

DPV-11

DPV-11 FUNC. DIAG  
CVDPVBO

AH-S035B-MC  
FICHE 1 OF 1

FEB 1981  
COPYRIGHT © 1980  
MADE IN USA





CVDPVBO DPV11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 1  
PROGRAM DOCUMENT

.NLIST TOC  
.REM @

IDENTIFICATION  
-----

PRODUCT CODE: AC-S033B-MC  
PRODUCT NAME: CVDPVBO DPV11 FUNC DIAG  
PRODUCT DATE: OCTOBER 1980  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: MIKE O'CONNOR

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.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF  
SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS  
AFFILIATED COMPANIES.

COPYRIGHT (C) 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGI AL  
DEC

PDP  
DECUS

UNIBUS  
DECTAPE

MASSBUS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

REVISION HISTORY:

REV ---	DATE ----	AUTHOR -----	REASON -----
A	JUNE 80	MIKE OCONNOR	ORIGINAL RELEASE
B	OCT. 80	MIKE OCONNOR	1. CHANGE CHARACTER LENGTH IN TEST 37
			2. CHANGE TIMEOUT IN TEST 29
			3. ENHANCEMENTS

CONTENTS

1	
2	
3	
4	
5	
6	1.0 INTRODUCTION
7	
8	2.0 HARDWARE REQUIREMENTS
9	
10	3.0 PRELIMINARY PROGRAM REQUIREMENTS
11	
12	4.0 GENERAL PROGRAM CONSIDERATIONS
13	4.1 DIAGNOSTIC SUPERVISOR
14	4.2 EXECUTION TIME
15	4.3 XXDP+
16	4.4 ACT/SLIDE
17	4.5 APT
18	4.6 MEMORY MANAGEMENT
19	4.7 MEMORY PARITY OPTION
20	4.8 ERROR LOGGING
21	
22	5.0 PROGRAM LOAD MEDIA
23	
24	6.0 OPERATING INSTRUCTIONS
25	6.1 LOADING AND STARTING PROCEDURES
26	6.1.1 LOADING PROCEDURES
27	6.1.2 STARTING PROCEDURES
28	6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION
29	6.2 INITIAL DIALOGUE
30	6.3 PROGRAM OPTIONS
31	6.3.1 START COMMAND
32	6.3.1.1 TESTS SWITCH
33	6.3.1.2 PASS SWITCH
34	6.3.1.3 FLAGS SWITCH
35	6.3.1.4 END OF PASS SWITCH
36	6.3.1.5 EFFECT OF START COMMAND
37	6.3.2 RESTART COMMAND
38	6.3.2.1 TESTS, PASS, AND FLAG SWITCHES
39	6.3.2.2 UNITS SWITCH
40	6.3.2.3 EFFECT OF RESTART COMMAND
41	6.3.3 CONTINUE COMMAND
42	6.3.3.1 PASS SWITCH
43	6.3.3.2 FLAGS SWITCH
44	6.3.3.3 EFFECT OF CONTINUE COMMAND
45	6.3.4 PROCEED COMMAND
46	6.3.4.1 FLAGS SWITCH
47	6.3.4.2 EFFECT OF PROCEED COMMAND
48	6.3.5 ADD COMMAND
49	6.3.5.1 UNITS SWITCH
50	6.3.5.2 EFFECT OF ADD COMMAND
51	6.3.6 DROP COMMAND
52	6.3.6.1 UNITS SWITCH
53	6.3.6.2 EFFECT OF DROP COMMAND
54	6.3.7 PRINT COMMAND
55	6.3.7.1 EFFECT OF PRINT COMMAND
56	6.3.8 DISPLAY COMMAND
57	6.3.8.1 UNITS SWITCH

58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74

6.3.8.2 EFFECT OF DISPLAY COMMAND  
6.3.9 FLAGS COMMAND  
6.3.9.1 EFFECT OF FLAGS COMMAND  
6.3.10 ZFLAGS COMMAND  
6.3.10.1 EFFECT OF ZFLAGS COMMAND  
6.3.11 CONTROL CHARACTERS  
6.3.12 HARDWARE PARAMETERS  
6.3.13 SOFTWARE PARAMETERS  
6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

## 7.0 DEVICE INFORMATION TABLES

## 8.0 TEST DESCRIPTIONS

### 8.1 DATA PATTERNS USED

## 9.0 ERROR INFORMATION

### 9.1 ERROR REPORTING

## 1.0 INTRODUCTION

THIS PROGRAM WILL BE IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN WILL CONFORM TO THE SUPERVISOR (STANDALONE VERSION) THE PROGRAM WILL BE COMPATIBLE WITH ACT, APT, XXDP+, AND SLIDE.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS LSI-BUS ADDRESS, VECTOR ADDRESSES AND DEVICE PRIORITY. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

DEVICE ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION OF THE ERROR, GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

## 2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DPV11 FUNCTIONAL DIAGNOSTIC TESTS:

A LSI11 OR PDP11/03  
16K MEMORY  
CONSOLE TERMINAL  
DPV11

## 3.0 PRELIMINARY PROGRAM REQUIREMENTS

IT IS ASSUMED THAT THE PROCESSOR IS IN PROPER WORKING CONDITION.

THE DEVICE ADDRESS AND THE INTERRUPT VECTOR MUST BE KNOWN BEFORE ANSWERING THE USER DIALOGUE. THE USER SHOULD ALSO KNOW WHETHER THE CPU IS A LSI11 (M7264), A LSI11/2 (M7270), OR A LSI11/23 (M8186). FINALLY THE USER MUST DECIDE THE TYPE OF TURNAROUND IN ORDER TO DETERMINE THE CONNECTOR (IF ANY) IS NECESSARY.

## 4.0 GENERAL PROGRAM CONSIDERATIONS

### 4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

EXECUTION TIME IS DEPENDENT ON THE PROCESSOR SPEED AND THE TYPE OF LOOPBACK  
THE FOLLOWING ARE THE TIMES TO COMPLETE THE 1ST PASS:

	RS423 (OR INTERNAL)	RS422
LSI11 (KD11-F M7264 MODULE):	10 SECONDS	30 SEC.
LSI11/2 (KD11-HA M7270 MODULE):	10 SECONDS	30 SEC.
LSI11/23(KDF11-AA M8186 MODULE):	7 SECONDS	5 SEC.

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN  
DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN  
IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING  
APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

THERE IS NO MEMORY MANAGEMENT USE IN THIS DIAGNOSTIC.

4.7 MEMORY PARITY OPTION

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE  
DISABLED BY THE PROGRAM.

4.8 ERROR LOGGING

AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT  
THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR  
RESTART COMMAND.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE  
ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM  
ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE  
ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST,  
FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE  
DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY  
THE DIAGNOSTIC PROGRAM.

## 6.0 OPERATING INSTRUCTIONS

### 6.1 LOADING AND STARTING PROCEDURES

#### 6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

#### 6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

#### 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+, WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

- A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
- B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION AND PROMPT (DRS-C>)
- C) ENTER STA<CR>
- D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
- E) GET END OF PASS MESSAGES OR ERROR MESSAGES
- F) TO END EXECUTION, ENTER CONTROL/C

### 6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED :

DRS LOADED  
DIAG. RUN-TIME SERVICES

DR>

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

### 6.3 PROGRAM OPTIONS

#### 6.3.1 START COMMAND

\*\*\*\*\*  
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
<FLAG-LIST>/EOP:<INCR>  
\*\*\*\*\*



6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
LOT	LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 6.3.1.5.

#### 6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 6.3.1.5.

#### 6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION '# UNITS?' TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM 'UNIT' REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION '# UNITS?' IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE 'TOO MANY UNITS' IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

#### EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

#### 6.3.2 RESTART COMMAND

\*\*\*\*\*  
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:

286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342

<FLAG-LIST>/UNITS:<UNIT-LIST>  
\*\*\*\*\*

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIALOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

\*\*\*\*\*  
CON(TINUE)/PASS:<PASS-CNT>/FLAGS:<FLAG-LIST>  
\*\*\*\*\*

6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

\*\*\*\*\*  
PRO(CEED)/FLAGS:<FLAG-LIST>  
\*\*\*\*\*

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

6.3.5 ADD COMMAND

\*\*\*\*\*  
ADD/UNITS:<UNIT-LIST>  
\*\*\*\*\*

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.



400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456

#### 6.3.6 DROP COMMAND

\*\*\*\*\*  
DRO(P)/UNITS:<UNIT-LIST>  
\*\*\*\*\*

##### 6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

##### 6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

#### 6.3.7 PRINT COMMAND

\*\*\*\*\*  
PRI(NT)  
\*\*\*\*\*

##### 6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

#### 6.3.8 DISPLAY COMMAND

\*\*\*\*\*  
DIS(PLAY)/UNITS:<UNIT-LIST>  
\*\*\*\*\*

##### 6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

##### 6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO DESIGNATED.

#### 6.3.9 FLAGS COMMAND

\*\*\*\*\*

FLA(GS)

\*\*\*\*\*

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

\*\*\*\*\*

ZFL(AGS)

\*\*\*\*\*

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES  
A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE OPERATOR  
DIALOGUES- HARD CORE QUESTIONS (SEE 6.2), HARDWARE DIALOGUE  
(SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE  
DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC  
CAUSES ALL TELETYPE OUTPUT TO BE SUPPRESSED FOR THE  
REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED,  
WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 4 QUESTIONS WILL BE ASKED ON A START COMMAND.  
THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE  
DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN  
RESPONSE.

1. ADDRESS : (O) 160010?

THIS IS THE ADDRESS AT WHICH THE DPV CSR REGISTERS RESIDE  
ON THE LSI-BUS. THE ALLOWABLE RANGE IS 160000-177776  
(OCTAL), AND THE DEFAULT VALUE IS 160010.

2. VECTOR : (O) 300 ?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS  
DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE  
DEFAULT VALUE IS 300.

3. LOOPBACK -

0 = INTERNAL, 1 = RS423, 2 = RS422  
3 = LOCAL MODEM LOOP, 4 = REMOTE MODEM LOOP (0) 1?

THIS IS THE USER SELECTED LOOPBACK. THE DEFAULT IS RS423.  
THE FOLLOWING SHOULD BE CONSIDERED:

- A. INTERNAL LOOPBACK RUNS THE DIAGNOSTIC THROUGH THE USYNR1 MAINTENANCE MODE LOOPBACK. THE DRIVERS WILL NOT BE TESTED. NO CONNECTOR IS REQUIRED.
- B. RS423 REQUIRES A H3260 ONBOARD CONNECTOR OR THE BC05C CABLE AND THE H3259 CONNECTOR. THIS TURNAROUND WILL PROVIDE A 2K CLOCK FOR DIAGNOSTICS. ALL TESTS SHOULD BE ABLE TO BE RUN ON ALL PROCESSORS.
- C. R2422 REQUIRES A MODIFIED H3260 ONBOARD CONNECTOR. THIS TURNAROUND WILL PROVIDE A 50K CLOCK FOR DIAGNOSTICS. THE TESTS RUN WILL DEPEND ON THE PROCESSOR.
  - 1. THE LSI11/23 SHOULD RUN ALL TESTS.
  - 2. THE LSI11/2 SHOULD RUN ALL TESTS EXCEPT TESTS 29-41.
  - 3. THE LSI11 WITHOUT PROCESSOR MEMORY REFRESH SHOULD RUN ALL TESTS EXCEPT TESTS 29-41.
  - 4. THE LSI11 WITH PROCESSOR MEMORY REFRESH SHOULD RUN ALL TESTS EXCEPT TESTS 29-43.
- D. LOOPBACK THROUGH THE MODEM SHOULD ONLY BE ATTEMPTED IF THE MODEM SUPPORTS THAT TYPE OF LOOPBACK.

#### 6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY TIS DIAGNOSTIC

#### 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "'# UNITS?'" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE

571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627

QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

# UNITS (D) ? 16  
UNIT 0  
<QUESTION 1> ? 75  
<QUESTION 2> ? 0-6  
<QUESTION 3> ? 76

UNIT 7  
<QUESTION 1> ?  
<QUESTION 2> ? 7-11,,13-15  
<QUESTION 3> ? 77

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM 'UNIT XX' AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).



7.0 DEVICE INFORMATION TABLES

SEE THE GLOBAL EQUATES SECTION FOR DEFINITIONS OF REGISTERS IN THE DPV  
AND BIT DEFINITIONS WITHIN THOSE REGISTERS.

8.0 TEST DESCRIPTIONS

```
*****
*                               TEST 1 - DPV-11
* VERIFY THAT ADDRESSING THE 4 LSI-BUS CSRS DOES NOT CAUSE A NON-
* EXISTENT MEMORY TRAP.
*
* THE DPV IS AN COMMUNICATION DEVICE RESIDING ON A LSI-BUS.
* COMMUNICATION BETWEEN THE MAIN CPU AND THE DPV IS ACCOMPLISHED
* THROUGH A SET OF FOUR 16-BIT LSI-BUS CONTROL AND STATUS REGISTERS
* (CSRS). THE FOUR REGISTERS ARE ASSIGNED ADDRESSES IN THE I/O PAGE
* FLOATING ADDRESS SPACE: 76XXX0 - 76XXX6
*
* AN ERROR IN THIS TEST COULD MEAN THAT THE DEVICE IS INCORRECTLY
* CONFIGURED, THAT THE ADDRESS IS WRONG OR THAT THE CRYSTAL CLOCK
* ON THE DPV IS NOT WORKING. THE SHIFT REGISTER CLOCK IS NEEDED
* FOR THE LS164 (E15) IN ORDER TO PROVIDE THE BUS REPLY (BRPLY/L ON
* PIN AF2).
*****
```

```
*****
*                               TEST 2 - DPV-11
* DPV RESET
* RESET THE DPV AND ENSURE THAT ALL REGISTERS ARE IN THEIR
* PROPER INITIALIZATION STATE. THE RESET IS ASYNCHRONOUS TO ALL
* DATA SET TIMING AND ANY DATA PORT ACCESSSES. THE FOLLOWING
* WILL BE CHECKED BY THE $RESET SUBROUTINE:
*   1. ALL BITS IN THE DATA PORT REGISTERS ARE CLEARED.
*   2. ALL OUTPUT INDICATORS ARE CLEARED.
*   3. TRANSMIT BUFFER EMPTY (TBE) IS SET
*
* SUBTEST 1 - AFTER RESET, CHECK THAT MAINTENANCE MODE AND
* TRANSMITTER CAN BE SET. ALSO CHECK THAT TRANSMITTER
* BUFFER EMPTY (TBE) IS CLEARED WHEN TDSR IS ACCESSED
* WITHOUT SETTING TRANSMITTER ENABLE.
* SUBTEST 2 - ON THE FIRST PASS ONLY, CHECK THAT A BUS RESET, DOES
* A DPV11 RESET.
*
* NOTE: DATA MODE, CTS, RR (RECEIVER READY) AND IC (INCOMING CALL)
* ARE UNAFFECTED BY A RESET.
*****
```

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

```
*****
*                               TEST 3 - DPV-11
* WRITE/READ DATA PATTERNS
* THIS TEST IS INTENDED TO TEST THE READ/WRITE BITS IN THE CSRS. THERE
* IS NO INTENTION TO CHECK THE USYNR/T; IT IS DESIRED TO ONLY CHECK THE
* READING AND WRITING OF THE CSRS. IN ALL THE SUBTESTS THE BITS ARE
* CHECKED TOGETHER AND INDIVIDUALLY.
* SUBTEST 1 - RXCSR (LOW BYTE CSR0)
* CHECK BITS 0-6
* SUBTEST 2 - PCR (HIGH BYTE CSR4)
* CHECK BITS 0-7
* SUBTEST 3 - TDSR (LOW BYTE OF CSR6) - TRANSMIT DATA BUFFER
* BITS 0-7
* SUBTEST 4 - TDSR (HIGH BYTE OF CSR6) - TRANSMIT STATUS REGISTER.
* BITS 0-3
* SUBTEST 5 - TDSR - CHECK BYTE OP SIGNAL FOR USYNRT
*
*****

*****
*                               TEST 4 - DMR-11
* TRANSMIT ENABLE/ TRANSMIT ACTIVE
* AFTER A DEVICE RESET, SET TRANSMIT START OF MESSAGE (TSOM). ENSURE
* THAT TRANSMIT ACTIVE (TXACT) IS SET.
*
* TXACT IS USED TO INDICATE THE CURRENT STATE OF THE TRANSMITTER
* DATA PATH. THIS BIT WILL BE ASSERTED WHEN BOTH THE TRANSMITTER IS
* ENABLED AND TSOM ARE INTERNALLY SYNCHRONIZED. TXACT WILL BE CLEARED
* UPON RESET OR WHEN THE TRANSMITTER ENTERS THE IDLE STATE.
*
*****

*****
*                               TEST 5 - DPV-11
* TRANSMIT BUFFER EMPTY
* VERIFY THAT TBE (TRANSMIT BUFFER EMPTY) IS ASSERTED WHENEVER
* THE DEVICE IS RESET OR WHENEVER THE TDSR IS AVAILABLE FOR DATA.
* TBE IS CLEARED AFTER WRITING TO THE TDSR.
*
*****
```

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

```
*****
*                               TEST 6 - DPV-11
* TRANSMIT INTERRUPT
* VERIFY THAT A TRANSMIT INTERRUPT IS RECEIVED WHEN TRANSMIT
* BUFFER EMPTY (TBE) IS ASSERTED.
*
*****

*****
*                               TEST 7 - DPV-11
* RECEIVER ENABLE, RECEIVER ACTIVE AND RECEIVER DATA READY
* MODE: BCP, 8 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK
* ENABLE THE RECEIVER. AFTER TRANSMITTING A CHARACTER WAIT FOR
* RECEIVER DATA AVAILABLE AND CHECK THAT THE RECEIVER IS ACTIVE.
* AFTER CLEARING RECEIVER ENABLE, ENSURE THAT THE RECEIVER IS INACTIVE.
*
* RECEIVER ENABLE - CONTROLS THE OPERATION OF THE RECEIVER DATA PATH (RDP)
* RECEIVER ACTIVE - THIS OUTPUT IS ASSERTED WHEN THE RDP PRESENTS THE 1ST
*                   DATA CHARACTER OF A MESSAGE TO THE USYRT. IT REMAINS
*                   ASSERTED UNTIL THE RDP ENTERS THE IDLE STATE..
* RECEIVE DATA    - THIS OUTPUT IS SET WHEN THE RDP HAS ASSEMBLED A DATA
*                   CHARACTER THAT IS READY TO BE PRESENTED.
*
*****

*****
*                               TEST 8 - DPV-11
* RECEIVE DATA INTERRUPT
* MODE: BCP, 8 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK
* ENABLE THE RECEIVER AND SET RECEIVER INTERRUPT. TRANSMIT DATA.
* CHECK THAT THE RECEIVE INTERRUPT WAS GENERATED. AFTER THE INTERRUPT
* WAS GENERATED DISABLE THE RECEIVER. CHECK THAT THE RECEIVER BECOMES
* INACTIVE.
*
*****
```

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

```
*****
*                                     *****
*                                     TEST 9 - DPV-11
* THERE ARE 3 SUBTESTS IN THIS TEST WHICH ARE INTENDED TO CHECK
* RECEIVER STATUS.
* SUBTEST 1 - REOM (RECEIVE END OF MESSAGE)
*               THIS SUBTEST WILL TRANSMIT A DATA MESSAGE THAT IS
*               ENDED WITH A TEOM (TRANSMIT END OF MESSAGE). A
*               CHECK WILL BE MADE THAT THE RECEIVER GETS THE DATA
*               AND THAT THE REOM IS RECEIVED WHEN RECEIVE
*               STATUS IS AVAILABLE.
*
* SUBTEST 2 - RECEIVER OVERRUN
*               THIS SUBTEST WILL TRANSMIT DATA CORRECTLY. THE
*               RECEIVER AFTER BECOMING ACTIVE WILL NOT SERVICE
*               THE RECEIVE BUFFER CORRECTLY. THIS SHOULD RESULT IN
*               A RECEIVE OVERRUN. THIS SUBTEST WILL ENSURE THAT
*               WHEN RECEIVE STATUS IS AVAILABLE, THE RECEIVER OVERRUN
*               IS SET.
*
* SUBTEST 3 - RECEIVER ABORT
*               THIS SUBTEST WILL TRANSMIT A DATA MESSAGE THAT IS ENDED
*               WITH A TRANSMIT ABORT. THE SUBTEST WILL ENSURE THAT
*               RECEIVE STATUS AVAILABLE IS RECEIVED AND THAT THE
*               ABORT IS RECEIVED.
*
*                                     *****
```

```
*****
*                                     *****
*                                     TEST 10 - DPV-11
* THIS TEST WILL ENSURE THAT INTERRUPTS MAY BE GENERATED WHEN
* RECEIVE STATUS IS AVAILABLE. EACH OF THE FOLLOWING SUBTESTS
* WILL GENERATE THE STATUS AS FOLLOWS:
* SUBTEST 1 - REOM
* SUBTEST 2 - RECEIVER OVERRUN
* SUBTEST 3 - RECEIVER ABORT
*
*                                     *****
```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50

```
*****
*                               TEST 11 - DPV-11
* RECEIVE AND TRANSMIT INTERRUPT
* TRANSMIT AND RECEIVE DATA USING INTERRUPT ROUTINES. THIS TEST
* WILL TRANSMIT 4 DATA CHARACTERS. AFTER ENSURING THAT A TRANSMIT
* INTERRUPT WAS COMPLETED, THE TEST WILL CHECK TO MAKE SURE THAT AT
* LEAST 1 RECEIVE INTERRUPT WAS GENERATED.
*
*****

*****
*                               TEST 12 - DPV-11
* MODEM STATUS
* IF A PROPER TURNAROUND (H3259 OR H3260) IS ON, THIS TEST WILL
* CHECK THAT THE FOLLOWING MODEM SIGNALS ARE TURNED AROUND
* 1. RTS (REQUEST TO SEND)      TURNED AROUND TO CTS (CLEAR TO SEND)
*                                & RR (RECEIVER READY)
* 2. DTR (DATA TERMINAL READY) TURNED AROUND TO IC (INCOMING CALL OR RING)
* 3. SF (SELECT FREQUENCY)     TURNED AROUND TO SQ (SIGNAL QUALITY)
* 4. LL (LOCAL LOOPBACK)       TURNED AROUND TO DM (DATA MODE)
*
*****

*****
*                               TEST 13 - DPV-11
* MODEM STATUS INTERRUPT
* IF A PROPER TURNAROUND (H3259 OR H3260) IS ON, THIS TEST WILL CHECK
* THAT THE FOLLOWING SUBTESTS WORK CORRECTLY.
* SUBTEST 1 - SET DTR (DATA TERMINAL READY), LOCAL LOOP (LL), RTS (REQUEST
* TO SEND) WITH ONLY RECEIVE INTERRUPT ENABLED. ENSURE THAT AN
* INTERRUPT IS NOT RECEIVED.
* SUBTEST 2 - SET DTR, LL AND RTS WITH ONLY DATA SET INTERRUPT ENABLED.
* ENSURE THAT AN INTERRUPT IS NOT RECEIVED.
* SUBTEST 3 - SET DTR, LL AND RTS WITHOUT ANY INTERRUPTS ENABLED. ENSURE
* THAT AN INTERRUPT IS NOT RECEIVED.
* SUBTEST 4 - SET RTS WITH RECEIVE AND DATA SET INTERRUPT ENABLED. ENSURE
* THAT AN INTERRUPT IS RECEIVED.
* SUBTEST 5 - SET DTR WITH RECEIVE AND DATA SET INTERRUPT ENABLED. ENSURE
* THAT AN INTERRUPT IS RECEIVED.
* SUBTEST 6 - SET LL WITH RECEIVE AND DATA SET INTERRUPT ENABLED. ENSURE
* THAT AN INTERRUPT IS RECEIVED.
*
*****
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39

```
*****
*                               TEST 14 - DPV-11
* RECEIVE AND MODEM STATUS INTERRUPTS
* CHANGE THE MODEM STATUS WHILE HANDLING A RECEIVE INTERRUPT.
* ENSURE THAT THE MODEM STATUS INTERRUPT IS RECEIVED.
* SUBTEST 1 - CHANGE RTS DURING THE RECEIVE INTERRUPT. ENSURE THAT
*               THE DATA SET INTERRUPT WAS RECEIVED.
* SUBTEST 2 - CHANGE DTR DURING THE RECEIVE INTERRUPT. ENSURE THAT
*               THE DATA SET INTERRUPT WAS RECEIVED.
* SUBTEST 3 - CHANGE LL DURING THE RECEIVE INTERRUPT. ENSURE THAT
*               THE DATA SET INTERRUPT WAS RECEIVED.
*
*****
```

```
*****
*                               TEST 15 - DPV-11
* SUBTEST 1 - SECONDARY ADDRESS
*   SEGMENT 1 - SELECT SECONDARY ADDRESS AND SEND THE CORRECT
*               ADDRESS. CHECK THE DATA IS PROPERLY RECEIVED.
*   SEGMENT 2 - SELECT SECONDARY ADDRESS AND SEND A MESSAGE WITHOUT
*               SENDING USING THE SECONDARY ADDRESS. CHECK THAT A
*               TIME OUT IS RECEIVED.
*
* SUBTEST 2 - ALL PARTIES ADDRESSING
*   SEGMENT 1 - SELECT ALL PARTIES AND SECONDARY ADDRESS. SEND A
*               MESSAGE USING THE ALL PARTIES ADDRESS. ENSURE THAT
*               THE MESSAGE IS CORRECTLY RECEIVED.
*   SEGMENT 2 - SELECT ALL PARTIES AND SECONDARY ADDRESS. SEND A
*               MESSAGE WITHOUT ALL PARTIES OR SECONDARY ADDRESS.
*               CHECK THAT A TIME OUT IS RECEIVED.
*   SEGMENT 3 - SELECT ALL PARTIES AND SECONDARY ADDRESS. SEND A
*               MESSAGE WITH A SECONDARY ADDRESS. CHECK THAT A
*               TIME OUT IS RECEIVED.
*
*****
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50

```
*****
*                                     TEST 16 - DPV-11
* ABORT TEST
* SUBTEST 1 - ABORT WITH IDLE CLEAR. ABORT CHARACTERS TRANSMITTED WHEN
*               THE ABORT BIT IS ASSERTED.
*               SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO 1,
*                               5 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*
* SUBTEST 2 - ABORT WITH IDLE SET. FLAGS TRANSMITTED WHEN THE ABORT BIT
*               IS ASSERTED.
*               SELECTED OPTIONS: BOP MODE, NO ERROR CHECKING, IDLE SET,
*                               5 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*
*****

*****
*                                     TEST 17 - DPV-11
* EXTENDED CONTROL AND ADDRESSING TEST
* CHECK THAT THE RECEIVER CAN RECOGNIZE EXTENDED ADDRESSING AND CONTROL
* CHARACTERS.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO 1,
*                   3 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK,
*                   EXTENDED CONTROL AND ADDRESSING SELECTED
*
*****

*****
*                                     TEST 18 - DPV-11
* TRANSMIT GO AHEAD
* TERMINATE A MESSAGE USING TRANSMIT GO AHEAD. CHECK THAT THE RECEIVE
* ABORT BIT IS SET WHEN THE END OF MESSAGE IS RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO 1, LOOP SET,
*                   5 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*
*****

*****
*                                     TEST 19 - DPV-11
* ASSEMBLED BIT COUNT
* TRANSMIT VARIOUS BIT LENGTHS WHILE RECEIVING AN 8 BIT CHARACTER.
* ENSURE THAT THE ASSEMBLED BIT COUNT (ABC) IS CORRECT UPON THE END
* OF MESSAGE.
*   SELECTED OPTIONS: BOP MODE, NO ERROR CHECKING, VARIOUS BIT
*                   LENGTH CHARACTERS, MAINTENANCE MODE LOOPBACK.
*
*****
```

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

```
*****
*                                     TEST 20 - DPV-11
* SPECIAL SPACE SEQUENCE
* START A MESSAGE USING A SPECIAL SPACE SEQUENCE. CHECK THAT THE
* MESSAGE IS CORRECTLY TRANSMITTED AND RECEIVED.
* NOTE: CERTAIN USYNRTS ONLY TRANSMIT A SPECIAL START SEQUENCE WHEN
*       TRANSMIT START AND END OF MESSAGE ARE SET BY A BYTE OPERATION.
*
*       SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO 1,
*                        5 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*****
```

```
*****
*                                     TEST 21 - DPV-11
* SYNCH CHARACTER
* CHECK THAT A SYNCH CHARACTER OF 271 CAN BE USED TO COMMENCE A MESSAGE.
* VERIFY THAT THE MESSAGE IS CORRECTLY TRANSMITTED AND RECEIVED.
*       SELECTED OPTIONS: BCP MODE, VRC-EVEN PARITY,
*                        7 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*****
```

```
*****
*                                     TEST 22 - DPV-11
* SYNCH FROM TRANSMIT DATA PATH
* TRANSMIT A MESSAGE USING THE SYNCH FROM THE TRANSMIT DATA PATH.
* VERIFY THAT THE MESSAGE IS CORRECTLY TRANSMITTED AND RECEIVED.
*       SELECTED OPTIONS: BCP MODE, VRC-ODD PARITY, IDLE SET
*                        5 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*****
```

```
*****
*                                     TEST 23 - DPV-11
* STRIP SYNCHS
* SEND MORE THAN 2 SYNCHS WITH THE STRIP SYNCH BIT SET. CHECK THAT
* THE MESSAGE IS CORRECTLY TRANSMITTED AND RECEIVED.
*       SELECTED OPTIONS: BCP MODE, VRC-ODD PARITY, STRIP SYNCH SET
*                        6 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*****
```



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

```
*****
*                                     *
*          TEST 24 - DPV-11          *
* CRC-CCITT PRESET TO ONES.          *
* CHECK TO ENSURE THAT THE ERROR CHECK BIT (BIT 15 OF RDSR) IS *
* SET WHEN AN ABORT IS RECEIVED. IN BOP MODE THIS BIT IS SET WHEN THE *
* CRC IS IN ERROR. THE ERROR CHECK BIT SHOULD BE ZERO WHEN REOM=1, *
* IF THE CRC WERE CORRECTLY RECEIVED. BY FORCING AN ABORT WE INTENTIONALLY *
* LOOK AT THE ERROR BIT WHEN IT SHOULD BE IN AN ERROR STATE. *
*          SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO 1, LOOP SET, *
*          4 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK. *
*                                     *
*****
```

```
*****
*                                     *
*          TEST 25 - DPV-11          *
* CRC-CCITT PRESET TO ZERO.          *
* CHECK TO ENSURE THAT THE ERROR CHECK BIT (BIT 15 OF RDSR) IS *
* SET WHEN AN ABORT IS RECEIVED. IN BOP MODE THIS BIT IS SET WHEN THE *
* CRC IS IN ERROR. THE ERROR CHECK BIT SHOULD BE ZERO WHEN REOM=1, *
* IF THE CRC WERE CORRECTLY RECEIVED. BY FORCING AN ABORT WE INTENTIONALLY *
* LOOK AT THE ERROR BIT WHEN IT SHOULD BE IN AN ERROR STATE. *
*          SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO 0, LOOP SET, *
*          8 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK. *
*                                     *
*****
```

```
*****
*                                     *
*          TEST 26 - DPV-11          *
* CRC-16 PRESET TO 0                *
*                                     *
* SUBTEST 1 - CRC-16 ERROR          *
* CHECK TO ENSURE THAT THE ERROR CHECK BIT (BIT 15 OF RDSR) IS *
* CLEAR IF THE RECEIVER IS SHUTDOWN BEFORE THE CRC IS RECEIVED. *
* IN BCP MODE THIS BIT IS CLEAR WHEN THE CRC IS IN ERROR. *
* THE ERROR CHECK BIT SHOULD BE SET WHEN THE LAST CHARACTER IS RECEIVED, *
* IF THE CRC WERE GOOD. *
*          SELECTED OPTIONS: BCP MODE, CRC-16 PRESET TO 0, LOOP SET, *
*          8 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK. *
*                                     *
* SUBTEST 2 - CRC-16 CHECK          *
* CHECK THAT THE CORRECT CRC-16 IS RECEIVED FOR THE DATA MESSAGE. *
* THE CRC FOR THIS DATA MESSAGE WAS PREDETERMINED. *
*                                     *
*****
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50

```
*****
*                               TEST 27 - DPV-11
* VRC ODD PARITY ERROR
* BY SELECTING DIFFERENT CHARACTER LENGTHS IN THE RECEIVER AND
* TRANSMITTER, CAUSE A PARITY ERROR TO OCCUR.
*   SELECTED OPTIONS: BCP MODE, VRC-ODD PARITY, XMIT=7 &
*                     RCV=6 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*****

*****
*                               TEST 28 - DPV-11
* VRC EVEN PARITY ERROR
* BY SELECTING DIFFERENT CHARACTER LENGTHS IN THE RECEIVER AND
* TRANSMITTER, CAUSE A PARITY ERROR TO OCCUR.
*   SELECTED OPTIONS: BCP MODE, VRC-EVEN PARITY, XMIT=5 &
*                     RCV=4 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*****

*****
*                               TEST 29 - DPV-11
* DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE WITHOUT THE USE OF INTERRUPT
* SERVICE ROUTINES. CHECK THAT THE DATA IS CORRECT.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ONES,
*                     8 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK.
*****

*****
*                               TEST 30 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ZERO,
*                     6 BIT CHARACTERS, USER SELECTED LOOPBACK.
*****

*****
*                               TEST 31 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ONES,
*                     5 BIT CHARACTERS, USER SELECTED LOOPBACK.
*****
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52

```
*****
*                               TEST 32 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ZERO,
*                     7 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****

*****
*                               TEST 33 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ONES,
*                     8 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****

*****
*                               TEST 34 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
* NOTE: CERTAIN USYNRTS ONLY TRANSMIT A SPECIAL START SEQUENCE WHEN
*       TRANSMIT START AND END OF MESSAGE ARE SET BY A BYTE OPERATION.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ONES,
*                     6 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****

*****
*                               TEST 35 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ZEROS,
*                     7 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****

*****
*                               TEST 36 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ZERO, LOOP SET,
*                     8 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****
```

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

```
*****
*                               TEST 37 - DPV-11
* BCP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BCP MODE, VRC-ODD PARITY, IDLE BIT SET
*                     6 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****

*****
*                               TEST 38 - DPV-11
* BCP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BCP MODE, VRC-EVEN PARITY,
*                     5 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****

*****
*                               TEST 39 - DPV-11
* BCP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BCP MODE, CRC-16 PRESET TO ONES, STRIP SYNCHS,
*                     7 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****

*****
*                               TEST 40 - DPV-11
* BCP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BCP MODE, CRC-16 PRESET TO ONES,
*                     8 BIT CHARACTERS, USER SELECTED LOOPBACK.
*
*****
```

```
*****
*                               TEST 41 - DPV-11
* DDCMP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE USING THE
* DDCMP MESSAGE FORMAT. CHECK THAT THE DATA IS CORRECTLY RECEIVED
* AND THAT THE CRC CHARACTERS ARE RECEIVED IN THE PROPER DDCMP
* ORDER.
*   SELECTED OPTIONS: BCP MODE, CRC-16 PRESET TO ONES, STRIP SYNCHS
*   8 BIT CHARACTERS, USER SELECTED LOOPBACK.
*****

*****
*                               TEST 42 - DPV-11
* BCP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BCP MODE, CRC-16 PRESET TO ONES,
*   5 OR 8 BIT CHARACTERS, USER SELECTED LOOPBACK.
*****

*****
*                               TEST 43 - DPV-11
* BOP DATA TEST
* TRANSMIT AND RECEIVE A COMPLETE DATA MESSAGE. CHECK THAT THE
* DATA IS CORRECTLY RECEIVED.
*   SELECTED OPTIONS: BOP MODE, CRC-CCITT PRESET TO ONES,
*   5 OR 8 BIT CHARACTERS, USER SELECTED LOOPBACK.
*****
```

## 9.0 ERROR INFORMATION

### 9.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES AN "TIME OUT" ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE PC OF THE CALL TO THE SUBROUTINE REPORTING IT, THE FAILING REGISTER NAME, AND DEVICE REGISTER CONTENTS :

```
DPV DVC FTL ERR 00002 ON UNIT 00 TST 020 SUB 000 PC: 004756
TIME OUT - DURING INTERRUPT EXERCISE
ERROR IN SUBROUTINE CALLED AT PC: 031706
RXCSR: 000160
RDSR : 000000
```

58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72

TXCSR: 122432  
TDSR : 001402  
DPV EOP 1  
1 CUMULATIVE ERRORS

a



```
9          002000          .=2000
10
11
12
13
14          .MCALL  SVC
15 002000          SVC          ; INITIALIZE SUPERVISOR MACROS
16
17
18 002000          BGNMOD
19
20
21          000001          $LSTIN= 1          ; LIST INSTRUCTIONS
22          000001          $LSTTAG= 1
23          000001          SVCINS= 1          ; LIST INSTRUCTIONS, SHIFTED RIGHT
24          000001          SVCTST= 1          ; LIST TEST TAGS, SHIFTED RIGHT
25          000001          SVCSUB= 1          ; LIST SUBTEST TAGS, SHIFTED RIGHT
26          000001          SVCGBL= 1          ; LIST GLOBAL TAGS, SHIFTED RIGHT
27          000001          SVCTAG= 1          ; LIST OTHER TAGS, SHIFTED RIGHT
28
29          ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
30          ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
31          ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YCU MAY
32          ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
33
34 002000          POINTFR BGNDL
35
43
44
45
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11

• ♦ ♦

THE PROGRAM HEADER MACRO CHARACTERIZES THIS DIAGNOSTIC. THE HEADER MACRO'S ARGUMENTS ARE FILE NAME, RELEASE LEVEL, PATCH DISPOSITION OF THE MOST RECENT PATCH, MAXIMUM TEST TIME IN SEC., AND THE TYPE OF DIAGNOSTIC (0-SEQUENTIAL, 1-EXERCISER). THESE ARGUMENTS ARE IN RESPECTIVE ORDER.

HEADER CVDPV,A,0,200.,0

```
.ASCII /C/
.ASCII /V/
.ASCII /D/
.ASCII /P/
.ASCII /V/
.BYTE 0
.BYTE 0
.BYTE 0
```

.ASCII /A/

.ASCII /0/

.WORD 0

.WORD 200.

.WORD LSHARD

WORD 0

.WORD LSHW

WORD 0

WORD LAST

.WORD 0

WORD 0

WORD 0

WORD 0

WORD LSDISPATCH

WORD 0

WORD 0

WORD 0

```
.BYTE CSREVISION
.BYTE CSEDIT
```

```

002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 003674
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 000000
002072
002072 017750
002074
002074 000000
002076
002076 003702
002100
002100 104035
002102
002102 000000
002104
002104 015372
002106
002106 016564
002110
002110 016500
002112
002112 015364
002114
002114 000000
002116
002116 000000
002120
002120 000000

```

12  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

.EVEN

```

LSEF::
      .WORD 0
      .WORD 0
LSSPC::
      .WORD 0
LSDEVP::
      .WORD LSDVTYP
LSREPP::
      .WORD 0
LSEXP4::
      .WORD 0
LSEXP5::
      .WORD 0
LSAUT::
      .WORD 0
LSDUT::
      .WORD LSDU
LSLUN::
      .WORD 0
LSDESP::
      .WORD LSDESC
LSLOAD::
      EMT ESLOAD
LSETP::
      .WORD 0
LSICP::
      .WORD LSINIT
LSCCP::
      .WORD LSCLEAN
LSACP::
      .WORD LSAUTO
LSPRT::
      .WORD LSPROT
LSTEST::
      .WORD 0
LSDLY::
      .WORD 0
LSHIME::
      .WORD 0

```

1  
2  
3  
4  
5  
6  
7  
8

.SBTTL DISPATCH TABLE

////////////////////////////////////  
// THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.  
// IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.  
////////////////////////////////////

DISPATCH 43

002122 000053  
002124 020030  
002126 020434  
002130 020660  
002132 021410  
002134 021752  
002136 022204  
002140 022362  
002142 022614  
002144 023112  
002146 024120  
002150 025234  
002152 025500  
002154 025770  
002156 026700  
002160 030004  
002162 030714  
002164 031240  
002166 031412  
002170 031622  
002172 032072  
002174 032240  
002176 032424  
002200 032610  
002202 032774  
002204 033164  
002206 033352  
002210 033770  
002212 034160  
002214 034350  
002216 034746  
002220 035122  
002222 035304  
002224 035474  
002226 035646  
002230 036022  
002232 036204  
002234 036366  
002236 036570  
002240 036772  
002242 037174  
002244 037352  
002246 037624  
002250 040034

.WORD 43  
LSDISPATCH::  
.WORD T1  
.WORD T2  
.WORD T3  
.WORD T4  
.WORD T5  
.WORD T6  
.WORD T7  
.WORD T8  
.WORD T9  
.WORD T10  
.WORD T11  
.WORD T12  
.WORD T13  
.WORD T14  
.WORD T15  
.WORD T16  
.WORD T17  
.WORD T18  
.WORD T19  
.WORD T20  
.WORD T21  
.WORD T22  
.WORD T23  
.WORD T24  
.WORD T25  
.WORD T26  
.WORD T27  
.WORD T28  
.WORD T29  
.WORD T30  
.WORD T31  
.WORD T32  
.WORD T33  
.WORD T34  
.WORD T35  
.WORD T36  
.WORD T37  
.WORD T38  
.WORD T39  
.WORD T40  
.WORD T41  
.WORD T42  
.WORD T43

```
1          .SBTTL  DEFAULT HARDWARE P-TABLE
2
3          ;////////////////////////////////////
4          ;// THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES FOR
5          ;// THE TEST-DEVICE PARAMETERS.
6          ;////////////////////////////////////
7
8          002252      BGNHW  DFPTBL
9          002252      000003
10         002254
11         002254
12         002254
13
14         002254      160010      .WORD  160010      ;DPV11 CSR UNIBUS ADDRESS
15         002256      000300      .WORD      300      ;DPV11 INTERRUPT VECTOR
16         002260      000001      .WORD      1      ;TURNAROUND (DEFAULT = RS423)
17
18         002262      ENDPHW
19
20
21
22
23
24
```

LSHW:: .WORD L10000-LSHW/2  
DFPTBL::  
L10000:

1  
2  
3  
4  
5  
6  
7  
8 002262

.SBTTL GLOBAL EQUATES SECTION

////////////////////////////////////  
:/ THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT  
:/ ARE USED IN MORE THAN ONE TEST.  
////////////////////////////////////

EQUALS

: BIT DEFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1
001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

: EVENT FLAG DEFINITIONS  
EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	: START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	: RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	: CONTINUE COMMAND WAS ISSUED
000035	EF.NEW== 29.	: A NEW PASS HAS BEEN STARTED
000034	EF.PWR== 28.	: A POWER-FAIL/POWER-UP OCCURRED

: PRIORITY LEVEL DEFINITIONS

000340	PRI07== 340
000300	PRI06== 300
000240	PRI05== 240
000200	PRI04== 200
000140	PRI03== 140
000100	PRI02== 100

```

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

000040
000000

000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000

PRI01== 40
PRI00== 0
;
; OPERATOR FLAG BITS
;
EVL== 4
LOT== 10
ADR== 20
IDU== 40
ISR== 100
UAM== 200
BOE== 400
PNT== 1000
PRI== 2000
IXE== 4000
IBE== 10000
IER== 20000
LOE== 40000
HOE== 100000
;*****
;*****
; SWITCH REGISTER OPTIONS
SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1
;*****
; CSR AND STATUS WORD DEFINITIONS
; RXCSR - CSR0 (EXTERNAL REGISTER) READ/WRITE BITS 0 - 6
SF= BIT0 ; SELECT FREQUENCY.
RL= BIT0 ; REMOTE LOOPBACK - IF WIRE WRAPPED
; SELECTED.
DTR= BIT1 ; DATA TERMINAL READY R/W
RTS= BIT2 ; REQUEST TO SEND R/W
LL= BIT3 ; LOCAL LOOPBACK
RXENA= BIT4 ; RECEIVER ENALBLE R/W
DSITEN= BIT5 ; DATA SET INTERRUPT ENABLE R/W
RXITEN= BIT6 ; RECEIVER INTERRUPT ENABLE R/W
; ** BITS 7 - 15 READ ONLY **
RDATRY= BIT7 ; RECEIVE DATA READY READ ONLY

```



```

47      000400      SFR=      BIT8      ;SYNCH OR FLAG DETECT READ ONLY
48      001000      DM=      BIT9      ;DATA MODE READ ONLY
49      002000      RSTARY= BIT10      ;RECEIVER STATUS READY READ ONLY
50      004000      RXACT=  BIT11      ;RECEIVER ACTIVE READ ONLY
51      010000      RR=      BIT12      ;RECEIVER READY READ ONLY
52      020000      CTS=     BIT13      ;CLEAR TO SEND READ ONLY
53      040000      IC=      BIT14      ;INCOMING CALL READ ONLY
54      100000      DSCNG=   BIT15      ;DATA SET CHANGE READ ONLY
55
56
57      ;;PCSAR - CSR2 (INTERNAL USNYR/T REGISTERS 4 AND 5) WRITE ONLY
58
59      ;BITS 0-7 SYNCH CHARACTER OR SECONDARY STATION
60      ;ADDRESS. LOWER BYTE OF THE PCSAR IS THE
61      ;SYNCH CHARACTER USED WITH IN BCP MODE OR
62      ;THE SECONDARY ADDRESS USED IN BOP MODE.
63
64      ;BITS 8-10 ERROR DETECTION SELECTION
65      000000      CCITT1= 0      ;CRC CCITT INITIALIZED TO ONES
66      000400      CCITT0= BIT8      ;CRC CCITT INITIALIZED TO ZEROS
67      001400      CRC16= BIT8!BIT9 ;CRC 16
68      002000      VRCE=  BIT10      ;VRC ODD PARITY
69      002400      VRCE=  BIT8!BIT10 ;VRC EVEN PARITY
70      003400      NOERR= BIT8!BIT9!BIT10 ;ALL ERROR DETECTION INHIBITED.
71      001000      NONE1= BIT9      ;NOT USED
72      003000      NONE2= BIT9!BIT10 ;NOT USED
73
74      004000      IDLE=  BIT11      ;IDLE MODE SELECT
75      010000      SECADR= BIT12      ;SECONDARY ADDRESS SELECT
76      020000      SSYNCH= BIT13      ;STRIP SYNCH - BCP
77      020000      LOOP=  BIT13      ;LOOP MODE - BOP
78      040000      PROTO=  BIT14      ;PROTOCOL SELECT.
79      100000      APA=      BIT15      ;ALL PARTIES ADDRESSED.
80
81
82      ;;RDSR - CSR2 (INTERNAL USNYR/Y REGISTERS 0 AND 1) READ ONLY
83
84      ;BITS 0-7 RECEIVE DATA BUFFER
85      000400      RSOM=  BIT8      ;RECEIVED START OF MESSAGE.
86      001000      REOM=  BIT9      ;RECEIVED END OF MESSAGE.
87      002000      RABORT= BIT10      ;RECEIVER ABORT OR GO AHEAD
88      004000      ROVER=  BIT11      ;RECEIVER OVERRUN.
89      ;BITS 12-14 ASSEMBLED BIT COUNT (ABC)
90      000000      ALL=  0      ;ALL BITS VALID
91      010000      ONE=  BIT12      ;ONE BIT VALID
92      020000      TWO=  BIT13      ;TWO BITS VALID
93      030000      THREE= BIT12!BIT13 ;THREE BITS VALID
94      040000      FOUR=  BIT14      ;FOUR BITS VALID
95      050000      FIVE=  BIT12!BIT14 ;FIVE BITS VALID
96      060000      SIX=   BIT13!BIT14 ;SIX BITS VALID
97      070000      SEVEN= BIT12!BIT13!BIT14 ;SEVEN BITS VALID
98
99      100000      ERR=  BIT15      ;ERROR CHECK
100
101
102      ;;TXCSR - CSR4 (EXTERNAL LO BYTE - INTERNAL 7 HI BYTE) PEAD/WRITE
103

```

104	000001	RESET=	BIT0	;DEVICE RESET - WRITE ONLY
105	000002	TXACT=	BIT1	;TRANSMITTER ACTIVE - READ ONLY
106	000004	TBE=	BIT2	;TRANSMITTER BUFFER EMPTY - READ ONLY
107	000010	MM=	BIT3	;MAINTENANCE MODE - R/W
108	000020	TXENA=	BIT4	;TRANSMITTER ENABLE - R/W
109	000040	SQ=	BIT5	;SIGNAL QUALITY -READ ONLY
110	000040	TM=	BIT5	;TEST MODE - READ ONLY WIRE WRAPPED FOR
111				;TEST MODE
112	000100	TXIE=	BIT6	;TRANSMIT INTERRUPT ENABLE - R/W
113				
114		;;PCR	- HI BYTE CSR4 (INTERNAL USNYR/T REGISTER 7)	
115				
116	000010	EXCON=	BIT3	;EXTENDED CONTROL FIELD
117	000020	EXADD=	BIT4	;EXTENDED ADDRESS FIELD.
118				
119		;;TDCSR	- CSR6 (INTERNAL USNYR/T REGISTERS 7 AND 7) READ/WRITE	
120				
121				;BITS 0-7 TRANSMITTER DATA
122	000400	TSOM=	BIT8	;TRANSMIT START OF MESSAGE - R/W
123	001000	TEOM=	BIT9	;TRANSMIT END OF MESSAGE - R/W
124	002000	TXABO=	BIT10	;TRANSMIT ABORT - R/W
125	004000	TGA=	BIT11	;TRANSMIT GO AHEAD - R/W
126				;BITS 12 - 14 RESERVED
127	100000	TERR=	BIT15	;TRANSMIT DATA LATE ERROR. - READ ONLY
128				
129				
130				
131		*****		
132		*****		
133		;; MISC. EQUATES		
134				
135	000226	SYN=	226	;DDCMP SYNCH CHARACTER
136	000207	RETURN=	207	;RETURN, FROM SUB. [= JSR PC]
137	100000	BOP=	BIT15	;BIT SET IN MODE WHEN IN BOP MODE
138	000015	CR=	15	;ASCII CARRIAGE RETURN
139	000012	LF=	12	;ASCII LINE FEED
140	000007	MFPT=	7	;OPCODE FOR LSI 11/23 TO MOVE PROCESSOR TYPE
141				;TO R0 R0=3 MEANS LSI 11/23 - ILLEGAL INSTRUCTION
142				;ON AN LSI 11 OR LSI 11/2
143	000332	CRCLO=	332	;LOW BYTE OF CRC IN TEST 26.
144	000266	CRCHI=	266	;HIGH BYTE OF CRC IN TEST 26.
145				

```

1      .SBTTL GLOBAL DATA SECTION
2
3      ;////////////////////////////////////
4      ;/ THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
5      ;/ IN MORE THAN ONE TEST.
6      ;////////////////////////////////////
7
8
9
10     ;*****
11     ;DPV11 VECTOR AND REGISTER INDIRECT POINTERS
12     002262 000000 RCVEC: .WORD 0 ;DPV11 RECEIVER INTERRUPT VECTOR
13     002264 000000 XMTVEC: .WORD 0 ;DPV11 TRANSMITTER INT. VECTOR
14     002266 000000 CSR0: .WORD 0 ;POINTER TO DPV11 CSR0
15     002270 000000 CSR2: .WORD 0 ;POINTER TO DPV11 CSR2
16     002272 000000 CSR4: .WORD 0 ;POINTER TO DPV11 CSR4
17     002274 000000 CSR6: .WORD 0 ;POINTER TO DPV11 CSR6
18     002276 000000 CSR1: .WORD 0 ;POINTER TO HIGH BYTE OF CSR0
19     002300 000000 CSR3: .WORD 0 ;POINTER TO HIGH BYTE OF CSR2
20     002302 000000 CSR5: .WORD 0 ;POINTER TO HIGH BYTE OF CSR4
21     002304 000000 CSR7: .WORD 0 ;POINTER TO HIGH BYTE OF CSR6
22
23     002266 RXCSR= CSR0 ;RECEIVER CSR (READ/WRITE)
24     002270 PCSAR= CSR2 ;PARAMETER CONTROL SYNCH/ADDRESS REG.
25     ;(WRITE ONLY)
26     002270 RDSR= CSR2 ;RECEIVE DATA/STATUS REGISTER (READ ONLY)
27     002272 TXCSR= CSR4 ;TRANSMITTER CSR (READ/WRITE)
28     002274 TDSR= CSR6 ;TRANSMIT DATA/STATUS REGISTER (READ ONLY)
29     002302 PCR= CSR5 ;PCR = PARAMETER CONTROL REGISTER
30
31     ;:OTHER HARDWARE PARAMETERS
32
33     002306 000000 TURN: .WORD 0 ;TURN AROUND TYPE (0-7)
34
35     ;*****
36     ;PROGRAM CONTROL PARAMETERS
37
38
39
40     002310 000000 FRSTIM: .WORD 0 ;FLAG=0 IF PROGRAM JUST LOADED
41     002312 000000 FRSPAS: .WORD 0 ;FLAG=0 IF FIRST PASS AFTER LOAD
42     002314 000000 STARES: .WORD 0 ;FLAG=0 IF 1ST TIME THRU AFTER STA OR RES
43
44
45
46     ;*****
47     ;PROGRAM VARIABLES
48
49     ;* MISCELLANEOUS STORAGE
50     002316 000000 ABORT: .WORD 0 ;FLAG TO ALLOW AN ABORT TO BE ISSUED.
51     002320 000000 BITS: .WORD 0 ;BITS TO BE SET IN THE CSR REGISTER
52     002322 000000 COUNTER: .WORD 0 ;COUNTER FOR # OF CHARACTERS TO RCV. (RDATA2)
53     002324 000000 CPU: .WORD 0 ;PROCESSOR TYPE
54     ;(3 = LSI11/23, 0 = LSI 11 OR LSI 11/2)
55     002326 000000 DATA: .WORD 0 ;COUNTER FOR # OF DATA CHARACTERS TRANSMITTED.
56     002330 000000 ERROR: .WORD 0 ;ERROR STORAGE
57     002332 000000 EXERR: .WORD 0 ;FLAG THAT AN ERROR IS EXPECTED IN DATA

```

58	002334	000000	FLAG:	.WORD	0	; SCRATCH WORD USED FOR MISC. FLAG IN SUB.
59	002336	000000	HEADER:	.WORD	0	; FLAG USED TO MARK DDCMP HEADER.
60	002340	000000	HIGH:	.WORD	0	; FLAG USED TO INDICATE HIGH SPEED ISR WHEN SET
61	002342	000000	IPCR:	.WORD	0	; IMAGE OF PCR
62	002344	000000	IPCSAR:	.WORD	0	; IMAGE OF PCSAR
63	002346	000000	IRXCSR:	.WORD	0	; IMAGE OF RXCSR
64	002350	000000	IRDSR:	.WORD	0	; IMAGE OF RDSR.
65	002352	000000	LENGTH:	.WORD	0	; CHARACTER LENGTH.
66	002354	000000	LOGDEV:	.WORD	0	; LOGICAL DEVICE NUMBER
67	002356	000000	MAINT:	.WORD	0	; MAINTENANCE MODE LOOPBACK FLAG
68	002360	000000	MCFLAG:	.WORD	0	; WORD USED IN TO TRACK MODEM CONTROL INT.
69	002362	000000	MODE:	.WORD	0	; PROTOCOL TYPE
70	002364	000000	NESTPC:	.WORD	0	; FLAG TO NOTIFY WHEN A SUBR IS NESTED
71	002366	000000	NXMFLG:	.WORD	0	; WORD USED WHEN ADDRESS IS NXM.
72	002370	000000	OVER:	.WORD	0	; FLAG TO ALLOW RECEIVE OVERRUN.
73	002372	000000	PSTACK:	.WORD	0	; CONTAINS BASE LEVEL PROGRAM SP
74	002374	000000	REG:	.WORD	0	; STORAGE OF A CSR ADDRESS
75	002376	000000	RFLAG:	.WORD	0	; WORD USED IN RECEIVE ROUTINE.
76	002400	000000	RSAVE:	.WORD	0	; TEMPORARY LOCATION TO SAVE RDSR ON INTERRUPT
77	002402	000000	RXINI:	.WORD	0	; RECEIVER INITIALIZATION
78	002404	000000	RXINIT:	.WORD	0	; RECEIVER INITIALIZATION WITH INT ENABLED.
79	002406	000000	RXMINI:	.WORD	0	; RECEIVER INIT WITH MAINTENANCE LOOPBACK.
80	002410	000000	SAVE:	.WORD	0	; SCRATCH WORD USED FOR MISC. STORAGE IN SUB.
81	002412	000000	SAVTIM:	.WORD	0	; STORAGE TO SAVE TIMER VALUE
82	002414	000000	START:	.WORD	0	; CONTER FOR # OF START CHARACTERS TO XMIT.
83	002416	000000	SUBRPC:	.WORD	0	; PC OF SUBR CALL FOR ERROR REPORTS
84	002420	000000	TEMP:	.WORD	0	; SCRATCH WORD USED FOR MISC. STORAGE IN SUB.
85	002422	000000	TEND:	.WORD	0	; TRANSMIT END
86	002424	000000	TFLAG:	.WORD	0	; WORD USED IN TRANSMIT INTERRUPT ROUTINE
87	002426	000000	TIMEO:	.WORD	0	; FLAG TO MARK TIME OUT IN \$DATA SUBROUTINE.
88	002430	000000	TIMER:	.WORD	0	; TIMER VALUE
89	002432	000000	TOGGLE:	.WORD	0	; FLAG TO ALLOW TOGGLE OF RTS IN TEST.
90	002434	000000	TSTART:	.WORD	0	; TRANSMIT START
91	002436	000000	TXINI:	.WORD	0	; TRANSMITTER INITIALIZATION
92	002440	000000	TXINIT:	.WORD	0	; TRANSMITTER INITIALIZATION WITH INT ENABLED.
93	002442	000000	TXMINI:	.WORD	0	; TRANSMITTER INIT WITH MAINTENANCE LOOPBACK
94						
95						
96						
97						
98						
99						
100						
101						
102						
103						
104						
105						
106						
107	002444		MODEM:	.BLKW	10.	; BUFFER AREA FOR MODEM STATUS
108						
109						
110						
111						
112						
113	002470	000000	XTYPE:	.WORD	0	; POINTER TO DATA TYPE TO LOAD INTO XMIT BUFFER
114	002472	000000	XCOUNT:	.WORD	0	; # OF CHARACTERS TO TRANSMIT.

115 002474 000000  
116 002476 000000  
117 002500 000000

ECOUNT: .WORD 0 ;# OF CHARACTERS FOR END OF MSG. IN BCP MODE.  
XMITD: .WORD 0 ;# OF CHARACTERS TRANSMITTED.  
RCOUNT: .WORD 0 ;# OF CHARACTERS RECEIVED.

118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129

::\*\*\*\*\*

;; \*\* CCITT PSUEDO-RANDOM TEST PATTERN \*\*  
; THE FOLLOWING 32 WORDS TRANSLATE INTO A 512 BIT PATTERN  
; THAT WAS GENERATED ACCORDING TO CCITT RECOMMENDATION V.52. THIS  
; PATTERN WAS GENERATED BY A 9 BIT SHIFT REGISTER (INITIALIZED  
; AS 1S) WHOSE 5TH AND 9TH BITS ARE XORED. THIS XOR RESULT IS SHIFTED  
; INTO THE 1ST BIT OF THE REGISTER AS THE REGISTER IS SHIFTED RIGHT.  
; THE 9TH BIT (OR BIT SHIFTED OUT) IS SHIFTED INTO THE BIT PATTERN.  
; NOTE: CCITT RECOMMENDED 511 BITS, I'VE EXTENDED THIS BY 1 BIT TO END  
; ON A WORD BOUNDARY.

130 002502  
131 002502 177603 157427 031011  
132 002510 047321 163715 105221  
133 002516 143325 142304 040041  
134 002524 014116 052606 172334  
135 002532 105025 123754 111337  
136 002540 111523 030030 145064  
137 002546 137642 143531 063617  
138 002554 135015 066730 026575  
139 002562 052012 053627 070071  
140 002570 151172 165044 031605  
141 002576 166632 016741

\$CCITT:  
.WORD 177603,157427,031011  
.WORD 047321,163715,105221  
.WORD 143325,142304,040041  
.WORD 014116,052606,172334  
.WORD 105025,123754,111337  
.WORD 111523,030030,145064  
.WORD 137642,143531,063617  
.WORD 135015,066730,026575  
.WORD 052012,053627,070071  
.WORD 151172,165044,031605  
.WORD 166632,016741

142  
143  
144  
145

::\*\*\*\*\*  
;; ALPHANUMERIC DATA

146 002602 101 102 103  
002605 104 105 106  
002610 107 110 111  
002613 112 113 114  
002616 115 116 117  
002621 120 121 122  
002624 123 124 125  
002627 126 127 130  
002632 131 132 060  
002635 061 062 063  
002640 064 065 066  
002643 067 070 071  
002646 000

ALPHA: .ASCIZ /ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789/

147 000045  
148  
149  
150  
151  
152  
153 002650 201  
154 002651 064 000  
155 002653 000  
156 002654 000  
157 002655 001  
158 000006  
159

ACOUNT= .-ALPHA ;CHARACTER COUNT  
.EVEN

::\*\*\*\*\*  
;; DDCMP BUFFER

DDCMP: .BYTE 201 ;SOH (START OF HEADER)  
.BYTE 64,0 ;COUNT AND FLAGS (BITS 0 AND 1 FLAGS)  
.BYTE 0 ;RESPONSE NUMBER  
.BYTE 0 ;TRANSMIT NUMBER  
.BYTE 1 ;STATION ADDRESS  
DDCMP1= .-DDCMP ;2 BYTES OF CRC16

160	002656	104	104	103	DDMSG: .ASCII /DDCMP MESSAGE/
	002661	115	120	040	
	002664	115	105	123	
	002667	123	101	107	
	002672	105			

161						
162		000015		DDCMP2= .-DDMSG		;2 BYTES OF CRC16
163						

164				::*****
165				:: TRANSMIT BUFFER

166				
167	002673			XMTBUF: .BLKB 256.

168				::*****
169				:: RECEIVE BUFFER
170				

171					
172	003273			RCVBUF: .BLKB 256.	;256. BYTE BUFFER
173		000400		RSIZE= .-RCVBUF	
174				.EVEN	
175					
176					

```

1      .SBTTL  GLOBAL TEXT SECTION
2
3      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4      ;X      THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5      ;X      MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
6      ;X      MORE THAN ONE TEST.
7      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8
9
10     ;*****
11     ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
12     ;*****
13     003674      DEVTYP  <DPV11>
14     003674
15     003674      104      120      126      LSDVTYP::
16     003677      061      061      000      .ASCIZ  /DPV11/
17
18     .EVEN
19
20     ;*****
21     ;* TITLE OF PROGRAM
22     ;*****
23     003702      DESCRIPT  <DIAGNOSTIC TESTS>
24     003702
25     003702      104      111      101      LSDDESC::
26     003705      107      116      117      .ASCIZ  /DIAGNOSTIC TESTS/
27     003710      123      124      111
28     003713      103      040      124
29     003716      105      123      124
30     003721      123      000
31
32     .EVEN
33
34     ;
35     ; FORMAT STATEMENTS USED IN PRINT CALLS
36     ;

```

.SBTTL GLOBAL SUBROUTINES

////////////////////////////////////  
// THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST  
////////////////////////////////////

\*\*\*\*\*  
\*\*\*\*\*

\*\*\*\*\*  
CALL MACRO - CALL ROUTINE = JSR PC, ROUTINE  
(NOTE: RETURN IS EQUATED TO A RTS PC)  
\*\*\*\*\*

.MACRO CALL ROUTIN  
.IF B,ROUTIN  
.ERROR ROUTINE; ## MISSING ROUTINE-EXPANSION ABORT ##  
.MEXIT  
.ENDC  
JSR PC,ROUTIN  
.ENDM

\*\*\*\*\*  
PUSH REGS MACRO  
\*\*\*\*\*

.MACRO PUSH REGS  
.IRP X,<REGS>  
MOV X,-(SP) ;PUSH REG ON STACK.  
.ENDR  
.ENDM PUSH

\*\*\*\*\*  
POP REGS MACRO  
\*\*\*\*\*

.MACRO POP REGS  
.IRP X,<REGS>  
MOV (SP)+,X ;POP REG OFF STACK.  
.ENDR  
.ENDM POP

\*\*\*\*\*  
WAIT MACRO  
\*\*\*\*\*

.MACRO WAIT \$BIT,ADDRESS  
.IF B,\$BIT  
.ERROR ROUTINE; ## MISSING ROUTINE-EXAPNSION ABORT ##  
.MEXIT  
.ENDC

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57



58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109

```
.NLIST
.LIST ME
.LIST

;***** MACRO EXPANSION *****

      .IF      B, ADDRESS
      .IF      IDN $BIT, TBE
      JSR      PC,$WAIT      ;CALL WAIT ROUTINE -
      .WORD    TBE           ;WAIT FOR TBE TO BE SET
      .WORD    TXCSR         ;IN TRANSMITTER CSR.

      .IFF
      JSR      PC,$WAIT      ;CALL WAIT ROUTINE -
      .WORD    $BIT          ;WAIT FOR BIT TO BE SET
      .WORD    RXCSR         ;IN RECEIVER CSR.

      .ENDC
      .IFF
      JSR      PC,$WAIT      ;CALL WAIT ROUTINE -
      .WORD    $BIT          ;WAIT FOR BIT TO BE SET
      .WORD    ADDRESS       ;IN THE GIVEN ADDRESS.

      .ENDC

;*****

.NLIST ME
.ENDM
```

```
::*****
::DELAY MACRO
::*****
```

```
.MACRO $DELAY $TIME
      .IF      B, $TIME
      .ERROR ROUTINE; ## MISSING ROUTINE-EXAPNSION ABORT ##
      .MEXIT
      .ENDC
```

```
.NLIST
.LIST ME
.LIST
```

```
JSR      PC,$DLAY      ;***** MACRO EXPANSION *****
      .WORD    $TIME    ;CALL DELAY SUBROUTINE
                        ;NUMBER OF DELAY LOOPS
                        ;*****
```

```
.NLIST ME
.ENDM
```

				SUBROUTINE \$WAIT	
				FUNCTION - TO WAIT FOR A BIT TO BE SET IN A GIVEN ADDRESS (USUALLY A DPV REGISTER).	
				CALLING FORMAT:	JSR PC,\$WAIT ;WORD         ;BIT ;WORD         ;ADDRESS
				ENTRY CONDITIONS -	
				EXIT CONDITIONS - EXIT WHEN BIT SET OR UPON TIME OUT. IF TIME OUT, PRINT TIME OUT ERROR.	
				CALLED BY - TESTS 4,5,7	
				REGISTERS DESTROYED - R0-R2 SAVED AND RESTORED	
				***** *****	
	\$WAIT:			MOV (\$PC),SUBRPC ;SAVE THE PC THAT CALLED THE ROUTINE.	
				SUB #4,SUBRPC ;CORRECT THE PC.	
				MOV @(\$PC),BITS ;SAVE THE BITS THAT WE ARE CHECKING.	
				ADD #2,(\$PC) ;UPDATE THE ADDRESS ON THE STACK.	
				MOV @(\$PC),R0 ;SAVE THE ADDRESS OF THE CSR POINTER	
				MOV @R0,R0 ;SAVE THE ACTUAL CSR ADDRESS.	
				ADD #2,(\$PC) ;UPDATE THE ADDRESS ON THE STACK.	
				PUSH <R2,R1,R0> ;PUSH REGS ON THE STACK	
				CLR R0 ;USE R0 AS A LOOP TIMER.	
	10\$:			MOV @R0,R1 ;SAVE THE CONTENTS OF THE CSR.	
				BIT BITS,R1 ;IS THE BIT SET ?	
				BNE 20\$ ;BRANCH IF SET	
				BREAK ;BREAK FOR SUPERVISOR.	
					TRAP CSBK
				DEC R0 ;DECREMENT TIMER	
				BNE 10\$ ;CONTINUE IF TIMER NOT EXPIRED.	
				MOV R1,R2 ;SAVE EXPECTED RESULTS FOR ERROR MESSAGE.	
				BIS BITS,R2 ;SET THE EXPECTED BITS.	
				ERRDF 0,EMG1,ERRG12 ;PRINT TIME OUT ERROR.	
					TRAP C\$ERDF
					.WORD 0
					.WORD EMG1
					.WORD ERRG12
				BIT #TBE,BITS ;WERE WE WAITING FOR TBE?	
				BEO 20\$ ;IF NOT, EXIT.	
				PRINTEB #FMS1 ;SUGGEST THAT THE XMIT CLOCK IS INOP.	
					MOV #FMS1,-(SP)
					MOV #1,-(SP)
					MOV SP,R0

ADDRESS	INSTR	OPCODE	OPERANDS	COMMENT	TRAP	CSPNTB
004060	104414					
004062	062706	000004			ADD	#4,SP
50 004066				20\$:		
51 004066				POP	<R0,R1,R2>	;POP REGISTERS OFF STACK.
52 004074	005037	002416		CLR	SUBRPC	
53						
54 004100	000207			RETURN		
55						
56 004102	045	101	050	FMS1:	.ASCIZ	/%A(CHECK THE XMIT CLOCK)%N/
004105	103	110	105			
004110	103	113	040			
004113	124	110	105			
004116	040	130	115			
004121	111	124	040			
004124	103	114	117			
004127	103	113	051			
004132	045	116	000			
57				.EVEN		
58						

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

004136			
004136	012777	000001	176126
004144	105777	176116	
004150	001015		
004152	005777	176112	
004156	001012		
004160	032777	000004	176104
004166	001406		
004170	105777	176106	
004174	001003		
004176	005777	176072	
004202	001413		
004204			
004204	011637	002416	
004210	162737	000004	002416
004216			
004216	104455		
004220	000001		
004222	013540		
004224	007572		
004226	005037	002416	
004232			
004232	017737	176030	002444
004240	042737	006760	002444
004246	032777	000040	176016
004254	001417		
004256	052737	000040	002444
004264	122777	000162	176004
004272	001010		
004274			
004274	012746	011522	
004300	012746	000001	
004304	010600		
004306	104414		
004310	062706	000004	

```

*****
*****
SUBROUTINE $RESET
FUNCTION - TO PERFORM A MASTER RESET AND TO CHECK THAT
           THE DPV IS IN THE PROPER INIT STATE.

CALLING FORMAT:      JSR      PC,$RESET

ENTRY CONDITIONS -

EXIT CONDITIONS - DEVICE IS RESET CORRECTLY OR AN ERROR IS REPORTED

CALLED BY           - TESTS 2-43

REGISTERS NOT AFFECTED

*****
*****
$RESET:
MOV      #RESET,@TXCSR      ;RESET THE DPV.
TSTB     @RXCSR             ;IS THE RECEIVE CSR = 0?
BNE      10$                ;IF NOT ERROR.
TST      @RDSR              ;IS THE RECEIVE STATUS AND DATA REG = 0?
BNE      10$                ;IF NOT, ERROR.
BIT      #4,@TXCSR          ;IS TBE SET?
BEQ      10$                ;IF NOT, ERROR.
TSTB     @PCR               ;IS THE PARAMETER CONTROL REG = 0?
BNE      10$                ;IF NOT, ERROR.
TST      @TDSR              ;IS THE XMIT STATUS AND DATA REG = 0?
BEQ      20$                ;IF YES - RESET OK.

10$:
MOV      (SP),SUBRPC         ;FLAG WHERE THIS SUBR. WAS CALLED.
SUB      #4,SUBRPC           ;ADJUST THE PC
ERRDF    1,EMG3,ERRG11       ;PRINT ERROR MESSAGE

TRAP     C$ERDF
        .WORD 1
        .WORD EMG3
        .WORD ERRG11

20$:
CLR      SUBRPC              ;CLEAR THE FLAG

MOV      @RXCSR,MODEM        ;SAVE THE MODEM STATUS.
BIC      #6760,MODEM         ;CLEAR ALL BUT MODEM
BIT      #TM,@TXCSR          ;IS TEST MODE SET?
BEQ      30$                 ;IF NOT OK
BIS      #TM,MODEM           ;OTHERWISE SET TM IN MODEM
;ALSO CHECK FOR -12V
CMPB     #162,@CSR1          ;ARE RING, CTS, CD AND DM ALSO SET?
BNE      30$                 ;IF NOT, PROBABLY HAVE -12V
PRINTB   #FMG9               ;PROMPT USER TO CHECK -12V.

MOV      #FMG9,-(SP)
MOV      #1,-(SP)
MOV      SP,R0
TRAP     C$PNTB
ADD      #4,SP

```

CVDPVBO DPV11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 29-1 K 4  
GLOBAL SUBROUTINES

49 004314

308:

50

51 004314 000207

RETURN

52

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31 004316  
32 004316 013701 002352  
33 004322 013702 002470  
34 004326 012703 002673  
35 004332 013704 002472  
36 004336 005737 002362  
37 004342 001444  
38  
39 004344 032737 100000 002344  
40 004352 001403  
41 004354 112723 000377  
42 004360 000422  
43 004362  
44 004362 032737 010000 002344  
45 004370 001403  
46 004372 113723 002344  
47 004376 000413  
48 004400  
49 004400 112223  
50 004402 032737 000020 002342  
51 004410 001406  
52 004412 142763 000001 177777  
53 004420 112223  
54 004422 005237 002472  
55 004426  
56 004426 112223  
57 004430 032737 000010 002342

```

*****
*****
SUBROUTINE $BUFRS
FUNCTION - TO SET UP THE TRANSMIT BUFFER WITH A DATA
          PATTERN AND TO CLEAR THE RECEIVE BUFFER

CALLING FORMAT:      JSR      PC,$BUFRS

ENTRY CONDITIONS - IPCSAR = IMAGE OF THE PCSAR (CSR 2 OF THE DPV)
                   IPCR   = IMAGE OF THE PCR (CSR 5 OF THE DPV)
                   XTYPE  = ADDRESS OF THE XMIT TYPE
                   XCOUNT = # OF CHARACTERS TO TRANSMIT
                   LENGTH = CHARACTER LENGTH
                   MODE    = PROTOCOL TYPE (BCP OR BOP)

EXIT CONDITIONS - ECOUNT = # OF CHARACTERS TO TRANSMIT (MODIFIED
                      XCOUNT)
                  XMTBUF = CONTAINS XMIT DATA TYPE PATTERN
                  RCVBUF = RECEIVE BUFFER CLEARED

CALLED BY      - TESTS 15-40

REGISTERS R1-R4 DESTROYED

*****
*****
$BUFRS:
MOV     LENGTH,R1      ;GET THE CHARACTER LENGTH
MOV     XTYPE,R2       ;ADDRESS OF DATA TYPE
MOV     #XMTBUF,R3     ;ADDRESS OF TRANSMIT BUFFER.
MOV     XCOUNT,R4      ;CHARACTER COUNT.
TST     MODE           ;WHAT MODE?
BEQ     10$            ;IF BCP, SKIP ADDRESS CHECK.

BIT     #APA,IPCSAR     ;IS APA DESIRED?
BEQ     5$             ;IF NOT CHECK SECONDARY ADDRESS.
MOVB    #377,(R3)+      ;PUT APA IN THE XMIT BUFFER
BR      7$

5$:
BIT     #SECADR,IPCSAR  ;IS THE SECONDARY ADDRESS DESIRED?
BEQ     6$             ;IF NOT - JUST LOAD DATA
MOVB    IPCSAR,(R3)+    ;PUT SECONDARY ADDRESS IN THE XMIT BUFFER.
BR      7$

6$:
MOVB    (R2)+,(R3)+     ;LOAD ADDRESS CHARACTER
BIT     #EXADD,IPCR     ;IS EXTENDED ADDRESS REQUESTED?
BEQ     7$             ;BR IF NOT
BICB    #BIT0,-1(R3)    ;MAKE SURE THE LSB OF THE ADDRESS IS 0
MOVB    (R2)+,(R3)+     ;GET THE EXTENDED ADDRESS BYTE.
INC     XCOUNT          ;COMPENSATE TRANSMIT COUNT.

7$:
MOVB    (R2)+,(R3)+     ;LOAD CONTROL CHARACTER
BIT     #EXCON,IPCR     ;IS EXTENDED CONTROL DESIRED?

```

PC	OPCODE	OPERAND 1	OPERAND 2	OPERAND 3	OPERAND 4	INSTR	COMMENT
58	004436	001403				BEQ	8\$
59	004440	112223				MOV	(R2)+,(R3)+
60	004442	005237	002472			INC	XCOUNT
61	004446						8\$:
62	004446	062737	000002	002472		ADD	#2,XCGUNT
63	004454						10\$:
64	004454	013737	002472	002474		MOV	XCOUNT,ECOUNT
65	004462						11\$:
66	004462	112213				MOV	(R2)+,(R3)
67	004464	146123	004514			BIC	MASK(R1),(R3)+
68	004470	005304				DEC	R4
69	004472	001373				BNE	11\$
70							
71	004474	012701	003273			MOV	#RCVBUF,R1
72	004500	012702	000400			MOV	#RSIZE,R2
73	004504						20\$:
74	004504	105021				CLRB	(R1)+
75	004506	005302				DEC	R2
76	004510	001375				BNE	20\$
77							
78	004512	000207				RETURN	
79							
80	004514	000	376	374		MASK:	.BYTE 0,376,374,370,360,340,300,200,0
	004517	370	360	340			
	004522	300	200	000			
81							.EVEN
82							

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

004526

004526	005037	002376
004532	005037	002424
004536	005037	002360
004542	005037	002330
004546	005037	002426
004552	005037	002476
004556	012701	003273
004562	012702	002673
004566	013703	002472
004572	005037	002500
004576	005737	002340
004602	001435	

\*\*\*\*\*  
\*\*\*\*\*

# SUBROUTINE \$DATA

## FUNCTION -

CALLING FORMAT: JSR PC,\$DATA  
JSR PC,\$DATA1

ENTRY CONDITIONS - RCVBUF = CLEARED RECEIVE BUFFER  
XMTBUF = XMIT BUFFER  
MAINT = MAINTENANCE MODE FLAG  
IF SET, MAINT. MODE DESIRED  
RXMINI = RECEIVER INIT WITH MAINTENANCE MODE SET.  
RXINIT = USER SELECTED RECEIVER INIT WORD.  
TXMINI = XMIT INIT WORD WITH MAINTENANCE MODE SET.  
TXINIT = USER SELECTED XMIT INIT WORD  
TIMER = TIME OUT VALUE (DETERMINED IN INIT -  
DEPENDENT ON PROCESSOR TYPE)  
EXERR = FLAG FOR EXPECTED ERROR.  
0 = NO ERROR EXPECTED.  
NONO = ERROR EXPECTED.

EXIT CONDITIONS - IF A CORRECT DATA TRANSMISSION - CARRY CLEAR  
IF ERROR IN TRANSMISSION - CARRY SET AND ERROR  
FLAG SET. IF ERROR WAS NOT EXPECTED, A MESSAGE  
WILL BE OUTPUT.

CALLED BY - \$DATA - TESTS 15-28 & 30 - 40  
\$DATA1 - TESTS 41 -43

REGISTERS R1-R5 DESTROYED

\*\*\*\*\*  
\*\*\*\*\*  
\$DATA:

CLR	RFLAG	;CLEAR THE RECEIVE FLAG
CLR	TFLAG	;CLEAR THE TRANSMIT FLAG
CLR	MCFLAG	;CLEAR THE MODEM CONTROL FLAG
CLR	ERROR	;ERROR CONDITION FLAG
CLR	TIMEO	;CLEAR TIMEOUT FLAG
CLR	XMITD	;CLEAR XMIT COUNTER.
MOV	#RCVBUF,R1	;RECEIVE BUFFER
MOV	#XMTBUF,R2	;TRANSMIT BUFFER
MOV	XCOUNT,R3	;TRANSMIT COUNTER
CLR	RCOUNT	;CLEAR RECEIVE COUNTER.
		;SET UP THE VECTORS.
TST	HIGH	;IS THIS A HIGH SPEED TEST?
BEQ	5\$	;BRANCH IF LOW SPEED
		;SET VECTORS WITH THE HIGH SPEED ISRS



```

58 004604          SETVEC  XMTVEC,#XDATA2,#PRI04 ;HIGH SPEED BOP XMIT ISR.
    004604 012746 000200
    004610 012746 017512
    004614 013746 002264
    004620 012746 000003
    004624 104437
    004626 062706 000010
59 004632          SETVEC  RCVEC,#RDATA2,#PRI04 ;HIGH SPEED RECV VECTOR
    004632 012746 000200
    004636 012746 017164
    004642 013746 002262
    004646 012746 000003
    004652 104437
    004654 062706 000010
60 004660 042737 000040 002404
61 004666 013737 002472 002322
62 004674 000426
63 004676
64 004676          SETVEC  XMTVEC,#XDATA.#PRI04 ;XMIT VECTOR
    004676 012746 000200
    004702 012746 017340
    004706 013746 002264
    004712 012746 000003
    004716 104437
    004720 062706 000010
65 004724          SETVEC  RCVEC,#RDATA,#PRI04 ;RECV VECTOR
    004724 012746 000200
    004730 012746 016732
    004734 013746 002262
    004740 012746 000003
    004744 104437
    004746 062706 000010
66 004752
67 004752          SETPRI  #PRI00 ;ENABLE INTERRUPTS
    004752 012700 000000
    004756 104441
68 004760 005737 002356
69 004764 001407
70 004766 053777 002406 175272
71 004774 053777 002442 175270
72 005002 000411
73
74 005004          $DATA1:
75 005004 053777 002404 175254
76 005012 053777 002440 175252
77 005020 052737 000040 002404
78
79 005026          $GO:
80 005026 011637 002416
81 005032 162737 000004 002416
82 005040 013704 002430
83 005044
84 005044 012705 001000
85 005050
86 005050 005777 175220
87 005054 100426
88 005056 005737 002376

    MOV      #PRI04,-(SP)
    MOV      #XDATA2,-(SP)
    MOV      XMTVEC,-(SP)
    MOV      #3,-(SP)
    TRAP     C$$SVEC
    ADD      #10,SP

    MOV      #PRI04,-(SP)
    MOV      #RDATA2,-(SP)
    MOV      RCVEC,-(SP)
    MOV      #3,-(SP)
    TRAP     C$$SVEC
    ADD      #10,SP

    BIC      #DSITEN,RXINIT ;IGNORE DATA SET INTERRUPTS IN HIGH SPEED.
    MOV      XCOUNT,COUNTER ;SET UP COUNTER FOR INT SERVICE ROUTINE RDATA2
    BR       7$

    MOV      #PRI04,-(SP)
    MOV      #XDATA,-(SP)
    MOV      XMTVEC,-(SP)
    MOV      #3,-(SP)
    TRAP     C$$SVEC
    ADD      #10,SP

    MOV      #PRI04,-(SP)
    MOV      #RDATA,-(SP)
    MOV      RCVEC,-(SP)
    MOV      #3,-(SP)
    TRAP     C$$SVEC
    ADD      #10,SP

    MOV      #PRI00,RO
    TRAP     C$$SPRI

    TST      MAINT          ;SET MAINTENANCE MODE?
    BEQ      $DATA1         ;BR IF NOT
    BIS      RXMINI,@RXCSR   ;INIT RECEIVER WITH MAINTENANCE MODE
    BIS      TXMINI,@TXCSR   ;INIT TRANSMITTER WITH MAINT. MODE.
    BR       $GO

    BIS      RXINIT,@RXCSR   ;ISSUE RECEIVER INIT (DETERMINED IN INIT CODE)
    BIS      TXINIT,@TXCSR   ;ISSUE XMIT INIT (DETERMINED IN INIT CODE)
    BIS      #DSITEN,RXINIT  ;RESET THE DATA SET INTERRUPT (MAY BE CLEARED
                                ;IF THIS IS A HIGH SPEED TEST).

    MOV      (SP),SUBRPC     ;FLAG WHERE THIS SUBR. WAS CALLED.
    SUB      #4,SUBRPC       ;ADJUST THE PC
    MOV      TIMER,R4        ;SET UP TIMER

    MOV      #1000,R5       ;INNER LOOP COUNTER

    TST      @TDSR          ;IS THERE A TRANSMITTER ERROR?
    BMI      20$            ;BR IF YES
    TST      RFLAG          ;IS THE RECEIVER DONE?

```

89	005062	001033		BNE	22\$	;EXIT LOOP IF YES		
90	005064	005305		DEC	R5	;DECREMENT INNER LOOP COUNTER		
91	005066	001370		BNE	10\$	;LOOP UNTIL DONE		
92	005070	022737	000002 002306	CMP	#2,TURN	;IS THIS RS422?		
93	005076	001401		BEQ	11\$	;IF YES - DON'T ALLOW A SUPERVISOR BREAK.		
94	005100			BREAK		;BREAK FOR SUPERVISOR INTERRUPT	TRAP	C\$BRK
	005100	104422						
95	005102		11\$:					
96	005102	005304		DEC	R4	;DECREMENT OUTSIDE LOOP COUNTER		
97	005104	001357		BNE	8\$	;LOOP UNTIL DONE		
98	005106	005237	002426	INC	TIMED	;SET TIME OUT FLAG.		
99								
100	005112	005737	002332	TST	EXERR	;WAS AN ERROR EXPECTED?		
101	005116	001036		BNE	25\$	;IF YES - EXIT WITHOUT ERROR MESSAGE.		
102	005120			ERRDF	2,EMG2,ERRG2	;TIME OUT		
	005120	104455					TRAP	C\$ERDF
	005122	000002					.WORD	2
	005124	013473					.WORD	EMG2
	005126	006700					.WORD	ERRG2
103	005130	000422		BR	24\$			
104	005132		20\$:					
105	005132	042777	000020 175132	BIC	#TXENA,@TXCSR	;DISABLE THE TRANSMITTER.		
106	005140			ERRDF	3,EMG30,ERRG2	;TRANSMIT UNDERRUN		
	005140	104455					TRAP	C\$ERDF
	005142	000003					.WORD	3
	005144	014766					.WORD	EMG30
	005146	006700					.WORD	ERRG2
107	005150	000412		BR	24\$			
108	005152		22\$:					
109	005152	005737	002376	TST	RFLAG	;WAS THIS THE END OF MESSAGE?		
110	005156	100016		BPL	25\$	;OK - IF YES		
111	005160	005737	002332	TST	EXERR	;WAS AN ERROR EXPECTED?		
112	005164	001013		BNE	25\$	;IF YES - EXIT WITHOUT ERROR MESSAGE.		
113	005166			ERRDF	4,EMG31,ERRG2	;RECEIVER ERROR		
	005166	104455					TRAP	C\$ERDF
	005170	000004					.WORD	4
	005172	015004					.WORD	EMG31
	005174	006700					.WORD	ERRG2
114	005176		24\$:					
115	005176	012737	000001 002330	MOV	#1,ERROR	;FLAG ERROR		
116	005204	005037	002416	CLR	SUBRPC	;CLEAR THE SUBR PC FLAG		
117	005210	000261		SEC		;SET CARRY - ERROR		
118	005212	000403		BR	30\$			
119								
120	005214		25\$:					
121	005214	000241		CLC		;CLEAR CARRY - NO ERROR		
122	005216	005037	002416	CLR	SUBRPC	;CLEAR THE SUBR PC FLAG		
123	005222		30\$:					
124	005222	052777	000001 175042	BIS	#RESET,@TXCSR	;RESET THE DPV		
125	005230			CLRVEC	XMTVEC	;RESTORE VECTORS		
	005230	013700	002264				MOV	XMTVEC,RO
	005234	104436					TRAP	C\$CVEC
126	005236			CLRVEC	RCVEC	:		
	005236	013700	002262				MOV	RCVEC,RO
	005242	104436					TRAP	C\$CVEC
127	005244	000207		RETURN				
128								

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

005246

005246 005737 002340  
005252 001046  
005254 005737 002362  
005260 001404  
005262 005737 002350  
005266 100410  
005270 000421  
005272  
005272 032737 002000 002344  
005300 001015  
005302 005737 002350  
005306 100412  
005310  
005310 011637 002416  
005314 162737 000004 002416  
005322  
005322 104455  
005324 000005  
005326 015226  
005330 006652  
005332 000444  
005334  
023737 002472 002500

```

*****
*****
SUBROUTINE $CHECK
FUNCTION - AFTER A DATA TRANSMISSION CHECK
1. THE ERROR CHECK BIT 2. THAT THE XMIT AND RCV
CHARACTER COUNTS ARE EQUAL 3. THAT THE XMIT AND
RCV BUFFERS ARE IDENTICAL

CALLING FORMAT:      JSR      PC,$CHECK
                     JSR      PC,$CHK1

ENTRY CONDITIONS - IRDSR = IMAGE OF THE LAST RECEIVED RDSR
                  XCOUNT = TRANSMIT CHARACTER COUNT.
                  RCOUNT = RECEIVER CHARACTER COUNT.
                  XMTBUF = THE TRANSMIT BUFFER STARTING ADDRESS.
                  RCVBUF = THE RECEIVE BUFFER STARTING ADDRESS.
                  MODE = PROTOCOL MODE: 0 = BCP, NON0 = BOP

EXIT CONDITIONS - IF ERROR DETECTED, A MESSAGE WILL BE OUTPUT.

CALLED BY          - $CHECK - TESTS 15, 17-23, 29-40
                   $CHK1 - TESTS 41-43

REGISTERS R1 - R3 DESTROYED
*****
*****
$CHECK:
      .ENABL  LSB          ;ENABLE LOCAL SYMBOL BLOCK.

      TST     HIGH        ;IS THIS A HIGH SPEED TEST (HIGH SPEED 1SR5)
      BNE     5$          ;IF YES SKIP CRC ERROR CHECK AND
      TST     MODE        ;IS THIS BCP MODE?
      BEQ     1$          ;BR IF YES
      TST     IRDSR       ;IS THE ERROR BIT SET (BIT 15)
      BMI     3$          ;IF YES - CRC ERROR.
      BR      4$

1$:   BIT     #BIT10,IPCSAR ;WAS CRC16 USED? (ONLY TIME BIT 10 IS SET)
      BNE     4$          ;IF NOT DON'T CHECK BIT.
      TST     IRDSR       ;IS THE ERROR BIT SET (BIT 15)?
      BMI     4$          ;IF YES - OK

3$:   MOV     (SP),SUBRPC   ;FLAG WHERE THIS SUBP. WAS CALLED.
      SUB     #4,SUBRPC    ;ADJUST THE PC
      ERDF    5,EMG37,ERRG1 ;CRC ERROR

                                TRAP    C$ERDF
                                .WORD   5
                                .WORD   EMG37
                                .WORD   ERRG1

4$:   BR      30$

      CMP     XCOUNT,RCOUNT ;ARE THE CHARACTER COUNTS THE SAME.

```

```

54 005342 001412          BEQ      5$      ;IF YES - CONTINUE
55 005344 011637 002416    MOV      (SP),SUBRPC ;FLAG WHERE THIS SUBR. WAS CALLED.
56 005350 162737 000004 002416 SUB      #4,SUBRPC ;ADJUST THE PC
57 005356          ERRDF      6,EMG25,ERRG14 ;CHARACTER COUNTS DIFFERENT
    005356 104455          TRAP      C$ERDF
    005360 000006          .WORD      6
    005362 014725          .WORD      EMG25
    005364 010760          .WORD      ERRG14
58 005366 000426          BR        30$
59 005370          5$:
60 005370 012701 002673    MOV      #XMTBUF,R1 ;GET THE ADDRESS OF THE XMIT BUFFER.
61 005374 012702 003273    MOV      #RCVBUF,R2 ;GET THE ADDRESS OF THE RECV BUFFER.
62 005400 013703 002472    MOV      XCOUNT,R3 ;GET THE CHARACTER COUNT
63 005404          $CHK1:
64 005404 122122          CMPB      (R1)+,(R2)+ ;ARE THE CHARACTERS THE SAME
65 005406 001003          BNE      20$      ;IF NOT, REPORT THE ERROR
66 005410 005303          DEC      R3      ;DECREMENT THE COUNT.
67 005412 001414          BEQ      30$      ;LOOP UNTIL DONE
68 005414 000773          BR        $CHK1
69 005416          20$:
70 005416 011637 002416    MOV      (SP),SUBRPC ;FLAG WHERE THIS SUBR. WAS CALLED.
71 005422 162737 000004 002416 SUB      #4,SUBRPC ;ADJUST THE PC
72 005430 005301          DEC      R1      ;POINT TO DATA IN ERROR
73 005432 005302          DEC      R2      ;POINT TO DATA IN ERROR.
74 005434          ERRDF      7,EMG26,ERRG3 ;CHARACTERS DON'T MATCH
    005434 104455          TRAP      C$ERDF
    005436 000007          .WORD      7
    005440 014753          .WORD      EMG26
    005442 007014          .WORD      ERRG3
75 005444          30$:
76 005444 005037 002416    CLR      SUBRPC ;CLEAR THE SUBR PC FLAG
77          .DSABL      LSB      ;DISABLE LOCAL SYMBOL BLOCK.
78 005450 000207          RETURN
79

```

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

\*\*\*\*\*  
\*\*\*\*\*

# SUBROUTINE \$MODEM

FUNCTION - TO PRINT OUT THE MODEM STATUS FROM A TEST

CALLING FORMAT: JSR PC,\$MODEM

ENTRY CONDITIONS - ERROR = FLAG SET IF THERE WAS AN ERROR IN \$DATA  
MCFLAG = # OF MODEM CONTROL INTERRUPTS RECEIVED.  
ALSO USED AS THE INDEX INTO THE MODEM  
STATUS TABLE.  
MODEM = ADDRESS OF MODEM STATUS TABLE

EXIT CONDITIONS - IF THERE IS AN ERROR OR MORE THAN 1 MODEM  
CONTROL INTERRUPT, PRINT OUT MODEM STATUS.  
OTHERWISE, UNEVENTFUL EXIT.

CALLED BY - TESTS 30-40

REGISTERS R1-R3 DESTROYED

\*\*\*\*\*  
\*\*\*\*\*

\$MODEM:

MOV (SP),SUBRPC ;FLAG WHERE THIS SUBR. WAS CALLED.  
SUB #4,SUBRPC ;ADJUST THE PC  
TST ERROR ;WAS THERE AN ERROR IN THE \$DATA ROUTINE  
BNE 1\$ ;IF YES PRINT OUT STATUS  
CMP #1,MCFLAG ;WAS THERE MORE THAN 1 MODEM CONTROL INT?  
BGE 35\$ ;IF NOT - SKIP PRINT OUT  
ERRDF 8,EMG40,ERRG1 ;MULTIPLE INTERRUPTS RECEIVED

TRAP C\$ERDF  
.WORD 8  
.WORD EMG40  
.WORD ERRG1

PRINTB #FMODEM,MCFLAG ;PRINT THE NUMBER OF INTERRUPTS.

MOV MCFLAG,-(SP)  
MOV #FMODEM,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP

CMP #9,MCFLAG ;WERE MORE THAN NINE INTERRUPTS RECEIVED?  
BGE 1\$ ;IF NOT, SKIP THE NEXT MESSAGE.  
MOV #9,MCFLAG ;ONLY PRINT OUT 9 INTERRUPTS  
PRINTB #FMODE6 ;INFORM THE USER INTERRUPTS DISABLED.

MOV #FMODE6,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP

1\$:

43	005574	012701	002444	MOV	#MODEM,R1	;ADDRESS OF MODEM STATUS	
44	005600			PRINTB	#FMODE0		
	005600	012746	006115				MOV #FMODE0,-(SP)
	005604	012746	000001				MOV #1,-(SP)
	005610	010600					MOV SP,R0
	005612	104414					TRAP C\$PNTB
	005614	062706	000004				ADD #4,SP
45	005620			PRINTB	#FMODE1	;PRINT INITIAL MODEM STATUS	
	005620	012746	006144				MOV #FMODE1,-(SP)
	005624	012746	000001				MOV #1,-(SP)
	005630	010600					MOV SP,R0
	005632	104414					TRAP C\$PNTB
	005634	062706	000004				ADD #4,SP
46	005640			PRINTB	#FMODE2		
	005640	012746	006233				MOV #FMODE2,-(SP)
	005644	012746	000001				MOV #1,-(SP)
	005650	010600					MOV SP,R0
	005652	104414					TRAP C\$PNTB
	005654	062706	000004				ADD #4,SP
47	005660	005002		CLR	R2	;CLEAR COUNTER	
48	005662			5\$:			
49	005662	012703	006420	MOV	#MMASK,R3		
50	005666	012704	000012	MOV	#10.,R4	;# OF BITS TO CHECK IN THE MODEM STATUS	
51							
52	005672			10\$:			
53	005672	032311		BIT	(R3)+,(R1)	;CHECK THE BIT	
54	005674	001011		BNE	12\$	;IS IT SET?	
55	005676			PRINTB	#FMODE3	;IF NOT, PRINT A 0	
	005676	012746	006257				MOV #FMODE3,-(SP)
	005702	012746	000001				MOV #1,-(SP)
	005706	010600					MOV SP,R0
	005710	104414					TRAP C\$PNTB
	005712	062706	000004				ADD #4,SP
56	005716	000410		BR	15\$		
57	005720			12\$:			
58	005720			PRINTB	#FMODE4	;PRINT A 1	
	005720	012746	006266				MOV #FMODE4,-(SP)
	005724	012746	000001				MOV #1,-(SP)
	005730	010600					MOV SP,R0
	005732	104414					TRAP C\$PNTB
	005734	062706	000004				ADD #4,SP
59	005740			15\$:			
60	005740	005304		DEC	R4	;DECREMENT BIT COUNTER	
61	005742	001353		BNE	10\$	;LOOP UNTIL DONE.	
62							
63							
64	005744	005737	002360	TST	MCFLAG	;IS THIS THE LAST STATUS	
65	005750	001416		BEQ	30\$	;IF YES, EXIT	
66	005752	005721		TST	(R1)+	;INCREMENT MODEM STATUS POINTER.	
67	005754	005337	002360	DEC	MCFLAG	;DECREMENT MC FLAG	
68	005760	005202		INC	R2	;INCREMENT COUNTER.	
69							
70	005762			PRINTB	#FMODE5,R2	;PRINT NEXT MODEM	
	005762	010246					MOV R2,-(SP)
	005764	012746	006275				MOV #FMODE5,-(SP)
	005770	012746	000002				MOV #2,-(SP)
	005774	010600					MOV SP,R0

[illegible]

```

80 006232      000
    006233      045      116      045  FMODE2: .ASCIZ  /%N%MODEM ON RESET:/
    006236      101      115      117
    006241      104      105      115
    006244      040      117      116
    006247      040      122      105
    006252      123      105      124
    006255      072      000
81 006257      045      123      064  FMODE3: .ASCIZ  /%S4%A0/
    006262      045      101      060
    006265      070
82 006266      045      123      064  FMODE4: .ASCIZ  /%S4%A1/
    006271      045      101      061
    006274      000
83 006275      045      116      045  FMODE5: .ASCIZ  /%N%MODEM CHANGE %D1%A:/
    006300      101      115      117
    006303      104      105      115
    006306      040      103      110
    006311      101      116      107
    006314      105      040      045
    006317      104      061      045
    006322      101      072      000
84 006325      045      101      052  FMODE6: .ASCIZ  /%A** MODEM CONTROL INTERRUPT DISABLED AFTER 9 CHANGES **%N/
    006330      052      040      115
    006333      117      104      105
    006336      115      040      103
    006341      117      116      124
    006344      122      117      114
    006347      040      111      116
    006352      124      105      122
    006355      122      125      120
    006360      124      040      104
    006363      111      123      101
    006366      102      114      105
    006371      104      040      101
    006374      106      124      105
    006377      122      040      071
    006402      040      103      110
    006405      101      116      107
    006410      105      123      040
    006413      052      052      045
    006416      116      000
85                                     .EVEN
86                                     ;MASKS OF EACH BIT
87 006420      000001      000002      000004  MMASK: .WORD  SF,DTR,RTS,LL,IM,DM,RR,CTS,IC,DSCNG
    006426      000010      000040      001000
    006434      010000      020000      040000
    006442      100000
88
89
90

```



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

```

*****
*****
SUBROUTINE $TURN
FUNCTION - DETERMINE IF TURNAROUND IS AVAILABLE

CALLING FORMAT:      JSR      PC,$TURN

ENTRY CONDITIONS -   TURN - 0 = TURNAROUND OFF
                     STARES = START RESTART COUNT

EXIT CONDITIONS -    TURNAROUND ON - CARRY CLEAR (DO THE TEST)
                     TURNAROUND OFF - CARRY SET (DON'T DO THE TEST)
                     IF TURNAROUND OFF AND IF ON FIRST PASS, OUTPUT
                     A MESSAGE TO THE USER.

CALLED BY            - TESTS 12 - 14

REGISTERS NOT EFFECTED

```

```

*****
*****
$TURN:
TST      TURN          ;IS THERE A TURNAROUND
BNE      5$            ;IF YES - CLEAR CARRY TO DO THE TEST.
CMP      #1,STARES     ;IS THIS THE FIRST PASS
BNE      1$            ;IF NOT, DON'T OUTPUT MESSAGE JUST SET FLAG.
PRINTX   #FMGO,L$TEST,LOGDEV ;INFORM THE USER THAT TEST CAN'T BE RUN
                                MOV      LOGDEV,-(SP)
                                MOV      L$TEST,-(SP)
                                MOV      #FMGO,-(SP)
                                MOV      #3,-(SP)
                                MOV      SP,R0
                                TRAP     C$PNTX
                                ADD      #10,SP

                                ;WITHOUT THE TURNAROUND.

1$:      SEC
BR       10$           ;FLAG NOT TO DO THE TEST.
                                ;BR TO RETURN
5$:      CLC
                                ;FLAG TO DO THE TEST.
10$:     RETURN

```

```

006444 005737 002306
006444 001022
006450 022737 000001 002314
006452 001014
006460 001014
006462 013746 002354
006466 013746 002114
006472 012746 011070
006476 012746 000003
006502 010600
006504 104415
006506 062706 000010
006512
006512 000261
006514 000401
006516
006516 000241
006520
006520 000207

```

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

```

*****
*****
SUBROUTINE $SPEED
FUNCTION - DETERMINE IF THE TEST CAN BE RUN WITH
          WITH THE SELECTED TURNAROUND AND/OR PROCESSOR.

CALLING FORMAT:      JSR      PC,$SPEED

ENTRY CONDITIONS -   TURN = 1 - RS423
                     TURN = 2 - RS422
                     CPU  = 0 - LSI 11 OR LSI 11/2
                     CPU  = 3 - LSI 11/23

EXIT CONDITIONS -    OK TO RUN TEST - CARRY CLEAR
                     DON'T RUN TEST - CARRY SET
                     IF TEST CAN'T BE RUN, THE USER WILL BE
                     INFORMED ON THE FIRST PASS.

CALLED BY            - $SPEED CALLED BY TESTS 29 - 41

REGISTERS NOT EFFECTED

*****
*****
$SPEED:
TST      CPU          ;IS THIS A LSI 11/23?
BNE      5$           ;IF YES - CLEAR CARRY TO DO THE TEST.
CMP      #2,TURN      ;IS THIS RS422?
BNE      5$           ;IF NOT - CLEAR CARRY AND DO THE TEST.
CMP      #1,STARES    ;IS THIS THE FIRST PASS?
BNE      1$           ;IF NOT, DON'T OUTPUT MESSAGE JUST SET FLAG.
PRINTX   #FMG27,L$TEST ;INFORM THE USER THAT THE TEST CAN'T BE RUN
                                MOV      L$TEST,-(SP)
                                MOV      #FMG27,-(SP)
                                MOV      #2,-(SP)
                                MOV      SP,R0
                                TRAP     C$PNTX
                                ADD      #6,SP

                                ;WITH THIS CPU AND RS422.

1$:      SEC
BR       10$          ;FLAG NOT TO DO THE TEST.
                                ;BR TO RETURN.

5$:      CLC
                                ;FLAG TO DO THE TEST.

10$:     RETURN

```

```

006522 005737 002324
006522 001024
006530 022737 000002 002306
006536 001020
006540 022737 000001 002314
006546 001012
006550
006550 013746 002114
006554 012746 013011
006560 012746 000002
006564 010600
006566 104415
006570 062706 000006

006574
006574 000261
006576 000401
006600
006600 000241
006602
006602 000207

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32

006604  
006604 017600 000000  
006610 062716 000002  
006614  
006614 012727 000001  
006620 000000  
006622 013727 002116  
006626 000000  
006630 005367 177772  
006634 001375  
006636 005367 177756  
006642 001367  
006644 005300  
006646 001362  
006650 000207

```

*****
*****
SUBROUTINE $DLAY
FUNCTION - TO SAVE PROGRAM SPACE BY USING ONLY 1
          EXPANSION OF THE SUPERVISOR MACRO "DELAY"

CALLING FORMAT:      JSR      PC,$DLAY
                      .WORD    #

ENTRY CONDITIONS -   @ (SP) = # OF DELAY LOOPS TO USE.

EXIT CONDITIONS -

CALLED BY           - TESTS  2, 5-9, 12, 13

REGISTER R0 RESTORED

*****
*****
$DLAY:
MOV      @ (SP),R0      ;GET THE # OF DELAY LOOPS
ADD      #2,(SP)        ;UPDATE THE PC
10$:
DELAY    1              ;1 DELAY LOOP

MOV      #1,(PC)+
        .WORD    0
MOV      L$DLY,(PC)+
        .WORD    0
DEC      -6(PC)
BNE      .-4
DEC      -22(PC)
BNE      .-20

DEC      R0              ;DECREMENT VARIABLE LOOP COUNTER
BNE      10$            ;LOOP UNTIL DONE
RETURN

```

```

1      .SBTTL GLOBAL ERROR REPORT REPORT SECTION
2      :///////////////////////////////////////////////////
3      :// THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
4      :// THAT ARE USED IN MORE THAN ONE TEST.
5      :///////////////////////////////////////////////////
6      .EVEN
7
8      006652      BGNMSG  ERRG1
9      006652      PRINTB  #FMG3,SUBRPC      ;PC THAT SUBROUTINE WAS CALLED.      ERRG1::
10     006652      013746  002416      MOV      SUBRPC,-(SP)
11     006656      012746  011260      MOV      #FMG3,-(SP)
12     006662      012746  000002      MOV      #2,-(SP)
13     006666      010600      MOV      SP,R0
14     006670      104414      TRAP     C$PNTB
15     006672      062706  000006      ADD      #6,SP
16
17     006676      ENDMSG
18     006676      104423      L10001:  TRAP     C$MSG
19
20     006700      BGNMSG  ERRG2
21     006700      TST      SUBRPC      ;IS THE ERROR IN A SUBROUTINE?      ERRG2::
22     006704      005737  002416      BEQ      10$      ;IF NOT, DON'T PRINT SUBR. PC
23     006706      001412      PRINTB  #FMG3,SUBRPC      ;PC THAT SUBROUTINE WAS CALLED.
24     006706      013746  002416      MOV      SUBRPC,-(SP)
25     006712      012746  011260      MOV      #FMG3,-(SP)
26     006716      012746  000002      MOV      #2,-(SP)
27     006722      010600      MOV      SP,R0
28     006724      104414      TRAP     C$PNTB
29     006726      062706  000006      ADD      #6,SP
30
31     006732      10$:
32     006732      PRINTB  #FMG1,@CSR0,@CSR2 ;PRINT CSR0 AND CSR2 CONTENTS.
33     006732      017746  173332      MOV      @CSR2,-(SP)
34     006736      017746  173324      MOV      @CSR0,-(SP)
35     006742      012746  011166      MOV      #FMG1,-(SP)
36     006746      012746  000003      MOV      #3,-(SP)
37     006752      010600      MOV      SP,R0
38     006754      104414      TRAP     C$PNTB
39     006756      062706  000010      ADD      #10,SP
40
41     006762      PRINTB  #FMG2,@CSR4,@CSR6 ;PRINT CSR4 AND CSR2 CONTENTS.
42     006762      017746  173306      MOV      @CSR6,-(SP)
43     006766      017746  173300      MOV      @CSR4,-(SP)
44     006772      012746  011223      MOV      #FMG2,-(SP)
45     006776      012746  000003      MOV      #3,-(SP)
46     007002      010600      MOV      SP,R0
47     007004      104414      TRAP     C$PNTB
48     007006      062706  000010      ADD      #10,SP
49
50     007012      ENDMSG
51     007012      104423      L10002:  TRAP     C$MSG
52
53     007014      BGNMSG  ERRG3
54     007014      PRINTB  #FMG3,SUBRPC      ;PC THAT SUBROUTINE WAS CALLED.      ERRG3::
55     007014      013746  002416      MOV      SUBRPC,-(SP)

```

007020	012746	011260				MOV	#FMG3,-(SP)
007024	012746	000002				MOV	#2,-(SP)
007030	010600					MOV	SP,R0
007032	104414					TRAP	C\$PNTB
007034	062706	000006				ADD	#6,SP
24 007040			PRINTB	#FMG8,<B,@R1>,<B,@R2>			
007040	005046					CLR	-(SP)
007042	151216					BISB	@R2,(SP)
007044	005046					CLR	-(SP)
007046	151116					BISB	@R1,(SP)
007050	012746	011457				MOV	#FMG8,-(SP)
007054	012746	000003				MOV	#3,-(SP)
007060	010600					MOV	SP,R0
007062	104414					TRAP	C\$PNTB
007064	062706	000010				ADD	#10,SP
25 007070			ENDMSG				
007070					L10003:	TRAP	C\$MSG
007070	104423						
26							
27							
28 007072			BGNMSG	ERRG4			
007072					ERRG4::		
29 007072			PRINTB	#FMG4	;PRINT HEADER		
007072	012746	011332				MOV	#FMG4,-(SP)
007076	012746	000001				MOV	#1,-(SP)
007102	010600					MOV	SP,R0
007104	104414					TRAP	C\$PNTB
007106	062706	000004				ADD	#4,SP
30 007112			PRINTB	#FMG7,CSRO,<B,@CSRO>	;PRINT THE LOW BYTE OF CSRO		
007112	005046					CLR	-(SP)
007114	157716	173146				BISB	@CSRO,(SP)
007120	013746	002266				MOV	CSRO,-(SP)
007124	012746	011413				MOV	#FMG7,-(SP)
007130	012746	000003				MOV	#3,-(SP)
007134	010600					MOV	SP,R0
007136	104414					TRAP	C\$PNTB
007140	062706	000010				ADD	#10,SP
31 007144			PRINTB	#FMG5,<B,R1>	;PRINT EXPECTED CONIENTS		
007144	005046					CLR	-(SP)
007146	150116					BISB	R1,(SP)
007150	012746	011377				MOV	#FMG5,-(SP)
007154	012746	000002				MOV	#2,-(SP)
007160	010600					MOV	SP,R0
007162	104414					TRAP	C\$PNTB
007164	062706	000006				ADD	#6,SP
32 007170			ENDMSG				
007170					L10004:	TRAP	C\$MSG
007170	104423						
33							
34							
35 007172			BGNMSG	ERRG7			
007172					ERRG7::		
36 007172			PRINTB	#FMG4	;PRINT HEADER		
007172	012746	011332				MOV	#FMG4,-(SP)
007176	012746	000001				MOV	#1,-(SP)
007202	010600					MOV	SP,R0
007204	104414					TRAP	C\$PNTB

37	007206	062706	000004					ADD #4,SP
	007212			PRINTB #FMG10,CSR4,<B,@CSR4>	;PRINT THE LOW BYTE OF	CSR4		CLR -(SP)
	007212	005046						BISB @CSR4,(SP)
	007214	157716	173052					MOV CSR4, -(SP)
	007220	013746	002272					MOV #FMG10, -(SP)
	007224	012746	011574					MOV #3, -(SP)
	007230	012746	000003					MOV SP, R0
	007234	010600						TRAP C\$PNTB
	007236	104414						ADD #10,SP
38	007240	062706	000010					
	007244			PRINTB #FMG5,<B,R1>	;PRINT EXPECTED CONTENTS			CLR -(SP)
	007244	005046						BISB R1,(SP)
	007246	150116						MOV #FMG5, -(SP)
	007250	012746	011377					MOV #2, -(SP)
	007254	012746	000002					MOV SP, R0
	007260	010600						TRAP C\$PNTB
	007262	104414						ADD #6,SP
	007264	062706	000006					
39	007270			ENDMSG				
	007270							
	007270	104423						L10005: TRAP C\$MSG
40								
41	007272			BGNMSG ERRG8				
	007272							ERRG8::
42	007272			PRINTB #FMG4	;PRINT HEADER			
	007272	012746	011332					MOV #FMG4, -(SP)
	007276	012746	000001					MOV #1, -(SP)
	007302	010600						MOV SP, R0
	007304	104414						TRAP C\$PNTB
	007306	062706	000004					ADD #4,SP
43	007312			PRINTB #FMG11,CSR5,<B,@PCR>	;PRINT THE HIGH BYTE OF	CSR4		CLR -(SP)
	007312	005046						BISB @PCR,(SP)
	007314	157716	172762					MOV CSR5, -(SP)
	007320	013746	002302					MOV #FMG11, -(SP)
	007324	012746	011640					MOV #3, -(SP)
	007330	012746	000003					MOV SP, R0
	007334	010600						TRAP C\$PNTB
	007336	104414						ADD #10,SP
	007340	062706	000010					
44	007344			PRINTB #FMG5,<B,R1>	;PRINT EXPECTED CONTENTS			CLR -(SP)
	007344	005046						BISB R1,(SP)
	007346	150116						MOV #FMG5, -(SP)
	007350	012746	011377					MOV #2, -(SP)
	007354	012746	000002					MOV SP, R0
	007360	010600						TRAP C\$PNTB
	007362	104414						ADD #6,SP
	007364	062706	000006					
45	007370			ENDMSG				
	007370							
	007370	104423						L10006: TRAP C\$MSG
46								
47	007372			BGNMSG ERRG9				
	007372							ERRG9::
48	007372			PRINTB #FMG4	;PRINT HEADER			
	007372	012746	011332					MOV #FMG4, -(SP)
	007376	012746	000001					MOV #1, -(SP)
	007402	010600						MOV SP, R0

007404	104414				TRAP	C\$PNTB
007406	062706	000004			ADD	#4,SP
49 007412			PRINTB	#FMG12,CSR6,<B,@CSR6>	;PRINT THE LOW BYTE OF CSR6	
007412	005046				CLR	-(SP)
007414	157716	172654			BISB	@CSR6,(SP)
007420	013746	002274			MOV	CSR6, -(SP)
007424	012746	011704			MOV	#FMG12, -(SP)
007430	012746	000003			MOV	#3, -(SP)
007434	010600				MOV	SP,RO
007436	104414				TRAP	C\$PNTB
007440	062706	000010			ADD	#10,SP
50 007444			PRINTB	#FMG5,<B,R1>	;PRINT EXPECTED CONTENTS	
007444	005046				CLR	-(SP)
007446	150116				BISB	R1,(SP)
007450	012746	011377			MOV	#FMG5, -(SP)
007454	012746	000002			MOV	#2, -(SP)
007460	010600				MOV	SP,RO
007462	104414				TRAP	C\$PNTB
007464	062706	000006			ADD	#6,SP
51 007470			ENDMSG			
007470						
007470	104423				L10007:	TRAP C\$MSG
52						
53 007472			BGNMSG	ERRG10		
007472					ERRG10::	
54 007472			PRINTB	#FMG4	;PRINT HEADER	
007472	012746	011332			MOV	#FMG4, -(SP)
007476	012746	000001			MOV	#1, -(SP)
007502	010600				MOV	SP,RO
007504	104414				TRAP	C\$PNTB
007506	062706	000004			ADD	#4,SP
55 007512			PRINTB	#FMG13,CSR7,<B,@CSR7>	;PRINT THE HIGH BYTE OF CSR6	
007512	005046				CLR	-(SP)
007514	157716	172564			BISB	@CSR7,(SP)
007520	013746	002304			MOV	CSR7, -(SP)
007524	012746	011750			MOV	#FMG13, -(SP)
007530	012746	000003			MOV	#3, -(SP)
007534	010600				MOV	SP,RO
007536	104414				TRAP	C\$PNTB
007540	062706	000010			ADD	#10,SP
56 007544			PRINTB	#FMG5,<B,R1>	;PRINT EXPECTED CONTENTS	
007544	005046				CLR	-(SP)
007546	150116				BISB	R1,(SP)
007550	012746	011377			MOV	#FMG5, -(SP)
007554	012746	000002			MOV	#2, -(SP)
007560	010600				MOV	SP,RO
007562	104414				TRAP	C\$PNTB
007564	062706	000006			ADD	#6,SP
57 007570			ENDMSG			
007570						
007570	104423				L10010:	TRAP C\$MSG
58						
59						
60 007572			BGNMSG	ERRG11		
007572					ERRG11::	
61 007572	005737	002416	TST	SUBRPC	;WAS THE RESET ERROR FOUND IN THE SUB	
62 007576	001412		BEQ	5\$	;IF NOT SKIP	

63	007600		PRINTB #FMG23,SUBRPC	;PRINT WHERE CALLED	MOV SUBRPC,-(SP)
	007600	013746 002416			MOV #FMG23,-(SP)
	007604	012746 012536			MOV #2,-(SP)
	007610	012746 000002			MOV SP,R0
	007614	010600			TRAP C\$PNTB
	007616	104414			ADD #6,SP
	007620	062706 000006			
64	007624		5\$: PRINTB #FMG4	;PRINT HEADER	MOV #FMG4,-(SP)
65	007624				MOV #1,-(SP)
	007624	012746 011332			MOV SP,R0
	007630	012746 000001			TRAP C\$PNTB
	007634	010600			ADD #4,SP
	007636	104414			
	007640	062706 000004			
66	007644		PRINTB #FMG7,CSRO,<B,@CSRO>	;PRINT THE LOW BYTE OF CSRO	CLR -(SP)
	007644	005046			BISB @CSRO,(SP)
	007646	157716 172414			MOV CSRO,-(SP)
	007652	013746 002266			MOV #FMG7,-(SP)
	007656	012746 011413			MOV #3,-(SP)
	007662	012746 000003			MOV SP,R0
	007666	010600			TRAP C\$PNTB
	007670	104414			ADD #10,SP
	007672	062706 000010			
67	007676		PRINTB #FMG5,#0	;PRINT EXPECTED CONTENTS	MOV #0,-(SP)
	007676	012746 000000			MOV #FMG5,-(SP)
	007702	012746 011377			MOV #2,-(SP)
	007706	012746 000002			MOV SP,R0
	007712	010600			TRAP C\$PNTB
	007714	104414			ADD #6,SP
	007716	062706 000006			
68	007722		PRINTB #FMG10,CSR4,<B,@CSR4>	;PRINT THE LOW BYTE OF CSR4	CLR -(SP)
	007722	005046			BISB @CSR4,(SP)
	007724	157716 172342			MOV CSR4,-(SP)
	007730	013746 002272			MOV #FMG10,-(SP)
	007734	012746 011574			MOV #3,-(SP)
	007740	012746 000003			MOV SP,R0
	007744	010600			TRAP C\$PNTB
	007746	104414			ADD #10,SP
	007750	062706 000010			
69	007754		PRINTB #FMG5,#TBE	;PRINT EXPECTED CONTENTS	MOV #TBE,-(SP)
	007754	012746 000004			MOV #FMG5,-(SP)
	007760	012746 011377			MOV #2,-(SP)
	007764	012746 000002			MOV SP,R0
	007770	010600			TRAP C\$PNTB
	007772	104414			ADD #6,SP
	007774	062706 000006			
70	010000		PRINTB #FMG11,PCR,<B,@PCR>	;PRINT THE HIGH BYTE OF CSR4	CLR -(SP)
	010000	005046			BISB @PCR,(SP)
	010002	157716 172274			MOV PCR,-(SP)
	010006	013746 002302			MOV #FMG11,-(SP)
	010012	012746 011640			MOV #3,-(SP)
	010016	012746 000003			MOV SP,R0
	010022	010600			TRAP C\$PNTB
	010024	104414			ADD #10,SP
	010026	062706 000010			
71	010032		PRINTB #FMG5,#0	;PRINT EXPECTED CONTENTS	MOV #0,-(SP)
	010032	012746 000000			



010036	012746	011377				MOV	#FMG5,-(SP)
010042	012746	000002				MOV	#2,-(SP)
010046	010600					MOV	SP,R0
010050	104414					TRAP	C\$PNTB
010052	062706	000006				ADD	#6,SP
72 010056			PRINTB	#FMG12,CSR6,<B,@CSR6>	;PRINT THE LOW BYTE OF CSR6	CLR	-(SP)
010056	005046					BISB	@CSR6,(SP)
010060	157716	172210				MOV	CSR6,-(SP)
010064	013746	002274				MOV	#FMG12,-(SP)
010070	012746	011704				MOV	#3,-(SP)
010074	012746	000003				MOV	SP,R0
010100	010600					TRAP	C\$PNTB
010102	104414					ADD	#10,SP
010104	062706	000010					
73 010110			PRINTB	#FMG5,#0	;PRINT EXPECTED CONTENTS	MOV	#0,-(SP)
010110	012746	000000				MOV	#FMG5,-(SP)
010114	012746	011377				MOV	#2,-(SP)
010120	012746	000002				MOV	SP,R0
010124	010600					TRAP	C\$PNTB
010126	104414					ADD	#6,SP
010130	062706	000006					
74 010134			PRINTB	#FMG13,CSR7,<B,@CSR7>	;PRINT THE HIGH BYTE OF CSR6	CLR	-(SP)
010134	005046					BISB	@CSR7,(SP)
010136	157716	172142				MOV	CSR7,-(SP)
010142	013746	002304				MOV	#FMG13,-(SP)
010146	012746	011750				MOV	#3,-(SP)
010152	012746	000003				MOV	SP,R0
010156	010600					TRAP	C\$PNTB
010160	104414					ADD	#10,SP
010162	062706	000010					
75 010166			PRINTB	#FMG5,#0	;PRINT EXPECTED CONTENTS	MOV	#0,-(SP)
010166	012746	000000				MOV	#FMG5,-(SP)
010172	012746	011377				MOV	#2,-(SP)
010176	012746	000002				MOV	SP,R0
010202	010600					TRAP	C\$PNTB
010204	104414					ADD	#6,SP
010206	062706	000006					
76 010212			ENDMSG				
010212							
010212	104423					L10011:	TRAP C\$MSG
77 010214							
78 010214			BGNMSG	ERRG12			
010214					ERRG12::		
79 010214			PRINTB	#FMG3,SUBRPC	;PC THAT SUBROUTINE WAS CALLED.	MOV	SUBRPC,-(SP)
010214	013746	002416				MOV	#FMG3,-(SP)
010220	012746	011260				MOV	#2,-(SP)
010224	012746	000002				MOV	SP,R0
010230	010600					TRAP	C\$PNTB
010232	104414					ADD	#6,SP
010234	062706	000006					
80 010240			PRINTB	#FMG14,REG,R1,R2	;PRINT TIME OUT ERROR	MOV	R2,-(SP)
010240	010246					MOV	R1,-(SP)
010242	010146					MOV	REG,-(SP)
010244	013746	002374				MOV	#FMG14,-(SP)
010250	012746	012014				MOV	#4,-(SP)
010254	012746	000004				MOV	SP,R0
010260	010600						

	010262	104414					TRAP	C\$PNTB
	010264	062706	000012				ADD	#12,SP
81	010270			ENDMSG				
	010270					L10012:		
	010270	104423					TRAP	C\$MSG
82								
83	010272			BGNMSG	ERRG13			
	010272						ERRG13::	
84	010272	032777	000004	171766	BIT	#RTS,@RXCSR		:IS RTS SET
85	010300	001413			BEQ	5\$		
86	010302	012701	030004		MOV	#RTS!CTS:RR,R1		:SET UP THE EXPECTED RESULTS
87	010306				PRINTB	#FMG17		
	010306	012746	012144				MOV	#FMG17,-(SP)
	010312	012746	000001				MOV	#1,-(SP)
	010316	010600					MOV	SP,R0
	010320	104414					TRAP	C\$PNTB
	010322	062706	000004				ADD	#4,SP
88	010326	000544			BR	20\$		
89	010330			5\$:				
90	010330	032777	000002	171730	BIT	#DTR,@RXCSR		:IS DTR SET?
91	010336	001413			BEQ	10\$		:BR IF NOT
92	010340	012701	040002		MOV	#DTR!IC,R1		:SET UP THE EXPECTED RESULTS
93	010344				PRINTB	#FMG18		
	010344	012746	012221				MOV	#FMG18,-(SP)
	010350	012746	000001				MOV	#1,-(SP)
	010354	010600					MOV	SP,R0
	010356	104414					TRAP	C\$PNTB
	010360	062706	000004				ADD	#4,SP
94	010364	000525			BR	20\$		
95	010366			10\$:				
96	010366	032777	000010	171672	BIT	#LL,@RXCSR		:IS LOCAL LOOP SET
97	010374	001433			BEQ	15\$		
98	010376	012701	001010		MOV	#LL!DM,R1		:SET UP THE EXPECTED RESULTS
99	010402				PRINTB	#FMG19		
	010402	012746	012270				MOV	#FMG19,-(SP)
	010406	012746	000001				MOV	#1,-(SP)
	010412	010600					MOV	SP,R0
	010414	104414					TRAP	C\$PNTB
	010416	062706	000004				ADD	#4,SP
100	010422				PRINTB	#FMG21		
	010422	012746	012422				MOV	#FMG21,-(SP)
	010426	012746	000001				MOV	#1,-(SP)
	010432	010600					MOV	SP,R0
	010434	104414					TRAP	C\$PNTB
	010436	062706	000004				ADD	#4,SP
101	010442				PRINTB	#FMG29		
	010442	012746	013221				MOV	#FMG29,-(SP)
	010446	012746	000001				MOV	#1,-(SP)
	010452	010600					MOV	SP,R0
	010454	104414					TRAP	C\$PNTB
	010456	062706	000004				ADD	#4,SP
102	010462	000466			BR	20\$		
103								
104	010464			15\$:				
105	010464	032777	000001	171574	BIT	#SF,@RXCSR		:IS SEL FREQ SET?
106	010472	001520			BEQ	25\$		:IF NONE OF THOSE BITS SET - ERROR
107	010474	012701	000040		MOV	#SQ,R1		:SET UP THE EXPECTED RESULTS

108	010500		PRINTB #FMG20		MOV #FMG20,-(SP)
	010500	012746 012335			MOV #1,-(SP)
	010504	012746 000001			MOV SP,R0
	010510	010600			TRAP C\$PNTB
	010512	104414			ADD #4,SP
	010514	062706 000004			
109	010520		PRINTB #FMG21		MOV #FMG21,-(SP)
	010520	012746 012422			MOV #1,-(SP)
	010524	012746 000001			MOV SP,R0
	010530	010600			TRAP C\$PNTB
	010532	104414			ADD #4,SP
	010534	062706 000004			
110	010540		PRINTB #FMG4	;PRINT HEADER	MOV #FMG4,-(SP)
	010540	012746 011332			MOV #1,-(SP)
	010544	012746 000001			MOV SP,R0
	010550	010600			TRAP C\$PNTB
	010552	104414			ADD #4,SP
	010554	062706 000004			
111	010560		PRINTB #FMG10,CSR4,<B,@CSR4>	;PRINT THE LOW BYTE OF CSR4	CLR -(SP)
	010560	005046			BISB @CSR4,(SP)
	010562	157716 171504			MOV CSR4,-(SP)
	010566	013746 002272			MOV #FMG10,-(SP)
	010572	012746 011574			MOV #3,-(SP)
	010576	012746 000003			MOV SP,R0
	010602	010600			TRAP C\$PNTB
	010604	104414			ADD #10,SP
	010606	062706 0000'0			
112	010612		PRINTB #FMG5,<B,R1>	;PRINT EXPECTED CONTENTS	CLR -(SP)
	010612	005046			BISB R1,(SP)
	010614	150116			MOV #FMG5,-(SP)
	010616	012746 011377			MOV #2,-(SP)
	010622	012746 000002			MOV SP,R0
	010626	010600			TRAP C\$PNTB
	010630	104414			ADD #6,SP
	010632	062706 000006			
113	010636	000447	BR 30\$		
114					
115	010640		20\$:		
116	010640		PRINTB #FMG4	;PRINT HEADER	MOV #FMG4,-(SP)
	010640	012746 011332			MOV #1,-(SP)
	010644	012746 000001			MOV SP,R0
	010650	010600			TRAP C\$PNTB
	010652	104414			ADD #4,SP
	010654	062706 000004			
117	010660		PRINTB #FMG15,CSRO,@CSRO	;PRINT THE LOW BYTE OF CSRO	MOV @CSRO,-(SP)
	010660	017746 171402			MOV CSRO,-(SP)
	010664	013746 002266			MOV #FMG15,-(SP)
	010670	012746 012071			MOV #3,-(SP)
	010674	012746 000003			MOV SP,R0
	010700	010600			TRAP C\$PNTB
	010702	104414			ADD #10,SP
	010704	062706 000010			
118	010710		PRINTB #FMG16,R1	;PRINT EXPECTED CONTENTS	MOV R1,-(SP)
	010710	010146			MOV #FMG16,-(SP)
	010712	012746 012133			MOV #2,-(SP)
	010716	012746 000002			MOV SP,R0
	010722	010600			

Line	Address	Hex	Dec	Label	Instruction	Comment	Register	Value
119	010724	104414	000006					
120	010726	062706	000006					
121	010732	000411		25\$:	BR 30\$			
121	010734				PRINTB #FMG22,R2	;PRINT BIT THAT ISN'T WRITTEN.		
121	010734	010246					MOV	R2,-(SP)
121	010736	012746	012471				MOV	#FMG22,-(SP)
121	010742	012746	000002				MOV	#2,-(SP)
121	010746	010600					MOV	SP,R0
121	010750	104414					TRAP	C\$PNTB
121	010752	062706	000006				ADD	#6,SP
122	010756			30\$:				
123	010756			ENDMSG				
123	010756							
123	010756	104423						
124								
125	010760			BGNMSG	ERRG14			
125	010760							
126	010760	005737	002416		TST SUBRPC	ERRG'4::		
127	010764	001412			BEQ 10\$	;IS THE ERROR IN A SUBROUTINE?		
128	010766				PRINTB #FMG3,SUBRPC	;IF NOT, DON'T PRINT SUBR. PC		
128	010766	013746	002416			;PC THAT SUBROUTINE WAS CALLED.		
128	010772	012746	011260				MOV	SUBRPC,-(SP)
128	010776	012746	000002				MOV	#FMG3,-(SP)
128	011002	010600					MOV	#2,-(SP)
128	011004	104414					MOV	SP,R0
128	011006	062706	000006				TRAP	C\$PNTB
129	011012			10\$:			ADD	#6,SP
130	011012				PRINTB #FMG24,XMITD,RCGUNT	;PRINT CHARACTERS XMITTED AND RCVD.		
130	011012	013746	002500				MOV	RCOUNT,-(SP)
130	011016	013746	002476				MOV	XMITD,-(SP)
130	011022	012746	012605				MOV	#FMG24,-(SP)
130	011026	012746	000003				MOV	#3,-(SP)
130	011032	010600					MOV	SP,R0
130	011034	104414					TRAP	C\$PNTB
130	011036	062706	000010				ADD	#10,SP
131	011042			ENDMSG				
131	011042							
131	011042	104423						
132								
133	011044			BGNMSG	ERRG15			
133	011044							
134	011044				PRINTB #FMG25,R2	;PRINT BIT THAT ISN'T CLEARED.		
134	011044	010246					MOV	R2,-(SP)
134	011046	012746	012652				MOV	#FMG25,-(SP)
134	011052	012746	000002				MOV	#2,-(SP)
134	011056	010600					MOV	SP,R0
134	011060	104414					TRAP	C\$PNTB
134	011062	062706	000006				ADD	#6,SP
135	011066			ENDMSG				
135	011066							
135	011066	104423						
136								
137	011070	045	101	FMG0:</				

	011104	122	125	116	
	011107	040	124	105	
	011112	123	124	040	
	011115	045	104	062	
	011120	045	101	040	
	011123	117	116	040	
	011126	125	116	111	
	011131	124	040	045	
	011134	104	062	045	
	011137	101	040	127	
	011142	111	124	110	
	011145	117	125	124	
	011150	040	124	125	
	011153	122	116	101	
	011156	122	117	125	
	011161	116	104	045	
	011164	116	000		
138	011166	045	101	122	FMG1: .ASCIIZ /%ARXCSR: %06%N%ARDSR : %06%N/
	011171	130	103	123	
	011174	122	072	040	
	011177	045	117	066	
	011202	045	116	045	
	011205	101	122	104	
	011210	123	122	040	
	011213	072	040	045	
	011216	117	066	045	
	011221	116	000		
139	011223	045	101	120	FMG2: .ASCIIZ /%APCSCR: %06%N%ATDSR : %06%N/
	011226	103	123	103	
	011231	122	072	040	
	011234	045	117	066	
	011237	045	116	045	
	011242	101	124	104	
	011245	123	122	040	
	011250	072	040	045	
	011253	117	066	045	
	011256	116	000		
140	011260	045	101	105	FMG3: .ASCIIZ /%AERROR IN SUBROUTINE CALLED AT PC: %06%N/
	011263	122	122	117	
	011266	122	040	111	
	011271	116	040	123	
	011274	125	102	122	
	011277	117	125	124	
	011302	111	116	105	
	011305	040	103	101	
	011310	114	114	105	
	011313	104	040	101	
	011316	124	040	120	
	011321	103	072	040	
	011324	045	117	066	
	011327	045	116	000	
141	011332	045	123	071	FMG4: .ASCIIZ /%S9%S9%S9%S3%AFOUND: %S2%AEXPECTED: %N/
	011335	045	123	071	
	011340	045	123	071	
	011343	045	123	063	
	011346	045	101	106	
	011351	117	125	116	

	011354	104	072	045	
	011357	123	062	045	
	011362	101	105	130	
	011365	120	105	103	
	011370	124	105	104	
	011373	072	045	116	
	011376	000			
142	011377	045	123	067	FMG5: .ASCIIZ /%S7%03%N/
	011402	045	117	063	
	011405	045	116	000	
143	011410	045	116	000	FMG6: .ASCIIZ /%N/
144	011413	045	101	122	FMG7: .ASCIIZ /%ARXCSR = %06%A (EXTERNAL): %03/
	011416	130	103	123	
	011421	122	040	040	
	011424	040	040	075	
	011427	040	045	117	
	011432	066	045	101	
	011435	040	050	105	
	011440	130	124	105	
	011443	122	116	101	
	011446	114	051	072	
	011451	040	040	045	
	011454	117	063	000	
145	011457	045	101	130	FMG8: .ASCIIZ /%AXMIT DATA: %03%A RCV DATA: %03%N/
	011462	115	111	124	
	011465	040	104	101	
	011470	124	101	072	
	011473	040	045	117	
	011476	063	045	101	
	011501	040	122	103	
	011504	126	040	104	
	011507	101	124	101	
	011512	072	040	045	
	011515	117	063	045	
	011520	116	000		
146	011522	045	116	045	FMG9: .ASCIIZ /%N%A** CHECK -V FROM THE CHARGE PUMP **%N/
	011525	101	052	052	
	011530	040	103	110	
	011533	105	103	113	
	011536	040	055	126	
	011541	040	106	122	
	011544	117	115	040	
	011547	124	110	105	
	011552	040	103	110	
	011555	101	122	107	
	011560	105	040	120	
	011563	125	115	120	
	011566	040	052	052	
	011571	045	116	000	
147	011574	045	101	120	FMG10: .ASCIIZ /%APCSCR = %06%A (EXTERNAL): %03/
	011577	103	123	103	
	011602	122	040	040	
	011605	040	040	075	
	011610	040	045	117	
	011613	066	045	101	
	011616	040	050	105	
	011621	130	124	105	

	011624	122	116	101	
	011627	114	051	072	
	011632	040	040	045	
	011635	117	063	000	
148	011640	045	101	120	FMG11: .ASCIIZ /%APCR = %06%A (USYNRT R7): %03/
	011643	103	122	040	
	011646	040	040	040	
	011651	040	040	075	
	011654	040	045	117	
	011657	066	045	101	
	011662	040	050	125	
	011665	123	131	116	
	011670	122	124	040	
	011673	122	067	051	
	011676	072	040	045	
	011701	117	063	000	
149	011704	045	101	124	FMG12: .ASCIIZ /%AT. DATA = %06%A (USYNRT R2): %03/
	011707	056	040	104	
	011712	101	124	101	
	011715	040	040	075	
	011720	040	045	117	
	011723	066	045	101	
	011726	040	050	125	
	011731	123	131	116	
	011734	122	124	040	
	011737	122	062	051	
	011742	072	040	045	
	011745	117	063	000	
150	011750	045	101	124	FMG13: .ASCIIZ /%AT. STATUS= %06%A (USYNRT R3): %03/
	011753	056	040	123	
	011756	124	101	124	
	011761	125	123	075	
	011764	040	045	117	
	011767	066	045	101	
	011772	040	050	125	
	011775	123	131	116	
	012000	122	124	040	
	012003	122	063	051	
	012006	072	040	045	
	012011	117	063	000	
151	012014	045	101	103	FMG14: .ASCIIZ /%A CONTENTS OF %06%A = %06%A EXPECTED %06%N/
	012017	117	116	124	
	012022	105	116	124	
	012025	123	040	117	
	012030	106	040	045	
	012033	117	066	045	
	012036	101	040	075	
	012041	040	045	117	
	012044	066	045	101	
	012047	040	040	040	
	012052	105	130	120	
	012055	105	103	124	
	012060	105	104	040	
	012063	045	117	066	
	012066	045	116	000	
152	012071	045	101	122	FMG15: .ASCIIZ /%ARXCSR = %06%A (EXTERNAL): %06/
	012074	130	103	123	

	012077	122	040	040	
	012102	040	075	040	
	012105	045	117	066	
	012110	045	101	040	
	012113	050	105	130	
	012116	124	105	122	
	012121	116	101	114	
	012124	051	072	040	
	012127	045	117	066	
	012132	000			
153	012133	045	123	063	FMG16: .ASCIZ /%S3%06%N/
	012136	045	117	066	
	012141	045	116	000	
154	012144	045	101	122	FMG17: .ASCIZ /%ARTS NOT TURNED AROUND TO CTS AND RR (CD)%N/
	012147	124	123	040	
	012152	116	117	124	
	012155	040	124	125	
	012160	122	116	105	
	012163	104	040	101	
	012166	122	117	125	
	012171	116	104	040	
	012174	124	117	040	
	012177	103	124	123	
	012202	040	101	116	
	012205	104	040	122	
	012210	122	040	050	
	012213	103	104	051	
	012216	045	116	000	
155	012221	045	101	104	FMG18: .ASCIZ /%ADTR NOT TURNED AROUND TO IC (RING)%N/
	012224	124	122	040	
	012227	116	117	124	
	012232	040	124	125	
	012235	122	116	105	
	012240	104	040	101	
	012243	122	117	125	
	012246	116	104	040	
	012251	124	117	040	
	012254	111	103	040	
	012257	050	122	111	
	012262	116	107	051	
	012265	045	116	000	
156	012270	045	101	114	FMG19: .ASCIZ /%ALL NOT TURNED AROUND TO DM (DSR)%N/
	012273	114	040	116	
	012276	117	124	040	
	012301	124	125	122	
	012304	116	105	104	
	012307	040	101	122	
	012312	117	125	116	
	012315	104	040	124	
	012320	117	040	104	
	012323	115	040	050	
	012326	104	123	122	
	012331	051	045	116	
	012334	000			
157	012335	045	101	122	FMG20: .ASCIZ /%ARL NOT TURNED AROUND TO TEST MODE (SIG. QUALITY)%N/
	012340	114	040	116	
	012343	117	124	040	



	012346	124	125	122	
	012351	116	105	104	
	012354	040	101	122	
	012357	117	125	116	
	012362	104	040	124	
	012365	117	040	124	
	012370	105	123	124	
	012373	040	115	117	
	012376	104	105	040	
	012401	050	123	111	
	012404	107	056	040	
	012407	121	125	101	
	012412	114	111	124	
	012415	131	051	045	
	012420	116	000		
158	012422	045	101	103	FMG21: .ASCIIZ /%ACHECK THAT THE JUMPER IS INSTALLED%N/
	012425	110	105	103	
	012430	113	040	124	
	012433	110	101	124	
	012436	040	124	110	
	012441	105	040	112	
	012444	125	115	120	
	012447	105	122	040	
	012452	111	123	040	
	012455	111	116	123	
	012460	124	101	114	
	012463	114	105	104	
	012466	045	116	000	
159	012471	045	101	103	FMG22: .ASCIIZ /%ACAN'T WRITE BIT %06XA INTO RXCSR%N/
	012474	101	116	047	
	012477	124	040	127	
	012502	122	111	124	
	012505	105	040	102	
	012510	111	124	040	
	012513	045	117	066	
	012516	045	101	040	
	012521	111	116	124	
	012524	117	040	122	
	012527	130	103	123	
	012532	122	045	116	
	012535	000			
160	012536	045	101	122	FMG23: .ASCIIZ /%ARESET SUBROUTINE CALLED AT PC: %06%N/
	012541	105	123	105	
	012544	124	040	123	
	012547	125	102	122	
	012552	117	125	124	
	012555	111	116	105	
	012560	040	103	101	
	012563	114	114	105	
	012566	104	040	101	
	012571	124	040	120	
	012574	103	072	040	
	012577	045	117	066	
	012602	045	116	000	
161	012605	045	101	124	FMG24: .ASCIIZ /%ATRANSMITTED: %D2XA RECEIVED: %D2%N/
	012610	122	101	116	
	012613	123	115	111	

	012616	124	124	105	
	012621	104	072	040	
	012624	045	104	062	
	012627	045	101	040	
	012632	122	105	103	
	012635	105	111	126	
	012640	105	104	072	
	012643	040	045	104	
	012646	062	045	116	
	012651	000			
162	012652	045	101	103	FMG25: .ASCIIZ /%ACAN'T CLEAR BIT %06%A IN RXCSR%N/
	012655	101	116	047	
	012660	124	040	103	
	012663	114	105	101	
	012666	122	040	102	
	012671	111	124	040	
	012674	045	117	066	
	012677	045	101	040	
	012702	111	116	040	
	012705	122	130	103	
	012710	123	122	045	
	012713	116	000		
163	012715	045	101	116	FMG26: .ASCIIZ /%ANOTE: DATA SET INTERRUPT MAY BE DISABLED - CHECK JUMPER%N/
	012720	117	124	105	
	012723	072	040	104	
	012726	101	124	101	
	012731	040	123	105	
	012734	124	040	111	
	012737	116	124	105	
	012742	122	122	125	
	012745	120	124	040	
	012750	115	101	131	
	012753	040	102	105	
	012756	040	104	111	
	012761	123	101	102	
	012764	114	105	104	
	012767	040	055	040	
	012772	103	110	105	
	012775	103	113	040	
	013000	112	125	115	
	013003	120	105	122	
	013006	045	116	000	
164	013011	045	101	124	FMG27: .ASCII /%ATEST %D2%A WILL ONLY RUN WITH RS422 ON A ISI-11/
	013014	105	123	124	
	013017	040	045	104	
	013022	062	045	101	
	013025	040	127	111	
	013030	114	114	040	
	013033	117	116	114	
	013036	131	040	122	
	013041	125	116	040	
	013044	127	111	124	
	013047	110	040	122	
	013052	123	064	062	
	013055	062	040	117	
	013060	116	040	101	
	013063	040	114	123	

	013066	111	055	061	
	013071	061			
165	013072	057			.BYTE 57
166	013073	062	063	045	.ASCIIZ /23%N/
	013076	116	000		
167	013100	045	101	111	FMG28: .ASCII /%AIF CPU IS A M7264 WITH MEMORY REFRESH ENABLED, A HIGH/
	013103	106	040	103	
	013106	120	125	040	
	013111	111	123	040	
	013114	101	040	115	
	013117	067	062	066	
	013122	064	040	127	
	013125	111	124	110	
	013130	040	115	105	
	013133	115	117	122	
	013136	131	040	122	
	013141	105	106	122	
	013144	105	123	110	
	013147	040	105	116	
	013152	101	102	114	
	013155	105	104	054	
	013160	040	101	040	
	013163	110	111	107	
	013166	110			
168	013167	045	101	040	.ASCIIZ /%A SPEED TEST CAN'T RUN%N/
	013172	123	120	105	
	013175	105	104	040	
	013200	124	105	123	
	013203	124	040	103	
	013206	101	116	047	
	013211	124	040	122	
	013214	125	116	045	
	013217	116	000		
169	013221	045	101	052	FMG29: .ASCIIZ /%A** IF M8020 JUMPERED FOR RS422, THIS ERROR EXPECTED **%N/
	013224	052	040	111	
	013227	106	040	115	
	013232	070	060	062	
	013235	060	040	112	
	013240	125	115	120	
	013243	105	122	105	
	013246	104	040	106	
	013251	117	122	040	
	013254	122	123	064	
	013257	062	062	054	
	013262	040	124	110	
	013265	111	123	040	
	013270	105	122	122	
	013273	117	122	040	
	013276	105	130	120	
	013301	105	103	124	
	013304	105	104	040	
	013307	052	052	045	
	013312	116	000		
170	013314	045	101	052	FMG30: .ASCIIZ /%A** CHECK BYTE OP SIGNAL - STUCK LOW ?? **%N/
	013317	052	040	103	
	013322	110	105	103	
	013325	113	040	102	

	013330	131	124	105	
	013333	040	117	120	
	013336	040	123	111	
	013341	107	116	101	
	013344	114	040	055	
	013347	040	123	124	
	013352	125	103	113	
	013355	040	114	117	
	013360	127	040	077	
	013363	077	040	052	
	013366	052	045	116	
	013371	000			
171					
172	013372	122	105	123	EMG0: .ASCIIZ /RESET ERROR AFTER BUS RESET (DETECTED ONLY ON 1ST PASS)/
	013375	105	124	040	
	013400	105	122	122	
	013403	117	122	040	
	013406	101	106	124	
	013411	105	122	040	
	013414	102	125	123	
	013417	040	122	105	
	013422	123	105	124	
	013425	040	050	104	
	013430	105	124	105	
	013433	103	124	105	
	013436	104	040	117	
	013441	116	114	131	
	013444	040	117	116	
	013447	040	061	123	
	013452	124	040	120	
	013455	101	123	123	
	013460	051	000		
173	013462	124	111	115	EMG1: .ASCIIZ /TIME OUT/
	013465	105	040	117	
	013470	125	124	000	
174	013473	124	111	115	EMG2: .ASCIIZ /TIME OUT - DURING INTERRUPT EXERCISE/
	013476	105	040	117	
	013501	125	124	040	
	013504	055	040	104	
	013507	125	122	111	
	013512	116	107	040	
	013515	111	116	124	
	013520	105	122	122	
	013523	125	120	124	
	013526	040	105	130	
	013531	105	122	103	
	013534	111	123	105	
	013537	000			
175	013540	122	105	123	EMG3: .ASCIIZ /RESET ERROR/
	013543	105	124	040	
	013546	105	122	122	
	013551	117	122	000	
176	013554	103	123	122	EMG4: .ASCIIZ /CSR READ-WRITE ERROR/
	013557	040	122	105	
	013562	101	104	055	
	013565	127	122	111	
	013570	124	105	040	

	013573	105	122	122	
	013576	117	122	000	
177	013601	125	123	131	EMG5: .ASCIIZ /USYNRT XMIT ACTIVE NOT SET/
	013604	116	122	124	
	013607	040	130	115	
	013612	111	124	040	
	013615	101	103	124	
	013620	111	126	105	
	013623	040	116	117	
	013626	124	040	123	
178	013631	105	124	000	EMG6: .ASCIIZ /USYNRT XMIT ACTIVE NOT CLEAR/
	013634	125	123	131	
	013637	116	122	124	
	013642	040	130	115	
	013645	111	124	040	
	013650	101	103	124	
	013653	111	126	105	
	013656	040	116	117	
	013661	124	040	103	
	013664	114	105	101	
	013667	122	000		
179	013671	124	102	105	EMG7: .ASCIIZ /TBE NOT CLFAR/
	013674	040	116	117	
	013677	124	040	103	
	013702	114	105	101	
	013705	122	000		
180	013707	124	102	105	EMG8: .ASCIIZ /TBE NOT SET/
	013712	040	116	117	
	013715	124	040	123	
	013720	105	124	000	
181	013723	130	115	111	EMG9: .ASCIIZ /XMIT INTERRUPT NOT RECEIVED/
	013726	124	040	111	
	013731	116	124	105	
	013734	122	122	125	
	013737	120	124	040	
	013742	116	117	124	
	013745	040	122	105	
	013750	103	105	111	
	013753	126	105	104	
	013756	000			
182	013757	130	115	111	EMG10: .ASCIIZ /XMIT INTERRUPT RECEIVED WHEN NOT EXPECTED/
	013762	124	040	111	
	013765	116	124	105	
	013770	122	122	125	
	013773	120	124	040	
	013776	122	105	103	
	014001	105	111	126	
	014004	105	104	040	
	014007	127	110	105	
	014012	116	040	116	
	014015	117	124	040	
	014020	105	130	120	
	014023	105	103	124	
	014026	105	104	000	
183	014031	122	105	103	EMG11: .ASCIIZ /RECEIVER NOT DEACTIVATED/
	014034	105	111	126	
	014037	105	122	040	

	014042	116	117	124	
	014045	040	104	105	
	014050	101	103	124	
	014053	111	126	101	
	014056	124	105	104	
	014061	000			
184	014062	122	105	103	EMG12: .ASCIZ /RECEIVER NOT ACTIVE/
	014065	105	111	126	
	014070	105	122	040	
	014073	116	117	124	
	014076	040	101	103	
	014101	124	111	126	
	014104	105	000		
185	014106	122	105	103	EMG13: .ASCIZ /RECEIVER NOT INITIALIZED AFTER RECEIVER DISABLED/
	014111	105	111	126	
	014114	105	122	040	
	014117	116	117	124	
	014122	040	111	116	
	014125	111	124	111	
	014130	101	114	111	
	014133	132	105	104	
	014136	040	101	106	
	014141	124	105	122	
	014144	040	122	105	
	014147	103	105	111	
	014152	126	105	122	
	014155	040	104	111	
	014160	123	101	102	
	014163	114	105	104	
	014166	000			
186	014167	122	105	103	EMG14: .ASCIZ /RECEIVER ACTIVE BEFORE FIRST DATA CHARACTER/
	014172	105	111	126	
	014175	105	122	040	
	014200	101	103	124	
	014203	111	126	105	
	014206	040	102	105	
	014211	106	117	122	
	014214	105	040	106	
	014217	111	122	123	
	014222	124	040	104	
	014225	101	124	101	
	014230	040	103	110	
	014233	101	122	101	
	014236	103	124	105	
	014241	122	000		
187	014243	122	103	126	EMG15: .ASCIZ /RCV INTERRUPT NOT RECEIVED/
	014246	040	111	116	
	014251	124	105	122	
	014254	122	125	120	
	014257	124	040	116	
	014262	117	124	040	
	014265	122	105	103	
	014270	105	111	126	
	014273	105	104	000	
188	014276	122	103	126	EMG16: .ASCIZ /RCV INTERRUPT RECEIVED BEFORE EXPECTED/
	014301	040	111	116	
	014304	124	105	122	

	014307	122	125	120	
	014312	124	040	122	
	014315	105	103	105	
	014320	111	126	105	
	014323	104	040	102	
	014326	105	106	117	
	014331	122	105	040	
	014334	105	130	120	
	014337	105	103	124	
	014342	105	104	000	
189	014345	122	103	126	EMG17: .ASCIZ /RCV END OF MESSAGE NOT RECEIVED/
	014350	040	105	116	
	014353	104	040	117	
	014356	106	040	115	
	014361	105	123	123	
	014364	101	107	105	
	014367	040	116	117	
	014372	124	040	122	
	014375	105	103	105	
	014400	111	126	105	
	014403	104	000		
190	014405	122	103	126	EMG18: .ASCIZ /RCV STATUS NOT CLEARED/
	014410	040	123	124	
	014413	101	124	125	
	014416	123	040	116	
	014421	117	124	040	
	014424	103	114	105	
	014427	101	122	105	
	014432	104	000		
191	014434	122	103	126	EMG19: .ASCIZ /RCV OVERRUN NOT RECEIVED/
	014437	040	117	126	
	014442	105	122	122	
	014445	125	116	040	
	014450	116	117	124	
	014453	040	122	105	
	014456	103	105	111	
	014461	126	105	104	
	014464	000			
192	014465	122	103	126	EMG20: .ASCIZ /RCV ABORT NOT RECEIVED/
	014470	040	101	102	
	014473	117	122	124	
	014476	040	116	117	
	014501	124	040	122	
	014504	105	103	105	
	014507	111	126	105	
	014512	104	000		
193	014514	122	103	126	EMG21: .ASCIZ /RCV STATUS INTERRUPT NOT RECEIVED/
	014517	040	123	124	
	014522	101	124	125	
	014525	123	040	111	
	014530	116	124	105	
	014533	122	122	125	
	014536	120	124	040	
	014541	116	117	124	
	014544	040	122	105	
	014547	103	105	111	
	014552	126	105	104	

	014555	000			
194	014556	115	117	104	EMG22: .ASCIZ /MODEM LOOPBACK ERROR/
	014561	105	115	040	
	014564	114	117	117	
	014567	120	102	101	
	014572	103	113	040	
	014575	105	122	122	
	014600	117	122	000	
195	014603	115	117	104	EMG23: .ASCIZ /MODEM STATUS INTERRUPT RECEIVED WHEN DISABLED/
	014606	105	115	040	
	014611	123	124	101	
	014614	124	125	123	
	014617	040	111	116	
	014622	124	105	122	
	014625	122	125	120	
	014630	124	040	122	
	014633	105	103	105	
	014636	111	126	105	
	014641	104	040	127	
	014644	110	105	116	
	014647	040	104	111	
	014652	123	101	102	
	014655	114	105	104	
	014660	000			
196	014661	115	117	104	FMG24: .ASCIZ /MODEM STATUS INTERRUPT NOT RECEIVED/
	014664	105	115	040	
	014667	123	124	101	
	014672	124	125	123	
	014675	040	111	116	
	014700	124	105	122	
	014703	122	125	120	
	014706	124	040	116	
	014711	117	124	040	
	014714	122	105	103	
	014717	105	111	126	
	014722	105	104	000	
197	014725	103	110	101	EMG25: .ASCIZ /CHARACTER COUNT ERROR/
	014730	122	101	103	
	014733	124	105	122	
	014736	040	103	117	
	014741	125	116	124	
	014744	040	105	122	
	014747	122	117	122	
	014752	000			
198	014753	104	101	124	EMG26: .ASCIZ /DATA ERROR/
	014756	101	040	105	
	014761	122	122	117	
	014764	122	000		
199	014766	130	115	111	EMG30: .ASCIZ /XMIT UNDERRUN/
	014771	124	040	125	
	014774	116	104	105	
	014777	122	122	125	
	015002	116	000		
200	015004	122	105	103	EMG31: .ASCIZ /RECEIVER ERROR/
	015007	105	111	126	
	015012	105	122	040	
	015015	105	122	122	



	015020	117	122	000	
201	015023	101	102	117	EMG32: .ASCIZ /ABORT NOT RECEIVED/
	015026	122	124	040	
	015031	116	117	124	
	015034	040	122	105	
	015037	103	105	111	
	015042	126	105	104	
	015045	000			
202	015046	107	117	040	EMG33: .ASCIZ /GO AHEAD NOT RECEIVED/
	015051	101	110	105	
	015054	101	104	040	
	015057	116	117	124	
	015062	040	122	105	
	015065	103	105	111	
	015070	126	105	104	
	015073	000			
203	015074	101	102	117	EMG34: .ASCIZ /ABORT RECEIVED WHEN NOT EXPECTED/
	015077	122	124	040	
	015102	122	105	103	
	015105	105	111	126	
	015110	105	104	040	
	015113	127	110	105	
	015116	116	040	116	
	015121	117	124	040	
	015124	105	130	120	
	015127	105	103	124	
	015132	105	104	000	
204	015135	101	104	104	EMG35: .ASCIZ /ADDRESS INCORRECTLY RECOGNIZED/
	015140	122	105	123	
	015143	123	040	111	
	015146	116	103	117	
	015151	122	122	105	
	015154	103	124	114	
	015157	131	040	122	
	015162	105	103	117	
	015165	107	116	111	
	015170	132	105	104	
	015173	000			
205	015174	101	123	123	EMG36: .ASCIZ /ASSEMBLED BIT COUNT ERROR/
	015177	105	115	102	
	015202	114	105	104	
	015205	040	102	111	
	015210	124	040	103	
	015213	117	125	116	
	015216	124	040	105	
	015221	122	122	117	
	015224	122	000		
206	015226	103	122	103	EMG37: .ASCIZ /CRC ERROR/
	015231	040	105	122	
	015234	122	117	122	
	015237	000			
207	015240	103	122	103	EMG38: .ASCIZ /CRC ERROR NOT DETECTED/
	015243	040	105	122	
	015246	122	117	122	
	015251	040	116	117	
	015254	124	040	104	
	015257	105	124	105	

	015262	103	124	105	
	015265	104	000		
208	015267	120	101	122	EMG39: .ASCIIZ /PARITY ERROR NOT DETECTED/
	015272	111	124	131	
	015275	040	105	122	
	015300	122	117	122	
	015303	040	116	117	
	015306	124	040	104	
	015311	105	124	105	
	015314	103	124	105	
	015317	104	000		
209	015321	115	125	114	EMG40: .ASCIIZ /MULTIPLE MODEM CONTROL INTERRUPTS/
	015324	124	111	120	
	015327	114	105	040	
	015332	115	117	104	
	015335	105	115	040	
	015340	103	117	116	
	015343	124	122	117	
	015346	114	040	111	
	015351	116	124	105	
	015354	122	122	125	
	015357	120	124	123	
	015362	000			
210					.EVEN
211					

```
1
2
3
4
5
6
7
8 015364          .SBTTL  LOAD DEVICE PROTECTION TABLE
   015364          ;////////////////////////////////////
10 015364 177777   ;// THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
11 015366 177777   ;// PROTECTED FROM TESTING. IF DESIRED.
12 015370 177777   ;////////////////////////////////////
13
14 015372          BGNPROT
15
16
17
18
19
                                L$PROT::
                                .WORD  -1          ;DON'T CHECK CSR ADDRESS
                                .WORD  -1          ;DON'T CHECK MASSBUS UNIT NUMBER
                                .WORD  -1          ;DON'T CHECK DRIVE NUMBER
                                ENDPROT
```

```

1      .SBTTL  INITIALIZE SECTION
2
3      ;////////////////////////////////////
4      ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
5      ;/ AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
6      ;////////////////////////////////////
7
8 015372 BGNINIT
015372
9
10 015372          SETPRI  #PRI07          ;SET DIAGNOSTIC PRIORITY = 7
    015372 012700 000340
    015376 104441
    015376          MOV      #PRI07,RO
    015376          TRAP     C$SPRI
11 015400          MOV      SP,PSTACK      ;STORE BASE LEVEL PROGRAM STACK POINTER
12 015404 005037 002372          CLR      SUBRPC      ;CLEAR STORAGE WORD FOR SUBROUTINE PC CALL
13 015410 005037 002330          CLR      ERROR      ;CLEAR ERROR FLAGS
14
15 015414 005037 002334          CLR      FLAG      ;CLEAR MISC. FLAGS
16 015420 005037 002376          CLR      RFLAG
17 015424 005037 002424          CLR      TFLAG
18 015430 005037 002366          CLR      NXMFLG
19 015434 005037 002316          CLR      ABORT
20 015440 005037 002432          CLR      TOGGLE
21 015444 005037 002370          CLR      OVER
22 015450 005037 002340          CLR      HIGH
23
24 015454 005737 002310          TST      FRSTIM      ;IS THIS THE TIME THROUGH AFTER LOAD?
25 015460 001005          BNE      1$      ;IF NOT - ERROR TRAP VECTOR ALREADY SAVED
26 015462 012737 000001 002310          MOV      #1,FRSTIM      ;FLAG THAT WE'VE BEEN THRU THE 1ST TIME
27 015470 005037 002312          CLR      FRSPAS      ;CLEAR COUNTER FOR # OF PASSES AFTER LOAD
28
29          1$:
30          CLRVEC  #4          ;ENSURE VECTOR 4 IS IN NORMAL STATE.
    015474 012700 000004          MOV      #4,RO
    015500 104436          TRAP     C$CVEC
31
32 015502          READEF  #EF.START      ;IS THIS JUST STARTED?
    015502 012700 000040          MOV      #EF.START,RO
    015506 104447          TRAP     C$REFG
33          BCOMPLETE STARST      ;IF YES - BRANCH.
    015510 103416          BCS      STARST
34          READEF  #EF.RESTART      ;IS THIS A RESTART ?
    015512 012700 000037          MOV      #EF.RESTART,RO
    015516 104447          TRAP     C$REFG
35          BCOMPLETE STARST      ;IF YES - BRANCH.
    015520 103412          BCS      STARST
36          READEF  #EF.NEW      ;IS THIS A NEW PASS?
    015522 012700 000035          MOV      #EF.NEW,RO
    015526 104447          TRAP     C$REFG
37          BCOMPLETE NEWST      ;IF YES - BRANCH
    015530 103410          BCS      NEWST
38          READEF  #EF.CONTINUE      ;IS THIS A CONTINUATION?
    015532 012700 000036          MOV      #EF.CONTINUE,RO
    015536 104447          TRAP     C$REFG
39          BNCOMPLETE GETPRM      ;IF NOT - GET PARAMETERS
    015540 103013          BCC      GETPRM
40 015542 000137 016310          JMP      END      ;OTHERWISE - DON'T INITIALIZE.

```

```

41
42 015546          STARST:
43 015546 005037 002314      CLR      STARES          ;CLEAR THE FLAG TO SHOW START/RESTART.
44
45 015552          NEWST:
46 015552 012737 177777 002354      MOV      #-1,LOGDEV      ;INITIALIZE LOGICAL UNIT NUMBER.
47 015560 005237 002312          INC      FRSPAS          ;INCREMENT # OF PASSES AFTER LOAD.
48 015564 005237 002314          INC      STARES          ;INCREMENT # OF PASSES SINCE START/RESTART.
49 015570          GETPRM:
50 015570 005237 002354          INC      LOGDEV          ;NEXT LOGICAL UNIT TO BE TESTED
51 015574 023737 002354 002012      CMP      LOGDEV,LSUNIT      ;IS THE MAXIMUM UNIT # EXCEEDED?
52 015602 002363          BGE      NEWST          ;IF YES - DO A NEW START
53 015604          GPWARD LOGDEV,R1      ;GET THE P-TABLE POINTER INTO R1
      MOV      LOGDEV,R0
      TRAP     CS$PHRD
      MOV      R0,R1
54 015614          BNCOMPLETE GETPRM      ;IF NOT AVAILABLE, GET THE NEXT ONE
      BCC      GETPRM
      MOV      (R1),R0          ;SAVE THE ADDRESS
55 015616 011100          BIT      #7,R0          ;DOES THIS DEVICE ADDRESS END IN NON-ZERO?
56 015620 032700 000007          BEQ      10$          ;IF NOT - OK (76XXX0)
57 015624 001414          BIC      #7,(R1)        ;MAKE IT 76XXX0
58 015626 042711 000007          PRINTB #FINIT1,(R1),R0 ;INFORM THE USER
59 015632          MOV      R0,-(SP)
      MOV      (R1),-(SP)
      MOV      #FINIT1,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      TRAP     CS$PNTB
      ADD      #10,SP
60 015656          10$:
61 015656 011137 002266          MOV      (R1),CSR0      ;CSR ADDRESS 0 = RECEIVER CSR (RXCSR)
62                                     READ/WRITE
63 015662 013737 002266 002276          MOV      CSR0,CSR1      ;SAVE HIGH BYTE ADDRESS
64 015670 005237 002276          INC      CSR1
65 015674 011137 002270          MOV      (R1),CSR2
66 015700 062737 000002 002270          ADD      #2,CSR2      ;CSR ADDRESS 2 = RECEIVE DATA/STATUS (RDSR)
67                                     READ ONLY
68                                     ;CSR ADDRESS 2 = PARAMETER CONTROL/SYNCH ADDR
69                                     (PCSR) - WRITE ONLY
70 015706 013737 002270 002300          MOV      CSR2,CSR3      ;SAVE HIGH BYTE ADDRESS
71 015714 005237 002300          INC      CSR3
72 015720 011137 002272          MOV      (R1),CSR4
73 015724 062737 000004 002272          ADD      #4,CSR4      ;CSR ADDRESS 4 = TRANSMITTER CSR (TXCSR)
74                                     READ/WRITE
75
76 015732 013737 002272 002302          MOV      CSR4,CSR5      ;CSR ADDRESS 5 = PARAMETER CONTROL REG (PCR)
77                                     READ/WRITE
78 015740 005237 002302          INC      CSR5          ;PCR IS HI BYTE OF TXCSR
79 015744 012137 002274          MOV      (R1)+,CSR6
80 015750 062737 000006 002274          ADD      #6,CSR6      ;CSR ADDRESS 6 = TRANSMIT DATA/STATUS (TDSR)
81                                     READ/WRITE
82 015756 013737 002274 002304          MOV      CSR6,CSR7      ;SAVE HIGH BYTE ADDRESS
83 015764 005237 002304          INC      CSR7
84 015770 011100          MOV      (R1),R0          ;GET VECTOR
85 015772 032700 000007          BIT      #7,R0          ;DOES THIS VECTOR END IN NON-ZERO?
86 015776 001414          BEQ      11$          ;IF NOT - OK (XX0)

```

87	016000	042711	000007		BIC	#7,(R1)	;MAKE IT XX0		
88	016004				PRINTB	#FINIT2,(R1),R0	;INFORM THE USER		
	016004	010046						MOV	R0,-(SP)
	016006	011146						MOV	(R1),-(SP)
	016010	012746	016405					MOV	#FINIT2,-(SP)
	016014	012746	000003					MOV	#3,-(SP)
	016020	010600						MOV	SP,R0
	016022	104414						TRAP	C\$PNTB
	016024	062706	000010					ADD	#10,SP
89	016030			11\$:					
90	016030	011137	002262		MOV	(R1),RCVEC	;RCV. VECTOR		
91	016034	012137	002264		MOV	(R1)+,XMTVEC	;TRANSMIT VECTOR		
92	016040	062737	000004	002264	ADD	#4,XMTVEC	;ADJUST XMIT VECTOR		
93									
94	016046	011137	002306		MOV	(R1),TURN	;TURNAROUND.		
95	016052	012737	000020	002402	MOV	#RXENA,RXINI	;RECEIVER INIT WORD		
96	016060	012737	000020	002436	MOV	#TXENA,TXINI	;TRANSMITTER INIT WORD		
97	016066	005737	002306		TST	TURN	;WHAT WAS THE TURNAROUND		
98	016072	001004			BNE	15\$	;IF ACTUAL TURNAROUND DON'T SET MAINT MODE		
99	016074	052737	000010	002436	BIS	#MM,TXINI	;SET THE MAINT. MODE BIT.		
100	016102	000422			BR	20\$			
101	016104			15\$:					
102	016104	052737	000004	002402	BIS	#RTS,RXINI	;SET RTS FOR TURNAROUND LOOP.		
103	016112	022737	000003	002306	CMP	#3,TURN	;LOCAL LOOPBACK?		
104	016120	001004			BNE	17\$	;IF NOT SKIP.		
105	016122	052737	000012	002402	BIS	#LL!DTR,RXINI	;SET LOCAL LOOP AND DTR.		
106	016130	000407			BR	20\$			
107	016132			17\$:					
108	016132	022737	000004	002306	CMP	#4,TURN	;REMOTE LOOPBACK?		
109	016140	001003			BNE	20\$			
110	016142	052737	000003	002402	BIS	#DTR!RL,RXINI	;SET REMOTE LOOP AND DTR		
111	016150			20\$:					
112	016150	013737	002402	002404	MOV	RXINI,RXINIT	;SAVE RECEIVER INIT WORD		
113	016156	052737	000140	002404	BIS	#RXITEN!DSITEN,RXINIT	;MAKE IT AN INTERRUPT INIT WORD		
114	016164	013737	002436	002440	MOV	TXINI,TXINIT	;SAVE TRANSMITTER INIT WORD		
115	016172	052737	000100	002440	BIS	#TXIE,TXINIT	;MAKE IT AN INTERRUPT INIT WORD		
116	016200	012737	000120	002406	MOV	#RXITEN!RXENA,RXMINI	;RCV INIT FOR MAINT. LOOP.		
117	016206	012737	000130	002442	MOV	#TXIE!TXENA!MM,TXMINI	;TRANS INIT WITH MAINT. LOOP.		
118							;DETERMINE PROCESSOR TYPE		
119	016214				SETVEC	#10,#ILLGL,#PRI07	;SET UP ILLEGAL INSTRUCTION TRAP		
	016214	012746	000340					MOV	#PRI07,-(SP)
	016220	012746	017744					MOV	#ILLGL,-(SP)
	016224	012746	000010					MOV	#10,-(SP)
	016230	012746	000003					MOV	#3,-(SP)
	016234	104437						TRAP	C\$SVEC
	016236	062706	000010					ADD	#10,SP
120	016242	000007			MFPT		;MOVE PROCESSOR TYPE TO R0		
121							;FOR AN LSI 11/23 R0 = 3		
122							;FOR OTHER LSI THIS WILL RESULT IN AN		
123							;ILLEGAL INSTRUCTION (R0=0).		
124	016244	010037	002324		MOV	R0,CPU	;SAVE THE PROCESSOR TYPE		
125	016250				CLRVEC	#10	;RESTORE TRAP TO THE SUPERVISOR		
	016250	012700	000010					MOV	#10,R0
	016254	104436						TRAP	C\$CVEC
126	016256	005737	002324		TST	CPU	;IS THE CPU A LSI11/23 ?		
127	016262	001004			BNE	25\$	;BR IF YES		
128	016264	012737	000020	002430	MOV	#20,TIMER	;SET THE TIMER FOR A LSI11 OR 11/2.		

```

129 016272 000403
130 016274
131 016274 012737 000050 002430
132 016302
133 016302 013737 002430 002412
134 016310
135 016310
    016310
    016310 104411
136 016312 045 101 052
    016315 052 040 127
    016320 101 122 116
    016323 111 116 107
    016326 040 055 040
    016331 127 111 114
    016334 114 040 101
    016337 123 123 125
    016342 115 105 040
    016345 104 120 126
    016350 040 101 104
    016353 104 122 105
    016356 123 123 040
    016361 045 117 066
    016364 045 101 040
    016367 050 116 117
    016372 124 040 045
    016375 117 066 045
    016400 101 051 045
    016403 116 000
137 016405 045 101 052
    016410 052 040 127
    016413 101 122 116
    016416 111 116 107
    016421 040 055 040
    016424 127 111 114
    016427 114 040 101
    016432 123 123 125
    016435 115 105 040
    016440 104 120 126
    016443 040 126 105
    016446 103 124 117
    016451 122 040 040
    016454 045 117 063
    016457 045 101 040
    016462 050 116 117
    016465 124 040 045
    016470 117 063 045
    016473 101 051 045
    016476 116 000

```

```

BR 30$
25$: MOV #50,TIMER ;SET THE TIMER FOR A LSI-11/23.
30$: MOV TIMER,SAVTIM ;STORE THE TIMER VALUE.
END:
ENDINIT

```

L10017: TRAP C\$INIT

FINIT1: .ASCIIZ /%A\*\* WARNING - WILL ASSUME DPV ADDRESS %06XA (NOT %06XA)%N/

FINIT2: .ASCIIZ /%A\*\* WARNING - WILL ASSUME DPV VECTOR %03XA (NOT %03XA)%N/

.EVEN

```

138
139
140
141
142
143

```

```

1      .SBTTL  AUTO DROP UNIT SECTION
2
3      ;////////////////////////////////////
4      ;// THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
5      ;// WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
6      ;////////////////////////////////////
7
8      016500      BGNAUTO
9      016500
10     016500      SETVEC  #4,#NXM,#PRI07 ;SET UP NON -EXISTENT MEMORY TRAP VECTOR.
11     016500      012746 000340      MOV      #PRI07,-(SP)
12     016504      012746 017734      MOV      #NXM,-(SP)
13     016510      012746 000004      MOV      #4,-(SP)
14     016514      012746 000003      MOV      #3,-(SP)
15     016520      104437      TRAP      C$SVEC
16     016522      062706 000010      ADD      #10,SP
17     016526      005037 002366      CLR      NXMFLG      ;CLEAR FLAG USED IN TEST
18     016532      005777 163530      TST      @CSRO      ;REFERENCE MEMORY ADDRESS FOR THE DEVICE
19                                ;TO SEE IF IT EXISTS.
20
21     ;*****
22     ; IF THE DEVICE DOESN'T EXIST THE RESULTANT TRAP TO VECTOR 04 WILL
23     ; CAUSE THE DEVICE TO BE DROPPED (SEE INTERRUPT ROUTINE 'DROPO4').
24     ; OTHERWISE THE MEMORY REFERENCE IS UNEVENTFUL AND THE DEVICE IS READY.
25     ;*****
26     016536      005737 002366      TST      NXMFLG      ;WAS THERE A TRAP?
27     016542      001407      BEQ      10$      ;BR IF NOT
28     016544      013700 002354      DODU      LOGDEV      ;DROP THE DEVICE
29     016550      104451      MOV      LOGDEV,R0
30     016552      104444      TRAP      C$DODU
31     016554      012700 000004      DOCLN      ;CLEAN UP CODE.
32     016560      104436      CLRVEC  #4      ;RETURN VECTOR 04 TO NORMAL STATE
33     016562      104461      MOV      #4,R0
34                                TRAP      C$CVEC
35
36     10$:
37     ENDAUTO
38
39     L10020:
40     TRAP      C$AUTO
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17

.SBTTL CLEANUP CODING SECTION  
:////////////////////  
:// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED AT THE  
:// END OF THE TEST SEQUENCE ON A PARTICULAR UNIT. THIS SECTION IS REQUIRED  
:// EVEN IF IT IS A NULL CLEANUP  
:////////////////////

BGNCLN  
L\$CLEAN::  
TST NXMFLG ;WAS THERE A NXM TRAP  
BNE 10\$ ;IF YES, SKIP RESET  
MOV #RESET,@TXCSR ;RESET THE DPV  
10\$:  
ENDCLN  
L10021: TRAP C\$CLEAN

016564  
016564  
016564 005737 002366  
016570 001003  
016572 012777 000001 163472  
016600  
016600  
016600 104412

[illegible]

CVDPVBO DVP11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 44-1  
GLOBAL INTERRUPT HANDLING ROUTINES

```
57 016722 017737 163342 002400      MOV      @RDSR,RSAVE      ;SAVE RECEIVE DATA AND STATUS.  
58  
59 016730                                ENDSRV  
   016730                                L10022:  
   016730 000002                                RTI  
60
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24 016732
    016732
25
26 016732 017737 163330 002346
27 016740 100040
28
29 016742 032737 000040 002346
30 016750 001434
31 016752 005237 002360
32 016756 022737 0C0011 002360
33 016764 002004
34 016766 042777 000040 163272
35 016774 000422
36 016776
37 016776
38 017000 013705 002360
39 017004 006305
40 017006 013765 002346 002444
41 017014 042765 006760 002444
42 017022 032777 000040 163242
43 017030 001403
44 017032 052765 000040 002444
45 017040
46 017040
47
48 017042
49 017042 032737 002200 002346
50 017050 001444
51 017052 017737 163212 002350
52 017060 032737 000200 002346
53 017066 001404
54 017070 113721 002350
55 017074 005237 002500
56 017100

```

\*\*\*\*\*  
RDATA - INTERRUPT SERVICE ROUTINE  
FUNCTION - GENERAL PURPOSE RECEIVE INTERRUPT ROUTINE  
ENTRY CONDITIONS  
    ECOUNT = # OF CHARACTERS TO BE RECEIVED.  
    R1      = ADDRESS OF BUFFER FOR NEXT CHARACTER  
EXIT CONDITIONS  
    IRXCSR = IMAGE OF RXCSR  
    IRDSR  = IMAGE OF RDSR  
    RCOUNT = COUNT OF CHARACTERS RECEIVED  
    MODE   = PROTOCOL MODE ( 0 = BCP, NON 0 = BOP)  
    MCFLAG = COUNT OF MODEM CONTROL INTERRUPTS RECEIVED  
    MODEM  = ADDRESS OF MODEM CONTROL INTERRUPT TABLE  
    RFLAG  = RECEIVE END FLAG ( 1 = NO ERROR, -1 = ERROR)  
    R1      = INCREMENTED TO NEXT BYTE IN BUFFER.  
USED IN TESTS: 15-28 & 30-40 (CALLED IN SUBROUTINE \$DATA), 41  
\*\*\*\*\*

```

BGNSRV  RDATA
RDATA::
1$:
    MOV    @RXCSR,IRXCSR    ;SAVE THE RXCSR
    BPL    10$              ;IS DATA SET CHANGE? IF NOT SET, BR.
    BIT    #DSITEN,IRXCSR   ;WAS THE DATA SET CHANGE INT. ENABLED?
    BEQ    10$              ;IF NOT - DON'T KEEP TRACK OF THE CHANGES.
    INC    MCFLAG           ;INCR MODEM CONTROL FLAG.
    CMP    #9,MCFLAG        ;WERE TOO MANY INTERRUPTS RECEIVED?
    BGE    1$              ;IF NOT - PROCEED.
    BIC    #DSITEN,@RXCSR   ;CLEAR MODEM CONTROL INTERRUPT.
    BR     10$
1$:
    PUSH    <R5>            ;SAVE R5
    MOV     MCFLAG,R5        ;USE THE INTERRUPT # AS A TABLE INDEX.
    ASL     R5               ;CHANGE MODEM CONTROL TO ADDRESS OFFSET
    MOV     IRXCSR,MODEM(R5) ;SAVE THE MODEM STATUS
    BIC     #6760,MODEM(R5) ;SAVE ONLY THE MODEM STATUS.
    BIT     #TM,@TXCSR       ;WAS THE TEST MODE BIT SET?
    BEQ     5$              ;BR IF NOT
    BIS     #TM,MODEM(R5)    ;SAVE TEST MODE STATUS.
5$:
    POP     <R5>            ;RESTORE R5
10$:
    BIT     #RSTARY!RDATRY,IRXCSR ;IS THE DATA OR STATUS BIT SET
    SBB     55$
    MOV     @RDSR,IRDSR      ;SAVE THE DATA AND STATUS REG.
    BIT     #RDATRY,IRXCSR   ;IS DATA SET?
    BEQ     20$
    MOVB    IRDSR,(R1)+      ;SAVE THE DATA.
    INC     RCOUNT           ;INCREMENT BYTE COUNT
20$:

```

```

57 017100 032737 002000 002346      BIT      #RSTARY,IRXCSR ;IS STATUS SET?
58 017106 001410                      BEQ      50$
59 017110 032737 106000 002350      BIT      #ERR.ROVER!RABORT,IRDSR ;WAS THERE AN ERROR?
60 017116 001413                      BEQ      53$ ;IF NOT - MUST BE END OF MESSAGE.
61 017120 012737 177777 002376      MOV      #1,RFLAG ;OTHERWISE, SET ERROR FLAG.
62 017126 000412                      BR       54$
63 017130                      50$:
64 017130 005737 002362                      TST      MODE ;IS THIS BCP?
65 017134 001012                      BNE      55$ ;IF NOT - EXIT
66 017136 023737 002500 002474      CMP      RCOUNT,ECOUNT ;HAVE WE RECEIVED ALL THE CHARACTERS
67 017144 001006                      BNE      55$ ;IF NOT - EXIT
68 017146                      53$:
69 017146 012737 000001 002376      MOV      #1,RFLAG ;SET FLAG
70 017154                      54$:
71 017154 042777 000100 163104      BIC      #RXITEN,#RXCSR ;DISABLE INTERRUPT
72 017162                      55$:
73
74 017162                      ENDSRV
   017162
   017162 000002                      L10023:
                                           RTI
75
76

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 017164      BGNSRV  RDATA2
   017164
21
22 017164  105777  163076      TSTB   @RXCSR      ;IS THIS DATA?
23 017170  100404              BMI     5$
24
25 017172  012737  177777  002376  MOV    #-1,RFLAG      ;DATA OR STATUS?
26 017200  000410              BR      20$      ;FLAG FOR ERROR
27
28 017202              5$:
29 017202  117721  163062      MOVB   @RDSR,(R1)+    ;SAVE THE DATA.
30 017206  005337  002322      DEC     COUNTER      ;DECREMENT COUNT
31 017212  001006              BNE     30$          ;BR IF NOT DONE.
32 017214  012737  000001  002376  MOV    #1,RFLAG      ;SET FLAG
33 017222
34 017222  042777  000100  163036  20$:      BIC     #RXITEN,@RXCSR  ;DISABLE INTERRUPT
35 017230      30$:
36
37 017230      ENDSRV
   017230
   017230  000002      L10024:
38                                RTI
39

```

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

```

*****
XINT - INTERRUPT SERVICE ROUTINE

FUNCTION - TRANSMIT INTERRUPT ROUTINE. SET A FLAG WHEN INTERRUPT
          GENERATED. THIS ISR WILL TRANSMIT 4 DATA CHARACTERS AND
          END A MESSAGE IN A SPECIFIED MANNER.

ENTRY CONDITIONS
          ABORT = FLAG, SET IF TERMINATE BY AN ABORT IS DESIRED.
          START = # OF START CHARACTERS (FLAGS OR SYNCHS) TO
                  BE SENT.

EXIT CONDITIONS
          TFLAG = FLAG SET WHEN THIS INTERRUPT IS SERVICED
          DATA = # OF DATA CHARACTERS TRANSMITTED

          USED IN TESTS. 6, 8-11, 14
*****

```

```

BGNSRV XINT
XINT::
MOV #1,TFLAG ;SET THE TRANSMIT FLAG
TST START ;SEND START
BEQ 5$ ;IS THIS DATA OR A START
MOV #TDOM,@TDSR ;TRANSMIT A SYNCH/FLAG.
DEC START ;DECREMENT START COUNTER.
CLR DATA ;CLEAR DATA COUNTER
BR 20$

5$:
CMP #4,DATA ;HAVE WE SENT 4 DATA CHARACTERS
BNE 10$
TST ABORT ;SEND AN ABORT?
BEQ 7$
BIS #TXABO,@TDSR ;SEND AN ABORT
BR 20$

7$:
MOV #TEOM!21,@TDSR ;SEND END OF MESSAGE
BR 20$

10$:
MOV #41,@TDSR ;TRANSMIT DATA.
INC DATA ;INCREMENT DATA

20$:
ENDSRV
L10025: RTI

```

```

017232
017232
017232 012737 000001 002424
017240 005737 002414
017244 001410
017246 012777 000400 163020
017254 005337 002414
017260 005037 002326
017264 000424
017266
017266 022737 000004 002326
017274 001013
017276 005737 002316
017302 001404
017304 052777 002000 162762
017312 000411
017314
017314 012777 001021 162752
017322 000405
017324
017324 012777 000041 162742
017332 005237 002326
017336
017336
017336 000002

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 017340
   017340
27
28 017340 005737 002414
29 017344 001426
30 017346 032737 000001 002434
31 017354 001407
32
33
34
35
36 017356 113777 002434 162720
37 017364 042737 000002 002434
38 017372 000403
39 017374
40 017374 013777 002434 162672
41 017402
42 017402 005337 002414
43 017406 001040
44 017410 005037 002476
45 017414 005037 002500
46 017420 000433
47 017422
48 017422 005737 002336
49 017426 001407
50 017430 100413
51 017432 042777 000400 162634
52 017440 005337 002336
53 017444 000405
54 017446
55 017446 005337 002336
56 017452 153777 002342 162622

*****
XDATA - INTERRUPT SERVICE ROUTINE

FUNCTION - GENERAL PURPOSE TRANSMIT INTERRUPT ROUTINE

ENTRY CONDITIONS
START = # OF START CHARACTERS (FLAGS OR SYNCHS) TO
        BE SENT.
TSTART= TRANSMIT START OF MESSAGE BIT/(OR BITS)
HEADER= # OF HEADER CHARACTERS (8 BIT CHARACTERS) TO
        TRANSMIT BEFORE, SETTING THE SELECTED
        CHARACTER LENGTH.
IPCR = IMAGE OF PCR. CHARACTER LENGTH TO SET AFTER
        THE HEADER CHARACTERS ARE SENT.

EXIT CONDITIONS
XMITD = # OF DATA CHARACTERS TRANSMITTED
RCOUNT= 0 (AFTER START OF MESSAGE TRANSMITTED)

USED IN TESTS: 15-28 & 30-40 (CALLED IN SUBROUTINE $DATA)

*****
BGNSRV XDATA

XDATA::

TST START ;ANY STARTS LEFT TO SEND?
BEQ 10$ ;IF NOT, SKIP.
BIT #BIT0,TSTART ;IS THIS SPECIAL START SEQUENCE.
BEQ 2$ ;IF NOT - SKIP.
; * NOTE: CERTAIN USYNRTS ONLY TRANSMIT
; * A SPECIAL START SEQUENCE WHEN
; * TRANSMIT START AND END OF MESSAGE
; * ARE SET BY A BYTE OPERATION.
MOVB TSTART,@CSR7 ;SEND SPECIAL SEQUENCE START OF MESSAGE.
BIC #BIT1,TSTART ;CLEAR END OF MESSAGE IN SPECIAL START
BR 5$

2$: MOV TSTART,@TDSR ;SEND START OF MESSAGE.

5$: DEC START ;DECREMENT COUNTER.
BNE 20$ ;IF NOT LAST START EXIT.
CLR XMITD ;CLEAR TRANSMIT COUNT.
CLR RCOUNT ;CLEAR RECEIVER COUNT.
BR 20$

10$: TST HEADER ;IS THIS A CONTROL CHARACTER?
BEQ 15$ ;IF DONE WITH CONTROL CHAR, SET LENGTH
BMI 16$ ;AFTERWARDS - BR TO TRANSMIT
BIC #TSOM,@TDSR ;CLEAR START OF MESSAGE.
DEC HEADER ;DECREMENT HEADER COUNT.
BR 16$

15$: DEC HEADER ;MAKE HEADER FLAG - NEGATIVE
BISB IPCR,@PCR ;SET CHARACTER LENGTH (BOP MODE)

```



```

57 017460          16$:
58 017460 112277 162610      MOVB  (R2)+,@TDSR      ;TRANSMIT A CHARACTER.
59 017464 005237 002476      INC   XMITD          ;INCR COUNT OF ACTUALLY SENT.
60 017470 005303              DEC   R3             ;DECREMENT COUNTER
61 017472 001006              BNE   20$
62 017474 053777 002422 162572  BIS   TEND,@TDSR      ;TRANSMIT END OF MESSAGE.
63 017502 042777 000100 162562  BIC   #TXIE,@TXCSR    ;DISABLE TRANSMITTER INTERRUPT.
64 017510          20$:
65
66 017510          ENDSRV
   017510
   017510 000002          L10026:
67                                     RTI

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 017512
    017512
19 017512 005737 002414
20 017516 100414
21 017520 001406
22 017522 052777 000400 162544
23 017530
24 017530 005337 002414
25 017534 000430
26 017536
27 017536 005337 002414
28 017542 042777 000400 162524
29 017550
30 017550 022737 000002 002476
31 017556 001003
32 017560 113777 002342 162514
33 017566
34 017566 112277 162502
35 017572 005237 002476
36 017576 005303
37 017600 001006
38 017602 052777 001000 162464
39 017610 042777 000100 162454
40 017616
41
42
43 017616
    017616
    017616 000002
44
45
46

```

```

*****
XDATA2 - INTERRUPT SERVICE ROUTINE
FUNCTION - HIGH SPEED TRANSMIT INTERRUPT ROUTINE FOR BOP MODE
ENTRY CONDITIONS
    START = # OF START CHARACTERS (FLAGS OR SYNCHS) TO
           BE SENT.
EXIT CONDITIONS
    XMITD = # OF DATA CHARACTERS TRANSMITTED
USED IN TESTS: 31,38,42,43
*****
BGNSRV  XDATA2
                                           XDATA2::
    TST    START                      ;ANY STARTS LEFT TO SEND?
    BMI    20$                        ;IF NEGATIVE SEND DATA
    BEQ    10$                        ;IF NOT, SKIP.
    BIS    #TSOM,@TDSR                ;SEND SYNCH (OR FLAG)
5$:
    DEC    START                      ;DECREMENT COUNTER.
    BR     30$
10$:
    DEC    START                      ;MAKE THE COUNTER NEGATIVE.
    BIC    #TSOM,@TDSR                ;CLEAR START OF MESSAGE
20$:
    CMP    #2,XMITD                   ;IS THIS THE 3RD CHARACTER.
    BNE    25$                        ;IF NOT SKIP
    MOVB   IPCR,@PCR                  ;CHANGE THE CHARACTER LENGTH
25$:
    MOVB   (R2)+,@TDSR                ;TRANSMIT A CHARACTER.
    INC    XMITD                      ;INCR COUNT OF ACTUALLY SENT.
    DEC    R3                         ;DECREMENT COUNTER
    BNE    30$
    BIS    #TEOM,@TDSR                ;TRANSMIT END OF MESSAGE.
    BIC    #TXIE,@TXCSR               ;DISABLE TRANSMITTER INTERRUPT.
30$:
ENDSRV
                                           L10027:
                                           RTI

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 017620
    017620
23
24 017620 005737 002414
25 017624 001413
26 017626 012777 000400 162440
27 017634 005337 002414
28 017640 001034
29 017642 005037 002476
30 017646 005037 002500
31 017652 000427
32 017654
33 017654 042777 001400 162412
34 017662 112277 162406
35 017666 005237 002476
36 017672 005303
37 017674 001016
38 017676 052777 001000 162370
39 017704 005737 002336
40 017710 001005
41 017712 005237 002336
42 017716 012703 000015
43 017722 000403
44 017724
45 017724 042777 000100 162340
46 017732
47
48
49 017732
    017732
    017732 000002
50

```

```

*****
XDDCMP - INTERRUPT SERVICE ROUTINE
*****
FUNCTION - DDCMP TRANSMIT INTERRUPT ROUTINE
ENTRY CONDITIONS
    START = # OF START CHARACTERS (FLAGS OR SYNCHS) TO
            BE SENT.
    HEADER= FLAG WHICH IS SET AFTER THE DDCMP HEADER HAS
            BEEN TRANSMITTED
    DDCMP2= # OF DATA CHARACTERS IN THE DDCMP DATA MESSAGE
EXIT CONDITIONS
    XMITD = # OF DATA CHARACTERS TRANSMITTED
    RCOUNT= 0 (AFTER START OF MESSAGE TRANSMITTED)
USED IN TESTS: 41
*****
BGNSRV  XDDCMP
XDDCMP::
    TST     START          ;ANY STARTS LEFT TO SEND?
    BEQ     10$            ;IF NOT, SKIP.
    MOV     #TSOM,@TDSR    ;SEND START OF MESSAGE.
    DEC     START          ;DECREMENT COUNTER.
    BNE     20$            ;
    CLR     XMITD          ;CLEAR TRANSMIT COUNT.
    CLR     RCOUNT        ;CLEAR RECEIVER COUNT.
    BR      20$
10$:
    BIC     #TEOM!TSOM,@TDSR ;CLEAR START OR END OF MESSAGE.
    MOVB    (R2)+,@TDSR    ;TRANSMIT A CHARACTER.
    INC     XMITD          ;INCR COUNT OF ACTUALLY SENT.
    DEC     R3             ;DECREMENT COUNTER
    BNE     20$            ;
    BIS     #TEOM,@TDSR    ;TRANSMIT END OF MESSAGE.
    TST     HEADER         ;IS THIS THE HEADER
    BNE     15$            ;IF NOT, DISABLE THE TRANSMITTER
    INC     HEADER         ;SET HEADER FLAG.
    MOV     #DDCMP2,R3     ;COUNTER FOR THE MESSAGE
    BR      20$
15$:
    BIC     #TXIE,@TXCSR   ;DISABLE TRANSMITTER INTERRUPT.
20$:
ENDSRV
L10030:
RTI

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 017734
   017734
19
20 017734 012737 000001 002366
21
22 017742
   017742
   017742 000002
23
24
25

```

\*\*\*\*\*

NXM - INTERRUPT SERVICE ROUTINE

FUNCTION - NXM INTERRUPT ROUTINE. THIS ROUTINE IS ASSIGNED TO VECTOR 4 WHEN ADDRESSING THE DPV FOR THE FIRST TIME. IF THIS INTERRUPT IS GENERATED THE DPV IS INCORRECTLY ADDRESSED.

ENTRY CONDITIONS

EXIT CONDITIONS NXMFLG= FLAG SET WHEN THIS INTERRUPT IS SERVICED.

USED IN TESTS: AUTO DROP

\*\*\*\*\*

```

BGNSRV NXM
NXM::
MOV #1,NXMFLG ;SET FLAG IF MEMORY IS NON-EXISTENT.
ENDSRV
L10031:
RTI

```

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

```
*****
ILLGL - INTERRUPT SERVICE ROUTINE

FUNCTION - ILLEGAL INSTRUCTION TRAP TO VECTOR 10
          THIS TRAP WILL OCCUR IF THE PROCESSOR IS AN
          LSI 11 OR LSI 11/2. THIS TRAP IS USED TO
          AUTO SIZE FOR PROCESSOR TYPE IN THE
          INITIALIZATION SECTION.

ENTRY CONDITIONS

EXIT CONDITIONS      R0 = 0

USED IN TESTS:  INIT CODE
*****

BGNSRV  ILLGL

                                ILLGL::

                                CLR      R0

ENDSRV

                                L10032:
                                RTI
```

```

1      .SBTTL  DROP UNIT SECTION
2
3      ;////////////////////////////////////
4      ;// THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5      ;// TO NO LONGER BE TESTED.
6      ;////////////////////////////////////
7
8      017750      BGNDU
9      017750
10     017750      BRESET          ;ISSUE LSI-BUS RESET TO CLEAN UP
11     017750      104433          TRAP      C$RESET
12     017752      013746 002354          MOV      LOGDEV,-(SP)
13     017756      012746 020000          MOV      #FMDROP,-(SP)
14     017762      012746 000002          MOV      #2,-(SP)
15     017766      010600          MOV      SP,R0
16     017770      104417          TRAP      C$PNTF
17     017772      062706 000006          ADD      #6,SP
18
19     017776      ENDDU
20     017776
21     017776      104453          L10033:  TRAP      C$DU
22
23     020000      045      116      045  FMDROP: .ASCIZ  /%N%AUNIT %D2%A DROPPED/
24     020003      101      125      116
25     020006      111      124      040
26     020011      045      104      062
27     020014      045      101      040
28     020017      104      122      117
29     020022      120      120      105
30     020025      104      000
31
32     .EVEN
  
```

```

1      .SBTTL          TEST 1 - CSR ADDRESSING
2
3      *****
4      *              TEST 1 - DPV-11
5      *  VERIFY THAT ADDRESSING THE 4 LSI-BUS CSRS DOES NOT CAUSE A NON-
6      *  EXISTENT MEMORY TRAP.
7      *
8      *  THE DPV IS AN COMMUNICATION DEVICE RESIDING ON A LSI-BUS.
9      *  COMMUNICATION BETWEEN THE MAIN CPU AND THE DPV IS ACCOMPLISHED
10     *  THROUGH A SET OF FOUR 16-BIT LSI-BUS CONTROL AND STATUS REGISTERS
11     *  (CSRS).  THE FOUR REGISTERS ARE ASSIGNED ADDRESSES IN THE I/O PAGE
12     *  FLOATING ADDRESS SPACE: 76XXX0 - 76XXX6
13     *
14     *  AN ERROR IN THIS TEST COULD MEAN THAT THE DEVICE IS INCORRECTLY
15     *  CONFIGURED, THAT THE ADDRESS IS WRONG OR THAT THE CRYSTAL CLOCK
16     *  ON THE DPV IS NOT WORKING.  THE SHIFT REGISTER CLOCK IS NEEDED
17     *  FOR THE LS164 (E15) IN ORDER TO PROVIDE THE BUS REPLY (BRPLY/L ON
18     *  PIN AF2).
19     *  *****
20     BGNTS1
21
22     T1::
23     SETVEC  #4,#LOCATE,#PRI07 ;SET UP NON -EXISTENT MEMORY TRAP VECTOR.
24     MOV     #PRI07,-(SP)
25     MOV     #LOCATE,-(SP)
26     MOV     #4,-(SP)
27     MOV     #3,-(SP)
28     TRAP    C$SVEC
29     ADD     #10,SP
30
31     CLR     NXMFLG           ;FLAG USED IN THE TRAP ROUTINE.
32     CLR     R1              ;USE REGISTER TO REMEMBER WHICH OF THE
33                               ;4 CSRS WE ARE ADDRESSING.
34
35     ;*****
36     ; IF ADDRESSING ANY ONE OF THE CSRS RESULTS IN A TRAP TO VECTOR 04, THE TRAP
37     ; WILL REPORT THE ERROR (SEE INTERRUPT ROUTINE 'LOCATE').  OTHERWISE THE
38     ; MEMORY REFERENCE IS UNEVENTFUL AND THE DEVICE IS READY FOR FURTHER TESTS
39     ;*****
40
41     TST     @CSR0           ;TEST THE CSR AT 76XXX0
42     MOV     #2,R1          ;SAVE THE OFFSET OF THE NEXT CSR
43     TST     @CSR2           ;TEST THE CSR AT 76XXX2
44     MOV     #4,R1          ;SAVE THE OFFSET OF THE NEXT CSR
45     TST     @CSR4           ;TEST THE CSR AT 76XXX4
46     MOV     #6,R1          ;SAVE THE OFFSET OF THE NEXT CSR
47     TST     @CSR6           ;TEST THE CSR AT 76XXX6
48     TST     NXMFLG         ;WAS THERE A TRAP?
49     BEQ     10$            ;IF NOT - EXIT.
50     PRINTX  #FMT1          ;SUGGEST THE PROBLEM. (
51
52     MOV     #FMT1,-(SP)
53     MOV     #1,-(SP)
54     MOV     SP,R0
55     TRAP    C$PNTX
56     ADD     #4,SP
57
58     DODU    LOGDEV         ;DROP THE DEVICE
59
60     MOV     LOGDEV,R0
61     TRAP    C$DODU

```

```

44 020154          DOCLN          ;CLEAN UP CODE - FORCE BACK TO INIT.
    020154 104444          TRAP          CSDCLN
45
46
47 020156          10$:          CLRVEC  #4          ;RETURN VECTOR 04 TO NORMAL STATE
48 020156          012700 000004          MOV          #4,R0
    020156 104436          TRAP          CSCVEC
49
50 020164          ENDTST          L10034:          TRAP          C$ETST
    020164 104401
    020164
51
52
53 020166          BGNSRV LOCATE          ;INTERRUPT SERVICE ROUTINE
    020166          TST          NXMFLG          LOCATE::
54 020166 005737 002366          BNE          10$          ;HAVE WE HAD AT LEAST 1 PREVIOUS TRAP?
55 020172 001006          ERRDF  9,EMTO          ;IF YES, DON'T BOTHER DECLARING ANOTHER
56          ;DEVICE FATAL ERROR
57          ;NON-EXISTENT DEVICE ERROR
    020174 104455          TRAP          C$ERDF
    020176 000011          .WORD          9
    020200 020240          .WORD          EMTO
    020202 000000          .WORD          0
58 020204 005237 002366          10$:          INC          NXMFLG          ;SET THE FLAG
59 020210          PRINTX  #FMT0,R1,CSRO(R1) ;PRINT THE CSR THAT DOESN'T RESPOND.
60 020210          016146 002266          MOV          CSRO(R1),-(SP)
    020214 010146          MOV          R1, -(SP)
    020216 012746 020276          MOV          #FMT0, -(SP)
    020222 012746 000003          MOV          #3, -(SP)
    020226 010600          MOV          SP,R0
    020230 104415          TRAP          C$PNTX
    020232 062706 000010          ADD          #10,SP
61 020236          ENDSRV          L10035:          RTI
    020236 000002
    020236
62
63 020240          103          123          122  EMT0:  .ASCIIZ  /CSR ADDRESSING ERROR - TRAP 4/
    020243          040          101          104
    020246          104          122          105
    020251          123          123          111
    020254          116          107          040
    020257          105          122          122
    020262          117          122          040
    020265          055          040          124
    020270          122          101          120
    020273          040          064          000
64 020276          045          123          063  FMT0:  .ASCIIZ  /%S3%ACSR%D1%A AT %06%A DOES NOT RESPOND%/
    020301          045          101          103
    020304          123          122          045
    020307          104          061          045
    020312          101          040          101
    020315          124          040          045
    020320          117          066          045
    020323          101          040          104
    020326          117          105          123

```



	020331	040	116	117	
	020334	124	040	122	
	020337	105	123	120	
	020342	117	116	104	
	020345	045	116	000	
65	020350	045	101	050	FMT1: .ASCIIZ /%A(CONFIGURATION ERROR OR NO BUS REPLY SIGNAL)%N2/
	020353	103	117	116	
	020356	106	111	107	
	020361	125	122	101	
	020364	124	111	117	
	020367	116	040	105	
	020372	122	122	117	
	020375	122	040	040	
	020400	117	122	040	
	020403	040	116	117	
	020406	040	102	125	
	020411	123	040	122	
	020414	105	120	114	
	020417	131	040	123	
	020422	111	107	116	
	020425	101	114	051	
	020430	045	116	062	
	020433	000			
66					.EVFN
67					
68					
69					

```

1      .SBTTL          TEST 2 - DPV RESET
2
3      *****
4      *              TEST 2 - DPV-11
5      *
6      * DPV RESET
7      * RESET THE DPV AND ENSURE THAT ALL REGISTERS ARE IN THEIR
8      * PROPER INITIALIZATION STATE. THE RESET IS ASYNCHRONOUS TO ALL
9      * DATA SET TIMING AND ANY DATA PORT ACCESSSES. THE FOLLOWING
10     * WILL BE CHECKED BY THE $RESET SUBROUTINE:
11     *   1. ALL BITS IN THE DATA PORT REGISTERS ARE CLEARED.
12     *   2. ALL OUTPUT INDICATORS ARE CLEARED.
13     *   3. TRANSMIT BUFFER EMPTY (TBE) IS SET
14     *
15     * SUBTEST 1 - AFTER RESET, CHECK THAT MAINTENANCE MODE AND
16     * TRANSMITTER CAN BE SET. ALSO CHECK THAT TRANSMITTER
17     * BUFFER EMPTY (TBE) IS CLEARED WHEN TDSR IS ACCESSED
18     * WITHOUT SETTING TRANSMITTER ENABLE.
19     * SUBTEST 2 - ON THE FIRST PASS ONLY, CHECK THAT A BUS RESET, DOES
20     * A DPV11 RESET.
21     *
22     * NOTE: DATA MODE, CTS, RR (RECEIVER READY) AND IC (INCOMING CALL)
23     * ARE UNAFFECTED BY A RESET.
24     *
25     *****
26     RGNTST
27
28     BGNSUB
29
30     CALL $RESET          ;RESET THE DPV
31     ESCAPE TST           ;IF ERROR, EXIT THE TEST
32
33     CLR R1               ;BITS SHOULD BE CLEAR.
34     CLR @TDSR            ;CLEAR TBE
35     TST @TXCSR           ;IS TBE CLEARED?
36     BNE 10$              ;ERROR IF NOT CLEAR
37     MOV #10,R1           ;REMEMBER BITS TO SET.
38     BIS R1,@TXCSR        ;SET THOSE BITS
39     CMP R1,@TXCSR        ;WERE THOSE BITS SET
40     BNE 10$              ;NEXT BIT TO SET
41     MOV #20,R1
42     MOVB R1,@TXCSR
43     CMP R1,@TXCSR
44     BNE 10$
45     MOV #30,R1
46     MOVB #TXENA!MM,@TXCSR ;SET THE ENABLE AND MAINT. MODE.
47     CMP R1,@TXCSR        ;ARE THOSE BITS SET?
48     BNE 10$              ;BR IF IN ERROR.
49     MOV #100,R1          ;SET TX INTERRUPT ENABLE.
50     MOVB R1,@TXCSR        ;SET THE INTERRUPT BIT
51     CMP R1,@TXCSR        ;IS THE BIT SET?
52     BEQ 20$              ;IF YES - OK.
53
54     10$:
55     ERRDF 10,EMG4,ERRG7
56
57     TRAP C$BSCUB
58     TRAP C$ESCAPE
59     TRAP C$ERDF
60
61     104402
62     104410
63     000212
64     005001
65     161620
66     161612
67     000010
68     161600
69     161574
70     000020
71     161562
72     161556
73     000030
74     000030 161542
75     161536
76     000100
77     161524
78     161520
79     001404
80     104455
81     000012

```

Line	Address	Offset	Label	Instruction	Comments	Trap	Word
51	020564		20\$:				
52	020564		ENDSUB				
53	020564	104403				L10037:	TRAP C\$ESUB
54							
55							
56	020566		BGNSUB				
57	020570	022737	000001 002314	CMP #1,STARES	;IS THIS THE FIRST PASS?	T2.2:	TRAP C\$BSUB
58	020576	001026		BNE 30\$	;IF NOT - SKIP THIS TEST.		
59	020600			BRESET	;BUS RESET.		
60	020602	104433		\$DELAY 10	;DELAY 1 MSEC.	TRAP	C\$RESET
	020602	004737	006604	JSR PC,\$DLAY	;***** MACRO EXPANSION *****		
	020606	000010		.WORD 10	;CALL DELAY SUBROUTINE		
					;NUMBER OF DELAY LOOPS		
					;*****		
61	020610	005001		CLR R1	;EXPECT 0 IN ALL R/W REGISTERS		
62	020612	105777	161450	TSTB @RXCSR	;IS THE RECEIVE CSR 0?		
63	020616	001012		BNE 20\$	;BRANCH ON ERROR.		
64	020620	005777	161444	TST @RDSR	;EXPECT READ DATA/STATUS TO BE 0		
65	020624	001007		BNE 20\$	;BR IF NOT		
66	020626	022777	000004 161436	CMP #4,@TXCSR	;EXPECT TBE TO BE SET.		
67	020634	001003		BNE 20\$	;BR IF NOT		
68	020636	005777	161432	TST @TDSR	;EXPECT XMIT. DATA/STATUS TO BE 0.		
69	020642	001404		BFO 30\$	;BRANCH IF OK.		
70	020644		20\$:				
71	020644			ERRDF 11,EMGO,ERRG11	;PRINT ERROR MESSAGE		
	020644	104455				TRAP	C\$ERDF
	020646	000013				.WORD	11
	020650	013372				.WORD	EMGO
	020652	007572				.WORD	ERRG11
72	020654		30\$:				
73	020654		ENDSUB				
	020654					L10040:	
	020654	104403				TRAP	C\$ESUB
74							
75	020656		ENDTST				
	020656					L10036:	
	020656	104401				TRAP	C\$TST

```

1      .SBTTL          TEST 3 - CSR READ/WRITE
2
3      .....
4      *              TEST 3 - DPV-11
5      * WRITE/READ DATA PATTERNS
6      * THIS TEST IS INTENDED TO TEST THE READ/WRITE BITS IN THE CSRS. THERE
7      * IS NO INTENTION TO CHECK THE USYNR/T; IT IS DESIRED TO ONLY CHECK THE
8      * READING AND WRITING OF THE CSRS. IN ALL THE SUBTESTS THE BITS ARE
9      * CHECKED TOGETHER AND INDIVIDUALLY.
10     * SUBTEST 1 - RXCSR (LOW BYTE CSR0)
11     *              CHECK BITS 0-6
12     * SUBTEST 2 - PCR (HIGH BYTE CSR4)
13     *              CHECK BITS 0-7
14     * SUBTEST 3 - TDSR (LOW BYTE OF CSR6) - TRANSMIT DATA BUFFER
15     *              BITS 0-7
16     * SUBTEST 4 - TDSR (HIGH BYTE OF CSR6) - TRANSMIT STATUS REGISTER.
17     *              BITS 0-3
18     * SUBTEST 5 - TDSR - CHECK BYTE OP SIGNAL FOR USYNRT
19     *
20     .....
21     BGNTST
22
23     CALL      $RESET          ;RESET THE DPV
24     ESCAPE    TST             ;IF ERROR, EXIT THE TEST
25
26     TRAP      CS$ESCAPE
27     .WORD     L10041-.
28
29     BGNSUB
30
31     T3.1:
32     TRAP      CS$SUB
33
34     10$:
35     MOV       #BIT0,R1        ;START ROTATE PATTERN
36     MOV       #7,R2           ;COUNTER - WRITE INTO BITS 0-6.
37
38     BISB      R1,@RXCSR        ;WRITE BIT.
39     CMPB      R1,@RXCSR        ;IS THE BIT WRITTEN?
40     BNE       20$             ;IF NOT - REPORT IT.
41     ROL       R1               ;ROTATE THE BIT PATTERN.
42     CLRB      @RXCSR          ;CLEAR REGISTER
43     DEC       R2
44     BNE       10$             ;CONTINUE UNTIL DONE.
45
46     MOV       #137,R1         ;WRITE ALL BITS EXCEPT MODEM CONTROL INT.
47     ;MODEM CONTROL NOT WRITTEN BECAUSE WE DON'T
48     ;WANT TO ACTUALLY GENERATE AN INTERRUPT.
49
50     MOVB      R1,@RXCSR        ;WRITE BITS.
51     CMPB      R1,@RXCSR        ;IS THE PATTERN WRITTEN?
52     BNE       20$             ;IF NOT REPORT IT
53     CLR       R1               ;REMEMBER DATA PATTERN
54     CLRB      @RXCSR          ;CLEAR THOSE BITS.
55     TSTB      @RXCSR          ;ARE THOSE BITS CLEARED?
56     BEQ       30$             ;IF YES, OK.
57
58     20$:
59     ERRDF     12,EMG4,ERRG4
60
61     TRAP      CS$ERDF
62     .WORD     12
63     .WORD     EMG4
64     .WORD     ERRG4

```

49	020770			30\$:					
50	020770	105077	161272		CLRB	@RXCSR		;CLEAR THE REGISTER	
51									
52	020774				ENDSUB				
	020774								
	020774	104403							L10042: TRAP C\$ESUB
53									
54									
55	020776				BGNSUB				
	020776								
	020776	104402							T3.2: TRAP C\$BSUB
56	021000	012701	000377		MOV	#377,R1		;WRITE DATA PATTERN	
57	021004	110177	161272		MOVB	R1,@PCR		;WRITE THE PATTERN.	
58	021010	120177	161266		CMPB	R1,@PCR		;IS THE PATTERN WRITTEN?	
59	021014	001025			BNE	20\$		;IF NOT REPORT IT	
60	021016	005001			CLR	R1		;REMEMBER THE DATA PATTERN	
61	021020	105077	161256		CLRB	@PCR		;CLEAR THOSE BITS	
62	021024	105777	161252		TSTB	@PCR		;WERE THE BITS CLEARED?	
63	021030	001017			BNE	20\$		;IF NOT - REPORT IT	
64	021032	012701	000001		MOV	#BIT0,R1		;START ROTATE PATTERN	
65	021036	012702	000006		MOV	#6,R2		;ROTATE THE BIT 4 TIMES	
66	021042			10\$:					
67	021042	150177	161234		BISB	R1,@PCR		;WRITE PATTERN	
68	021046	120177	161230		CMPB	R1,@PCR		;IS THE PATTERN WRITTEN?	
69	021052	001006			BNE	20\$		;IF NOT - REPORT IT.	
70	021054	006101			ROL	R1		;ROTATE THE PATTERN	
71	021056	105077	161220		CLRB	@PCR		;CLEAR THE PCR.	
72	021062	005302			DEC	R2			
73	021064	001366			BNE	10\$		;CONTINUE UNTIL DONE.	
74	021066	000404			BR	30\$		;EXIT - WHEN DONE	
75	021070			20\$:					
76	021070				ERRDF	13,EMG4,ERRG8			
	021070	104455							TRAP C\$ERDF
	021072	000015							.WORD 13
	021074	013554							.WORD EMG4
	021076	007272							.WORD ERRG8
77	021100			30\$:					
78	021100	105077	161176		CLRB	@PCR		;CLEAR THE PCR	
79									
80	021104				ENDSUB				
	021104								
	021104	104403							L10043: TRAP C\$ESUB
81									
82	021106				BGNSUB				
	021106								
	021106	104402							T3.3: TRAP C\$BSUB
83									
84	021110	012701	000377		MOV	#377,R1		;WRITE DATA PATTERN	
85	021114	110177	161154		MOVB	R1,@TDSR		;WRITE THE PATTERN.	
86	021120	120177	161150		CMPB	R1,@TDSR		;IS THE PATTERN WRITTEN?	
87	021124	001025			BNE	20\$		;IF NOT REPORT IT	
88	021126	005001			CLR	R1		;REMEMBER DATA PATTERN	
89	021130	105077	161140		CLRB	@TDSR		;CLEAR THOSE BITS	
90	021134	105777	161134		TSTB	@TDSR		;IS THE DATA CLEAR?	
91	021140	001017			BNE	20\$		;IF NOT - REPORT IT.	
92	021142	012701	000001		MOV	#BIT0,R1		;START ROTATE PATTERN	
93	021146	012702	000006		MOV	#6,R2		;ROTATE THE BIT 4 TIMES	

Address	Offset	Value	Label	Operation	Comment	Trap	Value
94	021152		10\$:	BISB	R1,@TDSR		
95	021152	150177		CMPB	R1,@TDSR		
96	021156	120177		BNE	20\$		
97	021162	001006		CLRB	@TDSR		
98	021164	105077	161104	ROL	R1		
99	021170	006101		DEC	R2		
100	021172	005302		BNE	10\$		
101	021174	001366		BR	30\$		
102	021176	000404					
103	021200		20\$:	ERRDF	14,EMG4,ERRG9		
104	021200	104455				TRAP	C\$ERDF
	021202	000016				.WORD	14
	021204	013554				.WORD	EMG4
	021206	007372				.WORD	ERRG9
105	021210		30\$:	CLRB	@TDSR		
106	021210	105077	161060				
107							
108							
109	021214		ENDSUB				
	021214					L10044:	
	021214	104403				TRAP	C\$ESUB
110							
111	021216		BGNSUB				
	021216					T3.4:	
	021216	104402				TRAP	C\$BSUB
112	021220	012701		MOV	#17,R1		
113	021224	110177		MOVB	R1,@CSR7		
114	021230	120177		CMPB	R1,@CSR7		
115	021234	001025		BNE	20\$		
116	021236	005001		CLR	R1		
117	021240	105077	161040	CLRB	@CSR7		
118	021244	105777	161034	TSTB	@CSR7		
119	021250	001017		BNE	20\$		
120	021252	012701	000001	MOV	#BIT0,R1		
121	021256	012702	000003	MOV	#3,R2		
122	021262		10\$:				
123	021262	150177	161016	BISB	R1,@CSR7		
124	021266	120177	161012	CMPB	R1,@CSR7		
125	021272	001006		BNE	20\$		
126	021274	105077	161004	CLRB	@CSR7		
127	021300	006101		ROL	R1		
128	021302	005302		DEC	R2		
129	021304	001366		BNE	10\$		
130	021306	000404		BR	30\$		
131	021310		20\$:				
132	021310			ERRDF	15,EMG4,ERRG10		
	021310	104455				TRAP	C\$ERDF
	021312	000017				.WORD	15
	021314	013554				.WORD	EMG4
	021316	007472				.WORD	ERRG10
133	021320		30\$:				
134	021320	105077	160760	CLRB	@CSR7		
135							
136	021324		ENDSUB				
	021324					L10045:	
	021324	104403				TRAP	C\$ESUB

```

137
138 021326          BGNSUB
    021326          T3.5:
    021326 104402          TRAP      C$BSUB
139 021330 012777 007777 160736      MOV      #7777,@TDSR      ;WRITE TO TDSR
140 021336 105077 160742          CLR      @CSR7      ;CLEAR ONLY THE HIGH BYTE.
141 021342 105777 160726          TST      @CSR6      ;SEE IF THE LOW BYTE WAS ALSO CLEARED
142 021346 001016          BNE      10$      ;IF NOT, BYTE OP IS OK.
143 021350 012701 000377          MOV      #377,R1      ;DATA FOR ERROR PRINT OUT.
144 021354          ERRDF      16,EMG4,ERRG9      ;PRINT ERROR
    021354 104455          TRAP      C$ERDF
    021356 000020          .WORD      16
    021360 013554          .WORD      EMG4
    021362 007372          .WORD      ERRG9
145 021364          PRINTX      #FMG30      ;ALSO WARN THAT BYTE OP MAY BE STUCK LOW.
    021364 012746 013314          MOV      #FMG30,-(SP)
    021370 012746 000001          MOV      #1,-(SP)
    021374 010600          MOV      SP,R0
    021376 104415          TRAP      C$PNTX
    021400 062706 000004          ADD      #4,SP
146 021404          10$:
147
148 021404          ENDSUB
    021404          L10046:
    021404 104403          TRAP      C$ESUB
149
150 021406          ENDTST
    021406          L10041:
    021406 104401          TRAP      C$ETST
151
152
153

```

```

1      .SBTTL          TEST 4 - TRANSMIT ENABLE
2
3      ;*****
4      ;          TEST 4 - DMR-11
5      ; TRANSMIT ENABLE/ TRANSMIT ACTIVE
6      ; AFTER A DEVICE RESET, SET TRANSMIT START OF MESSAGE (TSOM).  ENSURE
7      ; THAT TRANSMIT ACTIVE (TXACT) IS SET.
8      ;
9      ; TXACT IS USED TO INDICATE THE CURRENT STATE OF THE TRANSMITTER
10     ; DATA PATH.  THIS BIT WILL BE ASSERTED WHEN BOTH THE TRANSMITTER IS
11     ; ENABLED AND TSOM ARE INTERNALLY SYNCHRONIZED.  TXACT WILL BE CLEARED
12     ; UPON RESET OR WHEN THE TRANSMITTER ENTERS THE IDLE STATE.
13     ;
14     ;*****
15     BGNTST
16
17     BGNSUB
18
19     T4::
20
21     T4.1:
22
23     TRAP      C$BSUB
24
25     CALL      $RESET      ;RESET THE DPV
26     ESCAPE    TST         ;IF ERROR, EXIT THE TEST
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
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
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
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
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
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
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```



021532	004737	003724	JSR	PC,\$WAIT	***** MACRO EXPANSION *****		
021536	000004			.WORD TBE	;CALL WAIT ROUTINE -		
021540	002272			.WORD TXCSR	;WAIT FOR TBE TO BE SET		
					;IN TRANSMITTER CSR.		
					*****		
39 021542			ESCAPE	TST	;IF ERROR, BRANCH TO END OF TEST.	TRAP	C\$ESCAPE
021542	104410					.WORD	L10047-.
021544	000204						
40 021546	032777	000002	BIT	#TXACT,@TXCSR	;IS THE TRANSMITTER INACTIVE?		
41 021554	001406		BEQ	20\$	;IF YES - OK.		
42 021556	012701	000004	MOV	#TBE,R1	;EXPECT ONLY TBE TO BE SET.		
43 021562			ERRDF	18,EMG6,ERRG7			
021562	104455					TRAP	C\$ERDF
021564	000022					.WORD	18
021566	013634					.WORD	EMG6
021570	007172					.WORD	ERRG7
44							
45 021572		20\$:	ESCAPE	TST	;IF ERROR, BRANCH TO END OF TEST	TRAP	C\$ESCAPE
46 021572						.WORD	L10047-.
021572	104410						
021574	000154						
47			ENDSUB				
48 021576							
021576							
021576	104403					TRAP	C\$ESUB
49							
50							
51 021600			BGNSUB				
021600							
021600	104402						
52 021602			CALL	\$RESET	;RESET THE DPV	TRAP	C\$BSUB
53 021606			ESCAPE	TST	;IF ERROR, EXIT THE TEST		
021606	104410					TRAP	C\$ESCAPE
021610	000140					.WORD	L10047-.
54 021612	005737	002306	TST	TURN	;TURNAROUND?		
55 021616	001003		BNE	5\$	;BR IF EXTERNAL.		
56 021620	052777	000010	BIS	#MM,@TXCSR	;SET MAINTENANCE MODE.		
57 021626							
58 021626	052777	000020	BIS	#TXENA,@TXCSR	;ENABLE THE TRANSMITTER.		
59 021634	052777	000400	BIS	#TSOM,@TDSR	;TRANSMIT START OF MESSAGE.		
60 021642			WAIT	TBE	;WAIT FOR TBE TO BE SET.		
021642	004737	003724	JSR	PC,\$WAIT	***** MACRO EXPANSION *****		
021646	000004			.WORD TBE	;CALL WAIT ROUTINE -		
021650	002272			.WORD TXCSR	;WAIT FOR TBE TO BE SET		
					;IN TRANSMITTER CSR.		
					*****		
61 021652			ESCAPE	TST	;IF ERROR, BRANCH TO END OF TEST.	TRAP	C\$ESCAPE
021652	104410					.WORD	L10047-.
021654	000074						
62 021656	032777	000002	BIT	#TXACT,@TXCSR	;IS THE TRANSMITTER ACTIVE?		
63 021664	001010		BNE	10\$	;IF YES - OK.		
64 021666	017701	160400	MOV	@TXCSR,R1	;SAVE THE TRANSMIT STATUS		
65 021672	052701	000020	BIS	#TXENA,R1	;EXPECT TXENA TO BE SET.		
66 021676			ERRDF	19,EMG5,ERRG7			

021676	104455				TRAP	C\$ERDF
021700	000023				.WORD	19
021702	013601				.WORD	EMG5
021704	007172				.WORD	ERRG7
67						
68	021706	10\$:				
69	021706					
70	021712					
	021712					
	021714				TRAP	C\$ESCAPE
	000034				.WORD	L10047-.
71	021716	000002	160346			
72	021724					
73	021726	000004				
74	021732					
	021732					
	021734				TRAP	C\$ERDF
	021736				.WORD	20
	021740				.WORD	EMG6
	007172				.WORD	ERRG7
75						
76	021742	20\$:				
77	021742					
	021742					
	021744				TRAP	C\$ESCAPE
	000004				.WORD	L10047-.
78						
79	021746	ENDSUB				
	021746					
	021746				L10051:	
	104403				TRAP	C\$ESUB
80						
81						
82	021750	ENDTST				
	021750					
	021750				L10047:	
	104401				TRAP	C\$ETST
83						
84						
85						

```

1      .SBTTL          TEST 5 - TRANSMIT BUFFER EMPTY
2
3      ;*****
4      ;          TEST 5 - DPV-11
5      ;          TRANSMIT BUFFER EMPTY
6      ;          VERIFY THAT TBE (TRANSMIT BUFFER EMPTY) IS ASSERTED WHENEVER
7      ;          THE DEVICE IS RESET OR WHENEVER THE TDSR IS AVAILABLE FOR DATA.
8      ;          TBE IS CLEARED AFTER WRITING TO THE TDSR.
9      ;*****
10
11
12     021752          BGNTST
13     021752
14     021752          BGNSUB
15     021752          104402
16     021754          CALL $RESET          ;RESET THE DPV
17     021760          ESCAPE TST          ;IF ERROR, EXIT THE TEST
18     021760          104410
19     021762          000220
20     021764          005077 160304
21     021770          CLP @TDSR          ;WRITE TO THE TDSR.
22     021770          DF AY 5          ;DELAY 500 MICROSECONDS. THIS WILL
23     021774          012727 000005          MOV #5,(PC)+
24     021776          000000          .WORD 0
25     021776          013727 002116          MOV L$DLY,(PC)+
26     022002          000000          .WORD 0
27     022004          005367 177772          DEC -6(PC)
28     022010          001375          BNE -4
29     022012          005367 177756          DEC -22(PC)
30     022016          001367          BNE -20
31
32     022020          032777 000004 160244          BIT #TBE,@TXCSR
33     022026          001410          BEQ 10$
34     022030          017701 160240          MOV @TDSR,R1
35     022034          042701 000004          BIC #TBE,R1
36     022040          ERRDF 21,EMG7,ERRG7          ;US TO ENSURE THAT TBE IS NOT
37     022040          104455          ;REASSERTED. BECAUSE THE TRANSMITTER
38     022042          000025          ;IS IDLE, TBE SHOULD STAY LOW.
39     022044          013671          ;IS TBE CLEARED?
40     022046          007172          ;IF YES - OK
41
42     022050          10$:
43     022050          ENDSUB
44     022050          104403
45
46     022052          BGNSUB
47     022052          104402
48     022054          CALL $RESET          ;RESET THE DPV
49     022060          ESCAPE TST          ;IF ERROR, EXIT THE TEST
50     022060          104410
51     022062          000120
52     022064          005737 002306          TST TURN          ;TURNAROUND?
53
54     T5::
55     T5.1:
56     TRAP C$BSUB
57     TRAP C$ESCAPE
58     .WORD L10052-.
59     MOV #5,(PC)+
60     .WORD 0
61     MOV L$DLY,(PC)+
62     .WORD 0
63     DEC -6(PC)
64     BNE -4
65     DEC -22(PC)
66     BNE -20
67
68     ;US TO ENSURE THAT TBE IS NOT
69     ;REASSERTED. BECAUSE THE TRANSMITTER
70     ;IS IDLE, TBE SHOULD STAY LOW.
71     ;IS TBE CLEARED?
72     ;IF YES - OK
73     ;SAVE THE TRANSMIT DATA/STATUS REG.
74     ;PUT EXPECTED RESULT IN R1 FOR MSG.
75
76     TRAP C$ERDF
77     .WORD 21
78     .WORD EMG7
79     .WORD ERRG7
80
81     10$:
82     ENDSUB
83
84     L10053:
85     TRAP C$ESUB
86
87     T5.2:
88     TRAP C$BSUB
89     TRAP C$ESCAPE
90     .WORD L10052-.

```

```

35 022070 001003      BNE 1$      ;BR IF EXTERNAL.
36 022072 052777 000010 160172 1$: BIS #MM,@TXCSR ;SET MAINTENANCE MODE.
37 022100
38
39 022100 052777 000020 160164      BIS #TXENA,@TXCSR ;ENABLE THE TRANSMITTER.
40 022106 012777 000400 160160      MOV #TSOM,@TDSR ;TRANSMIT START OF MESSAGE.
41 022114      WAIT TBE ;WAIT FOR TBE TO BE SET.

      ;***** MACRO EXPANSION *****
      ;CALL WAIT ROUTINE -
      ;WAIT FOR TBE TO BE SET
      ;IN TRANSMITTER CSR.
      ;*****

      JSR PC,$WAIT
      .WORD TBE
      .WORD TXCSR

42 022124      ESCAPE TST      ;IF ERROR, BRANCH TO END OF TEST.
      022124 104410      TRAP C$ESCAPE
      022126 000054      .WORD L10052-.

43
44 022130 012777 000014 160136      MOV #14,@TDSR ;TRANSMIT 1ST CHARACTER.
45 022136      WAIT TBE ;WAIT FOR TBE TO BE SET.

      ;***** MACRO EXPANSION *****
      ;CALL WAIT ROUTINE -
      ;WAIT FOR TBE TO BE SET
      ;IN TRANSMITTER CSR.
      ;*****

      JSR PC,$WAIT
      .WORD TBE
      .WORD TXCSR

46 022146      ESCAPE TST      ;IF ERROR, BRANCH TO END OF TEST.
      022146 104410      TRAP C$ESCAPE
      022150 000032      .WORD L10052-.
47 022152 012701 001000      MOV #1000,R1 ;SET UP COUNTER
48 022156 5$:
49 022156 005777 160112      TST @TDSR ;CHECK FOR TRANSMIT ERROR.
50 022162 100406      BMI 10$ ;WHEN SET OK.
51 022164 005301      DEC R1 ;DECREMENT COUNTER.
52 022166 001373      BNE 5$ ;CONTINUE UNTIL COUNTER 0
53 022170      ERRDF 22,EMG8,ERRG2
      022170 104455      TRAP C$ERDF
      022172 000026      .WORD 22
      022174 013707      .WORD EMG8
      022176 006700      .WORD ERRG2

54 022200      10$:
55 022200      ENDSUB
      022200      L10054:
      022200 104403      TRAP C$ESUB

56
57
58 022202      ENDTST
      022202      L10052:
      022202 104401      TRAP C$ETST

59
60
61

```

CVDPVBO LPV11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 62  
TEST 6 - TRANSMIT INTERRUPT

```

1      .SBTTL          TEST 6 - TRANSMIT INTERRUPT
2
3      :*****
4      :*              TEST 6 - DPV-11
5      :* TRANSMIT INTERRUPT
6      :* VERIFY THAT A TRANSMIT INTERRUPT IS RECEIVED WHEN TRANSMIT
7      :* BUFFER EMPTY (TBE) IS ASSERTED.
8      :*
9      :*****
10     BGNTST
11
12     022204          CALL    $RESET          ;RESET THE DPV
13     022210          ESCAPE  TST             ;IF ERROR, EXIT THE TEST
14     022210          104410
15     022212          000'46
16     022214          005037  002424          CLR      TFLAG          ;CLEAR THE FLAG USED IN THE INTERRUPT ROUTINE.
17
18     022220          SETVEC  XMTVEC,#XINT,#PRI04
19     022220          012746  000200          MOV      #PRI04,-(SP)
20     022224          012746  017232          MOV      #XINT,-(SP)
21     022230          013746  002264          MOV      XMTVEC,-(SP)
22     022234          012746  000003          MOV      #3,-(SP)
23     022240          104437
24     022242          062706  000010          TRAP     C$SVEC
25     022246          SETPRI  #PRI00          ;SET PROCESSOR PRIORITY. FOR LSI 11/03
26     022246          012700  000000          ADD      #10,SP
27     022252          104441
28
29          ;THIS WILL ENABLE INTERRUPTS. FOR 11/03
30          ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
31          ;LEVEL 4-7.
32          ;SET UP INTERRUPT VECTOR
33
34     022254          052777  000120  160010  BIS      #TXENA!TXIE,@TXCSR ;SET THE INTERRUPT ENABLE AND ENABLE
35     022262          005000          ;THE TRANSMITTER.
36     022264          CLR      R0             ;TIMER FOR LOOP
37     022264          005737  002424          10$:    TST      TFLAG          ;WAS THE INTERRUPT RECEIVED?
38     022270          001006          BNE      20$          ;IF YES - OK.
39     022272          005300          DEC      R0             ;DECREMENT TIMER.
40     022274          001373          BNE      10$          ;KEEP CHECKING UNTIL THE TIMER EXPIRES.
41     022276          ERRDF  23,EMG9,ERRG2    ;ERROR MESSAGE XMIT NOT RECEIVED.
42     022276          104455          TRAP     C$ERDF
43     022300          000027          .WORD    23
44     022302          013723          .WORD    EMG9
45     022304          006700          .WORD    ERRG2
46
47     022306          005037  002424          20$:    CLR      TFLAG          ;CLEAR THE FLAG
48     022312          012777  000001  157752    MOV      #RESET,@TXCSR    ;RESET THE DPV
49     022320          $DELAY  1             ;WAIT FOR 100 MICROSECONDS.
50
51          ;***** MACRO EXPANSION *****
52          ;CALL DELAY SUBROUTINE
53          ;NUMBER OF DELAY LOOPS
54          ;*****
55     022320          004737  006604          JSR      PC,$DLAY
56     022324          000001          .WORD    1

```

```

38 022326 005737 002334      TST      FLAG      ;WAS AN INTERRUPT RECEIVED
39 022332 001404              BEQ      30$      ;IF NOT - OK. (RESET SHOULD CLEAR INT ENABLE)
40 022334              ERRDF  24,EMG10,ERRG2 ;ERROR MESSAGE - TRANSMIT INT RECEIVED
    022334 104455              TRAP      C$ERDF
    022336 000030              .WORD    24
    022340 013757              .WORD    EMG10
    022342 0067C0              .WORD    ERRG2
41 022344              30$:
42 022344              SETPRI  #PRI07      ;SET PROCESSOR PRIORITY TO 7 (FOR
    022344 012700 000340              MOV      #PRI07,RO
    022350 104441              TRAP      C$SPRI
43 022352              ;LSI 11/03 THIS WILL DISABLE INTERRUPTS)
44 022352 013700 002264      CLRVEC  XMTVEC ;RESTORE THE XMIT INTERRUPT VECTOR
    022356 104436              MOV      XMTVEC,RO
    022356 104436              TRAP      C$CVEC
45 022360              ENDTST
46 022360              L10055:
    022360 104401              TRAP      C$SETST
47
48
49
50
  
```

```

1      .SBTTL          TEST 7 - RECEIVER ENABLE
2
3      *****
4      TEST 7 - DPV-11
5      * RECEIVER ENABLE, RECEIVER ACTIVE AND RECEIVER DATA READY
6      * MODE: BCP, 8 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK
7      * ENABLE THE RECEIVER. AFTER TRANSMITTING A CHARACTER WAIT FOR
8      * RECEIVER DATA AVAILABLE AND CHECK THAT THE RECEIVER IS ACTIVE.
9      * AFTER CLEARING RECEIVER ENABLE, ENSURE THAT THE RECEIVER IS INACTIVE.
10
11      * RECEIVER ENABLE - CONTROLS THE OPERATION OF THE RECEIVER DATA PATH (RDP)
12      * RECEIVER ACTIVE - THIS OUTPUT IS ASSERTED WHEN THE RDP PRESENTS THE 1ST
13      * DATA CHARACTER OF A MESSAGE TO THE USYNT. IT REMAINS
14      * ASSERTED UNTIL THE RDP ENTERS THE IDLE STATE..
15      * RECEIVE DATA - THIS OUTPUT IS SET WHEN THE RDP HAS ASSEMBLED A DATA
16      * CHARACTER THAT IS READY TO BE PRESENTED.
17      *****
18      BGNSTST
19
20      022362          CALL $RESET          ;RESET THE DPV
21      022362          ESCAPE TST           ;IF ERROR, EXIT THE TEST
22      022366          104410              TRAP C$ESCAPE
23      022370          000222              .WORD L10056-.
24      022372          012777 040252 157670 MOV #40252,@PCSR ;SET BCP MODE AND SYNCH CHARACTER.
25      022400          012777 000020 157660 MOV #RXENA,@RXCSR ;ENABLE THE RECEIVER.
26      022406          012777 000030 157656 MOV #TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER
27      022414          052777 000400 157652 BIS #TSOM,@TDSR ;SELECT THE MAINTENANCE LOOPBACK.
28      022422          052777 000400 157652 WAIT TBE ;TRANSMIT START OF MESSAGE
29      022422          052777 000400 157652 ;WAIT FOR TBE TO BE SET.
30
31      022422          004737 003724 JSR PC,$WAIT ;***** MACRO EXPANSION *****
32      022426          000004 ;CALL WAIT ROUTINE -
33      022430          002272 ;WAIT FOR TBE TO BE SET
34      022430          002272 ;IN TRANSMITTER CSR.
35      022430          002272 ;*****
36
37      022432          ESCAPE TST           ;IF ERROR, BRANCH TO END OF TEST.
38      022432          104410              TRAP C$ESCAPE
39      022434          000156              .WORD L10056-.
40      022436          032777 004200 157622 BIT #RXACT!RDATRY,@RXCSR ;CHECK RECEIVER ACTIVE AND DATA READY.
41      022444          001056 ;IF SET, REPORT ERROR.
42      022446          052777 000400 157620 BIS #TSOM,@TDSR ;RETRANSMIT START OF MESSAGE.
43      022454          052777 000400 157620 WAIT TBE ;WAIT FOR TBE TO BE SET.
44
45      022454          004737 003724 JSR PC,$WAIT ;***** MACRO EXPANSION *****
46      022460          000004 ;CALL WAIT ROUTINE -
47      022462          002272 ;WAIT FOR TBE TO BE SET
48      022462          002272 ;IN TRANSMITTER CSR.
49      022462          002272 ;*****
50
51      022464          ESCAPE TST           ;IF ERROR, BRANCH TO END OF TEST.
52      022464          104410              TRAP C$ESCAPE
53      022466          000124              .WORD L10056-.
54      022470          032777 004200 157570 BIT #RXACT!RDATRY,@RXCSR ;CHECK RECEIVER ACTIVE AND DATA READY.
55      022476          001041 ;IF SET, REPORT ERROR.
56      022500          012777 000123 157566 MOV #123,@TDSR ;TRANSMIT THE FIRST DATA CHARACTER.

```

```

37 022506          WAIT    RDATRY          ;WAIT FOR RECEIVE DATA.

          022506 004737 003724          JSR    PC,$WAIT          ;***** MACRO EXPANSION *****
          022512 000200                  .WORD    RDATRY          ;CALL WAIT ROUTINE -
          022514 002266                  .WORD    RXCSR          ;WAIT FOR BIT TO BE SET
                                          ;IN RECEIVER CSR.
                                          ;*****

38 022516          ESCAPE  TST              ;IF ERROR, BRANCH TO END OF TEST.
          022516 104410                  TRAP      C$ESCAPE
          022520 000072                  .WORD      L10056-.

39 022522 032777 004000 157536          BIT     #RXACT,@RXCSR      ;IS THE RECEIVER ACTIVE?
40 022530 001005                  BNE      10$                ;IF YES - OK.
41 022532          ERRDF   25,EMG12,ERRG2
          022532 104455                  TRAP      C$ERDF
          022534 000031                  .WORD      25
          022536 014062                  .WORD      EMG12
          022540 006700                  .WORD      ERRG2

42 022542 000423          BR      30$
43 022544          10$:
44 022544 042777 000020 157514          BIC     #RXENA,@RXCSR      ;DISABLE THE RECEIVER
45 022552          $DELAY  4                ;DELAY TO ALLOW DISABLE.

          022552 004737 006604          JSR     PC,$DLAY          ;***** MACRO EXPANSION *****
          022556 000004                  .WORD      4            ;CALL DELAY SUBROUTINE
                                          ;NUMBER OF DELAY LOOPS
                                          ;*****

46 022560 032777 004200 157500          BIT     #RXACT!RDATRY,@RXCSR ;CHECK RECEIVER ACTIVE AND DATA READY.
47 022566 001411                  BEQ      30$                ;IF CLEAR OK
48 022570          ERRDF   26,EMG13,ERRG2
          022570 104455                  TRAP      C$ERDF
          022572 000032                  .WORD      26
          022574 014106                  .WORD      EMG13
          022576 006700                  .WORD      ERRG2

49 022600 000404          BR      30$
50 022602          20$:
51 022602          ERRDF   27,EMG14,ERRG2
          022602 104455                  TRAP      C$ERDF
          022604 000033                  .WORD      27
          022606 014167                  .WORD      EMG14
          022610 006700                  .WORD      ERRG2

52 022612          30$:
53
54 022612          ENDTST
          022612
          022612 104401          L10056: TRAP      C$ETST

55
56
57

```



```

1      .SBTTL      TEST 8 - RECEIVE DATA INTERRUPT
2
3      *****
4      *          TEST 8 - DPV-11
5      *  RECEIVE DATA INTERRUPT
6      *          MODE: BCP, 8 BIT CHARACTERS, MAINTENANCE MODE LOOPBACK
7      *  ENABLE THE RECEIVER AND SET RECEIVER INTERRUPT. TRANSMIT DATA.
8      *  CHECK THAT THE RECEIVE INTERRUPT WAS GENERATED. AFTER THE INTERRUPT
9      *  WAS GENERATED DISABLE THE RECEIVER. CHECK THAT THE RECEIVER BECOMES
10     *  INACTIVE.
11     *
12     *****
13     022614      BGNTST
14     022614
15
16     022614      CALL  $RESET      ;RESET THE DPV
17     022620      ESCAPE  TST        ;IF ERROR, EXIT THE TEST
18     022620      104410
19     022622      000266
20     022624      005037  002424
21     022630      005037  002376
22     022634      005037  002360
23     022640      012737  000002  002414
24     022646      SETVEC  XMTVEC,#XINT,#PRI04
25     022646      012746  000200
26     022652      012746  017232
27     022656      013746  002264
28     022662      012746  000003
29     022666      104437
30     022670      062706  000010
31     022674      SETVEC  RCVEC,#RINT,#PRI04
32     022674      012746  000200
33     022700      012746  016602
34     022704      013746  002262
35     022710      012746  000003
36     022714      104437
37     022716      062706  000010
38     022722      022722      012700  000000
39     022726      104441
40
41     ;THIS WILL ENABLE INTERRUPTS. FOR 11/23
42     ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
43     ;LEVEL 4-7.
44     ;SET UP INTERRUPT VECTOR
45
46     MOV  #40252,@PCSR      ;SET BCP MODE AND SYNCH CHARACTER.
47     MOV  #RXENA!RXITEN,@RXCSR ;ENABLE THE RECEIVER AND SET
48     ;SET INTERRUPT ENABLE.
49     MOV  #TXIE!TXENA!MM,@TXCSR ;ENABLE THE XMITTER AND INT.
50     ;SELECT THE MAINTENANCE LOOPBACK.
51
52     CLR  R3                ;CLEAR COUNTER
53
54     BIT  #1,RFLAG          ;WAS DATA RECEIVED
55     BNE  10$              ;IF YES - OK.

```

```

41 022764 005303      DEC      R3          ;DECREMENT COUNTER.
42 022766 001372      BNE      5$
43
44 022770              ERRDF      28,EMG15,ERRG2
    022770 104455
    022772 000034      TRAP      C$ERDF
    022774 014243      .WORD     28
    022776 006700      .WORD     EMG15
    022776 000430      .WORD     ERRG2
45 023000
46 023002
47 023002 042777 000020 157256 10$: BIC      #RXENA,@RXCSR ;DISABLE THE RECEIVER
48 023010 005037 002376      CLR      RFLAG      ;CLEAR THE FLAG.
49 023014      $DELAY      5          ;DELAY TO ALLOW DISABLE.

    023014 004737 006604      JSR      PC,$DLAY      ;***** MACRO EXPANSION *****
    023020 000005      .WORD      5          ;CALL DELAY SUBROUTINE
    ;NUMBER OF DELAY LOOPS
    ;*****

50 023022 005737 002376      TST      RFLAG      ;WAS AN INTERRUPT RECEIVED?
51 023026 001011      BNE      20$          ;IF YES - REPORT EPROR.
52 023030 032777 004200 157230 BIT      #RXACT!RDATRY,@RXCSR ;CHECK RECEIVER ACTIVE AND DATA READY.
53 023036 001411      BEQ      30$          ;IF CLEAR OK
54 023040      ERRDF      29,EMG13,ERRG2
    023040 104455      TRAP      C$ERDF
    023042 000035      .WORD     29
    023044 014106      .WORD     EMG13
    023046 006700      .WORD     ERRG2
55 023050 000404
56 023052      BR        30$
57 023052      ERRDF      30,EMG16,ERRG2
    023052 104455      TRAP      C$ERDF
    023054 000036      .WORD     30
    023056 014276      .WORD     EMG16
    023060 006700      .WORD     ERRG2
58 023062
59 023062      30$:      CALL      $RESET      ;RESET THE DPV.
60 023066      SETPRI     #PRI07      ;SET THE PROCESSOR PRI TO 7
    023066 012700 000340      MOV      #PRI07,R0
    023072 104441      TRAP      C$SPRI

61
62 023074      CLRVEC     RCVEC      ;(THIS WILL DISABLE INTERRUPTS)
    023074 013700 002262      ;RESTORE THE RECV. VECTOR
    023100 104436      MOV      RCVEC,R0
    023102      TRAP      C$CVEC
63 023102      CLRVEC     XMTVEC      ;RESTORE THE XMIT. VECTOR
    023102 013700 002264      MOV      XMTVEC,R0
    023106 104436      TRAP      C$CVEC
64
65 023110      ENDTST
    023110
    023110 104401      L10057: TRAP      C$ETST
66
67
68

```

```

1      .SBTTL          TEST 9 - RECEIVER STATUS
2
3      :*****
4      :*              TEST 9 - DPV-11
5      :*  THERE ARE 3 SUBTESTS IN THIS TEST WHICH ARE INTENDED TO CHECK
6      :*  RECEIVER STATUS.
7      :*  SUBTEST 1 - REOM (RECEIVE END OF MESSAGE)
8      :*              THIS SUBTEST WILL TRANSMIT A DATA MESSAGE THAT IS
9      :*              ENDED WITH A TEOM (TRANSMIT END OF MESSAGE). A
10     :*              CHECK WILL BE MADE THAT THE RECEIVER GETS THE DATA
11     :*              AND THAT THE REOM IS RECEIVED WHEN RECEIVE
12     :*              STATUS IS AVAILABLE.
13     :*
14     :*  SUBTEST 2 - RECEIVER OVERRUN
15     :*              THIS SUBTEST WILL TRANSMIT DATA CORRECTLY. THE
16     :*              RECEIVER AFTER BECOMING ACTIVE WILL NOT SERVICE
17     :*              THE RECEIVE BUFFER CORRECTLY. THIS SHOULD RESULT IN
18     :*              A RECEIVE OVERRUN. THIS SUBTEST WILL ENSURE THAT
19     :*              WHEN RECEIVE STATUS IS AVAILABLE, THE RECEIVER OVERRUN
20     :*              IS SET.
21     :*
22     :*  SUBTEST 3 - RECEIVER ABORT
23     :*              THIS SUBTEST WILL TRANSMIT A DATA MESSAGE THAT IS ENDED
24     :*              WITH A TRANSMIT ABORT. THE SUBTEST WILL ENSURE THAT
25     :*              RECEIVE STATUS AVAILABLE IS RECEIVED AND THAT THE
26     :*              ABORT IS RECEIVED.
27     :*
28     :*****
29     BGNTST
30
31     BGNSUB
32
33     T9.:
34
35     T9.1:
36
37     TRAP      C$BSUB
38
39     TRAP      C$ESCAPE
40     .WORD     L'0060-.
41
42     CLR      TFLAG      ;CLEAR TRANSMIT INTERRUPT FLAG.
43     MOV      #1,START   ;# OF START OF MESSAGES.
44
45     SETVEC   XMTVEC,#XINT,#PRI04
46
47     MOV      #PRI04,-(SP)
48     MOV      #XINT,-(SP)
49     MOV      XMTVEC,-(SP)
50     MOV      #3,-(SP)
51     TRAP     C$SVEC
52     ADD      #10,SP
53
54     SETPRI   #PRI00      ;SET PROCESSOR PRIORITY. FOR LSI 11/03
55
56     MOV      #PRI00,R0
57     TRAP     C$SPRI
58
59     ;THIS WILL ENABLE INTERRUPTS. FOR 11/23
60     ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
61     ;LEVEL 4-7.
62     ;SET UP INTERRUPT VECTOR
63
64     BIS      #TXIE!TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER AND SELECT

```

CVDPVBO DVP11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 66-1  
TEST 9 - RECEIVER STATUS

```

45                                     ;MAINTENANCE MODE LOOPBACK.
46 023200 052777 000020 157060      BIS      #RXENA,@RXCSR    ;ENABLE THE RECEIVER
47                                     ;
48 023206 005003                      CLR      R3              ;INITIALIZE THE COUNTER
49 023210                                5$:
50 023210 032777 004000 157050      BIT      #RXACT,@RXCSR    ;IS THE RECEIVER ACTIVE?
51 023216 001007                      BNE     10$              ;BR IF YES
52 023220 005303                      DEC     R3              ;DECREMENT THE COUNTER
53 023222 001372                      BNE     5$
54 023224                                ERRDF   31,EMG12,ERRG2
                                     TRAP      C$ERDF
                                     .WORD     31
                                     .WORD     EMG12
                                     .WORD     ERRG2
55 023234 000444                      BR       45$
56 023236                                10$:
57 023236 005003                      CLR      R3              ;INITIALIZE THE COUNTER.
58 023240                                12$:
59 023240 032777 002200 157020      BIT      #RSTARY!RDATRY,@RXCSR ;IS DATA OR STATUS READY?
60 023246 001407                      BEQ     15$              ;BR IF NOT
61 023250 017737 157014 002400      MOV     @RDSR,RSAVE        ;SAVE THE CHARACTER
62 023256 032737 001000 002400      BIT      #REOM,RSAVE      ;WAS THE RECEIVE END OF MESSAGE RECEIVED?
63 023264 001007                      BNE     20$
64 023266                                15$:
65 023266 005303                      DEC     R3              ;DECREMENT THE COUNTER
66 023270 001363                      BNE     12$
67 023272                                ERRDF   32,EMG17,ERRG2
                                     TRAP      C$ERDF
                                     .WORD     32
                                     .WORD     EMG17
                                     .WORD     ERRG2
68 023302 000421                      BR       45$
69 023304                                20$:
70 023304 032777 002000 156754      BIT      #RSTARY,@RXCSR    ;IS THE STATUS DROPPED?
71 023312 001405                      BEQ     25$
72 023314                                ERRDF   33,EMG18,ERRG2
                                     TRAP      C$ERDF
                                     .WORD     33
                                     .WORD     EMG18
                                     .WORD     ERRG2
73 023324 000410                      BR       45$
74 023326                                25$:
75 023326 032777 004000 156732      BIT      #RXACT,@RXCSR    ;IS THE RECEIVER INACTIVE?
76 023334 001404                      BEQ     45$              ;BR IF YES
77 023336                                ERRDF   34,EMG11,ERRG2
                                     TRAP      C$ERDF
                                     .WORD     34
                                     .WORD     EMG11
                                     .WORD     ERRG2
78 023336 000455                      BR       45$
79 023346                                45$:
80                                     ENDSUB
81 023346                                L10061:
82 023346 000403                      TRAP     C$ESUB
83 023350                                BGNSUB

```

CVDPVBO DPV11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 66-2  
TEST 9 - RECEIVER STATUS

```

      023350
      023350 104402
84 023352
85 023356          CALL $RESET      ;RESET THE DPV
      023356 104410          ESCAPE TST      ;IF ERROR, EXIT THE TEST
      023360 000536          TRAP      C$BSSUB
                                .WORD      C$ESCAPE
                                .WORD      L10060-.
86
87 023362 005037 002424          CLR      TFLAG      ;CLEAR TRANSMIT INTERRUPT FLAG.
88 023366 012737 000001 002414          MOV      #1,START      ;# OF START OF MESSAGES.
89
90 023374          SETVEC      XMTVEC,#XINT,#PRIO4
      023374 012746 000200          MOV      #PRIO4,-(SP)
      023400 012746 017232          MOV      #XINT,-(SP)
      023404 013746 002264          MOV      XMTVEC,-(SP)
      023410 012746 000003          MOV      #3,-(SP)
      023414 104437          TRAP      C$SVEC
      023416 062706 000010          ADD      #10,SP
91 023422          SETPRI      #PRIO0      ;SET PROCESSOR PRIORITY. FOR LSI 11/03
      023422 012700 000000          MOV      #PRIO0,R0
      023426 104441          TRAP      C$SPRI
                                ;THIS WILL ENABLE INTERRUPTS. FOR 11/23
                                ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
                                ;LEVEL 4-7.
                                ;SET UP INTERRUPT VECTOR
92
93
94
95
96
97 023430 052777 000130 156634          BIS      #TXIE!TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER AND SELECT
98                                     ;MAINTENANCE MODE LOOPBACK.
99 023436 052777 000020 156622          BIS      #RXENA,@RXCSR ;ENABLE THE RECEIVER
100
101 023444 005003          CLR      R3      ;INITIALIZE THE COUNTER
102 023446          5$:
103 023446 032777 004000 156612          BIT      #RXACT,@RXCSR ;IS THE RECEIVER ACTIVE?
104 023454 001007          BNE      10$      ;BR IF YES
105 023456 005303          DEC      R3      ;DECREMENT THE COUNTER
106 023460 001372          BNE      5$
107 023462          ERRDF      35,EMG12,ERRG2
      023462 104455          TRAP      C$ERDF
      023464 000043          .WORD      35
      023466 014062          .WORD      EMG12
      023470 006700          .WORD      ERRG2
108 023472 000464          BR      55$
109 023474          10$:
110 023474 005003          CLR      R3      ;INITIALIZE THE COUNTER.
111 023476          12$:
112 023476 032777 002000 156562          BIT      #RSTARY,@RXCSR ;IS THE STATUS READY?
113 023504 001007          BNE      20$      ;DECREMENT THE COUNTER
114 023506 005303          DEC      R3
115 023510 001372          BNE      12$
116
117 023512          ERRDF      36,EMG1,ERRG2 ;TIME OUT
      023512 104455          TRAP      C$ERDF
      023514 000044          .WORD      36
      023516 013462          .WORD      EMG1
      023520 006700          .WORD      ERRG2
118 023522 000450          BR      55$
119
120 023524          20$:

```

121									
122	023524	032777	004000	156536	BIT	#ROVER,@RDSR	;WAS THE RECEIVE OVERRUN RECEIVED?		
123	023532	001005			BNE	40\$	;IF YES OK.		
124	023534				ERRDF	37,EMG19,ERRG2			
	023534	104455						TRAP	C\$ERDF
	023536	000045						.WORD	37
	023540	014434						.WORD	EMG19
	023542	006700						.WORD	ERRG2
125	023544	000437			BR	55\$			
126	023546				40\$:				
127									
128	023546	032777	002000	156512	BIT	#RSTARY,@RXCSR	;WAS THE STATUS CLEARED		
129	023554	001405			BEQ	42\$	;IF YES OK		
130	023556				ERRDF	38,EMG18,ERRG2			
	023556	104455						TRAP	C\$ERDF
	023560	000046						.WORD	38
	023562	014405						.WORD	EMG18
	023564	006700						.WORD	ERRG2
131	023566	000426			BR	55\$			
132	023570				42\$:				
133	023570	032777	002000	156470	BIT	#RSTARY,@RXCSR	;IS THE STATUS READY?		
134	023576	001007			BNE	47\$			
135	023600	005303			DEC	R3	;DECREMENT THE COUNTER		
136	023602	001372			BNE	42\$			
137									
138	023604				ERRDF	39,EMG1,ERRG2	;TIME OUT		
	023604	104455						TRAP	C\$ERDF
	023606	000047						.WORD	39
	023610	013462						.WORD	EMG1
	023612	006700						.WORD	ERRG2
139	023614	000413			BR	55\$			
140									
141									
142	023616				47\$:				
143	023616	042777	000020	156442	BIC	#RXENA,@RXCSR	;DISABLE THE RECEIVER.		
144									
145	023624	032777	002000	156434	BIT	#RSTARY,@RXCSR	;IS THE STATUS DROPPED?		
146	023632	001404			BEQ	55\$			
147	023634				50\$:				
148	023634				ERRDF	40,EMG18,ERRG2			
	023634	104455						TRAP	C\$ERDF
	023636	000050						.WORD	40
	023640	014405						.WORD	EMG18
	023642	006700						.WORD	ERRG2
149	023644				55\$:				
150									
151	023644				ENDSUB				
	023644							L10062:	
	023644	104403						TRAP	C\$ESUB
152									
153									
154									
155	023646				BGNSUB				
	023646							T9.3:	
	023646	104402						TRAP	C\$BSUB
156	023650				CALL	\$RESET	;RESET THE DPV		
157	023654				ESCAPE	TST	;IF ERROR, EXIT THE TEST		

CVDPVBO DPV11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 66-4  
TEST 9 - RECEIVER STATUS

	023654	104410						TRAP	C\$ESCAPE
	023656	000240						.WORD	L10060-
158	023660	005037	002424			CLR	TFLAG		;CLEAR TRANSMIT INTERRUPT FLAG.
159	023664	012737	000001	002414		MOV	#1,START		;# OF START OF MESSAGES.
160	023672	012737	000001	002316		MOV	#1,ABORT		;SEND AN ABORT
161									
162	023700					SETVEC	XMTVEC,#XINT,#PRI04		
	023700	012746	000200					MOV	#PRI04,-(SP)
	023704	012746	017232					MOV	#XINT,-(SP)
	023710	013746	002264					MOV	XMTVEC,-(SP)
	023714	012746	000003					MOV	#3,-(SP)
	023720	104437						TRAP	C\$SVEC
	023722	062706	000010					ADD	#10,SP
163	023726					SETPRI	#PRI00		;SET PROCESSOR PRIORITY. FOR LSI 11/03
	023726	012700	000000					MOV	#PRI00,RO
	023732	104441						TRAP	C\$SPRI
164									;THIS WILL ENABLE INTERRUPTS. FOR 11/23
165									;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
166									;LEVEL 4-7.
167									;SET UP INTERRUPT VECTOR
168									
169	023734	052777	000130	156330		BIS	#TXIE!TXENA!MM,@TXCSR		;ENABLE THE TRANSMITTER AND SELECT
170									;MAINTENANCE MODE LOOPBACK.
171	023742	052777	000020	156316		BIS	#RXENA,@RXCSR		;ENABLE THE RECEIVER
172									
173	023750	005003				CLR	R3		;INITIALIZE THE COUNTER
174	023752				5\$:				
175	023752	032777	004000	156306		BIT	#RXACT,@RXCSR		;IS THE RECEIVER ACTIVE?
176	023760	001007				BNE	10\$		;BR IF YES
177	023762	005303				DEC	R3		;DECREMENT THE COUNTER
178	023764	001372				BNE	5\$		
179	023766					ERRDF	41,EMG12,ERRG2		
	023766	104455						TRAP	C\$ERDF
	023770	000051						.WORD	41
	023772	014062						.WORD	EMG12
	023774	006700						.WORD	ERRG2
180	023776	000444				BR	45\$		
181	024000				10\$:				
182	024000	005003				CLR	R3		;INITIALIZE THE COUNTER.
183	024002				12\$:				
184	024002	032777	002000	156256		BIT	#RSTARY,@RXCSR		;IS THE STATUS READY?
185	024010	001016				BNE	20\$		
186	024012	032777	000200	156246		BIT	#RDATRY,@RXCSR		
187	024020	001403				BEQ	15\$		
188	024022					\$DELAY	5		;DELAY .5 MSEC.
									;***** MACRO EXPANSION *****
	024022	004737	006604			JSR	PC,\$DLAY		;CALL DELAY SUBROUTINE
	024026	000005					.WORD 5		;NUMBER OF DELAY LOOPS
									;*****
189	024030				15\$:				
190	024030	005303				DEC	R3		;DECREMENT THE COUNTER
191	024032	001363				BNE	12\$		
192	024034					ERRDF	42,EMG1,ERRG2		;TIME OUT
	024034	104455						TRAP	C\$ERDF
	024036	000052						.WORD	42

LINE	ADDRESS	DATA	INSTR	OPERANDS	COMMENT	TRAP	CS
193	024040	013462					EMG1
194	024042	006700					ERRG2
195	024044	000421	BR	45\$			
196	024046		20\$:				
197	024046	032777	002000	156214	BIT #RABORT, @RDSR ; WAS THE RECEIVE ABORT RECEIVED?		
198	024054	001005	BNE	40\$	; IF YES OK.		
199	024056		ERRDF	43, EMG20, ERRG2			
200	024056	104455				TRAP	C\$ERDF
201	024060	000053				.WORD	43
202	024062	014465				.WORD	EMG20
203	024064	006700				.WORD	ERRG2
204	024066	000410	BR	45\$			
205	024070		40\$:				
206	024070	032777	002000	156170	BIT #RSTARY, @RXCSR ; IS THE STATUS DROPPED?		
207	024076	001404	BEQ	45\$			
208	024100		ERRDF	44, EMG18, ERRG2			
209	024100	104455				TRAP	C\$ERDF
210	024102	000054				.WORD	44
211	024104	014405				.WORD	EMG18
212	024106	006700				.WORD	ERRG2
213	024110		45\$:				
214	024110	005037	002316		CLR ABORT ; CLEAR THE ABORT FLAG.		
215	024114		ENDSUB				
216	024114					L10063:	
217	024114	104403				TRAP	C\$ESUB
218	024116		ENDTST				
219	024116					L10060:	
220	024116	104401				TRAP	C\$ETST



CVDPVBO DPV11 FUNC DIAG MACRO V03.01 28-OCT-80 08:56:53 PAGE 67  
TEST 10 - RECEIVE STATUS INTERRUPT

```

1      .SBTTL          EST 10 - RECEIVE STATUS INTERRUPT
2
3      *****
4      *              TEST 10 - DPV-11
5      * THIS TEST WILL ENSURE THAT INTERRUPTS MAY BE GENERATED WHEN
6      * RECEIVE STATUS IS AVAILABLE. EACH OF THE FOLLOWING SUBTESTS
7      * WILL GENERATE THE STATUS AS FOLLOWS:
8      * SUBTEST 1 - REOM
9      * SUBTEST 2 - RECEIVER OVERRUN
10     * SUBTEST 3 - RECEIVER ABORT
11     *
12     *****
13     024120          BGNTST
14     024120
15
16     024120          BGNSUB
17     024120          104402
18     024122          CALL $RESET          ;RESET THE DPV
19     024126          ESCAPE TST          ;IF ERROR, EXIT THE TEST
20     024126          104410
21     024130          001102
22     024132          005037 002376
23     024136          005037 002424
24     024142          005037 002360
25     024146          012737 000001 002414
26     024154          SETVEC XMTVEC,#XINT,#PRI04
27     024154          012746 000200
28     024160          012746 017232
29     024164          013746 002264
30     024170          012746 000003
31     024174          104437
32     024176          062706 000010
33     024202          SETVEC RCVEC,#RINT,#PRI04
34     024202          012746 000200
35     024206          012746 016602
36     024212          013746 002262
37     024216          012746 000003
38     024222          104437
39     024224          062706 000010
40     024230          SETPRI #PRI00          ;SET PROCESSOR PRIORITY. FOR LSI 11/03
41     024230          012700 000000
42     024234          104441
43
44     ;THIS WILL ENABLE INTERRUPTS. FOR 11/23
45     ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
46     ;LEVEL 4-7.
47     ;SET UP INTERRUPT VECTOR
48
49     024236          052777 000130 156026
50     024244          052777 000120 156014
51
52     024252          005003
53     024254          032777 004000 156004
54
55     BIS #TXIE'TXENA'MM,@TXCSR ;ENABLE THE TRANSMITTER AND SELECT
56                                     ;MAINTENANCE MODE LOOPBACK.
57     BIS #RXITEN!RXENA,@RXCSR ;ENABLE THE RECEIVER
58
59     CLR R3          ;INITIALIZE THE COUNTER
60
61     BIT #RXACT,@RXCSR ;IS THE RECEIVER ACTIVE?

```

```

39 024262 001007      BNE      10$      ;BR IF YES
40 024264 005303      DEC      R3        ;DECREMENT THE COUNTER
41 024266 001372      BNE      5$
42 024270      ERRDF    45,EMG12,ERRG2
      024270 104455
      024272 000055      TRAP      C$ERDF
      024274 014062      .WORD    45
      024276 006700      .WORD    EMG12
      024276 006700      .WORD    ERRG2
43 024300 000434      BR       45$
44 024302      10$:
45 024302 005003      CLR      R3        ;INITIALIZE THE COUNTER.
46 024304      12$:
47 024304 032737 000002 002376      BIT      #2,RFLAG      ;WAS STATUS RECEIVED?
48 024312 001007      BNE      20$
49 024314 005303      DEC      R3        ;DECREMENT THE COUNTER
50 024316 001372      BNE      12$
51 024320      ERRDF    46,EMG21,ERRG2
      024320 104455      TRAP      C$ERDF
      024322 000056      .WORD    46
      024324 014514      .WORD    EMG21
      024326 006700      .WORD    ERRG2
52 024330 000420      BR       45$
53
54 024332      20$:
55 024332 032737 001000 002400      BIT      #REOM,RSAVE      ;WAS THE RECEIVE END OF MESSAGE RECEIVED?
56 024340 001004      BNE      40$      ;IF YES OK.
57 024342      ERRDF    47,EMG17,ERRG2
      024342 104455      TRAP      C$ERDF
      024344 000057      .WORD    47
      024346 014345      .WORD    EMG17
      024350 006700      .WORD    ERRG2
58 024352      40$:
59 024352 032777 002000 155706      BIT      #RSTARY,ARXCSR      ;IS THE STATUS DROPPED?
60 024360 001404      BEQ      45$
61 024362      ERRDF    48,EMG18,ERRG2
      024362 104455      TRAP      C$ERDF
      024364 000060      .WORD    48
      024366 014405      .WORD    EMG18
      024370 006700      .WORD    ERRG2
62 024372      45$:
63 024372      SETPRI   #PRI07      ;SET PROCESSOR PRI TO 7
      024372 012700 000340      MOV      #PRI07,R0
      024376 104441      TRAP      C$SPRI
64
65 024400      CLRVEC   RCVEC      ;(DISABLE INTERRUPT)
      024400 013700 002262      ;RESTORE THE INTERRUPT VECTOR.
      024404 104436      MOV      RCVEC,R0
66 024406      CLRVEC   XMTVEC      ;RESORE THE VECTOR.
      024406 013700 002264      MOV      XMTVEC,R0
      024412 104436      TRAP      C$CVEC
67
68 024414      ENDSUB
      024414
      024414 104403      L10065: TRAP      C$ESUB
69
70 024416      BGNSUB
      024416      T10.2:

```

Address	Offset	Label	Instruction	Comment	Trap Word	Trap Label
71	024416	104402				
72	024420		CALL	\$RESET		
	024424		ESCAPE	TST		
	024424	104410				
	024426	000604				
73						
74	024430	005037	002376	CLR	RFLAG	:CLEAR RECEIVE INTERRUPT
75	024434	005037	002424	CLR	TFLAG	:CLEAR TRANSMIT INTERRUPT FLAG.
76	024440	005037	002360	CLR	MCFLAG	:CLEAR MODEM CONTROL FLAG.
77	024444	012737	000001	MOV	#1,START	:# OF START OF MESSAGES.
78	024452	012737	000001	MOV	#1,OVER	:FLAG TO CREATE RECEIVE OVERRUN.
79						
80	024460			SETVEC	XMTVEC,#XINT,#PRI04	
	024460	012746	000200			MOV #PRI04,-(SP)
	024464	012746	017232			MOV #XINT,-(SP)
	024470	013746	002264			MOV XMTVEC,-(SP)
	024474	012746	000003			MOV #3,-(SP)
	024500	104437				TRAP C\$SVEC
	024502	062706	000010			ADD #10,SP
81	024506			SETVEC	RCVEC,#RINT,#PRI04	
	024506	012746	000200			MOV #PRI04,-(SP)
	024512	012746	016602			MOV #RINT,-(SP)
	024516	013746	002262			MOV RCVEC,-(SP)
	024522	012746	000003			MOV #3,-(SP)
	024526	104437				TRAP C\$SVEC
	024530	062706	000010			ADD #10,SP
82	024534			SETPRI	#PRI00	:SET PROCESSOR PRIORITY. FOR LSI 11/03
	024534	012700	000000			MOV #PRI00,R0
	024540	104441				TRAP C\$SPRI
83						:THIS WILL ENABLE INTERRUPTS. FOR 11/23
84						:THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
85						:LEVEL 4-7.
86						:SET UP INTERRUPT VECTOR
87						
88	024542	052777	000130	155522	BIS	#TXIE!TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER AND SELECT
89						;MAINTENANCE MODE LOOPBACK.
90	024550	052777	000120	155510	BIS	#RXITEN!RXENA,@RXCSR ;ENABLE THE RECEIVER
91						
92	024556	005003			CLR	R3 ;INITIALIZE THE COUNTER
93	024560					
94	024560	032777	004000	155500	BIT	#RXACT,@RXCSR ;IS THE RECEIVER ACTIVE?
95	024566	001007			BNE	10\$ ;BR IF YES
96	024570	005303			DEC	R3 ;DECREMENT THE COUNTER
97	024572	001372			BNE	5\$
98	024574				ERRDF	49,EMG12,ERRG2
	024574	104455				
	024576	000061				TRAP C\$ERDF
	024600	014062				.WORD 49
	024602	006700				.WORD EMG12
	024604	000434				.WORD ERRG2
99					BR	45\$
100	024606					
101	024606	005003			CLR	R3 ;INITIALIZE THE COUNTER.
102	024610					
103	024610	032737	000002	002376	BIT	#2,RFLAG ;WAS STATUS RECEIVED?
104	024616	001007			BNE	20\$
105	024620	005303			DEC	R3 ;DECREMENT THE COUNTER
106	024622	001372			BNE	12\$

Line	Address	Offset	Label	Instruction	Comment	Trap	Trap Word
107	024624		ERRDF	50,EMG21,ERRG2		TRAP	C\$ERDF
	024624	104455				.WORD	50
	024626	000062				.WORD	EMG21
	024630	014514				.WORD	ERRG2
	024632	006700					
108	024634	000420	BR	45\$			
109							
110	024636		20\$:				
111							
112	024636	032737	004000	002400	BIT #ROVER,RSAVE ;WAS THE RECEIVE OVERRUN RECEIVED?		
113	024644	001004			BNE 40\$ ;IF YES OK.		
114	024646		ERRDF	51,EMG19,ERRG2		TRAP	C\$ERDF
	024646	104455				.WORD	51
	024650	000063				.WORD	EMG19
	024652	014434				.WORD	ERRG2
	024654	006700					
115	024656		40\$:				
116	024656	032777	002000	155402	BIT #RSTARY,@RXCSR ;IS THE STATUS DROPPED?		
117	024664	001404			BEQ 45\$		
118	024666		ERRDF	52,EMG18,ERRG2		TRAP	C\$ERDF
	024666	104455				.WORD	52
	024670	000064				.WORD	EMG18
	024672	014405				.WORD	ERRG2
	024674	006700					
119	024676		45\$:				
120	024676		SETPRI	#PRI07	;SET PROCESSOR PRI TO 7	MOV	#PRI07 RO
	024676	012700	000340			TRAP	C\$SPRI
	024702	104441					
121					; (DISABLE INTERRUPT)		
122	024704		CLRVEC	RCVEC	;RESTORE THE INTERRUPT VECTOR.	MOV	RCVEC,RO
	024704	013700	002262			TRAP	C\$CVEC
	024710	104436					
123	024712		CLRVEC	XMTVEC		MOV	XMTVEC,RO
	024712	013700	002264			TRAP	C\$CVEC
	024716	104436					
124	024720	005037	002370		CLR OVER ;CLEAR OVERRUN FLAG.		
125							
126	024724		ENDSUB				
	024724					L10066:	
	024724	104403				TRAP	C\$ESUB
127							
128							
129							
130	024726		BGNSUB				
	024726					T10.3:	
	024726	104402				TRAP	C\$BSUB
131	024730		CALL	\$RESET	;RESET THE DPV		
132	024734		ESCAPE	TST	;IF ERROR, EXIT THE TEST	TRAP	C\$ESCAPE
	024734	104410				.WORD	L10064-
	024736	000274					
133							
134	024740	005037	002376		CLR RFLAG ;CLEAR RECEIVE INTERRUPT		
135	024744	005037	002424		CLR TFLAG ;CLEAR TRANSMIT INTERRUPT FLAG.		
136	024750	005037	002360		CLR MCFLAG ;CLEAR MODEM CONTROL FLAG.		
137	024754	012737	000001	002414	MOV #1,START ;# OF START OF MESSAGES.		
138	024762	012737	000001	002316	MOV #1,ABORT ;FLAG TO SEND ABORT		
139							

```

140 024770          SETVEC  XMTVEC,#XINT,#PRI04
    024770 012746 000200          MOV  #PRI04,-(SP)
    024774 012746 017232          MOV  #XINT,-(SP)
    025000 013746 002264          MOV  XMTVEC,-(SP)
    025004 012746 000003          MOV  #3,-(SP)
    025010 104437          TRAP  C$SVEC
    025012 052706 000010          ADD  #10,SP
141 025016          SETVEC  RCVEC,#RINT,#PRI04
    025016 012746 000200          MOV  #PRI04,-(SP)
    025022 012746 016602          MOV  #RINT,-(SP)
    025026 013746 002262          MOV  RCVEC,-(SP)
    025032 012746 000003          MOV  #3,-(SP)
    025036 104437          TRAP  C$SVEC
    025040 062706 000010          ADD  #10,SP
142 025044          SETPRI  #PRI00          ;SET PROCESSOR PRIORITY.  FOR LSI 11/03
    025044 012700 000000          MOV  #PRI00,R0
    025050 104441          TRAP  C$SPRI
143          ;THIS WILL ENABLE INTERRUPTS.  FOR 11/23
144          ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
145          ;LEVEL 4-7.
146          ;SET UP INTERRUPT VECTOR
147
148 025052 052777 000130 155212      BIS  #TXIE!TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER AND SELECT
149          ;MAINTENANCE MODE LOOPBACK.
150 025060 052777 000120 155200      BIS  #RXITEN!RXENA,@RXCSR ;ENABLE THE RECEIVER
151
152 025066 005003          CLR  R3          ;INITIALIZE THE COUNTER
153 025070          5$:
154 025070 032777 004000 155170      BIT  #RXACT,@RXCSR ;IS THE RECEIVER ACTIVE?
155 025076 001007          BNE  10$          ;BR IF YES
156 025100 005303          DEC  R3          ;DECREMENT THE COUNTER
157 025102 001372          BNE  5$
158 025104          ERRDF  53,EMG12,ERRG2
    025104 104455          TRAP  C$ERDF
    025106 000065          .WORD  53
    025110 014062          .WORD  EMG12
    025112 006700          .WORD  ERRG2
159 025114 000435          BR  45$
160 025116          10$:
161 025116 005003          CLR  R3          ;INITIALIZE THE COUNTER.
162 025120          12$:
163 025120 032737 000002 002376      BIT  #2,RFLAG ;WAS STATUS RECEIVED?
164 025126 001007          BNE  20$
165 025130 005303          DEC  R3          ;DECREMENT THE COUNTER
166 025132 001372          BNE  12$
167 025134          ERRDF  54,EMG21,ERRG2
    025134 104455          TRAP  C$ERDF
    025136 000066          .WORD  54
    025140 014514          .WORD  EMG21
    025142 006700          .WORD  ERRG2
168 025144 000421          BR  45$
169
170 025146          20$:
171 025146 032737 002000 002400      BIT  #RABORT,RSAVE ;WAS THE RECEIVE ABORT RECEIVED?
172 025154 001005          BNE  40$          ;IF YES OK.
173 025156          ERRDF  55,EMG20,ERRG2
    025156 104455          TRAP  C$ERDF
  
```

	025160	000067				.WORD	55
	025162	014465				.WORD	EMG20
	025164	006700				.WORD	ERRG2
174	025166	00C410		BR	458		
175	025170		408:				
176	025170	032777	002000	155070	BIT	#RSTARY,ARXCSR	;IS THE STATUS DROPPED?
177	025176	001404		BEQ	458		
178							
179	025200			ERRDF	56,EMG18,ERRG2		
	025200	104455				TRAP	C\$ERDF
	025202	00C070				.WORD	56
	025204	014405				.WORD	EMG18
	025206	006700				.WORD	ERRG2
180	025210		458:				
181	025210			SETPRI	#PRI07		;SET PROCESSOR PRI TO 7
	025210	012700	000340			MOV	#PRI07,R0
	025214	104441				TRAP	C\$SPRI
182							
183	025216			CLRVEC	RCVEC		; (DISABLE INTERRUPT) ;RESTORE THE INTERRUPT VECTOR.
	025216	013700	002262			MOV	RCVEC,R0
	025222	104436				TRAP	C\$CJEC
184	025224	005037	002316	CLR	ABORT		;CLEAR THE ABORT FLAG.
185							
186							
187	025230			ENDSUB			
	025230					L10067:	
	025230	104403				TRAP	C\$ESUB
188							
189							
190	025232			ENDTST			
	025232					L10064:	
	025232	104401				TRAP	C\$E'TST
191							
192							
193							
194							
195							
196							
197							

```

1          .SBTTL          TEST 11 - RECEIVE AND TRANSMIT INTERRUPT
2
3          :*****
4          :          TEST 11 - DPV-11
5          :          * RECEIVE AND TRANSMIT INTERRUPT
6          :          * TRANSMIT AND RECEIVE DATA USING INTERRUPT ROUTINES. THIS TEST
7          :          * WILL TRANSMIT 4 DATA CHARACTERS. AFTER ENSURING THAT A TRANSMIT
8          :          * INTERRUPT WAS COMPLETED, THE TEST WILL CHECK TO MAKE SURE THAT AT
9          :          * LEAST 1 RECEIVE INTERRUPT WAS GENERATED.
10         :*****
11         :
12         025234          BGNTST
13
14         025234          CALL    $RESET          ;RESET THE DPV
15         025240          ESCAPE  TST            ;IF ERROR, EXIT THE TEST
16         025240          104410
17         025242          000234
18         025244          005037 002424
19         025250          005037 002376
20         025254          005037 002360
21         025260          012746 000200
22         025264          012746 016602
23         025270          013746 002262
24         025274          012746 000003
25         025300          104437
26         025302          062706 000010
27         025306          012746 000200
28         025312          012746 017232
29         025316          013746 002264
30         025322          012746 000003
31         025326          104437
32         025330          062706 000010
33         025334          012700 000000
34         025340          104441
35
36         025342          012777 043652 154720
37         025350          012737 000002 002414
38         025356          012777 000120 154702
39
40

```

111::  
 TRAP C\$ESCAPE  
 .WORD L10070-.  
 ;CLEAR THE FLAGS USED IN THE ISRS.  
 ;CLEAR MODEM CONTROL FLAG.  
 MOV #PRI04,-(SP)  
 MOV #RINT,-(SP)  
 MOV RCVEC,-(SP)  
 MOV #3,-(SP)  
 TRAP C\$SVEC  
 ADD #10,SP  
 MOV #PRI04,-(SP)  
 MOV #XINT,-(SP)  
 MOV XMTVEC,-(SP)  
 MOV #3,-(SP)  
 TRAP C\$SVEC  
 ADD #10,SP  
 ;SET PROCESSOR PRIORITY. FOR LSI 11/03  
 MOV #PRI00,RO  
 TRAP C\$SPRI  
 ;THIS WILL ENABLE INTERRUPTS. FOR 11/23  
 ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS  
 ;LEVEL 4-7.  
 ;SET UP INTERRUPT VECTOR  
 MOV #43652,@PCSR ;SET BCP MODE, NO ERROR AND SYNCH CHARACTER.  
 MOV #2,START ;# OF STARTS TO SEND.  
 MOV #RXITEN!RXENA,@RXCSR ;ENABLE THE RECEIVER AND SET  
 ;SET INTERRUPT ENABLE.  
 MOV #TXIE!TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER AND INT.  
 ;SELECT THE MAINTENANCE LOOPBACK.  
 CLR R1 ;LOOP COUNTER  
 10\$:  
 CMP #4,DATA ;ARE THE 4 DATA CHARACTERS RECEIVED?  
 BEQ 20\$ ;IF YES - CHECK RECEIVE INTERRUPT.  
 DEC R1 ;DECREMENT COUNTER  
 BNE 10\$

Line	Address	Offset	Label	Instruction	Comment	Trap	CS
41	025410	005737	002424	TST	TFLAG		
42	025414	001005		BNE	20\$		
43	025416			ERRDF	57,EMG9,ERRG2		
	025416	104455				TRAP	C\$ERDF
	025420	000071				.WORD	57
	025422	013723				.WORD	EMG9
	025424	006700				.WORD	ERRG2
44	025426	000410		BR	30\$		
45							
46	025430						
47	025430	032737	000001 002376	BIT	#1,RFLAG		
48	025436	001004		BNE	30\$		
49	025440			ERRDF	58,EMG15,ERRG2		
	025440	104455				TRAP	C\$ERDF
	025442	000072				.WORD	58
	025444	014243				.WORD	EMG15
	025446	006700				.WORD	ERRG2
50							
51	025450						
52	025450						
53	025454			CALL	\$RESET		
	025454	012700	000340	SETPRI	#PRI07		
	025460	104441				MOV	#PRI07,R0
						TRAP	C\$SPRI
54							
55	025462			CLRVEC	RCVEC		
	025462	013700	002262			MOV	RCVEC,R0
	025466	104436				TRAP	C\$CVEC
56	025470			CLRVEC	XMTVEC		
	025470	013700	002264			MOV	XMTVEC,R0
	025474	104436				TRAP	C\$CVEC
57							
58							
59							
60	025476			ENDTST			
	025476						
	025476	104401				TRAP	C\$ETST
61							
62							



```

1      .SBTTL          TEST 12 - MODEM STATUS
2
3      :*****
4      :*              TEST 12 - DPV-11
5      :* MODEM STATUS
6      :* IF A PROPER TURNAROUND (H3259 OR H3260) IS ON, THIS TEST WILL
7      :* CHECK THAT THE FOLLOWING MODEM SIGNALS ARE TURNED AROUND
8      :* 1. RTS (REQUEST TO SEND)      TURNED AROUND TO CTS (CLEAR TO SEND)
9      :*                                & RR (RECEIVER READY)
10     :* 2. DTR (DATA TERMINAL READY) TURNED AROUND TO IC (INCOMING CALL OR RING)
11     :* 3. SF (SELECT FREQUENCY)     TURNED AROUND TO SQ (SIGNAL QUALITY)
12     :* 4. LL (LOCAL LOOPBACK)       TURNED AROUND TO DM (DATA MODE)
13     :*
14     :*****
15     BGNTST
16     CALL    $TURN          ;CHECK THE TURNAROUND.
17     BCS     40$            ;SKIP TEST IF NO TURNAROUND.
18
19     CALL    $RESET          ;RESET THE DPV
20     ESCAPE  TST            ;IF ERROR, EXIT THE TEST
21
22     TRAP    C$ESCAPE
23     .WORD   L10071-
24
25     MOV     #RTS,R2         ;SAVE RTS IN REGISTER (FOR ERROR REPORT).
26     MOV     R2,@RXCSR      ;SET RTS
27     $DELAY  1              ;DELAY 100 MICROSECONDS
28
29     JSR     PC,$DLAY        ;***** MACRO EXPANSION *****
30     .WORD   1              ;CALL DELAY SUBROUTINE
31
32     BIT     #CTS,@RXCSR     ;IS CTS ON?
33     BEQ     10$            ;IF NOT - REPORT.
34     BIT     #RR,@RXCSR     ;IS RR (CD) ON
35     BEQ     10$            ;IF NOT - ERROR.
36     MOV     #DTR,R2        ;SAVE DTR IN REGISTER (FOR ERROR REPORT).
37     MOV     R2,@RXCSR      ;SET DTR.
38     $DELAY  1              ;DELAY 100 MICROSECONDS
39
40     JSR     PC,$DLAY        ;***** MACRO EXPANSION *****
41     .WORD   1              ;CALL DELAY SUBROUTINE
42
43     BIT     #IC,@RXCSR      ;IS RING (IC) SET?
44     BEQ     10$            ;IF NOT - ERROR.
45     MOV     #SF,R2         ;SAVE SF IN REGISTER (FOR ERROR REPORT).
46     MOV     R2,@RXCSR      ;SET REMOTE LOOP/ SEC FREQ
47     $DELAY  1              ;DELAY 100 MICROSECONDS
48
49     JSR     PC,$DLAY        ;***** MACRO EXPANSION *****
50     .WORD   1              ;CALL DELAY SUBROUTINE
51
52     BIT     #SQ,@TXCSR      ;IS SIGNAL QUALITY SET?

```

```

37 025626 001413      BEQ      10$      ;IF NOT - ERROR.
38 025630 012702 000010  MOV      #LL,R2      ;SAVE LL IN REGISTER (FOR ERROR REPORT).
39 025634 010277 154426  MOV      R2,@RXCSR    ;SET LOCAL LOOP
40 025640      $DELAY 1      ;DELAY 100 MICROSECONDS

      025640 004737 006604      JSR      PC,$DLAY      ;***** MACRO EXPANSION *****
      025644 000001      .WORD 1      ;CALL DELAY SUBROUTINE
      ;NUMBER OF DELAY LOOPS
      ;*****

41 025646 032777 001000 154412  BIT      #DM,@RXCSR    ;IS DATA MODE SET?
42 025654 001004      BNE      20$
43
44 025656      10$:
45 025656      ERRDF 59,EMG22,ERRG13
      025656 104455      TRAP      C$ERDF
      025660 000073      .WORD 59
      025662 014556      .WORD EMG22
      025664 010272      .WORD ERRG13

46
47 025666      20$:
48
49 025666 042777 000017 154372  BIC      #RTS!DTR!SF!LL,@RXCSR ;CLEAR ALL THOSE BITS
50 025674      $DELAY 1      ;DELAY 100 MICRO SECONDS

      025674 004737 006604      JSR      PC,$DLAY      ;***** MACRO EXPANSION *****
      025700 000001      .WORD 1      ;CALL DELAY SUBROUTINE
      ;NUMBER OF DELAY LOOPS
      ;*****

51
52 025702 012702 000004      MOV      #RTS,R2      ;CHECK RTS.
53 025706 030277 154354      BIT      R2,@RXCSR    ;IS IT SET?
54 025712 001021      BNE      30$      ;IF YES, ERROR.
55 025714 012702 000002      MOV      #DTR,R2      ;CHECK DTR.
56 025720 030277 154342      BIT      R2,@RXCSR    ;IS IT SET?
57 025724 001014      BNE      30$      ;IF YES, ERROR.
58 025726 012777 000001 154332  MOV      #SF,@RXCSR    ;CHECK SF.
59 025734 030277 154326      BIT      R2,@RXCSR    ;IS IT SET?
60 025740 001006      BNE      30$      ;IF YES, ERROR.
61 025742 012777 000010 154316  MOV      #LL,@RXCSR    ;CHECK LL
62 025750 030277 154312      BIT      R2,@RXCSR    ;IS IT SET?
63 025754 001404      BEQ      40$      ;IF NOT, OK
64 025756      30$:
65 025756      ERRDF 60,EMG22,ERRG15
      025756 104455      TRAP      C$ERDF
      025760 000074      .WORD 60
      025762 014556      .WORD EMG22
      025764 011044      .WORD ERRG15

66 025766      40$:
67 025766      ENDTST
      025766      L10071:
      025766 104401      TRAP      C$ETST

68
69
70
71

```

```

1      .SBTTL          TEST 13 - MODEM STATUS INTERRUPT
2
3      ;*****
4      ;*          TEST 13 - DVP-11
5      ;* MODEM STATUS INTERRUPT
6      ;* IF A PROPER TURNAROUND (H3259 OR H3260) IS ON, THIS TEST WILL CHECK
7      ;* THAT THE FOLLOWING SUBTESTS WORK CORRECTLY.
8      ;* SUBTEST 1 - SET DTR (DATA TERMINAL READY), LOCAL LOOP (LL), RTS (REQUEST
9      ;*                TO SEND) WITH ONLY RECEIVE INTERRUPT ENABLED. ENSURE THAT AN
10     ;*                INTERRUPT IS NOT RECEIVED.
11     ;* SUBTEST 2 - SET DTR, LL AND RTS WITH ONLY DATA SET INTERRUPT ENABLED.
12     ;*                ENSURE THAT AN INTERRUPT IS NOT RECEIVED.
13     ;* SUBTEST 3 - SET DTR, LL AND RTS WITHOUT ANY INTERRUPTS ENABLED. ENSURE
14     ;*                THAT AN INTERRUPT IS NOT RECEIVED.
15     ;* SUBTEST 4 - SET RTS WITH RECEIVE AND DATA SET INTERRUPT ENABLED. ENSURE
16     ;*                THAT AN INTERRUPT IS RECEIVED.
17     ;* SUBTEST 5 - SET DTR WITH RECEIVE AND DATA SET INTERRUPT ENABLED. ENSURE
18     ;*                THAT AN INTERRUPT IS RECEIVED.
19     ;* SUBTEST 6 - SET LL WITH RECEIVE AND DATA SET INTERRUPT ENABLED. ENSURE
20     ;*                THAT AN INTERRUPT IS RECEIVED.
21     ;*
22     ;*****
23     BGNTST
24
25     CALL    $TURN          ;CHECK THE TURNAROUND.
26     BCC     1$             ;PROCEED IF TURNAROUND.
27     EXIT    TST
28
29     1$:
30
31     SETVEC  RCVEC,#RINI,#PRI04
32
33     MOV     #PRI04,-(SP)
34     MOV     #RINT,-(SP)
35     MOV     RCVEC,-(SP)
36     MOV     #3,-(SP)
37     TRAP    C$SVEC
38     ADD     #10,SP
39
40     SETPRI  #PRI00          ;SET PROCESSOR PRIORITY. FOR LSI 11/03
41
42     MOV     #PRI00,R0
43     TRAP    C$SPRI
44
45     ;THIS WILL ENABLE INTERRUPTS. FOR 11/23
46     ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
47     ;LEVEL 4-7.
48     ;SET UP INTERRUPT VECTOR
49
50     BGNSUB
51
52     T13.1:
53     TRAP    C$BSUB
54
55     CALL    $RESET          ;RESET THE DVP
56     ESCAPE  TST             ;IF ERROR, EXIT THE TEST
57
58     TRAP    C$ESCAPE
59     .WORD   L10072-
60
61     CLR     RFLAG           ;CLEAR THE FLAG USED IN THE ISR
62     CLR     MCFLAG          ;CLEAR MODEM CONTROL FLAG.

```

```

43                                     ;ENABLE RCV INT AND SET RTS, DTR AND L. LOOP
44 026060 012777 000116 154200      MOV    #RXITEN!LL!DTR!RTS,@RXCSR
45
46 026066                               10$:
47 026066                               $DELAY 10                               ;WAIT 1 MS

                                ;***** MACRO EXPANSION *****
                                ;CALL DELAY SUBROUTINE
                                ;NUMBER OF DELAY LOOPS
                                ;*****

026066 004737 006604                JSR    PC,$DLAY
026072 000010                .WORD    10

48 026074 005737 002360                TST    MCFLAG                ;WAS AN MODEM CONTROL INTERRUPT RECEIVED?
49 026100 001404                BEQ    30$                ;IF NOT OK.
50 026102                ERRDF    61,EMG23,ERRG2

                                TRAP    C$ERDF
                                .WORD    61
                                .WORD    EMG23
                                .WORD    ERRG2

51
52 026112                               30$:
53
54 026112                ENDSUB
026112
026112 104403                L10073:
                                TRAP    C$ESUB

55
56
57 026114                BGNSUB
026114
026114 104402                T13.2:
                                TRAP    C$BSUB

58 026116                CALL    $RESET                ;RESET THE DPV
59 026122                ESCAPE  TST                ;IF ERROR, EXIT THE TEST

                                TRAP    C$ESCAPE
                                .WORD    L10072-.

026122 104410
026124 000552
60 026126 005037 002376                CLR    RFLAG                ;CLEAR THE FLAG USED IN THE ISR
61 026132 005037 002360                CLR    MCFLAG                ;CLEAR MODEM CONTROL FLAG.

62
63                                     ;ENABLE DS. INT, SET RTS, DTR AND LL
64 026136 012777 000056 154122      MOV    #DSITEN!LL!RTS!DTR,@RYCSR
65
66 026144                               10$:
67 026144                               $DELAY 10                               ;WAIT 1 MS

                                ;***** MACRO EXPANSION *****
                                ;CALL DELAY SUBROUTINE
                                ;NUMBER OF DELAY LOOPS
                                ;*****

026144 004737 006604                JSR    PC,$DLAY
026150 000010                .WORD    10

68 026152 005737 002360                TST    MCFLAG                ;WAS AN MODEM CONTROL INTERRUPT RECEIVED?
69 026156 001404                BEQ    30$                ;IF NOT OK.
70 026160                ERRDF    62,EMG23,ERRG2

                                TRAP    C$ERDF
                                .WORD    62
                                .WORD    EMG23
                                .WORD    ERRG2

026160 104455
026162 000076
026164 014603
026166 006700

71
72 026170                               30$:
73

```

```

74 026170          ENDSUB                                L10074:
    026170          TRAP                                C$ESUB
    026170 104403
75
76
77
78 026172          BGNSUB                                T13.3:
    026172          TRAP                                C$BSUB
    026172 104402
79 026174          CALL $RESET                          ;RESET THE DPV
80 026200          ESCAPE TST                          ;IF ERROR, EXIT THE TEST
    026200 104410          TRAP                                C$ESCAPE
    026202 000474          .WORD                        L10072-.
81 026204 005037 002376          CLR RFLAG              ;CLEAR THE FLAG USED IN THE ISR
82 026210 005037 002360          CLR MCFLAG             ;CLEAR MODEM CONTROL FLAG.
83
84
85 026214 012777 000016 154044 10$: MOV #LL!RTS!DTR,@RXCSR ;SET LOCAL LOOP, DTR AND RTS.
86 026222
87 026222          $DELAY 10                          ;WAIT 1 MS

          026222 004737 006604          JSR PC,$DLAY      ;***** MACRO EXPANSION *****
          026226 000010          .WORD 10              ;CALL DELAY SUBROUTINE
          ;NUMBER OF DELAY LOOPS
          ;*****

88 026230 005737 002360          TST MCFLAG             ;WAS AN INTERRUPT RECEIVED?
89 026234 001404          BEQ 30$                      ;IF NOT OK.
90 026236          ERRDF 63,EMG23,ERRG2
    026236 104455          TRAP                                C$ERRDF
    026240 000077          .WORD 63
    026242 014603          .WORD EMG23
    026244 006700          .WORD ERRG2
91
92 026246          30$:
93
94 026246          ENDSUB                                L10075:
    026246          TRAP                                C$ESUB
    026246 104403
95
96
97 026250          BGNSUB                                T13.4:
    026250          TRAP                                C$BSUB
    026250 104402
98 026252          CALL $RESET                          ;RESET THE DPV
99 026256          ESCAPE TST                          ;IF ERROR, EXIT THE TEST
    026256 104410          TRAP                                C$ESCAPE
    026260 000416          .WORD                        L10072-.
100 026262 005037 002376          CLR RFLAG              ;CLEAR THE FLAG USED IN THE ISR
101 026266 005037 002360          CLR MCFLAG             ;CLEAR MODEM CONTROL FLAG.
102
103
104 026272 012777 000144 153766 10$: MOV #RXITEN!DSITEN!RTS,@RXCSR ;ENABLE INTERRUPTS AND SET RTS.
105
106 026300          $DELAY 10                          ;WAIT 1 MS
107 026300
  
```

```

026300 004737 006604      JSR      PC,$DLAY      ;***** MACRO EXPANSION *****
026304 000010                      ;CALL DELAY SUBROUTINE
                                      ;NUMBER OF DELAY LOOPS
                                      ;*****
108 026306 005737 002360      TST      MCFLAG      ;WAS AN INTERRUPT RECEIVED?
109 026312 001015                      BNE      20$      ;IF YES - CHECK FOR MULTIPLE INTERRUPTS.
110 026314                      ERRDF    64,EMG24,ERRG2
                                      TRAP    C$ERDF
                                      .WORD   64
                                      .WORD   EMG24
                                      .WORD   ERRG2
111 026314 104455                      MOV     #FMG26,-(SP)
026316 000100                      MOV     #1,-(SP)
026320 014661                      MOV     SP,R0
026322 006700                      TRAP    C$PNTB
111 026324                      ADD     #4,SP
026324 012746 012715      PRINTB  #FMG26
026330 012746 000001
026334 010600
026336 104414
026340 062706 000004
112 026344 000410      BR      30$
113 026346
114 026346 022737 000001 002360 20$:  CMP     #1,MCFLAG      ;WAS ONLY 1 RECEIVED?
115 026354 001404                      BEQ     30$      ;IF YES - OK
116 026356                      ERRDF    65,EMG40      ;REPORT MULTIPLE INTERRUPTS
                                      TRAP    C$ERDF
                                      .WORD   65
                                      .WORD   EMG40
                                      .WORD   0
026356 104455
026360 000101
026362 015321
026364 000000
117 026366      30$:
118
119 026366      ENDSUB
026366
026366 104403      L10076:  TRAP    C$ESUB
120
121
122 026370      BGNSUB
026370
026370 104402      T13.5:  TRAP    C$BSUB
123 026372      CALL    $RESET      ;RESET THE DVP
124 026376      ESCAPE  TST      ;IF ERROR, EXIT THE TEST
                                      TRAP    C$ESCAPE
                                      .WORD   L10072-.
026376 104410
026400 000276
125 026402 005037 002376      CLR     RFLAG      ;CLEAR THE FLAG USED IN THE ISR
126 026406 005037 002360      CLR     MCFLAG      ;CLEAR MODEM CONTROL FLAG.
127
128                      ;ENABLE INTERRUPTS AND SET DTR.
129 026412 012777 000142 153646      MOV     #RXITEN!DSITEN.DTR,$RXCSR
130
131 026420      10$:
132 026420      $DELAY  10      ;WAIT 1 MS
                                      ;***** MACRO EXPANSION *****
                                      ;CALL DELAY SUBROUTINE
                                      ;NUMBER OF DELAY LOOPS
                                      ;*****
133 026426 005737 002360      TST      MCFLAG      ;WAS AN INTERRUPT RECEIVED?
134 026432 001015                      BNE      20$      ;IF YES - CHECK FOR MULTIPLE INTERRUPTS.

```

```

135 026434          ERRDF  66,EMG24,ERRG2
    026434 104455
    026436 000102
    026440 014661
    026442 006700
136 026444          PRINTB #FMG26
    026444 012746 012715
    026450 012746 000001
    026454 010600
    026456 104414
    026460 062706 000004
137 026464 000410
138 026466
139 026466 022737 000001 002360
140 026474 001404
141 026476          ERRDF  67,EMG40
    026476 104455
    026500 000103
    026502 015321
    026504 000000
142 026506
143
144 026506          ENDSUB
    026506
    026506 104403
145
146
147 026510          BGNSUB
    026510
    026510 104402
148 026512          CALL  $RESET
149 026516          ESCAPE TST
    026516 104410
    026520 000156
150 026522 005037 002376
151 026526 005037 002360
152
153
154 026532 012777 000150 153526
155
156 026540
157 026540
    026540 004737 006604
    026544 000010
158 026546 005737 002360
159 026552 001025
160 026554          ERRDF  68,EMG24,ERRG2
    026554 104455
    026556 000104
    026560 014661
    026562 006700
161 026564          PRINTB #FMG26
    026564 012746 012715

```

20\$:  
 30\$:  
 30\$:  
 10\$:

;WAS ONLY 1 RECEIVED?  
 ;IF YES - OK  
 ;REPORT MULTIPLE INTERRUPTS

;RESET THE DPV  
 ;IF ERROR, EXIT THE TEST

;CLEAR THE FLAG USED IN THE ISR  
 ;CLEAR MODEM CONTROL FLAG.

;ENABLE INTERRUPTS AND SET LL.

;WAIT 1 MS

;\*\*\*\*\* MACRO EXPANSION \*\*\*\*\*  
 ;CALL DELAY SUBROUTINE  
 ;NUMBER OF DELAY LOOPS  
 ;\*\*\*\*\*

;WAS AN INTERRUPT RECEIVED?  
 ;IF YES - CHECK FOR MULTIPLE INTERRUPTS.

026570	012746	000001				MOV	#1,-(SP)
026574	010600					MOV	SP,R0
026576	104414					TRAP	C\$PNTB
026600	062706	000004				ADD	#4,SP
162 026604			PRINTB	#FMG29			
026604	012746	013221				MOV	#FMG29,-(SP)
026610	012746	000001				MOV	#1,-(SP)
026614	010600					MOV	SP,R0
026616	104414					TRAP	C\$PNTB
026620	062706	000004				ADD	#4,SP
163 026624	000410		BR	30\$			
164 026626			20\$:				
165 026626	022737	000001	002360	CMP	#1,MCFLAG	;WAS ONLY 1 RECEIVED?	
166 026634	001404			BEQ	30\$	;IF YES - OK	
167 026636				ERRDF	69,EMG40	;REPORT MULTIPLE INTERRUPTS	
026636	104455					TRAP	C\$ERDF
026640	000105					.WORD	69
026642	015321					.WORD	EMG40
026644	000000					.WORD	0
168 026646			30\$:				
169			ENDSUB				
170 026646						L10100:	
026646						TRAP	C\$ESUB
026646	104403						
171							
172							
173							
174 026650			CALL	\$RESET		;RESET THE DPV	
175 026654			SETPRI	#PRI07		;SET THE PROCESSOR PRI TO 7	
026654	012700	000340				MOV	#PRI07,R0
026660	104441					TRAP	C\$SPRI
176						;(THIS WILL DISABLE INTERRUPTS)	
177 026662			CLRVEC	RCVEC		;RESTORE THE RECV. VECTOR	
026662	013700	002262				MOV	RCVEC,R0
026666	104436					TRAP	C\$CVEC
178 026670			CLRVEC	XMTVEC		;RESTORE THE XMIT. VECTOR	
026670	013700	002264				MOV	XMTVEC,R0
026674	104436					TRAP	C\$CVEC
179							
180							
181 026676			ENDTST			L10072:	
026676						TRAP	C\$ETST
026676	104401						
182							
183							
184							



```

1      .SBTTL          TEST 14 - RECEIVE AND MODEM STATUS INTERRUPTS
2
3      :*****
4      :               TEST 14 - DPV-11
5      : * RECEIVE AND MODEM STATUS INTERRUPTS
6      : * CHANGE THE MODEM STATUS WHILE HANDLING A RECEIVE INTERRUPT.
7      : * ENSURE THAT THE MODEM STATUS INTERRUPT IS RECEIVED.
8      : * SUBTEST 1 - CHANGE RTS DURING THE RECEIVE INTERRUPT. ENSURE THAT
9      : *               THE DATA SET INTERRUPT WAS RECEIVED.
10     : * SUBTEST 2 - CHANGE DTR DURING THE RECEIVE INTERRUPT. ENSURE THAT
11     : *               THE DATA SET INTERRUPT WAS RECEIVED.
12     : * SUBTEST 3 - CHANGE LL DURING THE RECEIVE INTERRUPT. ENSURE THAT
13     : *               THE DATA SET INTERRUPT WAS RECEIVED.
14     :*****
15
16     026700          BGNTST
17     026700
18     026704          1C3002          CALL    $TURN          ;CHECK THE TURNAROUND.
19     026706          104432          BCC     1$              ;PROCEED, IF TURNAROUND ON.
20     026710          001072          EXIT    TST              ;IF NO TURNAROUND, EXIT.
21     026712          1$:              TRAP    C$EXIT
22     026712          BGNSUB              .WORD    L10101-.
23     026712          104402          T14.1:          TRAP    C$BSUB
24     026714          CALL    $RESET          ;RESET THE DPV
25     026720          ESCAPE  TST              ;IF ERROR, EXIT THE TEST
26     026722          104410          TRAP    C$ESCAPE
27     026724          005037 002424          .WORD    L10101-.
28     026730          005037 002376          CLR     TFLAG          ;CLEAR THE FLAGS USED IN THE ISRS.
29     026734          005037 002360          CLR     RFLAG
30     026740          012737 000004 002432    CLR     MCFLAG          ;CLEAR MODEM CONTROL FLAG.
31     026746          SETVEC  RCVEC,#RINT,#PRI04          ;TOGGLE RTS
32     026752          012746 000200          MOV     #RTS,TOGGLE
33     026756          013746 002262          SETVEC  RCVEC,#RINT,#PRI04
34     026762          012746 000003          MOV     #PRI04,-(SP)
35     026766          104437          MOV     #RINT,-(SP)
36     026770          062706 000010          MOV     RCVEC,-(SP)
37     026774          012746 000200          MOV     #3,-(SP)
38     027000          012746 017232          TRAP    C$SVEC
39     027004          013746 002264          ADD     #10,SP
40     027010          012746 000003          MOV     #PRI04,-(SP)
41     027014          104437          MOV     #XINT,-(SP)
42     027016          062706 000010          MOV     XMTVEC,-(SP)
43     027022          012700 000000          MOV     #3,-(SP)
44     027026          104441          TRAP    C$SVEC
45
46     ;SET PROCESSOR PRIORITY. FOR LSI 11/03
47     ;THIS WILL ENABLE INTERRUPTS. FOR 11/23
48     ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
49     ;LEVEL 4-7.
50     ;SET UP INTERRUPT VECTOR

```

```

37
38 027030 012777 043652 153232      MOV    #43652,@PCSR      ;SET BCP MODE, NO ERROR AND SYNCH CHARACTER.
39 027036 012737 000002 002414      MOV    #2,START        ;# OF START CHARACTERS.
40 027044 012777 000160 153214      MOV    #RXITEN!DSITEN!RXENA,@RXCSR ;ENABLE THE RECEIVER AND INT.
41 027052 012777 000130 153212      MOV    #TXIE!TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER AND INT.
42                                     ;SELECT THE MAINTENANCE LOOPBACK.
43 027060 005001                      CLR    R1                ;LOOP COUNTER
44 027062                                     10$:
45 027062 005737 002360              TST    MCFLAG              ;WAS A MODEM CHANGE INTERRUPT RECEIVED?
46 027066 001017                      BNE    20$                ;IF YES, EXIT.
47 027070 005301                      DEC    R1                ;DECREMENT COUNTER
48 027072 001373                      BNE    10$
49
50 027074                      ERRDF    70,EMG24,ERRG2
51                                     TRAP    C$ERDF
52 027074 104455                                     .WORD    70
53 027076 000106                                     .WORD    EMG24
54 027100 014661                                     .WORD    ERRG2
55 027102 006700
56
57 027104                      PRINTB    #FMG26                ;NOTIFY THAT INTERRUPT MAY BE DISABLED BY
58 027104 012746 012715              MOV    #FMG26,-(SP)
59 027110 012746 000001              MOV    #1,-(SP)
60 027114 010600              MOV    SP,R0
61 027116 104414              TRAP    C$PNTB
62 027120 062706 000004              ADD    #4,SP
63
64 027124 000410                      BR     30$                ;REMOVING THE WIRE WRAP.
65 027126                                     20$:
66 027126 022737 000001 002360      CMP    #1,MCFLAG        ;WAS ONLY 1 RECEIVED?
67 027134 001404                      BEQ    30$                ;IF YES - OK
68 027136                      ERRDF    71,EMG40              ;REPORT MULTIPLE INTERRUPTS
69 027136 104455                                     TRAP    C$ERDF
70 027140 000107                                     .WORD    71
71 027142 015321                                     .WORD    EMG40
72 027144 000000                                     .WORD    0
73
74 027146                                     30$:
75 027146                      CALL    $RESET                ;RESET THE DPV
76 027152                      SETPRI    #PRI07              ;SET THE PROCESSOR PRI TO 7
77 027152 012700 000340              MOV    #PRI07,R0
78 027156 104441                      TRAP    C$SPRI
79
80 027160                      CLRVEC    RCVEC              ;(THIS WILL DISABLE INTERRUPTS)
81 027160 013700 002262              ;RESTORE THE RECV. VECTOR
82 027164 104436                      MOV    RCVEC,R0
83 027166                      CLRVEC    XMTVEC              ;RESTORE THE XMIT. VECTOR
84 027166 013700 002264              MOV    XMTVEC,R0
85 027172 104436                      TRAP    C$CVEC
86
87 027174                      F$CAPE    TST                ;IF ERROR, ESCAPE
88 027174 104410                      TRAP    C$ESCAPE
89 027176 000604                      .WORD    L10101-.
90
91 027200                      ENDSUB
92 027200                                     L10102:
93 027200 104403                      TRAP    C$ESUB
94
95 027202                      BGNSUB
96 027202                                     T14.2:

```

70	027202	104402					TRAP	C\$BSUB
71	027204							
	027210							
	027210	104410					TRAP	C\$ESCAPE
	027212	000570					.WORD	L10101-
72	027214	005037	002424					
73	027220	005037	002376					
74	027224	005037	002360					
75	027230	012737	000002	002432				
76								
77	027236							
	027236	012746	000200				MOV	#PRI04,-(SP)
	027242	012746	016602				MOV	#RINT,-(SP)
	027246	013746	002262				MOV	RCVEC,-(SP)
	027252	012746	000003				MOV	#3,-(SP)
	027256	104437					TRAP	C\$SVEC
	027260	062706	000010				ADD	#10,SP
78	027264							
	027264	012746	000200				MOV	#PRI04,-(SP)
	027270	012746	017232				MOV	#XINT,-(SP)
	027274	013746	002264				MOV	XMTVEC,-(SP)
	027300	012746	000003				MOV	#3,-(SP)
	027304	104437					TRAP	C\$SVEC
	027306	062706	000010				ADD	#10,SP
79	027312							
	027312	012700	000000				MOV	#PRI00,RO
	027316	104441					TRAP	C\$SPRI
80								
81								
82								
83								
84								
85								
86	027320	012777	043652	152742				
87	027326	012737	000002	002414				
88	027334	012777	000160	152724				
89	027342	012777	000130	152722				
90								
91	027350	005001						
92	027352							
93	027352	005737	002360					
94	027356	001017						
95	027360	005301						
96	027362	001373						
97								
98	027364							
	027364	104455					TRAP	C\$ERDF
	027366	000110					.WORD	72
	027370	014661					.WORD	EMG24
	027372	006700					.WORD	ERRG2
99								
100	027374							
	027374	012746	012715				MOV	#FMG26,-(SP)
	027400	012746	000001				MOV	#1,-(SP)
	027404	010600					MOV	SP,RO
	027406	104414					TRAP	C\$PNTB
	027410	062706	000004				ADD	#4,SP

;ALL \$RESET ;RESET THE DPV  
 ESCAPE TST ;IF ERROR, EXIT THE TEST  
 ;CLEAR THE FLAGS USED IN THE ISRS.  
 CLR TFLAG  
 CLP RFLAG  
 CLR MCFLAG ;CLEAR MODEM CONTROL FLAG.  
 MOV #DTR,TOGGLE ;TOGGLE DTR.  
 SETVEC RCVEC,#RINT,#PRI04  
 SETVEC XMTVEC,#XINT,#PRI04  
 SETPRI #PRI00 ;SET PROCESSOR PRIORITY. FOR LSI 11/03  
 ;THIS WILL ENABLE INTERRUPTS. FOR 11/23  
 ;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS  
 ;LEVEL 4-7.  
 ;SET UP INTERRUPT VECTOR  
 MOV #43652,@PCSR ;SET BCP MODE, NO ERROR AND SYNCH CHARACTER.  
 MOV #2,START ;# OF START CHARACTERS.  
 MOV #RXITEN!DSITEN!RXENA,@RXCSR ;ENABLE THE RECEIVER AND INT.  
 MOV #TXIE!TXENA!MM,@TXCSR ;ENABLE THE TRANSMITTER AND INT.  
 ;SELECT THE MAINTENANCE LOOPBACK.  
 CLR R1 ;LOOP COUNTER  
 10\$-  
 TST MCFLAG ;WAS A MODEM CHANGE INTERRUPT RECEIVED?  
 BNE 20\$ ;IF YES, EXIT.  
 DEC R1 ;DECREMENT COUNTER  
 BNE 10\$  
 ERDF 72,EMG24,ERRG2  
 PRINTB #FMG26 ;NOTIFY THAT INTERRUPT MAY BE DISABLED BY

```

101                                     ;REMOVING THE WIRE WRAP.
102 027414 000410      BR      30$
103 027416                                     20$:
104 027416 022737 000001 002360      CMP      #1,MCFLAG      ;WAS ONLY 1 RECEIVED?
105 027424 001404      BEQ      30$      ;IF YES - OK
106 027426      ERDF      73,EMG40      ;REPORT MULTIPLE INTERRUPTS
      027426 104455      TRAP      C$ERDF
      027430 000111      .WORD      73
      027432 015321      .WORD      EMG40
      027434 000000      .WORD      0
107 027436                                     30$:
108 027436      CALL      $RESET      ;RESET THE DPV
109 027442      SETPRI      #PRI07      ;SET THE PROCESSOR PRI TO 7
      027442 012700 000340      MOV      #PRI07,RO
      027446 104441      TRAP      C$SPRI
110                                     ;(THIS WILL DISABLE INTERRUPTS)
111      CLRVEC      RCVEC      ;RESTORE THE RECV. VECTOR
      027450 013700 002262      MOV      RCVEC,RO
      027454 104436      TRAP      C$CVEC
112      CLRVEC      XMTVEC      ;RESTORE THE XMIT. VECTOR
      027456 013700 002264      MOV      XMTVEC,RO
      027462 104436      TRAP      C$CVEC
113      ESCAPE      TST      ;IF ERROR, ESCAPE
      027464 104410      TRAP      C$ESCAPE
      027466 000314      .WORD      L10101-.
114
115      ENDSUB
      027470
      027470 104403      L10103: TRAP      C$ESUB
116
117      BGNSUB
      027472
      027472 104402      T14.3: TRAP      C$BSUB
118      CALL      $RESET      ;RESET THE DPV
119      ESCAPE      TST      ;IF ERROR, EXIT THE TEST
      027500 104410      TRAP      C$ESCAPE
      027502 000300      .WORD      L10101-.
120 027504 005037 002424      CLR      TFLAG      ;CLEAR THE FLAGS USED IN THE ISRS.
121 027510 005037 002376      CLR      RFLAG
122 027514 005037 002360      CLR      MCFLAG      ;CLEAR MODEM CONTROL FLAG.
123 027520 012737 000010 002432      MOV      #LL,TOGGLE      ;TOGGLE LL
124
125      SETVEC      RCVEC,#RINT,#PRI04
      027526 012746 000200      MOV      #PRI04,-(SP)
      027532 012746 016602      MOV      #RINT,-(SP)
      027536 013746 002262      MOV      RCVEC,-(SP)
      027542 012746 000003      MOV      #3,-(SP)
      027546 104437      TRAP      C$SVEC
      027550 062706 000010      ADD      #10,SP
126      SETVEC      XMTVEC,#XINT,#PRI04
      027554 012746 000200      MOV      #PRI04,-(SP)
      027560 012746 017232      MOV      #XINT,-(SP)
      027564 013746 002264      MOV      XMTVEC,-(SP)
      027570 012746 000003      MOV      #3,-(SP)
      027574 104437      TRAP      C$SVEC
      027576 062706 000010      ADD      #10,SP
127 027602      SETPRI      #PRI00      ;SET PROCESSOR PRIORITY. FOR LSI 11/03

```

```

027602 012700 000000
027606 104441
128
129
130
131
132
133
134 027610 012777 043652 152452
135 027616 012737 000002 002414
136 027624 012777 000160 152434
137 027632 012777 000130 152432
138
139 027640 005001
140 027642
141 027642 005737 002360
142 027646 001027
143 027650 005301
144 027652 001373
145
146 027654
027654 104455
027656 000112
027660 014661
027662 006700
147
148 027664
027664 012746 012715
027670 012746 000001
027674 010600
027676 104414
027700 062706 000004
149
150 027704
027704 012746 013221
027710 012746 000001
027714 010600
027716 104414
027720 062706 000004
151 027724 000410
152 027726
153 027726 022737 000001 002360
154 027734 001404
155 027736
027736 104455
027740 000113
027742 015321
027744 000000
156 027746
157 027746
158 027752

;THIS WILL ENABLE INTERRUPTS. FOR 11/23
;THIS WILL ALLOW ACKNOWLEDGMENT OF INTERRUPTS
;LEVEL 4-7.
;SET UP INTERRUPT VECTOR

MOV #PRI00,R0
TRAP C$SPRI

;SET BCP MODE, NO ERROR AND SYNCH CHARACTER.
;# OF START CHARACTERS.
MOV #43652,@PCSR
MOV #2,START
MOV #RXITEN!DSITEN!RXENA,@RXCSR
MOV #TXIE!TXENA!MM,@TXCSR
;ENABLE THE RECEIVER AND INT.
;ENABLE THE TRANSMITTER AND INT.
;SELECT THE MAINTENANCE LOOPBACK.
CLR R1
;LOOP COUNTER
10$:
TST MCFLAG
BNF 20$
DEC R1
BNE 10$
;WAS A MODEM CHANGE INTERRUPT RECEIVED?
;IF YES, EXIT.
;DECREMENT COUNTER
ERRDF 74,EMG24,ERRG2
TRAP C$ERDF
WORD 74
WORD EMG24
WORD ERRG2

PRINTB #FMG26
;NOTIFY THAT INTERRUPT MAY BE DISABLED BY
MOV #FMG26,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP

PRINTB #FMG29
;REMOVING THE WIRE WRAP.
MOV #FMG29,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP

BR 30$
20$:
CMP #1,MCFLAG
BEQ 30$
ERRDF 75,EMG40
;WAS ONLY 1 RECEIVED?
;IF YES - OK
;REPORT MULTIPLE INTERRUPTS
TRAP C$ERDF
WORD 75
WORD EMG40
WORD 0

30$:
CALL $RESET
SETPRI #PRI07
;RESET THE DPV
;SET THE PROCESSOR PRI TO 7

```