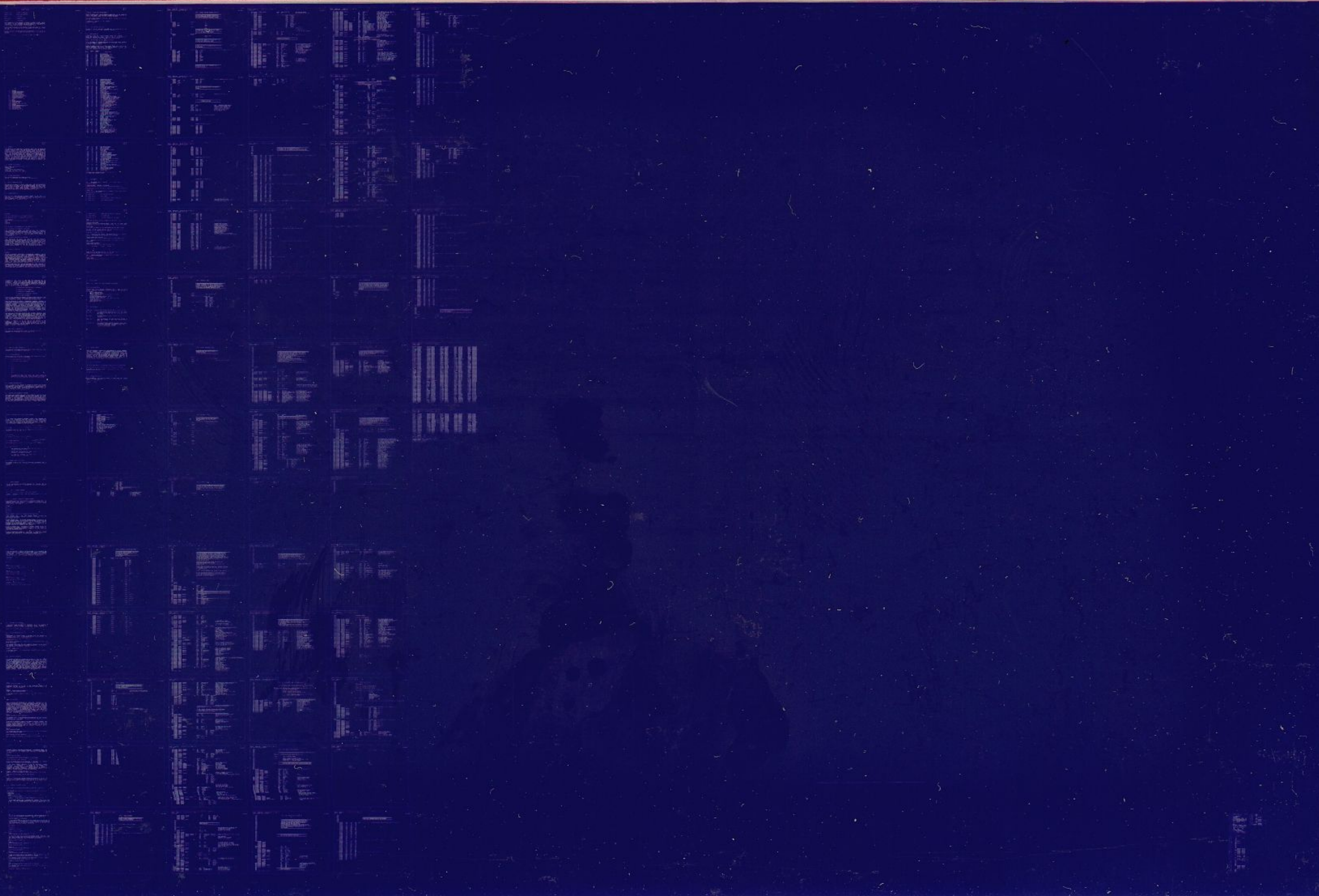


PDP11

SERIAL BUS EXERCISER
MD-11-DZKCH-A

EP-DZKCH-A-DL-A
COPYRIGHT © 1977
FICHE 1 OF 1

DEC 1977
digital
MADE IN USA



PRODUCT CODE: MAINDEC-11-DZKCH-A-D
PRODUCT NAME: SERIAL BUS EXERCISER
DATE: JUNE 1977
MAINTAINER: DIAGNOSTIC GROUP
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 DOCUMENT

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	HARDWARE REQUIREMENTS
3.0	SOFTWARE REQUIREMENTS
4.0	PROGRAM GENERATION
4.1	DIAGNOSTIC SUPERVISOR
5.0	PDP11 FAMILY COMPATIBILITY
6.0	OPERATING SYSTEM COMPATIBILITY
7.0	TESTING METHODOLOGY
8.0	PROGRAM CONSIDERATIONS
9.0	XXDP
10.0	ACT/APT
11.0	MEMORY MANAGEMENT
12.0	EXECUTION TIME
13.0	PROGRAM CONTROL PROCEDURES
14.0	LOADING
15.0	PROGRAM PARAMETER SELECTION
16.0	SWITCH REGISTER FUNCTIONS
17.0	ERROR REPORTING
18.0	TEST DESCRIPTION
19.0	PROGRAM TERMINATION

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 CURRENT 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 SUPER 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 COMPATIBILITY, 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 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.

8.1.1 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 THEREFORE 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 -

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.

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

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.

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 : (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

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'URN.

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 (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 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 (0) ? 63 <CR>
UNIT 1
ENTER CONTROLLER NUMBER : (0) ? <CP>
```

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 DZKCH

DCP-A>STA

NUMBER UNITS (0)?: 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-SDSW=Y"

WHERE THE ERROR CODES Y ARE RETURNED BY DIRECTIVES IN THE DIRECTIVE STATUS WORD (SDSW). 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 PRECEDED 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
.DAG	-13	363	DATA OVERRUN

SRE	-14	362	SEND/RECEIVE FAILURE
ABO	-15	361	REQUEST TERMINATED
PRI	-16	360	PRIVILEGE VIOLATION
RSU	-17	357	SHAREABLE RESOURCE IN USE
OVR	-18	356	ILLEGAL OVERLAY REQUEST
BYT	-L9	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 (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 REQUEST 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 ASSIGNS 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

ERROR-TYPE 3, ERROR-TYPE 5 ARE CONSIDERED 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.

18.2 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>

2-	32	DIAGNOSTIC SUPERVISOR HEADER
3-	43	EQUATES
5-	89	P-TABLE MESSAGES
6-	103	GENERAL AND DPB STORAGE LOCATIONS
7-	285	HARDWARE P-TABLE
8-	306	DEFAULT HARDWARE P-TABLE
9-	323	SOFTWARE P-TABLE
10-	346	DISPATCH TABLE
11-	360	INIT CODE
13-	623	MAIN ROUTINE
14-	714	DECLARE A SIGNIFICANT EVENT PER 5 MINS.
15-	737	SET BYTE PATTERN IN OUTPUT BUFFER
16-	774	GET NEXT DEVICE AND LUN INFORMATION
17-	806	POLL AND SET OUTPUT BUFFERS
18-	849	CHECK ERRORS-CHECK FOR BAD CRC
19-	983	CLEANUP CODE
20-	999	CRC-AST ROUTINE
21-	1029	AST SERVICE ROUTINE
25-	1164	PRINT MODULE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
23

000000

000001
000001
000001
000001
000001
000001

SVC

SVCINS= 1
SVCTST= 1
SVC SUB= 1
SVCGBL= 1
SVCTAG= 1

. TITLE DZKCH
. ENABL AMA
. NLIST MD, ME
. LIST MEB
. MCALL SVC
. MCALL ASTXSS, QIOS, ALUNS, DIRS, GLUNS, EXITSS
. MCALL MRKTSS, CMKTSS, GTIMSS
. MCALL WTSESS
. GLOBL STADR

;LIST GENERATED SYMBOLS
;LIST TEST NUMBERS
;LIST SUBTEST NUMBERS
;LIST BLOBALS
;LIST GENERATED TAGS

32
 33
 34
 35
 36
 37
 38
 39 000000
 40
 41 000000
 000000
 000000 104
 000001 132
 000002 113
 000003 103
 000004 110
 000005 000
 000006 000
 000007 000
 000010
 000010 101
 000011
 000011 060
 000012
 000012 001
 000013 004
 000014
 000014 000000
 000016
 000016 000063
 000020
 000020 000300
 000022
 000022 000001
 000024
 000024 000000
 000026 000000
 000030
 000030 000000
 000032
 000032 000000
 000034
 000034 000000
 000036
 000036 000000
 000040
 000040 007326'
 000042
 000042 007330'
 000044
 000044 013262'
 000046
 000046 007266'
 000050
 000050 000000
 000052
 000052 000210'

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

STADR: HEADER DZKCH, A, 0, 63, 300, 1
 LSNAME: :

. ASCII @D@
 . ASCII @Z@
 . ASCII @K@
 . ASCII @C@
 . ASCII @H@
 . BYTE 0
 . BYTE 0
 . BYTE 0
 LSREV: :
 . ASCII @AA@
 LSDEPO: :
 . ASCII @0@
 LSMREV: :
 . BYTE C\$REVISION
 . BYTE C\$EDIT
 LSUNIT: :
 . WORD 0
 LSTIM1: :
 . WORD 63
 LSTIMU: :
 . WORD 300
 LSTIML: :
 . WORD 1
 LSEF: :
 . WORD 0
 . WORD 0
 LSSPC: :
 . WORD 0
 LSEXP1: :
 . WORD 0
 LSEXP2: :
 . WORD 0
 LSEXP3: :
 . WORD 0
 LSDTP: :
 . WORD LSDISPATCH
 LSICP: :
 . WORD LSINIT
 LSCCP: :
 . WORD LSCLEAN
 LSHPCP: :
 . WORD LSHARD
 LSSPCP: :
 . WORD 0
 LSDEVP: :
 . WORD LSDVTYP

000054		LSREPP: :		
000054	000000		. WORD	0
000056		LSHPTP: :		
000056	007314'		. 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		LSPWRU: :		
000074	000000		. WORD	0
000076		LSLADP: :		
000076	015444'		. 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          000003      DATA   =3          ;BYTE POSITION IN OUTPUT BUFFER
52          000011      LUN     =11        ;DEFINE LOGICAL UNIT NUMBER FROM 11
53
54          ;OFFSETS IN DEVICE PARAMETER BLOCK (DCB)
55
56          000002      IO. FUN= 2
57          000004      IO. LUN= 4
58          000006      IO. EFN= 6
59          000010      IO. IST= 10
60          000012      IO. AST= 12
61          000014      IO. A1=  14
62          000016      IO. A2=  16
63          000020      IO. A3=  20
64          000022      IO. A4=  22
65          000024      IO. A5=  24
66
67          ; RSX I/O FUNCTION CODES
68
69          002400      IO. WPC= 2400     ;WRITE PERIPHERAL BLOCK
70          003400      IO. RXP= 3400     ;WRITE TRANSPARENT WITH LOOP BACK
```

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	115	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	

GETCNT: .ASCIZ/ENTER CONTROLLER NUMBER :/

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		

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
    000200 000001 .WORD 1
    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: QI0S 10. FUN, LUN, , IOST, 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 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
133 ;
134 ; *****
135 ; THE ASSISN LOGICAL UNIT PARAMETER BLOCK IS
136 ; DEFINED AS FOLLOWS.....
137 ; *****
138 ;
    
```



```

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 ;
144 ;
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   ERBRT
151 ;
152 ;
153 ;
154 ;
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 ;
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 ;
178 ;
179 001016      001041'          OBUF:  . 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
188 ;

```

 THE GET LUN PARAMETER BLOCK IS DEFINED AS
 FOLLOWS.....

 STORAGE LOCATIONS

RSX-11M DEVICE DEPENDENT PARAMETERS...

STORAGE FOR OUTPUT BUFFERS...

;END OF BUFFER TABLE

189			. ODD		
190	001041		OBUFO:	. BLKB	132.
191	001245		OBUF1:	. BLKB	132.
192	001451		OBUF2:	. BLKB	132.
193	001655		OBUF3:	. BLKB	132.
194	002061		OBUF4:	. BLKB	132.
195	002265		OBUF5:	. BLKB	132.
196	002471		OBUF6:	. BLKB	132.
197	002675		OBUF7:	. BLKB	132.
198			. EVEN		

; STORAGE FOR INPUT BUFFERS

201					
202	003102	003122'	INBUFF:	. WORD	IBUFO
203	003104	003322'		. WORD	IBUF1
204	003106	003522'		. WORD	IBUF2
205	003110	003722'		. WORD	IBUF3
206	003112	004122'		. WORD	IBUF4
207	003114	004322'		. WORD	IBUF5
208	003116	004522'		. WORD	IBUF6
209	003120	004722'		. WORD	IBUF7

210					
211	003122		IBUFO:	. BLKB	128.
212	003322		IBUF1:	. BLKB	128.
213	003522		IBUF2:	. BLKB	128.
214	003722		IBUF3:	. BLKB	128.
215	004122		IBUF4:	. BLKB	128.
216	004322		IBUF5:	. BLKB	128.
217	004522		IBUF6:	. BLKB	128.
218	004722		IBUF7:	. BLKB	128.

; STORAGE FOR I/O STATUS BLOCK

219					
220					
221					
222					
223					
224	005122	005142'	IOSB:	. WORD	ISTATO
225	005124	005146'		. WORD	ISTAT1
226	005126	005152'		. WORD	ISTAT2
227	005130	005156'		. WORD	ISTAT3
228	005132	005162'		. WORD	ISTAT4
229	005134	005166'		. WORD	ISTAT5
230	005136	005172'		. WORD	ISTAT6
231	005140	005176'		. WORD	ISTAT7

232					
233	005142		ISTATO:	. BLKW	2
234	005146		ISTAT1:	. BLKW	2
235	005152		ISTAT2:	. BLKW	2
236	005156		ISTAT3:	. BLKW	2
237	005162		ISTAT4:	. BLKW	2
238	005166		ISTAT5:	. BLKW	2
239	005172		ISTAT6:	. BLKW	2
240	005176		ISTAT7:	. BLKW	2

241					
242	005202	000000	ISB:	. WORD	0
243	005204	000000	IOSW:	. WORD	0
244	005206		BUF:	. BLKW	10
245					

; TEMP LOCATION USED IN AST
 ; TEMPORARY LOCATION FOR DSW ERROR
 ; TIME PARAMETER BUFFER

246	005226	377	PATRN:	. 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 PAIR
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	TMP:	. 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			

```
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298 007264  
    007264 000012  
    007266  
299 007266  
    007266 000032  
    007270 000100  
    007272 177400  
    007274 000000  
    007276 000007  
300 007300  
    007300 000052  
    007302 000132  
    007304 000377  
    007306 000001  
    007310 000077  
301  
302  
303 007312  
    007312  
304
```

.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.....  
*****
```

BGNHRD

.WORD L10000-LSHARD/2

LSHARD: :

GPRMD GETCNT, 0, 0, 177400, 0, 7, YES

.WORD TSCODE

.WORD GETCNT

.WORD 177400

.WORD TSLOLIM

.WORD TSHILIM

GPRMD GETERM, 0, 0, 377, 1, 63, YES

.WORD TSCODE

.WORD GETERM

.WORD 377

.WORD TSLOLIM

.WORD TSHILIM

ENDHRD

.EVEN

L10000:

306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321

.SBTTL DEFAULT HARDWARE P-TABLE

PARAMETERS HERE ARE USED AS DEFAULT IN
OPERATOR DIALOGUE.....

007312
007312 000001
007314
007314 000001
007316
007316

BGNHW
LSHW: :
ENDHW
L10001:

.WORD L10001-LSHW/2
.WORD 1 ; CONTROLLER/TERMINAL DEFAULT

```
323 . SBTTL SOFTWARE P-TABLE
324 ;
325 ;
326 ; *****
327 ; ANOTHER SEPERATE AND OPTIONAL PARAMETER
328 ; TABLE ASSEMBLED WITH THE DIAGNOSTIC
329 ; PROGRAM. ....
330 ; *****
331 007316 BGNSFT
    007316 000000 . WORD L10002-LSSOFT/2
    007320 LSSOFT: :
332 007320 ENDSFT
    007320 L10002: . EVEN
333
334 ; BUILD SOFTWARE P-TABLE
335 ;
336 ;
337 007320 BGNSW
    007320 000000 . WORD L10003-LSSW/2
    007322 LSSW: :
338
339 007322 ENDSW
    007322 L10003:
340
341 ;
342 007322 BGNRPT
    007322 LSRPT: :
343
344 007322 ENDRPT
    007322 L10004:
    007322 104025 EMT CSRPT
```


346
347
348
349
350
351
352
353
354
355
356
357
358

007324
007324 000001
007326
007326 011370'

.SBTTL DISPATCH TABLE
:
:
:
:
:
:
:
:
:
:
DISPATCH 1
LSDISPATCH: .WORD 1
.WORD T1

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.....

360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411

007330
007330
007330 012746
007332 033 001
007334 104377
007336 005037 005236'
007342 005037 005246'
007346 012701 000262'
007352 005021
007354 020127 001006'
007360 001374
007362 005037 000226'
007366 012737 000000G 000216'
007374 012705 000011
007400 010537 000220'
007404 012746 000214'

BGNINIT
LSINIT:

.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,
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...
++
--

CMKTS\$; CANCEL ANY OUTSTANDING MARK TIME REQUESTS
MOV (PC)+, -(SP)
.BYTE 27, 1
EMT 0<377>
CLR PASFG
CLR PASS

CLEAR STORAGE LOCATIONS FOR SUBSEQUENT INITIALIZATION CODE.

MOV #ERRBRT, R1
139\$: CLR (R1)+
CMP R1, #SCBADR
BNE 139\$

CLR Q10DPB+10. AST
MOV #10. DET, Q10DPB+10. FUN
MOV #LUN, R5
30\$: MOV R5, Q10DPB+10. LUN
DIRS #Q10DPB
MOV #Q10DPB, -(SP)

412	007410	104377		EMT	0<377>		
413	007412	005205		INC	R5		
414	007414	020527	000113	CMP	R5, #75.		
415	007420	003767		BLE	30%		
416	007422	012705	000306'	MOV	#UNTBL, R5		; 64 ONE-WORD UNIT TABLE
417	007426	013703	000014'	MOV	LSUNIT, 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			15: GPHARD	R4, R0		; GET POINTER TO HARDWARE P-TABLE
	007440	010400					
	007442	104055					
421	007444	011025		MOV	(R0), (R5)+	MOV R4, R0	
422				EMT		C\$GPHRD	
423	007446	005204		INC	R4		; GET TERM & CONTROLLER # FROM HARDWARE
424	007450	005303		DEC	R3		; P-TABLE, PUT IN UNIT TABLE
425	007452	001372		BNE	1%		; NEXT TABLE
426	007454	012715	177777	MOV	#-1, (R5)		; LAST P-TABLE?
427	007460	005000		CLR	R0		; BRANCH IF NO---
428	007462	005003		CLR	R3		; ELSE INSERT END OF TABLE FLAG
429	007464	012702	177777	MOV	#-1, R2		; OFFSETS TO TEMPORARY TABLES IN IBUFO
430	007470	005004		CLR	R4		; SB # INDICATOR
431	007472	012737	000011 000246' 25:	MOV	#LUN, ALUDPB+2		; CONTROLLER # INDICATOR FIRST TIME THRU
432	007500	010337	000252'	MOV	R3, ALUDPB+6		; TERMINAL # INDICATOR
433	007504			DIR\$	#ALUDPB		; SET THE LUN =11
	007504	012746	000244'	MOV	#ALUDPB, -(SP)		; SET THE DEVICES
	007510	104377		EMT	0<377>		; ISSUE I/O REQUEST
434	007512	103444		BCS	4%		
435	007514	012737	000011 000256'	MOV	#LUN, GLUDPB+2		; CHECK IF REQUEST IS REJECTED
436	007522	012737	000262' 000260'	MOV	#ERRBRT, GLUDPB+4		; ELSE SET THE LUN FOR THE G- DIRECTIVE
437	007530			DIR\$	#GLUDPB		; GET THE LUN INFO BUFFER ADDRESS
	007530	012746	000254'	MOV	#GLUDPB, -(SP)		; ISSUE I/O DIRECTIVE
	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, IBUFO+0(R0)		; STORE TERMINAL # IN TABLE
441	007550	110260	003123'	MOVB	R2, IBUFO+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		35: 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, IBUFO+0(R0)		; STORE TERMINAL #
450	007602	110260	003123'	MOVB	R2, IBUFO+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' 45:	MOV	#-1, IBUFO(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		55: CLR	R1		; POINTER TO IBUFO TABLE


```

462 007642 016402 000306'      MOV      UNTBL(R4),R2      ;GET CONTROLLER # TERM # FROM UNIT TABLE
463 007646 020261 003122'      65:      CMP      R2,IBUFO(R1)      ;CHECK IF SAME FROM MAPPED TABLE
464 007652 001406                BEQ      75                ;BRANCH IF SAME
465 007654 005761 003122'      TST      IBUFO(R1)        ;CHECK IF END OF TABLE
466 007660 100413                BMI      85                ;BRANCH IF END
467 007662 062701 000002      ADD      #2,R1            ;GET NEXT ITEM
468 007666 000767                BR       65                ;CONTINUE CHECKING
469 007670 116164 004122' 000506' 75:  MOVB    IBUF4(R1),DEVTAB+0(R4) ;STORE LOGICAL SB NUMBER
470 007676 062704 000002      ADD      #2,R4            ;GET NEXT UNTBL ENTRY
471 007702 005303                DEC      R3                ;LAST PHYSICAL TERMINAL TO BE STORED
472 007704 001355                BNE     55                ;BRANCH IF NO--
473 007706 000476                BR      115                ;ELSE GET NEXT DEVICE
474 007710 016437 000306' 007234' 85:  MOV      UNTBL(R4),TEMP    ;TEMPORARY STORAGE FOR PRINT ROUTINE
475 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$PNTF
      007746 062706 000010          ADD      #10,SP
476                                ;PRINTED THE NON-SYSGENED MESSAGE
477                                ;
478                                ;*****
479                                ;BUBBLE ROUTINE...
480                                ;++
481                                ;IF ONE , OR ALL TERMINALS ENTERED ARE UNASSIGNABLE
482                                ;OR UNATTACHABLE THIS ROUTINE ELIMINATES THEM OFF
483                                ;THE TABLE AND ABORTS THE PROGRAM...
484                                ;--
485                                ;*****
485 007752 005303                DEC      R3                ;ONE TOO MANY UNITS? UNASSIGNABLE?
486 007754 001433                BEQ     105                ;BRANCH IF ONE TOO MANY
487 007756                PUSH    <R3,R4>           ;SAVE
      007756 010346
      007760 010446
488 007762 016464 000310' 000306' 95:  MOV      UNTBL+2(R4),UNTBL(R4) ;BUBBLE UP FROM BELOW TO THIS LOCATION
489 007770 062704 000002      ADD      #2,R4            ;NEXT LOCATION
490 007774 005303                DEC      R3                ;DECREASE LOCATION COUNT
491 007776 001371                BNE     95                ;BRANCH IF NOT DONE
492 010000                POP     <R4,R3>           ;RESTORE
      010000 012604
      010002 012603
493 010004 005337 005250'      DEC      UNITS            ;ELIMINATE ONE UNIT OFF TABLE
494 010010 005737 005250'      TST     UNITS            ;ARE THERE ANY MORE DEVICES
495 010014 001311                BNE     55                ;YES--BRANCH
496 010016                PRINTF  #MABORT           ;NO-PRINT ABORT MESSAGE
      010016 012746 010767'          MOV      #MABORT,-(SP)
      010022 012746 000001          MOV      #1,-(SP)
      010026 010600                MOV      SP,RO
      010030 104017                EMT     C$PNTF
      010032 062706 000004          ADD      #4,SP
497 010036                EXITSS
      010036 012746                MOV      (PC)+,-(SP)
      010040 063                .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$$          ; AND EXIT
      010076 012746                MOV      (PC)+, -(SP)
      010100      063      001      . BYTE   51, 1
      010102 104377                EMT      0<377>
503 010104 005001                11$: CLR      R1      ; DEVICE TABLE POINTER
504 010106 012705 000011      MOV      #LUN, R5      ; SET THE LUN
505 010112 013703 005250'      MOV      UNITS, R3      ; KEEP COUNT
506 010116 012737 000000G 000216'  MOV      #10, ATT, Q10DPB+10. FUN ; SET FOR ATTACHES
507 010124 110537 000246'      12$: MOV     R5, ALUDPB+2      ; SET LUN FOR DIR$
508 010130 116102 000506'      MOV     DEVTAB+0(R1), R2 ; GET THE SB DEVICE
509 010134 110237 000252'      MOV     R2, ALUDPB+6      ; SET THE DEVICE NUMBER
510 010140                DIR$      #ALUDPB      ; ISSUE THE I/O REQUEST
      010140 012746 000244'      MOV     #ALUDPB, -(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$
515
516
517 010206 110561 000507'      13$: MOV     R5, DEVTAB+1(R1) ; DO NOT TRY TO ATTACH--
518 010212 010537 000220'      MOV     R5, Q10DPB+10. LUN ; BUT GO GET RID OF UNIT
519 010216                DIR$      #Q10DPB      ; FROM THE DEVICE TABLE
      010216 012746 000214'      MOV     #Q10DPB, -(SP) ; PUT THE LUN IN THE DEVICE TABLE
      010222 104377                EMT      0<377> ; ISSUE I/O REQUEST
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'
010270 012746 000003
010274 010600
010276 104017
010300 062706 000010
530
531
532 ; *****
533 ; BUBBLE ROUTINE.....
534 ; *****
535
536
537 010304 005303 175: DEC R3 ; ONE TOO MANY UNITS IN DEVICE TABLE
538 ; BECAUSE DEVICE IS UNASSIGNABLE
539 ; OR CANNOT BE ATTACHED
540 010306 001417 BEQ 195 ; BRANCH IF ONE TOO MANY
541 010310 PUSH <R3,R1>
010310 010346
010312 010146
542 010314 016161 000310' 000306' 185: 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 185 ; BRANCH IF NOT FINISHED
547 010340 POP <R1,R3>
010340 012601
010342 012603
548 010344 000667 BR 125 ; GO DO NEXT ASSIGN I. E. ALUN$
549 010346 005337 005250' 195: 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 155 ; 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,RO
010372 104017 EMT C$PNTF
010374 062706 000004 ADD #4,SP
554 010400 EXIT$$ ; AND EXIT
010400 012746 MOV (PC)+,-(SP)
010402 063 001 .BYTE 51,1
010404 104377 EMT 0<377>
555 010406 225:
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,RO
010430 104017 EMT C$PNTF
010432 062706 000006 ADD #6,SP
558 ; AND HEADER FOR ACTIVE DEVICE MAP
559 010436 005000 CLR RO ; POINTER TO TABLES
560 010440 116001 000507' 205: MOVB DEVTAB+1(RO),R1 ; GET LUN TO BE PRINTED
561 010444 116002 000506' MOVB DEVTAB+0(RO),R2 ; GET LOGICAL SB NUMBER
562 010450 116004 000307' MOVB UNTBL+1(RO),R4 ; GET CONTROLLER NUMBER

```



```

563 010454 116005 000306'      MOV B  UNTBL+0(R0),R5      ;GET PHYSICAL TERM#
564 010460 010046              PUSH RO                   ;SAVE RO**SUPERVISOR USES IT
565 010462 005046              PRINTF #MSG3,<B,R5>,<B,R4>,<B,R2>,<B,R1> ;
      010462 150116              CLR -(SP)
      010464 005046              BISB R1,(SP)
      010466 150216              CLR -(SP)
      010470 005046              BISB R2,(SP)
      010472 150416              CLR -(SP)
      010474 005046              BISB R4,(SP)
      010476 150516              CLR -(SP)
      010500 012746 011306'      BISB R5,(SP)
      010502 012746 000005      MOV #MSG3,-(SP)
      010506 010600              MOV #5,-(SP)
      010512 104017              MOV SP,RO
      010514 062706 000014      EMT C$PNTF
      010516 012600              ADD #14,SP
566 010522 012600              POP RO                   ;RESTORE RO
567 010524 062700 000002      ;PRINT ACTIVE DEVICE MAP
568 010530 005303              ADD #2,RO
569 010532 001342              DEC R3
570 010534 005005              BNE 20$
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              CLR B (R2)+          ;CLR 1'ST BYTE IN BUFFER
585 010560 005202              INC R2               ;GET NEXT BYTE
586 010562 112712 000014      MOV B #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'  MOV B #-1,ERRTBL     ;INIT ERROR TABLES
591 010602 005201              10$: INC R1
592 010604 105061 005252'      CLR B ERRTBL(R1)
593 010610 022701 001750      CMP #1000.,R1
594 010614 001372              BNE 10$
595 010616 105037 007222'      CLR B 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.,RO  ;CLEAR OFFLINE TABLE
599 010636 005040              14$: CLR -(RO)
600 010640 020027 000706'      CMP RO,#OFFLIN     ;END OF TABLE
601 010644 101374              BHI 14$
602 010646              MRKTSS ,#60.,#2,#MRKAST
      010646 012746 011744'      MOV #MRKAST,-(SP)
      010652 012746 000002      MOV #2,-(SP)

```

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

605
606
607
608
609
610
611
612
613

;
;
;
;
;

GLOBAL ASCII FOR INITIALIZATION SECTION
ALL MESSAGES FOR THE PROGRAM INIT ARE CONTAINED HERE.

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 - SDSW=%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-SDSW=%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	117	105
	011145	104	055	044
	011150	104	123	127
	011153	075	045	117
	011156	063	045	116
	011161	000		
618	011162	045	116	045
	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
	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
	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

MSG2: . ASCII /%N%ANUMBER OF ENTERED TERMINALS=%D2%N/

. ASCII2 /%N%AUNIT ** CONTROLLER ** LOGICAL SB ** LUN **/

MSG3: . ASCII2/%N%D3%A %D3%A %D3%A %D3%N/

011355	040	040	040
011360	040	040	045
011363	104	063	045
011366	116	000	

621

. EVEN

623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643 011370
 011370
 644
 645
 646
 647 011370 005037 007226'
 648 011374 004737 011760'
 649 011400 103460
 650
 651
 652
 653 011402 012737 000400 005240'
 654 011410 004737 012256'
 655 011414 006237 005240'
 656 011420 001765
 657
 658
 659
 660 011422 004737 012054'
 661 011426 103770
 662 011430 004737 012134'
 663
 664
 665
 666 011434 013703 005242'
 667 011440 012737 003400 000216'
 668 011446 016337 001016' 000236'
 669 011454 005237 000236'
 670 011460 013737 005240' 000240'
 671 011466 062737 000002 000240'
 672 011474 016337 003102' 000230'
 673 011502 013737 005240' 000232'
 674 011510 016337 005122' 000224'
 675 011516 012737 013362' 000226'
 676 011524 113737 005244' 000220'
 677 011532
 011532 012746 000214'

.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...
; *****

BGNTST
T1:
; EXECUTE A PASS
;
START: CLR PATPT ; CLEAR PATTERN POINTER
1$: JSR PC,C1 ; SET BYTE PATTERN IN OUTPUT BUFFER
BCS 65 ; END OF TEST

; EXECUTE A PARTIAL PASS
;
3$: MOV #256,LENGTH ; SET RECORD LENGTH
JSR PC,D3 ; CHECK ERRORS
ASR LENGTH ; VARY RECORD LENGTH
BEQ 1$ ; BRANCH IF END RECORD LENGTH

; PROCESS A LIST OF TERMINALS
;
4$: JSR PC,E1 ; SUBROUTINE (GET A TERMINAL FROM LIST)
BCS 3$ ; IF END OF TERMINAL LIST CHECK ERRORS
JSR PC,E2 ; CALL POLL AND SET OUTPUT BUFFER ROUTINE

; PROCESS A TERMINAL
;
MOV BFFPT,R3 ; GET BUFFER POINTER
MOV #10,RXP,Q10DPB+10.FUN ; READ TRANSPARENT TO THE DEVICE
MOV OBUFF(R3),Q10DPB+10.A4 ; LOAD OUTPUT BUFFER ADDRESS
INC Q10DPB+10.A4 ; SKIP THE FREE FLAG
MOV LENGTH,Q10DPB+10.A5 ; LOAD OUTPUT BUFFER SIZE
ADD #2,Q10DPB+10.A5 ; 2 BYTES LESS THAN OUT-BUFF
MOV INBUFF(R3),Q10DPB+10.A1 ; LOAD INPUT BUFFER ADDRESS
MOV LENGTH,Q10DPB+10.A2 ; LOAD INPUT BUFFER SIZE
MOV IOSB(R3),Q10DPB+10.IST ; LOAD STATUS WORD
MOV #AST,Q10DPB+10.AST ; LOAD TERM AST ADDRESS
MOVB LUNS,Q10DPB+10.LUN ; LOAD LOGICAL UNIT NUMBER
DIR$ #Q10DPB ; ISSUE I/O DIRECTIVE
MOV #Q10DPB,-(SP)
    
```



```

011536 104377      EMT      0<377>
678 011540 000730  BR      45          ;GET NEXT TERMINAL.
679          ; *****
680          ; KEEP PASS COUNT ONLY WHEN DEVICES ARE ACTIVE.
681          ; *****
682 011542          65:  PUSH     RO          ;SAVE
    011542 010046
683 011544 005000          CLR     RO          ;INIT POINTER TO OFFLINE TABLE
684 011546 023700 005250' 75:  CMP     UNITS,RO      ;CHECK IF END OF OFFLINE TABLE
685 011552 001407          BEQ     95          ;DO NEXT PASS IF END
686 011554 105760 000706'  TSTB   OFFLIN(RO)    ;CHECK FOR LIVE TERMINAL
687 011560 001402          BEQ     85          ;IF LIVE KEEP PASS COUNT
688 011562 105720          TSTB   (RO)+         ;IF OFFLINE UPDATE POINTER
689 011564 000770          BR      75          ;CONTINUE CHECKING
690 011566 005237 005246' 85:  INC     PASS        ;LOG PASS
691 011572          95:  BREAK          ;RETURN FOR A PEEK AT SUPERVISOR
    011572 104022
692 011574          POP     RO          ;RESTORE
    011574 012600
693 011576 005737 005236'  TST     PASFG
694 011602 001672          BEQ     START       ;TIME TO PRINT PASS COUNT?
695          ;NO-CONTINUE TESTING.
696 011604          MRKTSS  ,#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          135:  CMP     R1,R2       ;END OF OFFLINE TABLE
701 011642 001406          BEQ     115         ;BRANCH IF END
702 011644 105761 000706'  TSTB   OFFLIN(R1)   ;CHECK IF TERMINAL IS OFFLIN
703 011650 001001          BNE     125         ;IF OFFLINE BRANCH
704 011652 005203          INC     R3          ;LOG ACTIVE TERMINALS
705 011654 005201          125:  INC     R1          ;UPDATE OFFLINE TABLE POINTER
706 011656 000770          BR      135         ;CHECK NEXT TERMINAL
707 011660          115:  GTIMSS #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,-(SF)
    011710 012746 015311'  MOV     #EOP,-(SP)
    011714 012746 000005  MOV     #5,-(SP)
    011720 010600          MOV     SP,RO
    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
    
```

711 011740 000613
712 011742
011742
011742 104001

105: BR
ENDTST
L10006:

START

; DO NEXT PASS

EMT CSETST

714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735

.SBTTL DECLARE A SIGNIFICANT EVENT PER 5 MINS.

```
*****  
; THIS ROUTINE USES THE MRKTSS DIRECTIVE TO  
; PRINT THE NUMBER OF PROGRAM PASSES AND  
; THE NUMBER OF ACTIVE TERMINALS IN PRESENT PASS.  
; EVERY 5 MINUTES.  
*****
```

011744 005237 005236'
011750 005726
011752 012746
011754 163 001
011756 104377

```
MRKAST: INC PASFG ; SET THE 1 MIN. PASS FLAG.  
TST (SP)+ ; REMOVE EVENT FLAG PER RSX  
ASTXSS ; RETURN  
MOV (PC)+, -(SP)  
. BYTE 115, 1  
EML 0<377>
```

.SBTTL SET BYTE PATTERN IN OUTPUT BUFFER

737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 770
 771
 772

011760 013704 007226'
 011764 122764 000001 005226'
 011772 001004
 011774 005037 007226'
 012000 000261
 012002 000423
 012004 012700 001016'
 012010 005003
 012012 012001
 012014 012702 000200
 012020 116461 005226' 000003
 012026 005201
 012030 005302
 012032 001372
 012034 005203
 012036 022703 000010
 012042 001363
 012044 005237 007226'
 012050 000241
 012052 000207

C1:
 15:
 25:
 35:
 45:

```

MOV    PATPT,R4
CMPB   #1,PATRN(R4)
BNE    15
CLR    PATPT
SEC
BR     45
MOV    #0BUFF,R0
CLR    R3
MOV    (R0)+,R1
MOV    #128,R2
MOVB   PATRN(R4),DATA(R1)
INC    R1
DEC    R2
BNE    35
INC    R3
CMP    #8,R3
BNE    25
INC    PATPT
CLC
RTS    PC
    
```

```

*****
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.
*****
; GET PATTERN POINTER
; END OF PATTERN FLAG
; IF NO, GET BUFFERS
; RESET THE POINTER
; SET END OF PATTERN INDICATOR
; AND RETURN
; GET OUPUT BUFFER ADDRESS
; BUFFER NUMBER COUNTER
; GET BUFFERS 0 THRU 7
; SET MAXIMUM DATA BYTE LENGTH
; LOAD DATA PER BYTE IN BUFFER
; GET NEXT BYTE
; COUNT DATA BYTE LENGTH
; BRANCH UNTIL FINISHED
; UPDATE BUFFER NUMBER COUNT
; HAVE ALL 8 BUFFERS BEEN FILLED
; IF NO!! FILL NEXT BUFFER
; INCREMENT THE PATTERN POINTER
; RESET END FLAG FOR PATTERN TABLE
; RETURN
    
```



```

774 . SBTTL GET NEXT DEVICE AND LUN INFORMATION
775 ;
776 ;
777 ; *****
778 ;
779 ;++
780 ;ROUTINE TO GET THE NEXT PAIR OF DEVICE NUMBER AND ASSOCIATED LUN
781 ;RETURNS A -1 IN LUN, DEV AT END OF TABLE
782 ;
783 ; INPUT:      DEVPTR = POINTER TO NEXT DEVICE
784 ;            DEVTAB = TABLE OF ACTIVE DEVICES & LUNS
785 ;
786 ; OUTPUT:     TERM = TERMINAL NUMBER OF NEXT AVAILABLE UNIT
787 ;            LUNS = RSX ASSIGNED LUN NUMBER
788 ;
789 ;--
790 ;
791 ; *****
792 012054 013700 000302' E1: MOV DEVPTR,RO ;GET POINTER TO NEXT AVAILABLE UNIT
793 012060 006300 ASL RO ;CHANGE TO WORD OFFSET
794 012062 116037 000306' 007222' MOVB UNTBL+0(RO), TERM ;GET THE TERMINAL NUMBER
795 012070 116037 000507' 005244' MOVB DEVTAB+1(RO), LUNS ;GET LUN NUMBER
796 012076 116037 000506' 007256' MOVB DEVTAB+0(RO), SB ;SAVE SB (LOGICAL DEVICE
797 012104 005237 000302' INC DEVPTR ;POINT TO NEXT SET
798 012110 023737 000302' 000304' CMP DEVPTR, NUMDEV ;END OF LIST
799 012116 003404 BLE 15 ;NO---RETURN
800 012120 005037 000302' CLR DEVPTR ;POINT TO TOP OF LIST
801 012124 000261 SEC ;SET END OF LIST INDICATOR
802 012126 000401 BR 25 ;AND RETURN
803 012130 000241 15: CLC ;RESET END OF LIST INDICATOR
804 012132 000207 25: RTS PC ;RETURN
  
```

```

806 . SBTTL POLL AND SET OUTPUT BUFFERS
807 ;
808 ;
809 ; *****
810 ;++
811 ;ROUTINE TO DETERMINE THE NEXT OUTPUT BUFFER TO USE
812 ;
813 ; INPUT: TERM = TERMINAL NUMBER
814 ;
815 ; OUTPUT: BFFPT = OFFSET TO BUFFER TO USE
816 ;          OUTBUF BYTE 0 = SB - IN USE, IF 0-NOT IN USE
817 ;          OUTBUF BYTE 1 = TERMINAL NUMBER
818 ;
819 ; *****
820 ; NOTE: ROUTINE LOOPS FOREVER UNTIL A BUFFER BECOMES FREE
821 ; *****
822 ; --
823 ;
824 ;
825 ;
826 012134 E2: MRKTSS #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 J05 . BYTE 23, 5
    012156 104377 EMT 0<377>
827 012160 012700 001016' MOV #0BUFF, R0 ; START OF BUFFER TABLE
828 012164 005710 15: TST (R0) ; TEST FOR END OF TABLE
829 012166 001006 BNE 124$ ; BRANCH IF NOT
830 012170 WTSESS #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>
831 012202 000754 BR E2 ; START AT BEGINNING OF TABLE
832 012204 105770 000000 124$: TSTB @ (R0) ; SEE IF BUFFER IS IN USE
833 012210 001403 BEQ 2$ ; LOOP IF NON ZERO
834 012212 062700 000002 ADD #2, R0
835 012216 000762 BR 1$
836 012220 010037 005242' 2$: MOV R0, BFFPT ; SAVE IF BUFFER POINTER
837 012224 011000 MOV (R0), R0 ; GET ADDRESS OF TABLE
838 012226 113720 007256' MOV SB, (R0)+ ; SET SB
839 012232 113710 007222' MOVB TERM, (R0) ; INSERT DESIRED TERMINAL NUMBER
840 ; ; TERMINAL # MUST BE LEFT JUSTIFIED
841 ; ; IN THE BYTE WITH BIT0 SET
842 ; ; AND BIT#1 CLEARED
843 012236 106310 ASLB (R0)
844 012240 106310 ASLB (R0)
845 012242 152710 000001 BISB #BIT00, (R0) ; BITS IN BYTE HAS TO BE LEFT JUSTIFIED
846 012246 162737 001016' 005242' SUB #0BUFF, BFFPT ; COMPUTE AND SAVE OFFSET
847 012254 000207 RTS PC ; RETURN
    
```


.SBTTL CHECK ERRORS-CHECK FOR BAD CRC

849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874

```
;  
; *****  
; 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....  
; *****
```

875 012256
012256 010246
012260 010446
012262 010346
012264 010146
876 012266 005001
877 012270 005002
878 012272
012272 005046
012274 012746 000001
012300 012746 000005
012304 012746 000002
012310 012746
012312 027 005
012314 104377
879 012316 105771 001016'
880 012322 001406
881 012324
012324 012746 000002
012330 012746
012332 051 002
012334 104377
882 012336 000755
883 012340 005721
884 012342 022701 000020
885 012346 001351
886 012350 000137 012526'
887
888
889
890 012354 005037 007262'

```
D3: : PUSH <R2,R4,R3,R1> ;SAVE REGISTERS  
  
D3A: CLR R1  
CLR R2  
MRKTS #2, #5, #1 ;MARK TIME FOR SIG EVENT  
CLR -(SP)  
MOV #1, -(SP)  
MOV #5, -(SP)  
MOV #2, -(SP)  
MOV (PC)+, -(SP)  
. BYTE 23, 5  
EMT 0<377>  
TSTB @0BUFF(R1) ; IS BUFFER FREE?  
BEQ 123$ ;BRANCH IF FREE  
WTSESS #2 ;WAITFOR SIG EVENT  
MOV #2, -(SP)  
MOV (PC)+, -(SP)  
. BYTE 41, 2  
EMT 0<377>  
BR D3A ;CHECK FOR FREE BUFFER AGAIN  
123$: TST (R1)+ ;COUNT NUMBER OF BUFFERS FREE  
CMP #16, R1 ;8 BUFFERS FREE?  
BNE D3A ;BRANCH IF NO  
JMP 15$ ;TEMPO BYPASS...  
; *****  
; CHECK FOR BAD CRC.....  
; *****  
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$: ASL R2 ; READJUST POINTER
900 012406 116203 000306' MOVB UNTBL+0(R2),R3 ; GET LIVE TERMINAL
901 012412 110337 001042' MOVB R3, OBUFO+1 ; LOAD LIVE TERMINAL INTO BUFFER
902 012416 112737 000020 001043' MOVB #20, OBUFO+2 ; LOAD BAD CRC COMMAND
903 012424 012737 003400 000216' MOV #10, RXP, Q1ODPB+10. FUN ; LOAD WRITE TO DEVICE
904 012432 112737 001042' 000236' MOVB #OBUFO+1, Q1ODPB+10. A4 ; LOAD OUTPUT BUFFER ADDRESS
905 012440 013737 005240' 000240' MOV LENGTH, Q1ODPB+10. A5 ; LOAD OUTPUT BUFFER SIZE
906 012446 112737 003122' 000230' MOVB #IBUFO, Q1ODPB+10. A1 ; LOAD INPUT BUFFER ADDRESS
907 012454 013737 005240' 000232' MOV LENGTH, Q1ODPB+10. A2 ; LOAD INPUT BUFFER LENGTH
908 012462 012737 005142' 000224' MOV #ISTATO, Q1ODPB+10. IST ; LOAD STATUS WORD
909 012470 012737 013264' 000226' MOV #CRCAST, Q1ODPB+10. AST ; LOAD AST ADDRESS
910 012476 116237 000507' 000220' MOVB DEVTAB+1(R2), Q1ODPB+10. LUN ; LOAD LOGICAL UNIT NUMBER
911 012504 DIR$ #Q1ODPB ; ISSUE I/O DIRECTIVE
    012504 012746 000214' MOV #Q1ODPB, -(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, OBUFO+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
012644 062706 000004
943
944 ; AND DATA COMPARISON ERROR
945 ; *****
946 ; ERROR CALLS
947 ; *****
012650 ERRSOFT 4,MSG04,MSGCD4 ; DATA ERROR : LOG ERROR,
012650 104464 TRAP T$ERCODE
012652 000004 . WORD 4
012654 015213' . WORD MSG04
012656 014054' . WORD MSGCD4
948
949 012660 000723 BR 11$ ; PRINT ERROR AND
950 012662 15: PRINTF #LFCR ; BRANCH
012662 012746 015436' MOV #LFCR,-(SP) ; PRINT CARRIAGE RETURN LINE FEED
012666 012746 000001 MOV #1,-(SP)
012672 010600 MOV SP,RO
012674 104017 EMT C$PNTF
012676 062706 000004 ADD #4,SP
951 012702 ERRHRD 1,MSG01,MSGCD1 ; TIME-OUT ERROR : LOG ERROR,
012702 104463 TRAP T$ERCODE
012704 000001 . WORD 1
012706 015062' . WORD MSG01
012710 014370' . WORD MSGCD1
952 012712 000706 BR 11$ ; BRANCH
953 012714 25: PRINTF #LFCR ; PRINT CARRIAGE RETURN LINE FEED
012714 012746 015436' MOV #LFCR,-(SP)
012720 012746 000001 MOV #1,-(SP)
012724 010600 MOV SP,RO
012726 104017 EMT C$PNTF
012730 062706 000004 ADD #4,SP
954
955 012734 ERRSOFT 2,MSG02,MSGCD2 ; RETRY ERROR : LOG ERROR,
012734 104464 TRAP T$ERCODE
012736 000002 . WORD 2
012740 015127' . WORD MSG02
012742 014734' . WORD MSGCD2
956
957 012744 000671 BR 11$ ; PRINT ERROR AND
958 ; BRANCH
959 012746 35: PRINTF #LFCR ; PRINT CARRIAGE RETURN LINE FEED
012746 012746 015436' MOV #LFCR,-(SP)
012752 012746 000001 MOV #1,-(SP)
012756 010600 MOV SP,RO
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
962 012776 000654 BR 11$ ; PRINT ERROR AND
963 013000 75: PRINTF #LFCR ; BRANCH
013000 012746 015436' MOV #LFCR,-(SP) ; PRINT CARRIAGE RETURN LINE FEED
013004 012746 000001 MOV #1,-(SP)
013010 010600 MOV SP,RO
```

	013012	104017			EMT	C\$PNTF	
	013014	062706	000004		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, RO	
	013046	104017			EMT	C\$PNTF	
	013050	062706	000010		ADD	#10, SP	
965							;PRINT CRC MESSAGE
966	013054	062703	000006		ADD	#6, R3	;UPDATE ERROR TABLE POINTER
967	013060	000137	012530'		JMP	115	
968	013064			55:	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, RO	
	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	115	
971	013122			105:	PRINTF	#LFCR	;PRINT CARRIAGE RETURN.
	013122	012746	015436'		MOV	#LFCR, -(SP)	
	013126	012746	000001		MOV	#1, -(SP)	
	013132	010600			MOV	SP, RO	
	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, RO	
	013154	104017			EMT	C\$PNTF	
	013156	062706	000004		ADD	#4, SP	
973	013162	116300	005255'		MOVB	ERRTBL+3(R3), RO	
974	013166	005400			NEG	RO	
975	013170				PRINTF	#FMT01, <B, ERRTBL+0(R3)>, <B, ERRTBL+1(R3)>, <RO>, <B, ERRTBL+5(R3)>	
	013170	005046			CLR	-(SP)	
	013172	156316	005257'		BISB	ERRTBL+5(R3), (SP)	
	013176	010046			MOV	RO, -(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, RO	
	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	115	
978							
979	013244	005037	007224'	65:	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

983
984
985
986
987
988
989
990
991
992
993
994

995 013262
013262
996
997 013262
013262
013262 104012

.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.....

BGNCLN
L\$CLEAN: :

ENDCLN
L10007:

EMT C\$CLEAN

999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027

013264 005737 005142'
 013270 100025
 013272 013703 007224'
 013276 020327 001742
 013302 103020
 013304 112763 000006 005252'
 013312 116263 000306' 005253'
 013320 116263 000307' 005254'
 013326 062703 000006
 013332 112763 177777 005252'
 013340 010337 007224'
 013344 005726
 013346 012737 000001 007262'
 013354 012746
 013356 163 001
 013360 104377

CRC-AST:
 15:

```

TST      ISTAT0      ; TEMPORARY
BPL      15          ; IF NO RETRY THEN EXIT
MOV      ERRPNT,R3   ; GET POINTER TO ERROR TABLE
CMP      R3,#994     ; ERROR TABLE FILLED?
BHS      15          ; IF FILLED EXIT
MOVB     #6,ERRTBL+0(R3) ; LOAD ERROR TYPE
MOVB     UNTBL+0(R2),ERRTBL+1(R3) ; LOAD TERMINAL NUMBER
MOVB     UNTBL+1(R2),ERRTBL+2(R3) ; LOAD CONTROLLER NUMBER
ADD      #6,R3       ; UPDATE ERROR POINTER
MOVB     #-1,ERRTBL(R3) ; END OF ERRORTABLE FLAG
MOV      R3,ERRPNT   ; SAVE ERROR POINTER
TST      (SP)+       ; AS PER RSX--ASTXSS
MOV      #1,CRCFLG   ; SET AST INDICATOR
ASTXSS
MOV      (PC)+,-(SP) ; RETURN
.BYTE    115,1
EMT      0<377>
  
```

```

.SBTTL  CRC-AST ROUTINE

*****
ROUTINE CHECKS FOR DEVICE I/O ERRORS UPON A
REQUEST FOR BAD CRC GENERATION, LOGS THE ERRORS
AND SET AN INDICATION FOR AST
COMPLETION...
*****
  
```

.SBTTL AST SERVICE ROUTINE

1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078

013362
 013362 012637 005202'
 013366 011637 005204'
 013372
 013372 010046
 013374 010146
 013376 010246
 013400 010346
 013402 010446
 013404 010546
 013406 013705 005202'
 013412 162705 005142'
 013416 006205
 013420 010537 007260'
 013424 016505 001016'
 013430 111501
 013432 105015
 013434 110105
 013436 005004
 013440 120564 000506'
 013444 001402
 013446 005724
 013450 000773
 013452 006204
 013454 013701 005202'
 013460 013703 007224'
 013464 020327 001742
 013470 103153
 013472 005737 005204'
 013476 100423
 013500 105711
 013502 100431
 013504 105764 000706'
 013510 001411
 013512 105064 000706'
 013516 112763 000005 005252'
 013524 110563 005253'
 013530 062703 000006
 013534 023727 007224' 001742 45:
 013542 103126
 013544 000446

```

; *****
; 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.
; *****
;
; AST: POP ISB ; GET ADDRESS OF I/O STATUS BLOCK
;
; MOV (SP), IODSW ; GET DSW PRIOR TO AST ROUTINE
; PUSH <R0, R1, R2, R3, R4, R5> ; SAVE REGISTERS
;
; MOV ISB, R5 ; SAVE ADDRESS OF I/O STATUS BLOCK
; SUB #1STATO, R5 ; CALCULATE TERMINAL WHICH INTERRUPTED
; ASR R5 ; GET BUFFER CONTAINING INTERRUPTING TERMINAL
; MOV R5, TSTBUF ; SAVE BUFFER FOR DATA COMPARE
; MOV OBUF(R5), R5 ; STORE ADDRESS OF OUTPUT BUFFER
; MOVB (R5), R1 ; GET AND SAVE TERMINAL SB
; CLRB (R5) ; CLEAR THE 1'ST BYTE IN THE BUFFER
; MOVB R1, R5 ; STORE SB IN R5
; CLR R4 ; OFFLINE TABLE POINTER
; CMPB R5, DEVTAB(R4) ; FIND OFFSET FOR OFFLINE TABLE
; BEQ 215 ; BRANCH IF FOUND
; TST (R4)+ ; UPDATE POINTER
; BR 225 ; CONTINUE
; ASR R4 ; ADJUST OFFLINE TABLE POINTER
; MOV ISB, R1 ; GET I/O STAT
; MOV ERRPNT, R3 ; GET ERROR POINTER
; CMP R3, #994 ; CHECK ERROR TABLE
; BHIS 85 ; BRANCH IF ERROR POINTER = 994
; TST IODSW ; CHECK DSW ERRORS
; BMI 15 ; BRANCH IF ERROR
; TSTB (R1) ; CHECK IOSTAT ERRORS
; BMI 25 ; BRANCH IF ERROR
; TSTB OFFLIN(R4) ; WAS DEVICE OFFLINE?
; BEQ 45 ; BRANCH IF NO-----
; CLRB OFFLIN(R4) ; MAKE IT ON LINE
; MOVB #5, ERRTBL+0(R3) ; STORE ON-LINE CODE
; MOVB R5, ERRTBL+1(R3) ; STORE TERM#
; ADD #6, R3 ; ADJUST POINTER
; CMP ERRPNT, #994 ; CHECK ERROR TABLE
; BHIS 85 ; BRANCH IF FILLED
; BR 35 ; TEMPORARY BYPASS
  
```


1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092

;RETRY ROUTINE ERROR STUB INSERTED HERE*****

```

1094
1095
1096
1097 013546 112763 000003 005252' 15:      MOVB  #3,ERRTBL+0(R3)      ;MOV DSW ERROR-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    85                 ;BRANCH TO ASTXSS EXIT
1101
1102
1103
1104
1105
1106
1107
1108
1109 013566 121127 000277      25:   CMPB  (R1),#277        ;OFFLINE
1110 013572 001012      BNE   115
1111 013574 105764 000706'      TSTB  OFFLIN(R4)        ;ALREADY OFFLINE?
1112 013600 001107      BNE   85                 ;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    125
1116 013620 112763 000001 005252' 115:   MOVB  #1,ERRTBL+0(R3)    ;PUT HARD ERROR TYPE IN ERROR TABLE
1117 013626 110563 005253' 125:   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  45                  ;IF YES CHECK ERROR TABLE
1125 013660 000457      BR    85                 ;ASTXSS - EXIT
1126
  
```



```

1128                                     ; FROM RETRY ERROR
1129
1130                                     ; COMPARE I/O DATA BUFFER.....
1131
1132 013662 013701 007260'          35:  MOV    TSTBUF,R1          ; GET OUTPUT BUFFER POINTER
1133 013666 016102 001016'          55:  MOV    OBUF(R1),R2        ; STARTING ADDRESS OF OUTPUT BUFFER
1134 013672 016104 003102'          MOV    INBUF(R1),R4      ; STARTING ADDRESS INPUT BUFFER
1135 013676 062702 000003          ADD    #3,R2            ; GET OUTPUT DATA BYTE
1136 013702 012737 000012 007232'  MOV    #10.,ERRTMP     ; I/O DATA POINTER
1137 013710 005001          CLR    R1              ; CLEAR BYTE COUNTER.
1138 013712 020137 005240'          65:  CMP    R1,LENGTH      ; AT END OF RECORD?
1139 013716 001440          BEQ    85              ; YES-EXIT.
1140 013720 005201          INC    R1              ; NO-UPDATE BYTE COUNTER BY ONE.
1141 013722 122224          CMPB  (R2)+,(R4)+      ; I/O DATA OK?
1142 013724 001772          BEQ    65              ; GET NEXT DATA & BRANCH IF YES
1143 013726 005737 007232'          TST   ERRTMP          ; MORE THAN 10 I/O ERRORS STORED?
1144 013732 001432          BEQ    85              ; IF YES - EXIT
1145 013734 020327 001742          CMP    R3,#994        ; ERROR TABLE FILLED?
1146 013740 103027          BHIS  85              ; BRANCH IF YES - EXIT
1147 013742 112763 000004 005252'  MOVB  #4,ERRTBL+0(R3)  ; FILL ERROR-TYPE IN ERROR TABLE
1148 013750 110563 005253'          MOVB  R5,ERRTBL+1(R3) ; FILL TERMINAL# IN ERROR TABLE
1149 013754 013700 005202'          MOV   ISB,R0
1150 013760 116063 000002 005254'  MOVB  2(R0),ERRTBL+2(R3) ; FILL BYTE COUNT
1151 013766 110163 005255'          MOVB  R1,ERRTBL+3(R3) ; SET BYTE COUNTER
1152 013772 005337 007232'          DEC   ERRTMP          ; I/O DATA ERROR INDICATOR
1153 013776 124244          CMPB  -(R2),-(R4)     ; DECREMENT DATA POINTERS
1154 014000 111263 005256'          MOVB  (R2),ERRTBL+4(R3) ; FILL GOOD DATA
1155 014004 111463 005257'          MOVB  (R4),ERRTBL+5(R3) ; FILL BAD DATA
1156 014010 122224          CMPB  (R2)+,(R4)+     ; INCREMENT DATA POINTER
1157 014012 062703 000006          ADD   #6,R3
1158 014016 000735          BR    65
1159 014020 112763 177777 005252' 85: MOVB  #-1,ERRTBL(R3)   ; SET TABLE TERMINATOR
1160 014026 010337 007224'          MOV   R3,ERRPNT       ; STORE TABLE POINTER
1161 014032          POP   <R5,R4,R3,R2,R1,R0>
      014032 012605
      014034 012604
      014036 012603
      014040 012602
      014042 012601
      014044 012600
1162 014046          ASTXSS          ; EXIT
      014046 012746          MOV   (PC)+,-(SP)
      014050 163 001          .BYTE 115,1
      014052 104377          EMT   0<377>
  
```

1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187 014054
 014054
 1188 014054
 014054 005046
 014056 156316 005254'
 014062 005046
 014064 156316 005253'
 014070 005046
 014072 156316 005252'
 014076 012746 014206'
 014102 012746 000004
 014106 010600
 014110 104014
 014112 062706 000012
 1189
 1190 014116
 014116 005046
 014120 156316 005257'
 014124 005046
 014126 156316 005256'
 014132 005046
 014134 156316 005255'
 014140 012746 014336'
 014144 012746 000004
 014150 010600
 014152 104014
 014154 062706 000012
 1191 014160 062703 000006
 1192 014164 122763 000004 005252'
 1193 014172 001004
 1194 014174 126363 005253' 005245'
 1195 014202 001745
 1196 014204
 1197 014204

```

        .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 SAME TERMINAL AND SAME TYPE OF ERROR
; --
BGNMSG MSGCD4
MSGCD4:
        PRINTB #FORM1, <B, ERRTBL+0(R3)>, <B, ERRTBL+1(R3)>, <B, ERRTBL+2(R3)>
        CLR      -(SP)
        BISB    ERRTBL+2(R3), (SP)
        CLR      -(SP)
        BISB    ERRTBL+1(R3), (SP)
        CLR      -(SP)
        BISB    ERRTBL+0(R3), (SP)
        MOV     #FORM1, -(SP)
        MOV     #4, -(SP)
        MOV     SP, RO
        EMT     C$PNTB
        ADD     #12, SP
;
2$:     PRINTB #FORM3, <B, ERRTBL+3(R3)>, <B, ERRTBL+4(R3)>, <B, ERRTBL+5(R3)>
        CLR      -(SP)
        BISB    ERRTBL+5(R3), (SP)
        CLR      -(SP)
        BISB    ERRTBL+4(R3), (SP)
        CLR      -(SP)
        BISB    ERRTBL+3(R3), (SP)
        MOV     #FORM3, -(SP)
        MOV     #4, -(SP)
        MOV     SP, RO
        EMT     C$PNTB
        ADD     #12, SP
;
        ADD     #6, R3          ; POINT TO NEXT PACKET
        CMPB   #4, ERRTBL(R3) ; CHECK NEXT ERROR TYPE
        BNE   3$              ; BRANCH IF NOT
        CMPB   ERRTBL+1(R3), ERRTBL-5(R3) ; TERMINAL NUMBER
        BEQ   2$              ; SAME TERM, OUTPUT ERROR
;
3$:
ENDMSG
    
```


014204
014204 104023
1198

L10010:

;

EMT C\$MSG

1200
1201
1202
1203
1204
1205
1206

:/

GLOBAL ASCII MESSAGES USED BY THE PROGRAM.....

1207 014206 045 101 011
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

FORM1: . ASCII /%A ERRORTYPE=%02%A SB#=%D2%A RECEIVED BYTE COUNT=%D3/

1208 014272 045 116 045
014275 101 111 102
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
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

. ASCIIZ /%N%A BYTE-NUMBER GDDATA BDDATA/

FORM3: . ASCIIZ "%N%S14%D3%S9%03%S15%03%N"

1210

. EVEN

1212	014370				BGNMSG	MSGCD1			
	014370				MSGCD1:				
1213	014370	116300	005255'		MOVB	ERRTBL+3(R3),RO			
1214	014374	005400			NEG	RO			
1215	014376				PRINTB	#FMT01,<B,ERRTBL+0(R3)>,<B,ERRTBL+1(R3)>,<RO>,<B,ERRTBL+5(R3)>			
	014376	005046			CLR	-(SP)			
	014400	156316	005257'		BISB	ERRTBL+5(R3),(SP)			
	014404	010046			MOV	RO,-(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,RO			
	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									
1220	014446				ENDMSG				;UPDATE THE POINTER
	014446				L10011:				
	014446	104023							
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		. ASCII /%A RSX-11M STATUS CODE= -%D3%A BYTE COUNT=%D4%/			
	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:	. ASCII "%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	. ASCII "%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	
1226	014732	116	000		. 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' B1SB ERRTBL+2(R3), (SP)
      014742 005046 CLR -(SP)
      014744 156316 005253' B1SB ERRTBL+1(R3), (SP)
      014750 005046 CLR -(SP)
      014752 156316 005252' B1SB 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
```

```

1236                                     :GLOBAL TEXT
1237
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 SB. %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%ACTIVE 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	. ASCIIZ "%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: . ASCIIZ "%N%N"
	015441	116	000		
1247					:
1248					:
1249					:
1250					: *****
1251					: THE FOLLOWING CALL GENERATES THE FIRST EVEN ADDRESS NOT
1252					: USED BY THE PROGRAM.
1253					: *****
1254	015443				: LASTAD
	015444				: EVEN
1255					: L\$LAST: .
1256	000001				: . BLKW 2024. ; FREE MEMORY AREA
					: . END

ALUDP8	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\$QIC =	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	GLUDP8	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\$REVI=	000001	G\$NO =	000000	ISB	005202R	L\$RPT	007322RG
C\$ABRT=	000021	C\$RPT =	000025	G\$OFFS=	000400	ISTATO	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\$WFG=	000063	G\$RADA=	000140	ISTAT5	005166R	L\$SW	007322RG
C\$BUFF=	000034	C\$WTM =	000030	G\$RADB=	000000	ISTAT6	005172R	L\$TML	000022RG
C\$CEFG=	000061	C\$WTU =	000031	G\$RADD=	000040	ISTAT7	005176R	L\$TIMU	000020RG
C\$CLEA=	000012	C1	011760RG	G\$RADF=	000200	ISAU =	000041	L\$TIM1	000016RG
C\$CLP1=	000006	DATA =	000003	G\$RADL=	000120	ISCLN =	000041	L\$UNIT	000014RG
C\$CVEC=	000043	DEVPTR	000302R	G\$RADO=	000020	ISDU =	000041	L10000	007312R
C\$DCLN=	000057	DEVTAB	000506R	G\$RADT=	000100	ISHRD =	000041	L10001	007316R
C\$DODU=	000070	DSTW1	007250R	G\$XFER=	000004	ISINIT=	000041	L10002	007320R
C\$DRPT=	000024	DSTW2	007252R	G\$YES =	000010	ISMOD =	000041	L10003	007322R
C\$DSTI=	000033	DSTW3	007254R	G.LUBA=	000004	ISMSG =	000041	L10004	007322R
C\$DU =	000072	D3	012256RG	G.LUBL=	000006	ISPWR =	000041	L10005	010672R
C\$EDIT=	000004	D3A	012272R	G.LUCW=	000004	ISRPT =	000041	L10006	011742R
C\$ENTI=	000032	EOP	015311R	G.LUFB=	000003	ISSEG =	000041	L10007	013262R
C\$ERDF=	000002	ERRBRT	000262R	G.LULU=	000002	ISSFT =	000041	L10010	014204R
C\$ERHR=	000003	ERRPNT	007224R	G.LUNA=	000000	ISSRV =	000041	L10011	014446R
C\$ERSF=	000001	ERRTBL	005252R	G.LUNU=	000002	ISSUB =	000041	L10012	015002R
C\$ERSO=	000004	ERRTMP	007232R	G.TICP=	000016	ISTST =	000041	MAORT	010767R
C\$ESCA=	000010	ETB1	007236R	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	007242R	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	MSG0	010674R
C\$GMEM=	000050	FMT01	014450R	G.TIYR=	000000	L\$CLEA	013262RG	MSG01	015062R
C\$GPHR=	000055	FMT02	015004R	IBUFO	003122R	L\$DEPO	000011RG	MSG02	015127P
C\$GPRI=	000045	FORM1	014206R	IBUF1	003322R	L\$DEVP	000052RG	MSG03	015145R

MSG04	015213R	OSBGNR=	000000	RTRY1	007244R	TSTBUF	007260R	TSSHAR=	010000
MSG1	011020R	OSBGNS=	000000	RTRY2	007245R	TSARGC=	000004	TSSHW =	010001
MSG2	011162R	OSDU =	000000	RTRY3	007246R	TSCODE=	000052	TSSINI=	010005
MSG3	011306R	OSGNSW=	000000	SB	007256R	TSERCO=	000064	TSSMSG=	010012
M200	014563R	OSPOIN=	000001	SCBADR	001006R	TSERRN=	000003	TSSRPT=	010004
NR	= 000001	OSPWR =	000000	SCBCNT	001010R	TSEXCP=	000000	TSSSOFF=	010002
NUMDEV	000304R	PASFG	005236R	STADR	000000RG	TSHILI=	000077	TSSSW =	010003
OBUFF	001016R	PASS	005246R	START	011370R	TSLOLI=	000001	TSSTES=	010006
OBUFO	001041R	PATPT	007226R	SVC CNT=	177777	TSLSYM=	010000	T1	011370RG
OBUF1	001245R	PATRN	005226R	SVCGBL=	000001	TSMCAL=	000000	UNITS	005250R
OBUF2	001451R	PCBADR	001012R	SVCINS=	000001	TSNEST=	177777	UNTBL	000306R
OBUF3	001655R	PCBCNT	001014R	SVCSTK=	177777	TSNSKO=	000011	X\$ALWA=	000000
OBUF4	002061R	Q10DPB	000214R	SVC SUB=	000001	TSSAVL=	177777	X\$FALS=	000040
OBUF5	002265R	Q.10AE=	000012	SVCTAG=	000001	TSSSEGL=	177777	X\$OFFS=	000400
OBUF6	002471R	Q.10EF=	000006	SVCTST=	000001	TSSUBN=	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	TMP	007235R	T\$SCLE=	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