022          T1=IGETSP(N)
0023          CALL ENABLE(1)
0024          PACK(1,T1)=0
0025          IF(MESSEQ .EQ. 126) MESSEQ=0
0027          MESSEQ=MESSEQ+1
0028          PACK(2,T1)=MESSEQ
0029          PACK(3,T1)=0
0030          PACK(4,T1)=0
0031          PACK(5,T1)=25
0032          PACK(6,T1)=ISLID
0033          DO 20 I=7,9
0034            PACK(I,T1)=LF
0035       20 CONTINUE
0036          DO 30 I=1,22
0037       30 PACK(I+9,T1)=M1(I)
0038          IF(IS .EQ. 1) GOTO 50
0040          DO 40 I=1,8
0041       40 PACK(I+31,T1)=LIN0(I)
0042          GOTO 70
0043       50 DO 60 I=1,8
0044       60 PACK(I+31,T1)=LIN1(I)
0045       70 DO 80 I=23,40
0046       80 PACK(I+17,T1)=M1(I)
0047          PACK(58,T1)=CR
0048          PACK(59,T1)=LF
0049          PACK(60,T1)=ETX
0050          CALL ENSTR(PACK(254,T1),60)
0051          CALL ENABLE(0)
0052          CALL ENQUE(PXINQ,XINQ,T1)
0053          CALL ENABLE(1)
```

```
0054          LLFLG=0
0055          RETURN
0056          END
```

```
0001          SUBROUTINE VSQC
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003          INTEGER*2 PINQ,FLWCNT,OUTFCT,OUTQ,RESNLM
0004          REAL*8 VTR
0005          REAL*4 RH,RL,AH,AL,RHT,RLT,AHT,ALT,DELTA,IRSV,VMEAS
0006          LOGICAL*1 PACK,ISLID,MONTOR,LF,ETX,CR,DUM,INQUE
0007          LOGICAL*1  XOUTBF,XINBF,ACKSEQ,LIDFD,IDATA
0008          COMMON PACK(256,64)
0009          COMMON /MESS/ MESSEQ
0010          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0011          COMMON /VCOM/ ISWTCH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSW,
             1     AL(6),AH(6),DELTA(6),ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),
             2     ICHAN
0012          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1     PACKQ(2),INQ(16),PINQ(2)
0013          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,RESNLM,
             1     IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0014          DATA RH(1),RH(2),RH(3),RH(4) /99.5,99.5,199.0,199.0/
0015          DATA RH(5),RH(6) /59.1,299.0/
0016          DATA RL(1),RL(2),RL(3),RL(4) /-90.5,-90.5,-270.0,-275.0/
0017          DATA RL(5),RL(6) /-118.9,-270.0/
0018          DATA AH(1),AH(2),AH(3),AH(4) /90.0,90.0,192.0,191.5/
0019          DATA AH(5),AH(6) /56.5,199.0/
0020          DATA AL(1),AL(2),AL(3),AL(4) /-50.0,-55.0,-251.0,-257.0/
0021          DATA AL(5),AL(6) /-113.5,-252.0/
0022          DATA DELTA(1),DELTA(2),DELTA(3) /.5,.5,3.0/
0023          DATA DELTA(4),DELTA(5),DELTA(6) /2.0,1.0,2.0/
0024          DECODE(2,20,XINBF(1),ERR=410) ICNT
0025          DECODE(4,30,XINBF(3),ERR=410) VMEAS(1)
0026          DECODE(4,30,XINBF(7),ERR=410) VMEAS(2)
0027          DECODE(4,30,XINBF(11),ERR=410) VMEAS(3)
0028          DECODE(4,30,XINBF(15),ERR=410) VMEAS(4)
0029          DECODE(4,30,XINBF(19),ERR=410) VMEAS(5)
0030          DECODE(4,30,XINBF(23),ERR=410) VMEAS(6)
0031          DECODE(2,20,XINBF(27),ERR=410) IVC
0032          DECODE(2,20,XINBF(37),ERR=410) MON
0033       20 FORMAT(A2)
0034       30 FORMAT(A4)
0035          IF(ICHNSW .EQ. 1) GOTO 400
0037          DO 10 I=1,6
0038            IRSV(I)=(AH(I)+AL(I)) / 2
0039       10 CONTINUE
0040          DO 350 ICTR=1,6
0041          ALT=AL(ICTR)+DELTA(ICTR)
0042          AHT=AH(ICTR)+DELTA(ICTR)
0043          RLT=RL(ICTR)+DELTA(ICTR)
0044          RHT=RH(ICTR)+DELTA(ICTR)
0045          IF(VMEAS(ICTR) .GE. ALT .AND.
             1    VMEAS(ICTR) .LE. AHT) GOTO 350
0047          IF((VMEAS(ICTR) .GT. AL(ICTR) .AND.
             1    VMEAS(ICTR) .LT. ALT) .OR.
             2   (VMEAS(ICTR) .GT. AHT .AND.
             3    VMEAS(ICTR) .LT. AH(ICTR))) GOTO 100
```

```
0049          IF((VMEAS(ICTR) .GE. AH(ICTR) .AND.
             1    VMEAS(ICTR) .LE. RHT) .OR.
             2   (VMEAS(ICTR) .GE. RLT .AND.
             3    VMEAS(ICTR) .LE. AL(ICTR))) GOTO 200
0051          IF((VMEAS(ICTR) .GT. RHT .AND.
             1    VMEAS(ICTR) .LT. RH(ICTR)) .OR.
             2   (VMEAS(ICTR) .GT. RLT .AND.
             3    VMEAS(ICTR) .LT. RL(ICTR))) GOTO 300
0053          IF(ISWTCH .EQ. 1) GOTO 45
0055          IOFLG=0
0056          RETURN
0057     45   CALL ENABLE(0)
0058          K1=IGETSP(N)
0059          CALL ENABLE(1)
0060          DO 46 I=1,256
0061          PACK(I,K1)=0
0062     46   CONTINUE
0063          IF(MESSEQ .EQ. 126) MESSEQ=0
0065          MESSEQ=MESSEQ+1
0066          PACK(1,K1)=0
0067          PACK(2,K1)=MESSEQ
0068          PACK(3,K1)=0
0069          PACK(4,K1)=0
0070          PACK(5,K1)=MONTOR
0071          PACK(6,K1)=ISLID
0072          DO 50 I=7,14
0073            PACK(I,K1)=XINBF(I+22)
0074     50   CONTINUE
0075          DO 60 I=15,16
0076            PACK(I,K1)=XINBF(I+12)
0077     60   CONTINUE
0078          PACK(17,K1)=1
0079          ENCODE(4,65,PACK(18,K1)) VMEAS(ICTR)
0080     65   FORMAT(A4)
0081          ENCODE(2,70,PACK(22,K1),) MON
0082     70   FORMAT(A2)
0083          PACK(24,K1)=ITERM
0084          CALL ENSTR(PACK(254,K1),24)
0085          CALL ENABLE(0)
0086          CALL ENQUE(PXINQ,XINQ,K1)
0087          CALL ENABLE(1)
0088          GOTO 410
0089    100   IF(ISWTCH .EQ. 1) GOTO 110
0091          GOTO 410
0092    110   CALL TREND(2,VMEAS(ICTR))
0093          GOTO 410
0094    200   IF(ISWTCH .EQ. 1) GOTO 210
0096          GOTO 410
0097    210   CALL TREND(2,VMEAS(ICTR))
0098          GOTO 410
0099    300   IF(ISWTCH .EQ. 1) GOTO 310
0101          GOTO 410
0102    310   CALL TREND(2,VMEAS(ICTR))
0103          GOTO 410
```

```
0104    350  CONTINUE
0105         GOTO 410
0106    400  CALL ENABLE(0)
0107         K1=IGETSP(N)
0108         CALL ENABLE(1)
0109         DO 405 I=1,256
0110         PACK(I,K1)=0
0111    405  CONTINUE
0112         IF(MESSEQ .EQ. 126) MESSEQ=0
0114         MESSEQ=MESSEQ+1
0115         PACK(1,K1)=0
0116         PACK(2,K1)=MESSEQ
0117         PACK(3,K1)=0
0118         PACK(4,K1)=0
0119         PACK(5,K1)=25
0120         PACK(6,K1)=ISLID
0121         PACK(7,K1)=CR
0122         PACK(8,K1)=LF
0123         ENCODE(4,360,PACK(9,K1)) ICHAN
0124         PACK(13,K1)='040
0125         ENCODE(12,370,PACK(14,K1)) VMEAS(1)
0126         PACK(26,K1)='040
0127         ENCODE(12,370,PACK(27,K1)) VMEAS(2)
0128         PACK(39,K1)='040
0129         ENCODE(12,370,PACK(40,K1)) VMEAS(3)
0130         PACK(52,K1)='040
0131         ENCODE(12,370,PACK(53,K1)) VMEAS(4)
0132         PACK(65,K1)='040
0133         ENCODE(12,370,PACK(66,K1)) VMEAS(5)
0134         PACK(78,K1)='040
0135         ENCODE(12,370,PACK(79,K1)) VMEAS(6)
0136         PACK(91,K1)=CR
0137         PACK(92,K1)=LF
0138         PACK(93,K1)=ETX
0139         CALL ENSTR(PACK(254,K1),93)
0140         CALL ENABLE(0)
0141         CALL ENQUE(PXINQ,XINQ,K1)
0142         CALL ENABLE(1)
0143         ICHNSW=0
0144    360  FORMAT(I4)
0145    370  FORMAT(F12.6)
0146    410  IOFLG=0
0147         RETURN
0148         END
```

```
0001          SUBROUTINE TREND(ICND,IVAL)
0002          REAL*4 RH,RL,AH,AL,RHT,RLT,AHT,ALT,DELTA,IRSV,VMEAS,IVAL
0003          REAL*8 VTR
0004          LOGICAL*1 PACK,ISLID,MONTOR,LF,ETX,CR,DUM,INQUE,IDATA
0005          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,PINQ
0007          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0008          COMMON PACK(256,64)
0009          COMMON /MESS/ MESSEQ
0010          COMMON /VCOM/ ISWTCH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSW,
             1     AL(6),AH(6),DELTA(6),ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),
             2     ICHAN
0011          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1     PACKQ(2),INQ(16),PINQ(2)
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,RESNLM,
             1     IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             $     IOFLG,LLFLG,IRSEND
0014           IU=(RH(ICTR)-RL(ICTR))/30
0015           IF((IRSV(ICTR)-IVAL) .GT. (4 * IU)) GOTO 20
0017           IF(NFL .GE. 3) GOTO 40
0019           IF((IRSV(ICTR)-IVAL) .GT. (2 * IU)) GOTO 30
0021           NFL=NFL+1
0022     10    IRSV(ICTR)=(.33 * IVAL + .67 * IRSV(ICTR))
0023           GOTO 60
0024     20    IF(SFL .GE. 2) GOTO 30
0026           SFL=SFL+1
0027           GOTO 50
0028     30    IRSV(ICTR)=IVAL
0029           NFL=0
0030           GOTO 50
0031     40    SFL=0
0032           GOTO 10
0033     50    IF(ICND .EQ. 2) RETURN
0035     60    CALL ENABLE(0)
0036           K1=IGETSP(N)
0037           CALL ENABLE(1)
0038           IF(MESSEQ .EQ. 126) MESSEQ=0
0040           MESSEQ=MESSEQ+1
0041           PACK(1,K1)=0
0042           PACK(2,K1)=MESSEQ
0043           PACK(3,K1)=0
0044           PACK(4,K1)=0
0045           PACK(5,K1)=MONTOR
0046           PACK(6,K1)=ISLID
0047           DO 70 I=7,14
0048             PACK(I,K1)=XINBF(I+22)
0049     70    CONTINUE
0050           DO 75 I=15,16
0051             PACK(I,K1)=XINBF(I+12)
0052     75    CONTINUE
0053           ENCODE(4,80,PACK(18,K1)) IVAL
0054     80    FORMAT(A4)
0055           PACK(17,K1)=ICND
```

```
0056          ENCODE(2,90,PACK(22,K1)) MON
0057     90   FORMAT(A2)
0058          PACK(24,K1)=ITERM
0059          CALL ENSTR(PACK(254,K1),24)
0060          CALL ENABLE(0)
0061          CALL ENQUE(PXINQ,XINQ,K1)
0062          CALL ENABLE(1)
0063          RETURN
0064          END
```



```
0001
0002
0003
0004
0005
0006
0007
0008     10
0009
0010
0012
0013
0014
0015
0016     20
0018
0019
0020     30
0021
0022
0023
0024     40
0025    999
0026
```

```
FORTRAN IV     V02.1-1    Mon 04-Jun-79 18:12:13              PAGE 001

0001          SUBROUTINE VSQCLP
0002          LOGICAL*1 IDATA
0003          REAL*4 RH,RL,AH,AL,RHT,RLT,AHT,ALT,DELTA,IRSV,VMEAS
0004          REAL*8 VTR
0005          COMMON /VCOM/ ISWTCH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSW,
             1     AL(6),AH(6),DELTA(6),ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),
             2     ICHAN
0006          IF(IDATA(1) .EQ. '117 .AND.
             1  IDATA(2) .EQ. '116) GOTO 50
0008          IF(IDATA(1) .EQ. '117 .AND.
             1  IDATA(2) .EQ. '106 .AND.
             2  IDATA(3) .EQ. '106) GOTO 60
0010          IF(IDATA(1) .EQ. '115) GOTO 70
0012          GOTO 80
0013     50   ISWTCH=1
0014          ITERM=IDATA(4)
0015          RETURN
0016     60   ISWTCH=0
0017          RETURN
0018     70   DECODE(4,75,IDATA(2),ERR=80) ICHAN
0019     75   FORMAT(I)
0020          CALL SOUT(ICHAN)
0021          ICHNSW=1
0022     80   RETURN
0023          STOP
0024          END
```

```
                                .TITLE  FDM.MACRO
                                .SBTTL  NODE22
                                .IDENT  /V3.0/
                                .GLOBL  LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
                                .GLOBL  WTOKEN,RAM,STATUS,STATBO
                                .GLOBL  LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
                                .NLIST  CND
000000                          .PSECT

000000          R0      =%0
000001          R1      =%1
000002          R2      =%2
000003          R3      =%3
000004          R4      =%4
000005          R5      =%5
000006          SP      =%6
000007          PC      =%7

                ;INTERFACE ADDRESS

172410          BAR=    172410
172412          WCR=    172412
172414          CSR=    172414
172416          IOBUF=  172416
172416          OPREG=  172416
177560          HRCSR=  177560
177562          HRBUF=  177562
177564          HXCSR=  177564
177566          HXBUF=  177566

                ;INTERFACE OPTIONS

000001          XCIO=1
000000          XSIO=0
000001          XGIO=1
000000          NDXX=0
000001          ND24=1
000000          NHOST=0
000001          DHOST=1

                ;LOCAL VARIBLES

000002          ARG1=   2
000004          ARG2=   4
000006          ARG3=   6
000010          ARG4=   10

100000          BIT15=  100000
040000          BIT14=  40000
000400          BIT08=  400
000200          BIT07=  200
000100          BIT06=  100
000040          BIT05=  40
000020          BIT04=  20
000010          BIT03=  10
000004          BIT02=  4
```

```
FDM.MACRO       MACRO V03.01 4-JUN-79 17:30:32 PAGE 1-1
NODE22

     66          000002                          BIT01=  2
     67          000001                          BIT00=  1
     68
```


FORTRAN

0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0094
0095
0096
0097
0098
0099
0101
0102
0103

```
                                ;*****************************************************
                                ;**** PROCEDURE MASTER START UP                      *
                                ;*****************************************************

000000  012700  000340          MASTER: MOV     #340,   R0                      ;PRI=7
000004  106400                          MTPS    R0
000006  012700  000124                  MOV     #124,   R0
000012  012720  000430'                 MOV     #LIO,   (R0)+                   ;LIU HANDLER
000016  012710  000340                  MOV     #340,   (R0)                    ;PRI=7
000022  000005                          RESET                                   ;RESET BUS
000024  012700  000060                  MOV     #60,    R0
000030  012720  002470'                 MOV     #SIO,   (R0)+                   ;SIG INTERFACE HANDLER
000034  012720  000340                  MOV     #340,   (R0)+
000040  012700  000100                  MOV     #100,   R0                      ;CLOCK VECTOR
000044  012710  001654'                 MOV     #TIME,  (R0)                    ;CLOCK
000050  042737  040100  172414         BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000056  004767  000014                  JSR     PC,     LIUINT                  ;CLEAR LIU
000062  005037  172414                  CLR     @#CSR                           ;CLEAR BLIUI CSR
000066  012700  000000                  MOV     #000,   R0                      ;PRI=0
000072  106400                          MTPS    R0                              ;CPU DOWN
000074  000207                          RTS     PC


                                ;*****************************************************
                                ;**** PROCEDURE INITIALIZE                          *
                                ;*****************************************************

                                        ;CLEAR ACRAM

000076  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000104  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000112  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000116  100375                          BPL     .-4                             ;NO RETRY
000120  105037  172414                  CLRB    @#CSR
000124  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
000132  012700  177400                  MOV     #-256., R0                      ;COUNTER
000136  012737  004407  172416  1$:     MOV     #2311., @#OPREG                 ;WRITE A NULL
000144  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000150  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
000152  105037  172414                  CLRB    @#CSR
000156  005200                          INC     R0                              ;COUNT+1
000160  001366                          BNE     1$

                                        ;CLEAR INPUT/OUTPUT BUFFERS

000162  012701  010410                  MOV     #10410, R1                      ;RDBUFADR CMD
000166  012700  010440                  MOV     #10440, R0                      ;SEL INBUF0
000172  004767  000106                  JSR     PC,     ZEROBP                  ;SET POINTER=0
000176  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000204  105737  172414                  TSTB    @#CSR                           ;GOOD READ
000210  100375                          BPL     .-4                             ;NO RETRY
000212  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
000216  012701  010610                  MOV     #10610, R1                      ;RDBUFADR COMMAND
000222  012700  010640                  MOV     #10640, R0                      ;SEL INBUF1 COMMAND
000226  004767  000052                  JSR     PC,     ZEROBP                  ;POINTER=0
```

```
  102 000232  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
  103 000240  105737  172414                  TSTB    @#CSR                   ;GOOD RD
  104 000244  100375                          BPL     .-4                     ;NO RETRY
  105 000246  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
  106 000252  012701  010510                  MOV     #10510, R1              ;RDBUFADR COMMAND
  107 000256  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
  108 000262  004767  000016                  JSR     PC,     ZEROBP
  109 000266  012701  010710                  MOV     #10710, R1              ;RDBUFADR COMMAND
  110 000272  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
  111 000276  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
  112 000302  000440                          BR      STAT$                   ;GO CLEAR STATUS
  113 000304  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
  114 000310  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
  115 000316  105737  172414                  TSTB    @#CSR                   ;GOOD RD
  116 000322  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
  117 000324  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
  118 000330  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
  119 000334  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
  120 000340  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
  121 000344  022702  000000          2$:     CMP     #0,     R2              ;POINTER=0?
  122 000350  001412                          BEQ     3$                      ;YES RECHECK
  123 000352  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
  124 000360  105737  172414                  TSTB    @#CSR                   ;GOOD RD
  125 000364  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
  126 000366  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
  127 000372  005302                          DEC     R2                      ;LOCAL POINTER-1
  128 000374  000763                          BR      2$                      ;BR UNTIL 0
  129 000376  010037  172416          3$:     MOV     R0,     @#OPREG         ;SEL BUFFER
  130 000402  000207                          RTS     PC                      ;RETURN
  131
  132                                         ;READ AND CLEAR STATUS
  133
  134 000404  012737  010400  172416  STAT$:  MOV     #4352., @#OPREG         ;WCR:RS
  135 000412  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
  136 000420  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
  137 000426  000207                          RTS     PC
  138
  139
```

```
                                ;*************************************************************
                                ;**** PROCEDURE LIU-HANDLER                                 *
                                ;*************************************************************

000430  010046                LIU:    MOV     R0,     -(SP)                   ;SAVE R0
000432  010146                        MOV     R1,     -(SP)                   ;SAVE R1
000434  010246                        MOV     R2,     -(SP)                   ;SAVE R2
000436  010346                        MOV     R3,     -(SP)                   ;SAVE R3
000440  010446                        MOV     R4,     -(SP)                   ;SAVE R4
000442  010546                        MOV     R5,     -(SP)                   ;SAVE R5
000444  005737  172414                TST     @#CSR                           ;? LIU INTERRUPTED
000450  100153                        BPL     RTI$                            ;NO
000452  042737  040000  172414        BIC     #BIT14, @#CSR                   ;YES/DISABLE LIU
000460  012737  010400  172416        MOV     #4352., @#OPREG                 ;WCR : RS(0)
000466  012737  002400  172416        MOV     #1280., @#OPREG                 ;RD
000474  013767  172416  000502        MOV     @#IOBUF, CAUSE
000502  042767  177400  000474        BIC     #177400, CAUSE                  ;CLEAR UNUSED BITS

                                ;PREFORM OPERATION DEPENDING ON BITS

000510  132767  000004  000466  B2$:  BITB    #BIT02, CAUSE                   ;OV-FL
000516  001406                        BEQ     B3$
000520  012702  010410                MOV     #4360., R2                      ;RDBUFADR IN0
000524  012703  010440                MOV     #4384., R3                      ;SEL INBUF0
000530  004767  000270                JSR     PC,     EMBF                    ;GO EMPTY
000534  132767  000010  000442  B3$:  BITB    #BIT03, CAUSE                   ;OV-FL
000542  001406                        BEQ     B4$
000544  012702  010610                MOV     #4488., R2                      ;RDBUFADR IN1
000550  012703  010640                MOV     #4512., R3                      ;SEL INBUF1
000554  004767  000244                JSR     PC,     EMBF                    ;GO EMPTY
000560  132767  000020  000416  B4$:  BITB    #BIT04, CAUSE                   ;LINE-LOSS PRIMARY
000566  001407                        BEQ     B5$
000570  132767  000200  000406  B7$:  BITB    #BIT07, CAUSE                   ;LATCH ON
000576  001403                        BEQ     B5$                             ;YES LOOP AROUND
000600  012767  000002  002002'       MOV     #2,     LLFLG                   ;FLAG PRIMARY SW. FAIL
000606  132767  000040  000370  B5$:  BITB    #BIT05, CAUSE                   ;BACKUP LINE LOSS
000614  001403                        BEQ     B6$
000616  012767  000001  002002'       MOV     #1,     LLFLG                   ;FLAG BACKUP SW. FAIL
000624  132767  000100  000352  B6$:  BITB    #BIT06, CAUSE                   ;WRITE TOKEN DETECT
000632  001404                        BEQ     B0$
000634  005067  001000'               CLR     IWRTTM                          ;CLEAR TOKEN VAR
000640  005067  001002'               CLR     IWRT

                                ;BUFFER FULL

000644  132767  000001  000332  B0$:  BITB    #BIT00, CAUSE                   ;INBUF0 FULL
000652  001423                        BEQ     B1$                             ;NO
000654  012701  000400'               MOV     #INBF,  R1                      ;BUFFER ADDRESS
000660  012702  010410                MOV     #4360., R2                      ;RDBUFADR COMM
000664  012703  010440                MOV     #4384., R3                      ;SEL BUF
000670  012704  000001                MOV     #BIT00, R4                      ;CRC BIT
000674  004767  000124                JSR     PC,     EMBF                    ;EMPTY BUFFER
000700  012705  001206'               MOV     #AREA,  R5                      ;DATA LINK AREA
000704  012767  000001  000274        MOV     #1,     AREA                    ;1 VARIBLE
000712  010267  000274                MOV     R2,     DATA                    ;CRC OR BYTE COUNT
000716  004767  000000G               JSR     PC,     LPINPT                  ;CALL FORTRAN QUE'ER
```

```
 58 000722  132767  000002  000254  B1$:    BITB    #BIT01, CAUSE           ;INBUF1 FULL
 59 000730  001423                   BEQ     RTI$                    ;NO WERE DONE
 60 000732  012701  000400'          MOV     #INBF,  R1              ;BUFFER ADDRESS
 61 000736  012702  010610           MOV     #4488., R2              ;RDBUFADR COMM
 62 000742  012703  010640           MOV     #4512., R3              ;SEL INBUF1
 63 000746  012704  000002           MOV     #BIT01, R4              ;CRC BIT
 64 000752  004767  000046           JSR     PC,     EMBF            ;GO EMPTY BUFFER
 65 000756  012705  001206'          MOV     #AREA,  R5              ;DATA LINK AREA
 66 000762  012767  000001  000216   MOV     #1,     AREA            ;ONE VARIBLE
 67 000770  010267  000216           MOV     R2,     DATA            ;CRC OR BYTE COUNT
 68 000774  004767  000006G          JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
 69
 70 001000  012605                   RTI$:   MOV     (SP)+,  R5              ;RESTORE REGISTERS
 71 001002  012604                   MOV     (SP)+,  R4
 72 001004  012603                   MOV     (SP)+,  R3
 73 001006  012602                   MOV     (SP)+,  R2
 74 001010  012601                   MOV     (SP)+,  R1
 75 001012  012600                   MOV     (SP)+,  R0
 76 001014  052737  040000  172414   BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
 77 001022  000002                   RTI                             ;RETURN FROM INTERRUPT
 78
 79 001024  012737  010600  172416   EMBF:   MOV     #4480., @#OPREG         ;READ STATUS 1
 80 001032  012737  002400  172416           MOV     #1280., @#OPREG         ;RS
 81 001040  013700  172416           MOV     @#IOBUF, R0             ;FETCH STATUS
 82 001044  130400                   BITB    R4,R0                   ;GOOD CRC
 83 001046  001002                   BNE     CRCOK
 84 001050  012704  177777           MOV     #-1,    R4              ;NO FLAG
 85 001054  010237  172416           CRCOK:  MOV     R2,     @#OPREG         ;RDBUFADR
 86 001060  012737  001400  172416   MOV     #768.,  @#OPREG         ;RD
 87 001066  105737  172414           TSTB    @#CSR                   ;GOOD RD
 88 001072  100375                   BPL     .-4
 89 001074  013702  172416           MOV     @#IOBUF, R2             ;REPLACE WITH POINTER
 90 001100  042702  177400           BIC     #177400, R2             ;CLEAR MST BITS
 91 001104  010200                   MOV     R2,     R0              ;SAVE IT
 92 001106  005400                   NEG     R0                      ;2'S COMP
 93 001110  010037  172412           MOV     R0,     @#WCR           ;BYTE COUNT NOW
 94 001114  010137  172410           MOV     R1,     @#BAR           ;ADDRESS IN MEMORY
 95 001120  010337  172416           MOV     R3,     @#OPREG         ;SEL BUFFER
 96 001124  012737  001400  172416   MOV     #768.,  @#OPREG         ;FALSE RD
 97 001132  105737  172414           TSTB    @#CSR                   ;DONE ON
 98 001136  100375                   BPL     .-4                     ;NO LOOP
 99 001140  012737  021000  172416   MOV     #8704., @#OPREG         ;FIRE DMA
100 001146  000240                   NOP                             ;DELAY
101 001150  105737  172414           TSTB    @#CSR                   ;GOOR DMA
102 001154  100401                   BMI     DMAOK
103 001156  000240                   NOP                             ;ERROR IF HERE
104 001160  012737  004400  172416   DMAOK:  MOV     #2304., @#OPREG         ;FALSE WD
105 001166  105737  172414           TSTB    @#CSR                   ;GOOD WD
106 001172  100375                   BPL     .-4
107 001174  005704                   TST     R4                      ;WAS CRC OK
108 001176  100001                   BPL     END$                    ;YES
109 001200  010402                   MOV     R4,     R2              ;NO FLAG IT
110 001202  000207                   END$:   RTS     PC                      ;RETURN
111 001204  000000                   CAUSE:  .WORD   0                       ;STATUS BYTE 0 HOLDER
112 001206  000000                   AREA:   .WORD   0
113 001210  001212'                  .WORD   DATA
114 001212  000000                   DATA:   .WORD   0
```

```
                                        .ENABLE LSB
                                    ;*************************************************************
                                    ;**** PROCEDURE ENABLE                                      *
                                    ;*************************************************************

001214  017501  000002          ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
001220  022701  000001                  CMP     #1,     R1
001224  001404                          BEQ     E1$
001226  042737  040000  172414  E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
001234  000207                          RTS     PC
001236  052737  040000  172414  E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
001244  000207                          RTS     PC

                                        ;FORTRAN ENCODE/DECODE MACRO

001246  017575  000004  000002  ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
001254  000207                          RTS     PC
001256  017575  000002  000004  DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECODE BYTE COUNT
001264  000207                          RTS     PC


                                    ;*************************************************************
                                    ;**** PROCEDURE LINE SWITCH                                 *
                                    ;*************************************************************

                                        ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                        ;
                                        ;04=SET PRIMARY LINE
                                        ;32=RESET PRIMARY LINR
                                        ;08=SET BACKUP LINE
                                        ;64=RESET BACKUP LINE

001266  017501  000002          SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
001272  012737  010420  172416          MOV     #4368., @#OPREG                 ;WCR : MODSTAT
001300  062701  004400                  ADD     #2304., R1
001304  010137  172416                  MOV     R1,     @#OPREG                 ;RD
001310  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001314  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001316  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001322  000207                          RTS     PC


```

```
                        ;*************************************************************
                        ;**** PROCEDURE STATUS                                      *
                        ;*************************************************************

                                ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
                                ;
                                ;X=0 STATUS BYTE 0
                                ;X=1 STATUS BYTE 1

001324  017501  000002          STATUS: MOV     @ARG1(R5), R1                   ;WHICH BYTE
001330  022701  000001                  CMP     #1,     R1                      ;COMPARE
001334  001415                          BEQ     2$
001336  012737  010400  172416  1$:     MOV     #4352., @#OPREG                 ;WCR : RS(0)
001344  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001352  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001356  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001362  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001366  000207                          RTS     PC
001370  012737  010600  172416  2$:     MOV     #4480., @#OPREG                 ;WCR : RS(1)
001376  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001404  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001410  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001414  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001420  000207                          RTS     PC
001422  012737  010400  172416  STATB0: MOV     #4352., @#OPREG                 ;STATUS BYTE 1
001430  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001436  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001442  042701  177400                  BIC     #177400, R1
001446  132701  000010                  BITB    #BIT03, R1                      ;BUFFER FULL
001452  001404                          BEQ     3$
001454  012775  000001  000002          MOV     #1,     @ARG1(R5)               ;LOAD RETURN
001462  000207                          RTS     PC
001464  012775  000000  000002  3$:     MOV     #0,     @ARG1(R5)               ;LOAD RETURN
001472  000207                          RTS     PC
```

```
                                ;*************************************************************
                                ;**** PROCEDURE WRITE TOKEN (WTOKEN)                        *
                                ;*************************************************************

                                        ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP

001474  012737  010600  172416  WTOKEN: MOV     #4480.,@#OPREG                  ;READ STATUS(1)
001502  012737  002400  172416          MOV     #1280.,@#OPREG                  ;RS
001510  013701  172416                  MOV     @#IOBUF,R1                      ;FETCH DATA
001514  132701  000010                  BITB    #BIT03,R1                       ;BUFFER LOADED
001520  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001522  012737  010420  172416          MOV     #4368.,@#OPREG                  ;MODSTAT
001530  012737  004423  172416          MOV     #2323.,@#OPREG                  ;MARK OB0F/OB1F/WTCMD
001536  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001542  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001544  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001550  000207                          RTS     PC
001552  012737  010740  172416  WTKG:   MOV     #4576.,@#OPREG                  ;SELECT BUFFER 1
001560  012737  001652' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001566  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001572  005402                          NEG     R2                              ;FORMAT IT
001574  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001600  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001606  000240                          NOP                                     ;DELAY
001610  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001614  000240                          NOP                                     ;ERROR IF HERE
001616  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001622  012737  010420  172416          MOV     #4368.,@#OPREG                  ;MODSTAT
001630  012737  004422  172416          MOV     #2322.,@#OPREG                  ;MARK OB1F/WTCMD
001636  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001642  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001644  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001650  000207                          RTS     PC                              ;RETURN
001652     000     377          WTD:    .BYTE 0,377                             ;A WTOKEN


                                ;*************************************************************
                                ;**** PROCEDURE TIMER (TIME)                                *
                                ;*************************************************************

001654  005267  001006'        TIME:   INC     LTIME                           ;TIME+1
001660  022767  077777  001006'        CMP     #077777,LTIME                   ;?TIME FULL
001666  001002                          BNE     4$
001670  005067  001006'                CLR     LTIME                           ;CLEAR CLOCK
001674  000002                  4$:     RTI
```

```
                        ;*************************************************************
                        ;**** PROCEDURE ACRAM                                       *
                        ;*************************************************************

                                ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
                                ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
                                ;
                                ;0110   =NREAD
                                ;0100   =DREAD
                                ;0111   =NULL
                                ;0000   =WTOKEN


001676  017500  000002         RAM:    MOV     @ARG1(R5), R0                   ; RAM
001702  022700  000000                 CMP     #0,     R0                      ;WHICH OPERATION
001706  001035                        BNE     RDRAM                           ;READ OP
001710  017500  000004         WTRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS
001714  017501  000006                 MOV     @ARG3(R5), R1                   ;WRITE DATA
001720  012737  010402  172416         MOV     #4354.,  @#OPREG                ;SEL LDACTR
001726  062700  004400                 ADD     #2304.,  R0                     ;WD/DATA (ADDR)
001732  010037  172416                 MOV     R0,     @#OPREG                 ;WRITE DATA
001736  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
001742  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
001744  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
001750  012737  010401  172416         MOV     #4353.,  @#OPREG                ;SEL ACRAM
001756  062701  004400                 ADD     #2304.,  R1                     ;WD/DATA (CMD)
001762  010137  172416                 MOV     R1,     @#OPREG                 ;WRITE
001766  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
001772  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
001774  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002000  000207                        RTS     PC
002002  017500  000004         RDRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS TO READ
002006  012737  010402  172416         MOV     #4354.,  @#OPREG                ;SEL LDACTR
002014  062700  004400                 ADD     #2304.,  R0                     ;WD/DATA (ADDR)
002020  010037  172416                 MOV     R0,     @#OPREG                 ;WRITE DATA
002024  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
002030  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
002032  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002036  012737  010401  172416         MOV     #4353.,  @#OPREG                ;SEL ACRAM
002044  012737  001400  172416         MOV     #768.,   @#OPREG                ;READ DATA
002052  105737  172414                 TSTB    @#CSR                           ;VALID READ
002056  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
002060  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002064  013700  172416                 MOV     @#IOBUF, R0                     ;FETCH DATA
002070  042700  177760                 BIC     #177760, R0                     ;CLEAR BITS
002074  010075  000006                 MOV     R0, @ARG3(R5)                   ;RTN DATA
002100  000207                        RTS     PC
```

```
                                .ENABLE LSB
                        ;*************************************************************
                        ;**** PROCEDURE WRITE LOOP (LPOUT)                          *
                        ;*************************************************************

002102  017502  000002          LPOUT:  MOV     @ARG1(R5), R2                   ;BYTE COUNT
002106  012700  000340                  MOV     #340,   R0                      ;PRI=7
002112  106400                          MTPS    R0
002114  012737  000000' 172410          MOV     #OUTBF, @#BAR                   ;BUS ADDRESS
002122  005402                          NEG     R2                              ; 2'S COMP COUNT
002124  010237  172412                  MOV     R2,     @#WCR                   ;COUNT
002130  012737  010540  172416          MOV     #4448., @#OPREG                 ;OBO COMMAND
002136  012737  024000  172416          MOV     #10240., @#OPREG                ;DMA GO
002144  000240                          NOP                                     ;INTERFACE TIME
002146  105737  172414                  TSTB    @#CSR                           ;DMA OK
002152  000240                          NOP                                     ;ERROR IF HERE
002154  105037  172414                  CLRB    @#CSR
002160  012737  010740  172416          MOV     #4576., @#OPREG                 ;OB1 COMMAND
002166  012737  001652' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
002174  012702  000002                  MOV     #2,     R2                      ;BYTE COUNT
002200  005402                          NEG     R2                              ;FORMAT FOR BLIUI
002202  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
002206  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
002214  000240                          NOP                                     ;DELAY
002216  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
002222  000240                          NOP                                     ;ERROR IF NOT
002224  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
002230  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
002236  012737  004403  172416          MOV     #2307., @#OPREG                 ;BUFFERS FULL
002244  105737  172414                  TSTB    @#CSR                           ;GOOD WRITE
002250  100375                          BPL     .-4                             ;NO LOOP UNTIL
002252  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002256  012700  000000                  MOV     #000,   R0                      ;PRI=0
002262  106400                          MTPS    R0                              ;LOWER CPU
002264  000207                          RTS     PC
```

```
                        ;*********************************************************
                        ;**** PROCEDURE READ BUFFER POINTER                     *
                        ;*********************************************************

002266  017500  000002  RDPNT:  MOV     @ARG1(R5), R0          ;FETCH COMMAND
002272  022700  000000          CMP     #0,     R0             ;INBUF0 ?
002276  001004                  BNE     1$                     ;NO
002300  012700  010410          MOV     #4360., R0             ;RDBUFADR IN0
002304  000167  000046          JMP     4$
002310  022700  000001  1$:     CMP     #1,     R0             ;INBUF1 ?
002314  001004                  BNE     2$                     ;NO
002316  012700  010610          MOV     #4488., R0             ;RDBUFADR IN1
002322  000167  000030          JMP     4$
002326  022700  000002  2$:     CMP     #2,     R0             ;OUTBUF0 ?
002332  001004                  BNE     3$                     ;NO
002334  012700  010510          MOV     #4424., R0             ;RDBUFADR OUT0
002340  000167  000012          JMP     4$
002344  022700  000004  3$:     CMP     #4,     R0             ;OUTBUF1 ?
002350  001020                  BNE     5$                     ;NO RETURN
002352  012700  010710          MOV     #4552., R0             ;RDBUFADR OUT1
002356  010037  172416  4$:     MOV     R0,     @#OPREG        ;RDBUFADR
002362  012737  001400  172416  MOV     #768.,  @#OPREG        ;RD
002370  105737  172414          TSTB    @#CSR                  ;READY
002374  100375                  BPL     .-4                    ;NO LOOP UNTIL
002376  013700  172416          MOV     @#IOBUF, R0            ;FETCH POINTER
002402  042700  177400          BIC     #177400, R0            ;CLEAR MST
002406  010075  000004          MOV     R0,     @ARG2(R5)      ;RETURN POINTER
002412  000207          5$:     RTS     PC
```

```
                        ;*************************************************************
                        ;**** PROCEDURE RESTART                                     *
                        ;*************************************************************

                                ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
                                ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
                                ;CALL RSTART(2) -LOAD MODE(173000)

002414  017500  000002  RSTART: MOV     @ARG1(R5), R0                   ;MODE
002420  022700  000000  RS0:    CMP     #0,     R0                      ;LOAD ADDRESS
002424  001002                  BNE     RS1
002426  000000                  HALT
002430  000207                  RTS     PC
002432  022700  000001  RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
002436  001005                  BNE     RS2
002440  004767  175432          JSR     PC,     LIUINT                  ;INT LIU FIRST
002444  012700  000040  G4$:    MOV     #40,    R0                      ;START ADDRESS
002450  000110                  JMP     (R0)
002452  022700  000002  RS2:    CMP     #2,     R0                      ;LOAD MODE
002456  001002                  BNE     RST
002460  000177  000002          JMP     @ROM
002464  000207          RST:    RTS     PC
002466  173000          ROM:    .WORD   173000
```

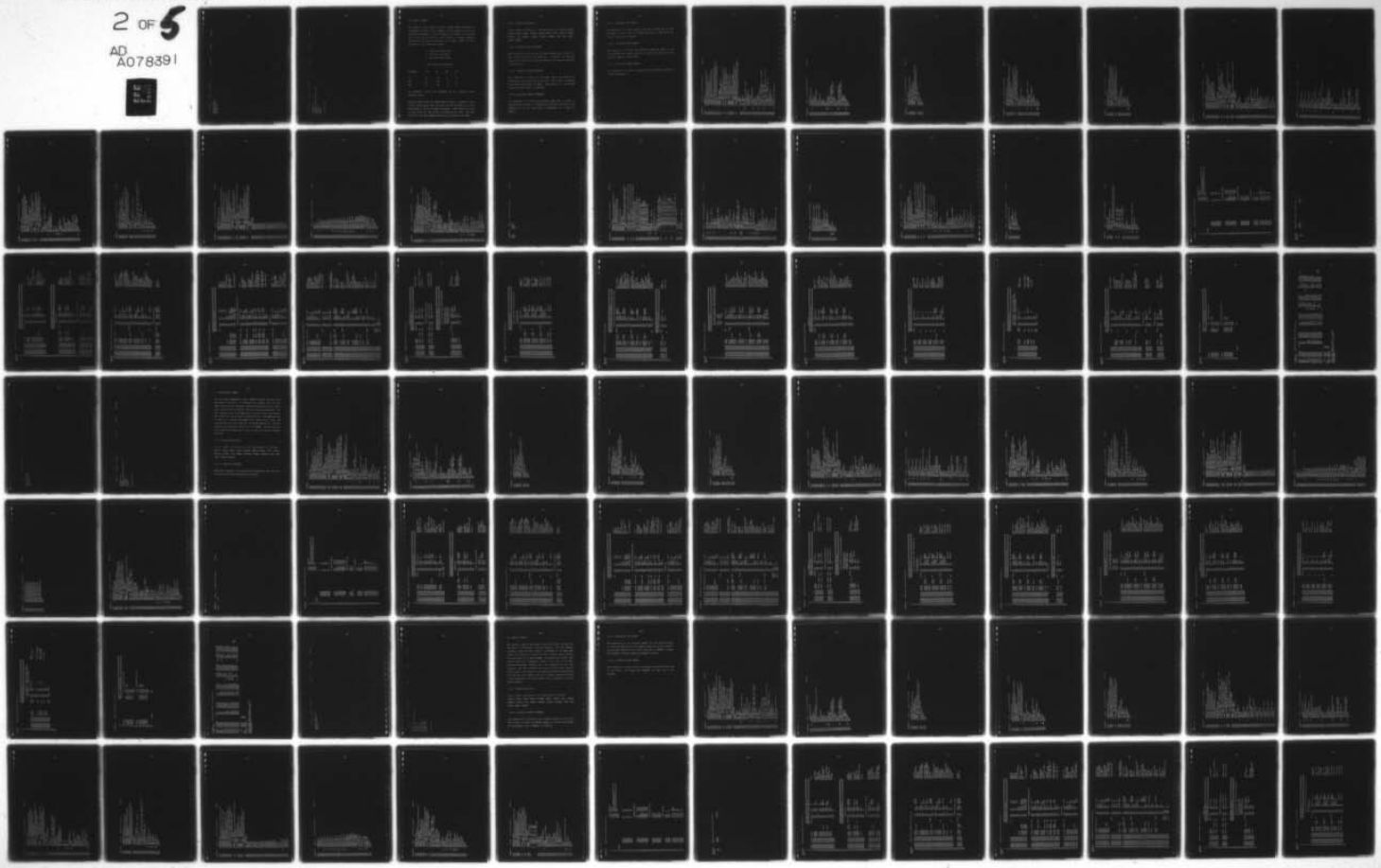
```
                                .ENABLE LSB
                        ;*************************************************************
                        ;****  PROCEDURE SIG INTERFACE                                *
                        ;*************************************************************

002470  010046          SIO::   MOV     R0,     -(SP)           ;SAVE REGISTERS
002472  010146                  MOV     R1,     -(SP)
002474  010246                  MOV     R2,     -(SP)
002476  012700  000400'          MOV     #XINBF, R0              ;BUFFER ADDRESS
002502  012701  000050          MOV     #40.,   R1              ;BYTE COUNT
002506  105737  177560  1$:     TSTB    @#HRCSR                 ;PORT READY
002512  100375                  BPL     .-4                     ;LOOP UNTIL
002514  113702  177562          MOVB    @#HRBUF, R2             ;FETCH BYTE
002520  042702  177400          BIC     #177400, R2             ;CLEAR MST
002524  010267  002000'          MOV     R2,     IOFLG           ;ITS THE BYTE COUNT
002530  110220                  MOVB    R2,     (R0)+
002532  005301                  DEC     R1                      ;COUNT-1
002534  105737  177560  2$:     TSTB    @#HRCSR                 ;PORT READY
002540  100375                  BPL     .-4                     ;LOOP UNTIL
002542  113702  177562          MOVB    @#HRBUF, R2             ;FETCH BYTE
002546  042702  177400          BIC     #177400, R2             ;CLEAR MST
002552  110220                  MOVB    R2,     (R0)+           ;STORE IT
002554  005301                  DEC     R1
002556  001366                  BNE     2$                      ;READ 39 BYTES
002560  042737  000100  177560  BIC     #100,   @#HRCSR         ;NO MORE UNTIL FORTRAN
002566  012602  3$:     MOV     (SP)+,  R2                      ;RETORE REGISTERS
002570  012601                  MOV     (SP)+,  R1
002572  012600                  MOV     (SP)+,  R0
002574  000002                  RTI

                                ;SEND OVER TO SIG

002576  017500  000002  SOUT::  MOV     @ARG1(R5),R0            ;BYTE COUNT
002602  105737  177564          TSTB    @#HXCSR                 ;READY TO SNED
002606  100375                  BPL     .-4                     ;LOOP UNTIL
002610  110037  177566          MOVB    R0,     @#HXBUF         ;SEND IT
002614  000207                  RTS     PC

                                ;REPORT NODE READY

002616  105737  177564  READY:: TSTB    @#HXCSR                 ;PORT READY
002622  100375                  BPL     .-4                     ;LOOP UNTIL
002624  112737  000122  177566  MOVB    #122, @#HXBUF           ;SEND R
002632  052737  000100  177560  BIS     #100, @#HRCSR           ;ENABLE INTERRUPTS
002640  000207                  RTS     PC                      ;RETURN
```

```
                        ;*************************************************************
                        ;****        COMMON DATA AREA                                 *
                        ;*************************************************************

000000                          .PSECT  DFM,RW,D,GBL,REL,OVR

000000                  OUTBF:  .BLKB   256.
000400                  INBF:   .BLKB   256.
001000                  IWRTTM: .BLKW
001002                  IWRT:   .BLKW
001004                  STAT:   .BLKW
001006                  LTIME:  .BLKW
                                .EVEN

000000                          .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                  XOUTBF: .BLKB   256.
000400                  XINBF:  .BLKB   256.
001000                  ACKSEQ: .BLKB   256.
001400                  LIDFD:  .BLKB   256.
002000                  IOFLG:  .BLKW
002002                  LLFLG:  .BLKW
002004                  IRSEND: .BLKW
                                .EVEN


000001                          .END
```

```
FDM.MACRO       MACRO V03.01 4-JUN-79 17:30:32 PAGE 12-1
SYMBOL TABLE

ACKSEQ  001000R     003 B1$     000722R         E1$     001236R         MASTER  000000RG        SOUT    002576RG
AREA    001206R         B2$     000510R         G4$     002444R         NDXX  = 000000          STAT    001004R     002
ARG1  = 000002          B3$     000534R         HRBUF = 177562          ND24  = 000001          STATBO  001422RG
ARG2  = 000004          B4$     000560R         HRCSR = 177560          NHOST = 000000          STATUS  001324RG
ARG3  = 000006          B5$     000606R         HXBUF = 177566          OPREG = 172416          STAT$   000404R
ARG4  = 000010          B6$     000624R         HXCSR = 177564          OUTBF   000000R     002 SWITCH  001266RG
BAR   = 172410          B7$     000570R         INBF    000400R     002 RAM     001676RG        TIME    001654RG
BIT00 = 000001          CAUSE   001204R         IOBUF = 172416          RDPNT   002266RG        WCR   = 172412
BIT01 = 000002          CRCOK   001054R         IOFLG   002000R     003 RDRAM   002002R         WTD     001652R
BIT02 = 000004          CSR   = 172414          IRSEND  002004R     003 READY   002616RG        WTKG    001552R
BIT03 = 000010          DATA    001212R         IWRT    001002R     002 ROM     002466R         WTOKEN  001474RG
BIT04 = 000020          DESTR   001256RG        IWRTTM  001000R     002 RST     002464R         WTRAM   001710R
BIT05 = 000040          DHOST = 000001          LIDFD   001400R     003 RSTART  002414RG        XCIO  = 000001
BIT06 = 000100          DMAOK   001160R         LIO     000430RG        RS0     002420R         XGIO  = 000001
BIT07 = 000200          EMBF    001024R         LIUINT  000076RG        RS1     002432R         XINBF   000400R     003
BIT08 = 000400          ENABLE  001214RG        LLFLG   002002R     003 RS2     002452R         XOUTBF  000000R     003
BIT14 = 040000          END$    001202R         LPINPT= ****** G         RTI$    001000R         XSIO  = 000000
BIT15 = 100000          ENSTR   001246RG        LPOUT   002102RG        SIO     002470RG        ZEROBP  000304R
BO$     000644R         EO$     001226R         LTIME   001006R     002

. ABS.  000000      000
        002642      001
DFM     001010      002
BUFS    002006      003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:FDM=DK:FDM
```

AD-A078 391 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66157 SBIE-AD-E100 313 NL

2 OF 5
AD A078391



```
FOR NODAL/NOLINE
FOR NODAL1/NOLINE
FOR VSOC/NOLINE
MAC FIM
```

```
ASSIGN DX0: DK:
R LINK
DX1:NODE22=DX1:MAP DX1:NODAL/C/I/W
DX1:E80/F
DX1:NODAL1/C
DX1:VSRC2/
$SIMRT


ASSIGN DX1: DK:
```

1.4 NODE 23 (DSQC)

The Digital Service Quality Control (DSQC) module assesses the performance of digital data channels for the purpose of detecting degrading performance of these channels with respect to increasing error rates, and to assist in fault isolation. Channels numbered 501-1000 are arbitrarily monitored by the DSQC. There are three parameters to be checked per channel:

1. Pseudo Error Rate (PE)
2. Bit Error Rate (BIE)
3. Block Error Rate (BLE)

DSQC Parameters Thresholds

Parameter	RH	RL	AH	AL
PE	10	-10	5	-5
BIE	10	-10	5	-5
BLE	10	-10	5	-5

The parameter values are simulated by the Simulated Input Generator (SIG).

Trending occurs when the DWQC module detects a parameter value within a delta value equal to one of the Red threshold or a delta value equal to two of the Amber threshold. Event reports are sent to FIAC using the same format as described for VSQC. The DSQC interprets the same commands generated by the DBMS as the VSQC.

1.4.1 Program Descriptions

1.4.1.1 Refer to Section 1.1 for descriptions of routines NODAL, IGETSP, ENQUE, DEQUE, ACKNAK, INPTQ, LPINPT, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME RAM, LPOUT, RDPNT, RSTART.

1.4.1.2 Subroutine DSQC (FORTRAN)

This subroutine is called from the nodal program when a packet has been received from the SIG interface. It decodes the simulated input values read from the SIG and performs the function described in Section 1.4.

1.4.1.3 Subroutine TREND (FORTRAN)

This subroutine is called by the DSQC routine to perform the trending on the values sent by the SIG. This value is compared with DELTA values; when a channel consistently is in the DELTA region, an event report is generated.

1.4.1.4 Subroutine DSQCLP (FORTRAN)

The subroutine is called from program NODAL when a packet is received from the loop. It performs one of three functions: turns reporting on or off, and takes a measurement of a specified channel.

1.4.1.5 Subroutine SIO (MACRO)

The subroutine is called from an interrupt received by the SIG interface; it then reads the message and sets a flag containing the byte count of the message.

1.4.1.6 Subroutine SOUT (MACRO)

This subroutine is called from subroutine DSQC and passes to the SIG processor the channel number of a channel for which the user language requests a measurement.

1.4.1.7 Subroutine READY (MACRO)

This subroutine is called to signal the SIG that DSQC is ready for another measurement.

```
0001          PROGRAM NODAL
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003          INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP
0005          INTEGER*2 RSTBKP,Q1,Q2,RESNLM,OUTQ,DEQUE
0006          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007          LOGICAL*1 IDATA
0008          LOGICAL*1 PACK,ICODE,ETX,CR,LF,MONTOR,ISLID,DUM
0009          INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0010          REAL*4 RH,RL,AH,AL,RHT,RLT,AHT,ALT,DELTA,IRSV,VMEAS
0011          REAL*8 VTR
0012          COMMON /MESS/ MESSEQ
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0014          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0015          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IOFLG,LLFLG,IRSEND
0016          COMMON /FRE/ FREE(64),IFR,IFRSZ
0017          COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0018          COMMON       PACK(256,64)
0019          COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0020          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0021          COMMON /DCOM/ ISWTCH,VMEAS(3),ICTR,RL(3),RH(3),AL(3),AH(3),
             1     ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),IRSV(3),ICHNSW,
             2     ICHAN
0022          CALL MASTER
0023          CALL INIT
0024          CALL ENABLE(1)
0025        5 CALL READY
0026       25 IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0028          CALL ENABLE(0)
0029          Q2=DEQUE(PINQ,INQ,1)
0030          CALL ENABLE(1)
0031          DO 30 I=1,10
0032            IDATA(I)=PACK(I+6,Q2)
0033       30 CONTINUE
0034          CALL DSQCLP
0035          IFR=IFR+1
0036          FREE(IFR)=Q2
0037       40 CALL STATBO(IS)
0038          IF(IS .EQ. 1) GOTO 100
0040          IF(IOFLG .GT. 0 .AND. IRSEND .EQ. 0) CALL DSQC
0042          IF(LLFLG .GT. 0) CALL LINLOS
0044       80 IF(OUTFCT .EQ. 1) GOTO 100
0046          IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0048          IF(IRSEND .EQ. 1) GOTO 85
0050          IF(PXINQ(1) .LT. PXINQ(2)) GOTO 100
0052          CALL ENABLE(0)
0053          Q1=DEQUE(PXINQ,XINQ,1)
0054          CALL ENABLE(1)
0055          OUTQ=Q1
0056       85 CALL DESTR(PACK(254,OUTQ),Q2)
```

```
0057          DO 90 I=1,Q2
0058            OUTBF(I)=PACK(I,OUTQ)
0059     90   CONTINUE
0060          OUTBF(Q2+1)=0
0061          IPT=OUTBF(5)
0062          OUTBF(Q2+2)=LIDFD(IPT)
0063          IFULL=1
0064          ISENT=1
0065          CALL LPOUT(Q2+2)
0066          IRSEND=1
0067          ACKTIM=0
0068          IWRTTM=0
0069          IWRT=1
0070    100   INFLCT=0
0071    110   OLDTIM=NEWTIM
0072          NEWTIM=LTIME
0073          IF(IWRT .EQ. 0) GOTO 120
0075          IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0076          IF(IWRTTM .LT. TIMLIM) GOTO 120
0078          CALL WTOKEN
0079          IWRTTM=0
0080          IWRT=0
0081    120   IF(ISENT .EQ. 0) GOTO 130
0083          ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0084          IF(ACKTIM .LT. ATIMLM) GOTO 130
0086          CALL ENABLE(0)
0087          CALL ACKNAK(0)
0088          CALL ENABLE(1)
0089    130   CONTINUE
0090          GOTO 5
0091          END
```

```
0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(64),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IF(IFR .LT. 1) CALL INIT
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```

```
0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008     10   IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016     20   IF(IQHEAD .GE. (IQLNTH)) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020     30   B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024     40   CALL INIT
0025    999   RETURN
0026          END
```

```
0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016       10 A(1)=IQHEAD-1
0017      999 RETURN
0018          END
```

```
0001          SUBROUTINE ACKNAK(N)
0002          INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ,PXOUTQ
0003          INTEGER*2 ACKQ,PACKQ,PINQ,FREE,OUTQ,XOUTQ
0004          INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006          LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008          INTEGER*2 LTIME,IWRTTM
0009          REAL*8 COMT(5)
0010          EQUIVALENCE(COMT,LCOMT)
0011          DATA COMT(1)/'MESSAGE '/,COMT(2)/'NOT SENT'/,COMT(3)/' FROM    '/,
             1     COMT(4)/'NODE     '/,COMT(5)/' TO NODE'/
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1       IWRTTM,IWRT,STAT,LTIME
0014          COMMON         PACK(256,64)
0015          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1           PACKQ(2),INQ(16),PINQ(2)
0016          COMMON /FRE/  FREE(64),IFR,IFRSZ
0017          COMMON /MESS/ MESSEQ
0018          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0019          IF (N .NE. 1) GOTO 10
0021          IFR=IFR+1
0022          DO 5 I=1,256
0023            PACK(I,OUTQ)=0
0024        5 CONTINUE
0025          IFULL=0
0026          ISENT=0
0027          IRSEND=0
0028          FREE(IFR)=OUTQ
0029          GOTO 999
0030       10 T2=PACK(256,OUTQ)
0031          T2=T2+1
0032          IF(T2 .LE. RESNLM) GOTO 120
0034          T3=PACK(3,OUTQ)
0035          T4=MOD(T3,64)
0036          IF(T4 .LT. 32) GOTO 110
0038          T5=IGETSP(N)
0039          IF(MESSEQ .EQ. 126) MESSEQ=0
0041          MESSEQ=MESSEQ+1
0042          PACK(1,T5)=0
0043          PACK(2,T5)=MESSEQ
0044          PACK(3,T5)=0
0045          PACK(4,T5)=0
0046          PACK(5,T5)=25
0047          PACK(6,T5)=ISLID
0048          DO 20 I=7,60
0049            PACK(I,T5)='040
0050       20 CONTINUE
0051          DO 30 I=7,9
0052            PACK(I,T5)=LF
0053       30 CONTINUE
```

```
0054          DO 40 I=1,8
0055            PACK(I+9,T5)=LCOMT(I)
0056     40   CONTINUE
0057          DO 50 I=9,16
0058            PACK(I+9,T5)=LCOMT(I)
0059     50   CONTINUE
0060          DO 60 I=17,22
0061            PACK(I+9,T5)=LCOMT(I)
0062     60   CONTINUE
0063          DO 70 I=25,29
0064            PACK(I+7,T5)=LCOMT(I)
0065     70   CONTINUE
0066          ENCODE(3,80,PACK(38,T5)) ISLID
0067     80   FORMAT(I3)
0068          DO 90 I=33,40
0069            PACK(I+11,T5)=LCOMT(I)
0070     90   CONTINUE
0071          ENCODE(3,80,PACK(52,T5))PACK(5,OUTQ)
0072          PACK(55,T5)=LF
0073          PACK(56,T5)=LF
0074          PACK(57,T5)=CR
0075          PACK(58,T5)=ETX
0076          CALL ENSTR(PACK(254,T5),58)
0077          CALL ENQUE(PXINQ,XINQ,T5)
0078          DO 102 I=1,256
0079            PACK(I,OUTQ)=0
0080    102   CONTINUE
0081          ISENT=0
0082          IRSEND=0
0083          IFULL=0
0084          IWRT=0
0085          IFR=IFR+1
0086          FREE(IFR)=OUTQ
0087          GOTO 999
0088     110  PACK(3,OUTQ)=PACK(3,OUTQ)+32
0089          PACK(256,OUTQ)=0
0090          ISENT=0
0091          IFULL=0
0092          IF(IRSEND .EQ. 1) GOTO 999
0094          CALL ENQUE(PXINQ,XINQ,OUTQ)
0095          GOTO 999
0096    120   PACK(256,OUTQ)=T2
0097          IFULL=0
0098          ISENT=0
0099          IF(IRSEND .EQ. 1) GOTO 999
0101          CALL ENQUE(PXINQ,XINQ,OUTQ)
0102    999   RETURN
0103          END
```

```
0001          SUBROUTINE INPTQ(L)
0002          INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
0003          INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005          LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
0006          INTEGER*2 LTIME,IWRTTM
0007          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IOFLG,LLFLG,IRSEND
0009          COMMON         PACK(256,64)
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          IF(L .LT. 1) GOTO 30
0014          LI=L-2
0015          II=INBF(6)
0016          IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0018          T1=IGETSP(N)
0019          DO 10 I=1,LI
0020       10 PACK(I,T1)=INBF(I)
0021          CALL ENSTR(PACK(254,T1),LI)
0022          CALL ENQUE(PINQ,INQ,T1)
0023          N=2
0024          GOTO 40
0025       20 N=1
0026          GOTO 40
0027       30 N=0
0028       40 DO 50 I=1,2
0029            OUTBF(I)=INBF(I)
0030       50 CONTINUE
0031          NN=1
0032          IF(N .EQ. 0) NN=128
0034          OUTBF(3)=NN
0035          OUTBF(4)=0
0036          OUTBF(5)=INBF(6)
0037          OUTBF(6)=ISLID
0038          OUTBF(7)=EXT
0039          OUTBF(8)=0
0040          OUTBF(9)=LIDFD(INBF(6))
0041          CALL LPOUT(9)
0042          IF(N .NE. 2) GOTO 999
0044          ACKSEQ(II)=INBF(2)
0045      999 RETURN
0046          END
```

```
FORTRAN IV     V02.1-1    Mon 04-Jun-79 17:49:19                PAGE 001

0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON/DFM/OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          CC1=INBF(3)
0009          CC2=INBF(4)
0010          IF(CC1 .LT. 0) GOTO 25
0012          IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0014          CC3=MOD(CC1,2)
0015          CC4=MOD(CC1,256)
0016          IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0018          IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0020          CC5=MOD(INBF(3),64)
0021          IF(CC5 .GE. 32) CALL INPTQ(LI)
0023          GOTO 99
0024      20  CC7=MOD(CC1,2)
0025          IF(CC7 .GE. 1) GOTO 30
0027      25  CALL ACKNAK(0)
0028          GOTO 99
0029      30  CALL ACKNAK(1)
0030          GOTO 99
0031      40  CALL INPTQ(LI)
0032      99  CONTINUE
0033          RETURN
0034          END
```

```
0001          SUBROUTINE INIT
0002          REAL*4 RH,RL,AH,AL,IRSV,VMEAS
0003          REAL*8 VTR
0004          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0005          INTEGER*2 STAT,FLWCNT,OUTFCT
0006          INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0007          LOGICAL*1 DUM,IDATA
0008          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0009          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID
0010          INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0011          COMMON /DFM/ OUTBF(256),INBF(256),
             1     IWRTTM,IWRT,STAT,LTIME
0012          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1              ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0013          COMMON /MESS/ MESSEQ
0014          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0015          COMMON /FRE/  FREE(64),IFR,IFRSZ
0016          COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0017          COMMON        PACK(256,64)
0018          COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0019          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0020          COMMON /DCOM/ ISWTCH,VMEAS(3),ICTR,RL(3),RH(3),AL(3),AH(3),
             1     ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),IRSV(3),ICHNSW,
             2     ICHAN
0021          CALL RAM(0,6,4)
0022          CALL RAM(0,255,0)
0023          MESSEQ=0
0024          IOFLG=0
0025          IRSEND=0
0026          LLFLG=0
0027          IFRSZ=64
0028          INFLCT=0
0029          ISLID=23
0030          ICHNSW=0
0031          CR='015
0032          LF='012
0033          ETX='003
0034          RESNLM=2
0035          IQLNTH=16
0036          IALTRT=0
0037          FLWCNT=0
0038          IWRTTM=0
0039          TIMLIM=50
0040          SETPRM=4
0041          RSTPRM=32
0042          SETBKP=8
0043          RSTBKP=64
0044          ATIMLM=200
0045          MONTOR=27
0046          ICHNSW=0
0047          IWRT=0
0048          ISWTCH=0
```

```
0049          OUTFCT=0
0050          IFULL=0
0051          ISENT=0
0052          DO 40 I=1,20
0053       40 LIDFD(I)=4
0054          LIDFD(21)=1
0055          LIDFD(22)=3
0056          LIDFD(23)=6
0057          LIDFD(24)=5
0058          LIDFD(25)=7
0059          LIDFD(26)=8
0060          LIDFD(27)=9
0061          LIDFD(28)=2
0062          DO 50 I=29,39
0063       50 LIDFD(I)=0
0064          DO 60 I=40,44
0065       60 LIDFD(I)=1
0066          DO 70 I=45,59
0067       70 LIDFD(I)=0
0068          DO 80 I=60,64
0069       80 LIDFD(I)=2
0070          DO 90 I=65,79
0071       90 LIDFD(I)=0
0072          DO 100 I=80,84
0073      100 LIDFD(I)=4
0074          DO 110 I=85,99
0075      110 LIDFD(I)=0
0076          DO 120 I=100,104
0077      120 LIDFD(I)=5
0078          DO 130 I=105,256
0079      130 LIDFD(I)=0
0080          PINQ(1)=IQLNTH
0081          PINQ(2)=IQLNTH+1
0082          PACKQ(1)=IQLNTH
0083          PACKQ(2)=IQLNTH+1
0084          PXOUTQ(1)=IQLNTH
0085          PXOUTQ(2)=IQLNTH+1
0086          PXINQ(1)=IQLNTH
0087          PXINQ(2)=IQLNTH+1
0088          DO 10 I=1,IFRSZ
0089            PACK(255,I)=0
0090       10 PACK(256,I)=0
0091          DO 20 I=1,IFRSZ
0092       20 FREE(I)=I
0093          DO 30 I=1,256
0094       30 ACKSEQ(I)=256
0095          IFR=IFRSZ
0096          RETURN
0097          END
```

```
0001          SUBROUTINE LINLOS
0002          REAL*8 RM1(5),LIN08,LIN18
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,PINQ
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006          LOGICAL*1 PACK,M1(40),LIN0(8),LIN1(8)
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             &           PACKQ(2),INQ(16),PINQ(2)
0009          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &            IOFLG,LLFLG,IRSEND
0010          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011          COMMON         PACK(256,64)
0012          COMMON /MESS/ MESSEQ
0013          DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0014          DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0015          DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0016          DATA RM1(5)/'DE 23   '/
0017          EQUIVALENCE(LIN0,LIN08)
0018          EQUIVALENCE(LIN1,LIN18)
0019          EQUIVALENCE(M1,RM1)
0020          IS=LLFLG
0021          CALL ENABLE(0)
0022          T1=IGETSP(N)
0023          CALL ENABLE(1)
0024          PACK(1,T1)=0
0025          IF(MESSEQ .EQ. 126) MESSEQ=0
0027          MESSEQ=MESSEQ+1
0028          PACK(2,T1)=MESSEQ
0029          PACK(3,T1)=0
0030          PACK(4,T1)=0
0031          PACK(5,T1)=25
0032          PACK(6,T1)=23
0033          DO 20 I=7,9
0034            PACK(I,T1)=LF
0035      20  CONTINUE
0036          DO 30 I=1,22
0037      30  PACK(I+9,T1)=M1(I)
0038          IF(IS .EQ. 1) GOTO 50
0040          DO 40 I=1,8
0041      40  PACK(I+31,T1)=LIN0(I)
0042          GOTO 70
0043      50  DO 60 I=1,8
0044      60  PACK(I+31,T1)=LIN1(I)
0045      70  DO 80 I=23,40
0046      80  PACK(I+17,T1)=M1(I)
0047          PACK(58,T1)=CR
0048          PACK(59,T1)=LF
0049          PACK(60,T1)=ETX
0050          CALL ENSTR(PACK(254,T1),60)
0051          CALL ENABLE(0)
0052          CALL ENQUE(PXINQ,XINQ,T1)
0053          CALL ENABLE(1)
```

```
0054          LLFLG=0
0055          RETURN
0056          END
```

```
0001          SUBROUTINE DSQC
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,IDATA
0005          LOGICAL*1 PACK,ISLID,MONTOR,LF,ETX,CR,DUM
0006          REAL*8 VTR
0007          REAL*4 RH,RL,AH,AL,VMEAS,IRSV
0008          COMMON PACK(256,64)
0009          COMMON /MESS/ MESSEQ
0010          COMMON /DCOM/ ISWTCH,VMEAS(3),ICTR,RL(3),RH(3),AL(3),AH(3),
             1     ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),IRSV(3),ICHNSW,
             2     ICHAN
0011          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0012          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1     PACKQ(2),INQ(16),PINQ(16)
0013          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,RESNLM,
             1     IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0014          DATA RH(1)/40.5/,RH(2)/19.9/,RH(3)/45.5/
0015          DATA RL(1)/-40.5/,RL(2)/-16.9/,RL(3)/-45.5/
0016          DATA AH(1)/35.5/,AH(2)/16.5/,AH(3)/40.5/
0017          DATA AL(1)/-35.5/,AL(2)/-16.5/,AL(3)/-40.5/
0018          DECODE(2,20,XINBF(1),ERR=410) ICNT
0019          DECODE(4,30,XINBF(3),ERR=410) VMEAS(1)
0020          DECODE(4,30,XINBF(7),ERR=410) VMEAS(2)
0021          DECODE(4,30,XINBF(11),ERR=410) VMEAS(3)
0022          DECODE(2,20,XINBF(15),ERR=410) IVC
0023          DECODE(2,20,XINBF(25),ERR=410) MON
0024    20    FORMAT(A2)
0025    30    FORMAT(A4)
0026          IF(ICHNSW .EQ. 1) GOTO 400
0028          DO 50 I=1,3
0029            IRSV(I)=(AH(I)+AL(I)) / 2
0030    50    CONTINUE
0031          DO 350 ICTR=1,3
0032            IF(VMEAS(ICTR) .GE. (AL(ICTR)+2) .AND.
             1     VMEAS(ICTR) .LE. (AH(ICTR)-2)) GOTO 350
0034            IF ((VMEAS(ICTR) .GT. (AH(ICTR)-2) .AND.
             1       VMEAS(ICTR) .LT. AH(ICTR)) .OR.
             2      (VMEAS(ICTR) .GT. AL(ICTR) .AND.
             3       VMEAS(ICTR) .LT. (AL(ICTR)+2))) GOTO 100
0036            IF ((VMEAS(ICTR) .GE. AH(ICTR) .AND.
             1       VMEAS(ICTR) .LE. (RH(ICTR)-1)) .OR.
             2      (VMEAS(ICTR) .GE. (RL(ICTR)+1) .AND.
             3       VMEAS(ICTR) .LE. AL(ICTR))) GOTO 200
0038            IF ((VMEAS(ICTR) .GT. (RH(ICTR)-1) .AND.
             1       VMEAS(ICTR) .LT. RH(ICTR)) .OR.
             2      (VMEAS(ICTR) .GT. RL(ICTR) .AND.
             3       VMEAS(ICTR) .LT. (RL(ICTR)+1))) GOTO 300
0040            IF(ISWTCH .EQ. 1) GOTO 40
0042            GOTO 410
0043    40      ISTAT=1
0044            GOTO 210
0045   100      IF(ISWTCH .EQ. 1) GOTO 110
```

```
0047          GOTO 410
0048    110   CALL TREND(2,VMEAS(ICTR))
0049          GOTO 410
0050    200   ISTAT=2
0051          IF(ISWTCH .EQ. 1) GOTO 210
0053          GOTO 410
0054    210   CALL ENABLE(0)
0055          K1=IGETSP(N)
0056          CALL ENABLE(1)
0057          IF(MESSEQ .EQ. 126) MESSEQ=0
0059          MESSEQ=MESSEQ+1
0060          PACK(1,K1)=0
0061          PACK(2,K1)=MESSEQ
0062          PACK(3,K1)=0
0063          PACK(4,K1)=0
0064          PACK(5,K1)=MONTOR
0065          PACK(6,K1)=ISLID
0066          DO 250 I=7,14
0067             PACK(I,K1)=XINBF(I+10)
0068    250   CONTINUE
0069          DO 260 I=15,16
0070            PACK(15,K1)=XINBF(I)
0071    260   CONTINUE
0072          PACK(17,K1)=ISTAT
0073          ENCODE(4,270,PACK(18,K1)) VMEAS(ICTR)
0074          ENCODE(2,280,PACK(22,K1)) MON
0075    270   FORMAT(A4)
0076    280   FORMAT(A2)
0077          PACK(24,K1)=ITERM
0078          CALL ENSTR(PACK(254,K1),24)
0079          CALL ENABLE(0)
0080          CALL ENQUE(PXINQ,XINQ,K1)
0081          CALL ENABLE(1)
0082          GOTO 410
0083    300   IF(ISWTCH .EQ. 1) GOTO 310
0085          GOTO 410
0086    310   CALL TREND(2,VMEAS(ICTR))
0087          GOTO 410
0088    350 CONTINUE
0089        GOTO 410
0090    400 CALL ENABLE(0)
0091        K1=IGETSP(N)
0092        CALL ENABLE(1)
0093        IF(MESSEQ .EQ. 126) MESSEQ=0
0095        MESSEQ=MESSEQ+1
0096        PACK(1,K1)=0
0097        PACK(2,K1)=MESSEQ
0098        PACK(3,K1)=0
0099        PACK(4,K1)=0
0100        PACK(5,K1)=25
0101        PACK(6,K1)=ISLID
0102        PACK(7,K1)=CR
0103        PACK(8,K1)=LF
0104        ENCODE(4,360,PACK(9,K1)) ICHAN
```

```
0105          PACK(13,K1)='40
0106          ENCODE(12,370,PACK(14,K1)) VMEAS(1)
0107          PACK(26,K1)='40
0108          ENCODE(12,370,PACK(27,K1)) VMEAS(2)
0109          PACK(39,K1)='40
0110          ENCODE(12,370,PACK(40,K1)) VMEAS(3)
0111          PACK(52,K1)=CR
0112          PACK(53,K1)=LF
0113          PACK(54,K1)=ETX
0114          CALL ENSTR(PACK(254,K1),54)
0115          CALL ENABLE(0)
0116          CALL ENQUE(PXINQ,XINQ,K1)
0117          CALL ENABLE(1)
0118          ICHNSW=0
0119    360   FORMAT(I4)
0120    370   FORMAT(F12.6)
0121    410   IOFLG=0
0122          RETURN
0123          END
```

```
0001          SUBROUTINE TREND(ICND,IVAL)
0002          REAL*4 RH,RL,AH,AL,VMEAS,IRSV,IVAL
0003          REAL*8 VTR
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,IDATA
0005          LOGICAL*1 PACK,ISLID,MONTOR,LF,ETX,CR,DUM
0006          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0007          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0008          COMMON PACK(256,64)
0009          COMMON /MESS/ MESSEQ
0010          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1     PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,RESNLM,
             1     IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0013          COMMON /DCOM/ ISWTCH,VMEAS(3),ICTR,RL(3),RH(3),AL(3),AH(3),
             1     ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),IRSV(3),ICHNSW,
             2     ICHAN
0014          IU=(RH(ICTR)-RL(ICTR))/30
0015          IF((IRSV(ICTR)-IVAL) .GT. (4 * IU)) GOTO 20
0017          IF(NFL .GE. 3) GOTO 40
0019          IF((IRSV(ICTR)-IVAL) .GT. (2 * IU)) GOTO 30
0021          NFL=NFL+1
0022     10   IRSV(ICTR)=(.33 * IVAL + .67 * IRSV(ICTR))
0023          GOTO 60
0024     20   IF(SFL .GE. 2) GOTO 30
0026          SFL=SFL+1
0027          GOTO 50
0028     30   IRSV(ICTR)=IVAL
0029          NFL=0
0030          GOTO 50
0031     40   SFL=0
0032          GOTO 10
0033     50   IF(ICND .EQ. 2) RETURN
0035     60   CALL ENABLE(0)
0036          K1=IGETSP(N)
0037          CALL ENABLE(1)
0038          IF(MESSEQ .EQ. 126) MESSEQ=0
0040          MESSEQ=MESSEQ+1
0041          PACK(1,K1)=0
0042          PACK(2,K1)=MESSEQ
0043          PACK(3,K1)=0
0044          PACK(4,K1)=0
0045          PACK(5,K1)=MONTOR
0046          PACK(6,K1)=ISLID
0047          DO 70 I=7,14
0048            PACK(I,K1)=XINBF(I+10)
0049     70   CONTINUE
0050          DO 75 I=15,16
0051            PACK(I,K1)=XINBF(I)
0052     75   CONTINUE
0053          ENCODE(4,80,PACK(18,K1)) IVAL
0054     80   FORMAT(A4)
0055          PACK(17,K1)=ICND
```

```
0056          ENCODE(2,90,PACK(22,K1)) MON
0057      90  FORMAT(A2)
0058          PACK(24,K1)=ITERM
0059          CALL ENSTR(PACK(254,K1),24)
0060          CALL ENABLE(0)
0061          CALL ENQUE(PXINQ,XINQ,K1)
0062          CALL ENABLE(1)
0063          RETURN
0064          END
```

```
0001          SUBROUTINE DSQCLP
0002          LOGICAL*1 IDATA
0003          REAL*4 RH,RL,AL,VMEAS,IRSV
0004          REAL*8 VTR
0005          COMMON /DCOM/ ISWTCH,VMEAS(3),ICTR,RL(3),RH(3),AL(3),AH(3),
             1     ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),IRSV(3),ICHNSW,
             2     ICHAN
0006          IF(IDATA(1) .EQ. '117 .AND.
             1  IDATA(2) .EQ. '116) GOTO 50
0008          IF(IDATA(1) .EQ. '117 .AND.
             1  IDATA(2) .EQ. '106 .AND.
             2  IDATA(3) .EQ. '106) GOTO 60
0010          IF(IDATA(1) .EQ. '115) GOTO 70
0012          GOTO 80
0013     50   ISWTCH=1
0014          ITERM=IDATA(4)
0015          RETURN
0016     60   ISWTCH=0
0017          RETURN
0018     70   DECODE(4,75,IDATA(2),ERR=80) ICHAN
0019     75   FORMAT(I4)
0020          CALL SOUT(ICHAN)
0021          ICHNSW=1
0022     80   RETURN
0023          END
```

```
                                .TITLE  FDM.MACRO
                                .SBTTL  NODE 23
                                .IDENT  /V3.0/
                                .GLOBL  LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
                                .GLOBL  WTOKEN,RAM,STATUS,STATBO
                                .GLOBL  LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
                                .NLIST  CND
000000                          .PSECT

000000          R0      =%0
000001          R1      =%1
000002          R2      =%2
000003          R3      =%3
000004          R4      =%4
000005          R5      =%5
000006          SP      =%6
000007          PC      =%7

                ;INTERFACE ADDRESS

172410          BAR=    172410
172412          WCR=    172412
172414          CSR=    172414
172416          IOBUF=  172416
172416          OPREG=  172416
177560          HRCSR=  177560
177562          HRBUF=  177562
177564          HXCSR=  177564
177566          HXBUF=  177566

                ;INTERFACE OPTIONS

000001          XCIO=1
000000          XSIO=0
000001          XGIO=1
000000          NDXX=0
000001          ND24=1
000000          NHOST=0
000001          DHOST=1

                ;LOCAL VARIBLES

000002          ARG1=   2
000004          ARG2=   4
000006          ARG3=   6
000010          ARG4=   10

100000          BIT15=  100000
040000          BIT14=  40000
000400          BIT08=  400
000200          BIT07=  200
000100          BIT06=  100
000040          BIT05=  40
000020          BIT04=  20
000010          BIT03=  10
000004          BIT02=  4
```

```
FDM.MACRO       MACRO V03.01 4-JUN-79 17:50:23 PAGE 1-1
NODE 23

     66          000002                          BIT01=  2
     67          000001                          BIT00=  1
     68
```

```
                                ;*************************************************************
                                ;**** PROCEDURE MASTER START UP                              *
                                ;*************************************************************

000000  012700  000340          MASTER: MOV     #340,   R0                      ;PRI=7
000004  106400                          MTPS    R0
000006  012700  000124                  MOV     #124,   R0
000012  012720  000430'                 MOV     #LIO,   (R0)+                   ;LIU HANDLER
000016  012710  000340                  MOV     #340,   (R0)                    ;PRI=7
000022  000005                          RESET                                   ;RESET BUS
000024  012700  000060                  MOV     #60,    R0
000030  012720  002470'                 MOV     #SIO,   (R0)+                   ;SIG INTERFACE HANDLER
000034  012720  000340                  MOV     #340,   (R0)+
000040  012700  000100                  MOV     #100,   R0                      ;CLOCK VECTOR
000044  012710  001654'                 MOV     #TIME,  (R0)                    ;CLOCK
000050  042737  040100  172414          BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000056  004767  000014                  JSR     PC,     LIUINT                  ;CLEAR LIU
000062  005037  172414                  CLR     @#CSR                           ;CLEAR BLIUI CSR
000066  012700  000000                  MOV     #000,   R0                      ;PRI=0
000072  106400                          MTPS    R0                              ;CPU DOWN
000074  000207                          RTS     PC


                                ;*************************************************************
                                ;**** PROCEDURE INITIALIZE                                  *
                                ;*************************************************************

                                        ;CLEAR ACRAM

000076  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000104  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000112  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000116  100375                          BPL     .-4                             ;NO RETRY
000120  105037  172414                  CLRB    @#CSR
000124  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
000132  012700  177400                  MOV     #-256., R0                      ;COUNTER
000136  012737  004407  172416  1$:     MOV     #2311., @#OPREG                 ;WRITE A NULL
000144  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000150  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
000152  105037  172414                  CLRB    @#CSR
000156  005200                          INC     R0                              ;COUNT+1
000160  001366                          BNE     1$

                                        ;CLEAR INPUT/OUTPUT BUFFERS

000162  012701  010410                  MOV     #10410, R1                      ;RDBUFADR CMD
000166  012700  010440                  MOV     #10440, R0                      ;SEL INBUF0
000172  004767  000106                  JSR     PC,     ZEROBP                  ;SET POINTER=0
000176  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000204  105737  172414                  TSTB    @#CSR                           ;GOOD READ
000210  100375                          BPL     .-4                             ;NO RETRY
000212  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
000216  012701  010610                  MOV     #10610, R1                      ;RDBUFADR COMMAND
000222  012700  010640                  MOV     #10640, R0                      ;SEL INBUF1 COMMAND
000226  004767  000052                  JSR     PC,     ZEROBP                  ;POINTER=0
```

```
    102 000232  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
    103 000240  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    104 000244  100375                          BPL     .-4                     ;NO RETRY
    105 000246  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    106 000252  012701  010510                  MOV     #10510, R1              ;RDBUFADR COMMAND
    107 000256  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
    108 000262  004767  000016                  JSR     PC,     ZEROBP
    109 000266  012701  010710                  MOV     #10710, R1              ;RDBUFADR COMMAND
    110 000272  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
    111 000276  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
    112 000302  000440                          BR      STAT$                   ;GO CLEAR STATUS
    113 000304  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
    114 000310  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
    115 000316  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    116 000322  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    117 000324  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    118 000330  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
    119 000334  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
    120 000340  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
    121 000344  022702  000000          2$:     CMP     #0,     R2              ;POINTER=0?
    122 000350  001412                          BEQ     3$                      ;YES RECHECK
    123 000352  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
    124 000360  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    125 000364  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    126 000366  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    127 000372  005302                          DEC     R2                      ;LOCAL POINTER-1
    128 000374  000763                          BR      2$                      ;BR UNTIL 0
    129 000376  010037  172416          3$:     MOV     R0,     @#OPREG         ;SEL BUFFER
    130 000402  000207                          RTS     PC                      ;RETURN
    131
    132                                         ;READ AND CLEAR STATUS
    133
    134 000404  012737  010400  172416  STAT$:  MOV     #4352.,  @#OPREG        ;WCR:RS
    135 000412  012737  002400  172416          MOV     #1280.,  @#OPREG        ;RS(FALSE)
    136 000420  012737  002400  172416          MOV     #1280.,  @#OPREG        ;RS
    137 000426  000207                          RTS     PC
    138
    139
```

```
 1                                ;*************************************************************
 2                                ;**** PROCEDURE LIU-HANDLER                                 *
 3                                ;*************************************************************
 4
 5
 6 000430  010046                 LIO:    MOV     R0,     -(SP)                   ;SAVE R0
 7 000432  010146                         MOV     R1,     -(SP)                   ;SAVE R1
 8 000434  010246                         MOV     R2,     -(SP)                   ;SAVE R2
 9 000436  010346                         MOV     R3,     -(SP)                   ;SAVE R3
10 000440  010446                         MOV     R4,     -(SP)                   ;SAVE R4
11 000442  010546                         MOV     R5,     -(SP)                   ;SAVE R5
12 000444  005737  172414                 TST     @#CSR                           ;? LIU INTERRUPTED
13 000450  100153                         BPL     RTI$                            ;NO
14 000452  042737  040000  172414        BIC     #BIT14, @#CSR                   ;YES/DISABLE LIU
15 000460  012737  010400  172416        MOV     #4352., @#OPREG                 ;WCR : RS(0)
16 000466  012737  002400  172416        MOV     #1280., @#OPREG                 ;RD
17 000474  013767  172416  000502        MOV     @#IOBUF, CAUSE
18 000502  042767  177400  000474        BIC     #177400, CAUSE                  ;CLEAR UNUSED BITS
19
20                                       ;PREFORM OPERATION DEPENDING ON BITS
21
22 000510  132767  000004  000466  B2$:  BITB    #BIT02, CAUSE                   ;OV-FL
23 000516  001406                       BEQ     B3$
24 000520  012702  010410                MOV     #4360., R2                      ;RDBUFADR IN0
25 000524  012703  010440                MOV     #4384., R3                      ;SEL INBUF0
26 000530  004767  000270                JSR     PC,     EMBF                    ;GO EMPTY
27 000534  132767  000010  000442  B3$:  BITB    #BIT03, CAUSE                   ;OV-FL
28 000542  001406                       BEQ     B4$
29 000544  012702  010610                MOV     #4488., R2                      ;RDBUFADR IN1
30 000550  012703  010640                MOV     #4512., R3                      ;SEL INBUF1
31 000554  004767  000244                JSR     PC,     EMBF                    ;GO EMPTY
32 000560  132767  000020  000416  B4$:  BITB    #BIT04, CAUSE                   ;LINE-LOSS PRIMARY
33 000566  001407                       BEQ     B5$
34 000570  132767  000200  000406  B7$:  BITB    #BIT07, CAUSE                   ;LATCH ON
35 000576  001403                       BEQ     B5$                             ;YES LOOP AROUND
36 000600  012767  000002  002002'      MOV     #2,     LLFLG                   ;FLAG PRIMARY SW. FAIL
37 000606  132767  000040  000370  B5$:  BITB    #BIT05, CAUSE                   ;BACKUP LINE LOSS
38 000614  001403                       BEQ     B6$
39 000616  012767  000001  002002'      MOV     #1,     LLFLG                   ;FLAG BACKUP SW. FAIL
40 000624  132767  000100  000352  B6$:  BITB    #BIT06, CAUSE                   ;WRITE TOKEN DETECT
41 000632  001404                       BEQ     B0$
42 000634  005067  001000'              CLR     IWRTTM                          ;CLEAR TOKEN VAR
43 000640  005067  001002'              CLR     IWRT
44
45                                       ;BUFFER FULL
46
47 000644  132767  000001  000332  B0$:  BITB    #BIT00, CAUSE                   ;INBUF0 FULL
48 000652  001423                       BEQ     B1$                             ;NO
49 000654  012701  000400'              MOV     #INBF,  R1                      ;BUFFER ADDRESS
50 000660  012702  010410                MOV     #4360., R2                      ;RDBUFADR COMM
51 000664  012703  010440                MOV     #4384., R3                      ;SEL BUF
52 000670  012704  000001                MOV     #BIT00, R4                      ;CRC BIT
53 000674  004767  000124                JSR     PC,     EMBF                    ;EMPTY BUFFER
54 000700  012705  001206'              MOV     #AREA,  R5                      ;DATA LINK AREA
55 000704  012767  000001  000274        MOV     #1,     AREA                    ;1 VARIBLE
56 000712  010267  000274                MOV     R2,     DATA                    ;CRC OR BYTE COUNT
57 000716  004767  000000G               JSR     PC,     LPINPT                  ;CALL FORTRAN QUE'ER
```

```
 58 000722  132767  000002  000254  BI$:    BITB    #BIT01, CAUSE                   ;INBUF1 FULL
 59 000730  001423                           BEQ     RTI$                            ;NO WERE DONE
 60 000732  012701  000400'                  MOV     #INBF,  R1                      ;BUFFER ADDRESS
 61 000736  012702  010610                   MOV     #4488., R2                      ;RDBUFADR COMM
 62 000742  012703  010640                   MOV     #4512., R3                      ;SEL INBUF1
 63 000746  012704  000002                   MOV     #BIT01, R4                      ;CRC BIT
 64 000752  004767  000046                   JSR     PC,     EMBF                    ;GO EMPTY BUFFER
 65 000756  012705  001206'                  MOV     #AREA,  R5                      ;DATA LINK AREA
 66 000762  012767  000001  000216          MOV     #1,     AREA                    ;ONE VARIBLE
 67 000770  010267  000216                   MOV     R2,     DATA                    ;CRC OR BYTE COUNT
 68 000774  004767  000000G                  JSR     PC,     LPINPT                  ;CALL FORTRAN QUE'ER
 69
 70 001000  012605                  RTI$:    MOV     (SP)+,  R5                      ;RESTORE REGISTERS
 71 001002  012604                           MOV     (SP)+,  R4
 72 001004  012603                           MOV     (SP)+,  R3
 73 001006  012602                           MOV     (SP)+,  R2
 74 001010  012601                           MOV     (SP)+,  R1
 75 001012  012600                           MOV     (SP)+,  R0
 76 001014  052737  040000  172414          BIS     #BIT14, @#CSR                   ;ENABLE INTERRUPTS
 77 001022  000002                           RTI                                     ;RETURN FROM INTERRUPT
 78
 79 001024  012737  010600  172416  EMBF:    MOV     #4480., @#OPREG                 ;READ STATUS 1
 80 001032  012737  002400  172416           MOV     #1280., @#OPREG                 ;RS
 81 001040  013700  172416                   MOV     @#IOBUF, R0                     ;FETCH STATUS
 82 001044  130400                           BITB    R4,R0                           ;GOOD CRC
 83 001046  001002                           BNE     CRCOK
 84 001050  012704  177777                   MOV     #-1,    R4                      ;NO FLAG
 85 001054  010237  172416          CRCOK:   MOV     R2,     @#OPREG                 ;RDBUFADR
 86 001060  012737  001400  172416           MOV     #768.,  @#OPREG                 ;RD
 87 001066  105737  172414                   TSTB    @#CSR                           ;GOOD RD
 88 001072  100375                           BPL     .-4
 89 001074  013702  172416                   MOV     @#IOBUF, R2                     ;REPLACE WITH POINTER
 90 001100  042702  177400                   BIC     #177400, R2                     ;CLEAR MST BITS
 91 001104  010200                           MOV     R2,     R0                      ;SAVE IT
 92 001106  005400                           NEG     R0                              ;2'S COMP
 93 001110  010037  172412                   MOV     R0,     @#WCR                   ;BYTE COUNT NOW
 94 001114  010137  172410                   MOV     R1,     @#BAR                   ;ADDRESS IN MEMORY
 95 001120  010337  172416                   MOV     R3,     @#OPREG                 ;SEL BUFFER
 96 001124  012737  001400  172416           MOV     #768.,  @#OPREG                 ;FALSE RD
 97 001132  105737  172414                   TSTB    @#CSR                           ;DONE ON
 98 001136  100375                           BPL     .-4                             ;NO LOOP
 99 001140  012737  021000  172416           MOV     #8704., @#OPREG                 ;FIRE DMA
100 001146  000240                           NOP                                     ;DELAY
101 001150  105737  172414                   TSTB    @#CSR                           ;GOOR DMA
102 001154  100401                           BMI     DMAOK
103 001156  000240                           NOP                                     ;ERROR IF HERE
104 001160  012737  004400  172416  DMAOK:   MOV     #2304., @#OPREG                 ;FALSE WD
105 001166  105737  172414                   TSTB    @#CSR                           ;GOOD WD
106 001172  100375                           BPL     .-4
107 001174  005704                           TST     R4                              ;WAS CRC OK
108 001176  100001                           BPL     END$                            ;YES
109 001200  010402                           MOV     R4,     R2                      ;NO FLAG IT
110 001202  000207                  END$:    RTS     PC                              ;RETURN
111 001204  000000                  CAUSE:   .WORD   0                               ;STATUS BYTE 0 HOLDER
112 001206  000000                  AREA:    .WORD   0
113 001210  001212'                          .WORD   DATA
114 001212  000000                  DATA:    .WORD   0
```

```
                                        .ENABLE LSB
                                ;*************************************************************
                                ;**** PROCEDURE ENABLE                                      *
                                ;*************************************************************

001214  017501  000002          ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
001220  022701  000001                  CMP     #1,     R1
001224  001404                          BEQ     E1$
001226  042737  040000  172414  E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
001234  000207                          RTS     PC
001236  052737  040000  172414  E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
001244  000207                          RTS     PC

                                        ;FORTRAN ENCODE/DECODE MACRO

001246  017575  000004  000002  ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
001254  000207                          RTS     PC
001256  017575  000002  000004  DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECODE BYTE COUNT
001264  000207                          RTS     PC


                                ;*************************************************************
                                ;**** PROCEDURE LINE SWITCH                                 *
                                ;*************************************************************

                                        ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                        ;
                                        ;04=SET PRIMARY LINE
                                        ;32=RESET PRIMARY LINR
                                        ;08=SET BACKUP LINE
                                        ;64=RESET BACKUP LINE

001266  017501  000002          SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
001272  012737  010420  172416          MOV     #4368., @#OPREG                 ;WCR : MODSTAT
001300  062701  004400                  ADD     #2304., R1
001304  010137  172416                  MOV     R1,     @#OPREG                 ;RD
001310  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001314  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001316  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001322  000207                          RTS     PC
```

```
 1                                ;*************************************************************
 2                                ;**** PROCEDURE STATUS                                      *
 3                                ;*************************************************************
 4
 5                                        ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
 6                                        ;
 7                                        ;X=0 STATUS BYTE 0
 8                                        ;X=1 STATUS BYTE 1
 9
10 001324  017501  000002          STATUS: MOV     @ARG1(R5), R1                   ;WHICH BYTE
11 001330  022701  000001                  CMP     #1,     R1                      ;COMPARE
12 001334  001415                          BEQ     2$
13 001336  012737  010400  172416  1$:     MOV     #4352., @#OPREG                 ;WCR : RS(0)
14 001344  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
15 001352  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
16 001356  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
17 001362  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
18 001366  000207                          RTS     PC
19 001370  012737  010600  172416  2$:     MOV     #4480., @#OPREG                 ;WCR : RS(1)
20 001376  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
21 001404  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
22 001410  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
23 001414  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
24 001420  000207                          RTS     PC
25 001422  012737  010400  172416  STATB0: MOV     #4352., @#OPREG                 ;STATUS BYTE 1
26 001430  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
27 001436  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
28 001442  042701  177400                  BIC     #177400, R1
29 001446  132701  000010                  BITB    #BIT03, R1                      ;BUFFER FULL
30 001452  001404                          BEQ     3$
31 001454  012775  000001  000002          MOV     #1,     @ARG1(R5)               ;LOAD RETURN
32 001462  000207                          RTS     PC
33 001464  012775  000000  000002  3$:     MOV     #0,     @ARG1(R5)               ;LOAD RETURN
34 001472  000207                          RTS     PC
35
36
```

```
                                ;*************************************************************
                                ;**** PROCEDURE WRITE TOKEN (WTOKEN)                        *
                                ;*************************************************************

                                        ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP

001474  012737  010600  172416  WTOKEN: MOV     #4480., @#OPREG                 ;READ STATUS(1)
001502  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001510  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001514  132701  000010                  BITB    #BIT03, R1                      ;BUFFER LOADED
001520  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001522  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001530  012737  004423  172416          MOV     #2323., @#OPREG                 ;MARK OB0F/OB1F/WTCMD
001536  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001542  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001544  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001550  000207                          RTS     PC
001552  012737  010740  172416  WTKG:   MOV     #4576., @#OPREG                 ;SELECT BUFFER 1
001560  012737  001652' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001566  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001572  005402                          NEG     R2                              ;FORMAT IT
001574  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001600  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001606  000240                          NOP                                     ;DELAY
001610  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001614  000240                          NOP                                     ;ERROR IF HERE
001616  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001622  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001630  012737  004422  172416          MOV     #2322., @#OPREG                 ;MARK OB1F/WTCMD
001636  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001642  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001644  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001650  000207                          RTS     PC                              ;RETURN
001652     000     377          WTD:    .BYTE 0,377                             ;A WTOKEN


                                ;*************************************************************
                                ;**** PROCEDURE TIMER (TIME)                                *
                                ;*************************************************************

001654  005267  001006' TIME:   INC     LTIME                           ;TIME+1
001660  022767  077777  001006' CMP     #077777, LTIME                  ;?TIME FULL
001666  001002                  BNE     4$
001670  005067  001006'         CLR     LTIME                           ;CLEAR CLOCK
001674  000002                  4$:     RTI
```

```
 1                              ;*************************************************************
 2                              ;**** PROCEDURE ACRAM                                       *
 3                              ;*************************************************************
 4
 5                                      ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
 6                                      ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
 7                                      ;
 8                                      ;0110   =NREAD
 9                                      ;0100   =DREAD
10                                      ;0111   =NULL
11                                      ;0000   =WTOKEN
12
13
14 001676  017500  000002          RAM:    MOV     @ARG1(R5), R0                   ; RAM
15 001702  022700  000000                  CMP     #0,     R0                      ;WHICH OPERATION
16 001706  001035                          BNE     RDRAM                           ;READ OP
17 001710  017500  000004          WTRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS
18 001714  017501  000006                  MOV     @ARG3(R5), R1                   ;WRITE DATA
19 001720  012737  010402  172416        MOV     #4354., @#OPREG                 ;SEL LDACTR
20 001726  062700  004400                  ADD     #2304., R0                      ;WD/DATA (ADDR)
21 001732  010037  172416                  MOV     R0,     @#OPREG                 ;WRITE DATA
22 001736  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
23 001742  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
24 001744  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
25 001750  012737  010401  172416        MOV     #4353., @#OPREG                 ;SEL ACRAM
26 001756  062701  004400                  ADD     #2304., R1                      ;WD/DATA (CMD)
27 001762  010137  172416                  MOV     R1,     @#OPREG                 ;WRITE
28 001766  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
29 001772  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
30 001774  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
31 002000  000207                          RTS     PC
32 002002  017500  000004          RDRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS TO READ
33 002006  012737  010402  172416        MOV     #4354., @#OPREG                 ;SEL LDACTR
34 002014  062700  004400                  ADD     #2304., R0                      ;WD/DATA (ADDR)
35 002020  010037  172416                  MOV     R0,     @#OPREG                 ;WRITE DATA
36 002024  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
37 002030  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
38 002032  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
39 002036  012737  010401  172416        MOV     #4353., @#OPREG                 ;SEL ACRAM
40 002044  012737  001400  172416        MOV     #768.,  @#OPREG                 ;READ DATA
41 002052  105737  172414                  TSTB    @#CSR                           ;VALID READ
42 002056  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
43 002060  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
44 002064  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
45 002070  042700  177760                  BIC     #177760, R0                     ;CLEAR BITS
46 002074  010075  000006                  MOV     R0, @ARG3(R5)                   ;RTN DATA
47 002100  000207                          RTS     PC
48
49
```

```
                                .ENABLE LSB
                        ;*************************************************************
                        ;**** PROCEDURE WRITE LOOP (LPOUT)                          *
                        ;*************************************************************

002102  017502  000002          LPOUT:  MOV     @ARG1(R5), R2                   ;BYTE COUNT
002106  012700  000340                  MOV     #340,   R0                      ;PRI=7
002112  106400                          MTPS    R0
002114  012737  000000' 172410          MOV     #OUTBF, @#BAR                   ;BUS ADDRESS
002122  005402                          NEG     R2                              ; 2'S COMP COUNT
002124  010237  172412                  MOV     R2,     @#WCR                   ;COUNT
002130  012737  010540  172416          MOV     #4448., @#OPREG                 ;OBO COMMAND
002136  012737  024000  172416          MOV     #10240., @#OPREG                ;DMA GO
002144  000240                          NOP                                     ;INTERFACE TIME
002146  105737  172414                  TSTB    @#CSR                           ;DMA OK
002152  000240                          NOP                                     ;ERROR IF HERE
002154  105037  172414                  CLRB    @#CSR
002160  012737  010740  172416          MOV     #4576., @#OPREG                 ;OB1 COMMAND
002166  012737  001652' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
002174  012702  000002                  MOV     #2,     R2                      ;BYTE COUNT
002200  005402                          NEG     R2                              ;FORMAT FOR BLIUI
002202  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
002206  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
002214  000240                          NOP                                     ;DELAY
002216  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
002222  000240                          NOP                                     ;ERROR IF NOT
002224  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
002230  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
002236  012737  004403  172416          MOV     #2307., @#OPREG                 ;BUFFERS FULL
002244  105737  172414                  TSTB    @#CSR                           ;GOOD WRITE
002250  100375                          BPL     .-4                             ;NO LOOP UNTIL
002252  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002256  012700  000000                  MOV     #000,   R0                      ;PRI=0
002262  106400                          MTPS    R0                              ;LOWER CPU
002264  000207                          RTS     PC
```

```
                                ;*************************************************************
                                ;**** PROCEDURE READ BUFFER POINTER                          *
                                ;*************************************************************

002266  017500  000002          RDPNT:  MOV     @ARG1(R5), R0                   ;FETCH COMMAND
002272  022700  000000                  CMP     #0,     R0                      ;INBUF0 ?
002276  001004                          BNE     1$                              ;NO
002300  012700  010410                  MOV     #4360., R0                      ;RDBUFADR IN0
002304  000167  000046                  JMP     4$
002310  022700  000001          1$:     CMP     #1,     R0                      ;INBUF1 ?
002314  001004                          BNE     2$                              ;NO
002316  012700  010610                  MOV     #4488., R0                      ;RDBUFADR IN1
002322  000167  000030                  JMP     4$
002326  022700  000002          2$:     CMP     #2,     R0                      ;OUTBUF0 ?
002332  001004                          BNE     3$                              ;NO
002334  012700  010510                  MOV     #4424., R0                      ;RDBUFADR OUT0
002340  000167  000012                  JMP     4$
002344  022700  000004          3$:     CMP     #4,     R0                      ;OUTBUF1 ?
002350  001020                          BNE     5$                              ;NO RETURN
002352  012700  010710                  MOV     #4552., R0                      ;RDBUFADR OUT1
002356  010037  172416          4$:     MOV     R0,     @#OPREG                 ;RDBUFADR
002362  012737  001400  172416        MOV     #768.,  @#OPREG                 ;RD
002370  105737  172414                TSTB    @#CSR                           ;READY
002374  100375                        BPL     .-4                             ;NO LOOP UNTIL
002376  013700  172416                MOV     @#IOBUF, R0                     ;FETCH POINTER
002402  042700  177400                BIC     #177400, R0                     ;CLEAR MST
002406  010075  000004                MOV     R0,     @ARG2(R5)               ;RETURN POINTER
002412  000207          5$:     RTS     PC
```

```
                        ;*************************************************************
                        ;**** PROCEDURE RESTART                                     *
                        ;*************************************************************

                                ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
                                ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
                                ;CALL RSTART(2) -LOAD MODE(173000)

002414  017500  000002  RSTART: MOV     @ARG1(R5), R0                   ;MODE
002420  022700  000000  RS0:   CMP     #0,     R0                      ;LOAD ADDRESS
002424  001002                 BNE     RS1
002426  000000                 HALT
002430  000207                 RTS     PC
002432  022700  000001  RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
002436  001005                 BNE     RS2
002440  004767  175432         JSR     PC,     LIUINT                  ;INT LIU FIRST
002444  012700  000040  G4$:    MOV     #40,    R0                      ;START ADDRESS
002450  000110                 JMP     (R0)
002452  022700  000002  RS2:    CMP     #2,     R0                      ;LOAD MODE
002456  001002                 BNE     RST
002460  000177  000002         JMP     @ROM
002464  000207          RST:    RTS     PC
002466  173000          ROM:    .WORD   173000
```

```
                                .ENABLE LSB
                        ;*************************************************************
                        ;****  PROCEDURE SIG INTERFACE                              *
                        ;*************************************************************

002470  010046          SIO::   MOV     R0,     -(SP)                   ;SAVE REGISTERS
002472  010146                  MOV     R1,     -(SP)
002474  010246                  MOV     R2,     -(SP)
002476  012700  000400'          MOV     #XINBF, R0                      ;BUFFER ADDRESS
002502  012701  000050          MOV     #40.,   R1                      ;BYTE COUNT
002506  105737  177560  1$:     TSTB    @#HRCSR                         ;PORT READY
002512  100375                  BPL     .-4                             ;LOOP UNTIL
002514  113702  177562          MOVB    @#HRBUF, R2                     ;FETCH BYTE
002520  042702  177400          BIC     #177400, R2                     ;CLEAR MST
002524  010267  002000'          MOV     R2,     IOFLG                   ;ITS THE BYTE COUNT
002530  110220                  MOVB    R2,     (R0)+
002532  005301                  DEC     R1                              ;COUNT-1
002534  105737  177560  2$:     TSTB    @#HRCSR                         ;PORT READY
002540  100375                  BPL     .-4                             ;LOOP UNTIL
002542  113702  177562          MOVB    @#HRBUF, R2                     ;FETCH BYTE
002546  042702  177400          BIC     #177400, R2                     ;CLEAR MST
002552  110220                  MOVB    R2,     (R0)+                   ;STORE IT
002554  005301                  DEC     R1
002556  001366                  BNE     2$                              ;READ 39 BYTES
002560  042737  000100  177560  BIC     #100,   @#HRCSR                 ;NO MORE UNTIL F@RTRAN
002566  012602          3$:     MOV     (SP)+,  R2                      ;RETORE REGISTERS
002570  012601                  MOV     (SP)+,  R1
002572  012600                  MOV     (SP)+,  R0
002574  000002                  RTI

                                ;SEND OVER TO SIG

002576  017500  000002  SOUT::  MOV     @ARG1(R5),R0                    ;BYTE COUNT
002602  105737  177564          TSTB    @#HXCSR                         ;READY TO SNED
002606  100375                  BPL     .-4                             ;LOOP UNTIL
002610  110037  177566          MOVB    R0,     @#HXBUF                 ;SEND IT
002614  000207                  RTS     PC


                                ;REPORT NODE READY

002616  105737  177564  READY:: TSTB    @#HXCSR                         ;PORT READY
002622  100375                  BPL     .-4                             ;LOOP UNTIL
002624  112737  000122  177566  MOVB    #122, @#HXBUF                   ;SEND R
002632  052737  000100  177560  BIS     #100, @#HRCSR                   ;ENABLE INTERRUPTS
002640  000207                  RTS     PC                              ;RETURN
```

```
                        ;*************************************************************
                        ;****        COMMON DATA AREA                                 *
                        ;*************************************************************

000000                          .PSECT  DFM,RW,D,GBL,REL,OVR

000000                  OUTBF:  .BLKB   256.
000400                  INBF:   .BLKB   256.
001000                  IWRTTM: .BLKW
001002                  IWRT:   .BLKW
001004                  STAT:   .BLKW
001006                  LTIME:  .BLKW
                                .EVEN

000000                          .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                  XOUTBF: .BLKB   256.
000400                  XINBF:  .BLKB   256.
001000                  ACKSEQ: .BLKB   256.
001400                  LIDFD:  .BLKB   256.
002000                  IOFLG:  .BLKW
002002                  LLFLG:  .BLKW
002004                  IRSEND: .BLKW
                                .EVEN


000001                          .END
```

```
FDM.MACRO       MACRO V03.01 4-JUN-79 17:50:23 PAGE 12-1
SYMBOL TABLE

ACKSEQ  001000R      003 B1$     000722R         E1$     001236R         MASTER  000000RG        SOUT    002576RG
AREA    001206R          B2$     000510R         G4$     002444R         NDXX  = 000000          STAT    001004R      002
ARG1  = 000002           B3$     000534R         HRBUF = 177562          ND24  = 000001          STATBO  001422RG
ARG2  = 000004           B4$     000560R         HRCSR = 177560          NHOST = 000000          STATUS  001324RG
ARG3  = 000006           B5$     000606R         HXBUF = 177566          OPREG = 172416          STAT$   000404R
ARG4  = 000010           B6$     000624R         HXCSR = 177564          OUTBF   000000R     002 SWITCH  001266RG
BAR   = 172410           .7$     000570R         INBF    000400R     002 RAM     001676RG        TIME    001654RG
BIT00 = 000001           CAUSE   001204R         IOBUF = 172416          RDPNT   002266RG        WCR   = 172412
BIT01 = 000002           CRCOK   001054R         IOFLG   002000R     003 RDRAM   002002R         WTD     001652R
BIT02 = 000004           CSR   = 172414          IRSEND  002004R     003 READY   002616RG        WTKG    001552R
BIT03 = 000010           DATA    001212R         IWRT    001002R     002 ROM     002466R         WTOKEN  001474RG
BIT04 = 000020           DESTR   001256RG        IWRTTM  001000R     002 RST     002464R         WTRAM   001710R
BIT05 = 000040           DHOST = 000001          LIDFD   001400R     003 RSTART  002414RG        XCIO  = 000001
BIT06 = 000100           DMAOK   001160R         LIO     000430RG        RS0     002420R         XGIO  = 000001
BIT07 = 000200           EMBF    001024R         LIUINT  000076RG        RS1     002432R         XINBF   000400R     003
BIT08 = 000400           ENABLE  001214RG        LLFLG   002002R     003 RS2     002452R         XOUTBF  000000R     003
BIT14 = 040000           END$    001202R         LPINPT= ****** G        RTI$    001000R         XSIO  = 000000
BIT15 = 100000           ENSTR   001246RG        LPOUT   002102RG        SIO     002470RG        ZEROBP  000304R
BO$     000644R          EO$     001226R         LTIME   001006R     002

. ABS.  000000      000
        002642      001
DFM     001010      002
BUFS    002006      003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:FDM=DK:FDM
```

```
FOR NODAL/NOLINE
FOR NODAL1/NOLINE
FOR DSQC/NOLINE
MAC FDM
```

```
ASSIGN DX0: DK:
R LINK
DX1:NODE23,DX1:MAP=DX1:NODAL/C/I/W
DX1:FDM/C
DX1:NODAL1/C
DX1:DSRC//
$SIMRT

ASSIGN DX1: DK:
```

1.5 NODE 24 DMCP (DBMS)

The Data Base Management Service (DBMS) performs the data base maintenance functions. It maintains the display files for the human interface User Language, system configuration files, equipment status files, and object files for the various modules. The User Language runs on the DBMS and is used to control the system. Mini disks are used to store the various files. The DBMS may also be used as a Program Development Unit (PDU) with a local CRT terminal when not being used for the MSCDM application. The PDU develops and maintains software for the MSCDM. The PDU runs the RT-11 Operating System and it may be used as a general purpose processor.

1.5.1 Program Descriptions

1.5.1.1 Refer to Section 1.1 for descriptions of routines - NODAL, IGETSP, ENQUE, DEQUE, ACKNAK, INPTQ, LPINPT, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME, RAM, LPOUT, RDPNT, RSTART.

1.5.1.2 Subroutine USRLANG

MSCDM user language is the applications program for node 24, refer to Section 1.10 for description and listings.

```
0001          PROGRAM NODAL
0002          INTEGER*2 XINQ,PXINQ
0003          INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP
0005          INTEGER*2 RSTBKP,Q1,Q2,RESNLM,OUTQ,DEQUE
0006          INTEGER*2 ST,DEVNUM,FTYPE,RECNUM
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,TEMP(6),OUTBF,INBF
0008          LOGICAL*1 F7,F8,F9,F10,F11,F12,IBUFF
0009          LOGICAL*1 PACK,ICODE,ETX,CR,LF,MONTOR,ISLID,ICON,DUM,IBUF
0010          INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0011          INTEGER*2 KEYREC
0012          COMMON /MESS/ MESSEQ
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0014          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0015          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1          LIDFD(256),IOFLG,LLFLG,IRSEND
0016          COMMON /FRE/ FREE(30),IFR,IFRSZ
0017          COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0018          COMMON       PACK(256,30)
0019          COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0020          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0021          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0022          COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             1     KEYREC,IBUF(80)
0023          COMMON /M3000/ IFUNT,ITYPE,INODE,IMEAS
0024          COMMON /M4000/ FTYPE,RECNUM,F7,F8,F9,F10,F11,F12
0025          COMMON /M6000/ IDISP,IEQUIP,ICHAN,INUM,IBUFF(20),IFLAG,ICNT
0026          CALL MASTER(0)
0027          CALL INIT(0)
0028          CALL ENABLE(1)
0029          IFIRST=0
0030          CALL USRLNG(0)
0031        5 IF(IABORT .EQ. 1) GOTO 999
0033          IF(IFLAG .EQ. 0 .OR. IRSEND .EQ. 1) GOTO 25
0035          CALL M6000
0036          DO 27 J=1,20000
0037       27 CONTINUE
0038       25 IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0040          CALL ENABLE(0)
0041          Q2=DEQUE(PINQ,INQ,1)
0042          CALL ENABLE(1)
0043          CALL DESTR(PACK(254,Q2),LEN)
0044          DO 26 I=1,80
0045            ICODE(I)=0
0046       26 CONTINUE
0047          DO 30 I=1,LEN
0048            ICODE(I)=PACK(I,Q2)
0049       30 CONTINUE
0050          CALL USRLNG(LEN)
0051          IFR=IFR+1
0052          FREE(IFR)=Q2
0053       40 CALL STATBO(IS)
```

```
0054          IF(IS .EQ. 1) GOTO 100
0056          IF(LLFLG .GT. 0) CALL LINLOS
0058     80   IF(OUTFCT .EQ. 1) GOTO 100
0060          IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0062          IF(IRSEND .EQ. 1) GOTO 85
0064          IF(PXINQ(1) .LT. PXINQ(2)) GOTO 100
0066          CALL ENABLE(0)
0067          Q1=DEQUE(PXINQ,XINQ,1)
0068          CALL ENABLE(1)
0069          OUTQ=Q1
0070     85   CALL DESTR(PACK(254,OUTQ),Q2)
0071          IF(Q2 .LT. 6) GOTO 100
0073          DO 90 I=1,Q2
0074            OUTBF(I)=PACK(I,OUTQ)
0075     90   CONTINUE
0076          OUTBF(Q2+1)=0
0077          IPT=OUTBF(5)
0078          OUTBF(Q2+2)=LIDFD(IPT)
0079          IF(Q2+2 .LT. 6 .OR. Q2+2 .GT. 255) GOTO 100
0081          IFULL=1
0082          ISENT=1
0083          CALL LPOUT(Q2+2)
0084          IRSEND=1
0085          ACKTIM=0
0086          IWRTTM=0
0087          IWRT=1
0088    100   INFLCT=0
0089    110   OLDTIM=NEWTIM
0090          NEWTIM=LTIME
0091          IF(IWRT .EQ. 0) GOTO 120
0093          IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0094          IF(IWRTTM .LT. TIMLIM) GOTO 120
0096          CALL WTOKEN
0097          IWRTTM=0
0098          IWRT=0
0099    120   IF(ISENT .EQ. 0) GOTO 130
0101          ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0102          IF(ACKTIM .LT. ATIMLM) GOTO 130
0104          CALL ENABLE(0)
0105          CALL ACKNAK(0)
0106          CALL ENABLE(1)
0107    130   CONTINUE
0108          GOTO 5
0109    999   CALL ENABLE(0)
0110          CALL MASTER(1)
0111          STOP
0112          END
```

```
0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(30),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IF(IFR .LT. 1) CALL INIT(1)
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```

```
FORTRAN IV     V02.1-1     Tue 05-Jun-79 13:42:14                PAGE 001

0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,OUTQ,A(2),B(15)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008     10   IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016     20   IF(IQHEAD .GE. IQLNTH) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020     30   B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024     40   CALL INIT(1)
0025    999   RETURN
0026          END
```

```
0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(15),DEQUE,OUTFCT,OUTQ,RESNLM
0005          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016       10 A(1)=IQHEAD-1
0017      999 RETURN
0018          END
```

```
0001          SUBROUTINE ACKNAK(N)
0002          INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ
0003          INTEGER*2 PINQ,FREE,OUTQ,FINDPK
0004          INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006          LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008          INTEGER*2 LTIME,IWRTTM
0009          REAL*8 COMT(5)
0010          EQUIVALENCE(COMT,LCOMT)
0011          DATA COMT(1)/'MESSAGE '/,COMT(2)/'NOT RECE'/,COMT(3)/'IVED FRO'/,
             1     COMT(4)/'M NODE  '/,COMT(5)/'TO NODE '/
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1       IWRTTM,IWRT,STAT,LTIME
0014          COMMON         PACK(256,30)
0015          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0016          COMMON /FRE/  FREE(30),IFR,IFRSZ
0017          COMMON /MESS/ MESSEQ
0018          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0019          IF (N .NE. 1) GOTO 10
0021          IFR=IFR+1
0022          DO 5 I=1,256
0023            PACK(I,OUTQ)=0
0024      5   CONTINUE
0025          IFULL=0
0026          ISENT=0
0027          IRSEND=0
0028          FREE(IFR)=OUTQ
0029          GOTO 999
0030     10   T2=PACK(256,OUTQ)
0031          T2=T2+1
0032          IF(T2 .LE. RESNLM) GOTO 120
0034          T3=PACK(3,OUTQ)
0035          T4=MOD(T3,64)
0036          IF(T4 .LT. 32) GOTO 110
0038          T5=IGETSP(N)
0039          IF(MESSEQ .EQ. 126) MESSEQ=0
0041          MESSEQ=MESSEQ+1
0042          PACK(1,T5)=0
0043          PACK(2,T5)=MESSEQ
0044          PACK(3,T5)=0
0045          PACK(4,T5)=0
0046          PACK(5,T5)=MONTOR
0047          PACK(6,T5)=ISLID
0048          DO 20 I=7,60
0049            PACK(I,T5)='40
0050     20   CONTINUE
0051          DO 30 I=7,9
0052            PACK(I,T5)=LF
0053     30   CONTINUE
0054          DO 40 I=1,8
```

```
0055            PACK(I+9,T5)=LCOMT(I)
0056     40   CONTINUE
0057          DO 50 I=9,16
0058            PACK(I+9,T5)=LCOMT(I)
0059     50   CONTINUE
0060          DO 60 I=17,24
0061            PACK(I+9,T5)=LCOMT(I)
0062     60   CONTINUE
0063          DO 70 I=25,31
0064            PACK(I+9,T5)=LCOMT(I)
0065     70   CONTINUE
0066          ENCODE(3,80,PACK(41,T5))PACK(6,OUTQ)
0067     80   FORMAT(I3)
0068          DO 90 I=33,40
0069            PACK(I+12,T5)=LCOMT(I)
0070     90   CONTINUE
0071          ENCODE(3,80,PACK(53,T5))PACK(5,OUTQ)
0072          PACK(56,T5)=LF
0073          PACK(57,T5)=LF
0074          PACK(58,T5)=CR
0075          PACK(59,T5)=ETX
0076          CALL ENSTR(PACK(254,T5),59)
0077          CALL ENQUE(PXINQ,XINQ,T5)
0078          DO 102 I=1,256
0079            PACK(I,OUTQ)=0
0080    102   CONTINUE
0081          ISENT=0
0082          IRSEND=0
0083          IFULL=0
0084          IWRT=0
0085          IFR=IFR+1
0086          FREE(IFR)=OUTQ
0087          GOTO 999
0088     110  PACK(3,OUTQ)=PACK(3,OUTQ)+32
0089          PACK(256,OUTQ)=0
0090          ISENT=0
0091          IFULL=0
0092          IF(IRSEND .EQ. 1) GOTO 999
0094          CALL ENQUE(PXINQ,XINQ,OUTQ)
0095          GOTO 999
0096    120   PACK(256,OUTQ)=T2
0097          IFULL=0
0098          ISENT=0
0099          IF(IRSEND .EQ. 1) GOTO 999
0101          CALL ENQUE(PXINQ,XINQ,OUTQ)
0102    999   RETURN
0103          END
```

```
0001          SUBROUTINE INPTQ(L)
0002          INTEGER*2 STAT,XINQ,PXINQ
0003          INTEGER*2 PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,TEMP,OUTBF,INBF
0005          LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
0006          INTEGER*2 LTIME,IWRTTM
0007          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1          LIDFD(256),IOFLG,LLFLG,IRSEND
0009          COMMON         PACK(256,30)
0010          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          IF(L .LT. 1) GOTO 30
0014          IF(L .GT. 256) GOTO 30
0016          LI=L-5
0017          II=INBF(6)
0018     10   IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0020          T1=IGETSP(N)
0021          DO 15 I=1,LI
0022     15   PACK(I,T1)=INBF(I)
0023          CALL ENSTR(PACK(254,T1),LI)
0024          CALL ENQUE(PINQ,INQ,T1)
0025          N=2
0026          GOTO 40
0027     20   N=1
0028          GOTO 40
0029     30   N=0
0030     40   DO 50 I=1,2
0031            OUTBF(I)=INBF(I)
0032     50   CONTINUE
0033          NN=1
0034          IF(N .EQ. 0) NN=128
0036          OUTBF(3)=NN
0037          OUTBF(4)=0
0038          OUTBF(5)=INBF(6)
0039          OUTBF(6)=ISLID
0040          OUTBF(7)=ETX
0041          OUTBF(8)=0
0042          OUTBF(9)=LIDFD(INBF(6))
0043          CALL LPOUT(9)
0044          IF(N .NE. 2) GOTO 999
0046          ACKSEQ(II)=INBF(2)
0047    999   RETURN
0048          END
```

```
0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON /DFM/ OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          CC1=INBF(3)
0009          CC2=INBF(4)
0010          IF(CC1 .LT. 0) GOTO 25
0012          IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0014          CC3=MOD(CC1,2)
0015          CC4=MOD(CC1,256)
0016          IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0018          IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0020          CC5=MOD(INBF(3),64)
0021          IF(CC5 .GE. 32) CALL INPTQ(LI)
0023          GOTO 99
0024     20   CC6=MOD(CC1,64)
0025          IF(CC6 .GE. 32) IALTRT=1
0027          CC7=MOD(CC1,2)
0028          IF(CC7 .GE. 1) GOTO 30
0030     25   CALL ACKNAK(0)
0031          GOTO 99
0032     30   CALL ACKNAK(1)
0033          GOTO 99
0034     40   CALL INPTQ(LI)
0035     99   CONTINUE
0036          RETURN
0037          END
```

```
0001          SUBROUTINE INIT(IDDM)
0002          INTEGER*2 ST,DEVNUM,FTYPE,RECNUM
0003          INTEGER*2 XINQ,PXINQ,PINQ,FREE
0004          INTEGER*2 STAT,FLWCNT,OUTFCT
0005          INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0006          LOGICAL*1 ICODE,ICON,DUM,IBUFF,IBUF
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0008          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID
0009          LOGICAL*1 F7,F8,F9,F10,F11,F12
0010          INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0011          COMMON /DFM/ OUTBF(256),INBF(256),
             1     IWRTTM,IWRT,STAT,LTIME
0012          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0013          COMMON /MESS/ MESSEQ
0014          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0015          COMMON /FRE/  FREE(30),IFR,IFRSZ
0016          COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0017          COMMON        PACK(256,30)
0018          COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0019          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0020          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0021          COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             &     KEYREC,IBUF(80)
0022          COMMON /M3000/ IFUNT,ITYPE,INODE,IMEAS
0023          COMMON /M4000/ FTYPE,RECNUM,F7,F8,F9,F10,F11,F12
0024          COMMON /M6000/ IDISP,IEQUIP,ICHAN,INUM,IBUFF(20),IFLAG,ICNT
      C
0025          CALL RAM(0,5,4)
0026          CALL RAM(0,255,0)
0027          MESSEQ=0
0028          IABORT=0
0029          IFLAG=0
0030          IOFLG=0
0031          IRSEND=0
0032          LLFLG=0
0033          IFRSZ=30
0034          INFLCT=0
0035          ISLID=24
0036          CR='015
0037          LF='012
0038          ETX='003
0039          RESNLM=2
0040          IQLNTH=15
0041          IALTRT=0
0042          FLWCNT=0
0043          IWRTTM=0
0044          TIMLIM=50
0045          SETPRM=4
0046          RSTPRM=32
0047          SETBKP=8
0048          RSTBKP=64
0049          ATIMLM=200
```

```
0050          MONTOR=25
0051          IWRT=0
0052          OUTFCT=0
0053          IFULL=0
0054          ISENT=0
0055          DO 40 I=1,20
0056       40 LIDFD(I)=4
0057          LIDFD(21)=1
0058          LIDFD(22)=3
0059          LIDFD(23)=6
0060          LIDFD(24)=5
0061          LIDFD(25)=7
0062          LIDFD(26)=8
0063          LIDFD(27)=9
0064          LIDFD(28)=2
0065          DO 50 I=29,39
0066       50 LIDFD(I)=0
0067          DO 60 I=40,44
0068       60 LIDFD(I)=1
0069          DO 70 I=45,59
0070       70 LIDFD(I)=0
0071          DO 80 I=60,64
0072       80 LIDFD(I)=2
0073          DO 90 I=65,79
0074       90 LIDFD(I)=0
0075          DO 100 I=80,84
0076      100 LIDFD(I)=4
0077          DO 110 I=85,99
0078      110 LIDFD(I)=0
0079          DO 120 I=100,104
0080      120 LIDFD(I)=5
0081          DO 130 I=105,256
0082      130 LIDFD(I)=0
0083          PINQ(1)=IQLNTH
0084          PINQ(2)=IQLNTH+1
0085          PXINQ(1)=IQLNTH
0086          PXINQ(2)=IQLNTH+1
0087          DO 10 I=1,IFRSZ
0088            PACK(255,I)=0
0089       10 PACK(256,I)=0
0090          DO 20 I=1,IFRSZ
0091       20 FREE(I)=I
0092          DO 30 I=1,256
0093       30 ACKSEQ(I)=256
0094          IFR=IFRSZ
0095          IF(IDDM .EQ. 1) RETURN
            D CALL ASSIGN(1,'TT:')
0097          CALL ASSIGN(2,'STATUS.DAT')
0098          DEFINE FILE 2(1016,10,U,J6)
0099          READ(2'1)(ICODE(I),I=1,20)
0100          CALL ASSIGN(3,'CDIR.DAT')
0101          DEFINE FILE 3(10,40,U,J1)
0102          READ(3'1)(ICODE(I),I=1,1,80)
0103          CALL ASSIGN(4,'TDIR.DAT')
```

```
0104          DEFINE FILE 4(10,40,U,J2)
0105          READ(4'1)(ICODE(I),I=1,80)
0106          CALL ASSIGN(5,'CKTD.DAT')
0107          DEFINE FILE 5(100,40,U,J3)
0108          READ(5'1)(ICODE(I),I=1,80)
0109          CALL ASSIGN(6,'TNKD.DAT')
0110          DEFINE FILE 6(100,40,U,J4)
0111          READ(6'1)(ICODE(I),I=1,80)
0112          CALL ASSIGN(7,'REPORT.DAT')
0113          DEFINE FILE 7(10,40,U,J5)
0114          READ(7'1)(ICODE(I),I=1,80)
0115          CALL ASSIGN(8,'MSG.DAT')
0116          DEFINE FILE 8(400,40,U,NRCNO)
0117          RETURN
0118          END
```

```
0001          SUBROUTINE LINLOS
0002          REAL*8 RM1(5),LIN08,LIN18
0003          INTEGER*2 XINQ,PXINQ,PINQ
0004          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,XOUTBF,XINBF,DUM
0006          LOGICAL*1 PACK,M1(40),LIN0(8),LIN1(8),ACKSEQ,LIDFD
0007          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0008          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009          COMMON         PACK(256,30)
0010          COMMON /MESS/ MESSEQ
0011          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &            IOFLG,LLFLG,IRSEND
0012          DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0013          DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0014          DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0015          DATA RM1(5)/'DE 24   '/
0016          EQUIVALENCE(LIN0,LIN08)
0017          EQUIVALENCE(LIN1,LIN18)
0018          EQUIVALENCE(M1,RM1)
0019          IS=LLFLG
0020          CALL ENABLE(0)
0021          T1=IGETSP(N)
0022          CALL ENABLE(1)
0023          PACK(1,T1)=0
0024          IF(MESSEQ .EQ. 126) MESSEQ=0
0026          MESSEQ=MESSEQ+1
0027          PACK(2,T1)=MESSEQ
0028          PACK(3,T1)=0
0029          PACK(4,T1)=0
0030          PACK(5,T1)=25
0031          PACK(6,T1)=24
0032          DO 20 I=7,9
0033            PACK(I,T1)=LF
0034      20  CONTINUE
0035          DO 30 I=1,22
0036      30  PACK(I+9,T1)=M1(I)
0037          IF(IS .EQ. 1) GOTO 50
0039          DO 40 I=1,8
0040      40  PACK(I+31,T1)=LIN0(I)
0041          GOTO 70
0042      50  DO 60 I=1,8
0043      60  PACK(I+31,T1)=LIN1(I)
0044      70  DO 80 I=23,40
0045      80  PACK(I+17,T1)=M1(I)
0046          PACK(58,T1)=CR
0047          PACK(59,T1)=LF
0048          PACK(60,T1)=ETX
0049          CALL ENSTR(PACK(254,T1),60)
0050          CALL ENABLE(0)
0051          CALL ENQUE(PXINQ,XINQ,T1)
0052          CALL ENABLE(1)
0053          LLFLG=0
0054          RETURN
```

```
0055        END
```

```
                                        .TITLE  FDM.MACRO
                                        .IDENT  /V3.0/
                                        .GLOBL  LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
                                        .GLOBL  WTOKEN,RAM,STATUS,STATBO
                                        .GLOBL  LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
                                        .NLIST  CND
000000                                  .PSECT

                000000                  R0      =%0
                000001                  R1      =%1
                000002                  R2      =%2
                000003                  R3      =%3
                000004                  R4      =%4
                000005                  R5      =%5
                000006                  SP      =%6
                000007                  PC      =%7

                                        ;INTERFACE ADDRESS

                172410                  BAR=    172410
                172412                  WCR=    172412
                172414                  CSR=    172414
                172416                  IOBUF=  172416
                172416                  OPREG=  172416
                177560                  HRCSR=  177560
                177562                  HRBUF=  177562
                177564                  HXCSR=  177564
                177566                  HXBUF=  177566

                                        ;INTERFACE OPTIONS

                000001                  XCIO=1
                000001                  XSIO=1
                000001                  XGIO=1

                                        ;LOCAL VARIBLES

                000002                  ARG1=   2
                000004                  ARG2=   4
                000006                  ARG3=   6
                000010                  ARG4=   10

                100000                  BIT15=  100000
                040000                  BIT14=  40000
                000400                  BIT08=  400
                000200                  BIT07=  200
                000100                  BIT06=  100
                000040                  BIT05=  40
                000020                  BIT04=  20
                000010                  BIT03=  10
                000004                  BIT02=  4
                000002                  BIT01=  2
                000001                  BIT00=  1
```

```
                                ;*************************************************************
                                ;**** PROCEDURE MASTER START UP                              *
                                ;*************************************************************

000000  017500  000002          MASTER: MOV     @ARG1(R5), R0                   ;START OR RESTORE
000004  022700  000001                  CMP     #1,     R0                      ;RESTORE?
000010  001432                          BEQ     RSTVEC                          ;YES JUMP OVER START
000012  012700  000340                  MOV     #340,   R0                      ;PRI=7
000016  106400                          MTPS    R0
000020  012700  000124                  MOV     #124,   R0
000024  012720  000444'                 MOV     #LIO,   (R0)+                   ;LIU HANDLER
000030  012710  000340                  MOV     #340,   (R0)                    ;PRI=7
000034  012700  000100                  MOV     #100,   R0                      ;CLOCK VECTOR
000040  011067  000044                  MOV     (R0),   VECSAV                  ;SAVE IT FOR RECOVERY
000044  012710  001670'                 MOV     #TIME,  (R0)                    ;CLOCK
000050  042737  040100  172414         BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000056  004767  000030                  JSR     PC,     LIUINT                  ;CLEAR LIU
000062  005037  172414                  CLR     @#CSR                           ;CLEAR BLIUI CSR
000066  012700  000000                  MOV     #000,   R0                      ;PRI=0
000072  106400                          MTPS    R0                              ;CPU DOWN
000074  000207                          RTS     PC
000076  012700  000100          RSTVEC: MOV     #100,   R0                      ;VECTOR SPOT
000102  016710  000002                  MOV     VECSAV, (R0)                    ;RESTORE IT
000106  000207                          RTS     PC
000110  000000                  VECSAV: .WORD   0                               ;VECTOR MAILBOX.


                                ;*************************************************************
                                ;**** PROCEDURE INITIALIZE                                  *
                                ;*************************************************************

                                        ;CLEAR ACRAM

000112  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000120  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000126  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000132  100375                          BPL     .-4                             ;NO RETRY
000134  105037  172414                  CLRB    @#CSR
000140  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
000146  012700  177400                  MOV     #-256., R0                      ;COUNTER
000152  012737  004407  172416  1$:     MOV     #2311., @#OPREG                 ;WRITE A NULL
000160  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000164  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
000166  105037  172414                  CLRB    @#CSR
000172  005200                          INC     R0                              ;COUNT+1
000174  001366                          BNE     1$

                                        ;CLEAR INPUT/OUTPUT BUFFERS

000176  012701  010410                  MOV     #10410, R1                      ;RDBUFADR CMD
000202  012700  010440                  MOV     #10440, R0                      ;SEL INBUF0
000206  004767  000106                  JSR     PC,     ZEROBF                  ;SET POINTER=0
000212  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000220  105737  172414                  TSTB    @#CSR                           ;GOOD READ
000224  100375                          BPL     .-4                             ;NO RETRY
```

```
 76 000226  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
 77 000232  012701  010610                  MOV     #10610, R1              ;RDBUFADR COMMAND
 78 000236  012700  010640                  MOV     #10640, R0              ;SEL INBUF1 COMMAND
 79 000242  004767  000052                  JSR     PC,     ZEROBP          ;POINTER=0
 80 000246  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
 81 000254  105737  172414                  TSTB    @#CSR                   ;GOOD RD
 82 000260  100375                          BPL     .-4                     ;NO RETRY
 83 000262  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
 84 000266  012701  010510                  MOV     #10510, R1              ;RDBUFADR COMMAND
 85 000272  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
 86 000276  004767  000016                  JSR     PC,     ZEROBP
 87 000302  012701  010710                  MOV     #10710, R1              ;RDBUFADR COMMAND
 88 000306  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
 89 000312  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
 90 000316  000440                          BR      STAT$                   ;GO CLEAR STATUS
 91 000320  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
 92 000324  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
 93 000332  105737  172414                  TSTB    @#CSR                   ;GOOD RD
 94 000336  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
 95 000340  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
 96 000344  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
 97 000350  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
 98 000354  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
 99 000360  022702  000000          2$:     CMP     #0,     R2              ;POINTER=0?
100 000364  001412                          BEQ     3$                      ;YES RECHECK
101 000366  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
102 000374  105737  172414                  TSTB    @#CSR                   ;GOOD RD
103 000400  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
104 000402  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
105 000406  005302                          DEC     R2                      ;LOCAL POINTER-1
106 000410  000763                          BR      2$                      ;BR UNTIL 0
107 000412  010037  172416          3$:     MOV     R0,     @#OPREG         ;SEL BUFFER
108 000416  000207                          RTS     PC                      ;RETURN
109
110                                         ;READ AND CLEAR STATUS
111
112 000420  012737  010400  172416  STAT$:  MOV     #4352., @#OPREG         ;WCR:RS
113 000426  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
114 000434  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
115 000442  000207                          RTS     PC
116
117
```

```
                                ;*************************************************************
                                ;**** PROCEDURE LIU-HANDLER                                 *
                                ;*************************************************************


000444  010046                  LIO:    MOV     R0,     -(SP)                   ;SAVE R0
000446  010146                          MOV     R1,     -(SP)                   ;SAVE R1
000450  010246                          MOV     R2,     -(SP)                   ;SAVE R2
000452  010346                          MOV     R3,     -(SP)                   ;SAVE R3
000454  010446                          MOV     R4,     -(SP)                   ;SAVE R4
000456  010546                          MOV     R5,     -(SP)                   ;SAVE R5
000460  005737  172414                  TST     @#CSR                           ;? LIU INTERRUPTED
000464  100153                          BPL     RTI$                            ;NO
000466  042737  040000  172414          BIC     #BIT14, @#CSR                   ;YES/DISABLE LIU
000474  012737  010400  172416          MOV     #4352., @#OPREG                 ;WCR ; RS(0)
000502  012737  002400  172416          MOV     #1280., @#OPREG                 ;RD
000510  013767  172416  000502          MOV     @#IOBUF, CAUSE
000516  042767  177400  000474          BIC     #177400, CAUSE                  ;CLEAR UNUSED BITS

                                        ;PREFORM OPERATION DEPENDING ON BITS

000524  132767  000004  000466  B2$:    BITB    #BIT02, CAUSE                   ;OV-FL
000532  001406                          BEQ     B3$
000534  012702  010410                  MOV     #4360., R2                      ;RDBUFADR IN0
000540  012703  010440                  MOV     #4384., R3                      ;SEL INBUF0
000544  004767  000270                  JSR     PC,     EMBF                    ;GO EMPTY
000550  132767  000010  000442  B3$:    BITB    #BIT03, CAUSE                   ;OV-FL
000556  001406                          BEQ     B4$
000560  012702  010610                  MOV     #4488., R2                      ;RDBUFADR IN1
000564  012703  010640                  MOV     #4512., R3                      ;SEL INBUF1
000570  004767  000244                  JSR     PC,     EMBF                    ;GO EMPTY
000574  132767  000020  000416  B4$:    BITB    #BIT04, CAUSE                   ;LINE-LOSS PRIMARY
000602  001407                          BEQ     B5$
000604  132767  000200  000406  B7$:    BITB    #BIT07, CAUSE                   ;LATCH ON
000612  001403                          BEQ     B5$                             ;YES LOOP AROUND
000614  012767  000002  002002'         MOV     #2,     LLFLG                   ;FLAG PRIMARY SW. FAIL
000622  132767  000040  000370  B5$:    BITB    #BIT05, CAUSE                   ;BACKUP LINE LOSS
000630  001403                          BEQ     B6$
000632  012767  000001  002002'         MOV     #1,     LLFLG                   ;FLAG BACKUP SW. FAIL
000640  132767  000100  000352  B6$:    BITB    #BIT06, CAUSE                   ;WRITE TOKEN DETECT
000646  001404                          BEQ     B0$
000650  005067  001000'                  CLR     IWRTTM                          ;CLEAR TOKEN VAR
000654  005067  001002'                  CLR     IWRT

                                        ;BUFFER FULL

000660  132767  000001  000332  B0$:    BITB    #BIT00, CAUSE                   ;INBUF0 FULL
000666  001423                          BEQ     B1$                             ;NO
000670  012701  000400'                  MOV     #INBF,  R1                      ;BUFFER ADDRESS
000674  012702  010410                  MOV     #4360., R2                      ;RDBUFADR COMM
000700  012703  010440                  MOV     #4384., R3                      ;SEL BUF
000704  012704  000001                  MOV     #BIT00, R4                      ;CRC BIT
000710  004767  000124                  JSR     PC,     EMBF                    ;EMPTY BUFFER
000714  012705  001222'                  MOV     #AREA,  R5                      ;DATA LINK AREA
000720  012767  000001  000274          MOV     #1,     AREA                    ;1 VARIBLE
000726  010267  000274                  MOV     R2,     DATA                    ;CRC OR BYTE COUNT
000732  004767  000000G                 JSR     PC,     LPINPT                  ;CALL FORTRAN QUE'ER
```

```
 58 000736  132767  000002  000254  B1$:    BITB    #BIT01, CAUSE           ;INBUF1 FULL
 59 000744  001423                   BEQ     RTI$                    ;NO WERE DONE
 60 000746  012701  000400'           MOV     #INBF,  R1              ;BUFFER ADDRESS
 61 000752  012702  010610           MOV     #4488.,  R2             ;RDBUFADR COMM
 62 000756  012703  010640           MOV     #4512.,  R3             ;SEL INBUF1
 63 000762  012704  000002           MOV     #BIT01, R4              ;CRC BIT
 64 000766  004767  000046           JSR     PC,     EMBF            ;GO EMPTY BUFFER
 65 000772  012705  001222'          MOV     #AREA,  R5              ;DATA LINK AREA
 66 000776  012767  000001  000216   MOV     #1,     AREA            ;ONE VARIBLE
 67 001004  010267  000216           MOV     R2,     DATA            ;CRC OR BYTE COUNT
 68 001010  004767  000000G          JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
 69
 70 001014  012605                   RTI$:   MOV     (SP)+,  R5              ;RESTORE REGISTERS
 71 001016  012604                   MOV     (SP)+,  R4
 72 001020  012603                   MOV     (SP)+,  R3
 73 001022  012602                   MOV     (SP)+,  R2
 74 001024  012601                   MOV     (SP)+,  R1
 75 001026  012600                   MOV     (SP)+,  R0
 76 001030  052737  040000  172414   BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
 77 001036  000002                   RTI                             ;RETURN FROM INTERRUPT
 78
 79 001040  012737  010600  172416   EMBF:   MOV     #4480.,  @#OPREG        ;READ STATUS 1
 80 001046  012737  002400  172416           MOV     #1280.,  @#OPREG        ;RS
 81 001054  013700  172416           MOV     @#IOBUF, R0             ;FETCH STATUS
 82 001060  130400                   BITB    R4,R0                   ;GOOD CRC
 83 001062  001002                   BNE     CRCOK
 84 001064  012704  177777           MOV     #-1,    R4              ;NO FLAG
 85 001070  010237  172416           CRCOK:  MOV     R2,     @#OPREG         ;RDBUFADR
 86 001074  012737  001400  172416   MOV     #768.,  @#OPREG         ;RD
 87 001102  105737  172414           TSTB    @#CSR                   ;GOOD RD
 88 001106  100375                   BPL     .-4
 89 001110  013702  172416           MOV     @#IOBUF, R2             ;REPLACE WITH POINTER
 90 001114  042702  177400           BIC     #177400, R2             ;CLEAR MST BITS
 91 001120  010200                   MOV     R2,     R0              ;SAVE IT
 92 001122  005400                   NEG     R0                      ;2'S COMP
 93 001124  010037  172412           MOV     R0,     @#WCR           ;BYTE COUNT NOW
 94 001130  010137  172410           MOV     R1,     @#BAR           ;ADDRESS IN MEMORY
 95 001134  010337  172416           MOV     R3,     @#OPREG         ;SEL BUFFER
 96 001140  012737  001400  172416   MOV     #768.,  @#OPREG         ;FALSE RD
 97 001146  105737  172414           TSTB    @#CSR                   ;DONE ON
 98 001152  100375                   BPL     .-4                     ;NO LOOP
 99 001154  012737  021000  172416   MOV     #8704.,  @#OPREG        ;FIRE DMA
100 001162  000240                   NOP                             ;DELAY
101 001164  105737  172414           TSTB    @#CSR                   ;GOOR DMA
102 001170  100401                   BMI     DMAOK
103 001172  000240                   NOP                             ;ERROR IF HERE
104 001174  012737  004400  172416   DMAOK:  MOV     #2304.,  @#OPREG        ;FALSE WD
105 001202  105737  172414           TSTB    @#CSR                   ;GOOD WD
106 001206  100375                   BPL     .-4
107 001210  005704                   TST     R4                      ;WAS CRC OK
108 001212  100001                   BPL     END$                    ;YES
109 001214  010402                   MOV     R4,     R2              ;NO FLAG IT
110 001216  000207                   END$:   RTS     PC                      ;RETURN
111 001220  000000                   CAUSE:  .WORD   0                       ;STATUS BYTE 0 HOLDER
112 001222  000000                   AREA:   .WORD   0
113 001224  001226'                  .WORD   DATA
114 001226  000000                   DATA:   .WORD   0
```

```
                                        .ENABLE LSB
                                ;*************************************************************
                                ;**** PROCEDURE ENABLE                                      *
                                ;*************************************************************

001230  017501  000002          ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
001234  022701  000001                  CMP     #1,     R1
001240  001404                          BEQ     E1$
001242  042737  040000  172414  E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
001250  000207                          RTS     PC
001252  052737  040000  172414  E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
001260  000207                          RTS     PC

                                        ;FORTRAN ENCODE/DECODE MACRO

001262  017575  000004  000002  ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
001270  000207                          RTS     PC
001272  017575  000002  000004  DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECODE BYTE COUNT
001300  000207                          RTS     PC


                                ;*************************************************************
                                ;**** PROCEDURE LINE SWITCH                                 *
                                ;*************************************************************

                                        ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                        ;
                                        ;04=SET PRIMARY LINE
                                        ;32=RESET PRIMARY LINR
                                        ;08=SET BACKUP LINE
                                        ;64=RESET BACKUP LINE

001302  017501  000002          SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
001306  012737  010420  172416          MOV     #4368.,@#OPREG                  ;WCR : MODSTAT
001314  062701  004400                  ADD     #2304.,R1
001320  010137  172416                  MOV     R1,     @#OPREG                 ;RD
001324  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001330  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001332  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001336  000207                          RTS     PC
```

```
                                ;*************************************************************
                                ;**** PROCEDURE STATUS                                      *
                                ;*************************************************************

                                        ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
                                        ;
                                        ;X=0 STATUS BYTE 0
                                        ;X=1 STATUS BYTE 1

001340  017501  000002          STATUS: MOV     @ARG1(R5), R1                   ;WHICH BYTE
001344  022701  000001                  CMP     #1,     R1                      ;COMPARE
001350  001415                          BEQ     2$
001352  012737  010400  172416  1$:     MOV     #4352.,@#OPREG                  ;WCR : RS(0)
001360  012737  002400  172416          MOV     #1280.,@#OPREG                  ;RS
001366  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001372  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001376  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001402  000207                          RTS     PC
001404  012737  010600  172416  2$:     MOV     #4480.,@#OPREG                  ;WCR : RS(1)
001412  012737  002400  172416          MOV     #1280.,@#OPREG                  ;RS
001420  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001424  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001430  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001434  000207                          RTS     PC
001436  012737  010400  172416  STATB0: MOV     #4352.,@#OPREG                  ;STATUS BYTE 1
001444  012737  002400  172416          MOV     #1280.,@#OPREG                  ;RS
001452  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001456  042701  177400                  BIC     #177400, R1
001462  132701  000010                  BITB    #BIT03, R1                      ;BUFFER FULL
001466  001404                          BEQ     3$
001470  012775  000001  000002          MOV     #1,     @ARG1(R5)               ;LOAD RETURN
001476  000207                          RTS     PC
001500  012775  000000  000002  3$:     MOV     #0,     @ARG1(R5)               ;LOAD RETURN
001506  000207                          RTS     PC
```

```
                                ;*********************************************************
                                ;**** PROCEDURE WRITE TOKEN (WTOKEN)                     *
                                ;*********************************************************
                                
                                        ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP
                                
001510  012737  010600  172416  WTOKEN: MOV     #4480., @#OPREG                 ;READ STATUS(1)
001516  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001524  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001530  132701  000010                  BITB    #BIT03, R1                      ;BUFFER LOADED
001534  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001536  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001544  012737  004423  172416          MOV     #2323., @#OPREG                 ;MARK OBOF/OB1F/WTCMD
001552  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001556  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001560  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001564  000207                          RTS     PC
001566  012737  010740  172416  WTKG:   MOV     #4576., @#OPREG                 ;SELECT BUFFER 1
001574  012737  001666' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001602  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001606  005402                          NEG     R2                              ;FORMAT IT
001610  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001614  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001622  000240                          NOP                                     ;DELAY
001624  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001630  000240                          NOP                                     ;ERROR IF HERE
001632  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001636  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001644  012737  004422  172416          MOV     #2322., @#OPREG                 ;MARK OB1F/WTCMD
001652  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001656  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001660  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001664  000207                          RTS     PC                              ;RETURN
001666     000     377          WTD:    .BYTE 0,377                             ;A WTOKEN
                                
                                
                                ;*********************************************************
                                ;**** PROCEDURE TIMER (TIME)                             *
                                ;*********************************************************
                                
001670  005267  001006'          TIME:   INC     LTIME                           ;TIME+1
001674  022767  077777  001006'          CMP     #077777, LTIME                  ;?TIME FULL
001702  001002                          BNE     4$
001704  005067  001006'                  CLR     LTIME                           ;CLEAR CLOCK
001710  000002                  4$:     RTI
                                
                                
                                
```

```
                        ;*************************************************************
                        ;**** PROCEDURE ACRAM                                       *
                        ;*************************************************************

                                ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
                                ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
                                ;
                                ;0110   =NREAD
                                ;0100   =DREAD
                                ;0111   =NULL
                                ;0000   =WTOKEN

001712  017500  000002         RAM:    MOV     @ARG1(R5), R0           ; RAM
001716  022700  000000                 CMP     #0,     R0              ;WHICH OPERATION
001722  001035                         BNE     RDRAM                   ;READ OP
001724  017500  000004         WTRAM:  MOV     @ARG2(R5), R0           ;ADDRESS
001730  017501  000006                 MOV     @ARG3(R5), R1           ;WRITE DATA
001734  012737  010402  172416         MOV     #4354., @#OPREG         ;SEL LDACTR
001742  062700  004400                 ADD     #2304., R0              ;WD/DATA (ADDR)
001746  010037  172416                 MOV     R0,     @#OPREG         ;WRITE DATA
001752  105737  172414                 TSTB    @#CSR                   ;VALID WRITE
001756  100375                         BPL     .-4                     ;NO LOOP UNTIL READY
001760  105037  172414                 CLRB    @#CSR                   ;CLEAR DONE BIT
001764  012737  010401  172416         MOV     #4353., @#OPREG         ;SEL ACRAM
001772  062701  004400                 ADD     #2304., R1              ;WD/DATA (CMD)
001776  010137  172416                 MOV     R1,     @#OPREG         ;WRITE
002002  105737  172414                 TSTB    @#CSR                   ;VALID WRITE
002006  100375                         BPL     .-4                     ;NO LOOP UNTIL READY
002010  105037  172414                 CLRB    @#CSR                   ;CLEAR DONE BIT
002014  000207                         RTS     PC
002016  017500  000004         RDRAM:  MOV     @ARG2(R5), R0           ;ADDRESS TO READ
002022  012737  010402  172416         MOV     #4354., @#OPREG         ;SEL LDACTR
002030  062700  004400                 ADD     #2304., R0              ;WD/DATA (ADDR)
002034  010037  172416                 MOV     R0,     @#OPREG         ;WRITE DATA
002040  105737  172414                 TSTB    @#CSR                   ;VALID WRITE
002044  100375                         BPL     .-4                     ;NO LOOP UNTIL READY
002046  105037  172414                 CLRB    @#CSR                   ;CLEAR DONE BIT
002052  012737  010401  172416         MOV     #4353., @#OPREG         ;SEL ACRAM
002060  012737  001400  172416         MOV     #768.,  @#OPREG         ;READ DATA
002066  105737  172414                 TSTB    @#CSR                   ;VALID READ
002072  100375                         BPL     .-4                     ;NO LOOP UNTIL READY
002074  105037  172414                 CLRB    @#CSR                   ;CLEAR DONE BIT
002100  013700  172416                 MOV     @#IOBUF, R0             ;FETCH DATA
002104  042700  177760                 BIC     #177760, R0             ;CLEAR BITS
002110  010075  000006                 MOV     R0, @ARG3(R5)           ;RTN DATA
002114  000207                         RTS     PC
```

```
 1                                      .ENABLE LSB
 2                                ;*********************************************************
 3                                ;**** PROCEDURE WRITE LOOP (LPOUT)                       *
 4                                ;*********************************************************
 5
 6 002116  017502  000002        LPOUT:  MOV     @ARG1(R5), R2                   ;BYTE COUNT
 7 002122  012700  000340                MOV     #340,   R0                      ;PRI=7
 8 002126  106400                        MTPS    R0
 9 002130  012737  000000' 172410        MOV     #OUTBF, @#BAR                   ;BUS ADDRESS
10 002136  005402                        NEG     R2                              ; 2'S COMP COUNT
11 002140  010237  172412                MOV     R2,     @#WCR                   ;COUNT
12 002144  012737  010540  172416        MOV     #4448., @#OPREG                 ;OBO COMMAND
13 002152  012737  024000  172416        MOV     #10240., @#OPREG                ;DMA GO
14 002160  000240                        NOP                                     ;INTERFACE TIME
15 002162  105737  172414                TSTB    @#CSR                           ;DMA OK
16 002166  000240                        NOP                                     ;ERROR IF HERE
17 002170  105037  172414                CLRB    @#CSR
18 002174  012737  010740  172416        MOV     #4576., @#OPREG                 ;OB1 COMMAND
19 002202  012737  001666' 172410        MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
20 002210  012702  000002                MOV     #2,     R2                      ;BYTE COUNT
21 002214  005402                        NEG     R2                              ;FORMAT FOR BLIUI
22 002216  010237  172412                MOV     R2,     @#WCR                   ;DMA BYTE COUNT
23 002222  012737  024000  172416        MOV     #10240.,@#OPREG                 ;FIRE DMA
24 002230  000240                        NOP                                     ;DELAY
25 002232  105737  172414                TSTB    @#CSR                           ;GOOD DMA
26 002236  000240                        NOP                                     ;ERROR IF NOT
27 002240  105037  172414                CLRB    @#CSR                           ;CLEAR DONE
28 002244  012737  010420  172416        MOV     #4368., @#OPREG                 ;MODSTAT
29 002252  012737  004403  172416        MOV     #2307., @#OPREG                 ;BUFFERS FULL
30 002260  105737  172414                TSTB    @#CSR                           ;GOOD WRITE
31 002264  100375                        BPL     .-4                             ;NO LOOP UNTIL
32 002266  105037  172414                CLRB    @#CSR                           ;CLEAR DONE BIT
33 002272  012700  000000                MOV     #000,   R0                      ;PRI=0
34 002276  106400                        MTPS    R0                              ;LOWER CPU
35 002300  000207                        RTS     PC
36
37
38
```

```
                                ;*******************************************************
                                ;**** PROCEDURE READ BUFFER POINTER                    *
                                ;*******************************************************

002302  017500  000002          RDPNT:  MOV     @ARG1(R5), R0           ;FETCH COMMAND
002306  022700  000000                  CMP     #0,     R0              ;INBUF0 ?
002312  001004                          BNE     1$                      ;NO
002314  012700  010410                  MOV     #4360., R0              ;RDBUFADR IN0
002320  000167  000046                  JMP     4$
002324  022700  000001          1$:     CMP     #1,     R0              ;INBUF1 ?
002330  001004                          BNE     2$                      ;NO
002332  012700  010610                  MOV     #4488., R0              ;RDBUFADR IN1
002336  000167  000030                  JMP     4$
002342  022700  000002          2$:     CMP     #2,     R0              ;OUTBUF0 ?
002346  001004                          BNE     3$                      ;NO
002350  012700  010510                  MOV     #4424., R0              ;RDBUFADR OUT0
002354  000167  000012                  JMP     4$
002360  022700  000004          3$:     CMP     #4,     R0              ;OUTBUF1 ?
002364  001020                          BNE     5$                      ;NO RETURN
002366  012700  010710                  MOV     #4552., R0              ;RDBUFADR OUT1
002372  010037  172416          4$:     MOV     R0,     @#OPREG         ;RDBUFADR
002376  012737  001400  172416          MOV     #768.,  @#OPREG         ;RD
002404  105737  172414                  TSTB    @#CSR                   ;READY
002410  100375                          BPL     .-4                     ;NO LOOP UNTIL
002412  013700  172416                  MOV     @#IOBUF, R0             ;FETCH POINTER
002416  042700  177400                  BIC     #177400, R0             ;CLEAR LSB
002422  010075  000004                  MOV     R0,     @ARG2(R5)       ;RETURN POINTER
002426  000207          5$:     RTS     PC
```

```
 1                              ;*************************************************************
 2                              ;**** PROCEDURE RESTART                                      *
 3                              ;*************************************************************
 4
 5                                      ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
 6                                      ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
 7                                      ;CALL RSTART(2) -LOAD MODE(173000)
 8
 9 002430  017500  000002      RSTART: MOV     @ARG1(R5), R0                   ;MODE
10 002434  022700  000000      RS0:    CMP     #0,     R0                      ;LOAD ADDRESS
11 002440  001002                      BNE     RS1
12 002442  000000                      HALT
13 002444  000207                      RTS     PC
14 002446  022700  000001      RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
15 002452  001005                      BNE     RS2
16 002454  004767  175432              JSR     PC,     LIUINT                  ;INT LIU FIRST
17 002460  012700  000040      G4$:    MOV     #40,    R0                      ;START ADDRESS
18 002464  000110                      JMP     (R0)
19 002466  022700  000002      RS2:    CMP     #2,     R0                      ;LOAD MODE
20 002472  001002                      BNE     RST
21 002474  000177  000002              JMP     @ROM
22 002500  000207              RST:    RTS     PC
23 002502  173000              ROM:    .WORD   173000
24
25
26
235
```

```
                        ;*********************************************************
                        ;****        COMMON DATA AREA                             *
                        ;*********************************************************

000000                          .PSECT  DFM,RW,D,GBL,REL,OVR

000000                  OUTBF:  .BLKB   256.
000400                  INBF:   .BLKB   256.
001000                  IWRTTM: .BLKW
001002                  IWRT:   .BLKW
001004                  STAT:   .BLKW
001006                  LTIME:  .BLKW
                                .EVEN

000000                          .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                  XOUTBF: .BLKB   256.
000400                  XINBF:  .BLKB   256.
001000                  ACKSEQ: .BLKB   256.
001400                  LIDFD:  .BLKB   256.
002000                  IOFLG:  .BLKW
002002                  LLFLG:  .BLKW
002004                  IRSEND: .BLKW
                                .EVEN


        000001                  .END
```

```
FDM.MACRO       MACRO V03.01 5-JUN-79 13:44:43 PAGE 11-1
SYMBOL TABLE

ACKSEQ  001000R     003 B0$     000660R         E0$     001242R         LPOUT   002116RG        STATB0  001436RG
AREA    001222R         B1$     000736R         E1$     001252R         LTIME   001006R     002 STATUS  001340RG
ARG1  = 000002          B2$     000524R         G4$     002460R         MASTER  000000RG        STAT$   000420R
ARG2  = 000004          B3$     000550R         HRBUF = 177562          OPREG = 172416          SWITCH  001302RG
ARG3  = 000006          B4$     000574R         HRCSR = 177560          OUTBF   000000R     002 TIME    001670RG
ARG4  = 000010          B5$     000622R         HXBUF = 177566          RAM     001712RG        VECSAV  000110R
BAR   = 172410          B6$     000640R         HXCSR = 177564          RDPNT   002302RG        WCR   = 172412
BIT00 = 000001          B7$     000604R         INBF    000400R     002 RDRAM   002016R         WTD     001666R
BIT01 = 000002          CAUSE   001220R         IOBUF = 172416          ROM     002502R         WTKG    001566R
BIT02 = 000004          CRCOK   001070R         IOFLG   002000R     003 RST     002500R         WTOKEN  001510RG
BIT03 = 000010          CSR   = 172414          IRSEND  002004R     003 RSTART  002430RG        WTRAM   001724R
BIT04 = 000020          DATA    001226R         IWRT    001002R     002 RSTVEC  000076R         XCIO  = 000001
BIT05 = 000040          DESTR   001272RG        IWRTTM  001000R     002 RS0     002434R         XGIO  = 000001
BIT06 = 000100          DMAOK   001174R         LIDFD   001400R     003 RS1     002446R         XINBF   000400R     003
BIT07 = 000200          EMBF    001040R         LIO     000444RG        RS2     002466R         XOUTBF  000000R     003
BIT08 = 000400          ENABLE  001230RG        LIUINT  000112RG        RTI$    001014R         XSIO  = 000001
BIT14 = 040000          END$    001216R         LLFLG   002002R     003 STAT    001004R     002 ZEROBP  000320R
BIT15 = 100000          ENSTR   001262RG        LPINPT= ****** G

. ABS.  000000      000
        002504      001
DFM     001010      002
BUFS    002006      003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:FDM=DK:FDM
```

```
FOR NODAL/UNITS:9/NOLINE
FOR NODAL1/NOLINE
MAC FDM
```

```
decwriter     program:   FNK24.COM                 01-JUL 79        12:00:00        PAGE 001

ASS DX0: DK:
R LINK
NODE24,DX1:MAP=DX1:NODAL/C/I/W
DX1:FDM/C
DX1:NODAL1/C
DX1:M0000/C
DX1:M5000/C
DX1:M1000/0:1/C
DX1:M2000/0:1/C
DX1:M3000/0:1/C
DX1:M4000/0:1/C
DX1:M6000/0:1//
$SHORT

ASS DX1: DK:
```

1.6 NODE 25 (OCRI)

The Operator Control and Report Interface (OCRI) performs the functions of interfacing to the site operator. The User Language interface, when the OCRI terminal is ATTACHED to the DMBS node, allows the operator to control the site, request status information concerning site and equipment performance and prepare site reports which are forwarded to other sites such as the ACOC. Operator-to-operator messages are also supported by the user Language. The OCRI terminal can be used to print event reports, fault reports, and alarms for the simulated equipment generated by the FIAC and SDCA modules, and error messages concerning the FDM itself generated by the nodal modules (e.g., Loop-Back in effect, queue overflow).

1.6.1 Program Descriptions

1.6.1.1 Refer to Section 1.1 for descriptions of routines-
NODAL, IGETSP, ENQUE, DEQUE, ACKNAK, INPTQ, LPINPT, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME, RAM, LPOUT, RDPNT, RSTART.

1.6.1.2 Subroutine HINPUT (FORTRAN)

This subroutine is called by the interrupt handler of the LA-36. This routine is where the ATTACH command is checked and decoded. If the message is not a command it is queued.

1.6.1.3 Subroutine CIO (MACRO)

This subroutine is the interrupt handler for the LA-36 terminal. It reads the ASCII data as it's typed, checks for deletes, control, and xon-xoff characters and stores them until a RETURN is typed. When RETURN is detected subroutine HINPUT is called.

1.6.1.4 Subroutine COUT (MACRO)

This subroutine is used to print a message received from the loop on the LA-36. It's passed one argument the byte count to be printed.

```
0001          PROGRAM NODAL
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003          INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP,IXON
0005          INTEGER*2 RSTBKP,Q1,Q2,RESNUM,OUTQ,DEQUE
0006          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0008          INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0009          COMMON /MESS/ MESSEQ
0010          COMMON/DFM/ OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0011          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0012          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1            LIDFD(256),IOFLG,LLFLG,IRSEND
0013          COMMON /FRE/ FREE(64),IFR,IFRSZ
0014          COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0015          COMMON       PACK(256,64)
0016          COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0017          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1      RESNUM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0018          COMMON/CRT/IXON
0019          IXON=0
0020          CALL MASTER
0021          CALL INIT
0022          CALL ENABLE(1)
0023      5   CONTINUE
0024     25   IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0026          CALL ENABLE(0)
0027          Q2=DEQUE(PINQ,INQ,1)
0028          CALL ENABLE(1)
0029          CALL DESTR(PACK(254,Q2),LEN)
0030          DO 30 I=1,LEN
0031            XOUTBF(I)=PACK(I,Q2)
0032     30   CONTINUE
0033          ITERMX=XOUTBF(6)
0034          IF(ITERMX .EQ. 27 .AND. IXON .EQ. 1) GOTO 31
0036          IF(ITERMX .EQ. 28 .AND. IXON .EQ. 1) GOTO 31
0038          CALL COUT(LEN)
0039     31   IFR=IFR+1
0040          FREE(IFR)=Q2
0041     40   CALL STATBO(IS)
0042          IF(IS .EQ. 1) GOTO 100
0044          IF(IOFLG .GT. 0) CALL HINPUT
0046          IF(LLFLG .GT. 0) CALL LINLOS
0048     80   IF(OUTFCT .EQ. 1) GOTO 100
0050          IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0052          IF(IRSEND .EQ. 1) GOTO 85
0054          IF(PXINQ(1) .LT. PXINQ(2)) GOTO 100
0056          CALL ENABLE(0)
0057          Q1=DEQUE(PXINQ,XINQ,1)
0058          CALL ENABLE(1)
0059          OUTQ=Q1
0060     85   CALL DESTR(PACK(254,OUTQ),Q2)
0061          DO 90 I=1,Q2
```

```
0062          OUTBF(I)=PACK(I,OUTQ)
0063     90   CONTINUE
0064          OUTBF(Q2+1)=0
0065          IPT=OUTBF(5)
0066          OUTBF(Q2+2)=LIDFD(IPT)
0067          IF(Q2+2 .LT. 6 .OR. Q2+2 .GT. 255) GOTO 100
0069          IFULL=1
0070          ISENT=1
0071          CALL LPOUT(Q2+2)
0072          IRSEND=1
0073          ACKTIM=0
0074          IWRTTM=0
0075          IWRT=1
0076    100   INFLCT=0
0077    110   OLDTIM=NEWTIM
0078          NEWTIM=LTIME
0079          IF(IWRT .EQ. 0) GOTO 120
0081          IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0082          IF(IWRTTM .LT. TIMLIM) GOTO 120
0084          CALL WTOKEN
0085          IWRTTM=0
0086          IWRT=0
0087    120   IF(ISENT .EQ. 0) GOTO 130
0089          ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0090          IF(ACKTIM .LT. ATIMLM) GOTO 130
0092          CALL ENABLE(0)
0093          CALL ACKNAK(0)
0094          CALL ENABLE(1)
0095    130   CONTINUE
0096          GOTO 5
0097          END
```

```
FORTRAN IV     V02.1-1    Tue 05-Jun-79 13:30:43              PAGE 001

0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(64),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IF(IFR .LT. 1) CALL INIT
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```



```
FDM.MAC
```

```
FORTRAN IV     V02.1-1    Tue 05-Jun-79 13:30:54              PAGE 001

0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1         ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008       10 IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016       20 IF(IQHEAD .GE. IQLNTH) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020       30 B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024       40 CALL INIT
0025      999 RETURN
0026          END
```


FDM.MAC

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

```
FORTRAN IV     V02.1-1    Tue 05-Jun-79 13:31:08              PAGE 001

0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1         ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016       10 A(1)=IQHEAD-1
0017      999 RETURN
0018          END
```


FDM.MA

```
0001          SUBROUTINE ACKNAK(N)
0002          INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ,PXOUTQ
0003          INTEGER*2 ACKQ,PACKQ,PINQ,FREE,OUTQ,XOUTQ
0004          INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006          LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007          INTEGER*2 LTIME,IWRTTM
0008          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0009          REAL*8 COMT(5)
0010          EQUIVALENCE(COMT,LCOMT)
0011          DATA COMT(1)/'MESSAGE '/,COMT(2)/'NOT SENT'/,COMT(3)/' FROM    '/,
             1    COMT(4)/'NODE    '/,COMT(5)/' TO NODE'/
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IDFLG,LLFLG,IRSEND
0014          COMMON/DFM/OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0015          COMMON        PACK(256,64)
0016          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1          PACKQ(2),INQ(16),PINQ(2)
0017          COMMON /FRE/  FREE(64),IFR,IFRSZ
0018          COMMON /MESS/ MESSEQ
0019          IF (N .NE. 1) GOTO 10
0021          IFR=IFR+1
0022          DO 5 I=1,256
0023            PACK(I,OUTQ)=0
0024       5  CONTINUE
0025          IFULL=0
0026          ISENT=0
0027          FREE(IFR)=OUTQ
0028          IRSEND=0
0029          GOTO 999
0030      10  T2=PACK(256,OUTQ)
0031          T2=T2+1
0032          IF(T2 .LE. RESNLM) GOTO 120
0034          T3=PACK(3,OUTQ)
0035          T4=MOD(T3,64)
0036          IF(T4 .LT. 32) GOTO 110
0038          DO 20 I=7,60
0039            XOUTBF(I)='40
0040      20  CONTINUE
0041          DO 30 I=7,9
0042            XOUTBF(I)=LF
0043      30  CONTINUE
0044          DO 40 I=1,8
0045            XOUTBF(I+9)=LCOMT(I)
0046      40  CONTINUE
0047          DO 50 I=9,16
0048            XOUTBF(I+9)=LCOMT(I)
0049      50  CONTINUE
0050          DO 60 I=17,22
0051            XOUTBF(I+9)=LCOMT(I)
0052      60  CONTINUE
0053          DO 70 I=25,29
```

```
0054          XOUTBF(I+7)=LCOMT(I)
0055     70   CONTINUE
0056          ENCODE(3,80,XOUTBF(38)) ISLID
0057     80   FORMAT(I3)
0058          DO 90 I=33,40
0059            XOUTBF(I+11)=LCOMT(I)
0060     90   CONTINUE
0061          ENCODE(3,80,XOUTBF(52)) MONTOR
0062          XOUTBF(55)=LF
0063          XOUTBF(56)=LF
0064          XOUTBF(57)=CR
0065          XOUTBF(58)=ETX
0066          CALL COUT(58)
0067          DO 102 I=1,256
0068            PACK(I,OUTQ)=0
0069    102   CONTINUE
0070          ISENT=0
0071          IFULL=0
0072          IWRT=0
0073          IFR=IFR+1
0074          FREE(IFR)=OUTQ
0075          IRSEND=0
0076          GOTO 999
0077    110   PACK(3,OUTQ)=PACK(3,OUTQ)+32
0078          PACK(256,OUTQ)=0
0079          ISENT=0
0080          IFULL=0
0081          IF(IRSEND .EQ. 1) GOTO 999
0083          CALL ENQUE(PXINQ,XINQ,OUTQ)
0084          GOTO 999
0085    120   PACK(256,OUTQ)=T2
0086          ISENT=0
0087          IFULL=0
0088          IF(IRSEND .EQ. 1) GOTO 999
0090          CALL ENQUE(PXINQ,XINQ,OUTQ)
0091    999   RETURN
0092          END
```


FDM.MACR

```
 1
 2
 3
 4
 5 C
 6 C
 7 C
 8 C
 9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
```

```
      SUBROUTINE INPTQ(L)
      INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
      INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
      LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
      LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
      INTEGER*2 LTIME,IWRTTM,IXON
      COMMON/DFM/ OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
      COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
     1          LIDFD(256),IOFLG,LLFLG,IRSEND
      COMMON        PACK(256,64)
      COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
     1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
     1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
      COMMON/CRT/IXON
      IF(L .LT. 1) GOTO 30
      IF(L .GT. 256) GOTO 30
      LI=L-2
      II=INBF(6)
      IF(II .EQ. 27 .AND. IXON .EQ. 1) GOTO 20
      IF(II .EQ. 28 .AND. IXON .EQ. 1) GOTO 20
      IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
      T1=IGETSP(N)
      DO 10 I=1,LI
   10 PACK(I,T1)=INBF(I)
      CALL ENSTR(PACK(254,T1),LI)
      CALL ENQUE(PINQ,INQ,T1)
      N=2
      GOTO 40
   20 N=1
      GOTO 40
   30 N=0
   40 DO 50 I=1,2
        OUTBF(I)=INBF(I)
   50 CONTINUE
      NN=1
      IF(N .EQ. 0) NN=128
      OUTBF(3)=NN
      OUTBF(4)=0
      OUTBF(5)=INBF(6)
      OUTBF(6)=ISLID
      OUTBF(7)=ETX
      OUTBF(8)=0
      OUTBF(9)=LIDFD(INBF(6))
      CALL LPOUT(9)
      IF(N .NE. 2) GOTO 999
      ACKSEQ(II)=INBF(2)
  999 RETURN
      END
```

```
0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON/DFM/ OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          CC1=INBF(3)
0009          CC2=INBF(4)
0010          IF(CC1 .LT. 0) GOTO 25
0012          IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0014          CC3=MOD(CC1,2)
0015          CC4=MOD(CC1,256)
0016          IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0018          IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0020          CC5=MOD(INBF(3),64)
0021          IF(CC5 .GE. 32) CALL INPTQ(LI)
0023          GOTO 99
0024      20  CC7=MOD(CC1,2)
0025          IF(CC7 .GE. 1) GOTO 30
0027      25  CALL ACKNAK(0)
0028          GOTO 99
0029      30  CALL ACKNAK(1)
0030          GOTO 99
0031      40  CALL INPTQ(LI)
0032      99  CONTINUE
0033          RETURN
0034          END
```

```
0001          SUBROUTINE INIT
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0003          INTEGER*2 STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0005          LOGICAL*1 DUM,IBF0,IBF1
0006          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID
0008          INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0009          COMMON/DFM/OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1              ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /MESS/ MESSEQ
0012          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1      IOFLG,LLFLG,IRSEND
0013          COMMON /FRE/  FREE(64),IFR,IFRSZ
0014          COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0015          COMMON        PACK(256,64)
0016          COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0017          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0018          CALL RAM(0,7,4)
0019          CALL RAM(0,255,0)
0020          IFRSZ=64
0021          MESSEQ=0
0022          LLFLG=0
0023          INFLCT=0
0024          ISLID=25
0025          IOFLG=0
0026          IRSEND=0
0027          CR='015
0028          LF='012
0029          ETX='003
0030          RESNLM=2
0031          IQLNTH=16
0032          IALTRT=0
0033          FLWCNT=0
0034          IWRTTM=0
0035          TIMLIM=50
0036          SETPRM=4
0037          RSTPRM=32
0038          SETBKP=8
0039          RSTBKP=64
0040          ATIMLM=200
0041          MONTOR=24
0042          IWRT=0
0043          OUTFCT=0
0044          IFULL=0
0045          ISENT=0
0046          DO 40 I=1,20
0047       40 LIDFD(I)=4
0048          LIDFD(21)=1
0049          LIDFD(22)=3
0050          LIDFD(23)=6
0051          LIDFD(24)=5
```

```
0052          LIDFD(25)=7
0053          LIDFD(26)=8
0054          LIDFD(27)=9
0055          LIDFD(28)=2
0056          DO 50 I=29,39
0057       50 LIDFD(I)=0
0058          DO 60 I=40,44
0059       60 LIDFD(I)=1
0060          DO 70 I=45,59
0061       70 LIDFD(I)=0
0062          DO 80 I=60,64
0063       80 LIDFD(I)=2
0064          DO 90 I=65,79
0065       90 LIDFD(I)=0
0066          DO 100 I=80,84
0067      100 LIDFD(I)=4
0068          DO 110 I=85,99
0069      110 LIDFD(I)=0
0070          DO 120 I=100,104
0071      120 LIDFD(I)=5
0072          DO 130 I=105,256
0073      130 LIDFD(I)=0
0074          PINQ(1)=IQLNTH
0075          PINQ(2)=IQLNTH+1
0076          PACKQ(1)=IQLNTH
0077          PACKQ(2)=IQLNTH+1
0078          PXOUTQ(1)=IQLNTH
0079          PXOUTQ(2)=IQLNTH+1
0080          PXINQ(1)=IQLNTH
0081          PXINQ(2)=IQLNTH+1
0082          DO 10 I=1,IFRSZ
0083            PACK(255,I)=0
0084       10 PACK(256,I)=0
0085          DO 20 I=1,IFRSZ
0086       20 FREE(I)=I
0087          DO 30 I=1,256
0088       30 ACKSEQ(I)=256
0089          DO 140 I=1,20
0090      140 ACKSEQ(I)=254
0091          IFR=IFRSZ
0092          RETURN
0093          END
```

```
0001          SUBROUTINE LINLOS
0002          REAL*8 RM1(5),LIN08,LIN18
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,PACKQ,ACKQ,T1,STAT
0004          LOGICAL*1 OUTBF,INBF
0005          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0007          LOGICAL*1 M1(40),LIN0(8),LIN1(8)
0008          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1             LIDFD(256),IOFLG,LLFLG,IRSEND
0010          COMMON /MESS/ MESSEQ
0011          DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0012          DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0013          DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0014          DATA RM1(5)/'DE 25   '/
0015          EQUIVALENCE(LIN0,LIN08)
0016          EQUIVALENCE(LIN1,LIN18)
0017          EQUIVALENCE(M1,RM1)
0018          IS=LLFLG
0019          DO 20 I=7,9
0020            XOUTBF(I)=LF
0021      20  CONTINUE
0022          DO 30 I=1,22
0023      30  XOUTBF(I+9)=M1(I)
0024          IF(IS .EQ. 1) GOTO 50
0026          DO 40 I=1,8
0027      40  XOUTBF(I+31)=LIN0(I)
0028          GOTO 70
0029      50  DO 60 I=1,8
0030      60  XOUTBF(I+31)=LIN1(I)
0031      70  DO 80 I=23,40
0032      80  XOUTBF(I+17)=M1(I)
0033          XOUTBF(58)=CR
0034          XOUTBF(59)=LF
0035          XOUTBF(60)=ETX
0036          CALL COUT(60)
0037          LLFLG=0
0038          RETURN
0039          END
```

```
0001          SUBROUTINE HINPUT
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0004          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID
0005          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,PACK,DUM
0006          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1              LIDFD(256),IOFLG,LLFLG,IRSEND
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0009          COMMON /MESS/ MESSEQ
0010          COMMON        PACK(256,64)
0011          IF ((XINBF(1) .EQ. 'A') .AND.
             1   (XINBF(2) .EQ. 'T') .AND.
             2   (XINBF(3) .EQ. 'T') .AND.
             3   (XINBF(4) .EQ. 'A') .AND.
             4   (XINBF(5) .EQ. 'C') .AND.
             5   (XINBF(6) .EQ. 'H')) GOTO 500
0013          CALL ENABLE(0)
0014          JJ=IGETSP(N)
0015          CALL ENABLE(1)
0016          DO 10 I=1,IOFLG
0017            PACK(I+6,JJ)=XINBF(I)
0018       10   CONTINUE
0019          IF(MESSEQ .EQ. 126) MESSEQ=0
0021          MESSEQ=MESSEQ+1
0022          PACK(1,JJ)=0
0023          PACK(2,JJ)=MESSEQ
0024          PACK(3,JJ)=0
0025          PACK(4,JJ)=0
0026          PACK(5,JJ)=MONTOR
0027          PACK(6,JJ)=ISLID
0028          PACK(IOFLG+8,JJ)=ETX
0029          CALL ENSTR(PACK(254,JJ),IOFLG+8)
0030          CALL ENABLE(0)
0031          CALL ENQUE(PXINQ,XINQ,JJ)
0032          CALL ENABLE(1)
0033          IOFLG=0
0034          RETURN
0035       500  DECODE(2,510,XINBF(7),ERR=900) ITERM
0036       510  FORMAT(I2)
0037          MONTOR=ITERM
0038          IOFLG=0
0039       900  RETURN
0040          END
```

```
                                        .TITLE  FDM.MACRO
                                        .SBTTL  NODE25
                                        .IDENT  /V3.0/
                                        .GLOBL  LIUINT,LID,ENABLE,SWITCH,MASTER,TIME
                                        .GLOBL  WTOKEN,RAM,STATUS,STATBO
                                        .GLOBL  LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
                                        .NLIST  CND
000000                                  .PSECT

                000000                  R0      =%0
                000001                  R1      =%1
                000002                  R2      =%2
                000003                  R3      =%3
                000004                  R4      =%4
                000005                  R5      =%5
                000006                  SP      =%6
                000007                  PC      =%7

                                        ;INTERFACE ADDRESS

                172410                  BAR=    172410
                172412                  WCR=    172412
                172414                  CSR=    172414
                172416                  IOBUF=  172416
                172416                  OPREG=  172416
                177560                  HRCSR=  177560
                177562                  HRBUF=  177562
                177564                  HXCSR=  177564
                177566                  HXBUF=  177566

                                        ;INTERFACE OPTIONS

                000000                  XCIO=0
                000001                  XSIO=1
                000001                  XGIO=1
                000000                  NDXX=0
                000001                  ND24=1
                000000                  NHOST=0
                000001                  DHOST=1

                                        ;LOCAL VARIBLES

                000002                  ARG1=   2
                000004                  ARG2=   4
                000006                  ARG3=   6
                000010                  ARG4=   10

                100000                  BIT15=  100000
                040000                  BIT14=  40000
                000400                  BIT08=  400
                000200                  BIT07=  200
                000100                  BIT06=  100
                000040                  BIT05=  40
                000020                  BIT04=  20
                000010                  BIT03=  10
                000004                  BIT02=  4
```

```
     66          000002                          BIT01=  2
     67          000001                          BIT00=  1
     68
```

```
                        ;*************************************************************
                        ;**** PROCEDURE MASTER START UP                              *
                        ;*************************************************************

000000  012700  000340  MASTER: MOV     #340,   R0                      ;PRI=7
000004  106400                  MTPS    R0
000006  012700  000124          MOV     #124,   R0
000012  012720  000436'         MOV     #LIO,   (R0)+                   ;LIU HANDLER
000016  012710  000340          MOV     #340,   (R0)                    ;PRI=7
000022  000005                  RESET                                   ;RESET BUS
000024  012700  000060          MOV     #60,    R0                      ;TERM VECTOR
000030  012720  002474'         MOV     #CIO,   (R0)+                   ;HANDLER
000034  012710  000340          MOV     #340,   (R0)                    ;PRI=340
000040  052737  000100  177560  BIS     #100,   @#HRCSR                 ;ENABLE INTERRUPTS
000046  012700  000100          MOV     #100,   R0                      ;CLOCK VECTOR
000052  012710  001662'         MOV     #TIME,  (R0)                    ;CLOCK
000056  042737  040100  172414  BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000064  004767  000014          JSR     PC,     LIUINT                  ;CLEAR LIU
000070  005037  172414          CLR     @#CSR                           ;CLEAR BLIUI CSR
000074  012700  000000          MOV     #000,   R0                      ;PRI=0
000100  106400                  MTPS    R0                              ;CPU DOWN
000102  000207                  RTS     PC


                        ;*************************************************************
                        ;**** PROCEDURE INITIALIZE                                   *
                        ;*************************************************************

                                ;CLEAR ACRAM

000104  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000112  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000120  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000124  100375                  BPL     .-4                             ;NO RETRY
000126  105037  172414          CLRB    @#CSR
000132  012737  010401  172416  MOV     #4353., @#OPREG                 ;SEL ACRAM
000140  012700  177400          MOV     #-256., R0                      ;COUNTER
000144  012737  004407  172416  1$:     MOV     #2311., @#OPREG         ;WRITE A NULL
000152  105737  172414          TSTB    @#CSR                           ;GOOD WD
000156  100375                  BPL     .-4                             ;NO LOOP UNTIL READY
000160  105037  172414          CLRB    @#CSR
000164  005200                  INC     R0                              ;COUNT+1
000166  001366                  BNE     1$

                                ;CLEAR INPUT/OUTPUT BUFFERS

000170  012701  010410          MOV     #10410, R1                      ;RDBUFADR CMD
000174  012700  010440          MOV     #10440, R0                      ;SEL INBUF0
000200  004767  000106          JSR     PC,     ZEROBP                  ;SET POINTER=0
000204  012737  001400  172416  MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000212  105737  172414          TSTB    @#CSR                           ;GOOD READ
000216  100375                  BPL     .-4                             ;NO RETRY
000220  105037  172414          CLRB    @#CSR                           ;CLEAR DONE BIT
000224  012701  010610          MOV     #10610, R1                      ;RDBUFADR COMMAND
000230  012700  010640          MOV     #10640, R0                      ;SEL INBUF1 COMMAND
```

```
 97 000234  004767  000052                    JSR     PC,     ZEROBP          ;POINTER=0
 98 000240  012737  001400  172416            MOV     #1400,  @#OPREG         ;FALSE READ DATA
 99 000246  105737  172414                    TSTB    @#CSR                   ;GOOD RD
100 000252  100375                            BPL     .-4                     ;NO RETRY
101 000254  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
102 000260  012701  010510                    MOV     #10510, R1              ;RDBUFADR COMMAND
103 000264  012700  010540                    MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
104 000270  004767  000016                    JSR     PC,     ZEROBP
105 000274  012701  010710                    MOV     #10710, R1              ;RDBUFADR COMMAND
106 000300  012700  010740                    MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
107 000304  004767  000002                    JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
108 000310  000440                            BR      STAT$                   ;GO CLEAR STATUS
109 000312  010137  172416          ZEROBP:   MOV     R1,     @#OPREG         ;RDBUFADR
110 000316  012737  001400  172416            MOV     #1400,  @#OPREG         ;RD
111 000324  105737  172414                    TSTB    @#CSR                   ;GOOD RD
112 000330  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
113 000332  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
114 000336  013702  172416                    MOV     @#IOBUF, R2             ;FETCH POINTER
115 000342  042702  177400                    BIC     #177400, R2             ;CLEAR MST BYTE
116 000346  010037  172416                    MOV     R0,     @#OPREG         ;SEL BUFFER
117 000352  022702  000000          2$:       CMP     #0,     R2              ;POINTER=0?
118 000356  001412                            BEQ     3$                      ;YES RECHECK
119 000360  012737  001400  172416            MOV     #1400,  @#OPREG         ;NO FALSE RD
120 000366  105737  172414                    TSTB    @#CSR                   ;GOOD RD
121 000372  100375                            BPL     .-4                     ;NO LOOP UNTIL READY
122 000374  105037  172414                    CLRB    @#CSR                   ;CLEAR DONE BIT
123 000400  005302                            DEC     R2                      ;LOCAL POINTER-1
124 000402  000763                            BR      2$                      ;BR UNTIL 0
125 000404  010037  172416          3$:       MOV     R0,     @#OPREG         ;SEL BUFFER
126 000410  000207                            RTS     PC                      ;RETURN
127
128                                           ;READ AND CLEAR STATUS
129
130 000412  012737  010400  172416  STAT$:   MOV     #4352., @#OPREG         ;WCR:RS
131 000420  012737  002400  172416            MOV     #1280., @#OPREG         ;RS(FALSE)
132 000426  012737  002400  172416            MOV     #1280., @#OPREG         ;RS
133 000434  000207                            RTS     PC
134
135
```

```
                                ;*************************************************************
                                ;**** PROCEDURE LIU-HANDLER                                 *
                                ;*************************************************************

000436  010046                LIU:    MOV     R0,     -(SP)                   ;SAVE R0
000440  010146                        MOV     R1,     -(SP)                   ;SAVE R1
000442  010246                        MOV     R2,     -(SP)                   ;SAVE R2
000444  010346                        MOV     R3,     -(SP)                   ;SAVE R3
000446  010446                        MOV     R4,     -(SP)                   ;SAVE R4
000450  010546                        MOV     R5,     -(SP)                   ;SAVE R5
000452  005737  172414              TST     @#CSR                           ;? LIU INTERRUPTED
000456  100153                      BPL     RTI$                            ;NO
000460  042737  040000  172414      BIC     #BIT14, @#CSR                   ;YES/DISABLE LIU
000466  012737  010400  172416      MOV     #4352., @#OPREG                 ;WCR : RS(0)
000474  012737  002400  172416      MOV     #1280., @#OPREG                 ;RD
000502  013767  172416  000502      MOV     @#IOBUF, CAUSE
000510  042767  177400  000474      BIC     #177400, CAUSE                  ;CLEAR UNUSED BITS

                                    ;PREFORM OPERATION DEPENDING ON BITS

000516  132767  000004  000466  B2$:  BITB    #BIT02, CAUSE                   ;OV-FL
000524  001406                      BEQ     B3$
000526  012702  010410              MOV     #4360., R2                      ;RDBUFADR IN0
000532  012703  010440              MOV     #4384., R3                      ;SEL INBUF0
000536  004767  000270              JSR     PC,     EMBF                    ;GO EMPTY
000542  132767  000010  000442  B3$:  BITB    #BIT03, CAUSE                   ;OV-FL
000550  001406                      BEQ     B4$
000552  012702  010610              MOV     #4488., R2                      ;RDBUFADR IN1
000556  012703  010640              MOV     #4512., R3                      ;SEL INBUF1
000562  004767  000244              JSR     PC,     EMBF                    ;GO EMPTY
000566  132767  000020  000416  B4$:  BITB    #BIT04, CAUSE                   ;LINE-LOSS PRIMARY
000574  001407                      BEQ     B5$
000576  132767  000200  000406  B7$:  BITB    #BIT07, CAUSE                   ;LATCH ON
000604  001403                      BEQ     B5$                             ;YES LOOP AROUND
000606  012767  000002  002002'      MOV     #2,     LLFLG                   ;FLAG PRIMARY SW. FAIL
000614  132767  000040  000370  B5$:  BITB    #BIT05, CAUSE                   ;BACKUP LINE LOSS
000622  001403                      BEQ     B6$
000624  012767  000001  002002'      MOV     #1,     LLFLG                   ;FLAG BACKUP SW. FAIL
000632  132767  000100  000352  B6$:  BITB    #BIT06, CAUSE                   ;WRITE TOKEN DETECT
000640  001404                      BEQ     B0$
000642  005067  001000'              CLR     IWRTTM                          ;CLEAR TOKEN VAR
000646  005067  001002'              CLR     IWRT

                                    ;BUFFER FULL

000652  132767  000001  000332  B0$:  BITB    #BIT00, CAUSE                   ;INBUF0 FULL
000660  001423                      BEQ     B1$                             ;NO
000662  012701  000400'              MOV     #INBF,  R1                      ;BUFFER ADDRESS
000666  012702  010410              MOV     #4360., R2                      ;RDBUFADR COMM
000672  012703  010440              MOV     #4384., R3                      ;SEL BUF
000676  012704  000001              MOV     #BIT00, R4                      ;CRC BIT
000702  004767  000124              JSR     PC,     EMBF                    ;EMPTY BUFFER
000706  012705  001214'              MOV     #AREA,  R5                      ;DATA LINK AREA
000712  012767  000001  000274      MOV     #1,     AREA                    ;1 VARIBLE
000720  010267  000274              MOV     R2,     DATA                    ;CRC OR BYTE COUNT
000724  004767  000006              JSR     PC,     LPINPT                  ;CALL FORTRAN QUE'ER
```

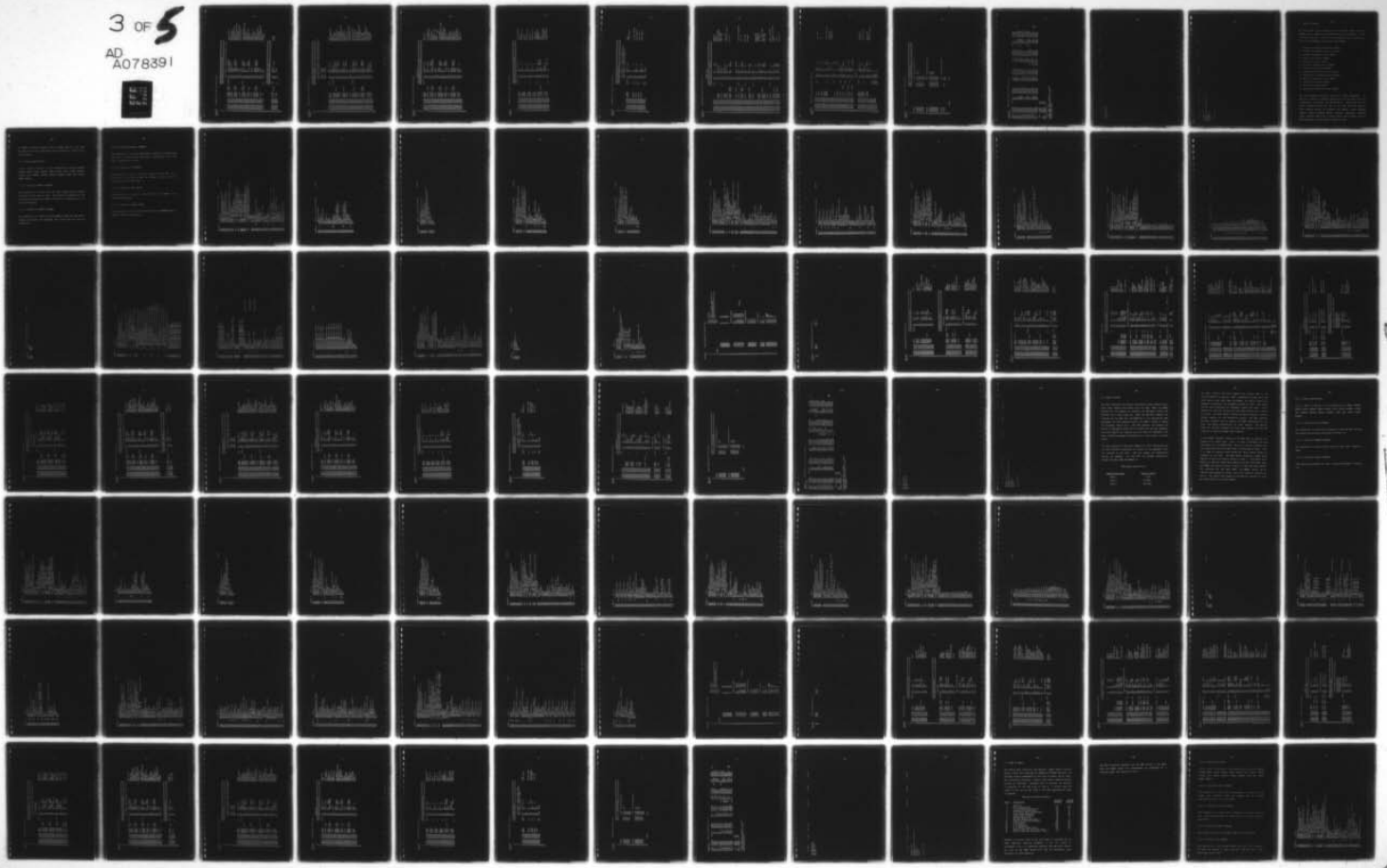
```
    58 000730  132767  000002  000254  B1$:    BITB    #BIT01, CAUSE           ;INBUF1 FULL
    59 000736  001423                   BEQ     RTI$                    ;NO WERE DONE
    60 000740  012701  000400'           MOV     #INBF,  R1              ;BUFFER ADDRESS
    61 000744  012702  010610            MOV     #4488., R2              ;RDBUFADR COMM
    62 000750  012703  010640            MOV     #4512., R3              ;SEL INBUF1
    63 000754  012704  000002            MOV     #BIT01, R4              ;CRC BIT
    64 000760  004767  000046            JSR     PC,     EMBF            ;GO EMPTY BUFFER
    65 000764  012705  001214'           MOV     #AREA,  R5              ;DATA LINK AREA
    66 000770  012767  000001  000216    MOV     #1,     AREA            ;ONE VARIBLE
    67 000776  010267  000216            MOV     R2,     DATA            ;CRC OR BYTE COUNT
    68 001002  004767  000006            JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
    69
    70 001006  012605                RTI$:   MOV     (SP)+,  R5              ;RESTORE REGISTERS
    71 001010  012604                        MOV     (SP)+,  R4
    72 001012  012603                        MOV     (SP)+,  R3
    73 001014  012602                        MOV     (SP)+,  R2
    74 001016  012601                        MOV     (SP)+,  R1
    75 001020  012600                        MOV     (SP)+,  R0
    76 001022  052737  040000  172414        BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
    77 001030  000002                        RTI                             ;RETURN FROM INTERRUPT
    78
    79 001032  012737  010600  172416  EMBF:   MOV     #4480., @#OPREG         ;READ STATUS 1
    80 001040  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
    81 001046  013700  172416                MOV     @#IOBUF, R0             ;FETCH STATUS
    82 001052  130400                        BITB    R4,R0                   ;GOOD CRC
    83 001054  001002                        BNE     CRCOK
    84 001056  012704  177777                MOV     #-1,    R4              ;NO FLAG
    85 001062  010237  172416        CRCOK:  MOV     R2,     @#OPREG         ;RDBUFADR
    86 001066  012737  001400  172416        MOV     #768.,  @#OPREG         ;RD
    87 001074  105737  172414                TSTB    @#CSR                   ;GOOD RD
    88 001100  100375                        BPL     .-4
    89 001102  013702  172416                MOV     @#IOBUF, R2             ;REPLACE WITH POINTER
    90 001106  042702  177400                BIC     #177400, R2             ;CLEAR MST BITS
    91 001112  010200                        MOV     R2,     R0              ;SAVE IT
    92 001114  005400                        NEG     R0                      ;2'S COMP
    93 001116  010037  172412                MOV     R0,     @#WCR           ;BYTE COUNT NOW
    94 001122  010137  172410                MOV     R1,     @#BAR           ;ADDRESS IN MEMORY
    95 001126  010337  172416                MOV     R3,     @#OPREG         ;SEL BUFFER
    96 001132  012737  001400  172416        MOV     #768.,  @#OPREG         ;FALSE RD
    97 001140  105737  172414                TSTB    @#CSR                   ;DONE ON
    98 001144  100375                        BPL     .-4                     ;NO LOOP
    99 001146  012737  021000  172416        MOV     #8704., @#OPREG         ;FIRE DMA
   100 001154  000240                        NOP                             ;DELAY
   101 001156  105737  172414                TSTB    @#CSR                   ;GOOR DMA
   102 001162  100401                        BMI     DMAOK
   103 001164  000240                        NOP                             ;ERROR IF HERE
   104 001166  012737  004400  172416  DMAOK: MOV     #2304., @#OPREG         ;FALSE WD
   105 001174  105737  172414                TSTB    @#CSR                   ;GOOD WD
   106 001200  100375                        BPL     .-4
   107 001202  005704                        TST     R4                      ;WAS CRC OK
   108 001204  100001                        BPL     END$                    ;YES
   109 001206  010402                        MOV     R4,     R2              ;NO FLAG IT
   110 001210  000207                END$:   RTS     PC                      ;RETURN
   111 001212  000000                CAUSE:  .WORD   0                       ;STATUS BYTE 0 HOLDER
   112 001214  000000                AREA:   .WORD   0
   113 001216  001220'                       .WORD   DATA
   114 001220  000000                DATA:   .WORD   0
```

```
                                        .ENABLE LSB
                                    ;************************************************************
                                    ;**** PROCEDURE ENABLE                                      *
                                    ;************************************************************

001222  017501  000002          ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
001226  022701  000001                  CMP     #1,     R1
001232  001404                          BEQ     E1$
001234  042737  040000  172414  E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
001242  000207                          RTS     PC
001244  052737  040000  172414  E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
001252  000207                          RTS     PC

                                        ;FORTRAN ENCODE/DECODE MACRO

001254  017575  000004  000002  ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
001262  000207                          RTS     PC
001264  017575  000002  000004  DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECODE BYTE COUNT
001272  000207                          RTS     PC


                                    ;************************************************************
                                    ;**** PROCEDURE LINE SWITCH                                 *
                                    ;************************************************************

                                        ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                        ;
                                        ;04=SET PRIMARY LINE
                                        ;32=RESET PRIMARY LINR
                                        ;08=SET BACKUP LINE
                                        ;64=RESET BACKUP LINE

001274  017501  000002          SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
001300  012737  010420  172416          MOV     #4368., @#OPREG                 ;WCR : MODSTAT
001306  062701  004400                  ADD     #2304., R1
001312  010137  172416                  MOV     R1,     @#OPREG                 ;RD
001316  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001322  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001324  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE.BIT
001330  000207                          RTS     PC
```

```
                        ;*************************************************************
                        ;**** PROCEDURE STATUS                                      *
                        ;*************************************************************

                                ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
                                ;
                                ;X=0 STATUS BYTE 0
                                ;X=1 STATUS BYTE 1

001332  017501  000002          STATUS: MOV     @ARG1(R5), R1                   ;WHICH BYTE
001336  022701  000001                  CMP     #1,     R1                      ;COMPARE
001342  001415                          BEQ     2$
001344  012737  010400  172416  1$:     MOV     #4352., @#OPREG                 ;WCR : RS(0)
001352  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001360  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001364  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001370  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001374  000207                          RTS     PC
001376  012737  010600  172416  2$:     MOV     #4480., @#OPREG                 ;WCR : RS(1)
001404  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001412  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001416  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001422  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001426  000207                          RTS     PC
001430  012737  010400  172416  STATB0: MOV     #4352., @#OPREG                 ;STATUS BYTE 1
001436  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001444  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001450  042701  177400                  BIC     #177400, R1
001454  132701  000010                  BITB    #BIT03, R1                      ;BUFFER FULL
001460  001404                          BEQ     3$
001462  012775  000001  000002          MOV     #1,     @ARG1(R5)               ;LOAD RETURN
001470  000207                          RTS     PC
001472  012775  000000  000002  3$:     MOV     #0,     @ARG1(R5)               ;LOAD RETURN
001500  000207                          RTS     PC
```

AD-A078 391 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66157 SBIE-AD-E100 313 NL

3 OF 5
AD A078391



```
                        ;*************************************************************
                        ;**** PROCEDURE WRITE TOKEN (WTOKEN)                        *
                        ;*************************************************************

                                ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP

001502  012737  010600' 172416  WTOKEN: MOV     #4480., @#OPREG                 ;READ STATUS(1)
001510  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001516  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001522  132701  000010                  BITB    #BIT03, R1                      ;BUFFER LOADED
001526  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001530  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001536  012737  004423  172416          MOV     #2323., @#OPREG                 ;MARK OB0F/OB1F/WTCMD
001544  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001550  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001552  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001556  000207                          RTS     PC
001560  012737  010740  172416  WTKG:   MOV     #4576., @#OPREG                 ;SELECT BUFFER 1
001566  012737  001660' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001574  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001600  005402                          NEG     R2                              ;FORMAT IT
001602  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001606  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001614  000240                          NOP                                     ;DELAY
001616  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001622  000240                          NOP                                     ;ERROR IF HERE
001624  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001630  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001636  012737  004422  172416          MOV     #2322., @#OPREG                 ;MARK OB1F/WTCMD
001644  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001650  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001652  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001656  000207                          RTS     PC                              ;RETURN
001660     000     377          WTD:    .BYTE 0,377                             ;A WTOKEN

                        ;*************************************************************
                        ;**** PROCEDURE TIMER (TIME)                                *
                        ;*************************************************************

001662  005267  001006' TIME:   INC     LTIME                           ;TIME+1
001666  022767  077777  001006'         CMP     #077777, LTIME                  ;LIMIT
001674  001002                          BNE     4$
001676  005067  001006'                 CLR     LTIME
001702  000002                  4$:     RTI
```

```
 1                                ;*************************************************************
 2                                ;**** PROCEDURE ACRAM                                       *
 3                                ;*************************************************************
 4
 5                                        ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
 6                                        ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
 7                                        ;
 8                                        ;0110   =NREAD
 9                                        ;0100   =DREAD
10                                        ;0111   =NULL
11                                        ;0000   =WTOKEN
12
13
14 001704  017500  000002          RAM:   MOV     @ARG1(R5), R0                    ; RAM
15 001710  022700  000000                 CMP     #0,      R0                      ;WHICH OPERATION
16 001714  001035                         BNE     RDRAM                            ;READ OP
17 001716  017500  000004          WTRAM: MOV     @ARG2(R5), R0                    ;ADDRESS
18 001722  017501  000006                 MOV     @ARG3(R5), R1                    ;WRITE DATA
19 001726  012737  010402  172416         MOV     #4354.,  @#OPREG                 ;SEL LDACTR
20 001734  062700  004400                 ADD     #2304.,  R0                      ;WD/DATA (ADDR)
21 001740  010037  172416                 MOV     R0,      @#OPREG                 ;WRITE DATA
22 001744  105737  172414                 TSTB    @#CSR                            ;VALID WRITE
23 001750  100375                         BPL     .-4                              ;NO LOOP UNTIL READY
24 001752  105037  172414                 CLRB    @#CSR                            ;CLEAR DONE BIT
25 001756  012737  010401  172416         MOV     #4353.,  @#OPREG                 ;SEL ACRAM
26 001764  062701  004400                 ADD     #2304.,  R1                      ;WD/DATA (CMD)
27 001770  010137  172416                 MOV     R1,      @#OPREG                 ;WRITE
28 001774  105737  172414                 TSTB    @#CSR                            ;VALID WRITE
29 002000  100375                         BPL     .-4                              ;NO LOOP UNTIL READY
30 002002  105037  172414                 CLRB    @#CSR                            ;CLEAR DONE BIT
31 002006  000207                         RTS     PC
32 002010  017500  000004          RDRAM: MOV     @ARG2(R5), R0                    ;ADDRESS TO READ
33 002014  012737  010402  172416         MOV     #4354.,  @#OPREG                 ;SEL LDACTR
34 002022  062700  004400                 ADD     #2304.,  R0                      ;WD/DATA (ADDR)
35 002026  010037  172416                 MOV     R0,      @#OPREG                 ;WRITE DATA
36 002032  105737  172414                 TSTB    @#CSR                            ;VALID WRITE
37 002036  100375                         BPL     .-4                              ;NO LOOP UNTIL READY
38 002040  105037  172414                 CLRB    @#CSR                            ;CLEAR DONE BIT
39 002044  012737  010401  172416         MOV     #4353.,  @#OPREG                 ;SEL ACRAM
40 002052  012737  001400  172416         MOV     #768.,   @#OPREG                 ;READ DATA
41 002060  105737  172414                 TSTB    @#CSR                            ;VALID READ
42 002064  100375                         BPL     .-4                              ;NO LOOP UNTIL READY
43 002066  105037  172414                 CLRB    @#CSR                            ;CLEAR DONE BIT
44 002072  013700  172416                 MOV     @#IOBUF, R0                      ;FETCH DATA
45 002076  042700  177760                 BIC     #177760, R0                      ;CLEAR BITS
46 002102  010075  000006                 MOV     R0, @ARG3(R5)                    ;RTN DATA
47 002106  000207                         RTS     PC
48
49
```

```
                                .ENABLE LSB
                        ;*************************************************************
                        ;**** PROCEDURE WRITE LOOP (LPOUT)                          *
                        ;*************************************************************

002110  017502  000002          LPOUT:  MOV     @ARG1(R5), R2                   ;BYTE COUNT
002114  012700  000340                  MOV     #340,   R0                      ;PRI=7
002120  106400                          MTPS    R0
002122  012737  000000' 172410          MOV     #OUTBF, @#BAR                   ;BUS ADDRESS
002130  005402                          NEG     R2                              ; 2'S COMP COUNT
002132  010237  172412                  MOV     R2,     @#WCR                   ;COUNT
002136  012737  010540  172416          MOV     #4448., @#OPREG                 ;OB0 COMMAND
002144  012737  024000  172416          MOV     #10240., @#OPREG                ;DMA GO
002152  000240                          NOP                                     ;INTERFACE TIME
002154  105737  172414                  TSTB    @#CSR                           ;DMA OK
002160  000240                          NOP                                     ;ERROR IF HERE
002162  105037  172414                  CLRB    @#CSR
002166  012737  010740  172416          MOV     #4576., @#OPREG                 ;OB1 COMMAND
002174  012737  001660' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
002202  012702  000002                  MOV     #2,     R2                      ;BYTE COUNT
002206  005402                          NEG     R2                              ;FORMAT FOR BLIUI
002210  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
002214  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
002222  000240                          NOP                                     ;DELAY
002224  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
002230  000240                          NOP                                     ;ERROR IF NOT
002232  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
002236  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
002244  012737  004403  172416          MOV     #2307., @#OPREG                 ;BUFFERS FULL
002252  105737  172414                  TSTB    @#CSR                           ;GOOD WRITE
002256  100375                          BPL     .-4                             ;NO LOOP UNTIL
002260  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002264  012700  000000                  MOV     #000,   R0                      ;PRI=0
002270  106400                          MTPS    R0                              ;LOWER CPU
002272  000207                          RTS     PC
```

```
                                ;*********************************************************
                                ;**** PROCEDURE READ BUFFER POINTER                      *
                                ;*********************************************************

002274  017500  000002          RDPNT:  MOV     @ARG1(R5), R0           ;FETCH COMMAND
002300  022700  000000                  CMP     #0,     R0              ;INBUF0 ?
002304  001004                          BNE     1$                      ;NO
002306  012700  010410                  MOV     #4360., R0              ;RDBUFADR IN0
002312  000167  000046                  JMP     4$
002316  022700  000001          1$:     CMP     #1,     R0              ;INBUF1 ?
002322  001004                          BNE     2$                      ;NO
002324  012700  010610                  MOV     #4488., R0              ;RDBUFADR IN1
002330  000167  000030                  JMP     4$
002334  022700  000002          2$:     CMP     #2,     R0              ;OUTBUF0 ?
002340  001004                          BNE     3$                      ;NO
002342  012700  010510                  MOV     #4424., R0              ;RDBUFADR OUT0
002346  000167  000012                  JMP     4$
002352  022700  000004          3$:     CMP     #4,     R0              ;OUTBUF1 ?
002356  001020                          BNE     5$                      ;NO RETURN
002360  012700  010710                  MOV     #4552., R0              ;RDBUFADR OUT1
002364  010037  172416          4$:     MOV     R0,     @#OPREG         ;RDBUFADR
002370  012737  001400  172416          MOV     #768.,  @#OPREG         ;RD
002376  105737  172414                  TSTB    @#CSR                   ;READY
002402  100375                          BPL     .-4                     ;NO LOOP UNTIL
002404  013700  172416                  MOV     @#IOBUF, R0             ;FETCH POINTER
002410  042700  177400                  BIC     #177400, R0             ;CLEAR MST
002414  010075  000004                  MOV     R0,     @ARG2(R5)       ;RETURN POINTER
002420  000207          5$:     RTS     PC
```

```
                        ;*************************************************************
                        ;**** PROCEDURE RESTART                                     *
                        ;*************************************************************

                                ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
                                ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
                                ;CALL RSTART(2) -LOAD MODE(173000)

002422  017500  000002  RSTART: MOV     @ARG1(R5), R0                   ;MODE
002426  022700  000000  RS0:    CMP     #0,     R0                      ;LOAD ADDRESS
002432  001002                  BNE     RS1
002434  000000                  HALT
002436  000207                  RTS     PC
002440  022700  000001  RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
002444  001003                  BNE     RS2
002446  013700  000040  G4$:    MOV     @#40,   R0                      ;START ADDRESS
002452  000110                  JMP     (R0)
002454  022700  000002  RS2:    CMP     #2,     R0                      ;LOAD MODE
002460  001003                  BNE     RST
002462  013700  002472'                 MOV     @#ROM,  R0              ;BOOT ADDRESS
002466  000110                  JMP     (R0)
002470  000207  RST:    RTS     PC
002472  173000  ROM:    .WORD   173000
```

```
                                        .ENABLE LSB
                                        .GLOBL  HINPUT
                                ;*******************************************************
                                ;****    PROCEDURE CRT INTERFACE                     *
                                ;*******************************************************

002474  010046                  CIO::   MOV     R0,     -(SP)           ;SAVE R0
002476  010146                          MOV     R1,     -(SP)           ;SAVE R1
002500  010246                          MOV     R2,     -(SP)           ;SAVE R2
002502  016700  000454                  MOV     CPBUF, R0               ;BUFFER POINTER
002506  012701  000400'                 MOV     #XINBF, R1              ;BUFFER ADDRESS
002512  060001                          ADD     R0,     R1              ;CHAR OFFSET
002514  105737  177560          1$:     TSTB    @#HRCSR                 ;CHAR READY
002520  100134                          BPL     7$
002522  113702  177562                  MOVB    @#HRBUF, R2
002526  042702  177600                  BIC     #177600, R2             ;CLEAR PAR.
002532  122702  000022                  CMPB    #022,   R2              ;TURN ON CRT(REPORT)
002536  001015                          BNE     20$
002540  012767  000000  000000'         MOV     #0,     IXON            ;IT'S ON
002546  012737  000007  177566          MOV     #007,   @#HXBUF
002554  000240                          NOP
002556  000240                          NOP
002560  000240                          NOP
002562  012737  000007  177566          MOV     #007,   @#HXBUF
002570  000510                          BR      7$
002572  122702  000023          20$:    CMPB    #023,   R2              ;TURN OFF CRT(REPORT)
002576  001007                          BNE     2$
002600  012767  000001  000000'         MOV     #1,     IXON            ;IT'S OFF
002606  012737  000007  177566          MOV     #007,   @#HXBUF
002614  000476                          BR      7$
002616  122702  000177          2$:     CMPB    #177,   R2              ;DEL CHAR
002622  001007                          BNE     3$
002624  005300                          DEC     R0                      ;POINTER-1
002626  012767  000134  000330          MOV     #134,   CHAR
002634  004767  000204                  JSR     PC,     ECHO
002640  000464                          BR      7$
002642  122702  000015          3$:     CMPB    #015,   R2
002646  001022                          BNE     4$
002650  110221                          MOVB    R2,     (R1)+           ;STORE CHAR
002652  005200                          INC     R0                      ;POINTER+1
002654  004767  000202                  JSR     PC,     CRLF            ;PRINT CHAR
002660  112721  000012                  MOVB    #012,   (R1)+           ;LF CHAR
002664  005200                          INC     R0
002666  012767  000007  000270          MOV     #007,   CHAR            ;ATT PACKET READY
002674  004767  000144                  JSR     PC,     ECHO            ;RING BELL
002700  010067  002000'                 MOV     R0,     IOFLG           ;BYTE COUNT
002704  005000                          CLR     R0
002706  005067  000246                  CLR     CLP
002712  000437                          BR      7$
002714  122702  000025          4$:     CMPB    #025,   R2              ;ERASE LINE CHAR
002720  001017                          BNE     5$
002722  016700  000232                  MOV     CLP,    R0              ;RESTORE LINE POINTER
002726  012767  000136  000230          MOV     #136,   CHAR            ;PRINT ^
002734  004767  000104                  JSR     PC,     ECHO
002740  012767  000125  000216          MOV     #125,   CHAR            ;PRINT U
002746  004767  000072                  JSR     PC,     ECHO
002752  004767  000104                  JSR     PC,     CRLF            ;PRINT CR LF
```

```
     58 002756  000415                          BR      7$
     59 002760  122702  000003          5$:     CMPB    #003,   R2              ;EOF CHAR
     60 002764  001020                          BNE     9$
     61 002766  012721  000012                  MOV     #012,   (R1)+           ;CR
     62 002772  005200                          INC     R0
     63 002774  012721  000015                  MOV     #015,   (R1)+           ;LF
     64 003000  005200                          INC     R0
     65 003002  004767  000054                  JSR     PC,     CRLF
     66 003006  010067  000146                  MOV     R0,     CLP             ;FLAG A NEW LINE
     67 003012  010067  000144          7$:     MOV     R0,     CPBUF
     68 003016  012602                          MOV     (SP)+,  R2              ;RESTORE R2
     69 003020  012601                          MOV     (SP)+,  R1              ;RESTORE R1
     70 003022  012600                          MOV     (SP)+,  R0              ;RESTORE R0
     71 003024  000002                          RTI
     72 003026  110221                  9$:     MOVB    R2,     (R1)+
     73 003030  010267  000130                  MOV     R2,     CHAR
     74 003034  005200                          INC     R0
     75 003036  004767  000002                  JSR     PC,     ECHO
     76 003042  000763                          BR      7$
     77 003044  105737  177564          ECHO:   TSTB    @#HXCSR
     78 003050  100375                          BPL     ECHO
     79 003052  116737  000106  177566          MOVB    CHAR, @#HXBUF
     80 003060  000207                          RTS     PC
     81 003062  105737  177564          CRLF:   TSTB    @#HXCSR
     82 003066  100375                          BPL     CRLF
     83 003070  012737  000012  177566          MOV     #012,   @#HXBUF
     84 003076  105737  177564          10$:    TSTB    @#HXCSR
     85 003102  100375                          BPL     10$
     86 003104  012737  000015  177566          MOV     #015,   @#HXBUF
     87 003112  000207                          RTS     PC
     88
     89                                         ;WRITE CRT
     90
     91 003114  017500  000002          COUT::  MOV     @ARG1(R5), R0           ;BYTE COUNT
     92 003120  012701  000000'                 MOV     #XOUTBF, R1             ;BUFFER
     93 003124  062700  000006                  ADD     #6,     R0
     94 003130  062701  000006                  ADD     #6,     R1              ;OFFSET PACKET
     95 003134  105737  177564          11$:    TSTB    @#HXCSR
     96 003140  100375                          BPL     11$
     97 003142  112137  177566                  MOVB    (R1)+, @#HXBUF          ;WRITE CHAR
     98 003146  122711  000003                  CMPB    #003,   (R1)            ;END YET
     99 003152  001401                          BEQ     12$                     ;YES STOP PRINTING
    100 003154  077011                          SOB     R0,     11$
    101 003156  000207                  12$:    RTS     PC
    102 003160  000000                  CLP:    .WORD   0
    103 003162  000000                  CPBUF:  .WORD   0
    104 003164  000000                  CHAR:   .WORD   0
    105 003166                          DUMP:   .BLKW   50.
    226
```

```
                        ;*************************************************************
                        ;****        COMMON DATA AREA                                 *
                        ;*************************************************************

000000                          .PSECT  DFM,RW,D,GBL,REL,OVR

000000                  OUTBF:  .BLKB   256.
000400                  INBF:   .BLKB   256.
001000                  IWRTTM: .BLKW
001002                  IWRT:   .BLKW
001004                  STAT:   .BLKW
001006                  LTIME:  .BLKW
                                .EVEN

000000                          .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                  XOUTBF: .BLKB   256.
000400                  XINBF:  .BLKB   256.
001000                  ACKSEQ: .BLKB   256.
001400                  LIDFD:  .BLKB   256.
002000                  IOFLG:  .BLKW
002002                  LLFLG:  .BLKW
002004                  IRSEND: .BLKW


000000                          .PSECT  CRT,RW,D,GBL,REL,OVR

000000                  IXON:   .BLKW
                                .EVEN


000001                          .END
```

```
FDM.MACRO       MACRO V03.01 5-JUN-79 13:33:26 PAGE 12-1
SYMBOL TABLE

ACKSEQ  001000R      003 B2$     000516R        ECHO    003044R        LIDFD   001400R      003 RS1     002440R
AREA    001214R          B3$     000542R        EMBF    001032R        LIO     000436RG         RS2     002454R
ARG1  = 000002           B4$     000566R        ENABLE  001222RG       LIUINT  000104RG         RTI$    001006R
ARG2  = 000004           B5$     000614R        END$    001210R        LLFLG   002002R      003 STAT    001004R      002
ARG3  = 000006           B6$     000632R        ENSTR   001254RG       LPINPT= ****** G         STATBO  001430RG
ARG4  = 000010           B7$     000576R        EO$     001234R        LPOUT   002110RG         STATUS  001332RG
BAR   = 172410           CAUSE   001212R        E1$     001244R        LTIME   001006R      002 STAT$   000412R
BIT00 = 000001           CHAR    003164R        G4$     002446R        MASTER  000000RG         SWITCH  001274RG
BIT01 = 000002           CIO     002474RG       HINPUT= ****** G       NDXX  = 000000           TIME    001662RG
BIT02 = 000004           CLP     003160R        HRBUF = 177562         ND24  = 000001           WCR   = 172412
BIT03 = 000010           COUT    003114RG       HRCSR = 177560         NHOST = 000000           WTD     001660R
BIT04 = 000020           CPBUF   003162R        HXBUF = 177566         OPREG = 172416           WTKG    001560R
BIT05 = 000040           CRCOK   001062R        HXCSR = 177564         OUTBF   000000R      002 WTOKEN  001502RG
BIT06 = 000100           CRLF    003062R        INBF    000400R  002 RAM     001704RG         WTRAM   001716R
BIT07 = 000200           CSR   = 172414         IOBUF = 172416         RDPNT   002274RG         XCIO  = 000000
BIT08 = 000400           DATA    001220R        IOFLG   002000R  003 RDRAM   002010R          XGIO  = 000001
BIT14 = 040000           DESTR   001264RG       IRSEND  002004R  003 ROM     002472R          XINBF   000400R      003
BIT15 = 100000           DHOST = 000001         IWRT    001002R  002 RST     002470R          XOUTBF  000000R      003
B0$     000652R          DMAOK   001166R        IWRTTM  001000R  002 RSTART  002422RG         XSIO  = 000001
B1$     000730R          DUMP    003166R        IXON    000000R  004 RS0     002426R          ZEROBP  000312R

. ABS.  000000    000
        003332    001
DFM     001010    002
BUFS    002006    003
CRT     000002    004
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  306 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  50 PAGES
DK:FDM,DK:FDM=DK:FDM
```

```
[illegible]    [illegible]:    FORT25.COM          01-JUN-79        12:00:00        PAGE 001

FOR NODAL-NOLINE
FOR NODAL1-NOLINE
MAC FDM
```

decwriter   Program: LNK25.COM   01-JUN-79   12:00:00   PAGE 001

```
ASSIGN DX0: DK:
R LINK
DX1:NODE25,DX1:MAP=DX1:NODAL/I/C/W
DX1:FDA/C
DX1:NODAL1//
$START

ASSIGN DX1: DK:
```

1.7 NODE 26 (BWBSA)

The Base Band Signal Analysis and Wide Band Signal Analysis (BWBSA) module assesses the performance of link equipment. Three links are monitored. There are seven parameters to be checked per link for the baseband, and six for the wideband:

1. Transmitter Percent Modulation (BTPM)
2. Transmitter Frequency Deviation (BTFD)
3. Relative Transmitter Power (BRTP)
4. Receiver AGC Voltage (BRAV)
5. Receiver IF Output (BRIO)
6. Multiplex Baseband Levels (BMBL)
7. Multiplex Pilot Levels (BMPL)
8. Transmitter Percent Deviation (WTFD)
9. Transmitter Frequency Deviation (WTFD)
10. Relative Transmitter Power (WRTP)
11. Receiver AGC Voltage (WRAV)
12. Receiver IF Output (WRIO)
13. Multiplexor Pseudo Error Rate (WMPE)

The SIG produces simulated values for three parameters. In addition, alarms are generated associated with the three links for transmitters, receivers, and multiplexers. Alarms and Red and Amber threshold values are sent to the FIAC as Event Reports. Event Reports consist of one-byte link number, 2-byte baseband number, 2-byte wideband number, one-byte supergroup, one-byte group, one-byte condition, a 3-byte monitor point number, and the node designator to which reports should be sent.

The BWBSA interprets commands from the DBMS similar to the VSQC and DSQC except that measurements are performed for 3 links rather than channels.

1.7.1 Program Descriptions

1.7.1.1 Refer to Section 1.1 for description of routines NODAL, IGETSP, ENQUE, DEQUE, ACKNAK, INPTQ, LPINPT, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME, RAM, LPOUT, RDPNT, RSTART.

1.7.1.2 Subroutine BWBSA (FORTRAN)

This subroutine is called from the nodal program when a message has been received from the SIG. The inputs are compared with the thresholds and subroutine SNDRPT is called if a measurement is out side the threshold.

1.7.1.3 Subroutine SNDRPT (FORTRAN)

This subroutine is called by routine BWBSA to send the FIAC node a message identifying the equipment that failed and the value of measurement.

1.7.1.4 Subroutine BWSALP (FORTRAN)

This subroutine is called by nodal when a message is received from the loop. It performs three functions: reporting on or off, and take a measurement of a link.

1.7.1.5 Subroutine SIO (MACRO)

This subroutine reads the simulated inputs from the SIG. It is called by an interrupt and when the message is read, it sets a flag containing the byte count.

1.7.1.6 Subroutine SOUT (MACRO)

This subroutine is used to send the SIG a link number for a requested measurement.

1.7.1.7 Subroutine READY (MACRO)

This subroutine is used to signal the SIG that the BWBSA module is ready for another measurement.

```
0001          PROGRAM NODAL
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003          INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP
0005          INTEGER*2 RSTBKP,Q1,Q2,RESNUM,OUTQ,DEQUE
0006          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007          LOGICAL*1 IDATA
0008          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0009          INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0010          REAL*4 RH,RL,AH,AL,VMEAS
0011          COMMON /MESS/ MESSEQ
0012          COMMON /DFM/ OUTBF(256),INBF(256),
             1       IWRTTM,IWRT,STAT,LTIME
0013          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1       ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0014          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1       LIDFD(256),IOFLG,LLFLG,IRSEND
0015          COMMON /FRE/ FREE(64),IFR,IFRSZ
0016          COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0017          COMMON       PACK(256,64)
0018          COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0019          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNUM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0020          COMMON /BCOM/ ISWTCH,RH(13,3),RL(13,3),AH(13,3),AL(13,3),
             &    ITERM,LINE,MON,VMEAS(13),ISUB,INUM,IDATA(10),
             &    ICHNSW
0021          CALL MASTER
0022          CALL INIT
0023          CALL ENABLE(1)
0024        5 CALL READY
0025       25 IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0027          CALL ENABLE(0)
0028          Q2=DEQUE(PINQ,INQ,1)
0029          CALL ENABLE(1)
0030          DO 30 I=1,10
0031            IDATA(I)=PACK(I+6,Q2)
0032       30 CONTINUE
0033          CALL BWSALP
0034          IFR=IFR+1
0035          FREE(IFR)=Q2
0036       40 CALL STATB0(IS)
0037          IF(IS .EQ. 1) GOTO 100
0039          IF(IOFLG .GT. 0 .A.D. IRSEND .EQ. 0) CALL BWBSA
0041          IF(LLFLG .GT. 0) CALL LINLOS
0043       80 IF(OUTFCT .EQ. 1) GOTO 100
0045          IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 120
0047          IF(IRSEND .EQ. 1) GOTO 85
0049          IF(PXINQ(1) .LT. PXINQ(2)) GOTO 100
0051          CALL ENABLE(0)
0052          Q1=DEQUE(PXINQ,XINQ,1)
0053          CALL ENABLE(1)
0054          OUTQ=Q1
0055       85 CALL DESTR(PACK(254,OUTQ),Q2)
0056          DO 90 I=1,Q2
```

```
0057            OUTBF(I)=PACK(I,OUTQ)
0058      90    CONTINUE
0059            OUTBF(Q2+1)=0
0060            IPT=OUTBF(5)
0061            OUTBF(Q2+2)=LIDFD(IPT)
0062            IFULL=1
0063            ISENT=1
0064            CALL LPOUT(Q2+2)
0065            IRSEND=1
0066            ACKTIM=0
0067            IWRTTM=0
0068            IWRT=1
0069    100     INFLCT=0
0070    110     OLDTIM=NEWTIM
0071            NEWTIM=LTIME
0072            IF(IWRT .EQ. 0) GOTO 120
0074            IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0075            IF(IWRTTM .LT. TIMLIM) GOTO 120
0077            CALL WTOKEN
0078            IWRTTM=0
0079            IWRT=0
0080    120     IF(ISENT .EQ. 0) GOTO 130
0082            ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0083            IF(ACKTIM .LT. ATIMLM) GOTO 130
0085            CALL ENABLE(0)
0086            CALL ACKNAK(0)
0087            CALL ENABLE(1)
0088    130     CONTINUE
0089            GOTO 5
0090            END
```

```
0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(64),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IF(IFR .LT. 1) CALL INIT
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```

```
0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008       10 IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016       20 IF(IQHEAD .GE. (IQLNTH)) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020       30 B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024       40 CALL INIT
0025      999 RETURN
0026          END
```

```
FORTRAN IV      V02.1-11   Mon 01-Oct-79 09:00:44              PAGE 001

0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(16),IEQUE,OUTFCT,OUTQ,RESNLM
0005          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016       10 A(1)=IQHEAD-1
0017      999 RETURN
0018          END
```

```
0001        SUBROUTINE ACKNAK(N)
0002        INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ,PXOUTQ
0003        INTEGER*2 ACKQ,PACKQ,PINQ,FREE,OUTQ,XOUTQ,FINDPK
0004        INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005        LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006        LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007        LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008        INTEGER*2 LTIME,IWRTIM
0009        REAL*8 COMT(5)
0010        EQUIVALENCE(COMT,LCOMT)
0011        DATA COMT(1)/'MESSAGE '/,COMT(2)/'NOT SENT'/,COMT(3)/' FROM    '/,
           1    COMT(4)/'NODE    '/,COMT(5)/' TO NODE'/
0012        COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
           1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013        COMMON /DFM/ OUTBF(256),INBF(256),
           1      IWRTTM,IWRT,STAT,LTIME
0014        COMMON         PACK(256,64)
0015        COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
           1          PACKQ(2),INQ(16),PINQ(2)
0016        COMMON /FRE/  FREE(64),IFR,IFRSZ
0017        COMMON /MESS/ MESSEQ
0018        COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
           1    IOFLG,LLFLG,IRSEND
0019        IF (N .NE. 1) GOTO 10
0021        IFR=IFR+1
0022        DO 5 I=1,256
0023          PACK(I,OUTQ)=0
0024      5 CONTINUE
0025        IFULL=0
0026        ISENT=0
0027        IRSEND=0
0028        FREE(IFR)=OUTQ
0029        GOTO 999
0030     10 T2=PACK(256,OUTQ)
0031        T2=T2+1
0032        IF(T2 .LE. RESNLM) GOTO 120
0034        T3=PACK(3,OUTQ)
0035        T4=MOD(T3,64)
0036        IF(T4 .LT. 32) GOTO 110
0038        T5=IGETSP(N)
0039        IF(MESSEQ .EQ. 126) MESSEQ=0
0041        MESSEQ=MESSEQ+1
0042        PACK(1,T5)=0
0043        PACK(2,T5)=MESSEQ
0044        PACK(3,T5)=0
0045        PACK(4,T5)=0
0046        PACK(5,T5)=25
0047        PACK(6,T5)=ISLID
0048        DO 20 I=7,60
0049          PACK(I,T5)="040
0050     20 CONTINUE
0051        DO 30 I=7,9
0052          PACK(I,T5)=LF
0053     30 CONTINUE
```

```
0054          DO 40 I=1,8
0055            PACK(I+9,T5)=LCOMT(I)
0056     40   CONTINUE
0057          DO 50 I=9,16
0058            PACK(I+9,T5)=LCOMT(I)
0059     50   CONTINUE
0060          DO 60 I=17,22
0061            PACK(I+9,T5)=LCOMT(I)
0062     60   CONTINUE
0063          DO 70 I=25,29
0064            PACK(I+7,T5)=LCOMT(I)
0065     70   CONTINUE
0066          ENCODE(3,80,PACK(38,T5)) ISLID
0067     80   FORMAT(I3)
0068          DO 90 I=33,40
0069            PACK(I+11,T5)=LCOMT(I)
0070     90   CONTINUE
0071          ENCODE(3,80,PACK(52,T5))PACK(5,OUTQ)
0072          PACK(55,T5)=LF
0073          PACK(56,T5)=LF
0074          PACK(57,T5)=CR
0075          PACK(58,T5)=ETX
0076          CALL ENSTR(PACK(254,T5),58)
0077          CALL ENQUE(PXINQ,XINQ,T5)
0078          DO 102 I=1,256
0079            PACK(I,OUTQ)=0
0080    102   CONTINUE
0081          ISENT=0
0082          IRSEND=0
0083          IFULL=0
0084          IWRT=0
0085          IFR=IFR+1
0086          FREE(IFR)=OUTQ
0087          GOTO 999
0088     110  PACK(3,OUTQ)=PACK(3,OUTQ)+32
0089          PACK(256,OUTQ)=0
0090          ISENT=0
0091          IFULL=0
0092          IF(IRSEND .EQ. 1) GOTO 999
0094          CALL ENQUE(PXINQ,XINQ,OUTQ)
0095          GOTO 999
0096    120   PACK(256,OUTQ)=T2
0097          IFULL=0
0098          ISENT=0
0099          IF(IRSEND .EQ. 1) GOTO 999
0101          CALL ENQUE(PXINQ,XINQ,OUTQ)
0102    999   RETURN
0103          END
```

```
0001          SUBROUTINE INPTQ(L)
0002          INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
0003          INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005          LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
0006          INTEGER*2 LTIME,IWRTTM
0007          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IOFLG,LLFLG,IRSEND
0009          COMMON         PACK(256,64)
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          IF(L .LT. 1) GOTO 30
0014          LI=L-2
0015          II=INBF(6)
0016          IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0018          T1=IGETSP(N)
0019          DO 10 I=1,LI
0020     10   PACK(I,T1)=INBF(I)
0021          CALL ENSTR(PACK(254,T1),LI)
0022          CALL ENQUE(PINQ,INQ,T1)
0023          N=2
0024          GOTO 40
0025     20   N=1
0026          GOTO 40
0027     30   N=0
0028     40   DO 50 I=1,2
0029            OUTBF(I)=INBF(I)
0030     50   CONTINUE
0031          NN=1
0032          IF(N .EQ. 0) NN=128
0034          OUTBF(3)=NN
0035          OUTBF(4)=0
0036          OUTBF(5)=INBF(6)
0037          OUTBF(6)=ISLID
0038          OUTBF(7)=ETX
0039          OUTBF(9)=LIDFD(INBF(6))
0040          CALL LPOUT(9)
0041          IF(N .NE. 2) GOTO 999
0043          ACKSEQ(II)=INBF(2)
0044    999   RETURN
0045          END
```

```
0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON/DFM/ OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          CC1=INBF(3)
0009          CC2=INBF(4)
0010          IF(CC1 .LT. 0) GOTO 25
0012          IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0014          CC3=MOD(CC1,2)
0015          CC4=MOD(CC1,256)
0016          IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0018          IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0020          CC5=MOD(INBF(3),64)
0021          IF(CC5 .GE. 32) CALL INPTQ(LI)
0023          GOTO 99
0024    20    CC7=MOD(CC1,2)
0025          IF(CC7 .GE. 1) GOTO 30
0027    25    CALL ACKNAK(0)
0028          GOTO 99
0029    30    CALL ACKNAK(1)
0030          GOTO 99
0031    40    CALL INPTQ(LI)
0032    99    CONTINUE
0033          RETURN
0034          END
```

```
0001          SUBROUTINE INIT
0002          REAL*4 RH,RL,AH,AL,VMEAS
0003          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0004          INTEGER*2 STAT,FLWCNT,OUTFCT
0005          INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0006          LOGICAL*1 IDATA,DUM
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0008          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID
0009          INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0010          COMMON /DFM/ OUTBF(256),INBF(256),
             1     IWRTTM,IWRT,STAT,LTIME
0011          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0012          COMMON /MESS/ MESSEQ
0013          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0014          COMMON /FRE/  FREE(64),IFR,IFRSZ
0015          COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0016          COMMON        PACK(256,64)
0017          COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0018          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0019          COMMON /BCOM/ ISWTCH,RH(13,3),RL(13,3),AH(13,3),AL(13,3),
             &     ITERM,LINK,MON,VMEAS(13),ISUB,INUM,IDATA(10),
             &     ICHNSW
0020          CALL RAM(0,8,4)
0021          CALL RAM(0,255,0)
0022          MESSEQ=0
0023          IOFLG=0
0024          IRSEND=0
0025          LLFLG=0
0026          IFRSZ=64
0027          INFLCT=0
0028          ISLID=26
0029          CR="015
0030          LF="012
0031          ETX="003
0032          RESNLM=2
0033          IQLNTH=16
0034          IALTRT=0
0035          FLWCNT=0
0036          IWRTTM=0
0037          TIMLIM=50
0038          SETPRM=4
0039          RSTPRM=32
0040          SETBKP=8
0041          RSTBKP=64
0042          ATIMLM=200
0043          MONTOR=27
0044          IWRT=0
0045          ISWTCH=0
0046          OUTFCT=0
0047          IFULL=0
0048          ISENT=0
```

```
0049          DO 40 I=1,20
0050     40   LIDFD(I)=4
0051          LIDFD(21)=1
0052          LIDFD(22)=3
0053          LIDFD(23)=6
0054          LIDFD(24)=5
0055          LIDFD(25)=7
0056          LIDFD(26)=8
0057          LIDFD(27)=9
0058          LIDFD(28)=2
0059          DO 50 I=29,39
0060     50   LIDFD(I)=0
0061          DO 60 I=40,44
0062     60   LIDFD(I)=1
0063          DO 70 I=45,59
0064     70   LIDFD(I)=0
0065          DO 80 I=60,64
0066     80   LIDFD(I)=2
0067          DO 90 I=65,79
0068     90   LIDFD(I)=0
0069          DO 100 I=80,84
0070    100   LIDFD(I)=4
0071          DO 110 I=85,99
0072    110   LIDFD(I)=0
0073          DO 120 I=100,104
0074    120   LIDFD(I)=5
0075          DO 130 I=105,256
0076    130   LIDFD(I)=0
0077          PINQ(1)=IQLNTH
0078          PINQ(2)=IQLNTH+1
0079          PACKQ(1)=IQLNTH
0080          PACKQ(2)=IQLNTH+1
0081          PXOUTQ(1)=IQLNTH
0082          PXOUTQ(2)=IQLNTH+1
0083          PXINQ(1)=IQLNTH
0084          PXINQ(2)=IQLNTH+1
0085          DO 10 I=1,IFRSZ
0086            PACK(255,I)=0
0087     10   PACK(256,I)=0
0088          DO 20 I=1,IFRSZ
0089     20   FREE(I)=I
0090          DO 30 I=1,256
0091     30   ACKSEQ(I)=256
0092          IFR=IFRSZ
0093          RETURN
0094          END
```

```
0001          SUBROUTINE LINLOS
0002          REAL*8 RM1(5),LIN08,LIN18
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,PINQ
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006          LOGICAL*1 PACK,M1(40),LIN0(8),LIN1(8)
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             &          PACKQ(2),INQ(16),PINQ(2)
0009          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &          IOFLG,LLFLG,IRSEND
0010          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011          COMMON         PACK(256,64)
0012          COMMON /MESS/ MESSEQ
0013          DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0014          DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0015          DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0016          DATA RM1(5)/'DE 26   '/
0017          EQUIVALENCE(LIN0,LIN08)
0018          EQUIVALENCE(LIN1,LIN18)
0019          EQUIVALENCE(M1,RM1)
0020          IS=LLFLG
0021          CALL ENABLE(0)
0022          T1=IGETSP(N)
0023          CALL ENABLE(1)
0024          PACK(1,T1)=0
0025          IF(MESSEQ .EQ. 126) MESSEQ=0
0027          MESSEQ=MESSEQ+1
0028          PACK(2,T1)=MESSEQ
0029          PACK(3,T1)=0
0030          PACK(4,T1)=0
0031          PACK(5,T1)=25
0032          PACK(6,T1)=ISLID
0033          DO 20 I=7,9
0034            PACK(I,T1)=LF
0035    20    CONTINUE
0036          DO 30 I=1,22
0037    30    PACK(I+9,T1)=M1(I)
0038          IF(IS .EQ. 1) GOTO 50
0039          DO 40 I=1,8
0041    40    PACK(I+31,T1)=LIN0(I)
0042          GOTO 70
0043    50    DO 60 I=1,8
0044    60    PACK(I+31,T1)=LIN1(I)
0045    70    DO 80 I=23,40
0046    80    PACK(I+17,T1)=M1(I)
0047          PACK(58,T1)=CR
0048          PACK(59,T1)=LF
0049          PACK(60,T1)=ETX
0050          CALL ENSTR(PACK(254,T1),60)
0051          CALL ENABLE(0)
0052          CALL ENQUE(PXINQ,XINQ,T1)
0053          CALL ENABLE(1)
```

```
0054          LLFLG=0
0055          RETURN
0056          END
```

```
0021          SUBROUTINE B4BSA
0002          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,IDATA,PACK
0003          LOGICAL*1 ISLID,MONTOR,LF,ETX,CR,DUM
0004          INTEGER*2 FL0CNT,OUTPCT,OUTQ,RESNUM
0005          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0006          REAL*4 RB,RL,AB,AL,TMEAS
0007          COMMON /MESS/ MESSEQ
0008          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &     ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0009          COMMON/GLOB/ISENT,FL0CNT,IQLNTH,OUTPCT,IFULL,OUTQ,RESNUM,
             &            IALTBT,ISLID,MONTOR,LF,ETX,CR,DUM
0010          COMMON/BCOM/IS0TOH,RB(13,3),RL(13,3),AB(13,3),AL(13,3),
             &            ITERM,LINK,MON,TMEAS(13),ISUB,INUM,
             &            IDATA(10),ICONSW,ICHAN
0011          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &            IOFLG,LLFLG,IRSEND
0012          COMMON PACK(256,64)
0013          DATA RB/1.2,20.7,3900.0,20.0,-20.0,10.0,19.5,11.15,23.5,
             &     3920.0,19.0,-15.0,21.5,
             &     2.91,39.2,3900.0,19.4,-13.5,13.9,19.0,12.96,39.0,
             &     3900.0,18.5,-16.0,24.5,
             &     4.6,57.0,3900.0,19.5,-16.0,13.8,18.3,14.0,40.0,
             &     3940.0,19.5,-18.0,24.0/
0014          DATA RL/-1.35,-21.0,190.0,-19.0,-100.0,-10.0,-50.0,-11.27,
             &     -24.5,130.0,-20.0,-100.0,-24.0,
             &     -2.8,-39.2,200.0,-19.2,-98.0,-9.0,-57.0,-12.75,-39.0,
             &     270.0,-18.5,-96.0,-24.5,
             &     -4.9,-52.0,180.0,-19.5,-103.0,-10.0,-62.0,-14.9,-40.0,
             &     180.0,-19.5,-102.0,-24.0/
0015          DATA AB/1.1,22.2,3740.0,18.0,-30.0,13.5,18.0,9.85,23.0,3740.0,
             &     18.0,-20.0,-21.0,
             &     2.8,34.0,3776.0,19.1,-15.5,13.1,14.5,13.85,37.5,3700.0,
             &     17.5,-20.0,22.5,
             &     4.3,49.0,3760.0,18.6,-18.0,13.5,17.0,13.0,25.0,3740.0,18.5,
             &     -26.0,19.0/
0016          DATA AL/-1.0,-20.0,400.0,-18.0,-100.0,-8.0,-60.0,-11.25,-23.0,
             &     -190.0,-19.0,188.0,-23.0,
             &     -2.5,-34.0,380.0,-18.7,-92.0,-5.0,-55.0,-12.5,-35.0,
             &     360.0,-17.5,-96.0,-23.0,
             &     -4.5,-48.0,210.0,-18.0,-102.0,-5.0,-55.0,-14.5,-25.0,
             &     380.0,-18.0,-92.0,-19.0/
0017          IF(XINBF(3) .EQ. 'X' .AND. XINBF(4) .EQ. 'X' .AND.
             &     XINBF(5) .EQ. 'X' .AND. XINBF(6) .EQ. 'X') GOTO 60
0018          DECODE(2,1,XINBF(1),ERR=80) ICNT
0019          DECODE(4,2,XINBF(3),ERR=80) TMEAS(1)
0020          DECODE(4,2,XINBF(7),ERR=80) TMEAS(2)
0021          DECODE(4,2,XINBF(11),ERR=80) TMEAS(3)
0022          DECODE(4,2,XINBF(15),ERR=80) TMEAS(4)
0023          DECODE(4,2,XINBF(19),ERR=80) TMEAS(5)
0024          DECODE(4,2,XINBF(23),ERR=80) TMEAS(6)
0025          DECODE(4,2,XINBF(27),ERR=80) TMEAS(7)
0026          DECODE(4,2,XINBF(31),ERR=80) TMEAS(8)
0027          DECODE(4,2,XINBF(35),ERR=80) TMEAS(9)
0028          DECODE(4,2,XINBF(39),ERR=80) TMEAS(10)
```

```
0030          DECODE(4,2,XINBF(43),ERR=80) VMEAS(11)
0031          DECODE(4,2,XINBF(47),ERR=80) VMEAS(12)
0032          DECODE(4,2,XINBF(51),ERR=80) VMEAS(13)
0033          DECODE(2,1,XINBF(55),ERR=80) LINK
0034          DECODE(2,1,XINBF(65),ERR=80) MON
0035        1 FORMAT(A2)
0036        2 FORMAT(A4)
0037          IF(ICHNSW .EQ. 1) GOTO 70
0039       10 IF(ISUB .GT. 3) ISUB=0
0041          ISUB=ISUB+1
0042          DO 20 INUM=1,13
0043          IF((AL(INUM,ISUB) .LE. VMEAS(INUM)) .AND.
             &    (AL(INUM,ISUB) .GE. VMEAS(INUM))) GOTO 20
0045          GOTO 40
0046       20 CONTINUE
0047          IOFLG=0
0048          RETURN
0049       40 IF((AH(INUM,ISUB) .LT. VMEAS(INUM) .AND.
             &    RH(INUM,ISUB) .GE. VMEAS(INUM)) .OR.
             &   (RL(INUM,ISUB) .LE. VMEAS(INUM) .AND.
             &    AL(INUM,ISUB) .GT. VMEAS(INUM))) GOTO 50
0051          IF(ISWTCH .EQ. 0) GOTO 80
0053          CALL SNDRPT(1)                          !SEND RED REPORT
0054          IOFLG=0
0055          RETURN
0056       50 IF(ISWTCH .EQ. 0) GOTO 80
0058          CALL SNDRPT(2)                          !SEND AMBER REPORT
0059          IOFLG=0
0060          RETURN
0061       60 IF(ISWTCH .EQ. 0) GOTO 80
0063          CALL SNDRPT(3)                          !SEND ALARM REPORT
0064          GOTO 80
0065       70 CALL ENABLE(0)
0066          K1=IGETSP(N)
0067          CALL ENABLE(1)
0068          DO 72 I=1,256
0069           PACK(I,K1)=0
0070       72 CONTINUE
0071          IF(MESSEQ .EQ. 126) MESSEQ=0
0073          MESSEQ=MESSEQ+1
0074          PACK(1,K1)=0
0075          PACK(2,K1)=MESSEQ
0076          PACK(3,K1)=0
0077          PACK(4,K1)=0
0078          PACK(5,K1)=25
0079          PACK(6,K1)=ISLID
0080          PACK(7,K1)=CR
0081          PACK(8,K1)=LF
0082          ENCODE(4,74,PACK(9,K1)) ICHAN
0083          PACK(13,K1)="040
0084          ENCODE(12,76,PACK(14,K1)) VMEAS(1)
0085          PACK(26,K1)="040
0086          ENCODE(12,76,PACK(27,K1)) VMEAS(2)
0087          PACK(39,K1)="040
```

```
0088          ENCODE(12,76,PACK(40,K1)) VMEAS(3)
0089          PACK(52,K1)="040
0090          ENCODE(12,76,PACK(53,K1)) VMEAS(4)
0091          PACK(65,K1)="040
0092          ENCODE(12,76,PACK(66,K1)) VMEAS(5)
0093          PACK(78,K1)="040
0094          ENCODE(12,76,PACK(79,K1)) VMEAS(6)
0095          PACK(91,K1)=CR
0096          PACK(92,K1)=LF
0097          ENCODE(12,76,PACK(93,K1)) VMEAS(7)
0098          PACK(105,K1)="040
0099          ENCODE(12,76,PACK(106,K1)) VMEAS(8)
0100          PACK(118,K1)="040
0101          ENCODE(12,76,PACK(119,K1)) VMEAS(9)
0102          PACK(131,K1)="040
0103          ENCODE(12,76,PACK(132,K1)) VMEAS(10)
0104          PACK(144,K1)="040
0105          ENCODE(12,76,PACK(145,K1)) VMEAS(11)
0106          PACK(157,K1)="040
0107          ENCODE(12,76,PACK(158,K1)) VMEAS(12)
0108          PACK(170,K1)="040
0109          ENCODE(12,76,PACK(171,K1)) VMEAS(13)
0110          PACK(183,K1)=CR
0111          PACK(184,K1)=LF
0112          PACK(185,K1)=ETX
0113    74    FORMAT(I4)
0114    76    FORMAT(F12.6)
0115          CALL ENSTR(PACK(254,K1),185)
0116          CALL ENABLE(0)
0117          CALL ENQUE(PXINQ,XINQ,K1)
0118          CALL ENABLE(1)
0119          ICHNSW=0
0120    80    IOFLG=0
0121          RETURN
0122          END
```

```
0001          SUBROUTINE SNDRPT(IC)
0002          REAL*4 RH,RL,AH,AL,VMEAS
0003          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0004          LOGICAL*1 PACK,IDATA,ISLID,MONTOR,LF,ETX,CR,DUM
0005          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0006          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0007          COMMON/BCOM/ISWITCH,RH(13,3),RL(13,3),AH(13,3),AL(13,3),
             &            ITERM,LINK,MON,VMEAS(13),ISUB,INUM,
             &            IDATA(10),ICHNSW,ICHAN
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &    IOFLG,LLFLG,IRSEND
0009          COMMON/MESS/MESSEQ
0010          COMMON PACK(256,64)
0011          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0012          COMMON/GLOB/ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             &          RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          CALL ENABLE(0)
0014          K1=IGETSP(N)
0015          CALL ENABLE(1)
0016          DO 10 I=1,256
0017           PACK(I,K1)=0
0018    10    CONTINUE
0019          IF(MESSEQ .EQ. 126) MESSEQ=0
0021          MESSEQ=MESSEQ+1
0022          PACK(1,K1)=0
0023          PACK(2,K1)=MESSEQ
0024          PACK(3,K1)=0
0025          PACK(4,K1)=0
0026          PACK(5,K1)=MONTOR
0027          PACK(6,K1)=ISLID
0028          IF(IC .EQ. 3) GOTO 50
0030          DO 20 I=7,14
0031            PACK(I,K1)=XINBF(I+50)
0032     20   CONTINUE
0033          ENCODE(2,30,PACK(15,K1)) LINK
0034    30    FORMAT(A2)
0035    40    FORMAT(A4)
0036          PACK(17,K1)=IC
0037          ENCODE(4,40,PACK(18,K1)) VMEAS(INUM)
0038          ENCODE(2,30,PACK(22,K1)) MON
0039          PACK(24,K1)=ITERM
0040          CALL ENSTR(PACK(254,K1),24)
0041          GOTO 70
0042    50    DO 60 I=7,10
0043           PACK(I,K1)='X'
0044    60    CONTINUE
0045          DO 62 I=11,14
0046            PACK(I,K1)=XINBF(I)
0047     62   CONTINUE
0048          PACK(15,K1)=CR
0049          PACK(16,K1)=LF
0050          PACK(17,K1)=ETX
0051          CALL ENSTR(PACK(254,K1),17)
```

```
0052    70    CALL ENABLE(0)
0053          CALL ENQUE(PXINQ,XINQ,K1)
0054          CALL ENABLE(1)
0055          RETURN
0056          END
```

```
0001          SUBROUTINE BWSALP
0002          LOGICAL*1 IDATA
0003          REAL*4 RH,RL,AH,AL,VMEAS
0004          COMMON/BCOM/ISWTCH,RH(13,3),RL(13,3),AH(13,3),AL(13,3),
             &            ITERM,LINK,MON,VMEAS(13),ISUB,INUM,
             &            IDATA(10),ICHNSW,ICHAN
0005          IF(IDATA(1) .EQ. "117 .AND.
             &  IDATA(2) .EQ. "116) GOTO 50
0007          IF(IDATA(1) .EQ. "117 .AND.
             &  IDATA(2) .EQ. "106 .AND.
             &  IDATA(3) .EQ. "106) GOTO 60
0009          IF(IDATA(1) .EQ. "115) GOTO 70
0011          GOTO 80
0012    50    ISWTCH=1
0013          ITERM=IDATA(4)
0014          RETURN
0015    60    ISWTCH=0
0016          RETURN
0017    70    DECODE(4,75,IDATA(2),ERR=80) ICHAN
0018    75    FORMAT(I4)
0019          CALL SOUT(ICHAN)
0020          ICHNSW=1
0021    80    RETURN
0022          END
```

```
                                        .TITLE  FDM.MACRO
                                        .SBTTL  NODE 26
                                        .IDENT  /V3.0/
                                        .GLOBL  LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
                                        .GLOBL  WTOKEN,RAM,STATUS,STATB0
                                        .GLOBL  LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
                                        .NLIST  CND
000000                                  .PSECT

000000                          R0      =%0
000001                          R1      =%1
000002                          R2      =%2
000003                          R3      =%3
000004                          R4      =%4
000005                          R5      =%5
000006                          SP      =%6
000007                          PC      =%7

                                ;INTERFACE ADDRESS

172410                          BAR=    172410
172412                          WCR=    172412
172414                          CSR=    172414
172416                          IOBUF=  172416
172416                          OPREG=  172416
177560                          HRCSR=  177560          ;CHANGED
177562                          HRBUF=  177562
177564                          HXCSR=  177564
177566                          HXBUF=  177566

                                ;INTERFACE OPTIONS

000001                          XCIO=1
000000                          XSIO=0
000001                          XGIO=1
000000                          NDXX=0
000001                          ND24=1
000000                          NHOST=0
000001                          DHOST=1

                                ;LOCAL VARIBLES

000002                          ARG1=   2
000004                          ARG2=   4
000006                          ARG3=   6
000010                          ARG4=   10

100000                          BIT15=  100000
040000                          BIT14=  40000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
000020                          BIT04=  20
000010                          BIT03=  10
000004                          BIT02=  4
```

```
FDM.MACRO       MACRO V03.02B1-OCT-79 09:01:28 PAGE 1-1
NODE 26

     66         000002                          BIT01=  2
     67         000001                          BIT00=  1
     68
```

```
                                ;*************************************************************
                                ;**** PROCEDURE MASTER START UP                              *
                                ;*************************************************************

000000  012700  000340          MASTER: MOV     #340,   R0                      ;PRI=7
000004  106400                          MTPS    R0
000006  012700  000124                  MOV     #124,   R0
000012  012720  000430'                 MOV     #LIO,   (R0)+                   ;LIU HANDLER
000016  012710  000340                  MOV     #340,   (R0)                    ;PRI=7
000022  000005                          RESET                                   ;RESET BUS
000024  012700  000060                  MOV     #60,    R0
000030  012720  002470'                 MOV     #SIO,   (R0)+                   ;SIG INTERFACE HANDLER
000034  012720  000340                  MOV     #340,   (R0)+
000040  012700  000100                  MOV     #100,   R0                      ;CLOCK VECTOR
000044  012710  001654'                 MOV     #TIME,  (R0)                    ;CLOCK
000050  042737  040100  172414          BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000056  004767  000014                  JSR     PC,     LIUINT                  ;CLEAR LIU
000062  005037  172414                  CLR     @#CSR                           ;CLEAR BLIUI CSR
000066  012700  000000                  MOV     #000,   R0                      ;PRI=0
000072  106400                          MTPS    R0                              ;CPU DOWN
000074  000207                          RTS     PC

                                ;*************************************************************
                                ;**** PROCEDURE INITIALIZE                                  *
                                ;*************************************************************

                                        ;CLEAR ACRAM

000076  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000104  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000112  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000116  100375                          BPL     .-4                             ;NO RETRY
000120  105037  172414                  CLRB    @#CSR
000124  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
000132  012700  177400                  MOV     #-256., R0                      ;COUNTER
000136  012737  004407  172416  1$:     MOV     #2311., @#OPREG                 ;WRITE A NULL
000144  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000150  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
000152  105037  172414                  CLRB    @#CSR
000156  005200                          INC     R0                              ;COUNT+1
000160  001366                          BNE     1$

                                        ;CLEAR INPUT/OUTPUT BUFFERS

000162  012701  010410                  MOV     #10410, R1                      ;RDBUFADR CMD
000166  012700  010440                  MOV     #10440, R0                      ;SEL INBUF0
000172  004767  000106                  JSR     PC,     ZEROBP                  ;SET POINTER=0
000176  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000204  105737  172414                  TSTB    @#CSR                           ;GOOD READ
000210  100375                          BPL     .-4                             ;NO RETRY
000212  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
000216  012701  010610                  MOV     #10610, R1                      ;RDBUFADR COMMAND
000222  012700  010640                  MOV     #10640, R0                      ;SEL INBUF1 COMMAND
000226  004767  000052                  JSR     PC,     ZEROBP                  ;POINTER=0
```

```
    102 000232  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
    103 000240  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    104 000244  100375                          BPL     .-4                     ;NO RETRY
    105 000246  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    106 000252  012701  010510                  MOV     #10510, R1              ;RDBUFADR COMMAND
    107 000256  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
    108 000262  004767  000016                  JSR     PC,     ZEROBP
    109 000266  012701  010710                  MOV     #10710, R1              ;RDBUFADR COMMAND
    110 000272  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
    111 000276  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
    112 000302  000440                          BR      STAT$                   ;GO CLEAR STATUS
    113 000304  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
    114 000310  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
    115 000316  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    116 000322  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    117 000324  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    118 000330  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
    119 000334  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
    120 000340  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
    121 000344  022702  000000          2$:     CMP     #0,     R2              ;POINTER=0?
    122 000350  001412                          BEQ     3$                      ;YES RECHECK
    123 000352  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
    124 000360  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    125 000364  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    126 000366  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    127 000372  005302                          DEC     R2                      ;LOCAL POINTER-1
    128 000374  000763                          BR      2$                      ;BR  NTIL 0
    129 000376  010037  172416          3$:     MOV     R0,     @#OPREG         ;SEL BUFFER
    130 000402  000207                          RTS     PC                      ;RETURN
    131
    132                                         ;READ AND CLEAR STATUS
    133
    134 000404  012737  010400  172416  STAT$:  MOV     #4352., @#OPREG         ;WCR:RS
    135 000412  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
    136 000420  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
    137 000426  000207                          RTS     PC
    138
    139
```

```
                                ;*********************************************************
                                ;**** PROCEDURE LIU-HANDLER                              *
                                ;*********************************************************


000430  010046                LIO:    MOV     R0,     -(SP)                   ;SAVE R0
000432  010146                        MOV     R1,     -(SP)                   ;SAVE R1
000434  010246                        MOV     R2,     -(SP)                   ;SAVE R2
000436  010346                        MOV     R3,     -(SP)                   ;SAVE R3
000440  010446                        MOV     R4,     -(SP)                   ;SAVE R4
000442  010546                        MOV     R5,     -(SP)                   ;SAVE R5
000444  005737  172414                TST     @#CSR                           ;? LIU INTERRUPTED
000450  100153                        BPL     RTI$                            ;NO
000452  042737  040000  172414        BIC     #BIT14, @#CSR                   ;YES/DISABLE LIU
000460  012737  010400  172416        MOV     #4352., @#OPREG                 ;WCR : RS(0)
000466  012737  002400  172416        MOV     #1280., @#OPREG                 ;RD
000474  013767  172416  000502        MOV     @#IOBUF, CAUSE
000502  042767  177400  000474        BIC     #177400, CAUSE                  ;CLEAR UNUSED BITS

                                ;PREFORM OPERATION DEPENDING ON BITS

000510  132767  000004  000466  B2$:  BITB    #BIT02, CAUSE                   ;OV-FL
000516  001406                        BEQ     B3$
000520  012702  010410                MOV     #4360., R2                      ;RDBUFADR IN0
000524  012703  010440                MOV     #4384., R3                      ;SEL INBUF0
000530  004767  000270                JSR     PC,     EMBF                    ;GO EMPTY
000534  132767  000010  000442  B3$:  BITB    #BIT03, CAUSE                   ;OV-FL
000542  001406                        BEQ     B4$
000544  012702  010610                MOV     #4488., R2                      ;RDBUFADR IN1
000550  012703  010640                MOV     #4512., R3                      ;SEL INBUF1
000554  004767  000244                JSR     PC,     EMBF                    ;GO EMPTY
000560  132767  000020  000416  B4$:  BITB    #BIT04, CAUSE                   ;LINE-LOSS PRIMARY
000566  001407                        BEQ     B5$
000570  132767  000200  000406  B7$:  BITB    #BIT07, CAUSE                   ;LATCH ON
000576  001403                        BEQ     B5$                             ;YES LOOP AROUND
000600  012767  000002  002002'      MOV     #2,     LLFLG                   ;FLAG PRIMARY SW. FAIL
000606  132767  000040  000370  B5$:  BITB    #BIT05, CAUSE                   ;BACKUP LINE LOSS
000614  001403                        BEQ     B6$
000616  012767  000001  002002'      MOV     #1,     LLFLG                   ;FLAG BACKUP SW. FAIL
000624  132767  000100  000352  B6$:  BITB    #BIT06, CAUSE                   ;WRITE TOKEN DETECT
000632  001404                        BEQ     B0$
000634  005067  001000'              CLR     IWRTTM                          ;CLEAR TOKEN VAR
000640  005067  001002'              CLR     IWRT

                                ;BUFFER FULL

000644  132767  000001  000332  B0$:  BITB    #BIT00, CAUSE                   ;INBUF0 FULL
000652  001423                        BEQ     B1$                             ;NO
000654  012701  000400'              MOV     #INBF,  R1                      ;BUFFER ADDRESS
000660  012702  010410                MOV     #4360., R2                      ;RDBUFADR COMM
000664  012703  010440                MOV     #4384., R3                      ;SEL BUF
000670  012704  000001                MOV     #BIT00, R4                      ;CRC BIT
000674  004767       54 000700  012705  001206'                  MOV     #AREA,  R5                      ;DATA LINK A
REA
000704  012767  000001  000274        MOV     #1,     AREA                    ;1 VARIBLE
000712  010267  000274                MOV     R2,     DATA                    ;CRC OR BYTE COUNT
000716  004767  000000G               JSR     PC,     LPINPT                  ;CALL FORTRAN QUE'ER
```

```
     58 000722  132767  000002  000254  B1$:    BITB    #BIT01, CAUSE           ;INBUF1 FULL
     59 000730  001423                           BEQ     RTI$                    ;NO WERE DONE
     60 000732  012701  000400'                  MOV     #INBF,  R1              ;BUFFER ADDRESS
     61 000736  012702  010610                   MOV     #4488., R2              ;RDBUFADR COMM
     62 000742  012703  010640                   MOV     #4512., R3              ;SEL INBUF1
     63 000746  012704  000002                   MOV     #BIT01, R4              ;CRC BIT
     64 000752  004767  000046                   JSR     PC,     EMBF            ;GO EMPTY BUFFER
     65 000756  012705  001206'                  MOV     #AREA,  R5              ;DATA LINK AREA
     66 000762  012767  000001  000216           MOV     #1,     AREA            ;ONE VARIBLE
     67 000770  010267  000216                   MOV     R2,     DATA            ;CRC OR BYTE COUNT
     68 000774  004767  000000G                  JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
     69
     70 001000  012605                   RTI$:   MOV     (SP)+,  R5              ;RESTORE REGISTERS
     71 001002  012604                           MOV     (SP)+,  R4
     72 001004  012603                           MOV     (SP)+,  R3
     73 001006  012602                           MOV     (SP)+,  R2
     74 001010  012601                           MOV     (SP)+,  R1
     75 001012  012600                           MOV     (SP)+,  R0
     76 001014  052737  040000  172414           BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
     77 001022  000002                           RTI                             ;RETURN FROM INTERRUPT
     78
     79 001024  012737  010600  172416  EMBF:    MOV     #4480., @#OPREG         ;READ STATUS 1
     80 001032  012737  002400  172416           MOV     #1280., @#OPREG         ;RS
     81 001040  013700  172416                   MOV     @#IOBUF, R0             ;FETCH STATUS
     82 001044  130400                           BITB    R4,R0                   ;GOOD CRC
     83 001046  001002                           BNE     CRCOK
     84 001050  012704  177777                   MOV     #-1,    R4              ;NO FLAG
     85 001054  010237  172416           CRCOK:  MOV     R2,     @#OPREG         ;RDBUFADR
     86 001060  012737  001400  172416           MOV     #768.,  @#OPREG         ;RD
     87 001066  105737  172414                   TSTB    @#CSR                   ;GOOD RD
     88 001072  100375                           BPL     .-4
     89 001074  013702  172416                   MOV     @#IOBUF, R2             ;REPLACE WITH POINTER
     90 001100  042702  177400                   BIC     #177400, R2             ;CLEAR MST BITS
     91 001104  010200                           MOV     R2,     R0              ;SAVE IT
     92 001106  005400                           NEG     R0                      ;2'S COMP
     93 001110  010037  172412                   MOV     R0,     @#WCR           ;BYTE COUNT NOW
     94 001114  010137  172410                   MOV     R1,     @#BAR           ;ADDRESS IN MEMORY
     95 001120  010337  172416                   MOV     R3,     @#OPREG         ;SEL BUFFER
     96 001124  012737  001400  172416           MOV     #768.,  @#OPREG         ;FALSE RD
     97 001132  105737  172414                   TSTB    @#CSR                   ;DONE ON
     98 001136  100375                           BPL     .-4                     ;NO LOOP
     99 001140  012737  021000  172416           MOV     #8704., @#OPREG         ;FIRE DMA
    100 001146  000240                           NOP                             ;DELAY
    101 001150  105737  172414                   TSTB    @#CSR                   ;GOOR DMA
    102 001154  100401                           BMI     DMAOK
    103 001156  000240                           NOP                             ;ERROR IF HERE
    104 001160  012737  004400  172416  DMAOK:   MOV     #2304., @#OPREG         ;FALSE WD
    105 001166  105737  172414                   TSTB    @#CSR                   ;GOOD WD
    106 001172  100375                           BPL     .-4
    107 001174  005704                           TST     R4                      ;WAS CRC OK
    108 001176  100001                           BPL     END$                    ;YES
    109 001200  010402                           MOV     R4,     R2              ;NO FLAG IT
    110 001202  000207                  END$:    RTS     PC                      ;RETURN
    111 001204  000000                  CAUSE:   .WORD   0                       ;STA US BYTE 0 HOLDER
    112 001206  000000                  AREA:    .WORD   0
    113 001210  001212'                          .WORD   DATA
    114 001212  000000                  DATA:    .WORD   0
```

```
                                .ENABLE LSB
                        ;*************************************************
                        ;**** PROCEDURE ENABLE                          *
                        ;*************************************************
 001214  017501  000002        ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
 001220  022701  000001                CMP     #1,     R1
 001224  001404                        BEQ     E1$
 001226  042737  040000  172414 E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
 001234  000207                        RTS     PC
 001236  052737  040000  172414 E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
 001244  000207                        RTS     PC

                                ;FORTRAN ENCODE/DECODE MACRO

 001246  017575  000004  000002 ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
 001254  000207                        RTS     PC
 001256  017575  000002  000004 DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECODE BYTE COUNT
 001264  000207                        RTS     PC


                        ;*************************************************
                        ;**** PROCEDURE LINE SWITCH                     *
                        ;*************************************************

                                ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                ;
                                ;04=SET PRIMARY LINE
                                ;32=RESET PRIMARY LINR
                                ;08=SET BACKUP LINE
                                ;64=RESET BACKUP LINE

 001266  017501  000002        SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
 001272  012737  010420  172416        MOV     #4368., @#OPREG                 ;WCR : MODSTAT
 001300  062701  004400                ADD     #2304., R1
 001304  010137  172416                MOV     R1,     @#OPREG                 ;RD
 001310  105737  172414                TSTB    @#CSR                           ;VALID WRITE
 001314  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
 001316  105037  172414                CLRB    @#CSR                           ;CLEAR DONE BIT
 001322  000207                        RTS     PC
```

```
                                ;*************************************************************
                                ;**** PROCEDURE STATUS                                       *
                                ;*************************************************************

                                        ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
                                        ;
                                        ;X=0 STATUS BYTE 0
                                        ;X=1 STATUS BYTE 1

001324  017501  000002          STATUS: MOV     @ARG1(R5), R1                   ;WHICH BYTE
001330  022701  000001                  CMP     #1,     R1                      ;COMPARE
001334  001415                          BEQ     2$
001336  012737  010400  172416  1$:     MOV     #4352., @#OPREG                 ;WCR : RS(0)
001344  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001352  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001356  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001362  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001366  000207                          RTS     PC
001370  012737  010600  172416  2$:     MOV     #4480., @#OPREG                 ;WCR : RS(1)
001376  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001404  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001410  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001414  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001420  000207                          RTS     PC
001422  012737  010400  172416  STATB0: MOV     #4352., @#OPREG                 ;STATUS BYTE 1
001430  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001436  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001442  042701  177400                  BIC     #177400, R1
001446  132701  000010                  BITB    #BIT03, R1                      ;BUFFER FULL
001452  001404                          BEQ     3$
001454  012775  000001  000002          MOV     #1,     @ARG1(R5)               ;LOAD RETURN
001462  000207                          RTS     PC
001464  012775  000000  000002  3$:     MOV     #0,     @ARG1(R5)               ;LOAD RETURN
001472  000207                          RTS     PC
```

```
                                ;*********************************************************
                                ;**** PROCEDURE WRITE TOKEN (WTOKEN)                     *
                                ;*********************************************************

                                        ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP

001474  012737  010600  172416  WTOKEN: MOV     #4480., @#OPREG                 ;READ STATUS(1)
001502  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001510  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001514  132701  000010                  BITB    #BIT03, R1                      ;BUFFER LOADED
001520  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001522  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001530  012737  004423  172416          MOV     #2323., @#OPREG                 ;MARK OB0F/OB1F/WTCMD
001536  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001542  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001544  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001550  000207                          RTS     PC
001552  012737  010740  172416  WTKG:   MOV     #4576., @#OPREG                 ;SELECT BUFFER 1
001560  012737  001652'  172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001566  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001572  005402                          NEG     R2                              ;FOR AT IT
001574  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001600  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001606  000240                          NOP                                     ;DELAY
001610  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001614  000240                          NOP                                     ;ERROR IF HERE
001616  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001622  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001630  012737  004422  172416          MOV     #2322., @#OPREG                 ;MARK OB1F/WTCMD
001636  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001642  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001644  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001650  000207                          RTS     PC                              ;RETURN
001652      000     377         WTD:    .BYTE 0,377                             ;A WTOKEN


                                ;*********************************************************
                                ;**** PROCEDURE TIMER (TIME)                             *
                                ;*********************************************************

001654  005267  001006'          TIME:   INC     LTIME                           ;TIME+1
001660  022767  077777  001006'          CMP     #077777, LTIME                  ;?TIME FULL
001666  001002                          BNE     4$
001670  005067  001006'                  CLR     LTIME                           ;CLEAR CLOCK
001674  000002                  4$:     RTI
```

```
                                ;*********************************************************
                                ;**** PROCEDURE ACRAM                                    *
                                ;*********************************************************

                                        ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
                                        ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
                                        ;
                                        ;0110   =NREAD
                                        ;0100   =DREAD
                                        ;0111   =NULL
                                        ;0000   =WTOKEN


001676  017500  000002         RAM:    MOV     @ARG1(R5), R0                   ; RAM
001702  022700  000000                  CMP     #0,     R0                      ;WHICH OPERATION
001706  001035                         BNE     RDRAM                           ;READ OP
001710  017500  000004         WTRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS
001714  017501  000006                  MOV     @ARG3(R5), R1                   ;WRITE DATA
001720  012737  010402  172416          MOV     #4354., @#OPREG                 ;SEL LDACTR
001726  062700  004400                  ADD     #2304., R0                      ;WD/DATA (ADDR)
001732  010037  172416                  MOV     R0,     @#OPREG                 ;WRITE DATA
001736  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001742  100375                         BPL     .-4                             ;NO LOOP UNTIL READY
001744  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001750  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
001756  062701  004400                  ADD     #2304., R1                      ;WD/DATA (CMD)
001762  010137  172416                  MOV     R1,     @#OPREG                 ;WRITE
001766  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001772  100375                         BPL     .-4                             ;NO LOOP UNTIL READY
001774  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002000  000207                         RTS     PC
002002  017500  000004         RDRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS TO READ
002006  012737  010402  172416          MOV     #4354., @#OPREG                 ;SEL LDACTR
002014  062700  004400                  ADD     #2304., R0                      ;WD/DATA (ADDR)
002020  010037  172416                  MOV     R0,     @#OPREG                 ;WRITE DATA
002024  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
002030  100375                         BPL     .-4                             ;NO LOOP UNTIL READY
002032  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002036  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
002044  012737  001400  172416          MOV     #768.,  @#OPREG                 ;READ DATA
002052  105737  172414                  TSTB    @#CSR                           ;VALID READ
002056  100375                         BPL     .-4                             ;NO LOOP UNTIL READY
002060  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002064  013700  172416                  MOV     @#IOBUF, R0                     ;FET H DATA
002070  042700  177760                  BIC     #177760, R0                     ;CLEAR BITS
002074  010075  000006                  MOV     R0, @ARG3(R5)                   ;RTN DATA
002100  000207                         RTS     PC
```

```
                                .ENABLE LSB
                        ;*****************************************************
                        ;**** PROCEDURE WRITE LOOP (LPOUT)                  *
                        ;*****************************************************

002102  017502  000002          LPOUT:  MOV     @ARG1(R5), R2           ;BYTE COUNT
002106  012700  000340                  MOV     #340,   R0              ;PRI=7
002112  106400                          MTPS    R0
002114  012737  000000' 172410          MOV     #OUTBF, @#BAR           ;BUS ADDRESS
002122  005402                          NEG     R2                      ; 2'S COMP COUNT
002124  010237  172412                  MOV     R2,     @#WCR           ;COUNT
002130  012737  010540  172416          MOV     #4448., @#OPREG         ;OB0 COMMAND
002136  012737  024000  172416          MOV     #10240., @#OPREG        ;DMA GO
002144  000240                          NOP                             ;INTERFACE TIME
002146  105737  172414                  TSTB    @#CSR                   ;DMA OK
002152  000240                          NOP                             ;ERROR IF HERE
002154  105037  172414                  CLRB    @#CSR
002160  012737  010740  172416          MOV     #4576., @#OPREG         ;OB1 COMMAND
002166  012737  001652' 172410          MOV     #WIL,   @#BAR           ;ADDRESS OF WRITE TOKEN
002174  012702  000002                  MOV     #2,     R2              ;BYTE COUNT
002200  005402                          NEG     R2                      ;FORMAT FOR BLIUI
002202  010237  172412                  MOV     R2,     @#WCR           ;DMA BYTE COUNT
002206  012737  024000  172416          MOV     #10240.,@#OPREG         ;FIRE DMA
002214  000240                          NOP                             ;DELAY
002216  105737  172414                  TSTB    @#CSR                   ;GOOD DMA
002222  000240                          NOP                             ;ERROR IF NOT
002224  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
002230  012737  010420  172416          MOV     #4368., @#OPREG         ;MODSTAT
002236  012737  004403  172416          MOV     #2307., @#OPREG         ;BUFFERS FULL
002244  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
002250  100375                          BPL     .-4                     ;NO LOOP UNTIL
002252  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
002256  012700  000000                  MOV     #000,   R0              ;PRI=0
002262  106400                          MTPS    R0                      ;LOWER CPU
002264  000207                          RTS     PC
```

```
                                ;*****************************************************
                                ;**** PROCEDURE READ BUFFER POINTER                  *
                                ;*****************************************************

002266  017500  000002  RDPNT:  MOV     @ARG1(R5), R0           ;FETCH COMMAND
002272  022700  000000          CMP     #0,     R0              ;INBUF0 ?
002276  001004                  BNE     1$                      ;NO
002300  012700  010410          MOV     #4360., R0              ;RDBUFADR IN0
002304  000167  000046          JMP     4$
002310  022700  000001  1$:     CMP     #1,     R0              ;INBUF1 ?
002314  001004                  BNE     2$                      ;NO
002316  012700  010610          MOV     #4488., R0              ;RDBUFADR IN1
002322  000167  000030          JMP     4$
002326  022700  000002  2$:     CMP     #2,     R0              ;OUTBUF0 ?
002332  001004                  BNE     3$                      ;NO
002334  012700  010510          MOV     #4424., R0              ;RDBUFADR OUT0
002340  000167  000012          JMP     4$
002344  022700  000004  3$:     CMP     #4,     R0              ;OUTBUF1 ?
002350  001020                  BNE     5$                      ;NO RETURN
002352  012700  010710          MOV     #4552., R0              ;RDBUFADR OUT1
002356  010037  172416  4$:     MOV     R0,     @#OPREG         ;RDBUFADR
002362  012737  001400  172416  MOV     #768.,  @#OPREG         ;RD
002370  105737  172414          TSTB    @#CSR                   ;READY
002374  100375                  BPL     .-4                     ;NO LOOP UNTIL
002376  013700  172416          MOV     @#IOBUF, R0             ;FETCH POINTER
002402  042700  177400          BIC     #177400, R0             ;CLEAR MST
002406  010075  000004          MOV     R0,     @ARG2(R5)       ;RETURN POINTER
002412  000207          5$:     RTS     PC
```

```
 1                                ;*****************************************************
 2                                ;**** PROCEDURE RESTART                             *
 3                                ;*****************************************************
 4
 5                                        ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
 6                                        ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
 7                                        ;CALL RSTART(2) -LOAD MODE(173000)
 8
 9 002414  017500  000002       RSTART: MOV     @ARG1(R5), R0                   ;MODE
10 002420  022700  000000       RS0:    CMP     #0,     R0                      ;LOAD ADDRESS
11 002424  001002                       BNE     RS1
12 002426  000000                       HALT
13 002430  000207                       RTS     PC
14 002432  022700  000001       RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
15 002436  001005                       BNE     RS2
16 002440  004767  175432               JSR     PC,     LIUINT                  ;INT LIU FIRST
17 002444  012700  000040       G4$:    MOV     #40,    R0                      ;START ADDRESS
18 002450  000110                       JMP     (R0)
19 002452  022700  000002       RS2:    CMP     #2,     R0                      ;LOAD MODE
20 002456  001002                       BNE     RST
21 002460  000177  000002               JMP     @ROM
22 002464  000207               RST:    RTS     PC
23 002466  173000               ROM:    .WORD   173000
24
25
26
```

```
                                        .ENABLE LSB
                                ;*****************************************************
                                ;****  PROCEDURE SIG INTERFACE                        *
                                ;*****************************************************

002470  010046                  SIO::   MOV     R0,     -(SP)                   ;SAVE REGISTERS
002472  010146                          MOV     R1,     -(SP)
002474  010246                          MOV     R2,     -(SP)
002476  012700  000400'                MOV     #XINBF, R0                      ;BUFFER ADDRESS
002502  012701  000120                  MOV     #80.,   R1                      ;BYTE COUNT
002506  105737  177560          1$:     TSTB    @#HRCSR                         ;PORT READY
002512  100375                          BPL     .-4                             ;LOOP UNTIL
002514  113702  177562                  MOVB    @#HRBUF, R2                     ;FETCH BYTE
002520  042702  177400                  BIC     #177400, R2                     ;CLEAR MST
002524  010267  002000'                MOV     R2,     IOFLG                   ;ITS THE BYTE COUNT
002530  110220                          MOVB    R2,     (R0)+
002532  005301                          DEC     R1                              ;COUNT-1
002534  105737  177560          2$:     TSTB    @#HRCSR                         ;PORT READY
002540  100375                          BPL     .-4                             ;LOOP UNTIL
002542  113702  177562                  MOVB    @#HRBUF, R2                     ;FETCH BYTE
002546  042702  177400                  BIC     #177400, R2                     ;CLEAR MST
002552  110220                          MOVB    R2,     (R0)+                   ;STORE IT
002554  005301                          DEC     R1
002556  001366                          BNE     2$                              ;READ 39 BYTES
002560  042737  000100  177560          BIC     #100,   @#HRCSR                 ;NO MORE UNTIL FORTRAN
002566  012602          3$:     MOV     (SP)+,  R2                              ;RETORE REGISTERS
002570  012601                  MOV     (SP)+,  R1
002572  012600                  MOV     (SP)+,  R0
002574  000002                  RTI

                                ;SEND OVER TO SIG

002576  017500  000002          SOUT::  MOV     @ARG1(R5),R0                    ;BYTE COUNT
002602  105737  177564                  TSTB    @#HXCSR                         ;READY TO SNED
002606  100375                          BPL     .-4                             ;LOOP UNTIL
002610  110037  177566                  MOVB    R0,     @#HXBUF                 ;SEND IT
002614  000207                          RTS     PC


                                ;REPORT NODE READY

002616  105737  177564          READY:: TSTB    @#HXCSR                         ;PORT READY
002622  100375                          BPL     .-4                             ;LOOP UNTIL
002624  112737  000122  177566          MOVB    #122, @#HXBUF                   ;SEND R
002632  052737  000100  177560          BIS     #100, @#HRCSR                   ;ENABLE INTERRUPTS
002640  000207                          RTS     PC                              ;RETURN
```

```
                        ;*********************************************************
                        ;****       COMMON DATA AREA                              *
                        ;*********************************************************

000000                          .PSECT  DFM,RW,D,GBL,REL,OVR

000000                  OUTBF:  .BLKB   256.
000400                  INBF:   .BLKB   256.
001000                  IWRTTM: .BLKW
001002                  IWRT:   .BLKW
001004                  STAT:   .BLKW
001006                  LTIME:  .BLKW
                                .EVEN

000000                          .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                  XOUTBF: .BLKB   256.
000400                  XINBF:  .BLKB   256.
001000                  ACKSEQ: .BLKB   256.
001400                  LIDFD:  .BLKB   256.
002000                  IOFLG:  .BLKW
002002                  LLFLG:  .BLKW
002004                  IRSEND: .BLKW
                                .EVEN


        000001                  .END
```

```
FDM.MACRO       MACRO V03.01 23-JUL-79 12:51:52 PAGE 12-1
SYMBOL TABLE

ACKSED  001000R      003 B1$     000722R     E1$     001236R         MASTER  000000RG        SOUT    002576RG
AREA    001206R          B2$     000510R     G4$     002444R         NDXX  = 000000          STAT    001004R     002
ARG1  = 000002           B3$     000534R     HRBUF = 177562          ND24  = 000001          STATB0  001422RG
ARG2  = 000004           B4$     000560R     HRCSR = 177560          NHOST = 000000          STATUS  001324RG
ARG3  = 000006           B5$     000606R     HXBUF = 177566          OPREG = 172416          STAT$   000404R
ARG4  = 000010           B6$     000624R     HXCSR = 177564          OUTBF   000000R     002 SWITCH  001266RG
BAR   = 172410           B7$     000570R     INBF    000400R     002 RAM     001676RG        TIME    001654RG
BIT00 = 000001           CAUSE   001204R     IOBUF = 172416          RDENT   002266RG        WCR   = 172412
BIT01 = 000002           CRCOK   001054R     IOFIG   002000R     003 RDRAM   002002R         WTD     001652R
BIT02 = 000004           CSR   = 172414      IRSEND  002004R     003 READY   002616RG        WTKG    001552R
BIT03 = 000010           DATA    001212R     IWRT    001002R     002 ROM     002466R         WTOKEN  001474RG
BIT04 = 000020           DESTR   001256RG    IWRTTM  001000R     002 RST     002464R         WTRAM   001710R
BIT05 = 000040           DHOST = 000001      LIDFD   001400R     003 RSTART  002414RG        XCIO  = 000001
BIT06 = 000100           DMAOK   001160R     LIO     000430RG        RS0     002420R         XGIO  = 000001
BIT07 = 000200           EMBF    001024R     LIUINT  000076RG        RS1     002432R         XINBF   000400R     003
BIT08 = 000400           ENABLE  001214RG    LIFLG   002002R     003 RS2     002452R         XOUTBF  000000R     003
BIT14 = 040000           END$    001202R     LPINPT= ****** G        RTI$    001000R         XSIO  = 000000
BIT15 = 100000           ENSTR   001246RG    LPOUT   002102RG        SIO     002470RG        ZEROBF  000304R
B0$     000644R          E0$     001226R     LTIME   001006R     002

. ABS.  000000      000
        002642      001
DFM     001010      002
BUFS    002006      003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  51 PAGES
,DK:FDM=DK:FDM
```

```
FOR NOUAL /NOLINE
FOR NOUAL1/NOLINE
FOR BWBSA/NOLINE
MAC FOR
```

```
ASSIGN DX0: DK:
K LINK
DX1:NODE36,DX1:MAP=DX1:NODAL/C/I/W
DX1:FDM/F
DX1:NODAL I/C
DX1:BWBSA//
$SIMKT


ASSIGN DX1: DK:
```

1.8 NODE 27 (FIAC)

The Fault Isolation and Control Coordination (FIAC) module interprets Event Reports and Alarms from the VSQC, DSQC, and BWBSA modules for the purpose of isolating the equipment causing the detection of a fault condition. Amber and Red Event Reports are received by the FIAC and retransmitted to the destination Node Designator for OCRI reporting and to the DBMS in order to update the Equipment Status File. The OCRI operator can display and modify the Equipment Status File via Mode 6 of the User Language. Red Event Reports are reported only once when the equipment first fails although subsequent measurements will also result in the Red Region.

The FIAC analyzes the Red Event Reports for fault indications and initiates isolation procedures to resolve to the equipment level the location of the fault. The FIAC checks the connectivity through the network. For the FDM, the assumed connectivity between the VSQC, DSQC and BWBSDA is:

FDM Assumed Connectivity

Connectivity Group	Monitor Points
Area 1	1-333
Area 2	334-666
Area 3	667-1000

The FIAC collects Red Event Reports and assigns them to the various connectivity groups. When a connectivity group has a Red Event Report from VSQC, DSQC and BWBSA, the monitor points are compared to determine the equipment causing the fault. The lowest monitor point indicates the equipment causing the fault. Fault Reports are then sent to both the local OCRI terminal and a remote site OCRI. The Fault Report contains the monitor points, link and channel numbers for the connectivity group. The FIAC software contains variables used to specify the Node Designators for the Local and Remote destinations for Fault Reports. The default nodes are the OCRI terminal for Local Fault Reports and the CRT terminal in loop 4 for Remote Fault Reports.

In the MSCDM, software resides on the SDCA node to simulate the behavior of a remote FIAC. Area 1 is used to represent the area associated with the MSCDM FIAC. Area 2 is used to represent the area associated with the remote FIAC (as simulated by SDCA). Area 3 is used to simulate other areas for which faults cannot be isolated by the FIAC. The SDCA sends occasional random Event Reports to FIAC with Monitor Points in Area 1. FIAC sends Event Reports to SDCA for those Event Reports received from VSQC, DSQC, and BWBSA with Monitor Points in Area 2. When Red Event Reports are collected for the VSQC, DSQC, and BWBSA, faults can be isolated and reported in Areas 1 and 2 but cannot be isolated for Area 3. The faults that cannot be isolated are reported to local and remote OCRI's via a Fault Report.

1.8.1 Program Descriptions

1.8.1.1 Refer to Section 1.1 for descriptions of NODAL, IGETSP, ENQUE, DEQUE, ACKNAK, INPTQ, LPINPT, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME, RAM, LPOUT, RDPNT, RSTART.

1.8.1.2 Subroutine FIAC (FORTRAN)

This subroutine is called when a message is received from the loop and perform the functions described in Section 1.8.

1.8.1.3 Subroutine SNDRPT (FORTRAN)

The subroutine is called by FIAC routine to send event reports to DBMS.

1.8.1.4 Subroutine FAULT (FORTRAN)

This subroutine performs the fault isolation described in Section 1.8.

```
0001          PROGRAM NODAL
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003          INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP
0005          INTEGER*2 RSTBKP,Q1,Q2,RESNUM,OUTQ,DEQUE
0006          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007          LOGICAL*1 IDATA,PRED
0008          LOGICAL*1 PACK,ETX,CR,LF,MONTOP,ISLID,DUM
0009          INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0010          REAL*4 FMEAS
0011          REAL*8 FTR
0012          COMMON /MESS/ MESSEQ
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0014          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0015          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1          LIDFD(256),IOFLG,LLFLG,IRSEND
0016          COMMON /FRE/ FREE(64),IFR,IFRSZ
0017          COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0018          COMMON       PACK(256,64)
0019          COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0020          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNUM,IALTRT,ISLID,MONTOP,LF,ETX,CR,DUM
0021          COMMON /FCOM/ ICONN(3,6),IDATA(80),ICHAN,MON,FTR,FMEAS,
             &     ITYPE,IAREA,ITERM,PRED(1014)
0022          CALL MASTER
0023          CALL INIT
0024          CALL ENABLE(1)
0025        5 CONTINUE
0026       25 IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0028          CALL ENABLE(0)
0029          Q2=DEQUE(PINQ,INQ,1)
0032          CALL ENABLE(1)
0031          DO 30 I=1,80
0032            IDATA(I)=PACK(I,Q2)
0033       30 CONTINUE
0034          CALL FIAC
0035          IFR=IFR+1
0036          FREE(IFR)=Q2
0037       40 CALL STATB0(IS)
0038          IF(IS .EQ. 1) GOTO 100
0040          IF(LLFLG .GT. 0) CALL LINLOS
0042       80 IF(OUTFCT .EQ. 1) GOTO 100
0044          IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0046          IF(IRSEND .EQ. 1) GOTO 85
0048          IF(PXINQ(1) .LT. PXINQ(2)) GOTO 100
0050          CALL ENABLE(0)
0051          Q1=DEQUE(PXINQ,XINQ,1)
0052          CALL ENABLE(1)
0053          OUTQ=Q1
0054       85 CALL DESTR(PACK(254,OUTQ),Q2)
0055          DO 90 I=1,Q2
0056            OUTBF(I)=PACK(I,OUTQ)
```

```
0057     90  CONTINUE
0058         OUTBF(Q2+1)=0
0059         IPT=OUTBF(5)
0060         OUTBF(Q2+2)=LIDFD(IPT)
0061         IFULL=1
0062         ISENT=1
0063         CALL LPOUT(Q2+2)
0064         IRSEND=1
0065         ACKTIM=0
0066         IWRTTM=0
0067         IWRT=1
0068    100  INFLOT=0
0069    110  OLDTIM=NEWTIM
0070         NEWTIM=LTIME
0071         IF(IWRT .EQ. 0) GOTO 120
0073         IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0074         IF(IWRTTM .LT. TIMLIM) GOTO 120
0076         CALL WTOKEN
0077         IWRTTM=0
0078         IWRT=0
0079    120  IF(ISENT .EQ. 0) GOTO 130
0081         ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0082         IF(ACKTIM .LT. ATIMLM) GOTO 130
0084         CALL ENABLE(0)
0085         CALL ACKNAK(0)
0086         CALL ENABLE(1)
0087    130  CONTINUE
0088         GOTO 5
0089         END
```

```
FORTRAN IV      V02.1-11                                    PAGE 001

0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(64),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,IF,ETX,CR,DUM
0007          IF(IFR .LT. 1) CALL INIT
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```

```
0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FL#CNT,OUTQ,A(2),B(16)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1         ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0007          COMMON /GLOB/ ISENT,FL#CNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008     10   IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016     20   IF(IQHEAD .GE. (IQLNTH)) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020     30   B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024     40   CALL INIT
0025    999   RETURN
0026          END
```

```
0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016    10    A(1)=IQHEAD-1
0017   999    RETURN
0018          END
```

```
0001        SUBROUTINE ACKNAK(N)
0002        INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ,PXOUTQ
0003        INTEGER*2 ACKQ,PACKQ,PINQ,FREE,OUTQ,XOUTQ
0004        INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005        LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006        LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007        LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008        INTEGER*2 LTIME,IWRTTM
0009        REAL*8 COMT(5)
0010        EQUIVALENCE(COMT,LCOMT)
0011        DATA COMT(1)/'MESSAGE '/,COMT(2)/'NOT SENT'/,COMT(3)/' FROM    '/,
           1    COMT(4)/'NODE    '/,COMT(5)/' TO NODE'/
0012        COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
           1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013        COMMON /DFM/ OUTBF(256),INBF(256),
           1       IWRTTM,IWRT,STAT,LTIME
0014        COMMON         PACK(256,64)
0015        COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
           1          PACKQ(2),INQ(16),PINQ(2)
0016        COMMON /FRE/  FREE(64),IFR,IFRSZ
0017        COMMON /MESS/ MESSEQ
0018        COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
           1    IOFLG,LLFLG,IRSEND
0019        IF (N .NE. 1) GOTO 10
0021        IFR=IFR+1
0022        DO 5 I=1,256
0023          PACK(I,OUTQ)=0
0024      5 CONTINUE
0025        IFULL=0
0026        ISENT=0
0027        IRSEND=0
0028        FREE(IFR)=OUTQ
0029        GOTO 999
0030     10 T2=PACK(256,OUTQ)
0031        T2=T2+1
0032        IF(T2 .LE. RESNLM) GOTO 120
0034        T3=PACK(3,OUTQ)
0035        T4=MOD(T3,64)
0036        IF(T4 .LT. 32) GOTO 110
0038        T5=IGETSP(N)
0039        IF(MESSEQ .EQ. 126) MESSEQ=0
0041        MESSEQ=MESSEQ+1
0042        PACK(1,T5)=0
0043        PACK(2,T5)=MESSEQ
0044        PACK(3,T5)=0
0045        PACK(4,T5)=0
0046        PACK(5,T5)=25
0047        PACK(6,T5)=ISLID
0048        DO 20 I=7,60
0049          PACK(I,T5)="040
0050     20 CONTINUE
0051        DO 30 I=7,9
0052          PACK(I,T5)=LF
0053     30 CONTINUE
```

```
0054          DO 40 I=1,8
0055            PACK(I+9,T5)=LCOMT(I)
0056     40   CONTINUE
0057          DO 50 I=9,16
0058            PACK(I+9,T5)=LCOMT(I)
0059     50   CONTINUE
0060          DO 60 I=17,22
0061            PACK(I+9,T5)=LCOMT(I)
0062     60   CONTINUE
0063          DO 70 I=25,29
0064            PACK(I+7,T5)=LCOMT(I)
0065     70   CONTINUE
0066          ENCODE(3,80,PACK(38,T5)) ISLID
0067     80   FORMAT(I3)
0068          DO 90 I=33,40
0069            PACK(I+11,T5)=LCOMT(I)
0070     90   CONTINUE
0071          ENCODE(3,80,PACK(52,T5))PACK(5,OUTQ)
0072          PACK(55,T5)=LF
0073          PACK(56,T5)=LF
0074          PACK(57,T5)=CR
0075          PACK(58,T5)=ETX
0076          CALL ENSTR(PACK(254,T5),58)
0077          CALL ENQUE(PXINQ,XINQ,T5)
0078          DO 102 I=1,256
0079            PACK(I,OUTQ)=0
0080    102   CONTINUE
0081          ISENT=0
0082          IRSEND=0
0083          IFULL=0
0084          IWRT=0
0085          IFR=IFR+1
0086          FREE(IFR)=OUTQ
0087          GOTO 999
0088     110  PACK(3,OUTQ)=PACK(3,OUTQ)+32
0089          PACK(256,OUTQ)=0
0090          ISENT=0
0091          IFULL=0
0092          IF(IRSEND .EQ. 1) GOTO 999
0094          CALL ENQUE(PXINQ,XINQ,OUTQ)
0095          GOTO 999
0096    120   PACK(256,OUTQ)=T2
0097          IFULL=0
0098          ISENT=0
0099          IF(IRSEND .EQ. 1) GOTO 999
0101          CALL ENQUE(PXINQ,XINQ,OUTQ)
0102    999   RETURN
0103          END
```

```
0001          SUBROUTINE INPTQ(L)
0002          INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
0003          INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005          LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
0006          INTEGER*2 LTIME,IWRTTM
0007          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IOFLG,LLFLG,IRSEND
0009          COMMON         PACK(256,64)
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          IF(L .LT. 1) GOTO 30
0014          LI=L-2
0015          II=INBF(6)
0016          IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0018          T1=IGETSP(N)
0019          DO 10 I=1,LI
0020     10   PACK(I,T1)=INBF(I)
0021          CALL ENSTR(PACK(254,T1),LI)
0022          CALL ENQUE(PINQ,INQ,T1)
0023          N=2
0024          GOTO 40
0025     20   N=1
0026          GOTO 40
0027     30   N=0
0028     40   DO 50 I=1,2
0029            OUTBF(I)=INBF(I)
0030     50   CONTINUE
0031          NN=1
0032          IF(N .EQ. 0) NN=128
0034          OUTBF(3)=NN
0035          OUTBF(4)=0
0036          OUTBF(5)=INBF(6)
0037          OUTBF(6)=ISLID
0038          OUTBF(7)=ETX
0039          OUTBF(8)=0
0040          OUTBF(9)=LIDFD(INBF(6))
0041          CALL LPOUT(9)
0042          IF(N .NE. 2) GOTO 999
0044          ACKSEQ(II)=INBF(2)
0045     999  RETURN
0046          END
```

```
0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON/DFM/ OUTBF(256),INBF(256),IWRTTM,IWRT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          CC1=INBF(3)
0009          CC2=INBF(4)
0010          IF(CC1 .LT. 0) GOTO 25
0012          IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0014          CC3=MOD(CC1,2)
0015          CC4=MOD(CC1,256)
0016          IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0018          IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0020          CC5=MOD(INBF(3),64)
0021          IF(CC5 .GE. 32) CALL INPTQ(LI)
0023          GOTO 99
0024     20   CC7=MOD(CC1,2)
0025          IF(CC7 .GE. 1) GOTO 30
0027     25   CALL ACKNAK(0)
0028          GOTO 99
0029     30   CALL ACKNAK(1)
0030          GOTO 99
0031     40   CALL INPTQ(LI)
0032     99   CONTINUE
0033          RETURN
0034          END
```

```
0001        SUBROUTINE INIT
0002        REAL*4 FMEAS
0003        REAL*8 FTR
0004        INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0005        INTEGER*2 STAT,FLWCNT,OUTFCT
0006        INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0007        LOGICAL*1 DUM,IDATA,PRED
0008        LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0009        LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID
0010        INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0011        COMMON /DFM/ OUTBF(256),INBF(256),
           1    IWRTTM,IWRT,STAT,LTIME
0012        COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
           1             ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0013        COMMON /MESS/ MESSEQ
0014        COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
           1    IOFLG,LLFLG,IRSEND
0015        COMMON /FRE/  FREE(64),IFR,IFRSZ
0016        COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0017        COMMON        PACK(256,64)
0018        COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0019        COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
           1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0020        COMMON /FCOM/ ICONN(3,6),IDATA(80),ICHAN,MON,FTR,FMEAS,
           &    ITYPE,IAREA,ITERM,PRED(1014)
0021        CALL RAM(0,9,4)
0022        CALL RAM(0,255,0)
0023        MESSEQ=0
0024        IOFLG=0
0025        IRSEND=0
0026        LLFLG=0
0027        IFRSZ=64
0028        INFLCT=0
0029        ISLID=27
0030        CR="015
0031        LF="012
0032        ETX="003
0033        ICHNSW=0
0034        RESNLM=2
0035        IQLNTH=16
0036        IALTRT=0
0037        FLWCNT=0
0038        IWRTTM=0
0039        TIMLIM=35
0040        SETPRM=4
0041        RSTPRM=32
0042        SETBKP=8
0043        RSTBKP=64
0044        ATIMLM=200
0045        MONTOR=25
0046        IWRT=0
0047        ISWTCH=0
0048        OUTFCT=0
0049        IFULL=0
```

```
0050          ISENT=0
0051          DO 40 I=1,20
0052     40   LIDFD(I)=4
0053          LIDFD(21)=1
0054          LIDFD(22)=3
0055          LIDFD(23)=6
0056          LIDFD(24)=5
0057          LIDFD(25)=7
0058          LIDFD(26)=8
0059          LIDFD(27)=9
0060          LIDFD(28)=2
0061          DO 50 I=29,39
0062     50   LIDFD(I)=0
0063          DO 60 I=40,44
0064     60   LIDFD(I)=1
0065          DO 70 I=45,59
0066     70   LIDFD(I)=0
0067          DO 80 I=60,64
0068     80   LIDFD(I)=2
0069          DO 90 I=65,79
0070     90   LIDFD(I)=0
0071          DO 100 I=80,84
0072    100   LIDFD(I)=4
0073          DO 110 I=85,99
0074    110   LIDFD(I)=0
0075          DO 120 I=100,104
0076    120   LIDFD(I)=5
0077          DO 130 I=105,256
0078    130   LIDFD(I)=0
0079          PINQ(1)=IQLNTH
0080          PINQ(2)=IQLNTH+1
0081          PACKQ(1)=IQLNTH
0082          PACKQ(2)=IQLNTH+1
0083          PXOUTQ(1)=IQLNTH
0084          PXOUTQ(2)=IQLNTH+1
0085          PXINQ(1)=IQLNTH
0086          PXINQ(2)=IQLNTH+1
0087          DO 10 I=1,IFRSZ
0088            PACK(255,I)=0
0089     10   PACK(256,I)=0
0090          DO 20 I=1,IFRSZ
0091     20   FREE(I)=I
0092          DO 30 I=1,256
0093     30   ACKSEQ(I)=256
0094          IFR=IFRSZ
0095          RETURN
0096          END
```

```
0001        SUBROUTINE LINLOS
0002        REAL*8 RM1(5),LIN08,LIN18
0003        INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004        INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,PINQ
0005        LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006        LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0007        LOGICAL*1 PACK,M1(40),LIN0(8),LIN1(8)
0008        COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
           &          PACKQ(2),INQ(16),PINQ(2)
0009        COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
           &           IOFLG,LLFLG,IRSEND
0010        COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
           1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011        COMMON         PACK(256,64)
0012        COMMON /MESS/ MESSEQ
0013        DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0014        DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0015        DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0016        DATA RM1(5)/'DE 27   '/
0017        EQUIVALENCE(LIN0,LIN08)
0018        EQUIVALENCE(LIN1,LIN18)
0019        EQUIVALENCE(M1,RM1)
0020        IS=LLFLG
0021        CALL ENABLE(0)
0022        T1=IGETSP(N)
0023        CALL ENABLE(1)
0024        PACK(1,T1)=0
0025        IF(MESSEQ .EQ. 126) MESSEQ=0
0027        MESSEQ=MESSEQ+1
0028        PACK(2,T1)=MESSEQ
0029        PACK(3,T1)=0
0030        PACK(4,T1)=0
0031        PACK(5,T1)=25
0032        PACK(6,T1)=ISLID
0033        DO 20 I=7,9
0034          PACK(I,T1)=LF
0035    20  CONTINUE
0036        DO 30 I=1,22
0037    30  PACK(I+9,T1)=M1(I)
0038        IF(IS .EQ. 1) GOTO 50
0040        DO 40 I=1,8
0041    40  PACK(I+31,T1)=LIN0(I)
0042        GOTO 70
0043    50  DO 60 I=1,8
0044    60  PACK(I+31,T1)=LIN1(I)
0045    70  DO 80 I=23,40
0046    80  PACK(I+17,T1)=M1(I)
0047        PACK(58,T1)=CR
0048        PACK(59,T1)=LF
0049        PACK(60,T1)=ETX
0050        CALL ENSTR(PACK(254,T1),60)
0051        CALL ENABLE(0)
0052        CALL ENQUE(PXINQ,XINQ,T1)
0053        CALL ENABLE(1)
```

```
0054          LLFLG=0
0055          RETURN
0056          END
```

```
0001          SUBROUTINE FIAC
0002          LOGICAL*1 PRED,IDATA
0003          REAL*4 FMEAS
0004          REAL*8 FTR
0005          COMMON /FCOM/ ICONN(3,6),IDATA(80),ICHAN,MON,FIR,FMEAS,
             &    ITYPE,IAREA,ITERM,PRED(1014)
0006          DECODE(2,4,ILATA(15),ERR=250) ICHAN
0007          DECODE(4,6,IDATA(19),ERR=250) FMEAS
0008          DECODE(2,4,IDATA(22),ERR=250) MON
0009        4 FORMAT(A2)
0010        6 FORMAT(A4)
0011          ITERM=IDATA(24)
0012       10 IF(IDATA(6) .EQ. 22) GOTO 20
0014          IF(IDATA(6) .EQ. 23) GOTO 30
0016          IF(IDATA(6) .EQ. 26) GOTO 40
0018          IF(IDATA(6) .EQ. 28) GOTO 50
0020          RETURN
0021       20 ITYPE=1
0022          GOTO 60
0023       30 ITYPE=3
0024          GOTO 60
0025       40 ITYPE=5
0026          IF((IDATA(7) .EQ. 'X') .AND.
             &   (IDATA(8) .EQ. 'X') .AND.
             &   (IDATA(9) .EQ. 'X') .AND.
             &   (IDATA(10) .EQ. 'X')) GOTO 45
0028          GOTO 60
0029       45 CALL SNDRPT(3)                        !SEND ALARM REPORT
0030          RETURN
0031       50 ITYPE=IDATA(25)
0032          IF(ITYPE .EQ. 1 .OR. ITYPE .EQ. 3 .OR.
             &  ITYPE .EQ. 5) GOTO 60
0034          RETURN
0035       60 IF(IDATA(17) .EQ. 1) GOTO 70
0037          IF(ITYPE .EQ. 5) ICHAN=ICHAN+1000
0039          CALL SNDRPT(2)                        !SEND AMBER REPORT
0040          RETURN
0041       70 IF(ITYPE .EQ. 5) ICHAN=ICHAN+1000
0043          PRED(ICHAN) = PRED(ICHAN)+1
0044          IF(PRED(ICHAN) .EQ. 1) GOTO 80
0046          IF(PRED(ICHAN) .GE. 21) PRED(ICHAN)=0
0048          RETURN
0049       80 CALL SNDRPT(1)                        !SEND RED REPORT
0050          IF(ITYPE .EQ. 5) ICHAN=ICHAN-1000
0052          IF((MON .GE. 1) .AND.
             &   (MON .LE. 333)) GOTO 90
0054          IF((MON .GE. 334) .AND.
             &   (MON .LE. 666)) GOTO 100
0056          IF((MON .GE. 667) .AND.
             &   (MON .LE. 1000)) GOTO 110
0058          RETURN
0059       90 IAREA=1
0060          GOTO 130
0061      100 IAREA=2
```

```
0062          CALL SNDRPT(4)
0063          GOTO 130
0064    110   IAREA=3
0065    130   IF(ICONN(IAREA,ITYPE) .EQ. 0) GOTO 140
0067          RETURN
0068    140   ICONN(IAREA,ITYPE)=ICHAN
0069          ICONN(IAREA,ITYPE+1)=MON
0070          DO 150 I=1,6
0071            IF(ICONN(IAREA,I) .EQ. 0) GOTO 250
0073    150   CONTINUE
0074          IF(ICONN(IAREA,2) .LT. ICONN(IAREA,4)) GOTO 190
0076          IF(ICONN(IAREA,4) .LT. ICONN(IAREA,6)) GOTO 200
0078    190   IF(ICONN(IAREA,2) .LT. ICONN(IAREA,6)) GOTO 210
0080          IT=6
0081          GOTO 220
0082    200   IT=4
0083          GOTO 220
0084    210   IT=2
0085    220   IF((IAREA .EQ. 1) .OR. (IAREA .EQ. 2)) GOTO 230
0087          CALL FAULT(2,IT)
0088          GOTO 240
0089    230   CALL FAULT(1,IT)
0090    240   DO 245 I=1,6
0091            ICONN(IAREA,I)=0
0092    245   CONTINUE
0093    250   RETURN
0094          END
```

```
0001          SUBROUTINE SNDRPT(IC)
0002          REAL*8 HEAD(3),FTR
0003          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0004          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0005          LOGICAL*1 PACK,DAEH(24),IDATA,PRED,ISLID,MONTOR
0006          LOGICAL*1 LF,CR,ETX,DUM
0007          REAL*4 FMEAS
0008          EQUIVALENCE (HEAD,DAEH)
0009          DATA HEAD(1)/'TRUNK   '/,HEAD(2)/'CHANNEL '/,HEAD(3)/'COND
0010          COMMON PACK(256,64)
0011          COMMON /FCOM/ ICONN(3,6),IDATA(80),ICHAN,MON,FTR,FMEAS,
             &    ITYPE,IAREA,ITERM,PRED(1014)
0012          COMMON /MESS/ MESSEQ
0013          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             &    RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0014          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &    ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0015          IF(IC .EQ. 3) GOTO 100
0017          IF(IC .EQ. 4) GOTO 200              !EVENT REPORT TO SDCA
0019          CALL ENABLE(0)
0020          K2=IGETSP(N)
0021          CALL ENABLE(1)
0022          IF(MESSEQ .EQ. 126) MESSEQ=0
0024          MESSEQ=MESSEQ+1
0025          PACK(1,K2)=0
0026          PACK(2,K2)=MESSEQ
0027          PACK(3,K2)=0
0028          PACK(4,K2)=0
0029          PACK(5,K2)=24
0030          PACK(6,K2)=ISLID
0031          PACK(7,K2)="177
0032          DO 70 I=7,14
0033            PACK(I+1,K2)=IDATA(I)
0034      70  CONTINUE
0035          ENCODE(2,80,PACK(16,K2)) ICHAN
0036      80  FORMAT(A2)
0037          PACK(18,K2)=IC
0038          DO 86 J=19,40
0039      86  PACK(J,K2)=0
0040          PACK(40,K2)=CR
0041          PACK(41,K2)=LF
0042          PACK(42,K2)=ETX
0043          CALL ENSTR(PACK(254,K2),42)
0044          CALL ENABLE(0)
0045          CALL ENQUE(PXINQ,XINQ,K2)
0046          CALL ENABLE(1)
0047          CONTINUE
0048          CALL ENABLE(0)
0049          K1=IGETSP(N)
0050          CALL ENABLE(1)
0051          DO 10 J=1,65
0052      10  PACK(J,K1)="040
0053          IF(MESSEQ .EQ. 126) MESSEQ=0
0055          MESSEQ=MESSEQ+1
```

```
0056          PACK(1,K1)=0
0057          PACK(2,K1)=MESSEQ
0058          PACK(3,K1)=0
0059          PACK(4,K1)=0
0060          PACK(5,K1)=ITERM
0061          PACK(6,K1)=ISLID
0062          DO 20 I=1,5
0063          PACK(I+9,K1)=DAEH(I)
0064     20   CONTINUE
0065          DO 30 I=9,15
0066          PACK(I+11,K1)=DAEH(I)
0067     30   CONTINUE
0068          DO 40 I=17,20
0069          PACK(I+14,K1)=DAEH(I)
0070     40   CONTINUE
0071          PACK(35,K1)=CR
0072          PACK(36,K1)=LF
0073          DO 50 I=7,14
0074          PACK(I+32,K1)=IDATA(I)
0075     50   CONTINUE
0076          ENCODE(4,60,PACK(51,K1)) ICHAN
0077     60   FORMAT(I4)
0078          ENCODE(2,65,PACK(61,K1)) IC
0079     65   FORMAT(I2)
0080          PACK(63,K1)=CR
0081          PACK(64,K1)=LF
0082          PACK(65,K1)=ETX
0083          CALL ENSTR(PACK(254,K1),65)
0084          CALL ENABLE(0)
0085          CALL ENQUE(PXINQ,XINQ,K1)
0086          CALL ENABLE(1)
0087          RETURN
0088    100   CALL ENABLE(0)
0089          K1=IGETSP(N)
0090          K2=IGETSP(N)
0091          CALL ENABLE(1)
0092          DO 110 I=1,256
0093            PACK(I,K1)="40
0094            PACK(I,K1)="40
0095    110   CONTINUE
0096          IF(MESSEQ .EQ. 126) MESSEQ=0
0098          MESSEQ=MESSEQ+1
0099          PACK(1,K1)=0
0100          PACK(2,K1)=MESSEQ
0101          PACK(3,K1)=0
0102          PACK(4,K1)=0
0103          PACK(5,K1)=ITERM
0104          PACK(6,K1)=ISLID
0105          DO 120 I=7,10
0106            PACK(I,K1)="007
0107    120   CONTINUE
0108          DO 130 I=7,10
0109            PACK(I+4,K1)=IDATA(I)
0110    130   CONTINUE
```

```
0111          ENCODE(2,140,PACK(16,K1)) IDATA(11)
0112          ENCODE(2,140,PACK(18,K1)) IDATA(13)
0113    140   FORMAT(I2)
0114          PACK(20,K1)=CR
0115          PACK(21,K1)=LF
0116          PACK(22,K1)=ETX
0117          CALL ENSTR(PACK(254,K1),22)
0118          IF(MESSEQ .EQ. 126) MESSEQ=0
0120          MESSEQ=MESSEQ+1
0121          PACK(1,K2)=0
0122          PACK(2,K2)=MESSEQ
0123          PACK(3,K2)=0
0124          PACK(4,K2)=0
0125          PACK(5,K2)=24
0126          PACK(6,K2)=ISLID
0127          DO 150 I=7,14
0128            PACK(I,K2)=IDATA(I)
0129    150   CONTINUE
0130          PACK(18,K2)=IC
0131          PACK(19,K2)="015
0132          PACK(20,K2)="012
0133          PACK(21,K2)="003
0134          CALL ENSTR(PACK(254,K2),21)
0135          CALL ENABLE(0)
0136          CALL ENQUE(PXINQ,XINQ,K1)
0137          CALL ENQUE(PXINQ,XINQ,K2)
0138          CALL ENABLE(1)
0139          RETURN
0140    200   CALL ENABLE(0)
0141          K1=IGETSP(N)
0142          CALL ENABLE(1)
0143          IF(MESSEQ .EQ. 126) MESSEQ=0
0145          MESSEQ=MESSEQ+1
0146          PACK(1,K1)=0
0147          PACK(2,K1)=MESSEQ
0148          PACK(3,K1)=0
0149          PACK(4,K1)=0
0150          PACK(5,K1)=28
0151          PACK(6,K1)=ISLID
0152          PACK(7,K1)='$'
0153          DO 210 I=7,14
0154            PACK(I+1,K1)=IDATA(I)
0155    210   CONTINUE
0156          ENCODE(2,220,PACK(16,K1)) ICHAN
0157    220   FORMAT(A2)
0158          PACK(18,K1)=IC
0159          CALL ENSTR(PACK(254,K1),18)
0160          CALL ENABLE(0)
0161          CALL ENQUE(PXINQ,XINQ,K1)
0162          CALL ENABLE(1)
0163          RETURN
0164          END
```

```
0001          SUBROUTINE FAULT(IC,IT)
0002          LOGICAL*1 IMSG1(40),IMSG2(56),PACK,IDATA,PRED
0003          LOGICAL*1 CR,ETX,DUM,ISLID,LF,MONTOR
0004          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0005          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0006          REAL*4 FMEAS
0007          REAL*8 MSG1(5),MSG2(7),FTR
0008          EQUIVALENCE (MSG1,IMSG1),(MSG2,IMSG2)
0009          COMMON PACK(256,64)
0010          COMMON /MESS/ MESSEQ
0011          COMMON /FCOM/ ICONN(3,6),IDATA(80),ICHAN,MON,FTR,FMEAS,
             &     ITYPE,IAREA,ITERM,PRED(1014)
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             &     RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &     ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0014          DATA MSG1(1)/'FAULT AR'/,MSG1(2)/'EA  CAUS'/,MSG1(3)/'ED BY   '/,
             &     MSG1(4)/'LINKCHAN'/,MSG1(5)/' MON PT '/
0015          DATA MSG2(1)/'FAULT CA'/,MSG2(2)/'NNOT BE '/,MSG2(3)/'ISOLATED'/,
             &     MSG2(4)/'POSSIBLE'/,MSG2(5)/' CAUSE  '/,MSG2(6)/'LINKCHAN'/,
             &     MSG2(7)/' MON PT '/
0016          CALL ENABLE(0)
0017          K1=IGETSP(N)
0018          K2=IGETSP(N)
0019          CALL ENABLE(1)
0020          IF(IC .EQ. 2) GOTO 100
0022          DO 80 ICRT=18,25,7
0023            K3=K1
0024            IF(ICRT .EQ. 25) K3=K2
0026            IF(MESSEQ .EQ. 126) MESSEQ=0
0028            MESSEQ=MESSEQ+1
0029            PACK(1,K3)=0
0030            PACK(2,K3)=MESSEQ
0031            PACK(3,K3)=0
0032            PACK(4,K3)=0
0033            PACK(5,K3)=ICRT
0034            PACK(6,K3)=ISLID
0035            DO 10 I=1,8
0036               PACK(I+6,K3)=IMSG1(I)
0037    10      CONTINUE
0038            DO 20 I=9,11
0039               PACK(I+6,K3)=IMSG1(I)
0040    20      CONTINUE
0041            ENCODE(2,25,PACK(18,K3)) IC
0042    25      FORMAT(I2)
0043            DO 30 I=12,16
0044               PACK(I+8,K3)=IMSG1(I)
0045    30      CONTINUE
0046            DO 35 I=17,22
0047               PACK(I+8,K3)=IMSG1(I)
0048    35      CONTINUE
0049            IF(IT .EQ. 6) GOTO 45
0051            DO 40 I=29,32
0052               PACK(I+2,K3)=IMSG1(I)
```

```
0053    40    CONTINUE
0054          GOTO 55
0055    45    DO 50 I=25,28
0056             PACK(I+6,K3)=IMSG1(I)
0057    50    CONTINUE
0058    55    ENCODE(5,60,PACK(35,K3)) ICONN(IAREA,IT-1)
0059    60    FORMAT(I5)
0060          DO 70 I=33,40
0061             PACK(I+7,K3)=IMSG1(I)
0062    70    CONTINUE
0063          ENCODE(5,60,PACK(48,K3)) ICONN(IAREA,IT)
0064          PACK(53,K3)=CR
0065          PACK(54,K3)=LF
0066          PACK(55,K3)=ETX
0067          CALL ENSTR(PACK(254,K3),55)
0068    80  CONTINUE
0069        GOTO 210
0070   100  DO 200 ICRT=18,25,7
0071          K3=K1
0072          IF(ICRT .EQ. 25) K3=K2
0074          IF(MESSEQ .EQ. 126) MESSEQ=0
0076          MESSEQ=MESSEQ+1
0077          PACK(1,K3)=0
0078          PACK(2,K3)=MESSEQ
0079          PACK(3,K3)=0
0080          PACK(4,K3)=0
0081          PACK(5,K3)=ICRT
0082          PACK(6,K3)=ISLID
0083          DO 110 I=1,8
0084             PACK(I+6,K3)=IMSG2(I)
0085   110    CONTINUE
0086          DO 120 I=9,16
0087             PACK(I+6,K3)=IMSG2(I)
0088   120    CONTINUE
0089          DO 130 I=17,24
0090             PACK(I+6,K3)=IMSG2(I)
0091   130    CONTINUE
0092          PACK(31,K3)=CR
0093          PACK(32,K3)=LF
0094          DO 140 I=25,32
0095             PACK(I+8,K3)=IMSG2(I)
0096   140    CONTINUE
0097          DO 150 I=33,40
0098             PACK(I+8,K3)=IMSG2(I)
0099   150    CONTINUE
0100          IF(IT .EQ. 6) GOTO 165
0102          DO 160 I=41,44
0103             PACK(I+8,K3)=IMSG2(I)
0104   160    CONTINUE
0105          GOTO 175
0106   165    DO 170 I=45,48
0107             PACK(I+4,K3)=IMSG2(I)
0108   170    CONTINUE
0109   175    ENCODE(5,180,PACK(53,K3)) ICONN(IAREA,IT-1)
```

```
0110   180    FORMAT(I5)
0111          DO 190 I=49,56
0112             PACK(I+9,K3)=IMSG2(I)
0113   190    CONTINUE
0114          ENCODE(5,180,PACK(66,K3)) ICONN(IAREA,IT)
0115          PACK(71,K3)=CR
0116          PACK(72,K3)=LF
0117          PACK(73,K3)=17
0118          PACK(74,K3)=ETX
0119          CALL ENSTR(PACK(254,K3),74)
0120   200  CONTINUE
0121   210  CALL ENABLE(0)
0122        CALL ENQUE(PXINQ,XINQ,K1)
0123        CALL ENQUE(PXINQ,XINQ,K2)
0124        CALL ENABLE(1)
0125        RETURN
0126        END
```

```
                                .TITLE  FDM.MACRO
                                .SBTTL  NODE 29
                                .IDENT  /V3.0/
                                .GLOBL  LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
                                .GLOBL  TOKEN,RAM,STATUS,STATB0
                                .GLOBL  LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
                                .NLIST  CND
000000                          .PSECT

000000          R0      =%0
000001          R1      =%1
000002          R2      =%2
000003          R3      =%3
000004          R4      =%4
000005          R5      =%5
000006          SP      =%6
000007          PC      =%7

                ;INTERFACE ADDRESS

172410          BAR=    172410
172412          WCR=    172412
172414          CSR=    172414
172416          IOBUF=  172416
172416          OPREG=  172416
177560          HRCSR=  177560          ;CHANGED
177562          HRBUF=  177562
177564          HXCSR=  177564
177566          HXBUF=  177566

                ;INTERFACE OPTIONS

000001          XCIO=1
000001          XSIO=1
000001          XGIO=1
000000          NDXX=0
000001          ND24=1
000000          NHOST=0
000001          DHOST=1

                ;LOCAL VARIBLES

000002          ARG1=   2
000004          ARG2=   4
000006          ARG3=   6
000010          ARG4=   10

100000          BIT15=  100000
040000          BIT14=  40000
000400          BIT08=  400
000200          BIT07=  200
000100          BIT06=  100
000040          BIT05=  40
000020          BIT04=  20
000010          BIT03=  10
000004          BIT02=  4
```

```
FDM.MACRO       MACRO V03.02B 00:36:52 PAGE 1-1
NODE 27

     66         000002                          BIT01=  2
     67         000001                          BIT00=  1
     68
```

```
                        ;*************************************************************
                        ;**** PROCEDURE MASTER START UP                              *
                        ;*************************************************************

000000  012700  000340          MASTER: MOV     #340,   R0                      ;PRI=7
000004  106400                          MTPS    R0
000006  012700  000124                  MOV     #124,   R0
000012  012720  000412'                 MOV     #LIO,   (R0)+                   ;LIU HANDLER
000016  012710  000340                  MOV     #340,   (R0)                    ;PRI=7
000022  012700  000100                  MOV     #100,   R0                      ;CLOCK VECTOR
000026  012710  001636'                 MOV     #TIME,  (R0)                    ;CLOCK
000032  042737  040100  172414          BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000040  004767  000014                  JSR     PC,     LIUINT                  ;CLEAR LIU
000044  005037  172414                  CLR     @#CSR                           ;CLEAR BLIUI CSR
000050  012700  000000                  MOV     #000,   R0                      ;PRI=0
000054  106400                          MTPS    R0                              ;CPU DOWN
000056  000207                          RTS     PC


                        ;*************************************************************
                        ;**** PROCEDURE INITIALIZE                                  *
                        ;*************************************************************

                                ;CLEAR ACRAM

000060  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000066  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000074  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000100  100375                          BPL     .-4                             ;NO RETRY
000102  105037  172414                  CLRB    @#CSR
000106  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
000114  012700  177400                  MOV     #-256., R0                      ;COUNTER
000120  012737  004407  172416  1$:     MOV     #2311., @#OPREG                 ;WRITE A NULL
000126  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000132  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
000134  105037  172414                  CLRB    @#CSR
000140  005200                          INC     R0                              ;COUNT+1
000142  001366                          BNE     1$

                                ;CLEAR INPUT/OUTPUT BUFFERS

000144  012701  010410                  MOV     #10410, R1                      ;RDBUFADR CMD
000150  012700  010440                  MOV     #10440, R0                      ;SEL INBUF0
000154  004767  000106                  JSR     PC,     ZEROBP                  ;SET POINTER=0
000160  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000166  105737  172414                  TSTB    @#CSR                           ;GOOD READ
000172  100375                          BPL     .-4                             ;NO RETRY
000174  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
000200  012701  010610                  MOV     #10610, R1                      ;RDBUFADR COMMAND
000204  012700  010640                  MOV     #10640, R0                      ;SEL INBUF1 COMMAND
000210  004767  000052                  JSR     PC,     ZEROBP                  ;POINTER=0
000214  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000222  105737  172414                  TSTB    @#CSR                           ;GOOD RD
000226  100375                          BPL     .-4                             ;NO RETRY
000230  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
```

```
106 000234  012701  010510                  MOV     #10510, R1              ;RDBUFADR COMMAND
107 000240  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
108 000244  004767  000216                  JSR     PC,     ZEROBP
109 000250  012701  010710                  MOV     #10710, R1              ;RDBUFADR COMMAND
110 000254  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
111 000260  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
112 000264  000440                          BR      STAT$                   ;GO CLEAR STATUS
113 000266  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
114 000272  012737  001400  172416        MOV     #1400,  @#OPREG         ;RD
115 000300  105737  172414                  TSTB    @#CSR                   ;GOOD RD
116 000304  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
117 000306  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
118 000312  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
119 000316  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
120 000322  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
121 000326  022702  000000          2$:     CMP     #0,     R2              ;POINTER=0?
122 000332  001412                          BEQ     3$                      ;YES RECHECK
123 000334  012737  001400  172416        MOV     #1400,  @#OPREG         ;NO FALSE RD
124 000342  105737  172414                  TSTB    @#CSR                   ;GOOD RD
125 000346  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
126 000350  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
127 000354  005302                          DEC     R2                      ;LOCAL POINTER-1
128 000356  000763                          BR      2$                      ;BR  NTIL 0
129 000360  010037  172416          3$:     MOV     R0,     @#OPREG         ;SEL BUFFER
130 000364  000207                          RTS     PC                      ;RETURN
131
132                                         ;READ AND CLEAR STATUS
133
134 000366  012737  010400  172416  STAT$: MOV     #4352., @#OPREG         ;WCR:RS
135 000374  012737  002400  172416        MOV     #1280., @#OPREG         ;RS(FALSE)
136 000402  012737  002400  172416        MOV     #1280., @#OPREG         ;RS
137 000410  000207                          RTS     PC
138
139
```

```
                                ;*********************************************************
                                ;**** PROCEDURE LIU-HANDLER                              *
                                ;*********************************************************


000412  010046          LIO:    MOV     R0,     -(SP)           ;SAVE R0
000414  010146                  MOV     R1,     -(SP)           ;SAVE R1
000416  010246                  MOV     R2,     -(SP)           ;SAVE R2
000420  010346                  MOV     R3,     -(SP)           ;SAVE R3
000422  010446                  MOV     R4,     -(SP)           ;SAVE R4
000424  010546                  MOV     R5,     -(SP)           ;SAVE R5
000426  005737  172414          TST     @#CSR                   ;? LIU INTERRUPTED
000432  100153                  BPL     RTI$                    ;NO
000434  042737  040000  172414  BIC     #BIT14, @#CSR           ;YES/DISABLE LIU
000442  012737  010400  172416  MOV     #4352., @#OPREG         ;WCR : RS(0)
000450  012737  002400  172416  MOV     #1280., @#OPREG         ;RD
000456  013767  172416  000502  MOV     @#IOBUF, CAUSE
000464  042767  177400  000474  BIC     #177400, CAUSE          ;CLEAR UNUSED BITS

                                ;PREFORM OPERATION DEPENDING ON BITS

000472  132767  000004  000466  B2$:    BITB    #BIT02, CAUSE   ;OV-FL
000500  001406                  BEQ     B3$
000502  012702  010410          MOV     #4360., R2              ;RDBUFADR IN0
000506  012703  010440          MOV     #4384., R3              ;SEL INBUF0
000512  004767  000270          JSR     PC,     EMBF            ;GO EMPTY
000516  132767  000010  000442  B3$:    BITB    #BIT03, CAUSE   ;OV-FL
000524  001406                  BEQ     B4$
000526  012702  010610          MOV     #4488., R2              ;RDBUFADR IN1
000532  012703  010640          MOV     #4512., R3              ;SEL INBUF1
000536  004767  000244          JSR     PC,     EMBF            ;GO EMPTY
000542  132767  000020  000416  B4$:    BITB    #BIT04, CAUSE   ;LINE-LOSS PRIMARY
000550  001407                  BEQ     B5$
000552  132767  000200  000406  B7$:    BITB    #BIT07, CAUSE   ;LATCH ON
000560  001403                  BEQ     B5$                     ;YES LOOP AROUND
000562  012767  000002  002002'        MOV     #2,     LLFLG   ;FLAG PRIMARY SW. FAIL
000570  132767  000040  000370  B5$:    BITB    #BIT05, CAUSE   ;BACKUP LINE LOSS
000576  001403                  BEQ     B6$
000600  012767  000001  002002'        MOV     #1,     LLFLG   ;FLAG BACKUP SW. FAIL
000606  132767  000100  000352  B6$:    BITB    #BIT06, CAUSE   ;WRITE TOKEN DETECT
000614  001404                  BEQ     B0$
000616  005067  001000'          CLR     IWRTTM                  ;CLEAR TOKEN VAR
000622  005067  001002'          CLR     IWRT

                                ;BUFFER FULL

000626  132767  000001  000332  B0$:    BITB    #BIT00, CAUSE   ;INBUF0 FULL
000634  001423                  BEQ     B1$                     ;NO
000636  012701  000400'          MOV     #INBF,  R1              ;BUFFER ADDRESS
000642  012702  010410          MOV     #4360., R2              ;RDBUFADR COMM
000646  012703  010440          MOV     #4384., R3              ;SEL BUF
000652  012704  000001          MOV     #BIT00, R4              ;CRC BIT
000656  004767  000124          JSR     PC,     EMBF            ;EMPTY BUFFER
000662  012705  001170'          MOV     #AREA,  R5              ;DATA LINK AREA
000666  012767  000001  000274  MOV     #1,     AREA            ;1 VARIBLE
000674  010267  000274          MOV     R2,     DATA            ;CRC OR BYTE COUNT
000700  004767  000000G         JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
```

```
 58 000704  132767  000002  000254  B1$:    BITB    #BIT01, CAUSE           ;INBUF1 FULL
 59 000712  001423                   BEQ     RTI$                    ;NO WERE DONE
 60 000714  012701  000400'                  MOV     #INBF,  R1              ;BUFFER ADDRESS
 61 000720  012702  010610                   MOV     #4488., R2              ;RDBUFADR COMM
 62 000724  012703  010612                   MOV     #4512., R3              ;SEL INBUF1
 63 000730  012704  000002                   MOV     #BIT01, R4              ;CRC BIT
 64 000734  004767  000046                   JSR     PC,     EMBF            ;GO EMPTY BUFFER
 65 000740  012705  001170'                  MOV     #AREA,  R5              ;DATA LINK AREA
 66 000744  012767  000001  000216          MOV     #1,     AREA            ;ONE VARIBLE
 67 000752  010267  000216                   MOV     R2,     DATA            ;CRC OR BYTE COUNT
 68 000756  004767  000000G                  JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
 69
 70 000762  012605                  RTI$:    MOV     (SP)+,  R5              ;RESTORE REGISTERS
 71 000764  012604                           MOV     (SP)+,  R4
 72 000766  012603                           MOV     (SP)+,  R3
 73 000770  012602                           MOV     (SP)+,  R2
 74 000772  012601                           MOV     (SP)+,  R1
 75 000774  012600                           MOV     (SP)+,  R0
 76 000776  052737  040000  172414          BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
 77 001004  000002                           RTI                             ;RETURN FROM INTERRUPT
 78
 79 001006  012737  010600  172416  EMBF:    MOV     #4480., @#OPREG         ;READ STATUS 1
 80 001014  012737  002400  172416           MOV     #1280., @#OPREG         ;RS
 81 001022  013700  172416                   MOV     @#IOBUF, R0             ;FETCH STATUS
 82 001026  130400                           BITB    R4,R0                   ;GOOD CRC
 83 001030  001002                           BNE     CRCOK
 84 001032  012704  177777                   MOV     #-1,    R4              ;NO FLAG
 85 001036  010237  172416           CRCOK:  MOV     R2,     @#OPREG         ;RDBUFADR
 86 001042  012737  001400  172416           MOV     #768.,  @#OPREG         ;RD
 87 001050  105737  172414                   TSTB    @#CSR                   ;GOOD RD
 88 001054  100375                           BPL     .-4
 89 001056  013702  172416                   MOV     @#IOBUF, R2             ;REPLACE WITH POINTER
 90 001062  042702  177400                   BIC     #177400, R2             ;CLEAR MST BITS
 91 001066  010200                           MOV     R2,     R0              ;SAVE IT
 92 001070  005400                           NEG     R0                      ;2'S COMP
 93 001072  010037  172412                   MOV     R0,     @#WCR           ;BYTE COUNT NOW
 94 001076  010137  172410                   MOV     R1,     @#BAR           ;ADDRESS IN MEMORY
 95 001102  010337  172416                   MOV     R3,     @#OPREG         ;SEL BUFFER
 96 001106  012737  001400  172416           MOV     #768.,  @#OPREG         ;FALSE RD
 97 001114  105737  172414                   TSTB    @#CSR                   ;DONE ON
 98 001120  100375                           BPL     .-4                     ;NO LOOP
 99 001122  012737  021000  172416           MOV     #8704., @#OPREG         ;FIRE DMA
100 001130  000240                           NOP                             ;DELAY
101 001132  105737  172414                   TSTB    @#CSR                   ;GOOR DMA
102 001136  100401                           BMI     DMAOK
103 001140  000240                           NOP                             ;ERROR IF HERE
104 001142  012737  004400  172416  DMAOK:   MOV     #2304., @#OPREG         ;FALSE WD
105 001150  105737  172414                   TSTB    @#CSR                   ;GOOD WD
106 001154  100375                           BPL     .-4
107 001156  005704                           TST     R4                      ;WAS CRC OK
108 001160  100001                           BPL     END$                    ;YES
109 001162  010402                           MOV     R4,     R2              ;NO FLAG IT
110 001164  000207                  END$:    RTS     PC                      ;RETURN
111 001166  000000                  CAUSE:   .WORD   0                       ;STATUS BYTE 0 HOLDER
112 001170  000000                  AREA:    .WORD   0
113 001172  001174'                          .WORD   DATA
114 001174  000000                  DATA:    .WORD   0
```

```
                                .ENABLE LSB
                        ;*****************************************************
                        ;**** PROCEDURE ENABLE                              *
                        ;*****************************************************

001176  017501  000002         ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
001202  022701  000001                 CMP     #1,     R1
001206  001404                         BEQ     E1$
001210  042737  040000  172414  E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
001216  000207                         RTS     PC
001220  052737  040000  172414  E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
001226  000207                         RTS     PC

                                ;FORTRAN ENCODE/DECODE MACRO

001230  017575  000004  000002  ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
001236  000207                         RTS     PC
001240  017575  000002  000004  DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECOD  BYTE COUNT
001246  000207                         RTS     PC


                        ;*****************************************************
                        ;**** PROCEDURE LINE SWITCH                         *
                        ;*****************************************************

                                ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                ;
                                ;04=SET PRIMARY LINE
                                ;32=RESET PRIMARY LINR
                                ;08=SET BACKUP LINE
                                ;64=RESET BACKUP LINE

001250  017501  000002         SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
001254  012737  010420  172416         MOV     #4368., @#OPREG                 ;WCR : MODSTAT
001262  062701  004400                 ADD     #2304., R1
001266  010137  172416                 MOV     R1,     @#OPREG                 ;RD
001272  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
001276  100375                         BPL     .-4                             ;NO LOOP UNTIL READY
001300  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
001304  000207                         RTS     PC
```

```
 1                                  ;*******************************************************
 2                                  ;**** PROCEDURE STATUS                                *
 3                                  ;*******************************************************
 4
 5                                          ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
 6                                          ;
 7                                          ;X=0 STATUS BYTE 0
 8                                          ;X=1 STATUS BYTE 1
 9
10 001306  017501  000002          STATUS: MOV     @ARG1(R5), R1                   ;WHICH BYTE
11 001312  022701  000001                  CMP     #1,     R1                      ;COMPARE
12 001316  001415                          BEQ     2$
13 001320  012737  010400  172416  1$:     MOV     #4352., @#OPREG                 ;WCR : RS(0)
14 001326  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
15 001334  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
16 001340  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
17 001344  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
18 001350  000207                          RTS     PC
19 001352  012737  010600  172416  2$:     MOV     #4480., @#OPREG                 ;WCR : RS(1)
20 001360  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
21 001366  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
22 001372  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
23 001376  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
24 001402  000207                          RTS     PC
25 001404  012737  010400  172416  STATB0: MOV     #4352., @#OPREG                 ;STATUS BYTE 1
26 001412  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
27 001422  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
28 001424  042701  177400                  BIC     #177400, R1
29 001430  132701  000010                  BITB    #BIT03, R1                      ;BUFFER FULL
30 001434  001404                          BEQ     3$
31 001436  012775  000001  000002          MOV     #1,     @ARG1(R5)               ;LOAD RETURN
32 001444  000207                          RTS     PC
33 001446  012775  000000  000002  3$:     MOV     #0,     @ARG1(R5)               ;LOAD RETURN
34 001454  000207                          RTS     PC
35
36
```

```
                                ;*******************************************************
                                ;**** PROCEDURE WRITE TOKEN (WTOKEN)                   *
                                ;*******************************************************

                                        ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP

001456  012737  010600  172416  WTOKEN: MOV     #4480., @#OPREG                 ;READ STATUS(1)
001464  012737  002422  172416          MOV     #1280., @#OPREG                 ;RS
001472  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001476  132701  000010                  BITB    #BIT03, R1                      ;BUFFER LOADED
001502  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001504  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001512  012737  004423  172416          MOV     #2323., @#OPREG                 ;MARK OB0F/OB1F/WTCMD
001520  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001524  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001526  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001532  000207                          RTS     PC
001534  012737  010740  172416  WTKG:   MOV     #4576., @#OPREG                 ;SELECT BUFFER 1
001542  012737  001634' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001550  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001554  005402                          NEG     R2                              ;FORMAT IT
001556  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001562  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001570  000240                          NOP                                     ;DELAY
001572  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001576  000240                          NOP                                     ;ERROR IF HERE
001600  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001604  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001612  012737  004422  172416          MOV     #2322., @#OPREG                 ;MARK OB1F/WTCMD
001620  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001624  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001626  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001632  000207                          RTS     PC                              ;RETURN
001634     000     377          WTD:    .BYTE 0,377                             ;A WTOKEN

                                ;*******************************************************
                                ;**** PROCEDURE TIMER (TIME)                           *
                                ;*******************************************************

001636  005267  001006' TIME:   INC     LTIME                           ;TIME+1
001642  022767  077777  001006'          CMP     #077777, LTIME                  ;?TIME FULL
001650  001002                          BNE     4$
001652  005067  001006'                  CLR     LTIME                           ;CLEAR CLOCK
001656  000002                  4$:     RTI
```

```
                                ;*********************************************************
                                ;**** PROCEDURE ACRAM                                    *
                                ;*********************************************************

                                        ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
                                        ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
                                        ;
                                        ;0110   =NREAD
                                        ;0100   =DREAD
                                        ;0111   =NULL
                                        ;0000   =WTOKEN


001660  017500  000002         RAM:    MOV     @ARG1(R5), R0                   ; RAM
001664  022700  000000                 CMP     #0,     R0                      ;WHICH OPERATION
001670  001035                        BNE     RDRAM                           ;READ OP
001672  017500  000004         WTRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS
001676  017501  000006                 MOV     @ARG3(R5), R1                   ;WRITE DATA
001702  012737  010402  172416         MOV     #4354., @#OPREG                 ;SEL LDACTR
001710  062700  004400                 ADD     #2304., R0                      ;WD/DATA (ADDR)
001714  010037  172416                 MOV     R0,     @#OPREG                 ;WRITE DATA
001720  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
001724  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
001726  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
001732  012737  010401  172416         MOV     #4353., @#OPREG                 ;SEL ACRAM
001740  062701  004400                 ADD     #2304., R1                      ;WD/DATA (CMD)
001744  010137  172416                 MOV     R1,     @#OPREG                 ;WRITE
001750  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
001754  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
001756  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
001762  000207                        RTS     PC
001764  017500  000004         RDRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS TO READ
001770  012737  010402  172416         MOV     #4354., @#OPREG                 ;SEL LDACTR
001776  062700  004400                 ADD     #2304., R0                      ;WD/DATA (ADDR)
002002  010037  172416                 MOV     R0,     @#OPREG                 ;WRITE DATA
002006  105737  172414                 TSTB    @#CSR                           ;VALID WRITE
002012  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
002014  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002020  012737  010401  172416         MOV     #4353., @#OPREG                 ;SEL ACRAM
002026  012737  001402  172416         MOV     #768.,  @#OPREG                 ;READ DATA
002034  105737  172414                 TSTB    @#CSR                           ;VALID READ
002040  100375                        BPL     .-4                             ;NO LOOP UNTIL READY
002042  105037  172414                 CLRB    @#CSR                           ;CLEAR DONE BIT
002046  013700  172416                 MOV     @#IOBUF, R0                     ;FETCH DATA
002052  042700  177760                 BIC     #177760, R0                     ;CLEAR BITS
002056  010075  000006                 MOV     R0, @ARG3(R5)                   ;RTN DATA
002062  000207                        RTS     PC
```

```
                                        .ENABLE LSB
                                ;*****************************************************
                                ;**** PROCEDURE WRITE LOOP (LPOUT)                  *
                                ;*****************************************************

002064  017502  000002          LPOUT:  MOV     @ARG1(R5), R2           ;BYTE COUNT
002070  012700  000340                  MOV     #340,   R0              ;PRI=7
002074  106400                          MTPS    R0
002076  012737  000000' 172410          MOV     #OUTBF, @#BAR           ;BUS ADDRESS
002104  005402                          NEG     R2                      ; 2'S COMP COUNT
002106  010237  172412                  MOV     R2,     @#WCR           ;COUNT
002112  012737  010540  172416          MOV     #4448., @#OPREG         ;OB0 COMMAND
002120  012737  024000  172416          MOV     #10240., @#OPREG        ;DMA GO
002126  000240                          NOP                             ;INTER ACE TIME
002130  105737  172414                  TSTB    @#CSR                   ;DMA OK
002134  000240                          NOP                             ;ERROR IF HERE
002136  105037  172414                  CLRB    @#CSR
002142  012737  010740  172416          MOV     #4576., @#OPREG         ;OB1 COMMAND
002150  012737  001634' 172410          MOV     #WTD,   @#BAR           ;ADDRESS OF WRITE TOKEN
002156  012702  000002                  MOV     #2,     R2              ;BYTE COUNT
002162  005402                          NEG     R2                      ;FOR AT FOR BLIUI
002164  010237  172412                  MOV     R2,     @#WCR           ;DMA BYTE COUNT
002170  012737  024000  172416          MOV     #10240.,@#OPREG         ;FIRE DMA
002176  000240                          NOP                             ;DELAY
002200  105737  172414                  TSTB    @#CSR                   ;GOOD DMA
002204  000240                          NOP                             ;ERROR IF NOT
002206  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE
002212  012737  010420  172416          MOV     #4368., @#OPREG         ;MODSTAT
002220  012737  004403  172416          MOV     #2307., @#OPREG         ;BUFFERS FULL
002226  105737  172414                  TSTB    @#CSR                   ;GOOD WRITE
002232  100375                          BPL     .-4                     ;NO LOOP UNTIL
002234  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
002240  012700  000000                  MOV     #000,   R0              ;PRI=0
002244  106400                          MTPS    R0                      ;LOWER CPU
002246  000207                          RTS     PC
```

```
                                ;*******************************************************
                                ;**** PROCEDURE READ BUFFER POINTER                    *
                                ;*******************************************************

002250  017500  000002          RDPNT:  MOV     @ARG1(R5), R0                   ;FETCH COMMAND
002254  022700  000000                  CMP     #0,     R0                      ;INBUF0 ?
002260  001004                          BNE     1$                              ;NO
002262  012700  010410                  MOV     #4360., R0                      ;RDBUFADR IN0
002266  000167  000046                  JMP     4$
002272  022700  000001          1$:     CMP     #1,     R0                      ;INBUF1 ?
002276  001004                          BNE     2$                              ;NO
002300  012700  010610                  MOV     #4488., R0                      ;RDBUFADR IN1
002304  000167  000030                  JMP     4$
002310  022700  000002          2$:     CMP     #2,     R0                      ;OUTBUF0 ?
002314  001004                          BNE     3$                              ;NO
002316  012700  010510                  MOV     #4424., R0                      ;RDBUFADR OUT0
002322  000167  000012                  JMP     4$
002326  022700  000004          3$:     CMP     #4,     R0                      ;OUTBUF1 ?
002332  001020                          BNE     5$                              ;NO RETURN
002334  012700  010710                  MOV     #4552., R0                      ;RDBUFADR OUT1
002340  010037  172416          4$:     MOV     R0,     @#OPREG                 ;RDBUFADR
002344  012737  001400  172416        MOV     #768.,  @#OPREG                 ;RD
002352  105737  172414                  TSTB    @#CSR                           ;READY
002356  100375                          BPL     .-4                             ;NO LOOP UNTIL
002360  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH POINTER
002364  042700  177400                  BIC     #177400, R0                     ;CLEAR MST
002370  010075  000004                  MOV     R0,     @ARG2(R5)               ;RETURN POINTER
002374  000207          5$:     RTS     PC
```

```
                                ;*******************************************************
                                ;**** PROCEDURE RESTART                                *
                                ;*******************************************************

                                        ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
                                        ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
                                        ;CALL RSTART(2) -LOAD MODE(173000)

002376  017500  000002          RSTART: MOV     @ARG1(R5), R0                   ;MODE
002402  022700  000000          RS0:    CMP     #0,     R0                      ;LOAD ADDRESS
002406  001002                          BNE     RS1
002410  000000                          HALT
002412  000207                          RTS     PC
002414  022700  000001          RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
002420  001005                          BNE     RS2
002422  004767  175432                  JSR     PC,     LIUINT                  ;INT LIU FIRST
002426  012700  000040          G4$:    MOV     #40,    R0                      ;START ADDRESS
002432  000110                          JMP     (R0)
002434  022700  000002          RS2:    CMP     #2,     R0                      ;LOAD MODE
002440  001002                          BNE     RST
002442  000177  000002                  JMP     @ROM
002446  000207                  RST:    RTS     PC
002450  173000                  ROM:    .WORD   173000
```

```
                                ;*********************************************************
                                ;****        COMMON DATA AREA                             *
                                ;*********************************************************

000000                                  .PSECT  DFM,RW,D,GBL,REL,OVR

000000                          OUTBF:  .BLKB   256.
000400                          INBF:   .BLKB   256.
001000                          IWRTTM: .BLKW
001002                          IWRT:   .BLKW
001004                          STAT:   .BLKW
001006                          LTIME:  .BLKW
                                        .EVEN

000000                                  .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                          XOUTBF: .BLKB   256.
000400                          XINBF:  .BLKB   256.
001000                          ACKSEQ: .BLKB   256.
001400                          LIDFD:  .BLKB   256.
002000                          IOFLG:  .BLKW
002002                          LLFLG:  .BLKW
002004                          IRSEND: .BLKW
                                        .EVEN


000001                                  .END
```

SYMBOL TABLE

```
ACKSEQ  001000R     003 B0$     000626R       ENSTR   001230RG      LPINPT= ****** G      RTI$    000762R
AREA    001170R         B1$     000704R       E0$     001210R       LPOUT   002264RG      STAT    001004R     002
ARG1  = 000002          B2$     000472R       E1$     001220R       LTIME   001006R   002 STATB0  001404RG
ARG2  = 000004          B3$     000516R       G4$     002426R       MASTER  000000RG      STATUS  001306RG
ARG3  = 000006          B4$     000542R       HRBUF = 177562        NDXX  = 000000        STAT$   000366R
ARG4  = 000010          B5$     000570R       HRCSR = 177560        ND24  = 000001        SWITCH  001250RG
BAR   = 172410          B6$     000606R       HXBUF = 177566        NHOST = 000000        TIME    001636RG
BIT00 = 000001          B7$     000552R       HXCSR = 177564        OPREG = 172416        WCR   = 172412
BIT01 = 000002          CAUSE   001166R       INBF    000400R   002 OUTBF   000000R   002 WTD     001634R
BIT02 = 000004          CRCOK   001036R       IOBUF = 172416        RAM     001660RG      WTKG    001534R
BIT03 = 000010          CSR   = 172414        IOFLG   002000R   003 RDPNT   002250RG      WTOKEN  001456RG
BIT04 = 000020          DATA    001174R       IRSEND  002004R   003 RDRAM   001764R       WTRAM   001672R
BIT05 = 000040          DESTR   001240RG      IWRT    001002R   002 ROM     002450R       XCIO  = 000001
BIT06 = 000100          DHOST = 000001        IWRTTM  001000R   002 RST     002446R       XGIO  = 000001
BIT07 = 000200          DMAOK   001142R       LIDFD   001400R   003 RSTART  002376RG      XINBF   000400R     003
BIT08 = 000400          EMBF    001006R       LIO     000412RG      RS0     002402R       XOUTBF  000000R     003
BIT14 = 040000          ENABLE  001176RG      LIUINT  000060RG      RS1     002414R       XSIO  = 000001
BIT15 = 100000          END$    001164R       LLFLG   002002R   003 RS2     002434R       ZEROBP  000266R

. ABS.  000000     000
        002452     001
DFM     001010     002
BUFS    002006     003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  56 PAGES
LK:FDM,DK:FDM=DK:FDM.M27
```

```
FOR NODAL/NOLINE
FOR NODAL1/NOLINE
FOR FIAC/NOLINE
MAC FDM
```

```
ASSIGN DX0: DK:
R LINK
DX1:NODE27,DX1:MAP=DX1:NODAL/C/I/W
DX1:FDM/C
DX1:NODAL1/C
DX1:FIAC27
$SIMKI

ASSIGN DX1: DK:
```

1.9 NODE 28 (SDCA)

The Switch Data Collection and Analysis (SDCA) module receives switch traffic data generated by AUTODIN or AUTOVON switches, and performs loading assessments on this data to detect switch equipment saturation conditions. Traffic flow control computations and actions are performed. Simulated data to represent two switches is generated by the PDP 11/40 in loop 2. A switch condition report is sent by the PDP 11/40 to the SDCA approximately every 2.5 seconds.

SDCA Switch Condition Report

Field	Description	Critical Values AUTODIN	AUTOVON
1	Switch #	1	2
2	# of Transactions	512	256
3	# of Blocked Transactions	25	10
4	Transaction Queue Depth	25	10
5	# of Prescripted Transactions	10	25
6	Trunk Group Occupancy	50	40
7	Trunk Group Overflow	50	40
8	Message Delay (sec.)	10	5
9	Maximum Message Age (sec.)	10	5
10	Number of Overflow Messages	10	10
11	# of Senders	128	64
12	# of Markers	128	64
13	# of Receivers	128	64
14	# of Pooled Crypto Units	128	64
15	Service Time for Dial Tone (sec.)	10	5
16	Service Time for Crypto Unit (sec.)	10	5

Whenever a critical value of the table above is exceeded and the Event Reporting Condition Parameter is ON, the switch is considered to be in a saturated condition and Red Event Reports are sent to the DBMS Status File and the destination Node Designator for OCRI reporting.

The SDCA interprets commands from the DBMS similar to the VSQC, DSQC and BWBSA except that measurements are performed for 2 switches rather than channels or links.

1.9.1 Program Descriptions

1.9.1.1 Refer to Section 1.1 for descriptions of routines NODAL, IGETSP, ENQUE, DEQUE, ACKNAK, INPTQ, LPINPT, INIT, LINLOS, MASTER, LIUINT, LIO, ENABLE, SWITCH, STATUS, WTOKEN, TIME RAM, LPOUT, RDPNT, RSTART.

1.9.1.2 Subroutine SDCA (FORTRAN)

This subroutine is called when a measurement is received for the 11/40. The simulated inputs are decoded and the function described in Section 1.9 is performed.

1.9.1.3 Subroutine SDCALP (FORTRAN)

This subroutine is called when a message is received from the loop. Functions performed are reporting on or off, and request a measurement.

1.9.1.4 Subroutine IMAGE (FORTRAN)

This subroutine converts a decimal number into ASCII format.

1.9.1.5 Subroutine SIO (MACRO)

This subroutine is the interupt handler for the 11/40 interface. It reads the message of input from the 11/40 and set a flag containing the byte count.

```
0001          PROGRAM NODAL
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0003          INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP
0005          INTEGER*2 RSTBKP,Q1,Q2,RESNLM,OUTQ,DEQUE
0006          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0007          LOGICAL*1 SAVE1,SAVE2
0008          LOGICAL*1 IDATA,ENC
0009          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0010          INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTTM,TIMLIM,ATIMLM
0011          COMMON /MESS/ MESSEQ
0012          COMMON /DFM/ OUTBF(256),INBF(256),
             1        IWRTTM,IWRT,STAT,LTIME
0013          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0014          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1          LIDFD(256),IOFLG,LLFLG,IRSEND
0015          COMMON /FRE/ FREE(64),IFR,IFRSZ
0016          COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0017          COMMON       PACK(256,64)
0018          COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0019          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0020          COMMON/SCOM/SAVE1(64),SAVE2(64),ITERM,ICOUNT,I1,I2,IDATA(256),
             &           ISWTCH
0021          COMMON/SCOM1/ISIM,IVC,IDC,IBC,ICC
0022          COMMON/END/ENC(4)
0023          CALL MASTER
0024          CALL INIT
0025          CALL ENABLE(1)
0026      5   CONTINUE
0027      25  IF(PINQ(1) .LT. PINQ(2)) GOTO 40
0029          CALL ENABLE(0)
0030          Q2=DEQUE(PINQ,INQ,1)
0031          CALL ENABLE(1)
0032          DO 30 I=1,100
0033            IDATA(I)=PACK(I+6,Q2)
0034      30  CONTINUE
0035          CALL SDCALP
0036          IFR=IFR+1
0037          FREE(IFR)=Q2
0038      40  CALL STATB0(IS)
0039          IF(IS .EQ. 1) GOTO 100
0041          IF(IOFLG .GT. 0 .AND. IRSEND .EQ. 0) CALL SDCA
0043          IF(LLFLG .GT. 0) CALL LINLOS
0045      80  IF(OUTFCT .EQ. 1) GOTO 100
0047          IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0049          IF(IRSEND .EQ. 1) GOTO 85
0051          IF(PXINQ(1) .LT. PXINQ(2)) GOTO 100
0053          CALL ENABLE(0)
0054          Q1=DEQUE(PXINQ,XINQ,1)
0055          CALL ENABLE(1)
0056          OUTQ=Q1
0057      85  CALL DESTR(PACK(254,OUTQ),Q2)
```

AD-A078 391 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66157 SBIE-AD-E100 313 NL

4 OF 5
AD A078391

```
0058          DO 90 I=1,Q2
0059            OUTBF(I)=PACK(I,OUTQ)
0060     90   CONTINUE
0061          OUTBF(Q2+1)=0
0062          IPT=OUTBF(5)
0063          OUTBF(Q2+2)=LIDFD(IPT)
0064          IFULL=1
0065          ISENT=1
0066          CALL LPOUT(Q2+2)
0067          IRSEND=1
0068          ACKTIM=0
0069          IWRTTM=0
0070          IWRT=1
0071    100   INFLCT=0
0072    110   OLDTIM=NEWTIM
0073          NEWTIM=LTIME
0074          IF(IWRT .EQ. 0) GOTO 120
0076          IWRTTM=IWRTTM+(NEWTIM-OLDTIM)
0077          IF(IWRTTM .LT. TIMIIM) GOTO 120
0079          CALL WTOKEN
0080          IWRTTM=0
0081          IWRT=0
0082    120   IF(ISENT .EQ. 0) GOTO 130
0084          ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0085          IF(ACKTIM .LT. ATIMLM) GOTO 130
0087          CALL ENABLE(0)
0088          CALL ACKNAK(0)
0089          CALL ENABLE(1)
0090    130   CONTINUE
0091          GOTO 5
0092          END
```

```
0001          FUNCTION IGETSP(N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 FREE,FLWCNT
0004          INTEGER*2 OUTFCT,OUTQ,RESNLM
0005          COMMON /FRE/ FREE(64),IFR,IFRSZ
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
      C
0007          IF(IFR .LT. 1) CALL INIT
0009          IGETSP=FREE(IFR)
0010          IFR=IFR-1
0011          RETURN
0012          END
```

```
0001          SUBROUTINE ENQUE(A,B,N)
0002          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005          INTEGER*2 RESNLM,OUTFCT
0006          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0007          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008       10 IQHEAD=A(1)
0009          IQTAIL=A(2)
0010          IF(IQTAIL .EQ. 1) GOTO 20
0012          IQTAIL=IQTAIL-1
0013          B(IQTAIL)=N
0014          A(2)=IQTAIL
0015          GOTO 999
0016       20 IF(IQHEAD .GE. (IQLNTH)) GOTO 40
0018          NN=IQHEAD-IQTAIL
0019          DO 30 I=1,NN+1
0020       30 B(IQLNTH+1-I)=B(IQHEAD+1-I)
0021          A(1)=IQLNTH
0022          A(2)=IQLNTH-NN
0023          GOTO 10
0024       40 CALL INIT
0025      999 RETURN
0026          END
```

```
0001          FUNCTION DEQUE(A,B,ID)
0002          LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004          INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005          COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0006          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007          IQHEAD=A(1)
0008          DEQUE=B(IQHEAD)
0009          IF(ID .NE. 1) GOTO 999
0011          IF(IQHEAD .NE. 0) GOTO 10
0013          A(1)=IQLNTH
0014          A(2)=IQLNTH+1
0015          GOTO 999
0016    10    A(1)=IQHEAD-1
0017   999    RETURN
0018          END
```

```
0001          SUBROUTINE ACKNAK(N)
0002          INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ,PXOUTQ
0003          INTEGER*2 ACKQ,PACKQ,PINQ,FREE,OUTQ,XOUTQ
0004          INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006          LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008          INTEGER*2 LTIME,IWRTTM
0009          REAL*8 COMT(5)
0010          EQUIVALENCE(COMT,LCOMT)
0011          DATA COMT(1)/'MESSAGE '/,COMT(2)/'NOT SENT'/,COMT(3)/' FROM   '/,
             1    COMT(4)/'NODE    '/,COMT(5)/' TO NODE'/
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON /DFM/ OUTBF(256),INBF(256),
             1       IWRTTM,IWRT,STAT,LTIME
0014          COMMON         PACK(256,64)
0015          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
             1          PACKQ(2),INQ(16),PINQ(2)
0016          COMMON /FRE/  FREE(64),IFR,IFRSZ
0017          COMMON /MESS/ MESSEQ
0018          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1    IOFLG,LLFLG,IRSEND
0019          IF (N .NE. 1) GOTO 10
0021          IFR=IFR+1
0022          DO 5 I=1,256
0023            PACK(I,OUTQ)=0
0024       5  CONTINUE
0025          IFULL=0
0026          ISENT=0
0027          IRSEND=0
0028          FREE(IFR)=OUTQ
0029          GOTO 999
0030      10  T2=PACK(256,OUTQ)
0031          T2=T2+1
0032          IF(T2 .LE. RESNLM) GOTO 120
0034          T3=PACK(3,OUTQ)
0035          T4=MOD(T3,64)
0036          IF(T4 .LT. 32) GOTO 110
0038          T5=IGETSP(N)
0039          IF(MESSEQ .EQ. 126) MESSEQ=0
0041          MESSEQ=MESSEQ+1
0042          PACK(1,T5)=0
0043          PACK(2,T5)=MESSEQ
0044          PACK(3,T5)=0
0045          PACK(4,T5)=0
0046          PACK(5,T5)=25
0047          PACK(6,T5)=ISLID
0048          DO 20 I=7,60
0049            PACK(I,T5)="040
0050      20  CONTINUE
0051          DO 30 I=7,9
0052            PACK(I,T5)=LF
0053      30  CONTINUE
```

```
0054          DO 40 I=1,8
0055            PACK(I+9,T5)=LCOMT(I)
0056     40   CONTINUE
0057          DO 50 I=9,16
0058            PACK(I+9,T5)=LCOMT(I)
0059     50   CONTINUE
0060          DO 60 I=17,22
0061            PACK(I+9,T5)=LCOMT(I)
0062     60   CONTINUE
0063          DO 70 I=25,29
0064            PACK(I+7,T5)=LCOMT(I)
0065     70   CONTINUE
0066          ENCODE(3,80,PACK(38,T5))ISLID
0067     80   FORMAT(I3)
0068          DO 90 I=33,40
0069            PACK(I+11,T5)=LCOMT(I)
0070     90   CONTINUE
0071          ENCODE(3,80,PACK(52,T5))PACK(5,OUTQ)
0072          PACK(55,T5)=LF
0073          PACK(56,T5)=LF
0074          PACK(57,T5)=CR
0075          PACK(58,T5)=ETX
0076          CALL ENSTR(PACK(254,T5),58)
0077          CALL ENQUE(PXINQ,XINQ,T5)
0078          DO 102 I=1,256
0079            PACK(I,OUTQ)=0
0080    102   CONTINUE
0081          ISENT=0
0082          IRSEND=0
0083          IFULL=0
0084          IERT=0
0085          IFR=IFR+1
0086          FREE(IFR)=OUTQ
0087          GOTO 999
0088     110  PACK(3,OUTQ)=PACK(3,OUTQ)+32
0089          PACK(256,OUTQ)=0
0090          ISENT=0
0091          IFULL=0
0092          IF(IRSEND .EQ. 1) GOTO 999
0094          CALL ENQUE(PXINQ,XINQ,OUTQ)
0095          GOTO 999
0096    120   PACK(256,OUTQ)=T2
0097          IFULL=0
0098          ISENT=0
0099          IF(IRSEND .EQ. 1) GOTO 999
0101          CALL ENQUE(PXINQ,XINQ,OUTQ)
0102    999   RETURN
0103          END
```

```
0001          SUBROUTINE INPTQ(L)
0002          INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
0003          INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005          LOGICAL*1 PACK,MONTOR,ISLID,ETX,CR,LF,DUM
0006          INTEGER*2 LTIME,IWRTTM
0007          COMMON /DFM/ OUTBF(256),INBF(256),
             1        I*RTTM,IWRT,STAT,LTIME
0008          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
             1           LIDFD(256),IOFLG,LLFLG,IRSEND
0009          COMMON         PACK(256,64)
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          IF(L .LT. 1) GOTO 30
0014          LI=L-2
0015          II=INBF(6)
0016          IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0018          T1=IGETSP(N)
0019          DO 10 I=1,LI
0020     10   PACK(I,T1)=INBF(I)
0021          CALL ENSTR(PACK(254,T1),LI)
0022          CALL ENQUE(PINQ,INQ,T1)
0023          N=2
0024          GOTO 40
0025     20   N=1
0026          GOTO 40
0027     30   N=0
0028     40   DO 50 I=1,2
0029            OUTBF(I)=INBF(I)
0030     50   CONTINUE
0031          NN=1
0032          IF(N .EQ. 0) NN=128
0034          OUTBF(3)=NN
0035          OUTBF(4)=0
0036          OUTBF(5)=INBF(6)
0037          OUTBF(6)=ISLID
0038          OUTBF(7)=ETX
0039          OUTBF(8)=0
0040          OUTBF(9)=LIDFD(INBF(6))
0041          CALL LPOUT(9)
0042          IF(N .NE. 2) GOTO 999
0044          ACKSEQ(II)=INBF(2)
0045    999   RETURN
0046          END
```

```
0001          SUBROUTINE LPINPT(LI)
0002          INTEGER*2 STAT,FLWCNT,OUTFCT,OUTQ,RESNLM
0003          INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004          LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005          INTEGER*2 LTIME,IWRTTM
0006          COMMON/DFM/OUTBF(256),INBF(256),I*RTTM,I*RT,STAT,LTIME
0007          COMMON /GLOB/ ISENT,FL*CNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008          CC1=INBF(3)
0009          CC2=INBF(4)
0010          IF(CC1 .LT. 0) GOTO 25
0012          IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0014          CC3=MOD(CC1,2)
0015          CC4=MOD(CC1,256)
0016          IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0018          IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0020          CC5=MOD(INBF(3),64)
0021          IF(CC5 .GE. 32) CALL INPTQ(LI)
0023          GOTO 99
0024     20   CC7=MOD(CC1,2)
0025          IF(CC7 .GE. 1) GOTO 30
0027     25   CALL ACKNAK(0)
0028          GOTO 99
0029     30   CALL ACKNAK(1)
0030          GOTO 99
0031     40   CALL INPTQ(LI)
0032     99   CONTINUE
0033          RETURN
0034          END
```

```
0001          SUBROUTINE INIT
0002          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0003          INTEGER*2 STAT,FLWCNT,OUTFCT
0004          INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0005          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0006          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM,IDATA
0007          LOGICAL*1 SAVE1,SAVE2
0008          INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTM
0009          COMMON /DFM/ OUTBF(256),INBF(256),
             1     IWRTTM,IWRT,STAT,LTIME
0010          COMMON /QUE/  XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             1              ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0011          COMMON /MESS/ MESSEQ
0012          COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             1     IOFLG,LLFLG,IRSEND
0013          COMMON /FRE/  FREE(64),IFR,IFRSZ
0014          COMMON /TIM/  OLDTIM,TIMLIM,ACKTIM,ATIMLM
0015          COMMON        PACK(256,64)
0016          COMMON /SWT/  SETPRM,RSTPRM,SETBKP,RSTBKP
0017          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0018          COMMON/SCOM/SAVE1(64),SAVE2(64),ITERM,ICOUNT,I1,I2,IDATA(256),
             &              ISWTCH
0019          COMMON/SCOM1/ISIM,IVC,IDC,IBC,ICC
0020          CALL RAM(0,2,4)
0021          CALL RAM(0,255,0)
0022          I1=0
0023          I2=0
0024          ICOUNT=0
0025          ISIM=5
0026          IVC=0
0027          IDC=500
0028          IBC=0
0029          ICC=0
0030          MESSEQ=0
0031          IOFLG=0
0032          IRSEND=0
0033          LLFLG=0
0034          IFRSZ=64
0035          INFLCT=0
0036          ISLID=28
0037          CR="015
0038          LF="012
0039          ETX="003
0040          RESNLM=2
0041          IQLNTH=16
0042          IALTRT=0
0043          FLWCNT=0
0044          IWRTTM=0
0045          TIMLIM=50
0046          SETPRM=4
0047          RSTPRM=32
0048          SETBKP=8
0049          RSTBKP=64
```

```
0050          ATIMLM=200
0051          MONTOR=25
0052          IWRT=0
0053          ISWTCH=2
0054          OUTFCT=0
0055          IFULL=0
0056          ISENT=0
0057          DO 40 I=1,20
0058       40 LIDFD(I)=4
0059          LIDFD(21)=1
0060          LIDFD(22)=3
0061          LIDFD(23)=6
0062          LIDFD(24)=5
0063          LIDFD(25)=7
0064          LIDFD(26)=8
0065          LIDFD(27)=9
0066          LIDFD(28)=2
0067          DO 50 I=29,39
0068       50 LIDFD(I)=0
0069          DO 60 I=40,44
0070       60 LIDFD(I)=1
0071          DO 70 I=45,59
0072       70 LIDFD(I)=0
0073          DO 80 I=60,64
0074       80 LIDFD(I)=2
0075          DO 90 I=65,79
0076       90 LIDFD(I)=0
0077          DO 100 I=80,84
0078      100 LIDFD(I)=4
0079          DO 110 I=85,99
0080      110 LIDFD(I)=0
0081          DO 120 I=100,104
0082      120 LIDFD(I)=5
0083          DO 130 I=105,256
0084      130 LIDFD(I)=0
0085          PINQ(1)=IQLNTH
0086          PINQ(2)=IQLNTH+1
0087          PACKQ(1)=IQLNTH
0088          PACKQ(2)=IQLNTH+1
0089          PXOUTQ(1)=IQLNTH
0090          PXOUTQ(2)=IQLNTH+1
0091          PXINQ(1)=IQLNTH
0092          PXINQ(2)=IQLNTH+1
0093          DO 10 I=1,IFRSZ
0094            PACK(255,I)=0
0095       10 PACK(256,I)=0
0096          DO 20 I=1,IFRSZ
0097       20 FREE(I)=I
0098          DO 30 I=1,256
0099       30 ACKSEQ(I)=256
0100          IFR=IFRSZ
0101          RETURN
0102          END
```

```
0001          SUBROUTINE LINLOS
0002          REAL*8 RM1(5),LIN08,LIN18
0003          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNUM,T1
0004          INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,PINQ
0005          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006          LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0007          LOGICAL*1 PACK,M1(40),LIN0(8),LIN1(8)
0008          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &          ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0009          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &          IOFLG,LLFLG,IRSEND
0010          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNUM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011          COMMON         PACK(256,64)
0012          COMMON /MESS/ MESSEQ
0013          DATA LIN08,LIN18/'PRIMARY ','BACKUP  '/
0014          DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'/
0015          DATA RM1(3),RM1(4)/'ON ON LO','OP AT NO'/
0016          DATA RM1(5)/'DE 28   '/
0017          EQUIVALENCE(LIN0,LIN08)
0018          EQUIVALENCE(LIN1,LIN18)
0019          EQUIVALENCE(M1,RM1)
0020          IS=LLFLG
0021          CALL ENABLE(0)
0022          T1=IGETSP(N)
0023          CALL ENABLE(1)
0024          PACK(1,T1)=0
0025          IF(MESSEQ .EQ. 126) MESSEQ=0
0027          MESSEQ=MESSEQ+1
0028          PACK(2,T1)=MESSEQ
0029          PACK(3,T1)=0
0030          PACK(4,T1)=0
0031          PACK(5,T1)=25
0032          PACK(6,T1)=ISLID
0033          DO 20 I=7,9
0034            PACK(I,T1)=LF
0035     20   CONTINUE
0036          DO 30 I=1,22
0037     30   PACK(I+9,T1)=M1(I)
0038          IF(IS .EQ. 1) GOTO 50
0040          DO 40 I=1,8
0041     40   PACK(I+31,T1)=LIN0(I)
0042          GOTO 70
0043     50   DO 60 I=1,8
0044     60   PACK(I+31,T1)=LIN1(I)
0045     70   DO 80 I=23,40
0046     80   PACK(I+17,T1)=M1(I)
0047          PACK(58,T1)=CR
0048          PACK(59,T1)=LF
0049          PACK(60,T1)=ETX
0050          CALL ENSTR(PACK(254,T1),60)
0051          CALL ENABLE(0)
0052          CALL ENQUE(PXINQ,XINQ,T1)
0053          CALL ENABLE(1)
```

```
0054          LLFLG=0
0055          RETURN
0056          END
```

```
0001          SUBROUTINE SDCA
0002          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,IIDFD,OUTBF,INBF
0003          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0004          LOGICAL*1 SAVE1,SAVE2,BHEAD(24),IDATA,ENC
0005          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0006          INTEGER*2 PINQ,K1,K2,K3,AME(16),AV(16),AD(16)
0007          INTEGER*2 OUTFCT,RESNLM,FLWCNT,OUTQ
0008          INTEGER*2 MESSEQ
0009          INTEGER*4 I14
0010          REAL*4    VMEAS
0011          REAL*8    RHEAD(3),VTRNK,DTRNK,BWID
0012          EQUIVALENCE(RHEAD,BHEAD)
0013          DATA AV/3,256,10,10,25,40,40,5,5,10,64,64,64,64,5,5/
0014          DATA AD/3,512,25,25,10,50,50,10,10,10,128,128,128,128,10,10/
0015          DATA RHEAD(1),RHEAD(2)/'SWITCH  ','FIELD   '/
0016          DATA RHEAD(3)          /'VALUE   '/
0017          DATA VTRNK,DTRNK,BWID/'AAAAAAAA','BBBBBBBB','  111110'/
0018          COMMON/MESS/ MESSEQ
0019          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &           ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0020          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(255),IIDFD(256),
             &           IOFIG,LLFLG,IRSEND
0021          COMMON/GLOB/ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             &           RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,IUM
0022          COMMON PACK(256,64)
0023          COMMON/SCOM/SAVE1(64),SAVE2(64),ITERM,ICOUNT,I1,I2,IDATA(256),
             &           ISWTCH
0024          COMMON/SCOM1/ISIM,IVC,IDC,IBC,ICC
0025          COMMON/END /ENC(4)
0026    1     IF(ISWTCH .EQ. 0) GOTO 999
0028          DECODE(4,2,XINBF(1),ERR=999) AME(1)
0029          DECODE(4,2,XINBF(5),ERR=999) AME(2)
0030          DECODE(4,2,XINBF(9),ERR=999) AME(3)
0031          DECODE(4,2,XINBF(13),ERR=999)AME(4)
0032          DECODE(4,2,XINBF(17),ERR=999)AME(5)
0033          DECODE(4,2,XINBF(21),ERR=999)AME(6)
0034          DECODE(4,2,XINBF(25),ERR=999)AME(7)
0035          DECODE(4,2,XINBF(29),ERR=999)AME(8)
0036          DECODE(4,2,XINBF(33),ERR=999)AME(9)
0037          DECODE(4,2,XINBF(37),ERR=999)AME(10)
0038          DECODE(4,2,XINBF(41),ERR=999)AME(11)
0039          DECODE(4,2,XINBF(45),ERR=999)AME(12)
0040          DECODE(4,2,XINBF(49),ERR=999)AME(13)
0041          DECODE(4,2,XINBF(53),ERR=999)AME(14)
0042          DECODE(4,2,XINBF(57),ERR=999)AME(15)
0043          DECODE(4,2,XINBF(61),ERR=999)AME(16)
0044    2     FORMAT(I4)
0045          IF(AME(1) .EQ. 2) GOTO 200
        C
        C     AUTODIN FUNCTION
        C
0047    100   DO 101 J=1,64
0048           SAVE1(J)=XINBF(J)
0049    101   CONTINUE
```

```
0050          DO 102 J=1,16
0051          JJ=J
0052           IF(AME(J) .LT. AD(J)) GOTO 102
0054           GOTO 300
0055    102   CONTINUE
0056          GOTO 999
        C
        C     AUTOVON FUNCTION
        C
0057    200   DO 201 J=1,64
0058           SAVE2(J)=XINBF(J)
0059    201   CONTINUE
0060          DO 202 J=1,16
0061          JJ=J
0062           IF(AME(J) .LT. AV(J)) GOTO 202
0064           GOTO 300
0065    202   CONTINUE
0066          GOTO 999
        C
        C      SEND REPORT TO CRT
        C
0067    300   CALL ENABLE(0)
0068          K1=IGETSP(N)
0069          K2=IGETSP(N)
0070          CALL ENABLE(1)
0071          IF(MESSEQ .EQ. 126) MESSEQ=0
0073          MESSEQ=MESSEQ+1
0074          PACK(1,K1)=0
0075          PACK(2,K1)=MESSEQ
0076          PACK(3,K1)=0
0077          PACK(4,K1)=0
0078          PACK(5,K1)=ITERM
0079          PACK(6,K1)=ISLID
0080          DO 301 J=1,24
0081    301   PACK(J+6,K1)=BHEAD(J)
0082          PACK(31,K1)=CR
0083          PACK(32,K1)=LF
0084          DO 302 J=1,4
0085    322   PACK(J+32,K1)=XINBF(J)
0086          DO 303 J=1,4
0087    303   PACK(J+36,K1)="040
0088          ENCODE(4,304,PACK(41,K1)) JJ
0089    304   FORMAT(I4)
0090          DO 305 J=1,4
0091    305   PACK(J+44,K1)="040
0092          JJI=JJ*4
0093          DO 306 J=1,4
0094    306   PACK(J+48,K1)=XINBF(JJI+J)
0095          PACK(53,K1)=CR
0096          PACK(54,K1)=LF
0097          PACK(55,K1)=ETX
0098          CALL ENSTR(PACK(254,K1),55)
0099          CALL ENABLE(0)
0100          CALL ENQUE(PXINQ,XINQ,K1)
```

```
0101          CALL ENABLE(1)
      C
      C       SEND REPORT TO DBMS
      C
0102          IF(MESSEQ .EQ. 126) MESSEQ=0
0104          MESSEQ=MESSEQ+1
0105          PACK(1,K2)=0
0106          PACK(2,K2)=MESSEQ
0107          PACK(3,K2)=0
0108          PACK(4,K2)=0
0109          PACK(5,K2)=24
0110          PACK(6,K2)=ISLID
0111          PACK(7,K2)="177
0112          DO 307 J=1,4
0113    307   PACK(J+7,K2)=XINBF(J)
0114          PACK(12,K2)="040
0115          ENCODE(4,304,PACK(13,K2)) JJ
0116          PACK(17,K2)="040
0117          DO 309 J=1,4
0118    309   PACK(J+17,K2)=XINBF(JJI+J)
0119          PACK(22,K2)="040
0120          PACK(23,K2)="040
0121          PACK(24,K2)="040
0122          PACK(25,K2)='1'
0123          PACK(26,K2)="040
0124          PACK(27,K2)="040
0125          PACK(28,K2)=CR
0126          PACK(30,K2)=LF
0127          PACK(29,K2)=ETX
0128          CALL ENSTR(PACK(254,K2),30)
0129          CALL ENABLE(0)
0130          CALL ENQUE(PXINQ,XINQ,K2)
0131          CALL ENABLE(1)
      C
      C       PACKET TO FIAC(RANDOM)
      C
0132    400   RN=RAN(I1,I2)
0133          MON=RN*1000
0134          IF(MON .GE. 334 .AND. MON .LE. 666) GOTO 401
0136          GOTO 999
0137    401   ISIM=ISIM+2
0138          IF(ISIM .EQ. 7) ISIM=1
0140          ICC=ICC+1
0141          IF(ICC .EQ. 3)  ICC=1
0143          VMEAS=RAN(I1,I2)
0144          CALL ENABLE(0)
0145          K3=IGETSP(N)
0146          CALL ENABLE(1)
0147          IF(MESSEQ .EQ. 126) MESSEQ=0
0149          MESSEQ=MESSEQ+1
0150          PACK(1,K3)=0
0151          PACK(2,K3)=MESSEQ
0152          PACK(3,K3)=0
0153          PACK(4,K3)=0
```

```
0154          PACK(5,K3)=27
0155          PACK(6,K3)=ISLID
0156          IF(ISIM .EQ. 3) GOTO 410
0158          IF(ISIM .EQ. 5) GOTO 420
0160          ENCODE(8,402,PACK(7,K3)) VTRNK
0161    402   FORMAT(A8)
0162          IVC=IVC+1
0163          IF(IVC .EQ. 501) IVC=1
0165          IS=JICVT(IVC,I14)
0166          CALL IMAGE(I14)
0167          DO 404 J=1,4
0168    404   PACK(J+10,K3)=ENC(J)
0169          ENCODE(2,405,PACK(15,K3)) IVC
0170    405   FORMAT(A2)
0171          GOTO 450
0172    410   ENCODE(8,402,PACK(7,K3)) DTRNK
0173          IDC=IDC+1
0174          IF(IDC .EQ. 1001) IDC=501
0176          IS=JICVT(IDC,I14)
0177          CALL IMAGE(I14)
0178          DO 411 J=1,4
0179    411   PACK(J+10,K3)=ENC(J)
0180          ENCODE(2,405,PACK(15,K3)) IDC
0181          GOTO 450
0182    420   ENCODE(8,402,PACK(7,K3)) BWID
0183          IBC=IBC+1
0184          IF(IBC .EQ. 4) IBC=1
0186          IS=JICVT(IBC,I14)
0187          CALL IMAGE(I14)
0188          PACK(14,K3)=ENC(4)
0189          ENCODE(2,405,PACK(15,K3)) IBC
0190    450   PACK(17,K3)=ICC
0191          ENCODE(4,451,PACK(18,K3)) VMEAS
0192    451   FORMAT(A4)
0193          ENCODE(2,452,PACK(22,K3)) MON
0194    452   FORMAT(A2)
0195          PACK(24,K3)=25
0196          PACK(25,K3)=ISIM
0197          PACK(26,K3)=CR
0198          PACK(27,K3)=LF
0199          PACK(28,K3)=ETX
0200          CALL ENSTR(PACK(254,K3),28)
0201          CALL ENABLE(0)
0202          CALL ENQUE(PXINQ,XINQ,K3)
0203          CALL ENABLE(1)
0204    999   IOFLG=0
0205          RETURN
0206          END
```

```
0001          SUBROUTINE SDCALP
0002          LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0003          LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0004          LOGICAL*1 SAVE1,SAVE2,IDATA
0005          INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ
0006          INTEGER*2 PINQ,K1,OUTFCT,RESNLM,FLWCNT,OUTQ
0007          COMMON/MESS/MESSEQ
0008          COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
             &            ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0009          COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
             &             IOFLG,LLFLG,IRSEND
0010          COMMON/GLOB/ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             &            RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011          COMMON PACK(256,64)
0012          COMMON/SCOM/SAVE1(64),SAVE2(64),ITERM,ICOUNT,I1,I2,IDATA(256),
             &            ISWTCH
0013          IF(IDATA(1) .EQ. "117 .AND. IDATA(2) .EQ. "116) GOTO 10
0015          IF(IDATA(1) .EQ. "117 .AND. IDATA(2) .EQ. "106 .AND.
             &  IDATA(3) .EQ. "106) GOTO 20
0017          IF(IDATA(1) .EQ. "115) GOTO 30
0019          IF(IDATA(1) .EQ. '$')  GOTO 999  ! FROM FIAC
0021    10    ISWTCH=1
0022          ITERM=IDATA(4)
0023          RETURN
0024    20    ISWTCH=0
0025          RETURN
0026    30    DECODE(4,31,IDATA(2),ERR=999) ITYPE
0027    31    FORMAT(I)
0028          IF(ITYPE .EQ. 1 .OR. ITYPE .EQ. 2) GOTO 32
0030          GOTO 999
0031    32    CALL ENABLE(0)
0032          K1=IGETSP(N)
0033          CALL ENABLE(1)
0034          IF(MESSEQ .EQ. 126) MESSEQ=0
0036          MESSEQ=MESSEQ+1
0037          PACK(1,K1)=0
0038          PACK(2,K1)=MESSEQ
0039          PACK(3,K1)=0
0040          PACK(4,K1)=0
0041          PACK(5,K1)=25
0042          PACK(6,K1)=28
0043          IF(ITYPE .EQ. 2) GOTO 34
0045          DO 35 J=1,64
0046           PACK(J+6,K1)=SAVE1(J)
0047    35    CONTINUE
0048          GOTO 40
0049    34    DO 36 J=1,64
0050          PACK(J+6,K1)=SAVE2(J)
0051    36    CONTINUE
0052    40    PACK(71,K1)=CR
0053          PACK(72,K1)=LF
0054          PACK(73,K1)=ETX
0055          CALL ENSTR(PACK(254,K1),73)
0056          CALL ENABLE(0)
```

```
0057          CALL ENQUE(PXINQ,XINQ,K1)
0058          CALL ENABLE(1)
0059    999   RETURN
0060          END
```

```
0001          SUBROUTINE IMAGE(IN)
0002          INTEGER*4 IN
0003          BYTE ENC
0004          COMMON/END/ENC(4)
0005          ENCODE(4,1,ENC) IN
0006    1     FORMAT(4I)
0007          DO 2 J=1,4
0008          IF(ENC(J) .LE. 57 .AND. ENC(J) .GE. 48) GOTO 2
0010          ENC(J)='0'
0011    2     CONTINUE
0012          RETURN
0013          END
```

```
                                        .TITLE  FDM.MACRO
                                        .SBTTL  NODE28
                                        .IDENT  /V3.0/
                                        .GLOBL  LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
                                        .GLOBL  WTOKEN,RAM,STATUS,STATB0
                                        .GLOBL  LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
                                        .NLIST  CND
000000                                  .PSECT

        000000                          R0      =%0
        000001                          R1      =%1
        000002                          R2      =%2
        000003                          R3      =%3
        000004                          R4      =%4
        000005                          R5      =%5
        000006                          SP      =%6
        000007                          PC      =%7

                                        ;INTERFACE ADDRESS

        172410                          BAR=    172410
        172412                          WCR=    172412
        172414                          CSR=    172414
        172416                          IOBUF=  172416
        172416                          OPREG=  172416
        177560                          HRCSR=  177560          ;CHANGED
        177562                          HRBUF=  177562
        177564                          HXCSR=  177564
        177566                          HXBUF=  177566

                                        ;INTERFACE OPTIONS

        000001                          XCIO=1
        000000                          XSIO=0
        000001                          XGIO=1
        000000                          NDXX=0
        000001                          ND24=1
        000000                          NHOST=0
        000001                          DHOST=1

                                        ;LOCAL VARIBLES

        000002                          ARG1=   2
        000004                          ARG2=   4
        000006                          ARG3=   6
        000010                          ARG4=   10

        100000                          BIT15=  100000
        040000                          BIT14=  40000
        000400                          BIT08=  400
        000200                          BIT07=  200
        000100                          BIT06=  100
        000040                          BIT05=  40
        000020                          BIT04=  20
        000010                          BIT03=  10
        000004                          BIT02=  4
```

```
FDM.MACRO       MACRO V03.02B 01:21:00 PAGE 1-1
NODE28

     66         000002                          BIT01=  2
     67         000001                          BIT00=  1
     68
```

```
                                ;*************************************************************
                                ;**** PROCEDURE MASTER START UP                              *
                                ;*************************************************************

000000  012700  000340          MASTER: MOV     #340,   R0                      ;PRI=7
000004  106400                          MTPS    R0
000006  012700  000124                  MOV     #124,   R0
000012  012720  000436'                 MOV     #LIO,   (R0)+                   ;LIU HANDLER
000016  012710  000340                  MOV     #340,   (R0)                    ;PRI=7
000022  000005                          RESET                                   ;RESET BUS
000024  012700  000060                  MOV     #60,    R0
000030  012720  002476'                 MOV     #SIO,   (R0)+                   ;SIG INTERFACE HANDLER
000034  012720  000340                  MOV     #340,   (R0)+
000040  052737  000100  177560          BIS     #100,   @#HRCSR                 ;ENABLE INTERRUPTS
000046  012700  000100                  MOV     #100,   R0                      ;CLOCK VECTOR
000052  012710  001662'                 MOV     #TIME,  (R0)                    ;CLOCK
000056  042737  040100  172414          BIC     #40100, @#CSR                   ;DISABLE INTERRUPTS
000064  004767  000014                  JSR     PC,     LIUINT                  ;CLEAR LIU
000070  005037  172414                  CLR     @#CSR                           ;CLEAR BLIUI CSR
000074  012700  000000                  MOV     #000,   R0                      ;PRI=0
000100  106400                          MTPS    R0                              ;CPU DOWN
000102  000207                          RTS     PC

                                ;*************************************************************
                                ;**** PROCEDURE INITIALIZE                                  *
                                ;*************************************************************

                                        ;CLEAR ACRAM

000104  012737  010402  172416  LIUINT: MOV     #4354., @#OPREG                 ;LDADR
000112  012737  004400  172416          MOV     #2304., @#OPREG                 ;ADDRESS=0
000120  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000124  100375                          BPL     .-4                             ;NO RETRY
000126  105037  172414                  CLRB    @#CSR
000132  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
000140  012700  177400                  MOV     #-256., R0                      ;COUNTER
000144  012737  004407  172416  1$:     MOV     #2311., @#OPREG                 ;WRITE A NULL
000152  105737  172414                  TSTB    @#CSR                           ;GOOD WD
000156  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
000160  105037  172414                  CLRB    @#CSR
000164  005200                          INC     R0                              ;COUNT+1
000166  001366                          BNE     1$

                                        ;CLEAR INPUT/OUTPUT BUFFERS

000170  012701  010410                  MOV     #10410, R1                      ;RDBUFADR CMD
000174  012700  010440                  MOV     #10440, R0                      ;SEL INBUFO
000200  004767  000106                  JSR     PC,     ZEROBP                  ;SET POINTER=0
000204  012737  001400  172416          MOV     #1400,  @#OPREG                 ;FALSE READ DATA
000212  105737  172414                  TSTB    @#CSR                           ;GOOD READ
000216  100375                          BPL     .-4                             ;NO RETRY
000220  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
000224  012701  010610                  MOV     #10610, R1                      ;RDBUFADR COMMAND
000230  012700  010640                  MOV     #10640, R0                      ;SEL INBUF1 COMMAND
```

```
    102 000234  004767  000052                  JSR     PC,     ZEROBP          ;POINTER=0
    103 000240  012737  001400  172416          MOV     #1400,  @#OPREG         ;FALSE READ DATA
    104 000246  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    105 000252  100375                          BPL     .-4                     ;NO RETRY
    106 000254  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    107 000260  012701  010510                  MOV     #10510, R1              ;RDBUFADR COMMAND
    108 000264  012700  010540                  MOV     #10540, R0              ;SEL OUTBUF0 COMMAND
    109 000270  004767  000016                  JSR     PC,     ZEROBP
    110 000274  012701  010710                  MOV     #10710, R1              ;RDBUFADR COMMAND
    111 000300  012700  010740                  MOV     #10740, R0              ;SEL OUTBUF0 COMMAND
    112 000304  004767  000002                  JSR     PC,     ZEROBP          ;ZERO BUFFER POINTER
    113 000310  000440                          BR      STAT$                   ;GO CLEAR STATUS
    114 000312  010137  172416          ZEROBP: MOV     R1,     @#OPREG         ;RDBUFADR
    115 000316  012737  001400  172416          MOV     #1400,  @#OPREG         ;RD
    116 000324  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    117 000330  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    118 000332  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    119 000336  013702  172416                  MOV     @#IOBUF, R2             ;FETCH POINTER
    120 000342  042702  177400                  BIC     #177400, R2             ;CLEAR MST BYTE
    121 000346  010037  172416                  MOV     R0,     @#OPREG         ;SEL BUFFER
    122 000352  022702  000000          2$:     CMP     #0,     R2              ;POINTER=0?
    123 000356  001412                          BEQ     3$                      ;YES RECHECK
    124 000360  012737  001400  172416          MOV     #1400,  @#OPREG         ;NO FALSE RD
    125 000366  105737  172414                  TSTB    @#CSR                   ;GOOD RD
    126 000372  100375                          BPL     .-4                     ;NO LOOP UNTIL READY
    127 000374  105037  172414                  CLRB    @#CSR                   ;CLEAR DONE BIT
    128 000400  005302                          DEC     R2                      ;LOCAL POINTER-1
    129 000402  000763                          BR      2$                      ;BR  NTIL 0
    130 000404  010037  172416          3$:     MOV     R0,     @#OPREG         ;SEL BUFFER
    131 000410  000207                          RTS     PC                      ;RETURN
    132
    133                                         ;READ AND CLEAR STATUS
    134
    135 000412  012737  010400  172416  STAT$:  MOV     #4352., @#OPREG         ;WCR:RS
    136 000420  012737  002400  172416          MOV     #1280., @#OPREG         ;RS(FALSE)
    137 000426  012737  002400  172416          MOV     #1280., @#OPREG         ;RS
    138 000434  000207                          RTS     PC
    139
    140
```

```
                                ;*************************************************************
                                ;**** PROCEDURE LIU-HANDLER                                  *
                                ;*************************************************************

000436  010046          LIO:    MOV     R0,     -(SP)           ;SAVE R0
000440  010146                  MOV     R1,     -(SP)           ;SAVE R1
000442  010246                  MOV     R2,     -(SP)           ;SAVE R2
000444  010346                  MOV     R3,     -(SP)           ;SAVE R3
000446  010446                  MOV     R4,     -(SP)           ;SAVE R4
000450  010546                  MOV     R5,     -(SP)           ;SAVE R5
000452  005737  172414          TST     @#CSR                   ;? LIU INTERRUPTED
000456  100153                  BPL     RTI$                    ;NO
000460  042737  040000  172414  BIC     #BIT14, @#CSR           ;YES/DISABLE LIU
000466  012737  010400  172416  MOV     #4352., @#OPREG         ;WCR : RS(0)
000474  012737  002400  172416  MOV     #1280., @#OPREG         ;RD
000502  013767  172416  000502  MOV     @#IOBUF, CAUSE
000510  042767  177400  000474  BIC     #177400, CAUSE          ;CLEAR UNUSED BITS

                                ;PREFORM OPERATION DEPENDING ON BITS

000516  132767  000004  000466  B2$:    BITB    #BIT02, CAUSE   ;OV-FL
000524  001406                  BEQ     B3$
000526  012702  010410          MOV     #4360., R2              ;RDBUFADR IN0
000532  012703  010440          MOV     #4384., R3              ;SEL INBUF0
000536  004767  000270          JSR     PC,     EMBF            ;GO EMPTY
000542  132767  000010  000442  B3$:    BITB    #BIT03, CAUSE   ;OV-FL
000550  001406                  BEQ     B4$
000552  012702  010610          MOV     #4488., R2              ;RDBUFADR IN1
000556  012703  010640          MOV     #4512., R3              ;SEL INBUF1
000562  004767  000244          JSR     PC,     EMBF            ;GO EMPTY
000566  132767  000020  000416  B4$:    BITB    #BIT04, CAUSE   ;LINE-LOSS PRIMARY
000574  001407                  BEQ     B5$
000576  132767  000200  000406  B7$:    BITB    #BIT07, CAUSE   ;LATCH ON
000604  001403                  BEQ     B5$                     ;YES LOOP AROUND
000606  012767  000002  002002' MOV     #2,     LLFLG           ;FLAG PRIMARY SW. FAIL
000614  132767  000040  000370  B5$:    BITB    #BIT05, CAUSE   ;BACKUP LINE LOSS
000622  001403                  BEQ     B6$
000624  012767  000001  002002' MOV     #1,     LLFLG           ;FLAG BACKUP SW. FAIL
000632  132767  000100  000352  B6$:    BITB    #BIT06, CAUSE   ;WHITE TOKEN DETECT
000640  001404                  BEQ     B0$
000642  005067  001000'          CLR     IWRTTM                  ;CLEAR TOKEN VAR
000646  005067  001002'          CLR     IWRT

                                ;BUFFER FULL

000652  132767  000001  000332  B0$:    BITB    #BIT00, CAUSE   ;INBUF0 FULL
000660  001423                  BEQ     B1$                     ;NO
000662  012701  000400'          MOV     #INBF,  R1              ;BUFFER ADDRESS
000666  012702  010410          MOV     #4360., R2              ;RDBUFADR COMM
000672  012703  010440          MOV     #4384., R3              ;SEL BUF
000676  012704  000001          MOV     #BIT00, R4              ;CRC BIT
000702  004767  000124          JSR     PC,     EMBF            ;EMPTY BUFFER
000706  012705  001214'          MOV     #AREA,  R5              ;DATA LINK AREA
000712  012767  000001  000274  MOV     #1,     AREA            ;1 VARIBLE
000720  010267  000274          MOV     R2,     DATA            ;CRC OR BYTE COUNT
000724  004767  000000G         JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
```

```
     58 000730  132767  000002  000254  B1$:    BITB    #BIT01, CAUSE           ;INBUF1 FULL
     59 000736  001423                           BEQ     RTI$                    ;NO WERE DONE
     60 000740  012701  000400'                  MOV     #INBF,  R1              ;BUFFER ADDRESS
     61 000744  012702  010610                   MOV     #4488., R2              ;RDBUFADR COMM
     62 000750  012703  010640                   MOV     #4512., R3              ;SEL INBUF1
     63 000754  012704  000002                   MOV     #BIT01, R4              ;CRC BIT
     64 000760  004767  000046                   JSR     PC,     EMBF            ;GO EMPTY BUFFER
     65 000764  012705  001214'                  MOV     #AREA,  R5              ;DATA LINK AREA
     66 000770  012767  000001  000216          MOV     #1,     AREA            ;ONE VARIBLE
     67 000776  010267  000216                   MOV     R2,     DATA            ;CRC OR BYTE COUNT
     68 001002  004767  000000G                  JSR     PC,     LPINPT          ;CALL FORTRAN QUE'ER
     69
     70 001006  012605                  RTI$:    MOV     (SP)+,  R5              ;RESTORE REGISTERS
     71 001010  012604                           MOV     (SP)+,  R4
     72 001012  012603                           MOV     (SP)+,  R3
     73 001014  012602                           MOV     (SP)+,  R2
     74 001016  012601                           MOV     (SP)+,  R1
     75 001020  012600                           MOV     (SP)+,  R0
     76 001022  052737  040000  172414          BIS     #BIT14, @#CSR           ;ENABLE INTERRUPTS
     77 001030  000002                           RTI                             ;RETURN FROM INTERRUPT
     78
     79 001032  012737  010600  172416  EMBF:    MOV     #4480., @#OPREG         ;READ STATUS 1
     80 001040  012737  002400  172416           MOV     #1280., @#OPREG         ;RS
     81 001046  013700  172416                   MOV     @#IOBUF, R0             ;FETCH STATUS
     82 001052  130400                           BITB    R4,R0                   ;GOOD CRC
     83 001054  001002                           BNE     CRCOK
     84 001056  012704  177777                   MOV     #-1,    R4              ;NO FLAG
     85 001062  010237  172416          CRCOK:   MOV     R2,     @#OPREG         ;RDBUFADR
     86 001066  012737  001400  172416           MOV     #768.,  @#OPREG         ;RD
     87 001074  105737  172414                   TSTB    @#CSR                   ;GOOD RD
     88 001100  100375                           BPL     .-4
     89 001102  013702  172416                   MOV     @#IOBUF, R2             ;REPLACE WITH POINTER
     90 001106  042702  177400                   BIC     #177400, R2             ;CLEAR MST BITS
     91 001112  010200                           MOV     R2,     R0              ;SAVE IT
     92 001114  005400                           NEG     R0                      ;2'S COMP
     93 001116  010037  172412                   MOV     R0,     @#WCR           ;BYTE COUNT NOW
     94 001122  010137  172410                   MOV     R1,     @#BAR           ;ADDRESS IN MEMORY
     95 001126  013337  172416                   MOV     R3,     @#OPREG         ;SEL BUFFER
     96 001132  012737  001400  172416           MOV     #768.,  @#OPREG         ;FALSE RD
     97 001140  105737  172414                   TSTB    @#CSR                   ;DONE ON
     98 001144  100375                           BPL     .-4                     ;NO LOOP
     99 001146  012737  021000  172416           MOV     #8704., @#OPREG         ;FIRE DMA
    100 001154  000242                           NOP                             ;DELAY
    101 001156  105737  172414                   TSTB    @#CSR                   ;GOOR DMA
    102 001162  100401                           BMI     DMAOK
    103 001164  000240                           NOP                             ;ERROR IF HERE
    104 001166  012737  004400  172416  DMAOK:   MOV     #2304., @#OPREG         ;FALSE WD
    105 001174  105737  172414                   TSTB    @#CSR                   ;GOOD WD
    106 001200  100375                           BPL     .-4
    107 001202  005704                           TST     R4                      ;WAS CRC OK
    108 001204  100001                           BPL     END$                    ;YES
    109 001206  010402                           MOV     R4,     R2              ;NO FLAG IT
    110 001212  000207                 END$:     RTS     PC                      ;RETURN
    111 001212  000000                 CAUSE:    .WORD   0                       ;STATUS BYTE 0 HOLDER
    112 001214  000000                 AREA:     .WORD   0
    113 001216  001220'                           .WORD   DATA
    114 001220  000000                 DATA:     .WORD   0
```

```
                                        .ENABLE LSB
                                ;*************************************************************
                                ;**** PROCEDURE ENABLE                                      *
                                ;*************************************************************

001222  017501  000002          ENABLE: MOV     @ARG1(R5), R1                   ;WHICH SUBROU.
001226  022701  000001                  CMP     #1,     R1
001232  001404                          BEQ     E1$
001234  042737  040000  172414  E0$:    BIC     #BIT14, @#CSR                   ;DISABLE LIU INTERRUPTS
001242  000207                          RTS     PC
001244  052737  040000  172414  E1$:    BIS     #BIT14, @#CSR                   ;ENABLE LIU
001252  000207                  *       RTS     PC

                                        ;FORTRAN ENCODE/DECODE MACRO

001254  017575  000004  000002  ENSTR:  MOV     @ARG2(R5),@ARG1(R5)             ;ENCODE BYTE COUNT
001262  000207                          RTS     PC
001264  017575  000002  000004  DESTR:  MOV     @ARG1(R5),@ARG2(R5)             ;DECODE BYTE COUNT
001272  000207                          RTS     PC


                                ;*************************************************************
                                ;**** PROCEDURE LINE SWITCH                                 *
                                ;*************************************************************

                                        ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
                                        ;
                                        ;04=SET PRIMARY LINE
                                        ;32=RESET PRIMARY LINR
                                        ;08=SET BACKUP LINE
                                        ;64=RESET BACKUP LINE

001274  017501  000002          SWITCH: MOV     @ARG1(R5),R1                    ;SW SET
001300  012737  010420  172416          MOV     #4368., @#OPREG                 ;WCR : MODSTAT
001306  062701  004400                  ADD     #2304., R1
001312  010137  172416                  MOV     R1,     @#OPREG                 ;RD
001316  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001322  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001324  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001330  000207                          RTS     PC
```

```
                                ;*********************************************************
                                ;**** PROCEDURE STATUS                                   *
                                ;*********************************************************

                                        ;CALL STATUS(X,DATA) -READ STATUS BYTES 0/1
                                        ;
                                        ;X=0 STATUS BYTE 0
                                        ;X=1 STATUS BYTE 1

001332  012501  000002          STATUS: MOV     @ARG1(R5), R1                   ;WHICH BYTE
001336  022701  000001                  CMP     #1,     R1                      ;COMPARE
001342  001415                          BEQ     2$
001344  012737  010400  172416  1$:     MOV     #4352., @#OPREG                 ;WCR : RS(0)
001352  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001360  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001364  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001370  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001374  000207                          RTS     PC
001376  012737  010600  172416  2$:     MOV     #4480., @#OPREG                 ;WCR : RS(1)
001404  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001412  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
001416  042700  177400                  BIC     #177400, R0                     ;CLEAR BITS
001422  010075  000004                  MOV     R0,     @ARG2(R5)               ;RTN DATA
001426  000207                          RTS     PC
001430  012737  010400  172416  STATB0: MOV     #4352., @#OPREG                 ;STATUS BYTE 1
001436  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001444  013701  172416                  MOV     @#IOBUF, R1                     ;FETCH DATA
001450  042701  177400                  BIC     #177400, R1
001454  132701  000010                  BITB    #BIT03, R1                      ;BUFFER FULL
001460  001404                          BEQ     3$
001462  012775  000001  000002          MOV     #1,     @ARG1(R5)               ;LOAD RETURN
001470  000207                          RTS     PC
001472  012775  000000  000002  3$:     MOV     #0,     @ARG1(R5)               ;LOAD RETURN
001500  000207                          RTS     PC
```

```
                                ;*************************************************************
                                ;**** PROCEDURE WRITE TOKEN (WTOKEN)                         *
                                ;*************************************************************

                                        ;CALL WTOKEN -LOADS AND WRITES A TOKEN TO LOOP

001502  012737  010600  172416  WTOKEN: MOV     #4480., @#OPREG                 ;READ STATUS(1)
001510  012737  002400  172416          MOV     #1280., @#OPREG                 ;RS
001516  013701  172416                  MOV     @#IOBUF, R1                     ;FET H DATA
001522  132701  000010                  BITB    #BIT03, R1                      ;BUFFER LOADED
001526  001414                          BEQ     WTKG                            ;NO GENERATE TOKEN
001530  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001536  012737  004423  172416          MOV     #2323., @#OPREG                 ;MARK OB0F/OB1F/WTCMD
001544  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001550  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001552  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001556  000207                          RTS     PC
001560  012737  010740  172416  WTKG:   MOV     #4576., @#OPREG                 ;SELECT BUFFER 1
001566  012737  001660' 172410          MOV     #WTD,   @#BAR                   ;ADDRESS OF WRITE TOKEN
001574  012702  000002                  MOV     #2,     R2                      ;# OF BYTES
001600  005402                          NEG     R2                              ;FOR AT IT
001602  010237  172412                  MOV     R2,     @#WCR                   ;DMA BYTE COUNT
001606  012737  024000  172416          MOV     #10240.,@#OPREG                 ;FIRE DMA
001614  000240                          NOP                                     ;DELAY
001616  105737  172414                  TSTB    @#CSR                           ;GOOD DMA
001622  000240                          NOP                                     ;ERROR IF HERE
001624  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE
001630  012737  010420  172416          MOV     #4368., @#OPREG                 ;MODSTAT
001636  012737  004422  172416          MOV     #2322., @#OPREG                 ;MARK OB1F/WTCMD
001644  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001650  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001652  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001656  000207                          RTS     PC                              ;RETURN
001660     000     377          WTD:    .BYTE 0,377                             ;A WTOKEN


                                ;*************************************************************
                                ;**** PROCEDURE TIMER (TIME)                                 *
                                ;*************************************************************

001662  005267  001006' TIME:   INC     LTIME                           ;TIME+1
001666  022767  077777  001006'          CMP     #077777, LTIME                  ;?TIME FULL
001674  001002                          BNE     4$
001676  005267  001006'                  CLR     LTIME                           ;CLEAR CLOCK
001702  000002                  4$:     RTI
```

```
                                ;*******************************************************
                                ;**** PROCEDURE ACRAM                                  *
                                ;*******************************************************

                                        ;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
                                        ;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
                                        ;
                                        ;0110   =NREAD
                                        ;0102   =DREAD
                                        ;0111   =NULL
                                        ;0000   =WTOKEN


001704  017500  000002          RAM:    MOV     @ARG1(R5), R0                   ; RAM
001710  022700  000000                  CMP     #0,     R0                      ;WHICH OPERATION
001714  001035                          BNE     RDRAM                           ;READ OP
001716  017500  000004          WTRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS
001722  017501  000006                  MOV     @ARG3(R5), R1                   ;WRITE DATA
001726  012737  010402  172416          MOV     #4354., @#OPREG                 ;SEL LDACTR
001734  062700  004400                  ADD     #2304., R0                      ;WD/DATA (ADDR)
001740  010037  172416                  MOV     R0,     @#OPREG                 ;WRITE DATA
001744  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
001750  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
001752  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
001756  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
001764  062701  004400                  ADD     #2304., R1                      ;WD/DATA (CMD)
001770  010137  172416                  MOV     R1,     @#OPREG                 ;WRITE
001774  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
002000  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
002002  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002006  000207                          RTS     PC
002010  017500  000004          RDRAM:  MOV     @ARG2(R5), R0                   ;ADDRESS TO READ
002014  012737  010402  172416          MOV     #4354., @#OPREG                 ;SEL LDACTR
002022  062700  004400                  ADD     #2304., R0                      ;WD/DATA (ADDR)
002026  010037  172416                  MOV     R0,     @#OPREG                 ;WRITE DATA
002032  105737  172414                  TSTB    @#CSR                           ;VALID WRITE
002036  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
002040  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002044  012737  010401  172416          MOV     #4353., @#OPREG                 ;SEL ACRAM
002052  012737  001400  172416          MOV     #768.,  @#OPREG                 ;READ DATA
002060  105737  172414                  TSTB    @#CSR                           ;VALID READ
002064  100375                          BPL     .-4                             ;NO LOOP UNTIL READY
002066  105037  172414                  CLRB    @#CSR                           ;CLEAR DONE BIT
002072  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH DATA
002076  042700  177760                  BIC     #177760, R0                     ;CLEAR BITS
002102  010075  000006                  MOV     R0, @ARG3(R5)                   ;RTN DATA
002106  000207                          RTS     PC
```

```
                                .ENABLE LSB
                               ;*********************************************************
                               ;**** PROCEDURE WRITE LOOP (LPOUT)                       *
                               ;*********************************************************

002110  017502  000002  LPOUT: MOV     @ARG1(R5), R2          ;BYTE COUNT
002114  012700  000340         MOV     #340,   R0             ;PRI=7
002120  106400                 MTPS    R0
002122  012737  000000' 172410 MOV     #OUTBF, @#BAR          ;BUS ADDRESS
002130  005402                 NEG     R2                     ; 2'S COMP COUNT
002132  010237  172412         MOV     R2,     @#WCR          ;COUNT
002136  012737  010540  172416 MOV     #4448., @#OPREG        ;OB0 COMMAND
002144  012737  024000  172416 MOV     #10240., @#OPREG       ;DMA GO
002152  000240                 NOP                            ;INTERFACE TIME
002154  105737  172414         TSTB    @#CSR                  ;DMA OK
002160  000240                 NOP                            ;ERROR IF HERE
002162  105037  172414         CLRB    @#CSR
002166  012737  010740  172416 MOV     #4576., @#OPREG        ;OB1 COMMAND
002174  012737  001660' 172410 MOV     #WTD,   @#BAR          ;ADDRESS OF WRITE TOKEN
002202  012702  000002         MOV     #2,     R2             ;BYTE COUNT
002206  005402                 NEG     R2                     ;FOR AT FOR BLIUI
002210  010237  172412         MOV     R2,     @#WCR          ;DMA BYTE COUNT
002214  012737  024000  172416 MOV     #10240.,@#OPREG        ;FIRE DMA
002222  000240                 NOP                            ;DELAY
002224  105737  172414         TSTB    @#CSR                  ;GOOD DMA
002230  000240                 NOP                            ;ERROR IF NOT
002232  105037  172414         CLRB    @#CSR                  ;CLEAR DONE
002236  012737  010420  172416 MOV     #4368., @#OPREG        ;MODSTAT
002244  012737  004403  172416 MOV     #2307., @#OPREG        ;BUFFERS FULL
002252  105737  172414         TSTB    @#CSR                  ;GOOD WRITE
002256  100375                 BPL     .-4                    ;NO LOOP UNTIL
002260  105037  172414         CLRB    @#CSR                  ;CLEAR DONE BIT
002264  012700  000000         MOV     #000,   R0             ;PRI=0
002270  106400                 MTPS    R0                     ;LOWER CPU
002272  000207                 RTS     PC
```

```
                                ;*********************************************************
                                ;**** PROCEDURE READ BUFFER POINTER                      *
                                ;*********************************************************

002274  017500  000002          RDPNT:  MOV     @ARG1(R5), R0                   ;FETCH COMMAND
002300  022700  000000                  CMP     #0,     R0                      ;INBUF0 ?
002304  001004                          BNE     1$                              ;NO
002306  012700  010410                  MOV     #4360., R0                      ;RDBUFADR IN0
002312  000167  000046                  JMP     4$
002316  022700  000001          1$:     CMP     #1,     R0                      ;INBUF1 ?
002322  001004                          BNE     2$                              ;NO
002324  012700  010610                  MOV     #4488., R0                      ;RDBUFADR IN1
002330  000167  000030                  JMP     4$
002334  022700  000002          2$:     CMP     #2,     R0                      ;OUTBUF0 ?
002340  001004                          BNE     3$                              ;NO
002342  012700  010510                  MOV     #4424., R0                      ;RDBUFADR OUT0
002346  000167  000012                  JMP     4$
002352  022700  000004          3$:     CMP     #4,     R0                      ;OUTBUF1 ?
002356  001020                          BNE     5$                              ;NO RETURN
002360  012700  010710                  MOV     #4552., R0                      ;RDBUFADR OUT1
002364  010037  172416          4$:     MOV     R0,     @#OPREG                 ;RDBUFADR
002370  012737  001400  172416          MOV     #768.,  @#OPREG                 ;RD
002376  105737  172414                  TSTB    @#CSR                           ;READY
002402  100375                          BPL     .-4                             ;NO LOOP UNTIL
002404  013700  172416                  MOV     @#IOBUF, R0                     ;FETCH POINTER
002410  042700  177400                  BIC     #177400, R0                     ;CLEAR MST
002414  010075  000004                  MOV     R0,     @ARG2(R5)               ;RETURN POINTER
002420  000207                  5$:     RTS     PC
```

```
                                ;*****************************************************
                                ;**** PROCEDURE RESTART                             *
                                ;*****************************************************

                                        ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
                                        ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
                                        ;CALL RSTART(2) -LOAD MODE(173000)

002422  017500  000002          RSTART: MOV     @ARG1(R5), R0                   ;MODE
002426  022700  000000          RS0:    CMP     #0,     R0                      ;LOAD ADDRESS
002432  001002                          BNE     RS1
002434  000000                          HALT
002436  000207                          RTS     PC
002440  022700  000001          RS1:    CMP     #1,     R0                      ;RESTART PROGRAM
002444  001005                          BNE     RS2
002446  004767  175432                  JSR     PC,     LIUINT                  ;INT LIU FIRST
002452  012700  000040          G4$:    MOV     #40,    R0                      ;START ADDRESS
002456  000110                          JMP     (R0)
002460  022700  000002          RS2:    CMP     #2,     R0                      ;LOAD MODE
002464  001002                          BNE     RST
002466  000177  000002                  JMP     @ROM
002472  000207                  RST:    RTS     PC
002474  173000                  ROM:    .WORD   173000
```

```
                                        .ENABLE LSB
                                ;*************************************************************
                                ;****  PROCEDURE SIO INTERFACE                                *
                                ;*************************************************************

002476  010046                  SIO::   MOV     R0,     -(SP)                   ;SAVE REGISTERS
002500  010146                          MOV     R1,     -(SP)
002502  010246                          MOV     R2,     -(SP)
002504  042737  000100  177560          BIC     #100,   @#HRCSR
002512  012700  000400'                 MOV     #XINBF, R0                      ;BUFFER ADDRESS
002516  012701  000103                  MOV     #67.,   R1                      ;BYTE COUNT
002522  105737  177560          1$:     TSTB    @#HRCSR                         ;PORT READY
002526  100375                          BPL     .-4                             ;LOOP UNTIL
002530  113702  177562                  MOVB    @#HRBUF, R2                     ;FETCH BYTE
002534  005301                          DEC     R1                              ;COUNT-1
002536  105737  177560          2$:     TSTB    @#HRCSR                         ;PORT READY
002542  100375                          BPL     .-4                             ;LOOP UNTIL
002544  113702  177562                  MOVB    @#HRBUF, R2                     ;FETCH BYTE
002550  042702  177400                  BIC     #177400, R2                     ;CLEAR MST
002554  110220                          MOVB    R2,     (R0)+                   ;STORE IT
002556  005301                          DEC     R1
002560  001366                          BNE     2$                              ;READ 39 BYTES
002562  052737  000100  177560          BIS     #100,   @#HRCSR
002570  012767  000001  002000'         MOV     #1,     IOFLG
002576  012602          3$:     MOV     (SP)+,  R2                              ;RETORE REGISTERS
002600  012601                  MOV     (SP)+,  R1
002602  012600                  MOV     (SP)+,  R0
002604  000002                  RTI
```

```
                        ;*********************************************************
                        ;****        COMMON DATA AREA                             *
                        ;*********************************************************

000000                          .PSECT  DFM,RW,D,GBL,REL,OVR

000000                  OUTBF:  .BLKB   256.
000400                  INBF:   .BLKB   256.
001000                  IWRTTM: .BLKW
001002                  IWRT:   .BLKW
001004                  STAT:   .BLKW
001006                  LTIME:  .BLKW
                                .EVEN

000000                          .PSECT  BUFS,RW,D,GBL,REL,OVR

000000                  XOUTBF: .BLKB   256.
000400                  XINBF:  .BLKB   256.
001000                  ACKSEQ: .BLKB   256.
001400                  LIDFD:  .BLKB   256.
002000                  IOFLG:  .BLKW
002002                  LLFLG:  .BLKW
002004                  IRSEND: .BLKW
                                .EVEN


000001                          .END
```

```
FDM.MACRO       MACRO V03.02B 01:21:00 PAGE 12-1
SYMBOL TABLE

ACKSEQ  001000R     003 B1$     000730R         E0$     001234R         LPOUT   002110RG        SIO     002476RG
AREA    001214R         B2$     000516R         E1$     001244R         LTIME   001006R     002 STAT    001004R     002
ARG1  = 000002          B3$     000542R         G4$     002452R         MASTER  000000RG        STATB0  001430RG
ARG2  = 000004          B4$     000566R         HRBUF = 177562          NDXX  = 000000          STATUS  001332RG
ARG3  = 000006          B5$     000614R         HRCSR = 177560          ND24  = 000001          STAT$   000412R
ARG4  = 000010          B6$     000632R         HXBUF = 177566          NHOST = 000000          SWITCH  001274RG
BAR   = 172410          B7$     000576R         HXCSR = 177564          OPREG = 172416          TIME    001662RG
BIT00 = 000001          CAUSE   001212R         INBF    000400R     002 OUTBF   000000R     002 WCR   = 172412
BIT01 = 000002          CRCOK   001062R         IOBUF = 172416          RAM     001704RG        WTD     001660R
BIT02 = 000004          CSR   = 172414          IOFLG   002000R     003 RDPNT   002274RG        WTKG    001560R
BIT03 = 000010          DATA    001220R         IRSEND  002004R     003 RDRAM   002010R         WTOKEN  001502RG
BIT04 = 000020          DESTR   001264RG        IWRT    001002R     002 ROM     002474R         WTRAM   001716R
BIT05 = 000040          DHOST = 000001          IWRTTM  001000R     002 RST     002472R         XCIO  = 000001
BIT06 = 000100          DMAOK   001166R         LIDFD   001400R     003 RSTART  002422RG        XGIO  = 000001
BIT07 = 000200          EMBF    001032R         LIO     000436RG        RS0     002426R         XINBF   000400R     003
BIT08 = 000400          ENABLE  001222RG        LIUINT  000104RG        RS1     002440R         XOUTBF  000000R     003
BIT14 = 040000          END$    001210R         LLFLG   002002R     003 RS2     002460R         XSIO  = 000000
BIT15 = 100000          ENSTR   001254RG        LPINPT= ****** G        RTI$    001006R         ZEROBP  000312R
B0$     000652R

. ABS.  000000      000
        002606      001
DFM     001010      002
BUFS    002006      003
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  300 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  56 PAGES
DK:FDM,DK:FDM=DK:FDM.M28
```

decwriter Program: COMP20.COM 01-JUL 79 12:00:00 PAGE 001

```
FOR NODAL/NOLINE
FOR NODAL1/NOLINE
FOR SDCA/NOLINE
MAC FDM
```

```
ASSIGN DX0: DK:
R LINK
DX1:NODDJB,DX1:MAP=DX1:NODDAL/E/I/W
DX1:FIRM/C
DX1:NODDAL I/C
DX1:SPCA//
$SIMRT


ASSIGN DX1: DK:
```

1.10 User Language

The User Language provides the human interface to the MSCDM and controls its operation. It is implemented as a dialogue between a processor and a terminal with displays stored as disk files. It is written to handle four ESM-MSCDM terminals. The User Language runs as the application program for node 24 (DBMS) and handles all disk-reads and writes.

1.10.1 Mode 1 CRT-to-CRT

This module allows messages to be sent from the MSCDM terminal to the three ESM terminals. In addition to Mode 1, an ATTACH command is provided so that a terminal can connect to another processor or to another terminal. When a terminal is attached to another terminal it can send messages directly without host processor intervention.

1.10.2 Mode 2 System Inquiry

System inquiry allows the MSCDM user to view displays that define the ESM Multiloop Network Simulation Facility. The default display values which correspond to the parameters generated in the nodal software are stored on the mini-disk. Updates to these parameters are stored on the PDP 11/40 HSTB in Loop 2. Updates are created via Mode 3 (System Update) of the User Language resident on HSTB. Node designators 20-28 are used for loop 5. The LID/FAD Conversion Tables are set up such that the FAD is given for the LID for that location on the display.

1.10.3 Mode 3 - Module Update

This mode of operation is used to modify parameters in the other MSCDM nodes. The user is given a list of parameters that may be changed. The parameter changes are performed in the remote nodes by special control packets that are generated by DBMS.

1.10.4 Mode 4 - File Access

This mode of operation allows records of files to be accessed, modified, added and deleted. The files are the circuit and trunk files.

1.10.5 Mode 5 - Report

This mode of operation is used to generate the system control reports. The operator is led through a menu-selection dialogue. This dialogue first asks whether a file is already open so that the operator can exit from mode 5 and return to complete the report later. Two types of reports can be generated: channel-link and switch. The user is prompted for the various elements of the report and the format of the response. A list of remarks can be added to the report. Simulation of a report being sent to a different System Control hierarchical level via the gateway node is done by entering the node designator of an ESM node. The report can then be sent to an ESM terminal or host processor.

1.10.6 Mode 6 - Status

This mode of operation displays the status of simulated equipment that is being monitored by MSCDM.

1.10.7 Program Descriptions

1.10.7.1 Subroutine USRLNG (FORTRAN)

This subroutine is called by the nodal program when a message is received from the loop. This module checks for abort commands and valid terminal input. The message is then passed to a subroutine of the mode the user requested. It also formats all records read from the disk into packets to be written to the loop.

1.10.7.2 Subroutine WRTDB (FORTRAN)

This subroutine updates records (events) in the status sent by the FIAC node.

1.10.7.3 Subroutine M1000 (FORTRAN)

This subroutine handles all the CRT-to-CRT messages it receives from node 25's LA-36 and reroutes the packet to the designated terminal.

1.10.7.4 Subroutine M2000 (FORTRAN)

This subroutine handles the system inquiries made by the user. It records from the MSG DAT file which contains all the system information and forms it into packets to be sent to the requesting terminal.

1.10.7.5 Subroutine M3000 (FORTRAN)

This subroutine handles the system update for modules VSQC, DSQC, BWBSA, and SDCA. It decodes commands inputed by the user and sends messages to each module concerned.

1.10.7.6 Subroutine M4000 (FORTRAN)

This subroutine handles the file access mode of the User Language. It determines which file (trunk, circuit) is open and which record is currently pointed to.

1.10.7.7 Subroutine M4001 (FORTRAN)

The subroutine is called by subroutine M4000 and performs the editing (addition, deletion, change) of a record in a file (circuit, trunk).

1.10.7.8 Subroutine M5000 (FORTRAN)

This subroutine handles the report mode of the User Language. It fills in a menu of questions and then writes them out to a selected terminal.

1.10.7.9 Subroutine SNDMES (FORTRAN)

This subroutine is called by subroutine M5000 to send the completed report to a terminal in message format.

1.10.7.10 Subroutine M6000 (FORTRAN)

This subroutine handles the status mode of the User Language. It modifies status of a channel or prints the status out to the terminal depending on the response to menu selection.

```
0001          SUBROUTINE USRLNG(LI)
0002          LOGICAL*1 F7,F8,F9,F10,F11,F12,IBUFF,TMP(80)
0003          LOGICAL*1 ICODE,IBUF,MODNUM,I7,I8,DUM
0004          LOGICAL*1 PACK,ICON,ETX,LF,CR,MONTOR,ISLID
0005          INTEGER*2 XINQ,PXINQ,PINQ
0006          INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0007          INTEGER*2 ST,DEVNUM,FTYPE,RECNUM,FREE
0008          COMMON PACK(256,30)
0009          COMMON /MESS/ MESSEQ
0010          COMMON /FRE/ FREE(30),IFR,IFRSZ
0011          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1       RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0012          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0013          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0014          COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             1     KEYREC,IBUF(80)
0015          COMMON /M3000/ IFUNT,ITYPE,INODE,IMEAS
0016          COMMON /M4000/ FTYPE,RECNUM,F7,F8,F9,F10,F11,F12
0017          COMMON /M6000/ IDISP,IEQUIP,ICHAN,INUM,IBUFF(20),IFLAG,ICNT
0018       15 FORMAT(1X,80A1)
      C     ********************************
      C     *   START DIALOGUE             *
      C     ********************************
0019          IF(IFIRST .EQ. 1) GOTO 14   !SET AFTER CALLED ONCE
0021          DO 20 IND=1,4
0022       20 ST(IND,1)=2
0023          IND=4
0024          DO 21 I=1,256
0025       21 ICODE(I)=0
0026          GOTO 51
      C     ********************************
      C     *   CHECK FOR ABORT            *
      C     ********************************
0027       14      IF (ICODE(7) .EQ. '101 .AND. ICODE(8) .EQ. '102 .AND.
             1      ICODE(9) .EQ. '117 .AND. ICODE(10) .EQ. '122 .AND.
             2      ICODE(11) .EQ. '124) GOTO 425  !CHECK FOR ABORT
0029          IF(ICODE(7) .NE. '177) GOTO 16
0031          CALL WRTDB
0032          GOTO 12
0033       16 IND=5
0034          MONTOR=ICODE(6)
0035          IF(ICODE(6) .EQ. 4) IND=1
0037          IF(ICODE(6) .EQ. 8) IND=2
0039          IF(ICODE(6) .EQ. 18) IND=3
0041          IF(ICODE(6) .EQ. 25) IND=4
0043          IF(IND .EQ. 5) RETURN
      C     ********************************
      C     *   CHECK FOR DS               *
      C     ********************************
0045          IF ((ICODE(7) .EQ. '104) .AND. (ICODE(8) .EQ. '123)) GOTO 18
      C     ********************************
      C     *   INTRODUCTION               *
      C     ********************************
0047          IF(ICODE(7) .EQ. 'A') GOTO 600              !TEST FOR ATTACH
```

```
0049          GOTO 630
0050     600  IF(ICODE(8) .EQ. 'T') GOTO 605
0052          GOTO 630
0053     605  IF(ICODE(9) .EQ. 'T') GOTO 610
0055          GOTO 630
0056     610  IF(ICODE(10) .EQ. 'A') GOTO 615
0058          GOTO 630
0059     615  IF(ICODE(11) .EQ. 'C') GOTO 620
0061          GOTO 630
0062     620  IF(ICODE(12) .EQ. 'H') GOTO 625
0064          GOTO 630
0065     625  CONTINUE
0066          ICON(7)=(((ICODE(13)-48) * 10) + (ICODE(14)-48))
0067          ICON(8)=*003
0068          ICON(1)=85
0069          ICON(2)=170
0070          ICON(3)=64
0071          ICON(5)=ICODE(6)
0072          ICON(6)=ICODE(5)
0073          CALL ENSTR(ICON(254),8)
0074          ICFLG=1
0075          GOTO 18
0076     630  IF (ST(IND,1) .EQ. 0001) GOTO 51     !DISPLAY INTRO MESSAGE
0078          IF (ST(IND,1) .EQ. 0002) GOTO 52     !DISPLAY ENTER USERCODE
0080          IF (ST(IND,1) .EQ. 0003) GOTO 53     !DISPLAY ENTER PASSWORD
0082          IF (ST(IND,1) .EQ. 0004) GOTO 54     !SELECT MODE OF OPERATION
0084          IF(ST(IND,1) .GE. 1000) GOTO 1
0086          GOTO 2
0087       1  IF(ST(IND,1) .LE. 1999) GOTO 61
0089       2  IF(ST(IND,1) .GE. 2000) GOTO 3
0091          GOTO 4
0092       3  IF(ST(IND,1) .LE. 2999) GOTO 62
0094       4  IF(ST(IND,1) .GE. 3000) GOTO 5
0096          GOTO 6
0097       5  IF(ST(IND,1) .LE. 3999) GOTO 63
0099       6  IF(ST(IND,1) .GE. 4000) GOTO 7
0101          GOTO 8
0102       7  IF(ST(IND,1) .LE. 4999) GOTO 64
0104       8  IF(ST(IND,1) .GE. 5000) GOTO 9
0106          GOTO 10
0107       9  IF(ST(IND,1) .LE. 5999) GOTO 65
0109      10  IF(ST(IND,1) .GE. 6000) GOTO 11
0111          GOTO 12
0112      11  IF(ST(IND,1) .LE. 6999) GOTO 66
0114      18  NRCNO=22                            !IF DS WAS ENTERED
0115          NOREC=1
0116          ST(IND,1)=0001
0117          GOTO 12
0118      51  NRCNO=1
0119          NOREC=2
0120          IFIRST=1
0121          ST(IND,1)=0002              !DISPLAY INTRO
0122          GOTO 12
      C     ********************************
```

```
      C     *   CHECK USERCODE            *
      C     ********************************
0123    52  IF ((ICODE(7) .EQ. '104 .AND. ICODE(8) .EQ. '101 .AND.
           1     ICODE(9) .EQ. '116) .OR. (ICODE(7) .EQ. '117 .AND.
           2     ICODE(8) .EQ. '127 .AND. ICODE(9) .EQ. '105 .AND.
           3     ICODE(10) .EQ. '116)) GOTO 57  !CHECK FOR USERCODE
0125        IF (ICODE(7) .EQ. '114 .AND. ICODE(8) .EQ. '105 .AND.
           4   ICODE(9) .EQ. '116) GOTO 57
0127        IF (ICODE(7) .EQ. '120 .AND. ICODE(8) .EQ. '105 .AND.
           5   ICODE(9) .EQ. '124 .AND. ICODE(10) .EQ. '105) GOTO 57
0129        IF (ICODE(7) .EQ. '120 .AND. ICODE(8) .EQ. '117 .AND.
           6   ICODE(9) .EQ. '124 .AND. ICODE(10) .EQ. '124 .AND.
           7   ICODE(11) .EQ. '105 .AND. ICODE(12) .EQ. '122) GOTO 57
0131        IF (ICODE(7) .EQ. '122 .AND. ICODE(8) .EQ. '117 .AND.
           8   ICODE(9) .EQ. '131) GOTO 57
0133        GOTO 18
0134    57  NRCNO=3                       !DISPLAY ENTER PASSWORD
0135        NOREC=1
0136        ST(IND,1)=0003
0137        GOTO 12
      C     ********************************
      C     *   CHECK PASSWORD            *
      C     ********************************
0138    53  IF ((ICODE(7) .EQ. '105) .AND.
           1    (ICODE(8) .EQ. '123) .AND.
           2    (ICODE(9) .EQ. '115)) GOTO 58   !CHECK PASSWORD
0140        IF (ICODE(7) .EQ. '114 .AND. ICODE(8) .EQ. '105 .AND.
           3   ICODE(9) .EQ. '107) GOTO 58
0142        IF (ICODE(7) .EQ. '115 .AND. ICODE(8) .EQ. '111 .AND.
           4   ICODE(9) .EQ. '103 .AND. ICODE(10) .EQ. '122 .AND.
           5   ICODE(11) .EQ. '117) GOTO 58
0144        IF (ICODE(7) .EQ. '102 .AND. ICODE(8) .EQ. '117 .AND.
           6   ICODE(9) .EQ. '123 .AND. ICODE(10) .EQ. '123) GOTO 58
0146        IF (ICODE(7) .EQ. '123 .AND. ICODE(8) .EQ. '110 .AND.
           7   ICODE(9) .EQ. '117 .AND. ICODE(10) .EQ. '106 .AND.
           8   ICODE(11) .EQ. '101 .AND. ICODE(12) .EQ. '122) GOTO 58
0148        GOTO 18
0149    58  NRCNO=4                             !DISPLAY MENU SELECTION
0150        NOREC=8                             !AND SELECT ONE
0151        ST(IND,1)=0004
0152        GOTO 12
      C     ********************************
      C     *   CHECK MODE SELECTION      *
      C     ********************************
0153    54  MODNUM=ICODE(7)-48  !WHICH MODE WAS SELECTED
0154    56  IF (MODNUM .EQ. 1) GOTO 101          !DETERMINE WHICH MODE
0156        IF (MODNUM .EQ. 2) GOTO 102          !OF OPERATION WAS
0158        IF (MODNUM .EQ. 3) GOTO 103          !CHOOSEN
0160        IF (MODNUM .EQ. 4) GOTO 104
0162        IF (MODNUM .EQ. 5) GOTO 105
0164        IF (MODNUM .EQ. 6) GOTO 106
0166    17  NRCNO=12             !IF INCORRECT NUMBER IS
0167        NOREC=1              !ENTERED - ERROR - RETRY
0168        GOTO 12
```

```
0169   101  NRCNO=13                    !BEGIN CRT-TO-CRT MODE
0170        NOREC=2
0171        ST(IND,1)=1000
0172        GOTO 12
0173   102  NRCNO=23              !BEGIN SYSTEM INQUIRY MODE
0174        NOREC=5
0175        ST(IND,1)=2000
0176        GOTO 12
0177   103  NRCNO=34              !BEGIN MODULE UPDATE MODE
0178        NOREC=3
0179        ST(IND,1)=3000
0180        GOTO 12
0181   104  NRCNO=49              !BEGIN FILE ACCESS MODE
0182        NOREC=3
0183        ST(IND,1)=4000
0184        GOTO 12
0185   105  NRCNO=80              !BEGIN REPORT MODE
0186        NOREC=3
0187        ST(IND,1)=5000
0188        GOTO 12
0189   106  NRCNO=255             !BEGIN STATUS MODE
0190        NOREC=5
0191        ST(IND,1)=6000
0192        GOTO 12
0193    61  CALL M1000(LI)
0194        GOTO 12
0195    62  CALL M2000
0196        GOTO 12
0197    63  CALL M3000
0198        GOTO 12
0199    64  CALL M4000
0200        READ(3'1,ERR=12,END=12)(TMP(I),I=1,80)
0201        READ(5'1,ERR=12,END=12)(TMP(I),I=1,80)
0202        READ(4'1,ERR=12,END=12)(TMP(I),I=1,80)
0203        READ(6'1,ERR=12,END=12)(TMP(I),I=1,80)
0204        GOTO 12
0205    65  CALL M5000
0206        READ(7'1,ERR=12,END=12)(TMP(I),I=1,80)
0207        GOTO 12
0208    66  CALL M6000
0209        GOTO 12
      C     ******************************
      C     *   WRITE PROCEDURE          *
      C     ******************************
0210    12  IF (NRCNO .EQ. 0) GOTO 450
0212        DO 200 J=7,256
0213   200  ICODE(J)=0
0214        IF (NRCNO .NE. 12) GOTO 210
0216        READ(8'12)(ICODE(I),I=9,87)          !ERROR CONDITION
0217        IF(MESSEQ .EQ. 126) MESSEQ=0
0219        MESSEQ=MESSEQ+1
0220        ICODE(1)=0
0221        ICODE(2)=MESSEQ
0222        ICODE(3)=0
```

```
0223          ICODE(4)=0
0224          ICODE(5)=MONTOR
0225          ICODE(6)=ISLID
0226          ICODE(88)=CR
0227          ICODE(89)=LF
0228          ICODE(90)=ETX
0229          CALL ENSTR(ICODE(254),91)
0230          CALL ENABLE(0)
0231          JSPC=IGETSP(N)
0232          CALL ENABLE(1)
0233          DO 205 I=1,256
0234            PACK(I,JSPC)=ICODE(I)
0235    205   CONTINUE
0236          CALL ENABLE(0)
0237          CALL ENQUE(PXINQ,XINQ,JSPC)
0238          CALL ENABLE(1)
0239          GOTO 450
0240    210   ICUR=6
0241          DO 310 J=1,NOREC              !START READ LOOP
0242            READ(8'NRCNO,ERR=450)(IBUF(I),I=1,80)
0243            DO 220 I=80,1,-1
0244              IEND=I
0245              IF(IBUF(I) .NE. 0) GOTO 230
0247    220     CONTINUE
0248    230     ICOUNT=IEND+ICUR
0249            IF(ICOUNT .LT. 240) GOTO 240
0251            IF(MESSEQ .EQ. 126) MESSEQ=0
0253            MESSEQ=MESSEQ+1
0254            ICODE(1)=0
0255            ICODE(2)=MESSEQ
0256            ICODE(3)=0
0257            ICODE(4)=0
0258            ICODE(5)=MONTOR
0259            ICODE(6)=ISLID
0260            ICODE(ICUR+1)=ETX
0261            CALL ENSTR(ICODE(254),ICUR+1)
0262            CALL ENABLE(0)
0263            JSPC=IGETSP(N)
0264            CALL ENABLE(1)
0265            DO 235 I=1,256
0266              PACK(I,JSPC)=ICODE(I)
0267    235     CONTINUE
0268            CALL ENABLE(0)
0269            CALL ENQUE(PXINQ,XINQ,JSPC)
0270            CALL ENABLE(1)
0271            ICUR=6
0272    240     DO 250 K=1,IEND
0273    250     ICODE(ICUR+K)=IBUF(K)
0274            ICUR=ICUR+IEND
0275            ICODE(ICUR+1)=CR
0276            ICODE(ICUR+2)=LF
0277            ICUR=ICUR+2
0278    310   CONTINUE
0279          ICODE(ICUR+1)=ETX
```

```
0280          CALL ENSTR(ICODE(254),ICUR+1)
0281          IF(MESSEQ .EQ. 126) MESSEQ=0
0283          MESSEQ=MESSEQ+1
0284          ICODE(1)=0
0285          ICODE(2)=MESSEQ
0286          ICODE(3)=0
0287          ICODE(4)=0
0288          ICODE(5)=MONTOR
0289          ICODE(6)=ISLID
0290          IF(ICFLG .EQ. 0) GOTO 320
0292          CALL ENABLE(0)
0293          K1=IGETSP(N)
0294          CALL ENABLE(1)
0295          DO 315 I=1,256
0296            PACK(I,K1)=ICON(I)
0297    315   CONTINUE
0298          ICFLG=0
0299          CALL ENABLE(0)
0300          CALL ENQUE(PXINQ,XINQ,K1)
0301          CALL ENABLE(1)
0302    320   CALL ENABLE(0)
0303          JSPC=IGETSP(N)
0304          CALL ENABLE(1)
0305          DO 330 J=1,256
0306            PACK(J,JSPC)=ICODE(J)
0307    330   CONTINUE
0308          DO 340 J=1,256
0309            ICODE(J)=0
0310    340   CONTINUE
0311          CALL ENABLE(0)
0312          CALL ENQUE(PXINQ,XINQ,JSPC)
0313          CALL ENABLE(1)
0314          GOTO 450
0315    425   IABORT=1
0316    450   RETURN
0317    500   END
```

330

```
FORTRAN IV     V02.1-1     Tue 05-Jun-79 14:02:26                PAGE 001

0001          SUBROUTINE WRTDB
0002          INTEGER*2 ST
0003          LOGICAL*1 ICODE,ICON,LBUF(20)
0004          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0005          IF(ICODE(6) .EQ. 28) GOTO 300
0007          IF(ICODE(8) .NE. 'X') GOTO 10
0009          IF(ICODE(9) .NE. 'X') GOTO 10
0011          IF(ICODE(10) .NE. 'X') GOTO 10
0013          IF(ICODE(11) .EQ. 'X') GOTO 100
0015     10   DECODE(2,30,ICODE(16),ERR=900) ICHAN
0016     30   FORMAT(A2)
0017          DO 40 I=1,20
0018     40   LBUF(I)=0
0019          IF(ICHAN .LT. 1 .OR. ICHAN .GT. 1014)  GOTO 900
0021          READ(2'ICHAN)(LBUF(I),I=1,20)
0022          ENCODE(1,45,LBUF(15)) ICODE(18)
0023     45   FORMAT(I1)
0024          WRITE(2'ICHAN)(LBUF(J),J=1,20)
0025          NRCNO=0
0026          GOTO 999
0027    100   DECODE(2,110,ICODE(11),ERR=900) LNKNUM
0028          DECODE(2,110,ICODE(13),ERR=900) LNKEQP
0029    110   FORMAT(A2)
0030          IF(LNKNUM .EQ. 1) LNUM=0
0032          IF(LNKNUM .EQ. 2) LNUM=3
0034          IF(LNKNUM .EQ. 3) LNUM=6
0036          READ(2'1005+LNUM+LNKEQP,ERR=900,END=900)(LBUF(I),I=1,20)
0037          ENCODE(1,45,LBUF(15)) ICODE(18)
0038          WRITE(2'1005+LNUM+LNKEQP,ERR=900,END=900)(LBUF(I),I=1,20)
0039          NRCNO=0
0040          GOTO 999
0041    300   LIREC=ICODE(11)-48
0042          DO 310 I=1,20
0043            LBUF(I)=0
0044    310   CONTINUE
0045          DO 320 I=1,16
0046           LBUF(I)=ICODE(I+10)
0047    320   CONTINUE
0048          WRITE(2'1003+LIREC,ERR=900,END=900)(LBUF(J),J=1,20)
0049          NRCNO=0
0050          GOTO 999
0051    900   NRCNO=109
0052          NOREC=3
0053          RETURN
0054    999   READ(2'1,ERR=900,END=900)(LBUF(J),J=1,20)
0055          RETURN
0056          STOP
0057          END
```

```
0001          SUBROUTINE M1000(LI)
0002          BYTE ICODE,I7,I8,A7,IBUF,ICON,PACK
0003          LOGICAL*1 ISLID,MONTOR,LF,ETX,CR,DUM
0004          INTEGER ST,DEVNUM,OUTQ,OUTFCT,RESNLM
0005          INTEGER FREE,FLWCNT,XINQ,PXINQ
0006          INTEGER PINQ,KEYREC
0007          COMMON PACK(256,30)
0008          COMMON /MESS/ MESSEQ
0009          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0010          COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             &    KEYREC,IBUF(80)
0011          COMMON /FRE/  FREE(30),IFR,IFRSZ
0012          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             1    RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0014          DATA LEOP,LF/'177,'12/
0015          IF (ST(IND,1) .EQ. 1000) GOTO 14     !VALIDATE ND
0017          IF (ST(IND,1) .EQ. 1003) GOTO 53     !MESSAGE TO SEND
0019          IF (ST(IND,1) .EQ. 1004) GOTO 54     !ASK WHAT TO DO NEXT
      C     ******************************
      C     *   CHECK NODE DESIGNATOR    *
      C     ******************************
0021     14   DECODE (2,10,ICODE(7),ERR=19)A7
0022     10   FORMAT(I)
0023          IF (A7 .EQ. 4) GOTO 20
0025          IF (A7 .EQ. 8) GOTO 21
0027          IF (A7 .EQ. 18) GOTO 22
0029          IF (A7 .EQ. 25) GOTO 23
0031     17   NRCNO=12                        !INVALID ND
0032          NOREC=3
0033          ST(IND,1)=1000
0034          RETURN
0035      5   NRCNO=13                              !LIST CRT ND INFO
0036          NOREC=2
0037          ST(IND,1)=1000
0038          RETURN
0039     20   ST(IND,2)=4                           !PUT ND INTO ST(IND,2)
0040          GOTO 7
0041     21   ST(IND,2)=8
0042          GOTO 7
0043     22   ST(IND,2)=18
0044          GOTO 7
0045     23   ST(IND,2)=25
0046      7   NRCNO=15
0047          NOREC=1                                     !DISPLAY 'TYPE
0048          ST(IND,1)=1003                              !IN MESSAGE'
0049          RETURN
0050     53   DO 70 I=1,80
0051     70   IBUF(I)=0
0052          READ(8'16)(IBUF(I),I=1,24)
0053          ENCODE(2,100,IBUF(26)) ST(IND,2)
0054    100   FORMAT(I2)
0055          WRITE(8'16)(IBUF(I),I=1,30)
0056          NRCNO=16
```

```
0057          NOREC=6
0058          ST(IND,1)=1004
0059          CALL ENABLE(0)
0060          I1=IGETSP(N)
0061          CALL ENABLE(1)
0062          IF(MESSEQ .EQ. 126) MESSEQ=0
0064          MESSEQ=MESSEQ+1
0065          PACK(1,I1)=0
0066          PACK(2,I1)=MESSEQ
0067          PACK(3,I1)=0
0068          PACK(4,I1)=0
0069          PACK(5,I1)=ST(IND,2)
0070          PACK(6,I1)=ISLID
0071          IF(ST(IND,2) .EQ. 25) GOTO 60
0073          PACK(7,I1)='024
0074          DO 61 I=8,23
0075     61   PACK(I,I1)=LF
0076          DO 62 I=1,LI-6
0077          JJ=I
0078     62   PACK(I+23,I1)=ICODE(I+6)
0079          PACK(JJ+25,I1)='024
0080          PACK(JJ+26,I1)='024
0081          PACK(JJ+27,I1)=0
0082          PACK(JJ+28,I1)=ETX
0083          CALL ENSTR(PACK(254,I1),JJ+28)
0084          CALL ENABLE(0)
0085          CALL ENQUE(PXINQ,XINQ,I1)
0086          CALL ENABLE(1)
0087          RETURN
0088     60   DO 52 I=7,LI
0089            PACK(I,I1)=ICODE(I)
0090     52   CONTINUE
0091          PACK(LI+1,I1)=CR
0092          PACK(LI+2,I1)=LF
0093          PACK(LI+3,I1)=ETX
0094          CALL ENSTR(PACK(254,I1),LI+3)
0095          CALL ENABLE(0)
0096          CALL ENQUE(PXINQ,XINQ,I1)
0097          CALL ENABLE(1)
0098          RETURN
          C   ******************************
          C   *   WHAT TO DO NEXT          *
          C   ******************************
0099     54   ANS=ICODE(7)-48
0100     58   IF(ANS .EQ. 1) GOTO 700
0102          IF(ANS .EQ. 2) GOTO 1
0104          IF(ANS .EQ. 3) GOTO 6
0106          IF(ANS .EQ. 4) GOTO 8
0108     19   NRCNO=12
0109          NOREC=1
0110          RETURN
0111    700   NRCNO=15                          !NEW MESSAGE TO SAME CRT
0112          NOREC=1
0113          ST(IND,1)=1003
```

```
0114          RETURN
0115       1  NRCNO=13                          !NEW MESSAGE TO ANOTHER CRT
0116          NOREC=2
0117          ST(IND,1)=1000
0118          RETURN
0119       6  NRCNO=22                          !LOGOUT
0120          NOREC=1
0121          ST(IND,1)=9999
0122          RETURN
0123       8  NRCNO=5                           !NEW MODE OF OPERATION
0124          NOREC=7
0125          ST(IND,1)=0004
0126          RETURN
0127          END
```

```
0001          SUBROUTINE M2000
0002          REAL*8 MOUT(10,10),LINE(10)
0003          BYTE ND,I7,I8,ANS,ICODE,ICON
0004          INTEGER ST
0005          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0006          IF (ST(IND,1) .EQ. 2000) GOTO 39       !FIRST TIME IN SYSTEM
0008          IF (ST(IND,1) .EQ. 2003) GOTO 53       !ASK FOR ND
0010          IF (ST(IND,1) .EQ. 2004) GOTO 54       !ASK WHAT TO DO NEXT
0012          IF (ST(IND,1) .EQ. 2005) GOTO 55       !FIND OUT WHAT TO DO
0014          IF (ST(IND,1) .EQ. 2006) GOTO 25
      C     ********************************
      C     *   WHAT TYPE REPORT           *
      C     ********************************
0016     39   I7=ICODE(7)-48                         !DETERMINE WHAT TYPE INFO
0017          IF(I7 .LT. 1 .OR. I7 .GT. 4) GOTO 71
0019     23   ST(IND,3)=I7
0020     24   IF(ST(IND,3) .EQ. 1) GOTO 72
0022     25   NRCNO=28                          !ENTER ND
0023          NOREC=2
0024          ST(IND,1)=2003
0025          RETURN
0026     72   NRCNO=127                         !NDI INFORMATION
0027          NOREC=15
0028          ST(IND,1)=2004
0029          RETURN
0030     80   NRCNO=127
0031          NOREC=15
0032          ST(IND,1)=2006
0033          RETURN
      C     ********************************
      C     *   CHECK NODE DESIGNATOR      *
      C     ********************************
0034     53   IF((ICODE(7) .EQ. '116) .AND.
             1  (ICODE(8) .EQ. '104) .AND.
             2  (ICODE(9) .EQ. '111)),GOTO 80
0036          DECODE (2,40,ICODE(7),ERR=71)ND
0037     40   FORMAT(I)
0038          IF(ND .EQ. 20) GOTO 71
0040     41   IF(ND .LT. 1) GOTO 71
0042          IF(ND .GT. 28) GOTO 71
0044          GOTO (201,201,201,
             1       202,202,202,202,
             2       203,203,203,203,
             3       204,204,204,204,204,204,204,204,
             4       205,205,205,205,205,205,205,205,205)ND
0045    201   IREC=1
0046          GOTO 22
0047    202   IREC=8
0048          IF(ST(IND,3) .EQ. 3) IREC=11
0050          GOTO 22
0051    203   IREC=15
0052          IF(ST(IND,3) .EQ. 3) IREC=21
0054          GOTO 22
0055    204   IREC=22
```

```
0056          IF(ST(IND,3) .EQ. 3) IREC=31
0058          GOTO 22
0059     205  IREC=29
0060          IF(ST(IND,3) .EQ. 3) IREC=137
0062      22  GOTO (72,51,52,10)ST(IND,3)
0063      51  NRCNO=141+IREC                          !INFO FOR LIDS 1-100
0064          NOREC=7
0065          ST(IND,1)=2004
0066          RETURN
0067      52  NRCNO=176+IREC                          !INFO FOR LIDS 101-255
0068          NOREC=10
0069          ST(IND,1)=2004
0070          RETURN
0071      10  DO 20 I=1,10
0072            DO 15 J=1,10
0073              MOUT(I,J)=0
0074      15    CONTINUE
0075      20  CONTINUE
0076      56  DO 57 J=1,10                            !READ FILE FROM DATA BASE
0077            READ(8'216+J)(MOUT(I,J),I=1,10)!INTO MOUT
0078      57  CONTINUE
0079          DO 30 I=1,10
0080            LINE(I)=0
0081      30  CONTINUE
0082          READ(8'226+ND)(LINE(I),I=1,10)       !MOVE INFO INTO LINE
0083          MOUT(1,2)=LINE(1)
0084          MOUT(5,2)=LINE(2)
0085          MOUT(8,2)=LINE(3)
0086          DO 58 J=3,9
0087            MOUT(8,J)=LINE(J+1)
0088      58  CONTINUE
0089          DO 60 J=1,10
0090            WRITE(8'274+J)(MOUT(I,J),I=1,10)
0091      60  CONTINUE
0092          NRCNO=275
0093          NOREC=10
0094          ST(IND,1)=2004
0095          RETURN
0096      54  NRCNO=30                        !ASK WHAT TO DO NEXT
0097          NOREC=4
0098          ST(IND,1)=2005
0099          RETURN
          C   ********************************
          C   *  WHAT TO DO NEXT             *
          C   ********************************
0100      55  ANS=ICODE(7)-48
0101      59  IF(ANS .EQ. 1) GOTO 2
0103          IF(ANS .EQ. 2) GOTO 6
0105          IF(ANS .EQ. 3) GOTO 8
0107          GOTO 71
0108       2  NRCNO=23                         !NEW SYSTEM INQUIRY
0109          NOREC=5
0110          ST(IND,1)=2000
0111          RETURN
```

```
0112      6  NRCNO=22                        !LOGOUT
0113         NOREC=1
0114         ST(IND,1)=9999
0115         RETURN
0116      8  NRCNO=5                         !NEW MODE OF OPERATION
0117         NOREC=7
0118         ST(IND,1)=0004
0119         RETURN
0120     71  NRCNO=12
0121         NOREC=1
0122         RETURN
0123         END
```

```
0001          SUBROUTINE M3000
0002          LOGICAL*1 ICODE,ICON,IBUF
0003          INTEGER ST,XINQ,PXINQ,PINQ,DEVNUM
0004          COMMON /MESS/ MESSEQ
0005          COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             &    KEYREC,IBUF(80)
0006          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0007          COMMON /M3000/ IFUNT,ITYPE,INODE,IMEAS
0008          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0009          IF(ST(IND,1) .EQ. 3000) GOTO 20      !WHAT FUNCTION
0011          IF(ST(IND,1) .EQ. 3005) GOTO 30      !WHAT TYPE
0013          IF(ST(IND,1) .EQ. 3010) GOTO 60      !WHAT NODE
0015          IF(ST(IND,1) .EQ. 3015) GOTO 90      !WHAT MEASURMENT
0017          IF(ST(IND,1) .EQ. 3020) GOTO 110!WHAT TO DO NEXT
0019      10  NRCNO=12
0020          NOREC=1
0021          RETURN
0022      20  IFUNT=ICODE(7)-48            !DETERMINE WHAT FUNCTION WAS
0023          IF(IFUNT .LT. 1 .OR. IFUNT .GT. 4) GOTO 10
0025          IF(IFUNT .EQ. 1) IF=22
0027          IF(IFUNT .EQ. 2) IF=23
0029          IF(IFUNT .EQ. 3) IF=26
0031          IF(IFUNT .EQ. 4) IF=28
0033          NRCNO=37                      !CHOSEN AND ASK WHAT TYPE OF
0034          NOREC=4                               !UPDATE
0035          ST(IND,1)=3005
0036          RETURN
0037      30  ITYPE=ICODE(7)-48            !DETERMINE WHAT TYPE UPDATE
0038          IF(ITYPE .EQ. 1) GOTO 40     !ENENT ON
0040          IF(ITYPE .EQ. 2) GOTO 70     !EVENT OFF
0042          IF(ITYPE .EQ. 3) GOTO 50     !MEASURMENT
0044          GOTO 10
0045      40  NRCNO=41                     !ENTER NODE DESIGNATOR
0046          NOREC=2
0047          ST(IND,1)=3010
0048          RETURN
0049       50 NRCNO=43                     !ENTER CHANNEL,LINK,OR SWITCH
0050          NOREC=1
0051          ST(IND,1)=3015
0052          RETURN
0053      60  DECODE(2,62,ICODE(7),ERR=10)INODE    !WHAT NODE TO SEND REPORT TO
0054      62  FORMAT(I)
0055          IF(INODE .NE. 8 .AND. INODE .NE. 4 .AND. INODE .NE. 18
             &  .AND. INODE .NE. 25) GOTO 10
0057          ICON(7)='0'                  !GENERATE ON PACKET
0058          ICON(8)='N'
0059          ICON(10)=INODE
0060          CALL ENSTR(ICON(254),10)
0061          GOTO 80
0062      70  ICON(7)='0'                  !GENERATE OFF PACKET
0063          ICON(8)='F'
0064          ICON(9)='F'
0065          CALL ENSTR(ICON(254),9)
0066      80  IF(MESSEQ .EQ. 126) MESSEQ=0
```

```
0068          MESSEQ=MESSEQ+1
0069          ICON(1)=0
0070          ICON(2)=MESSEQ
0071          ICON(3)=0
0072          ICON(4)=0
0073          ICON(5)=IF
0074          ICON(6)=24
0075          ICFLG=1
0076          GOTO 100
0077     90   IF(MESSEQ .EQ. 126) MESSEQ=0
0079          MESSEQ=MESSEQ+1
0080          ICON(1)=0
0081          ICON(2)=MESSEQ
0082          ICON(3)=0
0083          ICON(4)=0
0084          ICON(5)=IF
0085          ICON(6)=24
0086          ICON(7)='M'
0087          DO 92 I=7,10
0088            ICON(I+1)=ICODE(I)
0089     92   CONTINUE
0090          DECODE(4,93,ICODE(7),ERR=10) MEAS
0091     93   FORMAT(I4)
0092          IF(IFUNT .EQ. 1) GOTO 94
0094          IF(IFUNT .EQ. 2) GOTO 95
0096          IF(IFUNT .EQ. 3) GOTO 96
0098     94   IF(MEAS .GE. 1 .AND. MEAS .LE. 500) GOTO 98
0100          GOTO 10
0101     95   IF(MEAS .GE. 501 .AND. MEAS .LE. 1000) GOTO 98
0103          GOTO 10
0104     96   IF(MEAS .EQ. 1 .OR. MEAS .EQ. 2 .OR. MEAS .EQ. 3) GOTO 98
0106          GOTO 10
0107     98   CALL ENSTR(ICON(254),11)
0108          ICFLG=1
0109    100   NRCNO=44                        !WHAT TO DO NEXT
0110          NOREC=5
0111          ST(IND,1)=3020
0112          RETURN
0113    110   INEXT=ICODE(7)-48
0114          IF(INEXT .EQ. 1) GOTO 120       !NEW COLUMN FUNCTION
0116          IF(INEXT .EQ. 2) GOTO 130       !SAME COLUMN FUNCTION
0118          IF(INEXT .EQ. 3) GOTO 140       !NEW MODE OF OPERATION
0120          IF(INEXT .EQ. 4) GOTO 150       !LOGOFF
0122          GOTO 10
0123    120   NRCNO=34
0124          NOREC=3
0125          ST(IND,1)=3000
0126          RETURN
0127    130   NRCNO=37
0128          NOREC=4
0129          ST(IND,1)=3005
0130          RETURN
0131    140   NRCNO=5
0132          NOREC=7
```

```
0133          ST(IND,1)=0004
0134          RETURN
0135    150   NRCNO=22
0136          NOREC=1
0137          ST(IND,1)=9999
0138          RETURN
0139          END
```

```
0001          SUBROUTINE M4000
0002          BYTE ICODE,IBUF,JBUF(80),MTYPE,MOUT,ANS,ICON
0003          BYTE I7,I8,F7,F8,F9,F10,F11,F12
0004          INTEGER ST,RECNUM,FTYPE,DEVNUM
0005          DIMENSION MOUT(80)
0006          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0007          COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             1     KEYREC,IBUF(80)
0008          COMMON /M4000/ FTYPE,RECNUM,F7,F8,F9,F10,F11,F12
0009          DATA IPCNT/'45/
0010       15 FORMAT (1X,80A1)
0011          IF(ST(IND,1) .EQ. 4000) GOTO 72      !FIRST TIME IN MODE
0013          IF(ST(IND,1) .EQ. 4002) GOTO 22 !FIND OUT IF TO MODIFY
0015          IF(ST(IND,1) .EQ. 4005) GOTO 30 !ENTER KEY TO MODIFY
0017          IF(ST(IND,1) .EQ. 4006) GOTO 80 !CHECK TO SEE WHAR WAS ENTERED
0019          IF(ST(IND,1) .EQ. 4008) GOTO 61      !KEY WAS NOT FOUND
0021          IF(ST(IND,1) .EQ. 4009) GOTO 65      !DISPLAY SAME RECORD
0023          CALL M4001
0024          RETURN
      C       ********************************
      C       *   WHAT FILE WILL BE USED     *
      C       ********************************
0025       72 I7=ICODE(7)-48                       !DETERMINE WHAT FILE WILL
0026          IF(I7 .LT. 1 .OR. I7 .GT. 2) GOTO 74
0028          FTYPE=I7
0029          GOTO 75
0030       74 NRCNO=12                     !INVALID ENTRY
0031          NOREC=1
0032          RETURN
      C       ********************************
      C       *   DISPLAY BYTE SIZE          *
      C       ********************************
0033       75 READ (8'51+FTYPE,ERR=900,END=900)(JBUF(I),I=1,80)
0034          WRITE(8'54,ERR=900,END=900)(JBUF(I),I=1,80)
0035          NRCNO=54                        !ASK IF IT IS TO BE MODIFIED
0036          NOREC=4
0037          ST(IND,1)=4002
0038          RETURN
      C       ********************************
      C       *   WILL IT BE MODIFIED        *
      C       ********************************
0039       22 I7=ICODE(7)-48                          !IF IT IS TO MODIFIED,
0040          MTYPE=I7
0041       25 IF(MTYPE .EQ. 1) GOTO 26
0043          IF(MTYPE .EQ. 2) GOTO 27
0045          GOTO 74
0046       26 NRCNO=58                     !RECORD TO BE MODIFIED
0047          NOREC=1
0048          ST(IND,1)=4010
0049          RETURN
0050       27 NRCNO=59                     !ACCESS KEY
0051          NOREC=1
0052          ST(IND,1)=4005
0053          RETURN
```

```
0054    30  F7=ICODE(7)          !KEY IS ENTERED AND NOW WILL
0055        F8=ICODE(8)                  !COMPARE TO SEE IF EXISTS
0056        F9=ICODE(9)
0057        F10=ICODE(10)
0058        F11=ICODE(11)
0059        F12=ICODE(12)
0060    28  IF(FTYPE .EQ. 1) GOTO 50     !IF CIRCUIT FILE
      C     ********************************
      C     *   COMPARE TRUNK KEY          *
      C     ********************************
0062    31  RECNUM=0
0063        DO 40 J=1,10
0064          READ (4'J,ERR=900,ERR=900)(JBUF(I),I=1,80)
0065          DO 35 K=1,80,8
0066            RECNUM=RECNUM+1
0067            IF(JBUF(K) .NE. IPCNT) GOTO 33  !CHECK TO SEE IF EOF
0069            IF(JBUF(K+1) .NE. IPCNT) GOTO 33
0071            IF(JBUF(K+2) .EQ. IPCNT) GOTO 60
0073    33      IF(F7 .NE. JBUF(K)) GOTO 35     !START COMPARING FOR
0075            IF(F8 .NE. JBUF(K+1)) GOTO 35   !TRUNK KEY
0077            IF(F9 .NE. JBUF(K+2)) GOTO 35
0079            IF(F10 .NE. JBUF(K+3)) GOTO 35
0081            IF(F11 .NE. JBUF(K+4)) GOTO 35
0083            IF(F12 .EQ. JBUF(K+5)) GOTO 42
0085    35    CONTINUE
0086    40  CONTINUE
0087        GOTO 60
0088    42  READ (6'RECNUM,ERR=900,END=900)(MOUT(I),I=1,80)
0089    45  WRITE(8'72,ERR=900,END=900)(MOUT(I),I=1,80)
0090        NRCNO=72                     !ASK WHAT TO DO NEXT
0091        NOREC=7
0092        ST(IND,1)=4006
0093        RETURN
      C     ********************************
      C     *   COMPARE CIRCUIT KEY,       *
      C     ********************************
0094    50  RECNUM=0                     !CHECK CIRCUIT FILE
0095        DO 58 J=1,10
0096          READ (3'J,ERR=900,END=900)(JBUF(I),I=1,80)
0097          DO 56 K=1,80,4
0098            RECNUM=RECNUM+1
0099            IF(JBUF(K) .NE. IPCNT) GOTO 53  !CHECK FOR EOF
0101            IF(JBUF(K+1) .NE. IPCNT) GOTO 53
0103            IF(JBUF(K+2) .EQ. IPCNT) GOTO 60
0105    53      IF(F7 .NE. JBUF(K)) GOTO 56     !START COMPARING FOR
0107            IF(F8 .NE. JBUF(K+1)) GOTO 56   !CIRCUIT FILE
0109            IF(F9 .NE. JBUF(K+2)) GOTO 56
0111            IF(F10 .EQ. JBUF(K+3)) GOTO 59
0113    56    CONTINUE
0114    58  CONTINUE
0115        GOTO 60
0116    59  READ (5'RECNUM,ERR=900,END=900)(MOUT(I),I=1,80)     !IF KEY IF FOUND
0117        WRITE(8'72,ERR=900,END=900)(MOUT(I),I=1,80)
0118        NRCNO=72                     !ASK WHAT TO DO NEXT
```

```
0119          NOREC=7
0120          ST(IND,1)=4006
0121          RETURN
0122       60 NRCNO=79              !KEY WAS NOT FOUND
0123          NOREC=1
0124          ST(IND,1)=4008
0125          RETURN
0126       61 NRCNO=73
0127          NOREC=6
0128          ST(IND,1)=4006
0129          RETURN
0130       65 I7=ICODE(7)-48
0131          IF(I7 .EQ. 1) GOTO 68
0133          IF(I7 .EQ. 2) GOTO 28
0135          GOTO 74
0136       68 ST(IND,1)=4012
0137          CALL M4001
0138          RETURN
          C    ********************************
          C    *   WHAT TO DO NEXT            *
          C    ********************************
0139       80 I7=ICODE(7)-48                    !DETERMINE WHAT TO DO NEXT
0140          IF(I7 .LT. 1 .OR. I7 .GT. 5) GOTO 74
0142          ANS=I7
0143       95 IF(ANS .EQ. 1) GOTO 100           !NEW RECORD OF SAME FILE
0145          IF(ANS .EQ. 2) GOTO 105           !NEW FILE
0147          IF(ANS .EQ. 3) GOTO 110           !ANOTHER MODE OF OPERATION
0149          IF(ANS .EQ. 4) GOTO 115           !LOGOUT
0151          IF(ANS .EQ. 5) GOTO 120           !SAME RECORD
0153          GOTO 74
0154      100 NRCNO=55
0155          NOREC=3
0156          ST(IND,1)=4002
0157          RETURN
0158      105 NRCNO=49
0159          NOREC=3
0160          ST(IND,1)=4000
0161          RETURN
0162      110 NRCNO=5
0163          NOREC=7
0164          ST(IND,1)=0004
0165          RETURN
0166      115 NRCNO=22
0167          NOREC=1
0168          ST(IND,1)=9999
0169          RETURN
0170      120 NRCNO=55
0171          NOREC=3
0172          ST(IND,1)=4009
0173          RETURN
0174      900 NRCNO=109
0175          NOREC=3
0176          ST(IND,1)=4019
0177          RETURN
```

```
0178        END
```

```
0001          SUBROUTINE M4001
0002          BYTE ICODE,ICON,JBUF(80),IBUF,MOUT,ANS
0003          BYTE I7,I8,F7,F8,F9,F10,F11,F12
0004          INTEGER ST,RECNUM,DEVNUM,FTYPE
0005          DIMENSION MOUT(80)
0006          DATA IPCNT/'45/
0007          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0008          COMMON /FILE/J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             1     KEYREC,IBUF(80)
0009          COMMON /M4000/ FTYPE,RECNUM,F7,F8,F9,F10,F11,F12
0010       15 FORMAT(1X,80A1)
0011          IF(ST(IND,1) .EQ. 4010) GOTO 50      !ASKED FOR KEY TO BE MODIFIED
0013          IF(ST(IND,1) .EQ. 4012) GOTO 52 !CHECK FOR SAME RECORD
0015          IF(ST(IND,1) .EQ. 4014) GOTO 120!UPDATE OR DELETE RECORD
0017          IF(ST(IND,1) .EQ. 4016) GOTO 145!UPDATE FILE
0019          IF(ST(IND,1) .EQ. 4018) GOTO 180!ADD NEW RECORD
0021          IF(ST(IND,1) .EQ. 4019) GOTO 195!RESTART
0023          IF(ST(IND,1) .EQ. 4020) GOTO 230!TYPE IN NEW CIRCUIT RECORD
0025          IF(ST(IND,1) .EQ. 4030) GOTO 270!TYPE IN NEW TRUNK RECORD
      C       ********************************
      C       *   FILE WILL BE MODIFIED      *
      C       ********************************
0027       50 F7=ICODE(7)                    !MOVE KEY INTO FIELDS
0028          F8=ICODE(8)
0029          F9=ICODE(9)
0030          F10=ICODE(10)
0031          F11=ICODE(11)
0032          F12=ICODE(12)
0033       52 IF(FTYPE .EQ. 2) GOTO 80      !IF TRUNK FILE
      C       ********************************
      C       *   COMPARE CIRCUIT FILE       *
      C       ********************************
0035          RECNUM=0
0036          DO 70 KEYREC=1,10    !START COMPARING CIRCUIT FILE
0037            READ(3'KEYREC,ERR=900,END=900)(JBUF(I),I=1,80)
0038            DO 68 K=1,80,4
0039              RECNUM=RECNUM+1
0040              IF(JBUF(K) .NE. IPCNT) GOTO 65
0042              IF(JBUF(K+1) .NE. IPCNT) GOTO 65
0044              IF(JBUF(K+2) .EQ. IPCNT) GOTO 115
0046       65     IF(F7 .NE. JBUF(K)) GOTO 68
0048              IF(F8 .NE. JBUF(K+1)) GOTO 68
0050              IF(F9 .NE. JBUF(K+2)) GOTO 68
0052              IF(F10 .EQ. JBUF(K+3)) GOTO 72
0054       68   CONTINUE
0055       70 CONTINUE
0056          GOTO 115
0057       72 READ(5'RECNUM,ERR=900,END=900)(MOUT(I),I=1,80)       !FOUND THE KEY
0058          WRITE(8'60,ERR=900,END=900)(MOUT(I),I=1,80)
0059          NRCNO=60                     !ASK TO MODIFY OR DELETE
0060          NOREC=4
0061          ST(IND,1)=4014
0062          RETURN
      C       ********************************
```

```
      C     *   COMPARE TRUNK KEY        *
      C     ******************************
0063    80  RECNUM=0                      !START COMPARING TRUNK FILE
0064        DO 100 KEYREC=1,10
0065          READ(4'KEYREC,ERR=900,END=900)(JBUF(I),I=1,80)
0066          DO 96 K=1,80,8
0067            RECNUM=RECNUM+1
0068            IF(JBUF(K) .NE. IPCNT) GOTO 88
0070            IF(JBUF(K+1) .NE. IPCNT) GOTO 88
0072            IF(JBUF(K+2) .EQ. IPCNT) GOTO 115
0074    88      IF(F7 .NE. JBUF(K)) GOTO 96
0076            IF(F8 .NE. JBUF(K+1)) GOTO 96
0078            IF(F9 .NE. JBUF(K+2)) GOTO 96
0080            IF(F10 .NE. JBUF(K+3)) GOTO 96
0082            IF(F11 .NE. JBUF(K+4)) GOTO 96
0084            IF(F12 .EQ. JBUF(K+5)) GOTO 105
0086    96    CONTINUE
0087   100  CONTINUE
0088        GOTO 115
0089   105  READ(6'RECNUM,ERR=900,END=900)(MOUT(I),I=1,80)      !FOUND KEY
0090        WRITE(8'60,ERR=900,END=900)(MOUT(I),I=1,80)
0091        NRCNO=60
0092        NOREC=4
0093        ST(IND,1)=4014
0094        RETURN
0095   115  NRCNO=67                      !ASK IF TO ADD A NEW RECORD
0096        NOREC=4
0097        ST(IND,1)=4018
0098        RETURN
      C     ******************************
      C     *   UPDATE OR DELETE         *
      C     ******************************
0099   120  I7=ICODE(7)-48                        !UPDATE OR DELETE
0100        IF(I7 .LT. 1 .OR. I7 .GT. 2) GOTO 290
0102        ANS=I7
0103   130  IF(ANS .EQ. 1) GOTO 140
0105        IF(ANS .EQ. 2) GOTO 160
0107   140  NRCNO=65                      !MAKE ANY CHANGES
0108        NOREC=2
0109        ST(IND,1)=4016
0110        RETURN
      C     ******************************
      C     *   WRITE NEW RECORD         *
      C     ******************************
0111   145  IF(FTYPE .EQ. 2) GOTO 147    !WRITE NEW RECORD
0113        WRITE(5'RECNUM,ERR=900,END=900)(ICODE(I),I=7,80)
0114        GOTO 150
0115   147  WRITE(6'RECNUM,ERR=900,END=900)(ICODE(I),I=7,80)
0116   150  WRITE(8'72,ERR=900,END=900)(ICODE(I),I=7,80)
0117        NRCNO=71
0118        NOREC=8
0119        ST(IND,1)=4006
0120        RETURN
0121   160  IF(FTYPE .EQ. 2) GOTO 170
```

```
0123          JBUF(K)=0
0124          JBUF(K+1)=0
0125          JBUF(K+2)=0
0126          JBUF(K+3)=0
0127          WRITE(3'KEYREC,ERR=900,END=900)(JBUF(I),I=1,80)
0128          DO 162 I=1,80
0129            JBUF(I)=0
0130    162   CONTINUE
0131          GOTO 174
0132    170   JBUF(K)=0
0133          JBUF(K+1)=0
0134          JBUF(K+2)=0
0135          JBUF(K+3)=0
0136          JBUF(K+4)=0
0137          JBUF(K+5)=0
0138          WRITE(4'KEYREC,ERR=900,END=900)(JBUF(I),I=1,80)
0139          DO 172 I=1,80
0140            JBUF(I)=0
0141    172   CONTINUE
0142    174   WRITE(8'72,ERR=900,END=900)(JBUF(I),I=1,80)
0143          NRCNO=71
0144          NOREC=8
0145          ST(IND,1)=4006
0146          RETURN
      C     ******************************
      C     *   ADD A NEW RECORD         *
      C     ******************************
0147    180   I7=ICODE(7)-48                        !DETERMINE IF TO ADD A NEW
0148          IF(I7 .LT.1 .OR. I7 .GT. 2) GOTO 290
0150          ANS=I7
0151    190   IF(ANS .EQ. 1) GOTO 200
0153    195   NRCNO=73                    !DO NOT ADD A NEW RECORD
0154          NOREC=6
0155          ST(IND,1)=4006
0156          RETURN
0157    200   IF(FTYPE .EQ. 2) GOTO 240
      C     ******************************
      C     *   ADD CIRCUIT FILE         *
      C     ******************************
0159          RECNUM=0                    !ADD NEW CIRCUIT KEY
0160          DO 215 J=1,10                       !FIND FIRST EMPTY KEY TO
0161            READ(3'J,ERR=900,END=900)(JBUF(I),I=1,80)
0162            DO 210 K=1,80,4
0163              RECNUM=RECNUM+1
0164              IF(JBUF(K) .NE. 0) GOTO 205
0166              IF(JBUF(K+1) .NE. 0) GOTO 205
0168              IF(JBUF(K+2) .NE. 0) GOTO 205
0170              IF(JBUF(K+3) .EQ. 0) GOTO 220
0172    205       IF(JBUF(K) .NE. IPCNT) GOTO 210
0174              IF(JBUF(K+1) .NE. IPCNT) GOTO 210
0176              IF(JBUF(K+2) .EQ. IPCNT) GOTO 225
0178    210     CONTINUE
0179    215   CONTINUE
0180    220   JBUF(K)=F7
```

```
0181          JBUF(K+1)=F8
0182          JBUF(K+2)=F9
0183          JBUF(K+3)=F10
0184    224   WRITE(3'J,ERR=900,END=900)(JBUF(I),I=1,80)
0185          NRCNO=64                        !TYPE IN NEW CIRCUIT RECORD
0186          NOREC=1
0187          ST(IND,1)=4020
0188          RETURN
0189    225   JBUF(K)=F7
0190          JBUF(K+1)=F8
0191          JBUF(K+2)=F9
0192          JBUF(K+3)=F10
0193          JBUF(K+4)=IPCNT
0194          JBUF(K+5)=IPCNT
0195          JBUF(K+6)=IPCNT
0196          GOTO 224
0197    230   WRITE(5'RECNUM,ERR=900,END=900)(ICODE(I),I=7,80)
0198          GOTO 280
      C     ********************************
      C     *   ADD TRUNK FILE             *
      C     ********************************
0199    240   RECNUM=0                        !ADD NEW TRUNK KEY
0200          DO 255 J=1,10                   !FINK FIRST EMPTY KEY TO
0201            READ(4'J,ERR=900,END=900)(JBUF(I),I=1,80)
0202            DO 250 K=1,80,8
0203              RECNUM=RECNUM+1
0204              IF(JBUF(K) .NE. 0) GOTO 245
0206              IF(JBUF(K+1) .NE. 0) GOTO 245
0208              IF(JBUF(K+2) .NE. 0) GOTO 245
0210              IF(JBUF(K+3) .NE. 0) GOTO 245
0212              IF(JBUF(K+4) .NE. 0) GOTO 245
0214              IF(JBUF(K+5) .EQ. 0) GOTO 260
0216    245       IF(JBUF(K) .NE. IPCNT) GOTO 250
0218              IF(JBUF(K+1) .NE. IPCNT) GOTO 250
0220              IF(JBUF(K+2) .EQ. IPCNT) GOTO 265
0222    250     CONTINUE
0223    255   CONTINUE
0224    260   JBUF(K)=F7
0225          JBUF(K+1)=F8
0226          JBUF(K+2)=F9
0227          JBUF(K+3)=F10
0228          JBUF(K+4)=F11
0229          JBUF(K+5)=F12
0230    264   WRITE(4'J,ERR=900,END=900)(JBUF(I),I=1,80)
0231          NRCNO=64                        !TYPE IN NEW TRUNK RECORD
0232          NOREC=1
0233          ST(IND,1)=4030
0234          RETURN
0235    265   JBUF(K)=F7
0236          JBUF(K+1)=F8
0237          JBUF(K+2)=F9
0238          JBUF(K+3)=F10
0239          JBUF(K+4)=F11
0240          JBUF(K+5)=F12
```

```
0241          JBUF(K+8)=IPCNT
0242          JBUF(K+9)=IPCNT
0243          JBUF(K+10)=IPCNT
0244          GOTO 264
0245    270   WRITE(6'RECNUM,ERR=900,END=900)(ICODE(I),I=7,80)
0246    280   WRITE(8'72,ERR=900,END=900)(ICODE(I),I=7,80)
0247          NRCNO=71
0248          NOREC=8
0249          ST(IND,1)=4006
0250          RETURN
0251    290   NRCNO=12
0252          NOREC=1
0253          RETURN
0254    900   NRCNO=109
0255          NOREC=3
0256          ST(IND,1)=4019
0257          RETURN
0258          END
```

```
0001          SUBROUTINE M5000
0002          BYTE ICODE,I7,I8,RECBUF,IBUF,STORE,LBUF(80),ICON
0003          INTEGER ST,ANS,PARA,DEVNUM,KEYREC
0004          REAL*8 ERECBF,MOUT(10,9)
0005          DIMENSION STORE(80)
0006          DIMENSION ERECBF(10),RECBUF(80)
0007          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0008          COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
             1     KEYREC,IBUF(80)
0009          EQUIVALENCE (ERECBF,RECBUF)
0010          IF(ST(IND,1) .EQ. 5000) GOTO 50
0012          IF(ST(IND,1) .EQ. 5002) GOTO 56
0014          IF(ST(IND,1) .EQ. 5004) GOTO 65
0016          IF(ST(IND,1) .EQ. 5005) GOTO 133
0018          IF(ST(IND,1) .EQ. 5006) GOTO 140
0020          IF(ST(IND,1) .EQ. 5007) GOTO 146
0022          IF(ST(IND,1) .EQ. 5008) GOTO 150
0024          IF(ST(IND,1) .EQ. 5012) GOTO 310
0026          IF(ST(IND,1) .EQ. 5014) GOTO 365
0028          IF(ST(IND,1) .EQ. 5016) GOTO 375
0030          IF(ST(IND,1) .EQ. 5018) GOTO 385
0032          IF(ST(IND,1) .EQ. 5020) GOTO 395
0034          IF(ST(IND,1) .EQ. 5022) GOTO 405
0036          IF(ST(IND,1) .EQ. 5024) GOTO 415
0038          IF(ST(IND,1) .EQ. 5026) GOTO 425
0040          IF(ST(IND,1) .EQ. 5028) GOTO 435
0042          IF(ST(IND,1) .EQ. 5030) GOTO 510
0044          IF(ST(IND,1) .EQ. 5032) GOTO 95
0046          IF(ST(IND,1) .EQ. 5034) GOTO 100
0048          IF(ST(IND,1) .EQ. 5036) GOTO 105
0050          IF(ST(IND,1) .EQ. 5038) GOTO 110
0052          IF(ST(IND,1) .EQ. 5040) GOTO 115
0054          IF(ST(IND,1) .EQ. 5042) GOTO 120
0056          IF(ST(IND,1) .EQ. 5044) GOTO 125
0058          IF(ST(IND,1) .EQ. 5046) GOTO 130
0060          IF(ST(IND,1) .EQ. 5100) GOTO 137
0062          RETURN
0063       20 NRCNO=12                    !INVALID OPTION
0064          NOREC=1
0065          RETURN
0066       50 I7=ICODE(7)-48                      !ASK IF A FILE ALREADY OPEN
0067          IF(I7 .LT. 1 .OR. I7 .GT. 2) GOTO 20
0069          ANS=I7
0070       54 IF(ANS .EQ. 1) GOTO 300
0072          NRCNO=83                    !NO FILE OPEN, SO CREATE A
0073          NOREC=3                             !NEW CHANNEL OR SWITCH
0074          ST(IND,1)=5002                      !REPORT
0075          RETURN
0076       56 I7=ICODE(7)-48                      !DETERMINE WHAT TYPE REPORT
0077          IF(I7 .LT. 1 .OR. I7 .GT. 2) GOTO 20
0079          IUNIT=I7
0080       60 PARA=0
0081          JT=2
0082          DO 62 I=1,80
```

```
0083     62  LBUF(I)=0
0084         DO 64 I=1,4
0085         WRITE(7'I,ERR=900,END=900)(LBUF(J),J=1,80)
0086     64  CONTINUE
0087         NRCNO=86                        !CREATE A CHANNEL REPORT
0088         NOREC=4
0089         ST(IND,1)=5004
0090         RETURN
0091     65  NRCNO=90
0092         NOREC=1
0093         ST(IND,1)=5032
0094         RETURN
0095     95  DO 96 J=1,3                     !MOVE STATION ID TO ARRAY
0096           RECBUF(J)=ICODE(J+6)
0097     96  CONTINUE
0098         NRCNO=91
0099         NOREC=1
0100         ST(IND,1)=5034
0101         RETURN
0102    100  RECBUF(9)=ICODE(7)              !MOVE SEQUENCE NUMBER TO ARRAY
0103         RECBUF(10)=ICODE(8)
0104         NRCNO=92
0105         NOREC=1
0106         ST(IND,1)=5036
0107         RETURN
0108    105  DO 106 J=7,10                        !MOVE DATE-TIME TO ARRAY
0109           RECBUF(J+10)=ICODE(J)
0110    106  CONTINUE
0111         IF(IUNIT .EQ. 2) GOTO 107
0113         NRCNO=93
0114         NOREC=1
0115         GOTO 108
0116    107  NRCNO=123
0117         NOREC=1
0118    108  ST(IND,1)=5038
0119         RETURN
0120    110  DO 111 J=7,11                        !MOVE LINK TO ARRAY
0121           RECBUF(J+18)=ICODE(J)
0122    111  CONTINUE
0123         IF(IUNIT .EQ. 2) GOTO 118
0125         NRCNO=94
0126         NOREC=1
0127         ST(IND,1)=5040
0128         RETURN
0129    115  DO 116 J=7,9                         !MOVE CHANNEL NUMBER TO ARRAY
0130           RECBUF(J+26)=ICODE(J)
0131    116  CONTINUE
0132    118  NRCNO=95
0133         NOREC=1
0134         ST(IND,1)=5042
0135         RETURN
0136    120  DO 121 J=7,10                        !MOVE OUT TIME TO ARRAY
0137           RECBUF(J+34)=ICODE(J)
0138    121  CONTINUE
```

```
0139          NRCNO=96
0140          NOREC=1
0141          ST(IND,1)=5044
0142          RETURN
0143     125  DO 126 J=7,10                          !MOVE IN TIME TO ARRAY
0144            RECBUF(J+42)=ICODE(J)
0145     126  CONTINUE
0146          NRCNO=97
0147          NOREC=1
0148          ST(IND,1)=5046
0149          RETURN
0150     130  DO 131 J=7,9                           !MOVE RFO TO ARRAY
0151            RECBUF(J+50)=ICODE(J)
0152     131  CONTINUE
0153          WRITE(7'1,ERR=900,END=900)(RECBUF(I),I=1,80)
0154          NRCNO=98
0155          NOREC=3
0156          ST(IND,1)=5005
0157          RETURN
0158     133  IF(ICODE(7) .NE. '77) GOTO 134         !CHECK FOR ???
0160          IF(ICODE(8) .NE. '77) GOTO 134
0162          IF(ICODE(9) .EQ. '77) GOTO 136
0164     134  IF(ICODE(7) .NE. 'X') GOTO 135         !CHECK FOR XXXX AT END
0166          IF(ICODE(8) .NE. 'X') GOTO 135         !OF REMARKS SECTION
0168          IF(ICODE(9) .NE. 'X') GOTO 135
0170          IF(ICODE(10) .EQ. 'X') GOTO 136
0172     135  WRITE(7'JT,ERR=900,END=900)(ICODE(I),I=7,80)
0173          JT=JT+1
0174          NRCNO=0
0175          RETURN
0176     136  WRITE (7'JT,ERR=900,END=900)(ICODE(I),I=7,80)
0177          NRCNO=113                              !ASK IF REPORT IS COMPLETE
0178          NOREC=3
0179          JT=2
0180          ST(IND,1)=5100
0181          RETURN
0182     137  IANS=ICODE(7)-48                       !IS REPORT COMPLETE?
0183          IF(IANS .LT. 1 .OR. IANS .GT. 2) GOTO 20
0185          IF(IANS .EQ. 2) GOTO 146
0187     138  NRCNO=116                              !ASK FOR DESTINATION ID
0188          NOREC=2
0189          ST(IND,1)=5006
0190          RETURN
0191      140 DECODE(2,141,ICODE(7),ERR=20)I7
0192      141 FORMAT(I)
0193          IF(I7 .EQ. 4 .OR. I7 .EQ. 8 .OR. I7 .EQ. 18 .OR.
             &    I7 .EQ. 25) GOTO 143
0195          GOTO 20
0196     143  ANS=I7
0197          CALL SNDMES(ANS,IUNIT)
0198     146  NRCNO=118
0199          NOREC=5
0200          ST(IND,1)=5008
0201          RETURN
```

```
0202   150  I7=ICODE(7)-48                        !DETERMINE WHAT TO DO NEXT
0203        IF(I7 .LT. 1 .OR. I7 .GT. 4) GOTO 20
0205        JT=2
0206        ANS=I7
0207   160  IF(ANS .EQ. 1) GOTO 165
0209        IF(ANS .EQ. 2) GOTO 170
0211        IF(ANS .EQ. 3) GOTO 175
0213   165  NRCNO=80                      !NEW REPORT
0214        NOREC=3
0215        ST(IND,1)=5000
0216        RETURN
0217   170  NRCNO=5                               !NEW MODE OF OPERATION
0218        NOREC=7
0219        ST(IND,1)=0004
0220        RETURN
0221   175  NRCNO=22                      !LOGOUT
0222        NOREC=1
0223        ST(IND,1)=9999
0224        RETURN
0225   300  NRCNO=124                     !ASK WHAT TYPE REPORT
0226        NOREC=3
0227        ST(IND,1)=5012
0228        RETURN
0229   310  I7=ICODE(7)-48                        !DETERMINE WHAT TYPE REPORT
0230        IF(I7 .LT. 1 .OR. I7 .GT. 2) GOTO 20
0232        ANS=I7
0233   320  READ(7'1,ERR=900,END=900)(RECBUF(I),I=1,80)
0234        IF(RECBUF(1) .NE. '77) GOTO 325      !CHECK STATION ID
0236        IF(RECBUF(2) .NE. '77) GOTO 325
0238        IF(RECBUF(3) .EQ. '77) GOTO 360
0240   325  IF(RECBUF(9) .NE. '77) GOTO 330      !CHECK SEQUENCE NUMBER
0242        IF(RECBUF(10) .EQ. '77) GOTO 370
0244   330  IF(RECBUF(17) .NE. '77) GOTO 335     !CHECK DATE-TIME
0246        IF(RECBUF(18) .NE. '77) GOTO 335
0248        IF(RECBUF(19) .EQ. '77) GOTO 380
0250   335  IF(ANS .EQ. 2) GOTO 500
0252        IF(RECBUF(25) .NE. '77) GOTO 340!CHECK LINK ID
0254        IF(RECBUF(26) .NE. '77) GOTO 340
0256        IF(RECBUF(27) .EQ. '77) GOTO 390
0258   340  IF(RECBUF(33) .NE. '77) GOTO 345!CHECK CHANNEL NUMBER
0260        IF(RECBUF(34) .NE. '77) GOTO 345
0262        IF(RECBUF(35) .EQ. '77) GOTO 400
0264   345  IF(RECBUF(41) .NE. '77) GOTO 350!CHECK OUTAGE TIME
0266        IF(RECBUF(42) .NE. '77) GOTO 350
0268        IF(RECBUF(43) .EQ. '77) GOTO 410
0270   350  IF(RECBUF(49) .NE. '77) GOTO 355!CHECK IN TIME
0272        IF(RECBUF(50) .EQ. '77) GOTO 420
0274   355  IF(RECBUF(57) .NE. '77) GOTO 356!CHECK RFO CODE
0276        IF(RECBUF(58) .NE. '77) GOTO 356
0278        IF(RECBUF(59) .EQ. '77) GOTO 430
0280   356  WRITE(7'1,ERR=900,END=900)(RECBUF(I),I=1,80)
0281   357  READ(7'2,ERR=900,END=900)(RECBUF(I),I=1,80)
0282        IF(RECBUF(1) .NE. '77) GOTO 136
0284        IF(RECBUF(2) .NE. '77) GOTO 136
```

```
0286          IF(RECBUF(3) .EQ. '77) GOTO 450
0288          GOTO 136
0289    360   NRCNO=90                        !ENTER STATION ID
0290          NOREC=1
0291          ST(IND,1)=5014
0292          RETURN
0293    365   DO 366 J=1,3                            !MOVE STATION ID TO ARRAY
0294            RECBUF(J)=ICODE(J+6)
0295    366   CONTINUE
0296          GOTO 325
0297    370   NRCNO=91                        !ENTER SEQUENCE NUMBER
0298          NOREC=1
0299          ST(IND,1)=5016
0300          RETURN
0301    375   RECBUF(9)=ICODE(7)              !MOVE SEQ NUM TO ARRAY
0302          RECBUF(10)=ICODE(8)
0303          GOTO 330
0304    380   NRCNO=92                        !ENTER DATE-TIME
0305          NOREC=1
0306          ST(IND,1)=5018
0307          RETURN
0308    385   DO 386 J=7,11                           !MOVE DATE-TIME TO ARRAY
0309            RECBUF(J+10)=ICODE(J)
0310    386   CONTINUE
0311          GOTO 335
0312    390   NRCNO=93                        !ENTER LINK ID
0313          NOREC=1
0314          ST(IND,1)=5020
0315          RETURN
0316    395   DO 396 J=7,11                           !MOVE LINK ID TO ARRAY
0317            RECBUF(J+18)=ICODE(J)
0318    396   CONTINUE
0319          GOTO 340
0320    400   NRCNO=94                        !ENTER CHANNEL NUMBER
0321          NOREC=1
0322          ST(IND,1)=5022
0323          RETURN
0324    405   DO 406 J=7,9                            !MOVE CHAN NUM TO ARRAY
0325            RECBUF(J+26)=ICODE(J)
0326    406   CONTINUE
0327          GOTO 345
0328    410   NRCNO=95                        !ENTER OUTAGE TIME
0329          NOREC=1
0330          ST(IND,1)=5024
0331          RETURN
0332    415   DO 416 J=7,10                           !MOVE OUTAGE TIME TO ARRAY
0333            RECBUF(J+34)=ICODE(J)
0334    416   CONTINUE
0335          GOTO 350
0336    420   NRCNO=96                        !ENTER IN TIME
0337          NOREC=1
0338          ST(IND,1)=5026
0339          RETURN
0340    425   DO 426 J=7,10
```

```
0341          RECBUF(J+42)=ICODE(J)
0342    426   CONTINUE
0343          GOTO 355
0344    430   NRCNO=97                        !ENTER RFO CODE
0345          NOREC=1
0346          ST(IND,1)=5028
0347          RETURN
0348    435   DO 436 J=7,10                          !MOVE RFO CODE TO ARRAY
0349          RECBUF(J+50)=ICODE(J)
0350    436   CONTINUE
0351    438   WRITE(7'1,ERR=900,END=900)(RECBUF(I),I=1,80)
0352          GOTO 357
0353    450   NRCNO=98
0354          NOREC=3
0355          ST(IND,1)=5005
0356          JT=2
0357          RETURN
0358    500   IF(RECBUF(25) .NE. '77) GOTO 345!CHECK SWITCH ID
0360          IF(RECBUF(26) .NE. '77) GOTO 345
0362          IF(RECBUF(27) .EQ. '77) GOTO 505
0364          GOTO 345
0365    505   NRCNO=123
0366          NOREC=1
0367          ST(IND,1)=5030
0368          RETURN
0369    510   DO 511 J=7,11
0370          RECBUF(J+18)=ICODE(J)
0371    511   CONTINUE
0372          GOTO 410
0373    900   NRCNO=109
0374          NOREC=3
0375          RETURN
0376          END
```

```
0001          SUBROUTINE SNDMES(LT,LU)
0002          INTEGER*2 XINQ,PXINQ,PINQ
0003          LOGICAL*1 PACK,LBUF(80),RECBUF(80)
0004          COMMON PACK(256,30)
0005          COMMON /MESS/ MESSEQ
0006          COMMON/QUE/XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0007          DO 5 I=1,80
0008            LBUF(I)=0
0009            RECBUF(I)=0
0010       5  CONTINUE
0011          LCUR=1
0012          READ(7'1,ERR=900,END=900)(RECBUF(I),I=1,80)
0013          IF(LU .EQ. 2) GOTO 100
0015      10  DO 70 JJ1=101,108
0016            READ(8'JJ1,ERR=900,END=900)(LBUF(I),I=1,80)
0017            DO 20 I=80,1,-1
0018              ID=I
0019              IF(LBUF(I) .NE. 0) GOTO 30
0021      20    CONTINUE
0022      30    LBUF(ID+1)='040
0023            DO 40 J=LCUR,LCUR+7
0024              LBUF(ID+J+2-LCUR)=RECBUF(J)
0025      40    CONTINUE
0026            LBUF(ID+10)='015
0027            LBUF(ID+11)='012
0028            LBUF(ID+12)='003
0029            CALL ENABLE(0)
0030            K1=IGETSP(N)
0031            CALL ENABLE(1)
0032            DO 45 K=1,256
0033              PACK(K,K1)=0
0034      45    CONTINUE
0035            DO 50 JP=1,ID+12
0036              PACK(JP+6,K1)=LBUF(JP)
0037      50    CONTINUE
0038            IF(MESSEQ .EQ. 126) MESSEQ=0
0040            MESSEQ=MESSEQ+1
0041            PACK(1,K1)=0
0042            PACK(2,K1)=MESSEQ
0043            PACK(3,K1)=0
0044            PACK(4,K1)=0
0045            PACK(5,K1)=LT
0046            PACK(6,K1)=24
0047            CALL ENSTR(PACK(254,K1),ID+18)
0048            CALL ENABLE(0)
0049            CALL ENQUE(PXINQ,XINQ,K1)
0050            CALL ENABLE(1)
0051            LCUR=LCUR+8
0052            DO 60 I=1,80
0053              LBUF(I)=0
0054      60    CONTINUE
0055      70  CONTINUE
0056          GOTO 200
0057     100  DO 190 JJ1=285,291
```

```
0058          READ(8'JJ1,ERR=900,END=900)(LBUF(I),I=1,80)
0059          DO 110 I=80,1,-1
0060            ID=I
0061            IF(LBUF(I) .NE. 0) GOTO 120
0063    110   CONTINUE
0064    120   LBUF(ID+1)='040
0065          IF(JJ1 .EQ. 289) LCUR=LCUR+8
0067          DO 130 J=LCUR,LCUR+7
0068            LBUF(ID+J+2-LCUR)=RECBUF(J)
0069    130   CONTINUE
0070          LBUF(ID+10)='015
0071          LBUF(ID+11)='012
0072          LBUF(ID+12)='003
0073          CALL ENABLE(0)
0074          K1=IGETSP(N)
0075          CALL ENABLE(1)
0076          DO 135 K=1,256
0077            PACK(K,K1)=0
0078    135   CONTINUE
0079          DO 140 JP=1,ID+12
0080            PACK(JP+6,K1)=LBUF(JP)
0081    140   CONTINUE
0082          IF(MESSEQ .EQ. 126) MESSEQ=0
0084          MESSEQ=MESSEQ+1
0085          PACK(1,K1)=0
0086          PACK(2,K1)=MESSEQ
0087          PACK(3,K1)=0
0088          PACK(4,K1)=0
0089          PACK(5,K1)=LT
0090          PACK(6,K1)=24
0091          CALL ENSTR(PACK(254,K1),ID+18)
0092          CALL ENABLE(0)
0093          CALL ENQUE(PXINQ,XINQ,K1)
0094          CALL ENABLE(1)
0095          LCUR=LCUR+8
0096          DO 150 I=1,80
0097            LBUF(I)=0
0098    150   CONTINUE
0099    190 CONTINUE
0100    200 DO 250 IP=2,4
0101          DO 205 I=1,80
0102            LBUF(I)=0
0103    205   CONTINUE
0104          READ(7'IP,ERR=900,END=900)(LBUF(I),I=1,80)
0105          IF(LBUF(1) .EQ. '?' .AND. LBUF(2) .EQ. '?') GOTO 260
0107          IF(LBUF(1) .EQ. 'X' .AND. LBUF(2) .EQ. 'X' .AND.
     &       LBUF(3) .EQ. 'X') GOTO 260
0109          DO 210 IL=80,1,-1
0110            IF(LBUF(IL) .NE. 0) GOTO 220
0112    210   CONTINUE
0113    220   LBUF(IL)='015
0114          LBUF(IL+1)='012
0115          LBUF(IL+2)='003
0116          IF(IL .EQ. 1) GOTO 250
```

```
0118            CALL ENABLE(0)
0119            K1=IGETSP(N)
0120            CALL ENABLE(1)
0121            DO 225 K=1,258
0122              PACK(K,K1)=0
0123    225     CONTINUE
0124            DO 230 IR=1,IL+2
0125              PACK(IR+6,K1)=LBUF(IR)
0126    230     CONTINUE
0127            IF(MESSEQ .EQ. 126) MESSEQ=0
0129            MESSEQ=MESSEQ+1
0130            PACK(1,K1)=0
0131            PACK(2,K1)=MESSEQ
0132            PACK(3,K1)=0
0133            PACK(4,K1)=0
0134            PACK(5,K1)=LT
0135            PACK(6,K1)=24
0136            CALL ENSTR(PACK(254,K1),IL+8)
0137            CALL ENABLE(0)
0138            CALL ENQUE(PXINQ,XINQ,K1)
0139            CALL ENABLE(1)
0140    250   CONTINUE
0141    260   RETURN
0142    900   NRCNO=109
0143          NOREC=3
0144          RETURN
0145          END
```

```
0001          SUBROUTINE M6000
0002          BYTE ICODE,IBUFF,ICON,ISTAT,PACK
0003          BYTE ISLID,MONTOR,LF,ETX,CR,DUM
0004          INTEGER ST,FLWCNT,OUTFCT,OUTQ,RESNLM
0005          INTEGER PXINQ,XINQ,PINQ
0006          COMMON PACK(256,30)
0007          COMMON /MESS/ MESSEQ
0008          COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
             &    RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009          COMMON/QUE/ XINQ(15),PXINQ(2),INQ(15),PINQ(2)
0010          COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,NOREC,ICON(256)
0011          COMMON /M6000/ IDISP,IEQUIP,ICHAN,INUM,IBUFF(20),IFLAG,ICNT
0012          IF(IFLAG .EQ. 1) GOTO 30
0014          ICNT=0
0015          IF(ST(IND,1) .EQ. 6000) GOTO 20 !WHAT COMMAND WAS ENTERED
0017          IF(ST(IND,1) .EQ. 6010) GOTO 300!CHOSE NEXT COMMAND
0019          IF(ST(IND,1) .EQ. 6020) GOTO 140!WHAT EQUIPMENT
0021          IF(ST(IND,1) .EQ. 6030) GOTO 160!PICK DEVICE NUMBER
0023          IF(ST(IND,1) .EQ. 6040) GOTO 200!CHANGE STATUS CODE
0025          IF(ST(IND,1) .EQ. 6050) GOTO 310!PICK A NEW COMMAND
0027    10    NRCNO=12                      !ERROR
0028          NOREC=1
0029          RETURN
0030    20    IDISP=ICODE(7)-48             !WHAT TYPE INFO
0031          IF(IDISP .EQ. 1) GOTO 30      !RED CONDITIONS
0033          IF(IDISP .EQ. 2) GOTO 30      !AMBER CONDITIONS
0035          IF(IDISP .EQ. 3) GOTO 130     !SELECT STATUS
0037          IF(IDISP .EQ. 4) GOTO 130     !CHANGE STATUS
0039          GOTO 10                               !ERROR
0040    30    ISTAT='1'
0041          IF(IDISP .EQ. 2) ISTAT='2'    !AMBER STATUS
0043    35    ICNT=ICNT+1
0044          IF(ICNT .GT. 1016) GOTO 60
0046          READ(2'ICNT)(IBUFF(I),I=1,20)
0047          IF(IBUFF(15) .NE. ISTAT) GOTO 35
0049          CALL ENABLE(0)
0050          K1=IGETSP(N)
0051          CALL ENABLE(1)
0052          IF(MESSEQ .EQ. 126) MESSEQ=0
0054          MESSEQ=MESSEQ+1
0055          PACK(1,K1)=0
0056          PACK(2,K1)=MESSEQ
0057          PACK(3,K1)=0
0058          PACK(4,K1)=0
0059          PACK(5,K1)=MONTOR
0060          PACK(6,K1)=24
0061          PACK(7,K1)='S'
0062          PACK(8,K1)='T'
0063          PACK(9,K1)='A'
0064          PACK(10,K1)='T'
0065          PACK(11,K1)='U'
0066          PACK(12,K1)='S'
0067          PACK(13,K1)='040
0068          PACK(14,K1)='040
```

```
0069          PACK(15,K1)='040
0070          PACK(16,K1)='040
0071          PACK(17,K1)='040
0072          DO 40 I=1,20
0073            PACK(I+17,K1)=IBUFF(I)
0074     40   CONTINUE
0075          PACK(38,K1)='015
0076          PACK(39,K1)='012
0077          PACK(40,K1)='003
0078          CALL ENSTR(PACK(254,K1),40)
0079          CALL ENABLE(0)
0080          CALL ENQUE(PXINQ,XINQ,K1)
0081          CALL ENABLE(1)
0082          IFLAG=1
0083          NRCNO=0
0084          GOTO 999
0085     60   IFLAG=0
0086          ST(IND,1)=6010
0087          NRCNO=0
0088          GOTO 999
0089    130   NRCNO=260                    !PICK TYPE OF EQUIPMENT
0090          NOREC=4
0091          ST(IND,1)=6020
0092          RETURN
0093    140   IEQUIP=ICODE(7)-48           !WHAT WAS PICKED
0094          IF(IEQUIP .LT. 1 .OR. IEQUIP .GT. 6) GOTO 10
0096          IF(IDISP .EQ. 4) GOTO 150
0098          NRCNO=264                    !ENTER DEVICE NUMBER
0099          NOREC=1
0100          ST(IND,1)=6030
0101          RETURN
0102    150   NRCNO=265
0103          NOREC=2
0104          ST(IND,1)=6030
0105          RETURN
0106    160   DECODE(4,162,ICODE(7),ERR=10)INUM    !GET NUMBER ENTERED
0107    162   FORMAT(I4)
0108          IF(IEQUIP .EQ. 1) GOTO 170
0110          IF(IEQUIP .EQ. 3) GOTO 175
0112          IF(INUM .LT. 1 .OR. INUM .GT. 3) GOTO 10
0114          IF(IEQUIP .EQ. 2) INUM=INUM+1000
0116          IF(IEQUIP .EQ. 4) INUM=INUM+1005
0118          IF(IEQUIP .EQ. 5) INUM=INUM+1008
0120          IF(IEQUIP .EQ. 6) INUM=INUM+1011
0122          GOTO 180
0123    170   IF(INUM .LT. 1 .OR. INUM .GT. 1000) GOTO 10
0125          GOTO 180
0126    175   IF(INUM .EQ. 1 .OR. INUM .EQ. 2) GOTO 177
0128          GOTO 10
0129    177   INUM=INUM+1003
0130    180   IF(IDISP .EQ. 4) GOTO 200
0132          READ(2'INUM,ERR=900,END=900)(IBUFF(J),J=1,20)
0133          WRITE(8'273,ERR=900,END=900)(IBUFF(J),J=1,20)
0134          NRCNO=273
```

```
0135          NOREC=2
0136          ST(IND,1)=6010
0137          GOTO 999
0138     200  INO=ICODE(12)-48
0139          IF(INO .LT. 1 .OR. INO .GT. 3) GOTO 10
0141          READ(2'INUM,ERR=900,END=900)(IBUFF(I),I=1,20)
0142          IBUFF(15)=INO+48
0143          WRITE(2'INUM,ERR=900,END=900)(IBUFF(I),I=1,20)
0144          NRCNO=267
0145          NOREC=6
0146          ST(IND,1)=6050
0147          GOTO 999
0148     300  NRCNO=268
0149          NOREC=5
0150          ST(IND,1)=6050
0151          RETURN
0152     310  IANS=ICODE(7)-48
0153          IF(IANS .EQ. 1) GOTO 320
0155          IF(IANS .EQ. 2) GOTO 330
0157          IF(IANS .EQ. 3) GOTO 340
0159          IF(IANS .EQ. 4) GOTO 350
0161          GOTO 10
0162     320  ICNT=0
0163          IF(IDISP .EQ. 1) GOTO 30
0165          IF(IDISP .EQ. 2) GOTO 30
0167          NRCNO=260
0168          NOREC=4
0169          ST(IND,1)=6020
0170          RETURN
0171     330  NRCNO=5
0172          NOREC=7
0173          ST(IND,1)=0004
0174          RETURN
0175     340  NRCNO=255
0176          NOREC=5
0177          ST(IND,1)=6000
0178          RETURN
0179     350  NRCNO=22
0180          NOREC=1
0181          ST(IND,1)=9999
0182          RETURN
0183     900  NRCNO=109
0184          NOREC=3
0185          ST(IND,1)=6010
0186          RETURN
0187     999  READ(2'1,ERR=900,END=900)(IBUFF(I),I=1,20)
0188          RETURN
0189          STOP
0190          END
```

1.10.8 MSG.DAT File

This is a listing of the file MSG.DAT which the User Language uses for the menu formats of all five modes.

```
1     THIS IS THE FDM - (FEASIBILITY DEVELOPMENT MODEL)
2         ENTER USERCODE PLEASE
3         ENTER PASSWORD PLEASE
4         YOU ARE NOW LOGGED IN - (TO LOGOUT, ENTER *DS*)
5     PLEASE SELECT ONE MODE OF OPERATION:
6         1. CRT TO CRT
7         2. SYSTEM INQUIRY
8         3. MODULE UPDATE
9         4. FILE ACCESS
10        5. REPORT
11        6. STATUS
12    *INVALID ENTRY* - PLEASE TRY AGAIN (OR ENTER DS TO LOGOUT)
13    ENTER DEST CRT NODE DESIGNATOR (ND) - 04 FOR LP#2, 08 FOR LP#3
14        18 FOR LP#4, 25 FOR FDM.
15        PLEASE TYPE IN MESSAGE AND PRESS RETURN
16        MSG SENT TO CRT ND=
17    PLEASE SELECT ONE MODE OF OPERATION:
18        1. NEW MESSAGE TO SAME CRT
19        2. NEW MESSAGE TO ANOTHER CRT
20        3. LOGOUT
21        4. NEW MODE OF OPERATION
22    YOU ARE LOGGED OUT FROM FDM
23    PLEASE SELECT TYPE OF SYSTEM INFORMATION:
24        1. NETWORK DEVICE INFORMATION
25        2. LID/FAD CONVERSION TABLE (LID'S 1-100)
26        3. LID/FAD CONVERSION TABLE (LID'S 101-254)
27        4. WORKPAGE PARAMETERS OF NODES
28        PLEASE ENTER NODE DESIGNATOR (ND)
29        IF ND IS NOT KNOWN, ENTER NDI FOR NETWORK DEVICE INFORMATION
30    PLEASE SELECT ONE OF THE FOLLOWING:
31        1. NEW SYSTEM INQUIRY
32        2. LOGOUT
33        3. ANOTHER MODE OF OPERATION
34    PLEASE SELECT COLUMN FUNCTION TO BE UPDATED.
35        1. VSQC     3. BWBSA
36        2. DSQC     4. SDCA
37     PLEASE SELECT TYPE OF UPDATE
38        1. EVENT REPORTING ON
39        2. EVENT REPORTING OFF
40        3. MEASURMENT REQUESTED
41     PLEASE ENTER NODE DESIGNATOR FOR EVENT REPORTS
42        (E.G., 04,08,18,25)
43     PLEASE ENTER CHANNEL,LINK OR SWITCH NUMBER TO BE MEASURED (FORMAT I4)
44     PLEASE SELECT ONE OF THE FOLLOWING:
45        1. NEW COLUMN FUNCTION
46        2. SAME COLUMN FUNCTION
47        3. NEW MODE OF OPERATION
48        4. LOGOUT
49    PLEASE SELECT FILE TO BE ACCESSED:
50        1. CIRCUIT FILE
51        2. TRUNK FILE
52        THE KEY IS 4 BYTES
53        THE KEY IS 6 BYTES
54
55    DO YOU WISH TO MODIFY THIS RECORD
56        1. YES
57        2. NO
```

```
58      PLEASE ENTER KEY OF RECORD TO BE MODIFIED
59      PLEASE ENTER ACCESS KEY
60
61      FOR THIS RECORD, PLEASE SELECT TYPE OF DESIRED CHANGE
62          1. UPDATE
63          2. DELETE
64      TYPE IN NEW RECORD
65      MAKE ANY CHANGED YOU WISH USING CRT
66      WHEN CHANGES ARE COMPLETE, PRESS RETURN
67      THE RECORD DOES NOT EXIST
68      DO YOU WISH TO ADD A NEW RECORD TO THE FILE?
69          1. YES
70          2. NO
71      ** MODIFICATION COMPLETE **
72
73      PLEASE SELECT ONE OF THE FOLLOWING:
74          1. NEW RECORD OF FILE
75          2. NEW FILE
76          3. ANOTHER MODE OF OPERATION
77          4. LOGOUT
78          5. SAME RECORD
79      THE RECORD DOES NOT EXIST.  PRESS RETURN KEY
80      DO YOU HAVE A FILE ALREADY OPEN?
81          1. YES
82          2. NO
83      PLEASE SELECT TYPE OF REPORT TO BE GENERATED:
84          1. CHANNEL-LINK
85          2. SWITCH
86      IF YOU DO NOT HAVE AN ANSWER FOR A QUESTION
87      ENTER THREE(3) '???'
88      IF THE QUESTION IS NONAPPLICABLE ENTER 'N/A'
89      PRESS RETURN KEY WHEN READY
90      ENTER REPORTING STATION INDICATOR (3 CHARACTERS)
91      ENTER REPORT SEQUENCE NUMBER (2 DIGITS)
92      ENTER DATE-TIME THE REPORT IS MADE (DDTTTT)
93      ENTER LINK IDENTIFIER (5 CHARACTERS)
94      ENTER CHANNEL NUMBER (3 DIGITS)
95      ENTER TIME THE OUTAGE  BEGAN (TTTT)
96      ENTER TIME THE OUTAGE TERMINATED (TTTT)
97      ENTER REASON FOR OUTAGE CODE (3 CHARACTERS)
98      ENTER REMARKS -
99         MAXIMUM OF THREE LINES
100        USED 'XXXX' TO INDICATE END OF REMARKS:
101     REPORTING STATION INDICATOR
102     REPORT SEQUENCE NUMBER
103     DATE-TIME REPORT IS MADE
104     LINK IDENTIFIER
105     CHANNEL NUMBER
106     TIME THE OUTAGE BEGAN
107     TIME THE OUTAGE TERMINATED
108     REASON FOR OUTAGE CODE
109
110      **  ERROR IN READ/WRITE TO DATA BASE - RECORD NOT UPDATED **
111
112
113     IS REPORT COMPLETE
114         1. YES
```

```
        2. NO
 ENTER DESTINATION NODE DESIGNATOR
(04 FOR LP #1, OR FOR LP #2, 18 FOR LP #4, 25 FOR FDM)
PLEASE SELECT ONE OF THE FOLLOWING:
     1. NEW REPORT
     2. NEW MODE OF OPERATION
     3. LOGOUT

ENTER SWITCH IDENTIFIER (5 CHARACTERS)
WHAT TYPE REPORT?
     1. CHANNEL-LINK
     2. SWITCH
                NETWORK DEVICE INFORMATION

    NT   ND RDA  LP      NT   ND RDA  LP      NT   ND RDA  LP
  HSTA    1   1    1  GAT3-1   11   1    3    SSCI   21   1    5
 GAT1-2   2   2    1  GAT4-3   12   3    4    VSGC   22   3    5
 GAT1-3   3   3    1  GAT4-5   13   5    4    DSGC   23   6    5
   CRT4   4   4    2  AUTODN   14   1    4    HSTD   24   5    5
   HSTB   5   2    2    TCCF   15   2    4   CRT25   25   7    5
 GAT2-1   6   1    2    HSTC   16   4    4   RWRSA   26   8    5
 GAT2-3   7   3    2    SDLC   17   6    4    FIAC   27   9    5
   CRT8   8   3    3   CRT18   18   7    4    SDCA   28   2    5
 GAT3-4   9   4    3   SECUR   19   8    4
 GAT3-2  10   2    3
       NOTE: ND=NODE DESIGNATOR, RDA=READ ADDRESS, LP=LOOP
       --PRESS RETURN FOR NEXT INSTRUCTION--
      LOOP -1- LID/FAD CONVERSION TABLE, FOR LIDS 1-100
1   2   3   2   2   2   2   3   3   3   3   3   3   3   3   3   3   3   3   3
3   3   3   3   3   3   3   3
0
0
0
      -PRESS RETURN FOR NEXT INSTRUCTION-
      LOOP -2- LID/FAD CONVERSION TABLE, FOR LIDS 1-100
1   1   1   4   2   1   3   3   3   3   3   3   3   3   3   3   3   3   3   3
3   3   3   3   3   3   3   3
0
0
0
      -PRESS RETURN FOR NEXT INSTRUCTION-
      LOOP -3- LID/FAD CONVERSION TABLE, FOR LIDS 1-100
1   1   1   2   2   2   2   3   4   2   1   4   4   4   4   4   4   4   4   4
4   4   4   4   4   4   4   4
0
0
0
      -PRESS RETURN FOR NEXT INSTRUCTION-
      LOOP -4- LID/FAD CONVERSION TABLE, FOR LIDS 1-100
3   3   3   3   3   3   3   3   3   3   3   3   5   1   2   4   6   7   8   5
5   5   5   5   5   5   5   5
0
0
0
      -PRESS RETURN FOR NEXT INSTRUCTION-
      LOOP -5- LID/FAD CONVERSION TABLE, FOR LIDS 1-100
4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4
```

```
172     1  '3   6   5   7   8   9   2
173     0
174     0
175     0
176           -PRESS RETURN FOR NEXT INSTRUCTION-
177           LOOP -1- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
178     0
179     0
180     0
181     0
182     0
183     0
184     0
185     0
186           -PRESS RETURN FOR NEXT INSTRUCTION-
187           LOOP -2- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
188     0
189     0
190     0
191     0
192     0
193     0
194     0
195     0
196           -PRESS RETURN FOR NEXT INSTRUCTION-
197           LOOP -3- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
198     0
199     0
200     0
201     0
202     0
203     0
204     0
205     0
206           -PRESS RETURN FOR NEXT INSTRUCTION-
207           LOOP -4- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
208     0
209     0
210     0
211     0
212     0
213     0
214     0
215     0
216           -PRESS RETURN FOR NEXT INSTRUCTION-
217                       NODE WORKPAGE PARAMETERS
218              NODE HAS DESIGNATOR                    IN  LOOP#
219        ALTERNATE GATEWAY FUNCTIONAL ADDRESS  .............
220        MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) ............
221        MAXIMUM OUTPUT QUEUE SIZE (TO BITSTREAM) ..........
222        MAXIMUN PACKET XMISSIONS BEFORE MSG TERM ..........
223        TIMEOUT FOR WRITE TOKEN REGENERATION ..............
224        TIMEOUT FOR PACKET RETRANSMISSION .................
225        NUMBER OF NODES IN LOCAL LOOP .....................
226       PRESS RETURN FOR NEXT INSTRUCTION
227         HSTA    1        1     2,3      10        1        6      41       41       3
228       GATI-2    2        1     2,3      10        1       NA       7       NA       3
```

```
229      GAT1-3    3      1      2,3     10      1      NA      12      NA      3
230        CRT4    4      2      1,3      8      1       8      12      41      4
231        HSTB    5      2      1,3     10      1       6       4      42      4
232      GAT2-1    6      2      1,3     10      1      NA       7      NA      4
233      GAT2-3    7      2      1,3     10      1      NA       6      NA      4
234        CRT8    8      3      1,2      8      1       8      12      41      4
235      GAT3-4    9      3      1,2     10      1      NA      14      NA      4
236      GAT3-2   10      3      1,2     10      1      NA       4      NA      4
237      GAT3-1   11      3      1,2     10      1      NA       7      NA      4
238      GAT4-3   12      4      3,5     10      1      NA       7      NA      8
239      GAT4-5   13      4      3,5     10      1      NA       7      NA      8
240      AUTODN   14      4      3,5     10      1      NA       7      NA      8
241        TCCF   15      4      3,5     10      1      NA       7      NA      8
242        HSTA   16      4      3,5     10      1      NA       7      NA      8
243        SDLC   17      4      3,5     10      1      NA       7      NA      8
244       CRT18   18      4      3,5     10      1      NA       7      NA      8
245        SECUR   19      4      3,5     10      1      NA       7      NA      8
246          SIG   20     NA       NA     NA     NA      NA      NA      NA     NA
247        SSCI    21      5       NA     64      2       4      15      25      8
248        VSQC    22      5       NA     64      2       4      15      25      8
249        DSQC    23      5       NA     64      2       4      15      25      8
250        HSTD    24      5       NA     64      2       4      15      25      8
251       CRT25    25      5       NA     64      2       4      15      25      8
252       BWBSA    26      5       NA     64      2       4      15      25      8
253        FIAC    27      5       NA     64      2       4      15      25      8
254        SDCA    28      5       NA     64      2       4      15      25      8
255    PLEASE ENTER ONE OF THE FOLLOWING:
256        1. DISPLAY RED STATUS
257        2. DISPLAY AMBER STATUS
258        3. DISPLAY SELECTED EQUIPMENT
259        4. MANUALLY CHANGE STATUS OF SELECTED EQUIPMENT
260    PLEASE SELECT ONE OF THE FOLLOWING:
261        1. CHANNEL 1-1000     4. MULTIPLEXOR 1-3
262        2. LINK    1-3        5. TRANSMITTER 1-3
263        3. SWITCH  1 OR 2     6. RECEIVER    1-3
264    PLEASE ENTER DEVICE NUMBER TO BE DISPLAYED (FORMAT I4)
265    PLEASE ENTER DEVICE NUMBER TO BE CHANGED
266    FOLLOWED BY NEW VALUE (FORMAT I4,I1)
267    MODIFICATION COMPLETE
268    PLEASE SELECT ONE OF THE FOLLOWING:
269        1. NEW CHANNEL,LINK, OR SWITCH
270        2. NEW MODE OF OPERATION
271        3. SAME MODE OF OPERATION
272        4. LOGOUT
273
274    PRESS RETURN FOR NEXT INSTRUCTION
275
276
277
278
279
280
281
282
283
284
285    REPORTING STATION INDICATOR
```

```
286     REPORT SEQUENCE NUMBER
287     DATE TIME REPORT IS MADE
288     SWITCH IDENTIFIER
289     CHANNEL NUMBER
290     TIME THE OUTAGE BEGAN
291     TIME THE OUTAGE TERMINATED
292     REASON FOR OUTAGE CODE
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313           LOOP -5- LID/FAD CONVERSION TABLE, FOR LCDS 101-255
314     0
315     0
316     0
317     0
318     0
319     0
320     0
321     0
322           -PRESS RETURN FOR NEXT INSTRUCTION-
```

```
FOR M0000/NOLINE
FOR M1000/NOLINE
FOR M2000/NOLINE
FOR M3000/NOLINE
```

```
FOR M0000/NOLINE
FOR M1000/NOLINE
FOR M2000/NOLINE
FOR M3000/NOLINE
```

1.11 SIG, 11/40 Simulating Input Programs

The Simulated Input Generator (SIG) is a microprocessor that generates simulated inputs to the VSQC, DSQC, and BWBSA modules.

It acts as both a communications sensor and a scanner. It is used to simulate the measurements performed on 1000 channels and three links, multiplexors, transmitters, and receivers. Random numbers are generated for the measurements for the Red, Amber, and Green regions. Approximately 10% of the measurements are in the Amber region and 5% in the Red region. The SIG continually writes to three 9600 baud interfaces to the VSQC, DSQC, and BWBSA nodes. Each channel and link is simulated by a measurement report that is written to the interface. A VSQC measurement report consists of the six parameters given in Section 1.3 plus the channel number, trunk, name, and monitor point. A BWBSA measurement report consists of the thirteen parameters given in Section 1.7 plus the link number and monitor point. In addition, Multiplexor, Transmitter, and Receiver Alarms are generated.

The SIG also responds to commands from the VSQC, DSQC or BWBSA to perform a measurement on a specific channel or link. Equipments which are in the Red region continue to generate Red measurement parameters for a specified number of successive measurements in order to simulate hard equipment failures and repair time.

1.11.1 Program Descriptions

1.11.1.1 Program SIGGEN (FORTRAN)

This program is the simulated input generating program, the program loop that continually sends measurements to VSQC, DSQC, and BWBSA nodes. The function is described in Section 1.11.

1.11.1.2 Subroutine IMAGE (FORTRAN)

This subroutine takes a decimal number and converts it to ASCII with leading zeros placed.

1.11.1.3 Subroutine DELAY (FORTRAN)

This subroutine imposes a delay between measurements.

1.11.1.4 Subroutine INTVEC (MACRO)

This subroutine initializes the interrupt vectors of the interface boards.

1.11.1.5 Subroutine RDVSQC (MACRO)

This subroutine reads messages from the VSQC node.

1.11.1.6 Subroutine RDDSQC (MACRO)

This subroutine reads messages from the DSQC node.

1.11.1.7 Subroutine RDBWSA (MACRO)

This subroutine reads mesages from the BWBSA node.

```
0001          PROGRAM SIGGEN
0002          BYTE OUTQ(80),ENC
0003          INTEGER*4 K4,I14
0004          INTEGER*2 K2,DFLAG,VFLAG,BFLAG,DGO,VGO,BGO
0005          REAL*4 ALARM,VMEAS,PRED,VLL,BLL,DLL,VRANGE,DRANGE,BRANGE
0006          REAL*8 VTRNK,DTRNK,BWID,VTR,BTR,DTR
0007          COMMON/EXP/ENC(4)
0008          COMMON/INT/ DFLAG,VFLAG,BFLAG,DGO,VGO,BGO
0009          COMMON/SIG/VMEAS(13),PRED(1003,3),VLL(6),BLL(13,3),DLL,
             1            VRANGE(6),BRANGE(13,3),DRANGE,VTRNK,DTRNK,BWID,
             2            ALARM,VTR,DTR,BTR
0010          DATA VRANGE/14.0,14.0,160.0,140.0,120.0,160.0/
0011          DATA VLL  /-0.5,-0.5,-80.0,-70.0,-60.0,-70.0/
0012          DATA DRANGE,DLL/30.0,-15.0/
0013          DATA BRANGE/3.0,30.0,3600.0,22.0,100.0,4.0,70.0,3.0,
             &            30.0,3600.0,20.0,100.0,30.0,
             &            6.0,60.0,3530.0,19.0,91.0,4.0,70.0,6.0,
             &            60.0,3530.0,19.0,91.0,27.0,
             &            10.0,100.0,3600.0,20.0,100.0,3.9,69.0,
             &            10.0,100.0,3600.0,20.0,100.0,28.0/
0014          DATA BLL   /-1.5,-15.0,0.0,-10.0,-100.0,0.0,-60.0,-1.5,-15.0,
             &            0.0,-10.0,-100.0,-15.0,
             &            -3.0,-30.0,30.0,-9.5,-94.0,0.0,-60.0,-3.0,-30.0,
             &            30.0,-9.5,-94.0,-13.5,
             &            -5.0,-50.0,0.0,-10.0,-100.0,0.1,-59.0,-5.0,-50.0,
             &            0.0,-10.0,-100.0,-14.0/
0015          DATA VTRNK,DTRNK,BWID/'AAAAAAAA','BBBB1111','  111110'/
0016          DATA ALARM/'XXXX'/
0017          DFLAG=0
0018          VFLAG=0
0019          BFLAG=0
0020          VGO=1                !SET REPORT OFF
0021          DGO=1
0022          BGO=1
0023          I1=0
0024          I2=0
0025          DO 999 J=1,1000
0026   999    RN=RAN(I1,I2)
      C
0027          CALL INTVEC          !INITIALIZE VECTORS
0028     100  DO 150,I=1,500
      C
      C       THESE FLAGS ARE SET FROM A INTERRUPT REQUEST
      C      ->IF SET PROGRAM JUMPS TO OUTPUT VALUES AND
      C        THEN RETURNS TO NORMAL LOOP
      C
0029    650   IF(DFLAG .NE. 0) GOTO 651
0031    652   DFLAG=0
0032    653   IF(VFLAG .NE. 0) GOTO 654
0034    655   VFLAG=0
0035    656   IF(BFLAG .NE. 0) GOTO 657
0037    658   BFLAG=0
      C
      C**** VSQC ************************************************
```

```
      C
0038          IVC=I
0039      200 DO 350 J=1,6
0040          JJ=J
0041          RN=RAN(I1,I2)
0042          IF(RN .GE. .998 .OR. RN .LE. .003) GOTO 400
0044          IF(RN .GE. .990 .OR. RN .LE. .005) GOTO 450
0046      350 VMEAS(J)=RN*VRANGE(J)+VLL(J)                    !GREEN
0047          GOTO 500
0048      400 VMEAS(JJ)=RN*VRANGE(JJ)+VLL(JJ)                 !RED
0049          PRED(IVC,1)=PRED(IVC,1)+1
0050          PRED(IVC,2)=JJ
0051          PRED(IVC,3)=VMEAS(JJ)
0052      600 IF(JJ .EQ. 6) GOTO 500
0054          DO 550 K=JJ+1,6
0055      550 VMEAS(K)=.5*VRANGE(K)+VLL(K)
0056          GOTO 500
0057      450 VMEAS(JJ)=RN*VRANGE(JJ)+VLL(JJ)
0058          GOTO 600
0059      500 II=I
0060          IS=JICVT(II,I14)
0061          CALL IMAGE(I14)
0062          RN=RAN(I1,I2)
0063          MON=RN*1000                                     !MON PT
      C
      C     VSQC REQUEST A INTERRUPT IF FIAG SET
      C
0064          IF(VFLAG .EQ. 0) GOTO 659
0066      654 DO 660 K=1,6
0067          RN=RAN(I1,I2)
0068      660 VMEAS(K)=RN*VRANGE(K)+VLL(K)
0069          IF(PRED(VFLAG,1) .EQ. 0) GOTO 661
0071          IS=JAFIX(PRED(VFLAG,2),K4)
0072          IS=IJCVT(K4,K2)
0073          K=K2
0074          VMEAS(K)=PRED(VFLAG,3)
0075      661 IS=JICVT(VFLAG,I14)
0076          CALL IMAGE(I14)
0077          IVC=VFLAG
0078          RN=RAN(I1,I2)
0079          MON=RN*1000
0080          VGO=0
0081      659 CONTINUE
      C
0082          IF(VGO .EQ. 1) GOTO 914
0084          ICNT=38
0085          ENCODE(2,601,OUTQ(1))  ICNT
0086          ENCODE(4,501,OUTQ(3))  VMEAS(1)
0087          ENCODE(4,501,OUTQ(7))  VMEAS(2)
0088          ENCODE(4,501,OUTQ(11)) VMEAS(3)
0089          ENCODE(4,501,OUTQ(15)) VMEAS(4)
0090          ENCODE(4,501,OUTQ(19)) VMEAS(5)
0091          ENCODE(4,501,OUTQ(23)) VMEAS(6)
0092          ENCODE(2,502,OUTQ(27))  IVC
```

```
0093          ENCODE(8,503,OUTQ(29))  VTRNK
0094          DO 711 J=1,4
0095    711   OUTQ(J+32)=ENC(J)
0096          ENCODE(2,504,OUTQ(37))  MON
0097    501   FORMAT(A4)
0098    502   FORMAT(A2)
0099    503   FORMAT(A8)
0100    504   FORMAT(A2)
0101    601   FORMAT(A2)
      D       WRITE(7,900)(VMEAS(IC),IC=1,6),IVC,
      D      &            (OUTQ(II),II=29,36),MON
      D  900 FORMAT(1X,'WRVSQC-> ',6F12.6,/,10X,
      D      &                    I6,2X,8A1,2X,I6,//)
0102          DO 506 J1=1,40
0103          CALL IPOKEB("177416,OUTQ(J1))
0104    505   IF(IPEEK("177414) .NE. "200) GOTO 505
0106    506   CONTINUE
0107          CALL DELAY
0108    998   CONTINUE
0109          VGO=1
0110          IF(VFLAG .NE. 0) GOTO 655
      C
      C**** DSQC ************************************************
      C
0112    914   IDC=I+500
0113   1200   DO 1350 J=1,3
0114          JJ=J
0115          RN=RAN(I1,I2)
0116          IF(RN .GE. .998 .OR. RN .LE. .003) GOTO 1400
0118          IF(RN .GE. .990 .OR. RN .LE. .005) GOTO 1450
0120   1350   VMEAS(J)=RN*DRANGE+DLL
0121          GOTO 1500
0122   1400   VMEAS(JJ)=RN*DRANGE+DLL
0123          PRED(IDC,1)=PRED(IDC,1)+1
0124          PRED(IDC,2)=JJ
0125          PRED(IDC,3)=VMEAS(JJ)
0126   1600   IF(JJ .EQ. 3) GOTO 1500
0128          DO 1550 K=JJ+1,3
0129   1550   VMEAS(K)=.5*DRANGE+DLL
0130          GOTO 1500
0131   1450   VMEAS(JJ)=RN*DRANGE+DLL
0132          GOTO 1600
0133   1500   II=I+500
0134          IS=JICVT(II,I14)
0135          CALL IMAGE(I14)
0136          RN=RAN(I1,I2)
0137          MON=RN*1000
      C
      C       DSQC REQUESTS A INTERRUPT IF FLAG IS SET
      C
0138          IF(DFLAG .EQ. 0) GOTO 662
0140    651   DO 669 K=1,3
0141          RN=RAN(I1,I2)
0142    669   VMEAS(K)=RN*DRANGE+DLL
```

```
0143          IF(PRED(DFLAG,1) .EQ. 0) GOTO 668
0145          IS=JAFIX(PRED(DFLAG,2),K4)
0146          IS=IJCVT(K4,K2)
0147          K=K2
0148          VMEAS(K)=PRED(DFLAG,3)
0149    668   II4=VFLAG-500
0150          IS=JICVT(II4,I14)
0151          CALL IMAGE(I14)
0152          RN=RAN(I1,I2)
0153          MON=RN*1000
0154          DGO=0
0155    662   CONTINUE
      C
0156          IF(DGO .EQ. 1) GOTO 915
0158          ICNT=26
0159          ENCODE(2,601,OUTQ(1))  ICNT
0160          ENCODE(4,501,OUTQ(3))  VMEAS(1)
0161          ENCODE(4,501,OUTQ(7))  VMEAS(2)
0162          ENCODE(4,501,OUTQ(11)) VMEAS(3)
0163          ENCODE(2,508,OUTQ(15)) IDC
0164          ENCODE(8,509,OUTQ(17)) DTRNK
0165          DO 712 J=1,4
0166    712   OUTQ(J+20)=ENC(J)
0167          ENCODE(2,510,OUTQ(25)) MON
0168    508   FORMAT(A2)
0169    509   FORMAT(A8)
0170    510   FORMAT(A2)
      D       WRITE(7,901)(VMEAS(IC),IC=1,3),IDC,(OUTQ(ID),ID=17,24),
      D      &            MON
      D 901   FORMAT(1X,'WRDSQC-> ',3F12.6,/,10X,I6,2X,8A1,2X,I6,//)
0171          DO 512 J1=1,40
0172    511   IF(IPEEK("177514) .NE. "200) GOTO 511
0174          CALL IPOKEB("177516,OUTQ(J1))
0175    512   CONTINUE
0176          CALL DELAY
0177          DGO=1
0178          IF(DFLAG .NE. 0) GOTO 652
      C
      C**** BWBSA ******************************************
      C
0180    915   IF(I .EQ. 100) GOTO 2245
0182          IF(I .EQ. 200) GOTO 2245
0184          IF(I .EQ. 300) GOTO 2245
0186          GOTO 150
0187    2245  LINK=I/100                                    !LINK #
0188          IGRN=0
0189          IBX=1000+LINK
0190    2200  DO 2350 J=1,13
0191          JJ=J
0192          RN=RAN(I1,I2)
0193          IF(RN .GE. .998 .OR. RN .LE. .003) GOTO 2400
0195          IF(RN .GE. .990 .OR. RN .LE. .005) GOTO 2450
0197    2350  VMEAS(J)=RN*BRANGE(J,LINK)+BLL(J,LINK)
0198          IGRN=1                                       !GREEN VALS
```

```
0199          GOTO 2500
0200    2400  VMEAS(JJ)=RN*BRANGE(JJ,LINK)+BLL(JJ,LINK)
0201          PRED(IBX,1)=PRED(IBX,1)+1
0202          PRED(IBX,2)=JJ
0203          PRED(IBX,3)=VMEAS(JJ)
0204    2600  IF(JJ .EQ. 13) GOTO 2500
0206          DO 2550 K=JJ+1,13
0207    2550  VMEAS(K)=.5*BRANGE(K,LINK)+BLL(K,LINK)
0208          GOTO 2500
0209    2450  VMEAS(JJ)=RN*BRANGE(JJ,LINK)+BLL(JJ,LINK)
0210          GOTO 2600
0211    2500  IS=JICVT(LINK,I14)
0212          CALL IMAGE(I14)
0213          RN=RAN(I1,I2)
0214          MON=RN*1000
      C
      C       BWBSQ REQUESTS A INTERRUPT IF FLAG SET
      C
0215          IF(BFLAG .EQ. 0) GOTO 663
0217     657  IBX=1000+BFLAG
0218          DO 664 K=1,13
0219          RN=RAN(I1,I2)
0220     664  VMEAS(K)=RN*BRANGE(K,BFLAG)+BLL(K,BFLAG)
0221          IF(PRED(IBX,1) .EQ. 0) GOTO 665
0223          IS=JAFIX(PRED(IBX,2),K4)
0224          IS=IJCVT(K4,K2)
0225          K=K2
0226          VMEAS(K)=PRED(IBX,3)
0227     665  IS=JICVT(BFLAG,I14)
0228          CALL IMAGE(I14)
0229          RN=RAN(I1,I2)
0230          MON=RN*1000
0231          BGO=0
0232     663  CONTINUE
      C
0233          IF(BGO .EQ. 1) GOTO 150
0235          ICNT=66
0236          ENCODE(2,601,OUTQ(1))  ICNT
0237          ENCODE(4,501,OUTQ(3))  VMEAS(1)
0238          ENCODE(4,501,OUTQ(7))  VMEAS(2)
0239          ENCODE(4,501,OUTQ(11)) VMEAS(3)
0240          ENCODE(4,501,OUTQ(15)) VMEAS(4)
0241          ENCODE(4,501,OUTQ(19)) VMEAS(5)
0242          ENCODE(4,501,OUTQ(23)) VMEAS(6)
0243          ENCODE(4,501,OUTQ(27)) VMEAS(7)
0244          ENCODE(4,501,OUTQ(31)) VMEAS(8)
0245          ENCODE(4,501,OUTQ(35)) VMEAS(9)
0246          ENCODE(4,501,OUTQ(39)) VMEAS(10)
0247          ENCODE(4,501,OUTQ(43)) VMEAS(11)
0248          ENCODE(4,501,OUTQ(47)) VMEAS(12)
0249          ENCODE(4,501,OUTQ(51)) VMEAS(13)
0250          ENCODE(2,515,OUTQ(55)) LINK
0251          ENCODE(8,516,OUTQ(57)) BWID
0252          OUTQ(64)=ENC(4)
```

AD-A078 391 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/G 9/2
SOFTWARE MAINTENANCE MANUAL FOR THE MODULAR SYSTEM CONTROL DEVE--ETC(U)
NOV 79 DCA100-76-C-0083
UNCLASSIFIED 66157 SBIE-AD-E100 313 NL

5 OF 5
AD
A078391

END
DATE
FILMED
1-80
DDC

```
0253          ENCODE(2,517,OUTQ(65)) MON
0254    515   FORMAT(A2)
0255    516   FORMAT(A8)
0256    517   FORMAT(A2)
      D       WRITE(7,902)(VMEAS(IC),IC=1,13),LINK,(OUTQ(ID),ID=57,64),
      D      &          MON
      D 902   FORMAT(1X,'WRBWBSA-> ',5F12.6,/,10X,5F12.6,/,10X,
      D      &       3F12.6,/,10X,I6,2X,8A1,2X,I6,//)
0257          DO 519 J1=1,80
0258    518   IF(IPEEK("175614) .NE. "200) GOTO 518
0260          CALL IPOKEB("175616,OUTQ(J1))
0261    519   CONTINUE
0262          CALL DELAY
0263          BGO=1
0264          IF(BFLAG .NE. 0) GOTO 658
      C
0266          IF(IGRN .EQ. 0) GOTO 150
0268          RN=RAN(I1,I2)
0269          IF(RN .LT. .9) GOTO 150                          !NO ALARM
0271          RN=RAN(I1,I2)
0272          IEQ=3*RN+.5                                  !EQUIP
0273          IF(IEQ .EQ. 0) IEQ=1
0275          ICNT=14
0276          ENCODE(2 ,601,OUTQ(1)) ICNT
0277          ENCODE(4 ,520,OUTQ(3)) ALARM
0278          ENCODE(2 ,521,OUTQ(11)) LINK
0279          ENCODE(2 ,522,OUTQ(13))IEQ
0280    520   FORMAT(A4)
0281    521   FORMAT(A2)
0282    522   FORMAT(A2)
0283          DO 523 J1=1,80
0284    524   IF(IPEEK("175614) .NE. "200) GOTO 524
0286          CALL IPOKEB("175616,OUTQ(J1))
0287    523   CONTINUE
0288          CALL DELAY
      D       WRITE(7,903)LINK,IEQ
      D 903   FORMAT(1X,'WRBWSA-> ','XXXX',2X,I4,2X,I4,//)
      C
      C*************************************************
      C
0289    150   CONTINUE
0290          GOTO 100                                 !START OVER PROG
0291          STOP
0292          END
      C$$$$$$
```

```
0001          SUBROUTINE IMAGE(IN)
0002          BYTE ENC
0003          INTEGER*4 IN
0004          COMMON/EXP/ENC(4)
0005          ENCODE(4,1,ENC) IN
0006    1     FORMAT(4I)
0007          DO 2 J=1,4
0008          IF(ENC(J) .LE. 57 .AND. ENC(J) .GE. 48) GOTO 2
0010          ENC(J)='0'
0011    2     CONTINUE
0012          RETURN
0013          END
      C$$$$$$$
```

```
0001          SUBROUTINE DELAY
0002          DO 1 J1=1,2
0003          DO 2 J2=1,10000
0004     2    CONTINUE
0005     1    CONTINUE
0006          RETURN
0007          END
```

```
                                        .TITLE SIG.MACRO
                                        .IDENT/V1.0/
                                        .GLOBL  INTVEC,RDVSQC,RDDSQC,RDBWSA
000000                                  .PSECT

                                        ;INTERFACE ADDRESS

                                        ;DSQC
            177510                      RCSR1=177510
            177512                      RBUF1=177512
            177514                      XCSR1=177514
            177516                      XBUF1=177516

                                        ;VSQC
            177410                      RCSR2=177410
            177412                      RBUF2=177412
            177414                      XCSR2=177414
            177416                      XBUF2=177416

                                        ;BWBSA
            175610                      RCSR3=175610
            175612                      RBUF3=175612
            175614                      XCSR3=175614
            175616                      XBUF3=175616

                                        ;VECTOR INIT

000000  000005              INTVEC: RESET
000002  012700  000100              MOV     #100,   R0
000006  012710  000110'            MOV     #TIME,  (R0)
000012  012700  000320              MOV     #320,   R0
000016  012720  000112'            MOV     #RDVSQC,(R0)+
000022  012720  000340              MOV     #340,   (R0)+
000026  052737  000100  177410      BIS     #100,   @#RCSR2
000034  012700  000310              MOV     #310,   R0
000040  012720  000164'            MOV     #RDDSQC,(R0)+
000044  012720  000340              MOV     #340,   (R0)+
000050  052737  000100  177510      BIS     #100,   @#RCSR1
000056  012700  000300              MOV     #300,   R0
000062  012720  000226'            MOV     #RDBWSA,(R0)+
000066  012720  000340              MOV     #340,   (R0)+
000072  052737  000100  175610      BIS     #100,   @#RCSR3
000100  012700  000000              MOV     #000,   R0
000104  106400                      MTPS    R0
000106  000207                      RTS     PC
000110  000002              TIME:   RTI

000112  010046              RDVSQC: MOV     R0,     -(SP)
000114  105737  177410              TSTB    @#RCSR2
000120  100015                      BPL     1$
000122  113700  177412              MOVB    @#RBUF2, R0
000126  042700  177400              BIC     #177400, R0
000132  122700  000122              CMPB    #122,R0
000136  001004                      BNE     4$
000140  012767  000000  000010'     MOV     #0,     VGO
000146  000402                      BR      1$
```

```
000150  010067  000002'        4$:     MOV     R0,     VFLAG
000154  012600                  1$:     MOV     (SP)+,  R0
000156  000002                          RTI

000160  010046                  RDDSQC: MOV     R0,     -(SP)
000162  105737  177510                  TSTB    @#RCSR1
000166  100015                          BPL     2$
000170  113700  177512                  MOVB    @#RBUF1, R0
000174  042700  177400                  BIC     #177400, R0
000200  122700  000122                  CMPB    #122,   R0
000204  001004                          BNE     5$
000206  012767  000000  000006'        MOV     #0,     IGO
000214  000402                          BR      2$
000216  010067  000000'        5$:     MOV     R0,     DFLAG
000222  012600                  2$:     MOV     (SP)+,  R0
000224  000002                          RTI

000226  010046                  RDBWSA: MOV     R0,     -(SP)
000230  105737  175610                  TSTB    @#RCSR3
000234  100015                          BPL     3$
000236  113700  175612                  MOVB    @#RBUF3, R0
000242  042700  177400                  BIC     #177400, R0
000246  122700  000122                  CMPB    #122,   R0
000252  001004                          BNE     6$
000254  012767  000000  000012'        MOV     #0,     BGO
000262  000402                          BR      3$
000264  010067  000004'        6$:     MOV     R0,     BFLAG
000270  012600                  3$:     MOV     (SP)+,  R0
000272  000002                          RTI

000000                                  .PSECT  INT,RW,D,GBL,REL,OVR

000000  000000                  DFLAG:  .WORD   0
000002  000000                  VFLAG:  .WORD   0
000004  000000                  BFLAG:  .WORD   0
000006  000000                  IGO:    .WORD   0
000010  000000                  VGO:    .WORD   0
000012  000000                  BGO:    .WORD   0

        000000'                         .END INTVEC
```

```
SIG.MACRO       MACRO V03.02B 01:22:36 PAGE 1-2
SYMBOL TABLE

BFLAG   000004R     002 RBUF1 = 177512        RCSR3 = 175610        VFLAG   000002R     002 XBUF3 = 175616
BGO     000012R     002 RBUF2 = 177412        RDB%SA  000226RG      VGO     000010R     002 XCSR1 = 177514
DFLAG   000000R     002 RBUF3 = 175612        RDDSQC  000160RG      XBUF1 = 177516          XCSR2 = 177414
LGC     000006R     002 RCSR1 = 177510        RDVSQC  000112RG      XBUF2 = 177416          XCSR3 = 175614
INTVEC  000000RG        RCSR2 = 177414        TIME    000110R

. ABS.  000000      000
        000274      001
INT     000014      002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  294 WORDS  ( 2 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  56 PAGES
DK:SIGMAC,DK:SIGMAC=DK:SIGMAC.M20
```

```
    FORTRAN IV V01C-03          FRI 03-JUN-77 12:26:53          PAGE 001
CORE=08K, UIC=[20,20]                       SDCA5.OBJ,SDCA5.LST=SDCA5.FOR

      C
      C THIS PROGRAM -SIM1(SDCA5.FOR)  WRITES SIMULATED SWITCH
      C  CONDITION REPORTS TO SDCA LOOP 5
      C  THIS PROGRAM RUN ON HOST 11/40
      C
0001          DIMENSION ICRIT(2,16),IRPT(16)
0002          CALL ASSIGN(1,'TI:')
0003          DATA ICRIT(1,1),ICRIT(1,2),ICRIT(1,3)/1,512,25/
0004          DATA ICRIT(1,4),ICRIT(1,5),ICRIT(1,6)/25,10,50/
0005          DATA ICRIT(1,7),ICRIT(1,8),ICRIT(1,9)/50,10,10/
0006          DATA ICRIT(1,10),ICRIT(1,11),ICRIT(1,12)/10,128,128/
0007          DATA ICRIT(1,13),ICRIT(1,14),ICRIT(1,15)/128,128,10/
0008          DATA ICRIT(1,16)/10/
0009          DATA ICRIT(2,1),ICRIT(2,2),ICRIT(2,3)/2,256,10/
0010          DATA ICRIT(2,4),ICRIT(2,5),ICRIT(2,6)/10,25,40/
0011          DATA ICRIT(2,7),ICRIT(2,8),ICRIT(2,9)/40,5,5/
0012          DATA ICRIT(2,10),ICRIT(2,11),ICRIT(2,12)/10,64,64/
0013          DATA ICRIT(2,13),ICRIT(2,14),ICRIT(2,15)/64,64,5/
0014          DATA ICRIT(2,16)/5/
0015          I1=0
0016          I2=0
0017    200   IDO=1
0018    10    IRPT(1)=IDO
0019          R1=RAN(I1,I2)
0020          DO 20 I=3,16
0021    20    IRPT(I)=ICRIT(IDO,I)*R1
0022          IF(R1 .LE. .98) GOTO 100
      C         ELSE SATURATED COND.
0024          IR2=13*RAN(I1,I2)
0025          IF(IR2 .LT. 3) IR2=3
0027          IF(IR2 .GE. 16) IR2=16
0029          IRPT(IR2)=(R1*5)+ICRIT(IDO,IR2)
0030    100   IRPT(2)=IRPT(11)+IRPT(12)+IRPT(13)+IRPT(14)
0031          WRITE(1,110)(IRPT(I),I=1,16)
0032          WRITE(2,110)(IRPT(I),I=1,16)
0033    110   FORMAT(1X,16I4)
      C           WAIT 5.5 SEC
0034          T1=SECNDS(0.)
0035    150   DELTA=SECNDS(T1)
0036          IF(DELTA .LT. 5.5) GOTO 150
0038          IF(IDO .EQ. 2) GOTO 200
0040          IDO=2
0041          GOTO 10
0042          STOP
0043          END
```

```
FOR SIGGEN/NOLINE
MAC SIGMAC
```

```
ASS DX0 DK
R LINK
DX1:STGGEN=DX1:MAC=DX1:STGGEN/I/C/W:1
DX1:STGMAC//
$SIMKI

ASS DX1 DK
```