

The image shows a grid of 16 small diagnostic test screens, arranged in 4 rows and 4 columns. Each screen displays various data points, including memory addresses, hexadecimal values, and status indicators. The screens are arranged in a grid, with each screen containing a different set of diagnostic information. The data is presented in a structured, tabular format, typical of early computer diagnostic tools. The screens are arranged in a grid, with each screen containing a different set of diagnostic information. The data is presented in a structured, tabular format, typical of early computer diagnostic tools.

11
12
13

PRODUCT CODE: MAINDEC-11-DZKCI-A-D
PRODUCT NAME: REMOTE TERMINAL TESTER
PRODUCT DATE: DECEMBER 1977
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: CLEM WALSH

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

The software described in this document is furnished to the purchaser under a license for use on a single computer system and can be copied (with inclusion of Digital's copyright notice) only for use in such system, except as may otherwise be provided in writing by Digital.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

Copyright (C) 1977 Digital Equipment Corporation

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	HARDWARE REQUIREMENTS
3.0	SOFTWARE REQUIREMENTS
4.0	PROGRAM GENERATION
5.0	PROGRAM CONSIDERATIONS
6.0	TESTING PREQUISTES
7.0	XXDP
8.0	ACT/APT
9.0	MEMORY MANAGEMENT
10.0	SWITCH REGISTER FUNCTIONS
11.0	PROGRAM CONTROL PROCEDURES
12.0	LOADING
13.0	PROGRAM PARAMETER SELECTION
14.0	TEST DESCRIPTION
15.0	ERROR REPORTING
16.0	EXECUTION TIME
17.0	PROGRAM TERMINATION

1.0 ABSTRACT

THIS PROGRAM PERFORMS ON LINE REMOTE DIAGNOSTIC TESTS, BY STARTING THE INTERNAL DIAGNOSTICS IN SELECTED RT801, 803, OR 805 TERMINALS, CONNECTED TO THE SERIAL BUS FOR THE DPM SYSTEM. WITHIN THE PROGRAM, PROVISION IS MADE TO REPORT ALL ERRORS GENERATED BY EACH TERMINAL. STRUCTURALLY, THE PROGRAM CONSISTS OF ONE MAIN TEST, AND IS ORGANIZED WITH RESPECT TO THAT TEST.

2.0 HARDWARE REQUIREMENTS

PDP11 FAMILY OF COMPUTERS, EXCEPT THE LSI-11
CONSOLE TELETYPE
16K MEMORY
ISB11A (SERIAL BUS CONTROLLER)
SERIAL BUS
1-63 FACTORY DATA COLLECTION TERMINALS

3.0 SOFTWARE REQUIREMENTS

RSX-11M DISK BASED REAL TIME 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 DZKCI.TSK. IT IS GENERATED FROM THE SOURCE FILE DZKCI.MAC BY USING RSX-11M INDIRECT FILES. AN INDIRECT FILE DZKCI.CMD CONTAINS A LIST OF MCR COMMANDS, AND IS INVOKED BY TYPING:

```
"@LDZKCI"
```

UPON WHICH THE CONSOLE WILL PRINT THE FOLLOWING:

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

```
TKB @TKBDZKCI  
DZKCI.TSK,DZKCI.MAP=DZKCI.OBJ,DIAGSUPER  
/  
UNITS=75  
STACK=512  
//
```

THE FOLLOWING IS AN EXPLANATION OF THE CONSOLE PRINTOUT:

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

THIS ASSEMBLES THE SOURCE FILE DZKCI.MAC WITH THE DIAGNOSTIC SUPERVISOR MACRO PACKAGE DIAGSUPER/ML, INTO AN OBJECT FILE DZKCI.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

```
DZKCI.TSK,DZKCI.MAP=DZKCI.OBJ,DIAGSUPER
```

THIS TASK-BUILDS THE OBJECT FILE DZKCI.OBJ WITH THE DIAGNOSTIC SUPERVISOR OBJECT LIBRARY DIAGSUPER, INTO A TASK IMAGE DZKCI.TSK TO BE

RUN IN SYSTEM'S MEMORY. DZKCI. 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 "TKBDZKCI. 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 SUPER SUBSET OF THE BASIC STAND ALONE VERSION.

5.0 PROGRAM CONSIDERATIONS

THIS IS A FUNCTION LEVEL PROGRAM, DESIGNED FOR RSX-11M AND SUPERVISOR COMPATIBILITY, AND CREATED FOR DESIGN ENGINEERING, MANUFACTURING, OR FIELD ENGINEERING USAGE. AS A FUNCTION LEVEL PROGRAM, IT STARTS TERMINAL DIAGNOSTICS ON LINE, AND REPORTS ALL DIAGNOSTIC ERRORS, FOR EACH TERMINAL ON THE SERIAL BUS SYSTEM.

5.1 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 QIOS DIRECTIVES USED IN RSX-11M. IT SHOULD BE NOTED, HOWEVER, THAT RUNNING UNDER OPERATING SYSTEM CONTROL, STABLE PROGRAM LOOPS CANNOT BE GAURANTEED FOR FUNCTION LEVEL PROGRAMS.

5.1.1 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.

6.0 TESTING PREREQUISITES

BEFORE RUNNING THIS PROGRAM, THE FOLLOWING TESTS SHOULD BE SUCCESSFULLY RUN IN THE FOLLOWING ORDER:

1. ALL APPLICABLE PDP-11 CPU TESTS
2. ISB11A STANDALONE DIAGNOSTICS:
 1. DZKCC
 2. DZKCD
 3. DZKCA
 4. DZKCE
 5. DZKCF
 6. DZKCI (SERIAL BUS EXERCISER)

7. FOLLOWING THESE STANDALONE TESTS, THE ISB11A SHOULD BE REMAINED CONNECTED TO THE SERIAL BUS, WITH 1-63 RT801, 803, OR 805 TERMINALS, AND THE REMOTE TERMINAL EXERCISOR SHOULD BE RUN.

7.0 XXDP.

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

8.0 ACT/APT

THE PROGRAM IS NEITHER ACT NOR APT COMPATIBLE.

9.0 MEMORY MANAGEMENT

THE PROGRAM DOES NOT DIRECTLY UTILIZE OR TEST MEMORY MANAGEMENT.

10.0 SWITCH REGISTER FUNCTIONS

NO EXPLICIT SWITCH REGISTER SETTINGS WILL BE RECOGNIZED BY THE PROGRAM, SO AS TO FACILITATE INTEGRATION UNDER RSX-11M AND THE PDP-11 DIAGNOSTIC SUPERVISOR.

11.0 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.

11.1 COMMAND LANGUAGE

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

11.1.1 RSX-11M COMMAND LANGUAGE -

RUN - THE RUN DIRECTIVE CAUSES A TASK TO BE REQUESTED.

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

11.1.1.1 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

THESE DIAGNOSTIC 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 EXECUTION 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.

12.0 LOADING

SINCE THE EXERCISER IS PART OF THE DPM SOFTWARE, IT IS REQUESTED AND LOADED BY THE RSX-11M COMMAND "RUN DZKCI" WHERE "DZKCI" 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 DIALOGUE. (NOTE: OPERATOR'S RESPONSE IS UNDERLINED).

>RUN DZKCI

DCP-A>STA

UNITS (D) ? 8
--

UNIT 1
ENTER CONTROLLER NUMBER : (0) 0 ?
--

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

UNIT 5
ENTER CONTROLLER NUMBER : (0) 0 ? 1

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

UNIT 9
ENTER CONTROLLER NUMBER: (0) 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
4	2	14	20

12.1 EXPLANATION OF PRINTOUT

>"RUN DZKCI" 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 13.0

13.0 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.

13.1 SELECTED CONTROLLER

THE PROGRAM CAN EXERCISE 8 CONTROLLERS (0-7), BUT THE LARGEST NUMBER OF CONTROLLERS PRESENTLY PLANNED FOR DPM SYSTEMS IS 4(0-3). 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. 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 RETURN.

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

13.2 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. ONE TASK CAN EXERCISE 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. INTERROGATION HERE CONTINUES WITH:

UNIT 1
"ENTER TERMINALS ON THAT CONTROLLER (0) 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 : (0) 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 : (0) 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 I.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      XX      %
```

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.

13.3 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 DZKCI  
DCP-A> STA  
  UNITS (0) ? 63  
UNIT 1  
ENTER CONTROLLER NUMBER : (0) ?
```

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 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 DZKCI

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

E. T. C

E. T. C

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-EXISTENCE SYSGENED CONTROLLERS AND/OR TERMINALS, THE OPERATOR

WILL BE INFORMED WITH:

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

14.0 TEST DESCRIPTION

THE PROGRAM ESSENTIALLY CONSISTS OF ONE TEST, AND IS EXECUTED AS FOLLOWS:

AFTER INITIAL OPERATOR DIALOGUE, AND UPON TEST ENTRY, ALL ACTIVE F.D.C. TERMINALS ARE PLACED IN A UP TO 63 WORD TABLE, WITH THE CONTROLLER NUMBER IN THE HI-BYTE, AND THE TERMINAL NUMBER IN THE LO-BYTE OF THE WORD. DIAGNOSTICS ARE REQUESTED TO BEGIN, IN THE 1ST ACTIVE TERMINALS ON THE TABLE. THE PROGRAM THEN CHECKS FOR TERMINAL OFF LINE/ON LINE CONDITIONS, INFORMS THE OPERATOR OF THOSE CONDITIONS, LOGS THE CONDITIONS IN AN OFFLINE/ONLINE TABLE, CHECKS FOR DIAGNOSTIC ERRORS, AND THE PROCEDURE IS CONTINUED TO THE LAST TERMINAL. HAVING DONE THE LAST TERMINAL, A TWO MINUTE INTERVAL IS STARTED, THE 1ST ACTIVE TERMINAL IN THE TABLE IS CHECKED FOR OFFLINE/ONLINE CONDITIONS, AND A REQUEST IS ISSUED TO ONLINE TERMINALS. FOR TERMINALS RESPONDING WITHIN THE TWO MINUTE INTERVAL, DIAGNOSTIC ERRORS ARE CHECKED, LOGGED IN A TABLE, AND THE PROCEDURE IS CONTINUED TO THE LAST TERMINAL. IF THE TERMINAL DOES NOT RESPOND IN THE TWO MINUTE INTERVAL, IT IS LOGGED AS BEING OFFLINE, AND THE OPERATOR IS INFORMED THAT IT FAILED TO RETURN ON LINE FROM THE DIANOSTIC TESTS, AND THE PROCEDURE IS CONTINUED TO THE LAST TERMINAL. UPON COMPLETION OF THE LAST TERMINAL, THE ERROR TABLE IS CHECKED, ERRORS ARE REPORTED, AND THE 2 MINUTE INTERVAL IS CANCELLED IF NOT FINISHED. AN END OF PASS IS THEN RECORDED FOR ONLY ON-LINE TERMINALS, AND ANOTHER PASS IS STARTED.

15.0 ERROR REPORTING

AT THE END OF EVERY PASS, MAJOR ERRORS SENSED, ARE THE DIAGNOSTIC ERRORS GENERATED BY THE TERMINALS, AND ARE REPORTED AS FOLLOWS:

15.1 MAJOR ERRORS

EXAMPLE: "TERMINAL DIAGNOSTIC ERRORS ON PASS 1"

"SB"	TERMINAL-TYPE	FAILING-TEST
1	801	21

WHICH INDICATES DIAGNOSTIC ERRORS HAVE OCCURED DURING PASS 1, IN SB TERMINAL UNIT 1, OF TERMINAL-TYPE RT-801 AND THE TEST WHICH FAILED IS TEST 21.

FOR ANY TERMINAL NOT RESPONDING ON LINE AT ANYTIME, THE OPERATOR IS INFORMED WITH:

"TERMINAL X IS NOT RESPONDING"

FOR ANY TERMINAL COMING BACK ON LINE, THE OPERATOR IS INFORMED WITH:

"TERMINAL X IS NOW RESPONDING"

FOR ANY TERMINAL GOING OFFLINE DURING DIAGNOSTIC TESTS, THE OPERATOR IS INFORMED WITH:

"TERMINAL X DID NOT COME BACK ON LINE FROM DIAGNOSTIC TESTS"

FOR ALL TERMINALS GOING OFFLINE, THE OPERATOR IS INFORMED WITH:

"NO ACTIVE TERMINALS ON THE BUS" !!!!

FOR AN END OF PASS REPORT, THE OPERATOR IS INFORMED WITH:

EXAMPLE: END OF PASS 1, TIME=8:15

THIS END OF PASS REPORT CAN BE INHIBITED BY TYPING THE THE FOLLOWING, UPON RECEIVING THE DCP-A> PROMPT.

DCP-A>STA/FLA: IXE

16.0 EXECUTION TIME

EXECUTION-TIME PER PASS VARIES. MAXIMUM TIME = 2 MINS. PASS-TIME DEPENDS ON EXECUTION TIME OF DIAGNOSTIC TESTS. ERROR FREE DIAGNOSTIC TESTS WILL BE EXECUTED QUICKLY (APPROX 15 SECS.). A TERMINAL WHICH GENERATES ERRORS, WILL RUN THE DIAGNOSTIC (AT MINIMUM) 90 SECS, AT WHICH TIME THE DEVICE WILL GO OFFLINE. THE PROGRAM THEN WAITS AN ADDED 30 SECS TO ENSURE THAT THE DEVICE IS BACK ON LINE, WHICH ACCOUNTS FOR A MAXIMUM PASS TIME OF 2 MINS.

17.0 PROGRAM TERMINATION

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

EXAMPLE: CONTROL C

MCR>

DZKCI MACRO M11 10-OCT-77 10:33
TABLE OF CONTENTS

SEQ 0018

2-	32	DIAGNOSTIIC SUPERVISOR HEADER
3-	43	EQUATES
5-	89	P-TABLE MESSAGES
6-	103	GENERAL AND DPB STORAGE LOCATIONS
7-	202	HARDWARE P-TABLE
8-	223	DEFAULT HARDWARE P-TABLE
9-	240	SOFTWARE P-TABLE
10-	263	DISPATCH TABLE
11-	277	INIT CODE
12-	500	MAIN TEST

```

1          . TITLE DZKCI
2          . ENABL AMA
3          . NLIST MD, ME
4          . LIST  MEB
5          . MCALL SVC
6          . MCALL ASTXSS, Q10S, ALUNS, DIRS, GLUNS, EXITSS
7          . MCALL MRKTSS, CMKTSS, GTIMSS
8          . MCALL WTSESS
9          . GLOBL STADR
10         000000
11         000001 SVC
12         000001 SVC INS= 1 ;LIST GENERATED SYMBOLS
13         000001 SVCTST= 1 ;LIST TEST NUMBERS
14         000001 SVCSUB= 1 ;LIST SUBTEST NUMBERS
15         000001 SVCGBL= 1 ;LIST BLOBALS
23         000001 SVCTAG= 1 ;LIST GENERATED TAGS

```

```

32          . SBTTL  DIAGNOSTIC SUPERVISOR HEADER
33          ;
34          ; *****
35          ; THIS SECTION CONTAINS GENERAL INFO WHICH
36          ; DESCRIBES THE MAJOR CHARACTERISTICS OF
37          ; THE DIAGNOSTIC PROGRAM
38          ; *****
39 000000    POINTER NONE
40
41 000000    STADR:  HEADER DZKCI,A,0,63,300,1
      000000    LSNAME: :
      000000      104      . ASCII  @D@
      000001      132      . ASCII  @Z@
      000002      113      . ASCII  @K@
      000003      103      . ASCII  @C@
      000004      111      . ASCII  @I@
      000005      000      . BYTE   0
      000006      000      . BYTE   0
      000007      000      . BYTE   0
      000010          LSREV: :
      000010      101      . ASCII  @A@
      000011          LSDEPO: :
      000011      060      . ASCII  @O@
      000012          LSMREV: :
      000012      001      . BYTE   CSREVISION
      000013      004      . BYTE   CSREDIT
      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 005032'      . WORD   LSDISPATCH
      000042          LSICP: :
      000042 005034'      . WORD   LSINIT
      000044          LSCCP: :
      000044 007676'      . WORD   LSCLEAN
      000046          LSHPCP: :
      000046 004772'      . WORD   LSHARD
      000050          LSSPCP: :
      000050 000000      . WORD   0
      000052          LSDEVP: :
      000052 000210'      . WORD   LSDVTYP
  
```

000054		LSREPP::		
000054	000000		.WORD	0
000056		LSHPTP::		
000056	005020'		.WORD	LSHW
000060		LSSTPT::		
000060	000000		.WORD	0
000062		LSDRCT::		
000062	000202'		.WORD	LSDR
000064		LSDRS::		
000064	000206'		.WORD	LSDRST
000066		LSSTA::		
000066	000000		.WORD	0
000070		LSAUT::		
000070	000000		.WORD	0
000072		LSDUT::		
000072	000000		.WORD	0
000074		LSWRU::		
000074	000000		.WORD	0
000076		LSLADP::		
000076	011166'		.WORD	LSLAST

```
43 . SBTTL EQUATES
44 ;
45 ; *****
46 ; THIS FACILITATES PROGRAM ASSEMBLY BY EQUATING
47 ; SPECIFIC SYMBOLS WITH SPECIFIC VALUES VIA DIRECT
48 ; ASSIGNMENT STATEMENTS.
49 ; *****
50
51 000011 LUN =11 ;DEFINE LOGICAL UNIT NUMBER FROM 11
52
53 ;OFFSETS IN DEVICE PARAMETER BLOCK (DCB)
54
55 000002 10. FUN= 2
56 000004 10. LUN= 4
57 000006 10. EFN= 6
58 000010 10. IST= 10
59 000012 10. AST= 12
60 000014 10. A1= 14
61 000016 10. A2= 16
62 000020 10. A3= 20
63 000022 10. A4= 22
64 000024 10. A5= 24
65
66 ; RSX I/O FUNCTION CODES
67
68 002400 10. WPC= 2400 ;WRITE PERIPHERAL BLOCK
69 003400 10. RXP= 3400 ;READ TRANSPARENT
70 003000 10. RPC= 3000 ;READ PERIPHERAL BLOCK
```

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
90
91
92
93
94
95
96
97

.SBTTL P-TABLE MESSAGES

++NOTE++ P-TABLE MESSAGES ARE USED DURING
DIALOGUE WITH THE OPERATOR.

98 000100 105 116 124
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
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

GETCNT: .ASCIZ/ENTER CONTROLLER NUMBER : /

GETERM: .ASCIZ/ENTER TERMINALS ON THAT CONTROLLER : /

100
101

.EVEN

```
103 .SBTTL GENERAL AND DPB STORAGE LOCATIONS
104 ;
105 ; *****
106 ; THIS SECTION OF THE SOFTWARE CONTAINS ALL
107 ; DIRECTIVE PARAMETER BLOCKS AS USED BY RSX-11M.
108 ; ALSO CONTAINS PERMANENT/TEMPORARY STORAGE.
109 ; *****
110 ;
111 ;
112 ;
113 000200 DEVREG 1,1 .WORD 1
    000200 000001
    000202 LSDR: .WORD 1
    000202 000001 .WORD 1
    000204 000001 .WORD 1
    000206 LSDRST: .BLKW 1
114 ; *****
115 ; THE DEVREG CALL ASSEMBLES A PAIR OF TABLES
116 ; WHICH ARE USED TO DEFINE THOSE DEVICE REGISTERS
117 ; THAT WILL BE USED BY THE PROGRAM.
118 ; *****
119 000210 DEVTYPE SB
    000210 LSDVTYP: .ASCIZ @SB@
    000210 123 102 000 .EVEN
120 ;
121 ; *****
122 ; THE DEVTYP CALL SPECIFIES THE EQUIPMENT THAT
123 ; THE EXERCISER PROGRAM WILL SERVE.
124 ; *****
125 ;
126 ; *****
127 ; THE QUED I/O DIRECTIVE PARAMETER BLOCK IS SET
128 ; UP AS FOLLOWS.....
129 ; *****
130 ;
131 ;
132 000214 QI0DPB: QI05 10.FUN,LUN,,,I0ST,AST,<SCBADR,SCBCNT,40,PCBADR,PCBCNT>
    000214 001 014 .BYTE 1,12.
    000216 000002 .WORD 10.FUN
    000220 000011 .WORD LUN
    000222 000 000 .BYTE ,0
    000224 000744 .WORD I0ST
    000226 007634 .WORD AST
    000230 000700 .WORD SCBADR
    000232 000702 .WORD SCBCNT
    000234 000040 .WORD 40
    000236 000704 .WORD PCBADR
    000240 000706 .WORD PCBCNT
    000242 000000 .WORD 0
133 ;
134 ; *****
135 ; THE ASSIGN LOGICAL UNIT PARAMETER BLOCK IS
136 ; DEFINED AS FOLLOWS.....
137 ; *****
138 ;
```

```
139 000244          ALUDB: 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 ;
144 ;
145 ;
146 ;
147 ;
148 ;
149 000254          GLUDB: GLUN$  11,ERRBRT          ;PARAMETER BLOCK FOR GLUN
    000254      005      003          . BYTE  5, 3
    000256      000011          . WORD   11
    000260      000262          . WORD   ERBRT

150 ;
151 ;
152 ;
153 ;
154 ;
155 ;
156 ;
157 000262          ERBRT: . BLKW  6          ;GLUN INFO BUFFER I. E. BROADCAST ERROR
158 000276          UNTBL: . BLKW  64.         ;RUNTIME DEVICE UNIT TABLES
159 000476          DEVTBL: . BLKW  64.         ;ACTIVE LUN, DEV TBLLE
160 000676      000000          UNITS: . WORD  0          ;NUMBER OF ACTIVE TERMINALS
161 ;
162 ;
163 ;
164 ;
165 ;
166 000700      000000          SCBADR: . WORD  0
167 000702      000000          SCBCNT: . WORD  0
168 000704      000000          PCBADR: . WORD  0
169 000706      000000          PCBCNT: . WORD  0
170 ;
171 ;
172 ;
173 ;
174 000710      000000          TEMP:   . WORD  0          ;THE FOLLOWING FIVE LOCATIONS ARE TEMPORARY
175 000712      000000          TEMP1:  . WORD  0
176 000714      000000          TEMP2:  . WORD  0
177 000716      000000          TEMP4:  . WORD  0
178 000720      000000          TEMP5:  . WORD  0
179 000722      000000          TEMP6:  . WORD  0
180 000724      000000          TEMP7:  . WORD  0
181 000726      000001          PASFG:  . WORD  1          ;PROGRAM PASS INDICATOR
182 000730      000000          MINFLG: . WORD  0          ;2 MINUTE FLAG INDICATOR
183 000732      000000          ASTFLG: . WORD  0          ;AST INDICATOR
184 000734      000000          ERRFLG: . WORD  0          ;ERROR FLAG INDICATOR
185 000736      000000          PASS:  . WORD  0          ;PASS COUNT INDICATOR
186 000740          BUFF:   . BLKW  2          ;OUTPUT BUFFER ADDRESS
187 000744          IOST:   . BLKW  2          ;I/O STATUS ADDRESS
188 000750          BUF:    . BLKW  10         ;TIME PARAMETER BUFFER
```

189	000770	OFFLIN:	. BLKW	64.
190	001170	ERRBLK:	. BLKW	448.
191	002770	IBUF0:	. BLKB	128.
192	003170	IBUF1:	. BLKB	128.
193	003370	IBUF2:	. BLKB	128.
194	003570	IBUF3:	. BLKB	128.
195	003770	IBUF4:	. BLKB	128.
196	004170	IBUF5:	. BLKB	128.
197	004370	IBUF6:	. BLKB	128.
198	004570	IBUF7:	. BLKB	128.
199				
200		. EVEN		

; OFFLINE TABLE ADDRESS
; ERROR BLOCK ADDRESS
; THE FOLLOWING 8 BUFFER LOCATIONS ARE TEMP

202
203
204
205
206
207
208
209
210
211
212
213
214

.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.....

215 004770
004770 000012
004772
216 004772
004772 000032
004774 000100
004776 177400
005000 000000
005002 000007
217 005004
005004 000052
005006 000132
005010 000377
005012 000001
005014 000077

BGNHRD
LSHARD: : . WORD L10000-LSHARD/2
GPRMD GETCNT, 0, 0, 177400, 0, 7, YES
. WORD TSCODE
. WORD GETCNT
. WORD 177400
. WORD TSLOLIM
. WORD TSHILIM
GPRMD GETERM, 0, D, 377, 1, 63, YES
. WORD TSCODE
. WORD GETERM
. WORD 377
. WORD TSLOLIM
. WORD TSHILIM

218
219
220 005016
005016
221

ENDHRD
L10000: . EVEN

```
223 . SBTTL DEFAULT HARDWARE P-TABLE
224 ;
225 ;
226 *****
227 *****
228 *****
229 *****
230 *****
231 *****
232 ;
233 BGNHW
234 005016 000001 . WORD L10001-LSHW/2
235 005020 000001 ; CONTROLLER/TERMINAL DEFAULT
236 005022
237 005022
238
```

```
240 .SBTTL SOFTWARE P-TABLE
241 ;
242 ;
243 ; *****
244 ; ANOTHER SEPERATE AND OPTIONAL PARAMETER
245 ; TABLE ASSEMBLED WITH THE DIAGNOSTIC
246 ; PROGRAM. ....
247 ; *****
248 005022 BGNSFT
    005022 000000 .WORD L10002-L$SOFT/2
    005024 L$SOFT:
249 005024 ENDSFT
    005024 L10002: .EVEN
250
251 ;BUILD SOFTWARE P-TABLE
252
253
254 005024 BGNSW
    005024 000000 .WORD L10003-L$SW/2
    005026 L$SW:
255
256 005026 ENDSW
    005026 L10003:
257
258
259 005026 BGNRPT
    005026 L$RPT:
260
261 005026 ENDRPT
    005026 L10004:
    005026 104025 EMT CSRPT
```

263
264
265
266
267
268
269
270
271
272
273
274 005030
005030 000001
005032
005032 006274'
275

```
.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.....  
*****  
;  
DISPATCH 1 .WORD 1  
LSDISPATCH: .WORD T1
```


277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309 005034
005034
310 005034
005034 012746
005036 033 001
005040 104377
311 005042 005037 000726'
312 005046 005037 000736'
313
314
315
316 005052 012701 000262'
317 005056 005021 139\$:
318 005060 020127 002770'
319 005064 001374
320 005066 012701 002770'
321 005072 105021 50\$:
322 005074 020127 004570'
323 005100 001374
324
325 005102 005037 000226'
326 005106 012737 000000G 000216'
327 005114 012705 000011
328 005120 010537 000220' 30\$:
329 005124

.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,
QIOS, DIRS, ALUNS, GLUNS, EXITSS...
RESOURCES USED FROM THE DIAGNOSTIC SUPERVISER,
ARE THE MACROS GPHARD, PRINT...
--
GPHARD: THE GET PARAMETER HARD CALL, REQUEST A POINTER
TO THE TABLE, WHICH HOLDS THE CONTROLLERS AND TERMINALS
ON THAT CONTROLLER...
PRINT: OUTPUT MESSAGES AND ERRORS TO THE TELETYPE...
--
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...
++
*****

```

BGNINIT
LSINIT:

```

CMKTSS ;CANCEL ANY OUTSTANDING MARK TIME REQUEST
MOV (PC)+, -(SP)
.BYTE 27, 1
EMT 0<377>
CLR PASFG ;INIT INDICATOR
CLR PASS ;INIT INDICATOR
*****
CLEAR STORAGE LOCATIONS FOR SUBSEQUENT INITIALIZATION.
*****
MOV #ERRBRT, R1
139$: CLR (R1)+
CMP R1, #IBUFD
BNE 139$
MOV #IBUFD, R1
50$: CLRB (R1)+
CMP R1, #IBUF7
BNE 50$
*****
CLR QI0DPB+10. AST ;NO AST FOR DETACH
MOV #10. DET, QI0DPB+10. FUN ;I/O FUNCTION CODE
MOV #LUN, R5 ;LOGICAL UNIT NUMBER
30$: MOV R5, QI0DPB+10. LUN ;LUN DPB
DIRS #QI0DPB ;ISSUE I/O REQUEST

```

	005124	012746	000214'		MOV	#Q10DPB, -(SP)		
	005130	104377			EMT	0<377>		
330	005132	005205			INC	R5		; UPDATE LUN POINTER
331	005134	020527	000113		CMP	R5, #75.		; REQUEST FINISHED?
332	005140	003767			BLE	30%		; CONTINUE REQUEST IF NOT
333	005142	012705	000276'		MOV	#UNTBL, R5		; GET 64 ONE-WORD UNIT TABLE
334	005146	013703	000014'		MOV	LSUNIT, R3		; NUMBER OF UNITS FROM HEADER
335	005152	010337	000676'		MOV	R3, UNITS		; SAFE KEEP NUMBER OF UNITS
336	005156	005004			CLR	R4		; INIT POINTER TO P-TABLE
337	005160			15:	GPHARD	R4, RO		; GET POINTER TO HARDWARE P-TABLE
	005160	010400					MOV	R4, RO
	005162	104055					EMT	CSGPHRD
338	005164	011025			MOV	(RO), (R5)+		; GET TERM & CONTROLLER # FROM HARDWARE
339								; P-TABLE, PUT IN UNIT TABLE
340	005166	005204			INC	R4		; NEXT TABLE
341	005170	005303			DEC	R3		; LAST P-TABLE?
342	005172	001372			BNE	1%		; BRANCH IF NO---
343	005174	012715	177777		MOV	#-1, (R5)		; ELSE INSERT END OF TABLE FLAG
344	005200	005000			CLR	RO		; OFFSETS TO TEMPORARY TABLES IN IBUFO
345	005202	005003			CLR	R3		; SB # INDICATOR
346	005204	012702	177777		MOV	#-1, R2		; CONTROLLER # INDICATOR FIRST TIME THRU
347	005210	005004			CLR	R4		; TERMINAL # INDICATOR
348	005212	012737	000011	000246'	25:	MOV	#LUN, ALUDPB+2	; SET THE LUN =11
349	005220	010337	000252'		MOV	R3, ALUDPB+6		; SET THE DEVICES
350	005224				DIRS	#ALUDPB		; ISSUE I/O REQUEST
	005224	012746	000244'		MOV	#ALUDPB, -(SP)		
	005230	104377			EMT	0<377>		
351	005232	103444			BCS	4%		; CHECK IF REQUEST IS REJECTED
352	005234	012737	000011	000256'	MOV	#LUN, GLUDPB+2		; ELSE SET THE LUN FOR THE G- DIRECTIVE
353	005242	012737	000262'	000260'	MOV	#ERRBRT, GLUDPB+4		; GET THE LUN INFO BUFFER ADDRESS
354	005250				DIRS	#GLUDPB		; ISSUE I/O DIRECTIVE
	005250	012746	000254'		MOV	#GLUDPB, -(SP)		
	005254	104377			EMT	0<377>		
355	005256	005737	000266'		TST	ERRBRT+4		; CHECK FOR BROADCAST CHANNEL
356	005262	001413			BEQ	3%		; BRANCH IF BROADCAST CHANNEL
357	005264	110460	002770'		MOVB	R4, IBUFO+0(RO)		; STORE TERMINAL # IN TABLE
358	005270	110260	002771'		MOVB	R2, IBUFO+1(RO)		; STORE CONTROLLER #
359	005274	010360	003770'		MOV	R3, IBUF4(RO)		; STORE SB #
360	005300	005204			INC	R4		; UPDATE PHYSICAL TERMINAL #
361	005302	062700	000002		ADD	#2, RO		; UPDATE POINTER
362	005306	005203			INC	R3		; UPDATE SB #
363	005310	000740			BR	2%		; CONTINUE I/O REQUEST TO NEXT TERMINAL
364	005312	005202		35:	INC	R2		; INCREMENT CONTROLLER # I. E. LOG NEXT CONTROLLER
365	005314	005004			CLR	R4		; CLEAR TERMINAL #. I. E. TERMINALS ON NEXT CONTROLLER
366	005316	110460	002770'		MOVB	R4, IBUFO+0(RO)		; STORE TERMINAL #
367	005322	110260	002771'		MOVB	R2, IBUFO+1(RO)		; STORE CONTROLLER #
368	005326	010360	003770'		MOV	R3, IBUF4(RO)		; STORE SB
369	005332	005204			INC	R4		; UPDATE TERMINAL #
370	005334	005203			INC	R3		; UPDATE SB #
371	005336	062700	000002		ADD	#2, RO		; UPDATE POINTER
372	005342	000723			BR	2%		; ASSIGN NEXT TERMINAL
373	005344	012760	177777	002770'	45:	MOV	#-1, IBUFO(RO)	; END OF SB TABLE
374	005352	013703	000676'		MOV	UNITS, R3		; GET RUNTIME UNIT #
375	005356	005004			CLR	R4		; POINTER TO UNTBL CONTAINS
376								; PHYSICAL # IN LO-BYTE
377								; & CONTROLLER # IN HI-BYTE
378	005360	005001		55:	CLR	R1		; POINTER TO IBUFO TABLE

```

379 005362 016402 000276'      MOV      UNTBL(R4),R2      ;GET CONTROLLER # TERM # FROM UNIT TABLE
380 005366 020261 002770'      6$:     CMP      R2,IBUFO(R1)    ;CHECK IF SAME FROM MAPPED TABLE
381 005372 001406                BEQ      7$              ;BRANCH IF SAME
382 005374 005761 002770'      TST      IBUFO(R1)      ;CHECK IF END OF TABLE
383 005400 100413                BMI      8$              ;BRANCH IF END
384 005402 062701 000002      ADD      #2,R1          ;GET NEXT ITEM
385 005406 000767                BR       6$              ;CONTINUE CHECKING
386 005410 116164 003770' 000476' 7$:  MOVB    IBUF4(R1),DEVTBL+0(R4) ;STORE LOGICAL SB NUMBER
387 005416 062704 000002      ADD      #2,R4          ;GET NEXT UNTBL ENTRY
388 005422 005303                DEC      R3              ;LAST PHYSICAL TERMINAL TO BE STORED
389 005424 001355                BNE     5$              ;BRANCH IF NO--
390 005426 000476                BR       11$             ;ELSE GET NEXT DEVICE
391 005430 016437 000276' 000710' 8$:  MOV      UNTBL(R4),TEMP  ;TEMPORARY STORAGE FOR PRINT
392 005436                PRINTF  #MSGO,<B,TEMP+1>,<B,TEMP+0> ;PRINT THE NON-SYSGEN MESSAGE
    005436 005046                CLR      -(SP)
    005440 153716 000710'      BISB    TEMP+0,(SP)
    005444 005046                CLR      -(SP)
    005446 153716 000711'      BISB    TEMP+1,(SP)
    005452 012746 007700'      MOV     #MSGO,-(SP)
    005456 012746 000003      MOV     #3,-(SP)
    005462 010600                MOV     SP,RO
    005464 104017                EMT     C$PNTF
    005466 062706 000010      ADD     #10,SP

393 ;
394 ;
395 ;
396 ;
397 ;
398 ;
399 ;
400 ;
401 ;
402 ;
403 ;
404 005472 005303                DEC      R3              ;ONE TOO MANY UNITS? UNASSIGNABLE?
405 005474 001433                BEQ     10$             ;BRANCH IF ONE TOO MANY
406 005476                PUSH    <R3,R4>        ;SAVE
    005476 010346
    005500 010446
407 005502 016464 000300' 000276' 9$:  MOV     UNTBL+2(R4),UNTBL(R4) ;BUBBLE UP FROM BELOW TO THIS LOCATION
408 005510 062704 000002      ADD     #2,R4          ;NEXT LOCATION
409 005514 005303                DEC     R3              ;DECREASE LOCATION
410 005516 001371                BNE    9$              ;BRANCH IF NOT DONE
411 005520                POP     <R4,R3>        ;RESTORE
    005520 012604
    005522 012603
412 005524 005337 000676'      DEC     UNITS           ;ELIMINATE ONE UNIT OFF TABLE
413 005530 005737 000676'      TST     UNITS           ;ARE THERE ANY MORE DEVICES
414 005534 001311                BNE    5$              ;YES--BRANCH
415 005536                PRINTF #MABORT         ;NO-PRINT ABORT MESSAGE
    005536 012746 010342'      MOV     #MABORT,-(SP)
    005542 012746 000001      MOV     #1,-(SP)
    005546 010600                MOV     SP,RO
    005550 104017                EMT     C$PNTF
    005552 062706 000004      ADD     #4,SP
416 005556                EXIT$S
    005556 012746                MOV     (PC)+,-(SP)    ;AND EXIT TEST

```

```

*****
BUBBLE ROUTINE... IF ONE, OR ALL TERMINALS ENTERED, ARE
UNASSIGNABLE OR UNATTACHABLE, THIS ROUTINE ELIMINATES THEM
OFF THE TABLE, AND ABORTS THE PROGRAM...
--
*****

```

```

005560 063 001 . BYTE 51,1
005562 104377 EMT 0<377>
417 005564 005337 000676' 10$: DEC UNITS ; ONE TOO MANY UNITS
418 005570 005737 000676' TST UNITS ; LAST DEVICE?
419 005574 001013 BNE 11$ ; BRANCH IF NO
420 005576 PRINTF #MABORT ; ELSE PRINT ABORT MESSAGE
005576 012746 010342' MOV #MABORT,-(SP)
005602 012746 000001 MOV #1,-(SP)
005606 010600 MOV SP,RO
005610 104017 EMT C$PNTF
005612 062706 000004 ADD #4,SP
421 005616 EXIT$$ ; AND EXIT
005616 012746 MOV (PC)+,-(SP)
005620 063 001 . BYTE 51,1
005622 104377 EMT 0<377>
422 005624 005001 11$: CLR R1 ; DEVICE TABLE POINTER
423 005626 012705 000011 MOV #LUN,R5 ; SET THE LUN
424 005632 013703 000676' MOV UNITS,R3 ; KEEP COUNT
425 005636 012737 000000G 000216' MOV #10.ATT,Q10DPB+10.FUN ; SET FOR ATTACHES
426 005644 110537 000246' 12$: MOV R5,ALUDPB+2 ; SET LUN FOR DIR$
427 005650 116102 000476' MOV DEVTBL+0(R1),R2 ; GET THE SB DEVICE
428 005654 110237 000252' MOV R2,ALUDPB+6 ; SET THE DEVICE NUMBER
429 005660 DIR$ #ALUDPB ; ISSUE THE I/O REQUEST
005660 012746 000244' MOV #ALUDPB,-(SP)
005664 104377 EMT 0<377>
430 005666 103017 BCC 13$ ; BRANCH IF REQUEST IS ACCEPTED
431 005670 013704 000J00G MOV $DSW,R4 ; IF NOT GET DSW ERROR CODE
432 005674 PRINTF #MSG1,<B,R2>,<B,R4> ; PRINT ERROR MESSAGE
005674 005046 CLR -(SP)
005676 150416 BISB R4,(SP)
005700 005046 CLR -(SP)
005702 150216 BISB R2,(SP)
005704 012746 007773' MOV #MSG1,-(SP)
005710 012746 000003 MOV #3,-(SP)
005714 010600 MOV SP,RO
005716 104017 EMT C$PNTF
005720 062706 000010 ADD #10,SP
433 ; UNASSIGNABLE TERMINAL
434 005724 000437 BR 17$ ; DO NOT TRY TO ATTACH--
435 ; BUT GO GET RID OF UNIT
436 ; FROM THE DEVICE TABLE
437 005726 110561 000477' 13$: MOV R5,DEVTBL+1(R1) ; PUT THE LUN IN THE DEVICE TABLE
438 005732 010537 000220' MOV R5,Q10DPB+10.LUN ; SET THE LUN FOR AN ATTACH
439 005736 DIR$ #Q10DPB ; ISSUE I/O REQUEST
005736 012746 000214' MOV #Q10DPB,-(SP)
005742 104377 EMT 0<377>
440 005744 103411 BCS 16$ ; ERROR INDICATOR
441 005746 062701 000002 14$: ADD #2,R1 ; GET NEXT SB (LOGICAL)
442 005752 005205 INC R5 ; NEXT LUN
443 005754 005303 DEC R3 ; CHECK FOR LAST SB
444 005756 001332 BNE 12$ ; BRANCH IF NOT--GO DO NEXT ASSIGN
445 005760 15$:
446 005760 012761 177777 000476' MOV #-1,DEVTBL(R1) ; INSERT END OF FLAG IN DEVICE TABLE
447 005766 000457 BR 22$ ; AND PRINT ACTIVE DEVICE MAP
448 005770 013704 000000G 16$: MOV $DSW,R4 ; PRINT ERROR MESSAGE I.E UNATTACHABLE DEVICE
449 005774 PRINTF #MSG,<B,R2>,<B,R4>
005774 005046 CLR -(SP)

```



```

480 006160 116001 000477'      20$:  MOVB  DEVTBL+1(RO),R1      ;GET LUN TO BE PRINTED
481 006164 116002 000476'      MOVB  DEVTBL+0(RO),R2      ;GET LOGICAL SB NUMBER
482 006170 116004 000277'      MOVB  UNTBL+1(RO),R4      ;GET CONTROLLER NUMBER
483 006174 116005 000276'      MOVB  UNTBL+0(RO),R5      ;GET PHYSICAL TERM#
484 006200                                PUSH  RO                  ;SAVE RO**SUPERVISOR USES IT
      006200 010046
485 006202                                PRINTF #MSG3,<B,R5>,<B,R4>,<B,R2>,<B,R1> ;
      006202 005046                                CLR  -(SP)
      006204 150116                                BISB R1,(SP)
      006206 005046                                CLR  -(SP)
      006210 150216                                BISB R2,(SP)
      006212 005046                                CLR  -(SP)
      006214 150416                                BISB R4,(SP)
      006216 005046                                CLR  -(SP)
      006220 150516                                BISB R5,(SP)
      006222 012746 010176'      MOV  #MSG3,-(SP)
      006226 012746 000005      MOV  #5,-(SP)
      006232 010600      MOV  SP,RO
      006234 104017      EMT  C$PNTF
      006236 062706 000014      ADD  #14,SP
486 006242                                POP  RO                  ;RESTORE RO
      006242 012600
487                                ;PRINT ACTIVE DEVICE MAP
488 006244 062700 000002      ADD  #2,RO
489 006250 005303      DEC  R3
490 006252 001342      BNE  20$
491 006254 005237 000736'      21$: INC  PASS          ;INIT PASS LOCATION
492 006260 013702 000676'      MOV  UNITS,R2          ;GET NUMBER OF TERMINALS
493 006264 006302      ASL  R2                ;POINTER TO END OF TERMINALS
494 006266 005037 000722'      CLR  TEMP6            ;USED AS PRINT FLAG
495
496
497 006272                                ENDINIT
      006272                                L10005:
      006272 104011                                EMT  C$INIT
498

```

```

500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525 006274
      006274
526
527
528 006274
529 006274 005037 000734'
530 006300 005037 000730'
531 006304 005001
532 006306 005037 000732'
533 006312 012737 040500 000740'
534 006320 012737 040100 000742'
535
536
537
538 006326 005061 001170'
539 006332 005721
540 006334 022701 000700
541 006340 001372
542 006342 005001
543
544
545
546
547 006344 012737 002400 000216'
548 006352 012737 000744' 000224'
549 006360 012737 007634' 000226'
550 006366 005037 000236'
551 006372 005037 000240'
552 006376 012737 000740' 000230'
553 006404 012737 000004 000232'
554 006412 116137 000477' 000220'
555 006420

```

```

.SBTTL MAIN TEST

*****
THIS PROGRAM ESSENTIALLY CONSISTS OF ONE TEST.
AT ENTRY A TABLE HOLDS LOGICAL UNIT NUMBERS, ASSOCIATED WITH
LOGICAL DEVICES. ANOTHER TABLE HOLDS PHYSICAL DEVICES ON ANY
ONE OF THREE CONTROLLERS. A DIRECTIVE PARAMETER BLOCK IS THEN
BUILT, FOLLOWED BY A REQUEST TO ALL LIVE TERMINALS ON THE BUS,
TO START THEIR INTERNAL DIAGNOSTICS. THE PROGRAM WAITS FOR ALL
TERMINALS TO COMPLETE THE DIAGNOSTICS, BY IMPLEMENTING A --
SYSTEM DELAY DIRECTIVE, THEN READS, LOGS, AND REPORTS ALL
ERRORS FROM EACH TERMINAL.
--
AT EXIT, PROGRAM PASSES, LOGICAL SB'S, TERMINAL TYPES,
ERRORS, AND TERMINAL DIAGNOSTIC TEST #'S ARE REPORTED...
--
RESOURCES USED FROM RSX-11M ARE THE DIRECTIVES,
QIOS, DIRS, MRKTS, AND ASYNCHRONOUS TRAP (AST).
THE PRINT MACRO IS USED FROM THE DIAGNOSTIC SUPERVISOR.
--
*****

BGNTST
T1:

START:
CLR   ERRFLG           ;INIT ERROR INDICATOR FOR EACH PASS
CLR   MINFLG           ;INIT 2 MIN FLAG INDICATOR
CLR   R1               ;INIT POINTER FOR DEVICE TABLE
CLR   ASTFLG           ;INIT AST INTERRUPT INDICATOR
MOV   #40500,BUFF      ;LOAD PCB
MOV   #40100,BUFF+2    ;LOAD TERMINAL DIAGNOSTICS COMMANDS
*****
INITIALIZE THE ERROR TABLE
*****
205: CLR   ERRBLK(R1)
      TST  (R1)+
      CMP  #448.,R1
      BNE  205
      CLR  R1
*****
BUILD QIO DIRECTIVE PARAMETER BLOCK.
AND ISSUE THE DIRECTIVE TO TERMINAL.
*****
MOV   #10.WPC,QIOPB+10.FUN ;START DPB WITH FUNCTION CODE
MOV   #10ST,QIOPB+10.1ST  ;LOAD STATUS WORD ADDRESS
MOV   #AST,QIOPB+10.AST   ;LOAD TERMINAL WRITE AST
CLR   QIOPB+10.A4        ;CLEAR INPUT BUFFER ADDRESS
CLR   QIOPB+10.A5        ;LOAD INPUT BUFFER SIZE
MOV   #BUFF,QIOPB+10.A1   ;LOAD OUTPUT BUFFER ADDRESS
MOV   #4,QIOPB+10.A2     ;LOAD OUTPUT BUFFER SIZE.
15:  MOVB DEVTBL+1(R1),QIOPB+10.LUN ;LOAD DPB WITH LUN
      DIRS #QIOPB          ;ISSUE I/O REQUEST WPC

```

```

006420 012746 000214'      MOV      #Q10DPB, -(SP)
006424 104377              EMT      0<377>
556 006426 005737 000732'  2$:      TST      ASTFLG      ;CHECK FOR I/O COMPLETION (AST)
557 006432 001775              BEQ      2$           ;CONTINUE CHECK IF NOT COMPLETE
558 006434 005037 000732'      CLR      ASTFLG      ;RESET AST INDICATOR
559 006440 005737 000734'      TST      ERRFLG      ;CHECK FOR ERRORS FROM AST
560 006444 001423              BEQ      4$           ;BRANCH IF NONE
561 006446 005761 000770'      TST      OFFLIN(R1)  ;WAS THE TERMINAL OFF LINE?
562 006452 001015              BNE      3$           ;BRANCH IF YES-ONT
563 006454 005261 000770'      INC      OFFLIN(R1)  ;SET TERMINAL OFFLINE FLAG
564 006460              PRINTF  #MSG4, <B, DEVTBL+0(R1)> ; INFORM OPERATOR OF NO RESPONSE
006460 005046              CLR      -(SP)
006462 156116 000476'      BISB    DEVTBL+0(R1), (SP)
006466 012746 010373'      MOV      #MSG4, -(SP)
006472 012746 000002              MOV      #2, -(SP)
006476 010600              MOV      SP, RO
006500 104017              EMT      C$PNTF
006502 062706 000006              ADD      #6, SP
565 006506 005037 000734'  3$:      CLR      ERRFLG      ;RESET ERROR INDICATOR
566 006512 000420              BR       5$           ;GET NEXT TERMINAL
567 006514 005761 000770'  4$:      TST      OFFLIN(R1)  ;CHECK IF TERMINAL WAS PREVIOUSLY OFFLINE
568 006520 001415              BEQ      5$           ;BRANCH IF NO
569 006522 005061 000770'      CLR      OFFLIN(R1)  ;ELSE MAKE HER ON LINE
570 006526              PRINTF  #MSG5, <B, DEVTBL+0(R1)> ; INFORM OPERATOR TERMINAL IS RESPONDING
006526 005046              CLR      -(SP)
006530 156116 000476'      BISB    DEVTBL+0(R1), (SP)
006534 012746 010443'      MOV      #MSG5, -(SP)
006540 012746 000002              MOV      #2, -(SP)
006544 010600              MOV      SP, RO
006546 104017              EMT      C$PNTF
006550 062706 000006              ADD      #6, SP
571 006554 005721 000006  5$:      TST      (R1)+      ;UPDATE POINTER
572 006556 020102              CMP      R1, R2      ;IS IT LAST TERMINAL
573 006560 001314              BNE      1$          ;NO.. GET NEXT TERMINAL.
  
```



```
575  
576  
577 ; *****  
578 ; WAIT 2 MINUTES FOR TERMINAL TO COME BACK ON LINE  
579 ; AFTER COMPLETING DIAGNOSTICS. IF LESS THAN 2 MINS  
580 ; CHECK FOR ERRORS AND GET THE NEXT TERMINAL. IF MORE  
581 ; THAN 2 MINS, CHECK FOR ERRORS AND LOG AS AN OFFLINE  
582 ; TERMINAL.  
583 ; --  
584 ; MRKTSS IS AN RSX-11M SYSTEM DIRECTIVE MEANING,  
585 ; DECLARE A SIGNIFICANT EVENT...  
586 ; --  
587 ; WTSESS IS AN RSX-11M SYSTEM DIRECTIVE MEANING,  
588 ; WAIT FOR THE COMPLETION OF A SIGNIFICANT EVENT...  
589 ; --  
590 ; *****  
591  
592 006562 65: MRKTSS , #120. , #2, #MAST ; START 2 MINS PERIOD  
006562 012746 007662' MOV #MAST, -(SP)  
006566 012746 000002 MOV #2, -(SP)  
006572 012746 000170 MOV #120. , -(SP)  
006576 005046 CLR -(SP)  
006600 012746 MOV (PC)+, -(SP)  
006602 027 005 . BYTE 23. , 5  
006604 104377 EMT 0<377>  
593  
594 ; *****  
595 ; READ AND CHECK FOR ERRORS FROM ALL LIVE TERMINALS  
596 ; *****  
597 006606 012737 040102 000740' MOV #40102, BUFF ; PCB FOR READ  
598 006614 005037 000734' CLR ERRFLG ; RESET ERROR INDICATOR  
599 006620 005001 CLR R1 ; POINTER TO TERMINAL TABLES  
600 006622 005003 CLR R3 ; POINTER TO ERROR BLOCK  
601 006624 012737 003000 000216' MOV #10. RPC, Q10DPB+10. FUN ; LOAD DPB WITH READ FUNCTION  
602 006632 012737 000744' 000224' MOV #10ST, Q10DPB+10. 1ST ; LOAD STATUS WORD ADDRESS  
603 006640 012737 007634' 000226' MOV #AST, Q10DPB+10. AST ; LOAD READ AST ADDRESS  
604 006646 012737 000016 000232' MOV #14. , Q10DPB+10. A2 ; LOAD INPUT BUFFER SIZE  
605 006654 012737 000740' 000236' MOV #BUFF, Q10DPB+10. A4 ; LOAD OUTPUT BUFFER ADDRESS  
606 006662 012737 000004 000240' MOV #4, Q10DPB+10. A5 ; LOAD OUTPUT BUFFER SIZE  
607 006670 005761 000770' 75: TST OFFLIN(R1) ; CHECK OFFLINE TABLE  
608 006674 001111 BNE 95 ; BRANCH IF OFFLINE  
609 006676 012737 001170' 000710' MOV #ERRBLK, TEMP ; GET ADDRESS OF ERROR BLOCK  
610 006704 060337 000710' ADD R3, TEMP ; UPDATE ADDRESS OF EACH TERMINAL  
611 006710 013737 000710' 000230' MOV TEMP, Q10DPB+10. A1 ; GET UPDATED ADDRESS OF ERROR BLOCK  
612 006716 116137 000477' 000220' MOV DEVTL+1(R1), Q10DPB+10. LUN ; LOAD LOGICAL UNIT NUMBER  
613 006724 255:  
614 006724 DIR5 #Q10DPB ; ISSUE I/O REQUEST  
006724 012746 000214' MOV #Q10DPB, -(SP)  
006730 104377 EMT 0<377>  
615 006732 103020 BCC 85 ; REQUEST ACCEPTED  
616 006734 013737 000000G 000724' MOV $DSW, TEMP7 ; GET DSW ERROR  
617 006742 PRINTF #MSG10, TEMP7 ; QIO NOT ACCEPTED  
006742 013746 000724' MOV TEMP7, -(SP)  
006746 012746 010751' MOV #MSG10, -(SP)  
006752 012746 000002 MOV #2, -(SP)  
006756 010600 MOV SP, R0  
006760 104017 EMT C$PNTF
```

```

006762 062706 000006
618 006766 005261 000770'
619 006772 000452
620 006774
006774 005046
006776 012746 000001
007002 012746 000005
007006 012746 000001
007012 012746
007014 027 005
007016 104377
621 007020 005737 000732'
622 007024 001006
623 007026
007026 012746 000001
007032 012746
007034 051 002
007036 104377
624 007040 000755
625 007042 005037 000732'
626 007046 005737 000734'
627 007052 001422
628 007054 005037 000734'
629 007060 005737 000730'
630 007064 001717
631 007066 005261 000770'
632 007072
007072 005046
007074 156116 000476'
007100 012746 010513'
007104 012746 000002
007110 010600
007112 104017
007114 062706 000006
633 007120 005721
634 007122 062703 000016
635 007126 020102
636 007130 001257
637 007132 005001
638 007134 005003
639 007136 005037 000734'
640 007142 005005
641 007144 010304
642 007146 105764 001176'
643 007152 001513
644 007154 005737 000734'
645 007160 001024
646 007162
007162 013746 000736'
007166 012746 010621'
007172 012746 000002
007176 010600
007200 104017
007202 062706 000006
647 007206
007206 012746 010676'
007212 012746 000001
    ADD #6, SP
    INC OFFLIN(R1) ;DROP TERMINAL
    BR 9$
    MRKTSS #1., #5., #1 ;MARK FOR SIG EVENT
    CLR -(SP)
    MOV #1., -(SP)
    MOV #5., -(SP)
    MOV #1., -(SP)
    MOV (PC)+, -(SP)
    . BYTE 23., 5
    EMT 0<377>
    TST ASTFLG ;CHECK FOR I/O COMPLETION (AST)
    BNE 50$ ;BRANCH IF AST COMPLETE
    WTSESS #1. ;WAIT FOR SIG EVENT
    MOV #1., -(SP)
    MOV (PC)+, -(SP)
    . BYTE 41., 2
    EMT 0<377>
    BR 8$
    CLR ASTFLG ;RESET AST INDICATOR
    TST ERRFLG ;ANY STATUS CODE ERROR?
    BEQ 9$ ;BRANCH IF NO ---
    CLR ERRFLG ;YES - TERMINAL OFFLINE, THEN RESET ERROR FLAG
    TST MINFLG ;IS 2 MINS UP?
    BEQ 25$ ;IF NO- AGAIN ISSUE I/O REQUEST
    INC OFFLIN(R1) ;SET OFFLINE FLAG
    PRINTF #MSG6, <B, DEVTBL+O(R1)> ;INFORM OPERATOR TERMINAL DIED
    CLR -(SP)
    BISB DEVTBL+O(R1), (SP)
    MOV #MSG6, -(SP)
    MOV #2., -(SP)
    MOV SP, RO
    EMT C$PNTF
    ADD #6, SP
    TST (R1)+ ;GET NEXT TERMINAL
    ADD #14., R3 ;GET NEXT ERROR BLOCK
    CMP R1, R2 ;LAST TERMINAL?
    BNE 7$ ;GO CHECK PRESENT TERMINAL IF NO
    CLR R1 ;INIT DEVICE TABLE POINTER
    CLR R3 ;INIT POINTER TO ERROR BLOCK
    CLR ERRFLG ;RESET ERROR INDICATOR
    CLR R5 ;INIT TERMINAL ERROR COUNT
    MOV R3, R4 ;ERROR POINTER
    TSTB ERRBLK+6(R4) ;CHECK FOR ERRORS IN TABLE
    BEQ 13$ ;IF NO ERRORS-GET NEXT ERROR BLOCK
    TST ERRFLG ;IS HEADER PRINTED?
    BNE 11$ ;YES - PRINT ERRORS
    PRINTF #MSG7, PASS ;PRINT HEADER
    MOV PASS, -(SP)
    MOV #MSG7, -(SP)
    MOV #2., -(SP)
    MOV SP, RO
    EMT C$PNTF
    ADD #6, SP
    PRINTF #MSG8 ;PRINT SUB-HEAD
    MOV #MSG8, -(SP)
    MOV #1., -(SP)
    8$:
    50$:
    9$:
    12$:
    10$:
    11$:
    13$:
    14$:
    15$:
    16$:
    17$:
    18$:
    19$:
    20$:
    21$:
    22$:
    23$:
    24$:
    25$:
    26$:
    27$:
    28$:
    29$:
    30$:
    31$:
    32$:
    33$:
    34$:
    35$:
    36$:
    37$:
    38$:
    39$:
    40$:
    41$:
    42$:
    43$:
    44$:
    45$:
    46$:
    47$:
    48$:
    49$:
    50$:
    51$:
    52$:
    53$:
    54$:
    55$:
    56$:
    57$:
    58$:
    59$:
    60$:
    61$:
    62$:
    63$:
    64$:
    65$:
    66$:
    67$:
    68$:
    69$:
    70$:
    71$:
    72$:
    73$:
    74$:
    75$:
    76$:
    77$:
    78$:
    79$:
    80$:
    81$:
    82$:
    83$:
    84$:
    85$:
    86$:
    87$:
    88$:
    89$:
    90$:
    91$:
    92$:
    93$:
    94$:
    95$:
    96$:
    97$:
    98$:
    99$:
    100$:
    
```

007216	010600				MOV	SP, R0	
007220	104017				ENT	CSPNTF	
007222	062706	000004			ADD	#4, SP	
648	007226	005237	000734'		INC	ERRFLG	; HEADER PRINT INDICATOR
649							*****
650							; SAVE ALL ERRORS FOR PRINT-OUT...
651							*****
652	007232	116137	000476'	000712'	115:	MOVB	DEVTBL(R1), TEMP1 ; SAVE TERMINAL SB NUMBER
653	007240	116337	001174'	000714'		MOVB	ERRBLK+4(R3), TEMP2 ; SAVE ASCII-TERMINAL TYPE
654							-----
655							; TRANSFORM ASCII CHARACTER TO TERMINAL TYPE
656							-----
657							
658	007246	042737	177700	000714'		BIC	#177700, TEMP2 ; 101 BECOMES 001
659							; 102 BECOMES 002
660							; 103 BECOMES 003
661	007254	006337	000714'			ASL	TEMP2 ; 001 BECOMES 002
662							; 002 BECOMES 004
663							; 003 BECOMES 006
664	007260	005337	000714'			DEC	TEMP2 ; 002 BECOMES 001
665							; 004 BECOMES 003
666							; 006 BECOMES 005
667	007264	062737	001440	000714'		ADD	#800., TEMP2 ; 001 BECOMES 801
668							; 003 BECOMES 803
669							; 004 BECOMES 805
670							
671	007272	116437	001176'	000716'		MOVB	ERRBLK+6(R4), TEMP4 ; SAVE HI-BYTE TEST NUMBER IN ASCII
672	007300	116437	001177'	000720'		MOVB	ERRBLK+7(R4), TEMP5 ; SAVE LO-BYTE TEST NUMBER IN ASCII
673	007306	042737	000300	000716'		BIC	#300, TEMP4 ; TRANSFORM ASCII
674	007314	042737	000300	000720'		BIC	#300, TEMP5 ; TRANSFORM ASCII
675							*****
676							; PRINT ALL ERRORS AND END PASS
677							*****
678	007322					PRINTF	#FMT, <B, TEMP1>, TEMP2, <B, TEMP4>, <B, TEMP5>
	007322	005046				CLR	-(SP)
	007324	153716	000720'			BISB	TEMP5, (SP)
	007330	005046				CLR	-(SP)
	007332	153716	000716'			BISB	TEMP4, (SP)
	007336	013746	000714'			MOV	TEMP2, -(SP)
	007342	005046				CLR	-(SP)
	007344	153716	000712'			BISB	TEMP1, (SP)
	007350	012746	011011'			MOV	#FMT, -(SP)
	007354	012746	000005			MOV	#5, -(SP)
	007360	010600				MOV	SP, R0
	007362	104017				ENT	CSPNTF
	007364	062706	000014			ADD	#14, SP
679	007370	005205				INC	R5 ; UPDATE TERMINAL ERROR COUNT
680	007372	005724				TST	(R4)+ ; UPDATE POINTER BY 2
681	007374	020527	000004			CMP	R5, #4 ; CHECK IF END OF THIS BLOCK
682	007400	001262				BNE	10\$; IF NOT END, CHECK FOR MORE ERRORS
683	007402	062703	000016		135:	ADD	#14., R3 ; GET NEXT ERROR BLOCK
684	007406	005721				TST	(R1)+ ; UPDATE TERMINAL POINTER
685	007410	005005				CLR	R5 ; RESET ERROR COUNTER
686	007412	020102				CMP	R1, R2 ; CHECK FOR END OF ERROR TABLE
687	007414	001253				BNE	12\$; CONTINUE CHECKING ERRORS IF NOT END
688	007416	005737	000730'			TST	MINFLG ; IS 2 MINS UP?
689	007422	001006				BNE	27\$; BRANCH IF YES

```
690 007424 CMKTSS ,,ERR ;ELSE CANCEL 2 MINS PERIOD
      007424 012746 MOV (PC)+,-(SP)
      007426 033 001 .BYTE 27.,1
      007430 104377 EMT 0<377>
      007432 103002 .IIF EQ 0<$$$T1-37>, BCC .+6
      007434 004737 007606' JSR PC,ERR
691 ;*****
692 ;PRINT PASS ONLY FOR ACTIVE TERMINALS
693 ;*****
694 007440 005004 275: CLR R4 ;POINTER TO OFFLINE TERMINALS
695 007442 005001 CLR R1 ;POINTER TO OFFLINE TABLE
696 007444 005761 000770' 305: TST OFFLIN(R1) ;IS TERMINAL OFFLINE?
697 007450 001423 BEQ 265 ;BRANCH IF NO AND DO END OF PASS
698 007452 005204 INC R4 ;KEEP TRACK OF NON-ACTIVE TERMINALS
699 007454 005721 TST (R1)+ ;UPDATE POINTER TO OFFLINE TABLE
700 007456 023704 000676' CMP UNITS,R4 ;LAST TERMINAL?
701 007462 001370 BNE 305 ;CONTINUE CHECKING IF NO
702 007464 005737 000722' TST TEMP6 ;IS PRINT FLAG SET
703 007470 001012 BNE 245 ;IF YES DO NOT PRINT
704 007472 PRINTF #NACT ;INFORM OF NO ACTIVE TERMINALS
      007472 012746 011113' MOV #NACT,-(SP)
      007476 012746 000001 MOV #1,-(SP)
      007502 010600 MOV SP,RO
      007504 104017 EMT C$PNTF
      007506 062706 000004 ADD #4,SP
705 007512 005237 000722' INC TEMP6 ;SET PRINT FLAG
706 007516 000427 245: BR 465 ;DO NEXT PASS
707 007520 005037 000722' 265: CLR TEMP6 ;RESET PRINT FLAG
708 007524 GTIMSS #BUF ;GET THE SYSTEM TIME
      007524 012746 000750' MOV #BUF,-(SP)
      007530 012746 MOV (PC)+,-(SP)
      007532 075 002 .BYTE 61.,2
      007534 104377 EMT 0<377>
709 007536 PRINTX #EOP,PASS,BUF+G.TIHR,BUF+G.TIMI ;REPORT PASS COUNT
      007536 013746 000760' MOV BUF+G.TIMI,-(SP)
      007542 013746 000756' MOV BUF+G.TIHR,-(SP)
      007546 013746 000736' MOV PASS,-(SP)
      007552 012746 011042' MOV #EOP,-(SP)
      007556 012746 000004 MOV #4,-(SP)
      007562 010600 MOV SP,RO
      007564 104015 EMT C$PNTX
      007566 062706 000012 ADD #12,SP
710 007572 005237 000736' 465: INC PASS ;UPDATE PASS COUNT
711 007576 BREAK EMT C$BRK
      007576 104022 JMP START ;GO DO NEXT PASS
712 007600 000137 006274'
713
714
715
716 007604 ENDTST
      007604 L10006: EMT C$SETST
      007604 104001
717 ;
718 ; *****
719 ; IF CANCEL MARK TIME DIRECTIVE IS NOT ACCEPTED
720 ; ABORT PROGRAM, AND EXIT TO MCR.
721 ; *****
```

```
722  
723 007606 ;  
724 007606 ERR: PRINTF #MABORT ;PRINT ABORT MESSAGE  
007606 012746 010342' MOV #MABORT,-(SP)  
007612 012746 000001 MOV #1,-(SP)  
007616 010600 MOV SP,R0  
007620 104017 EMT C$PNTF  
007622 062706 000004 ADD #4,SP  
725 007626 EXIT$$ ;AND EXIT  
007626 012746 MOV (PC)+,-(SP)  
007630 063 001 .BYTE 51,1  
007632 104377 EMT 0<377>
```

727
728
729
730
731
732
733
734
735
736
737
738
739

```
*****  
++AST SERVICE ROUTINE...  
THIS SYSTEM TRAP IS ENTERED UPON A WRITE/READ I/O REQUEST  
COMPLETION. UPON ENTRY, A TWO WORD LOCATION CONTAINS I/O  
ERROR CODES. AN ERROR FLAG IS SET, FOR ANY TERMINAL  
GENERATING ERRORS AT COMPLETION OF AN I/O REQUEST. AT EXIT,  
THE ROUTINE INDICATES THAT THE TRAP HAS OCCURRED, AND THE  
STACK POINTER IS READJUSTED, FOR THE FOLLOWING AST...  
*****
```

740 007634
741 007634 105737 000744'
742 007640 100002
743 007642 005237 000734'
744 007646 005237 000732'
745 007652 005726
746 007654
007654 012746
007656 163 001
007660 104377

```
AST: ; TERMINAL WRITE-READ AST  
TSTB IOST ; CHECK FOR ERROR CODE  
BPL 1$ ; BRANCH IF NO ERRORS  
INC ERRFLG ; SET ERROR INDICATOR  
1$: INC ASTFLG ; SET AST INDICATION  
TST (SP)+ ; AS PER RSX-ASTXSS  
ASTXSS ; RETURN  
MOV (PC)+, -(SP)  
. BYTE 115, 1  
EMT 0<377>
```

747
748
749
750
751
752
753

```
*****  
; THIS MARK-TIME IS ENTERED UPON COMPLETION OF A 2 MINUTE  
; PERIOD. HERE A 2 MINUTE COMPLETION FLAG IS SET.  
*****
```

754 007662
755 007662 005237 000730'
756 007666 005726
757 007670
007670 012746
007672 163 001
007674 104377

```
MAST: ; SET 2 MINUTE FLAG  
INC MINFLG ; INC AS PER RSX-ASTXSS  
TST (SP)+ ; RETURN  
ASTXSS  
MOV (PC)+, -(SP)  
. BYTE 115, 1  
EMT 0<377>
```

759
760 007676
007676
761
762 007676
007676
007676 104012
763

BGNCLN
LSCLEAN: :

ENDCLN
L10007:

EMT CSCLEAN

765	007700	045	116	045	MSG0: .ASCIZ "%N%ACONTROLLER %D1%A, TERMINAL %D3%A IS NOT SYSGENED IN. %N"
	007703	101	103	117	
	007706	116	124	122	
	007711	117	114	114	
	007714	105	122	040	
	007717	045	104	061	
	007722	045	101	054	
	007725	040	124	105	
	007730	122	115	111	
	007733	116	101	114	
	007736	040	045	104	
	007741	063	045	101	
	007744	040	111	123	
	007747	040	116	117	
	007752	124	040	123	
	007755	131	123	107	
	007760	105	116	105	
	007763	104	040	111	
	007766	116	056	045	
	007771	116	000		
766	007773	045	116	045	MSG1: .ASCIZ "%N%ATERMINAL %D2%A CAINOT BE ASSIGNED-%DSW=%06"
	007776	101	124	105	
	010001	122	115	111	
	010004	116	101	114	
	010007	040	045	104	
	010012	062	045	101	
	010015	040	103	101	
	010020	116	116	117	
	010023	124	040	102	
	010026	105	040	101	
	010031	123	123	111	
	010034	107	116	105	
	010037	104	055	044	
	010042	104	123	127	
	010045	075	045	117	
	010050	066	000		
767	010052	045	116	045	MSG2: .ASCII "%N%ANUMBER OF ENTERED TERMINALS=%D2%N"
	010055	101	116	125	
	010060	115	102	105	
	010063	122	040	117	
	010066	106	040	105	
	010071	116	124	105	
	010074	122	105	104	
	010077	040	124	105	
	010102	122	115	111	
	010105	116	101	114	
	010110	123	075	045	
	010113	104	062	045	
	010116	116			
768	010117	045	116	045	.ASCIZ "%N%AUNIT ** CONTROLLER ** LOGICAL SB ** LUN **"
	010122	101	125	116	
	010125	111	124	040	
	010130	052	052	040	
	010133	103	117	116	
	010136	124	122	117	
	010141	114	114	105	
	010144	122	040	052	

	010147	052	040	114	
	010152	117	107	111	
	010155	103	101	114	
	010160	040	123	102	
	010163	040	052	052	
	010166	040	114	125	
	010171	116	040	052	
	010174	052	000		
769	010176	045	116	045	MSG3: . ASCIZ "%N%D3%A %D3%A %D3%A %D3%A"
	010201	104	063	045	
	010204	101	040	040	
	010207	040	040	040	
	010212	040	040	040	
	010215	040	040	040	
	010220	040	045	104	
	010223	063	045	101	
	010226	040	040	040	
	010231	040	040	040	
	010234	040	040	040	
	010237	040	045	104	
	010242	063	045	101	
	010245	040	040	040	
	010250	040	040	040	
	010253	045	104	063	
	010256	045	116	000	
770	010261	045	116	045	MSG: . ASCIZ "%N%ATERMINAL %D2%A CANNOT BE ATTACHED-\$DSW=%03%A"
	010264	101	124	105	
	010267	122	115	111	
	010272	116	101	114	
	010275	040	045	104	
	010300	062	045	101	
	010303	040	103	101	
	010306	116	116	117	
	010311	124	040	102	
	010314	105	040	101	
	010317	124	124	101	
	010322	103	110	105	
	010325	104	055	044	
	010330	104	123	127	
	010333	075	045	117	
	010336	063	045	116	
	010341	000			
771	010342	045	116	045	MABORT: . ASCIZ "%N%APROGRAM IS ABORTED%A"
	010345	101	120	122	
	010350	117	107	122	
	010353	101	115	040	
	010356	111	123	040	
	010361	101	102	117	
	010364	122	124	105	
	010367	104	045	116	
	010372	000			
772	010373	045	116	045	MSG4: . ASCIZ "%N%ATERMINAL %D3%A IS NOT RESPONDING %N"
	010376	101	124	105	
	010401	122	115	111	
	010404	116	101	114	
	010407	011	045	104	
	010412	063	045	101	

	010415	040	111	123	
	010420	040	116	117	
	010423	124	040	122	
	010426	105	123	120	
	010431	117	116	104	
	010434	111	116	107	
	010437	040	045	116	
	010442	000			
773	010443	045	116	045	MSG5: .ASCIZ "%N%ATERMINAL %D3%A IS NOW RESPONDING %N"
	010446	101	124	105	
	010449	122	115	111	
	010454	116	101	114	
	010457	011	045	104	
	010462	063	045	101	
	010465	040	111	123	
	010470	040	116	117	
	010473	127	040	122	
	010476	105	123	120	
	010501	117	116	104	
	010504	111	116	107	
	010507	040	045	116	
	010512	000			
774	010513	045	116	045	MSG6: .ASCII "%N%ATERMINAL %D3%A DID NOT COME BACK ON LINE "
	010516	101	124	105	
	010521	122	115	111	
	010524	116	101	114	
	010527	011	045	104	
	010532	063	045	101	
	010535	040	104	111	
	010540	104	040	116	
	010543	117	124	040	
	010546	103	117	115	
	010551	105	040	102	
	010554	101	103	113	
	010557	040	117	116	
	010562	040	114	111	
	010565	116	105	040	
775	010570	106	122	117	.ASCIZ "FROM DIAGNOSTIC TESTS %N"
	010573	115	040	104	
	010576	111	101	107	
	010601	116	117	123	
	010604	124	111	103	
	010607	040	124	105	
	010612	123	124	123	
	010615	040	045	116	
	010620	000			
776	010621	045	116	045	MSG7: .ASCIZ "%N%ATERMINAL DIAGNOSTIC ERRORS ON PASS %D5%N"
	010624	101	124	105	
	010627	122	115	111	
	010632	116	101	114	
	010635	040	104	111	
	010640	101	107	116	
	010643	117	123	124	
	010646	111	103	040	
	010651	105	122	122	
	010654	117	122	123	
	010657	040	117	116	

	010662	040	120	101			
	010665	123	123	040			
	010670	045	104	065			
	010673	045	116	000			
777	010676	045	116	045	MSG8:	. ASCIZ	"%N%ASB #
	010701	101	123	102			TERMINAL-TYPE
	010704	040	043	011			FAILING TEST: %N"
	010707	040	040	040			
	010712	124	105	122			
	010715	115	111	116			
	010720	101	114	055			
	010723	124	131	120			
	010726	105	011	011			
	010731	106	101	111			
	010734	114	111	116			
	010737	107	040	124			
	010742	105	123	124			
	010745	072	045	116			
	010750	000					
778	010751	045	116	045	MSG10:	. ASCIZ	"%N%AQ10 IS REJECTED--%DSW=%06%N"
	010754	101	121	111			
	010757	117	040	111			
	010762	123	040	122			
	010765	105	112	105			
	010770	103	124	105			
	010773	104	055	055			
	010776	044	104	123			
	011001	127	075	045			
	011004	117	066	045			
	011007	116	000				
779	011011	045	116	045	FMT:	. ASCIZ	"%N%D3%S12%D3%S28%01%01%N"
	011014	104	063	045			
	011017	123	061	062			
	011022	045	104	063			
	011025	045	123	062			
	011030	070	045	117			
	011033	061	045	117			
	011036	061	045	116			
	011041	000					
780	011042	045	116	045	EOP:	. ASCII	"%N%AREND OF PASS %06"
	011045	101	105	116			
	011050	104	040	117			
	011053	106	040	120			
	011056	101	123	123			
	011061	040	045	104			
	011064	066					
781	011065	045	101	054		. ASCIZ	"%A, TIME=%02%A: %Z2%N"
	011070	040	040	124			
	011073	111	115	105			
	011076	075	045	104			
	011101	062	045	101			
	011104	072	045	132			
	011107	062	045	116			
	011112	000					
782	011113	045	116	045	NACT:	. ASCIZ	"%N%ANO ACTIVE TERMINALS ON THE BUS!!!!%N"
	011116	101	116	117			
	011121	040	101	103			

011124	124	111	126	
011127	105	040	124	
011132	105	122	115	
011135	111	116	101	
011140	114	123	040	
011143	117	116	040	
011146	124	110	105	
011151	040	102	125	
011154	123	041	041	
011157	041	041	041	
011162	045	116	000	
783				. EVEN
784				
785				
786	011166	LASTAD		; GENERATES 1'ST WORD OF FREE MEMORY
				. EVEN
	011166	L\$LAST::		
787		. BLKW	2024	
788				
789				
790	000001	. END		

ALUDP8 000244R
AST 007634R
ASTFLG 000732R
A. LULU= 000002
A. LUNA= 000004
A. LUNU= 000006
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BUF 000750R
BUFF 000740R
CSABRT= 000021
CSADR = 000020
CSAU = 000071
CSBRK = 000022
CSBSEG= 000004
CSBSUB= 000002
CSBUFF= 000034
CSCEFG= 000061
CSCLER= 000012
CSCLP1= 000006
CSVEC = 000043
CSDCLN= 000057
CSDODU= 000070
CSDRPT= 000024
CSDSTI= 000033
CSDU = 000072
CSEDIT= 000004
CSENTI= 000032
CSERDF= 000002
CSERHR= 000003
CSERSF= 000001
CSERSO= 000004
CSESCA= 000010
CSESEG= 000005
CSESUB= 000003
CSETST= 000001
CSEXIT= 000036
CSGMAN= 000056
CSGMEM= 000050
CSGPHR= 000055
CSGPRI= 000045
CSGTIM= 000066
CSINIT= 000011

CSINLP= 000020
CSKWF= 000042
CSKWON= 000041
CSLGF= 000027
CSLGN= 000026
CSLOOP= 000100
CSMANI= 000065
CSMPME= 000051
CSMSG = 000023
CSPNTB= 000014
CSPNTF= 000017
CSPNTN= 000013
CSPNTS= 000016
CSPNTX= 000015
CSPOIN= 000040
CSPWR = 000073
CSQIO = 000377
CSREFG= 000064
CSRELA= 000052
CSRELM= 000053
CSRELO= 000037
CSREQT= 000060
CSRESE= 000040
CSREVI= 000001
CSRPT = 000025
CSSEFG= 000062
CSSPRI= 000046
CSSVEC= 000044
CSUNBU= 000035
CSWEFG= 000063
CSWTM = 000030
CSWTU = 000031
DEVTBL 000476R
EOP 011042R
ERR 007606R
ERRBLK 001170R
ERRBRT 000262R
ERRFLG 000734R
FMT 011011R
FSAU = 000015
FSBGN = 000040
FSCLEA= 000007
FSDU = 000016
FSEND = 000041
FSHARD= 000004
FSHW = 000013
FSINIT= 000006
FSMOD = 000000
FSMSG = 000011
FSPWR = 000017
FSRPT = 000012
FSSEG = 000003
FSSOFT= 000005
FSSRV = 000010
FSSUB = 000002
FSSW = 000014
FSTEST= 000001

GETCNT 000100R
GETERM 000132R
GLUDP8 000254R
GSEXCP= 000400
GSHILI= 000002
GSLOLI= 000001
GSNO = 000000
GSOFFS= 000400
GSOFSI= 000376
GSPRMA= 000001
GSPRMD= 000002
GSPRML= 000000
GSRADA= 000140
GSRADB= 000000
GSRADD= 000040
GSRADF= 000200
GSRADL= 000120
GSRADO= 000020
GSRADT= 000100
GSXFER= 000004
GYES = 000010
G. LUBA= 000004
G. LUBL= 000006
G. LUCH= 000004
G. LUFB= 000003
G. LULU= 000002
G. LUNA= 000000
G. LUNU= 000002
G. TICP= 000016
G. TICT= 000014
G. TIDA= 000004
G. TIHR= 000006
G. TIMI= 000010
G. TIMO= 000002
G. TISC= 000012
G. TIYR= 000000
IBUFO 002770R
IBUF1 003170R
IBUF2 003370R
IBUF3 003570R
IBUF4 003770R
IBUF5 004170R
IBUF6 004370R
IBUF7 004570R
IOST 000744R
IO. AST= 000012
IO. ATT= ***** GX
IO. A1 = 000014
IO. A2 = 000016
IO. A3 = 000020
IO. A4 = 000022
IO. A5 = 000024
IO. DET= ***** GX
IO. EFN= 000006
IO. FUN= 000002
IO. IST= 000010
IO. LUN= 000004

IO. RPC= 003000
IO. RXP= 003400
IO. WPC= 002400
ISAU = 000041
ISCLN = 000041
ISDU = 000041
ISHRD = 000041
ISINIT= 000041
ISMOD = 000041
ISMSG = 000041
ISPR = 000041
ISRPT = 000041
ISSEG = 000041
ISSFT = 000041
ISSRV = 000041
ISSUB = 000041
ISTST = 000041
LUN = 000011
LSAUT 000070RG
LSCCP 000044RG
LSCLEA 007676RG
LSDEPO 000011RG
LSDEVP 000052RG
LSDISP 005032RG
LSDR 000202RG
LSDRCT 000062RG
LSDRS 000064RG
LSDRST 000206RG
LSDTP 000040RG
LSDUT 000072RG
LSDVTY 000210RG
LSEF 000024RG
LSEXP1 000032RG
LSEXP2 000034RG
LSEXP3 000036RG
LSHARD 004772RG
LSHPCP 000046RG
LSHPTP 000056RG
LSHW 005020RG
LSICP 000042RG
LSINIT 005034RG
LSLADP 000076RG
LSLAST 011166RG
LSMREV 000012RG
LSNAME 000000RG
LSPWRU 000074RG
LSREPP 000054RG
LSREV 000010RG
LSRPT 005026RG
LSSOFT 005024RG
LSSPC 000030RG
LSSPCP 000050RG
LSSPTP 000060RG
LSSTA 000066RG
LSSW 005026RG
LSTIML 000022RG
LSTIMU 000020RG

LSTIM1 000016RG
LSUNIT 000014RG
L10000 005016R
L10001 005022R
L10002 005024R
L10003 005026R
L10004 005026R
L10005 006272R
L10006 007604R
L10007 007676R
MABORT 010342R
MAST 007662R
MINFLG 000730R
MSG 010261R
MSG0 007700R
MSG1 007773R
MSG10 010751R
MSG2 010052R
MSG3 010176R
MSG4 010373R
MSG5 010443R
MSG6 010513R
MSG7 010621R
MSG8 010676R
NACT 011113R
NR = 000001
OFFLIN 000770R
OSAPTS= 000000
OSAU = 000000
OSBGNR= 000000
OSBGN= 000000
OSDU = 000000
OSGNSW= 000000
OSPOIN= 000001
OSPR = 000000
PASFG 000726R
PASS 000736R
PCBADR 000704R
PCBCNT 000706R
QIODPB 000214R
Q. IOAE= 000012
Q. IOEF= 000006
Q. IOFN= 000002
Q. IOLU= 000004
Q. IOPL= 000014
Q. IOPR= 000007
Q. IOSB= 000010
SCBADR 000700R
SCBCNT 000702R
STADR 000000RG
START 006274R
SVCNT= 177777
SVCBL= 000001
SVCINS= 000001
SVCSTK= 177777
SVCSUB= 000001
SVCTAG= 000001

SVCTST= 000001	T\$CODE= 000052	T\$SAVL= 177777	T\$SHW = 010001	X\$ALWA= 000000
\$SLSYM= 010000	T\$ERRN= 000000	T\$SEGL= 177777	T\$SINI= 010005	X\$FALS= 000040
TEMP 000710R	T\$EXCP= 000000	T\$SUBN= 000000	T\$SRPT= 010004	X\$OFFS= 000400
TEMP1 000712R	T\$HILI= 000077	T\$TAGL= 177777	T\$SSOF= 010002	X\$TRUE= 000020
TEMP2 000714R	T\$LOLI= 000001	T\$TAGN= 010010	T\$SSW = 010003	\$DSW = ***** GX
TEMP4 000716R	T\$LSYM= 010000	T\$TEMP= 000007	T\$STES= 010006	\$SARG= 000004
TEMP5 000720R	T\$MCAL= 000000	T\$TEST= 000001	T1 006274RG	\$SOST= 000020
TEMP6 000722R	T\$NEST= 177777	T\$SCLE= 010007	UNITS 000676R	\$SST1 = 000037
TEMP7 000724R	T\$NSKO= 000007	T\$SHAR= 010000	UNTBL 000276R	\$SST2 = 000250R
T\$ARGC= 000001				

. ABS. 000000 000
015236 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 21948 WORDS (86 PAGES)

DYNAMIC MEMORY: 21140 WORDS (81 PAGES)

ELAPSED TIME: 00:02:26

DZKC1, DZKC1=DIAGSUPER/ML, DZKC1