

DL11-W

DL11-W/1144 MFM SLU
CZDLDD0

AH-8529D-MC

COPYRIGHT 75-80

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA

This microfiche card contains a grid of frames, each displaying a different set of data. The data is organized into columns and rows, with some frames showing binary code (0s and 1s) and others showing more complex patterns of characters and symbols. The frames are arranged in a regular grid pattern across the card.

.REM @

IDENTIFICATION

PRODUCT CODE: AC-8528D-MC
PRODUCT NAME: C2DLDD0 DL11-W/1144 MFM SLU
DATE CREATED: MARCH 1979
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: DAN CASALETIO
REVISED BY: JOHN CIJKAJ

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) 1975, 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

HISTORY SECTION

CZDLDDO WAS RELEASED OCT 79

THE FOLLOWING CHANGES WERE MADE. ALL CHANGES ARE INDICATED
BY ;** IN THE COMMENT FIELD:

1. USE THE MFPT INSTRUCTION, AND IF THE CPU IS A 11/44:
 - A. DO NOT PERFORM READER ENABLE AND RECEIVER ACTIVE TESTS.
 - B. IF ERROR FLAGS TESTS AND BREAK TESTS ARE ENABLED, PERFORM THESE TESTS ONLY FOR THE SLU AT 176500. PERFORM THESE TESTS FOR THE CONSOLE SLU IF BIT03 OF SWR IS ADDITIONALLY SET.
 - C. WHILE IN MAINTENANCE MODE DO NOT WRITE AND READ A ^P OR AN ASCII 220. THIS WILL FORCE THE CONSOLE INTO 'CONSOLE MODE'.
 - D. IN THE CLOCK REPEATABILITY TEST, ALLOW FOR A TOLERANCE OF 2 BETWEEN CLOCK COUNTS. THIS IS TO ALLOW ENOUGH TOLERANCE WHEN MOS MEMORY IS USED AND REFRESH CYCLES ARE OCCURRING.
2. BECAUSE 11/44 SLU INTERRUPT REQUESTS ARE DEPENDANT ON A MFM CLOCK ENOUGH TIME MUST BE GIVEN FOR THEM TO OCCUR. THEREFORE, ALL TESTS ASSOCIATED WITH XMIT & RECEIVE INTERRUPTS SHOULD HAVE A MINIMUM OF 4 NOP'S ACTING AS WAIT FOR INTERRUPT.
3. IT WAS FOUND THAT AN ATTACHED TU58 WOULD BE ACTIVATED BY THE DIAGNOSTIC, AND, AS A RESULT, CHARACTERS WOULD BE SENT TO THE RECEIVER BUFFER. THIS WOULD CAUSE SOME TESTS TO FAIL. THEREFORE TO INHIBIT ANY COMMUNICATION TO THE TU58 FROM THE DIAGNOSTIC:
 - A. ENABLE MAINTENANCE MODE IN ALL TESTS THAT WRITE TO THE XMITTER.
 - B. IN ALL TESTS THAT WITE TO XMITTER AND BEFORE LEAVING TEST: DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS THAT MIGHT BE IN THE PROCESS OF BEING XMITTED TO FINISH.
 - C. DISABLE MAINTENANCE MODE FOR PURPOSE OF ERROR PRINTING ONLY TO THE SLU WHICH THE TERMINAL IS ATTACHED.
4. IT WAS FOUND THAT UPON POWERUP AND WITH THE TU58 ATTACHED, CHARACTERS WOULD BE SENT TO THE RECEIVER FROM THE TU58. SOME RECEIVER TESTS WOULD FAIL DUE TO THIS. THEREFORE, ON ALL TESTS THAT PERFORM RECEIVER TESTS AND FOLLOWING MAINTENANCE MODE BEING ENABLED, DELAY ENOUGH TIME TO ALLOW TO ALLOW ANY CHARACTERS THAT MIGHT BE IN THE PROCESS OF BEING RECEIVED TO FINISH.
5. ADD SOFTWARE THAT WILL IMPLEMENT AUTO INITIATION OF 11/44 T/A CONSOLE TEST VIA WRAP CABLE FROM TU58 TO CONSOLE

PORTS. IT IS SELECTED BY SWR BIT02 AND IS PERFORMED
ONLY AFTER ALL SLUS ARE TESTED.

1.0 GENERAL INFORMATION

1.1 ABSTRACT

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE
THE FOLLOWING MODULES:

1. DL11-W SERIAL LINE/REAL TIME CLOCK INTERFACE
2. 11/44 MULTIFUNCTION MODULE

THE PROGRAM WILL RUN WITHOUT ANY SPECIAL TEST FIXTURES BY DEFAULT
HOWEVER, THE FOLLOWING OPTIONAL TESTING IS PROVIDED:

1. TEST TO VERIFY XMIT AND RECEIVE OF THE UARTS WITH
A WRAP CABLE. THE UART UNDER TEST IS LOOPED BACK ON
ITSELF. THIS IS SELECTED VIA OPERATIONAL SWITCH SETTING
BIT 7, AND IS APPLICABLE TO THE DL11W AND 11/44 MFM.
2. AUTOMATIC INITIATION OF THE T/A CONSOLE TEST OF THE 11/44
MFM. THE TU58 PORTS ARE LOOPED TO THE CONSOLE PORTS
THIS IS SELECTED VIA OPERATIONAL SWITCH SETTING BIT 2
AND IS APPLICABLE TO 11/44 MFM ONLY.
(SEE CKKFBA0 FOR T/A CONSOLE TEST EXPLANATION)

THIS DIAGNOSTIC OPERATES ON THE CONSOLE SERIAL LINE AND CLOCK INTERFACES
AS WELL AS UP TO FIFTEEN(15) ADDITIONAL IDENTICALLY CONFIGURED
SERIAL LINE INTERFACES. THE DEFAULT ADDRESSES ARE:

A. CONSOLE - 177560 SERIAL LINE
177546 CLOCK

B. OTHER SERIAL LINE - 776500 FIRST SERIAL LINE ADDRESS
OF 15 CONSECUTIVE SERIAL
LINE ADDRESSES

THE PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 8K OF MEMORY
. IT CAN BE RUN UNDER XXDP, APT, AND ACT MONITORS,
AND ON PROCESSORS WITH NO HARDWARE SWITCH REGISTER,
SOFTWARE SWITCH REGISTER = LOCATION 1/6

POWER FAILURE IS SUPPORTED FOR SYSTEMS WITH CORE MEMORY.

NOTE: THIS DIAGNOSTIC WITH THE SWR - 000020
(CLOCK TESTS ONLY) SHOULD BE USED ON SWITCHLESS CPU'S
TO TEST KW-11L LINE CLOCK MODULES.

1.2 SYSTEM REQUIREMENTS

1.2.1 EQUIPMENT

STANDARD 11 FAMILY (COMPUTER WITH A CONSOLE OUTPUT DEVICE

AND 8K OF MEMORY.

OPTIONAL:

1. LOOP CABLE FOR UART LOOP BACK ON ITSELF
2. WRAP CABLE FOR 11/44 MFM T/E TESTING
(SEE DWG. #7016942 WRAP AROUND CABLE
AND SECTION 5.0 OF THIS DOCUMENT)

1.2.2 STORAGE

THE PROGRAM USES 5K WORDS OF MEMORY

1.3 ASSUMPTIONS

- A. IF THE UNIT UNDER TEST (UUT) IS THE CONSOLE, THE PROGRAM WILL ASSUME THE REAL TIME CLOCK (RTC) IS ENABLED AND WILL TEST IT UNLESS THE TESTS ARE DISABLED BY BIT6 OF THE SWR.
- B. IF THE UUT IS NOT THE CONSOLE, THE RTC IS NOT TESTED FOR THAT DEVICE.
- C. FOR THE DL11-W:
- THE PROGRAM WILL ASSUME THE ERROR FLAG BITS AND THE BREAK FUNCTION OF THE DL11-W ARE DISABLED AND WILL NOT TEST THESE FUNCTIONS UNLESS ENABLED BY BIT10 (FOR ERROR FLAGS) AND BIT8 (FOR BREAK) OF THE SWR. THE DEFAULT CHARACTER SIZE IS 8 BITS (SEE PARA 2.3.2).
- FOR THE 11/44:
- THE PROGRAM ASSUMES THAT THE ERROR FLAG BITS AND THE BREAK FUNCTION ARE DISABLED, AND WILL NOT TEST THESE FUNCTIONS HOWEVER, IF BIT 10 (FOR ERROR FLAGS) AND BIT 08 (FOR BREAK) OF SWR ARE SET, THEN ERROR FLAGS AND BREAK TESTS ARE PERFORMED FOR THE TU58 SLU ONLY. IF BIT03 OF SWR IS ALSO SET THEN THESE TESTS WILL BE ENABLED FOR THE CONSOLE.
- READER ENABLE AND RECEIVER ACTIVE TESTS ARE NOT PERFORMED SINCE THE 11/44 MFM DOES NOT IMPLEMENT THESE FUNCTIONS.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES.

2.2 STARTING PROCEDURE

LOAD THE SWITCH REGISTER WITH SETTING

NOTE: IF USING A CPU WITHOUT HARDWARE SWITCH REGISTER
SOFTWARE SWITCH REGISTER LOCATION = 176.
(FOR A 11/44 CPU USE MFM CONSOLE FOR DEPOSITING
SWITCH REGISTER. TYPE ^P TO ENTER CONSOLE)

- A. START AT 200.
AFTER CHECKING THE TRANSMITTER, THE PROGRAM WILL PRINT ITS IDENTIFICATION AND REPORT THE NUMBER OF DEVICES UNDER TEST (NUMBER IS OCTAL).
'END PASS' IS PRINTED AFTER A FULL PASS HAS BEEN MADE ON ALL DEVICES UNDER TEST.
- B. START AT 204. *****NOTE*****
THE 'ECHO' TEST WILL BE EXECUTED. AN '*' IS PRINTED AT THE BEGINNING OF THE TEST. THE ECHO TEST READS A CHARACTER FROM

THE TERMINAL WRITES THAT CHARACTER TO THE TERMINAL AND
REPORTS ANY ERROR FLAGS SET IN THE RECEIVER BUFFER.
A CONTROL-C HALTS THE TEST AND PRINTS "STOP" AT THE TERMINAL
CONTINUING RESTARTS THE ECHO TEST.

C. START AT 210. *****NOTE*****
THE TERMINAL OUTPUT TEST WILL BE EXECUTED. DEPRESSING ANY CHARACTER
AT THE TERMINAL, HALTS THE TEST. CONTINUING RESTARTS THE TEST.
THE TEST OUTPUTS 32 CHARACTERS ON A LINE AND REPEATS THE
PATTERN EVERY THREE LINES. THE PATTERN IS AS FOLLOWS
(OCTAL CODE 040 --> 377):

.'#\$%&'()*+,-./0123456789:;< >? (OCTAL CODE)
(040 --> 077)

@ABCDEFGHIJKLMN O PQRSTU VWXYZ[\]^_ (100 --> 137)
'ABCDEFGHIJKLMN O PQRSTU VWXYZ' (140 --> 177) [LOWER CASE ALPHA]

THIS BOTTOM LINE COULD BE THE FOLLOWING IF THE TERMINAL
DOES NOT HAVE LOWER CASE:

@ABCDEFGHIJKLMN O PQRSTU VWXYZ[\] [UPPER CASE ALPHA]

*****NOTE*****

IF THE TESTING ON TERMINALS OTHER THAN THE CONSOLE IS DESIRED
FOR TESTS B OR C, SEE SECTION 2.3.4. AND 2.3.5. OF THIS DOCUMENT. ::++C
::++C

@
.REM

2.3 OPERATING PROCEDURE

2.3.1 OPERATIONAL SWITCH SETTINGS

THE DIAGNOSTIC WILL CHECK FOR EXISTENCE OF SWITCH REGISTER AT 177570.

IF NO SWITCH REGISTER IS AVAILABLE THE PROGRAM WILL AUTOMATICALLY USE THE CONTENTS OF LOCATION 176 AS THE SOFTWARE SWITCH REGISTER. THE USER SHOULD SET THIS LOCATION BEFORE STARTING THE PROGRAM. IF A HARDWARE SWITCH IS AVAILABLE AND A SOFTWARE SWR(LOC. 176) IS DESIRED, LOAD ALL 1'S INTO LOCATION 177570. (ALL SWITCHES UP IF PHYSICAL SWITCHES ARE AVAILABLE)

BIT15	- HALT ON ERROR
BIT14	- SCOPE LOOP
BIT13	- INHIBIT ERROR TIMEOUT
BIT12	- UNUSED
BIT11	- UNUSED
BIT10	- ENABLE ERROR FLAGS TESTS
BIT09	- LOOP ON ERROR
BIT08	- ENABLE BREAK FUNCTION TESTS
BIT07	- ENABLE DATA TEST WITH WRAP CABLE
BIT06	- INHIBIT RTC TESTS (ALLOW ONLY SLU TESTS)
BIT05	- ALLOW MANUAL SETTING OF 'SDEV' (DEVICE MAP)
BIT04	- INHIBIT SLU TESTS (ALLOW ONLY LINE CLOCK TESTS)
BIT03	- FOR 11/44 MFM: ENABLE BOTH 'BREAK TESTS' AND 'ERROR FLAG TESTS' FOR THE CONSOLE SLU. (THIS BIT IS VALID ONLY IF BIT10 OR BIT08 IS SET.)
BIT02	- 11/44 MFM OPTION: SELECT AUTO INITIATION OF T/A CONSOLE TEST VIA WRAP CABLE

FOR DL11-W:

IF THE SOFTWARE SWITCH REGISTER IS USED(LOC. 176) THEN BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS. BECAUSE THIS DIAGNOSTIC USES THE MAINTENANCE BIT OF THE SERIAL LINE, THE CONTROL-G SHOULD BE ISSUED DURING PROGRAM TYPEOUTS AT THAT TIME THE MAINTENANCE BIT IS SURE TO BE CLEAR.

IF A CONTROL-G IS DETECTED, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3. ^U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

FOR 11/44 CPU:

SINCE THE 11/44 HAS A HARDWARE SWITCH REGISTER LOADED BY THE MFM CONSOLE THEN THE DIAGNOSTIC SHOULD ALWAYS FIND EXISTENCE OF 177570. WHEN OPERATING ON THE 11/44 CPU, DYNAMIC CHANGING OF SWREG(177570) DURING PROGRAM EXECUTION CAN BE ACCOMPLISHED BY USING THE MFM CONSOLE. TYPING ^P<CR> ON THE CONSOLE TTY WILL ENTER THE CONSOLE. THIS IS CONSIDERED "CONSOLE MODE". TO EXAMINE SWREG TYPE ' E SW<CR> ' TO THE CONSOLE PROMPT. TO LOAD THE SWREG TYPE ' D SW DATA<CR> ' WHERE 'DATA' IS AN OCTAL NUMBER. IN ORDER FOR THE DIAGNOSTIC TO TYPE TO THE TTY IT IS NECESSARY TO HAVE THE 11/44 MFM IN 'PROGRAM I/O MODE'. THIS CAN BE ACCOMPLISHED BY TYPING ' C<CR> ' TO THE CONSOLE PROMPT. THEREFORE, WHEN CONSOLE USE IS COMPLETED, PLACE THE MFM IN 'PROGRAM I/O MODE'. BECAUSE THIS DIAGNOSTIC USES THE MAINTENANCE BIT, ^P WILL NOT BE ACKNOWLEDGED DURING THESE TESTS. THEREFORE, ISSUE ^P DURING PROGRAM TYPEOUTS. AT THIS TIME THE MAINTENANCE BIT WILL BE CLEARED.

2.3.2 SETTING BITS PER CHARACTER

THIS PROGRAM DEFAULTS TO TESTING 8 BITS PER CHARACTER. IF THE SERIAL LINE IS SET FOR 5-->7 BITS PER CHARACTER, SET THE MEMORY LOCATION '\$USWR' AS FOLLOWS:

CHAR. SIZE (# OF BITS)	'\$USWR' CONTENTS	
	(BINARY)	(OCTAL)
8	100000000	400
7	010000000	200
6	001000000	100
5	000100000	40

'\$USWR' IS USED IN THE DATA PATH TESTS TO LIMIT THE BINARY COUNT TEST PATTERN TO THE NUMBER OF BITS SELECTED ON THE SERIAL LINE.

2.3.3 RUNNING UNDER APT

THE APT MAILBOX IS LOCATED AT LOCATION 500, TO ALLOW ADDITIONAL SERIAL LINE VECTOR ASSIGNMENTS TO THE 400 AREA OF MEMORY.

FOR DL11W:

THE DEFAULT EXECUTION TIMES PROVIDED(\$STM,\$PSTM) ARE FOR EXECUTION WITH AN 11/34 PROCESSOR, CORE MEMORY, AND 110 BAUD.

FOR 11/44:

THE EXECUTION TIMES PROVIDED IN THE APT SCRIPT THAT FOLLOWS AND IN SECT. 2.4 ARE FOR EXECUTION WITH A 11/44 PROCESSOR, CACHE, 16K CORE MEMORY, AND 300 BAUD.

THE FOLLOWING IS A PROGRAM LOAD FILE USED BY APT:

1. E TABLE 'A' IS USED FOR APT DUMP MODE.

A. TWO SLU'S ARE TESTED.(\$SWREG BIT05=1 AND \$DEVM=3)
A SLU AT 177560(DEFAULT CONSOLE SLU) AND AT 176500
(BASE ADDRESS CODE).

B. THE ERROR FLAG AND BREAK TESTS ARE SELECTED FOR THE SLU
AT 176500(TU58 SLU IN MFG.)
(\$SWREG BIT 10 AND 8 =1)

2. E TABLE 'B' IS USED FOR APT QV AND RUN TIME MODES. IT ACCOMPLISHES WHAT E TABLE 'A' DOES, BUT ADDITIONALLY IT SUPPRESSES ALL TYPEOUTS TO THE TERMINAL(\$ENVM=240) AND SELECTS AUTO TESTING OF T/A CONSOLE TEST VIA WRAP CABLE(\$SWREG BIT02=1).

1ST PASS RUN TIME	LONGEST TEST TIME	ADDITIONAL RUN TIME
60	50	45

.....	E TABLES	
E-MODE/S-MODE (\$ENVM/\$ENV)	A 200/000	B 240/001
SWITCH REGISTER 1 (\$SWREG)	002440	002404
SWITCH REGISTER 2	000400	000400
CPU TYPE/OPTIONS	00/0000	00/0000
MEMORY MAP CODE 1	000/00000000	000/00000000
MEMORY MAP CODE 2	000/00000000	000/00000000
MEMORY MAP CODE 3	000/00000000	000/00000000
MEMORY MAP CODE 4	000/00000000	000/00000000
BUS PRIORITY/INTERRUPT 1	0000	0000
BUS PRIORITY/INTERRUPT 2	0000	0000
BUS ADDRESS CODE	176500	176500
DEVICE MAP CODE (\$DEVN)	000003	000003
CTLR. SPECIFIC WORD 1	000000	000000
CTLR. SPECIFIC WORD 2	000000	000000

2.3.4 RUN WITH ALTERNATE CONSOLE ADDRESS

TO USE A CONSOLE ADDRESS OTHER THAN 177560, OR VECTOR OTHER THAN 60, THE OPERATOR MUST SUPPLY THE PROGRAM WITH THE CORRECT ADDRESSES BY INSERTING THEM AT THE TAG LABELED 'CRCSR':

CRCSR: ADDRESS OF RECEIVER RCSR
CRBUF: ADDRESS OF RECEIVER BUFFER
CTCSR: ADDRESS OF TRANSMITTER CSR
CTBUF: ADDRESS OF TRANSMITTER BUFFER
CRVECT: ADDRESS OF RECEIVER VECTOR
CRPSW: ADDRESS OF ASSOCIATED PSW
CTVECT: ADDRESS OF TRANSMITTER VECTOR
CTPSW: ADDRESS OF ASSOCIATED PSW

2.3.5 TESTING ADDITIONAL SERIAL LINES

THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE SLU'S. IT REQUIRES THE ADDRESS OF THE FIRST ADDITIONAL RCSR (STORED AT '\$BASE') AND ITS INTERRUPT VECTOR (STORED AT '\$VECT1'); AND WILL BE ABLE TO ADDRESS ANY SLU STARTING AT THE SPECIFIED BASE ADDRESS UP TO 15 CONSECUTIVE DEVICES.

EXAMPLE: \$BASE: 776500
 \$VECT1: 300

THE PROGRAM WILL BE ABLE TO TEST THE CONSOLE PLUS ANY ADDITIONAL DL11-W SLU'S WITHIN THE RANGE 776500 --> 776660

\$BASE AND \$VECT1 DEFAULT TO 776500 AND 300 RESPECTIVELY.

THE PROGRAM ASSOCIATES UNIT NUMBERS TO DEVICES AS FOLLOWS:
(NUMBERS IN PARENTHESIS ARE OCTAL)

UNIT# 0 -->	CONSOLE [ADDRESS STORED AT 'CRCSR']
UNIT# 1 -->	BASE ADDRESS STORED AT '\$BASE'
	ASSOCIATED BASE VECTOR STORED AT '\$VECT1'
UNIT# 2 -->	BASE ADDRESS + (10)
	BASE VECTOR + (10)
UNIT# 3 -->	BASE ADDRESS + (20)
	BASE VECTOR + (20)
UNIT# 4 -->	BASE ADDRESS + (30)
	BASE VECTOR + (30)
	...
	...
	...
	V
UNIT#15 -->	BASE ADDRESS + (160)
	BASE VECTOR + (160)

STARTING AT LOCATION 200 AND HAVING BITS OF SWR CLEAR, THE PROGRAM WILL SELF
 SIZE THE NUMBER OF DEVICES (STARTING AT THE BAS ADDRESS) AND
 STORE A BIT MAP AT '\$DEVN' (DEVICE MAP) TO INDICATE WHICH UNIT NUMBERS
 ARE PRESENT AND WILL BE TESTED:

```

-----
: UNIT : UNIT : ..... : UNIT : UNIT : CONSOLE :
: 15  : 14  : ..... : 2   : 1   :         :
-----
    
```

A BIT MAP CAN BE ENTERED AT '\$DEVN' PRIOR TO STARTING THE PROGRAM
 SETTING BITS OF THE SWR INHIBITS THE SELF-SIZING AND DEVICE MAP
 GENERATION, AND USES THE VALUE STORED BY THE OPERATOR.

EXAMPLE:

SWR = 000040 [BINARY 0 000 000 000 100 000]
 \$BASE: 776500
 \$VECT1: 300

\$DEVN: 13 [BINARY - 0 000 000 000 001 011]

THE PROGRAM WILL TEST -
 UNIT# 0 - CONSOLE
 UNIT# 1 = 776500 ; 300
 UNIT# 3 = 776520 ; 320

2.4 EXECUTION TIMES -

FOR DL11-W: (110 BAUD)
LONGEST SUBTEST TIME - 50 SECONDS
PASS TIME = 60 SECONDS
ADDITIONAL DEVICES = 55 SECONDS/DEVICE

FOR 11/44: (300 BAUD)
LONGEST SUBTEST TIME - 50 SECONDS
PASS TIME = 60 SECONDS
ADDITIONAL DEVICES 55 SECONDS/DEVICE

3.0 ERROR REPORTING

IF A ROUTINE FAILS AND THE INHIBIT ERROR TIMEOUT (BIT13) OF THE SWR IS NOT SET, A PRINTOUT RESULTS IN THE FORM:

```
''(SOME ASCII MESSAGE)''  
TEST#  ERR PC  RCSR  [ANY APPLICABLE DAT HEADINGS]  
XXXXXX XXXXXX XXXXXX [ANY APPLICABLE DATA]
```

NOTE: 'RCSR' IS DEPENDENT ON THE FAILURE
& THEREFORE COULD BE TCSR, RBUF, TBUF, OR LKS

WHERE 'XXXXXX' IS AN OCTAL NUMBER.
THIS ERROR PRINTOUT OCCURS PROVIDED THE ERROR THAT EXISTS
WOULD NOT HINDER THE TIMEOUT. IN CASES WHERE IT IS NOT POSSIBLE
TO PRINT AN ERROR MESSAGE (I.E. FATAL CONSOLE TRANSMITTER
FAILURES), A HALT OCCURS AND THE PC CAN BE EXAMINED BY THE OPERATOR
TO FIND THE ERROR INFORMATION IN THE PROGRAM LISTING.

NOTE: FOR SOFTWARE SWITCH OPERATION, THE SWITCH REGISTER CAN
BE CHANGED BY TYPING A CONTROL-G AT THE CONSOLE DURING ERROR PRINTOUTS.
AFTER CONTINUING FROM THE ERROR HALT THE OLD SWR CONTENTS
IS DISPLAYED AND THE NEW CONTENTS CAN BE ENTERED.
IF ERROR HALTS ARE DISABLED, THE CONTROL-G RESPONSE OCCURS
IMMEDIATELY FOLLOWING THE TIMEOUT.

4.0 SUBROUTINE ABSTRACTS

4.1 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH A BREAK POINT TRAP (000003). THUS AN ILLEGAL TRAP OR INTERRUPT CAUSES A TRAP THROUGH THE BPT VECTOR(14) WHICH POINTS TO THE 'CATCH' ROUTINE.

THE 'CATCH' ROUTINE REPORTS THE PC THAT CAUSED THE ORIGINAL TRAP AND THE LOCATION OF THE TRAP VECTOR (IF UNDER APT, AN ERROR IS INDICATED TO APT). AFTER REPORTING THE ERROR THE PROGRAM HALTS. THE PROGRAM MUST BE RESTARTED AT THIS POINT.

4.2 WRPSW

THIS ROUTINE IS USED TO WRITE THE PSW BY POPPING VALUES FROM THE STACK. THIS METHOD IS USED TO BE COMPATIBLE WITH ALL 11 FAMILY PROCESSORS.

4.3 SCOPE

THIS ROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED AND UPDATES THE TEST NUMBER. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST AT WHICH THE SCOPE LOOP IS REQUESTED.

4.4 ERROR

THIS ROUTINE CALL IS PLACED WHEREEVER AN ERROR REPORT IS DESIRED. THE LOWER BYTE OF THIS CALL IS USED AS THE ERROR NUMBER AND AS A POINTER INTO THE ERROR TABLE. THIS ROUTINE REPORTS ERRORS TO APT USING '\$APTYPE' AND TYPES ERROR REPORTS TO THE CONSOLE USING '\$ERRTYPE'.

4.5 \$POWER

THIS ROUTINE SAVES ALL GENERAL REGISTERS DURING POWER-DOWN AND RESTORES THEM AT POWER-UP. IF A POWER FAILURE OCCURS 'POWER' IS PRINTED AT THE CONSOLE AFTER POWER IS RESTORED.

4.6 CKSWR

THIS ROUTINE CALL IS USED TO DETECT THE RECEPTION OF A CONTROL-G FROM THE CONSOLE. THE CALL USES '\$READ' TO PERFORM THE CONTROL-G SEQUENCE OF DISPLAYING THE CONTENTS OF THE SOFTWARE SWITCH REGISTER AND THE ENTERING THE NEW CONTENTS FROM THE TERMINAL.

5.0 AUTOMATIC INITIATION OF 11/44 T/A CONSOLE TEST VIA WRAP CABLE

PURPOSE: THE T/E(OR T/A) CONSOLE TEST CAN BE IMPLEMENTED BY MANUALLY TYPING T/E(OR T/A) ON THE KEYBOARD OF THE TERMINAL WHEN IN CONSOLE MODE. IN ORDER TO IMPLEMENT THIS TEST IN AN AUTOMATIC WAY IN MANUFACTURING WHILE UNDER APT, THE USE OF A WRAP CABLE FROM THE TU58 TO THE CONSOLE CAN BE USED ALLOWING THIS DIAGNOSTIC TO ISSUE THE 'T/A' COMMAND TO THE CONSOLE. THE DIAGNOSTIC WILL ISSUE THE APPROPRIATE SEQUENCE OF COMMANDS TO THE CONSOLE VIA THE WRAP CABLE WHEN BIT02 OF SWR =1. THE DIAGNOSTIC WILL THEN MONITOR THE EXPECTED RESPONSE FROM THE CONSOLE VIA THE WRAP CABLE AND HALT IF THERE IS AN ERROR. THE T/A TEST IS DONE ONLY AFTER ALL SLUS ARE TESTED. IF T/A IS SUCCESSFUL 'END PASS' IS PRINTED AND THE DIAGNOSTIC STARTS AGAIN.

FIGURE 1 SHOWS THE PROPER WRAP CABLE SETUP AND REQUIREMENTS FOR AUTOMATICALLY INITIATING T/A FROM THE THE DIAGNOSTIC. NOTICE THAT THIS ARRANGEMENT ALLOWS FOR THE TERMINAL TO MONITOR ALL COMMUNICATION FROM THE CONSOLE OUTPUT DURING EXECUTION OF THE DIAGNOSTIC IN 'WRAP MODE'.

TYPEOUT EXAMPLES

1. WITH THE CONFIGURATION OF FIGURE 1 AND 'WRAP MODE' SELECTED

THE PROGRAM MAY BE LOADED BY APT. IF 'E TABLE B' OF SECTION 2.3.3 WERE USED WITH THE EXCEPTION OF ALLOWING TYPEOUTS(\$ENV=200) THE FOLLOWING WOULD BE SEEN ON THE LOCAL TERMINAL:

```
CZDLDDO DL11-W/1144 MFM SLU
02 DEVICES UNDER TEST ^P
CONSOLE
>>>T/A
```

```
CONSOLE-TESTB
END PASS ^P
CONSOLE
>>>T/A
```

```
CONSOLE-TESTB
END PASS ^P
CONSOLE
>>>T/A
```

```
CONSOLE-TESTB
END PASS ^P .....ETC.
```

DESCRIPTION: REFERRING TO THE ABOVE PRINTOUTS:

- A. THE DIAGNOSTIC IS LOADED WITH THE TITLE BEING PRINTED.
- B. TWO SLU DEVICES(CONSOLE,TU58) ARE SPECIFIED
- C. BOTH SLUS ARE TESTED COMPLETELY. AT THIS POINT THE THE DIAGNOSTIC ISSUES ^P TO THE WRAP CABLE. THE CONSOLE ENTERS CONSOLE MODE AND THE ^P TYPED ON THE TERMINAL IS THE MFM CONSOLE ECHO.
- D. THE CONSOLE PERFORMS ITS OWN SELF TEST BY TYPING 'CONSOLE' FOLLOWED BY >>>, THE CONSOLE PROMPT.
- E. THE DIAGNOSTIC THEN ISSUES T/A AND THE TERMINAL SHOWS THE CONSOLE ECHO OF THIS.
- F. THE MFM THEN PERFORMS THE T/A TEST SHOWN BY THE TERMINAL TYPING 'CONSOLE-TESTB'.THE DIAGNOSTIC LOOKS FOR THE 'B' IN THIS TYPEOUT,AND WHEN FOUND , CONSIDERS THE TEST A SUCCESS. THE DIAGNOSTIC WILL TYPE END PASS INDICATING A SUCCESSFUL PASS OF THE DIAGNOSTIC.
- G. THE DIAGNOSTIC CONTINUES WITH FURTHER PASSES OF THE DIAGNOSTIC.

2. IF 'E TABLE B' OF 2.3.3 WERE USED WITH TYPEOUTS SUPPRESSED
(SENV=240) AS STATED, THEN THE FOLLOWING TYPEOUTS WOULD BE
NOTICED:

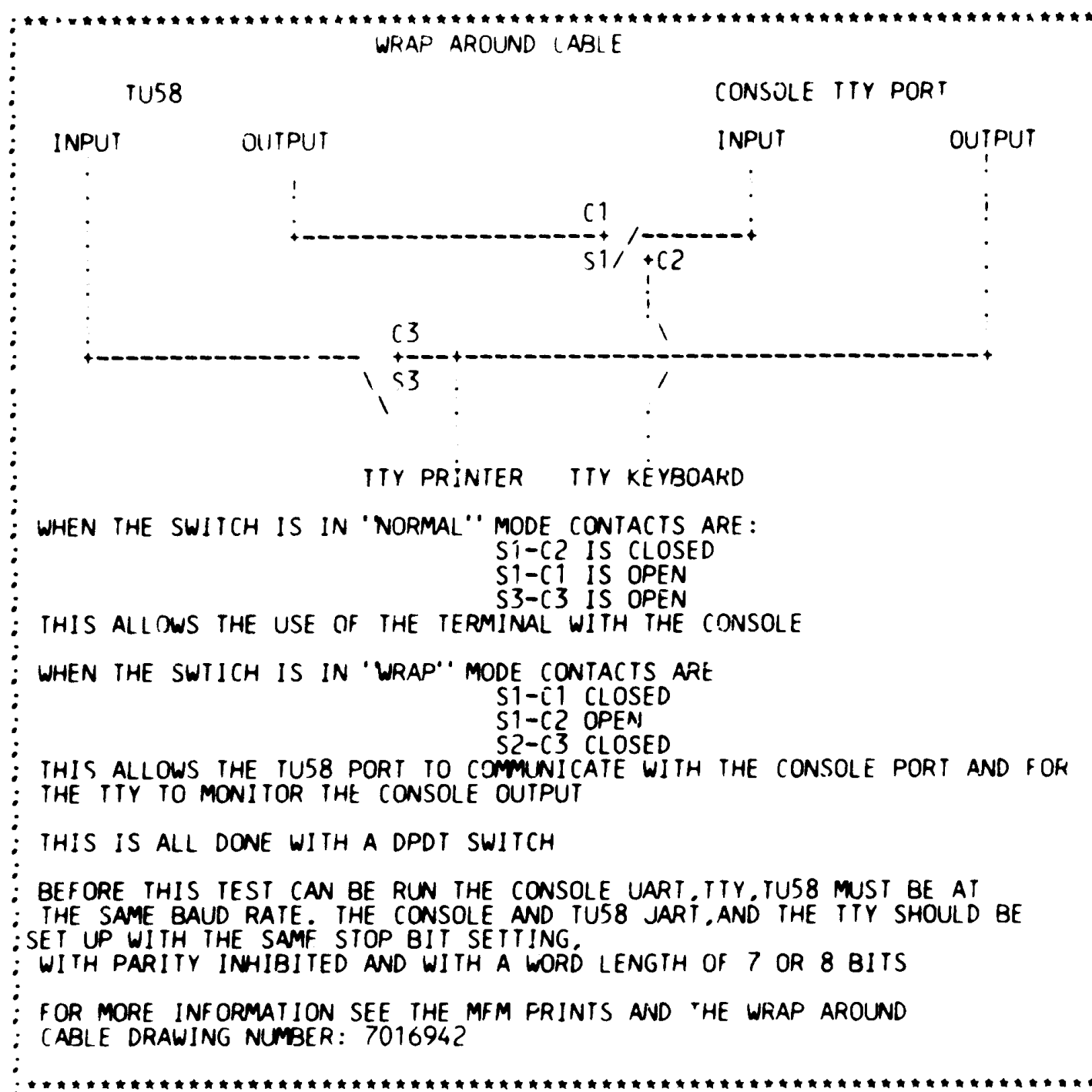
^P
CONSOLE
>>>T/A

CONSOLE-TESTB^P
CONSOLE
>>>T/A

CONSOLE-TESTB^P
CONSOLE
>>>T/A

CONSOLE-TESTB^PETC.

FIG. 1- WRAPAROUND CONFIGURATION - AUTO INITIATION OF
11/44 T/A CONSOLE TEST



784
791
795
799
803
804
811
812
818
828
843
844
845
846
847
848
849
850

```
.MCALL .STYPE,.$STYPOC,.$STRAP
.MCALL .SETUP,STARS,PUSH,POP,SETUP,.$EQUIV,.$ERRTYP
.MCALL .$APTHDR,.$APTBL,.$ACT11,.$SCOPE
.MCALL .SCMTAG,.$EOP,.$READ
.MCALL .EQUAT
.SBTTL BASIC DEFINITIONS
```

```
;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
```

001100
104000
000004

```
STACK= 1100
      ERROR=EMT
      SCOPE=JOT
```

```
;*MISCELLANEOUS DEFINITIONS
```

000011
000012
000015
000200
177776
177776
177774
177772
177570
177570

```
HT= 11          ;;CODE FOR HORIZONTAL TAB
LF= 12          ;;CODE FOR LINE FEED
CR= 15          ;;CODE FOR CARRIAGE RETURN
CRLF= 200       ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776      ;;PROCESSOR STATUS WORD
PSW=PS
STKLM= 177774   ;;STACK LIMIT REGISTER
PIRQ= 177772    ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570    ;;HARDWARE SWITCH REGISTER
DDISP= 177570   ;;HARDWARE DISPLAY REGISTER
```

```
;*GENERAL PURPOSE REGISTER DEFINITIONS
```

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

```
R0= %0          ;;GENERAL REGISTER
R1= %1          ;;GENERAL REGISTER
R2= %2          ;;GENERAL REGISTER
R3= %3          ;;GENERAL REGISTER
R4= %4          ;;GENERAL REGISTER
R5= %5          ;;GENERAL REGISTER
R6= %6          ;;GENERAL REGISTER
R7= %7          ;;GENERAL REGISTER
SP= %6          ;;STACK POINTER
PC= %7          ;;PROGRAM COUNTER
```

```
;*PRIORITY LEVEL DEFINITIONS
```

000000
000040
000100
000140
000200
000240
000300
000340

```
PR0= 0          ;;PRIORITY LEVEL 0
PR1= 40         ;;PRIORITY LEVEL 1
PR2= 100        ;;PRIORITY LEVEL 2
PR3= 140        ;;PRIORITY LEVEL 3
PR4= 200        ;;PRIORITY LEVEL 4
PR5= 240        ;;PRIORITY LEVEL 5
PR6= 300        ;;PRIORITY LEVEL 6
PR7= 340        ;;PRIORITY LEVEL 7
```

```

100000      ;*'SWITCH REGISTER' SWITCH DEFINITIONS
040000      SW15= 100000
020000      SW14= 40000
010000      SW13= 20000
004000      SW12= 10000
002000      SW11= 4000
001000      SW10= 2000
000400      SW09= 1000
000200      SW08= 400
000100      SW07= 200
000040      SW06= 100
000020      SW05= 40
000010      SW04= 20
000004      SW03= 10
000002      SW02= 4
000001      SW01= 2
001000      SW00= 1
000400      SW9=SW09
000200      SW8=SW08
000100      SW7=SW07
000040      SW6=SW06
000020      SW5=SW05
000010      SW4=SW04
000004      SW3=SW03
000002      SW2=SW02
000001      SW1=SW01
000000      SW0=SW00
```

```

100000      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
040000      BIT15= 100000
020000      BIT14= 40000
010000      BIT13= 20000
004000      BIT12= 10000
002000      BIT11= 4000
001000      BIT10= 2000
000400      BIT09= 1000
000200      BIT08= 400
000100      BIT07= 200
000040      BIT06= 100
000020      BIT05= 40
000010      BIT04= 20
000004      BIT03= 10
000002      BIT02= 4
000001      BIT01= 2
001000      BIT00= 1
000400      BIT9=BIT09
000200      BIT8=BIT08
000100      BIT7=BIT07
000040      BIT6=BIT06
000020      BIT5=BIT05
000010      BIT4=BIT04
000004      BIT3=BIT03
000002      BIT2=BIT02
000001      BIT1=BIT01
000000      BIT0=BIT00
```

```

000004      ;*BASIC 'CPU' TRAP VECTOR ADDRESSES
000010      ERRVEC= 4      ;;TIME OUT AND OTHER ERRORS
000014      RESVEC= 10     ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014      TBITVEC=14     ;;'T' BIT
000014      TRIVEC= 14     ;;TRACE TRAP
000014      BPTVEC= 14     ;;BREAKPOINT TRAP (BPT)
000020      IOTVEC= 20     ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024      PWRVEC= 24     ;;POWER FAIL
000030      EMTVEC= 30     ;;EMULATOR TRAP (EMT) **ERROR**
000034      TRAPVEC=34     ;;'TRAP' TRAP
000060      TKVEC= 60      ;;TTY KEYBOARD VECTOR
000064      TPVEC= 64      ;;TTY PRINTER VECTOR
000240      PIRQVEC=240    ;;PROGRAM INTERRUPT REQUEST VECTOR
851          000007      MFPT=7
852          176500      ABASE= 176500
853          000300      AVECT1= 300
854          000400      AUSWR= 400
855          000001      $TN= 1
856          161000      $SWR= 161000
857          000003      BPT= 000003      ;THIS IS THE COMMAND FOR A TRAP
858                                     ; THROUGH 14 (BPT TRAP)
859
860          000000      .-0
861      ;*****
862      ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A '.+2,BPT'
863      ;*SEQUENCE TO CATCH ILLEGAL TRAPS & INTERRUPTS
864      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
865
873
874          000014      . 14      ;THE BPT TRAP VECTOR POINTS TO THE
875 000014 014552      .WORD CATCH ; ILLEGAL TRAP HANDLER 'CATCH'
876 000016 000340      .WORD 340
877
878          000042      . 42
879 000042 000000      .WORD 0
880
881
883
884
885          000174      . = 174
886 000174 000000      DISPREG: .WORD 0
887 000176 000000      SWREG: .WORD 0
888
889          000200      . =200
890 000200 000167 002642      JMP START ;DO INTERFACE TEST
891 000204 000167 017114      JMP ECHO ;DO ECHO TEST
892 000210 000167 017330      JMP OUTTST ;DO OUTPUT TEST TO TERMINAL

```

```
894
895      000500
896      .SBTTL  . = 500
          ACT11 HOOKS
          ;*****
          ;HOOKS REQUIRED BY ACT11
          000500      $SVPC=.          ;SAVE PC
          000046      . = 46
          000046      $ENDAD          ;:1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
          000052      . = 52
          000052      .WORD 0          ;:2)SET LOC.52 TO ZERO
          000500      . = $SVPC      ;: RESTORE PC
897      .SBTTL  APT PARAMETER BLOCK
          ;*****
          ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
          ;*****
          000500      . $X=.          ;:SAVE CURRENT LOCATION
          000024      . = 24          ;:SET POWER FAIL TO POINT TO START OF PROGRAM
          000024      200          ;:FOR APT START UP
          000044      . = 44          ;:POINT TO APT INDIRECT ADDRESS PNTR.
          000044      $APTHDR          ;:POINT TO APT HEADER BLOCK
          000500      . = . $X      ;:RESET LOCATION COUNTER
          ;*****
          ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
          ;INTERFACE SPEC.
          000500      $APTHD:
          000500      $HIBTS: .WORD 0          ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
          000502      $MBADR: .WORD $MAIL      ;:ADDRESS OF APT MAILBOX (BITS 0-15)
          000504      $TSTM: .WORD 50          ;:RUN TIM OF LONGEST TEST
          000506      $PASTM: .WORD 60          ;:RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
          000510      $UNITM: .WORD 55          ;:ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
          000512      .WORD $ETEND-$MAIL/2 ;:LENGTH MAILBOX-ETABLE(WORDS)
898
```

900

.SBTTL COMMON TAGS

*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
*USED IN THE PROGRAM.

001000	001000	\$CMTAG: . =1000	:: START OF COMMON TAGS
001000	000000	\$STNM: .WORD 0	:: CONTAINS THE TEST NUMBER
001002	000	\$ERFLG: .BYTE 0	:: CONTAINS ERROR FLAG
001003	000	\$ICNT: .WORD 0	:: CONTAINS SUBTEST ITERATION COUNT
001004	000000	\$LPADR: .WORD 0	:: CONTAINS SCOPE LOOP ADDRESS
001006	000000	\$LPERR: .WORD 0	:: CONTAINS SCOPE RETURN FOR ERRORS
001010	000000	\$ERTTL: .WORD 0	:: CONTAINS TOTAL ERRORS DETECTED
001012	000000	\$ITEMB: .BYTE 0	:: CONTAINS ITEM CONTROL BYTE
001014	000	\$ERMAX: .BYTE 1	:: CONTAINS MAX. ERRORS PER TEST
001015	001	\$ERRPC: .WORD 0	:: CONTAINS PC OF LAST ERROR INSTRUCTION
001016	000000	\$GDADR: .WORD 0	:: CONTAINS ADDRESS OF 'GOOD' DATA
001020	000000	\$BDADR: .WORD 0	:: CONTAINS ADDRESS OF 'BAD' DATA
001022	000000	\$GDDAT: .WORD 0	:: CONTAINS 'GOOD' DATA
001024	000000	\$BDDAT: .WORD 0	:: CONTAINS 'BAD' DATA
001026	000000		:: RESERVED--NOT TO BE USED
001030	000000		
001032	000000		
001034	000	\$AUTOB: .BYTE 0	:: AUTOMATIC MODE INDICATOR
001035	000	\$INTAG: .BYTE 0	:: INTERRUPT MODE INDICATOR
001036	000000		
001040	177570	\$SWR: .WORD DSWR	:: ADDRESS OF SWITCH REGISTER
001042	177570	\$DISPLAY: .WORD DDISP	:: ADDRESS OF DISPLAY REGISTER
001044	177560	\$TKS: 177560	:: TTY KBD STATUS
001046	177562	\$TKB: 177562	:: TTY KBD BUFFER
001050	177564	\$TPS: 177564	:: TTY PRINTER STATUS REG. ADDRESS
001052	177566	\$TPB: 177566	:: TTY PRINTER BUFFER REG. ADDRESS
001054	000	\$NULL: .BYTE 0	:: CONTAINS NULL CHARACTER FOR FILLS
001055	002	\$FILLS: .BYTE 2	:: CONTAINS # OF FILLER CHARACTERS REQUIRED
001056	012	\$FILLC: .BYTE 12	:: INSERT FILL CHARS. AFTER A 'LINE FEED'
001057	000	\$TPFLG: .BYTE 0	:: 'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
001060	000000	\$ESCAPE: 0	:: ESCAPE ON ERROR ADDRESS
001062	077	\$QUES: .ASCII /?/	:: QUESTION MARK
001063	015	\$CRLF: .ASCII <15>	:: CARRIAGE RETURN
001064	012	\$LF: .ASCII <12>	:: LINE FEED

.SBTTL APT MAILBOX-ETABLE

001066		..EVEN	
001066	000000	\$MAIL: .	:: APT MAILBOX
001070	000000	\$MSGTY: .WORD AMSGTY	:: MESSAGE TYPE CODE
001072	000000	\$FATAL: .WORD AFATAL	:: FATAL ERROR NUMBER
001074	000000	\$TESTN: .WORD ATESTN	:: TEST NUMBER
001076	000000	\$PASS: .WORD APASS	:: PASS COUNT
001100	000000	\$DEVCT: .WORD ADEVCT	:: DEVICE COUNT
001102	000000	\$UNIT: .WORD AUNIT	:: I/O UNIT NUMBER
001104	000000	\$MSGAD: .WORD AMSGAD	:: MESSAGE ADDRESS
001106		\$MSGLG: .WORD AMSLG	:: MESSAGE LENGTH
001106		\$ETABLE: .	:: APT ENVIRONMENT TABLE
001106	000	\$ENV: .BYTE AENV	:: ENVIRONMENT BYTE

001107	000	\$ENVN:	.BYTE	AENVN	::ENVIRONMENT MODE BITS
001110	000000	\$SWREG:	.WORD	ASWREG	::APT SWITCH REGISTER
001112	000400	\$USWR:	.WORD	AUSWR	::USER SWITCHES
001114	000000	\$CPUOP:	.WORD	ACPUOP	::CPU TYPE,OPTIONS
		.*			BITS 15-11=CPU TYPE
		.*			11/04=01,11/05=02,11/20=03,11/40-04,11/45=05
		.*			11/70=06,PDQ=07,Q=10
		.*			BIT 10=REAL TIME CLOCK
		.*			BIT 9=FLOATING POINT PROCESSOR
		.*			BIT 8=MEMORY MANAGEMENT
001116	000	\$MAMS1:	.BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
001117	000	\$MTYP1:	.BYTE	AMTYP1	::MEM. TYPE,BLK#1
		.*			MEM.TYPE BYTE -- (HIGH BYTE)
		.*			900 NSEC CORE=001
		.*			300 NSEC BIPOLAR=002
		.*			500 NSEC MOS=003
001120	000000	\$MADR1:	.WORD	AMADR1	::HIGH ADDRESS,BLK#1
		.*			MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
001122	000	\$MAMS2:	.BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTE
001123	000	\$MTYP2:	.BYTE	AMTYP2	::MEM. TYPE,BLK#2
001124	000000	\$MADR2:	.WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
001126	000	\$MAMS3:	.BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
001127	000	\$MTYP3:	.BYTE	AMTYP3	::MEM. TYPE,BLK#3
001130	000000	\$MADR3:	.WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
001132	000	\$MAMS4:	.BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
001133	000	\$MTYP4:	.BYTE	AMTYP4	::MEM. TYPE,BLK#4
001134	000000	\$MADR4:	.WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
001136	000300	\$VECT1:	.WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
001140	000000	\$VECT2:	.WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
001142	176500	\$BASE:	.WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
001144	000000	\$DEVN:	.WORD	ADEVN	::DEVICE MAP
001146		\$ETEND:			
		.MEXIT			

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;* EM ::POINTS TO THE ERROR MESSAGE
;* DH ::POINTS TO THE DATA HEADER
;* DT ::POINTS TO THE DATA
;* DF ::POINTS TO THE DATA FORMAT

\$ERRTB:

.\$ERRTB:

901	001146			
902	001146		EM1	::'CAN NOT ACCESS TCSR'
903	001146	017630	DH1	::'TEST# ERR PC TCSR'
904	001150	024012	DT1	::\$TESTN,\$ERRPC,TCSR
905	001152	024572	0	
906	001154	000000		
907				
908	001156	017654	EM2	::'CAN NOT ACCESS TBUF'
909	001160	024037	DH2	::'TEST# ERR PC TBUF'
910	001162	024602	DT2	::\$TESTN,\$ERRPC,TBUF
911	001164	000000	0	
912				
913	001166	017700	EM3	::'TCSR DONE NOT CLEARED WITH TBUF FULL'
914	001170	024012	DH1	::'TEST# ERR PC TCSR'
915	001172	024572	DT1	::\$TESTN,\$ERRPC,TCSR
916	001174	000000	0	
917				
918	001176	017745	EM4	::'TCSR DONE NOT SET'
919	001200	024012	DH1	::'TEST# ERR PC TCSR'
920	001202	024572	DT1	::\$TESTN,\$ERRPC,TCSR
921	001204	000000	0	
922				
923	001206	017767	EM5	::TCSR DONE NOT SET WITH RESET
924	001210	024012	DH1	::'TEST# ERR PC TCSR'
925	001212	024572	DT1	::\$TESTN,\$ERRPC,TCSR
926	001214	000000	0	
927				
928	001216	020024	EM6	::'CAN NOT ACCESS RCSR'
929	001220	024064	DH6	::'TEST# ERR PC RCSR'
930	001222	024612	DT6	::\$TESTN,\$ERRPC,RCSR
931	001224	000000	0	
932				
933	001226	020050	EM7	::'CAN NOT ACCESS RBUF'
934	001230	024111	DH7	::'TEST# ERR PC RBUF'
935	001232	024622	DT7	::\$TESTN,\$ERRPC,RBUF
936	001234	000000	0	
937				
938	001236	020074	EM10	::'CAN NOT ACCESS LKS'
939	001240	024136	DH10	::'TEST# ERR PC LKS'
940	001242	024632	DT10	::\$TESTN,\$ERRPC,LKS
941	001244	000000	0	
942				

943	001246	020117	EM11	;'BIT0 OF TCSR NOT CLEAR AFTER RESET''
944	001250	024012	DH1	;'TEST# ERR PC TCSR''
945	001252	024572	DT1	;\$TESTN,\$ERRPC,TCSR
946	001254	000000	0	
947				
948	001256	020162	EM12	;'CAN NOT SET BIT0 OF TCSR''
949	001260	024012	DH1	;'TEST# ERR PC TCSR''
950	001262	024572	DT1	;\$TESTN,\$ERRPC,TCSR
951	001264	000000	0	
952				
953	001266	020213	EM13	;'CAN NOT CLEAR BIT0 OF TCSR''
954	001270	024012	DH1	;'TEST# ERR PC TCSR''
955	001272	024572	DT1	;\$TESTN,\$ERRPC,TCSR
956	001274	000000	0	
957				
958	001276	020246	EM14	;'RESET DID NOT CLEAR BIT0 OF TCSR''
959	001300	024012	DH1	;'TEST# ERR PC TCSR''
960	001302	024572	DT1	;\$TESTN,\$ERRPC,TCSR
961	001304	000000	0	
962				
963	001306	020307	EM15	;'BIT2 OF TCSR NOT CLEAR AFTER RESET''
964	001310	024012	DH1	;'TEST# ERR PC TCSR''
965	001312	024572	DT1	;\$TESTN,\$ERRPC,TCSR
966	001314	000000	0	
967				
968	001316	020352	EM16	;'CAN NOT SET BIT2 OF TCSR''
969	001320	024012	DH1	;'TEST# ERR PC TCSR''
970	001322	024572	DT1	;\$TESTN,\$ERRPC,TCSR
971	001324	000000	0	
972				
973	001326	020403	EM17	;'CAN NOT CLEAR BIT2 OF TCSR''
974	001330	024012	DH1	;'TEST# ERR PC TCSR''
975	001332	024572	DT1	;\$TESTN,\$ERRPC,TCSR
976	001334	000000	0	
977				
978	001336	020436	EM20	;'RESET DID NOT CLEAR BIT2 OF TCSR''
979	001340	024012	DH1	;'TEST# ERR PC TCSR''
980	001342	024572	DT1	;\$TESTN,\$ERRPC,TCSR
981	001344	000000	0	
982				
983	001346	020477	EM21	;'BIT6 OF TCSR NOT CLEAR AFTER RESET2
984	001350	024012	DH1	;'TEST# ERR PC TCSR''
985	001352	024572	DT1	;\$TESTN,\$ERRPC,TCSR
986	001354	000000	0	
987				
988	001356	020542	EM22	;'XMIT INTERRUPT WITH PRIORITY 7''
989	001360	024012	DH1	;'TEST# ERR PC TCSR''
990	001362	024572	DT1	;\$TESTN,\$ERRPC,TCSR
991	001364	000000	0	
992				
993	001366	020577	EM23	;'CAN NOT SET BIT6 OF TCSR''
994	001370	024012	DH1	;'TEST# ERR PC TCSR''
995	001372	024572	DT1	;\$TESTN,\$ERRPC,TCSR
996	001374	000000	0	
997				
998	001376	020630	EM24	;'CAN NOT CLEAR BIT6 OF TCSR''
999	001400	024012	DH1	;'TEST# ERR PC TCSR''

1000	001402	024572	DT1	;	\$TESTN,\$ERRPC,TCSR
1001	001404	000000	0		
1002					
1003	001406	020663	EM25	;	'RESET DID NOT CLEAR BIT6 OF TCSR'
1004	001410	024012	DH1	;	'TEST# ERR PC TCSR'
1005	001412	024572	DT1	;	\$TESTN,\$ERRPC,TCSR
1006	001414	000000	0		
1007					
1008	001416	020724	EM26	;	'BIT6 OF RCSR NOT CLEAR AFTER RESET'
1009	001420	024064	DH6	;	'TEST# ERR PC RCSR'
1010	001422	024612	DT6	;	\$TESTN,\$ERRPC,RCSR
1011	001424	000000	0		
1012					
1013	001426	020767	EM27	;	'RCVR INTERRUPT WITH PRIORITY 7'
1014	001430	024064	DH6	;	'TEST# ERR PC RCSR'
1015	001432	024612	DT6	;	\$TESTN,\$ERRPC,RCSR
1016	001434	000000	0		
1017					
1018	001436	021026	EM30	;	'CAN NOT SET BIT6 OF RCSR'
1019	001440	024064	DH6	;	'TEST# ERR PC RCSR'
1020	001442	024612	DT6	;	\$TESTN,\$ERRPC,RCSR
1021	001444	000000	0		
1022					
1023	001446	021057	EM31	;	'CAN NOT CLEAR BIT6 OF RCSR'
1024	001450	024064	DH6	;	'TEST# ERR PC RCSR'
1025	001452	024612	DT6	;	\$TESTN,\$ERRPC,RCSR
1026	001454	000000	0		
1027					
1028	001456	021112	EM32	;	'CAN NOT CLEAR BIT6 OF RCSR WITH RESET2
1029	001460	024064	DH6	;	'TEST# ERR PC RCSR'
1030	001462	024612	DT6	;	\$TESTN,\$ERRPC,RCSR
1031	001464	000000	0		
1032					
1033	001466	021160	EM33	;	'BIT6 OF LKS NOT CLEAR AFTER RESET'
1034	001470	024136	DH10	;	'TEST# ERR PC LKS'
1035	001472	024632	DT10	;	\$TESTN,\$ERRPC,LKS
1036	001474	000000	0		
1037					
1038	001476	021222	EM34	;	'LKS INTERRUPT WITH PRIORITY 7'
1039	001500	024136	DH10	;	'TEST# ERR PC LKS'
1040	001502	024632	DT10	;	\$TESTN,\$ERRPC,LKS
1041	001504	000000	0		
1042					
1043	001506	021260	EM35	;	'CAN NOT SET BIT6 OF LKS'
1044	001510	024136	DH10	;	'TEST# ERR PC LKS'
1045	001512	024632	DT10	;	\$TESTN,\$ERRPC,LKS
1046	001514	000000	0		
1047					
1048	001516	021310	EM36	;	'CAN NOT CLEAR BIT6 OF LKS'
1049	001520	024136	DH10	;	'TEST# ERR PC LKS'
1050	001522	024632	DT10	;	\$TESTN,\$ERRPC,LKS
1051	001524	000000	0		
1052					
1053	001526	021342	EM37	;	'RESET DID NOT CLEAR BIT6 OF LKS'
1054	001530	024136	DH10	;	'TEST# ERR PC LKS'
1055	001532	024632	DT10	;	\$TESTN,\$ERRPC,LKS
1056	001534	000000	0		

1057				
1058	001536	021402	EM40	;'DUAL ADDRESSING ERROR'
1059	001540	024162	DH40	;'TEST# ERR PC GOOD BAD'
1060	001542	024642	DT40	;\$TESTN,\$ERRPC,\$GDADR,\$BDCSR
1061	001544	000000	0	
1062				
1063	001546	021430	EM41	;'BIT7 OF LKS NOT SET AFTER RESET2
1064	001550	024136	DH10	;'TEST# ERR PC LKS'
1065	001552	024632	DT10	;\$TESTN,\$ERRPC,LKS
1066	001554	000000	0	
1067				
1068	001556	021470	EM42	;'CAN NOT CLEAR BIT7 OF LKS'
1069	001560	024136	DH10	;'TEST# ERR PC LKS'
1070	001562	024632	DT10	;\$TESTN,\$ERRPC,LKS
1071	001564	000000	0	
1072				
1073	001566	021522	EM43	;'BIT7 OF LKS DOES NOT SET'
1074	001570	024136	DH10	;'TEST# ERR PC LKS'
1075	001572	024632	DT10	;\$TESTN,\$ERRPC,LKS
1076	001574	000000	0	
1077				
1078	001576	021553	EM44	;'RTC INTERRUPT AT PRIORITY 7'
1079	001600	024136	DH10	;'TEST# ERR PC LKS'
1080	001602	024632	DT10	;\$TESTN,\$ERRPC,LKS
1081	001604	000000	0	
1082				
1083	001606	021607	EM45	;'RTC INTERRUPTS WHEN DISABLED'
1084	001610	024136	DH10	;'TEST# ERR PC LKS'
1085	001612	024632	DT10	;\$TESTN,\$ERRPC,LKS
1086	001614	000000	0	
1087				
1088	001616	021644	EM46	;'RTC INTERRUPT DID NOT OCCUR'
1089	001620	024136	DH10	;'TEST# ERR PC LKS'
1090	001622	024632	DT10	;\$TESTN,\$ERRPC,LKS
1091	001624	000000	0	
1092				
1093	001626	021644	EM47	;'RTC INTERRUPT DID NOT OCCUR'
1094	001630	024136	DH10	;'TEST# ERR PC LKS'
1095	001632	024632	DT10	;\$TESTN,\$ERRPC,LKS
1096	001634	000000	0	
1097				
1098	001636	021700	EM50	;'RTC DOUBLE INTERRUPT'
1099	001640	024136	DH10	;'TEST# ERR PC LKS'
1100	001642	024632	DT10	;\$TESTN,\$ERRPC,LKS
1101	001644	000000	0	
1102				
1103	001646	021725	EM51	;'RESET DID NOT CLEAR RTC INTERRUPT'
1104	001650	024136	DH10	;'TEST# ERR PC LKS'
1105	001652	024632	DT10	;\$TESTN,\$ERRPC,LKS
1106	001654	000000	0	
1107				
1108	001656	021755	EM52	;'RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS'
1109	001660	024136	DH10	;'TEST# ERR PC LKS'
1110	001662	024632	DT10	;\$TESTN,\$ERRPC,LKS
1111	001664	000000	0	
1112				
1113	001666	022032	EM53	

1114	001670	024216	DH53	;'TEST# ERR PC LKS CNT1 CNT2''
1115	001672	024654	DT53	;\$TESTN,\$ERRPC,LKS,FIRST,SECND
1116	001674	000000	0	
1117				
1118	001676	022056	EM54	;'XMIT INTERRUPTS WHEN DISABLED''
1119	001700	024012	DH1	;'TEST# ERR PC TCSR''
1120	001702	024572	DT1	;\$TESTN,\$ERRPC,TCSR
1121	001704	000000	0	
1122				
1123	001706	022214	EM55	;'XMIT DID NOT INTERRUPT''
1124	001710	024012	DH1	;'TEST# ERR PC TCSR''
1125	001712	024572	DT1	;\$TESTN,\$ERRPC,TCSR
1126	001714	000000	0	
1127				
1128	001716	022114	EM56	;'XMIT INTERRUPT AT PRIORITY 7''
1129	001720	024012	DH1	;'TEST# ERR PC TCSR''
1130	001722	024572	DT1	;\$TESTN,\$ERRPC,TCSR
1131	001724	000000	0	
1132				
1133	001726	022152	EM57	;'XMIT INTERRUPTS WITH ENABLE CLEAR''
1134	001730	024012	DH1	;'TEST# ERR PC TCSR''
1135	001732	024572	DT1	;\$TESTN,\$ERRPC,TCSR
1136	001734	000000	0	
1137				
1138	001736	022214	EM60	;'XMIT DID NOT INTERRUPT''
1139	001740	024012	DH1	;'TEST# ERR PC TCSR''
1140	001742	024572	DT1	;\$TESTN,\$ERRPC,TCSR
1141	001744	000000	0	
1142				
1143	001746	022243	EM61	;'XMIT RE-INTERRUPTED''
1144	001750	024012	DH1	;'TEST# ERR PC TCSR''
1145	001752	024572	DT1	;\$TESTN,\$ERRPC,TCSR
1146	001754	000000	0	
1147				
1148	001756	022267	EM62	;'LOADING TBUF DID NOT CLEAR INTERRUPT''
1149	001760	024012	DH1	;'TEST# ERR PC TCSR''
1150	001762	024572	DT1	;\$TESTN,\$ERRPC,TCSR
1151	001764	000000	0	
1152				
1153	001766	022334	EM63	;'RCVR ACTIVE NOT SET''
1154	001770	024064	DH6	;'TEST# ERR PC RCSR''
1155	001772	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1156	001774	000000	0	
1157				
1158	001776	022360	EM64	;'RECEIVER DONE NEVER SET''
1159	002000	024064	DH6	;'TEST# ERR PC RCSR''
1160	002002	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1161	002004	000000	0	
1162				
1163	002006	022404	EM65	;'RCVR ACTIVE NOT CLEARED WITH DONE SET''
1164	002010	024064	DH6	;'TEST# ERR PC RCSR''
1165	002012	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1166	002014	000000	0	
1167				
1168	002016	022452	EM66	;'RESET DID NOT CLEAR RCVR DONE''
1169	002020	024064	DH6	;'TEST# ERR PC RCSR''
1170	002022	024612	DT6	;\$TESTN,\$ERRPC,RCSR

1171	002024	000000	0	
1172				
1173	002026	022510	EM67	;'RDR ENABLE SET DID NOT CLEAR RCVR DONE''
1174	002030	024064	DH6	;'TEST# ERR PC RCSR''
1175	002032	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1176	002034	000000	0	
1177				
1178	002036	022553	EM70	;'READING RBUF DID NOT CLEAR RCVR DONE''
1179	002040	024064	DH6	;'TEST# ERR PC RCSR''
1180	002042	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1181	002044	000000	0	
1182				
1183	002046	022620	EM71	;'RCVR INTERRUPTS WITH ENABLE CLEAR''
1184	002050	024064	DH6	;'TEST# ERR PC RCSR''
1185	002052	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1186	002054	000000	0	
1187				
1188	002056	022767	EM72	;'RCVR DID NOT INTERRUPT''
1189	002060	024064	DH6	;'TEST# ERR PC RCSR''
1190	002062	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1191	002064	000000	0	
1192				
1193	002066	022662	EM73	;'RCVR INTERRUPTS AT PRIORITY 7''
1194	002070	024064	DH6	;'TEST# ERR PC RCSR''
1195	002072	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1196	002074	000000	0	
1197				
1198	002076	022720	EM74	;'RCVR INTERRUPT REQUEST PASSED WITH ENABLE CLEAR''
1199	002100	024064	DH6	;'TEST# ERR PC RCSR''
1200	002102	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1201	002104	000000	0	
1202				
1203	002106	022767	EM75	;'RCVR DID NOT INTERRUPT''
1204	002110	024064	DH6	;'TEST# ERR PC RCSR''
1205	002112	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1206	002114	000000	0	
1207	002116	023016	EM76	;'RECEIVER RE-INTERRUPTED''
1208	002120	024064	DH6	;'TEST# ERR PC RCSR''
1209	002122	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1210	002124	000000	0	
1211				
1212	002126	023042	EM77	;'READING RBUF DID NOT CLEAR INTERRUPT''
1213	002130	024064	DH6	;'TEST# ERR PC RCSR''
1214	002132	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1215	002134	000000	0	
1216				
1217	002136	023107	EM100	;'RESET DID NOT CLEAR RCVR INTERRUPT''
1218	002140	024064	DH6	;'TEST# ERR PC RCSR''
1219	002142	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1220	002144	000000	0	
1221				
1222				
1223	002146	023152	EM101	;'OR' FLAG DID NOT SET''
1224	002150	024064	DH6	
1225	002152	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1226	002154	000000	0	
1227				

1228	002156	023200	EM102	;'ERROR' NOT SET WITH 'OR' FLAG''
1229	002160	024064	DH6	;'TEST# ERR PC RCSR''
1230	002162	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1231	002164	000000	0	
1232	002166	023237	EM103	;'BREAK DID NOT TRANSMIT ALL ZEROES''
1233	002170	024263	DH103	;'TEST# ERR PC RCSR DATA''
1234	002172	024670	DT103	;\$TESTN,\$ERRPC,RCSR,\$BDDAT
1235	002174	000000	0	
1236				
1237	002176	023275	EM104	;'BREAK DID NOT SET FRAMING ERROR''
1238	002200	024064	DH6	;'TEST# ERR PC RCSR''
1239	002202	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1240	002204	000000	0	
1241				
1242	002206	023332	EM105	;'DATA COMPARE ERROR''
1243	002210	024320	DH105	;'TEST# ERR PC RCSR GOOD BAD''
1244	002212	024702	DT105	;\$TESTN,\$ERRPC,RCSR,GD,BD
1245	002214	000000	0	
1246				
1247	002216	023355	EM106	;'DATA COMPARE ERROR WITH CABLE''
1248	002220	024320	DH105	;'TEST# ERR PC RCSR GOOD BAD''
1249	002222	024702	DT105	;\$TESTN,\$ERRPC,RCSR,GD,BD
1250	002224	000000	0	
1251				
1252	002226	023413	EM107	;'TIMEOUT IN EXERCISER TEST''
1253	002230	024064	DH6	;'TEST# ERR PC RCSR''
1254	002232	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1255	002234	000000	0	
1256				
1257	002236	023445	EM110	;'INCORRECT RECEIVE COUNT
1258	002240	024364	DH110	;'TEST# ERR PC RCSR TRANS RCV''
1259	002242	024716	DT110	;\$TESTN,\$ERRPC,RCSR,XMTCNT,RCVCNT
1260	002244	000000	0	
1261				
1262	002246	023475	EM111	;'DATA COMPARE ERROR IN EXERCISER''
1263	002250	024320	DH105	;'TEST# ERR PC RCSR GOOD BAD''
1264	002252	024702	DT105	;\$TESTN,\$ERRPC,RCSR,GD,BD
1265	002254	000000	0	
1266				
1267	002256	023535	EM112	;'TRAP CATCHER''
1268	002260	024430	DH112	;'TEST# ERR PC RCSR OLDPC TRAP ADR''
1269	002262	024732	DT112	;\$TESTN,\$ERRPC,RCSR,OLDPC,BDVECT
1270	002264	000000	0	
1271				
1272	002266	023552	EM113	;'NO CLK INTERRUPT IN EXERCISER''
1273	002270	024136	DH10	;'TEST# ERR PC LKS''
1274	002272	024632	DT10	;\$TESTN,\$ERRPC,LKS
1275	002274	000000	0	
1276				
1277	002276	023610	EM114	;'ERROR' NOT SET WITH 'FR' FLAG''
1278	002300	024064	DH6	;'TEST# ERR PC RCSR''
1279	002302	024612	DT6	;\$TESTN,\$ERRPC,RCSR
1280	002304	000000	0	
1281				
1282	002306	023647	EM115	;'RCV ACTVIE NOT CLEAR WITH INIT
1283	002310	024064	DH6	;'TEST# ERR PC RCSR''
1284	002312	024612	DT6	;\$TESTN,\$ERRPC,RCSR

1285	002314	000000			0	
1286						
1287	002316	023706			EM116	;RCV ACTIVE WITHOUT 'START' BIT
1288	002320	024064			DH6	;'TEST# ERR PC RCSR'
1289	002322	024612			DT6	;\$TESTN,\$ERRPC,RCSR
1290	002324	000000			0	
1291						
1292	002326	023745			EM117	;RDR ENABLE NOT CLEAR WITH RCV ACTIVE
1293	002330	024064			DH6	;'TEST# ERR PC RCSR'
1294	002332	024612			DT6	;\$TESTN,\$ERRPC,RCSR
1295	002334	000000			0	
1296						
1297						;STORAGE LOCATIONS
1298	002336	000000			FIRST: .WORD	0
1299	002340	000000			LOC1: .WORD	0
1300	002342	000000			LCC2: .WORD	0
1301	002344	000000			SAVE0: .WORD	0
1302	002346				SAVLOC: .BLKW	22
1303	002412				ENDSTK: .BLKW	10
1304	002432	000000			JIMSTK: .WORD	0
1305	002434	000000			SAVEPS: .WORD	0
1306	002436	000000			OLDSUM: .WORD	0
1307	002440	000000			RCVCNT: .WORD	0
1308	002442	000000			XMTCNT: .WORD	0
1309	002444	000000			CLKCNT: .WORD	0
1310	002446	000000			STPSW: .WORD	0
1311	002450	000000			SRPSW: .WORD	0
1312	002452	000000			SCPSW: .WORD	0
1313	002454				BUF: .BLKW	44
1314	002564	020	000		CNTLP: .ASCIZ	<20>
1315	002566	136	120	000	PROMPT: .ASCII	/^P/<0><CR><LF>/CONSOLE/<CR><LF>/>>>/<377>
	002571	015	012	103		
	002574	117	116	123		
	002577	117	114	105		
	002602	015	012	076		
	002605	076	076	377		
1316	002610	124	057	101	TA: .ASCIZ	\$T/AS<CR><LF>
	002613	015	012	000		
1317						
1318					.EVEN	
1319	002616	000000			SECND: .WORD	0
1320	002620	000000			OLDPC: .WORD	0
1321	002622	000000			BDVECT: .WORD	0
1322						
1323						
1324						
1325						
1326						
1327	002624	000000			C*STFL: .WORD	0
1328	002626	000000			TMP1: .WORD	0
1329	002630	000000			TMP2: .WORD	0
1330	002632	000000			TMP3: .WORD	0
1331					;REGISTER AND VECTOR ADDRESSES FOR THE DL-11W UNDER TEST	
1332						
1333	002634	000000			RCSR: .WORD	0
1334	002636	000000			RBUF: .WORD	0
1335	002640	000000			TCSR: .WORD	0

```

1336 002642 000000      TBUF:  .WORD  0
1337 002644 000000      RVECT:  .WORD  0
1338 002646 000000      RPSW:   .WORD  0
1339 002650 000000      TVECT:  .WORD  0
1340 002652 000000      TPSW:   .WORD  0
1341
1342      ;CONSOLE REGISTER AND VECTOR ADDRESSES FOR THE DL-11W
1343
1344 002654 177560      CRCSR:  177560      ;ADDRESS OF RECEIVER COMMAND/STATUS REGISTER
1345 002656 177562      CRBUF:  177562      ;ADDRESS OF RECEIVER BUFFER
1346 002660 177564      CTCSR:  177564      ;ADDRESS OF TRANSMITTER COMMAND/STATUS REGISTER
1347 002662 177566      CTBUF:  177566      ;ADDRESS OF TRANSMITTER BUFFER
1348 002664 000060      CRVECT: 60          ;RECEIVER INTERRUPT VECTOR
1349 002666 000062      CRPSW:  62
1350 002670 000064      CTVECT: 64          ;TRANSMITTER INTERRUPT VECTOR
1351 002672 000066      CTPSW:  66
1352
1353      ;REAL TIME CLOCK REGISTER AND VECTOR ADDRESSES
1354 002674 177546      LKS:    .WORD  177546
1355 002676 000100      RTCVT:  .WORD  100
1356 002700 000102      RTCPSW: .WORD  102
1357
1358 002702      ADRTBL:  .BLKW  20
1359 002742      VCTTBL:  .BLKW  20
1360 003002 000000      FLAG44: .WORD  0
1361 003004 176500      TURCSR: .WORD  176500
1362 003006 176502      TURBUF: .WORD  176502
1363 003010 176504      TUTCSR: .WORD  176504
1364 003012 176506      TUTBUF: .WORD  176506
1365
1366      ;SUBROUTINE TO GENERATE DEVICE ADDRESS TABLE
1367
1368
1369 003014 012702 002702      DEVADR: MOV    #ADRTBL,R2      ;POINT R2 TO THE DEVICE ADDRESS TABLE
1370 003020 016700 176116      MOV    $BASE,R0      ;LOAD BASE DEVICE ADDRESS IN R0
1371 003024 010001      MOV    R0,R1      ;
1372 003026 062701 000170      ADD    #170,R1      ;POINT R1 TO LAST DEVICE ADDRESS
1373 003032 010022      1$:    MOV    R0,(R2)+      ;MOVE DEVICE ADDRESS TO TABLE
1374 003034 062700 000010      ADD    #10,R0      ;POINT R0 TO NEXT DEVICE ADDRESS
1375 003040 020001      CMP     R0,R1      ;FINISHED GENERATING TABLE?
1376 003042 003773      BLE     1$          ;BR, IF LAST DEVICE ADDRESS NOT LOADED
1377 003044 000207      RTS     PC
1378
1379
1380
1381 003046 005067 176016      START: CLR    $FATAL      ;CLEAR ERROR NO.
1382 003052 005067 176010      CLR    $MSGTYP      ;CLEAR MESSAGE TYPE
1383 003056 005067 176010      CLR    $TESTN      ;CLEAR TEST NO.
1384 003062 005067 177536      CLR    CTSTFL      ;CLEAR CONSOLE UNDER TEST FLAG
1385 003066 005067 176004      CLR    $DEVCT      ;CLEAR DEVICE COUNT
1386 003072 005067 176002      CLR    $UNIT      ;CLEAR UNIT NUMBER
1387 003076 005067 177700      CLR    FLAG44      ;** CLEAR 11/44 CPU FLAG
1388 003102 005767 176004      TST    $USWR      ;IS $USWR LOADED?
1389 003106 001003      BNE     1$          ;BR IF YES
1390 003110 012767 000400 175774      MOV    #400,$USWR      ;ELSE, DEFAULT TO $USWR=400
1391 003116 012737 000006 000004      1$:  MOV    #6,$A4      ;INITIALIZE TIMEOUT VECTORS TO TRAP
1392 003124 012737 000003 000006      MOV    #3,$A6      ;CATCHER ROUTINE

```

1393
1394

```

.SBTTL INITIALIZE THE COMMON TAGS
;;CLEAR THE COMMON TAGS ($CMTAG) AREA
003132 012706 001000      MOV    # $CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
003136 005026             CLR    (R6)+          ;;CLEAR MEMORY LOCATION
003140 022706 001040      CMP    #SWR,R6      ;;DONE?
003144 001374             BNE    -6             ;;LOOP BACK IF NO
003146 012706 001000      MOV    #1000,SP      ;;SETUP THE STACK POINTER
;;INITIALIZE A FEW VECTORS
003152 012737 015272 000020  MOV    # $SCOPE,@#IOTVEC  ;;IOT VECTOR FOR SCOPE ROUTINE
003160 012737 000340 000022  MOV    #340,@#IOTVEC+2  ;;LEVEL 7
003166 012737 014576 000030  MOV    # $ERROR,@#EMTVEC  ;;EMT VECTOR FOR ERROR ROUTINE
003174 012737 000340 000032  MOV    #340,@#EMTVEC+2  ;;LEVEL 7
003202 012737 017246 000034  MOV    # $TRAP,@#TRAPVEC  ;;TRAP VECTOR FOR TRAP CALLS
003210 012737 000340 000036  MOV    #340,@#TRAPVEC+2  ;;LEVEL 7
003216 012737 015114 000024  MOV    # $PWRDN,@#PWRVEC  ;;POWER FAILURE VECTOR
003224 012737 000340 000026  MOV    #340,@#PWRVEC+2  ;;LEVEL 7
003232 016767 011172 011162  MOV    $ENDCT,$EOPCT  ;;SETUP END-OF-PROGRAM COUNTER
003240 005067 175614             CLR    $ESCAPE      ;;CLEAR THE ESCAPE ON ERROR ADDRESS
003244 112767 000001 175543  MOV    #1,$ERMAX      ;;ALLOW ONE ERROR PER TEST
003252 012767 003252 175526  MOV    #.,$LPADR      ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
003260 012767 003260 175522  MOV    #.,$LPERR      ;;SETUP THE ERROR LOOP ADDRESS
;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
;;EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
003266 013746 000004             MOV    @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
003272 012737 003326 000004  MOV    #64,$@#ERRVEC  ;;SET UP ERROR VECTOR
003300 012767 177570 175532  MOV    #DSWR,SWR      ;;SETUP FOR A HARDWARE SWICH REGISTER
003306 012767 177570 175526  MOV    #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003314 022777 177777 175516  CMP    #-1,@SWR      ;;TRY TO REFERENCE HARDWARE SWR
003322 001012             BNE    66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
;;AND THE HARDWARE SWR IS NOT -1
003324 000403             BR     65$             ;;BRANCH IF NO TIMEOUT
003326 012716 003334 64$:     MOV    #65$,(SP)      ;;SET UP FOR TRAP RETURN
003332 000002             RTI
003334 012767 000176 175476 65$:     MOV    #SWREG,SWR      ;;POINT TO SOFTWARE SWR
003342 012767 000174 175472     MOV    #DISPREG,DISPLAY
003350 012637 000004 66$:     MOV    (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
003354 005067 175514             CLR    $PASS      ;;CLEAR PASS COUNT
003360 132767 000200 175521  BITB   #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
003366 001403             BEQ    67$             ;;YES,USE NON-APT SWITCH
003370 012767 001110 175442  MOV    # $SWREG,SWR      ;;NO,USE APT SWITCH REGISTER
003376
67$:

```

1395
1396
1397

```

;SIZE FOR 11/44 CPU
1398 003376 013746 000010      MOV    @#10,-(SP)      ;;** SAVE VECTOR
1399 003402 012737 003430 000010  MOV    #10$,@#10      ;;** SET UP FOR TRAP
1400 003410 000007             MFPT              ;;** WHAT CPU??
1401 003412 122700 000001      CMPB   #1,R0          ;;** ARE WE A 11/44
1402 003416 001007             BNE    11$            ;;** NO GO RESET VECTOR
1403 003420 052737 000001 003002  BIS    #1,@#FLAG44  ;;** YES SET FLAG
1404 003426 000403             BR     11$            ;;** SKIP RTI
1405 003430 012716 003436 10$:     MOV    #11$,(SP)      ;;** SET STACK RETURN
1406 003434 000002             RTI                  ;;** RETURN
1407 003436 012637 000010 11$:     MOV    (SP)+,@#10      ;;** RESTORE VECTOR
1408 003442 032777 000020 175370  BIT    #BIT4,@SWR      ;;TEST CLOCK ONLY?

```

```

1409 003450 001404      BEQ      INIT      ;BR IF NOT
1410 003452 005267 177146 INC      CTSTFL    ;ELSE, SET CONSOLE TEST FLAG TO ENABLE CLOCK TESTS
1411 003456 000167 001136 JMP      ID      ; AND JUMP TO TYPE PROGRAM ID
1412 003462 132767 000001 175416 INIT:  BITB    #BIT0,$ENV    ;CHECK IF ON APT
1413 003470 001404      BEQ      MANL      ;BR IF NOT APT
1414 003472 132767 000200 175407      BITB    #BIT7,$ENVM    ;DID APT SIZE
1415 003500 001056      BNE      APTSZD    ;BR, IF APT SIZED
1416 003502 032777 000040 175330 MANL:  BIT      #BIT5,@SWR    ;WAS '$DEV' MANUALLY SET?
1417 003510 001052      BNE      APTSZD    ;IF YES, SKIP SELF-SIZING
1418
1419 003512 004767 177276      SIZE:  JSR      PC,DEVADR    ;GENERATE DEVICE ADDRESS TABLE
1420 003516 005067 177106      CLR      TMP2      ;CLR TEMP LOCATION TO KEEP DEVICE COUNT
1421 003522 005067 175416      CLR      $DEV      ;CLEAR DEVICE MAP
1422 003526 013703 000004      MOV      @#4,R3      ;SAVE TIMEOUT VECTOR
1423 003532 012737 003562 000004      MOV      #4$,@#4      ;SET TIMEOUT POINTER
1424 003540 016700 175376      MOV      $BASE,R0    ;LOAD BASE ADDRESS
1425 003544 062700 000160      ADD      #160,R0    ;POINT R0 TO UNIT #15 (UNIT#0-CONSOLE)
1426 003550 005710      3$:  TST      (R0)      ;CHECK FOR DEVICE EXISTANCE
1427 003552 005267 175366      INC      $DEV      ;INDICATE DEVICE EXISTANCE IN DEVICE MAP
1428 003556 005267 177046      INC      TMP2      ;INCREMENT DEVICE COUNT
1429 003562 012706 001000      4$:  MOV      #1000,SP    ;RESET STACK POINTER
1430 003566 006367 175352      ASL      $DEV      ;ADJUST DEVICE MAP FOR NEXT UNIT CHECK
1431 003572 162700 000010      SUB      #10,R0      ;POINT R0 TO NEXT DEVICE NUMBER
1432 003576 026700 175340      CMP      $BASE,R0    ;FINISHED SIZING?
1433 003602 003762      BLE      3$      ;BR, IF BASE ADDRESS HAS NOT BEEN CHECKED
1434 003604 016700 177044      MOV      CRCSR,R0    ;LOAD CONSOLE DEVICE ADDRESS
1435 003610 012737 003630 000004      MOV      #5$,@#4      ;SET UP TIMEOUT POINTER
1436 003616 005710      TST      (R0)      ;TEST FOR CONSOLE EXISTANCE
1437 003620 005267 175320      INC      $DEV      ;INDICATE CONSOLE EXISTANCE IN DEVICE MAP
1438 003624 005267 177000      INC      TMP2      ;INCREMENT DEVICE COUNT
1439 003630 010337 000004      5$:  MOV      R3,@#4      ;RESTORE TIMEOUT VECTOR
1440
1441 003634 000415      BR      VCTADR    ;BR TO GENERATE VECTOR ADDRESS TABLE
1442
1443 003636 005067 176766      APTSZD: CLR      TMP2      ;CLEAR TEMP LOCATION TO KEEP DEVICE CNT
1444 003642 016702 175276      MOV      $DEV,R2      ;MOVE DEVICE MAP TO R2
1445 003646 005702      TSTDVM: TST      R2      ;TEST MSB OF DEVICE MAP
1446 003650 100002      BPL      1$      ;BR, IF MSB IS ZERO
1447 003652 005267 176752      INC      TMP2      ;INCREMENT DEVICE COUNT, IF MSB=1
1448 003656 006302      1$:  ASL      R2      ;SHIFT NEXT BIT INTO MSB POSITION
1449 003660 001401      BEQ      DVADT      ;BR, IF NO OTHER BITS ARE SET IN $DEV
1450 003662 000771      BR      TSTDVM    ;CONTINUE CHECKING $DEV, IF MORE BITS SET
1451 003664 004767 177124      DVADT: JSR      PC,DEVADR    ;GENERATE DEVICE ADDRESS TABLE
1452
1453      ;GENERATE VECTOR ADDRESS TABLE
1454
1455 003670 012702 002742      VCTADR: MOV      #VCTTBL,R2    ;GET LOCATION OF VECTOR TABLE
1456 003674 113700 001136      MOV      @#$VECT1,R0    ;COPY BASE VECTOR
1457 003700 042700 177400      BIC      #177400,R0    ;CLEAR BYTE SIGN EXTENSION
1458 003704 010001      MOV      R0,R1      ;
1459 003706 062701 000170      ADD      #170,R1      ;POINT R1 TO LAST DEVICE VECTOR
1460 003712 010022      1$:  MOV      R0,(R2)+    ;PUT VECTOR ADDRESS IN TABLE
1461 003714 062700 000010      ADD      #10,R0      ;POINT R0 TO NEXT VECTOR ADDRESS
1462 003720 020001      CMP      R0,R1      ;FINISHED GENERATING VECTOR TABLE?
1463 003722 003773      BLE      1$      ;BR, IF LAST VECTOR IS NOT LOADED
1464
1465      ;MOVE DEVICE COUNT INTO DEVICE COUNT MESSAGE

```

```

1466
1467 003724 016700 176700      MOV    TMP2,R0      ;COPY DEVICE COUNT INTO R0
1468 003730 005001      CLR     R1        ;CLEAR AUXILIARY REGISTER
1469 003732 000300      SWAB    R0        ;PUT DEVICE COUNT IN UPPER BYTE OF R0
1470 003734 006300      ASL     R0        ;MOVE MSB OF COUNT INTO
1471 003736 006300      ASL     R0        ;MSB OF R0
1472 003740 006300      SHIFT: ASL     R0        ;PUT MSB OF COUNT INTO CARRY
1473 003742 106101      ROLB    R1        ;MOVE MSB OF COUNT INTO R1
1474 003744 006300      ASL     R0        ;MOVE NEXT BIT TO CARRY
1475 003746 106101      ROLB    R1        ;MOVE INTO R1
1476 003750 006300      ASL     R0        ;MOVE LAST BIT OF DIGIT
1477 003752 106101      ROLB    R1        ;INTO R1
1478 003754 062701 000060      ADD    #60,R1      ;CONVERT DIGIT TO ASCII
1479 003760 000301      SWAB    R1        ;MOVE DIGIT TO UPPER BYTE
1480 003762 032701 000020      BIT    #BIT4,R1     ;HAVE BOTH DIGITS BEEN MOVED TO R1?
1481 003766 001764      BEQ     SHIFT      ;BR, IF NOT
1482 003770 010167 020546      MOV    R1,M2A      ;MOVE DEVICE COUNT TO OUTPUT MESSAGE
1483
1484
1485 003774 052767 000002 176630 BEGIN: BIS    #BIT1,TMP3      ;SET UP BIT MASK TO TEST $DEVM FOR DEVICES EXCEPT CONSOLE
1486 004002 005067 176620      CLR     TMP1      ;CLEAR LOCATION TO STORE TABLE OFFSETS
1487 004006 032767 000001 175130      BIT    #BIT0,$DEVM      ;IS CONSOLE TO BE TESTED?
1488 004014 001001      BNE     TCONS      ;BR, IF CONSOLE IS TO BE TESTED
1489 004016 000414      BR      TSTDEV      ;BR, TO TEST OTHER DEVICES
1490 004020 005267 176600      TCONS: INC    CTSTFL      ;INDICATE CONSOLE UNDER TEST
1491 004024 012700 002654      MOV    #CRCSR,R0      ;SET UP CONSOLE DEVICE ADDRESSES
1492 004030 012701 002634      MOV    #RCSR,R1      ;POINT R1 TO UUT ADDRESS TABLE
1493 004034 012021      1$:    MOV    (R0)+,(R1)+      ;TRANSFER CONSOLE ADDRESSES
1494 004036 022701 002652      CMP     #TPSW,R1      ;FINISHED TRANSFER?
1495 004042 002374      BGE     1$          ;BR, IF NOT
1496 004044 000167 000122      JMP     TST1        ;GO TEST CONSOLE INTERFACE
1497
1498      ;PREPARE ADDRESSES AND VECTORS FOR UUT
1499 004050 036767 176556 175066 TSTDEV: BIT    TMP3,$DEVM      ;CHECK TO SEE IF DEVICE IS TO BE TESTED
1500 004056 001010      BNE     SETADR      ;BR, IF YES
1501 004060 006367 176546      ASL     TMP3        ;SHIFT MASK TO CHECK NEXT $DEVM BIT
1502 004064 062767 000002 176534      ADD    #2,TMP1      ;INCREMENT TABLE INDEX
1503 004072 005267 175002      INC     $UNIT      ;INCREMENT UNIT NUMBER
1504 004076 000764      BR      TSTDEV      ;GO TEST NEXT BIT OF DEVICE MAP
1505
1506 004100 005267 174774      SETADR: INC    $UNIT      ;UPDATE UNIT NUMBER
1507 004104 006367 176522      ASL     TMP3        ;UPDATE DEVICE MAP TEST MASK
1508 004110 016702 176512      MOV    TMP1,R2      ;MOVE TABLE OFFSET TO R2
1509 004114 062767 000002 176504      ADD    #2,TMP1      ;UPDATE TABLE OFFSET FOR NEXT DEVICE
1510 004122 016200 002702      MOV    ADRTBL(R2),R0      ;PUT UUT ADDRESS INTO R0
1511 004126 012701 002634      MOV    #RCSR,R1      ;POINT R1 TO STORAGE AREA FOR UUT ADDRESSES
1512 004132 010021      ADR:    MOV    R0,(R1)+      ;TRANSFER UUT ADDRESS
1513 004134 062700 000002      ADD    #2,R0        ;POINT TO NEXT UUT REGISTER
1514 004140 030027 000006      BIT    R0,#6        ;FINISHED TRANSFER?
1515 004144 001372      BNE     ADR        ;BR, IF NOT
1516
1517 004146 016200 002742      VECT:  MOV    VCTTBL(R2),R0      ;PUT UUT VECTOR INTO R0
1518 004152 010021      MOV    R0,(R1)+      ;TRANSFER UUT VECTORS TO ACTIVE TABLE AREA
1519 004154 062700 000002      ADD    #2,R0        ;POINT TO NEXT VECTOR
1520 004160 030027 000006      BIT    R0,#6        ;FINISHED TRANSFER?
1521 004164 001372      BNE     VECT        ;BR, IF NOT
1522 004166 000167 000000      JMP     TST1        ;GO TEST DEVICE

```

.MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 77-12
INITIALIZE THE COMMON TAGS

M 3

SEQ 0038

1523
1524

1526
1527
1528

1529 004172 000004
1530 004174 013703 000004
1531 004200 012737 004214 000004
1532 004206 005777 176426
1533 004212 000412
1534 004214 022626
1535 004216 005767 176402
1536 004222 001002
1537 004224 104001
1538 004226 000404
1539 004230
004230 004767 011226
004234 000001
1540 004236 000000
1541 004240 010337 000004
1542
1543
1544
1545

```
*****  
*TEST 1      TEST ABILITY TO REFERENCE TCSR  
*****  
TST1:  SCOPE  
      MOV    @#4,R3          ;SAVE TIMEOUT VECTOR  
      MOV    #1$,@#4        ;SET UP TIMEOUT VECTOR  
      TST    @TCSR          ;REFERENCE THE XMIT COMMAND/STATUS REG.  
      BR     4$             ;GO TO END OF TEST  
  
1$:    CMP    (SP)+,(SP)+    ;RESTORE SP AFTER TIMEOUT  
      TST    CTSTFL         ;CHECK IF DEVICE IS CONSOLE  
      BNE    2$             ;IF YES, SKIP ERROR TYPEOUT  
      ERROR+1 ;REPORT ERROR TO APT & TTY  
      BR     4$             ;BR TO END OF TEST  
  
2$:    JSR    PC,$ATY4       ;:ONLY REPORT A FATAL ERROR  
      1      ;:THE ERROR NUMBER (FROM APT LIST)  
  
3$:    HALT  
4$:    MOV    R3,@#4         ;RESTORE TIMEOUT VECTOR
```

1546 004244 000004
1547 004246 013703 000004
1548 004252 012737 004266 000004
1549 004260 005777 176356
1550 004264 000412
1551 004266 022626
1552 004270 005767 176330
1553 004274 001002
1554 004276 104002
1555 004300 000404
1556 004302
004302 004767 011154
004306 000002
1557 004310 000000
1558 004312 010337 000004

```
*****  
*TEST 2      TEST ABILITY TO REFERENCE TBUF  
*****  
TST2:  SCOPE  
      MOV    @#4,R3          ;SAVE TIMEOUT VECTOR  
      MOV    #1$,@#4        ;SET UP TIMEOUT VECTOR  
      TST    @TBUF          ;REFERENCE THE XMIT BUFFER  
      BR     4$             ;GO TO END OF TEST  
  
1$:    CMP    (SP)+,(SP)+    ;RESTORE SP AFTER TIMEOUT  
      TST    CTSTFL         ;CHECK IF DEVICE IS CONSOLE  
      BNE    2$             ;IF YES, SKIP ERROR TYPEOUT  
      ERROR+2 ;REPORT ERROR TO APT & TTY  
      BR     4$             ;BR TO END OF TEST  
  
2$:    JSR    PC,$ATY4       ;:ONLY REPORT A FATAL ERROR  
      2      ;:THE ERROR NUMBER (FROM APT LIST)  
  
3$:    HALT  
4$:    MOV    R3,@#4         ;RESTORE TIMEOUT VECTOR
```

1560
1561
1562

: *TEST 3 TEST THAT BIT2(MAINT. BIT) CAN BE SET & RESET

TST3: SCOPE
RESET ;CLEAR EVERYTHING
BIT #BIT2,@TCSR ;TEST FOR BIT2 OF TCSR CLEAR
BEQ 3\$;BR IF CLEAR

TST CTSTFL ;CHECK IF DEVICE IS CONSOLE
BNE 1\$;IF YES, SKIP ERROR TYPEOUT
ERROR+15 ;BIT2 OF TCSR NOT CLEAR AFTER RESET
BR 3\$

1571
1572 004344 000004 011112 1\$:
004344 004767 011112 JSR PC,\$ATY4 ;:ONLY REPORT A FATAL ERROR
004350 000015 15 ;:THE ERROR NUMBER (FROM APT LIST)
1573 004352 000000 2\$:
1574 HALT

1575 004354 052777 000004 176256 3\$:
1576 004362 032777 000004 176250 BIS #BIT2,@TCSR ;SET BIT2 OF TCSR
1577 004370 001001 BIT #BIT2,@TCSR ;TEST FOR BIT2 SET
BNE 4\$;BR IF SET

1578
1579 004372 104016 ERROR+16 ;BIT2 OF TCSR WILL NOT SET

1580
1581 004374 042777 000004 176236 4\$:
1582 004402 032777 000004 176230 BIC #BIT2,@TCSR ;CLEAR BIT2 OF TCSR
1583 004410 001411 BIT #BIT2,@TCSR ;TEST BIT2 CLEAR
BEQ 7\$;BR IF CLEAR

1584
1585 004412 005767 176206 TST CTSTFL ;CHECK IF DEVICE IS CONSOLE
1586 004416 001002 BNE 5\$;IF YES, SKIP ERROR TYPEOUT
1587 004420 104017 ERROR+17
1588 004422 000404 BR 7\$

1589 004424 000000 5\$:
004424 004767 011032 JSR PC,\$ATY4 ;:ONLY REPORT A FATAL ERROR
004430 000017 17 ;:THE ERROR NUMBER (FROM APT LIST)
1590 004432 000000 6\$:
1591 HALT ;BIT0 OF TCSR WILL NOT CLEAR

1592 004434 052777 000004 176176 7\$:
1593 004442 000005 BIS #BIT2,@TCSR ;SET BIT2 OF TCSR
1594 004444 032777 000004 176166 RESET ;CLEAR BIT2 WITH RESET
1595 004452 001404 BIT #BIT2,@TCSR ;TEST FOR BIT2 CLEAR
BEQ 10\$;** IF CLEAR, GO TO NEXT TEST

1596
1597 004454 042777 000004 176156 BIC #BIT2,@TCSR ;CLEAR BIT2. TO PRINT ERROR
1598 004462 104020 ERROR+20 ;RESET DID NOT CLEAR BIT2 OF TCSR
1599

1600
1601
1602

10\$: NOP ;**

: *TEST 4 TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED

TST4: SCOPE
BIS #BIT2,@TCSR ;** ENABLE MAINT. WRAP
CLR @TBUF ;LOAD XBUF
TSTB @TCSR ;CHECK DONE
BPL 3\$;BR IF CLEAR

1603 004466 000004 000004 176142
1604 004470 052777 176140
1605 004476 005077 176132
1606 004502 105777
1606 004506 100021

```

1607                                     ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
1608                                     ; FIRST TEST TO FAIL
1609 004510 005077 176126          CLR    @TBUF          ;FILL DOUBLE BUFFER
1610 004514 105777 176120          TSTB   @TCSR          ;CHECK DONE
1611 004520 100014                  BPL     3$            ;BR IF CLEAR
1612
1613 004522 005767 176076          TST     CTSTFL         ;CHECK IF DEVICE IS CONSOLE
1614 004526 001005                  BNE     1$            ;IF YES, SKIP ERROR TYPEOUT
1615 004530 042777 000004 176122  BIC     #BIT2,@CTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                     ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                     ;** WITH TERMINAL.
1616 004536 104003          ERROR+3 3$          ;DONE NOT CLEARED WITH TBUF FULL
1617 004540 000404          BR         3$            ;BR TO END OF TEST
1618 004542                  1$:
      004542 004767 010714          JSR     PC,$ATY4         ;: ONLY REPORT A FATAL ERROR
      004546 000003                  3          ;: THE ERROR NUMBER (FROM APT LIST)
1619 004550 000000          2$: HALT
1620 004552 005000          3$: CLR     R0              ;TCSR 'DONE' NOT CLEARED WITH TBUF FULL
1621 004554 105777 176060          4$: TSTB   @TCSR          ;CLEAR TIMER
1622 004560 100417          BMI     ID              ;CHECK FOR XMIT DONE
1623 004562 005200          INC     R0              ;IF DONE SETS, BR TO END OF TEST
1624 004564 001373          BNE     4$            ;INCREMENT TIMER
1625                                     ;BR IF TIMER NOT DONE
1626 004566 005767 176032          TST     CTSTFL         ;CHECK IF DEVICE IS CONSOLE
1627 004572 001005          BNE     5$            ;** DISABLE MAINTENANCE MODE FOR
1628 004574 042777 000004 176056  BIC     #BIT2,@CTCSR   ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                     ;** WITH TERMINAL.
1629 004602 104004          ERROR+4 5$:          ;TCSR 'DONE' DOES NOT SET
1630 004604 000405          BR         ID            ;BR TO END OF TEST
1631 004606          5$:
      004606 004767 010650          JSR     PC,$ATY4         ;: ONLY REPORT A FATAL ERROR
      004612 000004                  4          ;: THE ERROR NUMBER (FROM APT LIST)
1632 004614 000000          HALT
1633 004616 000427          BR         ENDB7         ;** BR TO NEXT TEST,
1634                                     ;** AND SKIP THE TYPEOUT THAT FOLLOWS
1635                                     ;** BECAUSE OF THIS FAILURE
1636
1637 004620          ID:
1638 004620 042777 000004 176032  BIC     #BIT2,@CTCSR   ;** DISABLE MAINTENANCE MODE FOR
                                     ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                     ;** WITH TERMINAL.
1639 004626 023737 000042 000046  CMP     @#42,@#46      ;UNDER ACT11?
1640 004634 001412          BEQ     6$            ;IF YES, SKIP IDENT. TYPEOUT
1641 004636 005767 174232          TST     $PASS         ;IS THIS THE FIRST PASS?
1642 004642 001007          BNE     6$            ;IF NOT BR TO NFXT TEST & SKIP THE IDENTIFICATION TYPEOUTS
1643 004644 005767 174226          TST     $DEVCT        ;IS THIS THE FIRST SUBPASS?
1644 004650 001004          BNE     6$            ;IF NOT, BR TO NEXT TEST
1645 004652 104401          TYPE
1646 004654 024502          M1
1647 004656 104401          TYPE
1648 004660 024540          M2
1649 004662 032777 000020 174150 6$: BIT     #BIT4,@SWR   ;CLOCK TEST ONLY?
1650 004670 001402          BEQ     ENDB7         ;** BR IF NOT
1651 004672 000167 000640          JMP     TCLOCK        ;ELSE, JUMP TO TEST CLOCK
1652 004676          ENDB7:
1653 004676 005000          CLR     R0              ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS

```

.MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 79-2
T4 TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED

SEQ 0042

004700 012701 000002
004704 005300
004706 001376
004710 005301
004712 001374

40\$: MOV #2,R1
DEC R0
BNE 40\$
DEC R1
BNE 40\$

:** THAT MIGHT BE IN THE PROCESS OF BEING
:** TRANSMITTED TO FINISH BEFORE MAINTENANCE
:** WRAP FOR THE UART UNDER TEST IS DISABLED.
:** THIS WILL INHIBIT ANY COMMUNICATION
:** TO HARDWARE MEDIA SUCH AS TU58 THAT MIGHT
:** BE ATTACHED TO UART UNDER TEST.

1655
 1656
 1657

 :*TEST 5 TEST THAT TCSR 'DONE' SETS WITH RESET

004714 000004
 1658 004716 052777 000004 175714
 1659 004724 005077 175712
 1660 004730 105777 175704
 1661 004734 100375
 1662 004736 005077 175700
 1663 004742 000240
 1664 004744 000005
 1665 004746 105777 175666
 1666 004752 100401
 1667
 1668 004754 104005
 1669
 1670 004756 000240
 1671
 1672
 1673

TST5: SCOPE
 BIS #BIT2,@TCSR ;** ENABLE MAINT. WRAP
 CLR @TBUF ;LOAD TRANSMIT BUFFER
 1\$: TSTB @TCSR ;WAIT FOR DONE
 BPL 1\$
 CLR @TBUF ;LOAD SECOND BUFFER
 NOP
 RESET ;CLEAR DONE WITH RESET
 TSTB @TCSR ;CHECK FOR DONE SET
 BMI 10\$;** BR TO NEXT TEST IF DONE SET
 ERROR+5 ;TCSR 'DONE' DOES NOT SET WITH RESET
 10\$: NOP ;**

 :*TEST 6 TEST ABILITY TO ACCESS RCSR

004760 000004
 1674 004762 013703 000004
 1675 004766 012737 005002 000004
 1676 004774 005777 175634
 1677 005000 000402
 1678
 1679 005002 022626
 1680 005004 104006
 1681 005006 010337 000004

TST6: SCOPE
 MOV @#4,R3 ;SAVE TIMEOUT VECTOR
 MOV #1\$,@#4 ;SET UP TIMEOUT VECTOR
 TST @RCSR ;ACCESS RCSR
 BR 2\$;BR TO END OF TEST
 1\$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
 ERROR+6 ;CAN NOT ACCESS RCSR
 2\$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR

1683
1684
1685

: *TEST 7 TEST ABILITY TO ACCESS RBUF

1686 005012 000004
1687 005014 013703 000004
1688 005020 012737 005034 000004
1689 005026 005777 175604
1690 005032 000402

TST7: SCOPE
MOV @#4,R3 ;SAVE TIMEOUT VECTOR
MOV #1\$,@#4 ;SET UP TIMEOUT VECTOR
TST @RBUF ;ACCESS RBUF
BR 2\$;BR TO END OF TEST

1691 005034 022626
1692 005036 104007
1693 005040 010337 000004

1\$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
ERROR+7 ;CAN NOT ACCESS RBUF
2\$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR

1694
1695
1696
1697

1699
 1700
 1701

```

*****
*TEST 10      TEST THAT BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
*****
TST10: SCOPE
1702 005044 000004 000400 173764 BIT #BIT8,@SWR ;IS BREAK FUNCTION ENABLED?
1703 005054 001475 000400 173764 BEQ 10$ ;** BR TO NEXT TEST, IF NOT
1704 005056 032767 000001 175716 BIT #BIT0,FLAG44 ;** IS THIS A 11/44
1705 005064 001407 000001 175716 BEQ 9$ ;** NO
1706 005066 005767 175532 TST CTSTFL ;** YES THIS IS 11/44. IS THIS THE CONSOLE
1707 005066 005767 175532 BEQ 9$ ;** SLU
1708 005072 001404 000010 173736 BIT #BIT03,@SWR ;** NO
1709 005074 032777 000010 173736 BEQ 10$ ;** THIS IS THE CONSOLE SLU.SHOULD THE BREAK
1710 005102 001462 000010 173736 BEQ 10$ ;** TEST BE PERFORMED
1711 005102 001462 000010 173736 BEQ 10$ ;** NO
1712 005104 000005 000001 175524 9$: RESET ;CLEAR EVERYTHING
1713 005104 000005 000001 175524 BIT #BIT0,@TCSR ;CHECK BIT0 OF TCSR CLEAR
1714 005106 032777 000001 175524 BEQ 3$ ;BR IF CLEAR
1715 005114 001411 000001 175524 TST CTSTFL
1716 005116 005767 175502 BNE 1$ ;BIT0 WAS NOT CLEAR AFTER RESET
1717 005122 001002 000001 175502 ERROR+11 BR 3$
1718 005124 104011 000001 175502 1$: JSR PC,$ATY4 ;:ONLY REPORT A FATAL ERROR
1719 005126 000404 000001 175502 1$ 11 ;:THE ERROR NUMBER (FROM APT LIST)
1720 005130 000000 000001 175502 2$: HALT
1721 005136 000000 000001 175502 3$: BIS #BIT0,@TCSR ;SET BIT0 IN TCSR
1722 005140 052777 000001 175472 BIT #BIT0,@TCSR ;TEST BIT0 OF TCSR
1723 005146 032777 000001 175464 BNE 4$ ;BR IF SET
1724 005154 001001 000001 175464 ERROR+12 ;BIT0 OF TCSR WILL NOT SET
1725 005156 104012 000001 175464 4$: BIC #BIT0,@TCSR ;CLEAR BIT0 OF TCSR
1726 005160 042777 000001 175452 BIT #BIT0,@TCSR ;TEST BIT0 OF TCSR
1727 005166 032777 000001 175444 BEQ 7$
1728 005174 001411 000001 175444 TST CTSTFL
1729 005176 005767 175422 BNE 5$ ;BIT0 OF TCSR WILL NOT CLEAR
1730 005202 001002 000001 175422 ERROR+13 BR 7$
1731 005204 104013 000001 175422 5$: JSR PC,$ATY4 ;:ONLY REPORT A FATAL ERROR
1732 005206 000404 000001 175422 5$ 13 ;:THE ERROR NUMBER (FROM APT LIST)
1733 005210 000000 000001 175422 6$: HALT
1734 005216 000000 000001 175422 7$: BIS #BIT0,@TCSR ;SET BIT0 IN TCSR
1735 005220 052777 000001 175412 RESET ;CLEAR BIT0 WITH RESET
1736 005226 000005 000001 175412 BIT #BIT0,@TCSR ;TEST BIT0 CLEAR
1737 005230 032777 000001 175402 BEQ 10$ ;** BR IF CLEAR
1738 005236 001404 000001 175402 BIC #BIT0,@TCSR ;CLEAR BIT0, TO PRINT ERROR
1739 005240 042777 000001 175372 ERROR+14 ;RESET DID NOT CLEAR BIT0 OF TCSR
1740 005246 104014 000001 175372 10$: NOP ;**
1741 005250 000240 000001 175372

```

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 83
T10 TEST THAT BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET

H 4

SEQ 0046

1747

1749
 1750
 1751

```

005252 000004
1752 005254 000005
1753 005256 017703 175366
1754 005262 012777 005312 175360
1755 005270 004767 007244
1756 005274 000340
1757 005276 032777 000100 175334
1758 005304 001404
1759 005306 104021
1760
1761 005310 000402
1762
1763 005312 022626
1764 005314 104022
1765
1766
1767 005316 052777 000100 175314
1768 005324 032777 000100 175306
1769 005332 001001
1770
1771 005334 104023
1772
1773
1774 005336 042777 000100 175274
1775 005344 032777 000100 175266
1776 005352 001401
1777 005354 104024
1778
1779
1780 005356 052777 000100 175254
1781 005364 000005
1782 005366 032777 000100 175244
1783 005374 001401
1784
1785 005376 104025
1786
1787 005400 010377 175244

TST11: SCOPE
RESET
MOV @TVECT,R3
MOV #1$,@TVECT
JSR PC,WRPSW
        .WORD 340
        #BIT6,@TCSR
BEQ 2$
ERROR+21
BR 2$

1$: CMP (SP)+,(SP)+
ERROR+22

2$: BIS #BIT6,@TCSR
BIT #BIT6,@TCSR
BNE 3$
ERROR+23

3$: BIC #BIT6,@TCSR
BIT #BIT6,@TCSR
BEQ 4$
ERROR+24

4$: BIS #BIT6,@TCSR
RESET
BIT #BIT6,@TCSR
BEQ 5$
ERROR+25

5$: MOV R3,@TVECT

```

```

;CLEAR EVERYTHING
;SAVE XMIT VECTOR
;SET UP INTERRUPT VECTOR FOR ERROR REPORT
;SET PSW TO PRIORITY-7
;TEST BIT6 OF TCSR
;BR IF ZERO
;BIT6 IN TCSR NOT CLEAR AFTER RESET
;RESTORE SP AFTER INTERRUPT
;XMIT INTERRUPT OCCURRED PRIO-7
;SET BIT6 OF TCSR
;TEST BIT6 OF TCSR
;BR, IF SET
;CANNOT SET BIT6 OF TCSR
;CLEAR BIT6 OF TCSR
;TEST BIT6 OF TCSR
;BR IF CLEAR
;CANNOT CLEAR BIT6 OF TCSR
;SET BIT6 OF TCSR
;CLEAR BIT6 WITH RESET
;TEST BIT6 OF TCSR
;BR IF CLEAR
;CANNOT CLEAR BIT6 OF TCSR WITH RESET
;RESTORE XMIT VECTOR

```

1789
 1790
 1791

```

005404 000004
1792 005406 000005
1793 005410 017703 175230
1794 005414 012777 005444 175222
1795 005422 004767 007112
1796 005426 000340
1797 005430 032777 000100 175176
1798 005436 001404
1799 005440 104026
1800
1801 005442 000402
1802
1803 005444 022626
1804 005446 104027
1805
1806
1807 005450 052777 000100 175156
1808 005456 032777 000100 175150
1809 005464 001001
1810
1811 005466 104030
1812
1813
1814 005470 042777 000100 175136
1815 005476 032777 000100 175130
1816 005504 001401
1817
1818 005506 104031
1819
1820
1821 005510 052777 000100 175116
1822 005516 000005
1823 005520 032777 000100 175106
1824 005526 001401
1825
1826 005530 104032
1827
1828 005532 010377 175106
1829
  
```

```

*****
*TEST 12      TEST THAT BIT6 OF RCSR CAN BE SET & RESET
*****

```

```

TST12:  SCOPE
        RESET                ;CLEAR EVERYTHING
        MOV      @RVECT,R3   ;SAVE RECEIVE VECTOR
        MOV      #1$,@RVECT  ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
        JSR      PC,WRPSW    ;SET PSW TO PRIORITY=7
                                .WORD 340
        BIT      #BIT6,@RCSR  ;TEST BIT6 OF RCSR
        BEQ      2$
        ERROR+26
                                ;BIT6 OF RCSR NOT CLEAR AFTER RESET
        BR       2$
1$:      CMP      (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
        ERROR+27
                                ;RCVR INTERRUPT WITH PRIORITY-7
2$:      BIS      #BIT6,@RCSR  ;SET BIT6 OF RCSR
        BIT      #BIT6,@RCSR  ;TEST BIT6 OF RCSR
        BNE      3$           ;BR, IF SET
        ERROR+30
                                ;CANNOT SET BIT6 OF RCSR
3$:      BIC      #BIT6,@RCSR  ;CLEAR BIT6 OF RCSR
        BIT      #BIT6,@RCSR  ;TEST BIT6 OF RCSR
        BEQ      4$           ;BR, IF CLEAR
        ERROR+31
                                ;CANNOT CLEAR BIT6 OF RCSR
4$:      BIS      #BIT6,@RCSR  ;SET BIT6 OF RCSR
        RESET      ;CLEAR BIT6 OF RCSR WITH RESET
        BIT      #BIT6,@RCSR  ;TEST BIT6 OF RCSR
        BEQ      5$           ;BR, IF CLEAR
        ERROR+32
                                ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
5$:      MOV      R3,@RVECT    ;RESTORE RECEIVE VECTOR
  
```

```

1831
1832 005536 012767 000012 173236 TCLOCK: MOV #12,$TSTNM ;ADJUST TEST NUMBER TO (NEXT TEST - 1)
1833 ;*****
;TEST 13 TEST ABILITY TO ACCESS LKS
;*****
TST13: SCOPE
1834 005544 000004 TST CTSTFL ;IS CONSOLE UNDER TEST?
005546 005767 175052 BEQ TST14 ;IF NOT, SKIP THIS TEST
005552 001420 BIT #BIT6,@SWR ;ARE LINE CLOCK TESTS INHIBITED?
005554 032777 000100 173256 BNE TST14 ;IF YES, SKIP THIS TEST
005562 001014 MOV @#4,R3 ;SAVE TIMEOUT VECTOR
1835 005564 013703 000004 MOV #1$,@#4 ;SET UP TIMEOUT VECTOR
1836 005570 012737 005604 000004 TST @LKS ;ACCESS LKS
1837 005576 005777 175072 BR 2$ ;NO TIMEOUT - BR TO END OF TEST
1838 005602 000402
1839
1840 005604 022626 1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
1841 005606 104010 ERROR+10 ;CAN NOT ACCESS LKS
1842
1843 005610 010337 000004 2$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR
1844
1845 ;*****
;TEST 14 TEST THAT BIT6 OF LKS CAN BE SET & RESET
;*****
TST14: SCOPE
1846 005614 000004 TST CTSTFL ;IS CONSOLE UNDER TEST?
005616 005767 175002 BEQ TST15 ;IF NOT, SKIP THIS TEST
005622 001460 BIT #BIT6,@SWR ;ARE LINE CLOCK TESTS INHIBITED?
005624 032777 000100 173206 BNE TST15 ;IF YES, SKIP THIS TEST
005632 001054 RESET
1847 005634 000005 MOV @RTCVT,R3 ;SAVE LINE CLOCK VECTOR
1848 005636 017703 175034 MOV #1$,@RTCVT ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1849 005642 012777 005672 175026 JSR PC,WRPSW ;SET PSW TO PRIORITY 7
1850 005650 004767 006664 .WORD 340
1851 005654 000340 BIT #BIT6,@LKS ;TEST BIT6 OF LKS
1852 005656 032777 000100 175010 BEQ 2$
1853 005664 001404 ERROR+33 ;BIT6 OF LKS NOT CLEAR AFTER RESET
1854 005666 104033 BR 2$
1855
1856 005670 000402
1857
1858 005672 022626 1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1859 005674 104034 ERROR+34 ;LKS INTERRUPT WITH PRIORITY 7
1860
1861
1862 005676 052777 000100 174770 2$: BIS #BIT6,@LKS ;SET BIT6 OF LKS
1863 005704 032777 000100 174762 BIT #BIT6,@LKS ;TEST BIT6 OF LKS
1864 005712 001001 BNE 3$ ;BR IF SET
1865
1866 005714 104035 ERROR+35 ;CANNOT SET BIT6 OF LKS
1867
1868
1869 005716 042777 000100 174750 3$: BIC #BIT6,@LKS ;CLEAR BIT6 OF LKS
1870 005724 032777 000100 174742 BIT #BIT6,@LKS ;TEST BIT6 OF LK
1871 005732 001401 BEQ 4$
1872 005734 104036 ERROR+36 ;CANNOT CLEAR BIT6 OF LKS
1873
1874 005736 052777 000100 174730 4$: BIS #BIT6,@LKS ;SET BIT6 OF LKS
1875 005744 000005 RESET ;CLEAR BIT6 OF LKS WITH RESET

```

.MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 86-1
+14 TEST THAT BIT6 OF LKS CAN BE SET & RESET

L 4

SEQ 0050

```
1876 005746 032777 000100 174720      BIT    #BIT6,@LKS      ;TEST BIT6 OF LKS
1877 005754 001401                      BEQ     5$              ;BR IF CLEAR
1878                                     ERROR+37
1879 005756 104037                      ;CANNOT CLEAR BIT6 OF LKS WITH RESET
1880                                     ;RESTORE LINE CLOCK VECTOR
1881 005760 010377 174712      5$:    MOV     R3,@RTCVT
```

1883
1884
1885

005764 000004
1886 005766 013703 000004
1887 005772 013704 000006
1888 005776 012737 006130 000004
1889 006004 012737 000340 000006
1890 006012 000005
1891 006014 012700 000002
1892 006020 032777 000020 173012
1893 006026 001404
1894 006030 016767 174640 172762
1895 006036 000403
1896 006040 016767 174570 172752 1\$:
1897 006046 016767 172746 172746 2\$:
1898 006054 040067 172742
1899 006060 026767 172734 172734
1900 006066 001002
1901 006070 050067 172726
1902 006074 017767 172722 172722 3\$:
1903 006102 052777 000100 172712
1904 006110 032777 000100 172702
1905 006116 001011
1906 006120 016777 172700 172674
1907 006126 000401
1908 006130 022626 4\$:
1909 006132 006300 5\$:
1910 006134 105700
1911 006136 100343
1912 006140 000401
1913
1914 006142 104040 6\$:
1915
1916
1917
1918 006144 010337 000004 7\$:
1919 006150 010437 000006

*TEST 15 TEST FOR DUAL ADDRESSING OF REGISTERS

TST15: SCOPE
MOV @#4,R3 ;SAVE TIMEOUT VECTOR
MOV @#6,R4 ;SAVE TIMEOUT PSW
MOV #4\$,@#4 ;SET UP TIMEOUT VECTOR
MOV #340,@#6 ;KEEP PRIO=7
RESET ;CLEAR EVERYTHING
MOV #BIT1,R0 ;SET UP BIT MASK
BIT #BIT4,@SWR ;CLOCK TEST ONLY?
BEQ 1\$;BR IF NOT
MOV LKS,\$GDADR ;ELSE, MOVE GOOD LKS ADDRESS INTO \$GDADR
BR 2\$
MOV RCSR,\$GDADR ;MOVE GOOD RCSR ADDRESS INTO \$GDADR
MOV \$GDADR,\$BDADR ;MOVE GOOD ADDRESS INTO TEST ADDRESS LOCATION
BIC R0,\$BDADR ;CREATE BAD ADDRESS BY COMPLEMENTING ONE BIT
CMP \$GDADR,\$BDADR ;ARE ADDRESSES IDENTICAL?
BNE 3\$;IF NOT, TEST THIS ADDRESS
BIS R0,\$BDADR ;ELSE, BIT SET THIS BIT POSITION TO GENERATE BAD ADDRESS
MOV @BDADR,\$GDDAT ;SAVE CONTENTS OF BAD ADDRESS IF IT EXISTS
BIS #BIT6,@BDADR ;SET BIT6 USING BAD ADDRESS
BIT #BIT6,@SGDADR ;CHECK TO SEE IF GOOD ADDRESS CONTAINS BIT6
BNE 6\$;BR IF SET ---> ERROR
MOV \$GDDAT,@BDADR ;RESTORE ANY MEMORY LOCATION THAT WAS ALTERED
BR 5\$;BR TO CONTINUE TEST
CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
ASL R0 ;SHIFT BIT MASK TO NEXT POSITION
TSTB R0 ;COMPLEMENTED ALL BITS FROM 1 - 7?
BPL 2\$;BR, IF NOT.
BR 7\$;BR TO NEXT TEST
6\$: ERROR+40 ;DUAL ADDRESSING ERROR
;BDADR = DUAL ADDRESS
;GDADR = GOOD ADDRESS
7\$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR
MOV R4,@#6 ;RESTORE TIMEOUT PSW


```

1921
1922
1923
1924 006154 000004
      006156 005767 174442
      006162 001437
      006164 032777 000100 172646
      006172 001033
1925 006174 000005
1926 006176 105777 174472
1927 006202 100401
1928
1929 006204 104041
1930
1931 006206 042777 000200 174460
1932 006214 032777 000200 174452
1933 006222 001410
1934 006224 042777 000200 174442
1935 006232 032777 000200 174434
1936 006240 001401
1937
1938 006242 104042
1939
1940 006244 005000
1941 006246 105777 174422
1942 006252 100403
1943 006254 005200
1944 006256 001373
1945
1946 006260 104043
1947

;*****
; *TEST 16      TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
;*****
TST16: SCOPE
        TST      CTSTFL      ;IS CONSOLE UNDER TEST?
        BEQ      TST17      ;IF NOT, SKIP THIS TEST
        BIT      #BIT6,@SWR  ;ARE LINE CLOCK TESTS INHIBITED?
        BNE      TST17      ;IF YES, SKIP THIS TEST
        RESET    ;CLEAR EVERYTHING & SET BIT7 OF LKS
1$:      TSTB     @LKS        ;TEST FOR BIT7 OF LKS
        BMI      2$          ;BR IF SET
                                ;BIT7 OF LKS DID NOT SET WITH RESET
                                ;CLEAR BIT7 OF LKS
2$:      BIC      #BIT7,@LKS  ;TEST BIT7 OF LKS
        BIT      #BIT7,@LKS
        BEQ      3$
        BIC      #BIT7,@LKS  ;TRY ONE MORE TIME BECAUSE THE CLOCK
        BIT      #BIT7,@LKS  ; MAY HAVE SET IMMEDIATELY AFTER THE FIRST CLEAR
        BEQ      3$
                                ;CAN NOT CLEAR BIT7 OF LKS
                                ;CLEAR TIMER
3$:      CLR      R0          ;TEST FOR BIT7 OF LKS
        CONT:    TSTB     @LKS ;BR, IF SET
        BMI      TST17      ;INCREMENT TIMER
        INC      R0          ;CONTINUE UNTIL TIME EXPIRES
        BNE      CONT
                                ;BIT7 OF LKS DOES NOT SET
        ERROR+43
  
```

1949
1950

```

006262 000004
1951 006264 005767 174334
006270 001505
006272 032777 000100 172540
006300 001101
1952 006302 004767 006232
1953 006306 000340
1954 006310 017703 174362
1955 006314 0177C 174360
1956 006320 012777 006364 174350
1957 006326 012777 000340 174344
1958 006334 042777 000200 174332
1959 006342 052777 000100 174324
1960 006350 105777 174320
1961 006354 100375
1962 006356 000240
1963 006360 000240
1964 006362 000402
1965
1966 006364 022626
1967 006366 104044
1968
1969 006370 005077 174300
1970 006374 012777 006424 174274
1971 006402 004767 006132
1972 006406 000240
1973 006410 105777 174260
1974 006414 100375
1975 006416 000240
1976 006420 000240
1977 006422 000402
1978
1979 006424 022626
1980 006426 104045
1981
1982 006430 012777 006464 174240
1983 006436 042777 000200 174230
1984 006444 052777 000100 174222
1985 006452 105777 174216
1986 006456 100375
1987 006460 000240
1988
1989 006462 104046
1990
1991 006464 022626
1992 006466 042777 000100 174200
1993 006474 010377 174176
1994 006500 010477 174174
1995
1996

```

```

*****
: *TEST 17      TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
*****

```

```

TST17:  SCOPE
        TST      CTSTFL      ;IS CONSOLE UNDER TEST?
        BEQ      TST20      ;IF NOT, SKIP THIS TEST
        BIT      #BIT6,@SWR  ;ARE LINE CLOCK TESTS INHIBITED?
        BNE      TST20      ;IF YES, SKIP THIS TEST
        JSR      PC,WRPSW    ;SET PSW TO PRIORITY 7
                                .WORD 340
        MOV      @RTCVT,R3    ;SAVE LINE CLOCK VECTOR
        MOV      @RTCPW,R4    ;SAVE LINE CLOCK PSW VECTOR
        MOV      #2$,@RTCVT   ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
        MOV      #340,@RTCPW  ;KEEP PRIORITY AT 7
        BIC      #BIT7,@LKS   ;CLEAR CLOCK DONE FLAG
        BIS      #BIT6,@LKS   ;SET INTERRUPT ENABLE
1$:      TSTB     @LKS         ;WAIT FOR RTC DONE (INTERRUPT REQUEST)
        BPL      1$
        NOP
        NOP
        BR       3$
                                ;GIVE TIME FOR ANY INTERRUPTS
                                ;GIVE TIME FOR ANY INTERRUPTS
                                ;BR, IF NO INTERRUPT OCCURS
2$:      CMP      (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
        ERROR+44              ;RTC INTERRUPTS AT PRIORITY 7
3$:      CLR      @LKS        ;DISABLE RTC INTERRUPTS & CLEAR DONE
        MOV      #4$,@RTCVT   ;SET RTC INTERRUPT VECTOR FOR ERROR
        JSR      PC,WRPSW    ;CHANGE PSW TO PRIORITY 5
                                .WORD 240
20$:     TSTB     @LKS        ;WAIT FOR DONE (INTERRUPT REQUEST)
        BPL      20$
        NOP
        NOP
        BR       5$
                                ;GIVE TIME FOR ANY INTERRUPT
                                ;GIVE TIME FOR ANY INTERRUPT
                                ;IF NO INTERRUPT - BR TO CONTINUE TEST
4$:      CMP      (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
        ERROR+45              ;RTC INTERRUPTS WITH INTERRUPTS DISABLED
5$:      MOV      #7$,@RTCVT   ;POINT RTC VECTOR TO END OF TEST
        BIC      #BIT7,@LKS   ;CLEAR CLOCK DONE FLAG
        BIS      #BIT6,@LKS   ;ALLOW INTERRUPTS
6$:      TSTB     @LKS        ;WAIT FOR RTC DONE
        BPL      6$
        NOP
                                ;GIVE TIME FOR INTERRUPT
                                ;RTC INTERRUPT DID NOT OCCUR
7$:      CMP      (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
        BIC      #BIT6,@LKS   ;DISABLE INTERRUPTS
        MOV      R3,@RTCVT    ;RESTORE LINE CLOCK VECTOR
        MOV      R4,@RTCPW    ;RESTORE LINE CLOCK PSW VECTOR

```

1998
1999
2000
2001

```

2002 006504 000004
      006506 005767 174112
      006512 001462
      006514 032777 000100 172316
      006522 001056
2003 006524 000005
2004 006526 017703 174144
2005 006532 017704 174142
2006 006536 012777 006606 174132
2007 006544 012777 000340 174126
2008 006552 004767 005762
2009 006556 000240
2010 006560 042777 000200 174106
2011 006566 052777 000100 174100
2012 006574 105777 174074 1$:
2013 006600 100375
2014 006602 000240
2015
2016 006604 104047
2017
2018 006606 022626
2019 006610 012777 006636 174060 2$:
2020 006616 004767 005716
2021 006622 000240
2022 006624 000240
2023 006626 000240
2024 006630 000240
2025 006632 000240
2026 006634 000402
2027
2028 006636 022626 3$:
2029 006640 104050
2030
2031
2032 006642 042777 000100 174024 4$:
2033 006650 010377 174022
2034 006654 010477 174020

;*****
; *TEST 20 TEST RTC FOR DOUBLE INTERRUPTS
;*****
TST20: SCOPE
        TST CTSTFL ;IS CONSOLE UNDER TEST?
        BEQ TST21 ;IF NOT, SKIP THIS TEST
        BIT #BIT6,@SWR ;ARE LINE CLOCK TESTS INHIBITED?
        BNE TST21 ;IF YES, SKIP THIS TEST
        RESET ;CLEAR EVERYTHING
        MOV @RTCVT,R3 ;SAVE LINE CLOCK VECTOR
        MOV @RTCPSW,R4 ;SAVE LINE CLOCK PSW VECTOR
        MOV #2$,@RTCVT ;SET UP RTC INTERRUPT VECTOR
        MOV #340,@RTCPSW ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
        JSR PC,WRPSW ;SET PRIORITY TO 5
        .WORD 240
        BIC #BIT7,@LKS ;CLEAR CLOCK DONE FLAG
        BIS #BIT6,@LKS ;ENABLE CLOCK INTERRUPTS
        TSTB @LKS ;WAIT FOR DONE
        BPL 1$
        NOP ;GIVE TIME FOR ANY INTERRUPT
        ERROR+47 ;RTC INTERRUPT DID NOT OCCUR
        2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
        MOV #3$,@RTCVT ;POINT RTC VECTOR TO ERROR REPORT
        JSR PC,WRPSW ;SET PSW TO PRIORITY 5
        .WORD 240
        NOP ;GIVE SOME TIME FOR AN INTERRUPT
        NOP ;GIVE SOME TIME FOR AN INTERRUPT
        NOP ;GIVE SOME TIME FOR AN INTERRUPT
        NOP ;GIVE SOME TIME FOR AN INTERRUPT
        BR 4$ ;NO INTERRUPT - BR TO END OF TEST
        3$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
        ERROR+50 ;INTERRUPT SEQUENCE DID NOT CLEAR
        ;INTERRUPT REQUEST
        4$: BIC #BIT6,@LKS ;DISABLE CLOCK INTERRUPTS
        MOV R3,@RTCVT ;RESTORE LINE CLOCK VECTOR
        MOV R4,@RTCPSW ;RESTORE LINE CLOCK PSW VECTOR
```

2036
2037
2038

```
006660 000004
2039 006662 005767 173736
006666 001445
006670 032777 000100 172142
006676 001041
2040 006700 004767 005634
2041 006704 000340
2042 006706 017703 173764
2043 006712 012777 006772 173756
2044 006720 042777 000200 173746
2045 006726 052777 000100 173740
2046 006734 105777 173734
2047 006740 100375
2048 006742 000005
2049 006744 004767 005570
2050 006750 000240
2051 006752 000240
2052 006754 000240
2053 006756 000240
2054 006760 000240
2055 006762 042777 000100 173704
2056 006770 000402
2057
2058 006772 022626
2059 006774 104051
2060
2061 006776 010377 173674
```

```
*****
:TEST 21      TEST THAT RTC INTERRUPT CLEARS WITH RESET
*****
TST21:  SCOPE
        TST      CTSTFL      :IS CONSOLE UNDER TEST?
        BEQ      TST22      :IF NOT, SKIP THIS TEST
        BIT      #BIT6,@SWR  :ARE LINE CLOCK TESTS INHIBITED?
        BNE      TST22      :IF YES, SKIP THIS TEST
        JSR      PC,WRPSW    :SET PRIORITY TO 7
                                .WORD 340
        MOV      @RTCVT,R3   :SAVE LINE CLOCK VECTOR
        MOV      #28,@RTCVT  :POINT RTC VECTOR TO ERROR REPORT
        BIC      #BIT7,@LKS  :CLEAR CLOCK DONE FLAG
        BIS      #BIT6,@LKS  :ENABLE CLOCK INTERRUPTS
18:     TSTB      @LKS        :WAIT FOR DONE (INTERRUPT REQUEST)
        BPL      18
        RESET
        JSR      PC,WRPSW    :CLEAR PENDING INTERRUPT WITH RESET
                                .WORD 240
        NOP
        NOP
        NOP
        NOP
        BIC      #BIT6,@LKS  :DISALLOW INTERRUPTS
        BR       38          :BR TO END OF TEST
28:     CMP      (SP)+,(SP)+  :RESTORE SP AFTER INTERRUPT
        ERROR+51 :RESET DID NOT CLEAR INTERRUPT
38:     MOV      R3,@RTCVT   :RESTORE LINE CLOCK VECTOR
```

2063
 2064
 2065

```

*****
:TEST 22      TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
*****
TST22:  SCOPE
2066 007002 000004          TST      CTSTFL      ;IS CONSOLE UNDER TEST?
      007004 005767 173614 BEQ      TST23      ;IF NOT, SKIP THIS TEST
      007010 001452          BIT      #BIT6,@SWR    ;ARE LINE CLOCK TESTS INHIBITED?
      007012 032777 000100 172020 BNE      TST23      ;IF YES, SKIP THIS TEST
      007020 001046          JSR      PC,WRPSW      ;SET PRIORITY TO 7
2067 007022 004767 005512          .WORD 340
2068 007026 000340          MOV      @RTCVT,R3      ;SAVE LINE CLOCK VECTOR
2069 007030 017703 173642          MOV      #25,@RTCVT ;POINT RTC VECTOR TO ERROR REPORT
2070 007034 012777 007120 173634 BIC      #BIT7,@LKS ;CLEAR CLOCK DONE FLAG
2071 007042 042777 000200 173624 BIS      #BIT6,@LKS ;ENABLE CLOCK INTERRUPTS
2072 007050 052777 000100 173616 1$: TSTB     @LKS      ;WAIT FOR DONE (INTERRUPT REQUEST)
2073 007056 105777 173612          BPL      1$
2074 007062 100375          BIC      #BIT7,@LKS      ;CLEAR DONE & INTERRUPT
2075 007064 042777 000200 173602 JSR      PC,WRPSW ;ALLOW INTERRUPTS
2076 007072 004767 005442          .WORD 240
2077 007076 000240          NOP
2078 007100 000240          NOP
2079 007102 000240          NOP
2080 007104 000240          NOP
2081 007106 000240          NOP
2082 007110 042777 000100 173556 BIC      #BIT6,@LKS ;DISALLOW INTERRUPTS
2083 007116 000402          BR       3$             ;BR TO END OF TEST
2084
2085
2086 007120 022626          2$:  CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2087 007122 104052          ERROR+52 ;CLEARING BIT7 OF LKS DID NOT CLEAR INTERRUPT
2088
2089 007124 010377 173546          3$:  MOV      R3,@RTCVT ;RESTORE LINE CLOCK VECTOR
2090 007130 004767 005404          JSR      PC,WRPSW ;RESTORE PRIORITY TO 7
2091 007134 000340          .WORD 340
  
```

2093
 2094
 2095

```

*****
: *TEST 23      TEST CLOCK REPEATABILITY
*****
TST23: SCOPE
0096 007136 000004          TST CTSTFL      ;IS CONSOLE UNDER TEST?
007140 005767 173460      BEQ TST24      ;IF NOT, SKIP THIS TEST
007144 001467          BIT #BIT6,@SWR    ;ARE LINE CLOCK TESTS INHIBITED?
007146 032777 000100 171664 BNE TST24      ;IF YES, SKIP THIS TEST
007154 001063          BIC #BIT6,@LKS    ;DISALLOW INTERRUPTS
2097 007156 042777 000100 173510
2098
2099 007164 005000          CLR R0        ;CLEAR A TIMER
2100 007166 012701 177777      MOV #1,R1  ;SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
2101 007172 005002          1$: CLR R2    ;CLEAR CLOCK COUNTER
2102 007174 005077 173474      CLR @LKS   ;CLEAR DONE
2103 007200 105777 173470      2$: TSTB @LKS ;SYNC ON DONE
2104 007204 100375          BPL 2$
2105 007206 005077 173462      CLR @LKS   ;CLEAR DONE
2106 007212 105777 173456      3$: TSTB @LKS ;IS CLOCK DONE?
2107 007216 100003          BPL 4$      ;BR IF NOT, TO INCREMENT TIMER
2108 007220 005202          INC R2      ;IF DONE, INCREMENT CLOCK COUNT
2109 007222 005077 173446      CLR @LKS   ;CLEAR DONE
2110 007226 005200          4$: INC R0    ;INCREMENT TIMER
2111 007230 001370          BNE 3$      ;BR IF TIME REMAINS
2112 007232 005201          INC R1      ;INCREMENT LOOP PASS FLAG
2113 007234 001003          BNE CMPARE   ;BR IF TWO PASSES HAVE BEEN MADE
2114 007236 010267 173074      MOV R2,FIRST ;IF NOT, STORE FIRST CLOCK COUNT
2115 007242 000753          BR 1$      ;DO LOOP AGAIN
2116 007244 016701 173066      CMPARE: MOV FIRST,R1 ;RECALL FIRST CLOCK COUNT
2117 007250 160201          SUB R2,R1    ;CALCULATE DIFFERENCE OF TWO COUNTS
2118 007252 100001          BPL TOLER   ;IF POSITIVE, SKIP NEGATION OF DIFFERENCE
2119 007254 005401          NEG R1      ;MAKE DIFFERENCE A POSITIVE NUMBER
2120 007256 032767 000001 173516 TOLER: BIT #BIT0,FLAG44 ;** IS THIS A 11/44
2121 007264 001403          BEQ 6$      ;** NO
2122 007266 020127 000002          CMP R1,#2 ;** YES; COMPARE DIFFERENCE WITH DESIRED
2123          ;** TOLERANCE OF 2
2124 007272 000402          BR 7$      ;**
2125 007274 020127 000001      6$: CMP R1,#1 ;COMPARE DIFFERENCE WITH DESIRED TOLERANCE
2126 007300 003403          7$: BLE 5$   ;BR, IF LOWER/EQUAL TO TOLERANCE
2127
2128 007302 010267 173310      MOV R2,SECND ;STORE SECOND COUNT
2129 007306 104053          ERROR+53 ;CLOCK REPEATABILITY ERROR
2130
2131 007310 032777 000020 171522 5$: BIT #BIT4,@SWR ;CLOCK TESTS ONLY?
2132 007316 001402          BEQ TST24    ;BR IF NOT
2133 007320 000167 005054          JMP $EOP ;ELSE, JUMP TO END OF PASS ROUTINE
2134

```

2136
 2137
 2138

```

007324 000004
2139 007326 042777 000100 173304
2140 007334 017703 173310
2141 007340 012777 007364 173302
2142 007346 105777 173266
2143 007352 100375
2144 007354 004767 005160
2145 007360 000140
2146 007362 000402
2147
2148 007364 022626
2149 007366 104054
2150
2151 007370 012777 007416 173252
2152 007376 052777 000100 173234
2153 007404 000240
2154 007406 000240
2155 007410 000240
2156 007412 000240
2157
2158 007414 104055
2159
2160 007416 042777 000100 173214
2161 007424 022626
2162 007426 010377 173216
2163

```

```

*****
;TEST 24      TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
*****
TST24:  SCOPE
        BIC     #BIT6,@TCSR      ;CLEAR TRANSMIT INTERRUPT ENABLE
        MOV     @TVECT,R3        ;SAVE XMIT VECTOR
        MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
1$:     TSTB     @TCSR           ;WAIT FOR DONE
        BPL     1$
        JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
        .WORD   140
        BR      3$

        CMP     (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
        ERROR+54

        MOV     #4$,@TVECT      ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
        BIS     #BIT6,@TCSR      ;SET XMIT VECTOR TO END OF TEST
        ;ENABLE INTERRUPTS
        ;**
        ;**
        ;**
        ;**
        ;**
        ERROR+55                ;XMIT DID NOT INTERRUPT

        BIC     #BIT6,@TCSR      ;DISABLE INTERRUPTS
        CMP     (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
        MOV     R3,@TVECT        ;RESTORE XMIT VECTOR

```

2165
2166

```

007432 000004
2167 007434 042777 000100 173176
2168 007442 004767 005072
2169 007446 000340
2170 007450 017703 173174
2171 007454 012777 007510 173166
2172 007462 105777 173152 1$:
2173 007466 100375
2174 007470 052777 000100 173142
2175 007476 000240
2176 007500 000240
2177 007502 000240
2178 007504 000240
2179 007506 000402
2180
2181 007510 022626 2$:
2182 007512 104056
2183
2184 007514 042777 000100 173116 3$:
2185 007522 012777 007550 173120
2186 007530 004767 005004
2187 007534 000140
2188 007536 000240
2189 007540 000240
2190 007542 000240
2191 007544 000240
2192 007546 000402
2193
2194 007550 022626 4$:
2195 007552 104057
2196
2197 007554 010377 173070 5$:
```

```

*****
:TEST 25 TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
*****
TST25: SCOPE
BIC #BIT6,@TCR ;DISABLE INTERRUPTS
JSR PC,WRPSW ;SET PSW TO PRIORITY 7
;WORD 340
MOV @TVECT,R3 ;SAVE XMIT VECTOR
MOV #2$,@TVECT ;POINT XMIT VECTOR TO ERROR REPORT
1$: TSTB @TCR ;WAIT FOR DONE
BPL 1$
BIS #BIT6,@TCR ;ENABLE INTERRUPT
NOP
NOP
NOP
NOP
BR 3$ ;CONTINUE TEST

2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR+56
;XMIT INTERRUPTS AT PRIORITY=7
3$: BIC #BIT6,@TCR ;CLEAR INTERRUPT ENABLE
MOV #4$,@TVECT ;POINT XMIT VECTOR TO ERROR REPORT
JSR PC,WRPSW ;SET PSW TO PRIORITY 3
;WORD 140
NOP
NOP
NOP
NOP
BR 5$ ;BR TO END OF TEST-NO INTERRUPT

4$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR+57
;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
5$: MOV R3,@TVECT ;RESTORE XMIT VECTOR
```

2199
2200
2201

2202 007560 000004
2203 007562 042777 000100 173050
2204 007570 017703 173054
2205 007574 017704 173052
2206 007600 012777 007650 173042
2207 007606 012777 000340 173036
2208 007614 004767 004720
2209 007620 000140
2210 007622 105777 173012
2211 007626 100375
2212 007630 052777 000100 173002
2213 007636 000240
2214 007640 000240
2215 007642 000240
2216 007644 000240
2217 007646 104060
2218
2219 007650 022626
2220 007652 012777 007706 172770
2221 007660 004767 004654
2222 007664 000140
2223 007666 000240
2224 007670 000240
2225 007672 000240
2226 007674 000240
2227 007676 042777 000100 172734
2228 007704 000402
2229
2230 007706 022626
2231 007710 104061
2232
2233 007712 010377 172732
2234 007716 010477 172730
2235
2236

```
*****
*TEST 26      TEST TRANSMITTER FOR DOUBLE INTERRUPTS
*****
TST26:  SCOPE
        BIC      #BIT6,@TCSR      ;CLEAR INTERRUPT ENABLE
        MOV      @TVECT,R3        ;SAVE XMIT VECTOR
        MOV      @TPSW,R4        ;SAVE XMIT PSW VECTOR
        MOV      #2$,@TVECT      ;SET UP XMIT VECTOR
        MOV      #340,@TPSW      ;SET PIO 7 AFTER INTERRUPT
        JSR      PC,WRPSW        ;SET PSW TO PRIORITY 3
        .WORD    140
1$:     TSTB     @TCSR            ;WAIT FOR DONE
        BPL      1$
        BIS      #BIT6,@TCSR      ;ENABLE INTERRUPTS
        NOP
        NOP
        NOP
        NOP
        ERROR+60
2$:     CMP      (SP)+,(SP)+      ;XMIT INTERRUPT DID NOT OCCUR
        MOV      #4$,@TVECT      ;RESTORE SP AFTER INTERRUPT
        JSR      PC,WRPSW        ;POINT XMIT VECTOR TO ERROR
        .WORD    140            ;SET PSW TO PRIORITY 3
        NOP
        NOP
        NOP
        NOP
        NOP
        BIC      #BIT6,@TCSR      ;GIVE TIME FOR ANY INTERRUPTS
        BR       5$             ;GIVE TIME FOR ANY INTERRUPTS
        ;GIVE TIME FOR ANY INTERRUPTS
        ;GIVE TIME FOR ANY INTERRUPTS
        ;GIVE TIME FOR ANY INTERRUPTS
        BIC      #BIT6,@TCSR      ;DISABLE INTERRUPTS
        BR       5$             ;BR TO END OF TEST
4$:     CMP      (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
        ERROR+61
5$:     MOV      R3,@TVECT        ;XMIT RE-INTERRUPTED
        MOV      R4,@TPSW        ;RESTORE XMIT VECTOR
        ;RESTORE XMIT PSW VECTOR
*****
*TEST 27      TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF
*****
TST27:  SCOPE
        BIC      #BIT6,@TCSR      ;DISABLE INTERRUPTS
        JSR      PC,WRPSW        ;SET PSW TO PRIORITY 7
        .WORD    340
        MOV      @TVECT,R3        ;SAVE XMIT VECTOR
        MOV      #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR
        BIS      #BIT2,@TCSR      ;** ENABLE MAINT. WRAP
        BIS      #BIT6,@TCSR      ;ENABLE INTERRUPTS
        CLR      @TBUF            ;LOAD TBUF
1$:     TSTB     @TCSR            ;WAIT FOR DONE (INTERRUPT)
        BPL      1$
        CLR      @TBUF            ;FILL SECOND BUFFER TO RESET INT.
        JSR      PC,WRPSW        ;ALLOW INTERRUPTS
        .WORD    140
```

2250	010012	000240		NOP		;GIVE TIME FOR ANY INTERRUPTS
2251	010014	000240		NOP		;GIVE TIME FOR ANY INTERRUPTS
2252	010016	000240		NOP		;GIVE TIME FOR ANY INTERRUPTS
2253	010020	000240		NOP		;GIVE TIME FOR ANY INTERRUPTS
2254	010022	042777	000100 172610	BIC	#BIT6,@TCSR	;DISABLE INTERRUPTS
2255	010030	000405		BR	3\$	;BR TO END OF TEST
2256						
2257	010032	022626	2\$:	CMP	(SP)+,(SP)+	;RESTORE SP AFTER INTERRUPT
2258	010034	042777	000004 172616	BIC	#BIT2,@TCSR	** DISABLE MAINTENANCE MODE FOR
						** CONSOLE TO ALLOW FOR COMMUNICATION
						** WITH TERMINAL.
2259	010042	104062			ERROR+62	
2260						;LOADING TBUF DID NOT CLEAR INTERRUPT.
2261	010044	010377	172600	3\$:	MOV	R3,@TVECT
2262						;RESTORE XMIT VECTOR
2263	010050	005000		CLR	R0	** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
	010052	012701	000002	MOV	#2,R1	** THAT MIGHT BE IN THE PROCESS OF BEING
	010056	005300		DEC	R0	** TRANSMITTED TO FINISH BEFORE MAINTENANCE
	010060	001376	40\$:	BNE	40\$	** WRAP FOR THE UART UNDER TEST IS DISABLED.
	010062	005301		DEC	R1	** THIS WILL INHIBIT ANY COMMUNICATION
	010064	001374		BNE	40\$	** TO HARDWARE MEDIA SUCH AS TU58 THAT MIGHT
						** BE ATTACHED TO UART UNDER TEST.

Line	Address	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
------	---------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 98
 T30 TEST THAT RCVR ACTIVE & DONE SET & CLEAR

SEQ 0064

```

2330
2331
2332 010402 000004
2333 010404 000005
2334 010406 052777 000004 172224
2334 010414 005000
2334 010416 012701 000002

010422 105777 172206
010426 100405
010430 005300
010432 001373
010434 005301
010436 001371
010440 000402
010442 005777 172170
2335 010446 005077 172170
2336 010452 105777 172156
2337 010456 100375
2338 010460 017700 172152
2339 010464 042777 000004 172166

2340 010472 105777 172136
2341 010476 001401
2342 010500 104070
2343
2344 010502 000240
2345
2346

010504 000004
2347 010506 032767 000001 172266
2348 010514 001050
2349 010516 000005
2350 010520 052777 000001 172106
2351 010526 052777 000004 172104
2352 010534 005000
2352 010536 012701 000002

010542 105777 172066
010546 100405
010550 005300
010552 001373
010554 005301
010556 001371
010560 000402
010562 005777 172050
2353 010566 005077 172050
2354 010572 105777 172036
2355 010576 100375
2356 010600 032777 000001 172026

*****
*TEST 31 TEST THAT READING RBUF CLEARS RECEIVER DONE
*****
TST31: SCOPE
RESET
BIS #BIT2,@TCSR ;CLEAR EVERYTHING
CLR R0 ;SET MAINTENANCE WRAP
MOV #2,R1 ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
;** THAT MIGHT BE IN THE PROCESS OF BEING
;** RECEIVED TO FINISH AFTER MAINTENANCE
;** WRAP FOR THE UART UNDER TEST IS ENABLED.
;** CHECK FOR RECEIVER DONE

42$: TSTB @RCR
BMI 43$
DEC R0
BNE 42$
DEC R1
BNE 42$
BR .+6

43$: TST @RBUF ;** READ TO CLEAR DONE
CLR @TBUF ;LOAD TRANSMITTER
1$: TSTB @RCR ;WAIT FOR RECEIVER DONE
BPL 1$
MOV @RBUF,R0 ;READ RECEIVE BUFFER
BIC #BIT2,@TCSR ;** DISABLE MAINTENANCE MODE FOR
;** CONSOLE TO ALLOW FOR COMMUNICATION
;** WITH TERMINAL.
;CHECK FOR RECEIVE DONE CLEAR
;** BR, IF CLEAR TO NEXT TEST

;READING RBUF DID NOT CLEAR RCVR DONE

10$: NOP

*****
*TEST 32 TEST THAT RDR ENABLE CLEARS RECEIVER DONE FLAG
*****
TST32: SCOPE
BIT #BIT0,FLAG44 ;** 11/44 ??
BNE 10$ ;** YES DO NOT EXECUTE THIS TEST
RESET
BIS #BIT0,@RCR ;CLEAR EVERYTHING
BIS #BIT2,@TCSR ;SET RDR ENABLE
CLR R0 ;SET MAINTENANCE WRAP
MOV #2,R1 ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
;** THAT MIGHT BE IN THE PROCESS OF BEING
;** RECEIVED TO FINISH AFTER MAINTENANCE
;** WRAP FOR THE UART UNDER TEST IS ENABLED.
;** CHECK FOR RECEIVER DONE

42$: TSTB @RCR
BMI 43$
DEC R0
BNE 42$
DEC R1
BNE 42$
BR .+6

43$: TST @RBUF ;** READ TO CLEAR DONE
CLR @TBUF ;LOAD TRANSMITTER
1$: TSTB @RCR ;WAIT FOR RECEIVER DONE
BPL 1$
BIT #BIT0,@RCR ;VERIFY RCV ACTIVE CLEARED RDR ENABLE

```

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 98-1
T32 TEST THAT RDR ENABLE CLEARS RECEIVER DONE FLAG

N 5

SEQ 0065

```
2357 010606 001401          BEQ      2$          ;BR IF CLEAR
2358
2359 010610 104117          ERROR+117        ;RDR ENABLE NOT CLEARED WITH RCV ACTIVE
2360
2361 010612 052777 000001 172014 2$:  BIS      #BIT0,@RCSR      ;CLEAR DONE BY SETTING RDR ENABLE
2362 010620 105777 172010          TSTB      @RCSR      ;CHECK FOR DONE CLEAR
2363 010624 001404          BEQ      10$          ;** BR, IF CLEAR TO NEXT TEST
2364 010626 042777 000004 172024          BIC      #BIT2,@CTCSR      ;** DISABLE MAINTENANCE MODE FOR
                                           ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                           ;** WITH TERMINAL.

2365 010634 104067          ERROR+67
2366
2367 010636 000240          10$:  NOP          ;SETTING RDR ENABLE DID NOT CLEAR RCVR DONE
2368
2369
```

2371
2372
2373

	010640	000004		TST33:	SCOPE				
2374	010642	042777	000100	171770	BIC	#BIT6,@TCSR	:	DISABLE TRANSMIT INTERRUPTS	
2375	010650	042777	000100	171756	BIC	#BIT6,@RCSR	:	DISABLE RECEIVER INTERRUPTS	
2376	010656	052777	000004	171754	BIS	#BIT2,@TCSR	:	SET MAINTENANCE WRAP	
2377	010664	005000			CLR	R0	:	** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS	
	010666	012701	000002		MOV	#2,R1	:	** THAT MIGHT BE IN THE PROCESS OF BEING	
							:	** RECEIVED TO FINISH AFTER MAINTENANCE	
							:	** WRAP FOR THE UART UNDER TEST IS ENABLED.	
							:	** CHECK FOR RECEIVER DONE	
	010672	105777	171736	42\$:	TSTB	@RCSR	:	**	
	010676	100405			BMI	43\$	:	**	
	010700	005300			DEC	R0	:	**	
	010702	001373			BNE	42\$	:	**	
	010704	005301			DEC	R1	:	**	
	010706	001371			BNE	42\$	:	**	
	010710	000402			BR	.+6	:	**	
	010712	005777	171720	43\$:	TST	@RBUF	:	** READ TO CLEAR DONE	
2378	010716	017703	171722		MOV	@RVECT,R3	:	SAVE RECEIVE VECTOR	
2379	010722	012777	010760	171714	MOV	#2\$,@RVECT	:	POINT RCV VECTOR TO ERROR REPORT	
2380	010730	004767	003604		JSR	PC,WPSW	:	SET PSW TO PRIORITY 3	
2381	010734	000140				.WORD 140	:		
2382	010736	005077	171700		CLR	@TBUF	:	SEND A CHARACTER	
2383	010742	105777	171666	1\$:	TSTB	@RCSR	:	WAIT FOR RECEIVER DONE	
2384	010746	100375			BPL	1\$	:		
2385	010750	042777	000004	171702	BIC	#BIT2,@TCSR	:	** DISABLE MAINTENANCE MODE FOR	
							:	** CONSOLE TO ALLOW FOR COMMUNICATION	
							:	** WITH TERMINAL.	
2386	010756	000405			BR	3\$	:	CONTINUE TEST	
2387							:		
2388	010760			2\$:			:		
2389	010760	042777	000004	171672	BIC	#BIT2,@TCSR	:	** DISABLE MAINTENANCE MODE FOR	
							:	** CONSOLE TO ALLOW FOR COMMUNICATION	
							:	** WITH TERMINAL.	
2390	010766	022626			CMP	(SP)+,(SP)+	:	RESTORE SP AFTER INTERRUPT	
2391	010770	104071			ERROR+71		:		
2392							:	RECEIVER INTERRUPTS WITH INT. ENABLE CLEAR	
2393							:		
2394	010772	012777	011026	171644	3\$:	MOV	#4\$,@RVECT	:	POINT RCV VECTOR TO END OF TEST
2395	011000	052777	000100	171626	BIS	#BIT6,@RCSR	:	ENABLE RCV INTERRUPTS	
2396	011006	000240			NOP		:	** GIVE ANY INTERRUPTS TIME	
2397	011010	000240			NOP		:	** GIVE ANY INTERRUPTS TIME	
2398	011012	000240			NOP		:	** GIVE ANY INTERRUPTS TIME	
2399	011014	000240			NOP		:	** GIVE ANY INTERRUPTS TIME	
2400	011016	042777	000004	171634	BIC	#BIT2,@TCSR	:	** DISABLE MAINTENANCE MODE FOR	
							:	** CONSOLE TO ALLOW FOR COMMUNICATION	
							:	** WITH TERMINAL.	
2401	011024	104072			ERROR+72		:		
2402							:	RCVR DID NOT INTERRUPT	
2403							:		
2404	011026	042777	000100	171600	4\$:	BIC	#BIT6,@RCSR	:	DISABLE INTERRUPTS
2405	011034	042777	000004	171616	BIC	#BIT2,@TCSR	:	** DISABLE MAINTENANCE MODE FOR	
							:	** CONSOLE TO ALLOW FOR COMMUNICATION	
							:	** WITH TERMINAL.	

.MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 99-1
T33 TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED

SEQ 0067

2406	011042	022626		CMP	(SP)+,(SP)+	;RESTORE SP AFTER INTERRUPT
2407	011044	010377	171574	MOV	R3,@RVECT	;RESTORE RECEIVE VECTOR

2409
 2410
 2411

 :TEST 34 TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED

2412	011050	000004			TST34:	SCOPE				
2413	011052	000005				RESET			:CLEAR EVERYTHING	
2414	011054	004767	003460			JSR	PC,WRPSW		:SET PSW TO PRIORITY 7	
2415	011060	000340					.WORD 340			
2416	011062	017703	171556			MOV	@RVECT,R3		:SAVE RECEIVE VECTOR	
2417	011066	012777	011174	171550		MOV	#28,@RVECT		:POINT RCVR VECTOR TO ERROR REPORT	
2418	011074	052777	000004	171536		BIS	#B12,@TCR		:SET MAINTENANCE WRAP	
2419	011102	005000				CLR	R0		:** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS	
2420	011104	012701	000002			MOV	#2,R1		:** THAT MIGHT BE IN THE PROCESS OF BEING	
									:** RECEIVED TO FINISH AFTER MAINTENANCE	
									:** WRAP FOR THE UART UNDER TEST IS ENABLED.	
									:** CHECK FOR RECEIVER DONE	
	011110	105777	171520		428:	TSTB	@RCR			
	011114	100405				BMI	438			
	011116	005300				DEC	R0			
	011120	001373				BNE	428			
	011122	005301				DEC	R1			
	011124	001371				BNE	428			
	011126	000402				BR	.+6			
2421	011130	005777	171502		438:	TST	@RBUF		:** READ TO CLEAR DONE	
2422	011134	105777	167710		68:	TSTB	@STPS		:TEST FOR XMIT READY	
2423	011140	100375				BPL	68		:LOOP IF NOT	
2424	011142	005077	171474			CLR	@RBUF		:SEND A CHARACTER	
2425	011146	105777	171462		18:	TSTB	@RCR		:WAIT FOR RECEIVER DONE	
2426	011152	100375				BPL	18			
2427	011154	052777	000100	171452		BIS	#B16,@RCR		:ENABLE INTERRUPTS	
2428	011162	000240				NOP			:** GIVE TIME FOR INTERRUPT	
2429	011164	000240				NOP			:** GIVE TIME FOR INTERRUPT	
2430	011166	000240				NOP			:** GIVE TIME FOR INTERRUPT	
2431	011170	000240				NOP			:** GIVE TIME FOR INTERRUPT	
2432	011172	000405				BR	38		:CONTINUE TEST	
2433	011174	042777	000004	171456	28:	BIC	#B12,@TCR		:** DISABLE MAINTENANCE MODE FOR	
2434									:** CONSOLE TO ALLOW FOR COMMUNICATION	
2435									:** WITH TERMINAL.	
2436	011202	022626				CMPL	(SP)+,(SP)+		:RESTORE SP AFTER INTERRUPT	
2437	011204	104073				ERROR+73			:RCVR INTERRUPTS AT PRIORITY 7	
2438										
2439	011206	042777	000100	171420	38:	BIC	#B16,@RCR		:CLEAR INTERRUPT ENABLE	
2440	011214	012777	011244	171422		MOV	#48,@RVECT		:POINT RCVR VECTOR TO ERROR REPORT	
2441	011222	004767	003312			JSR	PC,WRPSW		:SET PSW TO PRIORITY 3	
2442	011226	000140					.WORD 140			
2443	011230	000240				NOP			:GIVE TIME FOR ANY INTERRUPT	
2444	011232	000240				NOP			:GIVE TIME FOR ANY INTERRUPT	
2445	011234	042777	000004	171416		BIC	#B12,@TCR		:** DISABLE MAINTENANCE MODE FOR	
2446									:** CONSOLE TO ALLOW FOR COMMUNICATION	
2447									:** WITH TERMINAL.	
2448	011242	000405				BR	58		:BR TO END OF TEST, IF NO INTERRUPT	
2449										
2450	011244				48:	BIC	#B12,@TCR		:** DISABLE MAINTENANCE MODE FOR	
2451									:** CONSOLE TO ALLOW FOR COMMUNICATION	

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 100-1
134 TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED

SEQ 0069

2447 011252 022626 CMP (SP)+,(SP)+
2448 011254 106074 ERROR+74
2449
2450 011256 010377 171362 FS. MOV R3,DRVECT

::: WITH TERMINAL.
:RESTORE SP AFTER INTERRUPT
:RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
:RESTORE RECEIVE VECTOR

2452
2453
2454

```

*****
*TEST 35      TEST RECEIVER FOR DOUBLE INTERRUPTS
*****
TST35:  SCOPE
        RESET
        MOV    @RVECT,R3      ;CLEAR EVERYTHING
        MOV    @RPSW,R4      ;SAVE RECEIVE VECTOR
        MOV    #2$,@RVECT    ;SAVE RECEIVE PSW VECTOR
        MOV    #340,@RPSW    ;POINT RCV VECTOR TO CONTINUE TEST
        JSR    PC,WRPSW      ;SET PRIORITY TO 7 AFTER INTERRUPT
        .WORD  140           ;SET PSW TO PRIORITY 3
        BIS    #BIT2,@TCSR    ;SET MAINTENANCE WRAP
        CLR    R0            ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
        MOV    #2,R1         ;** THAT MIGHT BE IN THE PROCESS OF BEING
        ;** RECEIVED TO FINISH AFTER MAINTENANCE
        ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
        ;** CHECK FOR RECEIVER DONE
        42$:  TSTB    @RCSR
        BMI     43$
        DEC     R0
        BNE     42$
        DEC     R1
        BNE     42$
        BR      .+6
        43$:  TST     @RBUF    ;** READ TO CLEAR DONE
        CLR     @TBUF        ;SEND A CHARACTER
        1$:    TSTB    @RCSR    ;WAIT FOR RCVR DONE
        BPL     1$
        BIC     #BIT2,@TCSR    ;** DISABLE MAINTENANCE MODE FOR
        ;** CONSOLE TO ALLOW FOR COMMUNICATION
        ;** WITH TERMINAL.
        BIS     #BIT6,@RCSR    ;ENABLE RCV INTERRUPTS
        NOP
        NOP
        NOP
        NOP
        2464 011360 005077 171256 43$:  TST     @RBUF
        2465 011364 105777 171244 1$:    TSTB    @RCSR
        2466 011370 100375
        2467 011372 042777 000004 171260 BIC     #BIT2,@TCSR
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        2468 011400 052777 000100 171226 BIS     #BIT6,@RCSR
        2469 011406 000240
        2470 011410 000240
        2471 011412 000240
        2472 011414 000240
        2473
        2474 011416 104075
        2475
        2476
        2477 011420 022626
        2478 011422 012777 011466 171214 2$:  CMP     (SP)+,(SP)+
        2479 011430 004767 003104
        2480 011434 000140
        2481 011436 000240
        2482 011440 000240
        2483 011442 000240
        2484 011444 000240
        2485 011446 042777 000100 171160 BIC     #BIT6,@RCSR
        2486 011454 010377 171164
        2487 011460 010477 171162
        2488 011464 000402
        2489
        2490 011466 022626
        2491 011470 104076
        2492
        3$:    CMP     (SP)+,(SP)+
        ERROR+76
        ;RCVR INTERRUPT DID NOT OCCUR
        ;RESTORE SP AFTER INTERRUPT
        ;POINT RCV VECTOR TO ERROR REPORT
        ;RESET PSW TO PRIORITY 3
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        ;** GIVE SOME TIME
        ;CLEAR INTERRUPT ENABLE
        ;RESTORE RECEIVE VECTOR
        ;RESTORE RECEIVE PSW VECTOR
        ;BR TO END OF TEST
        ;RESTORE SP AFTER INTERRUPT
        ;RECEIVER RE-INTERRUPTED

```

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 101-1
T35 TEST RECEIVER FOR DOUBLE INTERRUPTS

G 6

SEQ 0071

2493 011472 010377 171146 4\$: MOV R3, @RVECT ;RESTORE RECEIVE VECTOR

2495
 2496
 2497

```

*****
*TEST 36          TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
*****
TST36:  SCOPE
        RESET
        JSR      PC,WRPSW          ;CLEAR EVERYTHING
                                   ;SET PSW PRIORITY TO 7
        .WORD    340
        MOV      @RVECT,R3        ;SAVE RECEIVE VECTOR
        MOV      #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
        BIS      #BIT6,@RCSR      ;SET RCVR INTERRUPT ENABLE
        BIS      #BIT2,@TCSR      ;SET MAINTENANCE WRAP
        CLR      R0               ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
        MOV      #2,R1            ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                   ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                   ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
                                   ;** CHECK FOR RECEIVER DONE
        42$:  TSTB      @RCSR
        BMI      43$
        DEC      R0
        BNE      42$
        DEC      R1
        BNE      42$
        BR       .+6
        43$:  TST      @RBUF
        CLR      @TBUF
        1$:    TSTB      @RCSR
        BPL      1$
        BIC      #BIT2,@TCSR      ;** DISABLE MAINTENANCE MODE FOR
                                   ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                   ;** WITH TERMINAL.
        CLR      @RBUF            ;READ RBUF TO CLEAR PENDING INTERRUPT
        JSR      PC,WRPSW        ;SET PSW TO PRIORITY 3
        .WORD    140
        NOP
        NOP
        NOP
        NOP
        BIC      #BIT6,@RCSR      ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
        BR       3$              ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
                                   ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
                                   ;** ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
                                   ;NO INTERRUPT-CLEAR INT. ENABLE
        2$:    CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
        ERROR+77 ;READING RBUF DID NOT CLEAR INTERRUPT
        3$:    MOV      R3,@RVECT ;RESTORE RECEIVE VECTOR
  
```

2498	011476	000004			
2499	011500	000005			
2500	011502	004767	003032		
2501	011506	000340			
2502	011510	017703	171130		
2503	011514	012777	011642	171122	
2504	011522	052777	000100	171104	
2505	011530	052777	000004	171102	
2506	011536	005000			
	011540	012701	000002		
	011544	105777	171064		
	011550	100405			
	011552	005300			
	011554	001373			
	011556	005301			
	011560	001371			
	011562	000402			
	011564	005777	171046		
2507	011570	005077	171046		
2508	011574	105777	171034		
2509	011600	100375			
	011602	042777	000004	171050	
2510	011610	005077	171022		
2511	011614	004767	002720		
2512	011620	000140			
2513	011622	000240			
2514	011624	000240			
2515	011626	000240			
2516	011630	000240			
2517	011632	042777	000100	170774	
2518	011640	000402			
2519					
2520	011642	022626			
2521	011644	104077			
2522					
2523	011646	010377	170772		
2524					

2526
 2527
 2528

 ;*TEST 37 TEST THAT RESET CLEARS RECEIVE INTERRUPT
 ;*****

2529	011652	000004			TST37:	SCOPE			
	011654	000005				RESET			;CLEAR EVERYTHING
2530	011656	004767	002656			JSR	PC,WRPSW		;SET PSW TO PRIORITY 7
2531	011662	000340					.WORD 340		
2532	011664	017703	170754			MOV	@RVECT,R3		;SAVE RECEIVE VECTOR
2533	011670	012777	012010	170746		MOV	#2\$,@RVECT		;POINT RCV VECTOR TO ERROR REPORT
2534	011676	052777	000100	170730		BIS	#BIT6,@RCSR		;SET RCV INTERRUPT ENABLE
2535	011704	052777	000004	170726		BIS	#BIT2,@TCSR		;SET MAINTENANCE WRAP
2536	011712	005000				CLR	R0		;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
	011714	012701	000002			MOV	#2,R1		;** THAT MIGHT BE IN THE PROCESS OF BEING
									;** RECEIVED TO FINISH AFTER MAINTENANCE
									;** WRAP FOR THE UART UNDER TEST IS ENABLED.
									;** CHECK FOR RECEIVER DONE
	011720	105777	170710		42\$:	TSTB	@RCSR		;**
	011724	100405				BMI	43\$		;**
	011726	005300				DEC	R0		;**
	011730	001373				BNE	42\$		;**
	011732	005301				DEC	R1		;**
	011734	001371				BNE	42\$		;**
	011736	000402				BR	.+6		;**
	011740	005777	170672		43\$:	TST	@RBUF		;** READ TO CLEAR DONE
2537	011744	012777	000377	170670		MOV	#377,@TBUF		;SEND AN ALL 1'S CHARACTER
2538	011752	105777	170656		1\$:	TSTB	@RCSR		;WAIT FOR RCV DONE
2539	011756	100375				BPL	1\$		
2540	011760	000005				RESET			;CLEAR RCV INTERRUPT & RBUF
2541	011762	004767	002552			JSR	PC,WRPSW		;SET PSW TO PRIORITY 3
2542	011766	000140					.WORD 140		
2543	011770	000240				NOP			;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
2544	011772	000240				NOP			;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
2545	011774	000240				NOP			;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
2546	011776	000240				NOP			;** ALLOW TIME FOR AN ERRONEOUS INTERRUPT
2547	012000	042777	000100	170626		BIC	#BIT6,@RCSR		;NO INTERRUPT-CLEAR INT. ENABLE
2548	012006	000402				BR	3\$		;CONTINUE TEST
2549									
2550									
2551	012010	022626			2\$:	CMPL	(SP)+,(SP)+		;RESTORE SP AFTER INTERRUPT
2552	012012	104100				ERROR+100			;RESET DID NOT CLEAR RCVR INTERRUPT
2553									
2554	012014	010377	170624		3\$:	MOV	R3,@RVECT		;RESTORE RECEIVE VECTOR

.MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 104
 T37 TEST THAT RESET CLEARS RECEIVE INTERRUPT

SFQ 0074

2556
 2557
 2558

 *TEST 40 TEST THAT THE 'OR' ERROR & 'ERROR' CAN BE SET

 TST40: SCOPE

2559	012020	000004			BIT	#BIT10,@SWR	:IS THIS TEST ENABLED
2560	012022	032777	002000	167010	BEQ	TST41	:IF NOT ENABLED, BR TO NEXT TEST
2561	012030	001471			BIT	#BIT0,FLAG44	:** IS THIS A 11/44
2562	012032	032767	000001	170742	BEQ	9\$	:** NO
2563	012040	001407			TST	CTSTFL	:** YES THIS IS 11/44. IS THIS THE CONSOLE
2564	012042	005767	170556				:** SLU
2565	012046	001404			BEQ	9\$	:** NO
2566	012050	032777	000010	166762	BIT	#BIT03,@SWR	:** THIS IS THE CONSOLE SLU.SHOULD THE OVERRUN
2567							:** ERROR TEST BE PERFORMED
2568	012056	001456			BEQ	TST41	:** NO
2569	012060	000005			RESET		:CLEAR EVERYTHING
2570	012062	052777	000004	170550	BIS	#BIT2,@TCSR	:SET MAINTENANCE WRAP
2571	012070	005000			CLR	R0	:** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
	012072	012701	000002		MOV	#2,R1	:** THAT MIGHT BE IN THE PROCESS OF BEING
							:** RECEIVED TO FINISH AFTER MAINTENANCE
							:** WRAP FOR THE UART UNDER TEST IS ENABLED.
							:** CHECK FOR RECEIVER DONE
	012076	105777	170532		42\$: TSTB	@RCSR	
	012102	100405			BMI	43\$	
	012104	005300			DEC	R0	
	012106	001373			BNE	42\$	
	012110	005301			DEC	R1	
	012112	001371			BNE	42\$	
	012114	000402			BR	+6	
	012116	005777	170514		43\$: TST	@RBUF	:** READ TO CLEAR DONE
2572	012122	012700	000003		MOV	#3,R0	:SET CHARACTER COUNT TO SEND 3 CHAR.
2573	012126	005077	170510		1\$: CLR	@TBUF	:LOAD TRANSMIT BUFFER
2574	012132	105777	170502		2\$: TSTB	@TCSR	:WAIT FOR TRANSMIT DONE
2575	012136	100375			BPL	2\$	
2576	012140	005300			DEC	R0	:DECREMENT CHARACTER COUNT
2577	012142	001371			BNE	1\$	:BR IF ALL CHARACTERS NOT TRANSMITTED
2578	012144	042777	000004	170506	BIC	#BIT2,@TCSR	:** DISABLE MAINTENANCE MODE FOR
							:** CONSOLE TO ALLOW FOR COMMUNICATION
							:** WITH TERMINAL.
2579	012152	032777	040000	170456	BIT	#BIT14,@RBUF	:TEST FOR 'OR' ERROR FLAG
2580	012160	001001			BNE	3\$	:BR, IF SET
2581	012162	104101			ERROR+101		: 'OR' ERROR FLAG DID NOT SET
2582							
2583							
2584	012164	032777	100000	170444	3\$: BIT	#BIT15,@RBUF	:TEST 'ERROR' FLAG
2585	012172	001001			BNE	4\$	:BR, IF SET
2586	012174	104102			ERROR+102		: 'ERROR' FLAG DID NOT SET WITH 'OR' FLAG
2587							
2588	012176				4\$: CLR	R0	:** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
2589	012176	005000			MOV	#2,R1	:** THAT MIGHT BE IN THE PROCESS OF BEING
	012200	012701	000002		DEC	R0	:** TRANSMITTED TO FINISH BEFORE MAINTENANCE
	012204	005300			BNE	40\$	:** WRAP FOR THE UART UNDER TEST IS DISABLED.
	012206	001376			DEC	R1	:** THIS WILL INHIBIT ANY COMMUNICATION
	012210	005301			BNE	40\$	:** TO HARDWARE MEDIA SUCH AS TUS8 THAT MIGHT
	012212	001374					:** BE ATTACHED TO UART UNDER TEST.

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 105
T40 TEST THAT THE 'DR' ERROR & 'ERROR' CAN BE SET

SEQ 0075

2591
2592
2593

```

*****
*TEST 41      TEST THAT BREAK TRANSMITS ALL ZEROES
*****
TST41: SCOPE
2594 012214 000004          BIT    #BIT8,@SWR      ;IS BREAK FUNCTION TEST ENABLED?
2595 012216 032777 000400 166614    BEQ    TST42      ;BR TO NEXT TEST, IF NOT ENABLED
2596 012224 001505          BIT    #BIT0,FLAG44    ;** IS THIS A 11/44
2597 012226 032767 000001 170546    BEQ    9$        ;** NO
2598 012234 001407          TST    CTSTFL          ;** YES THIS IS 11/44. IS THIS THE CONSOLE
2599 012236 005767 170362          ;** SLU
2600 012242 001404          BEQ    9$        ;** NO
2601 012244 032777 000010 166566    BIT    #BIT03,@SWR    ;** THIS IS THE CONSOLE SLU.SHOULD THE BREAK
2602                                ;** TEST BE PERFORMED
2603 012252 001472          BEQ    TST42      ;** NO
2604 012254 000005          9$: RESET          ;CLEAR EVERYTHING
2605 012256 052777 000004 170354    BIS    #BIT2,@TCSR    ;SET MAINTENANCE WRAP
2606 012264 005000          CLR    R0          ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
2607 012266 012701 000002          MOV    #2,R1      ;** THAT MIGHT BE IN THE PROCESS OF BEING
2608                                ;** RECEIVED TO FINISH AFTER MAINTENANCE
2609                                ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
2610                                ;** CHECK FOR RECEIVER DONE
2611 012272 105777 170336          42$: TSTB    @RCR      ;**
2612 012276 100405          BMI    43$        ;**
2613 012300 005300          DEC    R0          ;**
2614 012302 001373          BNE    42$        ;**
2615 012304 005301          DEC    R1          ;**
2616 012306 001371          BNE    42$        ;**
2617 012310 000402          BR    .+6          ;**
2618 012312 005777 170320          43$: TST    @RBUF      ;** READ TO CLEAR DONE
2619 012316 012777 177777 170316    MOV    #-1,@RBUF    ;TRANSMIT ALL ONES TO RCVR
2620 012324 105777 170304          1$: TSTB    @RCR      ;WAIT FOR RCVR DONE
2621 012330 100375          BPL    1$          ;**
2622 012332 005077 170300          CLR    @RBUF      ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
2623 012336 052777 000001 170274    BIS    #BIT0,@TCSR    ;TRANSMIT BREAK
2624 012344 005000          CLR    R0          ;CLEAR A TIMER
2625 012346 117767 170262 166452    2$: MOVB    @RCR,$BDDAT ;WAIT FOR RCVR DONE
2626 012354 100411          BMI    CONT41      ;BR IF DONE
2627 012356 005200          INC    R0          ;IF NOT, INCREMENT TIMER
2628 012360 001372          BNE    2$          ;BR IF TIME REMAINS
2629 012362 042777 000001 170250    BIC    #BIT0,@TCSR    ;CLEAR BREAK BIT
2630 012370 042777 000004 170262    BIC    #BIT2,@TCSR    ;** DISABLE MAINTENANCE MODE FOR
2631                                ;** CONSOLE TO ALLOW FOR COMMUNICATION
2632                                ;** WITH TERMINAL.
2633                                ;BREAK DID NOT TRANSMIT ANYTHING
2634 012376 104103          ERROR+103
2635 012400 105777 170232          CONT41: TSTB    @RBUF      ;CHECK RECEIVE BUFFER FOR ZERO
2636 012404 001407          BEQ    3$          ;BR, IF ZERO
2637 012406 042777 000001 170224    BIC    #BIT0,@TCSR    ;CLEAR BREAK BIT
2638 012414 042777 000004 170236    BIC    #BIT2,@TCSR    ;** DISABLE MAINTENANCE MODE FOR
2639                                ;** CONSOLE TO ALLOW FOR COMMUNICATION
2640                                ;** WITH TERMINAL.
2641 012422 104103          ERROR+103
2642 012424 042777 000001 170206    3$: BIC    #BIT0,@TCSR    ;CLEAR BREAK BIT

```

.MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 105-1
T41 TEST THAT BREAK TRANSMITS ALL ZEROES

SEQ 0076

2630 012432 042777 000004 170220

BIC

#BIT2,@CTCSR

;** DISABLE MAINTENANCE MODE FOR
;** CONSOLE TO ALLOW FOR COMMUNICATION
;** WITH TERMINAL.

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 107
 T42 TEST THAT 'FR' ERROR CAN BE SET DURING BREAK

SEQ 0078

2668
 2669
 2670

```

*****
: *TEST 43      TEST DATA PATH FROM XMIT TO REC USING MAINT WRAP
*****
TST43. SCOPE

012632 000004
2671
2672 012634 000005      RESET
2673 012636 052777 000004 167774      BIS      #BIT2,@TCSR      ;CLEAR EVERYTHING
2674                                     ;SET MAINTENANCE WRAP
2675                                     ;TRANSMIT A BINARY COUNT PATTERN - UP
2676                                     ;TO THE BIT POSITION INDICATED BY THE
2677 012644 005000      CLR      R0      ;CONTENTS OF LOCATION 'SUSWR'
012646 012701 000002      MOV      #2,R1      ;** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
                                           ;** THAT MIGHT BE IN THE PROCESS OF BEING
                                           ;** RECEIVED TO FINISH AFTER MAINTENANCE
                                           ;** WRAP FOR THE UART UNDER TEST IS ENABLED.
                                           ;** CHECK FOR RECEIVER DONE
012652 105777 167756      42$: TSTB      @RCSR
012656 100405      BMI      43$
012660 005300      DEC      R0
012662 001373      BNE      42$
012664 005301      DEC      R1
012666 001371      BNE      42$
012670 000402      BR      .+6
012672 005777 167740      43$: TST      @RBUF
2678 012676 005001      CLR      R1      ;** READ TO CLEAR DONE
2679 012700 105201      1$: INCB      R1      ;CLEAR REGISTER FOR TEST DATA
2680 012702 032767 000001 170072      BIT      #BIT0,FLAG44      ;INCREMENT THE TEST DATA
2681 012710 001406      BEQ      5$      ;** 11/44 CPU
2682 012712 122701 000020      CMPB     #20,R1      ;** TEST ALL DATA
2683 012716 001770      BEQ      1$      ;** CHECK FOR CONT-P IN TEST DATA
2684 012720 122701 000220      CMPB     #220,R1      ;** DO NOT XMIT ^P
2685 012724 001765      BEQ      1$
2686 012726 010177 167710      5$: MOV      R1,@RBUF      ;** DO NOT XMIT 220
2687 012732 105777 167676      2$: TSTB      @RCSR      ;XMIT A CHARACTER
2688 012736 100375      BPL      2$      ;WAIT FOR RECEIVER DONE
2689 012740 017702 167672      MOV      @RBUF,R2      ;GET RECEIVED CHARACTER
2690 012744 043701 001112      BIC      @SUSWR,R1      ;CLEAR LOWEST UNUSED DATA BIT POSITITON IN TEST DATA
2691 012750 020102      CMP      R1,R2      ;COMPARE DATA
2692 012752 001003      BNE      3$      ;BR, IF NON-COMPARE
2693 012754 105701      TSTB      R1      ;TEST XMIT DATA FOR ZERO
2694 012756 001411      BEQ      4$      ;BR, IF FINISHED
2695 012760 000747      BR      1$      ;CONTINUE IF NOT
2696 012762 010167 166036      3$: MOV      R1,$GDDAT      ;STORE THE EXPECTED DATA
2697 012766 010267 166034      MOV      R2,$BDDAT      ;STORE RECEIVED DATA
2698 012772 042777 000004 167660      BIC      #BIT2,@CTCSR      ;** DISABLE MAINTENANCE MODE FOR
                                           ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                           ;** WITH TERMINAL.
                                           ;DATA COMPARE DATA

2699 013000 104105      ERROR+105
2700
2701 013002
2702 013002 042777 000004 167650      4$: BIC      #BIT2,@CTCSR      ;** DISABLE MAINTENANCE MODE FOR
                                           ;** CONSOLE TO ALLOW FOR COMMUNICATION
                                           ;** WITH TERMINAL.

2703
2704
2705
2706
*****

```

					; *TEST 44	TEST DATA PATHS USING WRAP CABLE	
					*****	*****	
2707	013010	000004			TST44:	SCOPE	
2708	013012	032777	000200	166020		BIT #BIT7,@SWR	:IS THIS TEST ENABLED
2709	013020	001444				BEQ TST45	:BR, IF NOT
2710	013022	000005				RESET	:CLEAR EVERYTHING
2711	013024	005001				C_L R R1	:CLEAR REGISTER FOR TEST DATA
2712							:TRANSMIT A BINARY COUNT PATTERN - UP
2713							:TO THE BIT POSITION INDICATED BY THE
2714	013026	105201					:CONTENTS OF LOCATION '\$USWR'
2715	013030	032767	000001	167744	1\$:	INCB R1	:INCREMENT THE TEST DATA
2716	013036	001406				BIT #BIT0,FLAG44	:11/44 CPU
2717	013040	122701	000020			BEQ 5\$	:TEST ALL DATA
2718	013044	001770				CMPB #20,R1	:CHECK TEST DATA FOR ^P
2719	013046	122701	000220			BEQ 1\$	:DO NOT XMIT ^P
2720	013052	001765				CMPB #220,R1	
2721	013054	010177	167562			BEQ 1\$	
2722	013060	005000			5\$:	MOV R1,@TBUF	:XMIT A CHARACTER
2723	013062	105777	167546			CLR R0	:CLEAR A TIMER
2724	013066	100403			2\$:	TSTB @RC SR	:WAIT FOR RECEIVER DONE
2725	013070	005200				BMI 3\$	:BR IF DONE
2726	013072	001373				INC R0	:INCREMENT TIMER IF NOT
2727						BNE 2\$	:BR IF TIME REMAINS
2728	013074	104064				ERROR+64	:RECEIVER DONE NOT SET
2729							
2730	013076	017702	167534		3\$:	MOV @RBUF,R2	:GET RECEIVED CHARACTER
2731	013102	043701	001112			BIC @USWR,R1	:CLEAR LOWEST UNUSED DATA BIT POSITIION IN TEST DATA
2732	013106	020102				CMP R1,R2	:COMPARE DATA
2733	013110	001003				BNE 4\$	:BR, IF NON-COMPARE
2734	013112	105701				TSTB R1	:TEST XMIT DATA FOR ZERO
2735	013114	001406				BEQ TST45	:BR, IF FINISHED
2736	013116	000743				BR 1\$	:CONTINUE IF NOT
2737	013120	010167	165700		4\$:	MOV R1,\$GDDAT	:STORE EXPECTED DATA
2738	013124	010267	165676			MOV R2,\$BDDAT	:STORE RECEIVED DATA
2739							
2740	013130	104106				ERROR+106	:DATA COMPARE ERROR WITH WRAP CABLE

2742
 2743
 2744

 :TEST 45 TEST DL11-W LOGIC BY EXERCISING THE XMIT, REC, & CLOCK
 :*****

2745	013132	000004			TST45:	SCOPE		
2746	013134	000005				RESET		:CLEAR EVERYTHING
2747	013136	004767	001376			JSR	PC,WRPSW	:SET PRIORITY TO 7
2748	013142	000340				.WORD	340	
2749	013144	017703	167500			MOV	@TVECT,R3	:SAVE XMIT VECTOR
2750	013150	017704	167470			MOV	@RVECT,R4	:SAVE RECEIVE VECTOR
2751	013154	017705	167516			MOV	@RTCVT,R5	:SAVE CLOCK VECTOR
2752	013160	017767	167466	167260		MOV	@TPSW,STPSW	:SAVE XMIT PSW VECTOR
2753	013166	017767	167454	167254		MOV	@RPSW,SRPSW	:SAVE RECEIVE PSW VECTOR
2754	013174	017767	167500	167250		MOV	@RTCPW,SCPSW	:SAVE CLOCK PSW VECTOR
2755	013202	012777	013614	167440		MOV	#XMIT,@TVECT	:POINT TRANSMIT VECTOR TO TRANSMIT ROUTINE
2756	013210	012777	000200	167434		MOV	#200,@TPSW	:NO MULTIPLE INTERRUPTS ALLOWED
2757	013216	012777	013666	167420		MOV	#RCV,@RVECT	:POINT RECEIVE VECTOR TO RECEIVE ROUTINE
2758	013224	012777	000200	167414		MOV	#200,@RPSW	:NO MULT INTERRUPTS
2759	013232	005767	167366			TST	CTSTFL	:IS CONSOLE UNDER TEST?
2760	013236	001415				BEQ	1\$	:IF NOT SKIP CLOCK SET UP
2761	013240	032777	000100	165572		BIT	#BIT6,@SWR	:IF YES, ARE CLOCK TEST DISABLED?
2762	013246	001011				BNE	1\$	:IF YES, SKIP CLOCK SET UP
2763	013250	012777	013702	167420		MOV	#CLK,@RTCVT	:POINT VECTOR TO CLOCK INTERRUPT ROUTINE
2764	013256	012777	000300	167414		MOV	#300,@RTCPW	:NO MULT INTERRUPTS
2765	013264	052777	000100	167402		BIS	#BIT6,@LKS	:ENABLE CLOCK INTERRUPTS
2766	013272	052777	000004	167340	1\$:	BIS	#BIT2,@TCSR	:SET MAINTENANCE WRAP
	013300	005000				CLR	R0	:** DELAY ENOUGH TIME TO ALLOW ANY CHARACTERS
	013302	012701	000002			MOV	#2,R1	:** THAT MIGHT BE IN THE PROCESS OF BEING
								:** RECEIVED TO FINISH AFTER MAINTENANCE
								:** WRAP FOR THE UART UNDER TEST IS ENABLED.
								:** CHECK FOR RECEIVER DONE
	013306	105777	167322		42\$:	TSTB	@RCSR	
	013312	100405				BMI	43\$	
	013314	005300				DEC	R0	
	013316	001373				BNE	42\$	
	013320	005301				DEC	R1	
	013322	001371				BNE	42\$	
	013324	000402				BR	.+6	
	013326	005777	167304		43\$:	TST	@RBUF	:** READ TO CLEAR DONE
2767	013332	052777	000100	167300		BIS	#BIT6,@TCSR	:ENABLE TRANSMIT INTERRUPTS
2768	013340	052777	000100	167266		BIS	#BIT6,@RCSR	:ENABLE RECEIVE INTERRUPTS
2769	013346	005067	167070			CLR	XMTCNT	:CLEAR XMIT INTERRUPT COUNTER
2770	013352	005067	167062			CLR	RCVCNT	:CLEAR RCV INTERRUPT COUNTER
2771	013356	005001				CLR	R1	:CLEAR A REGISTER FOR TEST DATA USE
2772	013360	005000				CLR	R0	:CLEAR TIMER
2773	013362	012702	002454			MOV	#BUF,R2	:POINT R2 TO RECEIVE DATA STORAGE
2774	013366	005077	167250			CLR	@TBUF	:SEND FIRST CHARACTER
2775	013372	004767	001142			JSR	PC,WRPSW	:SET PSW TO PRIORITY 3
2776	013376	000140				.WORD	140	
2777								:WAIT FOR INTERRUPTS
2778	013400	000240			2\$:	NOP		:STALL
2779	013402	000240				NOP		:STALL
2780	013404	062700	000000			ADD	#0,R0	:ADD INSTRUCTIONS ARE USED TO LENGTHEN LOOP TIME
2781	013410	062700	000001			ADD	#1,R0	: TO COVER THE SLOWEST BAUD RATE ON THE FASTEST CPU
2782	013414	001371				BNE	2\$	
2783	013416	032777	000100	167214		BIT	#BIT6,@TCSR	:FINISHED ENTIRE TRANSMISSION
2784	013424	001402				BEQ	3\$	:BR, IF INTERRUPTS ARE DISABLED (FINISHED)

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 108-1
145 TEST DL11-W LOGIC BY EXERCISING THE XMIT, REC. & CLOCK

SEQ 0081

```
2785 013426 000005      RESET      ;CLEAR EVERYTHING
2786
2787 013430 104107      ERROR+107    ;TRANSMIT INTERRUPT TIMEOUT IN MAIN. DATA TEST
2788
2789 013432 026767 167004 167000 38.  CMP      XMITCNT,RCVCNT    ;COMPARE THE NUMBER OF INTERRUPTS
2790 013440 001402      BEQ      48    ;BR, IF EQUAL
2791 013442 000005      RESET      ;CLEAR EVERYTHING
2792
2793 013444 104110      ERROR+110    ;RECEIVER DID NOT GET FULL TRANSMISSION
2794      ; IF RCVCNT=0, NO DATA RECEIVED
2795      ; IF RCVCNT=?, THEN (XMITCNT-RCVCNT)
2796      ; EQUALS THE NO. OF INTERRUPTS LOST.
2797
```

```

2799
2800
2801 013446 005767 167152 4$: TST CTSTFL ;IS CONSOLE UNDER TEST?
2802 013452 001410 BEQ 5$ ;IF NOT, SKIP CLOCK COUNT CHECK
2803 013454 032777 000100 165356 BIT #BIT6,@SWR ;IF YES, ARE CLOCK TESTS DISABLED?
2804 013462 001004 BNE 5$ ;IF YES, SKIP CLOCK COUNT CHECK
2805 013464 005767 166754 TST CLKCNT ;CHECK FOR AT LEAST ONE CLOCK INTERRUPT
2806 013470 001001 BNE 5$ ;BR IF INTERRUPTS OCCURRED
2807
2808 013472 104113 ERROR+113 ;NO CLOCK INTERRUPTS IN EXERCISER
2809
2810 013474 000005 5$: RESET ;CLEAR EVERYTHING
2811 013476 012700 002454 MOV #BUF,R0 ;LOAD RECEIVED DATA POINTER TO R0
2812 013502 005001 CLR R1 ;SET UP REGISTER FOR COMPARISON
2813 013504 022001 COMP: CMP (R0)+,R1 ;COMPARE XMIT & RCV DATA
2814 013506 001014 BNE 6$ ;BR, IF NOT EQUAL
2815 013510 105201 9$: INCB R1 ;INCREMENT COMPARE DATA
2816 013512 032767 000001 167262 BIT #BIT0,FLAG44 ;11/44 CPU
2817 013520 001403 BEQ 8$ ;YES SKIP
2818 013522 122701 000020 CMPB #20,R1 ;SKIP 20 DATA IF 11/44
2819 013526 001770 BEQ 9$ ;SKIP 20
2820 013530 032701 000040 8$: BIT #BIT5,R1 ;FINISHED CHECKING RECEIVED DATA?
2821 013534 001763 BEQ COMP ;BR, IF NOT FINISHED
2822 013536 000405 BR 7$ ;BR TO END OF TEST
2823
2824 013540 014067 165262 6$: MOV -(R0),SBDDAT ;STORE BAD DATA FOR ERROR REPORT
2825 013544 010167 165254 MOV R1,$GDDAT ;STORE GOOD DATA FOR ERROR REPORT
2826 013550 104111 ERROR+111 ;DATA COMPARE ERROR IN EXERCISER
2827
2828 013552 010377 167072 7$: MOV R3,@TVECT ;RESTORE XMIT VECTOR
2829 013556 010477 167062 MOV R4,@RVECT ;RESTORE RECEIVE VECTOR
2830 013562 010577 167110 MOV R5,@RCVT ;RESTORE CLOCK VECTOR
2831 013566 016777 166654 167056 MOV STPSW,@TPSW ;RESTORE XMIT PSW VECTOR
2832 013574 016777 166650 167044 MOV SRPSW,@RPSW ;RESTORE RECEIVE PSW VECTOR
2833 013602 016777 166644 167070 MOV SCPSW,@RCPSW ;RESTORE CLOCK PSW VECTOR
2834 013610 000167 000530 JMP ENDEV ;GOTO NEXT TEST
2835
2836 013614 005267 166622 XMIT: INC XMITCNT ;INCREMENT XMIT INTERRUPT COUNTER
2837 013620 105201 1$: INCB R1 ;INCREMENT TEST DATA
2838 013622 032767 000001 167152 BIT #BIT0,FLAG44 ;11/44 CPU
2839 013630 001403 BEQ 2$ ;TEST ALL DATA
2840 013632 122701 000020 CMPB #20,R1 ;CHECK DATA FOR ^P
2841 013636 001770 BEQ 1$ ;DO NOT XMIT ^P
2842 013640 032701 000040 2$: BIT #BIT5,R1 ;SEND DATA PATTERN FROM 00 --> 37
2843 013644 001404 BEQ XCONT ;BR, IF MORE DATA TO BE SENT
2844 013646 042777 000100 166764 BIC #BIT6,@TCSR ;CLEAR XMIT INTERRUPT ENABLE
2845 013654 000402 BR XRET ;RETURN, WITHOUT SENDING ANY MORE DATA
2846 013656 110177 166760 XCONT: MOVB R1,@TBUF ;SEND NEW CHARACTER
2847 013662 005000 XRET: CLR R0 ;CLEAR TIMER
2848 013664 000002 RTI ;RETURN
2849
2850 013666 017722 166744 RCV: MOV @RBUF,(R2)+ ;STORE RECEIVED DATA
2851 013672 005267 166542 INC RCVCNT ;INCREMENT RCV INTERRUPT COUNTER
2852 013676 005000 CLR R0 ;CLEAR TIMER
2853 013700 000002 RTI ;RETURN
2854
2855 013702 005267 166536 CLK: INC CLKCNT ;INCREMENT CLOCK INTERRUPT COUNT

```

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 109-1
T45 TEST DL11-W LOGIC BY EXERCISING THE XMIT, REC, & CLOCK

F 7

SEQ 0083

2856 013706 000002
2857

RTI

;RETURN

```

2859 :*****
2860 : WRAPAROUND TESTING- AUTO INITIATION OF T/A CONSOLE TEST
2861 :
2862 :
2863 :
2864 :          176500          RCSR58 = 176500
2865 :          176502          RBUF58 = 176502
2866 :          176504          TCSR58 = 176504
2867 :          176506          TBUF58 = 176506
2868 :          003014          BGNADD = DEVADR
2869 :          024746          ENDADD = ENDADR
2870 :
2871 :*****
2872 : TEST 46      TEST CONSOLE WITH WRAP AROUND
2873 :*****
2874 : TST46: SCOPE
2875 :*****
2876 : MAIN LINE TEST
2877 :
2878 : THIS TEST PUTS COMMANDS OUT OVER THE SERIAL LINE WHICH IS WRAPPED
2879 : AROUND TO THE CONSOLE AND RECEIVES THE RESPONSES
2880 :
2881 : CONTROL P IS SEND TO GET THE CONSOLE'S ATTENTION AND THEN
2882 : T/A(A FOR APT) IS SENT. SINCE THE CPU IS HALTED DURING THE TESTING
2883 : THE PROGRAM CAN NOT SEE THE CHARS RETURNING, THUS THE PROGRAM WILL
2884 : SIT IN A LOOP WAITING FOR A 'B' TO BE PRINTED BY THE CONSOLE AS A
2885 : SIGNAL TO APT THAT THE TESTING IS DONE. ALSO SINCE THE TESTING
2886 : INVOLVES WRITTING TO THE CPU'S MEMORY A CHECKSUM IS CALCULATED AND THE
2887 : MEMORY TO BE USED INSIDE THIS PROGRAMS SPACE IS SAVE BEFORE TESTING
2888 : AND RESTORED AFTER TEST WITH ANOTHER CHECKSUM CALCULATION
2889 :*****
2890 :
2891 : 013712 000004 165120 WRAP:      BIT      #BIT2,@SWR      ;RUN THIS TEST ONLY IF
2892 : 013720 001451          BEQ      10$      ;SW BIT 2 IS ON
2893 :
2894 : 013722 016767 164050 166504      MOV      PSW,SAVEPS      ;SAVE OLD PSW
2895 : 013730 012767 000340 164040      MOV      #340,PSW      ;PUT IN NOW PSW
2896 : 013736 012706 002432          MOV      #JIMSTK,SP      ;USE MY STACK (SO NO CLOBER)
2897 :
2898 : 013742 005067 166372          CLR      LOC1
2899 : 013746 012767 000100 166366      MOV      #100,LOC2      ;TIMING LOOP COUNTERS
2900 :
2901 : 013754 004567 000274          JSR      R5,SAVEETE      ;SAVE LOCATIONS T/A WRITES
2902 :
2903 : 013760 004567 000076          JSR      R5,CHKSUM      ;CALCULATE CHECKSUM
2904 : 013764 010467 166446          MOV      R4,OLDSUM      ;SAVE OLD CHECKSUM
2905 :
2906 : 013770 012700 002564          MOV      #CNTLP,R0
2907 : 013774 004567 000102          JSR      R5,PUTLIN      ;SEND OUT CONTROL P
2908 :
2909 : 014000 012700 002566          MOV      #PROMPT,R0
2910 : 014004 004567 000124          JSR      R5,GETLIN      ;GET CRLF CONSOLE>>>
2911 :
2912 : 014010 012700 002610          MOV      #TA,R0

```

```
2913 014014 004567 000062      JSR      R5,PUTLIN      ;SEND OUT T/A<CRLF>
2914                                ;
2915 014020 004567 000154      JSR      R5,GETB       ;GET THE A FROM '-TESTB'
2916                                ;
2917 014024 004567 000260      JSR      R5,RESTTE    ;RESTORE LOCATIONS T/A WRITES
2918                                ;
2919 014030 004567 000026      JSR      R5,CHKSUM     ;RECALCULATE CHECKSUM
2920 014034 020467 166376      CMP      R4,OLDSUM
2921 014040 001401             BEQ      10$
2922 014042 000000             HALT
2923 014044 012706 001000      MOV      #1000,SP    ;RETURN THEIR SP
2924 014050 016767 166360      MOV      SAVEPS,PSW  ;RETURN PSW
2925 014056 000167 000316      JMP      $EOP      ;BR TO END OF PASS ROUTINE
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
```

```
*****
: ROUTINE TO CALCULATE CHECKSUM ON PROGRAM
: INPUT CONDITIONS
:   BGNADD  ADDRESS TO START CHECK SUM (INCLUSIVE)
:   ENDADD  ADDRESS TO END CHECK SUM (EXCLUSIVE)
: OUTPUT CONDITIONS
:   REG4 = CHECK SUM
*****
```

```
2944 014062 012700 003014      CHKSUM:  MOV      #BGNADD,R0    ;GET STARTING ADDRESS
2945 014066 005004             CLR      R4      ;RESET SUM
2946 014070 062004             ADD      (R0)+,R4  ;ADD WORD TO SUM
2947 014072 022700 024746      CMP      #ENDADD,R0  ;CHECK FOR END
2948 014076 001374             BNE      1$      ;IF NOT DONE LOOP
2949 014100 000205             RTS      R5      ;DONE RETURN
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
```

```
*****
: THIS ROUTINE OUTPUTS TO THE SERIAL LINE CHARS STARTING AT
: THE ADDRESS IN REG0 UNTIL IT HITS A <377>
*****
```

```
2960 014102 112001             PUTLIN:  MOVB     (R0)+,R1    ;GET DATA
2961 014104 001412             BEQ      10$      ;END OF DATA
2962 014106 004567 000122      1$:      JSR      R5,TIMER    ;PROVIDE FOR TIMOUT
2963 014112 032767 000200      BIT      #BIT7,TCSR58  ;TEST FOR XMIT READY
2964 014120 001772             BEQ      1$      ;WAIT FOR XMIT READY
2965 014122 110167 162360      MOVB     R1,TBUF58    ;OUTPUT CHAR
2966 014126 000167 177750      JMP      PUTLIN    ;REPETE
2967 014132 000205             10$:     RTS      R5      ;RETURN
2968
2969
```

```

2970
2971      : THIS ROUTINE INPUTS A CHARS FROM THE SERIAL LINE AND COMPARES
2972      : IT WITH THE EXPECTED VALUES POINTED TO BY REGO UNTIL NULL<00>
2973      :
2974      :*****
2975
2976
2977 014134 112001      GETLIN:      MOVB      (R0)+,R1      ;GET EXPECTED CHAR
2978 014136 105201      INCB      R1
2979 014140 001416      BEQ      10$      ;IF NULL EXIT
2980 014142 105301      DECB      R1
2981 014144 004567 000064 1$:      JSR      R5,TIMER      ;PROVIDE FOR TIMOUT
2982 014150 032767 000200 162322  BIT      #BIT7,RC5R58      ;TEST FOR REC READY
2983 014156 001772      BEQ      1$      ;WAIT FOR REC READY
2984 014160 116702 162316      MOVB      RBUF58,R2
2985 014164 042702 000200      BIC      #BIT7,R2      ;STRIP PARITY
2986 014170 120102      CMPB      R1,R2      ;ARE THEY THE SAME
2987 014172 001760      BEQ      GETLIN      ;SAME GET MORE
2988 014174 000000      HALT
2989 014176 000205 10$:      RTS      R5      ;NOT SAME HALT
2990      ;RETURN
2991
2992      :*****
2993
2994      : THIS ROUTINE INPUTS CHARS UNTIL IT GETS THE CHAR 'B'
2995      : AND THEN RETURNS
2996      :
2997      :*****
2998
2999
3000 014200 004567 000030      GETB:      JSR      R5,TIMER      ;OUT TIMING LOOP OUT
3001 014204 032767 000200 162266  BIT      #BIT7,RC5R58      ;TEST OFR REC READY
3002 014212 001772      BEQ      GETB      ;NOT DONE YET
3003 014214 116702 162262      MOVB      RBUF58,R2
3004 014220 042702 000200      BIC      #BIT7,R2      ;STRIP PARITY
3005 014224 122702 000102      CMPB      #102,R2      ;CHECK FOR 'B'
3006 014230 001363      BNE      GETB      ;NOT 'B' REPETE
3007 014232 000205      RTS      R5      ;WAS 'B' RETURN
3008
3009      :*****
3010
3011      : THIS ROUTINE IS USED AS A TIME OUT FEATURE
3012      : WHEN THE TIMING LOOPS BOTH REACH 0 THIS ROUTINE WI
3013      : CAUSE A HALT
3014      :
3015      :*****
3016
3017
3018 014234 005367 166100      TIMER:      DEC      LOC1
3019 014240 001004      BNE      10$      ;DECREMPNT TIMING LOOPS
3020 014242 005367 166074      DEC      LOC2
3021 014246 001001      BNE      10$
3022 014250 000000      HALT
3023 014252 000205 10$:      RTS      R5      ;IF ZERO THIS ROUTINE
3024      ;EXECUTED R3 TIMES
3025
3026      :*****

```

```

3027
3028      : THIS ROUTINE SAVES THE LOCATIONS WRITTEN BY THE T/A CONSOLE TEST
3029      : SO THEY MAY BE RESTORED LATER
3030      :
3031      :*****
3032
3033 014254 016767 163520 166062 SAVETE:      MOV      0,SAVE0      ;SAVE LOCATION 0
3034 014262 012702 000002      MOV      #2,R2      ;SET UP INDIRECT PNTER
3035 014266 012701 002346      MOV      #SAVLOC,R1      ;SET UP STORAGE LOC. PNTER
3036 014272 011221      1$:      MOV      (R2),(R1)+      ;GET WORD AND SAVE
3037 014274 000241      CLC      ;DONT ROT IN ANY BITS
3038 014276 006102      ROL      R2      ;SET NEXT ADDRESS
3039 014300 020227 024746      CMP      R2,#ENDADD      ;SEE IF AT END
3040 014304 100772      BMI      1$      ;NO REPETE
3041 014306 000205      RTS      R5
3042
3043
3044      :*****
3045      : THIS ROUTINE RESTORES THE SAVED LOCATIONS THAT T/A WRITES INTO
3046      :
3047      :*****
3048
3049
3050 014310 016767 166030 163462 RESTTE:      MOV      SAVE0,0
3051 014316 012702 000002      MOV      #2,R2      ;SET UP INDIRECT PNTER
3052 014322 012701 002346      MOV      #SAVLOC,R1      ;SET UP STORAGE LOC. PNTER
3053 014326 012112      1$:      MOV      (R1)+,(R2)      ;GET WORD AND RESTORE
3054 014330 000241      CLC      ;DONT ROT IN ANY BITS
3055 014332 006102      ROL      R2      ;SET NEXT ADDRESS
3056 014334 020227 024746      CMP      R2,#ENDADD      ;SEE IF AT END
3057 014340 100772      BMI      1$      ;NO REPETE
3058 014342 000205      RTS      R5
3059
3060

```

MAIN. MACRO M1111 20-SEP-79 10:37 PAGE 111
 46 TEST CONSOLE WITH WRAP AROUND

SEQ 0088

```

3062
3063
3064
3065 014344 005067 164432
3066 014350 005267 164522
3067 014354 026767 166250 164514
3068 014362 001002
3069 014364 000167 177322
3070 014370 005067 166230
3071 014374 000167 167450
3072
3073
3074
3075

;END OF DEVICE PASS ROUTINE
.EVEN
ENDEV: CLR $TSTNM ;CLEAR TEST NO. COUNT FOR SCOPE ROUTINE
INC $DEVCT ;INCREMENT DEVICE COUNTER
CMP TMP2,$DEVCT ;ALL DEVICES TESTED
BNE NOEOP ;BR, IF NO
JMP WRAP ;EXECUTE WRAP AROUND AFTER ALL DEVICES
NOEOP: CLR CTSTFL ;CLEAR CONSOLE UNDER TEST FLAG
JMP TSTDEV ;GO TEST NEXT DEVICE

```

.SBTTL END OF PASS ROUTINE

```

;*****
;*INCREMENT THE PASS NUMBER ($PASS)
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO GOAGIN

```

```

014400
014400 000004
014402 005067 164374
014406 005267 164462
014412 042767 100000 164454
014420 005327
014422 000001
014424 003015
014426 012737
014430 000001
014432 014422
014434 104401 014470
014440 013700 000042
014444 001405
014446 000005
014450 004710
014452 000240
014454 000240
014456 000240
014460
014460 000137
014462 014504
014464 377 377 000

$EOP:
SCOPE
CLR $TSTNM ;;ZERO THE TEST NUMBER
INC $PASS ;;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;;YES
MOV (PC)+,@(PC)+ ;;RESTORE COUNTER
$ENDCT: .WORD 1
TYPE ,ENDMG ;;TYPE 'END PASS'
$GET42: MOV @#42,R0 ;;GET MONITOR ADDRESS
BEQ $DOAGN ;;BRANCH IF NO MONITOR
RESET ;;CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11
$DOAGN:
JMP @(PC)+ ;;RETURN
$RTNAD: .WORD GOAGIN
$ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
.EVEN
ENDMG: .ASCIIZ <CR><LF>/END PASS /

```

```

3076 014470 015 012 105
014473 116 104 040
014476 120 101 123
014501 123 040 000

```

```
3078
3079 014504 005067 164366      GOAGIN: CLR      $DEVCT      ;CLEAR DEVICE COUNT
3080 014510 022767 000001 166112 CMP      #1,TMP2      ;IS THERE ONLY ONE DEVICE UNDER TEST?
3081 014516 001004          BNE      RSTRT      ;BR, IF NOT
3082 014520 012706 001000          MOV      #1000,SP      ;RESET STACK POINTER
3083 014524 000167 167442          JMP      TST1      ;GO DO ANOTHER PASS
3084
3085 014530 005067 164344      RSTRT: CLR      $UNIT      ;CLEAR UNIT NUMBER
3086 014534 000167 167234          JMP      BEGIN
3087
3088 014540 011646          WRPSW: MOV(SP),-(SP)      ;COPY RETURN PC
3089 014542 013616          MOV      @ (SP)+,(SP)      ;MOVE NEW PSW TO STACK
3090 014544 062746 000002          ADD      #2,-(SP)      ;ADJUST JSR RETURN
3091 014550 000002          RTI      ;POP RETURN PC & NEW PSW
3092
3093      ;SUBROUTINE TO REPORT UNEXPECTED OR ERRONEOUS TRAPS OR INTERRUPTS
3094
3095 014552 012600          CATCH: MOV      (SP)+,R0      ;GET ADDRESS OF TRAP VECTOR + 4
3096 014554 162700 000004          SUB      #4,R0      ;ADJUST TO POINT TO TRAP ADDRESS
3097 014560 010067 166036          MOV      R0,BDVECT      ;STORE TRAP OR INTERRUPT ADDRESS
3098 014564 016667 000002 166026          MOV      2(SP),OLDPC      ;GET PC WHERE TRAP OR INTERRUPT OCCURRED
3099 014572 104112          ERROR+112      ;REPORT ERROR
3100
3101 014574 000000          HALT      ;PROGRAM MUST BE RESTARTED AT THIS POINT
3102
```

```

3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117 014576
3118 014576 105267 164201
3119 014602 001775
3120 014604 016777 164172 164230
3121 014612 005267 164174
3122 014616 011667 164174
3123 014622 162767 000002 164166
3124 014630 117767 164162 164156
3125 014636 032777 020000 164174
3126 014644 001004
3127 014646 004767 000106
3128 014652 104401 001063
3129 014656
3130 014656 122767 000001 164222
3131 014664 001007
3132 014666 116767 164122 000004
3133 014674 004767 000562
3134 014700 000
3135 014701 000
3136 014702 000777
3137 014704 005777 164130
3138 014710 100001
3139 014712 000000
3140 014714 104406
3141 014716 032777 001000 164114
3142 014724 001402
3143 014726 016716 164056
3144 014732 005767 164122
3145 014736 001402
3146 014740 016716 164114
3147 014744
3148 014744 022737 014450 000042
3149 014752 001001
3150 014754 000000
3151 014756
3152 014756 000002
3153

```

```

*****
*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
*SAVE THE ERROR ITEM NUMBER AND ADDRESS OF THE ERROR CALL
*AND GO TO $ERRTYP ON ERROR
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW15=1      HALT ON ERROR
*SW13=1      INHIBIT ERROR TYPEOUTS
*SW09=1      LOOP IN ERROR
*CALL
*          ERROR+N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
*****

$ERROR:
7$:      INCB      $ERFLG      ;SET THE ERROR FLAG
        BEQ       7$          ;DON'T LET FLAG GO TO ZERO
        MOV       $TSTNM,$DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
        INC       $ERTTL      ;INCREMENT ERROR COUNT
        MOV       (SP),$ERRPC  ;GET ADDRESS OF ERROR INSTRUCTION
        SUB       #2,$ERRPC
        MOVB      @ $ERRPC,$ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
        BIT       #BIT13,@SWR   ;SKIP TYPEOUT IF SET
        BNE       20$          ;SKIP TYPEOUTS
        JSR       PC,$ERRTYP   ;GO TO USER ERROR ROUTINE
        TYPE      , $CRLF

20$:
        CMPB      #APTENV,$ENV  ;RUNNING IN APT MODE
        BNE       2$           ;NO, SKIP APT ERROR REPORT
        MOVB      $ITEMB,21$    ;SET ITEM NUMBER AS ERROR NUMBER
        JSR       PC,$ATY4      ;REPORT FATAL ERROR TO APT

21$:
        .BYTE     0
        .BYTE     0

22$:
        BR        22$          ;APT ERROR LOOP

2$:
        TST       @SWR         ;HALT ON ERROR
        BPL       3$           ;SKIP IF CONTINUE
        HALT      ;HALT ON ERROR!

3$:
        CKSWR
        BIT       #BIT09,@SWR  ;TEST FOR CHANGE IN SOFT-SWR
        BEQ       4$           ;LOOP ON ERROR SWITCH SET?
        MOV       $LPERR,(SP)  ;BR IF NO
        TST       $ESCAPE      ;FUDGE RETURN FOR LOOPING
        BEQ       5$           ;CHECK FOR AN ESCAPE ADDRESS
        MOV       $ESCAPE,(SP) ;BR IF NONE
        ;FUDGE RETURN ADDRESS FOR ESCAPE

5$:
        CMP       # $ENDAD,@#42 ;ACT-11 AUTO-ACCEPT?
        BNE       6$           ;BR IF NO
        HALT      ;YES

6$:
        RTI                  ;RETURN

```

3155
3156

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

*THIS ROUTINE USES THE 'ITEM CONTROL TYPE' (\$ITEMB) TO DETERMINE WHICH
*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE 'ERROR TABLE' (\$ERRTB),
*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

014760			\$ERRTYP:			
014760	104401	001063		TYPE	, \$CRLF	:: 'CARRIAGE RETURN' & 'LINE FEED'
014764	010046			MOV	R0, -(SP)	:: SAVE R0
014766	005000			CLR	R0	:: PICKUP THE ITEM INDEX
014770	153700	001014		BISB	@\$ITEMB, R0	
014774	001004			BNE	1\$	:: IF ITEM NUMBER IS ZERO, JUST
						:: TYPE THE PC OF THE ERROR
014776	016746	164014		MOV	\$ERRPC, -(SP)	:: SAVE \$ERRPC FOR TYPEOUT
						:: ERROR ADDRESS
015002	104402			TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
015004	000426			BR	6\$	:: GET OUT
015006	005300		1\$:	DEC	R0	:: ADJUST THE INDEX SO THAT IT WILL
015010	006300			ASL	R0	:: WORK FOR THE ERROR TABLE
015012	006300			ASL	R0	
015014	006300			ASL	R0	
015016	062700	001146		ADD	#\$ERRTB, R0	:: FORM TABLE POINTER
015022	012067	000004		MOV	(R0)+, 2\$	:: PICKUP 'ERROR MESSAGE' POINTER
015026	001404			BEQ	3\$	:: SKIP TYPEOUT IF NO POINTER
015030	104401			TYPE		:: TYPE THE 'ERROR MESSAGE'
015032	000000		2\$:	.WORD	0	:: 'ERROR MESSAGE' POINTER GOES HERE
015034	104401	001063		TYPE	, \$CRLF	:: 'CARRIAGE RETURN' & 'LINE FEED'
015040	012067	000004	3\$:	MOV	(R0)+, 4\$	:: PICKUP 'DATA HEADER' POINTER
015044	001404			BEQ	5\$	:: SKIP TYPEOUT IF 0
015046	104401			TYPE		:: TYPE THE 'DATA HEADER'
015050	000000		4\$:	.WORD	0	:: 'DATA HEADER' POINTER GOES HERE
015052	104401	001063		TYPE	, \$CRLF	:: 'CARRIAGE RETURN' & 'LINE FEED'
015056	011000		5\$:	MOV	(R0), R0	:: PICKUP 'DATA TABLE' POINTER
015060	001004			BNE	7\$	:: GO TYPE THE DATA
015062	012600		6\$:	MOV	(SP)+, R0	:: RESTORE R0
015064	104401	001063		TYPE	, \$CRLF	:: 'CARRIAGE RETURN' & 'LINE FEED'
015070	000207			RTS	PC	:: RETURN
015072			7\$:			
015072	013046			MOV	@(R0)+, -(SP)	:: SAVE @(R0)+ FOR TYPEOUT
015074	104402			TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
015076	005710			TST	(R0)	:: IS THERE ANOTHER NUMBER?
015100	001770			BEQ	6\$	:: BR IF NO
015102	104401	015110		TYPE	, 8\$	:: TYPE TWO(2) SPACES
015106	000771			BR	7\$	:: LOOP
015110	040	040	000 8\$:	.ASCIIZ	/ /	:: TWO(2) SPACES
				.EVEN		

3157

```

3159
3160 .SBTTL POWER DOWN AND UP ROUTINES
3161 ::*****
3162 ;*POWER DOWN ROUTINE
3163 ::*****
3164 015114 012737 015254 000024 $PWRDN: MOV $SILLUP,@PWRVEC ;SET FOR FAST UP
3165 015122 012737 000340 000026 MOV #340,@PWRVEC+2 ;PRIO:7
3166 015130 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
3167 015132 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
3168 015134 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
3169 015136 010346 MOV R3,-(SP) ;PUSH R3 ON STACK
3170 015140 010446 MOV R4,-(SP) ;PUSH R4 ON STACK
3171 015142 010546 MOV R5,-(SP) ;PUSH R5 ON STACK
3172 015144 017746 163670 MOV @SWR,-(SP) ;PUSH @SWR ON STACK
3173 015150 010667 000104 MOV SP,$SAVR6 ;SAVE SP
3174 015154 012737 015166 000024 MOV $PWRUP,@PWRVEC ;SET UP VECTOR
3175 015162 000000 HALT
3176 015164 000776 BR .-2 ;HANG UP
3177
3178
3179 ::*****
3180 ;*POWER UP ROUTINE
3181 ::*****
3182 015166 012737 015254 000024 $PWRUP: MOV $SILLUP,@PWRVEC ;SET FOR FAST DOWN
3183 015174 016706 000060 MOV $SAVR6,SP ;GET SP
3184 015200 012677 163634 MOV (SP)+,@SWR ;POP STACK INTO @SWR
3185 015204 012605 MOV (SP)+,R5
3186 015206 012604 MOV (SP)+,R4 ;POP STACK INTO R4
3187 015210 012603 MOV (SP)+,R3 ;POP STACK INTO R3
3188 015212 012602 MOV (SP)+,R2 ;POP STACK INTO R2
3189 015214 012601 MOV (SP)+,R1 ;POP STACK INTO R1
3190 015216 012600 MOV (SP)+,R0 ;POP STACK INTO R0
3191 015220 012737 015114 000024 MOV $PWRDN,@PWRVEC ;SET UP THE POWER DOWN VECTOR
3192 015226 012737 000340 000026 MOV #340,@PWRVEC+2 ;PRIO:7
3193 015234 005067 000020 CLR $SAVR6 ;WAIT LOOP FOR THE TTY
3194 015240 005267 000014 1$: INC $SAVR6 ;WAIT FOR THE INC
3195 015244 001375 BNE 1$ ;OF WORD
3196 015246 104401 TYPE ;REPORT THE POWER FAILURE
3197 015250 015262 $PWRMG: .WORD $POWER ;POWER FAIL MESSAGE POINTER
3198 015252 000002 RTI
3199 015254 000000 $SILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
3200 015256 000776 BR .-2 ;BEFORE THE POWER DOWN WAS COMPLETE
3201 015260 000000 $SAVR6: 0 ;PUT THE SP HERE
3202 015262 015 012 120 $POWER: .ASCIIZ <15><12>'POWER'
3203 015265 117 127 105
015270 122 000

```

3205
3206

.SBTTL SCOPE HANDLER ROUTINE

```
*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPIAY<7:0>)
;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW14=1      LOOP ON TEST
;SW09=1      LOOP ON ERROR
;CALL
;SCOPE      ;;SCOPE-IOT
```

```
015272 $SCOPE:
015272 104406 CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
015274 032777 1$:      BIT    #BIT14,$SWR      ;;LOOP ON PRESENT TEST?
015302 001052      BNE    $OVER      ;;YES IF SW14=1
;#####START OF CODE FOR THE XOR TESTER#####
015304 000416 $XTSTR: BR    6$      ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
;THIS INSTRUCTION TO A 'NOP' (NOP=240)
015306 013746 000004      MOV    @ERRVEC,-(SP)      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
015312 012737 015332 000004      MOV    #5,$@ERRVEC      ;;SET FOR TIMEOUT
015320 005737 177060      TST    @177060      ;;TIME OUT ON XOR?
015324 012637 000004      MOV    (SP)+,@ERRVEC      ;;RESTORE THE ERROR VECTOR
015330 000421      BR    $SVLAD      ;;GO TO THE NEXT TEST
015332 022626 5$:      CMP    (SP)+,(SP)+      ;;CLEAR THE STACK AFTER A TIME OUT
015334 012637 000004      MOV    (SP)+,@ERRVEC      ;;RESTORE THE ERROR VECTOR
015340 000407      BR    7$      ;;LOOP ON THE PRESENT TEST
015342 6$:;#####END OF CODE FOR THE XOR TESTER#####
015342 105767 163435 2$:      TSTB    $ERFLG      ;;HAS AN ERROR OCCURRED?
015346 001412      BEQ    $SVLAD      ;;BR IF NO
015350 032777 001000 163462      BIT    #BIT09,$SWR      ;;LOOP ON ERROR?
015356 001404      BEQ    4$      ;;BR IF NO
015360 016767 163424 163420 7$:      MOV    $LPERR,$LPADR      ;;SET LOOP ADDRESS TO LAST SCOPE
015366 000420      BR    $OVER
015370 105067 163407 4$:      CLRB    $ERFLG      ;;ZERO THE ERROR FLAG
015374 105267 163402 $SVLAD: INCB    $TSTNM      ;;COUNT TEST NUMBERS
015400 116767 163376 163464      MOVB    $TSTNM,$TESTN      ;;SET TEST NUMBER IN APT MAILBOX
015406 011667 163374      MOV    (SP),$LPADR      ;;SAVE SCOPE LOOP ADDRESS
015412 011667 163372      MOV    (SP),$LPERR      ;;SAVE ERROR LOOP ADDRESS
015416 005067 163436      CLR    $ESCAPE      ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
015422 112767 000001 163365      MOVB    #1,$ERMAX      ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
015430 016777 163346 163404 $OVER: MOV    $TSTNM,@DISPLAY      ;;DISPLAY TEST NUMBER
015436 016716 163344      MOV    $LPADR,(SP)      ;;FUDGE RETURN ADDRESS
015442 000002      RTI      ;;FIXES PS
```

3207

```

3209
3210
3211
3212
3213
3214 015444 112767 000001 000236 SATY1: MOVB #1,$FFLG ;TO REPORT FATAL ERROR
3215 015452 112767 000001 000226 SATY3: MOVB #1,$MFLG ;TO TYPE A MESSAGE
3216 015460 000403 BR SATYC
3217 015462 112767 000001 000220 SATY4: MOVB #1,$FFLG ;TO ONLY REPORT FATAL ERROR
3218 015470 SATYC:
3219 015470 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
3220 015472 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
3221 015474 105767 000206 TSTB $MFLG ;SHOULD TYPE A MESSAGE?
3222 015500 001450 BEQ 5$ ;IF NOT: BR
3223 015502 122767 000001 163376 CMPB #APTENV,$ENV ;OPERATING UNDER APT?
3224 015510 001031 BNE 3$ ;IF NOT: BR
3225 015512 132767 000100 163367 BITB #APTSPOOL,$ENV ;SHOULD SPOOL MESSAGE?
3226 015520 001425 BEQ 3$ ;IF NOT: BR
3227 015522 017600 000004 MOV @4(SP),R0 ;GET MESSAGE ADDRESS
3228 015526 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS
3229 015534 005767 163326 1$: TST $MSGTYPE ;SEE IF DONE W/ LAST XMISSION?
3230 015540 001375 BNE 1$ ;IF NOT: WAIT
3231 015542 010067 163334 MOV R0,$MSGAD ;PUT ADDRESS IN MAILBOX
3232 015546 105720 2$: TSTB (R0)+ ;FIND END OF MESSAGE
3233 015550 001376 BNE 2$
3234 015552 166700 163324 SUB $MSGAD,R0 ;SUB START OF MESSAGE
3235 015556 006200 ASR R0 ;GET MESSAGE LENGTH IN WORDS
3236 015560 010067 163320 MOV R0,$MSGGLT ;PUT LENGTH IN MAILBOX
3237 015564 012767 000004 163274 MOV #4,$MSGTYPE ;TELL APT TO TAKE MESSAGE
3238 015572 000413 BR 5$
3239 015574 017667 000004 000016 3$: MOV @4(SP),4$ ;PUT MSG ADDR IN JSR LINKAGE
3240 015602 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS
3241 015610 016746 162162 MOV 177776,-(SP) ;PUSH 177776 ON STACK
3242 015614 004767 000072 JSR PC,$TYPE ;CALL TYPE MACRO
3243 015620 000000 4$: .WORD 0
3244 015622 5$:
3245 015622 105767 000062 10$: TSTB $FFLG ;SHOULD REPORT FATAL ERROR?
3246 015626 001413 BEQ 12$ ;IF NOT: BR
3247 015630 005767 163252 TST $ENV ;RUNNING UNDER APT?
3248 015634 001410 BEQ 12$ ;IF NOT: BR
3249 015636 005767 163224 11$: TST $MSGTYPE ;FINISHED LAST MESSAGE?
3250 015642 001375 BNE 11$ ;IF NOT: WAIT
3251 015644 017667 000004 163216 MOV @4(SP),$FATAL ;GET ERROR #
3252 015652 005267 163210 INC $MSGTYPE ;TELL APT TO TAKE ERROR
3253 015656 062766 000002 000004 12$: ADD #2,4(SP) ;BUMP RETURN ADDRESS
3254 015664 105067 000020 CLRB $FFLG ;CLEAR FATAL FLAG
3255 015670 105067 000013 CLRB $LFLG ;CLEAR LOG FLAG
3256 015674 105067 000006 CLRB $MFLG ;CLEAR MESSAGE FLAG
3257 015700 012601 MOV (SP)+,R1 ;POP STACK INTO R1
3258 015702 012600 MOV (SP)+,R0 ;POP STACK INTO R1
3259 015704 000207 RTS PC ;RETURN
3260 015706 000 $MFLG: .BYTE 0
3261 015707 000 $LFLG: .BYTE 0 ;LOG FLAG
3262 015710 000 $FFLG: .BYTE 0 ;FATAL FLAG
3263
3264
3265 000200 .EVEN
APTSIZE-200

```

3266	000001	APTENV=001
3267	000100	APTSPool=100
3268	000040	APTC SUP=040
3269		

3271
3272

.SBTTL TYPE ROUTINE

:ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
:THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
:NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
:NOTE2: \$FILLC CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
:NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.
:

:CALL:
:1) USING A TRAP INSTRUCTION
: TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
:OR
: TYPE
: MESADR
:

015712	105767	163141	\$TYPE:	TSTB	\$TFPLG	::IS THERE A TERMINAL?
015716	100002			BPL	1\$	::BR IF YES
015720	000000			HALT		::HALT HERE IF NO TERMINAL
015722	000430			BR	3\$	::LEAVE
015724	010046		1\$:	MOV	R0,-(SP)	::SAVE R0
015726	017600	000002		MOV	@2(SP),R0	::GET ADDRESS OF ASCIZ STRING
015732	122767	000001	163146	CMPB	#APTENV,\$ENV	::RUNNING IN APT MODE
015740	001011			BNE	62\$	::NO,GO CHECK FOR APT CONSOLE
015742	132767	000100	163137	BITB	#APTPOOL,\$ENV	::SPOOL MESSAGE TO APT
015750	001405			BEQ	62\$	::NO,GO CHECK FOR CONSOLE
015752	010067	000004		MOV	R0,61\$	::SETUP MESSAGE ADDRESS FOR APT
015756	004767	177470		JSR	PC,\$ATY3	::SPOOL MESSAGE TO APT
015762	000000		61\$:	.WORD	0	::MESSAGE ADDRESS
015764	132767	000040	163115	62\$:	BITB	#APTCSP,\$ENV
015772	001003			BNE	60\$	::APT CONSOLE SUPPRESSED
015774	112046		2\$:	MOVB	(R0)+,-(SP)	::YES,SKIP TYPE OUT
015776	001005			BNE	4\$	::PUSH CHARACTER TO BE TYPED ONTO STACK
016000	005726			TST	(SP)+	::BR IF IT ISN'T THE TERMINATOR
016002	012600		60\$:	MOV	(SP)+,R0	::IF TERMINATOR POP IT OFF THE STACK
016004	062716	000002	3\$:	ADD	#2,(SP)	::RESTORE R0
016010	000002			RTI		::ADJUST RETURN PC
016012	122716	000011	4\$:	CMPB	#HT,(SP)	::RETURN
016016	001430			BEQ	8\$	::BRANCH IF <HT>
016020	122716	000200		CMPB	#CRLF,(SP)	::BRANCH IF NOT <CRLF>
016024	001006			BNE	5\$	
016026	005726			TST	(SP)+	::POP <CR><LF> EQUIV
016030	104401			TYPE		::TYPE A CR AND LF
016032	001063			\$CRLF		
016034	105067	000200		CLRB	\$CHARCNT	::CLEAR CHARACTER COUNT
016040	000755			BR	2\$	::GET NEXT CHARACTER
016042	004767	000056	5\$:	JSR	PC,\$TYPEC	::GO TYPE THIS CHARACTER
016046	126726	163004	6\$:	CMPB	\$FILLC,(SP)+	::IS IT TIME FOR FILLER CHARS.?
016052	001350			BNE	2\$	::IF NO GO GET NEXT CHAR.
016054	016746	162774		MOV	\$NULL,-(SP)	::GET # OF FILLER CHARS. NEEDED
						::AND THE NULL CHAR.
016060	105366	000001	7\$:	DECB	1(SP)	::DOES A NULL NEED TO BE TYPED?
016064	002770			BLT	6\$	::BR IF NO--GO POP THE NULL OFF OF STACK
016066	004767	000032		JSR	PC,\$TYPEC	::GO TYPE A NULL
016072	105367	000142		DECB	\$CHARCNT	::DO NOT COUNT AS A COUNT

```

016076 000770          BR      7$          ;;LOOP
                                ;HORIZONTAL TAB PROCESSOR
016100 112716 000040      8$:   MOVB    #' , (SP)          ;;REPLACE TAB WITH SPACE
016104 004767 000014      9$:   JSR     PC, $TYPEC          ;;TYPE A SPACE
016110 132767 000007 000122  BITB    #7, $CHARCNT          ;;BRANCH IF NOT AT
016116 001372          BNE     9$          ;;TAB STOP
016120 005726          TST     (SP)+          ;;POP SPACE OFF STACK
016122 000724          BR      2$          ;;GET NEXT CHARACTER
016124 105777 162720      $TYPEC: TSTB    @ $TPS          ;;WAIT UNTIL PRINTER IS READY
016130 100375          BPL     $TYPEC
016132 116677 000002 162712  MOVB    2(SP), @ $TPB          ;;LOAD CHAR TO BE TYPED INTO DATA REG.
016140 105777 162700          TSTB    @ $TKS          ;;SEE IF KEYBOARD IS TALKING.
016144 100021          BPL     2$          ;;BRANCH IF IT ISN'T.
016146 017746 162674          MOV     @ $TKB, -(SP)          ;;PUSH CHARACTER ONTO STACK.
016152 042716 177600          BIC     #177600, (SP)          ;;BIT CLEAR TOP BYTE AND PARITY BIT.
016156 022726 000023          CMP     #23, (SP)+          ;;SEE IF THIS IS A ^S.
016162 001012          BNE     2$          ;;BRANCH TO CONTINUE IF IT ISN'T.
016164 105777 162654      3$:   TSTB    @ $TKS          ;;WAIT FOR ANOTHER INPUT.
016170 100375          BPL     3$          ;;BRANCH BACK IF NOT READY.
016172 017746 162650          MOV     @ $TKB, -(SP)          ;;PUSH NEXT CHARACTER ON STACK.
016176 042716 177600          BIC     #177600, (SP)          ;;BIT CLEAR TOP BYTE AND PARITY BIT.
016202 022726 000021          CMP     #21, (SP)+          ;;SEE IF THIS IS A ^Q.
016206 001366          BNE     3$          ;;BRANCH BACK FOR MORE WAIT IF NOT.
016210 122766 000015 000002  2$:   CMPB    #CR, 2(SP)          ;;IS CHARACTER A CARRIAGE RETURN?
016216 001003          BNE     1$          ;;BRANCH IF NO
016220 105067 000014          CLRB    $CHARCNT          ;;YES--CLEAR CHARACTER COUNT
016224 000406          BR      $TYPEX          ;;EXIT
016226 122766 000012 000002  1$:   CMPB    #LF, 2(SP)          ;;IS CHARACTER A LINE FEED?
016234 001402          BEQ     $TYPEX          ;;BRANCH IF YES
016236 105227          INCB     (PC)+          ;;COUNT THE CHARACTER
016240 000000      $CHARCNT: .WORD    0          ;;CHARACTER COUNT STORAGE
016242 000207      $TYPEX:  RTS     PC

```

3273

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

;*****
;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
;OCTAL (ASCII) NUMBER AND TYPE IT.
;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
;CALL:
;      MOV     NUM, -(SP)          ;;NUMBER TO BE TYPED
;      TYPOS          ;;CALL FOR TYPEOUT
;      .BYTE    N          ;;N-1 TO 6 FOR NUMBER OF DIGITS TO TYPE
;      .BYTE    M          ;;M-1 OR 0
;      ;;1-TYPE LEADING ZEROS
;      ;;0-SUPPRESS LEADING ZEROS
;*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
;*$TYPOS OR $TYPOC
;CALL:
;      MOV     NUM, -(SP)          ;;NUMBER TO BE TYPED
;      TYPON          ;;CALL FOR TYPEOUT
;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
;CALL:

```

```

      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
      ;*      TYPOC      ;;CALL FOR TYPEOUT

016244 017646 000000      $TYPOS: MOV      @ (SP),-(SP)      ;;PICKUP THE MODE
016250 116667 000001 000211      MOV      1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
016256 112667 000207      MOV      (SP)+, $OMODE+1      ;;NUMBER OF DIGITS TO TYPE
016262 062716 000002      ADD      #2, (SP)      ;;ADJUST RETURN ADDRESS
016266 000406      BR      $TYPON
016270 112767 000001 000171      $TYPOC: MOV      #1, $OFILL      ;;SET THE ZERO FILL SWITCH
016276 112767 000006 000165      MOV      #6, $OMODE+1      ;;SET FOR SIX(6) DIGITS
016304 112767 000005 000154      $TYPON: MOV      #5, $OCNT      ;;SET THE ITERATION COUNT
016312 010346      MOV      R3, -(SP)      ;;SAVE R3
016314 010446      MOV      R4, -(SP)      ;;SAVE R4
016316 010546      MOV      R5, -(SP)      ;;SAVE R5
016320 116704 000145      MOV      $OMODE+1, R4      ;;GET THE NUMBER OF DIGITS TO TYPE
016324 005404      NEG      R4
016326 062704 000006      ADD      #6, R4      ;;SUBTRACT IT FOR MAX. ALLOWED
016332 110467 000132      MOV      R4, $OMODE      ;;SAVE IT FOR USE
016336 116704 000125      MOV      $OFILL, R4      ;;GET THE ZERO FILL SWITCH
016342 016605 000012      MOV      12(SP), R5      ;;PICKUP THE INPUT NUMBER
016346 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
016350 006105      1$: ROL      R5      ;;ROTATE MSB INTO 'C'
016352 000404      BR      3$      ;;GO DO MSB
016354 006105      2$: ROL      R5      ;;FORM THIS DIGIT
016356 006105      ROL      R5
016360 006105      ROL      R5
016362 010503      MOV      R5, R3
016364 006103      3$: ROL      R3      ;;GET LSB OF THIS DIGIT
016366 105367 000076      DECB      $OMODE      ;;TYPE THIS DIGIT?
016372 100016      BPL      7$      ;;BR IF NO
016374 042703 177770      BIC      #177770, R3      ;;GET RID OF JUNK
016400 001002      BNE      4$      ;;TEST FOR 0
016402 005704      TST      R4      ;;SUPPRESS THIS 0?
016404 001403      BEQ      5$      ;;BR IF YES
016406 005204      4$: INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
016410 052703 000060      BIS      #'0, R3      ;;MAKE THIS DIGIT ASCII
016414 052703 000040      5$: BIS      #' , R3      ;;MAKE ASCII IF NOT ALREADY
016420 110367 000040      MOV      R3, 8$      ;;SAVE FOR TYPING
016424 104401 016464      TYPE      , 8$      ;;GO TYPE THIS DIGIT
016430 105367 000032      7$: DECB      $OCNT      ;;COUNT BY 1
016434 003347      BGT      2$      ;;BR IF MORE TO DO
016436 002402      BLT      6$      ;;BR IF DONE
016440 005204      INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
016442 000744      BR      2$      ;;GO DO THE LAST DIGIT
016444 012605      6$: MOV      (SP)+, R5      ;;RESTORE R5
016446 012604      MOV      (SP)+, R4      ;;RESTORE R4
016450 012603      MOV      (SP)+, R3      ;;RESTORE R3
016452 016666 000002 000004      MOV      2(SP), 4(SP)      ;;SET THE STACK FOR RETURNING
016460 012616      MOV      (SP)+, (SP)
016462 000002      RTI      ;;RETURN
016464 000      8$: .BYTE      0      ;;STORAGE FOR ASCII DIGIT
016465 000      .BYTE      0      ;;TERMINATOR FOR TYPE ROUTINE
016466 000      $OCNT: .BYTE      0      ;;OCTAL DIGIT COUNTER
016467 000      $OFILL: .BYTE      0      ;;ZERO FILL SWITCH
016470 000000      $OMODE: .WORD      0      ;;NUMBER OF DIGITS TO TYPE

```

3276
3277

.SBTTL TTY INPUT ROUTINE

.ENABL LSB

*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
*WHEN OPERATING IN TTY FLAG MODE.

016472	022767	000176	162340	\$CKSWR:	CMP	#SWREG,SWR	::IS THE SOFT-SWR SELECTED?
016500	001074				BNE	15\$	::BRANCH IF NO
016502	105777	162336			TSTB	@\$TKS	::CHAR THERE?
016506	100071				BPL	15\$	::IF NO, DON'T WAIT AROUND
016510	117746	162332			MOVB	@\$TKB,-(SP)	::SAVE THE CHAR
016514	042716	177600			BIC	#^C177,(SP)	::STRIP-OFF THE ASCII
016520	022726	000007			CMP	#7,(SP)+	::IS IT A CONTROL G?
016524	001062				BNE	15\$	::NO, RETURN TO USER
016526	126727	162302	000001		CMPB	\$AUTOB,#1	::ARE WE RUNNING IN AUTO-MODE?
016534	001456				BEQ	15\$	::BRANCH IF YES
016536	104401	017217			TYPE	,\$CNTLG	::ECHO THE CONTROL-G (^G)
016542	104401	017224		\$GTSWR:	TYPE	,\$MSWR	::TYPE CURRENT CONTENTS
016546	016746	161424			MOV	SWREG,-(SP)	::SAVE SWREG FOR TYPEOUT
016552	104402				TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
016554	104401	017235			TYPE	,\$MNEW	::PROMPT FOR NEW SWR
016560	005046			9\$:	CLR	-(SP)	::CLEAR COUNTER
016562	005046				CLR	-(SP)	::THE NEW SWR
016564	105777	162254		7\$:	TSTB	@\$TKS	::CHAR THERE?
016570	100375				BPL	7\$	::IF NOT TRY AGAIN
016572	117746	162250			MOVB	@\$TKB,-(SP)	::PICK UP CHAR
016576	042716	177600			BIC	#^C177,(SP)	::MAKE IT 7-BIT ASCII
016602	021627	000025		9\$:	CMP	(SP),#25	::IS IT A CONTROL-U?
016606	001005				BNE	10\$	::BRANCH IF NOT
016610	104401	017212			TYPE	,\$CNTLU	::YES, ECHO CONTROL-U (^U)
016614	062706	000006		20\$:	ADD	#6,SP	::IGNORE PREVIOUS INPUT
016620	000757				BR	19\$	::LET'S TRY IT AGAIN
016622	021627	000015		10\$:	CMP	(SP),#15	::IS IT A <CR>?
016626	001022				BNE	16\$	::BRANCH IF NO
016630	005766	000004			ST	4(SP)	::YES, IS IT THE FIRST CHAR?
016634	001403				BEQ	11\$	::BRANCH IF YES
016636	016677	000002	162174		MOV	2(SP),@SWR	::SAVE NEW SWR
016644	062706	000006		11\$:	ADD	#6,SP	::CLEAR UP STACK
016650	104401	001063		14\$:	TYPE	,\$CRLF	::ECHO <CR> AND <LF>
016654	126727	162155	000001		CMPB	\$INTAG,#1	::RE-ENABLE TTY KBD INTERRUPTS?
016662	001003				BNE	15\$	::BRANCH IF NOT
016664	012777	000100	162152		MOV	#100,@\$TKS	::RE-ENABLE TTY KBD INTERRUPTS
016672	000002			15\$:	RTI		::RETURN
016674	004767	177224		16\$:	JSR	PC,\$TYPEC	::ECHO CHAR
016700	021627	000060			CMP	(SP),#60	::CHAR < 0?

```

016704 002420          BLT      18$          ;;BRANCH IF YES
016706 021627 000067    CMP      (SP),#67     ;;CHAR > 7?
016712 003015          BGT      18$          ;;BRANCH IF YES
016714 042726 000060    BIC      #60,(SP)+    ;;STRIP-OFF ASCII
016720 005766 000002    TST      2(SP)        ;;IS THIS THE FIRST CHAR
016724 001403          BEQ      17$          ;;BRANCH IF YES
016726 006316          ASL      (SP)          ;;NO, SHIFT PRESENT
016730 006316          ASL      (SP)          ;;CHAR OVER TO MAKE
016732 006316          ASL      (SP)          ;;ROOM FOR NEW ONE.
016734 005266 000002    17$: INC      2(SP)        ;;KEEP COUNT OF CHAR
016740 056616 177776    BIS      -2(SP),(SP)  ;;SET IN NEW CHAR
016744 000707          BR       7$           ;;GET THE NEXT ONE
016746 104401 001062    18$: TYPE  $QUES     ;;TYPE ?<CR><LF>
016752 000720          BR       20$          ;;SIMULATE CONTROL-U
          .DSABL  LSB

```

```

*****
*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
*CALL:
*      RDCHR          ;;INPUT A SINGLE CHARACTER FROM THE TTY
*      RETURN HERE    ;;CHARACTER IS ON THE STACK
*                    ;;WITH PARITY BIT STRIPPED OFF
*

```

```

016754 011646          $RDCHR: MOV      (SP),-(SP)  ;;PUSH DOWN THE PC
016756 016666 000004 000002    MOV      4(SP),2(SP)  ;;SAVE THE PS
016764 105777 162054    1$:  TSTB      @TKS        ;;WAIT FOR
016770 100375          BPL      1$           ;;A CHARACTER
016772 117766 162050 000004    MOVB     @TKB,4(SP)    ;;READ THE TTY
017000 042766 177600 000004    BIC      #^C<17>,4(SP) ;;GET RID OF JUNK IF ANY
017006 026627 000004 000023    CMP      4(SP),#23    ;;IS IT A CONTROL-S?
017014 001013          BNE      3$           ;;BRANCH IF NO
017016 105777 162022    2$:  TSTB      @TKS        ;;WAIT FOR A CHARACTER
017022 100375          BPL      2$           ;;LOOP UNTIL ITS THERE
017024 117746 162016    MOVB     @TKB,-(SP)          ;;GET CHARACTER
017030 042716 177600          BIC      #^C<17>,(SP)  ;;MAKE IT 7-BIT ASCII
017034 022627 000021    CMP      (SP)+,#21          ;;IS IT A CONTROL-Q?
017040 001366          BNE      2$           ;;IF NOT DISCARD IT
017042 000750          BR       1$           ;;YES, RESUME
017044 026627 000004 000140    3$:  CMP      4(SP),#140  ;;IS IT UPPER CASE?
017052 002407          BLT      4$           ;;BRANCH IF YES
017054 026627 000004 000175    CMP      4(SP),#175    ;;IS IT A SPECIAL CHAR?
017062 003003          BGT      4$           ;;BRANCH IF YES
017064 042766 000040 000004    BIC      #40,4(SP)     ;;MAKE IT UPPER CASE
017072 000002          4$:  RTI              ;;GO BACK TO USER

```

```

*****
*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
*CALL:
*      RDLIN          ;;INPUT A STRING FROM THE TTY
*      RETURN HERE    ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
*                    ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
*

```

```

017074 010346          $RDLIN: MOV      R3, -(SP)    ;;SAVE R3
017076 012703 017202    1$:  MOV      #$TTYIN,R3    ;;GET ADDRESS
017102 022703 017212    2$:  CMP      #$TTYIN+8.,R3  ;;BUFFER FULL?
017106 101405          BLOS     4$           ;;BR IF YES

```

017110	104407			RDCHR		::GO READ ONE CHARACTER FROM THE TTY
017112	112613			MOVB	(SP)+,(R3)	::GET CHARACTER
017114	122713	000177		CMPB	#177,(R3)	::IS IT A RUBOUT
017120	001003			BNE	3\$	::SKIP IF NOT
017122	104401	001062		TYPE	,\$QUES	::TYPE A '?'
017126	000763			BR	1\$	::CLEAR THE BUFFER AND LOOP
017130	111367	000044		MOVB	(R3),9\$	::DO THE CHARACTER
017134	104401	017200		TYPE	,\$	
017140	122723	000015		CMPB	#15,(R3)+	::CHECK FOR RETURN
017144	001356			BNE	2\$	::LOOP IF NOT RETURN
017146	105063	177777		CLRB	-1(R3)	::CLEAR RETURN (THE 15)
017152	104401	001064		TYPE	,\$LF	::TYPE A LINE FEED
017156	012603			MOV	(SP)+,R3	::RESTORE R3
017160	011646			MOV	(SP),-(SP)	::ADJUST THE STACK AND PUT ADDRESS OF THE
017162	016666	000004	000002	MOV	4(SP),2(SP)	::FIRST ASCII CHARACTER ON IT
017170	012766	017202	000004	MOV	#\$TTYIN,4(SP)	
017176	000002			RTI		::RETURN
017200	000			9\$: .BYTE	0	::STORAGE FOR ASCII CHAR. TO TYPE
017201	000			.BYTE	0	::TERMINATOR
017202				\$TTYIN: .BLKB	8.	::RESERVE 8 BYTES FOR TTY INPUT
017212	136	125	015	\$CNTLU: .ASCIZ	/'^U/<15><12>	::CONTROL 'U'
017215	012	000				
017217	136	107	015	\$CNTLG: .ASCIZ	/'^G/<15><12>	::CONTROL 'G'
017222	012	000				
017224	015	012	123	\$MSWR: .ASCIZ	<15><12>/SWR - /	
017227	127	122	040			
017232	075	040	000			
017235	040	040	116	\$MNEW: .ASCIZ	/ NEW - /	
017240	105	127	040			
017243	075	040	000			

3278

3280
3281
3282

.SBTTL TRAP DECODER

*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
*GO TO THAT ROUTINE.

017246	010046		\$TRAP:	MOV	R0,-(SP)	::SAVE R0
017250	016600	000002		MOV	2(SP),R0	::GET TRAP ADDRESS
017254	005740			TST	-(R0)	::BACKUP BY 2
017256	111000			MOVB	(R0),R0	::GET RIGHT BYTE OF TRAP
017260	006300			ASL	R0	::POSITION FOR INDEXING
017262	016000	017302		MOV	\$TRPAD(R0),R0	::INDEX TO TABLE
017266	000200			RTS	R0	::GO TO ROUTINE

::THIS IS USE TO HANDLE THE 'GETPRI' MACRO

017270	011646		\$TRAP2:	MOV	(SP),-(SP)	::MOVE THE PC DOWN
017272	016666	000004 000002		MOV	4(SP),2(SP)	::MOVE THE PSW DOWN
017300	000002			RTI		::RESTORE THE PSW

.SBTTL TRAP TABLE

*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
*BY THE 'TRAP' INSTRUCTION.

			:	ROUTINE	
			:	-----	
017302	017270		\$TRPAD:	.WORD	\$TRAP2
017304	015712			\$TYPE	::CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
017306	016270			\$TYPOC	::CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
017310	016244			\$TYPOS	::CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
017312	016304			\$TYPON	::CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
017314	016542			\$GTSWR	::CALL=GTSWR TRAP+5(104405) GET SOFT-SWR SETTING
017316	016472			\$CKSWR	::CALL=CKSWR TRAP+6(104406) TEST FOR CHANGE IN SOFT-SWR
017320	016754			\$RDCHR	::CALL=RDCHR TRAP+7(104407) TTY TYPEIN CHARACTER ROUTINE
017322	017074			\$RDLIN	::CALL=RDLIN TRAP+10(104410) TTY TYPEIN STRING ROUTINE

3283

```

3285 .SBTTL ECHO TEST
3286 ::*****
3287 ;*THIS ROUTINE WILL ECHO ANY CHARACTER TYPED
3288 ;*AT THE CONSOLE
3289 ;*THE TEST IS HALTED BY TYPING A CONTROL-C
3290 ;*TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING
3291 ::*****
3292 ECHO:
3293 RESET ;CLEAR EVERYTHING
3294 MOV #*,@CTBUF ;TRANSMIT PROMPT '*'
3295 TSTB @CRCSR ;WAIT FOR INPUT
3296 BPL 2$
3297 MOV @CRBUF,@CTBUF ;ECHO INPUT
3298 MOV @CRBUF,R0 ;STORE INPUT
3299 BPL 5$ ;BR IF 'ERROR' NOT SET
3300 BIS #BIT12,R1 ;SET PARITY ERROR TEST MASK
3301 BIT R1,R0 ;CHECK FOR PARITY ERROR FLAG
3302 BEQ 3$ ;BR IF NOT SET
3303 JSR PC,MSG ;REPORT PARITY ERROR
3304 MPAR
3305 3$: ASL R1 ;SHIFT MASK TO TEST 'FR' FLAG
3306 BIT R1,R0 ;TEST FOR FRAMING ERROR FLAG
3307 BEQ 4$ ;BR IF NOT SET
3308 JSR PC,MSG ;REPORT FRAMING ERROR
3309 MFR
3310 4$: ASL R1 ;SHIFT MASK TO TEST 'OR' FLAG
3311 BIT R1,R0 ;TEST FOR OVERFLOW ERROR
3312 BEQ 5$ ;BR IF NOT SET
3313 JSR PC,MSG ;REPORT OVERFLOW ERROR
3314 MOR
3315 5$: BIC #BIT7,R0 ;CLEAR ANY PARITY BIT
3316 CMP #3,R0 ;WAS INPUT CONTROL-C
3317 BNE 2$ ;BR IF NOT
3318 JSR PC,MSG ;REPORT PROGRAM STOP
3319 MSTOP
3320 HALT ;END OF TEST HALT
3321 BR ECHO ;AFTER END OF TEST HALT
3322 ; PRESS CONTINUE TO RESTART ECHO TEST
3323
3324 MSG: MOV @ (SP)+,R0 ;PICK UP MESSAGE POINTER
3325 ADD #2,-(SP) ;ADJUST RETURN PC
3326 WAIT: TSTB @CTCSR ;WAIT FOR XMIT DONE
3327 BPL WAIT
3328 MOVB (R0)+,@CTBUF ;SEND CHARACTER
3329 TSTB (R0) ;IS THIS END OF MESSAGE?
3330 BNE WAIT ;BR IF NOT
3331 RTS PC ;RETURN
3332
3333 017476 015 012 120 MPAR: .ASCIIZ <CR><LF>/PARITY/
3334 017501 101 122 111
3335 017504 124 131 000
3336 017507 015 012 106 MFR: .ASCIIZ <CR><LF>/FRAMING/
3337 017512 122 101 115
3338 017515 111 116 107
3339 017520 000
3340 017521 015 012 117 MOR: .ASCIIZ <CR><LF>/OVERFLOW/
3341 017524 126 105 122

```

	017527	106	114	117	
	017532	127	000		
3336	017534	015	012	123	MSTOP: .ASCIZ <CR><LF>/STOP/
	017537	124	117	120	
	017542	000			

3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371

017544 012701 000040
017550 012700 000040
017554 105777 163100
017560 100375
017562 010177 163074
017566 105201
017570 005300
017572 001370
017574 004767 177650

017600 001063
017602 105777 163046
017606 100404
017610 032701 000200
017614 001353
017616 000754

017620 005077 163032
017624 000000
017626 000746

```
.EVEN
.SBTTL  TERMINAL OUTPUT TEST

;*****
; *THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE
; *THE OCTAL CODE 040 --> 377
; *32 CHARACTERS ARE PRINTED ON EACH LINE
; *THE PATTERN IS REPEATED EVERY THREE LINES
; *
;*****

OUTTST: MOV    #40,R1          ;LOAD FIRST WRITABLE CHARACTER
1$:     MOV    #40,R0          ;LOAD CHAR COUNT PER LINE
2$:     TSTB   @CTCSR          ;WAIT FOR DONE
        BPL    2$
        MOV    R1,@CTBUF       ;TRANSMIT A CHARACTER
        INCB   R1              ;INCREMENT CHARACTER CODE
        DEC    R0              ;DECREMENT CHAR COUNT
        BNE    2$              ;BR IF LINE NOT COMPLETE
        JSR    PC,MSG          ;SSUE CR,LINE FEED

        $CRLF
        TSTB   @CRCSR          ;ANY CHARACTER RECEIVED?
        BMI    3$              ;BR IF YES
        BIT    #BIT7,R1        ;FINISHED ONE PASS OF WRITABLE CHARACTERS?
        BNE    OUTTST          ;BR IF YES
        BR     1$              ;IF NOT WRITE NEXT LINE

3$:     CLR    @CRBUF          ;CLEAR RECEIVER
        HALT
        BR     OUTTST          ;RESTART TEST IF CONTINUED
```

```
3373
3374
3375 017630      103      101      116  EM1:  .ASCIZ  /CAN NOT ACCESS TCSR/
      017633      040      116      117
      017636      124      040      101
      017641      103      103      105
      017644      123      123      040
      017647      124      103      123
      017652      122      000
3376 017654      103      101      116  EM2:  .ASCIZ  /CAN NOT ACCESS TBUF/
      017657      040      116      117
      017662      124      040      101
      017665      103      103      105
      017670      123      123      040
      017673      124      102      125
      017676      106      000
3377 017700      124      103      123  EM3:  .ASCIZ  /TCSR DONE NOT CLEARED WITH TBUF FULL/
      017703      122      040      104
      017706      117      116      105
      017711      040      116      117
      017714      124      040      103
      017717      114      105      101
      017722      122      105      104
      017725      040      127      111
      017730      124      110      040
      017733      124      102      125
      017736      106      040      106
      017741      125      114      114
      017744      000
3378 017745      124      103      123  EM4:  .ASCIZ  /TCSR DONE NOT SET/
      017750      122      040      104
      017753      117      116      105
      017756      040      116      117
      017761      124      040      123
      017764      105      124      000
3379 017767      124      103      123  EM5:  .ASCIZ  /TCSR DONE NOT SET WITH RESET/
      017772      122      040      104
      017775      117      116      105
      020000      040      116      117
      020003      124      040      123
      020006      05      124      040
      020011      127      111      124
      020014      110      040      122
      020017      105      123      105
      020022      124      000
3380 020024      103      101      116  EM6:  .ASCIZ  /CAN NOT ACCESS RCSR/
      020027      040      116      117
      020032      124      040      101
      020035      103      103      105
      020040      123      123      040
      020043      122      103      123
      020046      122      000
3381 020050      103      101      116  EM7:  .ASCIZ  /CAN NOT ACCESS RBUF/
      020053      040      116      117
      020056      124      040      101
      020061      103      103      105
      020064      123      123      040
```

	020067	122	102	125	
	020072	106	000		
3382	020074	103	101	116	EM10: .ASCIZ /CAN NOT ACCESS L&S/
	020077	040	116	117	
	020102	124	040	101	
	020105	103	103	105	
	020110	123	123	040	
	020113	114	113	123	
	020116	000			
3383	020117	102	111	124	EM11: .ASCIZ /BIT0 OF TCSR NOT CLEAR AFTER RESET/
	020122	060	040	117	
	020125	106	040	124	
	020130	103	123	122	
	020133	040	116	117	
	020136	124	040	103	
	020141	114	105	101	
	020144	122	040	101	
	020147	106	124	105	
	020152	122	040	122	
	020155	105	123	105	
	020160	124	000		
3384	020162	103	101	116	EM12: .ASCIZ /CAN NOT SET BIT0 OF TCSR/
	020165	040	116	117	
	020170	124	040	123	
	020173	105	124	040	
	020176	102	111	124	
	020201	060	040	117	
	020204	106	040	124	
	020207	103	123	122	
	020212	000			
3385	020213	103	101	116	EM13: .ASCIZ /CAN NOT CLEAR BIT0 OF TCSR/
	020216	040	116	117	
	020221	124	040	103	
	020224	114	105	101	
	020227	122	040	102	
	020232	111	124	060	
	020235	040	117	106	
	020240	040	124	103	
	020243	123	122	000	
3386	020246	122	105	123	EM14: .ASCIZ /RESET DID NOT CLEAR BIT0 OF TCSR/
	020251	105	124	040	
	020254	104	111	104	
	020257	040	116	117	
	020262	124	040	103	
	020265	114	105	101	
	020270	122	040	102	
	020273	111	124	060	
	020276	040	117	106	
	020301	040	124	103	
	020304	123	122	000	
3387	020307	102	111	124	EM15: .ASCIZ /BIT2 OF TCSR NOT CLEAR AFTER RESET/
	020312	062	040	117	
	020315	106	040	124	
	020320	103	123	122	
	020323	040	116	117	
	020326	124	040	103	
	020331	114	105	101	

	020334	122	040	101	
	020337	106	124	105	
	020342	122	040	122	
	020345	105	123	105	
	020350	124	000		
3388	020352	103	101	116	EM16: .ASCIZ /CAN NOT SET BIT2 OF TCSR/
	020355	040	116	117	
	020360	124	040	123	
	020363	105	124	040	
	020366	102	111	124	
	020371	062	040	117	
	020374	106	040	124	
	020377	103	123	122	
	020402	000			
3389	020403	103	101	116	EM17: .ASCIZ /CAN NOT CLEAR BIT2 OF TCSR/
	020406	040	116	117	
	020411	124	040	103	
	020414	114	105	101	
	020417	122	040	102	
	020422	111	124	062	
	020425	040	117	106	
	020430	040	124	103	
	020433	123	122	000	
3390	020436	122	105	123	EM20: .ASCIZ /RESET DID NOT CLEAR BIT2 OF TCSR/
	020441	105	124	040	
	020444	104	111	104	
	020447	040	116	117	
	020452	124	040	103	
	020455	114	105	101	
	020460	122	040	102	
	020463	111	124	062	
	020466	040	117	106	
	020471	040	124	103	
	020474	123	122	000	
3391	020477	102	111	124	EM21: .ASCIZ /BIT6 OF TCSR NOT CLEAR AFTER RESET/
	020502	066	040	117	
	020505	106	040	124	
	020510	103	123	122	
	020513	040	116	117	
	020516	124	040	103	
	020521	114	105	101	
	020524	122	040	101	
	020527	106	124	105	
	020532	122	040	122	
	020535	105	123	105	
	020540	124	000		
3392	020542	130	115	111	EM22: .ASCIZ /XMIT INTERRUPT AT PRIORITY 7/
	020545	124	040	111	
	020550	116	124	105	
	020553	122	122	125	
	020556	120	124	040	
	020561	101	124	040	
	020564	120	122	111	
	020567	117	122	111	
	020572	124	131	040	
	020575	067	000		
3393	020577	103	101	116	EM23: .ASCIZ /CAN NOT SET BIT6 OF TCSR/

	020602	040	116	117	
	020605	124	040	123	
	020610	105	124	040	
	020613	102	111	124	
	020616	066	040	117	
	020621	106	040	124	
	020624	103	123	122	
	020627	000			
3394	020630	103	101	116	EM24: .ASCIZ /CAN NOT CLEAR BIT6 OF TCSR/
	020633	040	116	117	
	020636	124	040	103	
	020641	114	105	101	
	020644	122	040	102	
	020647	111	124	066	
	020652	040	117	106	
	020655	040	124	103	
	020660	123	122	000	
3395	020663	122	105	123	EM25: .ASCIZ /RESET DID NOT CLEAR BIT6 OF TCSR/
	020666	105	124	040	
	020671	104	111	104	
	020674	040	116	117	
	020677	124	040	103	
	020702	114	105	101	
	020705	122	040	102	
	020710	111	124	066	
	020713	040	117	106	
	020716	040	124	103	
	020721	123	122	000	
3396	020724	102	111	124	EM26: .ASCIZ /BIT6 OF RCSR NOT CLEAR AFTER RESET/
	020727	066	040	117	
	020732	106	040	122	
	020735	103	123	122	
	020740	040	116	117	
	020743	124	040	103	
	020746	114	105	101	
	020751	122	040	101	
	020754	106	124	105	
	020757	122	040	122	
	020762	105	123	105	
	020765	124	000		
3397	020767	122	103	126	EM27: .ASCIZ /RCVR INTERRUPT WITH PRIORITY 7/
	020772	122	040	111	
	020775	116	124	105	
	021000	122	122	125	
	021003	120	124	040	
	021006	127	111	124	
	021011	110	040	120	
	021014	122	111	117	
	021017	122	111	124	
	021022	131	040	067	
	021025	000			
3398	021026	103	101	116	EM30: .ASCIZ /CAN NOT SET BIT6 OF RCSR/
	021031	040	116	117	
	021034	124	040	123	
	021037	105	124	040	
	021042	102	111	124	
	021045	066	040	117	

	021050	106	040	122	
	021053	103	123	122	
	021056	000			
3399	021057	103	101	116	EM31: .ASCIIZ /CAN NOT CLEAR BIT6 OF RCSR/
	021062	040	116	117	
	021065	124	040	103	
	021070	114	105	101	
	021073	122	040	102	
	021076	111	124	066	
	021101	040	117	106	
	021104	040	122	103	
	021107	123	122	000	
3400	021112	103	101	116	EM32: .ASCIIZ /CAN NOT CLEAR BIT6 OF RCSR WITH RESET/
	021115	040	116	117	
	021120	124	040	103	
	021123	114	105	101	
	021126	122	040	102	
	021131	111	124	066	
	021134	040	117	106	
	021137	040	122	103	
	021142	123	122	040	
	021145	127	111	124	
	021150	110	040	122	
	021153	105	123	105	
	021156	124	000		
3401	021160	102	111	124	EM33: .ASCIIZ /BIT6 OF LKS NOT CLEAR AFTER RESET/
	021163	066	040	117	
	021166	106	040	114	
	021171	113	123	040	
	021174	116	117	124	
	021177	040	103	114	
	021202	105	101	122	
	021205	040	101	106	
	021210	124	105	122	
	021213	040	122	105	
	021216	123	105	124	
	021221	000			
3402	021222	114	113	123	EM34: .ASCIIZ /LKS INTERRUPT WITH PRIORITY 7/
	021225	040	111	116	
	021230	124	105	122	
	021233	122	125	120	
	021236	124	040	127	
	021241	111	124	110	
	021244	040	120	122	
	021247	111	117	122	
	021252	111	124	131	
	021255	040	067	000	
3403	021260	103	101	116	EM35: .ASCIIZ /CAN NOT SET BIT6 OF LKS/
	021263	040	116	117	
	021266	124	040	123	
	021271	105	124	040	
	021274	102	111	124	
	021277	066	040	117	
	021302	106	040	114	
	021305	113	123	000	
3404	021310	103	101	116	EM36: .ASCIIZ /CAN NOT CLEAR BIT6 OF LKS/
	021313	040	116	117	

	021316	124	040	103	
	021321	114	105	101	
	021324	122	040	102	
	021327	111	124	066	
	021332	040	117	106	
	021335	040	114	113	
	021340	123	000		
3405	021342	122	105	123	EM37: .ASCIZ /RESET DID NOT CLEAR BIT6 OF LKS/
	021345	105	124	040	
	021350	104	111	104	
	021353	040	116	117	
	021356	124	040	103	
	021361	114	105	101	
	021364	122	040	102	
	021367	111	124	066	
	021372	040	117	106	
	021375	040	114	113	
	021400	123	000		
3406	021402	104	125	101	EM40: .ASCIZ /DUAL ADDRESSING ERROR/
	021405	114	040	101	
	021410	104	104	122	
	021413	105	123	123	
	021416	111	116	107	
	021421	040	105	122	
	021424	122	117	122	
	021427	000			
3407	021430	102	111	124	EM41: .ASCIZ /BIT6 OF LKS NOT SET AFTER RESET/
	021433	066	040	117	
	021436	106	040	114	
	021441	113	123	040	
	021444	116	117	124	
	021447	040	123	105	
	021452	124	040	101	
	021455	106	124	105	
	021460	122	040	122	
	021463	105	123	105	
	021466	124	000		
3408	021470	103	101	116	EM42: .ASCIZ /CAN NOT CLEAR BIT7 OF LKS/
	021473	040	116	117	
	021476	124	040	103	
	021501	114	105	101	
	021504	122	040	102	
	021507	111	124	067	
	021512	040	117	106	
	021515	040	114	113	
	021520	123	000		
3409	021522	102	111	124	EM43: .ASCIZ /BIT7 OF LKS DOES NOT SET/
	021525	067	040	117	
	021530	106	040	114	
	021533	113	123	040	
	021536	104	117	105	
	021541	123	040	116	
	021544	117	124	040	
	021547	123	105	124	
	021552	000			
3410	021553	122	124	103	EM44: .ASCIZ /RTC INTERRUPT AT PRIORITY 7/
	021556	040	111	116	

	021561	124	105	122	
	021564	122	125	120	
	021567	124	040	101	
	021572	124	040	120	
	021575	122	111	117	
	021600	122	111	124	
	021603	131	040	067	
	021606	000			
3411	021607	122	124	103	EM45: .ASCIIZ /RTC INTERRUPTS WHEN DISABLED/
	021612	040	111	116	
	021615	124	105	122	
	021620	122	125	120	
	021623	124	123	040	
	021626	127	110	105	
	021631	116	040	104	
	021634	111	123	101	
	021637	102	114	105	
	021642	104	000		
3412	021644				EM47:
3413	021644	122	124	103	EM46: .ASCIIZ /RTC INTERRUPT DID NOT OCCUR/
	021647	040	111	116	
	021652	124	105	122	
	021655	122	125	120	
	021660	124	040	104	
	021663	111	104	040	
	021666	116	117	124	
	021671	040	117	103	
	021674	103	125	122	
	021677	000			
3414	021700	122	124	103	EM50: .ASCIIZ /RTC DOUBLE INTERRUPT/
	021703	040	104	117	
	021706	125	102	114	
	021711	105	040	111	
	021714	116	124	105	
	021717	122	122	125	
	021722	120	124	000	
3415	021725	122	105	123	EM51: .ASCIIZ /RESET DID NOT INTERRUPT/
	021730	105	124	040	
	021733	104	111	104	
	021736	040	116	117	
	021741	124	040	111	
	021744	116	124	105	
	021747	122	122	125	
	021752	120	124	000	
3416	021755	122	124	103	EM52: .ASCIIZ /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
	021760	040	111	116	
	021763	124	105	122	
	021766	122	125	120	
	021771	124	040	104	
	021774	111	104	040	
	021777	116	117	124	
	022002	040	103	114	
	022005	105	101	122	
	022010	040	127	111	
	022013	124	110	040	
	022016	102	111	124	
	022021	067	040	117	

Address	Hex	ASCII	Comment
3417	022024	106	040
	022027	113	123
	022032	103	114
	022035	103	113
	022040	122	105
	022043	105	101
	022046	101	102
	022051	114	111
	022054	131	000
3418	022056	130	115
	022061	124	040
	022064	116	124
	022067	122	122
	022072	120	124
	022075	040	127
	022100	105	116
	022103	104	111
	022106	101	102
	022111	105	104
3419	022114	130	115
	022117	124	040
	022122	116	124
	022125	122	122
	022130	120	124
	022133	040	101
	022136	040	120
	022141	111	117
	022144	111	124
	022147	040	067
3420	022152	130	115
	022155	124	040
	022160	116	124
	022163	122	122
	022166	120	124
	022171	040	127
	022174	124	110
	022177	105	116
	022202	102	114
	022205	040	103
	022210	105	101
	022213	000	
3421	022214		
3422	022214	130	115
	022217	124	040
	022222	111	104
	022225	116	117
	022230	040	111
	022233	124	105
	022236	122	125
	022241	124	000
3423	022243	130	115
	022246	124	040
	022251	105	055
	022254	116	124
	022257	122	122
	022262	120	124
	022265	104	000

3424	022267	114	117	101	EM62: .ASCIIZ /LOADING TBUF DID NOT CLEAR INTERRUPT/
	022272	104	111	116	
	022275	107	040	124	
	022300	102	125	106	
	022303	040	104	111	
	022306	104	040	116	
	022311	117	124	040	
	022314	103	114	105	
	022317	101	122	040	
	022322	111	116	124	
	022325	105	122	122	
	022330	125	120	124	
	022333	000			
3425	022334	122	103	126	EM63: .ASCIIZ /RCVR ACTIVE NOT SET/
	022337	122	040	101	
	022342	103	124	111	
	022345	126	105	040	
	022350	116	117	124	
	022353	040	123	105	
	022356	124	000		
3426	022360	122	103	126	EM64: .ASCIIZ /RCVR DONE NEVER SET/
	022363	122	040	104	
	022366	117	116	105	
	022371	040	116	105	
	022374	126	105	122	
	022377	040	123	105	
	022402	124	000		
3427	022404	122	103	126	EM65: .ASCIIZ /RCVR ACTIVE NOT CLEARED WITH DONE SET/
	022407	122	040	101	
	022412	103	124	111	
	022415	126	105	040	
	022420	116	117	124	
	022423	040	103	114	
	022426	105	101	122	
	022431	105	104	040	
	022434	127	111	124	
	022437	110	040	104	
	022442	117	116	105	
	022445	040	123	105	
	022450	124	000		
3428	022452	122	105	123	EM66: .ASCIIZ /RESET DID NOT CLEAR RCVR DONE/
	022455	105	124	040	
	022460	104	111	104	
	022463	040	116	117	
	022466	124	040	103	
	022471	114	105	101	
	022474	122	040	122	
	022477	103	126	122	
	022502	040	104	117	
	022505	116	105	000	
3429	022510	122	104	122	EM67: .ASCIIZ /RDR ENABLE DID NOT CLEAR RCVR DONE/
	022513	040	105	116	
	022516	101	102	114	
	022521	105	040	104	
	022524	111	104	040	
	022527	116	117	124	
	022532	040	103	114	

	022535	105	101	122	
	022540	040	122	103	
	022543	126	122	040	
	022546	104	117	116	
	022551	105	000		
3430	022553	122	105	101	EM70: .ASCIZ /READING RBUF DID NOT CLEAR RCVR DONE/
	022556	104	111	116	
	022561	107	040	122	
	022564	102	125	106	
	022567	040	104	111	
	022572	104	040	116	
	022575	117	124	040	
	022600	103	114	105	
	022603	101	122	040	
	022606	122	103	126	
	022611	122	040	104	
	022614	117	116	105	
	022617	000			
3431	022620	122	103	126	EM71: .ASCIZ /RCVR INTERRUPTS WITH ENABLE CLEAR/
	022623	122	040	111	
	022626	116	124	105	
	022631	122	122	125	
	022634	120	124	123	
	022637	040	127	111	
	022642	124	110	040	
	022645	105	116	101	
	022650	102	114	105	
	022653	040	103	114	
	022656	105	101	122	
	022661	000			
3432	022662	122	103	126	EM73: .ASCIZ /RCVR INTERRUPTS AT PRIORITY 7/
	022665	122	040	111	
	022670	116	124	105	
	022673	122	122	125	
	022676	120	124	123	
	022701	040	101	124	
	022704	040	120	122	
	022707	111	117	122	
	022712	111	124	131	
	022715	040	067	000	
3433	022720	122	103	126	EM74: .ASCIZ /RCVR INT RQST PASSED WITH ENABLE CLEAR/
	022723	122	040	111	
	022726	116	124	040	
	022731	122	121	123	
	022734	124	040	120	
	022737	101	123	123	
	022742	105	104	040	
	022745	127	111	124	
	022750	110	040	105	
	022753	116	101	102	
	022756	114	105	040	
	022761	103	114	105	
	022764	101	122	000	
3434	022767				EM72:
3435	022767	122	103	126	EM75: .ASCIZ /RCVR DID NOT INTERRUPT/
	022772	122	040	104	
	022775	111	104	040	

	023000	116	117	124	
	023003	040	111	116	
	023006	124	105	122	
	023011	122	125	120	
	023014	124	000		
3436	023016	122	103	126	EM76: .ASCIIZ /RCVR RE-INTERRUPTED/
	023021	122	040	122	
	023024	105	055	111	
	023027	116	124	105	
	023032	122	122	125	
	023035	120	124	105	
	023040	104	000		
3437	023042	122	105	101	EM77: .ASCIIZ /READING RBUF DID NOT CLEAR INTERRUPT/
	023045	104	111	116	
	023050	107	040	122	
	023053	102	125	106	
	023056	040	104	111	
	023061	104	040	116	
	023064	117	124	040	
	023067	103	114	105	
	023072	101	122	040	
	023075	111	116	124	
	023100	105	122	122	
	023103	125	120	124	
	023106	000			
3438	023107	122	105	123	EM100: .ASCIIZ /RESET DID NOT CLEAR RCVR INTERRUPT/
	023112	105	124	040	
	023115	104	111	104	
	023120	040	116	117	
	023123	124	040	103	
	023126	114	105	101	
	023131	122	040	122	
	023134	103	126	122	
	023137	040	111	116	
	023142	124	105	122	
	023145	122	125	120	
	023150	124	000		
3439	023152	047	117	122	EM101: .ASCIIZ /'OR' FLAG DID NOT SET/
	023155	047	040	106	
	023160	114	101	107	
	023163	040	104	111	
	023166	104	040	116	
	023171	117	124	040	
	023174	123	105	124	
	023177	000			
3440	023200	047	105	122	EM102: .ASCIIZ /'ERROR' NOT SET WITH 'OR' FLAG/
	023203	122	117	122	
	023206	047	040	116	
	023211	117	124	040	
	023214	123	105	124	
	023217	040	127	111	
	023222	124	110	040	
	023225	047	117	122	
	023230	047	040	106	
	023233	114	101	107	
	023236	000			
3441	023237	102	122	105	EM103: .ASCIIZ /BREAK DID NOT XMIT ALL ZEROES/

	023242	101	113	040	
	023245	104	111	104	
	023250	040	116	117	
	023253	124	040	130	
	023256	115	111	124	
	023261	040	101	114	
	023264	114	040	132	
	023267	105	122	117	
	023272	105	123	000	
3442	023275	102	122	105	EM104: .ASCIZ /BREAK DID NOT SET 'FR' ERROR/
	023300	101	113	040	
	023303	104	111	104	
	023306	040	116	117	
	023311	124	040	123	
	023314	105	124	040	
	023317	047	106	122	
	023322	047	040	105	
	023325	122	122	117	
	023330	122	000		
3443	023332	104	101	124	EM105: .ASCIZ /DATA COMPARE ERROR/
	023335	101	040	103	
	023340	117	115	120	
	023343	101	122	105	
	023346	040	105	122	
	023351	122	117	122	
	023354	000			
3444	023355	104	101	124	EM106: .ASCIZ /DATA COMPARE ERROR WITH CABLE/
	023360	101	040	103	
	023363	117	115	120	
	023366	101	122	105	
	023371	040	105	122	
	023374	122	117	122	
	023377	040	127	111	
	023402	124	110	040	
	023405	103	101	102	
	023410	114	105	000	
3445	023413	124	111	115	EM107: .ASCIZ /TIMEOUT IN EXERCISER TEST/
	023416	105	117	125	
	023421	124	040	111	
	023424	116	040	105	
	023427	130	105	122	
	023432	103	111	123	
	023435	105	122	040	
	023440	124	105	123	
	023443	124	000		
3446	023445	111	116	103	EM110: .ASCIZ /INCORRECT RECEIVE COUNT/
	023450	117	122	122	
	023453	105	103	124	
	023456	040	122	105	
	023461	103	105	111	
	023464	126	105	040	
	023467	103	117	125	
	023472	116	124	000	
3447	023475	104	101	124	EM111: .ASCIZ /DATA COMPARE ERROR IN EXERCISER/
	023500	101	040	103	
	023503	117	115	120	
	023506	101	122	105	

	023511	040	105	122	
	023514	122	117	122	
	023517	040	111	116	
	023522	040	105	130	
	023525	105	122	103	
	023530	111	123	105	
	023533	122	000		
3448	023535	124	122	101	EM112: .ASCIZ /TRAP CATCHER/
	023540	120	040	103	
	023543	101	124	103	
	023546	110	105	122	
	023551	000			
3449	023552	116	117	040	EM113: .ASCIZ /NO CLK INTERRUPT IN EXERCISER/
	023555	103	114	113	
	023560	040	111	116	
	023563	124	105	122	
	023566	122	125	120	
	023571	124	040	111	
	023574	116	040	105	
	023577	130	105	122	
	023602	103	111	123	
	023605	105	122	000	
3450	023610	042	105	122	EM114: .ASCIZ /'ERROR' NOT SET WITH 'FR' FLAG/
	023613	122	117	122	
	023616	042	040	116	
	023621	117	124	040	
	023624	123	105	124	
	023627	040	127	111	
	023632	124	110	040	
	023635	042	106	122	
	023640	042	040	106	
	023643	114	101	107	
	023646	000			
3451	023647	122	103	126	EM115: .ASCIZ /RCV ACTIVE NOT CLEAR WITH INIT/
	023652	040	101	103	
	023655	124	111	126	
	023660	105	040	116	
	023663	117	124	040	
	023666	103	114	105	
	023671	101	122	040	
	023674	127	111	124	
	023677	110	040	111	
	023702	116	111	124	
	023705	000			
3452	023706	122	103	126	EM116: .ASCIZ /RCV ACTIVE WITHOUT 'START' BIT/
	023711	040	101	103	
	023714	124	111	126	
	023717	105	040	127	
	023722	111	124	110	
	023725	117	125	124	
	023730	040	042	123	
	023733	124	101	122	
	023736	124	042	040	
	023741	102	111	124	
	023744	000			
3453	023745	122	104	122	EM117: .ASCIZ /RDR ENABLE NOT CLEAR WITH RCV ACTIVE/
	023750	040	105	116	

	023753	101	102	114					
	023756	105	040	116					
	023761	117	124	040					
	023764	103	114	105					
	023767	101	122	040					
	023772	127	111	124					
	023775	110	040	122					
	024000	103	126	040					
	024003	101	103	124					
	024006	111	126	105					
	024011	000							
3454									
3455	024012	124	105	123	DH1:	.ASCIIZ	/TEST#	ERROR#	TCSR/
	024015	124	043	040					
	024020	040	040	105					
	024023	122	122	117					
	024026	122	043	040					
	024031	040	124	103					
	024034	123	122	000					
3456	024037	124	105	123	DH2:	.ASCIIZ	/TEST#	ERR PC	TBUF/
	024042	124	043	040					
	024045	040	040	105					
	024050	122	122	040					
	024053	120	103	040					
	024056	040	124	102					
	024061	125	106	000					
3457	024064	124	105	123	DH6:	.ASCIIZ	/TEST#	ERR PC	RCSR/
	024067	124	043	040					
	024072	040	040	105					
	024075	122	122	040					
	024100	120	103	040					
	024103	040	122	103					
	024106	123	122	000					
3458	024111	124	105	123	DH7:	.ASCIIZ	/TEST#	ERR PC	RBUF/
	024114	124	043	040					
	024117	040	040	105					
	024122	122	122	040					
	024125	120	103	040					
	024130	040	122	102					
	024133	125	106	000					
3459	024136	124	105	123	DH10:	.ASCIIZ	/TEST#	ERR PC	LKS/
	024141	124	043	040					
	024144	040	040	105					
	024147	122	122	040					
	024152	120	103	040					
	024155	040	114	113					
	024160	123	000						
3460	024162	124	105	123	DH40:	.ASCIIZ	/TEST#	ERR PC	GOOD BAD/
	024165	124	043	040					
	024170	040	040	105					
	024173	122	122	040					
	024176	120	103	040					
	024201	040	107	117					
	024204	117	104	040					
	024207	040	040	040					
	024212	102	101	104					
	024215	000							

3461	024216	124	105	123	DH53: .ASCIZ /TEST#	ERR PC	LKS	CNT1	CNT2/
	024221	124	043	040					
	024224	040	040	105					
	024227	122	122	040					
	024232	120	103	040					
	024235	040	114	113					
	024240	123	040	040					
	024243	040	040	040					
	024246	103	116	124					
	024251	061	040	040					
	024254	040	040	103					
	024257	116	124	062					
	024262	000							
3462	024263	124	105	123	DH103: .ASCIZ /TEST#	ERR PC	RCSR	DATA/	
	024266	124	043	040					
	024271	040	040	105					
	024274	122	122	040					
	024277	120	103	040					
	024302	040	122	103					
	024305	123	122	040					
	024310	040	040	040					
	024313	104	101	124					
	024316	101	000						
3463	024320	124	105	123	DH105: .ASCIZ /TEST#	ERR PC	RCSR	GOOD	BAD/
	024323	124	043	040					
	024326	040	040	105					
	024331	122	122	040					
	024334	120	103	040					
	024337	040	122	103					
	024342	123	122	040					
	024345	040	040	040					
	024350	107	117	117					
	024353	104	040	040					
	024356	040	040	102					
	024361	101	104	000					
3464	024364	124	105	123	DH110: .ASCIZ /TEST#	ERR PC	RCSR	TRANS	RCV/
	024367	124	043	040					
	024372	040	040	105					
	024375	122	122	040					
	024400	120	103	040					
	024403	040	122	103					
	024406	123	122	040					
	024411	040	040	040					
	024414	124	122	101					
	024417	116	123	040					
	024422	040	040	122					
	024425	103	126	000					
3465	024430	124	105	123	DH112: .ASCIZ /TEST#	ERR PC	RCSR	OLDPC	TRAP ADR/
	024433	124	043	040					
	024436	040	040	105					
	024441	122	122	040					
	024444	120	103	040					
	024447	040	122	103					
	024452	123	122	040					
	024455	040	040	040					
	024460	117	114	104					
	024463	120	103	040					

	024466	040	040	124			
	024471	122	101	120			
	024474	040	101	104			
	024477	122	000				
3466						.EVEN	
3467	024502	015	012	103	M1:	.ASCIIZ	<CR><LF>'CZDLDDO DL11-W,1144 MFM SLU/
	024505	132	104	114			
	024510	104	104	060			
	024513	040	104	114			
	024516	061	061	055			
	024521	127	054	061			
	024524	061	064	064			
	024527	040	115	106			
	024532	115	040	123			
	024535	114	125	000			
3468						.EVEN	
3469	024540	015	012		M2:	.ASCII	<CR><LF>
3470	024542	040	040	040	M2A:	.ASCIIZ	/ DEVICES UNDER TEST /
	024545	104	105	126			
	024550	111	103	105			
	024553	123	040	125			
	024556	116	104	105			
	024561	122	040	124			
	024564	105	123	124			
	024567	040	040	000			
3471							
3472						.EVEN	
3473	024572	001072	001016	002640	DT1:	.WORD	\$TESTN,\$ERRPC,TCSR,0
	024600	000000					
3474	024602	001072	001016	002642	DT2:	.WORD	\$TESTN,\$ERRPC,TBUF,0
	024610	000000					
3475	024612	001072	001016	002634	DT6:	.WORD	\$TESTN,\$ERRPC,RCSR,0
	024620	000000					
3476	024622	001072	001016	002636	DT7:	.WORD	\$TESTN,\$ERRPC,RBUF,0
	024630	000000					
3477	024632	001072	001016	002674	DT10:	.WORD	\$TESTN,\$ERRPC,LKS,0
	024640	000000					
3478	024642	001072	001016	001020	DT40:	.WORD	\$TESTN,\$ERRPC,\$GDADR,\$BDADR,0
	024650	001022	000000				
3479	024654	001072	001016	002674	DT53:	.WORD	\$TESTN,\$ERRPC,LKS,FIRST,SECND,0
	024662	002336	002616	000000			
3480	024670	001072	001016	002634	DT103:	.WORD	\$TESTN,\$ERRPC,RCSR,\$BDDAT,0
	024676	001026	000000				
3481	024702	001072	001016	002634	DT105:	.WORD	\$TESTN,\$ERRPC,RCSR,\$GDDAT,\$BDDAT,0
	024710	001024	001026	000000			
3482	024716	001072	001016	002634	DT110:	.WORD	\$TESTN,\$ERRPC,RCSR,XMTCNT,RCVCNT,0
	024724	002442	002440	000000			
3483	024732	001072	001016	002634	DT112:	.WORD	\$TESTN,\$ERRPC,RCSR,OLDPC,BDVECT,0
	024740	002620	002622	000000			
3484	024746	000000			ENDADR:	.WORD	0 ;END ADDRESS FOR CHECKSUM AND SAVE AREA
3485							:OF WRAP AROUND CONSOLE TEST
3486							
3487		000001				.END	

ABASE = 176500	BGNADD= 003014	DH110 024364	EM27 020767	GETLIN 014134
ACDW1 = 000000	BIT0 = 000001	DH112 024430	EM3 017700	GOAGIN 014504
ACDW2 = 000000	BIT00 = 000001	DH2 024037	EM30 021026	GTSWR = 104405
ACPUOP= 000000	BIT01 = 000002	DH40 024162	EM31 021057	HT = 000011
ADDW0 = 000000	BIT02 = 000004	DH53 024216	EM32 021112	ID 004620
ADDW1 = 000000	BIT03 = 000010	DH6 024064	EM33 021160	INIT 003462
ADDW10= 000000	BIT04 = 000020	DH7 024111	EM34 021222	IOTVEC= 000020
ADDW11= 000000	BIT05 = 000040	DISPLA 001042	EM35 021260	JIMSTK 002432
ADDW12= 000000	BIT06 = 000100	DISPRE 000174	EM36 021310	LF = 000012
ADDW13= 000000	BIT07 = 000200	DSWR = 177570	EM37 021342	LKS 002674
ADDW14= 000000	BIT08 = 000400	DT1 024572	EM4 017745	LOC1 002340
ADDW15= 000000	BIT09 = 001000	DT10 024632	EM40 021402	LOC2 002342
ADDW2 = 000000	BIT1 = 000002	DT103 024670	EM41 021430	MANL 003502
ADDW3 = 000000	BIT10 = 002000	DT105 024702	EM42 021470	MFPT = 000007
ADDW4 = 000000	BIT11 = 004000	DT110 024716	EM43 021522	MFR 017507
ADDW5 = 000000	BIT12 = 010000	DT112 024732	EM44 021553	MUR 017521
ADDW6 = 000000	BIT13 = 020000	DT2 024602	EM45 021607	MPAR 017476
ADDW7 = 000000	BIT14 = 040000	DT40 024642	EM46 021644	MSG 017450
ADDW8 = 000000	BIT15 = 100000	DT53 024654	EM47 021644	MSTOP 017534
ADDW9 = 000000	BIT2 = 000004	DT6 024612	EM5 017767	M1 024502
ADEVCT= 000000	BIT3 = 000010	DT7 024622	EM50 021700	M2 024540
ADEVN = 000000	BIT4 = 000020	DVADT 003664	EM51 021725	M2A 024542
ADR 004132	BIT5 = 000040	ECHO 017324	EM52 021755	NOEOP 014370
ADRTBL 002702	BIT6 = 000100	EMTVEC= 000030	EM53 022032	OLDPC 002620
AENV = 000000	BIT7 = 000200	EM1 017630	EM54 022056	OLDSUM 002436
AENVN = 000000	BIT8 = 000400	EM10 020074	EM55 022214	OUTTST 017544
AFATAL= 000000	BIT9 = 001000	EM100 023107	EM56 022114	PIRQ = 177772
AMADR1= 000000	BPT = 000003	EM101 023152	EM57 022152	PIRQVE= 000240
AMADR2= 000000	BPTVEC= 000014	EM102 023200	EM6 020024	PROMPT 002566
AMADR3= 000000	BUF 002454	EM103 023237	EM60 022214	PR0 = 000000
AMADR4= 000000	CATCH 014552	EM104 023275	EM61 022243	PR1 = 000040
AMAMS1= 000000	CHKSUM 014062	EM105 023332	EM62 022267	PR2 = 000100
AMAMS2= 000000	CKSWR = 104406	EM106 023355	EM63 022334	PR3 = 000140
AMAMS3= 000000	CLK 013702	EM107 023413	EM64 022360	PR4 = 000200
AMAMS4= 000000	CLKCNT 002444	EM11 020117	EM65 022404	PR5 = 000240
AMSGAD= 000000	COMPARE 007244	EM110 023445	EM66 022452	PR6 = 000300
AMSGLG= 000000	CNTLP 002564	EM111 023475	EM67 022510	PR7 = 000340
AMSGTY= 000000	COMP 013504	EM112 023535	EM7 020050	PS = 177776
AMTYP1= 000000	CONT 006246	EM113 023552	EM70 022553	PSW = 177776
AMTYP2= 000000	CONT41 012400	EM114 023610	EM71 022620	PUTLIN 014102
AMTYP3= 000000	CR = 000015	EM115 023647	EM72 022767	PWRVEC= 000024
AMTYP4= 000000	CRBUF 002656	EM116 023706	EM73 022662	RBUF 002636
APASS = 000000	CRCR 002654	EM117 023745	EM74 022720	RBUF58= 176502
APRICR= 000000	CRLF = 000200	EM12 020162	EM75 022767	RCSR 002634
APTCSU= 000040	CRPSW 002666	EM13 020213	EM76 023016	RCSR58= 176500
APTENV= 000001	CRVECT 002664	EM14 020246	EM77 023042	RCV 013666
APTSIZ= 000200	CTBUF 002662	EM15 020307	ENDADD= 024746	RCVCNT 002440
APTSPO= 000100	CTCSR 002660	EM16 020352	ENDADR 024746	RCVDON 010266
APTSZD 003636	CTPSW 002672	EM17 020403	ENDB7 004676	RDCHR = 104407
ASWREG= 000000	CTSTFL 002624	EM2 017654	ENDEV 014344	RDLIN = 104410
ATESTN= 000000	CTVECT 002670	EM20 020436	ENDMG 014470	RESTTE 014310
AUNIT = 000000	DDISP = 177570	EM21 020477	FNDSTK 02412	RESVEC= 000010
AUSWR = 000400	DEVADR 003014	EM22 020542	ERROR = 104000	RPSW 002646
AVECT1= 000300	DH1 024012	EM23 020577	ERRVEC= 000004	RSTRT 014530
AVECT2 000000	DH10 024136	EM24 020630	FIRST 002336	RTCPSW 002700
BDVECT 002622	DH103 024263	EM25 020663	FLAG44 003002	RTCVT 002676
BEGIN 003774	DH105 024320	EM26 020724	GETB 014200	RVECT 002644

R6 = 0000006	TCSR 002640	TST7 005012	\$ERMAX 001015	\$PASS 001074
R7 = 0000007	TCSR58= 176504	TURBUF 003006	\$ERROR 014576	\$PASTM 000506
SAVEPS 002434	TIMER 014234	TURCSR 003004	\$ERRPC 001016	\$PCWER 015262
SAVETE 014254	TKVEC = 000060	TUTBUF 003012	\$ERRTB 001146	\$PWRDN 015114
SAVEO 002344	TMP1 002626	TUTCSR 003010	\$ERRTY 014760	\$PWRMG 015250
SAVLOC 002346	TMP2 002630	TVECT 002650	\$ERTTL 001012	\$PWRUP 015166
SCOPE = 000004	TMP3 002632	TYPE = 104401	\$ESCAP 001060	\$QUES 001062
SCPSW 002452	TOLER 007256	TYPOC = 104402	\$ETABL 001106	\$RDCHR 016754
SECND 002616	TPSW 002652	TYPON = 104404	\$ETEND 001146	\$RDLIN 017074
SETADR 004100	TPVEC = 000064	TYPOS = 104403	\$FATAL 001070	\$RDSZ = 000010
SHIFT 003740	TRAPVE= 000034	VCTADR 003670	\$FFLG 015710	\$RTNAD 014462
SIZE 003512	TRTVEC= 000014	VCTTBL 002742	\$FILLC 001056	\$SAVR6 015260
SRPSW 002450	TSTDEV 004050	VECT 004152	\$FILLS 001055	\$SCOPE 015272
STACK = 001100	TSTDVM 003646	WACTV 010150	\$GDADR 001020	\$SETUP= 000137
START 003046	TST1 004172	WAIT 017456	\$GDDAT 001024	\$STUP = 177777
STKLMT= 177774	TST10 005044	WDONE 010302	\$GET42 014440	\$SVLAD 015374
STPSW 002446	TST11 005252	WRAP 013712	\$GTSWR 016542	\$SVPC = 000500
SWR 001040	TST12 005404	WRPSW 014540	\$HIBTS 000500	\$SWR = 161000
SWREG 000176	TST13 005544	WT 010236	\$ICNT 001004	\$SWREG 001110
SW0 = 000001	TST14 005614	XCONT 013656	\$ILLUP 015254	\$SWRMK= 000000
SW00 = 000001	TST15 005764	XMIT 013614	\$INTAG 001035	\$TESTN 001072
SW01 = 000002	TST16 006154	XMTCNT 002442	\$ITEMB 001014	\$TKB 001046
SW02 = 000004	TST17 006262	XRET 013662	\$LFLG 001064	\$TKS 001044
SW03 = 000010	TST2 004244	\$APTHD 000500	\$LFLG 015707	\$TN = 000047
SW04 = 000020	TST20 006504	\$ATYC 015470	\$LPADR 001006	\$TPB 001052
SW05 = 000040	TST21 006660	\$ATY1 015444	\$LPERR 001010	\$TPFLG 001057
SW06 = 000100	TST22 007002	\$ATY3 015452	\$MADR1 001120	\$TPS 001050
SW07 = 000200	TST23 007136	\$ATY4 015462	\$MADR2 001124	\$TRAP 017246
SW08 = 000400	TST24 007324	\$AUTOB 001034	\$MADR3 001130	\$TRAP2 017270
SW09 = 001000	TST25 007432	\$BASE 001142	\$MADR4 001134	\$TRP = 000011
SW1 = 000002	TST26 007560	\$BDADR 001022	\$MAIL 001066	\$TRPAD 017302
SW10 = 002000	TST27 007722	\$BDDAT 001026	\$MAMS1 001116	\$TSTM 000504
SW11 = 004000	TST3 004316	\$CHARC 016240	\$MAMS2 001122	\$TSTNM 001002
SW12 = 010000	TST30 010066	\$CKSWR 016472	\$MAMS3 001126	\$TTYIN 017202
SW13 = 020000	TST31 010402	\$CMTAG 001000	\$MAMS4 001132	\$TYPE 015712
SW14 = 040000	TST32 010504	\$CM3 = 000000	\$MBADR 000502	\$TYPEC 016124
SW15 = 100000	TST33 010640	\$CNTLG 017217	\$MFLG 015706	\$TYPEX 016242
SW2 = 000004	TST34 011050	\$CNTLU 017212	\$MNEW 017235	\$TYPOC 016270
SW3 = 000010	TST35 011262	\$CPUOP 001114	\$MSGAD 001102	\$TYPON 016304
SW4 = 000020	TST36 011476	\$CRLF 001063	\$MSGLG 001104	\$TYPOS 016244
SW5 = 000040	TST37 011652	\$DEVCT 001076	\$MSGTY 001066	\$UNIT 001100
SW6 = 000100	TST4 004466	\$DEVVM 001144	\$MSWR 017224	\$UNITM 000510
SW7 = 000200	TST40 012020	\$DOAGN 014460	\$MTYP1 001117	\$USWR 001112
SW8 = 000400	TST41 012214	\$ENDAD 014450	\$MTYP2 001123	\$VECT1 001136
SW9 = 001000	TST42 012440	\$ENDCT 014430	\$MTYP3 001127	\$VECT2 001140
TA 002610	TST43 012632	\$ENULL 014464	\$MTYP4 001133	\$XTSTR 015304
TBITVE= 000014	TST44 013010	\$ENV 001106	\$NULL 001054	\$GET4= 000000
TBUF 002642	TST45 013132	\$ENVM 001107	\$NWTST= 000001	\$OFILL 016467
TBUF58= 176506	TST46 013710	\$EOP 014400	\$OCNT 016466	.\$ERRT 001146
TCLOCK 005536	TST5 004714	\$EOPCT 014422	\$OMODE 016470	.\$X = 000500
TCONS 004020	TST6 004760	\$ERFLG 001003	\$OVER 015430	

. ABS. 024750 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 60528 WORDS (237 PAGES)

DYNAMIC MEMORY: 20434 WORDS (78 PAGES)

H 10

SEQ 0124

[illegible]

三

ELAPSED TIME: 00:05:26

C7DLDDO.CZDLDDO/NL:TOC/CRF/-SP=CZDLDDO.SML,CZDLDDO.P11

CZDLDDO CREATED BY MACRO ON 20-SEP-79 AT 10:40 PAGE 1
 SYMBOL CROSS REFERENCE CREF
 J 10

SYMBOL	VALUE	REFERENCES
ABASE	= 176500	#73-852 76-900 76-900
ACDW1	= 000000	76-900
ACDW2	= 000000	76-900
ACPUOP	= 000000	76-900 76-900
ADDW0	= 000000	76-900
ADDW1	= 000000	76-900
ADDW10	= 000000	76-900
ADDW11	= 000000	76-900
ADDW12	= 000000	76-900
ADDW13	= 000000	76-900
ADDW14	= 000000	76-900
ADDW15	= 000000	76-900
ADDW2	= 000000	76-900
ADDW3	= 000000	76-900
ADDW4	= 000000	76-900
ADDW5	= 000000	76-900
ADDW6	= 000000	76-900
ADDW7	= 000000	76-900
ADDW8	= 000000	76-900
ADDW9	= 000000	76-900
ADEVCT	= 000000	76-900 76-900
ADEVN	= 000000	76-900 76-900
AC...	004132	#77-1512 77-1515
ADRTBL	002702	#77-1358 77-1369 77-1510
AENV	= 000000	76-900 76-900
AENVN	= 000000	76-900 76-900
AFATAL	= 000000	76-900 76-900
AMADR1	= 000000	76-900 76-900
AMADR2	= 000000	76-900 76-900
AMADR3	= 000000	76-900 76-900
AMADR4	= 000000	76-900 76-900
AMAMS1	= 000000	76-900 76-900
AMAMS2	= 000000	76-900 76-900
AMAMS3	= 000000	76-900 76-900
AMAMS4	= 000000	76-900 76-900
AMSGAD	= 000000	76-900 76-900
AMSGLG	= 000000	76-900 76-900
AMSGTY	= 000000	76-900 76-900
AMTYP1	= 000000	76-900 76-900
AMTYP2	= 000000	76-900 76-900
AMTYP3	= 000000	76-900 76-900
AMTYP4	= 000000	76-900 76-900
APASS	= 000000	76-900 76-900
APRIOR	= 000000	76-900
APTCSU	= 000040	#117-3268 118-3272
APTENV	= 000001	113-3130 117-3223 #117-3266 118-3272
APTSIZ	000200	77-1394 #117-3265
APTSPO	000100	117-3225 #117-3267 118-3272
APTSZD	003636	77-1415 77-1417 #77-1443
ASWREG	000000	76-900 76-900
ATESTN	000000	76-900 76-900
AUNIT	000000	76-900 76-900

SEQ 0126

CZDLDDO		CREATED BY		MACRO ON 20-SEP-79 AT 10:40		PAGE 3		L 10		SEQ 0128	
SYMBOL CROSS REFERENCE		SYMBOL VALUE		REFERENCES		CREF					
BPTVEC	=	000014	#73-850								
BUF		002454	#77-1313	108-2773	109-2811						
CATCH		014552	73-875	#112-3095							
CHKSUM		014062	110-2903	110-2919	#110-2944						
CKSWR	=	104406	113-3140	116-3206	#120-3282						
CLK		013702	108-2762	#109-2855							
CLKCNT		002444	#77-1309	109-2805	*109-2855						
CMFARE		007244	93-2113	#93-2116							
CNTLP		002564	#77-1314	110-2906							
COMP		013504	#109-2813	109-2821							
CONT		006246	#88-1941	88-1944							
CONT41		012400	105-2614	#105-2622							
CR	=	000015	#73-850	77-1315	77-1316	111-3076	118-3272	118-3272	121-3333	121-3334	
			121-3335	121-3336	123-3467	123-3469					
CRBUF		002656	#77-1345	121-3297	121-3298	122-3369					
CRCR		002654	#77-1344	77-1434	77-1491	121-3295	122-3363				
CRLF	=	000200	#73-850	118-3272	118-3272						
CRPSW		002666	#77-1349								
CRVECT		002664	#77-1348								
CTBUF		002662	#77-1347	121-3294	121-3297	121-3328	122-3356				
CTCSR		002660	#77-1346	79-1615	79-1628	79-1638	96-2258	97-2278	97-2297	97-2307	97-2315
			97-2323	97-2328	98-2339	98-2364	99-2385	99-2389	99-2400	99-2405	100-2431
			100-2442	100-2446	101-2467	102-2509	104-2578	105-2619	105-2625	105-2630	106-2655
			107-2698	107-2702	121-3326	122-3354					
CTPSW		002672	#77-1351								
CTSTFL		002624	#77-1327	*77-1384	*77-1410	*77-1490	78-1535	78-1552	79-1567	79-1585	79-1613
			79-1626	82-1706	82-1716	82-1732	86-1834	86-1846	88-1924	89-1951	90-2002
			91-2039	92-2066	93-2096	104-2563	105-2598	106-2641	108-2758	109-2801	*111-3070
CTVECT		002670	#77-1350								
DDISP	=	177570	#73-850	76-900	77-1394						
DEVADR		003014	#77-1369	77-1419	77-1451	110-2868					
DH1		024012	77-904	77-914	77-919	77-924	77-944	77-949	77-954	77-959	77-964
			77-969	77-974	77-979	77-984	77-989	77-994	77-999	77-1004	77-1119
			77-1124	77-1129	77-1134	77-1139	77-1144	77-1149	#123-3455		
DH10		024136	77-939	77-1034	77-1039	77-1044	77-1049	77-1054	77-1064	77-1069	77-1074
			77-1079	77-1084	77-1089	77-1094	77-1099	77-1104	77-1109	77-1273	#123-3459
DH103		024263	77-1233	#123-3462							
DH105		024320	77-1243	77-1248	77-1263	#123-3463					
DH110		024364	77-1258	#123-3464							
DH112		024430	77-1268	#123-3465							
DH2		024037	77-909	#123-3456							
DH40		024162	77-1059	#123-3460							
DH53		024216	77-1114	#123-3461							
DH6		024064	77-929	77-1009	77-1014	77-1019	77-1024	77-1029	77-1154	77-1159	77-1164
			77-1169	77-1174	77-1179	77-1184	77-1189	77-1194	77-1199	77-1204	77-1208
			77-1213	77-1218	77-1224	77-1229	77-1238	77-1253	77-1278	77-1283	77-1288
			77-1293	#123-3457							
DH7		024111	77-934	#123-3458							
DISPLA		001042	#76-900	*77-1394	*77-1394	113-3120	116-3206				
DISPRE		000174	#73-886	77-1394							
DSWR	-	177570	#73-850	76-900	77-1394						
DT1		024572	77-905	77-915	77-920	77-925	77-945	77-950	77-955	77-960	77-965

CZDLDDO		CREATED BY		MACRO ON 20-SEP-79 AT 10:40		PAGE 4		M 10		SEQ 0129	
SYMBOL CROSS REFERENCE		SYMBOL VALUE		REFERENCES		CREF					
DT10	024632	77-970	77-975	77-980	77-985	77-990	77-995	77-1000	77-1005	77-1120	
		77-1125	77-1130	77-1135	77-1140	77-1145	77-1150	#123-3473			
		77-940	77-1035	77-1040	77-1045	77-1050	77-1055	77-1065	77-1070	77-1075	
		77-1080	77-1085	77-1090	77-1095	77-1100	77-1105	77-1110	77-1274	#123-3477	
DT103	024670	77-1234	#123-3480								
DT105	024702	77-1244	77-1249	77-1264	#123-3481						
DT110	024716	77-1259	#123-3482								
DT112	024732	77-1269	#123-3483								
DT2	024602	77-910	#123-3474								
DT40	024642	77-1060	#123-3478								
DT53	024654	77-1115	#123-3479								
DT6	024612	77-930	77-1010	77-1015	77-1020	77-1025	77-1030	77-1155	77-1160	77-1165	
		77-1170	77-1175	77-1180	77-1185	77-1190	77-1195	77-1200	77-1205	77-1209	
		77-1214	77-1219	77-1225	77-1230	77-1239	77-1254	77-1279	77-1284	77-1289	
		77-1294	#123-3475								
DT7	024622	77-935	#123-3476								
DVADT	003664	77-1449	#77-1451								
ECHO	017324	73-891	#121-3292	121-3321							
EMTVEC	= 000030	#73-850	77-1394	77-1394							
EM1	017630	77-903	#123-3375								
EM10	020074	77-938	#123-3382								
EM100	023107	77-1217	#123-3438								
EM101	023152	77-1223	#123-3439								
EM102	023200	77-1228	#123-3440								
EM103	023237	77-1232	#123-3441								
EM104	023275	77-1237	#123-3442								
EM105	023332	77-1242	#123-3443								
EM106	023355	77-1247	#123-3444								
EM107	023413	77-1252	#123-3445								
EM11	020117	77-943	#123-3383								
EM110	023445	77-1257	#123-3446								
EM111	023475	77-1262	#123-3447								
EM112	023535	77-1267	#123-3448								
EM113	023552	77-1272	#123-3449								
EM114	023610	77-1277	#123-3450								
EM115	023647	77-1282	#123-3451								
EM116	023706	77-1287	#123-3452								
EM117	023745	77-1292	#123-3453								
EM12	020162	77-948	#123-3384								
EM13	020213	77-953	#123-3385								
EM14	020246	77-958	#123-3386								
EM15	020307	77-963	#123-3387								
EM16	020352	77-968	#123-3388								
EM17	020403	77-973	#123-3389								
EM2	017654	77-908	#123-3376								
EM20	020436	77-978	#123-3390								
EM21	020477	77-983	#123-3391								
EM22	020542	77-988	#123-3392								
EM23	020577	77-993	#123-3393								
EM24	020630	77-998	#123-3394								
EM25	020663	77-1003	#123-3395								
EM26	020724	77-1008	#123-3396								

SYMBOL	VALUE	REFERENCES
EM27	020767	77-1013 #123-3397
EM3	017700	77-913 #123-3377
EM30	021026	77-1018 #123-3398
EM31	021057	77-1023 #123-3399
EM32	021112	77-1028 #123-3400
EM33	021160	77-1033 #123-3401
EM34	021222	77-1038 #123-3402
EM35	021260	77-1043 #123-3403
EM36	021310	77-1048 #123-3404
EM37	021342	77-1053 #123-3405
EM4	017745	77-918 #123-3378
EM40	021402	77-1058 #123-3406
EM41	021430	77-1063 #123-3407
EM42	021470	77-1068 #123-3408
EM43	021522	77-1073 #123-3409
EM44	021553	77-1078 #123-3410
EM45	021607	77-1083 #123-3411
EM46	021644	77-1088 #123-3413
EM47	021644	77-1093 #123-3412
EM5	017767	77-923 #123-3379
EM50	021700	77-1098 #123-3414
EM51	021725	77-1103 #123-3415
EM52	021755	77-1108 #123-3416
EM53	022032	77-1113 #123-3417
EM54	022056	77-1118 #123-3418
EM55	022214	77-1123 #123-3421
EM56	022114	77-1128 #123-3419
EM57	022152	77-1133 #123-3420
EM6	020024	77-928 #123-3380
EM60	022214	77-1138 #123-3422
EM61	022243	77-1143 #123-3423
EM62	022267	77-1148 #123-3424
EM63	022334	77-1153 #123-3425
EM64	022360	77-1158 #123-3426
EM65	022404	77-1163 #123-3427
EM66	022452	77-1168 #123-3428
EM67	022510	77-1173 #123-3429
EM7	020050	77-933 #123-3381
EM70	022553	77-1178 #123-3430
EM71	022620	77-1183 #123-3431
EM72	022767	77-1188 #123-3434
EM73	022662	77-1193 #123-3432
EM74	022720	77-1198 #123-3433
EM75	022767	77-1203 #123-3435
EM76	023016	77-1207 #123-3436
EM77	023042	77-1212 #123-3437
ENDADD	= 024746	#110-2869 110-2947 110-3039 110-3056
ENDADR	024746	110-2869 #123-3484
ENDB7	004676	79-1633 79-1650 #79-1652
ENDEV	014344	109-2834 #111-3065
ENDMG	014470	111-3075 #111-3076
ENDSTK	002412	#77-1303

CZDLDDO		CREATED BY MACRO ON 20-SEP-79 AT 10:40		PAGE 6		B 11				SEQ 0131	
SYMBOL CROSS REFERENCE		REFERENCES		CREF							
SYMBOL	VALUE										
ERROR	= 104000	#73-850	78-1537	78-1554	79-1569	79-1579	79-1587	79-1598	79-1616	79-1629	
		80-1668	80-1680	81-1692	82-1718	82-1727	82-1734	82-1744	84-1759	84-1764	
		84-1771	84-1777	84-1785	85-1799	85-1804	85-1811	85-1818	85-1826	86-1841	
		86-1854	86-1859	86-1866	86-1872	86-1879	87-1914	88-1929	88-1938	88-1946	
		89-1967	89-1980	89-1989	90-2016	90-2029	91-2059	92-2087	93-2129	94-2149	
		94-2158	95-2182	95-2195	96-2217	96-2231	96-2259	97-2280	97-2287	97-2298	
		97-2308	97-2316	97-2324	98-2342	98-2359	98-2365	99-2391	99-2401	100-2433	
		100-2448	101-2474	101-2491	102-2521	103-2552	104-2581	104-2586	105-2620	105-2627	
		106-2659	106-2664	107-2699	107-2728	107-2740	108-2787	108-2793	109-2808	109-2826	
		112-3099									
ERRVEC	- 000004	#73-850	77-1394	77-1394	77-1394	116-3206	116-3206	116-3206	116-3206		
FIRST	002336	#77-1298	*93-2114	93-2116	123-3479						
FLAG44	003002	#77-1360	*77-1387	77-1403	82-1704	93-2120	97-2267	97-2311	98-2347	104-2561	
		105-2596	106-2639	107-2680	107-2715	109-2816	109-2838				
GETB	014200	110-2915	#110-3000	110-3002	110-3006						
GETLIN	014134	110-2910	#110-2977	110-2987							
GNS	= *****	120-3282	120-3282	120-3282	120-3282	120-3282	120-3282	120-3282	120-3282	120-3282	
		120-3282	120-3282	120-3282	120-3282	120-3282	120-3282	120-3282			
GOAGIN	014504	111-3075	#112-3079								
GTSWR	= 104405	#120-3282									
HT	= 000011	#73-850	118-3272	118-3272							
ID	004620	77-1411	79-1622	79-1630	#79-1637						
INIT	003462	77-1409	#77-1412								
IOTVEC	= 000020	#73-850	77-1394	77-1394							
JIMSTK	002432	#77-1304	110-2896								
LF	- 000012	#73-850	77-1315	77-1315	77-1316	111-3076	118-3272	118-3272	121-3333	121-3334	
		121-3335	121-3336	123-3467	123-3469						
LKS	002674	#77-1354	86-1837	86-1852	86-1862	86-1863	86-1869	86-1870	86-1874	86-1876	
		87-1894	88-1926	88-1931	88-1932	88-1934	88-1935	88-1941	89-1958	89-1959	
		89-1960	89-1969	89-1973	89-1983	89-1984	89-1985	89-1992	90-2010	90-2011	
		90-2012	90-2032	91-2044	91-2045	91-2046	91-2055	92-2071	92-2072	92-2073	
		92-2075	92-2082	93-2097	93-2102	93-2103	93-2105	93-2106	93-2109	108-2764	
		123-3477	123-3479								
LOC1	002340	#77-1299	*110-2898	*110-3018							
LOC2	002342	#77-1300	*110-2899	*110-3020							
MANL	003502	77-1413	#77-1416								
MFPT	- 000007	#73-851	77-1400								
MFR	017507	121-3309	#121-3334								
MOR	017521	121-3314	#121-3335								
MPAR	017476	121-3304	#121-3333								
MSG	017450	121-3303	121-3308	121-3313	121-3318	#121-3324	122-3360				
MSTOP	017534	121-3319	#121-3336								
M1	024502	79-1646	#123-3467								
M2	024540	79-1648	#123-3469								
M2A	024542	*77-1482	#123-3470								
NOEOP	014370	111-3068	#111-3070								
OLDPC	002620	#77-1320	*112-3098	123-3483							
OLDSUM	002436	#77-1306	*110-2904	110-2920							
OUTTST	017544	73-892	#122-3352	122-3366	122-3371						
PIRQ	177772	#73-850									
PIRQVE	000240	#73-850									
PROMPT	002566	#77-1315	110-2909								

SYMBOL CROSS REFERENCE

SYMBOL VALUE REFERENCES

PRO	= 000000	#73-850								
PR1	= 000040	#73-850								
PR2	= 000100	#73-850								
PR3	= 000140	#73-850								
PR4	= 000200	#73-850								
PR5	= 000240	#73-850								
PR6	= 000300	#73-850								
PR7	= 000340	#73-850								
PS	= 177776	#73-850	73-850							
PSW	= 177776	#73-850	110-2894	*110-2895	*110-2924					
PUTLIN	014102	110-2907	110-2913	#110-2960	110-2966					
PWRVEC	= 000024	#73-850	77-1394	77-1394	115-3164	115-3165	115-3174	115-3182	115-3191	115-3192
RBUF	002636	#77-1334	81-1688	97-2271	98-2334	98-2338	98-2352	99-2377	100-2418	101-2463
		102-2505	102-2510	103-2536	104-2571	104-2579	104-2584	105-2606	105-2610	105-2622
		106-2649	106-2656	106-2661	107-2677	107-2689	107-2730	108-2766	109-2850	123-3476
RBUF 58	= 176502	#110-2865	110-2984	110-3003						
RCSR	002634	#77-1333	77-1492	77-1511	80-1676	85-1797	85-1807	85-1806	85-1814	85-1815
		85-1821	85-1823	87-1896	97-2271	97-2274	97-2284	97-2294	97-2303	97-2313
		97-2320	98-2334	98-2336	98-2340	98-2350	98-2352	98-2354	98-2356	98-2361
		98-2362	99-2375	99-2377	99-2383	99-2395	99-2404	100-2418	100-2422	100-2424
		100-2436	101-2463	101-2465	101-2468	101-2485	102-2503	102-2505	102-2507	102-2517
		103-2534	103-2536	103-2538	103-2547	104-2571	105-2606	105-2608	105-2613	106-2649
		106-2652	107-2677	107-2687	107-2723	108-2766	108-2768	123-3475	123-3480	123-3481
		123-3482	123-3483							
RCSR58	= 176500	#110-364	110-2982	110-3001						
RCV	013666	108-2756	#109-2850							
RCVCNT	002440	#77-1307	*108-2770	108-2789	*109-2851	123-3482				
RCVDON	010266	97-2268	97-2295	#97-2300						
RDCHR	= 104407	119-3277	#120-3282							
RDLIN	= 104410	#120-3282								
RESTTE	014310	110-2917	#110-3050							
RESVEC	= 000010	#73-850								
RPSW	002646	#77-1338	101-2457	101-2459	101-2487	108-2752	108-2757	109-2832		
RSTRT	014530	112-3081	#112-3085							
RTCPSW	002700	#77-1356	89-1955	89-1957	89-1994	90-2005	90-2007	90-2034	108-2753	108-2763
		109-2833								
RTCVT	002676	#77-1355	86-1848	86-1849	86-1881	89-1954	89-1956	89-1970	89-1982	89-1993
		90-2004	90-2006	90-2019	90-2033	91-2042	91-2043	91-2061	92-2069	92-2070
		92-2089	108-2750	108-2762	109-2830					
RVECT	002644	#77-1337	85-1793	85-1794	85-1828	99-2378	99-2379	99-2394	99-2407	100-2415
		100-2416	100-2437	100-2450	101-2456	101-2458	101-2478	101-2486	101-2493	102-2501
		102-2502	102-2523	103-2532	103-2533	103-2554	108-2749	108-2756	109-2829	
R6	=%000006	#73-850	*77-1394	*77-1394	77-1394					
R7	%000007	#73-850								
SAVEPS	002434	#77-1305	*110-2894	110-2924						
SAVETE	014254	110-2901	#110-3033							
SAVEO	002344	#77-1301	*110-3033	110-3050						
SAVLOC	002346	#77-1302	110-3035	110-3052						
SCOPE	= 000004	#73-850	78-1528	78-1545	79-1562	79-1602	80-1657	80-1673	81-1685	82-1701
		84-1751	85-1791	86-1833	86-1845	87-1885	88-1923	89-1950	90-2001	91-2038
		92-2065	93-2095	94-2138	95-2166	96-2201	96-2236	97-2266	98-2331	98-2346
		99-2373	100-2411	101-2454	102-2497	103-2528	104-2558	105-2593	106-2634	107-2670

SYMBOL	VALUE
SCPSW	002452
SECND	002616
SETADR	004100
SHIFT	003740
SIZE	003512
SRPSW	002450
STACK	= 001100
START	003046
STKLMT	= 177774
STPSW	002446
SWR	001040

107-2706	108-2744	110-2871	111-3075
#77-1312	*108-2753	109-2833	
#77-1319	*93-2128	123-3479	
77-1500	#77-1506		
#77-1472	77-1481		
#77-1419			
#77-1311	*108-2752	109-2832	
#73-850			
73-890	#77-1381		
#73-850			
#77-1310	*108-2751	109-2831	
#76-900	77-1394	*77-1394	77-1394
82-1702	82-1709	86-1834	86-1846
92-2066	93-2096	93-2131	104-2559
106-2644	107-2707	108-2760	109-2803
115-3184	116-3206	116-3206	119-3277
#73-887	77-1394	119-3277	119-3277

SWREG		000176
SW0	=	000001
SW00	=	000001
SW01	=	000002
SW02	=	000004
SW03	=	000010
SW04	=	000020
SW05	=	000040
SW06	=	000100
SW07	=	000200
SW08	=	000400
SW09	=	001000
SW1	=	000002
SW10	=	002000
SW11	=	004000
SW12	=	010000
SW13	=	020000
SW14	=	040000
SW15	=	100000
SW2	=	000004
SW3	=	000010
SW4	=	000020
SW5	=	000040
SW6	=	000100
SW7	=	000200
SW8	=	000400
SW9	=	001000
TA		002610
TBITVE	=	000014
TBUF		002642
TBUF 58		176506
TCLOCK		005536
TCONS		004020
TCSR		002640

[illegible]

SYMBOL CROSS REFERENCE
SYMBOL VALUE

REFERENCES

		79-1597	79-1603	79-1605	79-1610	79-1621	80-1658	80-1660	80-1665	82-1714
		82-1723	82-1724	82-1729	82-1730	82-1739	82-1741	82-1743	84-1757	84-1767
		84-1768	84-1774	84-1775	84-1780	84-1782	94-2139	94-2142	94-2152	94-2160
		95-2167	95-2172	95-2174	95-2184	96-2202	96-2209	96-2211	96-2227	96-2237
		96-2242	96-2243	96-2245	96-2254	97-2270	97-2290	97-2300	98-2333	98-2351
		99-2374	99-2376	100-2417	101-2462	102-2504	103-2535	104-2570	104-2574	105-2605
		105-2611	105-2618	105-2624	105-2629	106-2648	106-2650	106-2654	107-2673	108-2765
		108-2767	108-2783	109-2844	123-3473					
TCSR58	= 176504	#110-2866	110-2963							
TIMER	014234	110-2962	110-2981	110-3000	#110-3018					
TKVEC	= 000060	#73-850								
TMP1	002626	#77-1328	*77-1486	*77-1502	77-1508	*77-1509				
TMP2	002630	#77-1329	*77-1420	*77-1428	*77-1438	*77-1443	*77-1447	77-1467	111-3067	112-3080
TMP3	002632	#77-1330	*77-1485	77-1499	*77-1501	*77-1507				
TOLER	007256	93-2118	#93-2120							
TPSW	002652	#77-1340	77-1494	96-2204	96-2206	96-2234	108-2751	108-2755	109-2831	
TPVEC	= 000064	#73-850								
TRAPVE	= 000034	#73-850	77-1394	77-1394						
TRTVEC	= 000014	#73-850								
TSTDEV	004050	77-1489	#77-1499	77-1504	111-3071					
TSTDVM	003646	#77-1445	77-1450							
TST1	004172	77-1496	77-1522	#78-1528	112-3083					
TST10	005044	#82-1701								
TST11	005252	#84-1751								
TST12	005404	#85-1791								
TST13	005544	#86-1833								
TST14	005614	86-1834	86-1834	#86-1845						
TST15	005764	86-1846	86-1846	#87-1885						
TST16	006154	#88-1923								
TST17	006262	88-1924	88-1924	88-1942	#89-1950					
TST2	004244	#78-1545								
TST20	006504	89-1951	89-1951	#90-2001						
TST21	006660	90-2002	90-2002	#91-2038						
TST22	007002	91-2039	91-2039	#92-2065						
TST23	007136	92-2066	92-2066	#93-2095						
TST24	007324	93-2096	93-2096	93-2132	#94-2138					
TST25	007432	#95-2166								
TST26	007560	#96-2201								
TST27	007722	#96-2236								
TST3	004316	#79-1562								
TST30	010066	#97-2266								
TST31	010402	#98-2331								
TST32	010504	#98-2346								
TST33	010640	#99-2373								
TST34	011050	#100-2411								
TST35	011262	#101-2454								
TST36	011476	#102-2497								
TST37	011652	#103-2528								
TST4	004466	#79-1602								
TST40	012020	#104-2558								
TST41	012214	104-2560	104-2568	#105-2593						
TST42	012440	105-2595	105-2603	#106-2634						

CZDLDDO CREATED BY MACRO ON 20-SEP-79 AT 10:40 PAGE 11 G 11										
SYMBOL CROSS REFERENCE		SEQ 0136								
SYMBOL	VALUE	REFERENCES								
\$CNTLU	017212	119-3277	#119-3277							
\$CPUOP	001114	#76-900								
\$CRLF	001063	#76-900	113-3128	114-3156	114-3156	114-3156	114-3156	118-3272	118-3272	118-3272
		119-3277	119-3277	119-3277	122-3362					
\$DEVCT	001076	#76-900	*77-1385	79-1643	*111-3066	111-3067	*112-3079			
\$DEVN	001114	#76-900	*77-1421	*77-1427	*77-1430	*77-1437	77-1444	77-1487	77-1499	
\$DOAGN	014460	111-3075	111-3075	#111-3075						
\$ENDAD	014450	74-896	#111-3075	113-3148						
\$ENDCT	014430	77-1394	#111-3075							
\$ENULL	014464	#111-3075								
\$ENV	001106	#76-900	77-1412	113-3130	117-3223	117-3247	118-3272			
\$ENVN	001107	#76-900	77-1394	77-1414	117-3225	118-3272	118-3272			
\$FOP	014400	93-2133	110-2925	#111-3075						
\$EOPCT	014422	*77-1394	#111-3075	111-3075						
\$ERFLG	001003	#76-900	*113-3118	116-3206	116-3206	*116-3206	116-3206	116-3206		
\$ERMAX	001015	#76-900	*77-1394	*116-3206	116-3206	116-3206				
\$ERROR	014576	77-1394	#113-3117							
\$ERRPC	001016	#76-900	*113-3122	*113-3123	113-3124	114-3156	123-3473	123-3474	123-3475	123-3476
		123-3477	123-3478	123-3479	123-3480	123-3481	123-3482	123-3483		
\$ERRTB	001146	#77-900	114-3156							
\$ERRTY	014760	113-3127	#114-3156							
\$ERTTL	001012	#76-900	*113-3121							
\$ESCAP	001060	#76-900	*77-1394	113-3144	113-3146	*116-3206				
\$ETABL	001106	#76-900								
\$ETEND	001146	74-897	#76-900							
\$FATAL	001070	#76-900	*77-1381	*117-3251						
\$FFLG	015710	*117-3214	*117-3217	117-3245	*117-3254	#117-3262				
\$FILLC	001056	#76-900	118-3272	118-3272	118-3272					
\$FILLS	001055	#76-900	118-3272	118-3272						
\$GDADR	001020	#76-900	*87-1894	*87-1896	87-1897	87-1899	87-1904	123-3478		
\$GDDAT	001024	#76-900	*87-1902	87-1906	*107-2696	*107-2737	*109-2825	123-3481		
\$GET42	014440	#111-3075								
\$GTSWR	016542	#119-3277	120-3282	120-3282						
\$HIBTS	000500	#74-897								
\$ICNT	001004	#76-900								
\$ILLUP	015254	115-3164	115-3182	#115-3199						
\$INTAG	001035	#76-900	119-3277	119-3277	119-3277					
\$ITEMB	001014	#76-900	*113-3124	113-3132	114-3156					
\$LF	001064	#76-900	118-3272	118-3272	119-3277	119-3277	119-3277			
\$LFLG	015707	*117-3255	#117-3261							
\$LPADR	001006	#76-900	*77-1394	*116-3206	*116-3206	116-3206	116-3206	116-3206	116-3206	
\$LPERR	001010	#76-900	*77-1394	113-3143	116-3206	*116-3206	116-3206	116-3206	116-3206	
\$MADR1	001120	#76-900								
\$MADR2	001124	#76-900								
\$MADR3	001130	#76-900								
\$MADR4	001134	#76-900								
\$MAIL	001066	74-897	74-897	#76-900	77-1394	116-3206	118-3272			
\$MAMS1	001116	#76-900								
\$MAMS2	001122	#76-900								
\$MAMS3	001126	#76-900								
\$MAMS4	001132	#76-900								
\$MBADR	000502	#74-897								

CZDLDDO
SYMBOL CROSS REFERENCE

CREATED BY MACRO ON 20-SEP-79 AT 10:40

PAGE 13
CREF

I 11

SEQ 0138

SYMBOL	VALUE	REFERENCES
\$SWR	= 161000	#73-856 76-900 76-900 76-900 77-1394 77-1394 77-1394 77-1394 77-1394 78-1528 78-1545 79-1562 79-1602 80-1657 80-1673 81-1685 82-1701 84-1751 85-1791 86-1833 86-1845 87-1885 88-1923 89-1950 90-2001 91-2038 92-2065 93-2095 94-2138 95-2166 96-2201 96-2236 97-2266 98-2331 98-2346 99-2373 100-2411 101-2454 102-2497 103-2528 104-2558 105-2593 106-2634 107-2670 107-2706 108-2744 110-2871 111-3075 111-3075 111-3075 111-3075 111-3075 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206 116-3206
\$SWREG	001110	#76-900 77-1394
\$SWRMK	- 000000	116-3206
\$TESTN	001072	#76-900 *77-1383 *116-3206 123-3473 123-3474 123-3475 123-3476 123-3477 123-3478 123-3479 123-3480 123-3481 123-3482 123-3483
\$TKB	001046	#76-900 118-3272 118-3272 119-3277 119-3277 119-3277 119-3277 119-3277 119-3277
\$TKS	001044	#76-900 118-3272 118-3272 119-3277 119-3277 119-3277 119-3277 119-3277 119-3277
\$TN	000047	119-3277 #73-855 78-1528 78-1528 #78-1528 78-1545 78-1545 #78-1545 79-1562 79-1562 #79-1562 79-1602 79-1602 #79-1602 80-1657 80-1657 #80-1657 80-1673 80-1673 #80-1673 81-1685 81-1685 #81-1685 82-1701 82-1701 #82-1701 84-1751 84-1751 #84-1751 85-1791 85-1791 #85-1791 *86-1832 86-1833 86-1833 #86-1833 86-1834 86-1845 86-1845 #86-1845 86-1846 87-1885 87-1885 #87-1885 88-1923 88-1923 #88-1923 88-1924 89-1950 89-1950 #89-1950 89-1951 90-2001 90-2001 #90-2001 90-2002 91-2038 91-2038 #91-2038 91-2039 92-2065 92-2065 #92-2065 92-2066 93-2095 93-2095 #93-2095 93-2096 94-2138 94-2138 #94-2138 95-2166 95-2166 #95-2166 96-2201 96-2201 #96-2201 96-2236 96-2236 #96-2236 97-2266 97-2266 #97-2266 98-2331 98-2331 #98-2331 98-2346 98-2346 #98-2346 99-2373 99-2373 #99-2373 100-2411 100-2411 #100-2411 101-2454 101-2454 #101-2454 102-2497 102-2497 #102-2497 103-2528 103-2528 #103-2528 104-2558 104-2558 #104-2558 105-2593 105-2593 #105-2593 106-2634 106-2634 #106-2634 107-2670 107-2670 #107-2670 107-2706 107-2706 #107-2706 108-2744 108-2744 #108-2744 110-2871 110-2871 #110-2871
\$TPB	001052	#76-900 118-3272 118-3272 118-3272 118-3272
\$TPFLG	001057	#76-900 118-3272 118-3272 118-3272 118-3272
\$IPS	001050	#76-900 100-2419 118-3272 118-3272 118-3272
\$TRAP	017246	77-1394 #120-3282
\$TRAP2	017270	#120-3282 120-3282
\$TRP	= 000011	#120-3282 120-3282 120-3282 120-3282 120-3282 #120-3282 120-3282 120-3282 120-3282 #120-3282 120-3282 120-3282 120-3282 120-3282 #120-3282 120-3282 120-3282 120-3282 #120-3282 120-3282 120-3282 120-3282 #120-3282 120-3282 120-3282 120-3282 120-3282 120-3282 #120-3282 120-3282 120-3282 #120-3282
\$TRPAD	017302	120-3282 #120-3282
\$STSM	000504	#74-897
\$STNM	001002	#76-900 *86-1832 *111-3065 *111-3075 113-3120 116-3206 *116-3206 116-3206 116-3206 116-3206 116-3206 119-3277 119-3277 119-3277 #119-3277
\$TTYIN	017202	120-3282
\$TYPBN	= *****	120-3282
\$TYPDS	- *****	120-3282
\$TYPE	015712	117-3242 #118-3272 120-3282 120-3282
\$YPEC	016124	118-3272 118-3272 118-3272 #118-3272 118-3272 119-3277
\$YPEX	016242	118-3272 118-3272 #118-3272
\$YPOC	016270	#118-3273 120-3282 120-3282
\$YPON	016304	118-3273 #118-3273 120-3282
\$YPOS	016244	#118-3273 120-3282

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
SUNIT		001100	#76-900 *77-1386 *77-1503 *77-1506 *112-3085
SUNITM		000510	#74-897
SUSWR		001112	#76-900 77-1388 *77-1390 107-2690 107-2731
SVECT1		001136	#76-900 77-1456
SVECT2		001140	#76-900
SXTSTR		015304	#116-3206
SSGET4	=	000000	#111-3075 111-3075
SOFILL		016467	*118-3273 *118-3273 118-3273 #118-3273
S4OCAT	=	*****	116-3206
.SERRT		001146	#77-902
.SX		000500	#74-897 74-897

