

AD-A078 391 BURROUGHS CORP PAOLI PA FEDERAL AND SPECIAL SYSTEMS GROUP F/6 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



**LEVEL**

AD-E 100 313

66157

Book 1

November 1979

(M)

ADA 078391

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

DDC  
RECEIVED  
DEC 17 1979  
E

DDC FILE COPY

for

THE DEFENSE COMMUNICATIONS AGENCY  
WASHINGTON, D.C. 20305

**Burroughs Corporation**

Federal and Special Systems Group

Paoli, Pa. 19301

This document has been approved  
for public release and sale; its  
distribution is unlimited.

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 DCS System Control Loop Network Distributed Computer System Ring Network Modular Architecture		
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.		

79 22 5 043

Book 1

11

14  
66157

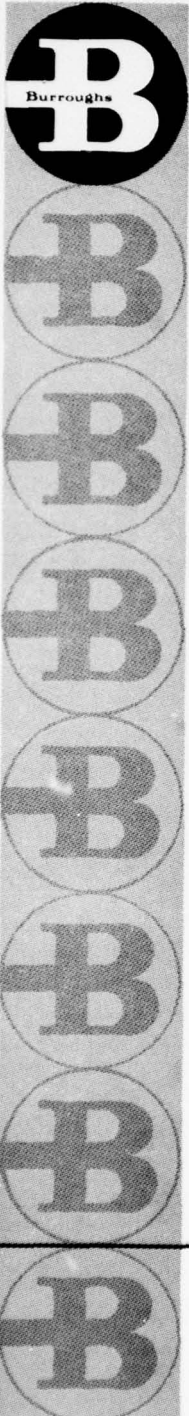
11  
November 1979

18 SBIE

12  
392

17  
AD-E-100 313

DDC  
RECEIVED  
DEC 17 1979  
E



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 DCA 100-76-C-0083

**Burroughs Corporation**

Federal and Special Systems Group

Paoli, Pa. 19301

070 040

This document has been approved  
for public release and sale; its  
distribution is unlimited.

YB

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.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	<input type="checkbox"/>
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

MSCDM SOFTWARE MAINTENANCE MANUAL

Book 1		PAGE
	INTRODUCTION	1
1.0	<u>MSCDM DMCP AND APPLICATION PROGRAMS</u>	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	<u>UTILITY PROGRAMS</u>	
2.1	PAGE PROGRAM	
2.2	FORMAT PROGRAM	
2.3	STATUS PROGRAM	
2.4	FDMLDR PROGRAM	
3.0	<u>MACRO-11 PROGRAMS</u>	
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: Interrupt 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.

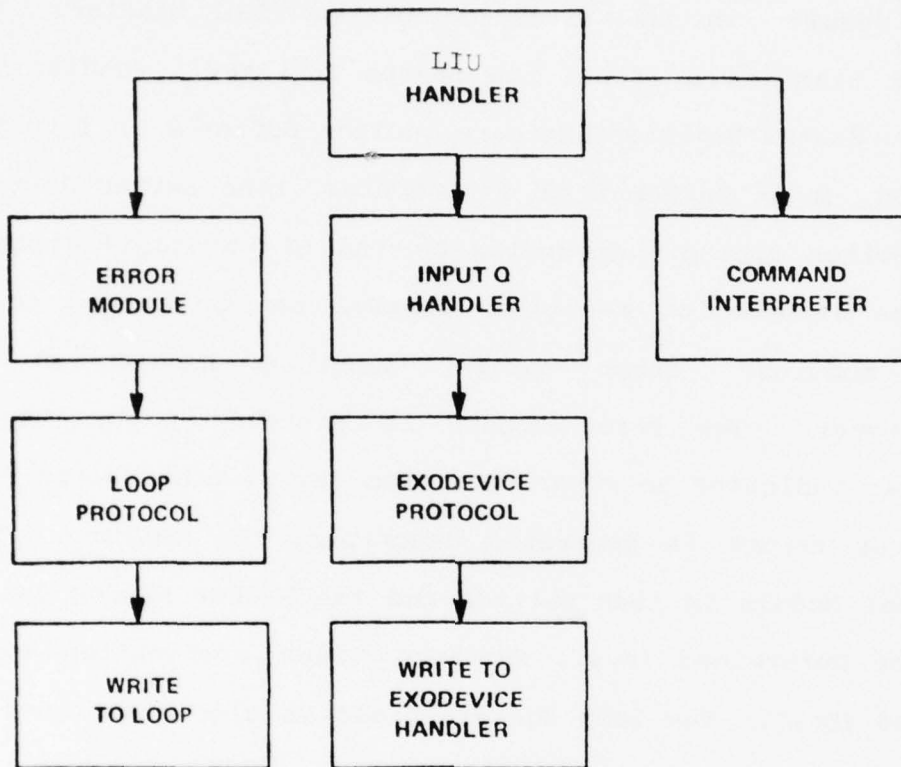


Figure 3. Input from Loop Flow Control

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.

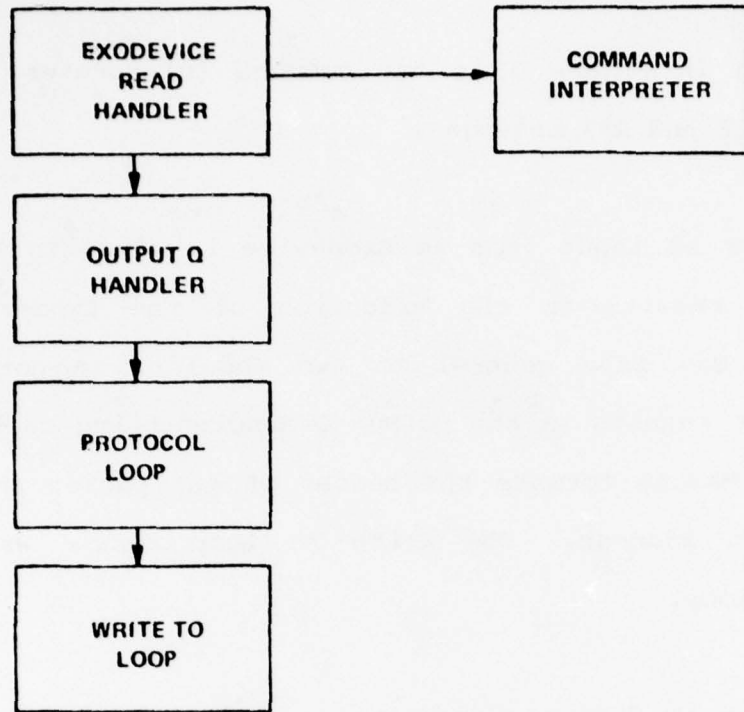


Figure 4. Input from Exodevice Flow Control

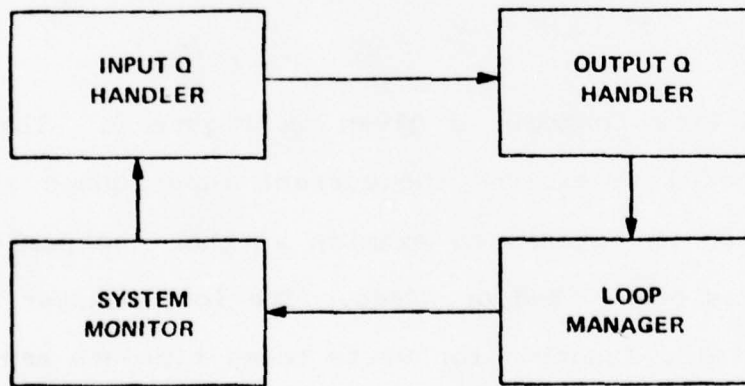


Figure 5. DMCP Idle Flow Control

#### 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 is entered by an 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 interrupts or raises priority thus disabling interrupts from being acknowledged.

#### 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 queuing 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.

PAGE 001

V02.1-1 Mon 04-Jun-79 17:07:01

FORTRAN IV

```

0001 PROGRAM NODAL
0002 INTEGER*2 XINQ,XOUTQ,ACKQ,FXINQ,FXOUTQ,PACKQ
0003 INTEGER*2 PINQ,FREE,STAT,FLWMT,OUTFCT
0004 INTEGER*2 SETPRM,RSTPRM,SETBKF,CCI,CC2
0005 INTEGER*2 RSTBKF,Q1,Q2,RESNLM,OUTQ,DEQUE
0006 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFQ,OUTRF,INBF
0007 LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0008 INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRITM,ITIMLIM,ATIMLM
0009 COMMON /MESS/ MESSEQ
0010 COMMON /DFM/ OUTBF(256),INBF(256),IWRITM,IWRT,STAT,LTIME
0011 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
0012 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0013 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
0014 LIDFQ(256),IOFLG,LLFLG,IRSEND
0015 COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0016 COMMON /SWT/ SETPRM,RSTPRM,SETBKF,RSTBKF
0017 COMMON /GLOB/ ISENT,FLWMT,IGLNTH,OUTFCT,IFULL,OUTQ,
0018 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0019 CALL MASTER
0020 CALL INIT
0021 CALL ENABLE(1)
0022 5 CONTINUE
0023 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 DESTP(PACK(254,Q2),LEN)
0028 DO 30 I=1,LEN
0029 XOUTBF(I)=PACK(I,Q2)
0030 CONTINUE
0031 CALL GOUT(LEN,IST)
0032 IFR=IFR+1
0033 FREE(IFR)=Q2
0034 CALL STATB(IS)
0035 IF(IS.EQ.1) GOTO 110
0037 IF(IOFLG.EQ.0) GOTO 81
0039 CALL ENABLE(0)
0040 Q2=IGETSP(N)
0041 CALL ENABLE(1)
0042 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 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)

```

PAGE 002

FORTRAN IV V02.1-1 Mon 04-Jun-79 17:07:01

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

PAGE 001

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

```
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      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 17:07:31      PAGE 001

0001      SUBROUTINE ENQUE(A,B,N)
0002      LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003      INTEGER*2 XING,PXING,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PING
0004      INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005      INTEGER*2 RESNLM,OUTFCT
0006      COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),
0007      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,IQLNTH,OUTFCT,IFULL,OUTQ,
0008      1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009      10      IQHEAD=A(1)
0010      IQTAIL=A(2)
0011      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
0017      NN=IQHEAD-IQTAIL
0018      DO 30 I=1,NN+1
0019      30      B(IQLNTH+1-I)=B(IQHEAD+1-I)
0020      A(1)=IQLNTH
0021      A(2)=IQLNTH-NN
0022      GOTO 10
0023      40      CALL INIT
0024      999      RETURN
0025      END
0026

```

```

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,FXOUTQ,ACKQ,FACKQ,INQ,FINQ
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),FINQ(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
0010      IF(IQHEAD.NE.0) GOTO 10
0011      A(1)=IQLNTH
0012      A(2)=IQLNTH+1
0013      GOTO 999
0014      10 A(1)=IQHEAD-1
0015      999 RETURN
0016      END
0017
0018

```

```

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

0001      SUBROUTINE INPTQ(L)
0002      INTEGER*2 STAT,XINQ,FXINQ,XOUTQ,FXOUTQ,ACKQ
0003      INTEGER*2 PACKQ,PING,FLWNT,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 /QUE/ XINQ(16),FXINQ(2),XOUTQ(16),FXOUTQ(2),
0010      COMMON /GLOB/ ISENT,FLWNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1          RESNLM,IALRT,ISLID,MONTOR,LF,ETX,CR,DUM
0011      IF(L.LT. 1) GOTO 999
0012      LI=L-2
0013      T1=IGETSP(N)
0014      DO 10 I=1,LI
0015      PACK(I,T1)=INBF(I)
0016      CALL ENSTR(PACK(254,T1),LI)
0017      CALL ENQUE(FINQ,ING,I1)
0018      RETURN
0019      999
0020      END
0021

```

PAGE 001

Mon 04-Jun-79 17:08:19

V02.1-1

FORTRAN IV

```
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,IWRITM
0006 COMMON/DFM/OUTBF(256),INBF(256),IWRITM,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
```

PAGE 001

Mon 04-Jun-79 17:08:30

FORTRAN IV

v02.1-1

```

0001 SUBROUTINE INIT
0002 REAL*4 RH,AL,AH,AL,IRSV,U,MEAS
0003 REAL*8 UTR
0004 INTEGER*2 XING,ACKQ,FXING,FXOUTQ,PACKQ,PING,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,DATA
0009 INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,IWRTTH
0010 COMMON /DFM/ OUTBF(256),INBF(256),
1 IWRTTH,IWRT,STAT,LTIME
0011 COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),FXOUTQ(2),
1 XINQ(16),PACKQ(2),INQ(16),PING(2)
0012 COMMON /MESS/ MESSQ
0013 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSER(256),LIDFD(256),
1 IOFLG,LLFLG,IRSEND
0014 COMMON /FRE/ FREE(64),IFR,IFRSZ
0015 COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0016 COMMON /PACK/ 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 MESSQ=0
0023 IOFLG=0
0024 IRSEND=0
0025 LLFLG=0
0026 IFRSZ=64
0027 INFLOT=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 IWRTTH=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 LIDFD(I)=4
0050 LIDFD(21)=1

```

PAGE 002

Mon 04-Jun-79 17:08:30

FORTRAN IV V02.1-1

```

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 LIDFD(I)=0
0060 DO 60 I=40,44
0061 LIDFD(I)=1
0062 DO 70 I=45,59
0063 LIDFD(I)=0
0064 DO 80 I=60,64
0065 LIDFD(I)=2
0066 DO 90 I=65,79
0067 LIDFD(I)=0
0068 DO 100 I=80,84
0069 LIDFD(I)=4
0070 DO 110 I=85,99
0071 LIDFD(I)=0
0072 DO 120 I=100,104
0073 LIDFD(I)=5
0074 DO 130 I=105,256
0075 LIDFD(I)=0
0076 PING(1)=IQLNTH
0077 PING(2)=IQLNTH+1
0078 PACKQ(1)=IQLNTH
0079 PACKQ(2)=IQLNTH+1
0080 PXOUTQ(1)=IQLNTH
0081 PXOUTQ(2)=IQLNTH+1
0082 PXING(1)=IQLNTH
0083 PXING(2)=IQLNTH+1
0084 DO 10 I=1,IFRSZ
0085 PACK(255,I)=0
0086 DO 20 I=1,IFRSZ
0087 FREE(I)=1
0088 DO 30 I=1,256
0089 ACKSEQ(I)=256
0090 IFR=IFRSZ
0091 RETURN
0092 END
0093

```

FORTRAN IV V02.1-1 Mon 04-Jun-79 17:08:50 PAGE 001

```

0001 SUBROUTINE LINLOS
0002 REAL*8 RMI(5),LINO8,LIN18
0003 INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004 INTEGER*2 XING,FXING,XOUTQ,ACKQ,PACKQ,FINQ
0005 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006 LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0007 LOGICAL*1 PACK,M1(40),LINO(8),LINI(8)
0008 COMMON/QUE/XING(16),PXING(2),XOUTQ(16),FXOUTQ(2),
      * ACKQ(16),PACKQ(2),ING(16),PING(2)
0009 COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
      * IOFLG,LLFLG,IRSEND
0010 COMMON /GLOB/ ISENT,FLWCNT,IOLNTH,OUTFCT,IFULL,OUTQ,
      * RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0011 COMMON PACK(256,64)
0012 COMMON /MESS/ MESSEQ
0013 DATA LINO8,LIN18/'PRIMARY ','BACKUP '//
0014 DATA RMI(1),RMI(2)/'LOSS OF ','MODULATI'//
0015 DATA RMI(3),RMI(4)/'ON ON LO','OP AT NO'//
0016 DATA RMI(5)/'DE 21 '//
0017 EQUIVALENCE(LINO,LINO8)
0018 EQUIVALENCE(LINI,LIN18)
0019 EQUIVALENCE(M1,RMI)
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   PACK(I+9,T1)=M1(I)
0038   IF(IS.EQ.1) GOTO 50
0040   DO 40 I=1,8
0041     PACK(I+31,T1)=LINO(I)
0042   GOTO 70
0043   DO 60 I=1,8
0044     PACK(I+31,T1)=LINI(I)
0045   DO 80 I=23,40
0046     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(FXING,XING,T1)
0053   CALL ENABLE(1)

```

PAGE 002

Mon 04-Jun-79 17:08:50

FORTRAN IV V02.1-1

0054 LLFLG=0  
0055 RETURN  
0056 END

```

1 .TITLE FDM.MACRO
2 .SBTTL NODE21
3 .IDENT /V3.0/
4 .GLOBL LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
5 .GLOBL WTKEN,PRAM,STATUS,STAI40
6 .GLOBL LFOUT,*RSTART,*RDFNT,*DESTR,ENSTR,LFINPT
7 .NLIST CND
8 .PSECT
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
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

```

000000
000001
000002
000003
000004
000005
000006
000007

R0 =%0
R1 =%1
R2 =%2
R3 =%3
R4 =%4
R5 =%5
SP =%6
PC =%7

```

!INTERFACE ADDRESS

```

172410 BAR= 172410
172412 WCR= 172412
172414 CSR= 172414
172416 IOBUF= 172416
172416 OFREG= 172416
177560 HRCR= 177560
177562 HRBUF= 177562
177564 HXCSR= 177564
177566 HXRUF= 177566

```

!INTERFACE OPTIONS

```

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

```

!LOCAL VARIABLES

```

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  
NODE21

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

000002  
000001

BIT01= 2  
BIT00= 1

66  
67  
68

```

1 *****
2 ***** PROCEDURE MASTER START UP *****
3 *****
4 *****
5 *****
6 MASTER: MOV #340, R0 ;PRI=7
7 MTPS R0
8 MOV #124, R0 ;LIU HANDLER
9 MOV #LI0, (R0)+ ;PRI=7
10 MOV #340, (R0) ;RESET BUS
11 RESET ;INTERFACE VECTOR
12 MOV #60, R0 ;GATE WAY HANDLER
13 MOV #610, (R0)+ ;PRI=7
14 MOV #340, (R0)+ ;CLOCK VECTOR
15 BIS #BIT06, @#HKCSR ;DISABLE INTERRUPTS
16 MOV #100, R0 ;CLEAR LIU
17 MOV #TIME, (R0) ;CLEAR BLIUI CSR
18 BIC #40100, @#CSR ;PRI=0
19 JSR FC, LIUINT ;CPU DOWN
20 CLR @#CSR
21 MOV #000, R0
22 MTPS R0
23 RTS PC
24
25 *****
26 ***** PROCEDURE INITIALIZE *****
27 *****
28 *****
29 *****
30 *****
31 iCLEAR AGRAM
32 LIUINT: MOV #4354,, @#DFREG
33 MOV #2304,, @#DFREG
34 TSTB @#CSR
35 BPL -4
36 CLRB @#CSR
37 MOV #4353,, @#DFREG
38 MOV #-256,, R0
39 TSTB @#CSR
40 BPL -4
41 CLRB @#CSR
42 INC R0
43 BNE 1$
44
45 iCLEAR INFUT/OUTPUT BUFFERS
46 MOV #10410, R1
47 MOV #10440, R0
48 JSR FC, ZEROPF
49 MOV #1400, @#DFREG
50 TSTB @#CSR
51 BPL -4
52 CLRB @#CSR
53 MOV #10610, R1
54 MOV #10640, R0
55
56 *****
57 ***** PROCEDURE INITIALIZE *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****

```

```

98 000234 004767 000052          JSR      ZEROBP
99 000240 012737 001400 172416  MOV     @#OPREG
100 000246 105737 172414          TSTB
101 000252 100375          BPL     -4
102 000254 105037 172414          CLRB   @#CSR
103 000260 012701 010510  MOV     @#CSR, R1
104 000264 012700 010540  MOV     @#CSR, R0
105 000270 004767 000016  JSR     ZEROBP
106 000274 012701 010710  MOV     @#CSR, R1
107 000300 012700 010740  MOV     @#CSR, R0
108 000304 004767 000002  JSR     ZEROBP
109 000310 000440          BR      STAT$
110 000312 010137 172416  MOV     @#OPREG
111 000316 012737 001400 172416  MOV     @#OPREG
112 000324 105737 172414          TSTB
113 000330 100375          BPL     -4
114 000332 105037 172414          CLRB   @#CSR
115 000336 013702 172416  MOV     @#OPREG, R2
116 000342 042702 177400  BIT     @#CSR
117 000346 010037 172416  MOV     @#CSR, R0
118 000352 022702 000000  CMP     @#CSR, R2
119 000356 001412  BEQ     3$
120 000360 012737 001400 172416  MOV     @#OPREG
121 000366 105737 172414          TSTB
122 000372 100375          BPL     -4
123 000374 105037 172414          CLRB   @#CSR
124 000400 005302  DEC     R2
125 000402 000763  BR      2$
126 000404 010037 172416  MOV     @#OPREG
127 000410 000207  RTS
128
129
130
131 000412 012737 010400 172416  MOV     @#OPREG
132 000420 012737 002400 172416  MOV     @#OPREG
133 000426 012737 002400 172416  MOV     @#OPREG
134 000434 000207  RTS
135
136

```

```

;PC=0
;FALSE READ DATA
;GOOD RD
;NO RETRY
;CLEAR DONE BIT
;RDBUFADR COMMAND
;SEL OUTBUFO COMMAND
;RDBUFADR COMMAND
;SEL OUTBUFO COMMAND
;ZERO BUFFER POINTER
;GO CLEAR STATUS
;RDBUFADR
;RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;FETCH POINTER
;CLEAR MST BYTE
;SEL BUFFER
;PC=0?
;YES RECHECK
;NO FALSE RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;LOCAL POINTER-1
;R UNTIL 0
;SEL BUFFER
;RETURN
;MCR:RS
;RS(FALSE)
;RS

```

\*\*\*\*\*  
\*\*\*\*\* PROCEDURE LIU-HANDLER \*\*\*\*\*  
\*\*\*\*\*

ISAVE R0  
ISAVE R1  
ISAVE R2  
ISAVE R3  
ISAVE R4  
ISAVE R5  
IF LIU INTERRUPTED  
END  
IF YES/DISABLE LIU  
IF WCR : RS(O)  
IF RD  
IF CLEAR UNUSED BITS

RO, -(SP)  
R1, -(SP)  
R2, -(SP)  
R3, -(SP)  
R4, -(SP)  
R5, -(SP)  
IF CCR  
R1, @CCR  
IF BIT14, @CCR  
IF @352, @OFFREG  
IF @1280, @OFFREG  
IF @IDRUF, CAUSE  
IF @177400, CAUSE

IF REFORM OPERATION DEPENDING ON BITS

IFOV-FL  
IFRDBUFADR INO  
IFSEL INBUFO  
IFGO EMPTY  
IFOV-FL  
IFRDBUFADR IN1  
IFSEL INBUF1  
IFGO EMPTY  
IFLINE-LOSS PRIMARY  
IFLATCH ON  
IFYES LOOP AROUND  
IFFLAG PRIMARY SW, FAIL  
IFBACKUP LINE LOSS  
IFFLAG BACKUP SW, FAIL  
IFWRITE TOKEN DETECT  
IFCLEAR TOKEN VAR

IF BIT02, CAUSE  
R3\$  
IF @4360, R2  
IF @4384, R3  
PC, EMBF  
IF BIT03, CAUSE  
R4\$  
IF @4488, R2  
IF @4512, R3  
PC, EMBF  
IF BIT04, CAUSE  
R5\$  
IF BIT07, CAUSE  
R2, LFLG  
IF BIT05, CAUSE  
R6\$  
IF LFLG  
IF BIT06, CAUSE  
R0\$  
IF WRTM  
IF WRT

IF BUFFER FULL  
IF INBUFO FULL  
END  
IF BUFFER ADDRESS  
IFRDBUFADR COMM  
IFSEL BUF  
IFCRC BIT  
IFEMPTY BUFFER  
IFDATA LINK AREA  
IFVARIABLE  
IFCRC OR BYTE COUNT  
IFCALL FORTRAN QUEUER

IF BIT00, CAUSE  
R1\$  
IF INBF, R1  
IF @4360, R2  
IF @4384, R3  
IF BIT00, R4  
PC, EMBF  
IF @AREA, R5  
IF @1, AREA  
IF @2, DATA  
IF @, LPRINT

Line	Address	Operation	Source	Destination
1	000436	MOV	010046	RO
2	000440	MOV	010146	R1
3	000442	MOV	010246	R2
4	000444	MOV	010346	R3
5	000446	MOV	010446	R4
6	000450	MOV	010546	R5
7	000452	IFST	172414	@CCR
8	000456	BIC	172414	@CCR
9	000460	BIC	040000	@BIT14
10	000466	MOV	012737	@OFFREG
11	000474	MOV	012737	@OFFREG
12	000502	MOV	172416	@IDRUF
13	000510	BIC	177400	@177400
14	000516	IF	000004	BIT02
15	000524	MOV	01406	R3
16	000526	MOV	012702	@4360
17	000532	MOV	010440	@4384
18	000536	JSR	000270	EMBF
19	000542	IF	000010	BIT03
20	000550	MOV	01406	R4
21	000552	MOV	010610	@4488
22	000556	MOV	010640	@4512
23	000562	JSR	000244	EMBF
24	000566	IF	000020	BIT04
25	000574	MOV	01407	R5
26	000576	MOV	132767	@BIT07
27	000604	IF	001403	R2
28	000606	MOV	002002	LFLG
29	000614	IF	000040	BIT05
30	000622	MOV	001403	R6
31	000624	MOV	002002	LFLG
32	000632	IF	000100	BIT06
33	000640	MOV	001404	R0
34	000642	CLR	001000	WRTM
35	000646	CLR	001002	WRT
36	000652	IF	000001	BUFFER FULL
37	000660	MOV	132767	INBUFO
38	000662	MOV	01423	END
39	000666	MOV	012701	ADDRESS
40	000672	MOV	012703	COMM
41	000676	MOV	012704	SEL
42	000702	JSR	004767	EMPTY
43	000706	MOV	012705	LINK AREA
44	000712	MOV	012767	VARIABLE
45	000720	MOV	010267	CRC OR BYTE COUNT
46	000724	MOV	004767	CALL FORTRAN QUEUER

58	000730	132767	000002	000254	R1\$:	R1B:	CAUSE
59	000736	001423			RTI\$	REG	#BIT01, FULL
60	000740	012701	000400		#INBF, R1	MOV	#NO WERE DONE
61	000744	012702	010610		#488., R2	MOV	#BUFFER ADDRESS
62	000750	012703	010640		#4512., R3	MOV	#RDBUFADR COMM
63	000754	012704	000002		#BIT01, R4	MOV	#SEL INBUFI
64	000760	004767	000046		FC, EMBF	JSR	#CRC BIT
65	000764	012705	001214		#AREA, R5	MOV	#GO EMPTY BUFFER
66	000770	012767	000001	000216	#1, AREA	MOV	#DATA LINK AREA
67	000776	010267	000216		R2, DATA	MOV	#ONE VARIABLE
68	001002	004767	000000G		FC, LFINFT	JSR	#CALL FORTRAN QUE'ER
69							
70	001006	012605			(SP)+, R5	MOV	#RESTORE REGISTERS
71	001010	012604			(SP)+, R4	MOV	
72	001012	012603			(SP)+, R3	MOV	
73	001014	012602			(SP)+, R2	MOV	
74	001016	012601			(SP)+, R1	MOV	
75	001020	012600			(SP)+, R0	MOV	
76	001022	052737	040000	172414	#BIT14, @#CSR	BIS	#ENABLE INTERRUPTS
77	001030	000002				RTI	#RETURN FROM INTERRUPT
78							
79	001032	012737	010600	172416	EMBF:	MOV	#READ STATUS 1
80	001040	012737	002400	172416		MOV	#RS
81	001046	013700	172416		@#IOBUF, R0	MOV	#FETCH STATUS
82	001052	130400			R4, R0	BITB	#GOOD CRC
83	001054	001002			CRCOK	BNE	
84	001056	012704	177777		#-1, R4	MOV	#NO FLAG
85	001062	010237	172416		R2, @#DFREG	MOV	#RDBUFADR
86	001066	012737	001400	172416	#768., @#DFREG	MOV	#RD
87	001074	105737	172414		@#CSR	TSTB	#GOOD RD
88	001100	100375			.-4	BPL	
89	001102	013702	172416		@#IOBUF, R2	MOV	#REPLACE WITH POINTER
90	001106	042702	177400		#177400, R2	BIC	#CLEAR MST BITS
91	001112	010200			R2, R0	MOV	#SAVE IT
92	001114	005400			NEG R0	MOV	#2'S COMP
93	001116	010037	172412		R0, @#WCR	MOV	#BYTE COUNT NOW
94	001122	010137	172410		R1, @#BAR	MOV	#ADDRESS IN MEMORY
95	001126	010337	172416		R3, @#DFREG	MOV	#SEL BUFFER
96	001132	012737	001400	172416	#768., @#DFREG	MOV	#FALSE RD
97	001140	105737	172414		@#CSR	TSTB	#DONE ON
98	001144	100375			.-4	BPL	#NO LOOP
99	001146	012737	021000	172416	#8704., @#DFREG	MOV	#FIRE DMA
100	001154	000240			NOP	MOV	#DELAY
101	001156	105737	172414		@#CSR	TSTB	#GOOD DMA
102	001162	100401			DMAOK	BMI	
103	001164	000240			NOP	MOV	#ERROR IF HERE
104	001166	012737	004400	172416	DMAOK:	MOV	#FALSE WD
105	001174	105737	172414		@#CSR	TSTB	#GOOD WD
106	001200	100375			.-4	BPL	
107	001202	005704			TST R4	TST	#WAS CRC OK
108	001204	100001			BPL	MOV	#YES
109	001206	010402			END\$	MOV	#NO FLAG IT
110	001210	000207			R4, R2	MOV	#RETURN
111	001212	000000			RTS	RTS	#STATUS BYTE 0 HOLDER
112	001214	000000			.WORD	0	
113	001216	001220			CAUSE:	.WORD	
114	001220	000000			AREA:	.WORD	
					DATA:	.WORD	

```

1  .ENABLE LSR
2  *****
3  ***** PROCEDURE ENABLE ***** *
4  *****
5
6  001222 017501 000002          @ARG1(R5), R1
7  001226 022701 000001          #1, R1
8  001232 001404
9  001234 042737 040000 172414 E0$: #BIT14, @#CSR
10 001242 000207          PC
11 001244 052737 040000 172414 E1$: #BIT14, @#CSR
12 001252 000207          PC
13
14
15
16 001254 017575 000004          @ARG2(R5), @ARG1(R5)
17 001262 000207          PC
18 001264 017575 000002 000004 @ARG1(R5), @ARG2(R5)
19 001272 000207          PC
20
21
22 *****
23 ***** PROCEDURE LINE SWITCH ***** *
24 *****
25
26
27
28
29
30
31
32
33 001274 017501 000002          @ARG1(R5), R1
34 001300 012737 010420 172416 #A368., @#OFREG
35 001306 062701 004400          #2304., R1
36 001312 010137 172416          MOV R1, @#OFREG
37 001316 105737 172414          TSTB @#CSR
38 001322 100375          BPL #-4
39 001324 105037 172414          CLRB @#CSR
40 001330 000207          RTS PC
41
42

```

```

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 STATUS: MOV @ARG1(R5), R1 ;WHICH BYTE
11 CMP #1, R1 ;COMPARE
12 REQ 2$
13 MOV #4352,, @#DFREG ;WCR : RS(0)
14 MOV #1280,, @#DFREG ;RS
15 MOV @#IOBUF, R0 ;FETCH DATA
16 MOV #177400, R0 ;CLEAR BITS
17 MOV R0, @ARG2(R5) ;RTN DATA
18 RTS PC ;
19 MOV #4480,, @#DFREG ;WCR : RS(1)
20 MOV #1280,, @#DFREG ;RS
21 MOV @#IOBUF, R0 ;FETCH DATA
22 BIC #177400, R0 ;CLEAR BITS
23 MOV R0, @ARG2(R5) ;RTN DATA
24 RTS PC ;
25 STATUS: MOV @#DFREG ;STATUS BYTE 1
26 MOV #1280,, @#DFREG ;RS
27 MOV @#IOBUF, R1 ;FETCH DATA
28 BIC #177400, R1 ;BUFFER FULL
29 BIT #BIT03, R1 ;
30 BEQ 3$ ;
31 MOV #1, @ARG1(R5) ;LOAD RETURN
32 RTS PC ;
33 MOV #0, @ARG1(R5) ;LOAD RETURN
34 MOV #000002, @ARG1(R5) ;
35 MOV #000000, @ARG1(R5) ;
36

```

```

1 *****
2 ***** PROCEDURE WRITE TOKEN (WTKEN) *****
3 *****
4 *****
5
6
7 001502 012737 010600 172416 WTKEN: MOV #4480., @DFREG
8 001510 012737 002400 172416 MOV #1280., @DFREG
9 001516 013701 172416 MOV @I0BUF, R1
10 001522 132701 000010 BITB #BIT03, R1
11 001526 001414 BEQ WTKG
12 001530 012737 010420 172416 MOV #4368., @DFREG
13 001536 012737 004423 172416 MOV #2323., @DFREG
14 001544 105737 172414 TSTR @CSR
15 001550 100375 BPL -4
16 001552 105037 172414 CLRB @CSR
17 001556 000207 RTS PC
18 001560 012737 010740 172416 WTKG: MOV #4576., @DFREG
19 001566 012737 001660' 172410 MOV #WTD, @BAR
20 001574 012702 000002 MOV #2, R2
21 001600 005402 NEG R2
22 001602 010237 172412 MOV R2, @WCR
23 001606 012737 024000 172416 MOV #10240., @DFREG
24 001614 000240 NDP
25 001616 105737 172414 TSTR @CSR
26 001622 000240 NDP
27 001624 105037 172414 CLRB @CSR
28 001630 012737 010420 172416 MOV #4368., @DFREG
29 001636 012737 004422 172416 MOV #2322., @DFREG
30 001644 105737 172414 TSTR @CSR
31 001650 100375 BPL -4
32 001652 105037 172414 CLRB @CSR
33 001656 000207 RTS PC
34 001660 000 377 WTD: .BYTE 0,377
35
36
37 *****
38 ***** PROCEDURE TIMER (TIME) *****
39 *****
40 *****
41 001662 005267 001006' TIME: INC LTIME
42 001666 022767 077777 001006' CMP #077777, LTIME
43 001674 001002 BNE 4$
44 001676 005067 001006' CLR LTIME
45 001702 000002 RTI
46
47
48

```

```

;READ STATUS(1)
;RS
;FETCH DATA
;BUFFER LOADED
;NO GENERATE TOKEN
;MODSTAT
;MARK OBUF/OBIF/WTCMD
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT

;SELECT BUFFER 1
;ADDRESS OF WRITE TOKEN
;# OF BYTES
;FORMAT IT
;DMA BYTE COUNT
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF HERE
;CLEAR DONE
;MODSTAT
;MARK OBIF/WTCMD
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;RETURN
;A WTKEN

```

;CALL WTKEN -LOADS AND WRITES A TOKEN TO LOOP

```

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

```

```
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 *****  
49 *****
```

14	001704	017500	000002	MOV	@R0(R5), R0	RAM:			
15	001710	022700	000000	COM	R0, R0				
16	001714	001035		BNE	RDRAM				
17	001716	017500	000004	MOV	@R0(R5), R0	WTRAM:			
18	001722	012501	000006	MOV	@R0(R5), R1				
19	001725	012737	010402	MOV	#4354., @OPREG		172416		
20	001734	062700	004400	ADD	#2304., R0				
21	001740	010037	172416	MOV	R0, @OPREG				
22	001744	105737	172414	TSTB	@CSR				
23	001750	100375		BPL	--4				
24	001752	105037	172414	CLRB	@CSR				
25	001756	012737	010401	MOV	#4353., @OPREG		172416		
26	001764	062701	004400	ADD	#2304., R1				
27	001770	010137	172416	MOV	R1, @OPREG				
28	001774	105737	172414	TSTB	@CSR				
29	002000	100375		BPL	--4				
30	002002	105037	172414	CLRB	@CSR				
31	002005	000207		RTS	PC	RDRAM:			
32	002010	017500	000004	MOV	@R0(R5), R0				
33	002014	012737	010402	MOV	#4354., @OPREG		172416		
34	002022	062700	004400	ADD	#2304., R0				
35	002026	010037	172416	MOV	R0, @OPREG				
36	002032	105737	172414	TSTB	@CSR				
37	002036	100375		BPL	--4				
38	002040	105037	172414	CLRB	@CSR				
39	002044	012737	010401	MOV	#4353., @OPREG		172416		
40	002052	012737	001400	MOV	#768., @OPREG		172416		
41	002060	105737	172414	TSTB	@CSR				
42	002064	100375		BPL	--4				
43	002068	105037	172414	CLRB	@CSR				
44	002072	013700	172416	MOV	@IOBUF, R0				
45	002076	042700	177760	BIC	#17760, R0				
46	002102	010075	000006	MOV	R0, @R0(R5)				
47	002106	000207		RTS	PC				

- RAM
- WHICH OPERATION
- READ OP
- ADDRESS
- WRITE DATA
- SEL LDATA
- WD/DA (ADDR)
- WRITE DATA
- VALID WRITE
- NO LOOP UNTIL READY
- CLEAR DONE BIT
- SEL AGRAM
- WD/DA (CMD)
- WRITE
- VALID WRITE
- NO LOOP UNTIL READY
- CLEAR DONE BIT
- ADDRESS TO READ
- SEL LDATA
- WD/DA (ADDR)
- WRITE DATA
- VALID WRITE
- NO LOOP UNTIL READY
- CLEAR DONE BIT
- SEL AGRAM
- READ DATA
- VALID READ
- NO LOOP UNTIL READY
- CLEAR DONE BIT
- FETCH DATA
- CLEAR BITS
- RTN DATA

```

1 1          .ENABLE LSB
2 2          *****
3 3          ***** PROCEDURE WRITE LOOP (LPOUT) *****
4 4          *****
5 5          LPOUT:  MOV  @ARG1(R5), R2
6 6                MOV  #340, R0
7 7                MTF5
8 8                MOV  @OUTBF, @#BAR
9 9                NEG  R2
10 10             MOV  R2, @#WCR
11 11             MOV  #4448., @#DFREG
12 12             MOV  #10240., @#DFREG
13 13             NOP
14 14             TSTB @#CSR
15 15             NOP
16 16             CLRB @#CSR
17 17             MOV  #4576., @#DFREG
18 18             MOV  @WID, @#BAR
19 19             MOV  #2, R2
20 20             NEG  R2
21 21             MOV  R2, @#WCR
22 22             MOV  #10240., @#DFREG
23 23             NOP
24 24             TSTB @#CSR
25 25             NOP
26 26             CLRB @#CSR
27 27             MOV  #4368., @#DFREG
28 28             MOV  #2307., @#DFREG
29 29             TSTB @#CSR
30 30             BPL  -4
31 31             CLRB @#CSR
32 32             MOV  #000, R0
33 33             MTF5
34 34             RTS
35 35
36 36
37 37
38 38

```

```

;BYTE COUNT
;PRI=7
;BUS ADDRESS
; 2'S COMP COUNT
;COUNT
;080 COMMAND
;DMA GO
;INTERFACE TIME
;DMA OK
;ERROR IF HERE
;081 COMMAND
;ADDRESS OF WRITE TOKEN
;BYTE COUNT
;FORMAT FOR BLUI
;DMA BYTE COUNT
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF NOT
;CLEAR DONE
;MODSTAT
;BUFFERS FULL
;GOOD WRITE
;NO LOOP UNTIL
;CLEAR DONE BIT
;PRI=0
;LOWER CPU

```

```

1 *****
2 *****
3 *****
4 *****
5 002274 017500 000002
6 002300 022700 000000
7 002304 001004
8 002306 012700 010410
9 002312 000167 000046
10 002316 022700 000001
11 002322 001004
12 002324 012700 010610
13 002330 000167 000030
14 002334 022700 000002
15 002340 001004
16 002342 012700 010510
17 002346 000167 000012
18 002352 022700 000004
19 002356 001020
20 002360 012700 010710
21 002364 010037 172416
22 002370 012737 001400 172416
23 002376 105737 172414
24 002402 100375
25 002404 013700 172416
26 002410 042700 177400
27 002414 010075 000004
28 002420 000207

RDPNT: MOV @ARG1(R5), R0
        CMP $0, R0
        BNE 1$
        MOV $4360., R0
        JMP 4$
        CMP $1, R0
        BNE 2$
        MOV $4488., R0
        JMP 4$
        CMP $2, R0
        BNE 3$
        MOV $4424., R0
        JMP 4$
        CMP $4, R0
        BNE 5$
        MOV $4552., R0
        MOV R0, @#0FREG
        TSTR @#CSR
        BFL -4
        MOV @#IOBUF, R0
        BIC $177400, R0
        MOV R0, @ARG2(R5)
        RTS PC

;*****
;*** PROCEDURE READ BUFFER POINTER ***
;*****
;FETCH COMMAND
;INBUFO ?
;NO
;RDBUFADR INO
;INBUFI ?
;NO
;RDBUFADR IN1
;OUTBUFO ?
;NO
;RDBUFADR OUT0
;OUTBUFI ?
;NO RETURN
;RDBUFADR OUT1
;RDBUFADR
;RD
;READY
;NO LOOP UNTIL
;FETCH POINTER
;CLEAR MST
;RETURN POINTER

```

FDM,MACRO  
NODE21

MACRO V03.01 4-JUN-79 17:09:17 PAGE 10

```
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 002422 017500 000002 RSTART: MOV @R0(R5), R0 ;MODE  
10 002426 022700 000000 RSO: CMP #0, R0 ;LOAD ADDRESS  
11 002432 001002 BNE RSI  
12 002434 000000 HALT  
13 002436 000207 RTS FC ;RESTART PROGRAM  
14 002440 022700 000001 RSI: CMP #1, R0 ;START ADDRESS  
15 002444 001003 BNE RS2 ;LOAD MODE  
16 002446 013700 000040 G4$: MOV @#40, R0 ;BOOT ADDRESS  
17 002452 000110 JMP (R0)  
18 002454 022700 000002 RS2: CMP #2, R0  
19 002460 001003 BNE RST ;R0M, R0  
20 002462 013700 002472' MOV @R0M, R0  
21 002466 000110 JMP (R0)  
22 002470 000207 RST: RTS FC  
23 002472 173000 ROM: .WORD 173000  
24  
25  
26
```

```

1 1 .ENABLE LSR
2 2 *****
3 3 ***** PROCEDURE GATE WAY INTERFACE *****
4 4 *****
5 5 *****
6 6
7 7 002474 010046      MOV      R0,      -(SF)
8 8 002476 010146      MOV      R1,      -(SP)
9 9 002500 010246      MOV      R2,      -(SP)
10 10 002502 042737 000100 177560      BIC      #100,  @#HRCSR
11 11 002510 005000      CLR      R0
12 12 002512 012701 000400'      MOV      #XINBF, R1
13 13
14 14 002516 105737 177560      CLR      @#HRCSR
15 15 002522 100375      BPL      .-4
16 16 002524 113702 177562      @#HRBUF, R2
17 17 002530 042702 177400      BIC      #17400, R2
18 18
19 19 002534 110221      XOR      R2,FXOR
20 20 002536 005200      MOV      R2,(R1)+
21 21 002540 122702 000003      INC      R0
22 22 002544 001364      CMP      #003,  R2
23 23 002546 022700 000006      BNE      2$,   R0
24 24 002552 003361      CMP      #6,   R0
25 25      RGT      2$,   R0
26 26      TSTB   @#HRCSR
27 27      BPL      .-4
28 28      MOV      @#HRBUF, R2
29 29      BIC      #17400, R2
30 30      CMFB   R2,FXOR
31 31      BNE      4$,   R0
32 32      TSTB   @#HXCSR
33 33      BPL      .-4
34 34 002554 010067 002000      MOV      #006,@#HXBUF
35 35      MOV      R0,  IOFLG
36 36      BR
37 37      TSTB   @#HXCSR
38 38      BPL      .-4
39 39 002560 052737 000100 177560      MOV      #100, @#HRCSR
40 40 002566 012602      (SP)+, R2
41 41 002570 012601      MOV      (SP)+, R1
42 42 002572 012600      MOV      (SP)+, R0
43 43 002574 000002      RTI
44 44
45 45
46 46
47 47
48 48 002576 012700 000340      SEND OVER GATE WAY
49 49 002602 106400      MOV      #340,  R0
50 50 002604 017500      MTPS   R0
51 51 002610 012701 000002'      MOV      @ARG1(R5),R0
52 52 002614 042737 000100 177560      MOV      #XOUTBF, R1
53 53 002622 105737 177564      BIC      #100,  @#HRCSR
54 54 002626 100375      TSTB   @#HXCSR
55 55 002630 012737 000005 177566      BPL      .-4
56 56 002636 105737 177560      MOV      #005,  @#HXBUF
57 57 002642 100375      TSTB   @#HRCSR
58 58      BPL      .-4

```

```

;SAVE REGISTERS
;CLEAR INTERRUPTS
;BYTE COUNTER
;BUFFER AREA
;OR BOX
;BYTE READY
;LOOP UNTIL
;FETCH A BYTE
;CLEAR MST
;XOR AND SAVE
;SAVE BYTE
;AND COUNT
;THE END (EOP)
;NO FETCH A BYTE

;READY TO RECEIVE B/C
;WAIT UNTIL
;FETCH IT
;CLEAR MST
;BLOCK CHECK
;NO NAK
;YES ACK
;LOOP TILL WE SEND
;SEND A ACK
;PASS BYTE COUNT
;EXIT
;INT. READY
;LOOP UNTIL
;SEND A NAK

;RESTORE REGISTERS

;RAISE CPU PRI
;SEND X BYTES
;BUFFER

```

FDM.MACRO MACRO V03.01 4-JUN-79 17:09:17 PAGE 11-1

NODE21

```

58 002644 113702 177562 @#HXBUFF, R2
59 002650 005002 CLR R2
60 002652 112102 @#HXBUFF, R2
61 002654 042702 177400 BIC (R1)+, R2
62 002660 105737 177564 TSTR @#HXCSR
63 002664 100375 BFL -4
64 002666 110237 177566 @#HXBUFF
65 002672 000240 MOV R2, @#HXBUFF
66 002674 000240 NOP
67 002676 077013 SOB R0, 6$
68 002700 012700 MOV $000, R0
69 002704 106400 MTPS R0
70 002706 052737 000100 177560 BIS $100, @#HRCR
71 002714 000207 RTS PC
72 002716 000000 FXOR: .WORD 0
73 002720 DUMP: .BLKW 50.
124

```

!FETCH BYTE

!READY TO SEND  
!LOOP UNTIL  
!SEND THE BYTE

!SEND ALL BYTES  
!LOWER PRI

!ALL DONE

FDM,MACRO  
NODE21

MACRO V03.01 4-JUN-79 17:09:17 PAGE 12

```
1 *****
2 *****
3 ***** COMMON DATA AREA *****
4 *****
5 000000 .FSECT DFM,RW,D,GBL,REL,OVR
6
7 000000 OUTBF: .BLKB 256.
8 000400 INBF: .BLKB 256.
9 001000 IWRTTM: .BLKW
10 001002 IWRT: .BLKW
11 001004 STAT: .BLKW
12 001006 LTIME: .BLKW
13 .EVEN
14
15 000000 .FSECT BUFS,RW,D,GBL,REL,OVR
16
17 000000 XOUTBF: .BLKB 256.
18 000400 XINBF: .BLKB 256.
19 001000 ACKSEQ: .BLKB 256.
20 001400 LIDFD: .BLKB 256.
21 002000 IOFLG: .BLKW
22 002002 LFLG: .BLKW
23 002004 IRSEND: .BLKW
24 .EVEN
25
26 000001 .END
27
```

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

ACKSEQ	001000R	003	B1\$	000730R	E0\$	001234R	LFINPT=	***** G	RTI\$	001006R
AREA	001214R		B2\$	000516R	E1\$	001244R	LPOUT	002110RG	STAT	001004R
ARG1	= 000002		B3\$	000542R	G10	002474RG	LTIME	001006RG	002	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	NH0ST	= 000000	TIME	001662RG
BIT00	= 000001		CAUSE	001212R	HXRUF	= 177566	OPREG	= 172416	WCR	= 172412
BIT01	= 000002		CRCK	001042R	HXRUF	= 177566	OUTBF	000000R	002	WTD
BIT02	= 000004		CSR	= 172414	INBF	000400R	002	PXOR	002716R	WTKG
BIT03	= 000010		DATA	001220R	IOBUF	= 172416	KAM	001704RG	WTKEN	001560R
BIT04	= 000020		DESTR	001264RG	IOFLG	002000R	003	RDFNT	WTRAM	001502RG
BIT05	= 000040		DH0ST	= 000001	IRSEND	002004R	003	RDRAM	XCIO	= 000001
BIT06	= 000100		DMADK	001166R	IURST	001002R	002	ROM	XG10	= 000000
BIT07	= 000200		DUMP	002720R	IWRTH	001000R	002	RST	XINBF	000400R
BIT08	= 000400		EMBF	001032R	LIDFD	001400R	003	RSTART	XOUTBF	000000R
BIT14	= 040000		ENABLE	001232RG	L10	000436RG	RS0	002422RG	XS10	= 000001
BIT15	= 100000		END\$	001210R	LIJINT	000104RG	RS1	002440R	ZER0BF	000312R
B0\$	000652R		ENSTR	001254RG	LLFLG	002002R	003	RS2		

. 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:FDH

PAGE 001

12:00:00

01-JUL-79

COMP21.COM

PROGRAM:

DECRYPT

FOR NOHAL/NOLENE  
FOR NOHAL1/NOLENE  
MAC FIM

PAGE 001

12:00:00

01-JUL-79

LNK21.COM

PROGRAM:

decipher

ASSIGN DX0: DN:  
 R LINK  
 DX1: NDBE21.SAV,DX1:MAP=DX1:NODAL/C/I/W  
 DX1: FDM/C  
 DX1: NDBH1//  
 \$SIMRT

ASSTGN 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

<u>Parameter</u>	<u>RH</u>	<u>RL</u>	<u>AH</u>	<u>AL</u>
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

<u>Parameter</u>	<u>Delta</u>	<u>RHT</u>	<u>RLT</u>	<u>AHT</u>	<u>ALT</u>
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.

FORTRAN IV V02.1-1 Mon 04-Jun-79 17:27:42 PAGE 001

```

0001 PROGRAM NODAL
0002 INTEGER*2 XING,XOUTQ,ACKQ,FXING,FXOUTQ,PACKQ
0003 INTEGER*2 PING,FREE,STAT,FLWCNT,OUTFCT
0004 INTEGER*2 SETPRM,RSTPRM,SETRP
0005 INTEGER*2 RSTBKP,Q1,Q2,RESNLM,OUTQ,DEQUE
0006 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INRF
0007 LOGICAL*1 IDATA
0008 LOGICAL*1 PACK,ICODE,ETX,CR,LF,MONTR,ISLID,DUM
0009 INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRITM,TIMLIM,ATIMLM
0010 REAL*4 RH,RL,AH,AL,RHT,RLT,AHT,ALTA,IRSV,VMEAS
0011 REAL*8 VTR
0012 COMMON /MESS/ MESSEQ
0013 COMMON /DFM/ OUTRF(256),INRF(256),
0014 1 IWRITM,IWRT,STAT,LTIME
0015 1 ACKQ(16),PACKQ(2),ING(16),PING(2),XOUTQ(16),FXOUTQ(2),
0016 1 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
0017 1 LIDFD(256),IOFLG,LLFLG,IRSEND
0018 COMMON /FRE/ FREE(64),IFR,IFRSZ
0019 COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
0020 COMMON /SWT/ SETPRM,RSTPRM,SETRP,RSTBKP
0021 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0022 RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0023 COMMON /VCOM/ ISWTVCH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSW,
0024 1 AL(6),AH(6),DELTA(6),ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10)
0025 CALL MASTER
0026 CALL INIT
0027 CALL ENABLE(1)
0028 CALL READY
0029 5 IF(PING(1) .LT. PING(2)) GOTO 40
0030 25 CALL ENABLE(0)
0031 Q2=DEQUE(PING,IND,1)
0032 CALL ENABLE(1)
0033 DO 30 I=1,10
0034 IDATA(I)=PACK(I+6,Q2)
0035 CONTINUE
0036 CALL USQCLP
0037 IFR=IFR+1
0038 FREE(IFR)=Q2
0039 CALL STATB0(IS)
0040 IF(IS .EQ. 1) GOTO 100
0041 IF(IDFLG .GT. 0 .AND. IRSEND .EQ. 0) CALL VSQC
0042 IF(LLFLG .GT. 0) CALL LINLOS
0043 IF(OUTFCT .EQ. 1) GOTO 100
0044 IF((IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0045 IF(IRSEND .EQ. 1) GOTO 85
0046 IF(PXING(1) .LT. PXING(2)) GOTO 100
0047 CALL ENABLE(0)
0048 Q1=DEQUE(PXING,XING,1)
0049 CALL ENABLE(1)
0050 OUTQ=Q1
0051 CALL DISTR(PACK(254,OUTQ),Q2)
0052 DO 90 I=1,Q2

```

PAGE 002

FORTRAN IV V02.1-1 Mon 04-Jun-79 17:27:42

```
0058   OUTBF(I)=PACK(I,OUTQ)
0059   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   ISEND=1
0067   ACKTM=0
0068   IWRTM=0
0069   IWRT=1
0070   INFCT=0
0071   100   OLDTM=NEWTM
0072   110   NEWTM=LTIME
0073   IF(IWRT .EQ. 0) GOTO 120
0075   IWRTM=IWRTM+(NEWTM-OLDTM)
0076   IF(IWRTM .LT. TIMLM) GOTO 120
0078   CALL WTOKEN
0079   IWRTM=0
0080   IWRT=0
0081   120   IF(ISENT .EQ. 0) GOTO 130
0083   ACKTM=ACKTM+(NEWTM-OLDTM)
0084   IF(ACKTM .LT. ATIMLM) GOTO 130
0086   CALL ENABLE(0)
0087   CALL ACKNAK(0)
0088   CALL ENABLE(1)
0089   130   CONTINUE
0090   GOTO 5
0091   END
```

PAGE 001

FORTRAN IV V02.1-1 Mon 04-Jun-79 17:28:03

```
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          /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1          RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007 IF (IFR .LT. 1) CALL INIT
0008 IGETSP=FREE(IFR)
0009 IFR=IFR-1
0010 RETURN
0011 END
0012
```

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,MONTR,ISLID,DUM
0003 INTEGER*2 XING,PXING,XOUTQ,PXOUTQ,ACKQ,PACKQ,ING,PING
0004 INTEGER*2 FLWNT,OUTQ,A(2),B(16)
0005 INTEGER*2 RESLN,OUTFCT
0006 COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),
1 ACKQ(16),PACKQ(2),ING(16),PING(2)
0007 COMMON /GLOB/ ISENT,FLWNT,IQLNTH,IQLNTH,OUTFCT,IFULL,OUTQ,
1 RESLN,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0008 10 IHEAD=A(1)
0009 IOTAIL=A(2)
0010 IF(IOTAIL .EQ. 1) GOTO 20
0011 IOTAIL=IOTAIL-1
0012 B(IOTAIL)=N
0013 A(2)=IOTAIL
0014 GOTO 999
0015 20 IF(IHEAD .GE. (IQLNTH)) GOTO 40
0016 NN=IHEAD-IOTAIL
0017 DO 30 I=1,NN+1
0018 B(IQLNTH+I-1)=B(IHEAD+I-1)
0019 A(1)=IQLNTH
0020 A(2)=IQLNTH-NN
0021 GOTO 10
0022 40 CALL INIT
0023 999 RETURN
0024 END
0025
0026

```

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,ACNO,FACNO,INQ,PIHQ
0004 INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
0006 ACNO(16),FACNO(2),INQ(16),PIHQ(2)
0007 COMMON /GLOB/ ISENT,FLWCNT,IOLNTH,IOLNTH+1,IFULL,OUTQ,
0008 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009 IHEAD=A(1) ! HEAD OF QUEUE
0010 DEQUE=B(IHEAD)
0011 IF(ID.NE.1) GOTO 999
0012 IF(IHEAD.NE.0) GOTO 10
0013 A(1)=IOLNTH
0014 A(2)=IOLNTH+1
0015 GOTO 999
0016 10 A(1)=IHEAD-1
0017 999 RETURN
0018 END
```

```

0001 SUBROUTINE ACKNAK(N)
0002 INTEGER*2 FLWCM,OUTFCT,STAT,XINQ,FXING,FXOUTQ,PXOUTQ
0003 INTEGER*2 ACKQ,PACKQ,PIHQ,FREE,OUTQ,XOUTQ
0004 INTEGER*2 I1,I2,I3,I4,I5,RESNLH
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,IWRTH
0009 REAL*8 COMT(5)
0010 EQUIVALENCE(COMT,LCOMT)
0011 DATA COMT(1)/'MESSAGE '//,COMT(2)/'NOT SENT',//,COMT(3)/' FROM '//,
1 COMT(4)/'MODE '//,COMT(5)/' TO MODE'//
0012 COMMON /GLOB/ ISENT,FLWCM,IQLNTH,OUTFCT,IFULL,OUTQ,
1 COMMON /RESNL/ IALTR,ISLID,MONTOR,LF,ETX,CR,DUM
0013 COMMON /DFM/ OUTBF(256),INBF(256),
1 IWRTH,IWRT,STAT,LTIME
0014 COMMON /ACK/ PACK(256,64)
0015 COMMON /QUE/ XINQ(16),PXING(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
1 PACKQ(2),INQ(16),PIHQ(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
0020 IFR=IFR+1
0021 DO 5 I=1,256
0022 PACK(I,OUTQ)=0
0023 CONTINUE
0024 IFULL=0
0025 ISENT=0
0026 IRSEND=0
0027 FREE(IFR)=OUTQ
0028 GOTO 999
0029 T2=PACK(256,OUTQ)
0030 I2=T2+1
0031 IF(I2.LE. RESNLH) GOTO 120
0032 T3=PACK(3,OUTQ)
0033 T4=MOD(T3,64)
0034 TS=IGETSP(N)
0035 IF(T4.LT. 32) GOTO 110
0036 IF(MESSEQ.EQ. 126) MESSEQ=0
0037 MESSEQ=MESSEQ+1
0038 PACK(1,I5)=0
0039 PACK(2,I5)=MESSEQ
0040 PACK(3,I5)=0
0041 PACK(4,I5)=0
0042 PACK(5,I5)=25
0043 PACK(6,I5)=ISLID
0044 DO 20 I=7,60
0045 PACK(I,I5)=040
0046 CONTINUE
0047 DO 30 I=7,9
0048 PACK(I,I5)=LF
0049 CONTINUE
0050 DO 30
0051 CONTINUE
0052 CONTINUE
0053 CONTINUE

```

EDM.N  
SYMBCC  
ACKSE  
AREA  
ARG2  
ARG3  
ARG4  
BAR  
BIT00  
BIT01  
BIT02  
BIT03  
BIT04  
BIT05  
BIT06  
BIT07  
BIT08  
BIT14  
BIT15  
B0\$  
. ABS  
DFM  
BUFS  
ERROR  
VIRTU  
DYNAM  
,DKIF

```

0054 DO 40 I=1,8
0055   PACK(I+9,T5)=LCOMT(I)
0056 CONTINUE
0057 DO 50 I=9,16
0058   PACK(I+9,T5)=LCOMT(I)
0059 CONTINUE
0060 DO 60 I=17,22
0061   PACK(I+9,T5)=LCOMT(I)
0062 CONTINUE
0063 DO 70 I=25,29
0064   PACK(I+7,T5)=LCOMT(I)
0065 CONTINUE
0066 ENCODE(3,80,PACK(38,T5))ISLID
0067 FORNAT(13)
0068 DO 90 I=33,40
0069   PACK(I+11,T5)=LCOMT(I)
0070 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(PXING,XING,T5)
0078 DO 102 I=1,256
0079   PACK(I,OUTQ)=0
0080 CONTINUE
0081 ISENT=0
0082 IRSEND=0
0083 IFULL=0
0084 IWR=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
0093 CALL ENQUE(PXING,XING,OUTQ)
0094 GOTO 999
0095 120 PACK(256,OUTQ)=T2
0096   IFULL=0
0097 ISENT=0
0098 IF(IRSEND.EQ.1) GOTO 999
0099 CALL ENQUE(PXING,XING,OUTQ)
0100 RETURN
0101 999
0102 END
0103

```

```

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

0001      SUBROUTINE INPTO(L)
0002      INTEGER*2 STAT,XING,PXING,XOUTG,XOUTQ,PXOUTQ,ACKQ
0003      INTEGER*2 PACKG,PING,FLWCNT,OUTFCT,T1,RESLHM,OUTG
0004      LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005      LOGICAL*1 PACK,MONTR,ISLID,ETX,CR,LF,DUM
0006      INTEGER*2 LTIME,IMRITM
0007      COMMON /DFM/ OUTBF(256),INBF(256),
0008      COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
0009      LIDFD(256),IOFLG,LLFLG,IRSEND
0010      COMMON /QUE/ XING(16),PXING(2),XOUTG(16),PXOUTG(2),
0011      ACKQ(16),PACKG(2),ING(16),PING(2)
0012      COMMON /GLOB/ ISENT,FLWCNT,IOLNTH,OUTFCT,IFULL,OUTG,
0013      RESLHM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0014      IF(L.LT. 1) GOTO 30
0015      LI=L-2
0016      II=INBF(6)
0017      IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0018      T1=IGETSP(N)
0019      DO 10 I=1,LI
0020      PACK(I,T1)=INBF(I)
0021      CALL ENSTR(PACK(254,T1),LI)
0022      CALL ENQUE(PING,ING,T1)
0023      N=2
0024      GOTO 40
0025      N=1
0026      GOTO 40
0027      N=0
0028      DO 50 I=1,2
0029      OUTBF(I)=INBF(I)
0030      CONTINUE
0031      NN=1
0032      IF(N .EQ. 0) NN=128
0033      OUTBF(3)=NN
0034      OUTBF(4)=0
0035      OUTBF(5)=INBF(6)
0036      OUTBF(6)=ISLID
0037      OUTBF(7)=ETX
0038      OUTBF(8)=0
0039      OUTBF(9)=LIDFD(INBF(6))
0040      CALL LPOUT(9)
0041      IF(N .NE. 2) GOTO 999
0042      ACKSEQ(II)=INBF(2)
0043      RETURN
0044      999      END
0045      999      END
0046      END

```

```

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

0001      SUBROUTINE LPIPT(LI)
0002      INTEGER*2 STAT,FLWCNT,OUTFCT,OUTG,RESNLM
0003      INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004      LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTR,ISLID,DUM
0005      INTEGER*2 LTIME,IWRTH
0006      COMMON/DFM/OUTBF(256),INBF(256),IWRTH,IWRT,STAT,LTIME
0007      COMMON /GLOB/ ISENT,FLWCNT,IGLNTH,OUTFCT,IFULL,OUTG,
1          RESNLM,IALTRT,ISLID,MONTR,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      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

```

```

FORTRAN IV      V02.1-1      Mon 04-Jun-79 17:29:42      PAGE 001

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,FINQ,FREE
0005      INTEGER*2 STAT,FLWCNT,OUTFCT
0006      INTEGER*2 SETPRM,RSTPRM,SETRKP,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,IWRITM
0010      COMMON /DFM/ OUTBF(256),INBF(256),
1          IWRITM,IMRT,STAT,LTIME
0011      COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
1          ACKQ(16),PACKQ(2),INQ(16),FINQ(2)
0012      COMMON /MESS/ MESSEQ
0013      COMMON /RUF/ 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 /SMT/ SETPRM,RSTPRM,SETRKP,OUTQ,RESNLM,RSTBKP
0018      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1          RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0019      COMMON /UCOM/ ISWTRCH,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      IWRITM=0
0038      TIMLIM=50
0039      SETPRM=4
0040      RSTPRM=32
0041      SETRKP=8
0042      RSTBKP=64
0043      ATIMLM=200
0044      MONTOR=27
0045      IMRT=0
0046      ISWTRCH=0
0047      OUTFCT=0
0048      IFULL=0
0049      ISENT=0

```

PAGE 002

Mon 04-Jun-79 17:29:42

FORTRAN IV V02.1-1

```

0050      DO 40 I=1,20
0051      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      LIDFD(I)=0
0062      DO 60 I=40,44
0063      LIDFD(I)=1
0064      DO 70 I=45,59
0065      LIDFD(I)=0
0066      DO 80 I=60,64
0067      LIDFD(I)=2
0068      DO 90 I=65,79
0069      LIDFD(I)=0
0070      DO 100 I=80,84
0071      LIDFD(I)=4
0072      DO 110 I=85,99
0073      LIDFD(I)=0
0074      DO 120 I=100,104
0075      LIDFD(I)=5
0076      DO 130 I=105,256
0077      LIDFD(I)=0
0078      PING(1)=IQLNTH
0079      PING(2)=IQLNTH+1
0080      PACKQ(1)=IQLNTH
0081      PACKQ(2)=IQLNTH+1
0082      PXOUTQ(1)=IQLNTH
0083      PXOUTQ(2)=IQLNTH+1
0084      PING(1)=IQLNTH
0085      PING(2)=IQLNTH+1
0086      DO 10 I=1,IFRSZ
0087      PACK(255,I)=0
0088      PACK(256,I)=0
0089      DO 20 I=1,IFRSZ
0090      FREE(I)=1
0091      DO 30 I=1,256
0092      ACKSEQ(I)=256
0093      IFR=IFRSZ
0094      RETURN
0095      END

```

```

FORTRAN IV      V02.1-1      Mon 04-Jun-79 17:30:04      PAGE 001

0001      SUBROUTINE LINLOS
0002      REAL*8 RM1(5),LIN08,LIN18
0003      INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004      INTEGER*2 XING,PXING,XOUTG,PXOUTG,ACKG,PACKG,PING
0005      LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006      LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0007      LOGICAL*1 PACK,M1(40),LINO(8),LIN1(8)
0008      COMMON/QUE/XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),
      &      ACKQ(16),PACKQ(2),INQ(16),PING(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,MONTR,LF,ETX,CR,DUM
0011      COMMON /MESS/ MESSEQ
0012      COMMON /MESS/ PACK(256,64)
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(LINO,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
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)=ISLID
0032      DO 20 I=7,9
0033          PACK(I,T1)=LF
0034      CONTINUE
0035      DO 30 I=1,22
0036          PACK(I+9,T1)=M1(I)
0037      IF(IS.EQ.1) GOTO 50
0038      DO 40 I=1,8
0039          PACK(I+31,T1)=LINO(I)
0040      GOTO 70
0041      DO 60 I=1,8
0042          PACK(I+31,T1)=LIN1(I)
0043      DO 80 I=23,40
0044          PACK(I+17,T1)=M1(I)
0045      PACK(58,T1)=CR
0046      PACK(59,T1)=LF
0047      PACK(60,T1)=ETX
0048      CALL ENSTR(PACK(254,T1),60)
0049      CALL ENABLE(0)
0050      CALL ENQUE(PXING,XING,T1)
0051      CALL ENABLE(1)
0052
0053

```

PAGE 002

Mon 04-Jun-79 17:30:04

F0RTRAN IV V02.1-1

0054 LLFLG=0  
0055 RETURN  
0056 END



```

FORTRAN IV      V02.1-1      Mon 04-Jun-79 18:11:26      PAGE 001

0001      SUBROUTINE VSQC
0002      INTEGER*2 XING,XOUTQ,ACKQ,PXING,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,MONTR,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),
0011      1 IOFLG,LLFLG,IRSEND
0012      COMMON /VCOM/ ISWICH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSW,
0013      1 AL(6),AH(6),DELTA(6),ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),
0014      2 ICHAN
0015      COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
0016      1 PACKQ(2),ING(16),PING(2)
0017      COMMON /GLOB/ ISENT,FLWCNT,IOLNTH,OUTFCT,IFULL,OUTQ,RESNLM,
0018      1 IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0019      DATA RH(1),RH(2),RH(3),RH(4) /99.5,99.5,199.0,199.0/
0020      DATA RH(5),RH(6) /59.1,299.0/
0021      DATA RL(1),RL(2),RL(3),RL(4) /-90.5,-90.5,-270.0,-275.0/
0022      DATA RL(5),RL(6) /-118.9,-270.0/
0023      DATA AH(1),AH(2),AH(3),AH(4) /90.0,90.0,192.0,191.5/
0024      DATA AH(5),AH(6) /56.5,199.0/
0025      DATA AL(1),AL(2),AL(3),AL(4) /-50.0,-55.0,-251.0,-257.0/
0026      DATA AL(5),AL(6) /-113.5,-252.0/
0027      DATA DELTA(1),DELTA(2),DELTA(3) /5.5,5.3,0/
0028      DATA DELTA(4),DELTA(5),DELTA(6) /2.0,1.0,2.0/
0029      DECODE(2,20,XINBF(1),ERR=410) ICNT
0030      DECODE(4,30,XINBF(3),ERR=410) VMEAS(1)
0031      DECODE(4,30,XINBF(7),ERR=410) VMEAS(2)
0032      DECODE(4,30,XINBF(11),ERR=410) VMEAS(3)
0033      DECODE(4,30,XINBF(15),ERR=410) VMEAS(4)
0034      DECODE(4,30,XINBF(19),ERR=410) VMEAS(5)
0035      DECODE(4,30,XINBF(23),ERR=410) VMEAS(6)
0036      DECODE(2,20,XINBF(27),ERR=410) IVC
0037      DECODE(2,20,XINBF(37),ERR=410) MON
0038      FORMAT(A2)
0039      20
0040      30
0041      IF(ICHNSW .EQ. 1) GOTO 400
0042      DO 10 I=1,6
0043      10 IRSV(I)=(AH(I)+AL(I)) / 2
0044      CONTINUE
0045      DO 350 ICTR=1,6
0046      ALT=AL(ICTR)+DELTA(ICTR)
0047      AHT=AH(ICTR)+DELTA(ICTR)
0048      RLT=RL(ICTR)+DELTA(ICTR)
0049      RHT=RH(ICTR)+DELTA(ICTR)
0050      IF(VMEAS(1) .GE. ALT .AND.
0051      1 VMEAS(1) .LE. AHT) GOTO 350
0052      IF((VMEAS(1) .GT. AL(1) .AND.
0053      1 VMEAS(1) .LT. ALT) .OR.
0054      2 (VMEAS(1) .GT. AHT .AND.
0055      3 VMEAS(1) .LT. AH(1))) 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(ISWITCH .EQ. 1) GOTO 45
IOFLG=0
RETURN
0056 CALL ENABLE(0)
0057 K1=IGETSP(N)
0058 CALL ENABLE(1)
0059 DO 46 I=1,256
0060 PACK(I,K1)=0
0061 CONTINUE
0062 IF(MESSEQ .EQ. 126) MESSEQ=0
0063 MESSEQ=MESSEQ+1
0064 PACK(1,K1)=0
0065 PACK(2,K1)=MESSEQ
0066 PACK(3,K1)=0
0067 PACK(4,K1)=0
0068 PACK(5,K1)=MONTOR
0069 PACK(6,K1)=ISLID
0070 DO 50 I=7,14
0071 PACK(I,K1)=XINBF(I+22)
0072 CONTINUE
0073 DO 60 I=15,16
0074 PACK(I,K1)=XINBF(I+12)
0075 CONTINUE
0076 DO 70 I=17,24
0077 PACK(I,K1)=ITERM
0078 ENCODE(4,65,PACK(18,K1)) VMEAS(ICTR)
0079 FORMAT(A4)
0080 ENCODE(2,70,PACK(22,K1)) MON
0081 FORMAT(A2)
0082 PACK(24,K1)=ITERM
0083 CALL ENSTR(PACK(254,K1),24)
0084 CALL ENABLE(0)
0085 CALL ENQUE(PXINQ,XINQ,K1)
0086 CALL ENABLE(1)
0087 GOTO 410
0088 IF(ISWITCH .EQ. 1) GOTO 110
0089 GOTO 410
0090 CALL TREND(2,VMEAS(ICTR))
0091 GOTO 410
0092 IF(ISWITCH .EQ. 1) GOTO 210
0093 GOTO 410
0094 CALL TREND(2,VMEAS(ICTR))
0095 GOTO 410
0096 IF(ISWITCH .EQ. 1) GOTO 310
0097 GOTO 410
0098 CALL TREND(2,VMEAS(ICTR))
0099 GOTO 410
0100 IF(ISWITCH .EQ. 1) GOTO 310
0101 GOTO 410
0102 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      CONTINUE
0112 405 IF(MESSED .EQ. 126) MESSED=0
0113      MESSED=MESSED+1
0114          PACK(1,K1)=0
0115          PACK(2,K1)=MESSED
0116          PACK(3,K1)=0
0117          PACK(4,K1)=0
0118          PACK(5,K1)=25
0119          PACK(6,K1)=ISLID
0120          PACK(7,K1)=CR
0121          PACK(8,K1)=LF
0122          ENCODE(4,360,PACK(9,K1)) ICHAN
0123          PACK(13,K1)='040
0124          ENCODE(12,370,PACK(14,K1)) VMEAS(1)
0125          PACK(26,K1)='040
0126          ENCODE(12,370,PACK(27,K1)) VMEAS(2)
0127          PACK(39,K1)='040
0128          ENCODE(12,370,PACK(40,K1)) VMEAS(3)
0129          PACK(52,K1)='040
0130          ENCODE(12,370,PACK(53,K1)) VMEAS(4)
0131          PACK(65,K1)='040
0132          ENCODE(12,370,PACK(66,K1)) VMEAS(5)
0133          PACK(78,K1)='040
0134          ENCODE(12,370,PACK(79,K1)) VMEAS(6)
0135          PACK(91,K1)=CR
0136          PACK(92,K1)=LF
0137          PACK(93,K1)=ETX
0138          CALL ENSTR(PACK(254,K1),93)
0139          CALL ENABLE(0)
0140          CALL ENQUE(PXIND,XIND,K1)
0141          CALL ENABLE(1)
0142          ICHNSW=0
0143          FORMAT(14)
0144          360 FORMAT(F12.6)
0145          370 IOFLG=0
0146          410 RETURN
0147          END
0148

```

FORTRAN  
 0038  
 0059  
 0060  
 0061  
 0062  
 0063  
 0064  
 0065  
 0066  
 0067  
 0068  
 0069  
 0070  
 0071  
 0072  
 0073  
 0075  
 0076  
 0078  
 0079  
 0080  
 0081  
 0083  
 0084  
 0086  
 0087  
 0088  
 0089  
 0090  
 0091

```

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,MONITOR,LF,ETX,CR,DUM,INQUE,IQDATA
0005 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006 INTEGER*2 XING,PXING,XOUTQ,F,XOUTQ,ACKQ,PACKQ,PING
0007 INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM
0008 COMMON PACK(256,64)
0009 COMMON /MESS/ MESSEQ
0010 COMMON /VCOM/ ISWITCH,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/ XING(16),PXING(2),XOUTR(16),FXOUTQ(2),ACKQ(16),
1 PACKQ(2),INQ(16),PING(2)
0012 COMMON /GLOB/ ISENT,FLWCNT,IOLNTH,OUTFCT,IFULL,OUTQ,RESNLM,
1 IALTRT,ISLID,MONITOR,LF,ETX,CR,DUM
0013 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
1 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 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)=MONITOR
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 FORMAT(A4)
0055 PACK(17,K1)=ICND

```

FORTRAN 1  
 0001  
 0002  
 0003  
 0004  
 0005  
 0006  
 0007  
 0009  
 0010  
 0011  
 0012

FORTRAN IV  
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

PAGE 002

V02.1-1 Mon 04-Jun-79 18:11:55

FORTRAN IV

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

```

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,AL,AL,RLT,ALT,DELTA,IRSV,VMEAS
0004      REAL*8 VTR
0005      COMMON /VCOM/ ISWITCH,VMEAS(6),IRSV(6),ICTR,RL(6),RH(6),ICHNSM,
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      ISWITCH=1
0014      ITERM=IDATA(4)
0015      RETURN
0016      ISWITCH=0
0017      RETURN
0018      70      DECODE(4,75, IDATA(2),ERR=80) ICHAN
0019      75      FORMAT(I)
0020      CALL SOUT(ICHAN)
0021      ICHNSM=1
0022      RETURN
0023      STOP
0024      END

```

FORTRAN  
0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0013  
0014  
0015  
0016  
0017  
0018

```

1 .TITLE FDM.MACRO
2 .SBTTL NODE22
3 .IDENT /V3.0/
4 .GLOBL LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
5 .GLOBL WTOKEN,RAM,STATUS,STATRO
6 .GLOBL LPOUT,RSTART,RDFNT,DESTR,ENSTR,LFINFT
7 .NLIST CND
8 .PSECT
9
10 000000
11 000001
12 000002
13 000003
14 000004
15 000005
16 000006
17 000007
18 000007
19
20
21
22
23 172410
24 172412
25 172414
26 172416
27 172416
28 177560
29 177562
30 177564
31 177566
32
33
34
35
36
37
38
39
40
41
42
43 000001
44 000000
45 000001
46 000000
47 000001
48 000000
49 000001
50
51
52 000002
53 000004
54 000006
55 000010
56
57 100000
58 040000
59 000400
60 000200
61 000100
62 000040
63 000020
64 000010
65 000004

```

FORTRAN I

0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0034  
0035  
0036  
0038  
0039  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053

FIM.MACRO  
NODE22

66  
67  
68

000002  
000001

BIT01= 2  
BIT00= 1

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

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

```

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 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****

```

```

*****
**** PROCEDURE MASTER START UP
*****

MASTER:  MOV    $340, R0
          MTPS   R0
          MOV    $124, R0
          MOV    $110, (R0)+
          MOV    $340, (R0)
          RESET
          MOV    $60, R0
          MOV    $510, (R0)+
          MOV    $340, (R0)+
          MOV    $100, R0
          MOV    $TIME, (R0)
          BIC    $40100, @#CSR
          JSR    PC, LIUINT
          CLR    @#CSR
          MOV    $000, R0
          MTPS  R0
          RTS   PC

*****
**** PROCEDURE INITIALIZE
*****

          ;CLEAR ACRAM
          LIUINT: MOV    $4354, @#OPREG
          TSTB  $2304, @#OPREG
          BPL  -4
          CLRB @#CSR
          MOV    $4353, @#OPREG
          MOV    $-256, R0
          MOV    $2311, @#OPREG
          TSTB @#CSR
          BPL  -4
          CLRB @#CSR
          INC  R0
          BNE  1$

          ;CLEAR INPUT/OUTPUT BUFFERS
          MOV    $10410, R1
          MOV    $10440, R0
          JSR   PC, ZEROPF
          MOV    $1400, @#OPREG
          TSTB @#CSR
          BPL  -4
          CLRB @#CSR
          MOV    $10610, R1
          MOV    $10640, R0
          JSR   PC, ZEROPF

          ;LIUI HANDLER
          ;PRI=7
          ;RESET BUS
          ;SIG INTERFACE HANDLER
          ;CLOCK VECTOR
          ;CLOCK
          ;DISABLE INTERRUPTS
          ;CLEAR LIUI
          ;PRI=0
          ;CPU DOWN

          ;LDADR
          ;ADDRESS=0
          ;GOOD WD
          ;NO RETRY
          ;SEL ACRAM
          ;COUNTER
          ;WRITE A NULL
          ;GOOD WD
          ;NO LOOP UNTIL READY
          ;COUNT+1

          ;DRUFADR CMD
          ;SEL INBUF0
          ;SET POINTER=0
          ;FALSE READ DATA
          ;GOOD READ
          ;NO RETRY
          ;CLEAR DONE BIT
          ;DRUFADR COMMAND
          ;SEL INBUF1 COMMAND
          ;POINTER=0

```

102	000232	012737	001400	172416	MOV	#1400, @#DFREG
103	000240	105737	172414		TSTB	@#CSR
104	000244	100375			BFL	.-4
105	000246	105037	172414		CLRB	@#CSR
106	000252	012701	010510		MOV	#10510, R1
107	000256	012700	010540		MOV	#10540, R0
108	000262	004767	000016		JSR	PC, ZER0BF
109	000266	012701	010710		MOV	#10710, R1
110	000272	012700	010740		MOV	#10740, R0
111	000276	004767	000002		JSR	PC, ZER0BF
112	000302	000440			BR	STAT\$
113	000304	010137	172416		ZER0BF:	MOV
114	000310	012737	001400	172416	MOV	#1400, @#DFREG
115	000316	105737	172414		TSTB	@#CSR
116	000322	100375			BFL	.-4
117	000324	105037	172414		CLRB	@#CSR
118	000330	013702	172416		MOV	@#IDBUF, R2
119	000334	042702	177400		RIC	#177400, R2
120	000340	010037	172416		MOV	R0, @#DFREG
121	000344	022702	000000		CMP	#0, R2
122	000350	001412			BEG	3\$
123	000352	012737	001400	172416	MOV	#1400, @#DFREG
124	000360	105737	172414		TSTB	@#CSR
125	000364	100375			BFL	.-4
126	000366	105037	172414		CLRB	@#CSR
127	000372	005302			DEC	R2
128	000374	000763			BR	2\$
129	000376	010037	172416		MOV	R0, @#DFREG
130	000402	000207			RTS	PC
131						
132						
133						
134	000404	012737	010400	172416	MOV	#4352., @#DFREG
135	000412	012737	002400	172416	MOV	#1280., @#DFREG
136	000420	012737	002400	172416	MOV	#1280., @#DFREG
137	000426	000207			RTS	PC
138						
139						

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 000430 010046          MOV      R0,      -(SP)
7 000432 010146          MOV      R1,      -(SP)
8 000434 010246          MOV      R2,      -(SP)
9 000436 010346          MOV      R3,      -(SP)
10 000440 010446          MOV      R4,      -(SP)
11 000442 010546          MOV      R5,      -(SP)
12 000444 005737 172414  TST      @#CSR
13 000450 100153          BFL     RTI$
14 000452 042737          RIC      #BIT14, @#CSR
15 000460 012737 010400 172414  MOV      #4352., @#DFREG
16 000466 012737 002400 172416  MOV      #1280., @#DFREG
17 000474 013767 172416 000502  @#IORUF, CAUSE
18 000502 042767 177400 000474  RIC      #177400, CAUSE
19
20
21
22 000510 132767 000004 000466  E2$:  BITB   #BIT02, CAUSE
23 000516 001406          BEQ     B3$
24 000520 012702 010410          MOV      #4360., R2
25 000524 012703 010440          MOV      #4384., R3
26 000530 004767 000270          JSR     PC,      EMBF
27 000534 132767 000010 000442  E3$:  BITB   #BIT03, CAUSE
28 000542 001406          BEQ     B4$
29 000544 012702 010610          MOV      #4488., R2
30 000550 012703 010640          MOV      #4512., R3
31 000554 004767 000244          JSR     PC,      EMBF
32 000560 132767 000020 000416  E4$:  BITB   #BIT04, CAUSE
33 000566 001407          BEQ     B5$
34 000570 132767 000200 000406  E7$:  BITB   #BIT07, CAUSE
35 000576 001403          BEQ     B5$
36 000600 012767 000002 002002'  MOV      #2,      LFLG
37 000606 132767 000040 000370  E5$:  BITB   #BIT05, CAUSE
38 000614 001403          BEQ     B6$
39 000616 012767 000001 002002'  MOV      #1,      LFLG
40 000624 132767 000100 000352  E6$:  BITB   #BIT06, CAUSE
41 000632 001404          BEQ     B0$
42 000634 005067 001000'  CLR     IWRTH
43 000640 005067 001002'  CLR     IWRTH
44
45
46
47 000644 132767 000001 000332  E0$:  BITB   #BIT00, CAUSE
48 000652 001423          BEQ     B1$
49 000654 012701 000400'  MOV      #INBF,  R1
50 000660 012702 010410          MOV      #4360., R2
51 000664 012703 010440          MOV      #4384., R3
52 000670 012704 000001          MOV      #BIT00, R4
53 000674 004767 000124          JSR     PC,      EMBF
54 000700 012705 001206'  MOV      #AREA,  R5
55 000704 012767 000001 000274  MOV      #1,      AREA
56 000712 010267 000274          MOV      R2,      DATA
57 000716 004767 000006          JSR     PC,      LPINFT

```

```

;SAVE R0
;SAVE R1
;SAVE R2
;SAVE R3
;SAVE R4
;SAVE R5
;? LIU INTERRUPTED
;NO
;YES/DISABLE LIU
;MCR : RS(0)
;RD
;CLEAR UNUSED BITS

;OV-FL
;RDBUFADR INO
;SEL INBUFO
;GO EMPTY
;OV-FL
;RDBUFADR IN1
;SEL INBUF1
;GO EMPTY
;LINE-LOSS PRIMARY
;LATCH ON
;YES LOOP AROUND
;FLAG PRIMARY SW. FAIL
;BACKUP LINE LOSS
;FLAG BACKUP SW. FAIL
;WRITE TOKEN DETECT
;CLEAR TOKEN VAR

;INBUFO FULL
;NO
;BUFFER ADDRESS
;RDBUFADR COMM
;SEL RUF
;CRC BIT
;EMPTY BUFFER
;DATA LINK AREA
;I VARIABL
;CRC OR BYTE COUNT
;CALL FORTRAN QUE'ER

```

;PERFORM OPERATION DEPENDING ON BITS

58	000722	132767	000002	000254	E1\$:	E1TB	#BIT01, CAUSE	
59	000730	001423				BER	RTI\$	
60	000732	012701	000400'			MOV	#INBF, R1	
61	000736	012702	010610			MOV	#4488., R2	
62	000742	012703	010640			MOV	#4512., R3	
63	000746	012704	000002			MOV	#BIT01, R4	
64	000752	004767	000046			JSR	FC, EMBF	
65	000756	012705	001206'			MOV	#AREA, R5	
66	000762	012767	000001	000216		MOV	#1, AREA	
67	000770	010267	000216			MOV	R2, DATA	
68	000774	004767	000000G			JSR	PC, LFINFT	
69								
70	001000	012605			RTI\$:	MOV	(SF)+, R5	
71	001002	012604				MOV	(SP)+, R4	
72	001004	012603				MOV	(SP)+, R3	
73	001006	012602				MOV	(SF)+, R2	
74	001010	012601				MOV	(SF)+, R1	
75	001012	012600				MOV	(SF)+, R0	
76	001014	052737	040000	172414		BIS	#BIT14, @#CSR	
77	001022	000002				RTI		
78								
79	001024	012737	010600	172416	EMBF:	MOV	#4480., @#DFREG	
80	001032	012737	002400	172416		MOV	#1280., @#DFREG	
81	001040	013700	172416			MOV	@#IORUF, R0	
82	001044	130400				E1TB	R4,R0	
83	001046	001002				BNE	CRCOK	
84	001050	012704	177777			MOV	#-1, R4	
85	001054	010237	172416		CRCOK:	MOV	R2, @#DFREG	
86	001060	012737	001400	172416		MOV	#768., @#DFREG	
87	001066	105737	172414			TSTB	@#CSR	
88	001072	100375				BPL	.-4	
89	001074	013702	172416			MOV	@#IORUF, R2	
90	001100	042702	177400			BIC	#177400, R2	
91	001104	010200				MOV	R0, R0	
92	001106	005400				NEG	R0	
93	001110	010037	172412			MOV	R0, @#WCR	
94	001114	010137	172410			MOV	R1, @#BAR	
95	001120	010337	172416			MOV	R3, @#DFREG	
96	001124	012737	001400	172416		MOV	#768., @#DFREG	
97	001132	105737	172414			TSTB	@#CSR	
98	001136	100375				BPL	.-4	
99	001140	012737	021000	172416		MOV	#8704., @#DFREG	
100	001146	000240				NOP		
101	001150	105737	172414			TSTB	@#CSR	
102	001154	100401				BMI	DMAOK	
103	001156	000240				NOP		
104	001160	012737	004400	172416	DMAOK:	MOV	#2304., @#DFREG	
105	001166	105737	172414			TSTB	@#CSR	
106	001172	100375				BPL	.-4	
107	001174	005704				TST	R4	
108	001176	100001				BFL	END\$	
109	001200	010402				MOV	R4, R2	
110	001202	000207				RTS	FC	
111	001204	000000			END\$:	.WORD	0	
112	001206	000000			CAUSE:	.WORD	0	
113	001210	001212'			AREA:	.WORD	DATA	
114	001212	000000			DATA:	.WORD	0	

```

;INBUF1 FULL
;NO WERE DONE
;BUFFER ADDRESS
;RDBUFADR COMM
;SEL INBUF1
;CRC BIT
;GO EMPTY BUFFER
;DATA LINK AREA
;ONE VARIABLE
;CRC OR BYTE COUNT
;CALL FORTRAN QUE'ER
;RESTORE REGISTERS

;ENABLE INTERRUPTS
;RETURN FROM INTERRUPT

;READ STATUS 1
;RS
;FETCH STATUS
;GOOD CRC

;NO FLAG
;RDBUFADR
;RD
;GOOD RD

;REPLACE WITH POINTER
;CLEAR MST BITS
;SAVE IT
;2'S COMP
;BYTE COUNT NOW
;ADDRESS IN MEMORY
;SEL BUFFER
;FALSE RD
;DONE ON
;NO LOOP
;FIRE DMA
;DELAY
;GOOD DMA

;ERROR IF HERE
;FALSE WD
;GOOD WD

;WAS CRC OK
;YES
;NO FLAG IT
;RETURN
;STATUS BYTE 0 HOLDER
  
```

```

1  .ENABLE LSR
2  *****
3  ;*** PROCEDURE ENABLE ***** *
4  ;***
5
6  001214 017501 000002 000002 000002 000002 000002 000002 000002 000002
7  001220 022701 000001
8  001224 001404
9  001226 042737 040000 172414 000000 172414 @#CSR
10 001234 000207 RTS
11 001236 052737 040000 172414 @#CSR
12 001244 000207 RTS
13
14 ;FORTRAN ENCODE/DECODE MACRO
15
16 001246 017575 000004 000002 ENSTR: MOV @ARG2(R5),@ARG1(R5)
17 001254 000207 RTS PC
18 001256 017575 000002 000004 DESTR: MOV @ARG1(R5),@ARG2(R5)
19 001264 000207 RTS PC
20
21 *****
22 ;*** PROCEDURE LINE SWITCH ***** *
23 ;***
24 *****
25
26
27 ;CALL SWITCH(X) -SET OR RESET LINE SWITCH
28 ;
29 ;04=SET PRIMARY LINE
30 ;32=RESET PRIMARY LINE
31 ;08=SET BACKUP LINE
32 ;64=RESET BACKUP LINE
33
34 001266 017501 000002 SWITCH: MOV @ARG1(R5),R1
35 001272 012737 010420 172416 MOV #4368., @#OPREG
36 001300 062701 004400 ADD #2304., R1
37 001304 010137 172416 MOV R1, @#OPREG
38 001310 105737 172414 TSTB @#CSR
39 001314 100375 BFL .-4
40 001316 105037 CLRB @#CSR
41 001322 000207 RTS PC
42
;SW SET
;WCR ; MODSTAT
;RU
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT

```

```

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 *****

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

STATUS: MOV @ARG1(R5), R1
          CMP #1, R1
          BEQ 2$
          MOV #4352., @OPREG
          MOV #1280., @OPREG
          MOV @IOBUF, R0
          BIC #177400, R0
          MOV R0, @ARG2(R5)
          RTS
          MOV #4480., @OPREG
          MOV #1280., @OPREG
          MOV @IOBUF, R0
          BIC #177400, R0
          MOV R0, @ARG2(R5)
          RTS
          MOV #4352., @OPREG
          MOV #1280., @OPREG
          MOV @IOBUF, R1
          BIC #177400, R1
          BITB #BIT03, R1
          BEQ 3$
          MOV #1, @ARG1(R5)
          RTS
          MOV #0, @ARG1(R5)
          RTS
          MOV #0, @ARG1(R5)
          RTS
          MOV #0, @ARG1(R5)
          RTS

;WHICH BYTE
;COMPARE
;WCR : RS(0)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;WCR : RS(1)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;STATUS BYTE 1
;RS
;FETCH DATA
;BUFFER FULL
;LOAD RETURN
;LOAD RETURN

```

```
1 *****  
2 ***** PROCEDURE WRITE TOKEN (WTKEN) *****  
3 *****  
4 *****  
5 *****  
6 *****  
7 001474 012737 010600 172416 WTKEN: MOV #4480., @#DPREG  
8 001502 012737 002400 172416 MOV #1280., @#DPREG  
9 001510 013701 172416 MOV #1080UF, R1  
10 001514 132701 000010 BITB @BIT03, R1  
11 001520 001414 REQ WTKG  
12 001522 012737 010420 172416 MOV #4368., @#DPREG  
13 001530 012737 004423 172416 MOV #2323., @#DPREG  
14 001536 105737 172414 TSTB @#CSR  
15 001542 100375 BPL -4  
16 001544 105037 172414 CLR @#CSR  
17 001550 000207 RTS FC  
18 001552 012737 010740 172416 WTKG: MOV #4576., @#DPREG  
19 001560 012737 001652 172410 MOV #WTD, @#BAR  
20 001566 012737 000002 MOV #2, R2  
21 001572 005402 NEG R2  
22 001574 010237 172412 MOV R2, @#WCR  
23 001600 012737 024000 172416 MOV #10240., @#DPREG  
24 001606 000240 NOP  
25 001610 105737 172414 TSTB @#CSR  
26 001614 000240 NOP  
27 001616 105037 172414 CLR @#CSR  
28 001622 012737 010420 172416 MOV #4368., @#DPREG  
29 001630 012737 004422 172416 MOV #2322., @#DPREG  
30 001636 105737 172414 TSTB @#CSR  
31 001642 100375 BPL -4  
32 001644 105037 172414 CLR @#CSR  
33 001650 000207 RTS FC  
34 001652 000 377 WTD: .BYTE 0,377  
35  
36  
37 *****  
38 ***** PROCEDURE TIMER (TIME) *****  
39 *****  
40 *****  
41 001654 005267 001006' TIME: INC LTIME  
42 001660 022767 077777 001006' CMP #077777, LTIME  
43 001666 001002 BNE 4$  
44 001670 005067 001006' CLR LTIME  
45 001674 000002 RTI  
46  
47  
48
```

```

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 *****
49 *****

;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 =WTKEN

RAM:      MOV    @ARG1(R5), R0
          CMP    #0, R0
          BNE   KDRAM
          MOV    @ARG2(R5), R0
          MOV    @ARG3(R5), R1
          MOV    #4354, @OPREG
          ADD   #2304, R0
          MOV    R0, @OPREG
          TSTB  @#CSR
          BPL  -4
          CLRB  @#CSR
          MOV    #4353, @OPREG
          ADD   #2304, R1
          MOV    R1, @OPREG
          TSTB  @#CSR
          BPL  -4
          CLRB  @#CSR
          RTS   PC

WTRAM:    MOV    @ARG2(R5), R0
          MOV    @ARG3(R5), R1
          MOV    #4354, @OPREG
          ADD   #2304, R0
          MOV    R0, @OPREG
          TSTB  @#CSR
          BPL  -4
          CLRB  @#CSR
          MOV    #4353, @OPREG
          ADD   #2304, R1
          MOV    R1, @OPREG
          TSTB  @#CSR
          BPL  -4
          CLRB  @#CSR
          MOV    @#IOBUF, R0
          BIC   #17760, R0
          MOV    R0, @ARG3(R5)
          RTS   PC

RDARAM:   MOV    @ARG2(R5), R0
          MOV    #4354, @OPREG
          ADD   #2304, R0
          MOV    R0, @OPREG
          TSTB  @#CSR
          BPL  -4
          CLRB  @#CSR
          MOV    #4353, @OPREG
          ADD   #2304, R1
          MOV    R1, @OPREG
          TSTB  @#CSR
          BPL  -4
          CLRB  @#CSR
          MOV    @#IOBUF, R0
          BIC   #17760, R0
          MOV    R0, @ARG3(R5)
          RTS   PC

; WHICH OPERATION
; READ OF
; ADDRESS
; WRITE DATA
; SEL LDATA
; WD/DATA (ADDR)
; WRITE DATA
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; WD/DATA (CMD)
; WRITE
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; ADDRESS TO READ
; SEL LDATA
; WD/DATA (ADDR)
; WRITE DATA
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; READ DATA
; VALID READ
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; FEICH DATA
; CLEAR BITS
; RTN DATA

```

```

1  .ENABLE LSB
2  *****
3  ***** PROCEDURE WRITE LOOP (LPOUT) *****
4  *****
5
6  LPOUT:  MOV  @ARG1(R5), R2
7          MOV  #340, R0
8          MTF5
9          MOV  $OUTBF, @#BAR
10         NEG  R2
11         MOV  R2, @#WCR
12         MOV  #4448, @#DPREG
13         MOV  #10240, @#DPREG
14         NOP
15         TSTB @#CSR
16         NOP
17         CLR8 @#CSR
18         MOV  #4576, @#DPREG
19         MOV  $WTD, @#BAR
20         MOV  #2, R2
21         NEG  R2
22         MOV  R2, @#WCR
23         MOV  #10240, @#DPREG
24         NOP
25         TSTB @#CSR
26         NOP
27         CLR8 @#CSR
28         MOV  #4368, @#DPREG
29         MOV  #2307, @#DPREG
30         TSTB @#CSR
31         BPL  ,-4
32         CLR8 @#CSR
33         MOV  #000, R0
34         MTF5
35         RTS  FC

```

```

;BYTE COUNT
;PRI=7
;BUS ADDRESS
; 2'S COMP COUNT
;COUNT
;OBO COMMAND
;DMA GO
;INTERFACE TIME
;DMA OK
;ERROR IF HERE
;OBI COMMAND
;ADDRESS OF WRITE TOKEN
;BYTE COUNT
;FORMAT FOR BLIUI
;DMA BYTE COUNT
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF NOT
;CLEAR DONE
;MODSTAT
;BUFFERS FULL
;GOOD WRITE
;NO LOOP UNTIL
;CLEAR DONE BIT
;PRI=0
;LOWER CPU

```

```

1 002266 017500 000002
2 002272 022700 000000
3 002276 001004
4 002300 012700 010410
5 002304 000167 000046
6 002310 022700 000001
7 002314 001004
8 002316 012700 010610
9 002322 000167 000030
10 002326 022700 000002
11 002332 001004
12 002334 012700 010510
13 002340 000167 000012
14 002344 022700 000004
15 002350 001020
16 002352 012700 010710
17 002356 010037 172416
18 002362 012737 001400
19 002370 105737 172414
20 002374 100375
21 002376 013700 172416
22 002402 042700 177400
23 002406 010075 000004
24 002412 000207
25 002416
26 002420
27 002424
28 002412 000207

*****
**** PROCEDURE READ BUFFER POINTER *****
*****
RDFNT:  MOV @ARG1(R5), R0
        CMP #0, R0
        BNE 1$
        MOV #4360., R0
        JMP 4$
        CMP #1, R0
        BNE 2$
        MOV #4488., R0
        JMP 4$
        CMP #2, R0
        BNE 3$
        MOV #4424., R0
        JMP 4$
        CMP #4, R0
        BNE 5$
        MOV #4552., R0
        MOV R0, @#DFREG
        MOV #768., @#DFREG
        TSTB @#CSR
        BPL .-4
        MOV @#IOBUF, R0
        BIC #177400, R0
        MOV R0, @ARG2(R5)
        RTS
        PC

1$:
2$:
3$:
4$:
5$:

;FETCH COMMAND
;INBUFO ?
;NO
;RDBUFADR INO
;INBUF1 ?
;NO
;RDBUFADR IN1
;OUTBUFO ?
;NO
;RDBUFADR OUTO
;OUTBUF1 ?
;NO RETURN
;RDBUFADR OUT1
;RDBUFADR
;RD
;READY
;NO LOOP UNTIL
;FETCH POINTER
;CLEAR MST
;RETURN POINTER

```

FDM.MACRO MACRO V03.01 4-JUN-79 17:30:32 PAGE 10

MODE22

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 *****
7 *****
8 *****
9 002414 017500 000002          ;CALL RSTART(0) -CAUSES A SOFTWARE HALT
10 002420 022700 000000          ;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
11 002424 001002          ;CALL RSTART(2) -LOAD MODE(173000)
12 002426 000000          *****
13 002430 000207          *****
14 002432 022700 000001          ;MODE
15 002436 001005          ;LOAD ADDRESS
16 002440 004767 175432          ;RESTART PROGRAM
17 002444 012700 000040          ;INT LIU FIRST
18 002450 000110          ;START ADDRESS
19 002452 022700 000002          ;LOAD MODE
20 002456 001002          *****
21 002460 000177 000002          *****
22 002464 000207          *****
23 002466 173000          *****
24
25
26

```

```

1  .ENABLE LSB
2  *****
3  ***** PROCEDURE SIG INTERFACE *****
4  *****
5
6  002470 010046      MOV     R0,      -(SP)
7  002472 010146      MOV     R1,      -(SP)
8  002474 010246      MOV     R2,      @#XINBF, R0
9  002476 012700      MOV     @#40.,  R1
10 002502 012701      MOV     @#HRCR, R1
11 002506 105737      TSTB   @#HRCR
12 002512 100375      BFL    @#HRCR, R2
13 002514 113702      MOVB   @#HRCR, R2
14 002520 042702      BIC    @#177400, R2
15 002524 010267      MOV     R2,      IOFLG
16 002530 110220      MOVB   R2,      (R0)+
17 002532 005301      DEC    R1
18 002534 105737      TSTB   @#HRCR
19 002540 100375      BFL    @#HRCR, R2
20 002542 113702      MOVB   @#HRCR, R2
21 002546 042702      BIC    @#177400, R2
22 002552 110220      MOVB   R2,      (R0)+
23 002554 005301      DEC    R1
24 002556 001366      BNE    2$,      @#HRCR
25 002560 042737      BIC    (SP)+,  R2
26 002566 012602      MOV     (SP)+,  R1
27 002570 012601      MOV     (SP)+,  R1
28 002572 012600      MOV     (SP)+,  R0
29 002574 000002      RTI
30
31
32
33 002576 017500      SOUT:: MOV     @#ARG1(R5),R0
34 002602 105737      TSTB   @#HRCR
35 002606 100375      BFL    @#HRCR, R2
36 002610 110037      MOVB   R0,      @#HXBUF
37 002614 000207      RTS
38
39
40
41 002616 105737      ;REPORT NODE READY
42 002622 100375      TSTB   @#HRCR
43 002624 112737      BFL    @#HRCR, R2
44 002626 000122      MOVB   @#122, @#HXBUF
45 002632 052737      BIS    @#100, @#HRCR
46 002640 000207      RTS
47
48

```

;SAVE REGISTERS

;BUFFER ADDRESS  
;BYTE COUNT  
;PORT READY  
;LOOP UNTIL  
;FETCH BYTE  
;CLEAR MST  
;ITS THE BYTE COUNT

;COUNT-1  
;PORT READY  
;LOOP UNTIL  
;FETCH BYTE  
;CLEAR MST  
;STORE IT

;READ 39 BYTES  
;NO MORE UNTIL FORTRAN  
;RESTORE REGISTERS

;BYTE COUNT  
;READY TO SNED  
;LOOP UNTIL  
;SEND IT

;PORT READY  
;LOOP UNTIL  
;SEND R  
;ENABLE INTERRUPTS  
;RETURN

∞  
∞

FDM-MACRO MACRO V03.01 4-JUN-79 17:30:32 PAGE 12  
NODE22

```
1 *****  
2 *****  
3 ***** COMMON DATA AREA *****  
4 *****  
5 0000000 .PSECT DFM,RW,D,GBL,REL,OVR  
6  
7 0000000 OUTBF: .BLKB 256.  
8 0004000 INBF: .BLKB 256.  
9 0010000 IWRTTM: .BLKW  
10 0010020 IWRT: .BLKW  
11 0010040 STAT: .BLKW  
12 0010060 LTIME: .BLKW  
13 .EVEN  
14  
15 0000000 .PSECT BUFS,RW,D,GBL,REL,OVR  
16  
17 0000000 XOUTBF: .BLKB 256.  
18 0004000 XINBF: .BLKB 256.  
19 0010000 ACKSER: .BLKB 256.  
20 0014000 LIDFD: .BLKB 256.  
21 0020000 IOFLG: .BLKW  
22 0020020 LLFLG: .BLKW  
23 0020040 IRSEND: .BLKW  
24 .EVEN  
25  
26  
27 0000001 .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	002
AREA	001206R		B2\$	000510R	G4\$	002444R	NXX =	000000	STAT	001004R	
ARG1	= 000002		B3\$	000534R	HREUF =	177562	ND24 =	000001	STATO	001422RG	
ARG2	= 000004		B4\$	000560R	HRC5R =	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	002 TIME	001654RG	
BIT00	= 000001		CAUSE	001204R	IOBUF =	172416	KDPNT	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	WTKEN	001474RG	
BIT04	= 000020		DESTR	001256RG	IWRTH	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	LIJINT	000076RG	RS1	002432R	XINBF	000400R	003
BIT08	= 000400		ENABLE	001214RG	LLFLG	002002R	RS2	002452R	XOUTBF	000000R	003
BIT14	= 040000		END\$	001202R	LPINFT =	***** G	RTI\$	001000R	XSID =	000000	
BIT15	= 100000		ENSTR	001246RG	LPOUT	002102RG	SID	002470RG	ZER0BF	000304R	
B0\$	000644R		E0\$	001228R	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/6 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



PAGE 001

12:00:00

01-JUL-79

COMF23.COM

PROGRAM:

decryptor

FOR NODAL /NOD INF  
FOR NODAL I/NOD INF  
FOR 0-300/NOD INF  
MAC: FBI

FA65 001

12:00:00

01-JUL-79

PROGRAM: 1802.00M

devout. for

ASSIGN DYO: BK:  
 R LINK  
 DCL:MODE.22-DXL:MAP DCL:MODAL/C/I/W  
 DCL:MODE/C  
 DCL:MODAL I/C  
 DCL:VSURE /  
 \$START

63516M DYL: BK:



## 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

<u>Parameter</u>	<u>RH</u>	<u>RL</u>	<u>AH</u>	<u>AL</u>
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.

FORTRAN IV V02.1-1 Mon 04-Jun-79 17:47:35

```

0001 PROGRAM MODAL
0002 INTEGER*2 XING,XOUTQ,ACKQ,FXING,FXOUTQ,PACKQ
0003 INTEGER*2 PING,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,MONTR,ISLID,DUM
0009 INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTIM,TIMLIM,ATIMLM
0010 REAL*4 RH,RL,AH,AL,RHT,RLT,AHT,ALT,DELTA,IRSV,VMEAS
0011 REAL*8 VTR
0012 COMMON /MESS/ MESSED
0013 COMMON /DFM/ OUTBF(256),INBF(256),
1     COMMON IWRTIM,IWRT,STAT,LTIME
0014 COMMON /QUE/ XING(16),FXING(2),XOUTQ(16),FXOUTQ(2),
1     ACKQ(16),PACKQ(2),INQ(16),PING(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
1     COMMON /PACK/ PACK(256,64)
0018 COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0019 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0020 RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0021 COMMON /DCOM/ ISWCH,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 CALL READY
0026 5 IF(PING(1) .LT. PING(2)) GOTO 40
0028 CALL ENABLE(0)
0029 Q2=DEQUE(PING,INQ,1)
0030 CALL ENABLE(1)
0031 DO 30 I=1,10
0032 IDATA(I)=PACK(I+6,Q2)
0033 CONTINUE
0034 CALL DSQCLP
0035 IFR=IFR+1
0036 FREE(IFR)=Q2
0037 CALL STATB0(15)
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(PXING(1) .LT. PXING(2)) GOTO 100
0052 CALL ENABLE(0)
0053 Q1=DEQUE(PXING,XING,1)
0054 CALL ENABLE(1)
0055 OUTQ=Q1
0056 85 CALL DESTR(PACK(254,OUTQ),Q2)

```

PAGE 002

Mon 04-Jun-79 17:47:35

FORTRAN IV U02.1-1

```
0057 DO 90 I=1,Q2
0058   OUTBF(I)=PACK(I,OUTQ)
0059   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   IWRTIM=0
0069   IWRT=1
0070   INFCT=0
0071   100 OLDTIM=NEWTIM
0072   NEWTIM=LTIME
0073   IF(IWRT .EQ. 0) GOTO 120
0075   IWRTIM=IWRTIM+(NEWTIM-OLDTIM)
0076   IF(IWRTIM .LT. TIMLIM) GOTO 120
0078   CALL WOKEN
0079   IWRTIM=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
```

PAGE 001

Mon 04-Jun-79 17:47:55

V02.1-1

FORTRAN IV

```
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
0008 IGETSP=FREE(IFR)
0009 IFR=IFR-1
0010 RETURN
0011 END
0012
```

PAGE 001

Mon 04-Jun-79 17:48:07

V02.1-1

FORTRAN IV

```

0001 SUBROUTINE ENQUE(A,B,N)
0002 LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003 INTEGER*2 XINQ,PXINQ,XOUTQ,FXOUTQ,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),FXOUTQ(2),
0007 1 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0008 1 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009 10 IQHEAD=A(1)
0010 10 IQTAIL=A(2)
0011 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
0017 NN=IQHEAD-IQTAIL
0018 DO 30 I=1,NN+1
0019 30 B(IQLNTH+I-I)=B(IQHEAD+I-I)
0020 A(1)=IQLNTH
0021 A(2)=IQLNTH-NN
0022 GOTO 10
0023 40 CALL INIT
0024 999 RETURN
0025 END
0026

```

```
FORTRAN IV      V02.1-1      Mon 04-Jun-79 17:48:20      PAGE 001

0001      FUNCTION DEQUE(A,B,ID)
0002      LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003      INTEGER*2 XING,PXING,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,FIND
0004      INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005      COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),
0006      1      ACKQ(16),PACKQ(2),INQ(16),FIND(2)
0007      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0008      1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009      IQHEAD=A(1)
0010      DEQUE=B(IQHEAD)
0011      IF(ID.NE.1) GOTO 999
0012      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 17:48:41      PAGE 001

0001      SUBROUTINE ACKNAK(N)
0002      INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,FXING,FXOUTQ,PXOUTQ
0003      INTEGER*2 ACKQ,PACKQ,PING,FREE,OUTQ,XOUTQ
0004      INTEGER*2 I1,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,IMRTM
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,IALIRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013      COMMON /DFM/ OUTBF(256),INBF(256),
1      IMRTM,IMRT,STAT,LTIME
0014      COMMON /QUE/  PACK(256,64)
0015      COMMON /QVE/  XINQ(16),PXING(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
1      PACKQ(2),INQ(16),PING(2)
0016      COMMON /FRE/  FREE(64),IFR,IFRSZ
0017      COMMON /MESS/ MESSED
0018      COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
1      IOFLG,LLFLG,IRSEND
0019      IF (N.EQ. 1) GOTO 10
0020      IFR=IFR+1
0021      DO 5 I=1,256
0022          PACK(I,OUTQ)=0
0023      5      CONTINUE
0024      IFULL=0
0025      ISENT=0
0026      IRSEND=0
0027      FREE(IFR)=OUTQ
0028      GOTU 999
0029      T2=PACK(256,OUTQ)
0030      T2=T2+1
0031      IF(T2.LE. RESNLM) GOTO 120
0032      T3=PACK(3,OUTQ)
0033      T4=MOD(T3,64)
0034      IF(T4.LT. 32) GOTO 110
0035      T5=IGETSP(N)
0036      IF(MESSEQ.EQ. 126) MESSEQ=0
0037      MESSEQ=MESSEQ+1
0038      PACK(I,T5)=0
0039      PACK(2,T5)=MESSEQ
0040      PACK(3,T5)=0
0041      PACK(4,T5)=0
0042      PACK(5,T5)=25
0043      PACK(6,T5)=ISLID
0044      DO 20 I=7,60
0045          PACK(I,T5)='040'
0046      20      CONTINUE
0047      DO 30 I=7,9
0048          PACK(I,T5)=LF
0049      30      CONTINUE

```

PAGE 002

F0RTRAN IV V02.1-1 Mon 04-Jun-79 17:48:41

```

0054 DO 40 I=1,8
0055   PACK(I+9,T5)=LCOMT(I)
0056 CONTINUE
0057 DO 50 I=9,16
0058   PACK(I+9,T5)=LCOMT(I)
0059 CONTINUE
0060 DO 60 I=17,22
0061   PACK(I+9,T5)=LCOMT(I)
0062 CONTINUE
0063 DO 70 I=25,29
0064   PACK(I+7,T5)=LCOMT(I)
0065 CONTINUE
0066 ENCODE(3,80,PACK(38,T5)) ISLID
0067 FORMAT(I3)
0068 DO 90 I=33,40
0069   PACK(I+11,T5)=LCOMT(I)
0070 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(PXING,XING,T5)
0078 DO 102 I=1,256
0079   PACK(I,OUTQ)=0
0080 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 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(PXING,XING,OUTQ)
0095 GOTO 999
0096 PACK(256,OUTQ)=T2
0097 IFULL=0
0098 ISENT=0
0099 IF(IRSEND.EQ.1) GOTO 999
0101 CALL ENQUE(PXING,XING,OUTQ)
0102 RETURN
0103 END

```

PAGE 001

FORTRAN IV V02.1-1 Mon 04-Jun-79 17:49:03

```

0001 SUBROUTINE INPTQ(L)
0002 INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,FXOUTQ,ACKQ
0003 INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005 LOGICAL*1 PACK,MONTR,ISLID,ETX,CR,LF,DUM
0006 INTEGER*2 LTIME,IWRITM
0007 COMMON /DFM/ OUTBF(256),INBF(256),
0008 IWRITM,IWRT,STAT,LTIME
0009 COMMON /RFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
0010 LIDFD(256),IOFLG,LLFLG,IRSEND
0011 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),FXOUTQ(2),
0012 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0013 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0014 RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0015 IF(L.LT. 1) GOTO 30
0016 LI=L-2
0017 II=INBF(6)
0018 IF(INBF(2).EQ. ACKSEQ(II)) GOTO 20
0019 T1=IGETSP(N)
0020 DO 10 I=1,LI
0021 PACK(I,T1)=INBF(I)
0022 CALL ENSTR(PACK(254,T1),LI)
0023 CALL ENQUE(PINQ,INQ,T1)
0024 N=2
0025 GOTO 40
0026 N=1
0027 GOTO 40
0028 N=0
0029 DO 50 I=1,2
0030 OUTBF(I)=INBF(I)
0031 CONTINUE
0032 NN=1
0033 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 LFOUT(9)
0042 IF(N.NE. 2) GOTO 999
0043 ACKSEQ(II)=INBF(2)
0044 RETURN
0045 999
0046 END

```

```

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

0001      SUBROUTINE LPIMPT(LI)
0002      INTEGER*2 STAT,FLWCNT,OUTFCT,OUTG,RESNLM
0003      INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004      LOGICAL*1 OUTRF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005      INTEGER*2 LTIME,IMRTM
0006      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTG,
0007      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008      CC1=INBF(3)
0009      CC2=INBF(4)
0010      IF(CCC1 .LT. 0) GOTO 25
0011      IF(CCC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0012      IF(CCC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0013      CC3=MOD(CCC1,2)
0014      CC4=MOD(CCC1,256)
0015      IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0016      IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0017      CC5=MOD(INBF(3),64)
0018      IF(CCC5 .GE. 32) CALL INPTQ(LI)
0019      GOTO 99
0020      20      CC7=MOD(CCC1,2)
0021      IF(CCC7 .GE. 1) GOTO 30
0022      25      CALL ACKNAK(0)
0023      GOTO 99
0024      30      CALL ACKNAK(1)
0025      GOTO 99
0026      40      CALL INPTQ(LI)
0027      99      CONTINUE
0028      RETURN
0029      END
0030
0031
0032
0033
0034

```

```

FORTRAN IV      V02.1-1      Mon. 04-Jun-79 17:49:34      PAGE 001

0001      SUBROUTINE INIT
0002      REAL*4 RH,RL,AH,AL,IRSV,VMEAS
0003      REAL*8 VTR
0004      INTEGER*2 XINQ,XOUTQ,ACKQ,PXING,PXOUTQ,PACKQ,PING,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,IWRITM
0011      COMMON /DFM/ OUTBF(256),INBF(256),
1 IWRITM,IWRT,STAT,LTIME
0012      COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
1 ACKQ(16),PACKQ(2),INQ(16),PING(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/ ISWTRCH,VMEAS(3),ICTR,RL(3),RH(3),AL(3),AH(3),
1 ITERM,NEL,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      INFCT=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      IWRITM=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      ISWTRCH=0

```

PAGE 002

Mon 04-Jun-79 17:49:34

FORTRAN IV V02.1-1

```

0049 QUITFCT=0
0050 IFULL=0
0051 ISENT=0
0052 DO 40 I=1,20
0053   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     LIDFD(I)=0
0064     DO 60 I=40,44
0065       LIDFD(I)=1
0066       DO 70 I=45,59
0067         LIDFD(I)=0
0068         DO 80 I=60,64
0069           LIDFD(I)=2
0070           DO 90 I=65,79
0071             LIDFD(I)=0
0072             DO 100 I=80,84
0073               LIDFD(I)=4
0074               DO 110 I=85,99
0075                 LIDFD(I)=0
0076                 DO 120 I=100,104
0077                   LIDFD(I)=5
0078                   DO 130 I=105,256
0079                     LIDFD(I)=0
0080                     PING(1)=IQLNTH
0081                     PING(2)=IQLNTH+1
0082                     PACKQ(1)=IQLNTH
0083                     PACKQ(2)=IQLNTH+1
0084                     PXOUTQ(1)=IQLNTH
0085                     PXOUTQ(2)=IQLNTH+1
0086                     PXING(1)=IQLNTH
0087                     PXING(2)=IQLNTH+1
0088                     DO 10 I=1,IFRSZ
0089                       PACK(255,I)=0
0090                       PACK(256,I)=0
0091                       DO 20 I=1,IFRSZ
0092                         FREE(I)=I
0093                         DO 30 I=1,256
0094                           ACKSEQ(I)=256
0095                           IFR=IFRSZ
0096                           RETURN
0097                           END

```

```

FORTRAN IV      V02.1-1      Mon 04-Jun-79 17:49:55      PAGE 001

0001      SUBROUTINE LINLOS
0002      REAL*8 RM1(5),LINO8,LIN18
0003      INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0004      INTEGER*2 XINQ,PXINQ,XOUTQ,FXOUTQ,ACKQ,PACKQ,PINQ
0005      LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0006      LOGICAL*1 PACK,M1(40),LINO(8),LINI(8)
0007      LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008      COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),FXOUTQ(2),ACKQ(16),
          *      PACKQ(2),INO(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,MONTR,LF,ETX,CR,DUM
          *      COMMON      PACK(256,64)
0011      COMMON /MESS/ MESSEQ
0012
0013      DATA LINO8,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(LINO,LINO8)
0018      EQUIVALENCE(LINI,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      CONTINUE
0036      DO 30 I=1,22
0037          PACK(I+9,T1)=M1(I)
0038      IF(IS.EQ.1) GOTO 50
0040      DO 40 I=1,8
0041          PACK(I+31,T1)=LINO(I)
0042      GOTO 70
0043      DO 60 I=1,8
0044          PACK(I+31,T1)=LINI(I)
0045      DO 80 I=23,40
0046          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)

```

PAGE 002

Mon 04-Jun-79 17:49:55

V02.1.1-1

FORTRAN IV

0054 LLFLG=0  
0055 RETURN  
0056 END

```

FORTRAN IV      V02.1-1      Mon 04-Jun-79 18:07:15      PAGE 001

0001      SUBROUTINE DSQC
0002      INTEGER*2 XING,XOUTQ,ACKQ,PXING,PXOUTQ,PACKQ,PING
0003      INTEGER*2 FLWNT,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 /UCOM/ ISWICH,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/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
1          PACKQ(2),ING(16),PING(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      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
1          IF ((VMEAS(ICTR) .GT. (AH(ICTR)-2) .AND.
2              VMEAS(ICTR) .LT. AH(ICTR)) .OR.
3              VMEAS(ICTR) .GT. AL(ICTR) .AND.
4              VMEAS(ICTR) .LT. (AL(ICTR)+2))) GOTO 100
1          IF ((VMEAS(ICTR) .GE. AH(ICTR) .AND.
2              VMEAS(ICTR) .LE. (RH(ICTR)-1)) .OR.
3              VMEAS(ICTR) .GE. (RL(ICTR)+1) .AND.
4              VMEAS(ICTR) .LE. AL(ICTR))) GOTO 200
1          IF ((VMEAS(ICTR) .GT. (RH(ICTR)-1) .AND.
2              VMEAS(ICTR) .LT. RH(ICTR)) .OR.
3              VMEAS(ICTR) .GT. RL(ICTR) .AND.
4              VMEAS(ICTR) .LT. (RL(ICTR)+1))) GOTO 300
0040      IF(ISWICH .EQ. 1) GOTO 40
0042      GOTO 410
0043      ISTAT=1
0044      GOTO 210
0045      100      IF(ISWICH .EQ. 1) GOTO 110

```

FORTRAN IV V02.1-1 Mon 04-Jun-79 18:07:15

```

0047      GOTO 410
0048 110  CALL TREND(2,VMEAS(ICTR))
0049      GOTO 410
0050 200  ISTAT=2
0051      IF(ISTWCH .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(PXIND,XIND,K1)
0081      CALL ENABLE(1)
0082      GOTO 410
0083 300  IF(ISTWCH .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

```

PAGE 003

FORTRAN IV V02.1-1 Mon 04-Jun-79 18:07:15

```
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(PXIND,XIND,K1)
0117 CALL ENABLE(1)
0118 ICHNSW=0
0119 FORMAT(I4)
0120 360 FORMAT(F12.6)
0121 410 IOFLG=0
0122 RETURN
0123 END
```

```

FORTRAN IV      V02.1-1      Mon 04-Jun-79 18:07:41      PAGE 001

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 XING,XOUTQ,ACKQ,PXING,PXOUTQ,PACKQ,FINQ
0007      INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLH
0008      COMMON PACK(256,64)
0009      COMMON /MESS/ MESSEQ
0010      COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
0011      PACKQ(2),INQ(16),FINQ(2)
0012      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,RESNLH,
0013      IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0014      COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
0015      IOFLG,LLFLG,IRSEND
0016      COMMON /DCOM/ ISWTCB,VMEAS(3),ICTR,RL(3),RH(3),AL(3),AH(3),
0017      ITERM,NFL,SFL,VTR,IVC,MON,IDATA(10),IRSV(3),ICHNSW,
0018      ICHAN
0019      IU=(RH(ICTR)-RL(ICTR))/30
0020      IF((IRSV(ICTR)-IVAL).GT.(4 * IU)) GOTO 20
0021      IF(NFL.GE.3) GOTO 40
0022      IF((IRSV(ICTR)-IVAL).GT.(2 * IU)) GOTO 30
0023      NFL=NFL+1
0024      IRSV(ICTR)=(.33 * IVAL + .67 * IRSV(ICTR))
0025      GOTO 60
0026      IF(SFL.GE.2) GOTO 30
0027      SFL=SFL+1
0028      GOTO 50
0029      IRSV(ICTR)=IVAL
0030      NFL=0
0031      GOTO 50
0032      SFL=0
0033      GOTO 10
0034      IF(ICND .EQ. 2) RETURN
0035      CALL ENABLE(0)
0036      K1=IGETSP(N)
0037      CALL ENABLE(1)
0038      IF(MESSEQ .EQ. 126) MESSEQ=0
0039      MESSEQ=MESSEQ+1
0040      PACK(1,K1)=0
0041      PACK(2,K1)=MESSEQ
0042      PACK(3,K1)=0
0043      PACK(4,K1)=0
0044      PACK(5,K1)=MONTOR
0045      PACK(6,K1)=ISLID
0046      DO 70 I=7,14
0047      PACK(I,K1)=XINBF(I+10)
0048      CONTINUE
0049      DO 75 I=15,16
0050      PACK(I,K1)=XINBF(I)
0051      CONTINUE
0052      ENCODE(4,80,PACK(18,K1)) IVAL
0053      FORMAT(A4)
0054      PACK(17,K1)=ICND
0055

```

PAGE 002

FORTRAN IV V02.1-1 Mon 04-Jun-79 18:07:41

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

```
FORTRAN IV      V02.1-1      Mon 04-Jun-79 18:07:59      PAGE 001

0001      SUBROUTINE DSQCCLF
0002      LOGICAL*1 IDATA
0003      REAL*4 RH,RL,AL,VMEAS,IRSV
0004      REAL*8 VTR
0005      COMMON /DCOM/ ISWICH,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      ISWICH=1
0014      ITERM=IDATA(4)
0015      RETURN
0016      ISWICH=0
0017      RETURN
0018      DECODE(4,75,IDATA(2),ERR=80) ICHAN
0019      FORMAT(I4)
0020      CALL SOUT(ICHAN)
0021      ICHNSW=1
0022      RETURN
0023      END
```

```

1 .TITLE FDM,MACRO
2 .SBTTL NODE_23
3 .IDENT /V3.0/
4 .GLOBL LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
5 .GLOBL WTKEN,RAM,STATUS,STATBO
6 .GLOBL LFOUT,KSTART,RDPNT,DEST,ENSTR,LPINPT
7 .NLIST CND
8 .PSECT
9

```

```

10 RO =%0
11 R1 =%1
12 R2 =%2
13 R3 =%3
14 R4 =%4
15 R5 =%5
16 SP =%6
17 PC =%7

```

!INTERFACE ADDRESS

```

20 BAK= 172410
21 WCR= 172412
22 CSK= 172414
23 IDBUF= 172416
24 OFREG= 172416
25 HRCRS= 177560
26 HRBUF= 177562
27 HXCSR= 177564
28 HXBUF= 177566

```

!INTERFACE OPTIONS

```

29 XCIO=1
30 XSIO=0
31 XGIO=1
32 NDXX=0
33 ND24=1
34 NHOST=0
35 ZHOST=1

```

!LOCAL VARIABLES

```

36 ARG1= 2
37 ARG2= 4
38 ARG3= 6
39 ARG4= 10
40 BIT15= 100000
41 BIT14= 40000
42 BIT08= 400
43 BIT07= 200
44 BIT06= 100
45 BIT05= 40
46 BIT04= 20
47 BIT03= 10
48 BIT02= 4

```

```

49 000000
50 000001
51 000002
52 000003
53 000004
54 000005
55 000006
56 000007

```

```

57 172410
58 172412
59 172414
60 172416
61 177560
62 177562
63 177564
64 177566

```

```

65 000001
66 000000
67 000001
68 000000
69 000001
70 000000
71 000001
72 000000

```

```

73 000002
74 000004
75 000006
76 000010
77 100000
78 040000
79 000400
80 000200
81 000100
82 000040
83 000020
84 000010
85 000004

```

FIM.MACRO  
NODE 23

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

66 000002  
67 000001  
68

BIT01= 2  
BIT00= 1

FDM:MACRO MACRO V03.01 4-JUN-79 17:50:23 PAGE 2  
 NODE 23

```

1 *****
2 ***** PROCEDURE MASTER START UP *****
3 *****
4 *****
5
6 MASTER: MOV    $340, R0
7 MTPS
8 MOV    $124, R0
9 MOV    $110, (R0)+
10 MOV   $340, (R0)
11 RESET
12 MOV    $60, R0
13 MOV    $SID, (R0)+
14 MOV    $340, (R0)+
15 MOV    $100, R0
16 MOV    $TIME, (R0)
17 BIC   $40100, $CSR
18 JSR   PC, LIUINT
19 CLR   $CSR
20 MOV   $000, R0
21 MTPS
22 RTS   PC
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
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

    MOV    $340, R0
    MTPS
    MOV    $124, R0
    MOV    $110, (R0)+
    MOV   $340, (R0)
    RESET
    MOV    $60, R0
    MOV    $SID, (R0)+
    MOV    $340, (R0)+
    MOV    $100, R0
    MOV    $TIME, (R0)
    BIC   $40100, $CSR
    JSR   PC, LIUINT
    CLR   $CSR
    MOV   $000, R0
    MTPS
    RTS   PC

    *****
    ***** PROCEDURE INITIALIZE *****
    *****

    ICLEAR ACRAM
    LIUINT: MOV   $4354, $OPREG
            MOV   $2304, $OPREG
            TSTB $CSR
            BPL  --4
            CLRB $CSR
            MOV   $4353, $OPREG
            MOV   $-235, R0
            MOV   $2311, $OPREG
            TSTB $CSR
            BPL  --4
            CLRB $CSR
            INC  R0
            BNE  1$
            ICLEAR INPUT/OUTPUT BUFFERS
            MOV   $10410, R1
            MOV   $10440, R0
            JSR   PC, ZEROP
            MOV   $1400, $OPREG
            TSTB $CSR
            BPL  --4
            CLRB $CSR
            MOV   $10410, R1
            MOV   $10640, R0
            JSR   PC, ZEROP
    1$:
            MOV   $4354, $OPREG
            MOV   $2304, $OPREG
            TSTB $CSR
            BPL  --4
            CLRB $CSR
            MOV   $4353, $OPREG
            MOV   $-235, R0
            MOV   $2311, $OPREG
            TSTB $CSR
            BPL  --4
            CLRB $CSR
            INC  R0
            BNE  1$
            ICLEAR INPUT/OUTPUT BUFFERS
            MOV   $10410, R1
            MOV   $10440, R0
            JSR   PC, ZEROP
            MOV   $1400, $OPREG
            TSTB $CSR
            BPL  --4
            CLRB $CSR
            MOV   $10410, R1
            MOV   $10640, R0
            JSR   PC, ZEROP
    *****
  
```

102	000232	012737	001400	172416	MOV	#1400, @#0FREG
103	000240	105737	172414	TSTB	@#CSR	!FALSE READ DATA
104	000244	100375		BFL	.-4	!GOOD RD
105	000246	105037	172414	CLRB	@#CSR	!NO RETRY
106	000252	012701	010510	MOV	#10510, R1	!CLEAR DONE BIT
107	000256	012700	010540	MOV	#10540, R0	!RDBUFADR COMMAND
108	000262	004767	000016	JSR	PC, ZEROFB	!SEL OUTBUFO COMMAND
109	000266	012701	010710	MOV	#10710, R1	!RDBUFADR COMMAND
110	000272	012700	010740	MOV	#10740, R0	!SEL OUTBUFO COMMAND
111	000276	004767	000002	JSR	PC, ZEROFB	!ZERO BUFFER POINTER
112	000302	000440		BR	STAT\$	!GO CLEAR STATUS
113	000304	010137	172416	ZEROFB: MOV	R1, @#0FREG	!RDBUFADR
114	000310	012737	001400	MOV	#1400, @#0FREG	!RD
115	000316	105737	172414	TSTB	@#CSR	!GOOD RD
116	000322	100375		BFL	.-4	!NO LOOP UNTIL READY
117	000324	105037	172414	CLRB	@#CSR	!CLEAR DONE BIT
118	000330	013702	172416	MOV	@#I0RUF, R2	!FETCH POINTER
119	000334	042702	177400	BIC	#177400, R2	!CLEAR MST BYTE
120	000340	010037	172416	MOV	R0, @#0FREG	!SEL BUFFER
121	000344	022702	000000	CMF	#0, R2	!POINTER=0?
122	000350	001412		BEQ	3\$	!YES RECHECK
123	000352	012737	001400	MOV	#1400, @#0FREG	!NO FALSE RD
124	000360	105737	172414	TSTB	@#CSR	!GOOD RD
125	000364	100375		BFL	.-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	MOV	R0, @#0FREG	!SEL BUFFER
130	000402	000207		RTS	PC	!RETURN
131						
132						
133						
134	000404	012737	010400	172416	STAT\$: MOV	!MCR:RS
135	000412	012737	002400	172416	MOV	!RS:(FALSE)
136	000420	012737	002400	172416	MOV	!RS
137	000426	000207			RTS	
138						
139						

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 000430 010046          LIO:      MOV      R0,      -(SP)
7 000432 010146          MOV      R1,      -(SP)
8 000434 010246          MOV      R2,      -(SP)
9 000436 010346          MOV      R3,      -(SP)
10 000440 010446          MOV      R4,      -(SP)
11 000442 010546          MOV      R5,      -(SP)
12 000444 005737          TST     @#CSR
13 000450 100153          BPL     RTI$
14 000452 042737          BIC     $BIT14, @#CSR
15 000460 012737          MOV     $4352, @#DPREG
16 000466 012737          MOV     $1280, @#DPREG
17 000474 013767          MOV     @#IOBUF, CAUSE
18 000502 042767          BIC     $177400, CAUSE
19
20
21
22 000510 132767          ;PERFORM OPERATION DEPENDING ON BITS
23 000516 001406          BITB   $BIT02, CAUSE
24 000520 012702          BEQ    B3$
25 000524 012703          MOV     $4360, R2
26 000530 004767          JSR    PC,      EMBF
27 000534 132767          BITB   $BIT03, CAUSE
28 000542 001406          BEQ    B4$
29 000544 012702          MOV     $4488, R2
30 000550 012703          MOV     $4512, R3
31 000554 004767          JSR    PC,      EMBF
32 000560 132767          BITB   $BIT04, CAUSE
33 000566 001407          BEQ    B5$
34 000570 132767          BITB   $BIT07, CAUSE
35 000576 001403          BEQ    B6$
36 000600 012767          MOV     $2,      LLFLG
37 000606 132767          BITB   $BIT05, CAUSE
38 000614 001403          BEQ    B6$
39 000616 012767          MOV     $1,      LLFLG
40 000624 132767          BITB   $BIT06, CAUSE
41 000632 001404          BEQ    B0$
42 000634 005067          CLR    IWRTHM
43 000640 005067          CLR    IWRTHM
44
45
46
47 000644 132767          ;BUFFER FULL
48 000652 001423          BITB   $BIT00, CAUSE
49 000654 012701          BEQ    B1$
50 000660 012702          MOV     $INRF, R1
51 000664 012703          MOV     $4360, R2
52 000670 012704          MOV     $4384, R3
53 000674 004767          JSR    PC,      EMBF
54 000700 012705          MOV     $AREA, R5
55 000704 012767          MOV     $1,      AREA
56 000712 010267          MOV     $2,      DATA
57 000716 004767          JSR    PC,      LPINFT

```

58	000722	132767	000002	000254	R1\$:	R1B	#BIT01, CAUSE		
59	000730	001423				REQ	RTI\$		
60	000732	012701	000400			MOV	#INBF, R1		;INRUF1 FULL
61	000736	012702	010610			MOV	#4488., R2		;NO MORE DONE
62	000742	012703	010640			MOV	#4512., R3		;BUFFER ADDRESS
63	000746	012704	000002			MOV	PC, EMBF		;RDRUFADR COMM
64	000752	004767	000046			JSR	#BIT01, R4		;SEL INBUF1
65	000756	012705	001206			MOV	#AREA, R5		;CRC BIT
66	000762	012767	000001	000216		MOV	#1, AREA		;GO EMPTY BUFFER
67	000770	010267	000216			MOV	R2, DATA		;DATA LINK AREA
68	000774	004767	000000G			JSR	PC, LPINFT		;CRC OK BYTE COUNT
69									;CALL FORTRAN QUE'ER
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., @#DFREG		;READ STATUS 1
80	001032	012737	002400	172416		MOV	#1280., @#DFREG		;RS
81	001040	013700	172416			MOV	@#RDRUF, R0		;FETCH STATUS
82	001044	130400				R1B	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, @#DFREG		;RDRUFADR
86	001060	012737	001400	172416		MOV	#768., @#DFREG		;RD
87	001066	105737	172414			TSTB	@#CSR		;GOOD RD
88	001072	100375				BPL	.-4		
89	001074	013702	172416			MOV	@#RDRUF, 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, @#DFREG		;SEL BUFFER
96	001124	012737	001400	172416		MOV	#768., @#DFREG		;FALSE RD
97	001132	105737	172414			TSTR	@#CSR		;DONE ON
98	001136	100375				BPL	.-4		;NO LOOP
99	001140	012737	021000	172416		MOV	#8704., @#DFREG		;FIRE DMA
100	001146	000240				NOP			;DELAY
101	001150	105737	172414			TSTR	@#CSR		;GOOD DMA
102	001154	100401				BMI	DMAOK		
103	001156	000240				NOP			
104	001160	012737	004400	172416	DMAOK:	MOV	#2304., @#DFREG		;ERROR IF HERE
105	001166	105737	172414			TSTR	@#CSR		;FALSE WD
106	001172	100375				BPL	.-4		;GOOD WD
107	001174	005704				TST	R4		
108	001176	100001				BFL	END\$		;WAS CRC OK
109	001200	010402				MOV	R4, R2		;YES
110	001202	000207				RIS	FC		;NO FLAG IT
111	001204	000000			END\$:	CAUSE:	.WORD 0		;RETURN
112	001206	000000			CAUSE:	AREA:	.WORD 0		;STATUS BYTE 0 HOLDER
113	001210	001212			AREA:	DATA:	.WORD 0		
114	001212	000000			DATA:		.WORD 0		

FDM.MACRO  
NODE 23

MACRO V03.01 4-JUN-79 17:50:23 PAGE 4

```
1          .ENABLE LSR
2          *****
3          ***** PROCEDURE ENABLE *****
4          *****
5          *****
6          001214 000002          ENABLE: MOV @ARG1(R5), R1
7          001220 022701 000001          CMP #1, R1
8          001224 001404          BEQ E1$
9          001226 042737 040000 172414 E0$: BIC #BIT14, @#CSR
10         001234 000207          RTS PC
11         001236 052737 040000 172414 E1$: BIC #BIT14, @#CSR
12         001244 000207          RTS PC
13
14         ;FORTRAN ENCODE/DECODE MACRO
15
16         001246 017575 000004 000002 ENSTR: MOV @ARG2(R5),@ARG1(R5)
17         001254 000207          RTS PC
18         001256 017575 000002 000004 DESTR: MOV @ARG1(R5),@ARG2(R5)
19         001264 000207          RTS PC
20
21
22         *****
23         ***** PROCEDURE LINE SWITCH *****
24         *****
25
26         ;CALL SWITCH(X) -SET OR RESET LINE SWITCHES
27         ;
28         ;04=SET PRIMARY LINE
29         ;32=RESET PRIMARY LINE
30         ;08=SET BACKUP LINE
31         ;64=RESET BACKUP LINE
32
33         001266 017501 000002          SWITCH: MOV @ARG1(R5),R1
34         001272 012737 010420 172416          MOV #4368., @#OPREG
35         001300 042701 004400          ADD #2304., R1
36         001304 010137 172416          MOV R1, @#OPREG
37         001310 105737 172414          TSTB @#CSR
38         001314 100375          BPL #-4
39         001316 105037 172414          CLRB @#CSR
40         001322 000207          RTS PC
41
42
```

```

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 STATUS: MOV @ARG1(R5), R1
11          CMP #1, R1
12          BEQ 2$
13          MOV #4352., @#DPREG
14          MOV #1280., @#DPREG
15          MOV @#IOBUF, R0
16          BIC #177400, R0
17          MOV R0, @ARG2(R5)
18          RTS
19          MOV #4480., @#DPREG
20          MOV #1280., @#DPREG
21          MOV @#IOBUF, R0
22          BIC #177400, R0
23          MOV R0, @ARG2(R5)
24          RTS
25          MOV #4352., @#DPREG
26          MOV #1280., @#DPREG
27          MOV @#IOBUF, R1
28          BIC #177400, R1
29          BITB #BIT03, R1
30          BEQ 3$
31          MOV #1, @ARG1(R5)
32          RTS
33          MOV #0, @ARG1(R5)
34          RTS
35
36
;WHICH BYTE
;COMPARE
;MCR : RS(0)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;MCR : RS(1)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;STATUS BYTE 1
;RS
;FETCH DATA
;BUFFER FULL
;LOAD RETURN
;LOAD RETURN

```

```

1 *****
2 ***** PROCEDURE WRITE TOKEN (WTKEN) ***** *
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 *****

;CALL WTKEN -LOADS AND WRITES A TOKEN TO LOOP

;READ STATUS(1)
;RS
;FETCH DATA
;BUFFER LOADED
;NO GENERATE TOKEN
;MODSTAT
;MARK OB0F/OB1F/WTCMD
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT

;SELECT BUFFER 1
;ADDRESS OF WRITE TOKEN
;# OF BYTES
;FORMAT IT
;DMA BYTE COUNT
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF HERE
;CLEAR DONE
;MODSTAT
;MARK OB1F/WTCMD
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;RETURN
;A WTKEN

;TIME+1
;?TIME FULL
;CLEAR CLOCK

;4480,, @#OPREG
;1280,, @#OPREG
@#IOBUF, R1
;BIT03, R1
REQ
;4368,, @#OPREG
;2323,, @#OPREG
@#CSR
.-4
@#CSR
FC
RTS
;4576,, @#OPREG
;WTD, @#BAR
;2, R2
NEG
R2, @#MCR
;10240,, @#OPREG
NOF
TSTB
@#CSR
NOF
@#CSR
CLRB
;4368,, @#OPREG
;2322,, @#OPREG
@#CSR
.-4
BPL
@#CSR
RTS
PC
;WTD: .BYTE 0,377

TIME: INC LTIME
CMF #077777, LTIME
BNE 4$
CLR LTIME
RTI

;01474 012737 010600 172416 WTKEN: MOV #4480,, @#OPREG
;01502 012737 002400 172416 MOV #1280,, @#OPREG
;01510 013701 172416 @#IOBUF, R1
;01514 013701 000010 ;BIT03, R1
;01520 001414 REQ
;01522 012737 010420 ;4368,, @#OPREG
;01530 012737 004423 ;2323,, @#OPREG
;01536 105737 172414 @#CSR
;01542 100375 .-4
;01544 105037 172414 @#CSR
;01550 000207 FC
;01552 012737 010740 WTKG: MOV #4576,, @#OPREG
;01560 012737 001652 172410 ;WTD, @#BAR
;01566 012702 000002 ;2, R2
;01572 005402 NEG
;01574 010237 172412 R2, @#MCR
;01600 012737 024000 ;10240,, @#OPREG
;01606 000240 NOF
;01610 105737 172414 TSTB
;01614 000240 @#CSR
;01616 105037 172414 CLRB
;01622 012737 010420 ;4368,, @#OPREG
;01630 012737 004422 ;2322,, @#OPREG
;01636 105737 172414 @#CSR
;01642 100375 .-4
;01644 105037 172414 BPL
;01650 000207 @#CSR
;01652 000 000 RTS
;PC
;WTD: .BYTE 0,377

TIME: INC LTIME
CMF #077777, LTIME
BNE 4$
CLR LTIME
RTI

```

MACRO V03.01 4-JUN-79 17:50:23 PAGE 7

FIM.MACRO  
NODE 23

```

1 *****
2 ***** PROCEDURE ACKRAM
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 =WITOKEN
12
13
14 001676 017500 000002 000002 @ARG1(R5), R0
15 001702 022700 000000 000000 #0, R0
16 001706 001035 RDRAM
17 001710 017500 000004 @ARG2(R5), R0
18 001714 017501 000006 @ARG3(R5), R1
19 001720 012737 010402 #4354., @0FREG
20 001726 062700 004400 #2304., R0
21 001732 010037 172416 MOV R0, @0FREG
22 001736 105737 172414 TSTB @CSR
23 001742 100375 BPL --4
24 001744 105037 172414 CLR @CSR
25 001750 012737 010401 MOV #4353., @0FREG
26 001756 062701 004400 ADD #2304., R1
27 001762 010137 172416 MOV R1, @0FREG
28 001766 105737 172414 TSTB @CSR
29 001772 100375 BPL --4
30 001774 105037 172414 CLR @CSR
31 002000 000207 RTS PC
32 002002 017500 000004 RDRAM: MOV @ARG2(R5), R0
33 002006 012737 010402 MOV #4354., @0FREG
34 002014 062700 004400 ADD #2304., R0
35 002020 010037 172416 MOV R0, @0FREG
36 002024 105737 172414 TSTB @CSR
37 002030 100375 BPL --4
38 002032 105037 172414 CLR @CSR
39 002036 012737 010401 MOV #4353., @0FREG
40 002044 012737 001400 MOV #68., @0FREG
41 002052 105737 172414 TSTB @CSR
42 002056 100375 BPL --4
43 002060 105037 172414 CLR @CSR
44 002064 013700 MOV @I0RUF, R0
45 002070 042700 177760 BIC #177760, R0
46 002074 010075 MOV R0, @ARG3(R5)
47 002100 000207 RTS PC
48
49

```

```

; RAM
; WHICH OPERATION
; READ OF
; ADDRESS
; WRITE DATA
; SEL L/DACR
; WD/DATA (ADDR)
; WRITE DATA
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; WD/DATA (CMD)
; WRITE
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; ADDRESS TO READ
; SEL L/DACR
; WD/DATA (ADDR)
; WRITE DATA
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; READ DATA
; VALID READ
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; FETCH DATA
; CLEAR BITS
; RTN DATA

```

```

1
2
3
4
5
6 002102 017502 000002 000000' 172410
7 002106 012700 000340
8 002112 106400
9 002114 012737 000000' 172410
10 002122 005402
11 002124 010237 172412
12 002130 012737 010540 172416
13 002136 012737 024000 172416
14 002144 000240
15 002146 105737 172414
16 002152 000240
17 002154 105037 172414
18 002160 012737 010740 172416
19 002166 012737 001652' 172410
20 002174 012702 000002
21 002200 005402
22 002202 010237 172412
23 002206 012737 024000 172416
24 002214 000240
25 002216 105737 172414
26 002222 000240
27 002224 105037 172414
28 002230 012737 010420 172416
29 002236 012737 004403 172416
30 002244 105737 172414
31 002250 100375
32 002252 105037 172414
33 002256 012700 000000
34 002262 106400
35 002264 000207
36
37
38

```

```

*****ENABLE LSB*****
*****PROCEDURE WRITE LOOP (LFOUT)*****
*****

```

```

LFOUT:  MOV  @ARG1(R5), R2
        MOV  #340, R0
        MTF5 R0
        MOV  @OUTBF, @BBAR
        NEG  R2
        MOV  R2, @#WCR
        MOV  #4448., @#DFREG
        MOV  #10240., @#DFREG
        NOP
        TSTB @#CSR
        NOP
        CLR8 @#CSR
        MOV  #4576., @#DFREG
        MOV  @WTD, @BBAR
        MOV  #2, R2
        NEG  R2
        MOV  R2, @#WCR
        MOV  #10240., @#DFREG
        NOP
        TSTB @#CSR
        NOP
        CLR8 @#CSR
        MOV  #4368., @#DFREG
        MOV  #2307., @#DFREG
        TSTB @#CSR
        BPL  -4
        CLR8 @#CSR
        MOV  #000, R0
        MTF5 R0
        RTS

```

```

;BYTE COUNT
;PRI=7
;BUS ADDRESS
; 2'S COMP COUNT
;COUNT
;JOB COMMAND
;DMA GO
;INTERFACE TIME
;DMA OK
;ERROR IF HERE
;JOB1 COMMAND
;ADDRESS OF WRITE TOKEN
;BYTE COUNT
;FORMAT FOR BLUI
;DMA BYTE COUNT
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF NOT
;CLEAR DONE
;MODSTAT
;BUFFERS FULL
;GOOD WRITE
;NO LOOP UNTIL
;CLEAR DONE BIT
;PRI=0
;FLOWER CPU

```

```

1 5 002246 017500 000002
2 6 002272 022700 000000
3 7 002276 001004
4 8 002300 012700 010410
5 9 002304 000167 000046
6 10 002310 022700 000001
7 11 002314 001004
8 12 002316 012700 010610
9 13 002322 000167 000030
10 14 002326 022700 000002
11 15 002332 001004
12 16 002334 012700 010510
13 17 002340 000167 000012
14 18 002344 022700 000004
15 19 002350 001020
16 20 002352 012700 010710
17 21 002356 010037 172416
18 22 002362 012737 001400
19 23 002370 105737 172414
20 24 002374 100375
21 25 002376 013700 172416
22 26 002402 042700 177400
23 27 002406 010075 000004
24 28 002412 000207

```

```

*****
**** PROCEDURE READ BUFFER POINTER
*****
RDPNT:  MOV @ARG1(R5), R0
        CMP #0, R0
        BNE 1$, R0
        MOV #4360., R0
        JMP 4$, R0
1$:     CMP #1, R0
        BNE 2$, R0
        MOV #4488., R0
        JMP 4$, R0
2$:     CMP #2, R0
        BNE 3$, R0
        MOV #4424., R0
        JMP 4$, R0
3$:     CMP #4, R0
        BNE 5$, R0
        MOV #4552., R0
        MOV R0, @#DFREG
        MOV #768., @#DFREG
        TSTB @#CSR
        BFL --4
        MOV @#IDRUF, R0
        BIC #177400, R0
        MOV R0, @ARG2(R5)
        RTS
5$:

```

```

;FETCH COMMAND
;INBUFO ?
;NO
;RDBUFADR INO
;INBUF1 ?
;NO
;RDBUFADR IN1
;OUTBUFO ?
;NO
;RDBUFADR OUTO
;OUTBUF1 ?
;NO RETURN
;RDBUFADR OUT1
;RDBUFADR
;RD
;READY
;NO LOOP UNTIL
;FETCH POINTER
;CLEAR MST
;RETURN POINTER

```

FDM. MACRO MACRO V03.01 4-JUN-79 17:50:23 PAGE 10

NODE 23

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 *****
7 *****
8 *****
9 002414 017500 000002 KSTART: MOV @ARG1(R5), R0 ;MODE
10 002420 022700 000000 RS0:  #0, R0 ;LOAD ADDRESS
11 002424 001002 BNE R51
12 002426 000000 HALT
13 002430 000207 RTS PC
14 002432 022700 000001 RS1:  CMP #1, R0 ;RESTART PROGRAM
15 002436 001005 BNE RS2 ;INT LIU FIRST
16 002440 004767 175432 JSR PC, LIUINT ;START ADDRESS
17 002444 012700 000040 G4:  MOV #40, R0
18 002450 000110 JMP (R0) ;LOAD MODE
19 002452 022700 000002 RS2:  CMP #2, R0
20 002456 001002 BNE RST
21 002460 000177 000002 JMP BRDM
22 002464 000207 RTS PC
23 002466 173000 ROM:  .WORD 173000
24
25
26

```

```

1
2
3
4
5
6 002470 010046
7 002472 010146
8 002474 010246
9 002476 012700 000400'
10 002502 012701 000050
11 002506 105737 177560
12 002512 100375
13 002514 113702 177562
14 002520 042702 177400
15 002524 010267 002000'
16 002530 110220
17 002532 005301
18 002534 105737 177560
19 002540 100375
20 002542 113702 177562
21 002546 042702 177400
22 002552 110220
23 002554 005301
24 002556 001366
25 002560 042737 000100 177560
26 002566 012602
27 002570 012601
28 002572 012600
29 002574 000002
30
31
32
33 002576 017500 000002
34 002602 105737 177564
35 002606 100375
36 002610 110037 177566
37 002614 000207
38
39
40
41
42 002616 105737 177564
43 002622 100375
44 002624 112737 000122 177566
45 002632 052737 000100 177560
46 002640 000207
47
48

```

```

*****
***** PROCEDURE SIG INTERFACE *****
*****

```

```

SIO::  MOV R0, -(SP)
        MOV R1, -(SP)
        MOV R2, -(SP)
        #XINBF, R0
        MOV #40, R1
        TSTB @#HRCR
        BPL -4
        MOV @#HRCRUF, R2
        BIC #177400, R2
        MOV R2, IOFLG
        MOV R2, (R0)+
        DEC R1
        TSTB @#HRCR
        BPL -4
        MOV @#HRCRUF, R2
        BIC #177400, R2
        MOV R2, (R0)+
        DEC R1
        BNE 2$
        BIC #100, @#HRCR
        MOV (SP)+, R2
        MOV (SP)+, R1
        MOV (SP)+, R0
        RTI

        ;SEND OVER TO SIG

SOUT::  MOV @ARG1(R5),R0
        TSTB @#HRCR
        BPL -4
        MOV R0, @#HXBUF
        RTS

        ;REPORT NODE READY

READY:: TSTB @#HRCR
        BPL -4
        MOV #122, @#HXBUF
        BIS #100, @#HRCR
        RTS

```

```

;SAVE REGISTERS
;BUFFER ADDRESS
;BYTE COUNT
;PORT READY
;LOOP UNTIL
;FETCH BYTE
;CLEAR MST
;ITS THE BYTE COUNT
;COUNT-1
;PORT READY
;LOOP UNTIL
;FETCH BYTE
;CLEAR MST
;STORE IT
;READ 39 BYTES
;NO MORE UNTIL FORTRAN
;RETURN REGISTERS
;BYTE COUNT
;READY TO SNEED
;LOOP UNTIL
;SEND IT
;PORT READY
;LOOP UNTIL
;SEND R
;ENABLE INTERRUPTS
;RETURN

```

MACRO V03.01 4-JUN-79 17:50:23 PAGE 12

FDH,MACRO  
NODE 23

\*\*\*\*\*  
;\*\*\*\* COMMON DATA AREA  
;\*\*\*\*\*

.FSECT DFH,RW,D,GBL,REL,OVR

OUTBF: .BLNB 256.  
INBF: .BLNB 256.

IWRTH: .BLNW  
IWRIT: .BLNW  
STAT: .BLNW  
LTIME: .BLNW  
.EVEN

.FSECT BUFS,RW,D,GBL,REL,OVR

XOUTBF: .BLNB 256.  
XINBF: .BLNB 256.  
ACKSEG: .BLNB 256.  
LIDFD: .BLNB 256.

IOFLG: .BLNW  
LLFLG: .BLNW  
IRSEND: .BLNW  
.EVEN

.END

000001

1 000000  
2  
3  
4 000000  
5  
6  
7 000000  
8 000400  
9 001000  
10 001002  
11 001004  
12 001006  
13  
14  
15 000000  
16  
17 000000  
18 000400  
19 001000  
20 001400  
21 002000  
22 002002  
23 002004  
24  
25  
26  
27

MACRO V03.01 4-JUN-79 17:50:23 PAGE 12-1

FDM-MACRO  
SYMBOL TABLE

ACKSER	001000F	003	B1\$	000722R	E1\$	001236R	MASTER	000000RG	SOUT	002576RG	002
AREA	001206R		B2\$	000510R	G4\$	002444R	NDXX =	000000	STAT	001004R	
ARG1	= 000002		B3\$	000534R	HRBUF =	177562	ND24 =	000001	STAT0	001422RG	
ARG2	= 000004		B4\$	000560R	HRCRSR =	177560	NHOST =	000000	STATUS	001324RG	
ARG3	= 000006		B5\$	000606R	HXBUF =	177566	OFREG =	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	KDFNT	002266RG	WCR =	172412	
BIT01	= 000002		CRCHK	001054R	IOFLG	002000R	003 RDRAM	002002R	WTD	001652R	
BIT02	= 000004		CSR =	172414	IRSEND	002004R	003 READY	002616RG	WTKG	001552R	
BIT03	= 000010		DATA	001212R	IWR	001002R	002 K0M	002466R	WTKEN	001474RG	
BIT04	= 000020		DESTR	001256RG	IWRITM	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	XINEF	000400R	003
BIT08	= 000400		ENABLE	001214RG	LLFLG	002002R	003 RS2	002452R	XOUTBF	000000R	003
BIT14	= 040000		END\$	001202R	LPINFT =	***** G	RTI\$	001000R	XSIO =	000000	
BIT15	= 100000		ENSTR	001246RG	LPOUT	002102RG	SIO	002470RG	ZERORP	000304R	
B0\$	000644R		E0\$	001226R	LTIME	001006R					

\* ABS. 000000 000  
002642 001  
DFM 001010 002  
RUF\$ 002006 003  
ERRORS DETECTED: 0

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

PAGE 001

12:00:00

01-08-79

COM 23.000

PROGRAM:

INQUIRY

FOR NODAL/NDI LINE  
FOR NODAL/L/NDI LINE  
FOR DISC/NDI LINE  
MAC FDR

PAGE 001

12:00:00

01-JUL-79

LNK23.00M

Program:

Computer:

ASSIGN DX0: DN:  
 R LINK  
 DX1:NDRE 3,DX1:MAF=DX1:NDRAL /C/L/W  
 DX1:FIM/C  
 DX1:NDRAL I/C  
 DX1:DS-DC//  
 \$STHR1

ASSIGN DX1: DN:

## 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.

FORTRAN IV V02.1-1 Tue 05-Jun-79 13:41:40 PAGE 001

```

0001 PROGRAM NODIAL
0002 INTEGER*2 XING,PXING
0003 INTEGER*2 PINQ,FREE,STAT,FLWCNT,OUTFCT
0004 INTEGER*2 SETPRM,RSTPRM,SETBKF
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,IWRITM,TIMLIM,ATIMLM
0011 INTEGER*2 KEYREC
0012 COMMON /HES/ HESSEQ
0013 COMMON /DFM/ OUTBF(256),INBF(256),
1 COMMON IWRITM,IWRT,STAT,LTIME
0014 COMMON/QUE/ XING(15),PXING(2),INQ(15),PING(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 /SMT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0020 COMMON /GLOB/ ISENT,FLWCNT,IOLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRT,ISLID,MONTR,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 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 CONTINUE
0047 DO 30 I=1,LEN
0048 ICODE(I)=PACK(I,Q2)
0049 CONTINUE
0050 CALL USRLNG(LEN)
0051 IFR=IFR+1
0052 FREE(IFR)=Q2
0053 40 CALL STATB0(IS)

```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 13:41:40      PAGE 002

0054      IF(IS .EQ. 1) GOTO 100
0056      IF(LFLG .GT. 0) CALL LINLDS
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(PXING(1) .LT. PXING(2)) GOTO 100
0066      CALL ENABLE(0)
0067      Q1=DEQUE(PXING,XING,1)
0068      CALL ENABLE(1)
0069      OUTQ=Q1
0070      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      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      INFLCT=0
0089      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 WTKEN
0097      IWRTTM=0
0098      IWRT=0
0099      IF(ISENT .EQ. 0) GOTO 130
0101      ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0102      IF(ACKTIM .LT. ATIMLH) GOTO 130
0104      CALL ENABLE(0)
0105      CALL ACKNAK(0)
0106      CALL ENABLE(1)
0107      CONTINUE
130      GOTO 5
999      CALL ENABLE(0)
0109      CALL MASTER(1)
0110      CALL MASTER(1)
0111      STOP
0112      END

```

PAGE 001

FORTRAN IV V02.1-1 Tue 05-Jun-79 13:42:02

```
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)
0008 IGETSP=FREE(IFR)
0009 IFR=IFR-1
0010 RETURN
0011 END
0012
```

PAGE 001

Tue 05-Jun-79 13:42:14

V02.1-1

FORTRAN IV

```

0001 SUBROUTINE ENQUE(A,B,N)
0002 LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0003 INTEGER*2 XING,PXING,ING,PING
0004 INTEGER*2 FLWCNT,OUTQ,A(2),B(15)
0005 INYGER*2 RESNLM,OUTFCT
0006 COMMON/QUE/ XING(15),PXING(2),ING(15),PING(2)
0007 COMMON/GLOB/ ISENT,FLWCNT,IQLNTH,IQLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0008 10 IQHEAD=A(1)
0009 IQTAIL=A(2)
0010 IF(IQTAIL.EQ. 1) GOTO 20
0011 IQTAIL=IQTAIL-1
0012 B(IQTAIL)=N
0013 A(2)=IQTAIL
0014 GOTO 999
0015
0016 20 IF(IQHEAD.GE. IQLNTH) GOTO 40
0017 NN=IQHEAD-IQTAIL
0018 DO 30 I=1,NN+1
0019 30 B(IQLNTH+1-I)=B(IQHEAD+1-I)
0020 A(1)=IQLNTH
0021 A(2)=IQLNTH-NN
0022 GOTO 10
0023
0024 40 CALL INIT(1)
0025 999 RETURN
0026 END

```

PAGE 001

Tue 05-Jun-79 13:42:27

V02.1-1

FORTRAN IV

```
0001 FUNCTION DEQUE(A,B,ID)
0002 LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003 INTEGER*2 XING,PXING,ING,PING
0004 INTEGER*2 FLWENT,A(2),B(15),DEQUE,OUTFCT,OUTQ,RESNLM
0005 COMMON/QUE/ XING(15),PXING(2),ING(15),PING(2)
0006 COMMON /GLOB/ ISENT,FLWENT,IQLNTH,IQLNTH+1,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   Tue 05-Jun-79 13:42:56          PAGE 001

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,LIDFU
0008  INTEGER*2 LTIME,IMRTH
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,IOLNTH,OUTFCT,IFULL,OUTQ,
1    RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013  COMMON /DFM/ OUTBF(256),INBF(256),
1    IMRTH,IMRT,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),
0019  IF (N .NE. 1) GOTO 10
0020  IFR=IFR+1
0021  DO 5 I=1,256
0022      PACK(I,OUTQ)=0
0023  5    CONTINUE
0024  IFULL=0
0025  ISENT=0
0026  IRSEND=0
0027  FREE(IFR)=OUTQ
0028  GOTO 999
0029  10  T2=PACK(256,OUTQ)
0030  T2=T2+1
0031  IF(T2 .LE. RESNLM) GOTO 120
0032  T3=PACK(3,OUTQ)
0033  T4=MOD(T3,64)
0034  IF(T4 .LT. 32) GOTO 110
0035  T5=IGETSP(N)
0036  IF(MESSEQ .EQ. 126) MESSEQ=0
0037  MESSEQ=MESSEQ+1
0038  PACK(1,T5)=0
0039  PACK(2,T5)=MESSEQ
0040  PACK(3,T5)=0
0041  PACK(4,T5)=0
0042  PACK(5,T5)=MONTOR
0043  PACK(6,T5)=ISLID
0044  DO 20 I=7,60
0045      PACK(I,T5)='40
0046  20  CONTINUE
0047  DO 30 I=7,9
0048      PACK(I,T5)=LF
0049  30  CONTINUE
0050  DO 40 I=1,8
0051      PACK(I,T5)=LF
0052  40  CONTINUE
0053  DO 40 I=1,8
0054

```

PAGE 002

FORTRAN IV V02.1-1 Tue 05-Jun-79 13:42:56

```

0055   PACK(I+9,TS)=LCOMT(I)
0056   CONTINUE
0057   DO 50 I=9,16
0058     PACK(I+9,TS)=LCOMT(I)
0059   CONTINUE
0060   DO 60 I=17,24
0061     PACK(I+9,TS)=LCOMT(I)
0062   CONTINUE
0063   DO 70 I=25,31
0064     PACK(I+9,TS)=LCOMT(I)
0065   CONTINUE
0066   ENCODE(3,80,PACK(41,TS))PACK(6,OUTQ)
0067   FORMAT(I3)
0068   DO 90 I=33,40
0069     PACK(I+12,TS)=LCOMT(I)
0070   CONTINUE
0071   ENCODE(3,80,PACK(53,TS))PACK(5,OUTQ)
0072   PACK(56,TS)=LF
0073   PACK(57,TS)=LF
0074   PACK(58,TS)=CR
0075   PACK(59,TS)=ETX
0076   CALL ENSTR(PACK(254,TS),59)
0077   CALL ENQUE(PXING,XING,TS)
0078   DO 102 I=1,256
0079     PACK(I,OUTQ)=0
0080   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(PXING,XING,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(PXING,XING,OUTQ)
0102   RETURN
0103   END

```

PAGE 001

V02.1-1 Tue 05-Jun-79 13:43:17

FORTRAN IV

```

0001 SUBROUTINE INPTQ(L)
0002 INTEGER*2 STAT,XINQ,PXINQ
0003 INTEGER*2 PING,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
0004 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,TEMP,OUTBF,INBF
0005 LOGICAL*1 PACK,MONTR,ISLID,ETX,CR,LF,DUM
0006 INTEGER*2 LTIME,IWRITM
0007 COMMON /DFM/ OUTBF(256),INBF(256),
1 IWRITM,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),PING(2)
0011 COMMON /GLOB/ ISENT,FLWCNT,IOLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRI,ISLID,MONTR,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 IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0020 T1=IGETSP(N)
0021 DO 15 I=1,LI
0022 PACK(I,T1)=INBF(I)
0023 CALL ENSTR(PACK(254,T1),LI)
0024 CALL ENQUE(PING,INQ,T1)
0025 N=2
0026 GOTO 40
0027 N=1
0028 GOTO 40
0029 N=0
0030 DO 50 I=1,2
0031 OUTBF(I)=INBF(I)
0032 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 RETURN
0048 END
999

```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 13:43:34      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,IWRTTH
0006      COMMON /DFM/ OUTBF(256),INBF(256),IWRTM,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

```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 13:43:49      PAGE 001

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,IOLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0020      COMMON /USER/ ST(4,9),IND,ICD(256),NRCNO,NOREC,ICON(256)
0021      COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
2 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      INFLECT=0
0035      ISLID=24
0036      CR='015'
0037      LF='012'
0038      ETX='003'
0039      RESNLM=2
0040      IOLNTH=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

```

PAGE 002

Tue 05-Jun-79 13:43:49

FORTRAN IV V02.1-1

```

0050 MONTOR=25
0051 IWRT=0
0052 OUTFACT=0
0053 IFULL=0
0054 ISENT=0
0055 DO 40 I=1,20
0056 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 LIDFD(I)=0
0067 DO 60 I=40,44
0068 LIDFD(I)=1
0069 DO 70 I=45,59
0070 LIDFD(I)=0
0071 DO 80 I=60,64
0072 LIDFD(I)=2
0073 DO 90 I=65,79
0074 LIDFD(I)=0
0075 DO 100 I=80,84
0076 LIDFD(I)=4
0077 DO 110 I=85,99
0078 LIDFD(I)=0
0079 DO 120 I=100,104
0080 LIDFD(I)=5
0081 DO 130 I=105,256
0082 LIDFD(I)=0
0083 PING(1)=IQLNTH
0084 PING(2)=IQLNTH+1
0085 PXING(1)=IQLNTH
0086 PXING(2)=IQLNTH+1
0087 DO 10 I=1,IFRSZ
0088 PACK(255,I)=0
0089 DO 20 I=1,IFRSZ
0090 DO 20 I=1,IFRSZ
0091 FREE(I)=I
0092 DO 30 I=1,256
0093 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)(CODE(I),I=1,20)
0100 CALL ASSIGN(3,'CDIR.DAT')
0101 DEFINE FILE 3(10,40,U,J1)
0102 READ(3,1)(CODE(I),I=1,1,80)
0103 CALL ASSIGN(4,'TDIR.DAT')

```

PAGE 003

FORTRAN IV V02.1-1 Tue 05-Jun-79 13:43:49

```
0104 DEFINE FILE 4(10,40,U,J2)
0105 READ(4,1)(ICODE(I),I=1,80)
0106 CALL ASSIGN(5,'CATD.DAT')
0107 DEFINE FILE 5(100,40,U,J3)
0108 READ(5,1)(ICODE(I),I=1,80)
0109 CALL ASSIGN(6,'TKND.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,NRCND)
0117 RETURN
0118 END
```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 13:44:14      PAGE 001

0001      SUBROUTINE LINLOS
0002      REAL*8 RMI(5),LIN08,LIN18
0003      INTEGER*2 XING,PXING,PING
0004      INTEGER*2 FLWCNT,OUTFCT,OUTQ,RESNLM,T1
0005      LOGICAL*1 ETX,CR,LF,MONTR,ISLID,XOUTBF,XINBF,DUM
0006      LOGICAL*1 PACK,M1(40),LINO(8),LINI(8),ACKSEQ,LIDFD
0007      COMMON/QUE/ XING(15),PXING(2),INQ(15),FINQ(2)
0008      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0009      1      RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0010      COMMON /PACK/ PACK(256,30)
0011      COMMON /MESS/ MESSEQ
0012      COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
0013      *      IOFLG,LLFLG,IRSEND
0014      DATA LIN08,LINI0/'PRIMARY ','BACKUP '//
0015      DATA RMI(1),RMI(2)/'LOSS OF ','MODULATI'/
0016      DATA RMI(3),RMI(4)/'ON ON LO','OP AT NO'/
0017      DATA RMI(5)/'DE 24 '//
0018      EQUIVALENCE(LINO,LIN08)
0019      EQUIVALENCE(LINI,LINI8)
0020      EQUIVALENCE(MI,RMI)
0021      IS=LLFLG
0022      CALL ENABLE(0)
0023      T1=IGETSP(N)
0024      CALL ENABLE(1)
0025      PACK(1,T1)=0
0026      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)=24
0033      DO 20 I=7,9
0034      PACK(I,T1)=LF
0035      20      CONTINUE
0036      DO 30 I=1,22
0037      PACK(I+9,T1)=M1(I)
0038      IF(I5 .EQ. 1) GOTO 50
0039      DO 40 I=1,8
0040      PACK(I+31,T1)=LINO(I)
0041      GOTO 70
0042      DO 60 I=1,8
0043      PACK(I+31,T1)=LINI(I)
0044      DO 80 I=23,40
0045      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(PXING,XING,T1)
0052      CALL ENABLE(1)
0053      LLFLG=0
0054      RETURN

```

PAGE 002

V02.1-1 Tue 05-Jun-79 13:44:14

FDRTRAN IV

0055 END

```

1  .TITLE FDM.MACRO
2  .IDENT /V3.0/
3  .GLOBL LIJINT,LIO,ENABLE,SWITCH,MASTER,TIME
4  .GLOBL WTONEN,RAM,STATUS,STATBO
5  .GLOBL LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINFT
6  .NLIST CND
7  .PSECT
8
9
10 000000
11 000001
12 000002
13 000003
14 000004
15 000005
16 000006
17 000007
18
19
20
21 172410
22 172412
23 172414
24 172416
25 172416
26 177560
27 177562
28 177564
29 177566
30
31
32
33 000001
34 000001
35 000001
36
37
38
39 000002
40 000004
41 000006
42 000010
43
44 100000
45 040000
46 000400
47 000200
48 000100
49 000040
50 000020
51 000010
52 000004
53 000002
54 000001
55

```

```

R0 =X0
R1 =X1
R2 =X2
R3 =X3
R4 =X4
R5 =X5
SP =X6
PC =X7

```

```

; INTERFACE ADDRESS
BAR= 172410
WCR= 172412
CSR= 172414
IDBUF= 172416
OPREG= 172416
HRCSR= 177560
HRBUF= 177562
HXCSR= 177564
HXBUF= 177566

```

```

; INTERFACE OPTIONS
XCIO=1
XSIO=1
XGID=1

```

```

; LOCAL VARIABLES
ARG1= 2
ARG2= 4
ARG3= 6
ARG4= 10
BIT15= 100000
BIT14= 40000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1

```

```

1 *****
2 *****
3 *****
4 *****
5 MASTER: MOV @ARG1(R5), R0 ;START OR RESTORE
6 CMP #1, R0 ;RESTORE?
7 BEQ RSTVEC ;YES JUMP OVER START
8 MOV #340, R0 ;PRI=7
9 MTPS R0
10 MOV #124, R0 ;LIU HANDLER
11 MOV #10, (R0)+ ;PRI=7
12 MOV #340, (R0) ;CLOCK VECTOR
13 MOV #100, R0 ;SAVE IT FOR RECOVERY
14 MOV (R0), VECSAV ;CLOCK
15 MOV #TIME, (R0) ;DISABLE INTERRUPTS
16 BIC #40100, @#CSR ;CLEAR LIU
17 JSR PC, LIUINT ;CLEAR BLUI CSR
18 CLR @#CSR ;PRI=0
19 MOV #000, R0 ;CPU DOWN
20 MTPS R0
21 RTS PC ;VECTOR SPOT
22 RSTVEC: MOV #100, R0 ;RESTORE IT
23 MOV VECSAV, (R0)
24 RTS PC
25 VECSAV: .WORD 0 ;VECTOR MAILBOX,
26
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****

```

FDM-MACRO MACRO V03.01 5-JUN-79 13:44:43 PAGE 2-1

```

76 000226 105037 172414 CLR B
77 000232 012701 010610 MOV
78 000236 012700 010640 MOV
79 000242 004767 000052 JSR
80 000246 012737 001400 MOV 172416
81 000254 105737 172414 TST B
82 000260 100375 BFL
83 000262 105037 @#CSR
84 000266 012701 010510 MOV 172414
85 000272 012700 010540 MOV
86 000276 004767 000016 JSR
87 000302 012701 010710 MOV
88 000306 012700 010740 MOV
89 000312 004767 000002 JSR
90 000316 000440 BR
91 000320 010137 172416 MOV
92 000324 012737 001400 MOV 172416
93 000332 105737 172414 TST B
94 000336 100375 BFL
95 000340 105037 172414 CLR B
96 000344 013702 172416 MOV
97 000350 042702 177400 BIC
98 000354 010037 172416 MOV
99 000360 022702 000000 CMP
100 000364 001412 BEQ 3$
101 000366 012737 001400 MOV 172416
102 000374 105737 172414 TST B
103 000400 100375 BFL
104 000402 105037 172414 CLR B
105 000406 005302 DEC
106 000410 000763 BR
107 000412 010037 172416 MOV
108 000416 000207 RTS
109
110
111
112 000420 012737 010400 MOV 172416
113 000426 012737 002400 MOV 172416
114 000434 012737 002400 MOV 172416
115 000442 000207 RTS
116
117

@#CSR
#10610, R1
#10640, R0
PC, ZEROBF
#1400, @#DPREG
@#CSR
.-4
@#CSR
#10510, R1
#10540, R0
PC, ZEROBF
#10710, R1
#10740, R0
PC, ZEROBF
STAT$
R1, @#DPREG
R1, @#DPREG
@#CSR
.-4
@#CSR
@#IOBUF, R2
#177400, R2
R0, @#DPREG
#0, R2
3$
#1400, @#DPREG
@#CSR
.-4
@#CSR
R2
2$
R0, @#DPREG
PC
;READ AND CLEAR STATUS
#4352., @#DPREG
#1380., @#DPREG
#1280., @#DPREG
PC

;CLEAR DONE BIT
;RDBUFADR COMMAND
;SEL INBUFI COMMAND
;POINTER=0
;FALSE READ DATA
;GOOD R/U
;NO RETRY
;CLEAR DONE BIT
;RDBUFADR COMMAND
;SEL OUTBUFO COMMAND
;RDBUFADR COMMAND
;SEL OUTBUFO COMMAND
;ZERO BUFFER POINTER
;GO CLEAR STATUS
;RDBUFADR
;RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;FETCH POINTER
;CLEAR MST BYTE
;SEL BUFFER
;POINTER=0?
;YES RECHECK
;NO FALSE RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;LOCAL POINTER-1
;BR UNTIL 0
;SEL BUFFER
;RETURN

;MCR:RS
;RS(FALSE)
;RS

```

```

1 *****
2 ***** PROCEDURE LIU-HANDLER *****
3 *****
4 *****
5
6 000444 010046 LIO: MOV R0, -(SF)
7 000446 010146 MOV R1, -(SF)
8 000450 010246 MOV R2, -(SF)
9 000452 010346 MOV R3, -(SF)
10 000454 010446 MOV R4, -(SF)
11 000456 010546 MOV R5, -(SF)
12 000460 005737 TST @#CSR
13 000464 100133 RPL RTI$
14 000466 042737 BIC #BIT14, @#CSR
15 000474 012737 MOV #4352, @#DFREG
16 000502 012737 MOV #1280, @#DFREG
17 000510 013767 MOV @#IDRUF, CAUSE
18 000516 042767 BIC #177400, CAUSE
19
20
21
22 000524 132767 BITB #BIT02, CAUSE
23 000532 001406 BEQ B3$
24 000534 012702 MOV #4360, R2
25 000540 012703 MOV #4384, R3
26 000544 004767 JSR PC, EMBF
27 000550 132767 BITB #BIT03, CAUSE
28 000556 001406 BEQ B4$
29 000560 012702 MOV #4488, R2
30 000564 012703 MOV #4512, R3
31 000570 004767 JSR PC, EMBF
32 000574 132767 BITB #BIT04, CAUSE
33 000602 001407 BEQ B5$
34 000604 132767 BITB #BIT07, CAUSE
35 000612 001403 BEQ B5$
36 000614 012767 MOV #02002, LFLG
37 000622 132767 BITB #BIT05, CAUSE
38 000630 001403 BEQ B6$
39 000632 012767 MOV #1, LFLG
40 000640 132767 BITB #BIT06, CAUSE
41 000646 001404 BEQ B6$
42 000650 005067 CLR IWRTH
43 000654 005067 CLR IWR
44
45
46
47 000660 132767 BITB #BIT00, CAUSE
48 000666 001423 BEQ B1$
49 000670 012701 MOV #INBF, R1
50 000674 012702 MOV #4360, R2
51 000700 012703 MOV #4384, R3
52 000704 012704 MOV #BIT00, R4
53 000710 004767 JSR PC, EMBF
54 000714 012705 MOV #AREA, R5
55 000720 012767 MOV #1, AREA
56 000726 010267 MOV #2, DATA
57 000732 004767 JSR PC, LFINPT

```

```

;SAVE R0
;SAVE R1
;SAVE R2
;SAVE R3
;SAVE R4
;SAVE R5
;? LIU INTERRUPTED
;NO
;YES/DISABLE LIU
;WCR ; RS(0)
;RD
;CLEAR UNUSED BITS
;OV-FL
;RDRUFADR INO
;SEL INBUFO
;GO EMPTY
;OV-FL
;RDRUFADR INI
;SEL INBUF1
;GO EMPTY
;LINE-LOSS PRIMARY
;LATCH ON
;YES LOOP AROUND
;FLAG PRIMARY SW. FAIL
;BACKUP LINE LOSS
;FLAG BACKUP SW. FAIL
;WRITE TOKEN DETECT
;CLEAR TOKEN VAR
;INBUFO FULL
;NO
;BUFFER ADDRESS
;RDRUFADR COMM
;SEL BUF
;CRC BIT
;EMPTY BUFFER
;DATA LINK AREA
;1 VARIABLE
;CRC OR BYTE COUNT
;CALL FORTRAN QUE'ER

```

;PERFORM OPERATION DEPENDING ON BITS

;BUFFER FULL

FDM.MACRO MACRO V03.01 5-JUN-79 13:44:43 PAGE 3-1

58	000736	132767	000002	000254	BI\$:	BITB	#BIT01, CAUSE	!INBUF1 FULL
59	000744	001423				BEG	RTI\$	!NO WERE DONE
60	000746	012701	000400'			MOV	#INBF, R1	!BUFFER ADDRESS
61	000752	012702	010510			MOV	#4488., R2	!RBUFADR COMM
62	000756	012703	010640			MOV	#512., 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 VARIABLE
67	001004	010267	000316			MOV	R2, DATA	!CRC OR BYTE COUNT
68	001010	004767	0000006			JSR	PC, LFINFT	!CALL FORTRAN QUE'ER
69					RTI\$:			!RESTORE REGISTERS
70	001014	012605				MOV	(SF)+, R5	
71	001016	012604				MOV	(SF)+, R4	
72	001020	012603				MOV	(SF)+, R3	
73	001022	012602				MOV	(SF)+, R2	
74	001024	012601				MOV	(SF)+, R1	
75	001026	012600				MOV	(SF)+, 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., @#DFREG	!READ STATUS 1
80	001046	012737	002400	172416		MOV	#1280., @#DFREG	!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, @#DFREG	!RDBUFADR
86	001074	012737	001400	172416		MOV	#768., @#DFREG	!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, @#MCR	!BYTE COUNT NOW
94	001130	010137	172410			MOV	R1, @#RAR	!ADDRESS IN MEMORY
95	001134	010337	172416			MOV	R3, @#DFREG	!SEL BUFFER
96	001140	012737	001400	172416		MOV	#768., @#DFREG	!FALSE RD
97	001146	105737	172414			TSTB	@#CSR	!DONE ON
98	001152	100375				BPL	.-4	!NO LOOP
99	001154	012737	021000	172416		MOV	#8704., @#DFREG	!FIRE DMA
100	001162	000240				NOF		!DELAY
101	001164	105737	172414			TSTB	@#CSR	!GOOR DMA
102	001170	100401				BMI	DMAOK	
103	001172	000240				NOF		!ERROR IF HERE
104	001174	012737	004400	172416	DMAOK:	MOV	#2304., @#DFREG	!FALSE WD
105	001202	105737	172414			TSTB	@#CSR	!GOOD WD
106	001206	100375				BPL	.-4	
107	001210	005704				TST	R4	!MAS CRC OK
108	001212	100001				BPL	END\$	!YES
109	001214	010402				MOV	R4, R2	!NO FLAG IT
110	001216	000207				RTS	PC	!RETURN
111	001220	000000			END\$:	.WORD	0	!STATUS BYTE 0 HOLDER
112	001222	000000			CAUSE:	.WORD	0	
113	001224	001226'			AREA:	.WORD	DATA	
114	001226	000000			DATA:	.WORD	0	

FDM-MACRO MACRO V03.01 5-JUN-79 13:44:43 PAGE 4

```

1      .ENABLE LSH
2      *****
3      ***** PROCEDURE ENABLE *****
4      *****
5
6      001230 017501 000002          @ARG1(R5), R1
7      001234 022701 000001          #1, R1
8      001240 001404          BEQ E1$
9      001242 042737 172414 E0$:    @BIT14, @#CSR
10     001250 000207          RTS PC
11     001252 052737 172414 E1$:    @BIT14, @#CSR
12     001260 000207          RTS PC
13
14     ;FDRTRAN ENCODE/DECODE MACRO
15
16     001262 017575 000004 ENSTR:   MOV @ARG2(R5),@ARG1(R5)
17     001270 000207          RTS PC
18     001272 017575 000002 DESTR:   MOV @ARG1(R5),@ARG2(R5)
19     001300 000207          RTS PC
20
21
22     *****
23     ***** PROCEDURE LINE SWITCH *****
24     *****
25
26     ;CALL SWITCH(X) -SET OR RESET LINE SWITCH
27     ;
28     ;04=SET PRIMARY LINE
29     ;32=RESET PRIMARY LINE
30     ;08=SET BACKUP LINE
31     ;64=RESET BACKUP LINE
32
33     SWITCH: MOV @ARG1(R5),R1
34     001302 017501 000002          MOV #4368, @#OPREG
35     001304 012737 010420          ADD #2304,, R1
36     001314 062701 004400          MOV R1, @#DFREG
37     001320 010137 172416          TSTB @#CSR
38     001324 105737 172414          BPL -4
39     001330 100375          CLRB @#CSR
40     001332 105037 172414          RTS PC
41     001336 000207          RTS PC
42
;WHICH SUBROU.
;DISABLE LIU INTERRUPTS
;ENABLE LIU
;ENCODE BYTE COUNT
;DECODE BYTE COUNT
;PSW SET
;WCR ; MODSTAT
;RD
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT

```

FDM.MACRO MACRO V03.01 5-JUN-79 13:44:43 PAGE 5

```

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 *****

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

STATUS: MOV @ARG1(R5), R1
CMP #1, R1
BEQ 2$
MOV #4352,, @#DFREG
MOV #1280,, @#DFREG
MOV @#IOBUF, R0
BIC #177400, R0
MOV R0, @ARG2(R5)
RTS PC
MOV #4480,, @#DFREG
MOV #1280,, @#DFREG
MOV @#IOBUF, R0
BIC #177400, R0
MOV R0, @ARG2(R5)
RTS PC
STATUS: MOV @ARG1(R5), R1
CMP #1, R1
BEQ 1$:
MOV #10400 172416
MOV #02400 172416
MOV #13700 172416
MOV #42700 177400
MOV #10075 000004
RTS PC
MOV #10600 172416
MOV #02400 172416
MOV #13700 172416
MOV #42700 177400
MOV #10075 000004
RTS PC
STATUS: MOV @ARG1(R5), R1
CMP #1, R1
BEQ 2$:
MOV #10400 172416
MOV #02400 172416
MOV #13701 172416
MOV #42701 177400
MOV #10075 000010
RTS PC
MOV #01404 000001
MOV #12775 000001
MOV #00207 000002
MOV #12775 000000
MOV #00207 000002
RTS PC
MOV #01466 001404
MOV #12775 012775
MOV #00207 000207
MOV #12775 012775
MOV #00207 000207
RTS PC
MOV #01506 000000
MOV #12775 000000
MOV #00207 000207
RTS PC
;WHICH BYTE
;COMPARE
;MCR : RS(0)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;MCR : RS(1)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;STATUS BYTE 1
;RS
;FETCH DATA
;BUFFER FULL
;LOAD RETURN
;LOAD RETURN

```

```

1 *****
2 ***** PROCEDURE WRITE TOKEN (WTKEN) *****
3 *****
4 *****
5 ;CALL WTKEN -LOADS AND WRITES, A TOKEN TO LOOP
6
7 001510 012737 010600 172416 WTKEN: MOV #4480., @OPFREG ;READ STATUS(1)
8 001516 012737 02400 172416 MOV #1280., @OPFREG ;RS
9 001524 013701 172416 MOV @BIT0BUF, R1 ;FETCH DATA
10 001530 132701 000010 BITB @BIT03, R1 ;BUFFER LOADED
11 001534 001414 BRR WTKG ;NO GENERATE TOKEN
12 001536 012737 010420 MOV #4368., @OPFREG ;MODSTAT
13 001544 012737 004423 MOV #2322., @OPFREG ;MARK OB0F/0B1F/WTCMD
14 001552 105737 172414 TSTB @DCSR ;VALID WRITE
15 001556 100375 BPL -4 ;NO LOOP UNTIL READY
16 001560 105037 CLRFB @DCSR ;CLEAR DONE BIT
17 001564 000207 RTS PC
18 001566 012737 WTKG: MOV #4576., @OPFREG ;SELECT BUFFER 1
19 001574 012737 172416 WTKG: MOV @WTD, @BAR ;ADDRESS OF WRITE TOKEN
20 001602 012702 000002 NEG R2 ;# OF BYTES
21 001606 005402 MOV R2, @WCR ;FORMAT IT
22 001610 010237 172412 MOV R2, @WCR ;DMA BYTE COUNT
23 001614 012737 024000 MOV #10240., @OPFREG ;FIRE DMA
24 001622 000240 NOP ;DELAY
25 001624 105737 172414 TSTB @DCSR ;GOOD DMA
26 001630 000240 NOP ;ERROR IF HERE
27 001632 105037 CLRFB @DCSR ;CLEAR DONE
28 001636 012737 172414 MOV #4368., @OPFREG ;MODSTAT
29 001644 012737 004422 MOV #2322., @OPFREG ;MARK OB0F/0B1F/WTCMD
30 001652 105737 172414 TSTB @DCSR ;VALID WRITE
31 001656 100375 BPL -4 ;NO LOOP UNTIL READY
32 001660 105037 CLRFB @DCSR ;CLEAR DONE BIT
33 001664 000207 RTS PC ;RETURN
34 001666 000 WTD: .BYTE 0,377 ;A WTKEN
35
36 *****
37 ***** PROCEDURE TIMER (TIME) *****
38 *****
39 *****
40
41 001670 005267 001006' TIME: INC LTIME ;TIME+1
42 001674 022767 077777 001006' CMP #077777, LTIME ;?TIME FULL
43 001702 001002 BNE 4$ ;
44 001704 005067 001006' CLR LTIME ;CLEAR CLOCK
45 001710 000002 RTI
46
47
48

```

FDM,MACRO MACRO V03.01 5-JUN-79 13:44:43 PAGE 7

```

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 *****
49 *****

;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 =WTKEN

RAM:
MOV @RG1(R5), R0
CMP #0, R0
BNE KDRAM
MOV @RG2(R5), R0
MTRAM:
MOV @RG3(R5), R1
MOV #4354., @#OPREG
MOV #2304., R0
ADD #2304., R0
MOV R0, @#OPREG
TSTB @#CSR
BFL -4
CLRB @#CSR
MOV #4353., @#DPREG
ADD #2304., R1
MOV R1, @#OPREG
TSTB @#CSR
BPL -4
CLRB @#CSR
RTS PC
RDRAM:
MOV @RG2(R5), R0
MOV #4354., @#OPREG
ADD #2304., R0
MOV R0, @#OPREG
TSTB @#CSR
BPL -4
CLRB @#CSR
MOV #4353., @#OPREG
MOV #768., @#DFREG
TSTB @#CSR
BFL -4
CLRB @#CSR
MOV @#IOBUF, R0
MOV #17760, R0
BIC R0, @RG3(R5)
MOV R0, @RG3(R5)
RTS PC

; RAM
;WHICH OPERATION
;READ OP
;ADDRESS
;WRITE DATA
;SEL LDATA
;WD/DATA (ADDR)
;WRITE DATA
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;SEL ACRAM
;WD/DATA (CMD)
;WRITE
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;ADDRESS TO READ
;SEL LDATA
;WD/DATA (ADDR)
;WRITE DATA
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;SEL ACRAM
;READ DATA
;VALID READ
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;FETCH DATA
;CLEAR BITS
;RTN DATA

```

```

1          .ENABLE LSR
2          *****
3          ***** PROCEDURE WRITE LOOP (LFOUT) *****
4          *****
5          *****
6          LFOUT:  MOV  @ARG1(R5), R2
7                  MOV  #340, R0
8                  MTF5
9                  MOV  @OUTBF, @BBAR
10                 NEG  R2, @WCR
11                 MOV  #4448., @DFREG
12                 MOV  #10240., @DFREG
13                 NOP
14                 TSTB @CSR
15                 NOP
16                 CLR8
17                 MOV  #4576., @DFREG
18                 MOV  @WD, @BBAR
19                 MOV  #2, R2
20                 NEG  R2
21                 MOV  R2, @WCR
22                 MOV  #10240., @DFREG
23                 NOP
24                 TSTB @CSR
25                 NOP
26                 CLR8
27                 MOV  #4368., @DFREG
28                 MOV  #2307., @DFREG
29                 TSTB @CSR
30                 BFL  -4
31                 MOV  @CSR, R0
32                 MTF5 R0
33                 RTS
34                 PC
35
36
37
38
          #BYTE COUNT
          #PRI=7
          #BUS ADDRESS
          # 2'S COMP COUNT
          #COUNT
          #OBO COMMAND
          #DMA GO
          #INTERFACE TIME
          #DMA OK
          #ERROR IF HERE
          #OBI COMMAND
          #ADDRESS OF WRITE TOKEN
          #BYTE COUNT
          #FORMAT FOR BLUI
          #DMA BYTE COUNT
          #FIRE DMA
          #DELAY
          #GOOD DMA
          #ERROR IF NOT
          #CLEAR DONE
          #MODSTAT
          #BUFFERS FULL
          #GOOD WRITE
          #NO LOOP UNTIL
          #CLEAR DONE BIT
          #PRI=0
          #LOWER CPU

```

```

1 5 002302 017500 000002
2 6 002306 022700 000000
3 7 002312 001004
4 8 002314 012700 010410
5 9 002320 000167 000046
6 10 002324 022700 000001
7 11 002330 001004
8 12 002332 012700 010610
9 13 002336 000167 000030
10 14 002342 022700 000002
11 15 002346 001004
12 16 002350 012700 010510
13 17 002354 000167 000012
14 18 002360 022700 000004
15 19 002364 001020
16 20 002366 012700 010710
17 21 002372 010037 172416
18 22 002376 012737 001400
19 23 002404 105737 172414
20 24 002410 100375
21 25 002412 013700 172416
22 26 002416 042700 177400
23 27 002422 010075 000004
24 28 002426 000207

```

```

*****
**** PROCEDURE READ BUFFER POINTER
*****

```

```

RUPNT:  MOV @ARG1(R5), R0
        CMP #0, R0
        BNE 1$
        MOV #4360., R0
        JMP 4$
1$:     CMP #1, R0
        BNE 2$
        MOV #4488., R0
        JMP 4$
2$:     CMP #2, R0
        BNE 3$
        MOV #4424., R0
        JMP 4$
3$:     CMP #4, R0
        BNE 5$
        MOV #4552., R0
        MOV R0, @#0FREG
        MOV #768., @#0FREG
        TSTB @#CSR
        BFL .-4
        MOV @#10BUF, R0
        BIC #177400, R0
        MOV R0, @ARG2(R5)
        RTS PC

```

```

;FETCH COMMAND
;INBUFO ?
;NO
;RDBUFADR INO
;INBUFI ?
;NO
;RDBUFADR INI
;OUTBUFO ?
;NO
;RDBUFADR OUTO
;OUTRUF1 ?
;NO RETURN
;RDBUFADR OUT1
;RD
;READY
;NO LOOP UNTIL
;FETCH POINTER
;CLEAR LIST
;RETURN POINTER

```

FDM.MACRO MACRO V03.01 5-JUN-79 13:44:43 PAGE 10

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 *****
7 *****
8 *****
9 002430 017500 000002 RSTART: MOV BARG1(R5), R0 ;MODE
10 002434 022700 000000 RSO:  #0, R0 ;LOAD ADDRESS
11 002440 001002 BNE RSI
12 002442 000000 HALT
13 002444 000207 RTS PC
14 002446 022700 000001 RS1:  #1, R0 ;RESTART PROGRAM
15 002452 001005 BNE RS2
16 002454 004767 175432 JSR PC, LIUINT ;INT LIU FIRST
17 002460 012700 000040 G4$:  #40, R0 ;START ADDRESS
18 002464 000110 JMP (R0)
19 002466 022700 000002 RS2:  #2, R0 ;LOAD MODE
20 002472 001002 BNE RST
21 002474 000177 000002 JMP BROM
22 002500 000207 RST:  PC
23 002502 173000 ROM:  .WORD 173000
24
25
26
235

```

FDM.MACRO MACRO V03.01 5-JUN-79 13:44:43 PAGE 11

```

*****
***** COMMON DATA AREA *****
*****

```

.PSECT IFM,RW,D,GBL,REL,OVR

```

OUTBF: .BLKB 256.
INBF: .BLKB 256.
IWRITM: .BLKW
IWRITM: .BLKW
STAT: .BLKW
LTIME: .BLKW
        .EVEN

```

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

```

XOUTBF: .BLKB 256.
XINBF: .BLKB 256.
ACKSEQ: .BLKB 256.
LIDFD: .BLKB 256.
IOFLG: .BLKW
LLFLO: .BLKW
IRSEND: .BLKW
        .EVEN

```

```

1 5 000000
2 7 000000
3 8 000400
4 9 001000
5 10 001002
6 11 001004
7 12 001006
8 13
9 14
10 15 000000
11 16
12 17 000000
13 18 000400
14 19 001000
15 20 001400
16 21 002000
17 22 002002
18 23 002004
19 24
20 25
21 26
22 27

```

.END

000001

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

SYMBOL TABLE

ACKSEQ	001000R	003	B0\$	000660R	E0\$	001242R	LFOUT	002116RG	002	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	HRCR\$	= 177560	OUTBF	000000R	002	TIME	001670RG	
ARG4	= 000010		B5\$	000622R	HXBUR	= 177566	RAM	001712RG		VECSAV	000110R	
BAR	= 172410		B6\$	000640R	HXCUR	= 177564	RDPNT	002302RG		WCR	= 172412	
BIT00	= 000001		B7\$	000604R	INBF	000400R	002	RDRAM		WTD	001666R	
BIT01	= 000002		CAUSE	001220R	IOBUF	= 172416	ROM	002502R		WTKG	001566R	
BIT02	= 000004		CRCOK	001070R	IOFLG	002000R	003	RST	002500R	WTKEN	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	IWRTH	001000R	002	RS0	002434R	XGIO	= 000001	
BIT06	= 000100		DMAOK	001174R	LIDFD	001400R	003	RS1	002446R	XINBF	000400R	
BIT07	= 000200		ENBF	001040R	LIO	000444RG	RS2	002466R		XOUTBF	000000R	
BIT08	= 000400		ENABLE	001230RG	LIUINT	000112RG	RTI\$	001014R		XSI0	= 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

PAGE 001

12:00:00

01-00-79

CONFIDENTIAL

PROGRAM:

DESCRIPTION:

FOR NOBAL/UNITS:9/NOI IRE  
FOR NOBAL 1/NOI IRE  
MAC EUM

F604 001

12:00:00

01-JUL-79

1 NR, 4, 404

PROF: 06:

00000000

055 BX1: BK1

R 1 DR

NOU 25 03 11 00P BX1: 00000000/01/00

BX1: 00000000

BX1: 00000000

BX1: 00000000

BX1: 00000000

BX1: 00000000

BX1: 00000000

BX1: 00000000

BX1: 00000000

BX1: 00000000

\*\$100K

055 BX1: BK1

## 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 XING,XOUTQ,ACKQ,PXING,FXOUTQ,PACKQ
0003 INTEGR*2 FING,FREE,STAT,FLWCNT,OUTFCT
0004 INTEGR*2 SETPRM,RSTPRM,SETBKP,IXON
0005 INTEGR*2 RSTBKP,01,02,RESNLM,OUTQ,DEQUE
0006 LOGICAL*1 XOUTBF,XINBF,ACKSEG,LIDFD,OUTBF,INBF
0007 LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
0008 INTEGR*2 LTIME,ACNTIM,NEWTIM,OLDTIM,IMLIM,ATIMLM
0009 COMMON /MESS/ MESSQ
0010 COMMON /DFM/ OUTBF(256),INBF(256),IMRTM,IMRT,STAT,LTIME
0011 COMMON /QE/ XING(16),PXING(2),XOUTQ(16),FXOUTQ(2),
0012 COMMON /ACKQ(16),PACKQ(2),INQ(16),PING(2)
0013 COMMON /LIDFD(256),IOFLG,LLFLG,IRSEND
0014 COMMON /FRE/ FREE(64),IFR,IFRSZ
0015 COMMON /TIM/ OLDTIM,IMLIM,ACKTIM,ATIMLM
0016 COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0017 COMMON /GLOB/ ISENT,FLWCNT,IGLNTH,OUTFCT,IFULL,OUTQ,
0018 RESNLM,IALTR,ISLID,MONTOR,LF,ETX,CR,DUM
0019 COMMON /CRT/IXON
0020 IXON=0
0021 CALL MASTER
0022 CALL INIT
0023 CALL ENABLE(1)
0024 CONTINUE
0025 IF(PING(1) .LT. PING(2)) GOTO 40
0026 CALL ENABLE(0)
0027 Q2=DEQUE(PING,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 CONTINUE
0033 ITERMX=XOUTBF(6)
0034 IF(ITERMX .EQ. 27 .AND. IXON .EQ. 1) GOTO 31
0035 IF(ITERMX .EQ. 28 .AND. IXON .EQ. 1) GOTO 31
0036 CALL COUT(LEN)
0037 IFR=IFR+1
0038 FREE(IFR)=Q2
0039 IF(IS .EQ. 1) GOTO 100
0040 IF(IDFLG .GT. 0) CALL HINPUI
0041 IF(LLFLG .GT. 0) CALL LINLOS
0042 IF(OUTFCT .EQ. 1) GOTO 100
0043 IF(IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
0044 IF(IRSEND .EQ. 1) GOTO 85
0045 IF(PXING(1) .LT. PXING(2)) GOTO 100
0046 CALL ENABLE(0)
0047 Q1=DEQUE(PXING,XING,1)
0048 CALL ENABLE(1)
0049 OUTQ=Q1
0050 CALL DESTR(PACK(254,OUTQ),Q2)
0051 DO 90 I=1,Q2
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061

```

```

0062 OUTBF(I)=PACK(I,OUTQ)
0063 CONTINUE
0064 OUTBF(Q2+1)=0
0065 IP1=OUTBF(5)
0066 OUTBF(Q2+2)=LIDFD(IP1)
0067 IF(Q2+2 .LT. 6 .OR. Q2+2 .GT. 255) GOTO 100
0068 IFULL=1
0069 ISENT=1
0070 CALL LPQOUT(Q2+2)
0071 IRSEND=1
0072 ACKTIM=0
0073 IWRTIM=0
0074 IWRT=1
0075 INFLECT=0
0076 OLDTIM=NEWTIM
0077 NEWTIM=LTIME
0078 IF(IWRT .EQ. 0) GOTO 120
0079 IWRTIM=IWRTIM+(NEWTIM-OLDTIM)
0080 IF(IWRTIM .LT. TIMLM) GOTO 120
0081 CALL WTOKEN
0082 IWRTIM=0
0083 IWRT=0
0084 IF(ISENT .EQ. 0) GOTO 130
0085 ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0086 IF(ACKTIM .LT. ATIMLM) GOTO 130
0087 CALL ENABLE(O)
0088 CALL ACKNAK(O)
0089 CALL ENABLE(I)
0090 GOTO 5
0091 CONTINUE
0092 GOTO 5
0093 END

```

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

```

0001 FUNCTION IGETSF(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,IGLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0007 IF (IFR .LT. 1) CALL INIT
0008 IGETSF=FREE(IFR)
0009 IFR=IFR-1
0010 RETURN
0011 END
0012

```

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

```

0001 SUBROUTINE ENQUE(A,B,N)
0002 LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003 INTEGER*2 XINQ,FXINQ,XOUTQ,ACKQ,PACKQ,INQ,FINQ
0004 INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005 INTEGER*2 RESNLM,OUTFCT
0006 COMMON /QUE/ XINQ(16),FXINQ(2),XOUTQ(16),FXOUTQ(2),
1 ACKQ(16),PACKQ(2),INQ(16),FINQ(2)
1 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,IQLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRI,ISLID,MONTOR,LF,ETX,CR,DUM
10 IHEAD=A(1)
IQTAIL=A(2)
IF(IQTAIL.EQ.1) GOTO 20
IQTAIL=IQTAIL-1
B(IQTAIL)=N
A(2)=IQTAIL
GOTO 999
20 IF(IHEAD.GE.IQLNTH) GOTO 40
NN=IHEAD-IQTAIL
DO 30 I=1,NN+1
0020 B(IQLNTH+1-I)=B(IHEAD+1-I)
0021 A(1)=IQLNTH
A(2)=IQLNTH-NN
GOTO 10
0023 40 CALL INIT
0024 999 RETURN
0025 END
0026

```

```

0001 FUNCTION DEQUE(A,B,ID)
0002 LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTR
0003 INTEGER*2 XING,PXING,XOUTQ,ACAQ,PACKQ,INQ,FINQ
0004 INTEGER*2 FLWCNT,A(2),B(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005 COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),
1   ACQ(16),PACKQ(2),INQ(16),FINQ(2)
0006 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1   RESNLM,IALTRT,ISLID,MONTR,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,XING,FXING,FXOUTQ
0003 INTEGER*2 ACKQ,PACKQ,PING,FREE,OUTQ,XOUTQ
0004 INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005 LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0006 LOGICAL*1 PACK,OUTBF,INBF,LCOMT(40)
0007 INTEGER*2 LTIME,IWRITM
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,IDLNT,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTR,ISLID,MONTR,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),IWRITM,STAT,LTIME
0015 COMMON /PACK(256,64)
0016 COMMON /QUE/ XING(16),FXING(2),XOUTQ(16),FXOUTQ(2),ACKQ(16),
1 COMMON /FRE/ FREE(64),IFR,IFRSZ
0017 COMMON /MESS/ MESSEQ
0018 IF (N.NE. 1) GOTO 10
0019 IFR=IFR+1
0020 DO 5 I=1,256
0021 PACK(I,OUTQ)=0
0022 5 CONTINUE
0023 IFULL=0
0024 ISENT=0
0025 FREE(IFR)=OUTQ
0026 IRSEND=0
0027 GOTO 999
0028 10 T2=PACK(256,OUTQ)
0029 T2=T2+1
0030 IF(T2.LE. RESNLM) GOJP 120
0031 T3=PACK(3,OUTQ)
0032 T4=MOD(T3,64)
0033 IF(T4.LT. 32) GOTO 110
0034 DO 20 I=7,60
0035 XOUTBF(I)=*40
0036 CONTINUE
0037 DO 30 I=7,9
0038 XOUTBF(I)=LF
0039 CONTINUE
0040 DO 40 I=1,8
0041 XOUTBF(I+9)=LCOMT(I)
0042 CONTINUE
0043 DO 50 I=9,16
0044 XOUTBF(I+9)=LCOMT(I)
0045 CONTINUE
0046 DO 60 I=17,22
0047 XOUTBF(I+9)=LCOMT(I)
0048 CONTINUE
0049 DO 70 I=25,29
0050 XOUTBF(I+9)=LCOMT(I)
0051 CONTINUE
0052 DO 70 I=25,29
0053

```

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

FORTRAN IV V02.1-1 Tue 05-Jun-79 13:31:46

```

0054 XOUTBF(I+7)=LCOMT(I)
0055 CONTINUE
0056 ENCODE(3,80,XOUTBF(38)) ISLID
0057 FORMAT(I3)
0058 DO 90 I=33,40
0059 XOUTBF(I+11)=LCOMT(I)
0060 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=1,556
0068 PACK(I,OUTQ)=0
0069 CONTINUE
0070 ISENT=0
0071 IFULL=0
0072 IMRT=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(PXING,XING,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(PXING,XING,OUTQ)
0091 RETURN
0092 END

```

PAGE 001

Tue 05-Jun-79 13:32:06

```

SUBROUTINE IMPTOIL)
INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
INTEGER*2 /ACKQ,PINQ,FLWCNT,OUTFCT,T1,RESNLM,OUTQ
LOGICAL*41 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
LOGICAL*41 FACK,MONTR,ISLID,ETX,CR,LF,DUM
INTEGER*2 LTIME,IWRTM,IXON
COMMON /DFM/ OUTBF(256),INBF(256),IWRTM,IWRT,STAT,LTIME
COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
LIDFD(256),IOFLG,LLFLG,IRSEND
COMMON /PACK(256,64)
COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
RESNLM,IALTRT,ISLID,MONTR,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 EMSTR(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
0053 END

```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 13:32:22      PAGE 001

0001      SUBROUTINE LPINFT(LI)
0002      INTEGER*2 STAT,FLWCNT,OUTFCT,OUTO,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,IWRITM
0006      COMMON/DFM/ OUTBF(256),INBF(256),IWRITM,IWRT,STAT,LTIME
0007      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTO,
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      CC7=MOD(CC1,2)
0025      IF(CC7 .GE. 1) GOTO 30
0027      CALL ACKNAK(0)
0028      GOTO 99
0029      CALL ACKNAK(1)
0030      GOTO 99
0031      CALL INPTQ(LI)
0032      CONTINUE
0033      RETURN
0034      END

```

FORTRAN IV V02.1-1 Tue 05-Jun-79 13:32:37 PAGE 001

```

0001 SUBROUTINE INIT
0002 INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,FINQ,FREE
0003 INTEGER*2 STAT,FLWCNT,OUTFCT
0004 INTEGER*2 SETPRM,RSTPRM,SETBKP,OUTQ,RESNLM,RSTBKP
0005 LOGICAL*1 DUM,IBFO,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/IFM/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/ MESSED
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 /SMT/ 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 MESSED=0
0022 LLFLG=0
0023 INFLECT=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 LIDFD(I)=4
0048 LIDFD(21)=1
0049 LIDFD(22)=3
0050 LIDFD(23)=6
0051 LIDFD(24)=5

```

40

PAGE 002

F0RTRAN IV V02.1-1 Tue 05-Jun-79 13:32:137

```

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

```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 13:32:59      PAGE 001

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,MONTR,ISLID,DUM
0007      LOGICAL*1 M1(40),LINO(8),LINI(8)
0008      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0009      1 RESNLM,IALTRY,ISLID,MONTR,LF,ETX,CR,DUM
0010      COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
0011      1 LIDFD(256),IOFLG,LLFLG,IRSEND
0012      COMMON /MESS/ MESSQ
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 25 '//
0017      EQUIVALENCE(LINO,LIN08)
0018      EQUIVALENCE(LINI,LIN18)
0019      IS=LLFLG
0020      DO 20 I=7,9
0021      XOUTBF(I)=LF
0022      CONTINUE
0023      DO 30 I=1,22
0024      XOUTBF(I+9)=M1(I)
0025      IF (IS .EQ. 1) GOTO 50
0026      DO 40 I=1,8
0027      XOUTBF(I+31)=LINO(I)
0028      GOTO 70
0029      DO 60 I=1,8
0030      XOUTBF(I+31)=LINI(I)
0031      DO 80 I=23,40
0032      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

```

PAGE 001

V02.1-1 Tue 05-Jun-79 13:31:21

FORTRAN IV

```

0001 SUBROUTINE HINPUT
0002 INTEGER*2 XINQ,XOUTQ,ACKQ,FXINQ,FXOUTQ,PACKQ,FINQ
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),FINQ(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

```

```

1 .TITLE FDM.MACRO
2 .SRITL NODE25
3 .IDENT /V3.0/
4 .GLOBL LIJINT,LID,ENABLE,SWITCH,MASTER,TIME
5 .GLOBL WTOKEN,RAM,STATUS,STATBO
6 .GLOBL LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINFT
7 .NLIST CND
8 .PSECT
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

```

000000 R0 =Z0
000001 R1 =Z1
000002 R2 =Z2
000003 R3 =Z3
000004 R4 =Z4
000005 R5 =Z5
000006 SP =Z6
000007 PC =Z7

```

! 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 VARIABLES

```

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

```

MACRO V03.01 5-JUN-79 13:33:26 PAGE 1-1

FDM.MACRO  
NODE25

66  
67  
68

000002  
000001

BIT01= 2  
BIT00= 1

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 MASTER: MOV    $340,  RO
7           MTPS    RO
8           MOV    $124,  RO
9           MOV    $L10, (RO)+
10          MOV    $340,  (RO)
11          RESET
12          MOV    $60,   RO
13          MOV    $C10, (RO)+
14          MOV    $340, (RO)+
15          RIS    @#HRCSR
16          MOV    $100, RO
17          MOV    $TIME, (RO)
18          RIC    $A0100, @#CSR
19          JSR    PC, LIUINT
20          CLR    @#CSR
21          MOV    $000,  RO
22          MTPS   RO
23          RTS    PC
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 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****

```

```

97 000234 004767 000052 JSR ZER0BF
98 000240 012737 001400 172416 MOV #1400, @#DFREG
99 000246 105737 172414 TSTB @#CSR
100 000252 100375 BPL -4
101 000254 105037 172414 CLRB @#CSR
102 000260 012701 010510 MOV #10510, R1
103 000264 012700 010540 MOV #10540, R0
104 000270 004767 000016 JSR PC, ZER0BF
105 000274 012701 010710 MOV #10710, R1
106 000300 012700 010740 MOV #10740, R0
107 000304 004767 000002 JSR PC, ZER0BF
108 000310 000440 ER STAT$
109 000312 010137 172416 ZER0BF: MOV R1, @#DFREG
110 000316 012737 001400 172416 JSR #1400, @#DFREG
111 000324 105737 172414 TSTB @#CSR
112 000330 100375 BPL -4
113 000332 105037 172414 CLRB @#CSR
114 000336 013702 172416 MOV @#I0BUF, R2
115 000342 042702 177400 BIC #177400, R2
116 000346 010037 172416 MOV R0, @#DFREG
117 000352 022702 000000 CMP #0, R2
118 000356 001412 BEQ 3$
119 000360 012737 001400 172416 MOV #1400, @#DFREG
120 000366 105737 172414 TSTB @#CSR
121 000372 100375 BPL -4
122 000374 105037 172414 CLRB @#CSR
123 000400 005302 DEC R2
124 000402 000763 ER 2$
125 000404 010037 172416 MOV R0, @#DFREG
126 000410 000207 RTS PC
127
128
129
130 000412 012737 010400 172416 STAT$: MOV #432, @#DFREG
131 000420 012737 002400 172416 MOV #1280, @#DFREG
132 000426 012737 002400 172416 MOV #1280, @#DFREG
133 000434 000207 RTS PC
134
135

;POINTER=0
;FALSE READ DATA
;GOOD RD
;NO RETRY
;CLEAR DONE BIT
;RDBUFADR COMMAND
;SEL OUTBUFO COMMAND
;RDBUFADR COMMAND
;SEL OUTBUFO COMMAND
;ZERO BUFFER POINTER
;GO CLEAR STATUS
;RDBUFADR
;RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;FETCH POINTER
;CLEAR MST BYTE
;SEL BUFFER
;POINTER=0
;YES RECHECK
;NO FALSE RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;LOCAL POINTER-1
;BR UNTIL 0
;SEL BUFFER
;RETURN

;MCR:RS
;RS(FALSE)
;RS

```

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 000436 010046 L10: MOV R0, -(SP) ;SAVE R0
7 000440 010146 MOV R1, -(SP) ;SAVE R1
8 000442 010246 MOV R2, -(SP) ;SAVE R2
9 000444 010346 MOV R3, -(SP) ;SAVE R3
10 000446 010446 MOV R4, -(SP) ;SAVE R4
11 000450 010546 MOV R5, -(SP) ;SAVE R5
12 000452 005737 172414 TST @#CSR ;? LIU INTERRUPTED
13 000456 100153 SPL RTI$ ;NO
14 000460 042737 040000 172414 BIC #BIT14, @#CSR ;YES/DISABLE LIU
15 000466 012737 010400 172416 MOV #4352, @#OFREG ;WCR : RS(0)
16 000474 012737 002400 172416 MOV #1250, @#OFREG ;RD
17 000502 013767 172416 000502 MOV @#IOBUF, CAUSE
18 000510 042767 177400 000474 BIC #177400, CAUSE ;CLEAR UNUSED BITS
19
20
21
22 000516 132767 000004 000466 B2$: BITB #BIT02, CAUSE ;OV-FL
23 000524 001406 BEQ B3$ ;RDBUFADR INO
24 000526 012702 010410 MOV #4360, R2 ;SEL INBUFO
25 000532 012703 010440 MOV #4384, R3 ;GO EMPTY
26 000536 004767 000270 JSR PC, EMBF ;OV-FL
27 000542 132767 000010 000442 B3$: BITB #BIT03, CAUSE ;RDBUFADR IN1
28 000550 001406 BEQ B4$ ;SEL INBUF1
29 000552 012702 010610 MOV #4488, R2 ;GO EMPTY
30 000556 012703 010640 MOV #4512, R3 ;LINE-LOSS PRIMARY
31 000562 004767 000244 JSR PC, EMBF ;LATCH ON AROUND
32 000566 132767 000020 000416 B4$: BITB #BIT04, CAUSE ;YES LOOP AROUND
33 000574 001407 BEQ B5$ ;BIT07, CAUSE ;FLAG PRIMARY SW. FAIL
34 000576 132767 000200 000406 B7$: BITB #BIT05, CAUSE ;BACKUP LINE LOSS
35 000604 001403 BEQ B6$ ;2, LFLG ;FLAG BACKUP SW. FAIL
36 000606 012767 000002 002002' MOV #1, LFLG ;WRITE TOKEN DETECT
37 000614 132767 000040 000370 B5$: BITB #BIT06, CAUSE ;CLEAR TOKEN VAR
38 000622 001403 BEQ B6$ ;BIT07, CAUSE ;INBUFO FULL
39 000624 012767 000001 002002' MOV #1, LFLG ;NO
40 000632 132767 000100 000352 B6$: BITB #BIT06, CAUSE ;BUFFER ADDRESS
41 000640 001404 BEQ B0$ ;CLR IWRITM ;RDBUFADR COMM
42 000642 005067 001000' CLR IWRITM ;SEL BUF
43 000646 005067 001002' CLR IWRITM ;FCRC BIT
44
45
46
47 000652 132767 000001 000332 B0$: BITB #BIT00, CAUSE ;EMPTY BUFFER
48 000660 001423 BEQ B1$ ;INBF, R1 ;DATA LINK AREA
49 000662 012701 000400' MOV #INBF, R1 ;? VARIABE
50 000666 012702 010410 MOV #4360, R2 ;AREA, R5
51 000672 012703 010440 MOV #4384, R3 ;R5
52 000676 012704 000001 MOV #BIT00, R4 ;CRC OR BYTE COUNT
53 000702 004767 JSR PC, EMBF ;CALL FORTRAN QUE'ER
54 000706 012705 001214' MOV #01214, ;
55 000712 012767 000001 000274 MOV #1, AREA ;
56 000720 010267 000274 MOV #R2, DATA ;
57 000724 004767 JSR PC, LFINFT ;

```

PREFORM OPERATION DEPENDING ON BITS

```

58 000730 132767 000002 000254 B1$: RITE      #BIT01, CAUSE
59 000736 001423      MOV          RTI$
60 000740 012701 000400'   #INBF, R1
61 000744 012702 010610   #4488., R2
62 000750 012703 010640   #4512., R3
63 000754 012704 000002   #BIT01, R4
64 000760 004767 000046   FC,      EMBF
65 000764 012705 001214'   #AREA,  R5
66 000770 012767 000001   #1,      AREA
67 000776 010267 000216   R2,      DATA
68 001002 004767 00000006  JSR          LFINFT
69
70 001006 012605      MOV          (SP)+, R5
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   #BIT14, @#CSR
77 001030 000002      BIS          RTI
78
79 001032 012737      MOV          #4480., @#DFREG
80 001040 012737 002400   #1280., @#DFREG
81 001046 013700      MOV          @#IOBUF, R0
82 001052 130400      BITB       R4,R0
83 001054 001002      RNE        CRCOK
84 001056 012704      MOV          #-1,      R4
85 001062 010237 172416   MOV          R2,      @#DFREG
86 001066 012737 001400   #768.,    @#DFREG
87 001074 105737 172414   TSTB       @#CSR
88 001100 100375      BFL        -4
89 001102 013702      MOV          @#IOBUF, R2
90 001106 042702      RIC        #177400, R2
91 001112 010200      MOV          R2,      R0
92 001114 005400      NEG        R0
93 001116 010037      MOV          R0,      @#WCR
94 001122 010137      MOV          R1,      @#BAR
95 001126 010337      MOV          R3,      @#DFREG
96 001132 012737 001400   #768.,    @#DFREG
97 001140 105737 172414   TSTB       @#CSR
98 001144 100375      BFL        -4
99 001146 012737 021000   #8704.,  @#DFREG
100 001154 000240      NOP
101 001156 105737      TSTB       @#CSR
102 001162 100401      BMI        DMAOK
103 001164 000240      NOP
104 001166 012737 004400   #2304.,  @#DFREG
105 001174 105737 172414   TSTB       @#CSR
106 001200 100375      BFL        -4
107 001202 005704      R4
108 001204 100001      END$
109 001206 010402      MOV          R4,      R2
110 001210 000207      RTS
111 001212 000000      #WORD
112 001214 000000      #CAUSE:
113 001216 001220'   #AREA:    .WORD   DATA
114 001220 000000      #DATA:    .WORD   0

```

```

;INBUF1 FULL
;NO WERE DONE
;BUFFER ADDRESS
;RD,UFADR COMM
;SEL INBUF1
;CRC.BIT
;GO EMPTY BUFFER
;DATA LINK AREA
;ONE VARIABLE
;CRC OR BYTE COUNT
;CALL FORTRAN QUE'ER
;RESTORE REGISTERS

```

```

;ENABLE INTERRUPTS
;RETURN FROM INTERRUPT

```

```

;READ STATUS 1
;RS
;FETCH STATUS
;GOOD CRC

```

```

;NO FLAG
;RDBUFADR
;RD
;GOOD RD

```

```

;REPLACE WITH POINTER
;CLEAR MST BITS
;SAVE IT
;2'S COMP

```

```

;BYTE COUNT NOW
;ADDRESS IN MEMORY
;SEL BUFFER
;FALSE RD
;DONE ON
;NO LOOP
;FIRE DMA
;DELAY
;GOOD DMA

```

```

;ERROR IF HERE
;FALSE WD
;GOOD WD

```

```

;WAS CRC OK
;YES
;NO FLAG IT
;RETURN
;STATUS BYTE 0 HOLDER

```

```

1  .ENABLE LSB
2  *****
3  *****
4  ***** PROCEDURE ENABLE ***** *
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 *****

        .WHICH SUBROU.
        ;DISABLE LIU INTERRUPTS
        ;ENABLE LIU

        ;FORTRAN ENCODE/DECODE MACRO
        ENSTR:  MOV    @ARG1(R5), R1
                CMP    #1, R1
                BEQ    E1$
                RIC    #BIT14, @#CSR
                RTS    PC
        DEST:   MOV    @ARG1(R5), @ARG2(R5)
                RTS    PC

        ;SWITCH
        SWITCH: MOV    @ARG1(R5), R1
                MOV    #4368., @#DFREG
                ADD    #2304., R1
                MOV    R1, @#DFREG
                TSTB  @#CSR
                BPL   .-4
                CLRB @#CSR
                RTS   PC

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

        ;ENCODE BYTE COUNT
        ;DECODE BYTE COUNT

        ;SM SET
        ;MCR : MODSTAT
        ;RD
        ;VALID WRITE
        ;NO LOOP UNTIL READY
        ;CLEAR DONE.BIT

```

```

1 *****
2 *****
3 ***** PROCEDURE STATUS *****
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 *****

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

STATUS: MOV @ARG1(R5), R1
CMP #1, R1
BEQ 2$
MOV #4352., @#0FREG
MOV #1280., @#0PREG
MOV @#IOBUF, R0
BIC #177400, R0
MOV R0, @ARG2(R5)
RTS PC
MOV #4480., @#0FREG
MOV #1280., @#0PREG
MOV @#IOBUF, R0
BIC #177400, R0
MOV R0, @ARG2(R5)
RTS PC
MOV #4352., @#0FREG
MOV #1280., @#0PREG
MOV @#IOBUF, R1
BIC #177400, R1
BIT #BIT03, R1
BEQ 3$
MOV #1, @ARG1(R5)
RTS PC
MOV #0, @ARG1(R5)
RTS PC

;WHICH BYTE
;COMPARE
;WCR : RS(0)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;WCR : RS(1)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;STATUS BYTE 1
;RS
;FETCH DATA
;BUFFER FULL
;LOAD RETURN
;LOAD RETURN

```

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



FDM.MACRO MACRO V03.01 5-JUN-79 13:33:26 PAGE 6

NODE25

```

1 *****
2 ***** PROCEDURE WRITE TOKEN (WTKEN) *****
3 *****
4 *****
5 *****
6 ;CALL WTKEN -LOADS AND WRITES A TOKEN TO LOOP
7
8 ;READ STATUS(1)
9 ;RS
10 ;FETCH DATA
11 ;BUFFER LOADED
12 ;NO GENERATE TOKEN
13 ;MODSTAT
14 ;MARK OBOF/OBIF/WTKMD
15 ;VALID WRITE
16 ;NO LOOP UNTIL READY
17 ;CLEAR DONE BIT
18
19 ;SELECT BUFFER 1
20 ;ADDRESS OF WRITE TOKEN
21 ;# OF BYTES
22 ;FORMAT IT
23 ;DMA BYTE COUNT
24 ;FIRE DMA
25 ;DELAY
26 ;GOOD DMA
27 ;ERROR IF HERE
28 ;CLEAR DONE
29 ;MODSTAT
30 ;MARK OBIF/WTKMD
31 ;VALID WRITE
32 ;NO LOOP UNTIL READY
33 ;CLEAR DONE BIT
34 ;RETURN
35 ;A WTKEN
36
37
38
39
40
41 *****
42 ***** PROCEDURE TIMER (TIME) *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****

```

LINE	ADDRESS	OPERATION	DATA	TIME	INC	RTI
7	001502	012737	010600	172416	WTKEN:	MOV
8	001510	012737	002400	172416		MOV
9	001516	013701	172416			MOV
10	001522	132701	000010			BITB
11	001526	001414				REQ
12	001530	012737	010420	172416		MOV
13	001536	012737	004423	172416		MOV
14	001544	105737	172414			TSTB
15	001550	100375				BPL
16	001552	105037	172414			CLRB
17	001556	000207				RTS
18	001560	012737	010740	172416	WTKG:	MOV
19	001566	012737	001660	172410		MOV
20	001574	012702	000002			MOV
21	001600	005402				NEG
22	001602	010237	172412			MOV
23	001606	012737	024000	172416		MOV
24	001614	000240				NOP
25	001616	105737	172414			TSTB
26	001622	000240				NOP
27	001624	105037	172414			CLRB
28	001630	012737	010420	172416		MOV
29	001636	012737	004422	172416		MOV
30	001644	105737	172414			TSTB
31	001650	100375				BPL
32	001652	105037	172414			CLRB
33	001656	000207				RTS
34	001660	000	377			FC
35						.BYTE 0,377
36						
37						
38						
39						
40						
41	001662	005267	001006'		TIME:	INC
42	001666	022767	077777	001006'	CMP	#077777, LTIME
43	001674	001002		4#	BNE	4#
44	001676	005067	001006'		CLR	LTIME
45	001702	000002				RTI
46						
47						
48						

```

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 *****
49 *****

*****
***** 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 =WOKEN

RAM:  MOV @ARG1(R5), R0
      CNP @R0, R0
      RDRAM
      MOV @ARG2(R5), R0
      WTRAM:  MOV @ARG3(R5), R1
              MOV @4354., @OPREG
              ADD @2304., R0
              TSTB @#CSR
              BPL -4
              CLRFB @#CSR
              MOV @4353., @OPREG
              ADD @2304., R1
              MOV R1, @OPREG
              TSTB @#CSR
              BPL -4
              CLRFB @#CSR
              RTS PC
RDRAM:  MOV @ARG2(R5), R0
         MOV @4354., @OPREG
         ADD @2304., R0
         MOV R0, @OPREG
         TSTB @#CSR
         BPL -4
         CLRFB @#CSR
         MOV @4353., @OPREG
         MOV @768., @OPREG
         TSTB @#CSR
         BPL -4
         CLRFB @#CSR
         MOV @#IOBUF, R0
         MOV @17760, R0
         BIC @17760, R0
         MOV R0, @ARG3(R5)
         RTS PC

; RAM
; WHICH OPERATION
; READ OP
; ADDRESS
; WRITE DATA
; SEL LDACR
; WD/DATA (ADDR)
; WRITE DATA
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; WD/DATA (CMD)
; WRITE
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; ADDRESS TO READ
; SEL LDACR
; WD/DATA (ADDR)
; WRITE DATA
; VALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; READ DATA
; VALID READ
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; FETCH DATA
; CLEAR BITS
; RTN DATA

```

NODE25

```

1          .ENABLE LSR
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         *****

          LFOUT:  MOV  @ARG1(R5), R2
                   MOV  #340, R0
                   MTPS R0
                   MOV  #OUTBF, @#BAR
                   NEG  R2
                   MOV  @#WCR, R2
                   MOV  #4448., @#OPREG
                   MOV  #10240., @#OPREG
                   NOP
                   TSTB @#CSR
                   NOP
                   CLR  @#CSR
                   MOV  #4576., @#OPREG
                   MOV  #WTD, @#BAR
                   MOV  #2, R2
                   NEG  R2
                   MOV  @#WCR, R2
                   MOV  #10240., @#OPREG
                   NOP
                   TSTB @#CSR
                   NOP
                   CLR  @#CSR
                   MOV  #4368., @#OPREG
                   MOV  #2307., @#OPREG
                   TSTB @#CSR
                   BPL  ,-4
                   CLR  @#CSR
                   MOV  #000, R0
                   MTPS R0
                   RTS

          ;BYTE COUNT
          ;PRI=7
          ;BUS ADDRESS
          ;2'S COMP COUNT
          ;COUNT
          ;JOB COMMAND
          ;DMA GO
          ;INTERFACE TIME
          ;DMA OK
          ;ERROR IF HERE
          ;JOB1 COMMAND
          ;ADDRESS OF WRITE TOKEN
          ;BYTE COUNT
          ;FORMAT FOR BLIUI
          ;DMA BYTE COUNT
          ;FIRE DMA
          ;DELAY
          ;GOOD DMA
          ;ERROR IF NOT
          ;CLEAR DONE
          ;MODSTAT
          ;BUFFERS FULL
          ;GOOD WRITE
          ;NO LOOP UNTIL
          ;CLEAR DONE BIT
          ;PRI=0
          ;LOWER CPU

```

FDM-MACRO MACRO V03.01 5-JUN-79 13:33:26 PAGE 9  
NODE25

```

1 1 *****
2 2 *****
3 3 *****
4 4 *****
5 5 002274 017500 000002 @ARG1(R5), R0
6 6 002300 022700 000000 #0, R0
7 7 002304 001004 #1, R0
8 8 002306 012700 010410 #4360., R0
9 9 002312 000167 000046 #4, R0
10 10 002316 022700 000001 #1, R0
11 11 002322 001004 #2, R0
12 12 002324 012700 010610 #4488., R0
13 13 002330 000167 000030 #4, R0
14 14 002334 022700 000002 #2, R0
15 15 002340 001004 #3, R0
16 16 002342 012700 010510 #4424., R0
17 17 002346 000167 000012 #4, R0
18 18 002352 022700 000004 #4, R0
19 19 002356 001020 #5, R0
20 20 002360 012700 010710 #4552., R0
21 21 002364 010037 172416 R0, @#0PREG
22 22 002370 012737 001400 MOV #768., @#0PREG
23 23 002376 105737 172414 TSTB @#CSR
24 24 002402 100375 BFL #-4
25 25 002404 013700 MOV @#10BUF, R0
26 26 002410 042700 BIC #17400, R0
27 27 002414 010075 MOV R0, @ARG2(R5)
28 28 002420 000207 RTS PC

REPNT:
1$:
2$:
3$:
4$:
5$:

```

```

;FETCH COMMAND
;INBUFO ?
;NO
;RDBUFADR IN0
;INBUFO ?
;NO
;RDBUFADR IN1
;OUTBUFO ?
;NO
;RDBUFADR OUT0
;OUTBUFO ?
;NO RETURN
;RDBUFADR OUT1
;RDBUFADR
;RD
;READY
;NO LOOP UNTIL
;FETCH POINTER
;CLEAR MST
;RETURN POINTER

```

FIM.MACRO MACRO V03.01 5-JUN-79 13:33:26 PAGE 10  
NODE25

```

1 *****
2 *****
3 *****
4 *****
5 *****
6 *****
7 *****
8 *****
9 002422 017500 000002 RSTART: MOV @ARG1(R5), R0
10 002426 022700 000000 R50:  #0, R0
11 002432 001002 BNE R51
12 002434 000000 HALT
13 002436 000207 RTS FC
14 002440 022700 000001 RS1:  #1, R0
15 002444 001003 BNE R52
16 002446 013700 000040 G4:  @#40, R0
17 002452 000110 JMP (R0)
18 002454 022700 000002 RS2:  #2, R0
19 002460 001003 BNE RST
20 002462 013700 002472' MOV @#ROM, R0
21 002466 000110 JMP (R0)
22 002470 000207 RST:  RTS
23 002472 173000 ROM:  .WORD 173000
24
25
26

```

```

;MODE
;LOAD ADDRESS

;RESTART PROGRAM
;START ADDRESS
;LOAD MODE
;BOOT ADDRESS

```

```

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

```

```

1 7 002474 010046
2 8 002476 010146
3 9 002500 010246
4 10 002502 000454
5 11 002506 000400
6 12 002512 060001
7 13 002514 105737
8 14 002520 100134
9 15 002522 113702
10 16 002526 042702
11 17 002532 122702
12 18 002536 001015
13 19 002540 012767
14 20 002546 012737
15 21 002554 000240
16 22 002556 000240
17 23 002560 000240
18 24 002562 012737
19 25 002570 000510
20 26 002572 123702
21 27 002576 001007
22 28 002600 012767
23 29 002606 012737
24 30 002614 000476
25 31 002616 123702
26 32 002622 001007
27 33 002624 005300
28 34 002626 012767
29 35 002634 004767
30 36 002640 000464
31 37 002642 122702
32 38 002646 010222
33 39 002650 110221
34 40 002652 005200
35 41 002654 004767
36 42 002660 112721
37 43 002664 005200
38 44 002666 012767
39 45 002674 004767
40 46 002700 010067
41 47 002704 005000
42 48 002706 005067
43 49 002712 000437
44 50 002714 122702
45 51 002720 001017
46 52 002722 016700
47 53 002726 012767
48 54 002734 004767
49 55 002740 012767
50 56 002746 004767
51 57 002752 004767

      .ENABLE LSR
      .GLOBL HINFUT
*****
***** PROCEDURE CRT INTERFACE *****
*****
C10:  MOV R0, -(SP)
      MOV R1, -(SP)
      MOV R2, -(SP)
      MOV CPBUF, R0
      MOV XINBF, R1
      MOV R0, R1
      TSTB @#HRCSR
      BPL 7$
      @#HREUF, R2
      RIC @177600, R2
      CMFB @022, R2
      BNE 20$
      MOV @0, IXON
      MOV @007, @#HXBUF
      NOP
      NOP
      MOV @007, @#HXBUF
      BR 7$
      CMFB @023, R2
      BNE 2$
      MOV @1, IXON
      MOV @007, @#HXBUF
      BR 7$
      CMFB @177, R2
      BNE 3$
      DEC R0
      MOV @134, CHAR
      JSR FC, ECHO
      BR 7$
      CMFB @015, R2
      BNE 4$
      MOVB R2, (R1)+
      INC JSR FC, CRLF
      MOVB @012, (R1)+
      INC R0
      MOV @007, CHAR
      JSR FC, ECHO
      MOV R0, IOFLG
      CLR R0
      CLR CLP
      BR 7$
      CMFB @025, R2
      BNE 5$
      MOV CLP, R0
      MOV @136, CHAR
      JSR FC, ECHO
      MOV @007, CHAR
      JSR FC, ECHO
      MOV @0072, CHAR
      JSR FC, ECHO
      MOV @00104, CHAR
      JSR FC, CRLF

;SAVE R0
;SAVE R1
;SAVE R2
;BUFFER POINTER
;BUFFER ADDRESS
;CHAR OFFSET
;CHAR READY

;CLEAR PAR.
;TURN ON CRT(REPORT)

;IT'S ON

;TURN OFF CRT(REPORT)
;IT'S OFF

;DEL CHAR
;POINTER-1

;STORE CHAR
;POINTER+1
;PRINT CHAR
;LF CHAR

;ATT PACKET READY
;RING BELL
;BYTE COUNT

;ERASE LINE CHAR

;RESTORE LINE POINTER
;PRINT ~
;PRINT U
;PRINT CR LF

```



```
1 *****
2 *****
3 *****
4 *****
5 000000 .PSECT IFM,RW,D,GBL,REL,OVR
6
7 000000 .OUTRF: .BLKB 256.
8 000400 .INBF: .BLKB 256.
9 001000 .IWRTH: .BLKW
10 001002 .IMRT: .BLKW
11 001004 .STAT: .BLKW
12 001006 .LTIME: .BLKW
13 .EVEN
14
15 000000 .PSECT BUFS,RW,D,GBL,REL,OVR
16
17 000000 .XOUTBF: .BLKB 256.
18 000400 .XINBF: .BLKB 256.
19 001000 .ACKSED: .BLKB 256.
20 001400 .LIDFD: .BLKB 256.
21 002000 .IOFLG: .BLKW
22 002002 .LLFLG: .BLKW
23 002004 .IRSEND: .BLKW
24
25
26 000000 .PSECT CRT,RW,D,GBL,REL,OVR
27
28 000000 .IXON: .BLKW
29 .EVEN
30
31
32 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	003	RTI\$	001006R
ARG2	= 000004		B5\$	000614R	END\$	001210R	LLFLG	002002R	003	STAT	001004R
ARG3	= 000006		B6\$	000632R	ENSTR	001254RG	LFINPT=	***** G		STATUS	001430RG
ARG4	= 000010		B7\$	000576R	E0\$	001234R	LFOUT	002110RG	002	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	DFREG	= 172416		WTKG	001560R
BIT05	= 000040		CRCOK	001062R	HXCSR =	177564	OUTBF	000000R	002	WTKEN	001502RG
BIT06	= 000100		CRLF	003062R	INBF	000400R	002	RAM	001704RG	WTRAM	001716R
BIT07	= 000200		CSR	= 172414	IOBUF =	172416	KDPNT	002274RG		XCIO	= 000000
BIT08	= 000400		DATA	001220R	IOFLG	002000R	003	KDRAM	002010R	XGIO	= 000001
BIT14	= 040000		DESTR	001264RG	IRSEND	002004R	003	KOM	002472R	XINBF	000400R
BIT15	= 100000		DHOST	= 000001	IWRT	001002R	002	RST	002470R	XOUTBF	000000R
B0\$	000652R		DMADK	001166R	IWRTH	001000R	002	RSTART	002422RG	XSID	= 000001
B1\$	000730R		DUMP	003166R	IXON	000000R	004	RSO	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

F 604 001

12:00:00

01 .00 .79

0000 0000

0000 0000

0000 0000

FOR BUDGET - R00 198  
FOR BUDGET - R00 198  
0000 0000



PAGE 001

12100100

01 JOB 79

100.25.000

PROGRAM:

decmp110f

ASSTGR DCO: DKS:  
 R LINE  
 DX1:R000.25,DX1:MDP=DX1:MD000 / 1 / 1.4  
 DX1:FB07C  
 DX1:MD000 1 / 1  
 451DK1

ASSTGR DX1: DKS:

## 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 (B RTP)
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 outside 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.

FORTRAN IV 002.1-11 Mon 01-Oct-79 09:20:30 PAGE 001

```

0001 PROGRAM NOIAL
0002 INTEGER*2 XINQ,XOUTQ,ACKQ,FXINQ,FXOUTQ,PACKQ
0003 INTEGER*2 PINQ,FREE,STAT,FLWNT,OUTFCT
0004 INTEGER*2 SETPRM,RSTPRM,SETBKP
0005 INTEGER*2 RSTBKP,Q1,Q2,RESNM,OUTQ,DEQUE
0006 LOGICAL*1 YOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INPF
0007 LOGICAL*1 IDATA
0008 LOGICAL*1 PACK,RTX,CR,LF,MONTR,ISLID,IUM
0009 INTEGER*2 LTIME,ACKTIM,NE*TIM,OLDTIM,IWRTT*,TIMLIM,ATIMLM
0010 REAL*4 RH,RL,AH,AL,VMEAS
0011 COMMON /MESS/ MESSIC
0012 COMMON /DFM/ OUTBF(256),INBF(256),
1 IWRTT*,IWR*STAT,LTIME
0013 1 COMMON /QUE/ XINQ(16),PXINQ(2),YOUTQ(16),PKOUTQ(2),
1 COMMON /BUPS/ XOUTB(256),XINBF(256),ACKSEQ(256),
1 LIDFD(256),IOFLG,LLFLG,IRSEND
0015 1 COMMON /FR/ FREE(64),IFR,IFRSZ
0016 COMMON /TIM/ OLDTIP,TIMLIM,ACKTIM,ATIMLM
0017 COMMON /PACK(256,64)
0018 COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
0019 COMMON /GLOB/ ISENT,FLWNT,IGINTB,OUTFCT,IFULL,OUTQ,
1 RESNM,IALTR,ISLID,MONTR,LF,RTX,CR,DUM
0020 5 ITERM,LINK,MON,VMEAS(13),ISUB,INUM,IDATA(10),
5 ICHNSW
0021 CALL MASTER
0022 CALL INIT
0023 CALL ENABLE(1)
0024 CALL READY
0025 5 IF(PINQ(1)).LT.PINQ(2)) GOTO 40
0027 CALL ENABLE(0)
0028 QZ=DEQUE(PINQ,INQ,1)
0029 CALL ENABLE(1)
0030 DO 30 I=1,10
0031 IDATA(I)=PACK(I+6,Q2)
0032 CONTINUE
0033 CALL BWSALP
0034 IFR=IFR+1
0035 FREE(IFR)=Q2
0036 CALL STATE0(IIS)
0037 IF(IIS.EQ.1) GOTO 100
0039 IF(IOFLG.GT.2.AND.IRSEND.EQ.0) CALL EWBSA
0041 IF(LLFLG.GT.0) CALL LINLOS
0043 IF(OUTFCT.EQ.1) GOTO 100
0045 IF((IFULL.EQ.1).OR.(ISENT.EC.1)) GOTO 120
0047 IF(IRSEND.EC.1) GOTO 85
0049 IF(PXINQ(1)).LT.PXINQ(2)) GOTO 100
0051 CALL ENABLE(0)
0052 QI=DEQUE(PXINQ,XINQ,1)
0053 CALL ENABLE(1)
0054 OUTQ=Q1
0055 85 CALL DESTP(PACK(254,OUTQ),Q2)
0056 DO 90 I=1,Q2

```

PAGE 022

FORTRAN IV V02.1-11 Mon 01-Oct-79 09:00:30

```

0057   OUTBF(I)=PACK(I,OUTQ)
0058   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   IWRTIM=0
0068   IWRT=1
0069   INFLCT=0
0070   OLDTIM=NEWTIM
0071   NEWTIM=LTIME
0072   IF(IWRT.EQ.0) GOTO 120
0073   IWRTM=IWRTM+(NEWTIM-OLDTIM)
0074   IF(IWRTM.LT. TIMLIN) GOTO 120
0075   CALL WTKEN
0076   IWRTM=0
0077   IWRT=0
0078   IF(ISENT.EQ.0) GOTO 130
0079   ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0080   IF(ACKTIM.LT. ATIMLM) GOTO 130
0081   CALL ENABLE(0)
0082   CALL ACKNAK(0)
0083   CALL ENABLE(1)
0084   CONTINUE
0085   GOTO 5
0086   END

```

PAGE 001

V02.1-11 Mon 01-Oct-79 09:00:38

FORTRAN IV

```
0001 FUNCTION IGETSP(N)  
0002 LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM  
0003 INTEGER*2 FREE,FLWNT  
0004 INTEGER*2 OUTFCT,OUTQ,RESNLM  
0005 COMMON /PRE/ FREE(64),IFR,IFPSZ  
0006 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM  
1 IF(IFR.LT.1) CALL INIT  
0007 IGETSP=FREE(IFR)  
0008 IFR=IFR-1  
0009 RETURN  
0010  
0011  
0012 END
```

```

FORTRAN IV      V02.1-11   Mon 01-Oct-79 09:00:41      PAGE 001

0001  SUBROUTINE ENQUE(A,B,N)
0002  LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003  INTEGER*2 XINQ,PXINQ,XOUTQ,ACKQ,PACKQ,INO,PINQ
0004  INTEGER*2 FLAG,OUTC,A(2),E(16)
0005  INTEGER*2 RESNM,OUTFCT
0006  COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
0007  COMMON /GLOB/ ISENT,FLWONT,IQLNTH,OUTFCT,IFULL,OUTQ,
0008  RESNM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0009  IQHEAD=A(1)
0010  IQTAIL=A(2)
0011  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
0017  NN=IQHEAD-IQTAIL
0018  DO 30 I=1,NN+1
0019  B(IQLNTH+1-I)=B(IQHEAD+1-I)
0020  A(1)=IQLNTH
0021  A(2)=IQLNTH-NN
0022  GOTO 10
0023  40  CALL INIT
0024  999  RETURN
0025  END
0026

```

```

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,MONITOR
0003      INTEGER*2 XING,PXING,XOUTQ,XOUTQ,ACKQ,PACKQ,INQ,PING
0004      INTEGER*2 FLACNT,A(2),B(16),LEJCE,OUTFC,OUTQ,RESNLM
0005      COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),
0006      1      ACKQ(16),PACKQ(2),INQ(16),PING(2)
0007      1      COMMON /GLOB/ ISENT,FLACNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0008      1      RESNLM,IALTHT,ISLID,MONITOR,LF,ETX,CR,DUM
0009      ICHHEAD=A(1)
0010      DEQUE=B(ICHHEAD)
0011      IF(ID.NE.1) GOTO 999
0012      IF(ICHHEAD.NE.0) GOTO 10
0013      A(1)=IQLNTH
0014      A(2)=IQLNTH+1
0015      GOTO 999
0016      10      A(1)=ICHHEAD-1
0017      999      RETURN
0018      END

```

```

XOFRTRAN IV      V02.1-11      Mon 01-Oct-79 09:00:51      PAGE 001

0001      SUBROUTINE ACKMAX(N)
0002      INTEGER*2 FLWNT,OUTFCT,STAT,XINC,PXINC,PXOUTQ
0003      INTEGER*2 ACKQ,PACKQ,PINQ,FREE,OUTQ,XOUTQ,FINDPK
0004      INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005      LOGICAL*1 FTX,CR,LF,MONTOR,ISLID,LUM
0006      LOGICAL*1 PACK,OUTEF,INBF,LCOMT(40)
0007      LOGICAL*1 XOUTEF,XINBF,ACKSEQ,LIDFD
0008      INTEGER*2 LTIME,IWRTM
0009      REAL*8 COMT(5)
0010      EQUIVALENCE(COMT,LCOMT)
0011      DATA COMT(1)/'MESSAGE',/COMT(2)/'NOT SENT',COMT(3)/' FROM',/
0012      DATA COMT(4)/'NODE',/COMT(5)/' TO NODE',/
0013      COMMON /GLOB/ ISENT,FLWNT,FLQNT,IQLNIB,OUTFCT,IFULL,OUTQ,
0014      RESNLM,IALIRT,ISLID,MONTOR,LF,ETX,CR,DUM
0015      COMMON /DPM/ OUTBF(256),INBF(256),
0016      IWRTM,IWRT,STAT,LTIME
0017      COMMON /QUE/ XINQ(16),PXINC(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
0018      COMMON /PRE/ FREE(64),IFR,IFRSZ
0019      COMMON /MESS/ MESSEQ
0020      COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
0021      IF (N.NE.1) GOTO 10
0022      IFR=IFR+1
0023      DO 5 I=1,256
0024      PACK(I,OUTQ)=0
0025      IFULL=0
0026      ISENT=0
0027      IRSEND=0
0028      FREE(IFR)=OUTQ
0029      GOTO 999
0030      T2=PACK(256,OUTQ)
0031      T2=T2+1
0032      IF(T2.LE.RESNLM) GOTO 120
0033      T3=PACK(3,OUTQ)
0034      T4=MOD(T3,64)
0035      IF(T4.LT.32) GOTO 110
0036      T5=IGETSP(N)
0037      IF(MESSEQ.EQ.126) MESSEQ=0
0038      MESSEQ=MESSEQ+1
0039      PACK(1,T5)=0
0040      PACK(2,T5)=MESSEQ
0041      PACK(3,T5)=0
0042      PACK(4,T5)=0
0043      PACK(5,T5)=25
0044      PACK(6,T5)=ISLID
0045      DO 20 I=7,60
0046      PACK(I,T5)="040"
0047      CONTINUE
0048      DO 30 I=7,9
0049      PACK(I,T5)=LF
0050      CONTINUE
0051      DO 40 I=7,9
0052      PACK(I,T5)=LF
0053      CONTINUE

```

PAGE 002

FORTRAN IV V02.1-11 Mon 01-Oct-79 09:00:51

```

0054 DO 40 I=1,8
0055   PACK(I+9,T5)=LCOMT(I)
0056 40 CONTINUE
0057 DO 50 I=9,15
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(3,00,PACK(30,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),50)
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
0093 CALL ENQUE(PXINQ,XINQ,OUTQ)
0094 GOTO 999
0095 120 PACK(256,OUTQ)=T2
0096 IFULL=0
0097 0097 ISENT=0
0098 IF(IRSEND.EQ.1) GOTO 999
0099 CALL ENQUE(PXINQ,XINQ,OUTQ)
0101 RETURN
0102 999
0103 END

```

PAGE 001

FORTRAN IV V02.1-11 Mon 01-Oct-79 09:01:00

```

0001 SUBROUTINE INPTQ(L)
0002 INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ
0003 INTEGER*2 PACKQ,PINQ,FLCNT,OUTFCT,T1,RESNLM,OUTO
0004 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0005 LOGICAL*1 PACK,MONTR,ISLID,ETX,CR,LF,DUM
0006 INTEGER*2 LTIME,IWRITM
0007 COMMON /DFM/ OUTBF(256),INBF(256),
0008 COMMON /WRITM,IWRIT,STAT,LTIME
0009 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
0010 LIDFD(256),IOFLG,LLFLG,IRSEND
0011 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
0012 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0013 COMMON /GLOB/ ISENT,FLCNT,IQINTH,OUTFCT,IFULL,OUTQ,
0014 RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0015 IF(L .LT. 1) GOTO 30
0016 LI=L-2
0017 II=INBF(6)
0018 IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0019 TI=IGETSP(N)
0020 DO 10 I=1,LI
0021 PACK(I,TI)=INBF(I)
0022 CALL ENSTR(PACK(254,T1),LI)
0023 CALL ENQUE(PINQ,INQ,T1)
0024 N=2
0025 GOTO 40
0026 N=1
0027 GOTO 40
0028 N=0
0029 DO 50 I=1,2
0030 OUTBF(I)=INBF(I)
0031 CONTINUE
0032 NN=1
0033 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
0042 ACKSEQ(II)=INBF(2)
0043 RETURN
0044 999
0045 END

```

PAGE 001

FORTRAN IV V02.1-11 Mon 01-Oct-79 09:01:06

```

0001 SUBROUTINE LPRINT(LI)
0002 INTEGER*2 STAT,FLCNT,OUTFCT,OUTC,RESNLM
0003 INTEGER*2 CC1,CC2,CC3,CC4,CC5,CC6,CC7
0004 LOGICAL*1 QUIBF,INEF,ETX,CR,LF,MONTR,ISLID,DUM
0005 INTEGER*2 LTIME,IWRTM
0006 COMMON/DFM/OUTBF(256),INBF(256),IWRTM,IWRT,STAT,LTIME
0007 COMMON/GLOB/ISENT,FLCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRT,ISLID,MONTR,IF,ETX,CR,DUM
0008 CC1=INEF(3)
0009 CC2=INEF(4)
0010 IF(CC1 .LT. 0) GOTO 25
0011 IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0012 IF(CC1 .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0013 CC3=MOD(CC1,2)
0014 CC4=MOD(CC1,256)
0015 IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0016 IF ((INEF(1) .EQ. 85) .AND. INEF(2) .EQ. 170) GOTO 99
0017 CC5=MOD(INEF(3),64)
0018 IF(CC5 .GE. 32) CALL INPTQ(LI)
0019 GOTO 99
0020 CC7=MOD(CC1,2)
0021 IF(CC7 .GE. 1) GOTO 30
0022 CALL ACKNAK(0)
0023 GOTO 99
0024 CALL ACKNAK(1)
0025 GOTO 99
0026 CALL INPTQ(LI)
0027 CONTINUE
0028 RETURN
0029 END

```

PAGE 001

FORTRAN IV V02.1-11 Mon 01-Oct-79 09:01:10

```

0001 SUBROUTINE INIT
0002 REAL*4 RH,RL,AH,AL,VMEAS
0003 INTEGE#2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ,FREE
0004 INTEGE#2 STAT,LCNT,OUTFCT
0005 INTEGE#2 SETPRM,RSTPRM,SETBKF,OUTQ,RESNLM,RSTBKF
0006 LOGICAL*1 IDATA,DUM
0007 LOGICAL*1 XOUTB,XINBF,ACKSEC,LIDFD,OUTBF,INBF
0008 LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID
0009 INTEGE#2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLM,I*RTTM
0010 COMMON /DPM/ OUTBF(256),INBF(256),
1 I*RTTM,I*WRT,STAT,LTIME
1 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
1 COMMON /MESS/ MESSEQ
1 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEC(256),LIDFD(256),
1 IOFLG,LLFLG,IRSEND
1 COMMON /TIM/ FREE(64),I*PR,I*PRSZ
1 COMMON /OLDTIM/ TIMLIM,ACKTIM,ATIMLM
1 COMMON /PACK(256,64)
1 COMMON /SWT/ SETPRM,RSTPRM,SETBKF,RSTBKF
1 COMMON /GLOB/ ISENT,FLCNT,ICLNTH,OUTFCT,IFULL,OUTQ,
RESNLM,I*ALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
1 COMMON /BCOM/ ISWTRH(13,3),RL(13,3),AL(13,3),
& I*TERM,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 LFLG=0
0026 I*RSZ=64
0027 INFLCT=0
0028 ISLID=26
0029 CR="015
0030 LF="012
0031 ETX="023
0032 RESNLM=2
0033 ICLNTH=16
0034 I*ALTRT=0
0035 I*LCNT=0
0036 I*WRTTM=0
0037 TIMLIM=50
0038 SETPRM=4
0039 RSTPRM=32
0040 SETBKF=8
0041 RSTBKF=64
0042 ATIMLM=200
0043 MONTOR=27
0044 I*WRI=0
0045 ISWTRH=0
0046 OUTFCT=0
0047 IFULL=0
0048 ISENT=0

```

```

FORTRAN IV      V02.1-11      Mon 01-Oct-79 09:01:12      PAGE 002

0049      DO 40 I=1,20
0050      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      LIDFD(I)=0
0061      DO 60 I=40,44
0062      LIDFD(I)=1
0063      DO 70 I=45,59
0064      LIDFD(I)=0
0065      DO 80 I=60,64
0066      LIDFD(I)=2
0067      DO 90 I=65,79
0068      LIDFD(I)=0
0069      DO 100 I=80,84
0070      LIDFD(I)=4
0071      DO 110 I=85,99
0072      LIDFD(I)=0
0073      DO 120 I=100,104
0074      LIDFD(I)=5
0075      DO 130 I=105,256
0076      LIDFD(I)=0
0077      PING(1)=ICLNTH
0078      PING(2)=ICLNTH+1
0079      PACKQ(1)=ICLNTH
0080      PACKQ(2)=ICLNTH+1
0081      PXOUTQ(1)=ICLNTH
0082      PXOUTQ(2)=ICLNTH+1
0083      PXING(1)=ICLNTH
0084      PXING(2)=ICLNTH+1
0085      DO 10 I=1,IFRSZ
0086      PACK(255,I)=0
0087      DO 20 I=1,IFRSZ
0088      FREE(I)=1
0089      DO 30 I=1,256
0090      ACKSEQ(I)=256
0091      IFR=IFRSZ
0092      RETURN
0093      END
0094

```

```

FORTRAN IV      V02.1-11      Mon 01-Oct-79 09:01:19      PAGE 001

0001      SUBROUTINE LINLOS
0002      REAL*8 RM1(5),LINO0,LINI8
0003      INTEGER*2 FWCNT,OUTFCT,OUTQ,RESNLM,T1
0004      INTER*2 XINQ,PXINQ,XOUTQ,ACKQ,PACKQ,PINQ
0005      LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0006      LOGICAL*1 PACK,M1(40),LINO(8),LINI(8)
0007      LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008      COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
0009      &      PACKQ(2),INQ(16),PING(2)
0010      COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
0011      &      IOFLG,LLFLG,IRSEND
0012      COMMON /GLOB/ ISENT,FWCNT,IQLENTH,OUTFCT,IFULL,OUTQ,
0013      &      RESNLM,IALHT,ISLID,MONTR,LF,ETX,CR,DUM
0014      COMMON /MESS/ MESSEQ
0015      DATA LINO0,LINI8/'PRIMARY ','BACKUP '//
0016      DATA RM1(1),RM1(2)/'LOSS OF ','MODULATI'//
0017      DATA RM1(3),RM1(4)/'ON ON LO ','OP AT NO /
0018      DATA RM1(5)/'DE 26 /
0019      EQUIVALENCE(LINO0,LINI8)
0020      EQUIVALENCE(LINI,LINI8)
0021      EQUIVALENCE(M1,RM1)
0022      IS=LLFLG
0023      CALL ENABLE(0)
0024      T1=IOETSP(N)
0025      CALL ENABLE(1)
0026      PACK(1,T1)=0
0027      IF(MESSEQ.EQ.126) MESSEQ=0
0028      MESSEQ=MESSEQ+1
0029      PACK(2,T1)=MESSEQ
0030      PACK(3,T1)=0
0031      PACK(4,T1)=0
0032      PACK(5,T1)=25
0033      PACK(6,T1)=ISLID
0034      DO 20 I=7,9
0035      &      PACK(I,T1)=LF
0036      CONTINUE
0037      DO 30 I=1,22
0038      &      PACK(I+9,T1)=M1(I)
0039      &      IF(IS.EC.1) GOTO 50
0040      &      DO 40 I=1,8
0041      &      &      PACK(I+31,T1)=LINO(I)
0042      &      GOTO 70
0043      &      DO 60 I=1,8
0044      &      &      PACK(I+31,T1)=LINI(I)
0045      &      DO 80 I=23,40
0046      &      &      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)

```

PAGE 002

MON 21-OCT-79 23:21:19

V02.1-11

FORTRAN IV

LLFIC=0  
RETURN  
END

0054  
0055  
0056

```

SUBROUTINE B*BSA
LOGICAL*1 XOUTP,XINZF,ACXSG,LIDFD,DATA,PACK
LOGICAL*1 ISLID,MONTR,CR,CRJDM
INTEGER*2 FLOC,OFFIC,CR,RESUM
INTEGER*2 XING,FOUW,ACKG,XING,PXOUTG,PACKG,PING
REAL*4 RE,RE,AL,AL,VEAS
COMMON /MISS/ MISSIC
COMMON /CSE/ XING(10),XING(2),XOUTG(10),PXOUTG(2),
   &  ACK(10),PACK(2),ANG(10),PING(2)
COMMON /CLOS/ISNT,ICNT,ICNTR,COUTCT,ISLID,COUTG,RESUM,
   &  IALFT,ISLID,MONTR,CR,CRJDM
COMMON /BOOM/ISWTR,IR(3,3),AL(13,3),AE(13,3),AL(13,3),
   &  ITRM,LINK,MCN,VEAS(13),ISUB,INUM,
   &  IDATA(10),IGENSW,ICEN
COMMON /EUS/ XOUTF(256),XINZF(256),ACXSG(256),LIDFD(256),
   &  ICFIS,LIPLC,ISEND
COMMON PACK(255,64)
DATA RE/1.2,20.7,3500.0,200.0,-20.0,10.0,19.5,11.15,23.5,
   &  3920.0,19.0,-1.0,21.5,
   &  2.91,39.2,3900.0,19.4,-13.5,13.9,19.0,12.96,39.0,
   &  3920.0,18.15,-16.0,24.5,
   &  4.16,57.0,3920.0,19.45,-16.0,13.0,16.3,14.0,40.0,
   &  3940.0,19.5,-18.0,24.0/
DATA RI/-1.35,-21.0,100.0,-19.0,-100.0,-10.0,-55.0,-11.27,
   &  -24.5,130.0,-20.0,-100.0,-24.0,
   &  -2.0,-39.2,200.0,-19.2,-98.0,-9.0,-67.0,-12.75,-39.0,
   &  200.0,-18.5,-66.0,-24.5,
   &  4.0,-50.0,100.0,-19.5,-103.0,-10.0,-62.0,-14.9,-40.0,
   &  100.0,-19.45,-102.0,-24.0/
DATA AB/1.22,2.3740.0,10.0,-30.0,13.5,10.0,9.25,23.0,3740.0,
   &  10.0,-20.0,-21.0,
   &  2.16,34.0,3770.0,19.1,-15.5,13.1,14.5,13.55,37.5,3700.0,
   &  17.5,-20.0,22.5,
   &  4.3,49.0,3700.0,10.6,-10.0,13.5,17.0,13.0,25.0,3740.0,10.5,
   &  -20.0,19.0/
DATA AL/-1.0,-20.0,400.0,-10.0,-100.0,-0.0,-60.0,-11.25,-23.0,
   &  -190.0,-19.0,100.0,-23.0,
   &  -2.5,-34.0,360.0,-10.7,-92.0,-5.0,-65.0,-12.6,-35.0,
   &  300.0,-17.45,-90.0,-23.0,
   &  -4.5,-40.0,210.0,-10.0,-100.0,-5.0,-55.0,-14.5,-25.0,
   &  300.0,-10.0,-92.0,-10.0/
IF(XINZF(3).EQ.'X'.AND.XINZF(4).EQ.'Y'.AND.
   &  XINZF(5).EQ.'X'.AND.XINZF(6).EQ.'X') GOTO 60
LECODE(2,4,XINZF(1),IR=0) ICNT
LECODE(4,2,XINZF(3),IR=0) VEAS(1)
LECODE(4,2,XINZF(7),IR=0) VEAS(2)
LECODE(4,2,XINZF(11),IR=0) VEAS(3)
LECODE(4,2,XINZF(15),IR=0) VEAS(4)
LECODE(4,2,XINZF(19),IR=0) VEAS(5)
LECODE(4,2,XINZF(23),IR=0) VEAS(6)
LECODE(4,2,XINZF(27),IR=0) VEAS(7)
LECODE(4,2,XINZF(31),IR=0) VEAS(8)
LECODE(4,2,XINZF(35),IR=0) VEAS(9)
LECODE(4,2,XINZF(39),IR=0) VEAS(10)

```

PAGE 002

```

FORTRAN IV      V02.1-11
0030  DECODE(4,2,XIMBF(43),ERR=80) VMEAS(11)
0031  DECODE(4,2,XIMBF(47),ERR=80) VMEAS(12)
0032  DECODE(4,2,XIMBF(51),ERR=80) VMEAS(13)
0033  DECODE(2,1,XIMBF(55),ERR=90) LINK
0034  DECODE(2,1,XIMBF(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  5 ((AL(INUM,ISUB).LE.VMEAS(INUM)).AND.
      GOTO 40
0045  20 CONTINUE
0047  IOFLG=0
0048  RETURN
0049  40 IF((AR(INUM,ISUB).LT.VMEAS(INUM)).AND.
      & (R(INUM,ISUB).GE.VMEAS(INUM)).OR.
      & (AL(INUM,ISUB).GT.VMEAS(INUM))) GOTO 50
0051  IF(ISWITCH.EQ.0) GOTO 80
0053  CALL SDRPT(1)
0054  IOFLG=0
0055  RETURN
0056  50 IF(ISWITCH.EQ.0) GOTO 80
0058  CALL SDRPT(2)
0059  IOFLG=0
0060  RETURN
0061  60 IF(ISWITCH.EQ.0) GOTO 80
0063  CALL SDRPT(3)
0064  GOTO 80
0065  70 CALL ENABLE(0)
0066  K1=IGTSP(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)=2
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

```

ISEND RED REPORT

ISEND AMBER REPORT

ISEND ALARM REPORT

PAGE 003

```

FORTRAN IV      V02.1-11
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 ENOUR(PXINQ,XINQ,K1)
0118      CALL ENABLE(1)
0119      ICHNSW=2
0120      IOFLG=0
0121      RETURN
0122      END

```

PAGE 201

FORTRAN IV V02.1-11

```

0001 SUBROUTINE SDRFRT(IC)
0002 REAL*4 PH,PL,AH,AL,VMEAS
0003 LOGICAL*41 YOUTER,XINBF,ACKSEQ,LIDFD
0004 LOGICAL*41 PACK,DATA,ISLID,MONTR,LF,PTX,CP,DUM
0005 INTEGER*2 XING,XOUTG,ACKG,PXING,PXOUTG,PING
0006 INTEGER*2 FWCNT,OUTFCT,OUTG,RESNM
0007 COMMON/ECON/ISLID,PH(13,3),AL(13,3),AH(13,3),AL(13,3),
      5 ITERM,LINE,MON,VMEAS(13),ISUB,INUM,
      5 IDATA(12),ICRNSM,ICHAN
0008 COMMON /BUIS/ XOUTER(256),XINBF(256),ACKSEQ(256),LIDFD(256),
      5 IOFLG,LLFLG,IRSEND
0009 COMMON/MESS/MESSEQ
0010 COMMON PACK(256,64)
0011 COMMON/QUE/XING(15),PXING(2),XOUTG(16),PXOUTG(2),
      5 ACKQ(16),PACKQ(2),INQ(15),PING(2)
0012 COMMON/GLOB/ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTG,
      5 RESNM,IALTPT,ISLID,MONTR,LF,PTX,CP,DUM
      CALL ENABLE(0)
0013 K1=IGETSP(N)
0014 CALL ENABLE(1)
0015 DO 10 I=1,256
0016   PACK(I,K1)=0
0017   PACK(I,K1)=0
0018 CONTINUE
0019 IF(MESSEQ .EQ. 126) MESSEQ=0
0020 MESSEQ=MESSEQ+1
0021 PACK(1,K1)=0
0022 PACK(2,K1)=MESSEQ
0023 PACK(3,K1)=0
0024 PACK(4,K1)=0
0025 PACK(5,K1)=MONTR
0026 PACK(6,K1)=ISLID
0027 IF(IC .EQ. 3) GOTO 50
0028 DO 20 I=7,14
0029   PACK(I,K1)=XINBF(I+50)
0030 CONTINUE
0031 ENCODE(2,32,PACK(15,K1)) LINK
0032 FORMAT(A2)
0033 FORMAT(A4)
0034 PACK(17,K1)=IC
0035 ENCODE(4,40,PACK(12,K1)) VMEAS(INUM)
0036 ENCODE(2,30,PACK(22,K1)) MON
0037 PACK(24,K1)=ITERM
0038 CALL ENSTR(PACK(254,K1),24)
0039 GOTO 70
0040 DO 50 I=7,10,
0041   PACK(I,K1)=X'
0042 CONTINUE
0043 DO 60 I=11,14
0044   PACK(I,K1)=XINBF(I)
0045 CONTINUE
0046 PACK(15,K1)=CP
0047 PACK(16,K1)=LF
0048 PACK(17,K1)=PTX
0049 CALL ENSTR(PACK(254,K1),17)

```

PAGE 002

```
FORTRAN IV      V02.1-11
0052  70  CALL ENABLE(0)
0053      CALL ENQUE(PXING,XING,K1)
0054      CALL ENABLE(1)
0055      RETURN
0056      END
```

```

FORTRAN IV      V02.1-11                PAGE 021
0001      SUBROUTINE BWSALP
0002      LOGICAL*1 IDATA
0003      REAL*4 RH,RL,AH,AL,VMEAS
0004      COMMON/BCOM/ISWTC,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      ISWTC=1
0013      ITERM=IDATA(4)
0014      RETURN
0015      ISWTC=0
0016      RETURN
0017      DECODE(4,75, IDATA(2),ERR=80) ICHAN
0018      FORMAT(I4)
0019      CALL SOUT(ICHAN)
0020      ICHNSW=1
0021      RETURN
0022      END

```

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

```

1 .TITLE FDM-MACRO
2 .SETTL NODE 26
3 .IDENT /V3.0/
4 .GLOBL LIUINT,LIO,ENABLE,SWITCH,MASTER,TIME
5 .GLOBL WTOKEN,RAM,STATUS,STAB0
6 .GLOBL LPOUT,RSTART,RDPNT,DESTR,ENSTR,LPINPT
7 .NLIST CND
8 .PSECT
9

```

```

10 000000
11 000001
12 000002
13 000003
14 000004
15 000005
16 000006
17 000007
18
19
20
21

```

;INTERFACE ADDRESS

```

22 BAR= 172410
23 WCR= 172412
24 CSR= 172414
25 IOBUF= 172416
26 OPREG= 172418
27 HRCSH= 177560
28 HRBUF= 177562
29 HXCSR= 177564
30 HXBUF= 177566
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

;CHANGED

;INTERFACE OPTIONS

```

41 XCIO=1
42 XSIO=0
43 XGIO=1
44 NDXX=0
45 ND24=1
46 NHOST=0
47 DHOST=1
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

;LOCAL VARIABLES

```

51 ARG1= 2
52 ARG2= 4
53 ARG3= 6
54 ARG4= 10
55
56 BIT15= 100000
57 BIT14= 40000
58 BIT03= 400
59 BIT07= 200
60 BIT06= 100
61 BIT05= 40
62 BIT04= 20
63 BIT03= 10
64 BIT02= 4
65

```

MACRO V03.02B1-OCT-79 09:01:28 PAGE 1-1

FDX.MACRO  
NODE 26

66 00002  
67 000001  
68

BIT01= 2  
BIT00= 1

MACRO 003.02B1-OCT-79 09:01:28 PAGE 2

FDM-MACRO  
NODE 26

```

1 *****
2 *** PROCEDURE MASTER START UP ***
3 *****
4
5
6 MASTER: MOV #340, R0 ;PRI=7
7 MTPS R0
8 MOV #124, R0 ;LIU HANDLER
9 MOV #LIO, (R0)+ ;PRI=7
10 MOV #340, (R0) ;RESET BUS
11
12 RESET
13 MOV #60, R0 ;SIG INTERFACE HANDLER
14 MOV #SIO, (R0)+ ;CLOCK VECTOR
15 MOV #340, (R0)+ ;CLOCK
16 MOV #I00, R0 ;DISABLE INTERRUPTS
17 MOV #TIME, (R0) ;CLEAR LIU
18 BIC #40100, G#CSR ;CLEAR BLIUI CSR
19 JSR PC, LIUINT ;PRI=0
20 CLR G#CSR ;CPU DOWN
21 MTPS R0
22 RTS PC
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
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

```

*****
*** PROCEDURE INITIALIZE ***
*****
;CLEAR AGRAM
LDADR #4354, G#OPREG ;LDADR
ADDRESS=0 ;ADDRESS=0
GOOD WD ;GOOD WD
NO RETRY ;NO RETRY
;SEL AGRAM
COUNTER ;COUNTER
WRITE A NULL ;WRITE A NULL
GOOD WD ;GOOD WD
NO LOOP UNTIL READY ;NO LOOP UNTIL READY
COUNT+1 ;COUNT+1
;RBUFADR CMD
;SEL INBUFO
;SET POINTER=0
;FALSE READ DATA
;GOOD READ
;NO RETRY
;CLEAR DONE BIT
;RBUFADR COMMAND
;SEL INBUFI COMMAND
;POINTER=0
;CLEAR INPUT/OUTPUT BUFFERS
MOV #10410, R1
MOV #10440, R0
JSR PC, ZEROBP
MOV #1400, G#CPREG
TSTB G#CSR
BPL #-4
CLR G#CSR
INC R0
BNE 1$
;CLEAR INPUT/OUTPUT BUFFERS
MOV #10410, R1
MOV #10440, R0
JSR PC, ZEROBP
MOV #1400, G#CPREG
TSTB G#CSR
BPL #-4
CLR G#CSR
MOV #10610, R1
MOV #10640, R0
JSR PC, ZEROBP

```

FDM:MACRO  
NODE 26

MACRO V03.02B1-OCT-79 08:01:28 PAGE 2-1

```
102 000232 012737 001400 172416      MOV      #1400, @#OPREG
103 000240 105737 172414      TSTB
104 000244 100375          BPL
105 000246 105037 172414      CLRB
106 000252 012700 010510      MOV      #10510, R1
107 000256 012700 010540      MOV      #10540, R0
108 000262 004767 000016      JSR      PC, ZEROBP
109 000266 012700 010710      MOV      #10710, R1
110 000272 012700 010740      MOV      #10740, R0
111 000276 004767 000002      JSR      PC, ZEROBP
112 000302 000440          BR
113 000304 010137 172416      MOV      #1400, @#OPREG
114 000310 012737 001400      MOV      #1400, @#OPREG
115 000316 105737 172414      TSTB
116 000322 100375          BPL
117 000324 105037 172414      CLRB
118 000330 013702 172416      MOV      @#IOBUF, R2
119 000334 042702 172400      BIC      #177400, R2
120 000340 010037 172416      MOV      R0, @#OPREG
121 000344 022702 000000      CMP      #0, R2
122 000350 001412          BEQ      3$, @#OPREG
123 000352 012737 001400      MOV      #1400, @#OPREG
124 000360 105737 172414      TSTB
125 000364 100375          BPL
126 000366 105037 172414      CLRB
127 000372 005302          DEC      R2
128 000374 000763          BF
129 000376 010037 172416      MOV      R0, @#OPREG
130 000402 000207          RTS
131
132
133
134 000404 012737 010400      MOV      #4352, @#OPREG
135 000412 012737 002400      MOV      #1280, @#OPREG
136 000420 012737 002400      MOV      #1280, @#OPREG
137 000425 000207          RTS
138
139
```

```

;FALSE READ DATA
;GOOD RD
;NO RETRY
;CLEAR DONE BIT
;RDBUFADR COMMAND
;SEL OUTBUF0 COMMAND

;RDBUFADR COMMAND
;SEL OUTBUF0 COMMAND
;ZERO BUFFER POINTER
;GO CLEAR STATUS
;RDBUFADR
;RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;FETCH POINTER
;CLEAR MST BYTE
;SEL BUFFER
;POINTER=07
;YES RECHECK
;NO FALSE RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;LOCAL POINTER-1
;BR NTL 0
;SEL BUFFER
;RETURN

;WCR:RS
;RS(FALSE)
;RS

```

```

1 *****
2 *** PROCEDURE LIU-HANDLER ***
3 *****
4
5
6 000430 010046 LIO: MOV R0, -(SP)
7 000432 010146 MOV R1, -(SP)
8 000434 010246 MOV R2, -(SP)
9 000436 010346 MOV R3, -(SP)
10 000440 010446 MOV R4, -(SP)
11 000442 010546 MOV R5, -(SP)
12 000444 005737 TST G*CSR
13 000450 100153 BPL RTI$
14 000452 042737 BIC #BIT14, G*CSR
15 000460 012737 MOV #4352, G#OPREG
16 000466 012737 MOV #1280, G#OPREG
17 000474 013767 MOV G#IORUF, CAUSE
18 000502 042767 BIC #177400, CAUSE
19
20
21
22 000510 132767 BITB #BIT02, CAUSE
23 000516 001406 BEQ B3$,
24 000520 012702 MOV #4360, R2
25 000524 012703 MOV #4384, R3
26 000530 004767 JSR PC, EMBF
27 000534 132767 BITB #BIT03, CAUSE
28 000542 001406 BEQ B4$,
29 000544 012702 MOV #4488, R2
30 000550 012703 MOV #4512, R3
31 000554 004767 JSR PC, EMBF
32 000560 132767 BITB #BIT04, CAUSE
33 000566 001407 BEQ B5$,
34 000570 132767 BITB #BIT07, CAUSE
35 000576 001403 BEQ B5$,
36 000600 012767 MOV #2, LLFLG
37 000606 132767 BITB #BIT05, CAUSE
38 000614 001403 BEQ B6$,
39 000616 012767 MOV #1, LLFLG
40 000624 132767 BITB #BIT06, CAUSE
41 000632 001404 BEQ B0$,
42 000634 005067 CLR IWRTM
43 000640 005067 CLR IWRT
44
45
46
47 000644 132767 BITB #BIT00, CAUSE
48 000652 001423 BEQ B1$,
49 000654 012701 MOV #INBF, R1
50 000660 012702 MOV #4360, R2
51 000664 012703 MOV #4384, R3
52 000670 012704 MOV #BIT00, R4
53 000674 004767 MOV #AREA, R5
54 000700 012705
55 000704 012767 MOV #1, AREA
56 000712 010267 MOV #2, LATA
57 000716 004767 JSR PC, LPINPT

```

REA

;DATA LINK A

;SAVE R0  
;SAVE R1  
;SAVE R2  
;SAVE R3  
;SAVE R4  
;SAVE R5  
;? LIU INTERRUPTED  
;NO  
;YES/DISABLE LIU  
;WCR : RS(0)  
;RD  
;CLEAR UNUSED BITS  
;OV-PL  
;RDRUFADR IN0  
;SEL INBUF0  
;GO EMPTY  
;OV-PL  
;RDRUFADR IN1  
;SEL INBUF1  
;GO EMPTY  
;LINE-LOSS PRIMARY  
;LATCH ON  
;YES LOOP AROUND  
;FLAG PRIMARY SW. FAIL  
;BACKUP LINE LOSS  
;FLAG BACKUP SW. FAIL  
;WRITE TOKEN DETECT  
;CLEAR TOKEN VAR  
;INBUF0 FULL  
;NO  
;BUFFER ADDRESS  
;RDRUFADR COMM  
;SEL BUF  
;CRC BIT  
;1 VARIABLE  
;CRC OR BYTE COUNT  
;CALL FORTRAN QUE'ER

;PERFORM OPERATION DEPENDING ON BITS

```
58 000722 132767 000002 000254 B1$ BITB #BIT01, CAUSE  
59 000730 001423 BEQ RTI$  
60 000732 012701 000400 #INPF, R1  
61 000736 012702 010610 #4488., R2  
62 000742 012703 010640 #4512., R3  
63 000746 012704 000002 #BIT01, R4  
64 000752 004767 000046 PC #GO EMPTY BUFFER  
65 000756 012705 001206 #AREA, R5  
66 000762 012767 000001 #1, AREA  
67 000770 010267 000216 R2, DATA  
68 000774 004767 000000 JSR PC, LFINPT  
69  
70 001000 012605 RTI$ F5  
71 001002 012604 (SP)+, R4  
72 001004 012603 (SP)+, R3  
73 001006 012602 (SP)+, R2  
74 001010 012601 (SP)+, R1  
75 001012 012600 (SP)+, R0  
76 001014 052737 040000 BIS #BIT14, G#CSR  
77 001022 000000 RTI  
78  
79 001024 012737 010600 MOV #4480., G#OPREG  
80 001032 012737 002400 MOV #1280., G#OPREG  
81 001040 013700 172416 MOV #010BUF, R0  
82 001044 130400 BITB R4,R0  
83 001046 001002 BNE CRCOK  
84 001050 012704 MOV #1, R4  
85 001054 010237 MOV R2, G#OPREG  
86 001060 012737 001400 MOV #768., G#OPREG  
87 001066 105737 172414 TSTB G#CSR  
88 001072 100375 BPL -4  
89 001074 013702 MOV #010BUF, R2  
90 001100 042702 BIC #177400, R2  
91 001104 010200 MOV R0,  
92 001106 005400 NEG R0,  
93 001110 010037 MOV R0, G#OCR  
94 001114 010137 MOV R1, G#BAF  
95 001120 010337 MOV R2, G#OPREG  
96 001124 012737 001400 MOV #768., G#OPREG  
97 001132 105737 TSTB G#CSR  
98 001136 100375 BPL -4  
99 001140 012737 MOV #8704., G#OPREG  
100 001146 000240 NOP  
101 001150 105737 TSTB G#CSR  
102 001154 102401 EMI DMAOK  
103 001156 000240 NOP  
104 001160 012737 MOV #2304., G#OPREG  
105 001166 105737 TSTB G#CSR  
106 001172 100375 BPL -4  
107 001174 005704 TST R4  
108 001176 100001 BPL END$  
109 001200 010402 MOV R4, R2  
110 001202 000207 RTS PC  
111 001204 000000 CAUSE: .WORD 0  
112 001206 000000 AREA: .WORD 0  
113 001210 001212 .WORD DATA  
114 001212 000000 .WORD 0
```

EDM.MACRO MACRO V03.02B1-OCT-79 09:01:28 PAGE 4  
NODE 26

```

1      .ENABLE LSB
2      *****
3      ***** PROCEDURE ENABLE
4      *****
5
6      001214 017501 000002          ENAB: MOV  GARG1(R5), R1
7      001220 022701 000201          CMP   #1, R1
8      001224 001404          BEQ   E1$,
9      001226 042737          BIC   #BIT14, @GCSR
10     001234 000207          RTS   PC
11     001236 052737          BIS   #BIT14, @GCSR
12     001244 000207          RTS   PC
13
14     ;FORTRAN ENCODE/LFCODE MACRO
15
16     001246 017575 000004 000002 ENSTR: MOV  GARG2(R5),GARG1(R5)
17     001254 000207          RTS   PC
18     001256 017575 000002 000004 DEST:  MOV  GARG1(R5),GARG2(R5)
19     001264 000207          RTS   PC
20
21
22
23     *****
24     ***** PROCEDURE LINE SWITCH
25     *****
26
27     ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
28
29     ;
30     ;04=SET PRIMARY LINE
31     ;32=RESET PRIMARY LINR
32     ;08=SET BACKUP LINE
33     ;04=RESET BACKUP LINE
34
35     SWITCH: MOV  GARG1(R5),R1
36     001266 012737 010420 172416 MOV  #4368., @OPREG
37     001272 062701 004400          ADD  #2304., R1
38     001300 010137 172416          MOV  R1, @OPREG
39     001304 105737 172414          TSTB @GCSR
40     001310 100375          BPL  #-4
41     001314 105037 172414          CLRB @GCSR
42     001322 000207          RTS   PC
43
44     ;SW SET
45     ;WCR : MODSTAT
46
47     ;RD
48     ;VALID WRITE
49     ;NO LOOP UNTIL READY
50     ;CLEAR DONE BIT

```

MACRO Y23.2221-001-79 29:21:22 PAGE 5

FDL MACRO  
NOTE 20

\*\*\*\*\*  
PROCEDURE STATUS  
\*\*\*\*\*

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

IX=0 STATUS BYTE 0  
IX=1 STATUS BYTE 1

10	001324	017501	000002	STATUS: MOV	GAR0(R5), R1	FETCH BYTE
11	001328	002701	000001	COMP	R1, R1	COMPARE
12	001334	001415	000000	BEQ	Z5	
13	001336	012707	010400	MOV	#4300, @OPREG	PCR : RS(0)
14	001344	012707	002400	MOV	#1200, @OPREG	FETCH DATA
15	001352	013708	172416	MOV	@OP00, R0	CLEAR BITS
16	001356	042700	177400	BIC	#177400, R0	RTN DATA
17	001362	010005	000004	MOV	R0, GAR02(R5)	
18	001366	000007		RTS	PC	
19	001370	012707	010500	MOV	#4300, @OPREG	PCR : RS(1)
20	001376	012707	002400	MOV	#1200, @OPREG	RS
21	001404	013708	172416	MOV	@OP00, R0	FETCH DATA
22	001410	042700	177400	BIC	#177400, R0	CLEAR BITS
23	001414	010005	000004	MOV	R0, GAR02(R5)	RTN DATA
24	001420	000007		RTS	PC	
25	001422	012707	010400	MOV	#4300, @OPREG	STATUS BYTE 1
26	001430	012707	002400	MOV	#1200, @OPREG	RS
27	001436	013708	172416	MOV	@OP00, R1	FETCH DATA
28	001442	042701	177400	BIC	#177400, R1	RTN DATA
29	001446	132701	000010	BEQ	#B100, R1	BYTES FOLD
30	001452	001404		MOV	R1, GAR01(R5)	LOAD RETURN
31	001454	012705	000001	RTS	PC	
32	001452	000007		RTS	PC	
33	001464	012705	000000	MOV	R0, GAR01(R5)	LOAD RETURN
34	001472	000007		RTS	PC	
35						
36						



MACRO V03.02B1-OCT-79 09:01:2E PAGE 7

EDM-MACRO  
NOTE 2C

```

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 *****
49 *****

```

```

*****
***** PROCEDURE ACRAM
*****
;CALL RAM(0,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
;
;0110 =NHEAD
;0100 =DHEAD
;0111 =NULL
;0000 =WTKEN
;
RAM: MOV GARG1(R5), R0
CMP #0, R0
RDRAM
BNE GARG2(R5), R0
WTRAM: MOV GARG3(R5), R1
MOV #4354., @#OPREG
ADD #2304., R0
MOV R0, @#OPREG
TSTB @#CSR
BPL -4
CLRB @#CSR
MOV #4353., @#OPREG
ADD #2304., R1
MOV R1, @#OPREG
TSTB @#CSR
BPL -4
CLRB @#CSR
RIS
PC
RDRAM: MOV GARG2(R5), R0
MOV #4354., @#OPREG
ADD #2304., R0
MOV R0, @#OPREG
TSTB @#CSR
BPL -4
CLRB @#CSR
MOV #4353., @#OPREG
MOV #768., @#OPREG
TSTB @#CSR
BPL -4
CLRB @#CSR
MOV #10401, 172416
MOV #01400, 172416
TSTB @#CSR
BPL 172414
CLRB @#CSR
MOV #10401, 172414
MOV #01400, 172414
BIC #17760, R0
MOV R0, @GARG3(R0)
RIS
PC

```

```

; WHICH OPERATION
; READ OP
; ADDRESS
; WRITE DATA
; SEL LDATA
; WD/DATA (ADDR)
; WRITE DATA
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; WD/DATA (CMD)
; WRITE
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; ADDRESS TO READ
; SEL LDATA
; WD/DATA (ADDR)
; WRITE DATA
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; READ DATA
; INVALID READ
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; FET H DATA
; CLEAR BITS
; RTN DATA

```

MACRO V03.02B1-OCT-79 09:01:28 PAGE 8

FDM.MACRO  
NODE 26

```

1 002102 017502 000002 172414 000000 000000 000000 000000 000000 000000
2 002106 012700 000340 172416 000000 000000 000000 000000 000000 000000
3 002112 100400 000000 172416 000000 000000 000000 000000 000000 000000
4 002114 012737 000000 172410 000000 000000 000000 000000 000000 000000
5 002122 005402 000000 172412 000000 000000 000000 000000 000000 000000
6 002124 010237 172412 172416 000000 000000 000000 000000 000000 000000
7 002130 012737 010540 172416 000000 000000 000000 000000 000000 000000
8 002136 012737 024000 172416 000000 000000 000000 000000 000000 000000
9 002144 000240 000000 172414 000000 000000 000000 000000 000000 000000
10 002146 100573 172414 172414 000000 000000 000000 000000 000000 000000
11 002152 000240 000000 172414 000000 000000 000000 000000 000000 000000
12 002154 105037 172414 172416 000000 000000 000000 000000 000000 000000
13 002160 012737 010740 172416 000000 000000 000000 000000 000000 000000
14 002166 012737 001652 172410 000000 000000 000000 000000 000000 000000
15 002174 012702 000002 172416 000000 000000 000000 000000 000000 000000
16 002200 005402 000000 172412 000000 000000 000000 000000 000000 000000
17 002202 010237 172412 172416 000000 000000 000000 000000 000000 000000
18 002206 012737 024000 172416 000000 000000 000000 000000 000000 000000
19 002214 000240 000000 172414 000000 000000 000000 000000 000000 000000
20 002216 100573 172414 172414 000000 000000 000000 000000 000000 000000
21 002222 000240 000000 172414 000000 000000 000000 000000 000000 000000
22 002224 105037 172414 172416 000000 000000 000000 000000 000000 000000
23 002230 012737 010420 172416 000000 000000 000000 000000 000000 000000
24 002236 012737 004403 172416 000000 000000 000000 000000 000000 000000
25 002244 105737 172414 172414 000000 000000 000000 000000 000000 000000
26 002250 100375 172414 172414 000000 000000 000000 000000 000000 000000
27 002252 105037 172414 172414 000000 000000 000000 000000 000000 000000
28 002256 012700 000000 172414 000000 000000 000000 000000 000000 000000
29 002262 105400 000000 172414 000000 000000 000000 000000 000000 000000
30 002264 000207 000000 172414 000000 000000 000000 000000 000000 000000
31 002266 000000 000000 172414 000000 000000 000000 000000 000000 000000
32 002268 000000 000000 172414 000000 000000 000000 000000 000000 000000
33 002270 000000 000000 172414 000000 000000 000000 000000 000000 000000
34 002272 000000 000000 172414 000000 000000 000000 000000 000000 000000
35 002274 000000 000000 172414 000000 000000 000000 000000 000000 000000
36 002276 000000 000000 172414 000000 000000 000000 000000 000000 000000
37 002278 000000 000000 172414 000000 000000 000000 000000 000000 000000
38 002280 000000 000000 172414 000000 000000 000000 000000 000000 000000

```

```

*****
;***** PROCEDURE WRITE LOOP (LPOUT) *****
*****

```

```

;BYTE COUNT
;PRI=7
;BUS ADDRESS
;2'S COMP COUNT
;COUNT
;JOB COMMAND
;DMA GO
;INTERFACE TIME
;DMA OK
;ERROR IF HERE
;JOB1 COMMAND
;ADDRESS OF WRITE TOKEN
;BYTE COUNT
;FORMAT FOR BLUI
;DMA BYTE COUNT
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF NOT
;CLEAR DONE
;MODSTAT
;BUFFERS FULL
;GOOD WRITE
;NO LOOP UNTIL
;CLEAR DONE BIT
;PRI=0
;LOWER CPU

```

MACRO V03.02B1-OCT-79 09:01:28 PAGE 9

FDM-MACRO  
NODE 26

```

1 5 002266 017500 000002
2 6 002272 022700 000000
3 7 002275 001004
4 8 002300 012700 010410
5 9 002304 000167 000046
6 10 002310 022700 000001
7 11 002314 001004
8 12 002316 012700 010610
9 13 002322 000167 000030
10 14 002325 022700 000002
11 15 002332 001004
12 16 002334 012700 010510
13 17 002340 000167 000012
14 18 002344 022700 000004
15 19 002350 001020
16 20 002352 012700 010710
17 21 002355 010037 172416
18 22 002362 012737 001400
19 23 002370 100375 172414
20 24 002374 100375
21 25 002375 013700 172416
22 26 002402 042700 177400
23 27 002406 010075 000004
24 28 002412 000207

;*****
;***** PROCEDURE READ BUFFER POINTER *****
;*****
;*****
RDPNT:  MOV   GARG1(R5), R0
        CMP   #0, R0
        BNE  1$, R0
        MOV  #4360., R0
        JMP  4$, R0
        CMP  #1, R0
        BNE  2$, R0
        MOV  #4488., R0
        JMP  4$, R0
        CMP  #2, R0
        BNE  3$, R0
        MOV  #4424., R0
        JMP  4$, R0
        CMP  #4, R0
        BNE  5$, R0
        MOV  #4552., R0
        JMP  4$, R0
        CMP  #758., R0
        BNE  #OPREG
        TSTB #OPREG
        EPL  -4
        MOV  G#IOBUF, R0
        PIC  #177400, R0
        MOV  R0, GARG2(R5)
        RTS

1$:
2$:
3$:
4$:
5$:
;FETCH COMMAND
;INBUF? ?
;NO
;RDBUFADR IN0
;INBUF1 ?
;NO
;RDBUFADR IN1
;OUTBUF0 ?
;NO
;RDBUFADR OUT0
;OUTBUF1 ?
;NO RETURN
;RDBUFADR OUT1
;RD
;READY
;NO LOOP UNTIL
;FETCH POINTER
;CLEAR MST
;RETURN POINTER

```

FDM.MACRO  
NODE 26

MACRO V03.02B1-OCT-79 09:01:28 PAGE 10

```
1 *****  
2 *****  
3 *****  
4 *****  
5 *****  
6 *****  
7 *****  
8 *****  
9 002414 017500 000002 RSTART: MOV GARG1(R5), R0  
10 002420 022700 000000 RS0:  #0, R0  
11 002424 001002          BNE  RS1  
12 002426 000000          HALT  
13 002430 000207          RTS   PC  
14 002432 022700 000001          CMP  #1, R0  
15 002436 001005          BNE  RS2  
16 002440 004767 175432          JSR  PC, LIUINT  
17 002444 012700 000040          MOV  #40, R0  
18 002450 000110          JMP  (R0)  
19 002452 022700 000002          CMP  #2, R0  
20 002456 001002          BNE  RS1  
21 002460 000177 000002          JMP  GROM  
22 002464 000207          RTS   PC  
23 002466 173000          .WORD 173000  
24  
25  
26
```

```
*****  
*  
*****  
;MODE  
;LOAD ADDRESS  
  
;RESTART PROGRAM  
;INT LIU FIRST  
;START ADDRESS  
;LOAD MODE
```

FDM-MACRO  
NODE 26  
MACRO V03.02B1-OCT-79 09:01:28 PAGE 11

```

1 .ENABLE LSB
2 *****
3 ;***# PROCEDURE SIG INTERFACE ***
4 ;*****
5
6 SIO:: MOV R0, -(SP)
7 002470 010046 MOV R1, -(SP)
8 002472 010146 MOV R2, -(SP)
9 002474 010246 MOV #XINBF, R0
10 002502 012700 000400 MOV #X0, R1
11 002506 105737 000120 TSTB @#HRCR
12 002512 100375 BPL *-4
13 002514 113702 177562 MOVB @#HRBUF, R2
14 002520 042702 177400 BIC #177400, R2
15 002524 010267 002000 MOV R2, IOFLG
16 002530 110220 MOVB R2, (R0)+
17 002532 005301 DEC R1
18 002534 105737 177560 TSTB @#HRCR
19 002540 100375 BPL *-4
20 002542 113702 177562 MOVB @#HRBUF, R2
21 002546 042702 177400 BIC #177400, R2
22 002552 110220 MOVB R2, (R0)+
23 002554 005301 DEC R1
24 002556 001366 BNE 25
25 002560 042737 BIC #100, @#HRCR
26 002566 012602 MOV (SP)+, R2
27 002570 012601 MOV (SP)+, R1
28 002572 012600 MOV (SP)+, R0
29 002574 000002 RTI
30
31
32 ;SEND OVER TO SIG
33 SOUT:: MOV GARGI(R5),R0
34 002576 017500 000002 TSTB @#HRCR
35 002602 105737 177564 BPL *-4
36 002606 100375 MOVB R0, @#HXBUF
37 002610 110037 177566 RTS PC
38
39 ;REPORT NODE READY
40
41
42 002616 105737 177564 TSTB @#HRCR
43 002622 100375 BPL *-4
44 002624 112737 000122 MOVB #122, @#HXBUF
45 002632 052737 000100 BIS #100, @#HRCR
46 002640 000207 RTS PC
48

```

```

;SAVE REGISTERS
;BUFFER ADDRESS
;BYTE COUNT
;PORT READY
;LOOP UNTIL
;FETCH BYTE
;CLEAR MST
;ITS THE BYTE COUNT
;COUNT-1
;PORT READY
;LOOP UNTIL
;FETCH BYTE
;CLEAR MST
;STORE IT
;READ 39 BYTES
;NO MORE UNTIL FORTRAN
;RETORE REGISTERS
;BYTE COUNT
;READY TO SNED
;LOOP UNTIL
;SEND IT
;PORT READY
;LOOP UNTIL
;SEND R
;ENABLE INTERRUPTS
;RETURN

```

FDM MACRO  
MODE 26

MACRO V03.02B1-OCT-79 09:01:28 PAGE 12

```

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

*****
;**** COMMON DATA AREA ****
;*****
.PSECT DFM,RW,D,CBL,REL,OVR

OUTBF: .BLKB 256.
INBF: .BLKB 256.
IWRTM: .BLKW
STAT: .BLKW
LTIME: .BLKW
.EVEN

.PSECT BUFS,RW,D,CBL,REL,OVR

KOUTBF: .BLKB 256.
XINBF: .BLKB 256.
ACKSEQ: .BLKB 256.
LIDFD: .BLKB 256.
IOPLG: .BLKW
LLFLG: .BLKW
IRSEND: .BLKW
.EVEN

.END

000001

```

FDM-MACRO SYMBOL TABLE  
 MACRO 003.01 23-JUL-79 12:51:52 PAGE 12-1

ACKSED	001000R	003	H1\$	000222R	E1\$	001236R	MASTER	000000RG	SOUT	002576RG	002
AREA	001206R		H2\$	000510R	64\$	002444R	NIXX =	000000	STAT	001004R	
ARG1	= 000002		B3\$	000534R	HRBUF =	177562	ND24 =	000001	STAT0	001422RG	
ARG2	= 000004		B4\$	000560R	HRCSR =	177560	NHOST =	000000	STATUS	001324RG	
ARG3	= 000006		H5\$	000605R	HXRBUF =	177564	UPREG =	172416	STAT\$	000404R	
ARG4	= 000010		B6\$	000624R	HXCSCR =	177564	OUTRF	000000R	SWITCH	001266RG	
BAR	= 172410		B7\$	000570R	INBF	000400R	002	EAM	TIME	001654RG	
BIT00	= 000001		CAUSE	001204R	IOHUF =	172416	KDFNT	001676RG	WCR =	172412	
BIT01	= 000002		CRCOK	001054R	IOFLG	002000R	KIKAM	002002R	WTH	001652R	
BIT02	= 000004		CSR	= 172414	IRSEND	002004R	003	KIKAM	WTNG	001552R	
BIT03	= 000010		DATA	001212R	IMRT	001002R	002	KOM	WTKEN	001474RG	
BIT04	= 000020		DESTR	001256RG	IMRTTH	001000R	002	RST	WTRAM	001710R	
BIT05	= 000040		DHOST =	000001	LIHFD	001400R	003	RSTART	XCIO =	000001	
BIT06	= 000100		DMAOK	001160R	LIO	000430RG	R50	002420R	XGIO =	000001	
BIT07	= 000200		EMBF	001024R	LIJINT	000076RG	R51	002432R	XINBF	000400R	003
BIT08	= 000400		ENABLE	001214RG	LI FLG	002002R	003	R52	XOUTRF	000000R	003
BIT14	= 040000		END\$	001202R	LFINFT =	***** G	KTI\$	001000R	XSIO =	000000	
BIT15	= 100000		ENSTR	001246RG	LFOUT	002102RG	SIO	002470RG	ZERDEF	000304R	
BO\$	000644R		E0\$	001226R	LTIME	001006R	002				

\* ABS. 000000 000  
 002642 001  
 IFM 001010 002  
 IUF5 002006 003  
 ERRORS DETECTED: 0

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

PAGE 001

12:00:00

01-JUL-79

CHIPS.CUM

POSTUM

DESCRIPTION

FOR MONTH /NOI TRF  
FOR MONTH /NOI TRF  
FOR MONTH /NOI TRF  
MMW F 10M



Page 001

12:00:00

01-JUL-79

100.00.000

PROGRAM:

ALPHA

ASSIGN DX0: BK:  
 K LINE  
 DX1: N00E236 DX1: N0P=DX1: N00601 / C / L / M  
 DX1: F00 / Z  
 DX1: N00601 / C  
 DX1: B00601 / Z  
 \$SMRT

ASSIGN DX1: BK:

## 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

<u>Connectivity Group</u>	<u>Monitor Points</u>
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.

PAGE 001

```

FORTRAN IV      702.1-11      PROGRAM MODAL
2021      INTER=2 XING, XOUTQ, ACKQ, PING, XIN, PXOUTQ, PACKQ
2022      INTER=2 PING, FRG, STAT, PLW, CN, OUTPCT
2023      INTER=2 SETPRM, RSTPRM, SETPKP
2024      INTER=2 RSTPKP, Q1, Q2, RESM, AUTO, DEQUE
2025      LOGICAL#1 XOUTBF, XINBF, ACKSEQ, LIDF, OUTBF, INBF
2026      LOGICAL#1 IDATA, PRED
2027      LOGICAL#1 PACK, ETX, CR, LF, MONTOR, ISLID, DUM
2028      INTER=2 LTIME, ACDTIM, NEWTIM, OLDTIM, IRTIM, TIMLIM, ATIML
2029      REAL#4 FZEAS
2010      COMMON /MESS/ MESSQ
2011      COMMON /LPM/ OUTBF(256), INBF(256),
2012      IRTIM, INR, STAT, LTIME
2013      COMMON /QUE/ XING(16), PING(2), XOUTQ(16), PXOUTQ(2),
2014      ACKQ(16), PACKQ(2), INQ(16), PING(2),
2015      LIDF(256), OFLG, ALLFLG, IRSEND
2016      COMMON /RST/ RST(64), IPR, IPRSZ
2017      COMMON /TIM/ OLDTIM, TIMLIM, ACDTIM, ATIML
2018      COMMON /PACK/ PACK(256, 64)
2019      COMMON /SWT/ SETPRM, RSTPRM, SETPKP, RSTPKP
2020      COMMON /GLOB/ ISENT, PRCNT, ICLNTH, OUTPCT, IFULL, OUTQ,
2021      RESNM, IALRT, ISLID, MONTOR, LF, ETX, CR, DUM
2022      COMMON /FCOM/ ICOM(3, 6), IDATA(80), ICRAN, MON, PTR, FZEAS,
2023      ITYPE, IAREA, ITERM, PRED(12, 1)
2024      CALL MASIE
2025      CALL INIT
2026      CALL ENABLE(1)
2027      5 CONTINUE
2028      25 IF(PING(1)) .LT. PING(2)) GOTO 40
2029      Q2=DEQUE(PING, INQ, 1)
2030      CALL ENABLE(1)
2031      DO 30 I=1, 80
2032      IDATA(I)=PACK(I, Q2)
2033      CONTINUE
2034      CALL FIA
2035      IPR=IPR+1
2036      PRE(IPR)=Q2
2037      40 CALL STA20(15)
2038      IF(.EQ. 1) GOTO 100
2039      IF(LLFLG .GT. 0) CALL LINLOS
2040      80 IF(OUTPCT .EQ. 1) GOTO 100
2041      IF(IFULL .EQ. 1) .OR. (ISENT .EQ. 1)) GOTO 100
2042      IF(PING(1)) .LT. PING(2)) GOTO 100
2043      CALL ENABLE(0)
2044      Q1=DEQUE(PING, XING, 1)
2045      CALL ENABLE(1)
2046      OUTQ=Q1
2047      65 CALL DESTR(PACK(254, OUTQ), Q2)
2048      DO 90 I=1, Q2
2049      OUTBF(I)=PACK(I, OUTQ)
2050

```

PAGE 222

```

FORTRAN IV      V02.1-11
0057      90      CONTINUE
0058      OUTF(02+1)=0
0059      IP5=OUTF(5)
0060      OUTF(02+2)=LIDF(IPT)
0061      IFULL=1
0062      ISENT=1
0063      CALL LPOUT(02+2)
0064      IREND=1
0065      ACTIM=0
0066      IWRSTM=0
0067      IWR=1
0068      INFCT=0
0069      100      OLDTIM=NEWTIM
0070      110      NEWTIM=TIME
0071      IF(LWR.EQ.0) GOTO 120
0072      IWRSTM=IWRSTM+(NEWTIM-OLDTIM)
0073      IF((IWRSTM.LT. TIMLM) GOTO 120
0074      CALL WTKEN
0075      IWRSTM=0
0076      IWR=0
0077      IWR=0
0078      IWR=0
0079      120      IP(ISENT.EQ.0) GOTO 130
0080      ACTIM=ACTIM+(NEWTIM-OLDTIM)
0081      IF(ACTIM.LT. ATIMLM) GOTO 130
0082      CALL ENAB(0)
0083      CALL ACENAB(0)
0084      CALL ENAB(1)
0085      130      CONTINUE
0086      GOTO 5
0087      END
0088
0089

```

PAGE 001

```
FORTRAN IV      V02.1-11
0001      FUNCTION IGCTSP(N)
0002      LOGICAL*1 ETX,CR,LF,MONITOR,ISLID,DUM
0003      INTEGER*2 FREE,FLWCNT
0004      INTEGER*2 OUTPCT,OUTQ,RESNLM
0005      COMMON /FREE/ FREE(64),IFR,IFRSZ
0006      COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTPCT,IFULL,OUTQ,
0007      IFR,IFRSZ,RESNLM,IALTRT,ISLID,MONITOR,IF,ETX,CR,DUM
0008      IF(IFR.LT.1) CALL INIT
0009      IGCTSP=FREE(IFR)
0010      IFR=IFR-1
0011      RETURN
0012      END
```

PAGE 001

FORTRAN IV 002.1-11

```

0001 SUBROUTINE ENQUR(A,B,N)
0002 LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003 INTEGER*2 XINQ,FXINQ,XOUTQ,PACKQ,ACKQ,PACKQ,INC,PINC
0004 INTEGER*2 FLWCNT,OUTQ,A(2),B(16)
0005 INTEGER*2 RESNLM,OUTFCT
0006 COMMON /QUE/ XINQ(16),PXINQ(2),YOUTQ(16),PXOUTQ(2),
0007         ASQC(16),PACKQ(2),INC(16),PINC(2),
0008         /GLOS/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0009         /RESNLM,IALTRT,ISLID,MONTOR,IF,ETX,CR,DUM
0010 IQHEAD=A(1)
0011 IQTAIL=A(2)
0012 IF(IQTAIL.EQ.1) GOTO 20
0013 IQTAIL=IQTAIL-1
0014 B(IQTAIL)=N
0015 A(2)=IQTAIL
0016 GOTO 999
0017 20 IF(IQHEAD.GE.(IQLNTH)) GOTO 40
0018 NN=IQHEAD-IQTAIL
0019 DO 30 I=1,NN+1
0020 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

```

PAGE 001

FORTRAN IV V02.1-11

```

0001 FUNCTION DEQUE(A,B,LD)
0002 LOGICAL*1 ETX,CH,LF,DUM,ISLID,MONTR
0003 INTEGER*2 XINQ,PXINQ,XOUTQ,PXOUTQ,ACKQ,PACKQ,INQ,PINQ
0004 INTEGER*2 FLWNT,A(2),R(16),DEQUE,OUTFCT,OUTQ,RESNLM
0005 COMMON /QUE/ XINQ(16),PXINQ(2),YOUTQ(16),PXOUTQ(2),
0006 ACKQ(16),PACKQ(2),INQ(16),PING(2)
0007 COMMON /GLOB/ ISENT,FLWNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0008 RESNLM,IALTRT,ISLID,MONTR,LF,ETY,CH,DUM
0009 IQHEAD=A(1)
0010 DEQUE=B(IQHEAD)
0011 IF(ID.NE.1) GOTO 999
0012 A(1)=IQLNTH
0013 A(2)=IQLNTH+1
0014 GOTO 999
0015 A(1)=IQHEAD-1
0016 A(1)=IQHEAD-1
0017 RETURN
0018 END

```

PAGE 001

X0FORTRAN IV V02.1-11

```

0001 SUBROUTINE ACKNAK(N)
0002 INTEGER*2 FLWCNT,OUTFCT,STAT,XINQ,PXINQ,PXOUTQ
0003 INTEGER*2 ACKQ,PACKQ,PINQ,FRF,OUTQ,XOUTQ
0004 INTEGER*2 T1,T2,T3,T4,T5,RESNLM
0005 LOGICAL*1 EFX,CR,LF,MONTOH,ISLID,DUM
0006 LOGICAL*1 PACK,OUTFF,INBF,LCONT(40)
0007 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0008 INTEGER*2 LTIME,IWRITM
0009 REAL*8 COMT(5)
0010 EQUIVALENCE(COMT,LCONT)
0011 DATA COMT(1)/'MESSAGE',/COMT(2)/'NOT SENT',/COMT(3)/' FROM
0012 1 COMT(4)/'NODE',/COMT(5)/' TO NODE'
0013 COMMON /GLOB/ ISENT,FLWCNT,IQLENTH,OUTFCT,IFULL,OUTQ,
0014 COMMON /DFM/ OUTBF(256),INBF(256),
0015 1 IWRITM,IWRIT,STAT,LTIME
0016 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),ACKQ(16),
0017 1 PACKQ(2),INQ(16),PIHQ(2)
0018 COMMON /FRF/ FRF(64),IFR,IFRSZ
0019 COMMON /MESS/ MESSEQ
0020 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
0021 1 IOFLG,LLFLG,IRSEND
0022 IF (N .NE. 1) GOTO 10
0023 IFR=IFR+1
0024 DO 5 I=1,256
0025 5 PACK(I,OUTQ)=0
0026 CONTINUE
0027 IFULL=0
0028 ISENT=0
0029 IRSEND=0
0030 FREE(IFR)=OUTQ
0031 GOTO 999
0032 10 T2=PACK(256,OUTQ)
0033 T2=T2+1
0034 IF(T2 .LE. RESNLM) GOTO 120
0035 T3=PACK(3,OUTQ)
0036 T4=MOD(T3,64)
0037 IF(T4 .LT. 32) GOTO 110
0038 T5=IGETSP(N)
0039 IF(MESSEQ .EQ. 126) MESSEQ=0
0040 MESSEQ=MESSEQ+1
0041 PACK(1,T5)=0
0042 PACK(2,T5)=MESSEQ
0043 PACK(3,T5)=0
0044 PACK(4,T5)=0
0045 PACK(5,T5)=25
0046 PACK(6,T5)=ISLID
0047 DO 20 I=7,60
0048 20 PACK(I,T5)="040
0049 CONTINUE
0050 DO 30 I=7,9
0051 30 PACK(I,T5)=LF
0052 34 CONTINUE
0053

```

PAGE 002

```

FORTRAN IV      V02.1-11
0054          DO 40 I=1,8
0055             PACK(I+9,T5)=LCOMT(I)
0056          CONTINUE
0057          DO 50 I=9,16
0058             PACK(I+9,T5)=LCOMT(I)
0059          CONTINUE
0060          DO 60 I=17,22
0061             PACK(I+9,T5)=LCOMT(I)
0062          CONTINUE
0063          DO 70 I=25,29
0064             PACK(I+7,T5)=LCOMT(I)
0065          CONTINUE
0066          ENCODE(3,80,PACK(38,T5)) ISLID
0067          FORMAT(I3)
0068          DO 90 I=33,40
0069             PACK(I+11,T5)=LCOMT(I)
0070          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          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(IASND.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(IASND.EQ.1) GOTO 999
0101          CALL ENQUE(PXINQ,XINQ,OUTQ)
0102          RETURN
0103          END

```

PAGE 201

FORTRAN IV V02.1-11

```

0001 SUBROUTINE IMPQT(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,INEF
0005 LOGICAL*1 PACK_MONTOR,ISLID,ETX,CR,LF,DUM
0006 INTEGER*2 LTIME,IWRITM
0007 COMMON /DFM/ OUTBF(256),INBF(256),
1 IWRITM,IWRT,STAT,LTIME
1 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
1 LIDFD(256),IOFLG,LLFLG,IRSEND
1 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
1 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
1 COMMON /GLOB/ ISENT,FLWCNT,IQNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALERT,ISLID,MONTOR,LF,ETX,CR,DUM
1 IF(L.LT.1) GOTO 30
LI=L-2
LI=INBF(5)
II=INBF(6)
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
0046 END

```

PAGE 021

FORTRAN IV V02.1-11

```

0001 SUBROUTINE LPINT(LI)
0002 INTEGER*2 STAT,FLWNT,OUTFCT,OUTC,RESNLM
0003 INTEGER*2 CCI,CC2,CC3,CC4,CC5,CC6,CC7
0004 LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTR,ISLID,DUM
0005 INTEGER*2 LTIME,IWRITM
0006 COMMON/DFM/OUTBF(256),INBF(256),IWRITM,IWRT,STAT,LTIME
0007 COMMON/GLOB/ISENT,ELWNT,IQLNTH,OUTFCT,IFULL,OUTC,
1 RESNLM,IALRHT,ISLID,MONTR,LF,ETX,CR,DUM
0008 CCI=INBF(3)
0009 CC2=INBF(4)
0010 IF(CCI.LI.0) GOTO 25
0011 IF(CCI.EQ.0.AND.CC2.EQ.0) GOTO 40
0012 IF(CCI.EQ.0.AND.CC2.EQ.0) GOTO 40
0013 CC3=MOD(CCI,2)
0014 CC4=MOD(CCI,256)
0015 IF((CC3.GE.1.OR.CC4.GE.128).AND.ISENT.EQ.1) GOTO 20
0016 IF((INBF(1).EQ.85).AND.INBF(2).EQ.170) GOTO 99
0017 CC5=MOD(INBF(3),54)
0018 IF(CCI.GE.32) CALL INPTQ(LI)
0019 GOTO 99
0020 CC7=MOD(CCI,2)
0021 IF(CC7.GE.1) GOTO 30
0022 CALL ACKNAK(0)
0023 GOTO 99
0024 CALL ACKNAK(1)
0025 GOTO 99
0026 CALL INPTQ(LI)
0027 CONTINUE
0028 RETURN
0029 END

```

PAGE 001

FORTRAN IV V02.1-11

```

0001 SUBROUTINE INIT
0002 REAL*4 FMEAS
0003 REAL*8 FTR
0004 INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,FINQ,FREE
0005 INTEGER*2 STAT,FLCNT,OUTFCT
0006 INTEGER*2 SETPRM,RSTPRM,SETRP,OUTQ,RESNLM,RSTBKP
0007 LOGICAL*1 DUM,DATA,PRED
0008 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTBF,INBF
0009 LOGICAL*1 PACK,ETX,CR,LF,MONTR,ISLID
0010 INTEGER*2 LTIME,OLDTIM,TIMLIM,ACTIM,ATIMLM,IWRTTM
0011 COMMON /DFM/ OUTBF(256),INBF(256)
1 IWRTTM,IWRT,STAT,LTIME
1 COMMON /QUE/ XINQ(16),PACKQ(2),INC(16),PINQ(2)
1 COMMON /MESS/ MESSEQ
1 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
1 IOPLG,LLFLG,IHSEND
1 COMMON /FRE/ FREE(64),IFR,IFRSZ
1 COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
1 COMMON /SWT/ SETPRM,RSTPRM,SETRP,SETRP,RSTBKP
1 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
1 COMMON /FCOM/ ICONN(3,6),IDATA(80),ICHAN,MON,FTR,FMEAS,
5 ITYPE,IAREA,IITEM,PRED(1014)
CALL RAM(0,9,4)
CALL RAM(0,255,0)
MESSEQ=0
IOPLG=0
IRSEND=0
LLFLG=0
IFRSZ=64
INFLCT=0
ISLID=27
CR='015
LF='012
ETX= 003
ICHNSW=0
RESNLM=2
IQLNTH=16
IALTRT=0
FLWCNT=0
IWRTTM=0
TIMLIM=35
SETRP=4
RSTPRM=32
SETRP=8
RSTBKP=64
ATIMLM=200
MONTR=25
IWRT=0
ISWICH=0
OUTFCT=0
IFULL=0

```



PAGE 001

FORTRAN IV V02.1-11

```

0001 SUBROUTINE LINLOS
0002 REAL*8 RMI(5),LIN08,LIN18
0003 INTEGER*2 FLMCNT,OUTFC,OUTQ,RESNLM,TI
0004 INTEGER*2 XINQ,PXING,XOUT,PKOUT,ACKQ,PACKQ,PINQ
0005 LOGICAL*1 YOUTBF,XINBF,ACKSEQ,LIFED
0006 LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0007 LOGICAL*1 PACK_M1(40),LIN0(8),LIN1(8)
0008 COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PKOUTQ(2),ACKQ(16),
0009 & PACKQ(2),INQ(16),PINQ(2)
0010 & IOFLG,LLFLG,IRSEND
0011 COMMON /GLOB/ISENT,FIMCNT,IOINTH,OUTFCT,IFULL,OUTQ,
0012 & RESNLM,IALTRI,ISLID,MONTR,IF,ETX,CR,DUM
0013 COMMON /MESS/ MESSEQ(256,64)
0014 DATA LIN08,LIN18/'PRIMARY ','BACKUP '//
0015 DATA RMI(1),RMI(2)/'LOSS OF ','MODULATI'//
0016 DATA RMI(3),RMI(4)/'ON ON LO ','OF AT NO'//
0017 DATA RMI(5)/'DE 27'//
0018 EQUIVALENCE(LIN0,LIN08)
0019 EQUIVALENCE(LIN1,LIN18)
0020 IS=LLFLG
0021 CALL ENABLE(0)
0022 TI=ICETSP(N)
0023 CALL ENABLR(1)
0024 PACK(1,TI)=0
0025 IF(MESSEQ.EQ.126) MESSEQ=0
0026 MESSEQ=MESSEQ+1
0027 PACK(2,TI)=MESSEQ
0028 PACK(3,TI)=0
0029 PACK(4,TI)=0
0030 PACK(5,TI)=25
0031 PACK(6,TI)=ISLID
0032 DO 20 I=7,9
0033 PACK(I,TI)=LF
0034 CONTINUE
0035 DO 30 I=1,22
0036 PACK(I+9,TI)=M1(I)
0037 IF(IS.EQ.1) GOTO 50
0038 DO 40 I=1,8
0039 PACK(I+31,TI)=LIN0(I)
0040 GOTO 70
0041 DO 60 I=1,8
0042 PACK(I+31,TI)=LIN1(I)
0043 DO 80 I=23,40
0044 PACK(I+17,TI)=M1(I)
0045 PACK(58,TI)=CR
0046 PACK(59,TI)=LF
0047 PACK(60,TI)=ETX
0048 CALL ENSTR(PACK(254,TI),60)
0049 CALL ENABLR(0)
0050 CALL ENQUE(PXING,XINQ,TI)
0051 CALL ENABLE(1)
0052
0053

```

PAGE 002

FORTRAN IV V02.1-11  
0054 LLFLG=0  
0055 RETURN  
0056 END

PAGE 001

```

EFCSTRAN IV      V02.1-11
0001      SUBROUTINE F1AC
0002      LOGICAL*1 PRED, IDATA
0003      REAL*4 FMEAS
0004      REAL*8 FTR
0005      COMMON /FCOM/ ICONN(3,6), I1ATA(50), ICHAN, MON, FIR, FMEAS,
&      ITYPE, IAREA, ITERM, PRED(1014)
0006      DECODE(2,4, I1ATA(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      ITYPE=1
0022      GOTO 60
0023      ITYPE=3
0024      GOTO 60
0025      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      CALL SDRPT(3)
0030      RETURN
0031      ITYPE=IDATA(25)
0032      IF(ITYPE .EQ. 1 .OR. ITYPE .EQ. 3 .OR.
&      ITYPE .EQ. 5) GOTO 60
0034      RETURN
0035      IF(IDATA(17) .EQ. 1) GOTO 70
0037      IF(ITYPE .EQ. 5) ICHAN=ICHAN+1000
0039      CALL SDRPT(2)
0040      RETURN
0041      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      CALL SDRPT(1)
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      IAREA=1
0060      GOTO 130
0061      100    IAREA=2

```

ISEND ALARM REPORT

ISEND AMBER REPORT

ISEND RED REPORT

PAGE 002

```

FORTRAN IV      V02.1-11
0062      CALL SNIPT(4)
0063      GOTO 130
0064      IAREA=3
0065      130 IF(ICONN(IAREA,ITYPE) .EQ. 0) GOTO 140
0067      RETURN
0068      140 ICONN(IAREA,ITYPE)=ICRAN
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      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

```

PAGE 001

FORTRAN IV V02.1-11

```

0001 SUBROUTINE SDRPT(IC)
0002 REAL*8 HEAD(3),PTR
0003 INTEGER*2 XING,XOUTQ,ACKQ,PXING,XOUTQ,PACKQ,PINQ
0004 INTEGER*2 FLWNT,OUTFCT,OUTQ,RESALM
0005 LOGICAL*1 PACK,DAEH(24),IDATA,PRES,ISLID,MONTR
0006 LOGICAL*1 LF,CR,ETX,DUM
0007 REAL*4 FMEAS
0008 EQUIVALENCE (HEAD,DAEH)
0009 DATA HEAD(1),TRUNK
0010 COMMON PACK(256,64)
0011 & ITYPE,IAREA,ITERM,PRED(1014)
0012 COMMON /MESS/ MESSEQ
0013 COMMON /GLOB/ ISENT,FLWNT,IQINTH,OUTFCT,IFULL,OUTQ,
0014 & RESNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0015 COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),XOUTQ(2),
0016 & ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0017 IF (IC .EQ. 3) GOTO 100
0018 IF (IC .EQ. 4) GOTO 200
0019 CALL ENABLE(0)
0020 K2=IGETSP(N)
0021 CALL ENABLE(1)
0022 IF(MESSEQ .EQ. 126) MESSEQ=0
0023 MESSEQ=MESSEQ+1
0024 PACK(1,K2)=0
0025 PACK(2,K2)=MESSEQ
0026 PACK(3,K2)=0
0027 PACK(4,K2)=0
0028 PACK(5,K2)=24
0029 PACK(6,K2)=ISLID
0030 PACK(7,K2)=177
0031 DO 70 I=7,14
0032 PACK(I+1,K2)=IDATA(I)
0033 CONTINUE
0034 ENCODE(2,80,PACK(16,K2)) ICHAN
0035 FORMAT(A2)
0036 PACK(18,K2)=IC
0037 DO 86 J=19,40
0038 PACK(J,K2)=0
0039 PACK(40,K2)=CR
0040 PACK(41,K2)=LF
0041 PACK(42,K2)=ETX
0042 CALL ENSTR(PACK(254,K2),42)
0043 CALL ENABLE(0)
0044 CALL ENQUE(PXING,XING,K2)
0045 CALL ENABLE(1)
0046 CONTINUE
0047 CALL ENABLE(0)
0048 KI=IGETSP(N)
0049 CALL ENABLE(1)
0050 DO 10 J=1,62
0051 PACK(J,K1)=040
0052 IF(MESSEQ .EQ. 126) MESSEQ=0
0053 MESSEQ=MESSEQ+1

```

EVENT REPORT TO SDCA

PAGE 002

```

FORTRAN IV      V02.1-11
0056      PACK(1,K1)=0
0057      PACK(2,K1)=MESSEQ
0058      PACK(3,K1)=0
0059      PACK(4,K1)=0
0060      PACK(5,K1)=ITRM
0061      PACK(6,K1)=ISLID
0062      DO 20 I=1,5
0063      PACK(I+9,K1)=DAEH(I)
0064      CONTINUE
0065      DO 30 I=9,15
0066      PACK(I+11,K1)=DAEH(I)
0067      CONTINUE
0068      DO 40 I=17,20
0069      PACK(I+14,K1)=DAEH(I)
0070      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      CONTINUE
0076      ENCODE(4,60,PACK(51,K1)) ICHAN
0077      FORMAT(14)
0078      ENCODE(2,65,PACK(61,K1)) IC
0079      FORMAT(12)
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(PXING,XINQ,K1)
0086      CALL ENABLE(1)
0087      RETURN
0088      CALL ENABLE(0)
0089      K1=IGTSP(N)
0090      K2=IGTSP(N)
0091      CALL ENABLE(1)
0092      DO 110 I=1,256
0093      PACK(I,K1)="40
0094      PACK(I,K1)="40
0095      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)=ITRM
0104      PACK(6,K1)=ISLID
0105      DO 120 I=7,10
0106      PACK(I,K1)=007
0107      CONTINUE
0108      DO 130 I=7,10
0109      PACK(I+4,K1)=IDATA(I)
0110      CONTINUE

```

PAGE 003

```

FORTRAN IV      V02.1-11

0111  ENCODE(2,140,PACK(16,K1)) IDATA(11)
0112  ENCODE(2,140,PACK(18,K1)) IDATA(13)
0113  FORMAT(I2)
140   PACK(20,K1)=CR
0114  PACK(21,K1)=LF
0115  PACK(22,K1)=FTX
0116  CALL ENSTR(PACK(254,K1),22)
0117  IF(MESSEQ.EQ.126) MESSEQ=0
0118  MESSEQ=MESSEQ+1
0119  PACK(1,K2)=0
0120  PACK(2,K2)=MESSEQ
0121  PACK(3,K2)=0
0122  PACK(4,K2)=0
0123  PACK(5,K2)=24
0124  PACK(6,K2)=ISLID
0125  DO 150 I=7,14
0126  PACK(I,K2)=IDATA(I)
0127  CONTINUE
150   PACK(18,K2)=IC
0128  PACK(19,K2)='015'
0129  PACK(20,K2)='012'
0130  PACK(21,K2)='003'
0131  CALL ENSTR(PACK(254,K2),21)
0132  CALL ENABLE(0)
0133  CALL ENQUE(PXINQ,XINQ,K1)
0134  CALL ENQUE(PXINQ,XINQ,K2)
0135  CALL ENABLE(1)
0136  RETURN
0137  CALL ENABLE(0)
0138  K1=ICETSP(N)
0139  CALL ENABLE(1)
200   IF(MESSEQ.EQ.126) MESSEQ=0
0140  MESSEQ=MESSEQ+1
0141  PACK(1,K1)=0
0142  PACK(2,K1)=MESSEQ
0143  PACK(3,K1)=0
0144  PACK(4,K1)=0
0145  PACK(5,K1)=28
0146  PACK(6,K1)=ISLID
0147  PACK(7,K1)='5'
0148  DO 210 I=7,14
0149  PACK(I-1,K1)=IDATA(I)
210   CONTINUE
0150  ENCODE(2,220,PACK(16,K1)) ICHEAN
0151  FORMAT(A2)
0152  PACK(18,K1)=IC
0153  CALL ENSTR(PACK(254,K1),18)
0154  CALL ENABLE(0)
0155  CALL ENQUE(PXINQ,XINQ,K1)
0156  CALL ENABLE(1)
0157  RETURN
0158  END
2164

```

PAGE 001

FORTRAN IV V02.1-11

```

0001 SUBROUTINE FAULT(IC,IT)
0002 LOGICAL*1 IMSG1(40),IMSG2(56),PACK,IData,PRED
0003 LOGICAL*1 CR,CTX,DUM,ISLID,LF,MONTR
0004 INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,PXOUTQ,PACKQ,PINQ
0005 INTEGER*2 FLMCNT,OUTFCT,OUTQ,RESNLM
0006 REAL*4 FMEAS
0007 REAL*8 MSG1(5),MSG2(7),PTR
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,MONTR,LF,ETX,CF,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),'LINECHAN',MSG1(5),'MON PT',
& MSG2(4),'POSSIBLE',MSG2(5),'CAUSE',MSG2(6),'LINECHAN',
& MSG2(7),'MON PT'
0015 CALL ENABLE(0)
0016 K1=ICETSP(N)
0017 K2=ICETSP(N)
0018 CALL ENABLE(1)
0019 IF(IC.EQ.2) GOTO 100
0020 DO 80 ICRT=18,25,7
0021 K3=K1
0022 IF(ICRT.EQ.25) K3=K2
0023 IF(MESSEQ.EQ.126) MESSEQ=0
0024 MESSEQ=MESSEQ+1
0025 PACK(1,K3)=0
0026 PACK(2,K3)=MESSEQ
0027 PACK(3,K3)=0
0028 PACK(4,K3)=0
0029 PACK(5,K3)=ICRT
0030 PACK(6,K3)=ISLID
0031 DO 10 I=1,8
0032 PACK(I+6,K3)=IMSG1(I)
0033 CONTINUE
0034 DO 20 I=9,11
0035 PACK(I+6,K3)=IMSG1(I)
0036 CONTINUE
0037 ENCODE(2,25,PACK(18,K3)) IC
0038 FORMAT(I2)
0039 DO 30 I=12,16
0040 PACK(I+8,K3)=IMSG1(I)
0041 CONTINUE
0042 DO 35 I=17,22
0043 PACK(I+8,K3)=IMSG1(I)
0044 CONTINUE
0045 IF(IT.EQ.6) GOTO 45
0046 DO 40 I=29,32
0047 PACK(I+2,K3)=IMSG1(I)
0048 CONTINUE
0049 DO 45 I=33,36
0050 PACK(I+2,K3)=IMSG1(I)
0051 CONTINUE
0052

```

PAGE 002

```

FORTRAN IV      V02.1-11
0053      CONTINUE
0054      GOTO 55
0055      DO 50 I=25,28
0056      PACK(I+6,K3)=IMSG1(I)
0057      CONTINUE
0058      ENCODE(5,60,PACK(35,K3)) ICONN(IAREA,IT-1)
0059      FORMAT(I5)
0060      DO 70 I=33,40
0061      PACK(I+7,K3)=IMSG1(I)
0062      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)=BTX
0067      CALL ENSTR(PACK(254,K3),55)
0068      CONTINUE
0069      GOTO 210
0070      DO 200 ICRT=18,25,7
0071      K3=K1
0072      IF(ICRT.EQ.25) K3=K2
0073      IF(MESSEC.EQ.126) MESSEC=0
0074      MESSEC=MESSEC+1
0075      PACK(1,K3)=0
0076      PACK(2,K3)=MESSEC
0077      PACK(3,K3)=0
0078      PACK(4,K3)=0
0079      PACK(5,K3)=ICRT
0080      PACK(6,K3)=ISLID
0081      DO 110 I=1,8
0082      PACK(I+6,K3)=IMSG2(I)
0083      CONTINUE
0084      DO 120 I=9,16
0085      PACK(I+6,K3)=IMSG2(I)
0086      CONTINUE
0087      DO 130 I=17,24
0088      PACK(I+6,K3)=IMSG2(I)
0089      CONTINUE
0090      DO 140 I=25,32
0091      PACK(I+6,K3)=IMSG2(I)
0092      CONTINUE
0093      PACK(31,K3)=CR
0094      PACK(32,K3)=LF
0095      DO 150 I=33,40
0096      PACK(I+8,K3)=IMSG2(I)
0097      CONTINUE
0098      DO 150 I=33,40
0099      PACK(I+8,K3)=IMSG2(I)
0100      CONTINUE
0101      IF(IT.EQ.6) GOTO 165
0102      DO 160 I=41,44
0103      PACK(I+8,K3)=IMSG2(I)
0104      CONTINUE
0105      GOTO 175
0106      DO 170 I=45,48
0107      PACK(I+4,K3)=IMSG2(I)
0108      CONTINUE
0109      ENCODE(5,180,PACK(53,K3)) ICONN(IAREA,IT-1)

```

PAGE 003

```
FORTRAN IV      V22.1-11
0110 180      FORMAT(15)
0111      DO 190 I=49,56
0112          PACK(I+9,K3)=IMSG2(1)
0113 190      CONTINUE
0114          ENCODE(5,180,PACK(56,K3)) ICNN(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(PXING,XING,K1)
0123          CALL ENQUE(PXING,XING,K2)
0124          CALL ENABLE(1)
0125          RETURN
0126          END
```

FDM,MACRO MACRO V03.02B 00:35:52 PAGE 1

```

1 .TITLE FDM,MACRO
2 .SETTL MODE,27
3 .ILENT /V3.0/
4 .GLOFL LIUIMT,LIO,ENABLE,SWITCH,MASTER,TIME
5 .GLOBL WFOKEN,RAM,STATUS,STATB0
6 .GLOBL IROUT,RSSTART,RPNT,DESTR,ENSTR,LPIAPT
7 .NLIST CND
8 .PSECT

```

```

R0 =%0
R1 =%1
R2 =%2
R3 =%3
R4 =%4
R5 =%5
SP =%6
PC =%7

```

;INTERFACE ADDRESS

```

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

```

;CHANGED

;INTERFACE OPTIONS

```

XCIO=1
XSIO=1
XGIO=1
NDXX=0
ND24=1
NEOST=0
DHOST=1

```

;LOCAL VARIABLES

```

ARG1= 2
ARG2= 4
ARG3= 6
ARG4= 10

```

```

BIT15= 100000
BIT14= 40000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4

```

```

10 000000
11 000001
12 000002
13 000003
14 000004
15 000005
16 000006
17 000007
18
19
20
21 172410
22 172412
23 172414
24 172416
25 172416
26 177560
27 177562
28 177564
29 177566
30
31
32
33
34
35
36
37
38
39
40
41
42 000001
43 000001
44 000001
45 000000
46 000001
47 000000
48 000001
49
50
51
52 000002
53 000004
54 000006
55 000010
56
57 100000
58 040000
59 020400
60 000200
61 000100
62 000040
63 000020
64 000010
65 000004

```

FDM.MACHO  
NOTE 27

MACRO V03.02E 00:36:52 PAGE 1-1

66  
67  
68

000002  
000001

BIT01= 2  
BIT00= 1

```
1 0000000 012700 000340
2 0000004 105400
3 0000006 012700
4 0000012 012720 000412
5 0000016 012710 000340
6 0000022 012700 000100
7 0000026 012710 001636
8 0000032 042737 040100
9 0000040 004767 000014
10 000044 005037 172414
11 000050 012700 000000
12 000054 105400
13 000056 002207
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
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
```

```
*****
**** PROCEDURE MASTER START UP
*****
MASTER: MOV #340, R0
MTPS R0
MOV #124, R0
MOV #L10, (R0)+
MOV #340, (R0)
MOV #100, R0
MOV #TIME, (R0)
BIC #40100, @CSR
PC, LIUINT
CLR @CSR
MOV #000, R0
MTPS R0
RTS PC
;LIU HANDLER
;PRI=7
;CLOCK VECTOR
;DISABLE INTERRUPTS
;CLEAR LIU CSR
;PRI=0
;CPU DOWN
```

```
*****
**** PROCEDURE INITIALIZE
*****
;CLEAR ACRAM
LIUINT: MOV #4354., @OPREG
TSTB @CSR
BPL .-4
CLR @CSR
MOV #4353., @OPREG
MOV #-256., R0
MOV #2311., @OPREG
TSTB @CSR
BPL .-4
CLR @CSR
INC R0
BNE 1$
;CLEAR INPUT/OUTPUT BUFFERS
MOV #10410, R1
MOV #10440, R0
JSR PC, ZEROP
MOV #1400, @OPREG
TSTB @CSR
BPL .-4
CLR @CSR
MOV #10510, R1
MOV #10640, R0
JSR PC, ZEROP
MOV #1400, @OPREG
TSTB @CSR
BPL .-4
CLR @CSR
172416 010402
172416 010400
172414 172414
172414 172414
172416 010401
172400 177400
172416 004407
172414 172414
172414 100375
100134 105037
105200 172414
001366
```

```
*****
**** PROCEDURE INITIALIZE
*****
;CLEAR ACRAM
LDADR
ADDRESS=0
GOOD WD
NO RETRY
SEL ACRAM
COUNT
WRITE A NULL
GOOD WD
NO LOOP UNTIL READY
COUNT+1
RBUFADR CMD
SEL INBUFO
SET POINTER=0
FALSE READ DATA
GOOD READ
NO RETRY
CLEAR DONE BIT
RBUFADR COMMAND
SEL INBUF1 COMMAND
POINTER=0
FALSE READ DATA
GOOD RD
NO RETRY
CLEAR DONE BIT
```

```
105 000234 012701 010510 MOV #10510, R1
107 000240 010540 MOV #10540, R0
108 000244 004767 JSR PC, ZEROFP
109 000250 012701 MOV #10710, R1
110 000254 012700 MOV #10740, R0
111 000260 004767 JSR PC, ZEROFP
112 000264 000440 BR STAT$
113 000266 012137 MOV #172416, R1
114 000272 012737 MOV #1400, G#OPREG
115 000300 005375 TSTB G#CSR
116 000304 100375 BPL -4
117 000306 105037 CLR B G#CSR
118 000312 013702 MOV G#IOBUF, R2
119 000316 042702 BIC #177400, R2
120 000322 010037 MOV R0, G#OPREG
121 000326 022702 CMP #0, R2
122 000332 001412 BEQ 3$
123 000334 012737 MOV #1400, G#OPREG
124 000342 105737 TSTB G#CSR
125 000346 100375 BPL -4
126 000350 105037 CLR B G#CSR
127 000354 005302 DEC R2
128 000356 000763 BR 2$
129 000360 010037 MOV R0, G#OPREG
130 000364 000207 RTS PC
131
132
133
134 000366 012737 MOV #172416, STAT$
135 000374 012737 MOV #12800, G#OPREG
136 000402 012737 MOV #12800, G#OPREG
137 000410 000207 RTS PC
138
139
;RBUFADR COMMAND
;SEL OUTBUF0 COMMAND
;RBUFADR COMMAND
;SEL OUTBUF0 COMMAND
;ZERO BUFFER POINTER
;GO CLEAR STATUS
;RBUFADR
;RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;FETCH POINTER
;CLEAR MST BYTE
;SEL BUFFER
;POINTER=0?
;YES RECHECK
;NO FALSE RD
;GOOD RD
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;LOCAL POINTER-1
;BR NTL 0
;SEL BUFFER
;RETURN
;WCR:RS
;RS(FALSE)
;RS
```

\*\*\*\*\*  
;\*\*\*\* PROCEDURE LIU-HANDLER  
\*\*\*\*\*

1	MOV	R0,	-(SP)
2	MOV	R1,	-(SP)
3	MOV	R2,	-(SP)
4	MOV	R3,	-(SP)
5	MOV	R4,	-(SP)
6	MOV	R5,	-(SP)
7	TST	#0CSR	
8	BPL	RTI\$	
9	BIC	#BIT14, @#CSR	172414
10	MOV	#4352, @#OPREG	040000
11	MOV	#1280, @#OPREG	172415
12	MOV	@#IOBUF, CAUSE	002400
13	BIC	#17400, CAUSE	172416
14			000502
15			177400
16			000474
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			
49			
50			
51			
52			
53			
54			
55			
56			
57			

;PREFORM OPERATION DEPENDING ON BITS

22	BITB	#BIT02, CAUSE	
23	BEQ	B5\$	000004
24	MOV	#4360, R2	010410
25	MOV	#4384, R3	010440
26	JSR	PC, @#MBF	004767
27	BITB	#BIT03, CAUSE	000270
28	BEQ	B4\$	000010
29	MOV	#448, R2	010510
30	MOV	#4512, R3	010640
31	JSR	PC, @#MBF	004767
32	BITB	#BIT04, CAUSE	000244
33	BEQ	B5\$	000020
34	BITB	#BIT07, CAUSE	001407
35	BEQ	B7\$	000200
36	MOV	#2, LFLG	000002
37	BITB	#BIT05, CAUSE	000040
38	BEQ	B6\$	002002
39	MOV	#1, LFLG	002001
40	BITB	#BIT06, CAUSE	000100
41	BEQ	B0\$	001000
42	CLR	IWRITM	005067
43	CLR	IWRT	001002
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			

FORTRAN  
0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053

EDM MACRO MACRO V03.02B 00:36:52 PAGE 3-1  
NODE 27

```

58 000704 132767 000002 000254 BIT# CAUSE
59 000712 001423 BEQ
60 000714 012701 #1AFF, R1
61 000720 012702 #4488, R2
62 000724 012703 #4512, R3
63 000730 012704 #BIT01, R4
64 000734 004767 PC, EMBF
65 000740 012705 #AREA, R5
66 000744 012706 #1, AREA
67 000752 012767 #2, DATA
68 000756 004767 JSR LFINPT
69
70 000762 012605 (SP)+, R5
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 BIS #BIT14, @#CSR
77 001004 000002 RTI
78
79 001006 012737 010600 MOV #4480, @#OPREG
80 001014 012737 002400 MOV #1280, @#OPREG
81 001022 013700 172416 MOV @#IOBUF, R0
82 001026 130400 BITB R4, R0
83 001030 001002 CRCOK
84 001032 012704 #1, R4
85 001036 012737 MOV #1, @#OPREG
86 001042 012737 001400 MOV #768, @#OPREG
87 001050 105737 TSTB @#CSR
88 001054 100375 BPL #-4
89 001056 013702 MOV @#IOBUF, R2
90 001062 042702 BIC #177400, R2
91 001066 010200 MOV R2, R0
92 001070 005400 NE5
93 001072 010037 MOV R0, @#WCR
94 001076 010137 MOV R1, @#BAR
95 001102 010337 172416 MOV R3, @#OPREG
96 001106 012737 001400 MOV #768, @#OPREG
97 001114 105737 TSTB @#CSR
98 001120 100375 BPL #-4
99 001122 012737 MOV #2704, @#OPREG
100 001130 000240 NOP
101 001132 105737 TSTB @#CSR
102 001136 100401 BMI DMAOK
103 001140 000240 MOV #2304, @#OPREG
104 001142 012737 TSTB @#CSR
105 001150 100375 BPL #-4
106 001154 100375 TST R4
107 001156 005704 TST ENDS
108 001160 100001 BPL R4, R2
109 001162 010402 RTS PC
110 001164 000207 CAUSE: .WORD 0
111 001166 000000 AREA: .WORD 0
112 001170 000000 .WORD 0
113 001172 001174 .WORD DATA
114 001174 000000 .WORD 0

```

```

;INBUFI FULL
;NO WERE DONE
;BUFFER ADDRESS
;RDBUFADR COMM
;SEL INBUFI
;CRC BIT
;GO EMPTY BUFFER
;DATA LINK AREA
;ONE VARIABLE
;CRC OR BYTE COUNT
;CALL FORTRAN QUE'ER
;RESTORE REGISTERS
;ENABLE INTERRUPTS
;RETURN FROM INTERRUPT
;READ STATUS 1
;RS
;FETCH STATUS
;GOOD CRC
;NO FLAG
;RDBUFADR
;RD
;GOOD RD
;REPLACE WITH POINTER
;CLEAR MST BITS
;SAVE IT
;S COMP
;BYTE COUNT NOW
;ADDRESS IN MEMORY
;SEL BUFFER
;FALSE RD
;DONE ON
;NO LOOP
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF HERE
;FALSE WD
;GOOD WD
;WAS CRC OK
;YES FLAG IT
;RETURN
;STATUS BYTE 0 HOLDER

```

EFCI 0001 0002 0003 0004 0005 0006 0007 0008 0009 0010 0011 0012 0013 0014 0015 0016 0017 0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032 0033 0034 0035 0036 0037 0038 0039 0040 0041 0042 0043 0044 0045 0046 0047 0048 0049 0050 0051 0052 0053 0054 0055 0056 0057 0058 0059 0060

FDM.MACRO MACRO V03.02B 00:36:52 PAGE 4  
NODE 27

```

1 1 .ENABLE LSR
2 2 *****
3 3 ***** PROCEDURE ENABLE *****
4 4 *****
5 5 .ENABLE: MOV @ARG1(R5), R1
6 6 #1, R1
7 7 BEQ EI$
8 8 #BIT14, @#CSR
9 9 RTS
10 10 PC
11 11 #BIT14, @#CSR
12 12 RTS
13 13
14 14 ;FORTRAN ENCODE/DECODE MACRO
15 15
16 16 ENSTR: MOV @ARG2(R5), @ARG1(R5)
17 17 PC
18 18 DESTR: MOV @ARG1(R5), @ARG2(R5)
19 19 PC
20 20 RTS
21 21
22 22 *****
23 23 ***** PROCEDURE LINE SWITCH *****
24 24 *****
25 25
26 26
27 27 ;CALL SWITCH(X) -SET OR RESET LINE SWITCH
28 28 ;
29 29 ;04=SET PRIMARY LINE
30 30 ;32=RESET PRIMARY LINE
31 31 ;08=SET BACKUP LINE
32 32 ;64=RESET BACKUP LINE
33 33 SWITCH: MOV @ARG1(R5), R1
34 34 #436R., @#OPREG
35 35 ADD #2304., R1
36 36 MOV R1, @#OPREG
37 37 TSTB @#CSR
38 38 BPL .-4
39 39 CLRB @#CSR
40 40 RTS
41 41
42 42

```



FDM-MACRO  
NOTE 27

MACRO V03.02B 00:36:52 PAGE 6

```

1 2 001456 012737 010620 172416 *TOKEN: MOV #4480., @#OPREG
3 8 001464 012737 002402 172416 #1280., @#OPREG
4 9 001472 013701 172416 MOV @#IOBUF, R1
5 10 001478 132701 000010 B11B #BIT03, R1
6 11 001502 001414 BEQ WTKG
7 12 001504 012737 010420 MOV #4368., @#OPREG
8 13 001512 012737 004423 MOV #2323., @#OPREG
9 14 001520 105737 172414 TSTB @#CSR
10 15 001524 100375 BPL #-4
11 16 001526 105037 172414 CLRB @#CSR
12 17 001532 000207 RTS PC
13 18 001534 012737 010740 WTKG: MOV #4576., @#OPREG
14 19 001542 012737 001634 172410 #WTL, @#BAR
15 20 001550 012702 000002 MOV #2, R2
16 21 001554 005402 NEG R2
17 22 001556 010237 172412 MOV #WCR
18 23 001562 012737 024000 MOV #10240., @#OPREG
19 24 001570 000240 NOP
20 25 001572 105737 172414 TSTB @#CSR
21 26 001576 000240 NOP
22 27 001600 105037 172414 CLRB @#CSR
23 28 001604 012737 010420 MOV #4368., @#OPREG
24 29 001612 012737 004422 172416 MOV #2322., @#OPREG
25 30 001620 105737 172414 TSTB @#CSR
26 31 001624 100375 BPL #-4
27 32 001626 105037 172414 CLRB @#CSR
28 33 001632 000207 RTS PC
29 34 001634 000 377 WTL: .BYTE 0,377
30
31
32
33
34
35
36
37
38
39
40
41 001636 005267 001006' TIME: INC LTIME
42 001642 022767 077777 001006' TIME: CMP #077777, LTIME
43 001650 001002 BNE 45
44 001652 005057 001006' CLR LTIME
45 001656 000002 RTI
46
47
48

```

```

*****
;**** PROCEDURE WRITE TOKEN (WTKEN)
;*****

```

```

;CALL WTKEN -LOADS AND WRITES A TOKEN TO LOOP

```

```

;READ STATUS(1)
;RS
;FETCH DATA
;BUFFER LOADED
;NO GENERATE TOKEN
;MODSTAT
;MARK OB0F/OB1F/WTCMD
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT

;SELECT BUFFER 1
;ADDRESS OF WRITE TOKEN
;# OF BYTES
;FORMAT IT
;DMA BYTE COUNT
;DELAY
;GOOD DMA
;ERROR IF HERE
;CLEAR DONE
;MODSTAT
;MARK OB1F/WTCMD
;VALID WRITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;RETURN
;A WTKEN

```

```

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

```

```

TIME: INC LTIME
CMP #077777, LTIME
BNE 45
CLR LTIME
RTI

```

FM:MACRO MACRO V03.02B 20:36:52 PAGE 7

NOTE 27

```

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
49

***** PROCEDURE ACRAM *****
*****
;CALL RAM(2,ADDR,DATA) -WRITE RAM ADDRESS WITH DATA
;CALL RAM(1,ADDR,DATA) -READ RAM DATA AT ADDRESS
;
;0110 =NHEAD
;2100 =DREAD
;0111 =NULL
;0000 =WOKEN

RAM: MOV @ARG1(R5), R0
CMP #0, R0
BNE RDRAM
WTRAM: MOV @ARG2(R5), R0
MOV @ARG3(R5), R1
MOV #4354., @OPREG
ADD #2304., R0
TSTB @#CSR
BPL -4
CLR @#CSR
MOV #4353., @OPREG
ADD #2304., R1
MOV R1, @#OPREG
TSTB @#CSR
BPL -4
CLR @#CSR
RTS PC
RDRAM: MOV @ARG2(R5), R0
MOV #4354., @OPREG
ADD #2304., R0
MOV R0, @#OPREG
TSTB @#CSR
BPL -4
CLR @#CSR
MOV #755., @OPREG
MOV #0, @#CSR
TSTB @#CSR
BPL -4
CLR @#CSR
MOV #177750., R0
MOV R0, @ARG3(R5)
RTS PC

; RAM
; WHICH OPERATION
; READ OF
; ADDRESS
; WRITE DATA
; SEL LDCR
; RD/DATA (ADDR)
; WRITE DATA
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; RD/DATA (CMD)
; WRITE
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; ADDRESS TO READ
; SEL LDCR
; RD/DATA (ADDR)
; WRITE DATA
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; READ DATA
; INVALID READ
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; FETCH DATA
; CLEAR BITS
; RTN DATA

```

FDM, MACRO  
 NODE 27

MACRO V03.02B 00:36:52 PAGE 8

1	002064	017502	000002	LPOUT:	MOV	GARG1(R5), R2	; BYTE COUNT
2	002070	012700	000340		MOV	#340, R0	; PRI=7
3	002074	106400			MTPS	R0, R0	
4	002076	012737	000000	172410	MOV	#OUTBF, @#BAR	
5	002104	005402			NEG	R2	; BUS ADDRESS
6	002106	010237	172412		MOV	R2, @#WCR	; 2'S COMP COUNT
7	002112	012737	010540	172416	MOV	#4448., @#OPREG	; COUNT
8	002120	012737	024000	172416	MOV	#10240., @#OPREG	; COUNT COMMAND
9	002126	000240			NOP		; DMA GO
10	002130	105737	172414		TSTB	@#CSR	; INTER ACE TIME
11	002134	000240			NOP		; DMA OK
12	002136	105037	172414		CLRB	@#CSR	; ERROR IF HERE
13	002142	012737	010740	172416	MOV	#4576., @#OPREG	; OBI COMMAND
14	002150	012737	001634	172410	MOV	##TD, @#BAR	; ADDRESS OF WRITE TOKEN
15	002156	012702	000002		MOV	#2, R2	; BYTE COUNT
16	002162	005402			NEG	R2	; FOR AT FOR BLUI
17	002164	010237	172412		MOV	R2, @#WCR	; DMA BYTE COUNT
18	002170	012737	024000	172416	MOV	#10240., @#OPREG	; FIRE DMA
19	002176	000240			NOP		; DELAY
20	002200	105737	172414		TSTB	@#CSR	; GOOD DMA
21	002204	000240			NOP		; ERROR IF NOT
22	002206	105037	172414		CLRB	@#CSR	; CLEAR DONE
23	002212	012737	010420	172416	MOV	#4368., @#OPREG	; MODSTAT
24	002220	012737	004403	172416	MOV	#2307., @#OPREG	; BUFFERS FULL
25	002226	100375	172414		TSTB	@#CSR	; GOOD WRITE
26	002232	100375			BPL	.-4	; NO LOOP UNTIL
27	002234	105037	172414		CLRB	@#CSR	; CLEAR DONE BIT
28	002240	012700	000000		MOV	#000, R0	; PRI=0
29	002244	106400			MTPS	R0	; LOWER CPU
30	002246	000207			RTS	PC	

```

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

```



EDM-MACRO MACRO V03.02B 00:36:52 PAGE 10

NODE 27

```

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 *****
243 *****

```

```

*****
***** PROCEDURE RESTART *****
*****
*****
;CALL RSTART(0) -CAUSES A SOFTWARE HALT
;CALL RSTART(1) -RESTARTS PROGRAM (MASTER)
;CALL RSTART(2) -LOAD MODE(173000)
*****
*****
RSTART: MOV GARG1(R5), R0
RSG: CMP #0, R0
BNE RS1
HALT
RTS PC
RS1: CMP #1, R0
BNE RS2
JSR PC, LIUINT
G4: MOV #40, R0
JMP (R0)
RS2: CMP #2, R0
BNE RST
JMP GROM
RST: RTS PC
ROM: .WORD 173000
*****
*****
;MODE
;LOAD ADDRESS
*****
;RESTART PROGRAM
*****
;INT LIU FIRST
;START ADDRESS
*****
;LOAD MODE
*****

```

EDM,MACRO MACRO V03.02B 00:36:52 PAGE 11

NOTE 27

```

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 *****

```

COMMON DATA AREA

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

```

OUTBF: .BLKB 256.
INBF: .BLKB 256.
IWRITM: .BLKW
IWRIT: .BLKW
STAT: .BLKW
LTIME: .BLKW
.EVEN

```

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

```

XOUTBF: .BLKB 256.
XINBF: .BLKB 256.
ACKSEC: .BLKB 256.
LIDFD: .BLKB 256.
IOFLG: .BLKW
LLFLG: .BLKW
IRSEND: .BLKW
.EVEN

```

000001

.END

EDM.MACRO SYMBOL TABLE MACRO V03.02B 00:36:52 PAGE 11-1

ACKSEQ	001000R	003	B0\$	000626R	ENSTR	001230RG	LPINPT=	***** G	RTI\$	000762R
AREA	001170R		B1\$	000704R	E0\$	001210R	LPOUT	002264RG	STAT	001004R
ARG1	= 000002		B2\$	000472R	E1\$	001220R	LTIME	001000RG	STATUS	001404RG
ARG2	= 000004		B3\$	000516R	G4\$	002426R	MASTER	000000RG	STAT\$	001306RG
ARG3	= 000006		B4\$	000542R	HRBUF	= 177562	NMAX	000000	STAT\$	000366R
ARG4	= 000010		B5\$	000570R	HRCSE	= 177560	ND24	= 000001	SWITCH	001250PG
BAR	= 172410		B6\$	000606R	HXBUF	= 177566	NHOST	= 000000	TIME	001630RG
BIT00	= 000001		B7\$	000552R	HXCSE	= 177564	OPREG	= 172416	WCR	= 172412
BIT01	= 000002		CAUSE	001166R	INBF	000400R	OUTBF	000000R	WTD	001634R
BIT02	= 000004		CHCOK	001036R	IOBUF	= 172416	RAM	001660RG	WTKG	001534R
BIT03	= 000010		CSR	= 172414	IOFLG	002000R	RDPM	002250PG	WTKEN	001456RG
BIT04	= 000020		DATA	001174R	IRSEND	002004R	RDRAM	001764R	WTRAM	001672R
BIT05	= 000040		DEST	001240RG	IWRT	001002R	ROM	002450R	XCIO	= 000001
BIT06	= 000100		DHOST	= 000001	IWRITM	001000R	RST	002446R	XGIO	= 000001
BIT07	= 000200		DMAOK	001142R	LIDFD	001400R	RSTART	002376RG	XINBF	000400R
BIT08	= 000400		EMBF	001006R	LIO	000412RG	RS0	002402R	XOUTBF	000000R
BIT14	= 040000		ENABLE	001175RG	LIUINT	000050RG	RS1	002414R	XSIO	= 000001
BIT15	= 100000		END\$	001164R	LLFLG	002002R	RS2	002434R	ZEROP	000266R

. ABS. 000000 000  
 002452 001  
 DPM 001010 002  
 EUPS 002006 003  
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 300 WORDS ( 2 PAGES)  
 DYNAMIC MEMORY AVAILABLE FOR 56 PAGES  
 LK:EDM,PK:EDM-DK:EDM.M27

PAGE 001

12:00:00

01-JUL-79

CONF27.COM

Program:

deewriter

FOR NOHAL/NDI LNE  
FOR NOBAL1/NDI LNE  
FOR FIAC/NDI LNE  
MAC FDM

PAGE 001

12:00:00

01-JUL-79

LNK27.COH

PROGRAM:

DESCRIPTION:

ASSIGN DXO: DN:

R LINK

DXI:NDRE 27,DXI:MAP=DXI:NDRE/C/L/W

DXI:FDR/C

DXI:NDRE L/E

DXI:FDR //

\*SIMKI

ASSIGN DXI: DN:

## 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

<u>Field</u>	<u>Description</u>	<u>Critical AUTODIN</u>	<u>Values AUTOVON</u>
1	Switch #	1	2
2	# of Transactions	512	256
3	# of Blocked Transactions	25	10
4	Transaction Queue Depth	25	10
5	# of Prescribed 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 interrupt handler for the 11/40 interface. It reads the message of input from the 11/40 and set a flag containing the byte count.

PAGE 001

FORTRAN IV Y02.1-11

```

PROGRAM MODAL
INTEGER*2 XING,XOUTQ,ACKQ,PXING,PXOUTQ,PACKQ
INTEGER*2 PINQ,STAT,FLWCNT,OUTFCT
INTEGER*2 SETPRM,RSTPRM,SETBKP
INTEGER*2 RSTBKP,Q1,Q2,RESNLM,OUTQ,DEQUE
LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDED,OUTBF,INBF
LOGICAL*1 SAVE1,SAVE2
LOGICAL*1 IDATA,ENC
LOGICAL*1 PACK,ETX,CR,LF,MONTOR,ISLID,DUM
INTEGER*2 LTIME,ACKTIM,NEWTIM,OLDTIM,IWRTM,TIMLIM,ATIMLM
COMMON /MESS/ MESSEQ
COMMON /DFM/ OUTBF(256),INBF(256),
1 IWRTM,IWRT,STAT,LTIME
1 COMMON /QUE/ XING(16),PXING(2),XOUTQ(16),PXOUTQ(2),
1 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
1 LIDED(256),IOFLG,LLFLG,IRSEND
COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLM
COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
1 COMMON /SCJM/SAVE1(64),SAVE2(64),ITEM,ICOUNT,I1,I2,IDATA(256),
5 ISWCH
COMMON /SCOM1/ISIM,IVC,IDC,IBC,ICC
COMMON/END/ENC(4)
CALL MASTER
CALL INIT
CALL ENABLE(1)
5 CONTINUE
25 IF(PINQ(1).LT.PINQ(2)) GOTO 40
CALL ENABLE(0)
Q2=DEQUE(PINQ,INQ,1)
CALL ENABLE(1)
DO 30 I=1,100
IDATA(I)=PACK(I+6,Q2)
30 CONTINUE
CALL SDCALP
IFR=IFR+1
FREE(IFR)=Q2
40 CALL STATE2(IS)
IF(IS.EQ.1) GO TO 100
IF(IOFLG.GT.0.AND.IRSEND.EQ.0) CALL SICA
IF(LLFLG.GT.0) CALL LINDOS
80 IF(OUTFCT.EQ.1) GOTO 100
IF(IFULL.EQ.1).OR.(ISENT.EQ.1) GOTO 100
IF(IRSEND.EQ.1) GOTO 85
IF(PXING(1).LT.PXING(2)) GOTO 100
CALL ENABLE(0)
Q1=DEQUE(PXING,XING,1)
CALL ENABLE(1)
OUTQ=Q1
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  
AD  
A078391



PAGE 002

```

FORTRAN IV      V02.1-11
0058          DO 90 I=1,Q2
0059          OUTBF(I)=PACK(I,OUTQ)
0060          CONTINUE
0061          90  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          IWRTM=0
0070          IWRT=1
0071          INFLOT=0
0072          OLDTIM=NEWTIM
0073          NEWTIM=LTIME
0074          IF(IWRT.EQ.0) GOTO 120
0076          IWRTM=IWRTM+(NEWTIM-OLDTIM)
0077          IF(IWRTM.LT.TIMLIM) GOTO 120
0079          CALL WTOKEN
0080          IWRTM=0
0081          IWRT=0
0082          120 IF(ISENT.EQ.0) GOTO 130
0084          ACKTIM=ACKTIM+(NEWTIM-OLDTIM)
0085          IF(ACKTIM.LT.ATIMLIM) GOTO 130
0087          CALL ENABLE(0)
0088          CALL ACKNAK(0)
0089          CALL ENABLE(1)
0090          130 CONTINUE
0091          GOTO 5
0092          END

```

PAGE 001

```
FORTRAN IV      V02.1-11
0001      FUNCTION IGETSP(N)
0002      LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0003      INTEGER*2 FREE,FLCNT
0004      INTEGER*2 OUTFCT,OUTQ,RESNLM
0005      COMMON /FRE/ FREE(G4),IFR,IFRSZ
0006      COMMON /GLOB/ ISENT,FLCNT,ICLNTH,OUTFCT,IFULL,OUTQ,
1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CF,DUM
C
0007      IF (IFR .LT. 1) CALL INIT
0009      IGETSP=FREE(IFR)
0010      IFR=IFR-1
0011      RETURN
0012      END
```

PAGE 001

FORTRAN IV V02.1-11

```

0001 SUBROUTINE ENQUE(A,B,N)
0002 LOGICAL*1 EFX,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),
0007     ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0008     RESNLM,IALTRT,ISLID,MONTOR,LF,EFX,CR,DUM
0009 10 IQHEAD=A(1)
0010 IQTAIL=A(2)
0011 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
0017 NN=IQHEAD-IQTAIL
0018 DO 30 I=1,NN+1
0019 30 B(IQLNTH+1-I)=B(IQHEAD+1-I)
0020 A(1)=IQLNTH
0021 A(2)=IQLNTH-NN
0022 GOTO 10
0023 40 CALL INIT
0024 999 RETURN
0025 END
0026

```

PAGE 001

FORTRAN IV V02.1-11

```

0001 FUNCTION DEQUE(A,B,LD)
0002 LOGICAL*1 ETX,CR,LF,DUM,ISLID,MONTOR
0003 INTEGER*2 XINQ,PXINQ,XOUTQ,PACKQ,ACKQ,PACKQ,INQ,PINQ
0004 INTEGER*2 FLWNT,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,FLWNT,IQLNTH,OUTFCT,IFULL,OUTQ,
1 RESNLM,IALTR,ISLID,MONTOR,LF,ETX,CR,DUM
0007 IQHEAD=A(1)
0008 DEQUE=B(IQHEAD)
0009 IF(ID.NE.1) GOTO 999
0010 IF(IQHEAD.NE.0) GOTO 10
0011 A(1)=IQLNTH
0012 A(2)=IQLNTH+1
0013 GOTO 999
0014 10 A(1)=IQHEAD-1
0015 999 RETURN
0016 END

```

PAGE 001

X0FORTRAN IV 002.1-1-11

```

0001 SUBROUTINE ACKMAK(N)
0002 INTEGER*2 FLWCNT,OUTFCT,STAT,XING,PXING,PXOUTQ
0003 INTEGER*2 ACKQ,PACKQ,PINQ,FRFQ,OUTQ,XOUTQ
0004 INTEGER*2 I1,I2,I3,I4,I5,RESNLM
0005 LOGICAL*1 ETX,CR,LF,MONTOR,ISLID,DUM
0006 LOGICAL*1 PACK,OUTF,INBF,LCOMT(4)
0007 LOGICAL*1 XOUTF,XINBF,ACKSEQ,LIDFD
0008 INTEGER*2 LTIME,IWRTM
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 IWRTM,IWRT,STAT,LTIME
0014 COMMON /QUE/ PACK(256,64)
0015 COMMON /FRF/ FRF(64),IFR,IFRSZ
0016 COMMON /MESS/ MESSEQ
0017 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 T2=PACK(256,OUTQ)
0031 T2=T2+1
0032 IF(T2.LE.RESNLM) GOTO 120
0033 T3=PACK(3,OUTQ)
0034 T4=MOD(T3,64)
0035 IF(T4.LT.32) GOTO 110
0036 T5=IGTSP(N)
0038 IF(MESSEQ.EQ.126) MESSEQ=0
0039 MESSEQ=MESSEQ+1
0041 PACK(1,T5)=0
0042 PACK(2,T5)=MESSEQ
0043 PACK(3,T5)=0
0044 PACK(4,T5)=0
0045 PACK(5,T5)=25
0046 PACK(6,T5)=ISLID
0047 DO 20 I=7,60
0048 PACK(I,T5)=""
0049 20 CONTINUE
0050 DO 30 I=7,9
0051 PACK(I,T5)=LF
0052 30 CONTINUE
0053

```

PAGE 002

```

FORTRAN IV      V02.1-11
0054      DO 40 I=1,8
0055      PACK(I+9,T5)=LCOMT(I)
0056      CONTINUE
0057      DO 50 I=9,16
0058      PACK(I+9,T5)=LCOMT(I)
0059      CONTINUE
0060      DO 60 I=17,22
0061      PACK(I+9,T5)=LCOMT(I)
0062      CONTINUE
0063      DO 70 I=25,29
0064      PACK(I+7,T5)=LCOMT(I)
0065      CONTINUE
0066      ENCODE(3,80,PACK(38,T5))ISLID
0067      FORMAT(13)
0068      DO 90 I=33,40
0069      PACK(I+11,T5)=LCOMT(I)
0070      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)=EIX
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      CONTINUE
0081      ISENT=0
0082      IRSEND=0
0083      IFULL=0
0084      IWR=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(IASEND.EQ.1) GOTO 999
0101      CALL ENQUE(PXINQ,XINQ,OUTQ)
0102      999 RETURN
0103      END

```

PAGE 001

```

FORTRAN IV      V02.1-11
0001      SUBROUTINE INPTQ(L)
0002      INTEGER*2 STAT,XINQ,PXINQ,XOUTQ,PKOUTQ,ACKQ
0003      INTEGER*2 PACKQ,PINQ,FLWCNT,OUTFCT,T1,RBSNLM,OUTQ
0004      LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD,OUTFBF,INBF
0005      LOGICAL*1 PACK,MONTR,ISLID,ETX,CR,LF,DUM
0006      INTEGER*2 LTIME,IWRITM
0007      COMMON /DFM/ OUTBF(256),INBF(256),
0008      1 IWRITM,IWRIT,STAT,LTIME
0009      1 COMMON /BUFS/ XOUTBF(256),XINBF(256),ACKSEQ(256),
0010      1 LIDFD(256),IOFLG,LLFLG,INSEND
0011      COMMON /QUE/  PACK(256,64)
0012      COMMON /GLOB/ XINQ(16),PXINQ(2),XOUTQ(16),PKOUTQ(2),
0013      1 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0014      1 COMMON /RBSNLM,IALTRT,ISLID,MONTR,LF,ETX,CR,DUM
0015      1 IF(L .LT. 1) GOTO 30
0016      LI=L-2
0017      II=INBF(6)
0018      IF(INBF(2) .EQ. ACKSEQ(II)) GOTO 20
0019      T1=IGETSP(N)
0020      DO 10 I=1,LI
0021      10 PACK(I,T1)=INBF(I)
0022      CALL ENSTH(PACK(254,T1),LI)
0023      CALL ENQUE(PINQ,INQ,T1)
0024      N=2
0025      GOTO 40
0026      N=1
0027      GOTO 40
0028      N=0
0029      DO 50 I=1,2
0030      50 OUTBF(I)=INBF(I)
0031      NN=1
0032      CONTINUE
0033      IF(N .EQ. 0) NN=12E
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
0043      ACKSEQ(II)=INBF(2)
0044      999 RETURN
0045      END
0046

```

```

FORTRAN IV      V02.1-1-11      PAGE 001

0001      SUBROUTINE LPINPT(LI)
0002      INTEGER*2 STAT,FLWCNT,OUTFCT,OUTFC,RESNLM
0003      INTEGER*2 CCI,CC2,CC3,CC4,CC5,CC6,CC7
0004      LOGICAL*1 OUTBF,INBF,ETX,CR,LF,MONTOR,ISLID,DUM
0005      INTEGER*2 LTIME,IWRTTM
0006      COMMON /DEM/OUTBF(256),INBF(256),I*RTTM,I*RT,STAT,LTIME
0007      COMMON /GLOB/ ISENT,FLWCNT,IQLENH,OUTFCT,IFULL,OUTQ,
1      RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0008      CCI=INBF(3)
0009      CC2=INBF(4)
0010      IF(CCI .LT. 0) GOTO 25
0011      IF(CCI .EQ. 0 .AND. CC2 .EQ. 0) GOTO 40
0012      CC3=MOD(CCI,2)
0013      CC4=MOD(CCI,256)
0014      IF ((CC3 .GE. 1 .OR. CC4 .GE. 128) .AND. ISENT .EQ. 1) GOTO 20
0015      IF ((INBF(1) .EQ. 85) .AND. INBF(2) .EQ. 170) GOTO 99
0016      CC5=MOD(INBF(3),64)
0017      IF(CC5 .GE. 32) CALL INPTQ(LI)
0018      GOTO 99
0019      20      CC7=MOD(CCI,2)
0020      IF(CC7 .GE. 1) GOTO 30
0021      25      CALL ACKNAK(0)
0022      GOTO 99
0023      30      CALL ACKNAK(1)
0024      GOTO 99
0025      40      CALL INPTQ(LI)
0026      99      CONTINUE
0027      RETURN
0028      END
0029
0030
0031
0032
0033
0034

```

PAGE 001

FORTRAN IV V02.1-11

```

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,LIDED,OUTRF,INBF
0006 LOGICAL*1 PACK,ETA,CR,LF,MONFOR,ISLID,IUM,IData
0007 LOGICAL*1 SAVE1,SAVE2
0008 INTEGER*2 LTIME,OLDTIM,TIMLIM,ACKTIM,ATIMLIM,IWRITM
0009 COMMON /DFM/ OUTRF(256),INBF(256)
1 IWRITM,IWRT,STAT,LTIME
1 COMMON /QUE/ XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
1 COMMON /BUFS/ ACKQ(16),PACKQ(2),INQ(16),PINQ(2),
1 COMMON /MESS/ MESSEQ
1 COMMON /FREE/ FREE(64),IFR,IFRSZ
1 COMMON /TIM/ OLDTIM,TIMLIM,ACKTIM,ATIMLIM
1 COMMON /SWT/ SETPRM,RSTPRM,SETBKP,RSTBKP
1 COMMON /GLOB/ ISEMT,FLWCNT,IQLENTH,OUTFCT,IFULL,OUTQ,
1 COMMON /SCOM/SAVE1(64),SAVE2(64),ITERM,ICOUNT,I1,I2,IData(256),
& ISWITCH
1 COMMON /SCOM1/ISIM,IVC,IDC,IBC,ICC
1 CALL RAM(0,2,4)
1 CALL RAM(0,255,0)
I1=0
I2=0
ICOUNT=0
ISIM=5
IVC=0
IDC=500
IBC=0
ICC=0
MESSEQ=0
IOFLG=0
IRSEND=0
LFLG=0
IFRSZ=64
INFCT=0
ISLID=28
CP=015
LF=012
ETA="003"
RESNLM=2
IQLENTH=16
IALTRT=0
FLWCNT=0
IWRITM=0
TIMLIM=50
SETPRM=4
RSTPRM=32
SETBKP=8
RSTBKP=C4

```

PAGE 002

```

FORTRAN IV      V02.1-11
0050      ATIMLM=200
0051      MONTOR=25
0052      IWRT=0
0053      ISWTC=2
0054      OUTPUT=0
0055      IFULL=0
0056      ISENT=0
0057      DO 40 I=1,20
0058          LIDPD(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          LIDPD(I)=0
0069      DO 60 I=40,44
0070          LIDFD(I)=1
0071      DO 70 I=45,59
0072          LIDPD(I)=0
0073      DO 80 I=60,64
0074          LIDFD(I)=2
0075      DO 90 I=65,79
0076          LIDPD(I)=0
0077      DO 100 I=80,84
0078          LIDPD(I)=4
0079      DO 110 I=85,99
0080          LIDFD(I)=0
0081      DO 120 I=100,104
0082          LIDPD(I)=5
0083      DO 130 I=105,256
0084          LIDFD(I)=0
0085          PINQ(1)=IQLNTH
0086          PINQ(2)=IQLNTH+1
0087          PACKQ(1)=IQLNTH
0088          PACKQ(2)=IQLNTH+1
0089          POUTQ(1)=IQLNTH
0090          POUTQ(2)=IQLNTH+1
0091          PXINQ(1)=IQLNTH
0092          PXINQ(2)=IQLNTH+1
0093      DO 10 I=1,IPRSZ
0094          PACK(255,I)=0
0095          PACK(256,I)=0
0096      DO 20 I=1,IPRSZ
0097          FREE(I)=I
0098      DO 30 I=1,256
0099          ACKSEQ(I)=256
0100          IFE=IPRSZ
0101          RETURN
0102      END

```

PAGE 001

FORTRAN IV V02.1-11

```

0001 SUBROUTINE LINLOS
0002 REAL*8 RMI(5),LINO8,LINI8
0003 INTEGER*2 FL*CNT,OUTFOT,OUTQ,RESNLM,TI
0004 INTEGER*2 XINQ,FXINQ,XOUTQ,FXOUTQ,ACKQ,PACKQ,PINQ
0005 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,LIDFD
0006 LOGICAL*1 ETX,CR,LF,MONTR,ISLID,DUM
0007 LOGICAL*1 PACK,M1(40),LIN0(8),LINI(8)
0008 COMMON/QUE/XINQ(16),FXINQ(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,FL*CNT,IQ*INTH,OUTFOT,IFULL,OUTQ,
1 RESNLM,IALRBT,ISLID,MONTR,LF,ETX,CR,DUM
0011 COMMON /MESS/ MESSEQ
0012 DATA LINO8,LINI8/'PRIMARY ','BACKUP '//
0013 DATA RMI(1),RMI(2)'/LOSS OF ','MOLULATI'//
0014 DATA RMI(3),RMI(4)'/ON ON LO','OP AT NO'//
0015 DATA RMI(5)'/DE 28'//
0016 EQUIVALENCE(LINO,LINO8)
0017 EQUIVALENCE(LINI,LINI8)
0018 EQUIVALENCE(LINI,LINI8)
0019 EQUIVALENCE(M1,RMI)
0020 IS=LLFLG
0021 CALL ENABLE(0)
0022 TI=IGETSP(N)
0023 CALL ENABLE(1)
0024 PACK(1,TI)=0
0025 IF(MESSEQ.EQ.126) MESSEQ=0
0026 MESSEQ=MESSEQ+1
0027 PACK(2,TI)=MESSEQ
0028 PACK(3,TI)=0
0029 PACK(4,TI)=0
0030 PACK(5,TI)=25
0031 PACK(6,TI)=ISLID
0032 DO 20 I=7,9
0033 PACK(I,TI)=LF
0034 CONTINUE
0035 DO 30 I=1,22
0036 PACK(I+9,TI)=M1(I)
0037 IF(IS.EQ.1) GOTO 50
0038 DO 40 I=1,8
0039 PACK(I+31,TI)=LINO(I)
0040 GOTO 70
0041 DO 60 I=1,8
0042 PACK(I+31,TI)=LINI(I)
0043 DO 80 I=23,40
0044 PACK(I+17,TI)=M1(I)
0045 PACK(58,TI)=CH
0046 PACK(59,TI)=LF
0047 PACK(60,TI)=ETX
0048 CALL ENSTR(PACK(254,I),60)
0049 CALL ENABLE(0)
0050 CALL ENQUE(PXINQ,XINQ,TI)
0051 CALL ENABLE(1)
0052
0053

```

PAGE 002

FORTRAN IV    V02.1-11  
0054            LFLG=0  
0055            RETURN  
0056            END



```

0001 SUBROUTINE SDCA
0002 LOGICAL*1 XOUTBF,XINBF,ACKSEQ,ILDFD,OUTRF,INBF
0003 LOGICAL*1 PACK,ETY,CR,LF,MONTOR,ISLID,IUM
0004 LOGICAL*1 SAVE1,SAVE2,BHEAD(24),IDATA,ENC
0005 INTEGER*2 XINQ,XOUTQ,ACKQ,PXINQ,FXOUTQ,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,LTRNK,BWID
0012 EQUIVALENCE(RHEAD,BHEAD)
0013 DATA AV/3,256,10,10,25,40,50,64,5,5,10,64,64,64,5,5/
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/'AAAAAAA','BBBBBBB', '11110'/
0018 COMMON/MESS/ MESSEQ
0019 COMMON/QUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
0020 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0021 COMMON/GLOB/ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0022 IOTIG,ILFLG,IIRSEN
0023 COMMON/RESNLM,IALRRT,ISLID,MONTOR,LF,ETY,CR,IUM
0024 COMMON/SCOM/SAVE1(64),SAVE2(64),ITERM,ICOUNT,I1,I2,IDATA(256),
0025 ISWICH
0026 COMMON/SCOM1/ISIM,IVC,IDC,IBC,ICC
0027 COMMON/END/ENC(4)
0028 1 IF(ISWICH .EQ. 0) GOTO 999
0029 DECODE(4,2,XINBF(1),ERR=999) AME(1)
0030 DECODE(4,2,XINBF(5),ERR=999) AME(2)
0031 DECODE(4,2,XINBF(9),ERR=999) AME(3)
0032 DECODE(4,2,XINBF(13),ERR=999) AME(4)
0033 DECODE(4,2,XINBF(17),ERR=999) AME(5)
0034 DECODE(4,2,XINBF(21),ERR=999) AME(6)
0035 DECODE(4,2,XINBF(25),ERR=999) AME(7)
0036 DECODE(4,2,XINBF(29),ERR=999) AME(8)
0037 DECODE(4,2,XINBF(33),ERR=999) AME(9)
0038 DECODE(4,2,XINBF(37),ERR=999) AME(10)
0039 DECODE(4,2,XINBF(41),ERR=999) AME(11)
0040 DECODE(4,2,XINBF(45),ERR=999) AME(12)
0041 DECODE(4,2,XINBF(49),ERR=999) AME(13)
0042 DECODE(4,2,XINBF(53),ERR=999) AME(14)
0043 DECODE(4,2,XINBF(57),ERR=999) AME(15)
0044 FORMAT(I4)
0045 2 IF(AME(1) .EQ. 2) GOTO 200
0046
0047 C
0048 C
0049 C
0050 C
0051 C
0052 C
0053 C
0054 C
0055 C
0056 C
0057 C
0058 C
0059 C
0060 C
0061 C
0062 C
0063 C
0064 C
0065 C
0066 C
0067 C
0068 C
0069 C
0070 C
0071 C
0072 C
0073 C
0074 C
0075 C
0076 C
0077 C
0078 C
0079 C
0080 C
0081 C
0082 C
0083 C
0084 C
0085 C
0086 C
0087 C
0088 C
0089 C
0090 C
0091 C
0092 C
0093 C
0094 C
0095 C
0096 C
0097 C
0098 C
0099 C
0100 C
0101 C
0102 C
0103 C
0104 C
0105 C
0106 C
0107 C
0108 C
0109 C
0110 C
0111 C
0112 C
0113 C
0114 C
0115 C
0116 C
0117 C
0118 C
0119 C
0120 C
0121 C
0122 C
0123 C
0124 C
0125 C
0126 C
0127 C
0128 C
0129 C
0130 C
0131 C
0132 C
0133 C
0134 C
0135 C
0136 C
0137 C
0138 C
0139 C
0140 C
0141 C
0142 C
0143 C
0144 C
0145 C
0146 C
0147 C
0148 C
0149 C
0150 C
0151 C
0152 C
0153 C
0154 C
0155 C
0156 C
0157 C
0158 C
0159 C
0160 C
0161 C
0162 C
0163 C
0164 C
0165 C
0166 C
0167 C
0168 C
0169 C
0170 C
0171 C
0172 C
0173 C
0174 C
0175 C
0176 C
0177 C
0178 C
0179 C
0180 C
0181 C
0182 C
0183 C
0184 C
0185 C
0186 C
0187 C
0188 C
0189 C
0190 C
0191 C
0192 C
0193 C
0194 C
0195 C
0196 C
0197 C
0198 C
0199 C
0200 C
0201 C
0202 C
0203 C
0204 C
0205 C
0206 C
0207 C
0208 C
0209 C
0210 C
0211 C
0212 C
0213 C
0214 C
0215 C
0216 C
0217 C
0218 C
0219 C
0220 C
0221 C
0222 C
0223 C
0224 C
0225 C
0226 C
0227 C
0228 C
0229 C
0230 C
0231 C
0232 C
0233 C
0234 C
0235 C
0236 C
0237 C
0238 C
0239 C
0240 C
0241 C
0242 C
0243 C
0244 C
0245 C
0246 C
0247 C
0248 C
0249 C
0250 C
0251 C
0252 C
0253 C
0254 C
0255 C
0256 C
0257 C
0258 C
0259 C
0260 C
0261 C
0262 C
0263 C
0264 C
0265 C
0266 C
0267 C
0268 C
0269 C
0270 C
0271 C
0272 C
0273 C
0274 C
0275 C
0276 C
0277 C
0278 C
0279 C
0280 C
0281 C
0282 C
0283 C
0284 C
0285 C
0286 C
0287 C
0288 C
0289 C
0290 C
0291 C
0292 C
0293 C
0294 C
0295 C
0296 C
0297 C
0298 C
0299 C
0300 C
0301 C
0302 C
0303 C
0304 C
0305 C
0306 C
0307 C
0308 C
0309 C
0310 C
0311 C
0312 C
0313 C
0314 C
0315 C
0316 C
0317 C
0318 C
0319 C
0320 C
0321 C
0322 C
0323 C
0324 C
0325 C
0326 C
0327 C
0328 C
0329 C
0330 C
0331 C
0332 C
0333 C
0334 C
0335 C
0336 C
0337 C
0338 C
0339 C
0340 C
0341 C
0342 C
0343 C
0344 C
0345 C
0346 C
0347 C
0348 C
0349 C
0350 C
0351 C
0352 C
0353 C
0354 C
0355 C
0356 C
0357 C
0358 C
0359 C
0360 C
0361 C
0362 C
0363 C
0364 C
0365 C
0366 C
0367 C
0368 C
0369 C
0370 C
0371 C
0372 C
0373 C
0374 C
0375 C
0376 C
0377 C
0378 C
0379 C
0380 C
0381 C
0382 C
0383 C
0384 C
0385 C
0386 C
0387 C
0388 C
0389 C
0390 C
0391 C
0392 C
0393 C
0394 C
0395 C
0396 C
0397 C
0398 C
0399 C
0400 C
0401 C
0402 C
0403 C
0404 C
0405 C
0406 C
0407 C
0408 C
0409 C
0410 C
0411 C
0412 C
0413 C
0414 C
0415 C
0416 C
0417 C
0418 C
0419 C
0420 C
0421 C
0422 C
0423 C
0424 C
0425 C
0426 C
0427 C
0428 C
0429 C
0430 C
0431 C
0432 C
0433 C
0434 C
0435 C
0436 C
0437 C
0438 C
0439 C
0440 C
0441 C
0442 C
0443 C
0444 C
0445 C
0446 C
0447 C
0448 C
0449 C
0450 C
0451 C
0452 C
0453 C
0454 C
0455 C
0456 C
0457 C
0458 C
0459 C
0460 C
0461 C
0462 C
0463 C
0464 C
0465 C
0466 C
0467 C
0468 C
0469 C
0470 C
0471 C
0472 C
0473 C
0474 C
0475 C
0476 C
0477 C
0478 C
0479 C
0480 C
0481 C
0482 C
0483 C
0484 C
0485 C
0486 C
0487 C
0488 C
0489 C
0490 C
0491 C
0492 C
0493 C
0494 C
0495 C
0496 C
0497 C
0498 C
0499 C
0500 C

```

PAGE 022

```

FORTRAN IV      V02.1-11
0050      DO 102 J=1,16
0051      JJ=J
0052      IF(AME(J) .LT. AD(J)) GOTO 102
0054      GOTO 300
0055      CONTINUE
0056      GOTO 999
C
C      AUTOVON FUNCTION
C
0057      DO 201 J=1,64
0058      SAVE2(J)=XINBF(J)
0059      CONTINUE
0060      DO 202 J=1,16
0061      JJ=J
0062      IF(AME(J) .LT. AV(J)) GOTO 202
0064      GOTO 300
0065      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)=ITRM
0079      PACK(6,K1)=ISLID
0080      DO 301 J=1,24
0081      PACK(J+6,K1)=HEAD(J)
0082      PACK(31,K1)=CR
0083      PACK(32,K1)=LF
0084      DO 302 J=1,4
0085      PACK(J+32,K1)=XINBF(J)
0086      DO 303 J=1,4
0087      PACK(J+36,K1)=""040
0088      ENCODE(4,304,PACK(41,K1)) JJ
0089      FORMAT(I4)
0090      DO 305 J=1,4
0091      PACK(J+44,K1)=""040
0092      JJI=JJ#4
0093      DO 306 J=1,4
0094      PACK(J+48,K1)=XINBF(JJI+J)
0095      PACK(53,K1)=CR
0096      PACK(54,K1)=LF
0097      PACK(55,K1)=EIX
0098      CALL ENSTR(PACK(254,K1),55)
0099      CALL ENABLE(0)
0100      CALL ENQUE(PXINQ,XINQ,K1)

```

PAGE 003

```

FORTRAN IV      V02.1-11
0101      C      CALL ENABLE(1)
          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      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      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)=CR
0125      PACK(28,K2)=LF
0126      PACK(30,K2)=LF
0127      PACK(29,K2)=ETX
0128      CALL ENSTR(PACK(254,K2),30)
0129      CALL ENABLE(0)
0130      CALL ENQUE(PXING,XING,K2)
0131      CALL ENABLE(1)
          C
          C      PACKET TO FIAC(RANDOM)
          C
0132      RN=RN*(I1,I2)
0133      MON=RN*1000
0134      IF(MON.GE.334.AND.MON.LE.566) GOTO 401
0135      GOTO 999
0137      ISIM=ISIM+2
0138      IF(ISIM.EC.7) ISIM=1
0140      ICC=ICC+1
0141      IF(ICC.EC.3) ICC=1
0143      VMEAS=RN*(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

```

PAGE 004

```

FORTRAN IV      V02.1-11
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      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      PACK(J+10,K3)=ENC(J)
0169      ENCODE(2,405,PACK(15,K3)) IVC
0170      FORMAT(A2)
0171      GOTO 450
0172      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      PACK(J+10,K3)=ENC(J)
0180      ENCODE(2,405,PACK(15,K3)) IDC
0181      GOTO 450
0182      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      PACK(17,K3)=ICC
0191      ENCODE(4,451,PACK(18,K3)) VMEAS
0192      FORMAT(A4)
0193      ENCODE(2,452,PACK(22,K3)) MON
0194      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(2)
0202      CALL ENQUE(ENQING,XING,K3)
0203      CALL ENABLE(1)
0204      IOFLG=0
0205      RETURN
0206      END

```

PAGE 001

FORTRAN IV W02.1-11

```

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,PAI,Q,FXINQ,FXOUTQ,PACKQ
0006 INTEGER*2 PINQ,K1,OUTFCT,RESNLM,FLWCNT,OUTQ
0007 COMMON/MESS/MESSEQ
0008 COMMON/CUE/XINQ(16),PXINQ(2),XOUTQ(16),PXOUTQ(2),
0009 ACKQ(16),PACKQ(2),INQ(16),PINQ(2)
0010 COMMON/BUFS/XOUTBF(256),XINBF(256),ACKSEQ(256),LIDFD(256),
0011 IOPIC,LLFLG,IRSEND
0012 COMMON/GLOB/ISENT,FLCNT,IQLENTH,OUTFCT,IFULL,OUTQ,
0013 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0014 COMMON/SCOM/SAVE1(64),SAVE2(64),ITERM,ICOUNT,I1,I2,IDATA(256),
0015 ISWTCB
0016 IF(IDATA(1).EQ.117.AND.IDATA(2).EQ.116) GOTO 10
0017 IF(IDATA(1).EQ.117.AND.IDATA(2).EQ.106.AND.
0018 & IDATA(3).EQ.106) GOTO 20
0019 IF(IDATA(1).EQ.115) GOTO 30
0020 IF(IDATA(1).EQ.15) GOTO 999 ! FROM FIAC
0021 ISWTCB=1
0022 ITERM=IDATA(4)
0023 RETURN
0024 ISWTCB=0
0025 RETURN
0026 DECODE(4,31,IDATA(2),ERR=999) ITYPE
0027 FORMAT(I)
0028 IF(ITYPE.EQ.1.OR.ITYPE.EQ.2) GOTO 32
0029 GOTO 999
0030 CALL ENABLE(0)
0031 K1=IGTSP(N)
0032 CALL ENABLE(1)
0033 IF(MESSEQ.EQ.126) MESSEQ=0
0034 MESSEQ=MESSEQ+1
0035 PACK(1,K1)=0
0036 PACK(2,K1)=MESSEQ
0037 PACK(3,K1)=0
0038 PACK(4,K1)=0
0039 PACK(5,K1)=25
0040 PACK(6,K1)=28
0041 IF(ITYPE.EQ.2) GOTO 34
0042 DO 35 J=1,54
0043 PACK(J+6,K1)=SAVE1(J)
0044 CONTINUE
0045 DO 36 J=1,64
0046 PACK(J+6,K1)=SAVE2(J)
0047 CONTINUE
0048 PACK(71,K1)=CR
0049 PACK(72,K1)=LF
0050 PACK(73,K1)=ETX
0051 CALL ENSTE(PACK(254,K1),73)
0052 CALL ENABLE(0)
0053
0054
0055
0056

```

PAGE 002

```
FORTRAN IV      V02.1-11
0057      CALL ENQUE(PXING,XING,K1)
0058      CALL ENABLE(1)
0059      RETURN
0060      END
```

PAGE 001

```
FOFTRAN IV      V02.1-11
0001  SUBROUTINE IMAGE(IN)
0002  INTEGER*4 IN
0003  BYTE ENC
0004  COMMON/END/ENC(4)
0005  ENCODE(4,1,ENC) IN
0006  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  CONTINUE
0012  RETURN
0013  END
```

```

1 .TITLE FDM,MACRO
2 .SECT1 NOLE2B
3 .IDENT /V3.0/
4 .GLOBL LIUJNT,LIO,ENABLE,SWITCH,MASTER,TIME
5 .GLOBL WOKEN,RAM,STATUS,STATE0
6 .GLOBL LFOUT,RFSTART,REPNT,DESTR,LENSTR,LPINPT
7 .NLIST CND
8 .PSECT
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
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

```

R0 =%0
R1 =%1
R2 =%2
R3 =%3
R4 =%4
R5 =%5
R6 =%6
R7 =%7

;INTERFACE ADDRESS
BAR= 172410
WCR= 172412
CSR= 172414
IOBUF= 172416
OPREG= 172416
HRCR= 177560
HRBUF= 177562
HACR= 177564
HXBUF= 177566

;INTERFACE OPTIONS
XCIO=1
XSIO=0
XGIO=1
NDX=0
ND24=1
NHST=0
DHST=1

;LOCAL VARIABLES
ARG1= 2
ARG2= 4
ARG3= 6
ARG4= 10
BIT15= 100000
BIT14= 40000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4

```

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

FDM.MACRO  
NODE2B

66 000002  
67 000001  
68

BIT01= 2  
BIT00= 1

FDM.MACRO MACRO V03.02B 01:21:00 PAGE 2

NODE28

```

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 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****

MASTER: MOV #340, R0
          MTF5 R0
          MOV #124, R0
          MOV #110, (R0)+
          MOV #340, (R0)
          RESET
          MOV #60, R0
          MOV #SIO, (R0)+
          MOV #340, (R0)+
          MOV #100, @#HRC5R
          MOV #100, R0
          MOV #TIME, (R0)
          MOV #40100, @#CSR
          PC, LIUINT
          CLR @#CSR
          MOV #000, R0
          MTF5 R0
          RTS PC

          ;LIU HANDLER
          ;PRI=7
          ;RESET BUS
          ;SIG INTERFACE HANDLER
          ;ENABLE INTERRUPTS
          ;CLOCK VECTOR
          ;DISABLE INTERRUPTS
          ;CLEAR LIU
          ;CLEAR BLIUI CSR
          ;PRI=0
          ;CPU DOWN

          ;LDADR
          ;ADDRESS=0
          ;GOOD WD
          ;NO RETRY
          ;SEL ACRAM
          ;COUNTER
          ;WRITE A NULL
          ;GOOD WD
          ;NO LOOP UNTIL READY
          ;COUNT+1

          ;RDBUFADR CMD
          ;SEL INBUFO
          ;SET POINTER=0
          ;FALSE READ DATA
          ;GOOD READ
          ;NO RETRY
          ;CLEAR DONE BIT
          ;RDBUFADR COMMAND
          ;SEL INBUFI COMMAND

          ;CLEAR ACRAM
          LIUINT: MOV #4354., @#OPREG
                  MOV #2304., @#OPREG
                  TSTB @#CSR
                  BPL --4
                  CLR @#CSR
                  MOV #4353., @#OPREG
                  MOV #256., R0
                  MOV #2311., @#OPREG
                  TSTB @#CSR
                  BPL --4
                  CLR @#CSR
                  INC R0
                  BNE 1$
          ;CLEAR INPUT/OUTPUT BUFFERS
          MOV #12410, R1
          MOV #10440, R0
          JSR PC, ZEROBP
          MOV #1400, @#OPREG
          TSTB @#CSR
          BPL --4
          CLR @#CSR
          MOV #10510, R1
          MOV #10640, R0

```

FDM-MACRO  
NODE26  
MACRO V03.02B 01:21:00 PAGE 2-1

```

102 000234 004767 000052 JSR PC, ZEROBP
103 000240 012737 001400 MOV #1400, G#OPREG
104 000246 105737 172416 TSTB G#CSR
105 000252 100375 172414 BPL -4
106 000254 105037 010510 CLRB G#CSR
107 000260 012701 010510 MOV #10510, R1
108 000264 012700 010540 MOV #10540, R0
109 000270 004767 000015 JSR PC, ZEROBP
110 000274 012701 010710 MOV #10710, R1
111 000300 012700 010740 MOV #10740, R0
112 000304 004767 000002 JSR PC, ZEROBP
113 000310 000440 BR STAT$
114 000312 010137 172416 ZEROBP: MOV R1, G#OPREG
115 000316 012737 001400 MOV #1400, G#OPREG
116 000324 105737 172414 TSTB G#CSR
117 000330 100375 172414 BPL -4
118 000332 105037 172414 CLRB G#CSR
119 000336 013702 172416 MOV #10370, R2
120 000342 042702 177400 BIC #177400, R2
121 000346 010037 172416 MOV R0, G#OPREG
122 000352 022702 000000 CMP #0, R2
123 000356 001412 BEQ 3$, 3$
124 000360 012737 001400 MOV #1400, G#OPREG
125 000366 105737 172414 TSTB G#CSR
126 000372 100375 172414 BPL -4
127 000374 105037 172414 CLRB G#CSR
128 000400 005302 DEC R2
129 000402 000763 BR 2$, 2$
130 000404 010037 172416 MOV R0, G#OPREG
131 000410 000207 RTS PC
132
133
134
135 000412 012737 010400 ;HEAD AND CLEAR STATUS
136 000420 012737 002400 MOV #4352, G#OPREG
137 000426 012737 002400 MOV #1280, G#OPREG
138 000434 000207 MOV #1280, G#OPREG
139
140

```

```

; POINTER=0
; FALSE READ DATA
; GOOD RD
; NO RETRY
; CLEAR DONE BIT
; RDBUFADR COMMAND
; SEL OUTBUF0 COMMAND
; RDBUFADR COMMAND
; SEL OUTBUF0 COMMAND
; ZERO BUFFER POINTER
; GO CLEAR STATUS
; RDBUFADR
; RD
; GOOD RD
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; FETCH POINTER
; CLEAR MST BYTE
; SEL BUFFER
; POINTER=0?
; YES RECHECK
; NO FALSE RD
; GOOD RD
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; LOCAL POINTER-1
; BR NTL 0
; SEL BUFFER
; RETURN

```

```

; WCR:RS
; RS(FALSE)
; RS

```

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

49

50

51

52

53

54

55

56

57

\*\*\*\*\*  
;\*\*\*\* PROCEDURE LIU-HANDLER  
;\*\*\*\*\*

```

L10:      MOV      R0,      -(SP)
          MOV      R1,      -(SP)
          MOV      R2,      -(SP)
          MOV      R3,      -(SP)
          MOV      R4,      -(SP)
          MOV      R5,      -(SP)
          TST      @#CSR
          BPL      RTI$
          BIC      #BIT14, @#CSR
          MOV      #4352., @#OPREG
          MOV      #1280., @#OPREG
          MOV      @#IOPUF, CAUSE
          BIC      #177400, CAUSE

          000436 010046
          000440 010146
          000442 010246
          000444 010346
          000446 010446
          000450 010546
          000452 005737 172414
          000456 100153
          000460 042737 040000 172414
          000466 012737 010400 172416
          000474 012737 002400 172416
          000502 013767 172416 000502
          000510 042767 177400 000474

```

;PERFORM OPERATION DEPENDING ON BITS

```

          BITB    #BIT02, CAUSE
          BEC     B3$
          MOV     #4360., R2
          MOV     #4384., R3
          JSR    PC, EMBF
          BITB   #BIT03, CAUSE
          BEC     B4$
          MOV     #4488., R2
          MOV     #4512., R3
          JSR    PC, EMBF
          BITB   #BIT04, CAUSE
          BEC     B5$
          MOV     #4488., R2
          MOV     #4512., R3
          JSR    PC, EMBF
          BITB   #BIT07, CAUSE
          BEC     B6$
          MOV     #02002',
          MOV     #000370 000370
          MOV     #02002',
          MOV     #000352 000352
          CLR     I#RTM
          CLR     I#RT

          ;BUFFER FULL
          BITB   #BIT00, CAUSE
          BEC     B1$
          MOV     #1#BF, R1
          MOV     #4360., R2
          MOV     #4384., R3
          MOV     #BIT00, R4
          JSR    PC, EMBF
          MOV     #A#EA, R5
          MOV     #1, AREA
          MOV     #00001 000274
          MOV     #0267 000274
          JSR    PC, LPINPT

          ;SAVE R0
          ;SAVE R1
          ;SAVE R2
          ;SAVE R3
          ;SAVE R4
          ;SAVE R5
          ;? LIU INTERRUPTED
          ;NO
          ;YES/DISABLE LIU
          ;RCR : RS(0)
          ;RD
          ;CLEAR UNUSED BITS

          ;OV-FL
          ;RDBUFADR IN0
          ;SEL INBUF0
          ;GO EMPTY
          ;OV-FL
          ;RDBUFADR IN1
          ;SEL INBUFI
          ;GO EMPTY
          ;LINE-LOSS PRIMARY
          ;LATCH ON
          ;YES LOOP AROUND
          ;FLAG PRIMARY SW. FAIL
          ;BACKUP LINE LOSS
          ;FLAG BACKUP SW. FAIL
          ;WHITE TOKEN DETECT
          ;CLEAR TOKEN VAR

          ;INBUF0 FULL
          ;NO
          ;BUFFER ADDRESS
          ;RDBUFADR COMM
          ;SEL BUF
          ;CRC BIT
          ;EMPTY BUFFER
          ;DATA LINK AREA
          ;1 VARIABLE
          ;CRC OR BYTE COUNT
          ;CALL FORTRAN QUE'ER

```

MACRO V03.02B 01:21:00 PAGE 3-1

FDM-MACHO  
NODE28

```

58 000730 132767 000002 000254 E15: BITB #BIT01, CAUSE
59 000736 001423 BEQ RTI$
60 000740 012701 000400 #INER, R1
61 000744 012702 010610 #48B., R2
62 000750 012703 010640 #4512., R3
63 000754 012704 000002 #BIT01, R4
64 000760 004767 000046 PC, EMBF
65 000764 012705 001214 #AREA, R5
66 000770 012767 000001 #1, AREA
67 000776 010267 000216 R2, DATA
68 001002 004767 000000 JSR LPINPT
69
70 001006 012605 RTI5: R5
71 001010 012604 (SP)+, R4
72 001012 012603 (SP)+, R3
73 001014 012602 (SP)+, R2
74 001016 012601 (SP)+, R1
75 001020 012600 (SP)+, R0
76 001022 052737 040000 #BIT14, G#CSR
77 001030 000002 RTI
78
79 001032 012737 172416 EMBF: R5
80 001040 012737 002400 172416 #44B., G#OPREG
81 001046 013700 172416 #12B0., G#OPREG
82 001052 130400 MOV #10BUF, R0
83 001054 001002 BITB R4, R0
84 001056 012704 BNE CROCK R4
85 001062 010237 MOV #1., R2
86 001066 012737 001400 CROCK: MOV #75B., G#OPREG
87 001074 105737 172414 TSTB G#CSR
88 001100 100375 BPL -4
89 001102 013702 MOV #10BUF, R2
90 001106 042702 BIC #177400, R2
91 001112 010200 MOV R0, R0
92 001114 005400 NEG R0
93 001116 010037 MOV R0, G#WCR
94 001122 010137 172410 MOV R1, G#BAR
95 001126 010337 172416 MOV R3, G#OPREG
96 001132 012737 001400 172416 MOV #75B., G#OPREG
97 001140 105737 172414 TSTB G#CSR
98 001144 100375 BPL -4
99 001146 012737 021000 172416 MOV #6704., G#OPREG
100 001154 000240 NOP
101 001156 105737 172414 TSTB G#CSR
102 001162 100401 BMI DMA0K
103 001164 000240 MOV #2304., G#OPREG
104 001166 012737 004400 DMA0K: MOV G#CSR
105 001174 105737 172414 BPL -4
106 001200 100375 TSTB R4
107 001202 005704 TST END$
108 001204 100001 BPL R4
109 001206 010402 MOV R4, R2
110 001210 002207 RTS PC
111 001212 000000 .WORD 0
112 001214 000000 CAUSE: .WORD 0
113 001216 001220 AREA: .WORD DATA
114 001220 000000 DATA: .WORD 0

```

```

;INBUF1 FULL
;NO WERE DONE
;BUFFER ADDRESS
;RDBUFADR COMM
;SEL INEUF1
;CRC BIT
;GO EMPTY BUFFER
;DATA LINK AREA
;ONE VARIABLE
;CRC OR BYTE COUNT
;CALL FORTRAN QUE'ER
;RESTORE REGISTERS
;ENABLE INTERRUPTS
;RETURN FROM INTERRUPT
;READ STATUS 1
;RS
;FETCH STATUS
;GOOD CRC
;NO FLAG
;RDBUFADR
;RD
;GOOD RD
;REPLACE WITH POINTER
;CLEAR MST BITS
;2'S COMP
;BYTE COUNT NOW
;ADDRESS IN MEMORY
;SEL BUFFER
;FALSE RD
;DONE ON
;NO LOOP
;FIRE DMA
;DELAY DMA
;GOOD DMA
;ERROR IF HERE
;FALSE WD
;GOOD WD
;WAS CRC OK
;YES
;NO FLAG IT
;RETURN
;STATUS BYTE 0 HOLDER

```

MACRO V05.02B 01:21:00 PAGE 4

FDM-MACRO  
NOIE28

```

1  .ENABLE LSB
2  *****
3  ;***** PROCEDURE ENABLE *****
4  ;*****
5
6  ENABLE: MOV  @ARG1(R5), R1
7  CMP  #1, R1
8  BEQ  E1$,
9  BIC  #BIT14, @#CSR
10 RTS  PC
11 BIS  #BIT14, @#CSR
12 RTS  PC
13
14 ;FORTRAN ENCODE/DECODE MACRO
15
16 ENSTR: MOV  @ARG2(R5),@ARG1(R5)
17 RTS  PC
18 DESTR: MOV  @ARG1(R5),@ARG2(R5)
19 RTS  PC
20
21 *****
22 ;***** PROCEDURE LINE SWITCH *****
23 ;*****
24
25 ;CALL SWITCH(X) -SET OR RESET LINE SWITCHS
26 ;
27 ;04=SET PRIMARY LINE
28 ;32=RESET PRIMARY LINE
29 ;08=SET BACKUP LINE
30 ;64=RESET BACKUP LINE
31
32 SWITCH: MOV  @ARG1(R5),R1
33 MOV  #4368., @#OPREG
34 ADD  #2304., R1
35 MOV  R1, @#OPREG
36 TSTB @#CSR
37 BPL  .-4
38 CLR  @#CSR
39 RTS  PC
40
41 ;SW SET
42 ;WCR : MODSTAT
;RD
;VALID WHITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT

```

RDW-MACRO MACRO V03.02B 01:21:00 PAGE 5

RDW-MACRO  
NODE2B

```

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 *****

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

STATUS: MOV GARG1(R5), R1
          CMP #1, R1
          BEQ 2$
          MOV #4352,, G#OPREG
          MOV #1280,, G#CPREG
          MOV G#IOBUF, R0
          BIC #177400, R0
          MOV R0, GARG2(R5)
          RTS PC
          MOV #4480,, G#OPREG
          MOV #1280,, G#CPREG
          MOV G#IOBUF, R0
          BIC #177400, R0
          MOV R0, GARG2(R5)
          RTS PC
          MOV #4352,, G#OPREG
          MOV #1280,, G#CPREG
          MOV G#IOBUF, R1
          BIC #177400, R1
          BIT #BIT03, R1
          BEQ 3$
          MOV #1, GARG1(R5)
          RTS PC
          MOV #0, GARG1(R5)
          RTS PC

;WHICH BYTE
;COMPARE
;WCR : RS(0)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;WCR : RS(1)
;RS
;FETCH DATA
;CLEAR BITS
;RTN DATA
;STATUS BYTE 1
;RS
;FETCH DATA
;BUFFER FULL
;LOAD RETURN
;LOAD RETURN

```

```

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 *****

;*****
;**** PROCEDURE WRITE TOKEN (WTKEN)
;*****
;CALL WTKEN -LOADS AND WRITES A TOKEN TO LOOP

WTKEN: MOV #4400., @#OPREG
MOV #1280., @#OPREG
MOV @#10BUF, R1
MOV @#1003, R1
BEG WTKG
MOV #4368., @#OPREG
MOV #2323., @#OPREG
TSTB @#CSR
BPL *-4
CLR @#CSR
RTS PC
MOV #4576., @#OPREG
MOV @#10, @#BAR
MOV #2, R2
MOV #2, @#WCR
MOV #10240., @#OPREG
NOP
TSTB @#CSR
NOP
CLR @#CSR
MOV #4368., @#OPREG
MOV #2322., @#OPREG
TSTB @#CSR
BPL *-4
CLR @#CSR
RTS PC
WTD: .BYTE 0,377

;*****
;**** PROCEDURE TIMER (TIME)
;*****
TIME: INC LTIME
CMP #077777, LTIME
BNE 45
CLR LTIME
RTI 45

;READ STATUS(1)
;RS
;FET H DATA
;BUFFER LOADED
;NO GENERATE TOKEN
;MODSTAT
;MARK OB0F/OB1F/WTCMD
;VALID WHITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT

;SELECT BUFFER 1
;ADDRESS OF WHITE TOKEN
;# OF BYTES
;FOR AT IT
;DMA BYTE COUNT
;FIRE DMA
;DELAY
;GOOD DMA
;ERROR IF HERE
;CLEAR DONE
;MODSTAT
;MARK OB1P/WTCMD
;VALID WHITE
;NO LOOP UNTIL READY
;CLEAR DONE BIT
;RETURN
;A WTKEN

;TIME+1
;?TIME FULL
;CLEAR CLOCK

```

FDM.MACRO MACRO V03.02B 01:21:00 PAGE 7

```

1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
15 16
16 17
17 18
18 19
19 20
20 21
21 22
22 23
23 24
24 25
25 26
26 27
27 28
28 29
29 30
30 31
31 32
32 33
33 34
34 35
35 36
36 37
37 38
38 39
39 40
40 41
41 42
42 43
43 44
44 45
45 46
46 47
47 48
48 49

```

```

*****
***** PROCEDURE ACRAM
*****
;CALL RAM(0,ADDR,DATA) --WRITE RAM ADDRESS WITH DATA
;CALL RAM(1,ADDR,DATA) --READ RAM DATA AT ADDRESS
;
;0110 =NHEAD
;0100 =DHEAD
;0111 =NULL
;1000 =WOKEN

RAM:
MOV   (ARG1(R5), R0
CMP   #0, R0
BNE   RDRAM
MOV   (ARG2(R5), R0
MOV   (ARG3(R5), R1
MOV   #4354., @#OPREG
ADD   #2304., R0
MOV   R0, @#OPREG
TSTB @#CSR
BPL   .-4
CLR   @#CSR
MOV   #4353., @#OPREG
ADD   #2304., R1
MOV   R1, @#OPREG
TSTB @#CSR
BPL   .-4
CLR   @#CSR
RTS   PC
RDRAM:
MOV   (ARG2(R5), R0
MOV   #4354., @#OPREG
ADD   #2304., R0
MOV   R0, @#OPREG
TSTB @#CSR
BPL   .-4
CLR   @#CSR
MOV   #4353., @#OPREG
MOV   #768., @#OPREG
TSTB @#CSR
BPL   .-4
CLR   @#CSR
MOV   #1000., @#OPREG
MOV   #17760., R0
BIC   #17760., R0
MOV   R0, (ARG3(R5)
RTS   PC

```

```

; WHICH OPERATION
; READ OP
; ADDRESS
; WRITE DATA
; SEL LDACTR
; WD/DATA (ADDR)
; WRITE DATA
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; WD/DATA (CMD)
; WRITE
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; ADDRESS TO READ
; SEL LDACTR
; WD/DATA (ADDR)
; WRITE DATA
; INVALID WRITE
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; SEL ACRAM
; READ DATA
; INVALID READ
; NO LOOP UNTIL READY
; CLEAR DONE BIT
; FETCH DATA
; CLEAR BITS
; RIN DATA

```





FDM.MACRO  
NODE28

MACRO V03.02B 01:21:00 PAGE 10

```
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 RSTART: MOV @R0(R5), R0 ;MODE
10 RSD: CMP #0, R0 ;LOAD ADDRESS
11 BNE RS1
12 HALT
13 RTS PC
14 CMP #1, R0 ;RESTART PROGRAM
15 BNE RS2 ;INT LIU FIRST
16 JSR PC, LIUINT ;START ADDRESS
17 MOV #40, R0 ;LOAD MODE
18 JMP (R0)
19 CMP #2, R0
20 BNE RST
21 JMP GROM
22 RTS PC
23 ROM: .WORD 173000
24
25
26
```



MACRO V03.02B 01:21:00 PAGE 12

FDM,MACRO  
NOLE28

```
1 *****
2 ;****
3 ;***** COMMON DATA AREA *****
4 ;*****
5
```

```
6 .PSECT DFM,RW,D,CBL,REL,OVR
```

```
7 OUTBF: .BLKB 256.
8 INBF: .BLKB 256.
```

```
9 IWRITM: .BLKW
10 IWRT: .BLKW
11 STAT: .BLKW
12 LTIME: .BLKW
13 .EVEN
```

```
14 .PSECT BUFS,RW,D,CBL,REL,OVR
```

```
15 XOUTBF: .BLKB 256.
16 XINBF: .BLKB 256.
17 ACKSEC: .BLKB 256.
```

```
18 LIDFD: .BLKB
19 IOFLG: .BLKW
20 LFLG: .BLKW
21 IRSEND: .BLKW
22 .EVEN
```

```
23 .END
```

000001

MACRO V03.02B 01:21:00 PAGE 12-1

FDM.MACRO  
SYMBOL TABLE

ACKSEQ	001000R	003	B1\$	000730R	E0\$	001234R	LPOUT	002110PG	SIO	002476RG
AREA	001214R		B2\$	000516R	E1\$	001244R	LTIME	001000P	002	001004R
ARG1	= 000002		B3\$	000542R	G4\$	002452R	MASTER	000000PG	STATB0	001430RG
ARG2	= 000004		B4\$	000568R	HREUF =	177562	NDXX =	000000	STATUS	001332RG
ARG3	= 000006		B5\$	000614R	HRCR =	177560	ND24 =	000001	STAT\$	000412R
ARG4	= 000010		B6\$	000632R	HXBUF =	177566	NHOST =	000000	SWTCH	001274RG
BAR	= 172410		B7\$	000575R	HXCSR =	177564	OPREC =	172416	TIME	001662RG
BIT00	= 000001		CAUSE	001212R	INBF	000400R	002	OUTBF	000000R	
BIT01	= 000002		CRCOK	001062R	IOBUF =	172416	RAM	001704RG	WTD	001660R
BIT02	= 000004		CSR	= 172414	IOPLG	002000R	003	RDPT	WTKG	001560R
BIT03	= 000010		DATA	001220R	IRSEND	002004R	003	RDRAM	WTOKEN	001502RG
BIT04	= 000020		DESTR	001264RG	IVRT	001002R	002	ROM	WTRAM	001716R
BIT05	= 000040		DHOST =	000001	IWRITM	001000R	002	RST	XCIO =	000001
BIT06	= 000100		DMAOK	001166R	LIDFD	001400R	003	RSTART	XGIO =	000001
BIT07	= 000200		EMBF	001032R	LIO	000436RG	RS0	002422RG	XINBF	000400R
BIT08	= 000400		ENABLE	001222RG	LIUINT	000104RG	RS1	002440P	XOUTBF	000000R
BIT14	= 040000		END\$	001210R	LIPLC	002002R	RS2	002460R	XGIO =	000000
BIT15	= 100000		ENSTR	001254RG	LPINPT=	***** G	003	RTI\$	ZEROBP	000312R
B0\$	000652R									

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

VIRTUAL MEMORY USED: 300 WORDS ( 2 PAGES)  
 DYNAMIC MEMORY AVAILABLE FOR 56 PAGES  
 DK:FDM,DK:FDM=DK:FDM.M2B

FA01 001

12:00:00

01 JUL 79

COMPRESSOR

PROGRAM:

DECOMPER

FOR NUBOL /NOI LRF  
FOR NUBOL /NOI LRF  
FOR SICA /NOI LRF  
MAC FPH



PAGE 001

12:00:00

01-JUL-79

LINKS.COM

PROGRAM:

COMPILER:

ASSTGM DX0: BK:  
 R LINK  
 DX1:NDDB CB>DX1:MAP<-DX1:NDDBG1 Z/E/T/W  
 DX1:FLM ZC  
 DX1:NDDBG1 Z/E  
 DX1:SDG1 Z  
 \$LMRT

ASSTGM DX1: BK:

## 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 inputted 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.

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:01:42 PAGE 001

```

0001 SUBROUTINE USRLNG(LI)
0002 LOGICAL*1 F7,FB,F9,F10,F11,F12,IBUFF,IMP(80)
0003 LOGICAL*1 ICODE,IBUF,MODNUM,I7,I8,DUM
0004 LOGICAL*1 PACK,ICON,ETX,LF,CR,MONTOR,ISLID
0005 INTEGER*2 XING,PXING,PING
0006 INTEGER*2 FLWCNT,OUTFCT,OUTG,RESNLM
0007 INTEGER*2 ST,DEVNUM,FTYPE,RECNUM,FREE
0008 COMMON PACK(256,30)
0009 COMMON /MESS/ MESSEQ
0010 COMMON /FREE/ FREE(30),IFR,IFRSZ
0011 COMMON /GLOB/ ISENT,FLWCNT,ISLID,MONTOR,LF,ETX,CR,DUM
1012 RESNLM,IALTRT,ISLID,MONTOR,LF,ETX,CR,DUM
0013 COMMON /QUE/ XING(15),PXING(2),ING(15),PING(2)
0014 COMMON /USER/ ST(4,9),IND,ICODE(256),NRCNO,MOREC,ICON(256)
0015 COMMON /FILE/ J1,J2,J3,J4,J5,DEVNUM,IFIRST,ICFLG,IABORT,
1016 KEYREC,IBUF(80)
0017 COMMON /M3000/ IFUNT,ITYPE,INODE,IMEAS
0018 COMMON /M4000/ FTYPE,RECNUM,F7,F8,F9,F10,F11,F12
0019 COMMON /M6000/ IDISP,IEQUIP,ICHAN,INUM,IBUFF(20),IFLAG,ICNT
1020 FORMAT(1X,80A1)
C *****
C * START DIALOGUE *
C *****
0021 IF(IFIRST.EQ. 1) GOTO 14 !SET AFTER CALLED ONCE
0022 DO 20 IND=1,4
0023 IND=4
0024 DO 21 I=1,256
0025 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.
20 ICODE(9) .EQ. '117' .AND. ICODE(10) .EQ. '122' .AND.
ICODE(11) .EQ. '124') GOTO 425 !CHECK FOR ABORT
IF(ICODE(7) .NE. '177') GOTO 16
CALL WRTDB
GOTO 12
16 IND=5
MONTOR=ICODE(6)
IF(ICODE(6) .EQ. 4) IND=1
IF(ICODE(6) .EQ. 8) IND=2
IF(ICODE(6) .EQ. 18) IND=3
IF(ICODE(6) .EQ. 25) IND=4
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      IF(ICODE(8) .EQ. 'T') GOTO 605
0051      GOTO 630
0052      GOTO 630
0053      IF(ICODE(9) .EQ. 'T') GOTO 610
0054      GOTO 630
0055      GOTO 630
0056      IF(ICODE(10) .EQ. 'A') GOTO 615
0057      GOTO 630
0058      GOTO 630
0059      IF(ICODE(11) .EQ. 'C') GOTO 620
0060      GOTO 630
0061      IF(ICODE(12) .EQ. 'H') GOTO 625
0062      GOTO 630
0063      GOTO 630
0064      GOTO 630
0065      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      IF (ST(IND,1) .EQ. 0001) GOTO 51
0077      IF (ST(IND,1) .EQ. 0002) GOTO 52
0078      IF (ST(IND,1) .EQ. 0003) GOTO 53
0079      IF (ST(IND,1) .EQ. 0004) GOTO 54
0080      IF (ST(IND,1) .GE. 1000) GOTO 1
0081      GOTO 2
0082      IF(ST(IND,1) .LE. 1999) GOTO 61
0083      IF(ST(IND,1) .GE. 2000) GOTO 3
0084      GOTO 4
0085      IF(ST(IND,1) .LE. 2999) GOTO 62
0086      IF(ST(IND,1) .GE. 3000) GOTO 5
0087      GOTO 6
0088      IF(ST(IND,1) .LE. 3999) GOTO 63
0089      IF(ST(IND,1) .GE. 4000) GOTO 7
0090      GOTO 8
0091      IF(ST(IND,1) .LE. 4999) GOTO 64
0092      IF(ST(IND,1) .GE. 5000) GOTO 9
0093      GOTO 10
0094      IF(ST(IND,1) .LE. 5999) GOTO 65
0095      IF(ST(IND,1) .GE. 6000) GOTO 11
0096      GOTO 12
0097      IF(ST(IND,1) .LE. 6999) GOTO 66
0098      IF(ST(IND,1) .GE. 7000) GOTO 13
0099      GOTO 14
0100      IF(ST(IND,1) .LE. 7999) GOTO 67
0101      IF(ST(IND,1) .GE. 8000) GOTO 15
0102      IF(ST(IND,1) .LE. 8999) GOTO 68
0103      IF(ST(IND,1) .GE. 9000) GOTO 16
0104      GOTO 17
0105      IF(ST(IND,1) .LE. 9999) GOTO 69
0106      IF(ST(IND,1) .GE. 10000) GOTO 18
0107      GOTO 19
0108      IF(ST(IND,1) .LE. 10999) GOTO 70
0109      IF(ST(IND,1) .GE. 11000) GOTO 20
0110      GOTO 21
0111      IF(ST(IND,1) .LE. 11999) GOTO 71
0112      IF(ST(IND,1) .GE. 12000) GOTO 22
0113      GOTO 23
0114      NRCND=22
0115      MOREC=1
0116      ST(IND,1)=0001
0117      GOTO 12
0118      NRCND=1
0119      MOREC=2
0120      IFIRST=1
0121      ST(IND,1)=0002
0122      GOTO 12

```

```

!DISPLAY INTRO MESSAGE
!DISPLAY ENTER USERCODE
!DISPLAY ENTER PASSWORD
!SELECT MODE OF OPERATION

```

```

! IF DS WAS ENTERED

```

```

!DISPLAY INTRO

```

```

C *****

```

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:01:42 PAGE 003

```

C * CHECK USERCODE *
C *****
0123 52 IF ((IC0DE(7) .EQ. '104 .AND. IC0DE(8) .EQ. '101 .AND.
      IC0DE(9) .EQ. '116) .OR. (IC0DE(7) .EQ. '117 .AND.
      IC0DE(8) .EQ. '127 .AND. IC0DE(9) .EQ. '105 .AND.
      IC0DE(10) .EQ. '116)) GOTO 57 !CHECK FOR USERCODE
0125 4 IF (IC0DE(7) .EQ. '114 .AND. IC0DE(8) .EQ. '105 .AND.
      IC0DE(9) .EQ. '116) GOTO 57
0127 5 IF (IC0DE(7) .EQ. '120 .AND. IC0DE(8) .EQ. '105 .AND.
      IC0DE(9) .EQ. '124 .AND. IC0DE(10) .EQ. '105) GOTO 57
0129 6 IF (IC0DE(7) .EQ. '120 .AND. IC0DE(8) .EQ. '117 .AND.
      IC0DE(9) .EQ. '124 .AND. IC0DE(10) .EQ. '124 .AND.
      IC0DE(11) .EQ. '105 .AND. IC0DE(12) .EQ. '122) GOTO 57
0131 7 IF (IC0DE(7) .EQ. '122 .AND. IC0DE(8) .EQ. '117 .AND.
      IC0DE(9) .EQ. '131) GOTO 57
0133 8 GOTO 18
0134 GOTO 18
0135 NRCNO=3 !DISPLAY ENTER PASSWORD
0136 NOREC=1
0137 ST(IND,1)=0003
      GOTO 12
C *****
C * CHECK PASSWORD *
C *****
0138 53 IF ((IC0DE(7) .EQ. '105) .AND.
      (IC0DE(8) .EQ. '123) .AND.
      (IC0DE(9) .EQ. '115)) GOTO 58 !CHECK PASSWORD
0140 3 IF (IC0DE(7) .EQ. '114 .AND. IC0DE(8) .EQ. '105 .AND.
      IC0DE(9) .EQ. '107) GOTO 58
0142 4 IF (IC0DE(7) .EQ. '115 .AND. IC0DE(8) .EQ. '111 .AND.
      IC0DE(9) .EQ. '103 .AND. IC0DE(10) .EQ. '122 .AND.
      IC0DE(11) .EQ. '117) GOTO 58
0144 5 IF (IC0DE(7) .EQ. '102 .AND. IC0DE(8) .EQ. '117 .AND.
      IC0DE(9) .EQ. '123 .AND. IC0DE(10) .EQ. '123) GOTO 58
0146 6 IF (IC0DE(7) .EQ. '123 .AND. IC0DE(8) .EQ. '110 .AND.
      IC0DE(9) .EQ. '117 .AND. IC0DE(10) .EQ. '106 .AND.
      IC0DE(11) .EQ. '101 .AND. IC0DE(12) .EQ. '122) GOTO 58
0148 8 GOTO 18
0149 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=IC0DE(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 NRCNO=12 !IF INCORRECT NUMBER IS
0167 NOREC=1 !ENTERED - ERROR - RETRY
0168 GOTO 12

```

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:01:42 PAGE 004

!BEGIN CRT-TO-CRT MODE

!BEGIN SYSTEM INQUIRY MODE

!BEGIN MODULE UPDATE MODE

!BEGIN FILE ACCESS MODE

!BEGIN REPORT MODE

!BEGIN STATUS MODE

```

0169 MRCNO=13
0170 NOREC=2
0171 ST(IND,1)=1000
0172 GOTO 12
0173 MRCNO=23
0174 NOREC=5
0175 ST(IND,1)=2000
0176 GOTO 12
0177 MRCNO=34
0178 NOREC=3
0179 ST(IND,1)=3000
0180 GOTO 12
0181 MRCNO=49
0182 NOREC=3
0183 ST(IND,1)=4000
0184 GOTO 12
0185 MRCNO=80
0186 NOREC=3
0187 ST(IND,1)=5000
0188 GOTO 12
0189 MRCNO=295
0190 NOREC=5
0191 ST(IND,1)=6000
0192 GOTO 12
0193 CALL M1000(LI)
0194 GOTO 12
0195 CALL M2000
0196 GOTO 12
0197 CALL M3000
0198 GOTO 12
0199 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 CALL M5000
0206 READ(7,1,ERR=12,END=12)(TMP(I),I=1,80)
0207 GOTO 12
0208 CALL M6000
0209 GOTO 12
C *****
C * WRITE PROCEDURE *****
C *****
12 IF (MRCNO.EQ. 0) GOTO 450
200 DO 200 J=7,256
    ICODE(J)=0
214 IF (MRCNO.NE. 12) GOTO 210
216 READ(8,12)(ICODE(I),I=9,87)
217 IF(MESSEQ.EQ. 126) MESSEQ=0
219 MESSEQ=MESSEQ+1
220 ICODE(1)=0
221 ICODE(2)=MESSEQ
222 ICODE(3)=0

```

!ERROR CONDITION

PAGE 005

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:01:42

```

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 CONTINUE
0236 CALL ENABLE(0)
0237 CALL ENQUE(PXING,XING,JSPC)
0238 CALL ENABLE(1)
0239 GOTO 450
0240 210 ICUR=6
0241 DO 310 J=1,NOREC      !START READ LOOP
0242   READ(8,'MRCNO,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 CONTINUE
0248 ICOUNT=IEND+ICUR
0249 IF(ICOUNT.LT.240) GOTO 240
0251 IF(MESSED.EQ.126) MESSED=0
0253 MESSEQ=MESSEQ+1
0254 ICODE(1)=0
0255 ICODE(2)=MESSED
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 CONTINUE
0268 CALL ENABLE(0)
0269 CALL ENQUE(PXING,XING,JSPC)
0270 CALL ENABLE(1)
0271 ICUR=6
0272 DO 250 K=1,IEND
0273   ICODE(ICUR+K)=IBUF(K)
0274   ICUR=ICUR+IEND
0275   ICODE(ICUR+1)=CR
0276   ICODE(ICUR+2)=LF
0277   ICUR=ICUR+2
0278 CONTINUE
0279 ICODE(ICUR+1)=ETX

```

PAGE 006

FORTRAN IV U02.1-1 Tue 05-Jun-79 14:01:42

```
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 CONTINUE
0298 ICFLG=0
0299 CALL ENABLE(0)
0300 CALL ENQUE(PXING,XING,K1)
0301 CALL ENABLE(1)
0302 CALL ENABLE(0)
0303 JSPC=IGETSP(N)
0304 CALL ENABLE(1)
0305 DO 330 J=1,256
0306   PACK(J,JSPC)=ICODE(J)
0307 CONTINUE
0308 DO 340 J=1,256
0309   ICODE(J)=0
0310 CONTINUE
0311 CALL ENABLE(0)
0312 CALL ENQUE(PXING,XING,JSPC)
0313 CALL ENABLE(1)
0314 GOTO 450
0315 IABORT=1
0316 RETURN
0317 500 END
```

```

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

0001      SUBROUTINE WRITDB
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      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,MONITOR,LF,ETX,CR,DUM
0004 INTEGER ST,DEVNUM,OUTQ,OUTFCT,RESNLM
0005 INTEGER FREE,FLWCNT,XING,PXING
0006 INTEGER PING,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,
      8 KEYREC,IBUF(80)
0011 COMMON /FRE/ FREE(30),IFR,IFRSZ
0012 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
      1 RESNLM,IALTRT,ISLID,MONITOR,LF,ETX,CR,DUM
0013 COMMON/QUE/ XING(15),PXING(2),INQ(15),PING(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
0032 NOREC=3
0033 ST(IND,1)=1000
0034 RETURN
0035 5 NRCNO=13
0036 NOREC=2
0037 ST(IND,1)=1000
0038 RETURN
0039 20 ST(IND,2)=4
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
0048 ST(IND,1)=1003
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 FORMAT(I2)
0055 WRITE(8,16)(IBUF(I),I=1,30)
0056 NRCNO=16
      !INVALID ND
      !LIST CRT ND INFO
      !PUT ND INTO ST(IND,2)
      !DISPLAY 'TYPE
      !IN MESSAGE'

```

PAGE 002

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:02:51

```

0057 NOREC=6
0058 ST(IND,1)=1004
0059 CALL ENABLE(0)
0060 I1=IGTSP(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) GOTD 60
0073 PACK(7,I1)=*024
0074 DO 61 I=8,23
0075 PACK(I,I)=LF
0076 DO 62 I=1,LI-6
0077 JJ=1
0078 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 DO 52 I=7,LI
0089 PACK(I,I1)=ICODE(I)
0090 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 ANS=ICODE(7)-48
0100 IF(ANS.EQ.1) GOTD 700
0102 IF(ANS.EQ.2) GOTD 1
0104 IF(ANS.EQ.3) GOTD 6
0106 IF(ANS.EQ.4) GOTD 8
0108 NRCNO=12
0109 NOREC=1
0110 RETURN
0111 NRCNO=15
0112 NOREC=1
0113 ST(IND,1)=1003

```

NEW MESSAGE TO SAME CRT

```
FORTRAN IV      V02.1-1      Tue 05-Jun-79 14:02:51      PAGE 003  
0114      RETURN  
0115      1      NRCNO=13  
0116      NOREC=2  
0117      ST(IND,1)=1000  
0118      RETURN  
0119      6      NRCNO=22  
0120      NOREC=1  
0121      ST(IND,1)=9999  
0122      RETURN  
0123      8      NRCNO=5  
0124      NOREC=7  
0125      ST(IND,1)=0004  
0126      RETURN  
0127      END  
  
NEW MESSAGE TO ANOTHER CRT  
  
LOGOUT  
  
NEW MODE OF OPERATION
```

```

0001 SUBROUTINE M2000
0002 REAL*8 MOUT(10,10),LINE(10)
0003 BYTE ND,I7,IB,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) GO TO 39
0008 IF (ST(IND,1) .EQ. 2003) GO TO 53
0010 IF (ST(IND,1) .EQ. 2004) GO TO 54
0012 IF (ST(IND,1) .EQ. 2005) GO TO 55
0014 IF (ST(IND,1) .EQ. 2006) GO TO 25
*****
C * WHAT TYPE REPORT *
C *****
C *****
0016 I7=ICODE(7)-48
0017 IF (I7 .LT. 1 .OR. I7 .GT. 4) GO TO 71
0019 ST(IND,3)=I7
0020 24 IF (ST(IND,3) .EQ. 1) GO TO 72
0022 25 NRCNO=28
0023 NOREC=2
0024 ST(IND,1)=2003
0025 RETURN
0026 72 NRCNO=127
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 * CHECK NODE DESIGNATOR *
C *****
0034 53 IF ((ICODE(7) .EQ. '116') .AND.
1 (ICODE(8) .EQ. '104') .AND.
2 (ICODE(9) .EQ. '111')) GO TO 80
DECODE (2,40,ICODE(7),ERR=71)ND
0036 40 FORMAT(I)
0037 IF (ND .EQ. 20) GO TO 71
0038 IF (ND .LT. 1) GO TO 71
0040 41 IF (ND .GT. 28) GO TO 71
0042 GO TO (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)ND
0045 201 IREC=1
0046 GO TO 22
0047 202 IREC=8
0048 IF (ST(IND,3) .EQ. 3) IREC=11
0050 GO TO 22
0051 IREC=15
0052 IF (ST(IND,3) .EQ. 3) IREC=21
0054 GO TO 22
0055 204 IREC=22

```

DETERMINE WHAT TYPE INFO

INDI INFORMATION

ENTER ND

```

0054 IF(ST(IND,3) .EQ. 3) IREC=31
0058 GOTD 22
0059 IREC=29
0060 IF(ST(IND,3) .EQ. 3) IREC=137
0062 GOTD (72,51,52,10)ST(IND,3)
0063 51 NRCNO=141+IREC
0064 NOREC=7
0065 ST(IND,1)=2004
0066 RETURN
0067 NRCNO=176+IREC
0068 NOREC=10
0069 ST(IND,1)=2004
0070 RETURN
0071 DO 20 I=1,10
0072 DO 15 J=1,10
0073 MOUT(I,J)=0
0074 CONTINUE
0075 CONTINUE
0076 DO 57 J=1,10
0077 READ(8,'216+J')(MOUT(I,J),I=1,10),INTO MOUT
0078 CONTINUE
0079 DO 30 I=1,10
0080 LINE(I)=0
0081 CONTINUE
0082 READ(8,'226+ND')(LINE(I),I=1,10)
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 CONTINUE
0089 DO 60 J=1,10
0090 WRITE(8,'274+J')(MOUT(I,J),I=1,10)
0091 CONTINUE
0092 NRCNO=275
0093 NOREC=10
0094 ST(IND,1)=2004
0095 RETURN
0096 54 NRCNO=30
0097 NOREC=4
0098 ST(IND,1)=2005
0099 RETURN
C *****
C * WHAT TO DO NEXT *
C *****
0100 55 ANS=ICD(7)-4B
0101 59 IF(ANS .EQ. 1) GOTD 2
0103 IF(ANS .EQ. 2) GOTD 6
0105 IF(ANS .EQ. 3) GOTD 8
0107 GOTD 71
0108 2 NRCNO=23
0109 NOREC=5
0110 ST(IND,1)=2000
0111 RETURN
!INFO FOR LIDS 1-100
!INFO FOR LIDS 101-255
!READ FILE FROM DATA BASE
!MOVE INFO INTO LINE
!ASK WHAT TO DO NEXT
!NEW SYSTEM INQUIRY

```

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:03:22

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

!LOGOUT

\*NEW MODE OF OPERATION

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:03:51 PAGE 001

```

0001 SUBROUTINE M3000
0002 LOGICAL*1 ICODE,ICON,IBUF
0003 INTEGER ST,XING,PXING,PING,DEVNUM
0004 COMMON /MESS/ MESSER
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/ XING(15),PXING(2),INQ(15),PING(2)
0009 IF(ST,IND,1) .EQ. 3000) GOTO 20 !WHAT FUNCTION
0010 IF(ST,IND,1) .EQ. 3005) GOTO 30 !WHAT TYPE
0011 IF(ST,IND,1) .EQ. 3010) GOTO 60 !WHAT NODE
0012 IF(ST,IND,1) .EQ. 3015) GOTO 90 !WHAT MEASUREMENT
0013 IF(ST,IND,1) .EQ. 3020) GOTO 110 !WHAT TO DO NEXT
0014 NRCNO=12
0015 NOREC=1
0016 RETURN
0017
0022 IF(ICODE(7))-48 !DETERMINE WHAT FUNCTION WAS
0023 IF(IFUNT .LT. 1 .OR. IFUNT .GT. 4) GOTO 10
0024 IF(IFUNT .EQ. 1) IF=22
0025 IF(IFUNT .EQ. 2) IF=23
0026 IF(IFUNT .EQ. 3) IF=26
0027 IF(IFUNT .EQ. 4) IF=28
0028 NRCNO=37
0029 NOREC=4
0030 RETURN
0031 ST(IND,1)=3005
0032
0037 ITYPE=ICODE(7)-48 !DETERMINE WHAT TYPE UPDATE
0038 IF(ITYPE .EQ. 1) GOTO 40 !EVENT ON
0039 IF(ITYPE .EQ. 2) GOTO 70 !EVENT OFF
0040 IF(ITYPE .EQ. 3) GOTO 50 !MEASUREMENT
0041 GOTO 10
0042 NRCNO=41
0043 NOREC=2
0044 ST(IND,1)=3010
0045 RETURN
0046 NRCNO=43
0047 NOREC=1
0048 ST(IND,1)=3015
0049 RETURN
0050 DECODE(2,62,ICODE(7),ERR=10) INODE !WHAT NODE TO SEND REPORT TO
0051 FORMAT(I)
0052 IF(INODE .NE. 8 .AND. INODE .NE. 4 .AND. INODE .NE. 18
      * .AND. INODE .NE. 25) GOTO 10
0053 ICON(7)='0' !GENERATE ON PACKET
0054 ICON(8)='N'
0055 ICON(10)=INODE
0056 CALL ENSTR(ICON(254),10)
0057 GOTO 80
0058 ICON(7)='0' !GENERATE OFF PACKET
0059 ICON(8)='F'
0060 ICON(9)='F'
0061 CALL ENSTR(ICON(254),9)
0062 IF(MESSEQ .EQ. 126) MESSEQ=0
0063
0066

```

```

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 GO TO 100
0077 IF(MESSEQ .EQ. 126) MESSEQ=0
0078 MESSEQ=MESSEQ+1
0079
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
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132

90 IF(MESSEQ .EQ. 126) MESSEQ=0
MESSEQ=MESSEQ+1
ICON(1)=0
ICON(2)=MESSEQ
ICON(3)=0
ICON(4)=0
ICON(5)=IF
ICON(6)=24
ICFLG=1
GO TO 100
IF(MESSEQ .EQ. 126) MESSEQ=0
MESSEQ=MESSEQ+1
ICON(1)=0
ICON(2)=MESSEQ
ICON(3)=0
ICON(4)=0
ICON(5)=IF
ICON(6)=24
ICON(7)='M'
DO 92 I=7,10
ICON(I+1)=ICODE(I)
CONTINUE
92
DECODE(4,93,ICODE(7),ERR=10) MEAS
93
FORMAT(14)
IF(IFUNT .EQ. 1) GO TO 94
IF(IFUNT .EQ. 2) GO TO 95
IF(IFUNT .EQ. 3) GO TO 96
94 IF(MEAS .GE. 1 .AND. MEAS .LE. 500) GO TO 98
GO TO 10
95 IF(MEAS .GE. 501 .AND. MEAS .LE. 1000) GO TO 98
GO TO 10
96 IF(MEAS .EQ. 1 .OR. MEAS .EQ. 2 .OR. MEAS .EQ. 3) GO TO 98
GO TO 10
98 CALL ENSTR(ICON(254),11)
ICFLG=1
100 MRCNO=44
MRECF=5
ST(IND,1)=3020
RETURN
110 INEXT=ICODE(7)-48
IF(INEXT .EQ. 1) GO TO 120 !NEW COLUMN FUNCTION
IF(INEXT .EQ. 2) GO TO 130 !SAME COLUMN FUNCTION
IF(INEXT .EQ. 3) GO TO 140 !NEW MODE OF OPERATION
IF(INEXT .EQ. 4) GO TO 150 !LOGOFF
GO TO 10
120 MRCNO=34
MRECF=3
ST(IND,1)=3000
RETURN
130 MRCNO=37
MRECF=4
ST(IND,1)=3005
RETURN
140 MRCNO=5
MRECF=7

```

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:03:51 PAGE 003

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,MTOUT,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 FORMAT (IX,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 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

```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 14:44:32      PAGE 002

0054      F7=IC0DE(7)
0055      F8=IC0DE(8)
0056      F9=IC0DE(9)
0057      F10=IC0DE(10)
0058      F11=IC0DE(11)
0059      F12=IC0DE(12)
0060
28      IF(FTYPE .EQ. 1) GOTO 50      !IF CIRCUIT FILE
C      *****
C      * COMPARE TRUNK KEY
C      *****
31      RECNUM=0
      DO 40 J=1,10
      READ (4,J,ERR=900,END=900)(JBUF(I),I=1,80)
      DO 35 K=1,80,8
      RECNUM=RECNUM+1
      IF(JBUF(K) .NE. IPCNT) GOTO 33      !CHECK TO SEE IF EOF
      IF(JBUF(K+1) .NE. IPCNT) GOTO 33
      IF(JBUF(K+2) .EQ. IPCNT) GOTO 60
      IF(F7 .NE. JBUF(K)) GOTO 35      !START COMPARING FOR
      IF(F8 .NE. JBUF(K+1)) GOTO 35      !TRUNK KEY
      IF(F9 .NE. JBUF(K+2)) GOTO 35
      IF(F10 .NE. JBUF(K+3)) GOTO 35
      IF(F11 .NE. JBUF(K+4)) GOTO 35
      IF(F12 .EQ. JBUF(K+5)) GOTO 42
35      CONTINUE
40      CONTINUE
      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      MRCNO=72
0091      NOREC=7
0092      ST(IND,1)=4006
0093      RETURN
C      *****
C      * COMPARE CIRCUIT KEY
C      *****
50      RECNUM=0
      DO 58 J=1,10
      READ (3,J,ERR=900,END=900)(JBUF(I),I=1,80)
      DO 56 K=1,80,4
      RECNUM=RECNUM+1
      IF(JBUF(K) .NE. IPCNT) GOTO 53      !CHECK FOR EOF
      IF(JBUF(K+1) .NE. IPCNT) GOTO 53
      IF(JBUF(K+2) .EQ. IPCNT) GOTO 60
      IF(F7 .NE. JBUF(K)) GOTO 56      !START COMPARING FOR
      IF(F8 .NE. JBUF(K+1)) GOTO 56      !CIRCUIT FILE
      IF(F9 .NE. JBUF(K+2)) GOTO 56
      IF(F10 .EQ. JBUF(K+3)) GOTO 59
111      CONTINUE
113      CONTINUE
114      GOTO 60
115      GOTO 60
59      READ (5,RECNUM,ERR=900,END=900)(MOUT(I),I=1,80)
117      WRITE(8,72,ERR=900,END=900)(MOUT(I),I=1,80)
118      MRCNO=72
      !ASK WHAT TO DO NEXT
      !IF KEY IF FOUND

```

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:44:32

```

0119 NOREC=7
0120 ST(IND,1)=4006
0121 RETURN
0122 60 MRCNO=79 !KEY WAS NOT FOUND
0123 NOREC=1
0124 ST(IND,1)=4008
0125 RETURN
0126 61 MRCNO=73
0127 NOREC=6
0128 ST(IND,1)=4006
0129 RETURN
0130 65 I7=IC0DE(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=IC0DE(7)-48 !DETERMINE WHAT TO DO NEXT
0140 IF(I7.LT. 1 .OR. I7.GT. 5) GOTO 74
0142 ANS=I7
0143 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 MRCNO=55
0155 NOREC=3
0156 ST(IND,1)=4002
0157 RETURN
0158 105 MRCNO=49
0159 NOREC=3
0160 ST(IND,1)=4000
0161 RETURN
0162 110 MRCNO=5
0163 NOREC=7
0164 ST(IND,1)=0004
0165 RETURN
0166 115 MRCNO=22
0167 NOREC=1
0168 ST(IND,1)=9999
0169 RETURN
0170 120 MRCNO=55
0171 NOREC=3
0172 ST(IND,1)=4009
0173 RETURN
0174 900 MRCNO=109
0175 NOREC=3
0176 ST(IND,1)=4019
0177 RETURN

```

PAGE 004

Tue 05-Jun-79 14:44:32

FORTRAN IV V02.1-1

0178 END

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:44:57 PAGE 001

```

0001 SUBROUTINE M4001
0002 BYTE ICON,ICON,JBUF(80),IRUF(80),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,IRUF(80)
0009 COMMON /M4000/ FTYPE,RECNUM,F7,F8,F9,F10,F11,F12
0010 FORMAT(1X,80A1)
0011 IF(STIND,1).EQ.4010)GOTO 50 !ASKED FOR KEY TO BE MODIFIED
0013 IF(STIND,1).EQ.4012)GOTO 52 !CHECK FOR SAME RECORD
0015 IF(STIND,1).EQ.4014)GOTO 120 !UPDATE OR DELETE RECORD
0017 IF(STIND,1).EQ.4016)GOTO 145 !UPDATE FILE
0019 IF(STIND,1).EQ.4018)GOTO 180 !ADD NEW RECORD
0021 IF(STIND,1).EQ.4019)GOTO 195 !RESTART
0023 IF(STIND,1).EQ.4020)GOTO 230 !TYPE IN NEW CIRCUIT RECORD
0025 IF(STIND,1).EQ.4030)GOTO 270 !TYPE IN NEW TRUNK RECORD
      C *****
      C * FILE WILL BE MODIFIED *****
      C *****
0027 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 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 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 CONTINUE
0055 GOTO 115
0056 GOTO 115
0057 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
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)
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
0096 NOREC=4
0097 ST(IND,1)=4018
0098 RETURN
C *****
C * UPDATE OR DELETE *
C *****
0099 120 I7=ICD(7)-48
0100 IF(I7.LT. 1 .OR. I7.GT. 2) GOTO 290
0102 ANS=I7
0103 IF(ANS.EQ. 1) GOTO 140
0105 IF(ANS.EQ. 2) GOTO 160
0107 140 NRCNO=65
0108 NOREC=2
0109 ST(IND,1)=4016
0110 RETURN
C *****
C * WRITE NEW RECORD *
C *****
0111 145 IF(FTYPE.EQ. 2) GOTO 147
0113 WRITE(5,'RECNUM,ERR=900,END=900')(ICD(1),I=7,80)
0114 GOTO 150
0115 147 WRITE(6,'RECNUM,ERR=900,END=900')(ICD(1),I=7,80)
0116 150 WRITE(8,'72,ERR=900,END=900')(ICD(1),I=7,80)
0117 NRCNO=71
0118 NOREC=8
0119 ST(IND,1)=4006
0120 RETURN
0121 160 IF(FTYPE.EQ. 2) GOTO 170

```

PAGE 003

Tue 05-Jun-79 14:44:57

FORTRAN IV V02.1-1

```

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 CONTINUE
0131 GOTO 174
0132 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 CONTINUE
0142 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 I7=ICODE(7)-48
0148 IF(I7 .LT.1 .OR. I7 .GT. 2) GOTO 290
0149 ANS=I7
0150 IF(ANS .EQ. 1) GOTO 200
0151 NRCNO=73
0152 NOREC=6
0153 ST(IND,1)=4006
0154 RETURN
0155 IF(FTYPE .EQ. 2) GOTO 240
0156 *****
0157 * ADD CIRCUIT FILE *
0158 *****
0159 RECNUM=0
0160 DO 215 J=1,10
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
0165 IF(JBUF(K+1) .NE. 0) GOTO 205
0166 IF(JBUF(K+2) .NE. 0) GOTO 205
0167 IF(JBUF(K+3) .EQ. 0) GOTO 220
0168 IF(JBUF(K) .NE. IPCNT) GOTO 210
0169 IF(JBUF(K+1) .NE. IPCNT) GOTO 210
0170 IF(JBUF(K+2) .EQ. IPCNT) GOTO 225
0171 CONTINUE
0172 210 CONTINUE
0173 215 CONTINUE
0174 JBUF(K)=F7

```

```

0181 JBUF(K+1)=F8
0182 JBUF(K+2)=F9
0183 JBUF(K+3)=F10
0184 WRITE(3,'J,ERR=900,END=900')(JBUF(I),I=1,80)
0185 NRCNO=64
0186 NOREC=1
0187 ST(IND,1)=4020
0188 RETURN
0189 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 WRITE(5'RECNUM,ERR=900,END=900)(ICDCE(I),I=7,80)
0198 GOTO 280
C *****
C * ADD TRUNK FILE *
C *****
240 RECNUM=0
DO 255 J=1,10
  READ(4,'J,ERR=900,END=900')(JBUF(I),I=1,80)
  DO 250 K=1,80,8
    RECNUM=RECNUM+1
    IF(JBUF(K).NE.0) GOTO 245
    IF(JBUF(K+1).NE.0) GOTO 245
    IF(JBUF(K+2).NE.0) GOTO 245
    IF(JBUF(K+3).NE.0) GOTO 245
    IF(JBUF(K+4).NE.0) GOTO 245
    IF(JBUF(K+5).EQ.0) GOTO 260
    IF(JBUF(K).NE.0) GOTO 250
    IF(JBUF(K+1).NE.0) GOTO 250
    IF(JBUF(K+2).EQ.0) GOTO 265
  CONTINUE
250 CONTINUE
255 JBUF(K)=F7
260 JBUF(K+1)=F8
265 JBUF(K+2)=F9
270 JBUF(K+3)=F10
275 JBUF(K+4)=F11
280 JBUF(K+5)=F12
WRITE(4,'J,ERR=900,END=900')(JBUF(I),I=1,80)
NRCNO=64
NOREC=1
ST(IND,1)=4030
RETURN
285 JBUF(K)=F7
290 JBUF(K+1)=F8
295 JBUF(K+2)=F9
300 JBUF(K+3)=F10
305 JBUF(K+4)=F11
310 JBUF(K+5)=F12

```

```
FORTRAN IV      V03.1-1      Tue 05-Jun-79 14:44:57      PAGE 005

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')(ICDDE(I),I=7,80)
0246      280 WRITE(8,'72,ERR=900,END=900')(ICDDE(I),I=7,80)
0247      NRCND=71
0248      NOREC=8
0249      ST(IND,1)=4006
0250      RETURN
0251      290 NRCND=12
0252      NOREC=1
0253      RETURN
0254      900 NRCND=109
0255      NOREC=3
0256      ST(IND,1)=4019
0257      RETURN
0258      END
```

```

0001 SUBROUTINE M5000
0002 BYTE ICODE,I7,I8,RECRUF,IRUF,STORE,LRUF(80),ICON
0003 INTEGER ST,ANS,PARA,DEVNUM,KEYREC
0004 REAL*8 ERECRF,MOUT(10,9)
0005 DIMENSION STORE(80)
0006 DIMENSION ERECRF(10),RECRUF(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,IRUF(80)
0009 EQUIVALENCE (ERECRF,RECRUF)
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 NRCNO=12 !INVALID OPTION
0064 NOREC=1
0065 RETURN
0066 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 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 I7=ICODE(7)-48 !DETERMINE WHAT TYPE REPORT
0077 IF(I7.LT.1.OR.I7.GT.2) GOTO 20
0079 IUNIT=I7
0080 PARA=0
0081 JT=2
0082 DO 62 I=1,80

```

PAGE 002

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:45:40

```

0083 LBUF(I)=0
0084 DO 64 I=1,4
0085 WRITE(7,ERR=90,END=900)(LBUF(J),J=1,80)
0086 CONTINUE
0087 MRCNO=86
0088 MOREC=4
0089 ST(IND,1)=5004
0090 RETURN
0091 MRCNO=90
0092 MOREC=1
0093 ST(IND,1)=5032
0094 RETURN
0095 DO 96 J=1,3
0096 RECBUF(J)=ICODE(J+6)
0097 CONTINUE
0098 MRCNO=91
0099 MOREC=1
0100 ST(IND,1)=5034
0101 RETURN
0102 RECBUF(9)=ICODE(7)
0103 RECBUF(10)=ICODE(8)
0104 MRCNO=92
0105 MOREC=1
0106 ST(IND,1)=5036
0107 RETURN
0108 DO 106 J=7,10
0109 RECBUF(J+10)=ICODE(J)
0110 CONTINUE
0111 IF(IUNIT.EQ.2) GOTO 107
0112 MOREC=1
0113 MRCNO=93
0114 GOTO 108
0115 MOREC=1
0116 MRCNO=123
0117 MOREC=1
0118 ST(IND,1)=5038
0119 RETURN
0120 DO 111 J=7,11
0121 RECBUF(J+18)=ICODE(J)
0122 CONTINUE
0123 IF(IUNIT.EQ.2) GOTO 118
0124 MRCNO=94
0125 MOREC=1
0126 MRCNO=95
0127 ST(IND,1)=5040
0128 RETURN
0129 DO 116 J=7,9
0130 RECBUF(J+26)=ICODE(J)
0131 CONTINUE
0132 MRCNO=95
0133 MOREC=1
0134 ST(IND,1)=5042
0135 RETURN
0136 DO 121 J=7,10
0137 RECBUF(J+34)=ICODE(J)
0138 CONTINUE

```

!CREATE A CHANNEL REPORT

!MOVE STATION ID TO ARRAY

!MOVE SEQUENCE NUMBER TO ARRAY

!MOVE DATE-TIME TO ARRAY

!MOVE LINK TO ARRAY

!MOVE CHANNEL NUMBER TO ARRAY

!MOVE OUT TIME TO ARRAY

```

0139 NRCNO=96
0140 NOREC=1
0141 ST(IND,1)=5044
0142 RETURN
0143 DO 126 J=7,10
0144 RECBUF(J+2)=ICODE(J)
0145 CONTINUE
0146 NRCNO=97
0147 NOREC=1
0148 ST(IND,1)=5046
0149 RETURN
0150 DO 131 J=7,9
0151 RECBUF(J+50)=ICODE(J)
0152 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 IF(ICODE(7) .NE. '77') GOTO 134
0159 IF(ICODE(8) .NE. '77') GOTO 134
0160 IF(ICODE(9) .EQ. '77') GOTO 136
0161 IF(ICODE(9) .EQ. '77') GOTO 136
0162 IF(ICODE(7) .NE. 'X') GOTO 135
0163 IF(ICODE(8) .NE. 'X') GOTO 135
0164 IF(ICODE(9) .NE. 'X') GOTO 135
0165 IF(ICODE(10) .EQ. 'X') GOTO 136
0166 IF(ICODE(10) .EQ. 'X') GOTO 136
0167 IF(ICODE(10) .EQ. 'X') GOTO 136
0168 IF(ICODE(10) .EQ. 'X') GOTO 136
0169 IF(ICODE(10) .EQ. 'X') GOTO 136
0170 IF(ICODE(10) .EQ. 'X') GOTO 136
0171 IF(ICODE(10) .EQ. 'X') GOTO 136
0172 IF(ICODE(10) .EQ. 'X') GOTO 136
0173 IF(ICODE(10) .EQ. 'X') GOTO 136
0174 IF(ICODE(10) .EQ. 'X') GOTO 136
0175 IF(ICODE(10) .EQ. 'X') GOTO 136
0176 IF(ICODE(10) .EQ. 'X') GOTO 136
0177 IF(ICODE(10) .EQ. 'X') GOTO 136
0178 IF(ICODE(10) .EQ. 'X') GOTO 136
0179 IF(ICODE(10) .EQ. 'X') GOTO 136
0180 IF(ICODE(10) .EQ. 'X') GOTO 136
0181 IF(ICODE(10) .EQ. 'X') GOTO 136
0182 IF(ICODE(10) .EQ. 'X') GOTO 136
0183 IF(ICODE(10) .EQ. 'X') GOTO 136
0184 IF(ICODE(10) .EQ. 'X') GOTO 136
0185 IF(ICODE(10) .EQ. 'X') GOTO 136
0186 IF(ICODE(10) .EQ. 'X') GOTO 136
0187 IF(ICODE(10) .EQ. 'X') GOTO 136
0188 IF(ICODE(10) .EQ. 'X') GOTO 136
0189 IF(ICODE(10) .EQ. 'X') GOTO 136
0190 IF(ICODE(10) .EQ. 'X') GOTO 136
0191 IF(ICODE(10) .EQ. 'X') GOTO 136
0192 IF(ICODE(10) .EQ. 'X') GOTO 136
0193 IF(ICODE(10) .EQ. 'X') GOTO 136
0194 IF(ICODE(10) .EQ. 'X') GOTO 136
0195 IF(ICODE(10) .EQ. 'X') GOTO 136
0196 IF(ICODE(10) .EQ. 'X') GOTO 136
0197 IF(ICODE(10) .EQ. 'X') GOTO 136
0198 IF(ICODE(10) .EQ. 'X') GOTO 136
0199 IF(ICODE(10) .EQ. 'X') GOTO 136
0200 IF(ICODE(10) .EQ. 'X') GOTO 136
0201 IF(ICODE(10) .EQ. 'X') GOTO 136

```

! MOVE IN TIME TO ARRAY

! MOVE RFO TO ARRAY

! CHECK FOR ???

! CHECK FOR XXXX AT END  
! OF REMARKS SECTION

! ASK IF REPORT IS COMPLETE

! IS REPORT COMPLETE?  
! IANS .LT. 1 .OR. IANS .GT. 2) GOTO 20

! ASK FOR DESTINATION ID

! I7 .EQ. 4 .OR. 17 .EQ. 8 .OR. 17 .EQ. 17 .EQ. 18 .OR.  
! 17 .EQ. 25) GOTO 143

! ANS=17  
! CALL SNDMES(ANS,IUNIT)

! NRCNO=118  
! NOREC=5  
! ST(IND,1)=5008  
! RETURN

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 14:45:40      PAGE 004

0202 150 I7=IC00DE(7)-48
0203 IF(I7 .LT. 1 .OR. I7 .GT. 4) GOTO 20
0205 JT=2
0206 ANS=I7
0207 IF(ANS .EQ. 1) GOTO 165
0209 IF(ANS .EQ. 2) GOTO 170
0211 IF(ANS .EQ. 3) GOTO 175
0213 NRCND=80
0214 NREC=3
0215 ST(IND,1)=5000
0216 RETURN
0217 170 NRCND=5
0218 NREC=7
0219 ST(IND,1)=0004
0220 RETURN
0221 175 NRCND=22
0222 NREC=1
0223 ST(IND,1)=9999
0224 RETURN
0225 300 NRCND=124
0226 NREC=3
0227 ST(IND,1)=5012
0228 RETURN
0229 310 I7=IC00DE(7)-48
0230 IF(I7 .LT. 1 .OR. I7 .GT. 2) GOTO 20
0232 ANS=I7
0233 READ(7,1,ERR=900,END=900)(RECBUF(I),I=1,80)
0234 IF(RECBUF(1) .NE. '77) GOTO 325
0236 IF(RECBUF(2) .NE. '77) GOTO 325
0238 IF(RECBUF(3) .EQ. '77) GOTO 360
0240 IF(RECBUF(9) .NE. '77) GOTO 330
0242 IF(RECBUF(10) .EQ. '77) GOTO 370
0244 IF(RECBUF(17) .NE. '77) GOTO 335
0246 IF(RECBUF(18) .NE. '77) GOTO 335
0248 IF(RECBUF(19) .EQ. '77) GOTO 380
0250 IF(ANS .EQ. 2) GOTO 500
0252 IF(RECBUF(25) .NE. '77) GOTO 340
0254 IF(RECBUF(26) .NE. '77) GOTO 340
0256 IF(RECBUF(27) .EQ. '77) GOTO 390
0258 IF(RECBUF(33) .NE. '77) GOTO 345
0260 IF(RECBUF(34) .NE. '77) GOTO 345
0262 IF(RECBUF(35) .EQ. '77) GOTO 400
0264 IF(RECBUF(41) .NE. '77) GOTO 350
0266 IF(RECBUF(42) .NE. '77) GOTO 350
0268 IF(RECBUF(43) .EQ. '77) GOTO 410
0270 IF(RECBUF(49) .NE. '77) GOTO 355
0272 IF(RECBUF(50) .EQ. '77) GOTO 420
0274 IF(RECBUF(57) .NE. '77) GOTO 356
0276 IF(RECBUF(58) .NE. '77) GOTO 356
0278 IF(RECBUF(59) .EQ. '77) GOTO 430
0280 WRITE(7,1,ERR=900,END=900)(RECBUF(I),I=1,80)
0281 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

```

```

! DETERMINE WHAT TO DO NEXT

```

```

! NEW REPORT

```

```

! NEW MODE OF OPERATION

```

```

! LOGOUT

```

```

! ASK WHAT TYPE REPORT

```

```

! DETERMINE WHAT TYPE REPORT

```

```

! CHECK STATION ID

```

```

! CHECK SEQUENCE NUMBER

```

```

! CHECK DATE-TIME

```

```

! CHECK LINK ID

```

```

! CHECK CHANNEL NUMBER

```

```

! CHECK OUTAGE TIME

```

```

! CHECK IN TIME

```

```

! CHECK RFD CODE

```

```

! CHECK RFD CODE

```

```

! CHECK RFD CODE

```

```

! CHECK RFD CODE

```

```

FORTRAN IV  V02.1-1  Tue 05-Jun-79 14:45:40  PAGE 005

0286 IF(RECBUF(3) .EQ. '77) GOTO 450
0288 GOTO 136
0289 MRCNO=90
0290 NOREC=1
0291 ST(IND,1)=5014
0292 RETURN
0293 DO 366 J=1,3
0294 RECBUF(J)=ICODE(J+6)
0295 CONTINUE
0296 GOTO 325
0297 MRCNO=91
0298 NOREC=1
0299 ST(IND,1)=5016
0300 RETURN
0301 RECBUF(9)=ICODE(7)
0302 RECBUF(10)=ICODE(8)
0303 GOTO 330
0304 MRCNO=92
0305 NOREC=1
0306 ST(IND,1)=5018
0307 RETURN
0308 DO 386 J=7,11
0309 RECBUF(J+10)=ICODE(J)
0310 CONTINUE
0311 GOTO 335
0312 MRCNO=93
0313 NOREC=1
0314 ST(IND,1)=5020
0315 RETURN
0316 DO 396 J=7,11
0317 RECBUF(J+18)=ICODE(J)
0318 CONTINUE
0319 GOTO 340
0320 MRCNO=94
0321 NOREC=1
0322 ST(IND,1)=5022
0323 RETURN
0324 DO 406 J=7,9
0325 RECBUF(J+26)=ICODE(J)
0326 CONTINUE
0327 GOTO 345
0328 MRCNO=95
0329 NOREC=1
0330 ST(IND,1)=5024
0331 RETURN
0332 DO 416 J=7,10
0333 RECBUF(J+34)=ICODE(J)
0334 CONTINUE
0335 GOTO 350
0336 MRCNO=96
0337 NOREC=1
0338 ST(IND,1)=5026
0339 RETURN
0340 DO 426 J=7,10

! MOVE STATION ID TO ARRAY
! MOVE SEQUENCE NUMBER
! MOVE SEQ NUM TO ARRAY
! MOVE DATE-TIME
! MOVE DATE-TIME TO ARRAY
! MOVE LINK ID
! MOVE LINK ID TO ARRAY
! MOVE CHANNEL NUMBER
! MOVE CHAN NUM TO ARRAY
! MOVE OUTAGE TIME
! MOVE OUTAGE TIME TO ARRAY
! MOVE IN TIME

```

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 14:45:40      PAGE 006

0341      RECBUF(J+42)=ICODE(J)
0342      CONTINUE
0343      GOTO 355
0344      NRCNO=97
0345      NOREC=1
0346      ST(IND,1)=5028
0347      RETURN
0348      DO 436 J=7,10
0349          RECBUF(J+50)=ICODE(J)
0350      CONTINUE
0351      438      WRITE(7,1,ERR=900,END=900)(RECBUF(I),I=1,80)
0352      GOTO 357
0353      NRCNO=98
0354      NOREC=3
0355      ST(IND,1)=5005
0356      JT=2
0357      RETURN
0358      500      IF(RECBUF(25).NE.'77') GOTO 3451CHECK SWITCH ID
0360      IF(RECBUF(26).NE.'77') GOTO 345
0362      IF(RECBUF(27).EQ.'77') GOTO 505
0364      GOTO 345
0365      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      CONTINUE
0372      GOTO 410
0373      900      NRCNO=109
0374      NOREC=3
0375      RETURN
0376      END

```

! ENTER RFO CODE

! MOVE RFO CODE TO ARRAY

PAGE 001

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:46:20

```

0001 SUBROUTINE SMDMES(LT,LU)
0002 INTEGER*2 XINQ,PXINQ,PING
0003 LOGICAL*1 PACK,LBUF(80),RECBUF(80)
0004 COMMON PACK(256*30)
0005 COMMON /MESS/ MESSEQ
0006 COMMON/QUE/XINQ(15),PXINQ(2),ING(15),PING(2)
0007 DO 5 I=1,80
0008   LBUF(I)=0
0009   RECBUF(I)=0
0010 5 CONTINUE
0011   LCUR=1
0012   READ(7,ERR=900,END=900)(RECBUF(I),I=1,80)
0013   IF(LU .EQ. 2) GO TO 100
0014   DO 70 JJI=101,108
0015     READ(8'JJJ,ERR=900,END=900)(LBUF(I),I=1,80)
0016     DO 20 I=80,1,-1
0017       ID=I
0018       IF(LBUF(I) .NE. 0) GO TO 30
0019     CONTINUE
0020     LBUF(ID+1)=*040
0021     DO 40 J=LCUR,LCUR+7
0022       LBUF(ID+J+2-LCUR)=RECBUF(J)
0023     CONTINUE
0024     LBUF(ID+10)=*015
0025     LBUF(ID+11)=*012
0026     LBUF(ID+12)=*003
0027     CALL ENABLE(0)
0028     K1=IGETSP(N)
0029     CALL ENABLE(1)
0030     DO 45 K=1,256
0031       PACK(K,K1)=0
0032     CONTINUE
0033     DO 50 JP=1,ID+12
0034       PACK(JP+6,K1)=LBUF(JP)
0035     CONTINUE
0036     IF(MESSEQ .EQ. 126) MESSEQ=0
0037     MESSEQ=MESSEQ+1
0038     PACK(1,K1)=0
0039     PACK(2,K1)=MESSEQ
0040     PACK(3,K1)=0
0041     PACK(4,K1)=LT
0042     PACK(5,K1)=24
0043     CALL ENSTR(PACK(254,K1),ID+18)
0044     CALL ENABLE(0)
0045     CALL ENQUE(PXINQ,XINQ,K1)
0046     CALL ENABLE(1)
0047     LCUR=LCUR+8
0048     DO 60 I=1,80
0049       LBUF(I)=0
0050     CONTINUE
0051     GO TO 200
0052 60 CONTINUE
0053 70 CONTINUE
0054 80 CONTINUE
0055 100 DO 190 JJI=285,291

```

PAGE 002

```

FORTRAN IV      V02.1-1      Tue 05-Jun-79 14:46:20
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      CONTINUE
0064      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)=RECRUF(J)
0069      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      CONTINUE
0079      DO 140 JP=1,ID+12
0080          PACK(JP+6,K1)=LBUF(JP)
0081      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(PXIND,XIND,K1)
0094      CALL ENABLE(1)
0095      LCUR=LCUR+8
0096      DO 150 I=1,80
0097          LBUF(I)=0
0098      CONTINUE
0099      CONTINUE
0100      DO 250 IP=2,4
0101          DO 205 I=1,80
0102              LBUF(I)=0
0103      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      CONTINUE
0113      LBUF(IL)='015
0114      LBUF(IL+1)='012
0115      LBUF(IL+2)='003
0116      IF(IL .EQ. 1) GOTO 250

```

```
FORTRAN IV      V02,1-1      Tue 05-Jun-79 14:46:20      PAGE 003

0118      CALL ENABLE(0)
0119      K1=IGETSP(N)
0120      CALL ENABLE(1)
0121      DO 225 K=1,256
0122      PACK(K,K1)=0
0123      CONTINUE
0124      DO 230 IR=1,IL+2
0125      PACK(IR+6,K1)=LBUF(IR)
0126      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(PXIND,XIND,K1)
0139      CALL ENABLE(1)
0140      CONTINUE
0141      RETURN
0142      900      NRCND=109
0143      900      NOREC=3
0144      RETURN
0145      END
```

```

0001 SUBROUTINE M6000
0002 BYTE ICODE,IBUFF,ICON,ISTAT,PACK
0003 BYTE ISLID,MONITOR,LF,ETX,CR,DUM
0004 INTEGER ST,FLWCNT,OUTFCT,OUTQ,RESNLM
0005 INTEGER PXING,XING,PING
0006 COMMON PACK(256,30)
0007 COMMON /MESS/ MESSEQ
0008 COMMON /GLOB/ ISENT,FLWCNT,IQLNTH,OUTFCT,IFULL,OUTQ,
0009 RESNLM,IALTRT,ISLID,MONITOR,LF,ETX,CR,DUM
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
0013 ICNT=0
0014 IF(ST(IND,1) .EQ. 6000) GOTO 20 !WHAT COMMAND WAS ENTERED
0015 IF(ST(IND,1) .EQ. 6010) GOTO 300 !CHOOSE NEXT COMMAND
0016 IF(ST(IND,1) .EQ. 6020) GOTO 140 !WHAT EQUIPMENT
0017 IF(ST(IND,1) .EQ. 6030) GOTO 160 !PICK DEVICE NUMBER
0018 IF(ST(IND,1) .EQ. 6040) GOTO 200 !CHANGE STATUS CODE
0019 IF(ST(IND,1) .EQ. 6050) GOTO 310 !PICK A NEW COMMAND
0020 IF(ST(IND,1) .EQ. 6050) GOTO 310 !PICK A NEW COMMAND
0021 !ERROR
0022 NRCNO=12
0023 NOREC=1
0024 RETURN
0025
0026 IDISP=ICODE(7)--48 !WHAT TYPE INFO
0027 IF(IDISP .EQ. 1) GOTO 30 !RED CONDITIONS
0028 IF(IDISP .EQ. 2) GOTO 30 !AMBER CONDITIONS
0029 IF(IDISP .EQ. 3) GOTO 130 !SELECT STATUS
0030 IF(IDISP .EQ. 4) GOTO 130 !CHANGE STATUS
0031 GOTO 10
0032 ISTAT='1'
0033 IF(IDISP .EQ. 2) ISTAT='2' !AMBER STATUS
0034 ICNT=ICNT+1
0035 IF(ICNT .GT. 1016) GOTO 60
0036 READ(2,ICNT)(IBUFF(I),I=1,20)
0037 IF(IBUFF(15) .NE. ISTAT) GOTO 35
0038 CALL ENABLE(0)
0039 KI=IGETSP(N)
0040 CALL ENABLE(1)
0041 IF(MESSEQ .EQ. 126) MESSEQ=0
0042 MESSEQ=MESSEQ+1
0043 PACK(1,K1)=0
0044 PACK(2,K1)=MESSEQ
0045 PACK(3,K1)=0
0046 PACK(4,K1)=0
0047 PACK(5,K1)=MONITOR
0048 PACK(6,K1)=24
0049 PACK(7,K1)='S'
0050 PACK(8,K1)='T'
0051 PACK(9,K1)='A'
0052 PACK(10,K1)='T'
0053 PACK(11,K1)='U'
0054 PACK(12,K1)='S'
0055 PACK(13,K1)='040
0056 PACK(14,K1)='040

```

PAGE 002

FORTRAN IV 002.1-1 Tue 05-Jun-79 14:46:53

```

0069 PACK(I5,K1)='040
0070 PACK(I6,K1)='040
0071 PACK(I7,K1)='040
0072 DO 40 I=1,20
0073   PACK(I+17,K1)=IBUFF(I)
0074   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(PXING,XING,K1)
0081   CALL ENABLE(1)
0082   IFLAG=1
0083   NRCND=0
0084   GOTO 999
0085   60 IFLAG=0
0086   ST(IND,1)=6010
0087   NRCND=0
0088   GOTO 999
0089   130 NRCND=260
0090   NRECC=4
0091   ST(IND,1)=6020
0092   RETURN
0093   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   NRCND=264              !ENTER DEVICE NUMBER
0099   NRECC=1
0100   ST(IND,1)=6030
0101   RETURN
0102   150 NRCND=265
0103   NRECC=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
0109   IF(IEQUIP.EQ.3) GOTO 175
0110   IF(INUM.LT.1.OR. INUM.GT.3) GOTO 10
0111   IF(IEQUIP.EQ.2) INUM=INUM+1000
0112   IF(IEQUIP.EQ.4) INUM=INUM+1005
0113   IF(IEQUIP.EQ.5) INUM=INUM+1008
0114   IF(IEQUIP.EQ.6) INUM=INUM+1011
0115   GOTO 180
0116   170 IF(INUM.LT.1.OR. INUM.GT.1000) GOTO 10
0117   GOTO 180
0118   175 IF(INUM.EQ.1.OR. INUM.EQ.2) GOTO 177
0119   GOTO 10
0120   177 INUM=INUM+1003
0121   180 IF(IDISP.EQ.4) GOTO 200
0122   READ(2,INUM,ERR=900,END=900)(IBUFF(J),J=1,20)
0123   WRITE(8,273,ERR=900,END=900)(IBUFF(J),J=1,20)
0124   NRCND=273

```

PAGE 003

FORTRAN IV V02.1-1 Tue 05-Jun-79 14:46:53

```

0135 NOREC=2
0136 ST(IND,1)=6010
0137 GOTO 999
0138 INO=ICD(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 MRCNO=267
0145 NOREC=6
0146 ST(IND,1)=6050
0147 GOTO 999
0148 MRCNO=268
0149 NOREC=5
0150 ST(IND,1)=6050
0151 RETURN
0152 IANS=ICD(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 ICNT=0
0163 IF(IDISP.EQ.1)GOTO 30
0165 IF(IDISP.EQ.2)GOTO 30
0167 MRCNO=260
0168 NOREC=4
0169 ST(IND,1)=6020
0170 RETURN
0171 MRCNO=5
0172 NOREC=7
0173 ST(IND,1)=0004
0174 RETURN
0175 MRCNO=255
0176 NOREC=5
0177 ST(IND,1)=6000
0178 RETURN
0179 MRCNO=22
0180 NOREC=1
0181 ST(IND,1)=9999
0182 RETURN
0183 MRCNO=109
0184 NOREC=3
0185 ST(IND,1)=6010
0186 RETURN
0187 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 FIM - CREATILITY DEVELOPMENT MODEL

2 ENTER USERNAME 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 FIM.

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 FIM

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. VSOC 3. RWPSA

36 2. BSOC 4. SICA

37 PLEASE SELECT TYPE OF UPDATE

38 1. EVENT REPORTING ON

39 2. EVENT REPORTING OFF

40 3. MEASUREMENT 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 14)

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 CHANGES 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 (DDTTT)  
 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 IS REPORT COMPLETE  
 113 1. YES  
 114



```

172 1 3 4 5 6 7 8 9 2
173 0
174 0
175 0
176 0
177 -PRESS RETURN FOR NEXT INSTRUCTION--
178 LOOP -1- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
179 0
180 0
181 0
182 0
183 0
184 0
185 0
186 0
187 -PRESS RETURN FOR NEXT INSTRUCTION--
188 LOOP -2- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
189 0
190 0
191 0
192 0
193 0
194 0
195 0
196 0
197 -PRESS RETURN FOR NEXT INSTRUCTION--
198 LOOP -3- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
199 0
200 0
201 0
202 0
203 0
204 0
205 0
206 0
207 -PRESS RETURN FOR NEXT INSTRUCTION--
208 LOOP -4- LID/FAD CONVERSION TABLE, FOR LIDS 101-255
209 0
210 0
211 0
212 0
213 0
214 0
215 0
216 0
217 -PRESS RETURN FOR NEXT INSTRUCTION--
218 NODE HAS DESIGNATOR IN LOOP#
219 ALTERNATE GATEWAY FUNCTIONAL ADDRESS .....
220 MAXIMUM INPUT QUEUE SIZE (TO EXTERNAL) .....
221 MAXIMUM OUTPUT QUEUE SIZE (TO RTSTREAM) .....
222 MAXIMUM 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 GATT-2 2 1 2+3 10 1 NA 7 NA 3

```

229	GAT1-3	3	1	2,3	10	1	1	NA	12	NA	3
230	CRT4	4	2	1,3	8	1	1	8	12	41	4
231	HSIR	5	2	1,3	10	1	1	6	4	42	4
232	GAT2-1	6	2	1,3	10	1	1	NA	7	NA	4
233	GAT2-3	7	2	1,3	10	1	1	NA	6	NA	4
234	CRT8	8	3	1,2	8	1	1	8	12	41	4
235	GAT3-4	9	3	1,2	10	1	1	NA	14	NA	4
236	GAT3-2	10	3	1,2	10	1	1	NA	4	NA	4
237	GAT3-1	11	3	1,2	10	1	1	NA	7	NA	4
238	GAT4-3	12	4	3,5	10	1	1	NA	7	NA	8
239	GAT4-5	13	4	3,5	10	1	1	NA	7	NA	8
240	AUTOIN	14	4	3,5	10	1	1	NA	7	NA	8
241	TCCF	15	4	3,5	10	1	1	NA	7	NA	8
242	HSTA	16	4	3,5	10	1	1	NA	7	NA	8
243	SILC	17	4	3,5	10	1	1	NA	7	NA	8
244	CRT18	18	4	3,5	10	1	1	NA	7	NA	8
245	SECUR	19	4	3,5	10	1	1	NA	7	NA	8
246	SIG	20	NA	NA	NA	NA	NA	NA	NA	NA	NA
247	SSCI	21	5	NA	64	2	2	4	15	25	8
248	VSOC	22	5	NA	64	2	2	4	15	25	8
249	DSOC	23	5	NA	64	2	2	4	15	25	8
250	HSID	24	5	NA	64	2	2	4	15	25	8
251	CRT25	25	5	NA	64	2	2	4	15	25	8
252	BWBSA	26	5	NA	64	2	2	4	15	25	8
253	FIAC	27	5	NA	64	2	2	4	15	25	8
254	SDCA	28	5	NA	64	2	2	4	15	25	8

PLEASE ENTER ONE OF THE FOLLOWING:

1. DISPLAY RED STATUS
  2. DISPLAY AMBER STATUS
  3. DISPLAY SELECTED EQUIPMENT
  4. MANUALLY CHANGE STATUS OF SELECTED EQUIPMENT
- PLEASE SELECT ONE OF THE FOLLOWING:
1. CHANNEL 1-1000
  2. LINK 1-3
  3. SWITCH 1 OR 2
  4. MULTIFLEXOR 1-3
  5. TRANSMITTER 1-3
  6. RECEIVER 1-3
- PLEASE ENTER DEVICE NUMBER TO BE DISPLAYED (FORMAT I4)
- PLEASE ENTER DEVICE NUMBER TO BE CHANGED FOLLOWED BY NEW VALUE (FORMAT I4,I1)
- MODIFICATION COMPLETE
- PLEASE SELECT ONE OF THE FOLLOWING:
1. NEW CHANNEL, LINK, OR SWITCH
  2. NEW MODE OF OPERATION
  3. SAME MODE OF OPERATION
  4. LOGOUT

PRESS RETURN FOR NEXT INSTRUCTION

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  
 314  
 315  
 316  
 317  
 318  
 319  
 320  
 321  
 322

LOOP -5- LID/FAU CONVERSION TABLE FOR LIDS 101-255

0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0

-PRESS RETURN FOR NEXT INSTRUCTION-

PAGE 001

14:56:24

05-JUN-79

COMP2.COM

Program:

decwriter

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

PAGE 001

14:56:32

05-JUN-79

COMP2.COM

Program:

decwriter

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 messages from the BWBSA node.

```

0001 PROGRAM SIGGEN
0002 BITE 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,FRANGE,DRANGE,BRANGE
0006 REAL*8 VTRNK,DTRNK,BWID,VTR,BTR,DTR
0007 COMMON/EXP/ENC(4)
0008 COMMON/INT/ LFLAG,VFLAG,BFLAG,DGO,VGO,BGO
0009 COMMON/SIG/VMEAS(13),PRED(1203,3),VLL(6),BLL(13,3),DLL,
1 VMEAS(6),BRANGE(13,3),DRANGE,VTRNK,DTRNK,BWID,
2 ALARM,VTR,DTR,BTR
0010 DATA VRANGE/14.0,14.0,160.0,140.0,120.0,150.0/
0011 DATA VLL /-0.5,-0.5,-80.0,-70.0,-60.0,-70.0/
0012 DATA FRANGE,DLL/30.0,-15.0/
0013 DATA BRANGE/3.0,30.0,3600.0,20.0,100.0,4.0,70.0,3.0,
& 30.0,3600.0,20.0,100.0,30.0,
& 5.0,50.0,3530.0,19.0,91.0,4.0,70.0,6.0,
& 60.0,3530.0,19.0,91.0,27.0,
& 10.0,100.0,3600.0,20.0,100.0,3.9,69.0,
& 10.0,100.0,3600.0,20.0,100.0,25.0/
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/AAAAAAA',BBBB1111', 111110'/
0016 DATA ALARM/XXXX'/
0017 DFLAG=0
0018 VFLAG=0
0019 BFLAG=0
0020 VGO=1
0021 DGO=1
0022 BGO=1
0023 I1=0
0024 I2=0
0025 DO 999 J=1,1000
0026 RN=RN(I1,I2)
C
0027 CALL INTVEC
0028 DO 100 DO 150,I=1,500
C
C INITIALIZE VECTORS
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 IF(DFLAG .NE. 0) GOTO 651
0031 DFLAG=0
0032 IF(VFLAG .NE. 0) GOTO 654
0034 VFLAG=0
0035 IF(BFLAG .NE. 0) GOTO 657
0037 BFLAG=0
C
C ***** VSQC *****

```

PAGE 022

FORTRAN IV V02.1-11

```

0038      IVC=I
0039      DO 350 J=1,6
0040      JJ=J
0041      RN=RN*(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      VMEAS(J)=RN*VRANGE(J)+VLL(J)
0047      GOTO 500
0048      VMEAS(JJ)=RN*VRANGE(JJ)+VLL(JJ)
0049      PRED(IVC,1)=PRED(IVC,1)+1
0050      PRED(IVC,2)=JJ
0051      PRED(IVC,3)=VMEAS(JJ)
0052      IF(JJ .EQ. 6) GOTO 500
0054      DO 550 K=JJ+1,6
0055      VMEAS(K)=.5*VRANGE(K)+VLL(K)
0056      GOTO 520
0057      VMEAS(JJ)=RN*VRANGE(JJ)+VLL(JJ)
0058      GOTO 600
0059      II=I
0060      IS=JICVT(II,I14)
0061      CALL IMAGE(I14)
0062      RN=RN*(I1,I2)
0063      MON=RN*1000

C
C      VSQC REQUEST A INTERRUPT IF FLAG SET
C
0064      IF(VFLAG .EQ. 0) GOTO 659
0066      DO 660 K=1,6
0067      RN=RN*(I1,I2)
0068      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=JICVT(K4,K2)
0073      K=K2
0074      VMEAS(K)=PRED(VFLAG,3)
0075      IS=JICVT(VFLAG,I14)
0076      CALL IMAGE(I14)
0077      IVC=VFLAG
0078      RN=RN*(I1,I2)
0079      MON=RN*1000
0080      VGO=0
0081      CONTINUE

C
0082      IF(VGC .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

```

IMON PT

PAGE 003

```

FORTRAN IV      V02.1-11
0093      ENCODE(9,503,OUTQ(29)) VTRNE
0094      DO 711 J=1,4
0095      OUTQ(J+32)=ENC(J)
0096      ENCODE(2,504,OUTQ(37)) MON
0097      FORMAT(A4)
0098      502 FORMAT(A2)
0099      503 FORMAT(A8)
0100      504 FORMAT(A2)
0101      601 FORMAT(A2)
D          S WRITE(7,900)(VMEAS(IC),IC=1,5),IVC,
D          S (OUTQ(II),II=29,35),MON
D 900 FORMAT(1X,'#RVSCC->',I6,2X,BA1,2X,I6,/)
D          S
D          DO 506 J1=1,40
0102      CALL IFOKFE(177416,OUTQ(J1))
0103      505 IF(IPEEK(177414).NE.200) GOTO 505
0104      506 CONTINUE
0105      CALL DELAY
0106      998 CONTINUE
0107      VGO=1
0108      IF(VFLAG.NE.0) GOTO 655
0109      0110
C          C *****
C          C *****
C          C *****
C          C *****
0112      IDC=I+500
0113      DO 1350 J=1,3
0114      JJ=J
0115      RN=RN+(11,12)
0116      IF(RN.GE.998.OR.RN.LE.003) GOTO 1400
0117      IF(RN.GE.999.OR.RN.LE.005) GOTO 1450
0118      1350 VMEAS(J)=RN*DRANGE+DLL
0119      GOTO 1500
0120      1400 VMEAS(JJ)=RN*DRANGE+DLL
0121      PRED(IEC,1)=PRED(IDC,1)+1
0122      PRED(IDC,2)=JJ
0123      PRED(IDC,3)=VMEAS(JJ)
0124      1600 IF(JJ.EQ.3) GOTO 1500
0125      DO 1550 K=JJ+1,3
0126      VMEAS(K)=.5*DRANGE+DLL
0127      GOTO 1500
0128      1450 VMEAS(JJ)=RN*DRANGE+DLL
0129      1500 GOTO 1500
0130      1500 GOTO 1500
0131      1500 I1=I+500
0132      IS=JICVT(I1,I14)
0133      CALL IMAGE(I14)
0134      RN=RN+(11,12)
0135      MON=RN+1007
0136      C
0137      C DSOC REQUESTS A INTERRUPT IF FLAG IS SET
0138      C
0139      IF(DEFIAG.EQ.0) GOTO 662
0140      DO 669 K=1,3
0141      RN=RN+(11,12)
0142      VMEAS(K)=RN*DRANGE+DLL

```

PAGE 004

```

FORTRAN IV      V02.1-11
0143 IF(PRED(DFLAG,1),FQ,0) GOTO 668
0145 IS=JAIIX(PRED(DFLAG,2),K4)
0146 IS=IJCVT(K4,K2)
0147 K=K2
0148 VMEAS(K)=PRED(DFLAG,3)
0149 I14=VFLAG-500
0150 IS=IJCVT(I14,I14)
0151 CALL IMAGE(I14)
0152 RN=RN(J1,I2)
0153 MON=RN*1000
0154 DCO=0
0155 662 CONTINUE
C
0156 IF(DGO .EQ. 1) GOTO 915
0158 ICNT=26
0159 ENCODE(2,501,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(6,509,OUTQ(17)) DTRNK
0165 DO 712 J=1,4
0166 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
D 6
D 901 FORMAT(IX,WRDSQC->,'3F12.6/,10X,16,2X,8A1,2X,16,/)
0171 DO 512 J1=1,40
0172 511 IF(IPREK('177514').NE.'200') GOTO 511
0174 CALL IPOKER('177516,OUTQ(J1))
0175 512 CONTINUE
0176 CALL DELAY
0177 DGO=1
0178 IF(DFLAG.NE.0) GOTO 552
C
C***** EWBSA *****
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
0188 IGRN=0
0189 IBX=1000+LINK
0190 2200 DO 2350 J=1,13
0191 JJ=J
0192 RN=RN(J1,I2)
0193 IF(RN.GE..998 .OR. RN.LE..003) GOTO 2400
0195 IF(RN.GE..990 .OR. RN.LE..005) GOTO 2450
0197 VMEAS(J)=RN#BRANGE(J,LINK)+BLI(J,LINK)
0198 IGRN=1

```

IGRFEN VALS

PAGE 005

FORTRAN IV V02.1-11

```

0199      GOTO 2500
0200      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)=TMEAS(JJ)
0204      IF(JJ.EQ.13) GOTO 2500
0206      DO 2550 K=JJ+1,13
0207      VMEAS(K)=.5*BRANGE(K,LINK)+BLL(K,LINK)
0208      GOTO 2500
0209      VMEAS(JJ)=RN*BRANGE(JJ,LINK)+BLL(JJ,LINK)
0210      GOTO 2600
0211      IS=JICVT(LINK,I14)
0212      CALL IMAGE(I14)
0213      RN=RN*(I1,I2)
0214      MON=RN*1000
      C
      C      BWSQ REQUESTS A INTERRUPT IF FLAG SET
      C
0215      IF(BFLAG.EQ.0) GOTO 663
0217      IBX=1000+BFLAG
0218      DO 664 K=1,13
0219      RN=RN*(I1,I2)
0220      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      IS=JICVT(BFLAG,I14)
0228      CALL IMAGE(I14)
0229      RN=RN*(I1,I2)
0230      MON=RN*1000
0231      BGO=0
0232      GOTO 663
      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  
66157 SBIE-AD-E100 313 NL

UNCLASSIFIED

5 OF 5  
AD  
A078391



END  
DATE  
FILMED  
1-80  
DDC

```

FORTRAN IV      V02.1-11      PAGE 005

0253      ENCODE(2,517,OUTQ(65)) MON
0254      FORMAT(A2)
0255      FORMAT(A8)
0256      FORMAT(A2)
D        S      WRITE(7,902)(VMEAS(IC),IC=1,13),LINK,(OUTQ(ID),ID=57,64),
D        S      MON
D 902      FORMAT(1X,'WRBWSA->',5F12.6,/,12X,5F12.6,/,12X,
D        S      3F12.6,/,12X,16,2X,8A1,2X,16,/)
D        S      DO 519 J1=1,80
0257      IF(IPEEK('175614').NE.'200') GOTO 518
0258      CALL IPOKEB('175616,OUTQ(J1))
0259      CONTINUE
0260      CALL DELAY
0261      BGO=1
0262      IF(BFLAG.NE.0) GOTO 658
0263      C
0264      IF(ICRN.EQ.0) GOTO 150
0265      RN=RAN(11,12)
0266      IF(RN.LT..9) GOTO 150
0267      RN=RAN(11,12)
0268      IEQ=3*RN*.5
0269      IF(IEQ.EQ.0) IEQ=1
0270      ICNT=14
0271      ENCODE(2,601,OUTQ(1)) ICNT
0272      ENCODE(4,520,OUTQ(3)) ALARM
0273      ENCODE(2,521,OUTQ(11)) LINK
0274      ENCODE(2,522,OUTQ(13)) IEQ
0275      FORMAT(A4)
0276      FORMAT(A2)
0277      FORMAT(A2)
D 520      DO 523 J1=1,80
0278      IF(IPEEK('175614').NE.'200') GOTO 524
0279      CALL IPOKEB('175616,OUTQ(J1))
0280      CONTINUE
0281      CALL DELAY
0282      WRITE(7,903)LINK,IEQ
0283      FORMAT(1X,'WRBWSA->',XXXX',2X,14,2X,14,/)
0284      C*****
0285      C
0286      C*****
0287      C
0288      C*****

0289      150      CONTINUE
0290      GOTO 100
0291      STOP
0292      END
C$$$$$$

```

!NO ALARM

IEQUIP

!START OVER PPOG

PAGE 021

```
FORTRAN IV      V02.1-11
0001      SUBROUTINE IMAGE(IN)
0002      BYTE ENC
0003      INTEGER*4 IN
0004      COMMON/EXP/ENC(4)
0005      ENCODE(4,1,ENC) IN
0006      FORMAT(4I)
0007      DO 2 J=1,4
0008      IF(ENC(J)).LE.57 .AND. ENC(J).GE.48) GOTO 2
0009      ENC(J)=0
0010      CONTINUE
0011      RETURN
0012      END
0013      C$$$$$$$
```

PAGE 001

```
FORTRAN IV      V02.1-11
0001      SUBROUTINE DELAY
0002      DO 1 J1=1,2
0003      DC 2 J2=1,10000
0004      CONTINUE
0005      CONTINUE
0006      RETURN
0007      END
```

J  
SIG.MACRO

MACRO V03.02B 01:22:36 PAGE 1

.TITLE SIG.MACRO  
.IDENT/V1.0/  
.GLOBL INTVEC,REV5QC,RDDSQC,RDEWSA  
.PSECT

;INTERFACE ADDRESS

;DSQC  
RCSH1=177510  
RBUF1=177512  
XCSR1=177514  
XBUF1=177516

;VSQC  
RCSR2=177412  
RBUF2=177412  
XCSR2=177414  
XBUF2=177416

;BWSA  
RCSR3=175610  
RBUF3=175612  
XCSR3=175614  
XBUF3=175616

;VECTOR INIT

INTVEC: RESET  
MOV #100, R0  
MOV #TIME, (R0)  
MOV #320, R0  
MOV #RDV5QC, (R0)+  
MOV #340, (R0)+  
BIS #100, G#RCSR2  
MOV #310, R0  
MOV #RDDSQC, (R0)+  
MOV #340, (R0)+  
BIS #100, G#RCSR1  
MOV #300, R0  
MOV #RDEWSA, (R0)+  
MOV #340, (R0)+  
BIS #100, G#RCSR3  
MIPS R0  
RTS PC

TIME:

RDV5QC: MOV R0, -(SP)  
TSTB G#RCSR2  
BPL 1\$  
MOVE G#RBUF2, R0  
BIC #177400, R0  
CMPB #122, R0  
BNE 4\$  
MOV #0, VGO  
ER 1\$

1  
2  
3  
4 000000  
5  
6  
7  
8  
9 177510  
10 177512  
11 177514  
12 177516  
13  
14  
15 177410  
16 177412  
17 177414  
18 177416  
19  
20 175610  
21 175612  
22 175614  
23 175616  
24  
25  
26  
27  
28 000005  
29 000002  
30 000006  
31 000012  
32 000015  
33 000022  
34 000026  
35 000034  
36 000044  
37 000044  
38 000050  
39 000056  
40 000062  
41 000065  
42 000072  
43 000100  
44 000104  
45 000106  
46 000110  
47  
48  
49 000112  
50 000114  
51 000120  
52 000122  
53 000126  
54 000132  
55 000136  
56 000140  
57 000146

000100  
000110  
000320  
000112  
000340  
000100 177410  
000310  
000160  
000340  
000100 177510  
000300  
000226  
000340  
000100 175610  
000000  
000002  
010046  
105737 177410  
100015  
177412  
177400  
000122  
000104  
000000 000010  
000402

000000  
000112  
105737  
100015  
177412  
177400  
000122  
000104  
000000 000010  
000402

SIG.MACRO MACRO V06.02B 01:22:36 PAGE 1-1

```

58 000150 010067 000002' 4$: MOV R0, VFLAG
59 000154 012600 1$: MOV (SP)+, R0
60 000156 000002 RTI
61
62 000160 010046 RDDSOC: MOV R0, -(SP)
63 000162 105737 TSTB G#RCSH1
64 000166 100015 BPL 2$
65 000170 113700 MOV G#RBUF1, R0
66 000174 042700 BIC #177400, R0
67 000200 122700 CMPB #122, R0
68 000204 001004 BNE 5$
69 000206 012767 LGO #0, IGO
70 000214 000402 BR 2$
71 000216 010067 MOV R0, DFLAG
72 000222 012600 (SP)+, R0
73 000224 000002 RTI
74
75 000226 010046 EDWMSA: MOV R0, -(SP)
76 000230 105737 TSTB G#RCSH3
77 000234 100015 BPL 3$
78 000236 113700 MOV G#RBUF3, R0
79 000242 042700 BIC #177400, R0
80 000246 122700 CMPB #122, R0
81 000252 001004 BNE 6$
82 000254 012767 LGO #0, EGO
83 000262 000402 BR 3$
84 000264 010067 MOV R0, EFLAG
85 000270 012600 (SP)+, R0
86 000272 000002 RTI
87
88 000000 .PSECT INT,RW,D,GEL,BLL,OVR
89 000000
90
91 000000 DFLAG: .WORD 0
92 000002 VFLAG: .WORD 0
93 000004 BFLAG: .WORD 0
94 000006 IGO: .WORD 0
95 000010 VGO: .WORD 0
96 000012 EGO: .WORD 0
97
98 000000 .END INTVEC
99

```

SIG.MACRO MACRO 003.02B 01:22:36 PAGE 1-2

SYMBOL TABLE

BFLAG	000004P	002	REUF1 = 177512	HCSE3 = 175610	VFLAG	000002F	002	XBUF3 = 175616
EGC	000012R	002	REUF2 = 177412	RDEWCA	000226R	002	ICSE1 = 177514	
DFLAG	000000H	002	REUF3 = 175612	RDESWC	000120R	002	XCS2 = 177414	
LGC	000005R	002	RCSE1 = 177510	RDSQC	000128G		XCS3 = 175614	
INTVEC	000002G		RCSE2 = 177412	TIME	000110R			
.	ABS.	000000	000					
		000274	001					
	INT	000214	002					
	ERRORS DETECTED:	0						

VIRTUAL MEMORY USED: 294 WORDS ( 2 PAGES)  
 DYNAMIC MEMORY AVAILABLE FOR 56 PAGES  
 DK:SIGMAC,DK:SIGMAC=DR:SIGMAC.M20

FORTRAN IV V01C-03 FRI 03-JUN-77 12:26:53 PAGE 001  
 CORE=08K, UIC=E20,201 SDCAS.0BJ,SDCAS.LST=SDCAS.FOR

```

C THIS PROGRAM -SIMI(SDCAS.FOR) WRITES SIMULATED SWITCH
C CONDITION REPORTS TO SDCA LOOP 5
C THIS PROGRAM RUN ON HOST 11/40
C
0001 DIMENSION ICRT(2,16),IRPT(16)
0002 CALL ASSGN(1,'TI:')
0003 DATA ICRT(1,1),ICRT(1,2),ICRT(1,3)/1,51,25/
0004 DATA ICRT(1,4),ICRT(1,5),ICRT(1,6)/25,10,50/
0005 DATA ICRT(1,7),ICRT(1,8),ICRT(1,9)/50,10,10/
0006 DATA ICRT(1,10),ICRT(1,11),ICRT(1,12)/10,128,128/
0007 DATA ICRT(1,13),ICRT(1,14),ICRT(1,15)/128,128,10/
0008 DATA ICRT(1,16)/10/
0009 DATA ICRT(2,1),ICRT(2,2),ICRT(2,3)/2,25,10/
0010 DATA ICRT(2,4),ICRT(2,5),ICRT(2,6)/10,25,40/
0011 DATA ICRT(2,7),ICRT(2,8),ICRT(2,9)/40,5,5/
0012 DATA ICRT(2,10),ICRT(2,11),ICRT(2,12)/10,64,64/
0013 DATA ICRT(2,13),ICRT(2,14),ICRT(2,15)/64,64,5/
0014 DATA ICRT(2,16)/5/
0015 I1=0
0016 I2=0
0017 I10=1
0018 IRPT(1)=I10
0019 R1=IRAN(I1,I2)
0020 DO 20 I=3,16
0021 IRPT(I)=ICRT(I10,I)*R1
0022 IF(R1 .LE. .98) GOTO 100
C ELSE SATURATED COND.
0024 IR2=13*IRAN(I1,I2)
0025 IF(IR2 .LT. 3) IR2=3
0027 IF(IR2 .GE. 16) IR2=16
0029 IRPT(IR2)=(R1*5)+ICRT(I10,IR2)
0030 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 FORMAT(1X,16I4)
C WAIT 5.5 SEC
0034 T1=SECND(0.)
0035 DELTA=SECND(T1)
0036 IF(DELTA .LT. 5.5) GOTO 150
0038 IF(I10 .EQ. 2) GOTO 200
0040 I10=2
0041 GOTO 10
0042 STOP
0043 END

```

PAGE 001

12:00:00

01 JUL -79

COMP20.COM

PROGRAM:

COMPILER

FOR SIGGEN/NOI LINE  
MAC SIGMA



PAGE 001

1:00:00

01 .000 -79

PROGRAM: ERK20.F0M

00000000

ASS D20 BK  
R LINK  
DX1:STGCN,DX1:MAC=DX1:STGCN/RE/AC/W.1  
DX1:CLMRC/2  
451MK1

ASS DX1 BK