PDP11

SERIAL BUS EXERCISER

MD-11-DZKCH-A

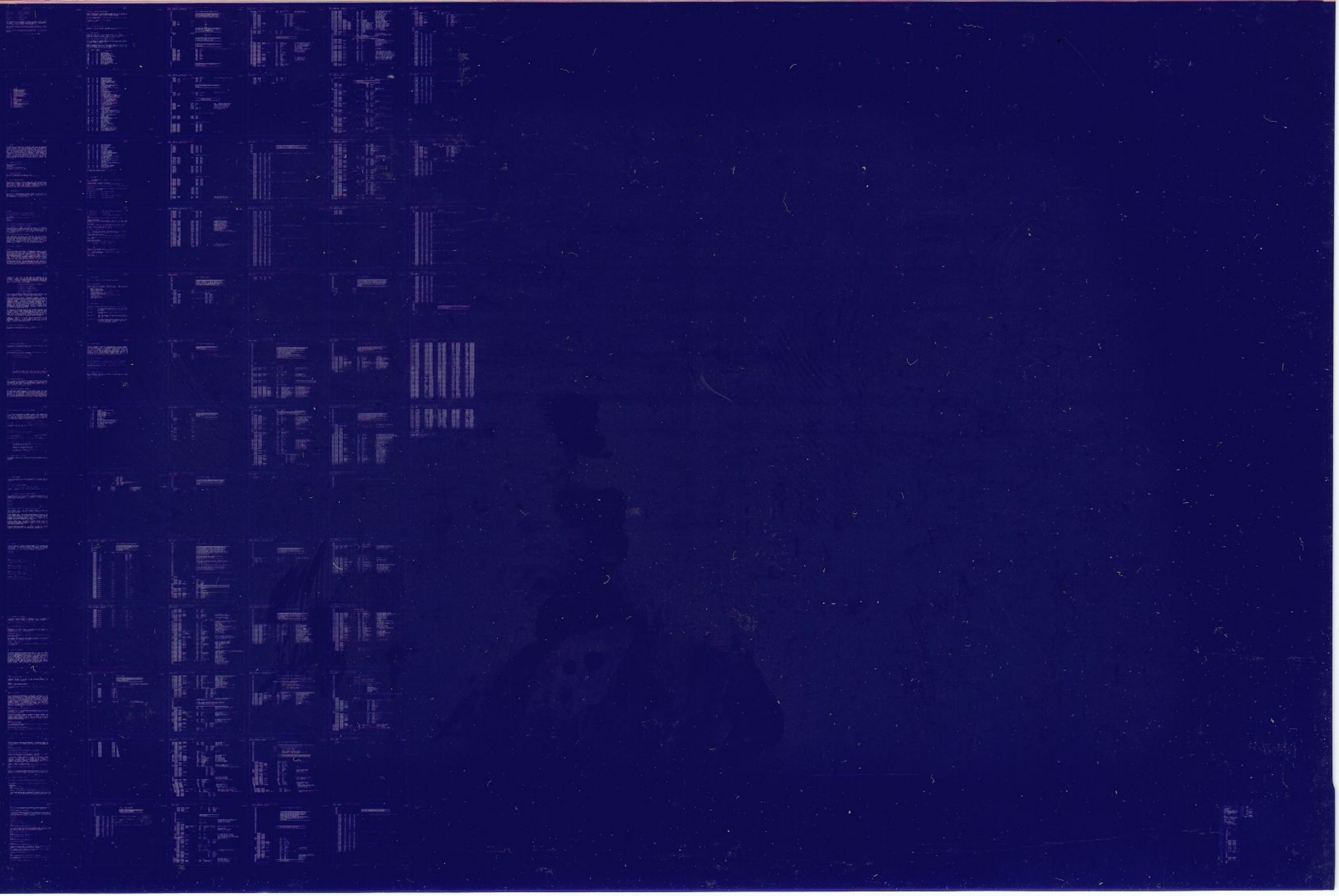
EP-DZKCH-A-DL-A

FICHE 1 OF 1

DEC 1977

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE	MAINDEC-11-DZKCH-A-D
PRODUCT NAME	SERIAL BUS EXERCISER
DATE	JUNE 1977
MAINTAINER	DIAGNOSTIC GROUP
AUTHOR.	CLEM WALSH

TABLE OF CONTENTS

1 0 ABSTRACT

THE PURPOSE OF THIS EXERCISER IS TO QUICKLY VERIFY THAT THE SERIAL BUS SYSTEM, (ISB11A, SERIAL BUS AND UP TO A TOTAL OF 63 FDC TERMINALS) SHOWS THE EXISTENCE OF NO MAJOR PROBLEMS, UNDER AN RSX11M TYPE OPERATING SYSTEM. THE PROGRAM ALLOWS THE OPERATOR TO EXERCISE ANY, OR ALL OF THE FDC TERMINALS (UP TO 63 AT ONE TIME) ATTACHED TO THE SERIAL BUS THE EXERCISER DRIVES ANY TERMINAL WHICH SUBSCRIBES TO THE CUPRENT SERIAL BUS PORT PROTOCOL(IE.-LOOP MODE). S DATA TRANSFER TO TERMINALS ON THE SERIAL BUS ARE PRECEDED BY A "LOOP" MAINTENANCE COMMAND THIS CAUSES THE ADDRESSED TERMINAL TO RETURN THE RECEIVED DATA TO THE HOST SYSTEM, EXACTLY AS IT WAS RECEIVED BY THE TERMINAL DATA PATTERNS AND DATA MESSAGE LENGTHS ARE VARIED TO EXERCISE THE SERIAL BUS SYSTEM TO MAXIMUM ALL ERRORS ARE REPORTED THROUGH A HOST SYSTEM CONSOLE

2 0 HARDWARE REQUIREMENTS

PDP-11 FAMILY OF COMPUTERS, EXCEPT THE LSI-11
CONSOLE TELETYPE
16K MEMORY
ISB11A (SERIAL BUS CONTROLLER)
SERIAL BUS - OR ISB11-A TEST CABLE
1-63 FACTORY DATA COLLECTION TERMINALS

3 0 SOFTWARE REQUIREMENTS

RSX-11M DISK BASED REALTIME OPERATING SYSTEM
THE DPM - "DISTRIBUTED PLANT MANAGEMENT" SOFTWARE PACKAGE

3 1 RSX-11M OPERATING SYSTEM

THE RSX-11M IS A SMALL TO MEDIUM SIZED REAL TIME MULTIPROGRAMMING SYSTEM WHICH UTILIZES 16K TO 124K WORDS OF MEMORY, OR 128K TO 1024K WORDS OF PDP-11/70 MEMORY THE 16K SYSTEM ALLOWS 8K FOR USER TASKS AND INCLUDES A SUBSET MACRO PACKAGE. A MINIMUM OF 24K WORDS ARE REQUIRED FOR FULL MACRO SUPPORT,CONCURRENT PROGRAM DEVELOPMENT AND APPLICATION TASKS EXECUTION, OR MEMORY MANAGEMENT SUPPORT.

4 0 PROGRAM GENERATION

THE FILE TO BE LOADED AND RUN IN SYSTEM'S MEMORY IS THE TASK FILE DZKCH.TSK. IT IS GENERATED FROM THE SOURCE FILE DZKCH MAC BY USING RSX-11M INDIRECT FILES. AN INDIRECT FILE DZKCH CMD CONTAINS A LIST OF MCR COMMANDS, AND IS INVOKED BY TYPING

'@LDZKCH"

UPON WHICH THE CONSOLE WILL PRINT THE FOLLOWING

```
MAC DZKCH OBJ,DZKCH.LST=DIAGSUPER/ML,DZKCH MAC

TKB @TKBDZKCH
DZKCH TSK,DZKCH MAP=DZKCH OBJ,DIAGSUPER
/ZK
UNITS=75
STACK=512
//
```

THE FOLLOWING IS AN EXPLANATION OF THE CONSOLE PRINTOUT

DZKCH OBJ,DZKCH LST=DIAGSUPER/ML,DZKCH MAC

THIS ASSEMBLES THE SOURCE FILE DZKCH MAC WITH THE DIAGNOSTIC SUPERVISOR MACRO PACKAGE DIAGSUPER/ML, INTO AN OBJECT FILE DZKCH.OBJ FOR TASK BUILDING BY THE RSX11-M TASK BUILDER. THE SWITCH /ML ON THE FILE DIAGSUPER SPECIFIES THE FILE AS A USER'S MACRO LIBRARY THE PDP-11 DIAGNOSTIC SUPERVISOR IS NEEDED FOR ASSEMBLING THE PROGRAM, AND IS DESCRIBED IN 4 1

DZKCH TSK,DZKCH MAP=DZKCH OBJ,DIAGSUPER

THIS TASK-BUILDS THE OBJECT FILE DZKCH OBJ WITH THE DIAGNOSTIC SUPERVISOR OBJECT LIBRARY DIAGSUPER, INTO A TASK IMAGE DZKCH.TSK TO BE RUN IN SYSTEM'S MEMORY. DZKCH.MAP IS A FILE WHICH CONTAINS THE MEMORY ALLOCATION MAP THE TASK BUILDER SWITCHES /, UNITS=75, STACK=512, //, ARE NOT PRINTED OUT ON THE CONSOLE, BUT ARE CONTAINED IN THE INDIRECT COMMAND FILE "TKBDZKCH CMD" AND ARE DESCRIBED IN THE RSX-11M TASK BUILDER REFERENCE MANUAL RSX-11M TASK BUILDER REFERENCE MANUAL

4 1 DIAGNOSTIC SUPERVISOR

GENERAL--------

THE PDP-11 DIAGNOSTIC SUPERVISOR IS A MODULARIZED DIAGNOSTIC MONITOR WHICH SERVES THE PDP-11 FAMILY OF COMPUTERS BY PROVIDING RUN-TIME SUPPORT FOR FUNCTIONAL OR REPAIR LEVEL DIAGNOSTICS THE DIAGNOSTICS ARE DESIGNED FOR SUPERVISOR COMPATIBILITY, AND CREATED FOR DESIGN ENGINEERING, MANUFACTURING, OR FIELD ENGINEERING USAGE FUNCTION LEVEL PROGRAMS PROVIDE FOR ERROR DIAGNOSIS AT A HARDWARE FUNCTION LEVEL, WHILE REPAIR LEVEL DIAGNOSTICS PROVIDE FOR REPAIR AT A MODULE LEVEL. AS SUCH, THEY EXERCISE, DIAGNOSE, OR TEST INDIVIDUAL PERIPHERALS, MEMORY SUBSYSTEMS, PROCESSOR OPTIONS, OR ENTIRE SYSTEMS

SEPARATE STAND ALONE (OFF LINE) AND USER MODE (ON LINE) VERSIONS OF THE SUPERVISOR COMMONLY PROVIDE NON-DIAGNOSTIC SERVICES FOR A SINGLE PROGRAM, OR A SCRIPT OF PROGRAMS, THAT HAVE BEEN PREVIOUSLY ASSEMBLED IN A STAND ALONE ENVIRONMENT THE SUPERVISOR SHARES RESIDENCY WITH THE

DIAGNOSTIC(S) ONLY, WHILE IN USER MODE THE SUPERVISOR AND THE DIAGNOSTIC(S) RESIDE WITH BOTH AN OPERATING SYSTEM AND THE USER PROGRAM(S) HOWEVER, IN EITHER ENVIRONMENT, SUPERVISOR SERVICES ARE ELICITED BY A RESIDENT DIAGNOSTIC VIA THE GENERATION OF UNIQUE MACRO CALLS TO THE SUPERVISOR IN THIS MANNER THE FOLLOWING NON-DIAGNOSTIC SERVICES CAN BE INITIATED FOR A PROGRAM.

THE EXECUTION OF EACH SECTION OF A DIAGNOSTIC

THE LOOPING OF TEST PROGRAMS

THE REPORTING OF HARDWARE ERRORS

THE REPORTING OF UNEXPECTED INTERRUPTS

THE PRINTING OF MESSAGES

FUNCTION LEVEL PROGRAMS CAN OPERATE IN EITHER A STAND ALONE OR A USER MODE ENVIRONMENT WITHOUT SOURCE CODE MODIFICATION. HOWEVER, REPAIR LEVEL PROGRAMS MUST OPERATE IN A STAND ALONE ENVIRONMENT ONLY.

ONCE THE SUPERVISOR IS LOADED A STANDARDIZED OPERATOR INTERFACE IS ESTABLISHED, PROVIDING A COMMUNICATIONS PATH THROUGH WHICH AN OPERATOR CAN DIRECT THE SUPERVISOR TO INITIATE THE LOADING AND EXECUTION OF A DIAGNOSTIC PROGRAM INTERFACE DIALOGUE ALSO ALLOWS AN OPERATOR TO EXAMINE AND/OR MODIFY THE CONTENT OF THE GENERAL REGISTERS, AND THE CONTENTS OF THE MEMORY LOCATIONS UTILIZED BY THE DIAGNOSTIC(S) IN ADDITION, THE INTERFACE ALLOWS AN OPERATOR TO ACCESS SEPARATE UTILITY PROGRAMS THAT ARE ALSO LOCATED ON THE DIAGNOSTIC LOAD MEDIA, AND ARE ASSOCIATED WITH THE SUPERVISOR PACKAGE THE UTILITY PROGRAMS PERMIT AN OPERATOR TO INSTALL FIELD CHANGES TO A DIAGNOSTIC, AND TO BUILD OR MODIFY SCRIPT AND CONFIGURATION FILES

THE SERVICING OF FUNCTIONAL OR REPAIR LEVEL PROGRAMS OPERATING UNDER THE CONTROL OF A STAND ALONE VERSION OF THE SUPERVISOR, CAN BE INDIRECTLY CONTROLLED FROM A SEPARATE COMPUTER BY A LOCAL OR REMOTE DIAGNOSTIC MONITOR, SUCH AS THE AUTOMATED PRODUCT TEST (APT/APT-RD) UNDER INDIRECT CONTROL, AN OPERATOR INTERFACE IS ESTABLISHED WITH THE LOCAL (E.G APT) OR REMOTE (E.G APT-RD) MONITOR AND THE SUPERVISOR IS DIRECTED TO LOAD AND EXECUTE PROGRAMS VIA MONITOR COMMANDS

FINALLY, IN ADDITION TO THE OFF AND ON LINE VERSIONS OF THE SUPERVISOR, THERE IS A SPECIAL VERSION DEFINED AS A PROGRAM DEVELOPMENT SYSTEM(PDS) THE PDS VERSION INCLUDES BOTH DEBUG AND UPDATE UTILITIES AS CORE RESIDENT FEATURES AND IS ESSENTIALLY A SUPEF SUBSET OF THE BASIC STAND ALONE VERSION

5 0 PDP11 FAMILY COMPATIBILITY

THIS TEST WILL RUN ON THE PDP-11 FAMILY OF COMPUTERS EQUIPPED WITH THE HARDWARE IN 2 0, TOGETHER WITH THE SOFTWARE IN 3 0

6 0 OPERATING SYSTEM COMPATIBILITY

THE EXERCISER IS DESIGNED TO RUN WITH RSX11M/S-BASED OPERATING SYSTEMS ASSEMBLED WITH THE DIAGNOSTIC SUPERVISOR

7 0 TESTING METHODOLOGY

BEFORE EXECUTION OF THIS TEST IS ATTEMPTED, THE FOLLOWING TESTS SHOULD BE SUCCESSFULLY RUN IN THE FOLLOWING ORDER·

1 ALL APPLICABLE PDP11 C P U TESTS

2 ISB11A STAND ALONE DIAGNOSTICS

A DZKCC

B DZKCD

C DZKCA

D DZKCE

E DZKCF

F FOLLOWING THESE STAND ALONE TESTS, THE ISB11A SHOULD BE CONNECTED TO 1-63 RT801, 803, OR 805 TERMINALS VIA THE SERIAL BUS, OR ISB11-A TEST CABLE AND THE EXERCISER SHOULD BE RUN

8 0 PROGRAM CONSIDERATIONS

THIS IS A FUNCTION LEVEL PROGRAM, AS OPPOSED TO A REPAIR LEVEL PROGRAM (AS EXPLAINED IN 8 1 AND 8 1 1), DESIGNED FOR RSX-11M AND SUPERVISOR COMPATABILITY, AND CREATED FOR DESIGN ENGINEERING, MANUFACTURING, OR FIELD ENGINEERING USAGE AS A FUNCTION LEVEL PROGRAM, IT EXERCISES THE FDC TERMINALS ON THE SERIAL BUS SYSTEM

8 1 FUNCTION LEVEL DIAGNOSTIC PROGRAMS

FUNCTIONAL LEVEL PROGRAMS ARE FIRST LEVEL DIAGNOSTICS THAT ARE SIMILAR TO USER MODE LEVEL-2 PROGRAMS IN THAT THEY REQUIRE THE TOTAL DEDICATION OF THE DEVICE UNDER TEST WHILE DIRECT ACCESS TO THE DEVICE REGISTERS IS NOT PERMITTED. HOWEVER, DATA CAN BE EXCHANGED WITH A DEVICE VIA THE IMPLEMENTATION OF I/O ROUTINES SUCH AS THOSE ENGENDERED BY THE QIO$ DIRECTIVES USED IN RSX-11M IT SHOULD BE NOTED, HOWEVER, THAT RUNNING UNDER OPERATING SYSTEM CONTROL, STABLE PROGRAM LOOPS

SEQ 0007

CANNOT BE GAUPANTEED FOR FUNCTION LEVEL PROGRAMS

3 1 1 REPAIR LEVEL DIAGNOSTIC PROGRAMS - REPAIR LEVEL PROGRAMS ARE SECOND LEVEL DIAGNOSTICS THAT ARE SIMILAR TO USER MODE LEVEL-3 PROGRAMS IN THAT THEY REQUIRE THE TOTAL DEDICATION OF THE DEVICE UNDER TEST, AND DIRECT ACCESS TO DEVICE REGISTERS IS ALLOWED IN ADDITION, SINCE REPAIR LEVEL DIAGNOSTICS CANNOT BE RESIGNED TO A USER MODE (ON LINE) ENVIRONMENT, PROGRAM LOOPS ARE APPLICABLE

9 0 XXDP

THE PROGRAM RUNS ON LINE TO THE OPERATING SYSTEM RSX-11M, AND THEREFOPE WILL NOT RUN UNDER XXDP

10 0 ACT/APT

THE PROGRAM IS NEITHER ACT NOR APT COMPATIBLE

11 0 MEMORY MANAGEMENT

THIS PROGRAM DOES NOT DIRECTLY UTILIZE OR TEST MEMORY MANAGEMENT

12 0 EXECUTION TIME

1 DATA TRANSACTION- ONE DATA PATTERN, ONE DATA LENGTH, TO ONE TERMINAL, ESTIMATED TIME 7MS

2 PARTIAL PASS- ONE PATTERN, ALL DATA LENGTHS, TO ALL (63) TERMINALS IS ESTIMATED TO BE 50MS

3 FULL PASS- ALL PATTERNS, ALL DATA LENGTHS, TO ALL (63) TERMINALS TAKES TO 2 5 SECONDS

13 0 PROGRAM CONTROL PROCEDURES

THE PROGRAM IS RUN ON LINE IN RSX-11M OPERATING ENVIRONMENT AND IS CONTROLLED BY RSX (MCR) OR DIAGNOSTIC SUPERVISOR MONITOR COMMAND LANGUAGE

13 1 COMMAND LANGUAGE

THIS SECTION DESCRIBES THE OPERATOR COMMANDS, THAT PROVIDES FOR THE LOADING AND MANIPULATION OF PROGRAMS BY RSX-11M AND THE DIAGNOSTIC SUPERVISOR

13 1 1 RSX-11M COMMAND LANGUAGE -

PUN - THE PUN DIRECTIVE CAUSES A TASK TO BE REQUESTED

CONTROL C - ACTIVATING CONTROL C GETS THE OPERATOR BACK TO MCR AND ABORTS THE PROGRAM

13 1 1 1 DIAGNOSTIC SUPERVISOR COMMAND LANGUAGE -

UPON PROGRAM EXECUTION, AND TO RETURN TO SUPERVISOR COMMAND MODE, THE OPERATOR SHOULD TYPE "CONTROL A" THE PROGRAM WILL RESPOND WITH THE PROMPT DCP-A>, WHEREIN ANY ONE OF THE DIAGNOSTIC SUPERVISOR COMMANDS CAN BE ENTERED FOR EXAMPLE

CONTROL A

DCP-A> STA

DCP-A>RES

DCP-A>CON

DCP-A>CON

THESE D AGNOSTIC SUPERVISOR COMMANDS ARE DESCRIBED BELOW

START PROGRAM (STA) - THE START PROGRAM COMMAND INITIATES THE EXECUTION OF THE PROGRAM CURRENTLY CONTAINED IN MAIN MEMORY, INCLUDING THE DIALOGUE PORTIONS

RESTART PROGRAM (RES) - THE RESTART PROGRAM COMMAND IS SIMILAR TO THE START PROGRAM COMMAND WITH THE EXCEPTION BEING THAT DIALOGUE PORTIONS OF THE PROGRAM CAN ONLY BE EXECUTED VIA AN APPROPRIATE OPERATOR RESPONSE TO A SUPERVISOR QUERY HOWEVER IT IS ASSUMED THAT THE REQUIRED CONFIGURATION PARAMETERS HAVE BEEN LOADED, ALONG WITH THE PROGRAM, PRIOR TO THE ISSUANCE OF THIS COMMAND.

RETURN TO PROGRAM (CON) - THE RETURN TO PROGRAM COMMAND ALLOWS THE FXECUTION OF THE DIAGNOSTIC PROGRAM TO RESUME AT THE FIRST INSTRUCTION FOLLOWING THE CURRENT SUPERVISOR CALL HOWEVER, IF DESIRED, NEW FLAG CONDITIONS MAY BE SPECIFIED

RETURN TO SUPERVISOR (CONTROL A) - THE RETURN TO SUPERVISOR COMMAND INITIATES THE EXECUTION OF THE CLEANUP CODE, CONTAINED IN THE ACTIVE PROGRAM, AND PROVIDES AN EXIT TO THE SUPERVISOR COMMAND MODE

SEQ 0009

14 0 LOADING

SINCE THE EXERCISER IS PART OF THE DPM SOFTWARE, IT IS REQUESTED AND LOADED BY THE RSX-11M COMMAND "RUN DZKCH" WHERE "DZKCH" IS THE PROGRAM RELEASED NAME A PROMPT IS THEN ISSUED TO THE CONSOLE WHICH INITIATED THE PROGRAM THE FOLLOWING IS A CONSOLE PRINTOUT OF AN EXAMPLE OF PROGRAM DIAGLOGUE (NOTE OPERATOR'S RESPONSE IS UNDERLINED)

```
>RUN DZKCH
------------
DCP-A>STA
      -----

# UNITS (D) ? 8
            --

UNIT 1
ENTER CONTROLLER NUMBER   (O) 0 ?
                             --
ENTER TERMINALS ON THAT CONTROLLER   (D) 1 ? 1-4
                                           ----

UNIT 5
ENTER CONTROLLER NUMBER   (O) 0 ? 1
                           ----
ENTER TERMINALS ON THAT CONTROLLER   (D) 4 ? 1,2,3,4
                                         ---------

UNIT 9
ENTER CONTROLLER NUMBER  (O) 2 ?
                          ---
ENTER TERMINALS ON THAT CONTROLLER  (D) 8 ? 1,2,3-4
                                         ---------

NUMBER OF ENTERED TERMINALS= 12

UNIT ** CONTROLLER ** LOGICAL SB ** LUN **
  1              0              1        9
  2              0              2       10
  3              0              3       11
  4              0              4       12
  1              1              6       13
  2              1              7       14
  3              1              8       15
  4              1              9       16
  1              2             11       17
  2              2             12       18
  3              2             13       19
```

SEQ 0010

4 2 14 20

14 1 EXPLANATION OF PRINTOUT

>"RUN DZKCH" CAUSES THE TASK TO BE REQUESTED, WHILE THE DIAGNOSTIC SUPERVISOR CONTROL PROGRAM -- REVISION-A (DCP-A), REQUESTS THE OPERATOR TO TYPE "STA" TO START THE EXERCISER EXAMPLE DCP-A>STA

DESCRIPTION OF THE PRINTOUT IS CONTINUED IN 15 0

15 0 PROGRAM PARAMETER SELECTION

THE PROGRAM AT STARTUP ENTERS A DIALOGUE WITH THE OPERATOR TO DETERMINE WHICH UNITS ON WHAT CONTROLLERS ARE TO BE EXERCISED THE OPERATOR AT THIS POINT IS INTERROGATED WITH

"NUMBER UNITS (D)?"

WHICH MEANS, ENTER THE TOTAL NUMBER OF DEVICES THE OPERATOR WANTS TO EXERCISE (UP TO 63)

THE OPERATOR THEN ENTERS THE TOTAL NUMBER OF TERMINALS (ON THE SERIAL BUS SYSTEM) TO BE EXERCISED IN DECIMAL REPRESENTATION, FOLLOWED BY A CARRIAGE RETURN NO DEFAULT IS PROVIDED

EXAMPLE NUMBER UNITS (D)? 12 <CR>

IN THE ABOVE EXAMPLE , THE OPERATOR WANTS A TOTAL OF TWELVE TERMINALS TO BE EXERCISED

15 1 SELECTED CONTROLLER

THE PROGRAM CAN EXERCISE 8 CONTROLLERS (0 THRU 7), BUT THE LARGEST NUMBER OF CONTROLLERS PRESENTLY PLANNED FOR DPM SYSTEMS IS 4 (0 THRU 3) THE OPERATOR CAN SELECT ANY OF ONE, OR ALL FOUR CONTROLLERS(0 THRU 3), IN ANY SEQUENCE ON THE SERIAL BUS. THE EXERCISER THEN BUILDS UP TO A 63 WORD TABLE FOR ALL SELECTED CONTROLLERS (MAX. OF 63 DEVICES PER CONTROLLER), PLACING EACH IN THE HIGH BYTE OF AN UP TO 63 WORD TABLE ONE TASK CAN EXERCISE UP TO 63 TERMINALS AT ONE TIME. IF A SYSTEM HAS MORE THAN 63 TERMINALS SPREAD OUT OVER MULTIPLE CONTROLLERS, MULTIPLE TASKS MUST BE RUN TO EXERCISE THOSE TERMINALS GREATER THAN 63 HOWEVER, EACH TASK MUST BE INITIATED FROM DIFFERENT CONSOLE TERMINALS HERE THE OPERATOR IS INTERROGATED WITH

```
UNIT 1
"ENTER CONTROLLER NUMBER (0) 0 ?"

OPERATOR'S RESPONSE  IS  TO  TYPE  IN  THE  CONTROLLER  NUMBER  TO  BE
EXERCISED,  OR  USE  THE  DEFAULT  FOLLOWED BY A CARRIAGE RETURN.  THE
DEFAULT IS CONTROLLER NUMBER 0

UNIT 1
EXAMPLE   ENTER CONTROLLER NUMBER (0)?  <CR>
WHERE <CR> MEANS CARRIAGE RET'JRN

IN THE ABOVE EXAMPLE CONTROLLER NUMBER  0  HAS  BEEN  ENTERED  TO  THE
PROGRAM



15 2  SELECTED UNITS


FOR A GIVEN SELECTED CONTROLLER, OR  CONTROLLERS,  TERMINALS  ON  THAT
SPECIFIC CONTROLLER CAN BE ENTERED INDIVIDUALLY, SEQUENTIALLY, OR IN ANY
ORDER, EACH SEPERATED BY  A  COMMA. THE  OPERATOR  IS  ALSO  GIVEN  THE
FLEXIBILITY  OF  ENTERING ALL 63 TERMINALS SIMULTANEOUSLY.  THE SYSTEM
IS THEN MAPPED, ONLINE TERMINALS EXERCISED, OFFLINE TERMINALS REPORTED
AS BEING OFFLINE, AND NON-EXISTENT TERMINALS REPORTED AS NOT SYSGENED
(A BRIEF  DESCRIPTION  OF  SYSGEN  IS  GIVEN  IN  1.1  OF  THE  SYSTEM
GENERATION  MANUAL).    THE  PROGRAM  THEN  EQUATES  THOSE TERMINALS TO
LOGICAL DEVICES, AS SYSGENED ON THE SYSTEM, STORES THEM  IN  THE  SAME
TABLE  AS  THE CONTROLLERS, WITH THE TERMINAL NUMBER IN THE LO-BYTE OF
THE WORD    INTERROGATION HERE CONTINUES WITH:

UNIT 1
"ENTER TERMINALS ON THAT CONTROLLER (D) 1 ?  1-4 "

THE OPERATOR TYPES IN THE TERMINALS TO BE EXERCISED, ON  THE  SELECTED
CONTROLLER  (0)    1-4 MEANS 4 TERMINALS HAVE BEEN ENTERED ALL AT ONCE
ON THE SELECTED CONTROLLER

UNITS KEEPS A SEQUENTIAL COUNT OF THE NUMBER OF DEVICES  ENTERED,  AND
POINTS  TO  THE  NEXT  SEQUENTIAL  DEVICE.    IT  DOES NOT POINT TO THE
PHYSICAL TERMINAL NUMBER.  AT  THIS  POINT  THE  TOTAL  12  UNITS  THE
OPERATOR  WANTED  EXERCISED  HAVE  NOT  BEEN ENTERED, THUS THE PROGRAM
ASSUMES THE REMAINING DEVICES ARE ON  DIFFERENT  CONTROLLERS,  AND  SO
CONTINUES WITH THE DIALOGUE

UNIT 5
ENTER CONTROLLER NUMBER    (0) ) ?  1

THE PROGRAM ASKS WHAT OTHER CONTROLLER IS TO BE UTILIZED, THE OPERATOR
SPECIFIES CONTROLLER NUMBER 1

ENTER TERMINALS ON THAT CONTROLLER    (D) 4 ?  1,2,3,4

HERE THE PROGRAM INTERROGATES FOR TERMINALS TO  BE  ENTERED,  AND  THE
```

OPERATOR ENTERS 4 TERMINALS ON CONTROLLER 1 IN SEQUENTIAL ORDER. NOT HAVING ENTERED THE TOTAL NUMBER OF TERMINALS (12) PREVIOUSLY ASKED FOR TO BE EXERCISED, THE PROGRAM AGAIN CONTINUES TO INTERROGATE THE OPERATOR

```
UNIT 9
ENTER CONTROLLER NUMBER    (0) 1 ? 2
```

THE OPERATOR ENTERS CONTROLLER NUMBER 2 TO THE PROGRAM

```
ENTER TERMINALS ON THAT CONTROLLER    (D) 8 ?  1,2,3-4
```

AFTER THE PROGRAM ASKS FOR THE REMAINING 4 TERMINALS, THE OPERATOR ENTERS THEM IN SEQUENTIAL AND SIMULTANEOUS COMBINATIONS

IN THE ABOVE EXAMPLE· TERMINALS 1,2 ON THE PREVIOUSLY ENTERED CONTROLLER 1 E (CONTROLLER 2) IS ENTERED TO THE PROGRAM TO BE EXERCISED TERMINALS 3 AND 4 IS ALSO ENTERED, BUT AS 3 THROUGH 4, ALLOWING FOR TYPING FLEXIBILITY. HAVING COUNTED THE TOTAL UNITS ENTERED TO BE EXERCISED, IMMEDIATELY THE OPERATOR IS INFORMED OF THE NUMBER OF TERMINALS ENTERED, IN THE FOLLOWING FORMAT

```
EXAMPLE   NUMBER OF ENTERED TERMINALS = 12
WHERE 12 IS THE TOTAL NUMBER OF LEGAL DEVICES ENTERED TO THE PROGRAM

**UNITS** **CONTROLLER** **LOGICAL SB** **LUN**
**U
   XX        YY              ZZ              %
```

WHERE XX IS THE PHYSICAL TERMINAL ENTERED ON CONTROLLER YY, ZZ IS THE LOGICAL SB UNIT NUMBER ASSIGNED TO EACH TERMINAL BY RSX DURING SYSGEN, AND % IS THE LOGICAL UNIT NUMBER THE PROGRAM ASSIGNS TO EACH UNIT XX

15 3 OPERATOR INTERFACE DIALOGUE

OTHER EXAMPLES OF OPERATOR AND PROGRAM DIALOGUE ARE LISTED BELOW

THE OPERATOR CAN TYPE IN ALL 63 TERMINALS SIMULTANEOUSLY AS FOLLOWS

```
>RUN DZKCH
DCP-A> STA
 UNITS (D) ?  63 <CR>
UNIT 1
ENTER CONTROLLER NUMBER    (0) ?  <CR>
```

EXAMPLE· "ENTER TERMINALS ON THAT CONTROLLER (1)?"1-63 <CR>

IN THE ABOVE EXAMPLE, ALL 63 TERMINALS ARE ENTERED INTO A 63 WORD TABLE, AND THE PROGRAM EXERCISES ALL 63 DEVICES IF THEY ARE ALL ON LINE IT IS TO BE NOTED THAT ONLY ONE CONTROLLER NUMBER(0 THRU 7)AND

SEQ 0013

UP TO 63 DEVICES ON ANY ONE CONTROLLER, CAN BE ENTERED AT ANY TIME, WHICH LIMITS THE PROGRAM TO EXERCISE A MAXIMUM OF 63 TERMINALS

ANOTHER EXAMPLE IS GIVEN BELOW:

IF THE OPERATOR HAS 63 UNITS ON THE DPM SYSTEM, BUT 15 DEVICES ARE ON CONTROLLER 0, 15 ON CONTROLLER 1, 15 ON CONTROLLER 2, AND 18 DEVICES ARE ON CONTROLLER 3 (A TOTAL OF 63 DEVICES), AND ALL DEVICES ARE TO BE EXERCISED THEN THE DIALOGUE IS AS FOLLOWS:

```
RUN DZKCH

DCP-A>STA

NUMBER UNITS (D)? 63

ENTER CONTROLLER NUMBER (0)?   0

UNIT 1
ENTER TERMINALS ON THAT CONTROLLER (1)? 1-15
```

AT THIS POINT THE TOTAL 63 UNITS THE OPERATOR WANTED EXERCISED, HAVE NOT BEEN ENTERED, THUS THE PROGRAM ASSUMES THE REMAINING DEVICES ARE ON DIFFERENT CONTROLLERS, AND SO CONTINUES THE DIALOGUE

```
ENTER CONTROLLER NUMBER (0)?   1

UNIT 16
ENTER TERMINALS ON THAT CONTROLLER (1)?. 16-30

ENTER CONTROLLER NUMBER (1)?: 2

UNIT 31
ENTER TERMINALS ON THAT CONTROLLER (1)?:  31-45

ENTER CONTROLLER NUMBER (2)?.  3

UNIT 46
ENTER TERMINALS ON THAT CONTROLLER (1)?·  46-63
```

(WHERE UNITS KEEPS COUNT OF THE NUMBER OF DEVICES ENTERED, AND THE NUMBER IN THE BRACKETS ARE DEFAULTED DEVICE NUMBERS)

THE OPERATOR IS THEN INFORMED OF THE TOTAL NUMBER OF ENTERED TERMINALS AS EXPLAINED BEFORE IN THE FOLLOWING FORMAT·

```
NUMBER OF ENTERED TERMINALS = 63

ETC
```

AND THE PROGRAM GOES ON TO EXERCISE ALL 63 TERMINALS

FOR TERMINALS THAT CANNOT BE ASSIGNED, OR ATTACHED, THE OPERATOR WILL BE INFORMED WITH:

"TERMINAL X CANNOT BE ASSIGNED-$DSW=Y"

"TERMINAL X CANNOT BE ATTACHED-$DSW=Y"

WHERE THE ERROR CODES Y ARE RETURNED BY DIRECTIVES IN THE DIRECTIVE STATUS WORD ($DSW). FOR ADDITIONAL INFORMATION, REFER TO THE RSX-11M EXECUTIVE REFERENCE MANUAL, OR RSX-11M POCKET REFERENCE.

FOR NON-EXISTENT SYSGENED CONTROLLERS AND/OR TERMINALS, THE OPERATOR WILL BE INFORMED WITH:

"CONTROLLER X, TERMINAL Y IS NOT SYSGENED IN"
PROGRAM IS ABORTED

16 0 SWITCH REGISTER FUNCTIONS

NO EXPLICIT SWITCH REGISTER SETTINGS WILL BE RECOGNIZED BY THE PROGRAM, SO AS TO FACILITATE INTEGRATION UNDER RSX11

17 0 ERROR REPORTING

ESSENTIALLY THERE ARE 7 TYPES OF ERRORS SENSED AND REPORTED, 4 OF WHICH ARE DEFINED AS MAJOR EACH OF THE 4 IS PRECEEDED BY A DIAGNOSTIC SUPERVISOR ERROR HEADER IN THE FOLLOWING FORMAT.

DZKCH HRD ERR 00001 TST 001 SUB 000 PC 015346

THIS LINE MEANS THAT PROGRAM DZKCH HAS DETECTED A HARD ERROR, NUMBERED 1, ON TEST NUMBER 1, SUBTEST NUMBER 0 (I E, NO SUBTEST), AND THE PC IS POINTING TO LOCATION 015346

ERRORS RETURNED BY RSX-11M I/O STATUS ERROR CODES ARE GIVEN BELOW PARTIAL MNEMONICS ARE LISTED, THE COMPLETE MNEMONIC IS IE XXX THE OCTAL ERROR NUMBER LISTED IS THE LOW-ORDER BYTE OF THE COMPLETE WORD VALUE (2'S COMPLEMENT OF THE DECIMAL NUMBER)

MNEM	ERROR DEC	NUMBER OCTAL	SIGNIFICANCE
BAD	- 1	377	BAD PARAMETERS
IFC	- 2	376	INVALID FUNCTION CODE
DNR	- 3	375	DEVICE NOT READY
VER	- 4	374	PARITY ERROR ON DEVICE
ONP	- 5	373	HARDWARE OPTION NOT PRESENT
SPC	- 6	372	ILLEGAL USER BUFFER
DNA	- 7	371	DEVICE NOT ATTACHED
DAA	- 8	370	DEVICE ALREADY ATTACHED
DUN	- 9	367	DEVICE NOT ATTACHABLE
EOF	-10	366	END-OF-FILE DETECTED
EOV	-11	365	END-OF-VOLUME DETECTED
WLK	-12	364	WRITE ATTEMPTED TO LOCK UNIT
DAO	-13	363	DATA OVERRUN

SRE	-14	362	SEND/RECEIVE FAILURE
ABO	-15	361	REQUEST TERMINATED
PPI	-16	360	PRIVILEGE VIOLATION
RSU	-17	357	SHAREABLE RESOURCE IN USE
OVR	-18	356	ILLEGAL OVERLAY REQUEST
BYT	-19	355	ODD BYTE COUNT (OR VIRTUAL ADDRESS)
BLK	-20	354	LOGICAL BLOCK NUMBER TOO LARGE
MOD	-21	353	INVALID UDC MODULE NUMBER
CON	-22	352	UDC CONNECT ERROR
NOD	-23	351	SYSTEM DYNAMIC MEMORY EXHAUSTED
DFU	-24	350	DEVICE FULL
IFU	-25	347	INDEX FILE FULL
NSF	-26	346	NO SUCH FILE
LCK	-27	345	LOCKED FROM READ/WRITE ACCESS
HFU	-28	344	FILE HEADER FULL
WAC	-29	343	ACCESSED FOR WRITE
CKS	-30	342	FILE HEADER CHECKSUM FAILURE
WAT	-31	341	ATTRIBUTE CONTROL LIST FORMAT ERROR
RER	-32	340	FILE PROCESSOR DEVICE READ ERROR
WER	-33	337	FILE PROCESSOR DEVICE WRITE ERROR
ALN	-34	336	FILE ALREADY ACCESSED ON LUN
SNC	-35	335	FILE ID, FILE NUMBER CHECK
SQC	-36	334	FILE ID, SEQUENCE NUMBER CHECK
NLN	-37	333	NO FILE ACCESSED ON LUN
CLO	-38	332	FILE WAS NOT PROPERLY CLOSED
NBF	-39	331	NO BUFFER SPACE AVAILABLE FOR FILE
RBG	-40	330	ILLEGAL RECORD SIZE
NBK	-41	327	FILE EXCEEDS SPACE ALLOCATED, NO BLOCKS
ILL	-42	326	ILLEGAL OPERATION ON FILE DESCRIPTOR BLOCK
BTP	-43	325	BAD RECORD TYPE
RAC	-44	324	ILLEGAL RECORD ACCESS BITS SET
RAT	-45	323	ILLEGAL RECORD ATTRIBUTES BITS SET
RCN	-46	322	ILLEGAL RECORD NUMBER - TOO LARGE
	-47		(NOT USED)
2DV	-48	320	RENAME - 2 DIFFERENT DEVICES
FEX	-49	317	RENAME - A NEW FILE NAME ALREADY IN USE
BDR	-50	316	BAD DIRECTORY FILE
RNM	-51	315	CANNOT RENAME OLD FILE SYSTEM
BDI	-52	314	BAD DIRECTORY SYNTAX
FOP	-53	313	FILE ALREADY OPEN
BNM	-54	312	BAD FILE NAME
BDV	-55	311	BAD DEVICE NAME
BBE	-56	310	BAD BLOCK ON DEVICE
DUP	-57	307	ENTER - DUPLICATE ENTRY IN DIRECTORY
STK	-58	306	NOT ENOUGH STACK SPACE (FCS OR FCP)
FHE	-59	305	FATAL HARDWARE ERROR ON DEVICE
NFI	-60	304	FILE ID WAS NOT SPECIFIED
ISQ	-61	303	ILLEGAL SEQUENTIAL OPERATION
EOT	-62	302	END-OF-TAPE DETECTED

BVR	-63	301	BAD VERSION NUMBER
BHD	-64	300	BAD FILE HEADER
OFL	-65	277	DEVICE OFFLINE
BCC	-66	276	BLOCK CHECK OR CRC ERROR
	-67		(NOT USED)
NNN	-68	274	NO SUCH NODE
NFW	-69	273	PATH LOST TO PARTNER
BLB	-70	272	BAD LOGICAL BUFFER
TMM	-71	271	TOO MANY OUTSTANDING MESSAGES
NDR	-72	270	NO DYNAMIC SPACE AVAILABLE
CNR	-73	267	CONNECTION REJECTED
TMO	-74	266	TIME OUT ON REQUEST
EXP	-75	265	FILE EXPIRATION DATE NOT REACHED
BTF	-76	264	BAD TAPE FORMAT
NNC	-77	263	NOT ANSI "D" FORMAT BYTE COUNT
NNL	-78	262	NOT A NETWORK LUN
NLK	-79	261	TASK NOT LINKED TO SPECIFIED ICS/ICR INTERRUPTS
NST	-80	260	SPECIFIED TASK NOT INSTALLED
FLN	-81	257	DEVICE OFFLINE WHEN OFFLINE PEQUEST WAS ISSUED
IES	-82	256	INVALID ESCAPE SEQUENCE
PES	-83	255	PARTIAL ESCAPE SEQUENCE
ALC	-84	254	ALLOCATION FAILURE
ULK	-85	253	UNLOCK ERROR

FOR ADDITIONAL INFORMATION REFER TO THE IAS/RSX-11M
I/O OPERATIONS REFERENCE MANUAL

17 1 MAJOR ERRORS

(1) NO RESPONSE FROM A TERMINAL - THIS ERROR RESULTS IN THE FOLLOWING MESSAGE

"TIME-OUT ERROR. TERMINAL X IS OFFLINE"
"ERROR-TYPE=8,SB=1-63,RSX-11M STATUS CODE=277,BYTE COUNT=0-128

THE FOLLOWING EXPLAINS THE VARIOUS PARAMETERS IN THE ERROR STATEMENTS

"ERROR-TYPE":
THE PROGRAM ASSIGN'S A NUMBER TO EVERY ERROR SENSED, AND ARE DEFINED AS FOLLOWS

A) ERROR-TYPE=1 ,RSX-11M DETECTED ERROR

B) ERROR-TYPE=2 ;RETRY ERROR.

C) ERROR-TYPE=3 ,RSX-11M DIRECTIVE STATUS WORD ERROR

D) ERROR-TYPE=4 ,DATA COMPARISON ERROR.

```
E) ERROR-TYPE=5          ,ERROR CODE FOR AN ONLINE TERMINAL

F) ERROR-TYPE=6          ,CRC ERROR CODE

G) ERROR-TYPE=8          ,TIME-OUT TERMINAL OFFLINE ERROR

'SB"
THESE ARE THE SERIAL BUS UNIT NUMBERS,WHICH ARE FROM 1 TO 63

"RSX-11M STATUS CODE"
APPENDIX I OF THE I/O OPERATIONS MANUAL  LISTS  THE  I/O  ERROR  CODES
RETURNED BY THE SYSTEM   THEY CAN ALSO BE FOUND IN SECTION 17 0 ABOVE

"BYTE-COUNT"
THIS KEEPS THE NUMBER OF BYTES TRANSFERRED WHEN THE ERROR OCCURED

FOR ANY  TERMINAL  GOING  OFFLINE,  AND  UPON  RETURNING  ONLINE,  THE
OPERATOR WILL BE INFORMED WITH THE MESSAGE

'TERMINAL X IS BACK ON LINE "

(2)     OPERATING SYSTEM ERRORS - FOR ERRORS RELATING TO THE OPERATING
SYSTEM,  THE OPERATOR WILL BE INFORMED WITH THE MESSAGE

"RSX-11M HARD ERROR DETECTED "
"ERROR-TYPE=1,SB=1-63,RSX-11M STATUS CODE=0-377,BYTE COUNT=0-128"

(3)     BAD DATA IN RECEIVED DATA MESSAGE - THIS ERROR WILL RESULT  IN
THE MESSAGE

"DATA COMPARISON ERROR "
"ERROR-TYPE=4,SB=1-63,BYTE COUNT RECEIVED=0-128"

        BYTE-NUMBER      GD DATA      BD DATA

           WW                YY           ZZ

           WW                YY           ZZ

WHERE WW IS THE BAD DATA POSITION  IN  THE  DATA  STREAM,  YY  IS  THE
TRANSMITTED DATA,  AND ZZ IS THE RECEIVED DATA

(4)     TRANSMISSION FAILURE IN TRANSMIT/RECEIVE MESSAGE - THIS  ERROR
WILL RESULT IN THE MESSAGE

"RETRY ERROR."
"ERROR-TYPE=2,SB=1-63,BYTE COUNT=0-128
```

17 2 MINOR ERRORS

EPPOR-TYPE 3, ERROR-TYPE 5 ARE CONSIDEPED MiNOR ERRORS

18 0 TEST DESCRIPTION

GENERAL AFTER INITIAL DIALOGUE IS PERFORMED AND A TABLE OF ACTIVE F D C TERMINALS CONSTRUCTED THE BASIC SEQUENCE OF THE EXERCISER IS AS FOLLOWS

SELECT A DATA PATTERN
SELECT A MESSAGE LENGTH
TRANSMIT DATA MESSAGE TO ALL TERMINALS IN THE TABLE
ON POLL CYCLE N
RECEIVE DATA MESSAGES FROM ALL SELECTED
TERMINALS IN POLL CYCLE N+1.
REPORT ANY ERRORS DETECTED IN THE FORMATS INDICATED
IN SECTION 6 0
UPDATE DATA PATTERN IN OUTPUT BUFFER
UPDATE MESSAGE LENGTH
ETC ETC

18 1 DATA PATTERNS

THERE ARE 7 DIFFERENT DATA PATTERNS AND THEY ARE UTILIZED AS FOLLOWS

ALL ONES-	THIS DATA PATTERN CHECKS THE ABILITY OF THE SERIAL BUS SYSTEM TO IMPLEMENT ZERO STUFFING AND ALSO CHECK BIT DROPOUT
ALL ZEROS -	THIS DATA PATTERN IS USED TO DETECT ANY BIT PICKUP TENDENCIES
ONES ZEROS -	THIS COMBINATION OF ONES AND ZEROS DATA PATTERN IS USED TO GENERALLY TEST THE SYSTEM RECOVERY ABILITIES.
FLOATING ZERO -	THIS DATA PATTERNS (FLOATING TWO ZEROS FROM LOWEST SIGNIFICANT POSITIONS TO HIGHEST POSITIONS) IS UTILIZED TO DETECT BYTE BOUNDARY FAILURES DUE TO LINE NOISE AND HARDWARE FAILURES

16 2 MESSAGE LENGTHS

EACH DATA PATTERN IS TRANSMITTED, ON ALTERNATE POLL CYCLES (TRANSMIT, RECEIVE, TRANSMIT, ETC), IN MESSAGES OF DECLINING MESSAGE LENGTHS. ALL DATA PATTERNS BEGINS WITH A LENGTH OF 128 BYTES, THEN 64, 32, ETC, ETC, UNTIL A MESSAGE OF 2 DATA BYTES IS TRANSMITTED THEN ANOTHER PATTERN IS SELECTED AND THE MESSAGE COUNT AGAIN REVERTS TO 128 AT PRESENT TIME ALL DEVICES HANDLES THE MAXIMUM MESSAGE LENGTHS OF 128 BYTES FOR FUTURE EQUIPMENTS MESSAGE LENGTHS MIGHT HAVE TO BE DETERMINED BY POLLING ALL DEVICES TO DETERMINE TYPES AND SETTING THE MAXIMUM MESSAGE LENGTHS EQUAL TO THE SMALLEST BUFFER SIZE

18 3 END OF PASS REPORT

AN END OF PASS REPORT IS GIVEN EVERY MINUTE IN THE FOLLOWING FORMAT

ACTIVE PASSES SINCE LAST REPORT=X, ACTIVE TERMINAL=Y, TIME=00 00

UPON RECEIVING THE DCP-A> PROMPT, THIS PRINTOUT CAN BE INHIBITED BY TYPING IN THE FOLLOWING

DCP-A>STA/FLA IXE

19 0 PROGRAM TERMINATION

THIS PROGRAM WILL CYCLE UNTIL A CONTROL C IS TYPED UPON THE CONSOLE WHICH INITIATED THE EXECRCISOR AT THAT TIME THE PROGRAM WILL ABORT AND EXIT TO THE MONITOR

EXAMPLE CONTROL C

MCR>

DZ1CH MACRO M11 10-OCT-77 10 16
TABLE OF CONTENTS

SEQ 0020

```
                                        TITLE   DZKCH
                                        ENABL   AMA
                                        NLIST   MD,ME
                                        .LIST   MEB
                                        MCALL   SVC
                                        MCALL   ASTX$S,QIO$,ALUN$,DIR$,GLUN$,EXIT$S
                                        MCALL   MRKT$S,CMKT$S,GTIM$S
                                        MCALL   WTSE$S
                                        GLOBL   STADR
000000                          SVC
        000001                  SVCINS= 1                       ;LIST GENERATED SYMBOLS
        000001                  SVCTST= 1                       ;LIST TEST NUMBERS
        000001                  SVCSUB= 1                       ;LIST SUBTEST NUMBERS
        000001                  SVCGBL= 1                       ;L ST BLOBALS
        000001                  SVCTAG= 1                       ;LIST GENERATED TAGS
```

SEQ 0022

```
                                        SBTTL  DIAGNOSTIIC SUPERVISOR HEADER
                                ;
                                ;       ******************************************
                                ;       THIS SECTION CONTAINS GENERAL INFO WHICH
                                ;       DESCRIBES THE MAJOR CHARACTERISTICS OF
                                ;       THE DIAGNOSTIC PROGRAM
                                ;       ******************************************
000000                          POINTER NONE

000000                  STADR   HEADER DZKCH,A,0,63,300,1
000000                  LSNAME:
000000  104                             .ASCII  @D@
000001  132                             ASCII   @Z@
000002  113                             ASCII   @K@
000003  103                             ASCII   @C@
000004  110                             ASCII   @H@
000005  000                             BYTE    0
000006  000                             BYTE    0
000007  000                             BYTE    0
000010                  LSREV:
000010  101                             ASCII   @A@
000011                  LSDEPO
000011  060                             ASCII   @0@
000012                  LSMREV:
000012  001                             .BYTE   CSREVISION
000013  004                             BYTE    CSEDIT
000014                  LSUNIT
000014  000000                          WORD    0
000016                  LSTIM1
000016  000063                          WORD    63
000020                  LSTIMU
000020  000300                          WORD    300
000022                  LSTIML
000022  000001                          WORD    1
000024                  LSEF
000024  000000                          WORD    0
000026  000000                          WORD    0
000030                  LSSPC
000030  000000                          WORD    0
000032                  LSEXP1
000032  000000                          WORD    0
000034                  LSEXP2
000034  000000                          WORD    0
000036                  LSEXP3
000036  000000                          WORD    0
000040                  LSDTP
000040  007326'                         WORD    LSDISPATCH
000042                  LSICP
000042  007330'                         WORD    LSINIT
000044                  LSCCP
000044  013262'                         WORD    LSCLEAN
000046                  LSHPCP
000046  007266'                         WORD    LSHARD
000050                  LSSPCP
000050  000000                          WORD    0
000052                  LSDEVP
000052  000210'                         WORD    LSDVTYP
```

DIAGNOSTIC SUPERVISOR HEADER

SEQ 0023

```
000054                          L$REPP
000054  000000                          WORD    0
000056                          L$HPTP
000056  007314'                          WORD    L$HW
000060                          L$SPTP
000060  000000                          WORD    0
000062                          L$DRCT
000062  000202'                          WORD    L$DR
000064                          L$DRS
000064  000206'                          WORD    L$DRST
000066                          L$STA .
000066  000000                          WORD    0
000070                          L$AUT
000070  000000                          WORD    0
000072                          L$DUT
000072  000000                          WORD    0
000074                          L$PWRU
000074  000000                          WORD    0
000076                          L$LADP
000076  015444'                          WORD    L$LAST
```

```
                                        SBTTL  EQUATES
                                ;
                                ;       ***************************************************
                                ;       THIS FACILITATES PROGRAM ASSEMBLY BY EQUATING
                                ;       SPECIFIC SYMBOLS WITH SPECIFIC VALUES VIA DIRECT
                                ;       ASSIGNMENT STATEMENTS.
                                ;       ***************************************************

000003                           DATA   =3                      ;BYTE POSITION IN OUTPUT BUFFER
000011                          LUN     =11                     ;DEFINE LOGICAL UNIT NUMBER FROM 11

                                ;OFFSETS IN DEVICE PARAMETER BLOCK (DCB)

000002                          IO FUN= 2
000004                          IO. LUN= 4
000006                          IO EFN= 6
000010                          IO IST= 10
000012                          IO AST= 12
000014                          IO A1=  14
000016                          IO A2=  16
000020                          IO A3=  20
000022                          IO A4=  22
000024                          IO A5=  24

                                ; RSX I/O FUNCTION CODES

002400                          IO WPC= 2400                    ;WRITE PERIPHERAL BLOCK
003400                          IO RXP= 3400                    ;WRITE TRANSPARENT WITH LOOP BACK
```

EQUATES

```
    72          000001          BIT00=  1
    73          000002          BIT01=  2
    74          000004          BIT02=  4
    75          000010          BIT03=  10
    76          000020          BIT04=  20
    77          000040          BIT05=  40
    78          000100          BIT06=  100
    79          000200          BIT07=  200
    80          000400          BIT08=  400
    81          001000          BIT09=  1000
    82          002000          BIT10=  2000
    83          004000          BIT11=  4000
    84          010000          BIT12=  10000
    85          020000          BIT13=  20000
    86          040000          BIT14=  40000
    87          100000          BIT15=  100000
```

```
     89                                          .SBTTL  P-TABLE MESSAGES
     90                                  ;
     91                                  ;       ************************************************
     92                                  ;       ++NOTE++ P-TABLE MESSAGES ARE USED DURING
     93                                  ;       DIALOGUE WITH THE OPERATOR.
     94                                  ;       ************************************************
     95
     96
     97
     98 000100      105     116     124  GETCNT: .ASCIZ/ENTER CONTROLLER NUMBER  /
        000103      105     122     040
        000106      103     117     116
        000111      124     122     117
        000114      114     114     105
        000117      122     040     116
        000122      125     115     102
        000125      105     122     040
        000130      072     000
     99 000132      105     116     124  GETERM:  ASCIZ/ENTER TERMINALS ON THAT CONTROLLER  /
        000135      105     122     040
        000140      124     105     122
        000143      115     111     116
        000146      101     114     123
        000151      040     117     116
        000154      040     124     110
        000157      101     124     040
        000162      103     117     116
        000165      124     122     117
        000170      114     114     105
        000173      122     040     072
        000176      000
    100
    101                                         .EVEN
```

```
                                        SBTTL  GENERAL AND DPB STORAGE LOCATIONS
                                ;
                                ;       ***********************************************
                                ;       THIS SECTION OF THE SOFTWARE CONTAINS ALL
                                ;       DIRECTIVE PARAMETER BLOCKS AS USED BY RSX-11M
                                ;       ALSO CONTAINS PERMANENT/TEMPORARY STORAGE
                                ;       ***********************************************
                                ;
                                ;

000200                          DEVREG  1,1
000200  000001                                 WORD    1
000202                          L$DR
000202  000001                                 WORD    1
000204  000001                                 WORD    1
000206                          L$DRST
                                        BLKW    1
                                ;       ***********************************************
                                ;       THE DEVREG CALL ASSEMBLES A PAIR OF TABLES
                                ;       WHICH ARE USED TO DEFINE THOSE DEVICE REGISTERS
                                ;       THAT WILL BE USED BY THE PROGRAM
                                ;       ***********************************************
000210                          DEVTYPE SB
000210                          L$DVTYP
000210      123     102     000                 .ASCIZ  @SB@
                                        .EVEN
                                ;
                                ;       ***********************************************
                                ;       THE DEVTYP CALL SPECIFIES THE EQUIPMENT THAT
                                ;       THE EXERCISER PROGRAM WILL SERVE.
                                ;       ***********************************************
                                ;
                                ;

                                ;       ***********************************************
                                ;       THE QUED I/O DIRECTIVE PARAMETER BLOCK IS SET
                                ;       UP AS FOLLOWS .... .
                                ;       ***********************************************

000214                          Q1ODPB  Q1O$    IO.FUN,LUN,,,IOST,AST,<SCBADR,SCBCNT,40,PCBADR,PCBCNT>
000214      001     014                 .BYTE   1,12.
000216  000002                          WORD    IO FUN
000220  000011                          WORD    LUN
000222      000     000                 BYTE    ,0
000224  000276'                         WORD    IOST
000226  013362'                         WORD    AST
000230  001006'                         WORD    SCBADR
000232  001010'                         WORD    SCBCNT
000234  000040                          WORD    40
000236  001012'                         WORD    PCBADR
000240  001014'                         .WORD   PCBCNT
000242  000000                          .WORD   0

                                ;
                                ;       ***********************************************
                                ;       THE ASSISN LOGICAL UNIT PARAMETER BLOCK IS
                                ;       DEFINED AS FOLLOWS .. . . .
                                ;       ***********************************************
```

```
DZKCH   MACRO M11    10-OCT-77 10 16  PAGE 6-1                                                    SEQ 0028
GENERAL AND DPB STORAGE LOCATIONS

    139 000244                          ALUDPB  ALUN$   LUN,SB,0            ,DIRECTIVE PARAMETER BLOCK FOR ALUN
        000244      007     004                  BYTE    7,4
        000246  000011                          WORD    LUN
        000250      123     102                  ASCII   /SB/
        000252  000000                          WORD    0
    140                                 ,
    141                                 ,
    142                                 ,       ********************************************
    143                                 ;       THE GET LUN PARAMETER BLOCK IS DEFINED AS
    144                                 ,       FOLLOWS. . . .
    145                                 ,       ********************************************
    146
    147                                 ,
    148                                 ,
    149                                 ,
    150 000254                          GLUDPB. GLUN$   11,ERRBRT           ,PARAMETER BLOCK FOR GLUN
        000254      005     003                  BYTE    5,3
        000256  000011                          .WORD   11
        000260  000262'                          WORD    ERRBRT
    151                                 .
    152                                 ,
    153                                 ,       ********************************************
    154                                 ,             STORAGE LOCATIONS
    155                                 ,       ********************************************
    156                                 ,
    157                                 ,
    158 000262                          ERRBRT  .BLKW   6                   ;GLUN    BROADCAST ERROR BUFFER
    159 000276                          IOST     BLKW   2                   ;WORDS FOR STATUS RETURN CODES
    160 000302  000000                  DEVPTR: 0                           ;POINTER TO NEXT TERMINAL,LUN
    161 000304  000000                  NUMDEV  0                           ,NUMBER OF ACTIVE TERMINALS
    162 000306                          UNTBL   .BLKW   64                  ;RUNTIME DEVICE UNIT TABLES
    163 000506                          DEVTAB:  BLKW   64                  ;ACTIVE LUN,,DEV TABLE
    164 000706                          OFFLIN  .BLKB   64                  ,OFFLINE TABLE
    165
    166
    167
    168                                 ,       RSX-11M DEVICE DEPENDENT PARAMETERS
    169
    170
    171 001006  000000                  SCBADR   WORD   0
    172 001010  000000                  SCBCNT   WORD   0
    173 001012  000000                  PCBADR   WORD   0
    174 001014  000000                  PCBCNT   WORD   0
    175
    176
    177                                 ,       STORAGE FOR OUTPUT BUFFERS
    178                                 ,
    179 001016  001041'                 OBUFF   .WORD   OBUF0
    180 001020  001245'                 .WORD   OBUF1
    181 001022  001451'                  WORD   OBUF2
    182 001024  001655'                 .WORD   OBUF3
    183 001026  002061'                  WORD   OBUF4
    184 001030  002265'                  WORD   OBUF5
    185 001032  002471'                  WORD   OBUF6
    186 001034  002675'                  WORD   OBUF7
    187 001036  000000                   WORD   0                                ,END OF BUFFER TABLE
    188
```

```
                                        ODD
001041                                  OBUF0   BLKB    132
001245                                  OBUF1   BLKB    132
001451                                  OBUF2   BLKB    132
001655                                  OBUF3   BLKB    132
002061                                  OBUF4   BLKB    132.
002265                                  OBUF5   BLKB    132
002471                                  OBUF6   BLKB    132
002675                                  OBUF7   BLKB    132.
                                        EVEN

                                ;       STORAGE FOR INPUT BUFFERS

003102  003122'                 INBUFF  WORD    IBUF0
003104  003322'                         WORD    IBUF1
003106  003522'                         WORD    IBUF2
003110  003722'                         WORD    IBUF3
003112  004122'                         WORD    IBUF4
003114  004322'                         WORD    IBUF5
003116  004522'                         WORD    IBUF6
003120  004722'                         WORD    IBUF7

003122                          IBUF0   BLKB    128
003322                          IBUF1:  BLKB    128
003522                          IBUF2.  BLKB    128
003722                          IBUF3:  BLKB    128
004122                          IBUF4.  BLKB    128
004322                          IBUF5   BLKB    128
004522                          IBUF6   BLKB    128
004722                          IBUF7   BLKB    128


                                ;       STORAGE FOR I/O STATUS BLOCK

005122  005142'                 IOSB    WORD    ISTAT0
005124  005146'                         WORD    ISTAT1
005126  005152'                         WORD    ISTAT2
005130  005156'                         WORD    ISTAT3
005132  005162'                         WORD    ISTAT4
005134  005166'                         WORD    ISTAT5
005136  005172'                         WORD    ISTAT6
005140  005176'                         WORD    ISTAT7

005142                          ISTAT0: BLKW    2
005146                          ISTAT1. BLKW    2
005152                          ISTAT2. BLKW    2
005156                          ISTAT3. BLKW    2
005162                          ISTAT4  BLKW    2
005166                          ISTAT5  BLKW    2
005172                          ISTAT6  BLKW    2
005176                          ISTAT7  BLKW    2

005202  000000                  ISB     WORD    0       ;TEMP LOCATION USED IN AST
005204  000000                  IODSW   WORD    0       ;TEMPORARY LOCATION FOR DSW ERROR
005206                          BUF     BLKW    10      ;TIME PARAMETER BUFFER
```

```
    246 005226  377                 PATTRN  BYTE  377
    247 005227  000                         BYTE  000
    248 005230  252                         BYTE  252
    249 005231  077                         BYTE  77
    250 005232  317                         BYTE  317
    251 005233  363                         BYTE  363
    252 005234  374                         BYTE  374
    253 005235  001                         BYTE  1          ;DATA PATTERN TERMINATOR
    254
    255
    256 005236  000000              PASFG   WORD  0          ;PROGRAM PASS INDICATOR
    257 005240  000000              LENGTH  WORD  0          ;MESSAGE LENGTH LOCATION
    258 005242  000000              BFFPT   WORD  0          ;POINTER TO CURRENT BUFFER PA R
    259 005244  000000              LUNS    WORD  0          ;LUN # TEMP LOCATION
    260 005246  000000              PASS    WORD  0          ;PROGRAM PASS INDICATOR
    261 005250  000000              UNITS   WORD  0          ;NUMBER OF ACTIVE TERMINALS
    262 005252                      ERRTBL  BLKB  1000       ;ERROR TABLE LOCATIONS
    263 007222  000000              TERM    WORD  0          ;TERMINAL NUMBER LOCATION
    264 007224  000000              ERRPNT  WORD  0          ;ERROR POINTER OFFSET
    265 007226  000000              PATPT   WORD  0          ;PATTERN POINTER
    266 007230  000000              TEM3    WORD  0          ;TEMPORARY LOCATION IN CRC MODULE
    267 007232  000000              ERRTMP  WORD  0          ;THE FOLLOWING 12 LOCATIONS ARE TEMPORARY
    268 007234  000                 TEMP    BYTE  0
    269 007235  000                 TMTP    BYTE  0
    270 007236  000                 ETB1    BYTE  0
    271 007237  000                 ETB2    BYTE  0          ;
    272 007240  000000              ETB3    WORD  0          ;
    273 007242  000000              ETB4    WORD  0          ;
    274 007244  000                 RTRY1   BYTE  0
    275 007245  000                 RTRY2   BYTE  0
    276 007246  000000              RTRY3   WORD  0
    277 007250  000000              DSTW1   WORD  0
    278 007252  000000              DSTW2   WORD  0
    279 007254  000000              DSTW3   WORD  0
    280 007256  000000              SB      WORD  0          ;LOCATION FOR LOGICAL SB DEVICES
    281 007260  000000              TSTBUF  WORD  0          ;OUTPUT BUFFER POINTER
    282 007262  000000              CRCFLG  WORD  0          ;CRC COMPLETE FLAG
    283                              EVEN
```

```
                        SBTTL  HARDWARE P-TABLE
                ;
                ;
                ;       ***********************************************************
                ;       P-TABLES IN GENERAL ARE USED TO PROVIDE CERTAIN
                ;       HARDWARE PARAMETERS I.E  VECTOR ADDRESSES,PRIORITY
                ;       LEVELS, E T C THE GET PARAMETER DATA (GPRMD) CALL
                ;       IS USED HERE FOR HARDWARE PARAMETER CODING
                ;       ***********************************************************
                ;
                ;
                ;

007264                  BGNHRD
007264  000012                        WORD L10000-L$HARD/2
007266                  L$HARD
007266                  GPRMD   GETCNT,0,0,177400,0,7,YES
007266  000032                        WORD    T$CODE
007270  000100'                       WORD    GETCNT
007272  177400                        WORD    177400
007274  000000                        WORD    T$LOLIM
007276  000007                        WORD    T$HILIM
007300                  GPRMD   GETERM,0,0,377,1,63.,YES
007300  000052                        WORD    T$CODE
007302  000132'                       WORD    GETERM
007304  000377                        WORD    377
007306  000001                        WORD    T$LOLIM
007310  000077                        WORD    T$HILIM

007312                  ENDHRD
                                      EVEN
007312                  L10000
                                      EVEN
```

```
                                        SBTTL  DEFAULT HARDWARE P-TABLE
                                ;
                                ;
                                ;       ****************************************
                                ;       PARAMETERS HERE ARE USED AS DEFAULT IN
                                ;       OPERATOR DIALOGUE
                                ;       ****************************************
                                ;
                                ;
007312                          BGNHW
007312  000001                                  WORD    L10001-L$HW/2
007314                          L$HW
007314  000001                          WORD    1               ;CONTROLLER/TERMINAL DEFAULT
007316                          ENDHW
007316                          L10001
```

```
                                        SBTTL  SOFTWARE P-TABLE
                                ;
                                ;
                                ;       ****************************************
                                ;       ANOTHER SEPERATE AND OPTIONAL PARAMETER
                                ;       TABLE ASSEMBLED WITH THE DIAGNOSTIC
                                ;       PROGRAM
                                ;       ****************************************
007316                          BGNSFT
007316  000000                                    WORD L10002-L$SOFT/2
007320                          L$SOFT
007320                          ENDSFT
                                                  EVEN
007320                          L10002

                                ;BUILD SOFTWARE P-TABLE

007320                          BGNSW
007320  000000                                    WORD   L10003-L$SW/2
007322                          L$SW

007322                          ENDSW
007322                          L10003

007322                          BGNRPT
007322                          L$RPT

007322                          ENDRPT
007322                          L10004
007322  104025                                  EMT     C$RPT
```

SEQ 0034

```
                                        SBTTL  DISPATCH TABLE
                                ;
                                ;
                                ;       ************************************************
                                ;       THIS CALL IS USED AS A DIRECTIVE FOR THE ASSEMBLY
                                ;       OF A DISPATCH TABLE WHICH WILL CONTAIN THE SYMBOLIC
                                ;       ADDRESS OF EACH TEST CONTAINED IN THE DIAGNOSTIC
                                ;       PROGRAM . . .
                                ;       ************************************************
                                ;
                                ;
 007324                         DISPATCH        1
 007324  000001                                        WORD    1
 007326                         L$DISPATCH
 007326  011370'                                        WORD    T1
```

```
                                .TITLE  DZKCH
                                .IDENT  /J 3/
                                .SBTTL  INIT CODE
                                ;
                                ;
                                ;       ********************************************************************
                                ;       THIS INITIALIZATION SECTION OF THE PROGRAM PROVIDES
                                ;       OPERATOR INTERFACE. AT ENTRY A TABLE HOLDS A MAXIMUM
                                ;       OF 3 CONTROLLERS AND 63 TERMINALS PER CONTROLLER
                                ;       THE SYSTEM GENERATION IS CHECKED SO THAT A TABLE OF
                                ;       LOGICAL UNITS IS BUILT. THIS IS THEN EQUATED TO PHYSICAL
                                ;       DEVICES ON A SPECIFIED CONTROLLER. HAVING DONE THE ABOVE
                                ;       THE ROUTINE ASSIGNS LOGICAL UNIT NUMBERS (LUNS) TO
                                ;       LOGICAL DEVICES (SB'S) ..
                                ;       --
                                ;       RESOURCES USED FROM RSX-11M ARE THE DIRECTIVES,
                                ;       QIO$, DIR$, ALUN$, GLUN$, EXIT$S
                                ;       RESOURCES USED FROM THE DIAGNOSTIC SUPERVISER ARE,
                                ;       GPHARD, PRINT
                                ;       --
                                ;       GPHARD-- THE GET PARAMETER HARD CALL, REQUEST A POINTER
                                ;       TO THE TABLE, WHICH HOLDS THE CONTROLLERS AND TERMINALS
                                ;       ON THAT CONTROLLER
                                ;       --
                                ;       PRINT-- OUTPUTS MESSAGES, AND  ERRORS TO THE TELETYPE E T C
                                ;
                                ;       AT EXIT THE ROUTINE PRINTS OUT , THE TOTAL NUMBER OF
                                ;       ACTIVE TERMINALS ON THE BUS, A PHYSICAL DEVICE ON A
                                ;       CONTROLLER, A CORRESPONDING LOGICAL DEVICE, AND AN
                                ;       ASSOCIATED LOGICAL UNIT NUMBER
                                ;       ++
                                ;       --
                                ;       ********************************************************************
                                ;
                                ;
                                ;
007330                          BGNINIT
007330                          LSINIT
                                
                                        CMKT$S                          ;CANCEL ANY OUTSTANDING MARK TIME REQUESTS
007330  012746                          MOV     (PC)+,-(SP)
007332     033     001                  .BYTE   27.,1
007334  104377                          EMT      0<377>
007336  005037  005236'                CLR     PASFG
007342  005037  005246'                CLR     PASS
                                ;       ********************************************************************
                                ;       CLEAR STORAGE LOCATIONS FOR SUBSEQUENT INITIALIZATION CODE
                                ;       ********************************************************************
007346  012701  000262'                MOV     #ERRBRT,R1
007352  005021                  139$    CLR     (R1)+
007354  020127  001006'                CMP     R1,#SCBADR
007360  001374                          BNE     139$
                                ;       ********************************************************************
007362  005037  000226'                CLR     Q10DPB+10.AST
007366  012737  000000G 000216'        MOV     #IO.DET,Q10DPB+10 FUN
007374  012705  000011                MOV     #LUN,R5
007400  010537  000220'          30$   MOV     R5,Q10DPB+10 LUN
007404                                  DIR$    #Q10DPB
007404  012746  000214'                MOV     #Q10DPB,-(SP)
```

```
     007410  104377                          EMT     0<377>
 412 007412  005205                          INC     R5
 413 007414  020527  000113                  CMP     R5,#75
 414 007420  003767                          BLE     30$
 415
 416 007422  012705  000306'                 MOV     #UNTBL,R5               ;64 ONE-WORD UNIT TABLE
 417 007426  013703  000014'                 MOV     L$UNIT,R3               ;NUMBER OF UNITS FROM HEADER
 418 007432  010337  005250'                 MOV     R3,UNITS
 419 007436  005004                          CLR     R4                      ;INIT POINTER TO P-TABLE
 420 007440                          1$      GPHARD  R4,R0                   ;GET POINTER TO HARDWARE P-TABLE
     007440  010400                                      MOV     R4,R0
     007442  104055                                      EMT     C$GPHRD
 421 007444  011025                          MOV     (R0),(R5)+              ;GET TERM & CONTROLLER # FROM HARDWAPE
 422                                                                         ;P-TABLE ,PUT IN UNIT TABLE
 423 007446  005204                          INC     R4                      ;NEXT TABLE
 424 007450  005303                          DEC     R3                      ;LAST P-TABLE?
 425 007452  001372                          BNE     1$                      ;BRANCH IF NO---
 426 007454  012715  177777                  MOV     #-1,(R5)                ;ELSE INSERT END OF TABLE FLAG
 427 007460  005000                          CLR     R0                      ;OFFSETS TO TEMPORARY TABLES IN IBUF0
 428 007462  005003                          CLR     R3                      ;SB # INDICATOR
 429 007464  012702  177777                  MOV     #-1,R2                  ;CONTROLLER # INDICATOR FIRST TIME THRU
 430 007470  005004                          CLR     R4                      ;TERMINAL # INDICATER
 431 007472  012737  000011  000246' 2$      MOV     #LUN,ALUDPB+2           ;SET THE LUN =11
 432 007500  010337  000252'                 MOV     R3,ALUDPB+6             ;SET THE DEVICES
 433 007504                                  DIR$    #ALUDPB                 ;ISSUE I/O REQUEST
     007504  012746  000244'                 MOV     #ALUDPB,-(SP)
     007510  104377                          EMT     0<377>
 434 007512  103444                          BCS     4$                      ;CHECK IF REQUEST IS REJECTED
 435 007514  012737  000011  000256'         MOV     #LUN,GLUDPB+2           ;ELSE SET THE LUN FOR THE G- DIRECTIVE
 436 007522  012737  000262' 000260'         MOV     #ERRBRT,GLUDPB+4        ;GET THE LUN INFO BUFFER ADDRESS
 437 007530                                  DIR$    #GLUDPB                 ;ISSUE I/O DIRECTIVE
     007530  012746  000254'                 MOV     #GLUDPB,-(SP)
     007534  104377                          EMT     0<377>
 438 007536  005737  000266'                 TST     ERRBRT+4                ;CHECK FOR BROADCAST CHANNEL
 439 007542  001413                          BEQ     3$                      ;BRANCH IF BROADCAST CHANNEL
 440 007544  110460  003122'                 MOVB    R4,IBUF0+0(R0)          ;STORE TERMINAL # IN TABLE
 441 007550  110260  003123'                 MOVB    R2,IBUF0+1(R0)          ;STORE CONTROLLER #
 442 007554  010360  004122'                 MOV     R3,IBUF4(R0)            ;STORE SB #
 443 007560  005204                          INC     R4                      ;UPDATE PHYSICAL TERMINAL #
 444 007562  062700  000002                  ADD     #2,R0                   ;UPDATE POINTER
 445 007566  005203                          INC     R3                      ;UPDATE SB #
 446 007570  000740                          BR      2$
 447 007572  005202                  3$      INC     R2                      ;INCREMENT CONTROLLER # I E  LOG NEXT CONTROLLER
 448 007574  005004                          CLR     R4                      ;CLEAR TERMINAL #.  I E TERMINALS ON NEXT CONTROLLER
 449 007576  110460  003122'                 MOVB    R4,IBUF0+0(R0)          ;STORE TERMINAL #
 450 007602  110260  003123'                 MOVB    R2,IBUF0+1(R0)          ;STORE CONTROLLER #
 451 007606  010360  004122'                 MOV     R3,IBUF4(R0)            ;STORE SB
 452 007612  005204                          INC     R4                      ;UPDATE TERMINAL #
 453 007614  005203                          INC     R3                      ;UPDATE SB #
 454 007616  062700  000002                  ADD     #2,R0                   ;UPDATE POINTER
 455 007622  000723                          BR      2$                      ;ASSIGN NEXT TERMINAL
 456 007624  012760  177777  003122'  4$      MOV     #-1,IBUF0(R0)           ;END OF SB TABLE
 457 007632  013703  005250'                 MOV     UNITS,R3                ;GET RUNTIME UNIT #
 458 007636  005004                          CLR     R4                      ;POINTER TO UNTBL CONTAINS
 459                                                                         ;PHYSICAL # IN LO-BYTE
 460                                                                         ;& CONTROLLER # IN HI-BYTE
 461 007640  005001                  5$      CLR     R1                      ;POINTER TO IBUF0 TABLE
```

```
    007642  016402  000306'                   MOV     UNTBL(R4),R2               ,GET CONTROLLER # TERM # FROM UNIT TABLE
    007646  020261  003122'           6$      CMP     R2,IBUFO(R1)               ,CHECK IF SAME FROM MAPPED TABLE
    007652  001406                            BEQ     7$                         ,BRANCH IF SAME
    007654  005761  003122'                   TST     IBUFO(R1)                  ,CHECK IF END OF TABLE
    007660  100413                            BMI     8$                         ,BRANCH IF END
    007662  062701  000002                    ADD     #2,R1                      ,GET NEXT ITEM
    007666  000767                            BR      6$                         ,CONTINUE CHECKING
    007670  116164  004122' 000506' 7$        MOVB    IBUF4(R1),DEVTAB+0(R4)     ;STORE LOGICAL SB NUMBER
    007676  062704  000002                    ADD     #2,R4                      ;GET NEXT UNTBL ENTRY
    007702  005303                            DEC     R3                         ,LAST PHYSICAL TERMINAL TO BE STORED
    007704  001355                            BNE     5$                         ,BRANCH IF NO--
    007706  000476                            BR      11$                        ,ELSE GET NEXT DEVICE
    007710  016437  000306' 007234' 8$        MOV     UNTBL(R4),TEMP             ,TEMPORARY STORAGE FOR PRINT ROUTINE
    007716                                    PRINTF  #MSGO,<B,TEMP+1>,<B,TEMP+0>
    007716  005046                                            CLR     -(SP)
    007720  153716  007234'                                   BISB    TEMP+0,(SP)
    007724  005046                                            CLR     -(SP)
    007726  153716  007235'                                   BISB    TEMP+1,(SP)
    007732  012746  010674'                                   MOV     #MSGO,-(SP)
    007736  012746  000003                                    MOV     #3,-(SP)
    007742  010600                                            MOV     SP,RO
    007744  104017                                            EMT     C$PNTF
    007746  062706  000010                                    ADD     #10,SP
                                                                                     ,PRINTED THE NON-SYSGENED MESSAGE
                                      ,       ***********************************************************
                                      ,       BUBBLE ROUTINE. .
                                      ,       ++
                                      ,       IF ONE , OR ALL TERMINALS ENTERED ARE UNASSIGNABLE
                                      ,       OR UNATTACHABLE THIS ROUTINE ELIMINATES THEM OFF
                                      ,       THE TABLE AND ABORTS THE PROGRAM. .
                                      ,       --
                                      ,       ***********************************************************
    007752  005303                            DEC     R3                         ,ONE TOO MANY UNITS? UNASSIGNABLE?
    007754  001433                            BEQ     10$                        ;BRANCH IF ONE TOO MANY
    007756                                    PUSH    <R3,R4>                    ;SAVE
    007756  010346
    007760  010446
    007762  016464  000310' 000306' 9$        MOV     UNTBL+2(R4),UNTBL(R4)      ,BUBBLE UP FROM BELOW  TO THIS LOCATION
    007770  062704  000002                    ADD     #2,R4                      ,NEXT LOCATION
    007774  005303                            DEC     R3                         ,DECREASE LOCATION COUNT
    007776  001371                            BNE     9$                         ,BRANCH IF NOT DONE
    010000                                    POP     <R4,R3>                    ,RESTORE
    010000  012604
    010002  012603
    010004  005337  005250'                   DEC     UNITS                      ,ELIMINATE ONE UNIT OFF TABLE
    010010  005737  005250'                   TST     UNITS                      ,ARE THERE ANY MORE DEVICES
    010014  001311                            BNE     5$                         ,YES--BRANCH
    010016                                    PRINTF  #MABORT                    ,NO-PRINT ABORT MESSAGE
    010016  012746  010767'                                   MOV     #MABORT,-(SP)
    010022  012746  000001                                    MOV     #1,-(SP)
    010026  010600                                            MOV     SP,RO
    010030  104017                                            EMT     C$PNTF
    010032  062706  000004                                    ADD     #4,SP
    010036                                    EXIT$S                             ,AND EXIT TEST
    010036  012746                            MOV     (PC)+,-(SP)
    010040     063     001                    .BYTE   51 ,1
    010042  104377                            EMT      0<377>
```

```
    498 010044  005337  005250'          10$     DEC     UNITS                   ,ONE TOO MANY UNITS
    499 010050  005737  005250'                  TST     UNITS                   ,LAST DEVICE?
    500 010054  001013                           BNE     11$                     ;BRANCH IF NO
    501 010056                                   PRINTF  #MABORT                 ,ELSE PRINT ABORT MESSAGE
        010056  012746  010767'                                   MOV     #MABORT,-(SP)
        010062  012746  000001                                   MOV     #1,-(SP)
        010066  010600                                           MOV     SP,R0
        010070  104017                                           EMT     C$PNTF
        010072  062706  000004                                   ADD     #4,SP
    502 010076                                   EXIT$S                          ;AND EXIT
        010076  012746                           MOV     (PC)+,-(SP)
        010100     063     001                   .BYTE   51 ,1
        010102  104377                           EMT      0<377>
    503 010104  005001                  11$      CLR     R1                      ,DEVICE TABLE POINTER
    504 010106  012705  000011                   MOV     #LUN,R5                 ,SET THE LUN
    505 010112  013703  005250'                  MOV     UNITS,R3                ,KEEP COUNT
    506 010116  012737  000000G 000216'         MOV     #IO.ATT,Q1ODPB+10 FUN   ,SET FOR ATTACHES
    507 010124  110537  000246'          12$     MOVB    R5,LUDPB+2              ;SET LUN FOR DIR$
    508 010130  116102  000506'                  MOVB    DEVTAB+0(R1),R2         ,GET THE SB DEVICE
    509 010134  110237  000252'                  MOVB    R2,LUDPB+6              ;SET THE DEVICE NUMBER
    510 010140                                   DIR$    #LUDPB                  ,ISSUE THE I/O REQUEST
        010140  012746  000244'                  MOV     #LUDPB,-(SP)
        010144  104377                           EMT      0<377>
    511 010146  103017                           BCC     13$                     ,BRANCH IF REQUEST IS ACCEPTED
    512 010150  013704  000000G                  MOV     $DSW,R4                 ,IF NOT GET ERROR CODE
    513 010154                                   PRINTF  #MSG1,<B,R2>,<B,R4>     ;PRINT ERROR MESSAGE *UNASSIGNABLE TERMINAL
        010154  005046                                   CLR     -(SP)
        010156  150416                                   BISB    R4,(SP)
        010160  005046                                   CLR     -(SP)
        010162  150216                                   BISB    R2,(SP)
        010164  012746  011020'                          MOV     #MSG1,-(SP)
        010170  012746  000003                           MOV     #3,-(SP)
        010174  010600                                   MOV     SP,R0
        010176  104017                                   EMT     C$PNTF
        010200  062706  000010                           ADD     #10,SP
    514 010204  000437                           BR      17$                     ;DO NOT TRY TO ATTACH--
    515                                                                          ,BUT GO GET RID OF UNIT
    516                                                                          ;FROM THE DEVICE TABLE
    517 010206  110561  000507'          13$     MOVB    R5,DEVTAB+1(R1)         ,PUT THE LUN IN THE DEVICE TABLE
    518 010212  010537  000220'                  MOV     R5,Q1ODPB+10.LUN
    519 010216                                   DIR$    #Q1ODPB                 ,ISSUE I/O REQUEST
        010216  012746  000214'                  MOV     #Q1ODPB,-(SP)
        010222  104377                           EMT      0<377>
    520 010224  103411                           BCS     16$                     ;ERROR INDICATOR
    521 010226  062701  000002          14$     ADD     #2,R1                   ,GET NEXT SB (LOGICAL)
    522 010232  005205                           INC     R5                      ,NEXT LUN
    523 010234  005303                           DEC     R3                      ,CHECK FOR LAST SB
    524 010236  001332                           BNE     12$                     ;BRANCH IF NOT--GO DO NEXT ASSIGN
    525 010240                          15$
    526 010240  012761  177777  000506'         MOV     #-1,DEVTAB(R1)          ,INSERT END OF FLAG IN DEVICE TABLE
    527 010246  000457                           BR      22$                     ,AND PRINT ACTIVE DEVICE MAP
    528 010250  013704  000000G          16$     MOV     $DSW,R4                 ,PRINT ERROR MESSAGE  I E UNATTACHABLE DEVICE
    529 010254                                   PRINTF  #MSG,<B,R2>,<B,R4>
        010254  005046                                   CLR     -(SP)
        010256  150416                                   BISB    R4,(SP)
        010260  005046                                   CLR     -(SP)
        010262  150216                                   BISB    R2,(SP)
```

```
        010264  012746  011101'                        MOV     #MSG,-(SP)
        010270  012746  000003                         MOV     #3,-(SP)
        010274  010600                                  MOV     SP,R0
        010276  104017                                  EMT     C$PNTF
        010300  062706  000010                          ADD     #10,SP
   530
   531
   532                                  ;       ****************************************
   533                                  ;       BUBBLE ROUTINE....
   534                                  ;       ****************************************
   535
   536
   537 010304  005303                   17$:    DEC     R3                      ;ONE TOO MANY UNITS IN DEVICE TABLE
   538                                                                          ;BECAUSE DEVICE IS UNASSIGNABLE
   539                                                                          ;OR CANNOT BE ATTACHED
   540 010306  001417                           BEQ     19$                     ;BRANCH IF ONE TOO MANY
   541 010310                                   PUSH    <R3,R1>
       010310  010346
       010312  010146
   542 010314  016161  000310' 000306'  18$:    MOV     UNTBL+2(R1),UNTBL(R1)   ;BUBBLE UP FROM BELOW TO THIS LOCATION
   543 010322  016161  000510' 000506'         MOV     DEVTAB+2(R1),DEVTAB(R1) ;"      "   "   "    "  "   "
   544 010330  062701  000002                   ADD     #2,R1                   ;NEXT LOCATION
   545 010334  005303                           DEC     R3                      ;DECREASE UNITS LOCATION
   546 010336  001366                           BNE     18$                     ;BRANCH IF NOT FINISHED
   547 010340                                   POP     <R1,R3>
       010340  012601
       010342  012603
   548 010344  000667                           BR      12$                     ;GO DO NEXT ASSIGN I E ALUN$
   549 010346  005337  005250'          19$:    DEC     UNITS                   ;ELIMINATE ONE UNIT OFF TABLE
   550                                                                          ;THIS INDICATES ONE UNIT LESS TO TEST
   551 010352  005737  005250'                   TST     UNITS                   ;CHECK FOR ONE DEVICE AND UNASSIGNABLE
   552 010356  001330                           BNE     15$                     ;BRANCH IF NOT------
   553 010360                                   PRINTF  #MABORT                 ;PRINT ABORT MESSAGE
       010360  012746  010767'                                MOV     #MABORT,-(SP)
       010364  012746  000001                                 MOV     #1,-(SP)
       010370  010600                                         MOV     SP,R0
       010372  104017                                         EMT     C$PNTF
       010374  062706  000004                                 ADD     #4,SP
   554 010400                                   EXIT$S                          ;AND EXIT
       010400  012746                           MOV     (PC)+,-(SP)
       010402     063      001                  .BYTE   51 ,1
       010404  104377                           EMT      0<377>
   555 010406                           22$:
   556 010406  013703  005250'                   MOV     UNITS,R3                ;GET NUMBER OF ACTIVE DEVICES
   557 010412                                   PRINTF  #MSG2,<B,R3>            ;PRINT NUMBER OF ACTIVE DEVICES
       010412  005046                                         CLR     -(SP)
       010414  150316                                         BISB    R3,(SP)
       010416  012746  011162'                                MOV     #MSG2,-(SP)
       010422  012746  000002                                 MOV     #2,-(SP)
       010426  010600                                         MOV     SP,R0
       010430  104017                                         EMT     C$PNTF
       010432  062706  000006                                 ADD     #6,SP
   558                                                                          ;AND HEADER FOR ACTIVE DEVICE MAP
   559 010436  005000                           CLR     R0                      ;POINTER TO TABLES
   560 010440  116001  000507'          20$:    MOVB    DEVTAB+1(R0),R1         ;GET LUN TO BE PRINTED
   561 010444  116002  000506'                   MOVB    DEVTAB+0(R0),R2         ;GET LOGICAL SB NUMBER
   562 010450  116004  000307'                   MOVB    UNTBL+1(R0),R4          ;GET CONTROLLER NUMBER
```

```
563 010454  116005  000306'                 MOVB    UNTBL+0(R0),R5          ;GET PHYSICAL TERM#
564 010460                                  PUSH    R0                      ;SAVE R0**SUPERVISOR USES IT
    010460  010046
565 010462                                  PRINTF  #MSG3,<B,R5>,<B,R4>,<B,R2>,<B,R1>  ,
    010462  005046                                          CLR     -(SP)
    010464  150116                                          BISB    R1,(SP)
    010466  005046                                          CLR     -(SP)
    010470  150216                                          BISB    R2,(SP)
    010472  005046                                          CLR     -(SP)
    010474  150416                                          BISB    R4,(SP)
    010476  005046                                          CLR     -(SP)
    010500  150516                                          BISB    R5,(SP)
    010502  012746  011306'                                MOV     #MSG3,-(SP)
    010506  012746  000005                                  MOV     #5,-(SP)
    010512  010600                                          MOV     SP,R0
    010514  104017                                          EMT     C$PNTF
    010516  062706  000014                                  ADD     #14,SP
566 010522                                  POP     R0                      ;RESTORE R0
    010522  012600
567                                                                         ;PRINT ACTIVE DEVICE MAP
568 010524  062700  000002                 ADD     #2,R0
569 010530  005303                         DEC     R3
570 010532  001342                         BNE     20$
571 010534  005005                 21$     CLR     R5                      ;ENTER AST WITH R5
572
573
574                                 ;       ****************************************
575                                 ;       PROGRAM INITIALIZATION
576                                 ;       ****************************************
577
578
579
580 010536  005037  000302'         INIT    CLR     DEVPTR                  ;SET DEVICE POINTER TO START OF TABLE
581 010542  013737  005250'  000304'         MOV     UNITS,NUMDEV            ;SET THE NUMBER OF DEVICES
582 010550  012701  001016'                 MOV     #OBUFF,R1               ;GET 1'ST OUTPUT BUFFER ADDRESS
583 010554  012102                 9$      MOV     (R1)+,R2                ;GET BUFFER,UPDATE ADDRESS
584 010556  105022                         CLRB    (R2)+                   ;CLR 1'ST BYTE IN BUFFER
585 010560  005202                         INC     R2                      ;GET NEXT BYTE
586 010562  112712  000014                 MOVB    #14,(R2)                ;LOAD TRANSPARENT READ COMMAND
587 010566  005711                         TST     (R1)                    ;CHECK FOR END OF BUFFER
588 010570  001371                         BNE     9$                      ;BRANCH IF NOT END----
589 010572  005001                         CLR     R1                      ;INITIALIZE REGISTER
590 010574  112737  177777  005252'       MOVB    #-1,ERRTBL              ;INIT ERROR TABLES
591 010602  005201                 10$     INC     R1
592 010604  105061  005252'               CLRB    ERRTBL(R1)
593 010610  022701  001750                 CMP     #1000.,R1
594 010614  001372                         BNE     10$
595 010616  105037  007222'               CLRB    TERM                    ;CLR TERMINAL # LOCS
596 010622  005037  005242'               CLR     BFFPT                   ;CLR CURRENT BUFF POINTER
597 010626  005037  007224'               CLR     ERRPNT                  ;CLR ERROR POINTER
598 010632  012700  001006'               MOV     #OFFLIN+64.,R0          ;CLEAR OFFLINE TABLE
599 010636  005040                 14$     CLR     -(R0)
600 010640  020027  000706'               CMP     R0,#OFFLIN              ;END OF TABLE
601 010644  101374                         BHI     14$
602 010646                                  MRKT$S  ,#60.,#2,#MRKAST
    010646  012746  011744'                MOV     #MRKAST,-(SP)
    010652  012746  000002                 MOV     #2,-(SP)
```

INIT CODE


```
        010656  012746  000074                  MOV     #60.,-(SP)
        010662  005046                          CLR     -(SP)
        010664  012746                          MOV     (PC)+,-(SP)
        010666     027     005                   BYTE    23,5
        010670  104377                          EMT      0<377>
    603 010672                          ENDINIT
        010672                          L10005
        010672  104011                                  EMT     C$INIT
```

```
    605                                 ;
    606                                 ;       ************************************************************
    607                                 ;       GLOBAL ASCII FOR INITIALIZATION SECTION
    608                                 ;       ALL MESSAGES FOR THE PROGRAM INIT ARE CONTAINED HERE.
    609                                 ;       ************************************************************
    610
    611
    612
    613
    614 010674  045  116  045  MSG0    ASCIZ  "%N%ACONTROLLER %D1%A, TERMINAL %D3%A IS NOT SYSGENED IN %N"
        010677  101  103  117
        010702  116  124  122
        010705  117  114  114
        010710  105  122  040
        010713  045  104  061
        010716  045  101  054
        010721  040  124  105
        010724  122  115  111
        010727  116  101  114
        010732  040  045  104
        010735  063  045  101
        010740  040  111  123
        010743  040  116  117
        010746  124  040  123
        010751  131  123  107
        010754  105  116  105
        010757  104  040  111
        010762  116  056  045
        010765  116  000
    615 010767  045  116  045  MABORT  ASCIZ  "%N%APROGRAM IS ABORTED%N"
        010772  101  120  122
        010775  117  107  122
        011000  101  115  040
        011003  111  123  040
        011006  101  102  117
        011011  122  124  105
        011014  104  045  116
        011017  000
    616 011020  045  116  045  MSG1    ASCIZ  /%N%ATERMINAL %D2%A CANNOT BE ASSIGNED - $DSW=%06/
        011023  101  124  105
        011026  122  115  111
        011031  116  101  114
        011034  040  045  104
        011037  062  045  101
        011042  040  103  101
        011045  116  116  117
        011050  124  040  102
        011053  105  040  101
        011056  123  123  111
        011061  107  116  105
        011064  104  040  055
        011067  040  044  104
        011072  123  127  075
        011075  045  117  066
        011100  000
    617 011101  045  116  045  MSG     ASCIZ  "%N%ATERMINAL %D2%A CANNOT BE ATTACHED-$DSW=%03%N"
        011104  101  124  105
```

```
        011107      122      115      111
        011112      116      101      114
        011115      040      045      104
        011120      062      045      101
        011123      040      103      101
        011126      116      116      117
        011131      124      040      102
        011134      105      040      101
        011137      124      124      101
        011142      103      110      105
        011145      104      055      044
        011150      104      123      127
        011153      075      045      117
        011156      063      045      116
        011161      000
   618  011162      045      116      045  MSG2    .ASCII  /%N%ANUMBER OF ENTERED TERMINALS=%D2%N/
        011165      101      116      125
        011170      115      102      105
        011173      122      040      117
        011176      106      040      105
        011201      116      124      105
        011204      122      105      104
        011207      040      124      105
        011212      122      115      111
        011215      116      101      114
        011220      123      075      045
        011223      104      062      045
        011226      116
   619  011227      045      116      045          ASCIZ  /%N%AUNIT ** CONTROLLER ** LOGICAL SB ** LUN **/
        011232      101      125      116
        011235      111      124      040
        011240      052      052      040
        011243      103      117      116
        011246      124      122      117
        011251      114      114      105
        011254      122      040      052
        011257      052      040      114
        011262      117      107      111
        011265      103      101      114
        011270      040      123      102
        011273      040      052      052
        011276      040      114      125
        011301      116      040      052
        011304      052      000
   620  011306      045      116      045  MSG3:   ASCIZ/%N%D3%A          %D3%A          %D3%A      %D3%N/
        011311      104      063      045
        011314      101      040      040
        011317      040      040      040
        011322      040      040      040
        011325      040      040      040
        011330      045      104      063
        011333      045      101      040
        011336      040      040      040
        011341      040      040      040
        011344      040      040      040
        011347      045      104      063
        011352      045      101      040
```

```
        011355      040     040     040
        011360      040     040     045
        011363      104     063     045
        011366      116     000
   621                                          EVEN
```

```
                                        SBTTL  MAIN ROUTINE


                                ;       ************************************************************
                                ;       THE MAIN ROUTINE EXECUTES A PROGRAM PASS,A PARTIAL PASS,
                                ;       PROCESSES A TERMINAL LIST,AND FINALLY ONE TERMINAL
                                ;       A PROGRAM PASS IS DEFINED HERE AS ALL PATTERNS OF
                                ;       ALL DATA LENGTHS TO ALL TERMINALS
                                ;       A PARTIAL PASS IS DEFINED AS ONE PATTERN ALL DATA
                                ;       LENGTHS TO ALL TERMINALS
                                ;       A LIST OF TERMINALS CONTAINS 1-63 UNITS
                                ;       TO PROCESS A TERMINAL, RSX-11M'S DIRECTIVE
                                ;       PARAMETER BLOCK IS LOADED AND AN I/O DIRECTIVE IS
                                ;       ISSUED TO THE DECICE
                                ;       ************************************************************
                                ;
                                ;

011370                          BGNTST
011370                          T1

                                ;       EXECUTE A PASS
                                ;
011370  005037  007226'         START   CLR     PATPT                   ;CLEAR PATTERN POINTER
011374  004737  011760'         1$      JSR     PC,C1                   ;SET BYTE PATTERN IN OUTPUT BUFFER
011400  103460                          BCS     6$                      ;END OF TEST

                                ;       EXECUTE A PARTIAL PASS

011402  012737  000400  005240'         MOV     #256.,LENGTH            ;SET RECORD LENGTH
011410  004737  012256'         3$      JSR     PC,D3                   ;CHECK ERRORS
011414  006237  005240'                 ASR     LENGTH                  ;VARY RECORD LENGTH
011420  001765                          BEQ     1$                      ;BRANCH IF END RECORD LENGTH

                                ;       PROCESS A LIST OF TERMINALS

011422  004737  012054          4$      JSR     PC,E1                   ;SUBROUTINE (GET A TERMINAL FROM LIST)
011426  103770                          BCS     3$                      ;IF END OF TERMINAL LIST CHECK ERRORS
011430  004737  012134                  JSR     PC,E2                   ; CALL POLL AND SET OUTPUT BUFFER ROUTINE

                                ;       PROCESS A TERMINAL

011434  013703  005242'                 MOV     BFFPT,R3                ;GET BUFFER POINTER
011440  012737  003400  000216'         MOV     #IO.RXP,Q1ODPB+10       ;READ TRANSPARENT TO THE DEVICE
011446  016337  001016'  000236'         MOV     OBUFF(R3),Q1ODPB+10.A4  ;LOAD OUTPUT BUFFER ADDRESS
011454  005237  000236'                 INC     Q1ODPB+10.A4            ;SKIP THE FREE FLAG
011460  013737  005240'  000240'         MOV     LENGTH,Q1ODPB+10.A5     ;LOAD OUTPUT BUFFER SIZE
011466  062737  000002  000240'         ADD     #2,Q1ODPB+10.A5         ;2 BYTES LESS THAN OUT-BUFF
011474  016337  003102'  000230'         MOV     INBUFF(R3),Q1ODPB+10.A1 ;LOAD INPUT BUFFER ADDRESS
011502  013737  005240'  000232'         MOV     LENGTH,Q1ODPB+10.A2     ;LOAD INPUT BUFFER SIZE
011510  016337  005122'  000224'         MOV     IOSB(R3),Q1ODPB+10.IST  ;LOAD STATUS WORD
011516  012737  013362'  000226'         MOV     #AST,Q1ODPB+10.AST      ;LOAD TERM AST ADDRESS
011524  113737  005244'  000220'         MOVB    LUNS,Q1ODPB+10.LUN      ;LOAD LOGICAL UNIT NUMBER
011532                          DIR$    #Q1ODPB                 ;ISSUE I/O DIRECTIVE
011532  012746  000214'                 MOV     #Q1ODPB,-(SP)
```

```
     011536  104377                          EMT     0(377)
 678 011540  000730                          BR      4$                          ;GET NEXT TERMINAL
 679                                         ;*******************************************************
 680                                         ;KEEP PASS COUNT ONLY WHEN DEVICES ARE ACTIVE
 681                                         ;*******************************************************
 682 011542                              6$  PUSH    R0                          ;SAVE
     011542  010046
 683 011544  005000                          CLR     R0                          ;INIT POINTER TO OFFLINE TABLE
 684 011546  023700  005250'             7$  CMP     UNITS,R0                    ;CHECK IF END OF OFFLINE TABLE
 685 011552  001407                          BEQ     9$                          ;DO NEXT PASS IF END
 686 011554  105760  000706'                TSTB    OFFLIN(R0)                  ;CHECK FOR LIVE TERMINAL
 687 011560  001402                          BEQ     8$                          ;IF LIVE KEEP PASS COUNT
 688 011562  105720                          TSTB    (R0)+                       ;IF OFFLINE UPDATE POINTER
 689 011564  000770                          BR      7$                          ;CONTINUE CHECKING
 690 011566  005237  005246'             8$  INC     PASS                        ;LOG PASS
 691 011572                              9$  BREAK                               ;RETURN FOR A PEEK AT SUPERVISOR
     011572  104022                                      EMT     C$BRK
 692 011574                                  POP     R0                          ;RESTORE
     011574  012600
 693 011576  005737  005236'                 TST     PASFG                       ;TIME TO PRINT PASS COUNT?
 694 011602  001672                          BEQ     START                       ;NO-CONTINUE TEST NG
 695
 696 011604                                  MRKT$S  ,#60 ,#2,#MRKAST
     011604  012746  011744'                MOV     #MRKAST,-(SP)
     011610  012746  000002                  MOV     #2,-(SP)
     011614  012746  000074                  MOV     #60 ,-(SP)
     011620  005046                          CLR     -(SP)
     011622  012746                          MOV     (PC)+,-(SP)
     011624     027     005                   BYTE    23 ,5
     011626  104377                          EMT      0(377)
 697 011630  005001                          CLR     R1                          ;POINTER TO OFFLINE TABLE
 698 011632  005003                          CLR     R3                          ; NUMBER OF ACTIVE TERMINALS PER PASS
 699 011634  013702  005250                  MOV     UNITS,R2                    ;GET MAXIMUM NUMBER OF UNITS
 700 011640  020102                      13$ CMP     R1,R2                       ;END OF OFFLINE TABLE
 701 011642  001406                          BEQ     11$                         ;BRANCH IF END
 702 011644  105761  000706'                TSTB    OFFLIN(R1)                  ;CHECK IF TERMINAL IS OFFLIN
 703 011650  001001                          BNE     12$                         ;IF OFFLINE BRANCH
 704 011652  005203                          INC     R3                          ;LOG ACTIVE TERMINALS
 705 011654  005201                      12$ INC     R1                          ;UPDATE OFFLINE TABLE POINTER
 706 011656  000770                          BR      13$                         ;CHECK NEXT TERMINAL
 707 011660                              11$ GTIM$S  #BUF                        ; GET THE SYSTEM TIME
     011660  012746  005206                  MOV     #BUF,-(SP)
     011664  012746                          MOV     (PC)+,-(SP)
     011666     075     002                   BYTE    61 ,2
     011670  104377                          EMT      0(377)
 708 011672                                  PRINTX  #EOP,PASS,R3,BUF+G TIHR,BUF+G TIMI          ;PRINT NUMBER OF PASSES E T
     011672  013746  005216'                            MOV     BUF+G TIMI,-(SP)
     011676  013746  005214'                            MOV     BUF+G TIHR,-(SP)
     011702  010346                                     MOV     R3,-(SP)
     011704  013746  005246'                            MOV     PASS,-(SP)
     011710  012746  015311'                            MOV     #EOP,-(SP)
     011714  012746  000005                              MOV     #5,-(SP)
     011720  010600                                     MOV     SP,R0
     011722  .04015                                     EMT     C$PNTX
     011724  062706  000014                            ADD     #14,SP
 709 011730  005037  005246'                 CLR     PASS                        ;RESET THE PASS COUNT
 710 011734  005037  005236'                 CLR     PASFG                       ;FOR THE NEXT PRINT REPORT
```

SEQ 0047

```
    711 011740  000613                 10$     BR      START                   ;DO NEXT PASS
    712 011742                          ENDTST
        011742                          L10006
        011742  104001                                  EMT     CSETST
```

SEQ 0048

```
                                        SBTTL  DECLARE A SIGNIFICANT EVENT PER 5 MINS
                                ;
                                ;
                                ;
                                ;
                                ;       ************************************************
                                ;       THIS ROUTINE USES THE MRKT$S DIRECTIVE TO
                                ;       PRINT THE NUMBER OF PROGRAM PASSES AND
                                ;       THE NUMBER OF ACTIVE TERMINALS IN PRESENT PASS
                                ;       EVERY 5 MINUTES
                                ;       ************************************************
                                ;
                                ;
                                ;
                                ;
011744  005237  005236'       MRKAST  INC     PASFG                   ;SET THE 1 MIN  PASS FLAG
011750  005726                        TST     (SP)+                   ;REMOVE EVENT FLAG PER RSX
011752                                  ASTX$S                          ;RETUPN
011752  012746                        MOV     (PC)+,-(SP)
011754     163     001                 .BYTE   115 ,1
011756  104377                        EMT      0<377>
                                ;
                                ;
                                ;
                                ;
```

```
                                        SBTTL  SET BYTE PATTERN IN OUTPUT BUFFER


                                ;       ************************************************************
                                ;       THIS ROUTINE CONSECUTIVELY GETS ONE OF SEVEN
                                ;       DATA PATTERNS,PACKS IT INTO EIGHT BUFFERS (128 BYTES LONG)
                                ;       FOR SUBSEQUENT TRANSMISSION VIA THE SERIAL BUS
                                ;       TO THE DEVICES.
                                ;       ++NOTE++ EIGHT TERMINALS ARE SIMULTANEOUSLY PROCESSED
                                ;       AT ANY GIVEN TIME
                                ;       ************************************************************


011760  013704  007226'         C1      MOV     PATPT,R4                ;GET PATTERN POINTER
011764  122764  000001  005226'          CMPB    #1,PATTRN(R4)           ;END OF PATTERN FLAG
011772  001004                          BNE     1$                      ;IF NO,GET BUFFERS
011774  005037  007226'                  CLR     PATPT                   ;RESET THE POINTER
012000  000261                          SEC                             ;SET END OF PATTERN INDICATOR
012002  000423                          BR      4$                      ;AND RETURN
012004  012700  001016'          1$      MOV     #OBUFF,R0               ;GET OUPUT BUFFER ADDRESS
012010  005003                          CLR     R3                      ;BUFFER NUMBER COUNTER
012012  012001                  2$      MOV     (R0)+,R1                ;GET BUFFERS 0 THRU 7
012014  012702  000200                  MOV     #128 ,R2                ;SET MAXIMUM DATA BYTE LENGTH
012020  116461  005226'  000003  3$      MOVB    PATTRN(R4),DATA(R1)     ;LOAD DATA PER BYTE IN BUFFER
012026  005201                          INC     R1                      ;GET NEXT BYTE
012030  005302                          DEC     R2                      ;COUNT DATA BYTE LENGTH
012032  001372                          BNE     3$                      ;BRANCH UNTIL  FINISHED
012034  005203                          INC     R3                      ;UPDATE BUFFER NUMBER COUNT
012036  022703  000010                  CMP     #8 ,R3                  ;HAVE ALL 8 BUFFERS BEEN FILLED
012042  001363                          BNE     2$                      ;IF NO!! FILL NEXT BUFFER
012044  005237  007226'                  INC     PATPT                   ;INCREMENT THE PATTERN POINTER
012050  000241                          CLC                             ;RESET END FLAG FOR PATTERN TABLE
012052  000207                  4$      RTS     PC                      ;RETURN
```

```
                        SBTTL GET NEXT DEVICE AND LUN INFORMATION
                ;
                ;
                ;       ******************************************************************
                ;++
                ;ROUTINE TO GET THE NEXT PAIR OF DEVICE NUMBER AND ASSCOIATED LUN
                ;RETURNS A -1 IN LUN,,DEV AT END OF TABLE
                ;
                ;INPUT           DEVPTR = POINTER TO NEXT DEV CE
                ;                DEVTAB = TABLE OF ACTIVE DEV CES & LUNS
                ;
                ;OUTPUT                  TERM = TERMINAL NUMBER OF NEXT AVAILABLE UNIT
                ;                LUNS = RSX ASSIGNED LUN NUMBER
                ;
                ;--
                ;       ******************************************************************
012054 013700 000302'   E1      MOV     DEVPTR,R0               ;GET POINTER TO NEXT AVAILABLE UNIT
012060 006300                   ASL     R0                      ;CHANGE TO WORD OFFSET
012062 116037 000306' 007222    MOVB    UNTBL+0(R0),TERM        ;GET THE TERMINAL NUMBER
012070 116037 000507' 005244'   MOVB    DEVTAB+1(R0),LUNS       ;GET LUN NUMBER
012076 116037 000506' 007256'   MOVB    DEVTAB+0(R0),SB         ;SAVE SB (LOGICAL DEVICE
012104 005237 000302'           INC     DEVPTR                  ;POINT TO NEXT SET
012110 023737 000302  000304'   CMP     DEVPTR,NUMDEV           ;END OF LIST
012116 003404                   BLE     1$                      ;NO---RETURN
012120 005037 000302'           CLR     DEVPTR                  ;POINT TO TOP OF LIST
012124 000261                   SEC                             ;SET END OF LIST INDICATOR
012126 000401                   BR      2$                      ;AND RETURN
012130 000241           1$      CLC                             ;RESET END OF LIST INDICATOR
012132 000207           2$      RTS     PC                      ;RETURN
```

```
                                        SBTTL  POLL AND SET OUTPUT BUFFERS
                                ;
                                ;
                                ;       ******************************************************************
                                ;++
                                ;ROUTINE TO DETERMINE THE NEXT OUTPUT BUFFER TO USE.
                                ;
                                ;INPUT           TERM = TERMINAL NUMBER
                                ;
                                ;OUTPUT          BFFPT = OFFSET TO BUFFER TO USE
                                ;                OUTBUF BYTE 0 = SB - IN USE, IF 0-NOT IN USE
                                ;                OUTBUF BYTE 1 = TERMINAL NUMBER
                                ;
                                ;       ******************************************************************
                                ;NOTE            ROUTINE LOOPS FOREVER UNTIL A BUFFER BECOMES FREE
                                ;       ******************************************************************
                                ;--
                                ;
                                ;
012134                          E2      MRKT$S  #1 ,#5 ,#1              ;MARK TIME FOR #5 TICKS
012134  005046                         CLR     -(SP)
012136  012746  000001                  MOV     #1,-(SP)
012142  012746  000005                  MOV     #5 ,-(SP)
012146  012746  000001                  MOV     #1. ,-(SP)
012152  012746                         MOV     (PC)+,-(SP)
012154     027     005                   BYTE    23.,5
012156  104377                         EMT      0<377>
012160  012700  001016'                 MOV     #OBUFF,R0               ;START OF BUFFER TABLE
012164  005710                  1$      TST     (R0)                    ;TEST FOR END OF TABLE
012166  001006                         BNE     12$                     ;BRANCH IF NOT
012170                                 WTSE$S  #1                      ;WAITFOR SIG EVENT
012170  012746  000001                  MOV     #1,-(SP)
012174  012746                         MOV     (PC)+,-(SP)
012176     051     002                   BYTE    41 ,2
012200  104377                         EMT      0<377>
012202  000754                         BR      E2                      ;START AT BEGINNING OF TABLE
012204  105770  000000          12$.    TSTB    @(R0)                   ;SEE IF BUFFER IS IN USE
012210  001403                         BEQ     2$                      ;LOOP IF NON ZERO
012212  062700  000002                  ADD     #2,R0
012216  000762                         BR      1$
012220  010037  005242'          2$      MOV     R0,BFFPT                ;SAVE IF BUFFER POINTER
012224  011000                         MOV     (R0),R0                 ;GET ADDRESS OF TABLE
012226  113720  007256'                 MOVB    SB,(R0)+                ;SET SB
012232  113710  007222'                 MOVB    TERM,(R0)               ;INSERT DESIRED TERMINAL NUMBER
                                                                        ;TERMINAL # MUST BE LEFT JUSTIFIED
                                                                        ;IN THE BYTE WITH BIT0 SET
                                                                        ;AND BIT#1 CLEARED
012236  106310                         ASLB    (R0)
012240  106310                         ASLB    (R0)
012242  152710  000001                  BISB    #BIT00,(R0)             ;BITS IN BYTE HAS TO BE LEFT JUSTIFIED
012246  162737  001016' 005242'         SUB     #OBUFF,BFFPT            ;COMPUTE AND SAVE OFFSET
012254  000207                         RTS     PC                      ;RETURN
```

SEQ 0052

```
                        SBTTL CHECK ERRORS-CHECK FOR BAD CRC


                ;       ************************************************************
                ;       THIS ROUTINE WAITS FOR ALL BUFFERS TO BE FREE,
                ;       (DONE BY THE AST SERVICE ROUTINE)CHECKS FOR LIVE
                ;       TERMINALS ON THE BUS, AND ISSUES A COMMAND TO
                ;       GENERATE BAD CRC FROM THE DEVICES
                ;       NON DETECTED BAD CRC ERRORS BY ANY CONTROLLER ARE
                ;       LOGGED TOGETHER WITH PROCESSED ERRORS FROM AST SERVICE
                ;       ROUTINE, FOR SUBSEQUENT PRINT-OUT BY THE
                ;       PRINT MODULE
                ;       ************************************************************
                ;
                ;
                ;*************************************************************
                ;       WAIT FOR ALL BUFFERS TO BE FREE
                ;*************************************************************
                ;
                ;
                ;
012256                  D3      PUSH    <R2,R4,R3,R1>           ;SAVE REGISTERS
012256  010246
012260  010446
012262  010346
012264  010146
012266  005001                  CLR     R1
012270  005002                  CLR     R2
012272                  D3A     MRKT$S  #2,#5,#1                ;MARK TIME FOR SIG EVENT
012272  005046                  CLR     -(SP)
012274  012746  000001          MOV     #1,-(SP)
012300  012746  000005          MOV     #5,-(SP)
012304  012746  000002          MOV     #2,-(SP)
012310  012746                  MOV     (PC)+,-(SP)
012312     027     005           BYTE    23,5
012314  104377                  EMT      0<377>
012316  105771  001016'        TSTB     @0BUFF(R1)              ;IS BUFFER FREE?
012322  001406                  BEQ     123$                    ;BRANCH IF FREE
012324                          WTSE$S  #2                      ;WAITFOR SIG EVENT
012324  012746  000002          MOV     #2,-(SP)
012330  012746                  MOV     (PC)+,-(SP)
012332     051     002           BYTE    41,2
012334  104377                  EMT      0<377>
012336  000755                  BR      D3A                     ;CHECK FOR FREE BUFFER AGAIN
012340  005721          123$    TST     (R1)+                   ;COUNT NUMBER OF BUFFERS FREE
012342  022701  000020          CMP     #16,R1                  ;8 BUFFERS FREE?
012346  001351                  BNE     D3A                     ;BRANCH IF NO
012350  000137  012526'        JMP     15$                     ;TEMPO BYPASS
                ;*************************************************************
                ;       CHECK FOR BAD CRC
                ;*************************************************************
012354  005037  007262'        CLR     CRCFLG                  ;CRC COMPLETE FLAG
```

```
    891 012360  005762  000306'          20$     TST     UNTBL(R2)                ;START SEARCH FOR LIVE TERMINAL
    892 012364  100460                           BMI     15$                      ;IF END OF TABLE BRANCH OUT
    893 012366  006202                           ASR     R2                       ;ADJUST POINTER
    894 012370  105762  000706'                  TSTB    OFFLIN(R2)               ;ELSE CHECK IF TERMINAL IS OFFLINE
    895 012374  001403                           BEQ     21$                      ;LIVE TERMINAL FOUND
    896 012376  006302                           ASL     R2                       ;READJUST POINTER
    897 012400  005722                   30$     TST     (R2)+                    ;UPDATE TABLE POINTER
    898 012402  000766                           BR      20$                      ;CONTINUE SEARCH
    899 012404  006302                   21$     ASL     R2                       ;READJUST POINTER
    900 012406  116203  000306'                  MOVB    UNTBL+0(R2),R3           ;GET LIVE TERMINAL
    901 012412  110337  001042'                  MOVB    R3,OBUF0+1               ;LOAD LIVE TERMINAL INTO BUFFER
    902 012416  112737  000020  001043'          MOVB    #20,OBUF0+2              ;LOAD BAD CRC COMMAND
    903 012424  012737  003400  000216'          MOV     #10 RXP,Q10DPB+10 FUN    ;LOAD WRITE TO DEVICE
    904 012432  112737  001042' 000236'          MOVB    #OBUF0+1,Q10DPB+10 A4    ;LOAD OUTPUT BUFFER ADDRESS
    905 012440  013737  005240' 000240'          MOV     LENGTH,Q10DPB+10 A5      ;LOAD OUTPUT BUFFER SIZE
    906 012446  112737  003122' 000230'          MOVB    #IBUF0,Q10DPB+10 A1      ;LOAD INPUT BUFFER ADDRESS
    907 012454  013737  005240' 000232'          MOV     LENGTH,Q10DPB+10 A2      ;LOAD INPUT BUFFER LENGTH
    908 012462  012737  005142' 000224'          MOV     #ISTAT0,Q10DPB+10 IST    ;LOAD STATUS WORD
    909 012470  012737  013264' 000226'          MOV     #CRCAST,Q10DPB+10 AST    ;LOAD AST ADDRESS
    910 012476  116237  000507' 000220'          MOVB    DEVTAB+1(R2),Q10DPB+10 LUN      ;LOAD LOGICAL UNIT NUMBER
    911 012504                                   DIR$    #Q10DPB                  ;ISSUE I/O DIRECTIVE
        012504  012746  000214'                 MOV     #Q10DPB,-(SP)
        012510  104377                           EMT      0<377>
    912 012512  005737  007262'          22$     TST     CRCFLG                   ;CHECK AST ROUTINE
    913 012516  001775                           BEQ     22$                      ;IF NOT COMPLETED CHECK AGAIN
    914 012520  112737  000014  001043           MOVB    #14,OBUF0+2              ;RESTORE LOOP COMMAND IN BUFFER
    915                                  ;*********************************************
    916                                  ;       START CHECKING ERRORS
    917                                  ;*********************************************
    918 012526  005003                   15$     CLR     R3                       ;INIT ERROR TABLE
    919 012530  116304  005252'          11$     MOVB    ERRTBL(R3),R4            ;GET ERROR ENTRY & CHECK FOR ERRORS
    920 012534  122704  177777                   CMPB    #-1,R4                   ;END OF ERROR TABLE?
    921 012540  001002                           BNE     18$                      ;BRANCH IF NO CONTINUE TO CHECK
    922 012542  000137  013244'                  JMP     6$                       ;ELSE BRANCH OUT THRU RTS
    923                                                                           ;ELSE CHECK FOR
    924 012546  122704  000001           18$     CMPB    #1,R4                    ;HARD-ERROR?
    925 012552  001002                           BNE     33$                      ;BRANCH IF NO--
    926 012554  000137  013122'                  JMP     10$
    927 012560  122704  000010           33$     CMPB    #10,R4                   ;IS TERMINAL OFFLINE
    928 012564  001436                           BEQ     1$                       ;ERROR CALL
    929
    930 012566  122704  000002                   CMPB    #2,R4                    ; ELSE CHECK FOR RETRY ERROR
    931 012572  001450                           BEQ     2$                       ; ERROR? BRANCH IF YES TO ERROR CALL
    932 012574  122704  000003                   CMPB    #3,R4                    ; ELSE CHECK FOR DSW ERROR?
    933 012600  001462                           BEQ     3$                       ; ERROR? BRANCH IF YES TO ERROR CALL
    934 012602  122704  000005                   CMPB    #5,R4                    ;ELSE CHECK FOR TERMINAL ON-LINE CODE
    935 012606  001526                           BEQ     5$                       ;YES- IT IS ONLINE - PRINT MESSAGE
    936 012610  122704  000006                   CMPB    #6,R4                    ;ELSE CHECK FOR CRC ERROR CODE
    937 012614  001471                           BEQ     7$                       ;BRANCH IF YES TO INFORM OPERTOR
    938 012616  122704  000004                   CMPB    #4,R4                    ;ELSE CHECK DATA COMPARE ERROR
    939 012622  001402                           BEQ     180$                     ;BRANCH OUT THROUGH RTS IF NO
    940 012624  000137  013244'                  JMP     6$
    941
    942 012630                           180$    PRINTF  #LFCR                    ;PRINT CARRIAGE RETURN LINE FEED
        012630  012746  015436'                                 MOV     #LFCR,-(SP)
        012634  012746  000001                                  MOV     #1,-(SP)
        012640  010600                                          MOV     SP,R0
```

```
        012642  104017                          EMT     C$PNTF
        012644  062706  000004                  ADD     #4,SP
    943
    944                                 ;AND DATA COMPARISON ERROR
    945                                 ;*****************************************************
    946                                 ;       ERROR CALLS
    947 012650                          ;*****************************************************
        012650  104464          ERRSOFT 4,MSG04,MSGCD4          ;DATA ERROR   LOG ERROR,
        012652  000004                          TRAP    T$ERCODE
        012654  015213'                          WORD   4
        012656  014054'                          WORD   MSG04
    948                                          .WORD  MSGCD4
    949 012660  000723                  BR      11$             ;PRINT ERROR AND
    950 012662                          1$      PRINTF  #LFCR   ;BRANCH
        012662  012746  015436                                  ;PRINT CARRIAGE RETURN LINE FEED
        012666  012746  000001                  MOV    #LFCR,-(SP)
        012672  010600                          MOV    #1,-(SP)
        012674  104017                          MOV    SP,R0
        012676  062706  000004                  EMT    C$PNTF
    951 012702                                  ADD    #4,SP
        012702  104463          ERRHRD  1,MSG01,MSGCD1          ;TIME-OUT ERROR   LOG ERROR,
        012704  000001                          TRAP    T$ERCODE
        012706  015062'                          WORD   1
        012710  014370                          WORD   MSG01
    952 012712  000706                          WORD   MSGCD1
    953 012714                          BR      11$             ;BRANCH
        012714  012746  015436'          2$      PRINTF  #LFCR   ;PRINT CARRIAGE RETURN L NE FEED
        012720  012746  000001                  MOV    #LFCR,-(SP)
        012724  010600                          MOV    #1,-(SP)
        012726  104017                          MOV    SP,R0
        012730  062706  000004                  EMT    C$PNTF
    954                                         ADD    #4,SP
    955 012734
        012734  104464          ERRSOFT 2,MSG02,MSGCD2          ;RETRY ERROR   LOG ERROR,
        012736  000002                          TRAP    T$ERCODE
        012740  015127'                          .WORD  2
        012742  014734'                          WORD   MSG02
    956                                          WORD   MSGCD2
    957 012744  000671                  BR      11$             ;PRINT ERROR AND
    958                                         ;BRANCH
    959 012746                          3$      PRINTF  #LFCR   ;PRINT CARRIAGE RETURN LINE FEED
        012746  012746  015436'                MOV    #LFCR,-(SP)
        012752  012746  000001                 MOV    #1,-(SP)
        012756  010600                         MOV    SP,R0
        012760  104017                         EMT    C$PNTF
        012762  062706  000004                 ADD    #4,SP
    960 012766                          ERRSOFT 3,MSG03,MSGCD2          ;DSW ERROR   LOG ERROR,
        012766  104464                          TRAP    T$ERCODE
        012770  000003                          WORD   3
        012772  015145'                          WORD   MSG03
        012774  014734'                          WORD   MSGCD2
    961                                 ;PRINT ERROR AND
    962 012776  000654                  BR      11$             ;BRANCH
    963 013000                          7$      PRINTF  #LFCR   ;PRINT CARRIAGE RETURN LINE FEED
        013000  012746  015436'                 MOV    #LFCR,-(SP)
        013004  012746  000001                  MOV    #1,-(SP)
        013010  010600                          MOV    SP,R0
```

```
                013012  104017
                013014  062706  000004                          EMT     C$PNTF
                                                                ADD     #4,SP
           964  013020                          PRINTF  #CRCMSG,<B,ERRTBL+1(R3)>,<B,ERRTBL+2(R3)>
                013020  005046                                  CLR     -(SP)
                013022  156316  005254'                        BISB    ERRTBL+2(R3),(SP)
                013026  005046                                  CLR     -(SP)
                013030  156316  005253'                        BISB    ERRTBL+1(R3),(SP)
                013034  012746  014627'                        MOV     #CRCMSG,-(SP)
                013040  012746  000003                        MOV     #3,-(SP)
                013044  010600                                  MOV     SP,R0
                013046  104017                                  EMT     C$PNTF
                013050  062706  000010                        ADD     #10,SP
           965
           966  013054  062703  000006                ADD     #6,R3                   ;UPDATE ERROR TABLE POINTER
           967  013060  000137  012530'                JMP     11$
           968  013064                          5$      PRINTF  #ONMSG,<B,ERRTBL+1(R3)> ;PRINT TO OPERATOR-ON-LINE MSG
                013064  005046                                  CLR     -(SP)
                013066  156316  005253'                        BISB    ERRTBL+1(R3),(SP)
                013072  012746  015242'                        MOV     #ONMSG,-(SP)
                013076  012746  000002                        MOV     #2,-(SP)
                013102  010600                                  MOV     SP,R0
                013104  104017                                  EMT     C$PNTF
                013106  062706  000006                        ADD     #6,SP
           969  013112  062703  000006                ADD     #6,R3                   ;ADJUST ERROR POINTER
           970  013116  000137  012530'                JMP     11$
           971  013122                          10$     PRINTF  #LFCR                   ;PRINT CARRIAGE RETURN
                013122  012746  015436'                        MOV     #LFCR,-(SP)
                013126  012746  000001                        MOV     #1,-(SP)
                013132  010600                                  MOV     SP,R0
                013134  104017                                  EMT     C$PNTF
                013136  062706  000004                        ADD     #4,SP
           972  013142                          PRINTF  #M200
                013142  012746  014563'                        MOV     #M200,-(SP)
                013146  012746  000001                        MOV     #1,-(SP)
                013152  010600                                  MOV     SP,R0
                013154  104017                                  EMT     C$PNTF
                013156  062706  000004                        ADD     #4,SP
           973  013162  116300  005255'                MOVB    ERRTBL+3(R3),R0
           974  013166  005400                          NEG     R0
           975  013170                          PRINTF  #FMT01,<B,ERRTBL+0(R3)>,<B,ERRTBL+1(R3)>,<R0>,<B,ERRTBL+5(R3)>
                013170  005046                                  CLR     -(SP)
                013172  156316  005257'                        BISB    ERRTBL+5(R3),(SP)
                013176  010046                                  MOV     R0,-(SP)
                013200  005046                                  CLR     -(SP)
                013202  156316  005253'                        BISB    ERRTBL+1(R3),(SP)
                013206  005046                                  CLR     -(SP)
                013210  156316  005252'                        BISB    ERRTBL+0(R3),(SP)
                013214  012746  014450'                        MOV     #FMT01,-(SP)
                013220  012746  000005                        MOV     #5,-(SP)
                013224  010600                                  MOV     SP,R0
                013226  104017                                  EMT     C$PNTF
                013230  062706  000014                        ADD     #14,SP
           976  013234  062703  000006                ADD     #6,R3                   ;UPDATE ERROR POINTER
           977  013240  000137  012530'                JMP     11$
           978
           979  013244  005037  007224'          6$    CLR     ERRPNT                  ;RESET ERROR POINTER
           980  013250                                  POP     <R1,R3,R4,R2>           ;RESTORE REGISTERS
```

```
        013250  012601
        013252  012603
        013254  012604
        013256  012602
    981 013260  000207                  RTS     PC                  ;RETURN
```

```
                                SBTTL  CLEANUP CODE
                        ;
                        ;
                        ;       ************************************************************
                        ;       THIS CALL RETURNS THE DEVICES UNDER TEST TO A STATIC
                        ;       STATE IF THE ENDINT CALL IS USED. THE CLEANUP CODE IS
                        ;       ESPECIALLY REQUIRED WHEN A FUNCTION LEVEL DIAGNOSTIC
                        ;       IS RESIDENT WITH BOTH AN ON LINE SUPERVISOR AND AN
                        ;       OPERATING SYSTEM TO WHICH THE TEST DEVICE MUST BE
                        ;       RETURNED . .
                        ;       ************************************************************

013262                  BGNCLN
013262                  LSCLEAN

013262                  ENDCLN
013262                  L10007
013262  104012                          EMT     CSCLEAN
```

```
  999                                   SBTTL  CRC-AST ROUTINE
 1000
 1001
 1002
 1003                           ;      ************************************************************
 1004                           ;      ROUTINE CHECKS FOR DEVICE I/O ERRORS UPON A
 1005                           ;      REQUEST FOR BAD CRC GENERATION,LOGS THE ERRORS
 1006                           ;      AND SET AN INDICATION FOR AST
 1007                           ;      COMPLETION...
 1008                           ;      ************************************************************
 1009
 1010
 1011
 1012
 1013 013264  005737  005142'        CRCAST  TST     ISTATO                  ;TEMPORARY
 1014 013270  100025                         BPL     1$                      ;IF NO RETRY THEN EXIT
 1015 013272  013703  007224'                MOV     ERRPNT,R3               ;GET POINTER TO ERROR TABLE
 1016 013276  020327  001742                 CMP     R3,#994                 ;ERROR TABLE FILLED?
 1017 013302  103020                         BHIS    1$                      ;IF FILLED EXIT
 1018 013304  112763  000006  005252'        MOVB    #6,ERRTBL+0(R3)         ;LOAD ERROR TYPE
 1019 013312  116263  000306' 005253'        MOVB    UNTBL+0(R2),ERRTBL+1(R3),LOAD TERMINAL NUMBER
 1020 013320  116263  000307' 005254'        MOVB    UNTBL+1(R2),ERRTBL+2(R3),LOAD CONTROLLER NUMBER
 1021 013326  062703  000006                 ADD     #6,R3                   ;UPDATE ERROR POINTER
 1022 013332  112763  177777  005252'        MOVB    #-1,ERRTBL(R3)          ;END OF ERRORTABLE FLAG
 1023 013340  010337  007224'                MOV     R3,ERRPNT               ;SAVE ERROR POINTER
 1024 013344  005726                   1$    TST     (SP)+                   ;AS PER RSX--ASTX$S
 1025 013346  012737  000001  007262'        MOV     #1,CRCFLG               ;SET AST INDICATOR
 1026 013354                                 ASTX$S                          ;RETURN
      013354  012746                         MOV     (PC)+,-(SP)
      013356     163     001                  BYTE    115.,1
      013360  104377                         EMT     0'377
 1027
```

```
                        SBTTL  AST SERVICE ROUTINE


                ;       ************************************************************
                ;       THIS ROUTINE CALCULATES THE TERMINAL NUMBER OF THE
                ;       DEVICE WHICH INTERRUPTED,PROCESS ANY ERRORS FROM
                ;       THOSE DEVICES. (I E TIME-OUT,DEVICE NOT READY,AND DATA
                ;        ERRORS E T.C)RECORDS ANY AND ALL
                ;       TERMINALS WHICH GOES OFFLINE,OR RETURNS ONLINE AND
                ;       FREES AN OUTPUT BUFFER
                        ************************************************************


                ;
013362                  AST     POP     ISB                     ;GET ADDRESS OF I/O STATUS BLOCK
013362  012637  005202'
013366  011637  005204'          MOV     (SP),IODSW              ;GET DSW PRIOR TO AST ROUTINE
013372                          PUSH    <R0,R1,R2,R3,R4,R5>     ;SAVE REGISTERS
013372  010046
013374  010146
013376  010246
013400  010346
013402  010446
013404  010546
013406  013705  005202'          MOV     ISB,R5                  ;SAVE ADDRESS OF I/O STATUS BLOCK
013412  162705  005142'          SUB     #ISTAT0,R5              ;CALCULATE TERMINAL WHICH INTERRUPTED
013416  006205                  ASR     R5                      ;GET BUFFER CONTAING INTERRUPTING TERMINAL
013420  010537  007260'          MOV     R5,TSTBUF               ;SAVE BUFFER FOR DATA COMPARE
013424  016505  001016'          MOV     OBUFF(R5),R5            ;STORE ADRRESS OF OUTPUT BUFFER
013430  111501                  MOVB    (R5),R1                 ;GET AND SAVE TERMINAL SB
013432  105015                  CLRB    (R5)                    ;CLEAR THE 1'ST BYTE IN THE BUFFER
013434  110105                  MOVB    R1,R5                   ;STORE SB IN R5
013436  005004                  CLR     R4                      ;OFFLINE TABLE POINTER
013440  120564  000506'  22$     CMPB    R5,DEVTAB(R4)           ;FIND OFFSET FOR OFFLINE TABLE
013444  001402                  BEQ     21$                     ;BRANCH IF FOUND
013446  005724                  TST     (R4)+                   ;UPDATE POINTER
013450  000773                  BR      22$                     ;CONTINUE
013452  006204          21$     ASR     R4                      ;ADJUST OFFLINE TABLE POINTER
013454  013701  005202'          MOV     ISB,R1                  ;GET I/O STAT
013460  013703  007224'          MOV     ERRPNT,R3               ;GET ERROR POINTER
013464  020327  001742           CMP     R3,#994                 ;CHECK ERROR TABLE
013470  103153                  BHIS    8$                      ;BRANCH IF ERROR POINTEP = 994
013472  005737  005204'          TST     IODSW                   ;CHECK DSW ERRORS
013476  100423                  BMI     1$                      ;BRANCH IF ERROR
013500  105711                  TSTB    (R1)                    ;CHECK IOSTAT ERRORS
013502  100431                  BMI     2$                      ;BRANCH IF ERROR
013504  105764  000706'          TSTB    OFFLIN(R4)              ;WAS DEVICE OFFLINE?
013510  001411                  BEQ     4$                       BRANCH IF NO-------
013512  105064  000706'          CLRB    OFFLIN(R4)              ;MAKE IT ON LINE
013516  112763  000005  005252'  MOVB    #5,ERRTBL+0(R3)         ;STORE ON-LINE CODE
013524  110563  005253'          MOVB    R5,ERRTBL+1(R3)         ;STORE TERM#
013530  062703  000006           ADD     #6,R3                   ;ADJUST POINTER
013534  023727  007224'  001742  4$      CMP     ERRPNT,#994     ;CHECK ERROR TABLE
013542  103126                  BHIS    8$                      ;BRANCH IF FILLED
013544  000446                  BR      3$                      ;TEMPORARY BYPASS
```

SEQ 0060

```
1080
1081
1082
1083                    ;PETRY ROUTINE ERROR STUB INSERTED HERE**************************
1084
1085
1086
1087
1088
1089
1090
1091
1092
```

```
1094
1095                                        ;PROCESS        DSW     ERRORS
1096
1097 013546  112763  000003  005252' 1$      MOVB    #3,ERRTBL+0(R3)         ;MOV DSW ERROP-TYPE TO ERROR TABLE
1098 013554  110563  005253'                 MOVB    R5,ERRTBL+1(R3)         ;LOAD TERMINAL# IN ERROR TABLE
1099 013560  062703  000006                 ADD     #6,R3                   ;UPDATE ERROR POINTER
1100 013564  000515                         BR      8$                      ;BRANCH TO ASTX$S EXIT
1101
1102
1103
1104
1105
1106
1107                                        ;PROCESS IOSTAT ERRORS
1108
1109 013566  121127  000277          2$      CMPB    (R1),#277               ;OFFLINE
1110 013572  001012                         BNE     11$
1111 013574  105764  000706'                 TSTB    OFFLIN(R4)              ;ALREADY OFFLINE?
1112 013600  001107                         BNE     8$                      ;YES - JUST IGNORE
1113 013602  112764  000001  000706'         MOVB    #1,OFFLIN(R4)           ;MARK IT OFFLINE
1114 013610  112763  000010  005252'         MOVB    #10,ERRTBL+0(R3)
1115 013616  000403                         BR      12$
1116 013620  112763  000001  005252  11$     MOVB    #1,ERRTBL+0(R3)         ;PUT HARD ERROR TYPE IN ERROR TABLE
1117 013626  110563  005253'          12$     MOVB    R5,ERRTBL+1(R3)         ;LOAD TERMINAL# INTO ERROR TABLE
1118 013632  112163  005255'                 MOVB    (R1)+,ERRTBL+3(R3)      ;LOAD 1ST WORD IOSTAT
1119 013636  005201                         INC     R1                      ;GET 2'ND I/O STAT WORD
1120 013640  111163  005257'                 MOVB    (R1),ERRTBL+5(R3)       ;LOAD 2ND WORD IOSTAT
1121 013644  062703  000006                 ADD     #6,R3                   ;UPDATE ERROR POINTER
1122 013650  005741                         TST     -(R1)                   ;CHECK 1ST WORD OF IOSTAT
1123 013652  122711  177765                 CMPB    #-13,(R1)               ;IS IOSTAT = -13?
1124 013656  001726                         BEQ     4$                      ;IF YES CHECK ERROR TABLE
1125 013660  000457                         BR      8$                      ;ASTX$S - EXIT
1126
```

```
                                        ;FROM RETRY ERROR
                                        ;COMPARE I/O DATA BUFFER
013662  013701  007260'          3$    MOV     TSTBUF,R1           ;GET OUTPUT BUFFER POINTER
013666  016102  001016'          5$    MOV     OBUFF(R1),R2        ;STARTING ADDRESS OF OUTPUT BUFFER
013672  016104  003102'                MOV     INBUFF(R1),R4       ;STARTING ADDRESS INPUT BUFFER
013676  062702  000003                ADD     #3,R2               ;GET OUTPUT DATA BYTE
013702  012737  000012  007232'        MOV     #10.,ERRTMP         ;I/O DATA POINTER
013710  005001                        CLR     R1                  ;CLEAR BYTE COUNTER
013712  020137  005240'          6$    CMP     R1,LENGTH           ;AT END OF RECORD?
013716  001440                        BEQ     8$                  ;YES-EXIT
013720  005201                        INC     R1                  ;NO-UPDATE BYTE COUNTER BY ONE
013722  122224                        CMPB    (R2)+,(R4)+         ;I/O DATA OK?
013724  001772                        BEQ     6$                  ;GET NEXT DATA & BRANCH IF YES
013726  005737  007232'                TST     ERRTMP              ;MORE THAN 10 I/O ERRORS STORED?
013732  001432                        BEQ     8$                  ;IF YES - EXIT
013734  020327  001742                CMP     R3,#994             ;ERROR TABLE FILLED?
013740  103027                        BHIS    8$                  ;BRANCH IF YES - EXIT
013742  112763  000004  005252'        MOVB    #4,ERRTBL+0(R3)     ;FILL ERROR-TYPE IN ERROR TABLE
013750  110563  005253'                MOVB    R5,ERRTBL+1(R3)     ;FILL TERMINAL# IN ERROR TABLE
013754  013700  005202'                MOV     ISB,R0
013760  116063  000002  005254'        MOVB    2(R0),ERRTBL+2(R3)  ;FILL BYTE COUNT
013766  110163  005255'                MOVB    R1,ERRTBL+3(R3)     ;SET BYTE COUNTER
013772  005337  007232'                DEC     ERRTMP              ;I/O DATA ERROR INDICATOR
013776  124244                        CMPB    -(R2),-(R4)         ;DECREMENT DATA POINTERS
014000  111263  005256'                MOVB    (R2),ERRTBL+4(R3)   ;FILL GOOD DATA
014004  111463  005257'                MOVB    (R4),ERRTBL+5(R3)   ;FILL BAD DATA
014010  122224                        CMPB    (R2)+,(R4)+         ;INCREMENT DATA POINTER
014012  062703  000006                ADD     #6,R3
014016  000735                        BR      6$
014020  112763  177777  005252' 8$    MOVB    #-1,ERRTBL(R3)      ;SET TABLE TERMINATOR
014026  010337  007224'                MOV     R3,ERRPNT           ;STORE TABLE POINTER
014032                                POP     <R5,R4,R3,R2,R1,R0>
014032  012605
014034  012604
014036  012603
014040  012602
014042  012601
014044  012600
014046                        ASTX$S                              ;EXIT
014046  012746                MOV     (PC)+,-(SP)
014050     163      001        .BYTE   115.,1
014052  104377                EMT      0<377>
```

```
                                        SBTTL  PRINT MODULE
                                ;++
                                ;PRINT MODULE
                                ;
                                ;INPUT  R3 = POINTER TO ERROR PACKET
                                ;
                                ;OUTPUT R3 = POINTER TO NEXT PACKET
                                ;
                                ;ERROR PACKET IS SIX BYTES IN THE FOLLOWING ORDER
                                ;
                                ;       BYTE            MEANING
                                ;
                                ;         0             ERROR CODE
                                ;         1             TERMINAL NUMBER
                                ;         2             RECEIVED BYTE COUNT
                                ;         3             ERROR BYTE COUNTER
                                ;         4             GOOD DATA
                                ;         5             BAD DATA
                                ;
                                ;NOTE    ROUTINE PRINTS REPEATING ERRORS FOR THE SAVE TERMINAL AND SAME TYPE OF ERRROR
                                ;
                                ;--

                                BGNMSG  MSGCD4
                                MSGCD4
                                        PRINTB  #FORM1,<B,ERRTBL+0(R3)>,<B,ERRTBL+1(R3)>,<B,ERRTBL+2(R3)>
014054  005046                                  CLR     -(SP)
014056  156316  005254'                         BISB    ERRTBL+2(R3),(SP)
014062  005046                                  CLR     -(SP)
014064  156316  005253'                         BISB    ERRTBL+1(R3),(SP)
014070  005046                                  CLR     -(SP)
014072  156316  005252'                         BISB    ERRTBL+0(R3),(SP)
014076  012746  014206'                         MOV     #FORM1,-(SP)
014102  012746  000004                         MOV     #4,-(SP)
014106  010600                                  MOV     SP,R0
014110  104014                                  EMT     C$PNTB
014112  062706  000012                         ADD     #12,SP

                                2$      PRINTB  #FORM3,<B,ERRTBL+3(R3)>,<B,ERRTBL+4(R3)>,<B,ERRTBL+5(R3)>
014116  005046                                  CLR     -(SP)
014120  156316  005257'                         BISB    ERRTBL+5(R3),(SP)
014124  005046                                  CLR     -(SP)
014126  156316  005256'                         BISB    ERRTBL+4(R3),(SP)
014132  005046                                  CLR     -(SP)
014134  156316  005255'                         BISB    ERRTBL+3(R3),(SP)
014140  012746  014336'                         MOV     #FORM3,-(SP)
014144  012746  000004                         MOV     #4,-(SP)
014150  010600                                  MOV     SP,R0
014152  104014                                  EMT     C$PNTB
014154  062706  000012                         ADD     #12,SP
014160  062703  000006                 ADD     #6,R3                   ;POINT TO NEXT PACKET
014164  122763  000004  005252'         CMPB    #4,ERRTBL(R3)           ;CHECK NEXT ERROR TYPE
014172  001004                         BNE     3$                      ;BRANCH IF NOT
014174  126363  005253'  005245'       CMPB    ERRTBL+1(R3),ERRTBL-5(R3)       ;TERMINAL NUMBER
014202  001745                         BEQ     2$                      ;SAME TERM, OUTPUT ERROR
014204                          3$
014204                          ENDMSG
```

```
        014204                          L10010
        014204  104023                          EMT     C$MSG
  1198                                  ,
```

```
1200                                    ;
1201                                    ;
1202                                    ;       ************************************************************
1203                                    ;       GLOBAL ASCII MESSAGES USED BY THE PROGRAM
1204                                    ;       ************************************************************
1205                                    ;
1206                                    ;
1207  014206    045     101     011     FORM1.  ASCII  /%A     ERRORTYPE=%02%A SB#=%D2%A RECEIVED BYTE COUNT=%D3/
      014211    105     122     122
      014214    117     122     124
      014217    131     120     105
      014222    075     045     117
      014225    062     045     101
      014230    040     123     102
      014233    043     075     045
      014236    104     062     045
      014241    101     040     122
      014244    105     103     105
      014247    111     126     105
      014252    104     040     102
      014255    131     124     105
      014260    040     103     117
      014263    125     116     124
      014266    075     045     104
      014271    063
1208  014272    045     116     045             ASCIZ  /%N%A   BYTE-NUMBER     GDDATA          BDDATA/
      014275    101     011     102
      014300    131     124     105
      014303    055     116     125
      014306    115     102     105
      014311    122     040     011
      014314    107     104     104
      014317    101     124     101
      014322    011     011     040
      014325    040     040     102
      014330    104     104     101
      014333    124     101     000
1209  014336    045     116     045     FORM3   ASCIZ  "%N%S14%D3%S9%03%S15%03%N"
      014341    123     061     064
      014344    045     104     063
      014347    045     123     071
      014352    045     117     063
      014355    045     123     061
      014360    065     045     117
      014363    063     045     116
      014366    000
1210                                    EVEN
```

```
1212 014370                          BGNMSG  MSGCD1
     014370                          MSGCD1
1213 014370  116300  005255'                MOVB    ERRTBL+3(R3),R0
1214 014374  005400                         NEG     R0
1215 014376                                 PRINTB  #FMT01,<B,ERRTBL+0(R3)>,<B,ERRTBL+1(R3)>,<R0>,<B,ERRTBL+5(R3)>
     014376  005046                                 CLR     -(SP)
     014400  156316  005257'                       BISB    ERRTBL+5(R3),(SP)
     014404  010046                                 MOV     R0,-(SP)
     014406  005046                                 CLR     -(SP)
     014410  156316  005253'                       BISB    ERRTBL+1(R3),(SP)
     014414  005046                                 CLR     -(SP)
     014416  156316  005252'                       BISB    ERRTBL+0(R3),(SP)
     014422  012746  014450'                       MOV     #FMT01,-(SP)
     014426  012746  000005                       MOV     #5,-(SP)
     014432  010600                               MOV     SP,R0
     014434  104014                               EMT     C$PNTB
     014436  062706  000014                       ADD     #14,SP
1216                                                                    ;NOTE THE DEVICE WENT
1217                                                                    ;OFF-LINE
1218 014442  062703  000006                 ADD     #6,R3
1219                                                                    ;UPDATE THE POINTER
1220 014446                          ENDMSG
     014446                          L10011
     014446  104023                         EMT     C$MSG
1221 014450     045     101     040 FMT01   .ASCII /%A ERROR-TYPE=%D2%A SB#=%D3/
     014453     105     122     122
     014456     117     122     055
     014461     124     131     120
     014464     105     075     045
     014467     104     062     045
     014472     101     040     123
     014475     102     043     075
     014500     045     104     063
1222 014503     045     101     040         .ASCIZ  /%A RSX-11M STATUS CODE= -%D3%A BYTE COUNT=%D4%N/
     014506     122     123     130
     014511     055     061     061
     014514     115     040     123
     014517     124     101     124
     014522     125     123     040
     014525     103     117     104
     014530     105     075     040
     014533     055     045     104
     014536     063     045     101
     014541     040     102     131
     014544     124     105     040
     014547     103     117     125
     014552     116     124     075
     014555     045     104     064
     014560     045     116     000
1223 014563     045     116     045 M200    .ASCIZ  "%N%A RSX-11M HARD ERROR DETECTED %N"
     014566     101     040     122
     014571     123     130     055
     014574     061     061     115
     014577     040     110     101
     014602     122     104     040
     014605     105     122     122
     014610     117     122     040
```

```
                014613  104     105     124
                014616  105     103     124
                014621  105     104     072
                014624  045     116     000
     1224       014627  045     116     045     CRCMSG  ASCII   "%N%ABAD CRC FROM TERMINAL%D3"
                014632  101     102     101
                014635  104     040     103
                014640  122     103     040
                014643  106     122     117
                014646  115     040     124
                014651  105     122     115
                014654  111     116     101
                014657  114     045     104
                014662  063
     1225       014663  045     101     054             ASCIZ   "%A, WAS NOT DETECTED BY CONTROLLER %D1%N"
                014666  040     127     101
                014671  123     040     116
                014674  117     124     040
                014677  104     105     124
                014702  105     103     124
                014705  105     104     040
                014710  102     131     040
                014713  103     117     116
                014716  124     122     117
                014721  114     114     105
                014724  122     040     045
                014727  104     061     045
                014732  116     000
     1226                                               EVEN
```

```
 1228 014734                                    BGNMSG  MSGCD2
      014734                                    MSGCD2
 1229 014734                                            PRINTB  #FMT02,<B,ERRTBL+0(R3)>,<B,ERRTBL+1(R3)>,<B,ERRTBL+2(R3)>
      014734  005046                                           CLR     -(SP)
      014736  156316  005254'                                 BISB    ERRTBL+2(R3),(SP)
      014742  005046                                           CLR     -(SP)
      014744  156316  005253'                                 BISB    ERRTBL+1(R3),(SP)
      014750  005046                                           CLR     -(SP)
      014752  156316  005252'                                 BISB    ERRTBL+0(R3),(SP)
      014756  012746  015004'                                 MOV     #FMT02,-(SP)
      014762  012746  000004                                   MOV     #4,-(SP)
      014766  010600                                           MOV     SP,R0
      014770  104014                                           EMT     C$PNTB
      014772  062706  000012                                   ADD     #12,SP
 1230 014776  062703  000006                            ADD     #6,R3
 1231 015002                                    ENDMSG
      015002                                    L10012
      015002  104023                                   EMT     C$MSG
 1232 015004     045     101     040    FMT02    ASCII   /%A ERROR-TYPE=%D2%A SB#=%D3/
      015007     105     122     122
      015012     117     122     055
      015015     124     131     120
      015020     105     075     045
      015023     104     062     045
      015026     101     040     123
      015031     102     043     075
      015034     045     104     063
 1233 015037     045     101     102            ASCIZ   /%ABYTE COUNT=%D4%N/
      015042     131     124     105
      015045     040     103     117
      015050     125     116     124
      015053     075     045     104
      015056     064     045     116
      015061     000
 1234                                           EVEN
```

```
                                                ;GLOBAL TEXT
                
1238 015062  124  111  115  MSG01   ASCIZ  /TIME-OUT ERROR  TERMINAL IS OFF LINE/
     015065  105  055  117
     015070  125  124  040
     015073  105  122  122
     015076  117  122  072
     015101  040  124  105
     015104  122  115  111
     015107  116  101  114
     015112  040  111  123
     015115  040  117  106
     015120  106  040  114
     015123  111  116  105
     015126  000
1239 015127  122  105  124  MSG02   ASCIZ  /RETRY ERROR  /
     015132  122  131  040
     015135  105  122  122
     015140  117  122  072
     015143  040  000
1240 015145  104  123  127  MSG03   ASCIZ  /DSW ERROR  DIRECTIVE WAS NOT ACCEPTED/
     015150  040  105  122
     015153  122  117  122
     015156  072  040  104
     015161  111  122  105
     015164  103  124  111
     015167  126  105  040
     015172  127  101  123
     015175  040  116  117
     015200  124  040  101
     015203  103  103  105
     015206  120  124  105
     015211  104  000
1241 015213  104  101  124  MSG04   ASCIZ  /DATA COMPARISON ERROR /
     015216  101  040  103
     015221  117  115  120
     015224  101  122  111
     015227  123  117  116
     015232  040  105  122
     015235  122  117  122
     015240  072  000
1242 015242  045  116  045  ONMSG   ASCIZ  "%N%ATERMINAL %B  %D2%A  BACK ON LINE%N"
     015245  101  124  105
     015250  122  115  111
     015253  116  101  114
     015256  040  123  102
     015261  056  040  045
     015264  104  062  045
     015267  101  040  040
     015272  102  101  103
     015275  113  040  117
     015300  116  040  114
     015303  111  116  105
     015306  045  116  000
1243 015311  045  116  045  EOP     ASCII  "%N%AACTIVE PASSES SINCE LAST REPORT=%D6"
     015314  101  101  103
     015317  124  111  126
```

```
        015322  105  040  120
        015325  101  123  123
        015330  105  123  040
        015333  123  111  116
        015336  103  105  040
        015341  114  101  123
        015344  124  040  122
        015347  105  120  117
        015352  122  124  075
        015355  045  104  066
  1244  015360  045  101  054          ASCII  "%A,  ACTIVE TERMINALS=%D3"
        015363  040  040  101
        015366  103  124  111
        015371  126  105  040
        015374  124  105  122
        015377  115  111  116
        015402  101  114  123
        015405  075  045  104
        015410  063
  1245  015411  045  101  054          ASCIZ  "%A, TIME=%D2%A %Z2%N'
        015414  040  124  111
        015417  115  105  075
        015422  045  104  062
        015425  045  101  072
        015430  045  132  062
        015433  045  116  000
  1246  015436  045  116  045  LFCR   ASCIZ  "%N%N"
        015441  116  000
  1247
  1248                               ;
  1249                               ;       ************************************************************
  1250                               ;       THE FOLLOWING CALL GENERATES THE FIRST EVEN ADDRESS NOT
  1251                               ;       USED BY THE PROGRAM
  1252                               ;       ************************************************************
  1253                               ;
  1254  015443                       LASTAD
                                                 EVEN
        015444                       L$LAST
  1255                                     BLKW   2024                  ;FREE MEMORY AREA
  1256          000001                     END
```

```
ALUDPB  000244R      C$GTIM= 000066      FORM3   014336R      IBUF2   003522R      L$DISP  007326RG
AST     013362R      C$INIT= 000011      F$AU  = 000015      IBUF3   003722R      L$DR    000202RG
A LULU= 000002       C$INLP= 000020      F$BGN = 000040      IBUF4   004122R      L$DRCT  000062RG
A LUNA= 000004       C$KWOF= 000042      F$CLEA= 000007      IBUF5   004322R      L$DRS   000064RG
A LUNU= 000006       C$KWON= 000041      F$DU  = 000016      IBUF6   004522R      L$DRST  000206RG
BFFPT   005242R      C$LGOF= 000027      F$END = 000041      IBUF7   004722R      L$DTP   000040RG
BIT00 = 000001       C$LGON= 000026      F$HARD= 000004      INBUFF  003102R      L$DUT   000072RG
BIT01 = 000002       C$LOOP= 000100      F$HW  = 000013      INIT    010536R      L$DVTY  000210RG
BIT02 = 000004       C$MANI= 000065      F$INIT= 000006      IODSW   005204R      L$EF    000024RG
BIT03 = 000010       C$MPME= 000051      F$MOD = 000000      IOSB    005122R      L$EXP1  000032RG
BIT04 = 000020       C$MSG = 000023      F$MSG = 000011      IOST    000276R      L$EXP2  000034RG
BIT05 = 000040       C$PNTB= 000014      F$PWR = 000017      IO AST= 000012       L$EXP3  000036RG
BIT06 = 000100       C$PNTF= 000017      F$RPT = 000012      IO ATT= ****** GX    L$HARD  007266RG
BIT07 = 000200       C$PNTN= 000013      F$SEG = 000003      IO A1 = 000014       L$HPCP  000046RG
BIT08 = 000400       C$PNTS= 000016      F$SOFT= 000005      IO A2 = 000016       L$HPTP  000056RG
BIT09 = 001000       C$PNTX= 000015      F$SRV = 000010      IO A3 = 000020       L$HW    007314RG
BIT10 = 002000       C$POIN= 000040      F$SUB = 000002      IO A4 = 000022       L$ICP   000042RG
BIT11 = 004000       C$PWR = 000073      F$SW  = 000014      IO A5 = 000024       L$INIT  007330RG
BIT12 = 010000       C$QIO = 000377      F$TEST= 000001      IO DET= ****** GX    L$LADP  000076RG
BIT13 = 020000       C$REFG= 000064      GETCNT  000100R     IO EFN= 000006       L$LAST  015444RG
BIT14 = 040000       C$RELA= 000052      GETERM  000132R     IO FUN= 000002       L$MREV  000012RG
BIT15 = 100000       C$RELM= 000053      GLUDPB  000254R     IO IST= 000010       L$NAME  000000RG
BUF     005206R      C$RELO= 000037      G$EXCP= 000400      IO LUN= 000004       L$PWRU  000074RG
CRCAST  013264R      C$REQT= 000060      G$HILI= 000002      IO RXP= 003400       L$REPP  000054RG
CRCFLG  007262R      C$RESE= 000040      G$LOLI= 000001      IO WPC= 002400       L$REV   000010RG
CRCMSG  014627R      C$PEVI= 000001      G$NO  = 000000      ISB     005202R      L$RPT   007322RG
C$ABRT= 000021       C$RPT = 000025      G$OFFS= 000400      ISTAT0  005142R      L$SOFT  007320RG
C$ADR = 000020       C$SEFG= 000062      G$OFSI= 000376      ISTAT1  005146R      L$SPC   000030RG
C$AU  = 000071       C$SPRI= 000046      G$PRMA= 000001      ISTAT2  005152R      L$SPCP  000050RG
C$BRK = 000022       C$SVEC= 000044      G$PRMD= 000002      ISTAT3  005156R      L$SPTP  000060RG
C$BSEG= 000004       C$UNBU= 000035      G$PRML= 000000      ISTAT4  005162R      L$STA   000066RG
C$BSUB= 000002       C$WEFG= 000063      G$RADA= 000140      ISTAT5  005166R      L$SW    007322RG
C$BUFF= 000034       C$WTM = 000030      G$RADB= 000000      ISTAT6  005172R      L$TIML  000022RG
C$CEFG= 000061       C$WTU = 000031      G$RADD= 000040      ISTAT7  005176R      L$TIMU  000020RG
C$CLEA= 000012       C1      011760RG    G$RADF= 000200      I$AU  = 000041       L$TIM1  000016RG
C$CLP1= 000006       DATA  = 000003      G$RADL= 000120      I$CLN = 000041       L$UNIT  000014RG
C$CVEC= 000043       DEVPTR  000302R     G$RADO= 000020      I$DU  = 000041       L10000  007312R
C$DCLN= 000057       DEVTAB  000506R     G$RADT= 000100      I$HRD = 000041       L10001  007316R
C$DODU= 000070       DSTW1   007250R     G$XFER= 000004      I$INIT= 000041       L10002  007320R
C$DRPT= 000024       DSTW2   007252R     G$YES = 000010      I$MOD = 000041       L10003  007322R
C$DSTI= 000033       DSTW3   007254R     G LUBA= 000004      I$MSG = 000041       L10004  007322R
C$DU  = 000072       D3      012256RG    G LUBL= 000006      I$PWR = 000041       L10005  010672R
C$EDIT= 000004       D3A     012272R     G LUCW= 000004      I$RPT = 000041       L10006  011742R
C$ENTI= 000032       EOP     015311R     G LUFB= 000003      I$SEG = 000041       L10007  013262R
C$ERDF= 000002       ERRBRT  000262R     G LULU= 000002      I$SFT = 000041       L10010  014204R
C$ERHR= 000003       ERRPNT  007224R     G LUNA= 000000      I$SRV = 000041       L10011  014446R
C$ERSF= 000001       ERRTBL  005252R     G LUNU= 000002      I$SUB = 000041       L10012  015002R
C$ERSO= 000004       ERRTMP  007232R     G TICP= 000016      I$TST = 000041       MABORT  010767R
C$ESCA= 000010       ETB1    007236P     G TICT= 000014      LENGTH  005240R      MRKAST  011744R
C$ESEG= 000005       ETB2    007237R     G TIDA= 000004      LFCR    015436R      MSG     011101R
C$ESUB= 000003       ETB3    007240R     G TIHR= 000006      LUN   = 000011      MSGCD1  014370RG
C$ETST= 000001       ETB4    007242P     G TIMI= 000010      LUNS    005244R      MSGCD2  014734RG
C$EXIT= 000036       E1      012054RG    G TIMO= 000002      L$AUT   000070RG     MSGCD4  014054RG
C$GMAN= 000056       E2      012134RG    G TISC= 000012      L$CCP   000044RG     MSGO    010674R
C$GMEM= 000050       FMTO1   014450R     G TIYR= 000000      L$CLEA  013262RG     MSGO1   015062P
C$GPHR= 000055       FMTO2   015004R     IBUFO   003122R     L$DEPO  000011RG     MSGO2   015127P
C$GPRI= 000045       FORM1   014206R     IBUF1   003322R     L$DEVP  000052RG     MSGO7   015145P
```

```
DZKCH  MACRO M11    10-OCT-77 10 16  PAGE 29-3                                  SEQ 0072
SYMBOL TABLE

M$GO4   015213R        O$BGNR= 000000        RTRY1   007244R        TSTBUF  007260R        T$$HAR= 010000
MSG1    011020R        O$BGNS= 000000        RTRY2   007245R        T$ARGC= 000004         T$$4W = 010001
MSG2    011162R        O$DU  = 000000        RTRY3   007246R        T$CODE= 000052         T$$INI= 010005
MSG3    011306R        O$GNSW= 000000        SB      007256R        T$ERCO= 000064         T$$MSG= 010012
M200    014563R        O$POIN= 000001        SCBADR  001006R        T$ERRN= 000003         T$$RPT= 010004
NP    = 000001         O$PWR = 000000        SCBCNT  001010R        T$EXCP= 000000         T$$SOF= 010002
NUMDEV  000304R        PASFG   005236R       STADR   000000RG       T$HILI= 000077         T$$SW = 010003
OBUFF   001016R        PASS    005246R       START   011370R        T$LOLI= 000001         T$$TES= 010006
OBUF0   001041R        PATPT   007226R       SVCCNT= 177777         T$LSYM= 010000         T1      011370RG
OBUF1   001245R        PATTRN  005226R       SVCGBL= 000001         T$MCAL= 000000         UNITS   005250R
OBUF2   001451R        PCBADR  001012R       SVCINS= 000001         T$NEST= 177777         UNTBL   000306R
OBUF3   001655R        PCBCNT  001014R       SVCSTK= 177777         T$NSKO= 000011         X$ALWA= 000000
OBUF4   002061R        Q10DPB  000214R       SVCSUB= 000001         T$SAVL= 177777         X$FALS= 000040
OBUF5   002265R        Q 10AE= 000012        SVCTAG= 000001         T$SEGL= 177777         X$OFFS= 000400
OBUF6   002471R        Q 10EF= 000006        SVCTST= 000001         T$SUBN= 000000         X$TRUE= 000020
OBUF7   002675R        Q 10FN= 000002        S$LSYM= 010000         T$TAGL= 177777         $DSW  = ****** GX
OFFLIN  000706R        Q 10LU= 000004        TEMP    007234R        T$TAGN= 010013         $$$ARG= 000004
ONMSG   015242R        Q 10PL= 000014        TEM3    007230R        T$TEMP= 000011         $$$OST= 000020
O$APTS= 000000         Q 10PR= 000007        TERM    007222R        T$TEST= 000001         $$$T1 = 000002
O$AU  = 000000         Q 10SB= 000010        TMTF    007235R        T$$CLE= 010007         $$$T2 = 000250R

  ABS   000000      000
        025364      001
ERRORS DETECTED   0

VIRTUAL MEMORY USED   22568 WORDS  ( 89 PAGES)
DYNAMIC MEMORY   21140 WORDS  ( 81 PAGES)
ELAPSED TIME   00 02 58
DZKCH/DZKCH=DIAGSUPER/ML/DZKCH
```