

DMR11-DMP11

DMR11 FCTNL DIAG  
CZDMIAO

AH-F832A-MC  
FICHE 1 OF 1

MAY 1980  
COPYRIGHT © 1980  
MADE IN USA



The main body of the document is a large grid of approximately 20 columns and 20 rows of small, dense tables. Each cell in the grid contains a small table with multiple columns and rows of text, likely representing data points or diagnostic results. The text is very small and difficult to read, but the overall structure is a regular grid of data tables. The tables are arranged in a grid pattern, with each cell containing a small table of data. The data appears to be organized in a structured manner, possibly representing a diagnostic or control sequence. The overall appearance is that of a technical manual or a data log.

.NLIST  
.TITLE CZDMIAO DMR-11 FUNCTIONAL TESTS  
.SBTTL PROGRAM DOCUMENT  
.LIST  
.NLIST TOC

.REM @

IDENTIFICATION

PRODUCT CODE: AC-F830A-MC  
PRODUCT NAME: CZDMIAO DMR-11 FCTNL DIAG  
PRODUCT DATE: FEBRUARY 1980  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: MIKE O'CONNOR

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

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

COPYRIGHT (C) 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL  
DEC

PDP  
DECUS

UNIBUS  
DECTAPE

MASSBUS

## CONTENTS

- 1.0 INTRODUCTION
- 2.0 HARDWARE REQUIREMENTS
- 3.0 PRELIMINARY PROGRAM REQUIREMENTS
- 4.0 GENERAL PROGRAM CONSIDERATIONS
  - 4.1 DIAGNOSTIC SUPERVISOR
  - 4.2 EXECUTION TIME
  - 4.3 XXDP+
  - 4.4 ACT/SLIDE
  - 4.5 APT
  - 4.6 MEMORY MANAGEMENT
  - 4.7 MEMORY PARITY OPTION
  - 4.8 ERROR LOGGING
- 5.0 PROGRAM LOAD MEDIA
- 6.0 OPERATING INSTRUCTIONS
  - 6.1 LOADING AND STARTING PROCEDURES
    - 6.1.1 LOADING PROCEDURES
    - 6.1.2 STARTING PROCEDURES
    - 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION
  - 6.2 INITIAL DIALOGUE
  - 6.3 PROGRAM OPTIONS
    - 6.3.1 START COMMAND
      - 6.3.1.1 TESTS SWITCH
      - 6.3.1.2 PASS SWITCH
      - 6.3.1.3 FLAGS SWITCH
      - 6.3.1.4 END OF PASS SWITCH
      - 6.3.1.5 EFFECT OF START COMMAND
    - 6.3.2 RESTART COMMAND
      - 6.3.2.1 TESTS, PASS, AND FLAG SWITCHES
      - 6.3.2.2 UNITS SWITCH
      - 6.3.2.3 EFFECT OF RESTART COMMAND
    - 6.3.3 CONTINUE COMMAND
      - 6.3.3.1 PASS SWITCH
      - 6.3.3.2 FLAGS SWITCH
      - 6.3.3.3 EFFECT OF CONTINUE COMMAND
    - 6.3.4 PROCEED COMMAND
      - 6.3.4.1 FLAGS SWITCH
      - 6.3.4.2 EFFECT OF PROCEED COMMAND
    - 6.3.5 ADD COMMAND
      - 6.3.5.1 UNITS SWITCH
      - 6.3.5.2 EFFECT OF ADD COMMAND
    - 6.3.6 DROP COMMAND
      - 6.3.6.1 UNITS SWITCH
      - 6.3.6.2 EFFECT OF DROP COMMAND
    - 6.3.7 PRINT COMMAND
      - 6.3.7.1 EFFECT OF PRINT COMMAND
    - 6.3.8 DISPLAY COMMAND
      - 6.3.8.1 UNITS SWITCH
      - 6.3.8.2 EFFECT OF DISPLAY COMMAND

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

7.0 DEVICE INFORMATION TABLES

8.0 TEST DESCRIPTIONS

9.0 ERROR INFORMATION

- 9.1 ERROR REPORTING

## 1.0 INTRODUCTION

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

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

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

## 2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DMR-11 FUNCTIONAL DIAGNOSTIC TESTS:

PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70  
16K MEMORY  
CONSOLE TERMINAL  
DMR-11

## 3.0 PRELIMINARY PROGRAM REQUIREMENTS

IT IS ADVISED THAT THE STATIC DIAGNOSTICS BE RUN BEFORE THESE FUNCTIONAL DIAGNOSTICS. IT IS ASSUMED THAT THE PROCESSOR IS IN PROPER WORKING CONDITION.

ENSURE THAT THE SWITCH 1 AT LOCATION E-85 ON THE M8207 IS ON. IF THIS SWITCH IS OFF, THE MAINTENANCE BITS IN BSEL1 CAN'T BE USED AND CERTAIN TESTS WILL BE NOT BE CORRECTLY RUN.

WHEN CHOSING A CABLE TEST CONNECTION, ENSURE THAT THE SWITCH PACK E-39 ON THE M8203 IS PROPERLY SET UP FOR THE DESIRED INTERFACE. IF CHOSING TEST CONFIGURATION OPTIONS 1-4, IT IS NOT NECESSARY TO SELECT THE INTERFACE; HOWEVER THE BAUD RATE MUST BE CORRECT. FOR EXAMPLE IF IT IS DESIRED TO RUN CONFIGURATION 3 (H3255-EIA), IT IS NOT NECESSARY TO HAVE SWITCH 7 OF THE SWITCH PACK IN THE OFF POSITION. IT IS, HOWEVER, NECESSARY TO HAVE THE BAUD RATE SELETCTED TO BE WITHIN THE EIA RANGE.

#### 4.0 GENERAL PROGRAM CONSIDERATIONS

##### 4.1 DIAGNOSTIC SUPERVISOR

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

##### 4.2 EXECUTION TIME

EXECUTION TIME IS DEPENDENT ON THE PROCESSOR SPEED AND THE DMR BAUD RATE. EXAMPLES OF EXECUTION TIME

11/70 WITH CACHE AND DMR AT 2.4K	4 AND 1/2 MINUTES
11/70 WITHOUT CACHE AND DMR AT 2.4K	5 AND 1/2 MINUTES
11/34 AND DMR AT 2.4K	10 MINUTES

##### 4.3 XXDP+

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

##### 4.4 ACT/SLIDE

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

##### 4.5 APT

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

##### 4.6 MEMORY MANAGEMENT

IF MEMORY MANAGEMENT IS AVAILABLE, IT IS USED BY CERTAIN TESTS IN THIS FUNCTIONAL DIAGNOSTIC.

##### 4.7 MEMORY PARITY OPTION

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

##### 4.8 ERROR LOGGING

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

## 5.0 PROGRAM LOAD MEDIA

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

## 6.0 OPERATING INSTRUCTIONS

### 6.1 LOADING AND STARTING PROCEDURES

#### 6.1.1 LOADING PROCEDURES

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

#### 6.1.2 STARTING PROCEDURES

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

#### 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

## 6.2 INITIAL DIALOGUE

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

```
DRS LOADED  
DIAG. RUN-TIME SERVICES
```

```
DR>
```

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

## 6.3 PROGRAM OPTIONS

H 1

SEQ 0007

### 6.3.1 START COMMAND

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

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

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

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

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

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

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

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



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

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

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

#### 6.3.1.5 EFFECT OF START COMMAND

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

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

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

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

#### EXAMPLE:

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

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

### 6.3.2 RESTART COMMAND

```
*****
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
<FLAG-LIST>/UNITS:<UNIT-LIST>
*****
```

#### 6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

#### 6.3.2.3 EFFECT OF RESTART COMMAND

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

### 6.3.3 CONTINUE COMMAND

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

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

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

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

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

### 6.3.3.3 EFFECT OF CONTINUE COMMAND

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

### 6.3.4 PROCEED COMMAND

```
*****
PRO(CCEED)/FLAGS:<FLAG-LIST>
*****
```

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

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

#### 6.3.4.2 EFFECT OF PROCEED COMMAND

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

### 6.3.5 ADD COMMAND

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

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

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

#### 6.3.5.2 EFFECT OF ADD COMMAND

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

### 6.3.6 DROP COMMAND

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

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

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

6.3.6.2 EFFECT OF DROP COMMAND

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

6.3.7 PRINT COMMAND

```
*****
PRI(NT)
*****
```

6.3.7.1 EFFECT OF PRINT COMMAND

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

6.3.8 DISPLAY COMMAND

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

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

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

6.3.8.2 EFFECT OF DISPLAY COMMAND

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

6.3.9 FLAGS COMMAND

```
*****
FLA(GS)
*****
```

## 6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

## 6.3.10 ZFLAGS COMMAND

\*\*\*\*\*  
ZFL(AGS)  
\*\*\*\*\*

## 6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

## 6.3.11 CONTROL CHARACTERS

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

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

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

## 6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 3 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

## 1. CSR ADDRESS: (O) 160070?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT VALUE IS 160070.

## 2. VECTOR ADDRESS: (O) 300 ?

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

## 3. TEST CONFIGURATION -

- 0 = INTERNAL (NO CONNECTOR)
- 1 = H3254 - V.35 (NOTE: MODE 1-4 ALLOWS
- 2 = H3254 - INTEGRAL PROGRAM INTERFACE SELECTION)
- 3 = H3255 - RS232C/423
- 4 = H3255 - RS422
- 5 = CABLE AND SW PACK INTERFACE SELECTED  
(V.35-H3250, INTEGRAL-BC55A-10, RS232C-H325, RS423/422-H3251)

\* SELECT THE FOLLOWING ONLY IF THE MODEM SUPPORTS LOOPBACK \*

6 = LOCAL LOOP  
7 = REMOTE LOOP  
(0) 5 ?

THIS QUESTION WILL COVER ALL THE POSSIBLE TEST CONFIGURATIONS. THE DEFAULT IS FOR ACTUAL CABLE LOOPBACK (5). CONFIGURATION 0 WILL ENABLE LINE UNIT (TTL) LOOPBACK. IF THIS IS SELECTED NO CABLES OR CONNECTORS SHOULD BE CONNECTED. CONFIGURATIONS 1-4 WILL SELECT THE INTERFACE REGARDLESS OF THE SWITCH SETTING AS LONG AS THE PROPER BAUD RATE IS SELECTED (I.E. EIA - 2.4K-19.2K).

### 6.3.13 SOFTWARE PARAMETERS

THE ONLY SOFTWARE PARAMETER QUESTION ASKED BY THE DIAGNOSTIC CONCERNS A SOFTWARE TIMEOUT VARIABLE THAT IS USED TO PREVENT SOFTWARE 'HUNG' CONDITIONS. THIS VARIABLE IS A VALUE FROM 1-5.

SELECTABLE PROGRAM LOOP TIME-OUT VARIABLE  
[REFER TO LISTING 6.3.13] (MAX=5; MIN=1) (0) 5 ?

THERE ARE TWO FACTORS THAT SHOULD BE CONSIDERED WHEN ANSWERING THIS QUESTION. THE FIRST IS PROCESSOR SPEED; THE FASTER THE PROCESSOR THE HIGHER THE VARIABLE SHOULD BE. THE SECOND IS BAUD RATE; THE SLOWER THE DMR BAUD RATE THE HIGHER THE VARIABLE SHOULD BE. FOR EXAMPLE:

11/70 WITH CACHE AND DMR AT 1 MEG.: 4  
11/34 AND DMR AT 56K: 2  
11/40 AND DMR AT 2.4K: 3

THE DEFAULT IS 5. THIS WILL COVER THE WORST CASE (I.E. 11/70 WITH CACHE AND THE DMR AT 2.4K).

### 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

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

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

THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

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

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

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

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

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

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

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

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

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

## 7.0 DEVICE INFORMATION TABLES

SEE THE GLOBAL EQUATES SECTION FOR DEVICE CSR BIT DEFINITIONS

## 8.0 TEST DESCRIPTIONS

```

*****
:*          TEST 1 - DMR-11
:* VERIFY THAT ADDRESSING THE 4 UNIBUS CSRS DOES NOT CAUSE A NON-
:* EXISTENT MEMORY TRAP.
:*
:* THE DMR IS AN NPR DEVICE RESIDING ON A UNIBUS. COMMUNICATION
:* BETWEEN THE MAIN CPU AND THE DMR IS ACCOMPLISHED THROUGH A
:* SET OF FOUR 16-BIT UNIBUS CONTROL AND STATUS REGISTERS (CSRS).
:* THE FOUR REGISTERS ARE ASSIGNED ADDRESSES IN THE I/O PAGE
:* FLOATING ADDRESS SPACE: 76XXX0 - 76XXX6
:*
:* NOTE: THIS TEST IS REDUNDANT IN THAT STATIC LOGIC TESTS SHOULD
:* HAVE BEEN RUN BEFORE THESE FREE-RUNNING TESTS WERE STARTED, AND
:* THEY SHOULD HAVE DETECTED ANY CSR ADDRESSING PROBLEMS.
:* BUT JUST IN CASE THOSE STATIC TESTS AREN'T RUN, WE'LL BE SAFE.
*****

```

```

*****
:*          TEST 2 - DMR-11
:* ROM CRC/CCITT - CHECK ROM POSITION AND CALCULATE CRC/CCITT. THE
:* LAST 4 BYTES CONTAIN INFORMATION ABOUT THE ROM TO CHECK. THE 1ST
:* OF THESE BYTES CONTAINS THE ASCII VERSION NUMBER. THE 2ND BYTE
:* CONTAINS THE ROM NUMBER. THE 3RD AND 4TH BYTES CONTAIN A NEGATIVE
:* CRC/CCITT WORD FOR THE ROM.
:*
:*          CHIP ADDRESS RANGE
:* LOCATION  CHIP NO.   BYTE   ADDRESS RANGE
:* E03       0         LOW    0000 - 1777
:* E02       1         HIGH   0000 - 1777
:* E04       2         LOW    2000 - 3777
:* E01       3         HIGH   2000 - 3777
:* E05       4         LOW    4000 - 5777
:* E14       5         HIGH   4000 - 5777

```

```

***** IMPORTANT !!!!!!!!!!!!! *****
:* FOR THIS TEST TO RUN CORRECTLY, ENSURE THAT SWITCH 1 AT LOCATION
:* E85 ON THE M8207 IS ON. IF THIS SWITCH IS OFF, BSEL1 WILL BE
:* LOCKED OUT AND THE MAINTENANCE FEATURES WILL NOT BE ENABLED.
*****

```

```

:*          SUBTEST 1 - ON THE FIRST PASS PRINT THE VERSION # IN EACH ROM
:*          SUBTEST 2 - GENERATE THE CRC-CCITT IN EACH ROM AND COMPARE IT
:*                      IT AGAINST THE CRC BLASTED IN THE ROM
:*          SUBTEST 3 - COMPARE THE ROM # BLASTED IN THE ROM AGAINST THE
:*                      EXPECTED ROM #.
*****

```



```

*****
:*          TEST 3 - DMR-11
:* MASTER CLEAR
:* THIS TEST WILL ISSUE 2 MASTER CLEARS.  EACH CALL TO THE MASTER
:* CLEAR ROUTINE WILL ENSURE THAT THE RUN BIT WILL BE SET.  ALSO
:* THE MASTER CLEAR WILL CAUSE THE DIAGNOSTIC MICROTESTS TO BE
:* RUN WHEN THE MICRODIAGNOSTIC BIT (BIT 13 IN SEL0) IS CORRECTLY
:* SET OR CLEARED.  BECAUSE THE RUNNING OF MICROTESTS DEPENDS ON THE
:* EXCLUSIVE OR OF THE HARDWARE SWITCH 10 ON E134 OF THE M8203 AND
:* THE MICRODIAGNOSTIC BIT, WE CAN'T KNOW WHETHER THE SETTING OR
:* CLEARING OF BIT 13 WILL RESULT IN THE RUNNING OF MICROTESTS.
:* THEREFORE THE MASTER CLEAR SUBROUTINE WILL TOGGLE (I.E. SET
:* BIT 13 ONLY ON EVERY OTHER MASTER CLEAR) THE SOFTWARE BIT.
:* THIS WILL ENSURE THAT REGARDLESS OF THE POSITION OF THE
:* HARDWARE SWITCH, MICROTESTS WILL BE RUN EVERY OTHER MASTER CLEAR.
:* WHEN RUNNING THIS TEST, WE EXPECT TO ADD THE RESULTS OF BSEL3
:* AFTER EACH MASTER CLEAR.
:* BSEL3 = 100      - MICROTESTS DISABLED
:* BSEL3 = 200      - MICROTESTS RUN SUCCESSFULLY
:* IF THE RESULT OF THE 2 MASTER CLEARS IS NOT 300, AN ERROR IS
:* REPORTED.
:*
:* ADDITIONALLY THIS ROUTINE WILL REPORT WHENEVER THE RESULT OF
:* BSEL3 IS 0.  THIS WILL MEAN THAT THE DEVICE IS NOT A DMR
:* (I.E. DMC)
*****

```

```

*****
:*          TEST 4 - DMR-11
:* BASE IN COMMANDS
:*
:* SUBTEST 1 - ISSUE A BASE IN - DMR MODE.
:*             ENSURE THAT THE DMR MODE BIT (BIT 4) IS SET IN
:*             THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
:*             MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
:* SUBTEST 2 - ISSUE A BASE IN - DMC MODE.
:*             ENSURE THAT THE DMC MODE BIT (BIT 4) IS CLEAR IN
:*             THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
:*             MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
*****

```

```

*****
:*          TEST 5 - DMR-11
:* DMR COMMANDS
:* SUBTEST 1 - ISSUE AN ENABLE EXTENDED ERROR COMMAND AND CHECK THAT
:*             THE EXT. ENABLE BIT IS SET IS SCRATCH PAD 13.  THEN
:*             DISABLE EXTENDED ERROR AND CHECK THAT THE ENABLE BIT
:*             IS CLEAR.
:* SUBTEST 2 - SET REP/SEL TIMER VALUE AND SET THE DMR THRESHOLD
:*             VALUES.  CHECK THAT THE VALUES ARE CORRECT IN
:*             THE BASE TABLE AFTER HALTING THE DMR.
*****

```

```

*****
:*          TEST 6 - DMR-11
:* CONTROL IN COMMAND TEST -
:* SUBTEST 1 - CONTROL IN, FULL DUPLEX, DDCMP MODE. ENSURE THAT
:*              THE HALF-DUPLEX BIT IS CLEAR IN THE MODEM STATUS WORD,
:*              ALSO ENSURE THAT DDCMP MODE BIT IS SET IN SCRATCH PAD 7.
:* SUBTEST 2 - CONTROL IN, HALF DUPLEX. ENSURE THAT THE HALF DUPLEX
:*              BIT IS SET.
:* SUBTEST 3 - CONTROL IN, MAINTENANCE MODE. ENSURE THAT MAINT. MODE
:*              BIT IS SET IN SCRATCH PAD 7.
:* SUBTEST 4 - CONTROL IN USING SELECTED LOOPBACK. ISSUE A CONTROL IN
:*              USING THE USER SELECTED LOOPBACK. IF THE LOOPBACK IS
:*              NOT CORRECT, DMR RUN MODE ACKNOWLEDGE WILL NOT BE
:*              RECEIVED.

```

```

*****
:*          TEST 7 - DMR-11
:* MODEM WRITE COMMAND
:* SUBTEST 1 - WRITE DATA PATTERNS INTO THE MODEM WRITE REGISTER.
:*              ENSURE THAT ON THE NEXT MODEM READ THAT THE
:*              MICROCODE RETURNS THE PATTERN WRITTEN INTO BSEL6.
:* SUBTEST 2 - ATTEMPT TO WRITE BOTH THE HALF-DUPLEX BIT AND THE
:*              RTS HOLD BIT. THE MICROCODE SHOULD NOT ALLOW THIS
:*              TO HAPPEN. WHEN READING THE MODEM STATUS, ONLY
:*              THE HALF-DUPLEX SHOULD BE SET.

```

```

*****
:*          TEST 8 - DMR-11
:* SUBTEST 1 - TRANSMIT A BUFFER THREE TIMES WIHOUT ASSIGNING A
:*              RECEIVE BUFFER. BY ASSIGNING A NO BUFFER THRESHOLD
:*              OF THREE, ENSURE THAT A NO BUFFER ERROR IS RECEIVED
:*              AFTER THE THIRD THRANSMISSION.
:* SUBTEST 2 - TRANSMIT A BUFFER WITHOUT A RECEIVE BUFFER.
:*              ASSIGN THE NAKS THRESHOLD OF 3 AND A NO BUFFER
:*              THRESHOLD OF 7. CHECK THAT THE NAKS ERROR COUNT IS
:*              THREE AFTER SHUTDOWN.

```

```

*****
:*          TEST 9 - DMR-11
:* NON-EXISTENT MEMORY (NXM) error check
:* Perform DMR commands using NXM addresses; verify that NXM error is
:* reported IN EACH OF THE FOLLOWING SUBTESTS:
:* SUBTEST 1 - BASE IN RESUME COMMAND - BASE TABLE ADDRESS IS NXM
:* SUBTEST 2 - BA/CC IN RECEIVE COMMAND - BA/CC IN ADDRESS IS NXM
:* SUBTEST 3 - BA/CC IN TRANSMIT COMMAND - BA/CC IN ADDRESS IS NXM

```

```

*****
:*          TEST 10 - DMR-11
:* TIME OUT - FORCE A TIMEOUT AND VERIFY THAT THE ERROR IS REPORTED
:*
*****

```

```

*****
:*          TEST 11 - DMR-11
:* MESSAGE TOO LONG - TRANSMIT A MESSAGE THAT IS TOO LONG FOR THE
:* RECEIVE BUFFER AND VERIFY THAT THE 'TOO LONG' ERROR IS RECEIVED.
:*
*****

```

```

*****
:*          TEST 12 - DMR-11
:* PROCEDURE ERRORS -
:* THE FOLLOWING SHOULD CAUSE THE DMR-11 TO HALT AND RESPOND WITH
:* A PROCEDURE ERROR:
:* SUBTEST 1 - A SECOND BASE IN COMMAND
:* SUBTEST 2 - A CONTROL IN BEFORE A BASE IN
:* SUBTEST 3 - A BA/CC IN BEFORE A BASE IN
:* SUBTEST 4 - A BA/CC IN RCV WITH A BUFFER LENGTH OF 0
:* SUBTEST 5 - A BA/CC IN XMIT. WITH A BUFFER LENGTH OF 0
:*
*****

```

```

*****
:*          TEST 13 - DMR-11
:* FREE RUNNING FLAG MODE DATA TEST
:* TRANSMIT A MESSAGE AND VERIFY THE RECEIVED DATA IS CORRECT.
:* IN THIS TEST NO INTERRUPTS ARE USED AND THE LINE UNIT IS IN
:* INTERNAL (TTL) LOOPBACK. THIS TEST IS THE FIRST TEST IN WHICH
:* THE DMR IS USED IN A DATA TRANSMISSION MODE.
:*
*****

```

```

*****
:*          TEST 14 - DMR-11
:* IN THIS TEST - SEE IF WE HAVE MEMORY MANAGEMENT, IF SO SEE IF WE
:* HAVE THE MEMORY TO CHECK BITS 16 & 17 IN SEL6. THIS WILL ALLOW
:* US TO TRANSFER DATA USING THOSE EXTENDED ADDRESSING BITS. AS IN
:* TEST 13 THE TEST IS NON-INTERRUPT AND INTERNAL (TTL) LOOPBACK IS
:* USED.
:*
*****

```

```

*****
: * TEST 15 - DMR-11
: * RESUME BASE IN - DMC MODE
: * ** WILL NOT RUN IF MODEM LOOPBACK IS SELECTED **
: * IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
: * TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
: * FOLLOWING MANNER:
: *
: * BASE IN
: * CONTROL IN
: * HALT - BASE IN RESUME
: * 2 BA/CC IN RECEIVE
: * HALT - BASE IN RESUME
: * 2 BA/CC IN RECEIVE
: * HALT - BASE IN RESUME
: * 2 BA/CC IN RECEIVE
: * HALT - BASE IN RESUME
: * 1 BA/CC IN RECEIVE
: * HALT - BASE IN RESUME
: * 2 BA/CC IN TRANSMIT
: * HALT - BASE IN RESUME
: * 2 BA/CC IN TRANSMIT
: * HALT - BASE IN RESUME
: * 2 BA/CC IN TRANSMIT
: * HALT - BASE IN RESUME
: * 1 BA/CC IN TRANSMIT
: * HALT - BASE IN RESUME
: *
: * ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
: * THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
: * THE RECEIVE/TRANSMIT TABLE.
: *
: * THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
: * SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
: * SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL
: * ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
: * HIERARCHY:
: *
: * A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
: * B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
: * THAN 2K BYTES, USE THAT MEMORY
: * C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
: * THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
: *
*****

```

```

*****
: *          TEST 16 - DMR-11
: * RESUME BASE IN - DMR MODE
: * IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
: * TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
: * FOLLOWING MANNER:
: *   BASE IN
: *   CONTROL IN
: *   HALT - BASE IN RESUME
: *   2 BA/CC IN RECEIVE
: *   HALT - BASE IN RESUME
: *   2 BA/CC IN RECEIVE
: *   HALT - BASE IN RESUME
: *   2 BA/CC IN RECEIVE
: *   HALT - BASE IN RESUME
: *   1 BA/CC IN RECEIVE
: *   HALT - BASE IN RESUME
: *   2 BA/CC IN TRANSMIT
: *   HALT - BASE IN RESUME
: *   2 BA/CC IN TRANSMIT
: *   HALT - BASE IN RESUME
: *   2 BA/CC IN TRANSMIT
: *   HALT - BASE IN RESUME
: *   1 BA/CC IN TRANSMIT
: *   HALT - BASE IN RESUME
: *
: * ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
: * THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
: * THE RECEIVE/TRANSMIT TABLE.
: *
: * THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
: * SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
: * SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL
: * ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
: * HIERARCHY:
: *   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
: *   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
: *      THAN 2K BYTES, USE THAT MEMORY
: *   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
: *      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
: *
*****

```

```

*****
:*          TEST 17 - DMR-11
:* INTERRUPT DRIVEN EXERCISE
:* IN THIS TEST 64 BUFFERS WILL BE TRANSMITTED AND RECEIVED
:*
:* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* 64 RECEIVE AND 64 TRANSMIT BUFFERS. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*
*****

```

```

*****
:*          TEST 18 - DMR-11
:* LARGE MESSAGE
:* IN THIS MODE TRANSMIT AND RECEIVE 1 LARGE BUFFER
:*
:* THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* ONE RECEIVE AND ONE TRANSMIT BUFFER. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*
*****

```

```

*****
:*          TEST 19 - DMR-11
:* MAINTENANCE MODE OPERATION
:*
:* THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* ONE RECEIVE AND ONE TRANSMIT BUFFER. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*
*****

```

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

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

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES AN "IRDY NOT SET" ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE PC OF THE CALL TO THE SUBROUTINE REPORTING IT, THE FAILING REGISTER NAME, AND DEVICE REGISTER CONTENTS :

```

CZDMR DVC FTL ERR 00002 ON UNIT 00 TST 006 SUB 000 PC: 016210
TIME OUT
ERROR IN SUBROUTINE CALLED AT PC: 036174
STATUS OF BUFFERS
NUMBER OF BUFFERS: 7
BUFFER SIZE: 2048
IN - RCV ASSIGNED: 7      XMIT ASSIGNED: 7
OUT - RCV RETURNED: 0     XMIT RETURNED: 0
DMR RUN ACKNOWLEDGMENT NOT RECEIVED
(CHECK INTERFACE, BAUD AND TURNAROUND)

```

ALL THE MESSAGES IN THE DIAGNOSTIC USE BASIC MESSAGE CALLS. THEREFORE THE INHIBIT EXTENDED ERROR FLAG WILL HAVE NO EFFECT ON THE MESSAGE OUTPUT. THE INHIBIT BASIC MESSAGES WILL INHIBIT THE ERROR MESSAGES.

@

```
2194  
2203          002000          .TITLE CZDMIAO DMR-11 FUNCTIONAL TESTS  
2204  
2205  
2206  
2207  
2208          .MCALL SVC  
2209 002000          SVC          ; INITIALIZE SUPERVISOR MACROS  
2210  
2211  
2212 002000          BGNMOD  
2213  
2214  
2215          000001          $LSTIN= 1      ; LIST INSTRUCTIONS  
2216          000001          $LSTTAG= 1  
2217          000001          SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT  
2218          000001          SVCTST= 1     ; LIST TEST TAGS, SHIFTED RIGHT  
2219          000001          SVCSUB= 1     ; LIST SUBTEST TAGS, SHIFTED RIGHT  
2220          000001          SVCGBL= 1     ; LIST GLOBAL TAGS, SHIFTED RIGHT  
2221          000001          SVCTAG= 1     ; LIST OTHER TAGS, SHIFTED RIGHT  
2222  
2223          ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH  
2224          ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE  
2225          ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY  
2226          ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.  
2227  
2228 002000          POINTER BGNSW,BGNDU,BGNSFT  
2229  
2237  
2238  
2239
```



2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251

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

HEADER CZDMI,A,0,600.,0

002000  
 (4) 002000  
 (4) 002000 103  
 (4) 002001 132  
 (4) 002002 104  
 (4) 002003 115  
 (4) 002004 111  
 (6) 002005 000  
 (6) 002006 000  
 (5) 002007 000  
 (5) 002010  
 (4) 002010 101  
 (5) 002011  
 (4) 002011 060  
 (5) 002012  
 (4) 002012 000000  
 (5) 002014  
 (4) 002014 001130  
 (5) 002016  
 (4) 002016 036534  
 (5) 002020  
 (4) 002020 037470  
 (5) 002022  
 (4) 002022 002174  
 (5) 002024  
 (4) 002024 002224  
 (5) 002026  
 (4) 002026 037744  
 (5) 002030  
 (4) 002030 000000  
 (5) 002032  
 (4) 002032 000000  
 (5) 002034  
 (4) 002034 000000  
 (5) 002036  
 (4) 002036 000000  
 (5) 002040  
 (4) 002040 002124  
 (5) 002042  
 (4) 002042 000000  
 (5) 002044  
 (4) 002044 000000  
 (5) 002046  
 (4) 002046 000000  
 (5) 002050  
 (4) 002050 003

```
L$NAME::
      .ASCII /C/
      .ASCII /Z/
      .ASCII /D/
      .ASCII /M/
      .ASCII /I/
      .BYTE 0
      .BYTE 0
      .BYTE 0
L$REV::
      .ASCII /A/
L$DEPO::
      .ASCII /0/
L$UNIT::
      .WORD 0
L$TIML::
      .WORD 600.
L$HPCP::
      .WORD L$SHARD
L$SPCP::
      .WORD L$SOFT
L$HPTP::
      .WORD L$HW
L$SPTP::
      .WORD L$SW
L$LADP::
      .WORD L$LAST
L$STA::
      .WORD 0
L$CO::
      .WORD 0
L$DTYP::
      .WORD 0
L$APT::
      .WORD 0
L$DTP::
      .WORD 0
L$DISPATCH::
      .WORD L$DISPATCH
L$PRIO::
      .WORD 0
L$ENVI::
      .WORD 0
L$EXP1::
      .WORD 0
L$MREV::
      .WORD 0
      .BYTE C$REVISION
```

(3) 002051 003  
 (5) 002052  
 (4) 002052 000000  
 (5) 002054 000000  
 (5) 002056  
 (4) 002056 000000  
 (5) 002060  
 (4) 002060 010230  
 (5) 002062  
 (4) 002062 000000  
 (5) 002064  
 (4) 002064 000000  
 (5) 002066  
 (4) 002066 000000  
 (5) 002070  
 (4) 002070 000000  
 (5) 002072  
 (4) 002072 023522  
 (5) 002074  
 (4) 002074 000000  
 (5) 002076  
 (4) 002076 010236  
 (5) 002100  
 (4) 002100 104035  
 (5) 002102  
 (4) 002102 000000  
 (5) 002104  
 (4) 002104 020514  
 (5) 002106  
 (4) 002106 021770  
 (5) 002110  
 (4) 002110 021700  
 (5) 002112  
 (4) 002112 020506  
 (5) 002114  
 (4) 002114 000000  
 (5) 002116  
 (4) 002116 000000  
 (5) 002120  
 (4) 002120 000000

LSEF:: .BYTE C\$EDIT  
 .WORD 0  
 .WORD 0  
 L\$SPC:: .WORD 0  
 L\$DEVP:: .WORD L\$DVTYP  
 L\$REPP:: .WORD 0  
 L\$EXP4:: .WORD 0  
 L\$EXP5:: .WORD 0  
 L\$AUT:: .WORD 0  
 L\$DUT:: .WORD 0  
 L\$LUN:: .WORD L\$DU  
 L\$DESP:: .WORD 0  
 L\$LOAD:: .WORD L\$DESC  
 EMT ESLOAD  
 L\$ETP:: .WORD 0  
 L\$ICP:: .WORD L\$INIT  
 L\$CCP:: .WORD L\$CLEAN  
 L\$ACP:: .WORD L\$AUTO  
 L\$PRT:: .WORD L\$PROT  
 L\$TEST:: .WORD 0  
 L\$DLY:: .WORD 0  
 L\$HIME:: .WORD 0

2252  
 2258  
 2259  
 2260  
 2261  
 2262  
 2263  
 2264  
 2265  
 2266  
 2267  
 2268

.EVEN



2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
(3)  
(3)  
(3)  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
(3)  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
(3)  
(3)  
(3)  
2326  
2327  
2328  
2329  
(3)  
2330  
2331  
2332  
2333  
2334

.SBTTL DEFAULT HARDWARE P-TABLE

:/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF  
:/ THE TEST-DEVICE PARAMETERS. \*\*NOTE - MANY OF THE P-TABLE VALUES LISTED  
:/ BELOW ARE NOT USED IN THIS DIAGNOSTIC BUT ARE INCLUDED TO AGREE WITH  
:/ M8207 & M8203 DIAGNOSTIC P-TABLES.

BGNHW DFPTBL

.WORD L10000-L\$HW/2

L\$HW::  
DFPTBL::

.WORD 0  
.WORD 160070  
.WORD 300  
.WORD 0  
.WORD 0  
.WORD 000  
.WORD 000  
.WORD 000  
.WORD 5  
.WORD 0  
.WORD 0

\*\*NOT USED - MICROPROCESSOR TYPE  
:DMR11 CSR UNIBUS ADDRESS DEFAULT  
:DMR11 INTERRUPT VECTOR DEFAULT  
\*\*NOT USED - PRIORITY LEVEL  
\*\*NOT USED - LINE UNIT  
\*\*NOT USED - SWITCH PACK #1 (REG 11)  
\*\*NOT USED - SWITCH PACK #2 (REG 15)  
\*\*NOT USED - SWITCH PACK #3 (REG 16)  
:CABLE TURNAROUND (DEFAULT = CABLE(5))  
\*\*NOT USED - BAUD RATE  
\*\*NOT USED - RUN SWITCH

ENDHW

L10000:

.SBTTL DEFAULT SOFTWARE P-TABLE

:/ THE SOFTWARE P-TABLE CONTAINS THE VALUE OF THE PROGRAM  
:/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.

BGNSW SFPTBL

.WORD L10001-L\$SW/2

L\$SW::  
SFPTBL::

SPEED: .WORD 5  
ENDSW

:PROCESSOR SPEED VARIABLE USED  
:TO ALTER THE WAIT VARIABLES.

L10001:



```

(1)      000100      PRI02== 100
(1)      000040      PRI01== 40
(1)      000000      PRI00== 0
(1)
(1)      ; OPERATOR FLAG BITS
(1)
(1)      000004      EVL==      4
(1)      000010      LOT==      10
(1)      000020      ADR==      20
(1)      000040      IDU==      40
(1)      000100      ISR==     100
(1)      000200      UAM==     200
(1)      000400      BOE==     400
(1)      001000      PNT==    1000
(1)      002000      PRI==    2000
(1)      004000      IXE==    4000
(1)      010000      IBE==   10000
(1)      020000      IER==   20000
(1)      040000      LOE==   40000
(1)      100000      HOE==  100000
2344      ;*****
2345
2346      ;*****
2347      ; SWITCH REGISTER OPTIONS
2348
2349      100000      SW15=   100000
2350      040000      SW14=   40000
2351      020000      SW13=   20000
2352      010000      SW12=   10000
2353      004000      SW11=   4000
2354      002000      SW10=   2000
2355      001000      SW09=   1000
2356      000400      SW08=   400
2357      000200      SW07=   200
2358      000100      SW06=   100
2359      000040      SW05=   40
2360      000020      SW04=   20
2361      000010      SW03=   10
2362      000004      SW02=   4
2363      000002      SW01=   2
2364      000001      SW00=   1
2365
2366      ;*****
2367      ; CSR AND STAU8 WORD DEFINITIONS
2368      ; SELO (CSR) - BSELO/BSEL1
2369      100000      RUN=    BIT15      ; SET IF RUNNING
2370      040000      MCLR=   BIT14      ; MASTER CLEAR OF PROCESSOR AND LINE UNIT
2371      020000      MDIAG=  BIT13      ; CSR MAINTENANCE - ENABLE MICRODIAGNOSTICS
2372      010000      STLU=   BIT12      ; CSR MAINTENANCE - STEP LINE UNIT
2373      004000      LPLU=   BIT11      ; CSR MAINTENANCE - LINE UNIT LOOP
2374      002000      ROMO=   BIT10      ; CSR MAINTENANCE
2375      001000      ROMI=   BIT9       ; CSR MAINTENANCE
2376      000400      STUP=   BIT8       ; CSR MAINTENANCE - USED WITH LOOP LU
2377      ; WHEN ASSERTED, XMITTER SHIFTS; CLEAR, REC. SHIFTS
2378      000200      RDI=    BIT7       ; CSR - DMR11 READY RESPONSE
2379      000100      IESET=  BIT6       ; CSR - INTERRUPT ENABLE INPUT - DMR11 INTERRUPTS
  
```

```

2380                                     :CPU WHEN RDI SET IN RESPONSE TO RQI BEING SET.
2381      000040      RQI=   BIT5      :CSR - REQUEST IN
2382      000020      IECLR= BIT4      :CSR - INTERRUPT ENABLE INPUT - DMR11 INTERRUPTS
2383                                     :CPU WHEN RDI CLEARS IN RESPONSE TO RQI BEING CLEAR.
2384                                     : (DMR RUN MODE ONLY)
2385      000004      RCV=   BIT2      :CSR - IF 0, TRANSMIT & IF 1, RECEIVE
2386
2387      ;;SEL2 - BSEL2/BSEL3
2388      000200      RDO=   BIT7      :SEL2 - DMR11 SETS TO INDICATE DATA READY FOR OUTPUT
2389      000100      IEO=   BIT6      :SEL2 - SET TO ENABLE DMR11 TO INTERRUPT WHEN RDO
2390
2391      ;;SEL6 - BSEL6/BSEL7
2392      020000      BASEUP= BIT13     :SEL6 - CONTROL OUT - RESPONSE TO DMR MODE BASE
2393                                     : TABLE UPDATE COMMAND.
2394      010000      RES=   BIT12     :SEL6 - BASE IN -- WHEN SET CAUSES
2395                                     : RESUMPTION OF OPERATION
2396      010000      CTS=   BIT12     :SEL6 - CONTROL OUT - CTS FAILED
2397      004000      SECN=  BIT11     :SEL6 - CONTROL IN -- START TIME (3 SEC IF SET
2398                                     : 1 SEC IF CLEAR)
2399      002000      HDX=   BIT10     :SEL6 - HALF-DUPLEX & CLEAR FOR FULL-DUPLEX
2400      002000      CD=    BIT10     :SEL6 - CONTROL OUT - CD GLITCHED
2401      001000      HALTC= BIT9      :SEL6 - EXTENDED CONTROL OUT - HALT COMPLETED
2402      000400      MAINT= BIT8      :SEL6 - DDCMP MAINTENANCE DURING CONTROL IN
2403      000522      DMR=   BIT8!122 :SEL6 - BASE IN -- SET FOR DMR11 MODE
2404                                     : 122 IS THE DMR PASSWORD FOR BSEL6 AND
2405                                     : BIT8 SETS THE DMR MODE BIT IN BSEL7
2406      000400      NXM=   BIT8      :SEL6 - CONTROL OUT - NON EXISTENT MEMORY
2407      000200      STREC= BIT7      :SEL6 - CONTROL OUT - START RECEIVED
2408      000100      DISCON= BIT6     :SEL6 - CONTROL OUT - DISCONNECT
2409      000100      DTR=   BIT6     :SEL6 - MODEM WRITE - DATA TERMINAL READY
2410      000040      DMRRUN= BIT5    :SEL6 - CONTROL OUT - DMR RUN MODE
2411      000020      TOLONG= BIT4    :SEL6 - CONTROL OUT - MESSAGE TOO LONG
2412      000010      MAINT1= BIT3    :SEL6 - MODEM WRITE - LOCAL MODEM LOOPBACK
2413      000010      MNTREC= BIT3    :SEL6 - CONTROL OUT - MAINTENANCE MSG. RECEIVED
2414      000004      NOBFR= BIT2     :SEL6 - CONTROL OUT - NO BUFFER
2415      000004      MAINT2= BIT2    :SEL6 - MODEM WRITE - REMOTE MODEM LOOPBACK
2416      000002      TOUT=  BIT1     :SEL6 - CONTROL OUT - TIME OUT
2417      000001      NAKS=  BIT0     :SEL6 - CONTROL OUT - NAKS THRESHOLD EXCEEDED
2418
2419
2420      ;;*****
2421      ;;DDCMP COMMANDS - BITS 0 & 1 IN SEL0 AND SEL2
2422
2423      ;INPUT (SEL0)
2424      000000      BACCT= 0          :BUF ADDRESS AND CHARACTER COUNT TRANSMIT
2425      000001      CNTRL= 1         :CONTROL COMMAND (IN OR OUT)
2426      000002      HLT=   2         :HALT COMMAND
2427      000003      BASEI= 3         :BASE IN COMMAND
2428      000004      BACCR= 4         :BUF ADDRESS AND CHARACTER COUNT RECEIVE
2429      000005      WMODEM= 5        :WRITE MODEM STATUS REGISTER
2430      000006      EXERR= 6         :ENABLE EXTENDED ERROR NOTIFICATION
2431      000007      DXERR= 7         :DISABLE EXTENDED ERROR NOTIFICATION
2432      000010      DDMC=  10        :DESELECT DMC LINE MODE
2433      000011      UPDATE= 11        :REQUEST BASE TABLE UPDATE
2434      000012      TIMER=  12        :SET REP/SELECT TIMER VALUE
2435      000013      THRESH= 13       :SET THE FOLLOWING THRESHOLDS:
    
```

```
2436                                     :NAKS RECVD
2437                                     :NAKS SENT
2438                                     :REP/SEL
2439                                     :NO BUFFER
2440 000014 RRAM= 14                       :READ M8207 RAM (0-377)
2441 000015 INTER= 15                      :WRITE INTERFACE IN AX3-15
2442 000017 RMODEM= 17                     :READ MODEM STATUS (=NOP)
2443
2444                                     :OUTPUT (SEL2)      NOTE: CNTRL IS USED FOR SEL2
2445 000007 CMD= 7                          ; ** MASK USED TO CLEAR COMMAND BITS 0-2 **
2446
2447 ::*****
2448 :::BASE TABLE OFFSETS
2449                                     :NOTE: THE OFFSETS FOR BASE+3.-BASE+10 WERE
2450                                     :INTENTIONALLY NOT LABELLED, BECAUSE THOSE LOCATIONS
2451                                     :MUST NOT BE CHANGED IN ORDER TO BE DMC COMPATIBLE.
2452                                     :THE LABELS BELOW CORRESPOND WITH THOSE USED IN THE
2453                                     :DMR MICROCODE.
2454 000042 R= 42
2455 000043 N= 43
2456 000044 A= 44
2457 000045 T= 45
2458 000046 X= 46
2459 000055 PRETIM= 55
2460 000060 TH1L= 60
2461 000062 TH2L= 62
2462 000064 TH3L= 64
2463 000066 TH4L= 66
2464 000072 ISP7= 72
2465 000076 ISP13= 76
2466                                     :#R - MESSAGE RECEIVED
2467                                     :#N - MESSAGE TRANSMITTED
2468                                     :#A - MESSAGE ACKNOWLEDGED
2469                                     :#T - NEXT MESSAGE TO BE TRANSMITTED
2470                                     :#X - LAST COMPLETED TRANSMISSION
2471                                     :PROGRAMMABLE REP/SEL TIMER VALUE.
2472                                     :THRESHOLD LEVEL - NAKS RECEIVED .
2473                                     :THRESHOLD LEVEL - NAKS SENT.
2474                                     :THRESHOLD LEVEL - REP SENT.
2475                                     :THRESHOLD LEVEL - NO BUFFER AVAILABLE.
2476                                     :IMAGE OF SCRATCH PAD 7
2477                                     :IMAGE OF SCRATCH PAD 13
2478 ::*****
:::INSTRUCTION DEFINITIONS
2470 000207 RETURN=207                       ;RETURN FROM SUB.      [= JSR PC]
2473 ::*****
2474 ::: MISC. EQUATES
2475 000015 CR= 15                          ;ASCII CARRIAGE RETURN
2476 000012 LF= 12                          ;ASCII LINE FEED
```



2480  
2481  
2482  
2483  
2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511  
2512  
2513  
2514  
2515  
2516  
2517  
2518  
2519  
2520  
2521  
2522  
2523  
2524  
2525  
2526  
2527  
2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535

.SBTTL GLOBAL DATA SECTION

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

::\*\*\*\*\*  
:DMR11 VECTOR AND REGISTER INDIRECT POINTERS

DMRVEC: .WORD 0 ;DMR11 RECEIVER INTERRUPT VECTOR  
DMTVEC: .WORD 0 ;DMR11 TRANSMITTER INT. VECTOR  
CSR: .WORD 0 ;POINTER TO DMR11 CONTROL STATUS REGISTER  
SEL2: .WORD 0 ;POINTER TO DMR11 CONTROL OUT REGISTER (SEL 2)  
SEL4: .WORD 0 ;POINTER TO DMR11 PORT REGISTER (SEL 4)  
SEL6: .WORD 0 ;POINTER TO DMR11 PORT REGISTER (SEL 6)  
SEL0= CSR ;CSR IS SEL0  
BSEL0= CSR ;LOW BYTE OF CSR  
BSEL1: .WORD 0 ;POINTER TO DMR11 CSR HIGH BYTE  
BSEL2= SEL2 ;LOW BYTE OF SEL2  
BSEL3: .WORD 0 ;POINTER TO SEL2 HIGH BYTE  
BSEL4= SEL4 ;LOW BYTE OF SEL4  
BSEL5: .WORD 0 ;POINTER TO SEL4 HIGH BYTE  
BSEL6= SEL6 ;LOW BYTE OF SEL6  
BSEL7: .WORD 0 ;POINTER TO SEL6 HIGH BYTE

::\*\*\*\*\*  
:OTHER HARDWARE PARAMETERS

WTYPE: .WORD 0 ;MICROPROCESSOR TYPE  
DMTURN: .WORD 0 ;TURN AROUND TYPE (0-7)  
MICRO: .WORD 0 ;MICRODIAGNOSTICS (IF 1(YES) - ENABLED)

::\*\*\*\*\*  
:PROGRAM CONTROL PARAMETERS

DMRFLG: .WORD 0 ;FLAG SET WHEN DMR MODE IS REQUESTED IN  
;THE BASE IN COMMAND. USED TO FLAG THAT  
;A DMR MODE ACKNOWLEDGE IS EXPECTED.  
INFACE: .WORD 0 ;FLAG TO ALLOW CHANGE OF INTERFACE TYPE  
;BY WRITING AX3-15. FLAG SET/CLEARED IN INIT.  
FRSTIM: .WORD 0 ;FLAG=0 IF PROGRAM JUST LOADED  
FRSPAS: .WORD 0 ;FLAG=0 IF FIRST PASS AFTER LOAD  
STARES: .WORD 0 ;FLAG=0 IF 1ST TIME THRU AFTER STA OR RES

:FOLLOWING PARAMETERS ARE USED IN THE  
:INTERRUPT TESTS (TESTS 15-19):  
START: .WORD 0 ;FLAG SET WHEN A CONTROL IN HAS BEEN ISSUED.  
RESUME: .WORD 0 ;FLAG SET WHEN A BASE IN WITH RESUME DESIRED.  
DMCMDE: .WORD 0 ;FLAG SET WHEN A BASE IN WITH DMC MODE DESIRED  
MNTMDE: .WORD 0 ;FLAG SET WHEN MAINTENANCE MODE IS DESIRED.  
MMANAG: .WORD 0 ;FLAG RETURNED IN THE SUBROUTINE \$BUFFS



2592	002360	000000	ERROR:	.WORD	0	:ERROR STORAGE
2593	002362	000000	LOGDEV:	.WORD	0	:LOGICAL DEVICE NUMBER
2594	002364	000000	PSTACK:	.WORD	0	:CONTAINS BASE LEVEL PROGRAM SP
2595	002366	000000	SUBRPC:	.WORD	0	:PC OF SUBR CALL FOR ERROR REPORTS
2596	002370	000000	NESTPC:	.WORD	0	:FLAG TO NOTIFY WHEN A SUBR IS NESTED
2597						:IN ANOTHER SUBROUTINE (WHEN SET)
2598	002372	000000	CLRNO:	.WORD	0	:THIS WORD IS INCREMENTED DURING EACH MASTER
2599						:CLEAR. THIS WILL ALLOW EVERY OTHER MASTER
2600						:CLEAR TO RUN THE MICRO TESTS.
2601						
2602						:ROM CHECK VARIABLES
2603	002374	000000	LOCRC:	.WORD	0	:CRC STORAGE FOR LOW BYTE CHIP
2604	002376	000000	HICRC:	.WORD	0	:CRC STORAGE FOR HIGH BYTE CHIP
2605	002400	000000	LOWORD:	.WORD	0	:TEMP. WORD CONTAINING 2 CONSECUTIVE LOW BYTES
2606	002402	000000	HIWORD:	.WORD	0	:TEMP. WORD CONTAINING 2 CONSECUTIVE HI BYTES
2607	002404	000000	ROMADR:	.WORD	0	:POINTER TO ROM ADDRESS.
2608	002406	000000	CHIPNO:	.WORD	0	:CHIP NUMBER BEING CHECKED.
2609						
2610						
2611						
2612						
2613						
2614						
2615						
2616						
2617						
2618						
2619						
2620						
2621						
2622						
2623	002410					
2624	002410	177603				
2625	002416	047321				
2626	002424	143325				
2627	002432	014116				
2628	002440	105025				
2629	002446	111523				
2630	002454	137642				
2631	002462	135015				
2632	002470	052012				
2633	002476	151172				
2634	002504	166632				
2635						
2636						
2637						
2638						
2639	002510	000000	TFLAG:	.WORD	0	:FLAG FOR STATUS OF TRANSMIT BUFFER
2640		000044	Tcount=	36.		:CHARACTER COUNT OF TBUF
2641	002512	041101	TBUF:	.ASCIZ	/ABCDEFGHIJKLMN	
	002520	044107			OPQRSTUVWXYZ0123456789/	
	002526	047115				
	002534	052123				
	002542	055131				
	002550	032464				
	002556	000				

::\*\*\*\*\*  
::\*\*\*\*\*  
::BUFFER AREA

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

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

::\*\*\*\*\*  
:: TRANSMIT BUFFER (SMALL)

```
2642          002560          .EVEN
2643
2644
2645          ::*****
2646          :: RECEIVE BUFFER (SMALL)
2647
2648 002560 000000  RFLAG: .WORD 0          ;FLAG FOR STATUS OF RECEIVE BUFFER
2649          000044  RCOUNT= 36.          ;CHARACTER COUNT OF RBUF
2650 002562 000046  RBUF: .BLKB 38.          ;36. BYTE BUFFER + 2 BYTES USED
2651          ;TO MARK THE END OF THE RECEIVE BUFFER
2652          .EVEN
2653
2654          ::*****
2655          :: BASE TABLE
2656
2657 002630 000400  BASE: .BLKB 256.          ;MICROPROCESSOR MEMORY ALLOCATION
2658
2659          ::*****
2660          :: TRANSMIT AND RECEIVE BUFFER POINTERS
2661
2662 003230 000200  XMTBUF: .BLKW 128.          ;POINTERS TO TRANSMIT BUFFERS (UP TO 64)
2663          ;1 WORD FOR ADDRESS AND 1 WORD FOR CHAR. COUNT
2664 003630 000200  RCVBUF: .BLKW 128.          ;POINTERS TO RECEIVE BUFFERS (UP TO 64).
2665
2666          ::*****
2667          :: BUFFER AREA (LARGE)
2668
2669 004230 004000  BIGBUF: .BLKB 4000          ;MAX BUFFER (2K BYTES)
2670
2671
2672
2673
```

2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699  
2706  
2707  
2708  
2709  
2710

.SBTTL GLOBAL TEXT SECTION

:XXX  
: THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,  
: MESSAGES, AND ASCII INFORMATION THAT ARE USED IN  
: MORE THAN ONE TEST.  
:XXX

:\*\*\*\*\*  
:\* NAMES OF DEVICES SUPPORTED BY PROGRAM  
:\*\*\*\*\*  
DEVTYP <DMR11>

010230  
(4) 010230  
(3) 010230 046504 030522 000061  
(2)

L\$DVTYP::  
.ASCIZ /DMR11/  
.EVEN

:\*\*\*\*\*  
:\* TITLE OF PROGRAM  
:\*\*\*\*\*  
DESCRIPT <DMR-11 FUNCTIONAL TESTS>

010236  
(4) 010236  
(3) 010236 046504 026522 030461  
(3) 010244 043040 047125 052103  
(3) 010252 047511 040516 020114  
(3) 010260 042524 052123 000123  
(2)

L\$DESC::  
.ASCIZ /DMR-11 FUNCTION  
  
.EVEN

:  
: FORMAT STATEMENTS USED IN PRINT CALLS  
:

2712  
2713  
2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767

```
.SBTTL GLOBAL SUBROUTINES

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

:*****
:
:   MACROS - THERE ARE 2 BASIC TYPES OF MACROS USED
:           1. NORMAL MACROS -
:           2. DMR11 FUNCTIONAL MACROS - THESE MACROS MAY
:              BE NOTHING MORE THAN A CALL TO A SUBROUTINE,
:              BUT THEY ARE DISTINCT DMR FUNCTIONS WHICH CAN
:              DISTINGUISHED BY THE IN-LINE MACRO NAME.
:*****

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

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

:*****
: WAIT $FLAG MACRO - THIS MACRO INTERPUTS THE $FLAG AS RDI, RQI OR RDO.
:                    IF RDI OR RDO, THE SUBROUTINE CALLED WILL WAIT UNTIL
:                    THE RESPECTIVE BIT IS SET. IF RQI, THE SUBROUTINE
:                    CALLED WILL CLEAR RQI AND WAIT UNTIL RDI IS CLEARED.
:*****

:   .MACRO WAIT $FLAG
:   .NLIST
:   .LIST ME
:   .LIST
:
:           ;**** MACRO EXPANSION ****
:   .IF B, $FLAG
:   .ERROR FLAG ;## MISSING FLAG FOR WAIT - EXPANSION ABORT ##
:   .MEXIT
:   .ENDC
:   .IF IDN $FLAG,RQI
:   JSR PC, $CLRQI ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
:   .ENDC
:   .IF IDN $FLAG,RDI
:   JSR PC, $WAIT ;CALL WAIT ROUTINE
:   .WORD 0 ;FLAG THAT WE'RE WAITING FOR RDI
:   .ENDC
:   .IF IDN $FLAG,RDO
:   JSR PC, $WAIT ;CALL WAIT ROUTINE
```

```
2768 .ENDC .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
2769
2770 ;*****
2771 .NLIST ME ;*****
2772 .ENDM
2773
2774
2775 ;:*****
2776 ;: CLEAR MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE
2777 ;: $MSCLR SUBROUTINE
2778 ;:*****
2779 .MACRO CLEAR
2780 .NLIST
2781 .LIST ME
2782 .LIST
2783
2784 JSR PC, $MSCLR ;**** MACRO EXPANSION ****
2785 ;ISSUE A DMR MASTER CLEAR
2786 ;****
2787 .NLIST ME
2788 .ENDM
```

2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799  
2800  
2801  
2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826  
2827  
2828  
2829  
2830  
2831  
2832  
2833  
2834  
2835  
2836  
2837  
2838

```
*****  
: BASEIN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $BASEIN SUBROUTINE (WITH DEFAULT ARGUMENTS  
: IF ARGUMENTS NOT GIVEN)  
*****  
: .MACRO BASEIN $A,$B,$C  
: .NLIST  
: .LIST ME  
: .LIST  
  
: .IF B $A ;**** MACRO EXPANSION ****  
: JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS  
: .WORD LPLU ;SET LINE UNIT LOOP  
: .WORD BASE ;BASE TABLE ADDRESS  
: .WORD DMR ;DMR-11 MODE  
  
: .IFF  
: JSR PC, $BASEI ;CALL BASE IN ROUTINE  
: .WORD $A ;MAINTENANCE MODE BITS TO SET IN BSEL1  
: .WORD $B ;BASE TABLE ADDRESS  
: .WORD $C ;MODE  
  
: .ENDC  
  
: .NLIST ME ;****  
: .ENDM ;****  
  
*****  
: CNTRIN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $CNTIN SUBROUTINE (WITH DEFAULT ARGUMENTS  
: IF ARGUMENTS NOT GIVEN)  
*****  
: .MACRO CNTRIN $A  
: .NLIST  
: .LIST ME  
: .LIST  
  
: .IF B $A ;**** MACRO EXPANSION ****  
: JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE WITH DEFAULT  
: .WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.  
  
: .IFF  
: JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE  
: .WORD $A ;SEL6 - (DUPLEX, MODE)  
  
: .ENDC  
  
: .NLIST ME ;****  
: .ENDM ;****
```



2840  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885

```
::*****  
: DMRIN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $DMRIN SUBROUTINE  
:*****
```

```
.MACRO DMRIN $A,$B,$C  
.NLIST  
.LIST ME  
.LIST
```

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

```
.IF B $A  
.ERROR DMRIN; ## MISSING ARGUMENTS-EXPANSION ABORT ##  
.MEXIT  
.ENDC
```

```
JSR PC, $DMRIN ;CALL DMR MODE INPUT ROUTINE  
.WORD $A ;INPUT COMMAND
```

```
.IF B $B  
.WORD 0 ;NO SEL4
```

```
.IFF  
.WORD $B ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
```

```
.ENDC  
.IF B $C  
.WORD 0 ;NO SEL6
```

```
.IFF  
.WORD $C ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
```

```
.ENDC
```

```
;**** ****
```

```
.NLIST ME  
.ENDM
```

```
::*****  
: SHUTDN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $HALT SUBROUTINE  
:*****
```

```
.MACRO SHUTDN  
.NLIST  
.LIST ME  
.LIST
```

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

```
JSR PC, $HALT ;DMR HALT ROUTINE.  
;**** ****
```

```
.NLIST ME  
.ENDM
```

2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896  
2897  
2898  
2899  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937

```
*****  
: BACCIR MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $BACC SUBROUTINE (WITH DEFAULT ARGUMENTS  
: IF ARGUMENTS NOT GIVEN)  
*****  
: .MACRO BACCIR $A,$B  
: .NLIST  
: .LIST ME  
: .LIST  
: .IF B $A ;**** MACRO EXPANSION ****  
: JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS  
: .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND  
: .WORD RBUF ;RECEIVE BUFFER  
: .WORD RCOUNT ;RECEIVE CHARACTER COUNT  
: .IFF  
: JSR PC, $BACC ;CALL BA/CC IN ROUTINE  
: .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND  
: .WORD $A ;BUFFER ADDRESS BITS 0-15  
: .WORD $B ;BA BITS 16/17 AND CHAR. COUNT  
: .ENDC  
: .NLIST ME ;**** ****  
: .ENDM
```

```
*****  
: BACCIT MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $BACC SUBROUTINE (WITH DEFAULT ARGUMENTS  
: IF ARGUMENTS NOT GIVEN)  
*****  
: .MACRO BACCIT $A,$B  
: .NLIST  
: .LIST ME  
: .LIST  
: .IF B $A ;**** MACRO EXPANSION ****  
: JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS  
: .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND  
: .WORD TBUF ;TRANSMIT BUFFER ADDRESS  
: .WORD TCOUNT ;TRANSMIT CHARACTER COUNT  
: .IFF  
: JSR PC, $BACC ;CALL BA/CC IN ROUTINE  
: .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND  
: .WORD $A ;BUFFER ADDRESS BITS 0-15  
: .WORD $B ;BA BITS 16 & 17 AND CHAR. COUNT  
: .ENDC  
: .NLIST ME ;**** ****  
: .ENDM
```

2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965  
2966  
2967  
2968  
2969  
2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991  
2992  
2993  
2994

010266  
010266 005037 002354  
010272 005737 002370  
010276 001005  
010300 011637 002366  
010304 162737 000004 002366  
010312  
010312 017637 000000 002334  
010320 062716 000002  
010324 010046  
010326 010146  
010330 013701 002310  
010334  
010334 005000  
010336  
010336 032777 000200 171670  
010344 001034  
010346 032777 000200 171656

```

*****
*****
SUBROUTINE $WAIT
    FUNCTION - TO WAIT FOR RDI TO BE SET IN SEL0
                OR RDO TO BE SET IN SEL2

CALLING FORMAT:      JSR    PC,    $WAIT
                    .WORD  FLAG
                    (MACRO CALL -- WAIT RDI)

NESTING LEVEL      - MAY BE CALLED FROM ANOTHER SUBROUTINE

ENTRY CONDITIONS - FLAG = 1 - WAIT FOR RDO
                    = 0 - WAIT FOR RDI
                    WAIT1 = DELAY COUNTER (DETERMINED IN INIT.)
                    NESTPC= 1 - ROUTINE NESTED WITHIN ANOTHER
                                SUBROUTINE.
                    = 0 - ROUTINE NOT NESTED.

EXIT CONDITIONS - EITHER RDI OR RDO BIT SET AS EXPECTED
                OR (ERROR CONDITONS):
    1. RDI OR RDO SET, BUT NOT THE EXPECTED ONE
       THE USER WILL BE INFORMED. HOWEVER,
       THIS WILL NOT NECESSARILY BE AN ERROR.
    2. BIT NOT SET BEFORE DELAY EXPIRED.
       THIS WILL RESULT IN A HARD ERROR MESSAGE
       AND THE CARRY BIT WILL BE SET. THE CARRY
       BIT SET FLAG THE ERROR CONDITION.

REGISTERS DESTROYED - RESTORED
*****
*****
$WAIT:
    CLR    ERRFLG      ;CLEAR ERROR FLAG
    TST    NESTPC      ;IS THIS NESTED IN ANOTHER SUBROUTINE?
    BNE    10$         ;YES - USE THE SUBRPC ALREADY CALCULATED.
    MOV    (SP),SUBRPC ;SAVE PC AFTER THE CALL TO $WAIT.
    SUB    #4,SUBRPC   ;BACKUP TO THE PC OF THE ACTUAL CALL

10$:
    MOV    @ (SP),TEMP ;GET THE FLAG FOR RDI OR RDO
    ADD    #2,(SP)     ;INC THE PC LEFT ON THE STACK TO POINT
                        ;PAST THE FLAG ARGUMENT
    MOV    R0,-(SP)    ;SAVE R0
    MOV    R1,-(SP)    ;SAVE R1
    MOV    WAIT1,R1    ;DELAY COUNTER DETERMINED BY BAUD RATE
                        ;(DETERMINED IN INIT ROUTINE).

30$:
    CLR    R0          ;INNER LOOP COUNT OF DELAY COUNTER

40$:
    BIT    #RDO,@SEL2 ;IS THE RDO BIT SET IN SEL2?
    BNE    60$         ;YES - EXIT BIT CHECK LOOP.
    BIT    #RDI,@SELO ;IS THE RDI BIT SET IN SEL0?

```

```

2995 010354 001062          BNE      70$          ;YES - EXIT
2996 010356          BREAK          ;CALL SUPERVISOR - ALLOW CONSOLE INTERRUPT.
      (3) 010356 104422          TRAP      C$BRK
2997 010360 005300          DEC      R0          ;LOOP UNTIL R0 RETURNS TO 0
2998 010362 001365          BNE      40$
2999 010364          DELAY     1          ;DELAY 100 MICROSECONDS
      (2) 010364 012727 000001          MOV      #1,(PC)+
      (2) 010370 000000          .WORD   0
      (2) 010372 013727 002116          MOV      L$DLY,(PC)+
      (2) 010376 000000          .WORD   0
      (2) 010400 005367 177772          DEC      -6(PC)
      (2) 010404 001375          BNE      -4
      (2) 010406 005367 177756          DEC      -22(PC)
      (2) 010412 001367          BNE      -20
3000
3001 010414 005301          DEC      R1          ;BETWEEN LOOPS.
3002 010416 001346          BNE      30$          ;REPEAT UNTIL MAXIMUM LOOP SATISFIED.
3003 010420          ERRDF   1,EMG1,ERRG2 ;TIME OUT ERROR
      (4) 010420 104455          TRAP      C$ERDF
      (5) 010422 000001          .WORD   1
      (5) 010424 020014          .WORD   EMG1
      (5) 010426 015070          .WORD   ERRG2
3004 010430 005237 002354          INC      ERRFLG      ;SET ERROR FLAG
3005 010434 000445          BR       100$        ;BRANCH TO COMMON EXIT.
3006 010436          60$:
3007 010436 005737 002334          TST      TEMP        ;WERE WE WAITING FOR THE RDO FLAG?
3008 010442 001042          BNE      100$        ;YES - OK, EXIT.
3009 010444 022737 0C0001 002360          CMP      #CNTRL,ERROR ;IS THIS CONTROL OUT ERROR EXPECTED?
3010 010452 001436          BEQ      100$        ;IF YES, DON'T REPORT THE FOLLOWING ERRORS.
3011 010454          PRINTB #FMS1        ;RECEIVED AN RDO, WHEN WAITING FOR RDI
      (7) 010454 012746 010600          MOV      #FMS1,-(SP)
      (6) 010460 012746 000001          MOV      #1,-(SP)
      (3) 010464 010600          MOV      SP,R0
      (4) 010466 104414          TRAP      C$PNTB
      (4) 010470 062706 000004          ADD      #4,SP
3012 010474 032777 000001 171532          BIT      #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
3013 010502 001422          BEQ      100$        ;NO NEED TO CHECK ERROR CODES.
3014 010504          ERRDF   9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT.
      (4) 010504 104455          TRAP      C$ERDF
      (5) 010506 000011          .WORD   9
      (5) 010510 020136          .WORD   EMG9
      (5) 010512 015070          .WORD   ERRG2
3015 010514 005237 002354          INC      ERRFLG      ;SET ERROR FLAG.
3016 010520 000413          BR       100$
3017 010522          70$:
3018 010522 005737 002334          TST      TEMP        ;WERE WE WAITING FOR THE RDI FLAG?
3019 010526 001410          BEQ      100$        ;YES - OK, EXIT
3020 010530          PRINTB #FMS2        ;RECEIVED AN RDI, WHEN WAITING FOR RDO
      (7) 010530 012746 010641          MOV      #FMS2,-(SP)
      (6) 010534 012746 000001          MOV      #1,-(SP)
      (3) 010540 010600          MOV      SP,R0
      (4) 010542 104414          TRAP      C$PNTB
      (4) 010544 062706 000004          ADD      #4,SP
3021 010550          100$:
3022 010550 005737 002370          TST      NESTPC      ;WAS THIS NESTED IN ANOTHER SUBROUTINE?
3023 010554 00100?          BNE      105$        ;IF YES - LEAVE THE SUBROUTINE PC ALONE
    
```

```

3024 010556 005037 002366          CLR      SUBRPC          :CLEAR THE PC
3025 010562          105$:          MOV      (SP)+,R1        :RESTORE R1
3026 010562 012601          MOV      (SP)+,R0        :RESTORE R0
3027 010564 012600          TST      ERRFLG         :WAS THERE AN ERROR (CARRY CLEARED ON TST)
3028 010566 005737 002354          BEQ      110$           :IF NOT, RETURN WITH CARRY CLEAR
3029 010572 001401          SEC                          :SET CARRY.
3030 010574 000261
3031 010576          110$:          RETURN
3032 010576 000207
3033
3034 010600 047045 040445 042122 FMS1:  .ASCIZ  /%N%ARDO SET WHEN EXPECTING RDI%N/
      010606 020117 042523 020124
      010614 044127 047105 042440
      010622 050130 041505 044524
      010630 043516 051040 044504
      010636 047045      000
3035 010641      045 022516 051101 FMS2:  .ASCIZ  /%N%ARDI SET WHEN EXPECTING RDO%N/
      010646 044504 051440 052105
      010654 053440 042510 020116
      010662 054105 042520 052103
      010670 047111 020107 042122
      010676 022517 000116
3036          .EVEN
3037
3038
    
```

3040  
3041  
3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
(2)  
(2)  
(2)  
(2)  
(2)  
(2)

010702  
010702 005037 002354  
010706 042777 000040 171316  
010714 005737 002370  
010720 001005  
010722 011637 002366  
010726 162737 000004 002366  
010734  
010734 010046  
010736 010146  
010740 013701 002312  
010744  
010744 005000  
010746  
010746 032777 000200 171256  
010754 001427  
010756 104422  
010760 005300  
010762 001371  
010764  
010764 012727 000001  
010770 000000  
010772 013727 002116  
010776 000000  
011000 005367 177772  
011004 001375

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

SUBROUTINE \$CLRqi

FUNCTION - TO CLEAR Rqi AND WAIT FOR RDI TO BE CLEARED

CALLING FORMAT: JSR PC, \$CLRqi  
(MACRO CALL -- WAIT Rqi)

NESTING LEVEL - MAY BE NESTED WITHIN ANOTHER SUBROUTINE

ENTRY CONDITIONS - WAIT2 = DELAY COUNTER (DETERMINED IN INIT. ROUTINE)  
 NESTPC= 1 - ROUTINE NESTED WITHIN ANOTHER SUBROUTINE.  
 = 0 - ROUTINE NOT NESTED.

EXIT CONDITIONS - 1. NON ERROR, DMR READY TO RECEIVE THE NEXT COMMAND  
 2. ERROR IF RDI DOES NOT CLEAR BEFORE THE DELAY ROUTINE EXPIRES. AN ERROR MESSAGE WILL OCCUR. ALSO A CARRY BIT WILL BE SET TO FLAG THE ERROR FOR THE USER.

REGISTERS DESTROYED - RESTORED

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

```

$CLRqi:
CLR    ERRFLG    ;CLEAR ERROR FLAG
BIC    #Rqi,@SELO ;REQUEST INPUT CLEAR
TST    NESTPC    ;IS THIS NESTED IN ANOTHER SUBROUTINE?
BNE    10$      ;YES - USE SUBRPC CALCULATED
MOV    (SP),SUBRPC ;SAVE THE PC AFTER THE CALL TO $WAIT.
SUB    #4,SUBRPC ;BACKUP TO THE PC OF THE ACTUAL CALL.

10$:
MOV    R0,-(SP) ;SAVE R0
MOV    R1,-(SP) ;SAVE R1
MOV    WAIT2,R1 ;GET THE DELAY COUNTER (DETERMINED BY
                ;BAUD RATE IN INIT ROUTINE)

12$:
CLR    R0       ;INNER LOOP COUNT

20$:
BIT    #RDI,@SELO ;IS THE RDI BIT CLEAR IN SELO?
BEQ    30$      ;YES - EXIT
BREAK ;CALL SUPERVISOR - ALLOW CONSOLE INTERRUPT.
                TRAP    C$BRK

DEC    R0       ;LOOP UNTIL R0 RETURNS TO 0
BNE    20$
DELAY 1         ;DELAY 100 MICROSECONDS

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

```

(2) 011006 005367 177756
(2) 011012 001367
3089 011014 005301
3090 011016 001352
3091 011020
(4) 011020 104455
(5) 011022 000001
(5) 011024 020014
(5) 011026 015070
3092 011030 005237 002354
3093 011034
3094 011034 005737 002370
3095 011040 001002
3096 011042 005037 002366
3097 011046
3098 011046 012601
3099 011050 012600
3100 011052 005737 002354
3101 011056 001401
3102 011060 000261
3103 011062
3104 011062 000207
3105
3106

```

					DEC	-22(PC)
					BNE	.-20
				DEC	R1	
				BNE	12\$	
				ERRDF	1,EMG1,ERRG2	
						:REPEAT UNTIL MAXIMUM LOOP SATISFIED.
						:TIME OUT ERROR
						TRAP
						.WORD
						.WORD
						.WORD
						C\$ERDF
						1
						EMG1
						ERRG2
				INC	ERRFLG	
		30\$:				:SET ERROR FLAG
				TST	NESTPC	
				BNE	40\$	
				CLR	SUBRPC	
		40\$:				:WAS THIS A NESTED ROUTINE?
						:IF YES - LEAVE THE SUBRPC ALONE
						:CLEAR THE PC
				MOV	(SP)+,R1	
				MOV	(SP)+,R0	
				TST	ERRFLG	
				BEQ	50\$	
				SEC		
		50\$:				:RESTORE R1
						:RESTORE R0
						:WAS THERE AN ERROR? (CARRY CLEARED ON TST)
						:IF NOT - RETURN WITH CARRY CLEAR
						:SET CARRY.
				RETURN		

3108  
3109  
3110  
3111  
3112  
3113  
3114  
3115  
3116  
3117  
3118  
3119  
3120  
3121  
3122  
3123  
3124  
3125  
3126  
3127  
3128  
3129  
3130  
3131  
3132  
3133  
3134  
3135  
3136  
3137  
3138  
3139  
3140  
3141  
3142  
3143  
3144  
3145  
3146  
3147  
3148  
3149  
3150  
3151  
3152  
3153  
3154  
3155  
3156  
3157  
3158  
3159  
3160  
3161  
3162  
3163

```

*****
*****
SUBROUTINE $MSCLR
FUNCTION - TO PERFORM A MASTER CLEAR FOR THE DMR11
CALLING FORMAT:      JSR    PC,    $MSCLR
                    (MACRO CALL -- CLEAR)
NESTING LEVEL - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST,
                    SUBTEST OR TEST SEGMENT)
ENTRY CONDITIONS - WAIT2 = DELAY COUNTER (DETERMINED BY INIT. ROUTINE)
                    CLRNO = EVEN OR ODD COUNT. THE ACTUAL # IS NOT
                    SIGNIFICANT, HOWEVER IF BIT 0 IS SET
                    THEN THE MICROTTEST IS SET ALONG WITH
                    THE MASTER CLEAR. THIS ROUTINE WILL INCR.
                    THE VALUE. THIS WILL RESULT IN THE MICRO
                    TESTS BEING RUN ON EVERY OTHER MASTER CLEAR
EXIT CONDITIONS - 1. NO ERROR - DMR11 MICROPROCESSOR INITIALIZED
                    2. IF RUN BIT NOT SET BEFORE DELAY TIMEOUT, ERROR
                    WILL RESULT. ADDITONALLY THE ERROR MESSAGE WILL
                    RELAY THE RESULTS OF THE MICROTTESTS IF THE RUN
                    BIT IS NOT SET.
NOTE:              THERE IS A PATCH AREA TO ALLOW THESE DIAGNOSTICS
                    TO RUN ON A M8206 (INSTEAD OF M8207). THIS
                    SHOULD BE FOR DEVELOPMENT USE ONLY.
REGISTERS DESTROYED - RESTORED
    
```

```

*****
*****
$MSCLR:
MOV    (SP),SUBRPC    ;SAVE PC AFTER THE CALL TO $WAIT.
SUB    #4,SUBRPC      ;BACKUP TO THE PC OF THE ACTUAL CALL
MOV    R0,-(SP)       ;SAVE R0
MOV    R1,-(SP)       ;SAVE R1
CLRB   @BSEL3         ;CLEAR BSEL3
NOP
NOP
NOP
NOP
;*****
;** PATCH AREA FOR 8206 IF NEEDED **
;CLR @#SEL6 -
;*****
BIT    #BIT0,CLRNO    ;IS THIS AN ODD MASTER CLEAR.
BNE    7$             ;IF YES - BR
MOV    #MCLR,@SELO    ;ISSUE A MASTER CLEAR.
BR     8$
7$:
MOV    #MCLR!MDIAG,@SELO ;ISSUE THE MASTER CLEAR AND TOGGLE
;MICRO TEST SWITCH.
8$:
    
```

```

011064 011637 002366
011064 162737 000004 002366
011076 010046
011100 010146
011102 105077 171136
011106 000240
011110 000240
011112 000240
011114 000240
011116 032737 000001 002372
011124 001004
011126 012777 040000 171076
011134 000403
011136 012777 060000 171066
011144
    
```



```

3164 011144 000240      NOP
3165 011146 000240      NOP
3166 011150 000240      NOP
3167 011152 000240      NOP
3168
3169 011154 005237 002372  INC   CLRNO
3170 011160 013701 002312  MOV   WAIT2,R1
3171
3172 011164          10$:
3173 011164 005000          CLR   R0
3174 011166          20$:
3175 011166 032777 100000 171036  BIT   #RUN,@SELO
3176 011174 001025          BNE   40$
3177 011176          BREAK
(3) 011176 104422          TRAP  C$BRK
3178 011200 005300          DEC   R0
3179 011202 001371          BNE   20$
3180 011204          DELAY 1
(2) 011204 012727 000001          ;DELAY 100 MICROSECONDS
(2) 011210 000000          MOV   #1,(PC)+
(2) 011212 013727 002116          .WORD 0
(2) 011216 000000          MOV   L$DLY,(PC)+
(2) 011220 005367 177772          .WORD 0
(2) 011224 001375          DEC   -6(PC)
(2) 011226 005367 177756          BNE   -4
(2) 011232 001367          DEC   -22(PC)
3181 011234 005301          BNE   -20
3182 011236 001352          DEC   R1
3183 011240          ERRDF 1,EMG1,ERRG3
(4) 011240 104455          ;REPEAT UNTIL MAX LOOP SATISFIED.
(5) 011242 000001          TRAP  C$ERDF
(5) 011244 020014          .WORD 1
(5) 011246 015204          .WORD EMG1
3184 011250          40$:
3185 011250 012601          .WORD ERRG3
3186 011252 012600          MOV   (SP)+,R1
3187 011254 005037 002366          MOV   (SP)+,R0
3188          CLR   SUBRPC
3189 011260 000207          ;RESTORE R1
3190          RETURN          ;RESTORE R0
3191          ;TIDY UP SUBRPC
  
```

3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208  
3209  
3210  
3211  
3212  
3213  
3214  
3215  
3216  
3217  
3218  
3219  
3220  
3221  
3222  
3223  
3224  
3225  
3226  
3227  
3228  
3229  
3230  
3231  
3232  
3233  
3234  
(1)  
(1)  
(1)  
(1)  
3235  
(2)  
3236  
3237  
3238  
3239  
3240  
3241  
3242  
3243

011262				
011262	011637	002366		
011266	162737	000004	002366	
011274	112777	000043	170730	
011302	012737	000001	002370	
011310				
011310	004737	010266		
011314	000000			
011316				
011316	103003			
011320	062716	000006		
011324	000467			
011326				
011326	057677	000000	170676	
011334	062716	000002		
011340	017677	000000	170670	
011346	062715	000002		
011352	017677	000000	170660	

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

SUBROUTINE \$BASEI

FUNCTION - TO PERFORM A BASE IN COMMAND

CALLING FORMAT: JSR PC, \$BASEI  
 .WORD A (SELO MAINTENANCE BITS)  
 .WORD B (SEL4 - ADDRESS)  
 .WORD C (SEL6 - MODE AND/OR RESUME)  
 (MACRO CALL -- BASEIN OR BASEIN A,B,C)

NESTING LEVEL - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST, SUBTEST OR TEST SEGMENT)

ENTRY CONDITIONS - A = MAINTENANCE BITS (I.E. LINE UNIT LOOP BACK)  
 B = BASE TABLE ADDRESS (SEL4)  
 C = MODE + RESUME (SEL6)  
 INFACE = 0 - NO INTERFACE WRITE REQUIRED  
 1 - WRITE INTERFACE (AX3-15)

EXIT CONDITIONS - 1. IF NO ERROR - DMR11 BASE TABLE ASSIGNED  
 2. IF IN DMR MODE, AND INTERFACE WRITE REQUESTED  
 WRITE REQUESTED AX3-15.  
 3. TIMEOUT ERRORS ARE DETECTED IN WAIT SUBROUTINES.  
 DMRFLG = -1 DMR MODE REQUESTED (USED IN CONTROL IN ROUTINE)  
 0 DMC MODE OR RESUME REQUESTED.

REGISTERS DESTROYED - RESTORED

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

```

$BASEI:
MOV    (SP),SUBRPC    ;SAVE PC AFTER THE CALL TO $WAIT.
SUB    #4,SUBRPC      ;BACKUP TO THE PC OF THE ACTUAL CALL

MOV    #RQI!BASEI,@BSELO ;ISSUE THE BASE IN COMMAND.
MOV    #1,NESTPC      ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT   RDI            ;WAIT FOR RDI
;**** MACRO EXPANSION ****
JSR    PC,$WAIT       ;CALL WAIT ROUTINE
;FLAG THAT WE'RE WAITING FOR RDI
;****
BNERROR 10$           ;IF NO ERROR, RDI SET - PROCEED
ADD    #6,(SP)        ;CORRECT STACK FOR ERROR EXIT
BR     30$            ;EXIT
;
10$:
BIS    @ (SP),@SELO   ;SET ANY MAINTENANCE BITS
ADD    #2,(SP)        ;INC. POINTER.
MOV    @ (SP),@SEL4   ;SET UP BASE ADDRESS
ADD    #2,(SP)        ;INC. POINTER AGAIN
MOV    @ (SP),@SEL6   ;SET UP RESUME BIT AND THE HIGH 2 BITS
    
```

```

3244
3245 011360 062716 000002      ADD    #2,(SP)      ;OF THE BASE TABLE ADDRESS
3246 011364                WAIT   RQI          ;INC. POINTER AGAIN (SHOULD BE AT RETURN PC)
(1)                ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
(1) 011364 004737 010702      JSR    PC,$CLRQI   ;**** MACRO EXPANSION ****
(1)                ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
3247 011370                BERROR 30$        ;****
(2) 011370 103445                ;IF ERROR, EXIT
3248 011372 122777 000122 170640  CMPB   #122,@BSEL6 ;WAS THIS A DMR BASE IN?
3249 011400 001004                BNE    15$         ;IF NOT, CLEAR DMR FLAG (DMC MODE)
3250 011402 032777 010000 170630  BIT    #RES,@SEL6 ;IS THIS A RESUME?
3251 011410 001403                BEQ    16$         ;IF NOT, PROCEED
3252 011412                15$:
3253 011412 005037 002260      CLR    DMRFLG     ;CLEAR DMR FLAG (NO DMR RUN ACKNOWLEDGE).
3254 011416 000432                BR     30$        ;SKIP - TO END
3255 011420                16$:
3256 011420 012737 177777 002260  MOV    #-1,DMRFLG ;FLAG THAT DMR MODE WAS REQUESTED.
3257 011426 005737 002262      TST   INFACE     ;IS AN INTERFACE WRITE REQUIRED?
3258 011432 001424                BEQ    30$        ;IF NOT - SKIP TO END
3259 011434 022737 000001 002360  CMP    #CNTRL,ERROR ;ARE WE EXPECTING AN ERROR (IN TEST THAT
3260                ;FORCES AN ERROR)
3261 011442 001004                BNE    17$        ;IF NOT PROCEED
3262 011444 032777 000200 170560  BIT    #RDO,@SELO ;IF EXPECTING AN ERROR - IS RDO SET
3263 011452 001014                BNE    30$        ;IF YES - DON'T BOTHER CHANGING THE INTERFACE.
3264 011454                17$:
3265 011454 112777 000055 170550  MOVB  #RQI!INTER,@BSELO ;ISSUE WRITE INTERFACE COMMAND.
3266 011462                WAIT   RDI          ;WAIT FOR RDI
(1)                ;**** MACRO EXPANSION ****
(1) 011462 004737 010266      JSR    PC,$WAIT   ;CALL WAIT ROUTINE
(1) 011466 000000                .WORD 0          ;FLAG THAT WE'RE WAITING FOR RDI
(1)                ;****
3267 011470                BERROR 30$        ;IF ERROR, BR TO END.
(2) 011470 103405                ;****
3268 011472 113777 002304 170550  MOVB  AX3,@BSEL7 ;WRITE AX3-15. INTERFACE SELECTED
3269                ;BY AX3 DETERMINED IN INIT. CODE.
3270 011500                WAIT   RQI          ;CLEAR RQI AND WAIT FOR RDI TO CLEAR.
(1)                ;**** MACRO EXPANSION ****
(1) 011500 004737 010702      JSR    PC,$CLRQI ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
(1)                ;****
3271 011504                30$:
3272 011504 005037 002370      CLR    NESTPC    ;CLEAR THE NEST FLAG
3273 011510 005037 002366      CLR    SUBRPC   ;TIDY UP SUBRPC
3274 011514 000207                RETURN
3275
3276
    
```

3278  
3279  
3280  
3281  
3282  
3283  
3284  
3285  
3286  
3287  
3288  
3289  
3290  
3291  
3292  
3293  
3294  
3295  
3296  
3297  
3298  
3299  
3300  
3301  
3302  
3303  
3304  
3305  
3306  
3307  
3308  
3309  
3310  
3311  
3312  
3313  
3314  
3315  
3316  
(1)  
(1)  
(1)  
(1)  
3317  
(2)  
3318  
3319  
3320  
3321  
3322  
3323  
3324  
3325  
3326  
3327  
(1)

011516			
011516	011637	002366	
011522	162737	000004	002366
011530	112777	000041	170474
011536	012737	000001	002370
011544			
011544	004737	010266	
011550	000000		
011552			
011552	103003		
011554	062716	000002	
011560	000463		
011562			
011562	017677	000000	170450
011570	062716	000002	
011574	032777	000400	170436
011602	001402		
011604	005037	002260	
011610			
011610			

```

*****
*****
SUBROUTINE $CNTIN
FUNCTION - TO PERFORM A CONTROL IN COMMAND
CALLING FORMAT:      JSR    PC,    $CNTIN
                     .WORD  A (SEL6 - MAINTENANCE MODE & HDX)
                     (MACRO CALL -- CNTRIN OR CNTRIN A)
NESTING LEVEL      - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST,
                     SUBTEST OR TEST SEGMENT)
ENTRY CONDITIONS   - DMRFLG = -1 EXPECT CONTROL OUT IF IN DMR MODE
                     = 0 NO CONTROL OUT, IN DMC MODE OR RESUME.

EXIT CONDITIONS - 1. IF NO ERROR - DMR11 CONTROL IN PERFORMED
                  2. TIMEOUTS REPORTED IN WAIT SUBROUTINES
                  3. IF THIS IS A DMR MODE START UP CONTROL IN,
                     THIS ROUTINE WILL WAIT FOR A CONTROL
                     OUT - DMR RUN. IF THIS CONTROL OUT IS
                     NOT RECEIVED, THIS WILL RESULT IN AN ERROR
                     MESSAGE AND A REMINDER TO CHECK THE BAUD RATE,
                     INTERFACE AND TURNAROUND (PROBABLE REASON).
    
```

REGISTERS DESTROYED

```

*****
*****
$CNTIN:
MOV    (SP),SUBRPC      ;SAVE PC FROM WHERE THIS SUBR. WAS CALLED.
SUB    #4,SUBRPC        ;BACKUP TO PC OF ACTUAL CALL
MOVB   #RQI+CNTRL,@BSEL0 ;SET UP CONTROL IN COMMAND
MOV    #1,NESTPC       ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT   RDI              ;WAIT FOR SETTING OF RDI
                     ;**** MACRO EXPANSION ****
JSR    PC,$WAIT        ;CALL WAIT ROUTINE
                     .WORD  0
                     ;FLAG THAT WE'RE WAITING FOR RDI
                     ;****          ****
BNERROR 1$             ;IF NO ERROR - PROCEED
                     BCC    1$
ADD    #2,(SP)         ;CORRECT RETURN ADDRESS
BR     20$             ;ERROR - EXIT
1$:
MOV    @(SP),@SEL6     ;SET MODE DESIRED
ADD    #2,(SP)         ;INC. RETURN PC LEFT ON STACK.
BIT    #MAINT,@SEL6   ;WAS MAINTENANCE MODE REQUESTED?
BEQ    5$              ;IF NOT, LEAVE DMRFLG AS IS.
CLR    DMRFLG         ;CLEAR FLAG - NO RUN MODE CONTROL OUT.
5$:
WAIT   RQI              ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
                     ;**** MACRO EXPANSION ****
    
```

```

(1) 011610 004737 010702      JSR    PC, $CLRQI      ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
(1)                               ;*****
3328 011614 005737 002260      TST    DMRFLG          ;WAS DMR MODE REQUESTED ON BASE IN?
3329 011620 001443              BEQ    20$              ;BR IF NOT (DMC MODE)
3330 011622 005037 002260      CLR    DMRFLG          ;CLEAR DMR RUN MODE FLAG
3331 011626                      WAIT   RDO              ;EXPECT RDO TO BE SET
(1)                               ;***** MACRO EXPANSION *****
(1) 011626 004737 010266      JSR    PC, $WAIT      ;CALL WAIT ROUTINE
(1) 011632 000001              .WORD  1              ;FLAG THAT WE'RE WAITING FOR RDO
(1)                               ;*****
3332 011634                      BERROR 7$            ;IF NO ERROR - PROCEED
(2) 011634 103011              PRINTB #FMS3          ;PRINT RUN ACKNOWLEDGE NOT RECEIVED.
3333 011636                      BCC    7$              ;
(7) 011636 012746 011742      MOV    #FMS3,-(SP)    ;
(6) 011642 012746 000001      MOV    #1,-(SP)      ;
(3) 011646 010600              MOV    SP,R0          ;
(4) 011650 104414              TRAP  C$PNTB         ;
(4) 011652 062706 000004      ADD   #4,SP          ;
3334 011656 000421              BR     15$            ;
3335 011660                      7$:
3336 011660 032777 000001 170346  BIT    #CNTRL,@SEL2   ;DID WE RECEIVE A CONTROL OUT?
3337 011666 001005              BNE   10$              ;IF YES - PROCEED.
3338 011670                      ERRDF 8,EMG8,ERRG2    ;EXPECTED CONTROL OUT NOT RECEIVED.
(4) 011670 104455              TRAP  C$ERDF         ;
(5) 011672 000010              .WORD 8                ;
(5) 011674 020072              .WORD EMG8             ;
(5) 011676 015070              .WORD ERRG2           ;
3339 011700 000410              BR     15$            ;
3340 011702                      10$:
3341 011702 032777 000040 170330  BIT    #DMRRUN,@SEL6  ;WAS THE DMR RUN MODE BIT SET?
3342 011710 0C1004              BNE   15$              ;BR IF OK.
3343 011712                      ERRDF 9,EMG9,ERRG2    ;WRONG CONTROL OUT RECEIVED.
(4) 011712 104455              TRAP  C$ERDF         ;
(5) 011714 000011              .WORD 9                ;
(5) 011716 020136              .WORD EMG9             ;
(5) 011720 015070              .WORD ERRG2           ;
3344 011722                      15$:
3345 011722 042777 000207 170304  BIC    #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS
3346 011730                      20$:
3347 011730 005037 002370      CLR    NESTPC          ;CLEAR THE NEST FLAG
3348 011734 005037 002366      CLR    SUBRPC          ;CLEAR PC
3349 011740 000207              RETURN
3350 011742 040445 046504 020122  FMS3: .ASCII /%ADMR RUN ACKNOWLEDGMENT NOT RECEIVED%/
3351 011750 052522 020116 041501
3352 011756 047113 053517 042514
3353 011764 043504 042515 052116
3354 011772 047040 052117 051040
3355 012000 041505 044505 042526
3356 012006 022504 116
3357 012011 045 024101 044103      .ASCIZ /%(CHECK INTERFACE, BAUD AND TURNAROUND)%/
3358 012016 041505 020113 047111
3359 012024 042524 043122 041501
3360 012032 026105 041040 052501
3361 012040 020104 047101 020104
  
```

012046 052524 047122 051101  
 012054 052517 042116 022451  
 012062 000116

3354  
3355  
3356  
3357  
3358  
3359  
3360  
3361  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385  
3386  
3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398  
3399  
3400  
3401  
3402  
3403  
3404  
3405  
3406

.EVEN

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

SUBROUTINE \$DMRIN

FUNCTION - TO PERFORM A DMR MODE INPUT COMMAND

CALLING FORMAT: JSR PC, \$DMRIN  
 .WORD COMMAND  
 .WORD B  
 .WORD C  
 (MACRO CALL -- DMRIN A,B,C)

NESTING LEVEL - MAY BE CALLED FROM IN-LINE CODE (TEST, SUBTEST OR TEST SEGMENT) OR FROM THE \$LOOP SUBROUTINE

ENTRY CONDITIONS - MUST BE IN DMR MODE  
 FOR ALL COMMANDS EXCEPT WRITE MODEM  
 B = SEL4  
 C = SEL6  
 FOR MODEM WRITE  
 B = BITS TO CLEAR IN SEL6  
 C = BITS TO SET IN SEL6  
 NESTPC = 1 - SUBROUTINE NESTED WITHIN ANOTHER SUB.  
 = 0 - SUBROUTINE NOT NESTED.

EXIT CONDITIONS - IF NO ERROR - DMR11 MODE INPUT COMMAND PERFORMED.

REGISTERS DESTROYED

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

\$DMRIN:  
 TST NESTPC ;IS THIS SUBROUTINE NESTED?  
 BNE 1\$ ;IF YES - DON'T CHANGE SUBRPC.  
 MOV (SP),SUBRPC ;SAVE PC FROM WHERE THIS SUBR. WAS CALLED.  
 SUB #4,SUBRPC ;BACKUP TO PC OF ACTUAL CALL  
 1\$:  
 MOVB @(SP),SAVE ;SAVE DMR INPUT COMMAND  
 MOVB @(SP),@BSELO ;SET UP DMR INPUT COMMAND.  
 ADD #2,(SP) ;INC RETURN PC LEFT ON STACK.  
 BIS #RQI,@SELO ;REQUEST INPUT.  
 MOV NESTPC,-(SP) ;SAVE THE CURRENT NEST FLAG.  
 MOV #1,NESTPC ;USE THE FLAG TO SHOW THE WAIT  
 ;ROUTINE IS NESTED.

012064 005737 002370  
 012070 001005  
 012072 011637 002366  
 012076 162737 000004 002366  
 012104  
 012104 117637 000000 002336  
 012112 117677 000000 170112  
 012120 062716 000002  
 012124 052777 000040 170100  
 012132 013746 002370  
 012136 012737 000001 002370

```

3407 012144          WAIT   RDI          ;WAIT FOR SETTING OF RDI
(1)
(1) 012144 004737 010266      JSR    PC, $WAIT          ;**** MACRO EXPANSION ****
(1) 012150 000000          .WORD   0          ;CALL WAIT ROUTINE
(1)          .WORD   0          ;FLAG THAT WE'RE WAITING FOR RDI
3408 012152 012637 002370      MOV    (SP)+,NESTPC      ;****          ****
3409 012156          BNEROR  5$          ;RESTORE THE ORIGINAL NEST FLAG.
(2) 012156 103003          ;IF NO ERROR, OK - PROCEED.
3410 012160 062716 000004      ADD    #4,(SP)          BCC    5$
3411 012164 000433          BR     10$             ;UPDATE RETURN ADDRESS.
3412 012166          ;ERROR EXIT.
3413 012166 122737 000005 002336 5$:  CMPB  #WMODEM,SAVE      ;IS THIS A MODEM WRITE?
3414 012174 001413          BEQ   6$             ;IF YES - SET/CLEAR BITS.
3415 012176 017677 000000 170032      MOV    @ (SP),@SEL4      ;PASS VALUE FOR SEL4 (VALUE, IF ANY,
3416          ;DEPENDS ON THE DMR COMMAND)
3417 012204 062716 000002          ADD    #2,(SP)          ;INC. RETURN PC LEFT ON STACK.
3418 012210 017677 000000 170022      MOV    @ (SP),@SEL6      ;PASS VALUE FOR SEL6 (VALUE, IF ANY,
3419          ;DEPENDS ON THE DMR COMMAND)
3420 012216 062716 000002          ADD    #2,(SP)          ;INC. RETURN PC LEFT ON STACK.
3421 012222 000412          BR     7$
3422 012224          ;
3423 012224 047677 000000 170006 6$:  BIC   @ (SP),@SEL6      ;CLEAR MODEM BITS
3424 012232 062716 000002          ADD    #2,(SP)          ;INC. RETURN PC LEFT ON STACK
3425 012236 057677 000000 167774      BIS   @ (SP),@SEL6      ;SET MODEM BITS
3426 012244 062716 000002          ADD    #2,(SP)          ;INC. RETURN PC LEFT ON STACK.
3427 012250          ;
3428 012250          7$:  WAIT   RQI          ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
(1)          ;**** MACRO EXPANSION ****
(1) 012250 004737 010702      JSR    PC, $CLRQI        ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
(1)          ;****          ****
3429 012254          10$:
3430 012254 005737 002370      TST   NESTPC          ;WAS THIS ROUTINE NESTED?
3431 012260 001002          BNE   15$            ;BR IF YES
3432 012262 005037 002366      CLR   SUBRPC          ;CLEAR PC
3433 012266          15$:
3434 012266 005037 002336      CLR   SAVE            ;RESTORE TEMP VALUE
3435 012272 000207          RETURN
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454

```

```

:*****
:*****
SUBROUTINE $BACC
FUNCTION - TO PERFORM A BUFFER ADDRESS/CHARACTER
COUNT IN COMMAND
CALLING FORMAT:
JSR    PC, $BACC
.WORD  SEL0 ;BA/CC IN COMMAND
.WORD  SEL4 ;BUFFER ADDRESS
.WORD  SEL6 ;BA BITS 16 & 17 AND
          ;CHARACTER COUNT
(MACRO CALL -- BACCIT OR BACCIT A,B)
OR (MACRO CALL -- BACCIR OR BACCIR A,B)

```

3455  
 3456  
 3457  
 3458  
 3459  
 3460  
 3461  
 3462  
 3463  
 3464  
 3465  
 3466  
 3467  
 3468  
 3469  
 3470  
 3471  
 3472  
 3473  
 3474  
 (1)  
 (1)  
 (1)  
 (1)  
 3475  
 (2)  
 3476  
 3477  
 3478  
 3479  
 3480  
 3481  
 3482  
 3483  
 3484  
 (1)  
 (1)  
 (1)  
 3485  
 3486  
 3487  
 3488  
 3489  
 3490  
 3491  
 3492  
 3493  
 3494  
 3495  
 3496  
 3497  
 3498  
 3499  
 3500  
 3501  
 3502

NESTING LEVEL - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST, SUBTEST OR TEST SEGMENT)

ENTRY CONDITIONS -

EXIT CONDITIONS - IF NO ERROR - DMR11 BA/CC COMMAND IN PERFORMED

REGISTERS DESTROYED - NOT AFFECTED

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

\$BACC:

```

MOV (SP), SUBRPC ;SAVE PC FROM WHERE THIS SUBR. WAS CALLED.
SUB #4, SUBRPC ;BACKUP TO PC OF ACTUAL CALL
MOVB @(SP), @BSELO ;SET UP BA/CC COMMAND IN (TRANSMIT OR RECEIVE)
ADD #2, (SP) ;INC POINTER ON STACK
MOV #1, NESTPC ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT RDI ;WAIT FOR SETTING OF RDI
;**** MACRO EXPANSION ****
JSR PC, $WAIT ;CALL WAIT ROUTINE
.WORD 0 ;FLAG THAT WE'RE WAITING FOR RDI
;**** ****
BNEROR 10$ ;IF NO ERROR - PROCEED
ADD #4, (SP) ;CORRECT STACK FOR ERROR EXIT. BCC 10$
BR 20$ ;EXIT
10$:
MOV @(SP), @SEL4 ;SET BUFFER ADDRESS
ADD #2, (SP) ;INC POINTER ON STACK
MOV @(SP), @SEL6 ;SET UP BUFFER COUNT AND BUFFER ADDRESS
;BITS 16 & 17
ADD #2, (SP) ;INC POINTER ON STACK
WAIT RQI ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
;**** MACRO EXPANSION ****
JSR PC, $CLRQI ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
;**** ****
20$:
CLR NESTPC ;CLEAR THE NEST FLAG
CLR SUBRPC ;CLEAR PC
RETURN
    
```

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

SUBROUTINE \$ERROR

FUNCTION - TO CHECK THE FIRST 8. BASE TABLE ERROR COUNTS FOR NON-ZERO VALUES.

CALLING FORMAT: JSR PC, \$ERROR



3503 : NESTING LEVEL - CAN BE NESTED WITHIN ANOTHER ROUTINE  
 3504 :  
 3505 : ENTRY CONDITIONS - SHOULD BE DONE AFTER PROPER SHUTDOWN  
 3506 : NESTPC = 1 - SUBROUTINE NESTED WITHIN ANOTHER SUB.  
 3507 : = 0 - SUBROUTINE NOT NESTED.  
 3508 :  
 3509 : EXIT CONDITIONS - IF ANY NON-ZERO VALUE FOUND IN THE BASE TABLE A  
 3510 : SOFT ERROR IS DECLARED.  
 3511 :  
 3512 :  
 3513 :  
 3514 :  
 3515 :  
 3516 :  
 3517 :  
 3518 :  
 3519 :  
 3520 :  
 3521 :  
 3522 :  
 3523 :  
 3524 :  
 3525 :  
 3526 :  
 3527 :  
 3528 :  
 3529 :  
 3530 :  
 3531 :  
 3532 :  
 3533 :  
 3534 :  
 3535 :  
 3536 :  
 3537 :  
 3538 :  
 3539 :  
 3540 :  
 3541 :  
 3542 :  
 3543 :  
 3544 :  
 3545 :  
 3546 :  
 (4) :  
 (5) :  
 (5) :  
 (5) :  
 3547 :  
 3548 :  
 3549 :  
 3550 :  
 3551 :  
 3552 :  
 3553 :  
 3554 :

REGISTERS DESTROYED - RESTORED

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

```

3517 012406          $ERROR:
3518 012406 005737 002370      TST     NESTPC      ;IS THIS ROUTINE NESTED?
3519 012412 001005          BNE     10$        ;BR IF YES (PC ALREADY SAVED)
3520 012414 011637 002366      MOV     (SP),SUBRPC ;SAVE PC AFTER THE CALL TO $WAIT.
3521 012420 162737 000004 002366 SUB     #4,SUBRPC   ;BACKUP TO THE PC OF THE ACTUAL CALL
3522                                     ;THE INSTRUCTION AFTER THE CALL.
3523 012426          10$:
3524 012426 010046          MOV     R0,-(SP)   ;SAVE R0
3525 012430 010146          MOV     R1,-(SP)   ;SAVE R1
3526 012432 012700 002633      MOV     #BASE+3,R0 ;POINTER TO ACTUAL BASE TABLE COUNTS.
3527 012436 012701 000006      MOV     #6.,R1    ;CHECK THE 6 NAK BYTES IN THE TABLE
3528 012442          20$:
3529 012442 105720          TSTB   (R0)+      ;IS THE NAK COUNT NON-ZERO?
3530 012444 001016          BNE     30$        ;IF YES - REPORT SOFT ERROR
3531 012446 005301          DEC     R1        ;LOOP UNTIL DONE.
3532 012450 001374          BNE     20$
3533 012452 122010          CMPB   (R0)+,(R0) ;ARE THE REPS THE SAME?
3534 012454 001012          BNE     30$        ;IF NOT - REPORT ERROR.
3535 012456 022737 000022 002114 CMP     #18.,L$TEST ;IS THIS TEST 18 (LARGE BUFFER TEST)
3536 012464 001403          BEQ     25$        ;IF YES - ALLOW 1 REP
3537 012466 105710          TSTB   (R0)      ;IF NOT TEST 18 - REPORT IF NON ZERO.
3538 012470 001004          BNE     30$
3539 012472 000407          BR      40$
3540 012474          25$:
3541 012474 122710 000001      CMPB   #1,(R0)   ;IS THE REP 0 OR 1?
3542 012500 002004          BGE     40$
3543                                     ;IF YES - OK (WE ALLOW 1 REP BECAUSE
3544                                     ;IN TEST 18 AT LOW BAUD RATES 1 REP IS
3545                                     ;EXPECTED.)
3546 012502          30$:
3547 012502          ERRSOFT 5,EMS3,ERRG4 ;REPORT SOFT ERROR
3548 (4) 012502 104457          TRAP   C$ERSOFT
3549 (5) 012504 000005          .WORD 5
3550 (5) 012506 012532          .WORD EMS3
3551 (5) 012510 015464          .WORD ERRG4
3552 012512          40$:
3553 012512 005737 002370      TST     NESTPC      ;IS THE ROUTINE NESTED?
3554 012516 001002          BNE     45$        ;BR IF YES
3555 012520 005037 002366      CLR     SUBRPC    ;CLEAR SAVED PC
3556 012524          45$:
3557 012524 012601          MOV     (SP)+,R1  ;RESTORE R1
3558 012526 012600          MOV     (SP)+,R0  ;RESTORE R0
3559 012530 000207          RETURN
    
```

3555  
 3556 012532 051105 047522 051522 EMS3: .ASCIZ /ERRORS IN BASE TABLE/  
 012540 044440 020116 040502  
 012546 042523 052040 041101  
 012554 042514 000  
 012560 .EVEN

3557  
 3558  
 3559  
 3560  
 3561  
 3562  
 3563  
 3564  
 3565  
 3566  
 3567  
 3568  
 3569  
 3570  
 3571  
 3572  
 3573  
 3574  
 3575  
 3576  
 3577

```

*****
*****
SUBROUTINE $HALT
FUNCTION - TO SHUTDOWN THE DMR11
ENTRY CONDITIONS - NONE
EXIT CONDITIONS - DMR SHUTDOWN
REGISTERS - NO EFFECT
*****
*****
$HALT:
  
```

3578 012560  
 3579 012560 011637 002366  
 3580 012564 162737 000004 002366  
 3581 012572 112777 000042 167432  
 3582 012600 105077 167430  
 3583 012604 012737 000001 002370  
 3584 012612  
 (1)  
 (1) 012612 004737 010266  
 (1) 012616 000000  
 (1)  
 3585 012620  
 (2) 012620 103430  
 3586 012622  
 (1)  
 (1) 012622 004737 010702  
 (1)  
 3587 012626  
 (2) 012626 103425  
 3588 012630  
 (1)  
 (1) 012630 004737 010266  
 (1) 012634 000001  
 (1)  
 3589 012636  
 (2) 012636 103421  
 3590 012640 032777 000001 167366  
 3591 012646 001005  
 3592 012650  
 (4) 012650 104455

```

MOV (SP),SUBRPC ;SAVE THE PC WHEN THE SUBROUTINE WAS CALLED.
SUB #4,SUBRPC ;BACK UP TO THE ADDRESS OF THE ACTUAL CALL.
MOVB #RQI!HLT,@SELO ;ISSUE A HALT
CLRB @SEL2 ;CLEAR ANY OUTPUT PENDING
MOV #1,NESTPC ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT RDI ;WAIT FOR RDI
;**** MACRO EXPANSION ****
JSR PC,$WAIT ;CALL WAIT ROUTINE
.WORD 0 ;FLAG THAT WE'RE WAITING FOR RDI
;****
BERROR 20$ ;IF ERROR, EXIT
WAIT RQI ;CLEAR RQI AND WAIT FOR RDI TO CLEAR BCS 20$
;**** MACRO EXPANSION ****
JSR PC,$CLRQI ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
;****
BERROR 20$ ;IF ERROR, EXIT
WAIT RDO ;WAIT FOR RDO BCS 20$
;**** MACRO EXPANSION ****
JSR PC,$WAIT ;CALL WAIT ROUTINE
.WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
;****
BERROR 20$ ;IF ERROR, EXIT
BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT? BCS 20$
BNE 10$ ;IF YES - PROCEED
ERRDF 4,EMS4,ERRG2 ;ERROR TRAP C$ERDF
  
```

```

(5) 012652 000004
(5) 012654 012722 .WORD 4
(5) 012656 015070 .WORD EMS4
3593 012660 000410 .WORD ERRG2
3594 012662
3595 012662 032777 001000 167350 10$: BR 20$
3596 012670 001004 BIT #HALTC,@SEL6 ;IS THE DMR HALTED?
3597 012672 ERRDF 4,EMS4,ERRG2 ;IF YES - EXIT
(4) 012672 104455 ;ERROR - NOT EXPECTED CONTROL OUT.
(5) 012674 000004 TRAP C$ERDF
(5) 012676 012722 .WORD 4
(5) 012700 015070 .WORD EMS4
3598 012702 .WORD ERRG2
3599 012702 042777 000207 167324 20$: BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND COMMAND BITS.
3600 012710 005037 002370 CLR NESTPC ;CLEAR THE NEST FLAG
3601 012714 005037 002366 CLR SUBRPC ;CLEAR THE PC.
3602 012720 000207 RETURN
3603
3604 012722 044123 052125 047504 EMS4: .ASCIZ /SHUTDOWN ERROR/
012730 047127 042440 051122
012736 051117 000
3605 012742 .EVEN
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625 012742
3626 012742 005077 167264
3627 012746 113777 002405 167260
3628 012754 052777 001000 167250
3629 012762 012777 121053 167250
3630
3631
3632
3633 012770 052777 000400 167234
3634 012776 042777 001400 167226
3635 013004 042737 000377 013060
3636 013012 153737 002404 013060
3637 013020 052777 001000 167204
3638 013026 013777 013060 167204
3639
    
```

```

*****
*****
SUBROUTINE $ROMO
FUNCTION - TO READ THE CONTENTS OF THE ROM
ENTRY CONDITIONS - ROMADR = ROM ADDRESS
EXIT CONDITIONS - BSEL6 = CONTENTS OF ROM ADDRESS
REGISTERS - NO EFFECT
*****
*****
    
```

```

$ROMO:
CLR @SELO ;INIT
MOVB ROMADR+1,@SEL2 ;SET HIGH BYTE OF ROM ADDRESS
BIS #ROMI,@SELO ;ENABLE SEL6 TO BE USED AS MAINTENANCE REG.
MOV #121053,@SEL6 ;SET UP MICROINSTRUCTION TO
;MOVE IBUS* 2 TO OBUS* 13
;(OBUS* 13 IS A SHADOW REGISTER FOR
;BITS 8-11 OF THE PC)
BIS #STUP,@SELO ;CLOCK THE INSTRUCTION
BIC #ROMI!STUP,@SELO ;CLEAR
BIC #377,1$ ;CLEAR ADDRESS FIELD OF BRANCH INST.
BISB ROMADR,1$ ;ADD ADDRESS OF BRANCH.
BIS #ROMI,@SELO ;ENABLE SEL6
MOV 1$,@SEL6 ;SET UP MICROINSTRUCTION TO
;BRANCH IMMEDIATELY TO PC. BRANCH IS
    
```

```

3640
3641 013034 052777 000400 167170      BIS      #STUP,@SELO      ;NECESSARY TO TRANSFER PC SHADOW REG TO PC
3642                                     ;CLOCK THE INSTRUCTION
3643 013042 042777 001400 167162      BIC      #ROMI!STUP,@SELO ;ROM PC = ROM ADDRESS
3644 013050 052777 002000 167154      BIS      #ROMO,@SELO      ;CLEAR
3645                                     ;CLOCK IN A MAINTENANCE ROM OUT
3646 013056 000207                                     ;ROM CONTENTS ARE NOW IN SEL6.
3647                                     RETURN
3648 013060 100000      1$:      .WORD 100000      ;MICRO INSTRUCTION OPCODE FOR IMMEDIATE
3649                                     ;BRANCH (ROM ADDRESS IS ADDED INTO BITS 0-7)
3650

```

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

SUBROUTINE \$LOOP

FUNCTION - TO ISSUE AN EXTENDED CONTROL IN TO SET UP THE MODEM LOOPBACK DESIRED BY THE USER.

ENTRY CONDITIONS - WMAINT = 0 - DON'T WRITE MAINT. BITS  
 WMAINT = 1 - SET BITS  
 (WMAINT SET IN INIT CODE)  
 DMCMD = 0 - DMR MODE  
 DMTURN = TURN AROUND CONNECTOR

EXIT CONDITIONS -  
 REGISTERS - NOT DESTROYED

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

```

3672 013062 005737 002276      $LOOP:  TST      DMCMD      ;IS THE DMR IN DMC MODE?
3673 013062 001041                                     ;IF SO, EXIT (CAN'T DO DMR MODE INPUT)
3674 013066 005737 002306      TST      WMAINT      ;DO WE NEED TO WRITE THE MAINTENANCE BITS?
3675 013070 001436      BEQ      30$          ;IF NOT - EXIT.
3676 013074 011637 002366      MOV      (SP),SUBRPC  ;SAVE THE PC AFTER THE CALL TO $LOOP
3677 013102 162737 000004 002366      SUB      #4,SUBRPC    ;BACKUP TO THE PC OF THE ACTUAL CALL.
3678 013110 022737 000006 002254      CMP      #6,DMTURN    ;IS LOCAL MODEM LOOPBACK DESIRED?
3679 013116 001007      BNE      10$          ;IF NOT - PROCEED.
3680 013120 012737 000004 013166      MOV      #MAINT2,100$ ;ENSURE REMOTE LOOPBACK IS CLEAR.
3681 013126 012737 000010 013170      MOV      #MAINT1,101$ ;SET MAINT BIT FOR LOCAL LOOPBACK
3682 013134 000406      BR      20$          ;IN ALL OTHER LOOPBACK CONFIGURATIONS
3683 013136                                     ;SET MAINTENANCE 2 (CONFIG. TYPE 1,3,7)
3684 013136      10$:      MOV      #MAINT1,100$ ;ENSURE REMOTE LOOPBACK IS CLEAR.
3685                                     ;SET MAINT BIT FOR REMOTE LOOPBACK
3686 013136 012737 000010 013166      MOV      #MAINT2,101$
3687 013144 012737 000004 013170      20$:      MOV      #1,NESTPC    ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
3688 013152 012737 000001 002370      CALL    $DMRIN      ;DMR MODE INPUT COMMAND
3689 013152 000005      .WORD  WMODEM      ;WRITE MODEM COMMAND
3690 013160 000000      .WORD  0            ;BITS TO CLEAR IN MODEM REGISTER
3691 013164 000000      100$:   .WORD  0
3692 013166 000000      101$:   .WORD  0            ;BITS TO SET IN MODEM REGISTER
3693 013170 000000
3694
3695

```

3696 013172  
 3697 013172 005037 002370  
 3698 013176 005037 002366  
 3699 013202 000207

30\$:

CLR NESTPC ;CLEAR THE NEST FLAG  
 CLR SUBRPC ;CLEAR PC.  
 RETURN

3700  
3701  
3702  
3703  
3704  
3705  
3706  
3707  
3708  
3709  
3710  
3711  
3712  
3713  
3714  
3715  
3716  
3717  
3718  
3719  
3720  
3721  
3722  
3723  
3724  
3725  
3726  
3727  
3728  
3729  
3730  
3731  
3732  
3733  
3734  
3735  
3736  
3737

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

SUBROUTINE \$BUFFS

FUNCTION - TO DETERMINE BUFFERS FOR TEST 15 - 19. THIS  
 SUBROUTINE WILL USE ONE OF THE FOLLOWING  
 THREE BUFFER AREAS:

1. IF MEMORY MANAGED, 32K - 48K
2. FREE MEMORY, IF MORE THAN 4K BYTES.
3. IF 2 OR 3 NOT POSSIBLE, DEFAULT 4K  
 DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.

CALL - JSR PC,\$BUFFS  
 NESTING LEVEL - CALLED ONLY BY TESTS 16-20  
 ENTRY CONDITIONS - BUFNUM = # OF RCV & XMIT BUFFERS  
 EXIT CONDITIONS - MMANAG = 1 MEMORY MANAGEMENT USED  
                   MMANAG = 0 MEMORY MANAGEMENT NOT USED  
                   RCVBUF = ADDRESS OF RECEIVE BUFFER (VIRTUAL)  
                   RCVBUF+2 = CHARACTER COUNT  
                   RCVBUF+4 = ADDRESS OF NEXT RECEIVE BUFFER  
                               (UP TO 64 ADDRESSES AND COUNTS)  
                   XMTBUF = ADDRESS OF TRANSMIT BUFFER (VIRTUAL)  
                               (UP TO 64 ADDRESSES AND COUNTS)

REGISTERS - NOT DESTROYED

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

3738 013204  
 3739 013204 011637 002366  
 3740 013210 162737 000004 002366  
 3741 013216 005037 002340  
 3742 013222  
 (7) 013222 012746 000340  
 (6) 013226 012746 023512  
 (5) 013232 012746 000004  
 (4) 013236 012746 000003  
 (3) 013242 104437  
 (2) 013244 062706 000010  
 3743 013250 005737 177572  
 3744 013254 005737 002340  
 3745

\$BUFFS:  
 MOV (SP),SUBRPC ;SAVE PC AFTER THE CALL TO \$BUFFS  
 SUB #4,SUBRPC ;BACKUP TO THE PC OF THE CALL.  
 CLR FLAG  
 SETVEC #4,#NOXMEM,#PRI07 ;SET UP TRAP 4 (WILL SET FLAG FOR NXM)  
                                   MOV #PRI07,-(SP)  
                                   MOV #NOXMEM,-(SP)  
                                   MOV #4,-(SP)  
                                   MOV #3,-(SP)  
                                   TRAP C\$SVEC  
                                   ADD #10,SP  
 TST @#177572 ;ADDRESS MEMORY MANAGEMENT REG  
 TST FLAG ;IS THE FLAG STILL CLEARED?  
 ;NOTE: THE FLAG WILL BE SET BY THE TRAP

```

3746
3747 013260 001143      BNE  30$
3748 013262 023727 002120 003000  CMP  L$HIMEM,#3000
3749
3750 013270 002537      BLT  30$
3751 013272 012737 000001 002302  MOV  #1,M$MANAG
3752 013300          SETPRI #P$RI07
    (3) 013300 012700 000340
    (3) 013304 104441
3753
3754
3755 013306 012701 172300      MOV  #172300,R1
3756 013312 012700 000010      MOV  #8.,R0
3757 013316
3758 013316 012721 077406      MOV  #77406,(R1)+
3759
3760 013322 005300      DEC  R0
3761 013324 001374      BNE  10$
3762 013326 012701 172340      MOV  #172340,R1
3763 013332 005011      CLR  (R1)
3764 013334 012761 000200 000002  MOV  #200,2(R1)
3765 013342 012761 000400 000004  MOV  #400,4(R1)
3766 013350 012761 002000 000006  MOV  #2000,6(R1)
3767 013356 012761 002200 000010  MOV  #2200,10(R1)
3768 013364 012761 002400 000012  MOV  #2400,12(R1)
3769 013372 012761 002600 000014  MOV  #2600,14(R1)
3770 013400 012761 007600 000016  MOV  #7600,16(R1)
3771
3772 013406 012703 000400      MOV  #256.,R3
3773 013412 012704 060000      MOV  #60000,R4
3774
3775
3776 013416 012737 000001 177572  MOV  #1,@#177572
3777 013424
3778 013424 012701 000040      MOV  #32.,R1
3779 013430 012702 002410      MOV  #$CCITT,R2
3780 013434
3781 013434 012224
3782
3783 013436 005737 002340      TST  FLAG
3784 013442 001050      BNE  29$
3785 013444 005301      DEC  R1
3786 013446 001372      BNE  16$
3787 013450 005303      DEC  R3
3788 013452 001364      BNE  15$
3789 013454 012701 020000      MOV  #20000,R1
3790 013460
3791 013460 005024
3792 013462 005737 002340      CLR  (R4)+
3793 013466 001036      TST  FLAG
3794 013470 005301      BNE  29$
3795 013472 001372      DEC  R1
3796 013474 005037 177572      BNE  17$
3797
3798 013500 012737 060000 003230  MOV  #60000,XMITBUF
3799 013506 012737 120000 003630  MOV  #120000,RCVBUF
  
```

```

:IF THERE IS NO MEMORY MANAGEMENT.
:BR TO USE NON-MEMORY MANAG. BUFFERS.
:IS THERE AT LEAST 48K WORDS? (16K WORDS
:FOR BUFFERS)
:IF NOT, USE NON-MEMORY MANAG. BUFFERS.
:FLAG THAT MEMORY MANAGEMENT IS USED
:MAKE SURE WE ARE IN KERNEL MODE.
      MOV  #P$RI07,R0
      TRAP C$SPRI
:SETTING PRI SHOULD SHOULD ALSO CLEAR
:BITS 14 & 15
:GET ADDRESS OF KERNEL PDR REG 0.
:WRITE PDR REG 0-7.
:WRITE BITS FOR THE FOLLOWING PAGE DESCRIPTION
:READ/WRITE ACCESS, 128. BLOCK PAGE LENGTH.
:WRITE ALL PDRS.
:ADDRESS OF KERNEL PAR 0
:PAR 0, ADDRS 0 - 17776
:PAR 1, ADDRS 20000 - 37776
:PAR 2, ADDRS 40000 - 57776
:PAR 3, ADDRS 200000 - 217776 (BUFFER PAGE 1)
:PAR 4, ADDRS 220000 - 237776 (BUFFER PAGE 2)
:PAR 5, ADDRS 240000 - 257776 (BUFFER PAGE 3)
:PAR 6, ADDRS 260000 - 277776 (BUFFER PAGE 4)
:PAR 7, ADDRS 160000 - 677776 (I/O PAGE)
:COUNTER FOR OUTER LOOP OF TEST PATTERN
:USE VIRTUAL ADDRESS TO MAP TO PAR 5
:GENERATE A TEST PATTERN IN THE 1ST 8K WORDS
:VIRTUAL ADDRESS 60000 - 111776
:ENABLE MEMORY MANAGEMENT
:COUNTER FOR INNER LOOP OF TEST PATTERN
:ADDRESS FOR 32. WORD TEST PATTERN.
:WRITE TEST PATTERN
:PHYSICAL ADDRESS 200000 - 237776
:FLAG WILL BE SET IF WE ADDRESS NXM.
:IF NXM - DON'T USE MEMORY MANAGEMENT.
:DO TH INNER LOOP 32. TIMES
:DO THE OUTER LOOP 256. TIMES
:COUNTER TO CLEAR THE NEXT 8K WORDS
:CLEAR VIRTUAL ADDRESS 120000 - 157776
:DOES AN NXM TRAP OCCUR?
:IF SO DON'T USE MEMORY MANAGEMENT.
:TURN OFF MEMORY MANAGEMENT
:VIRTUAL ADDRESS OF XMIT BUFFER
:VIRTUAL ADDRESS OF RCV. BUFFER
  
```

```

3800 013514 022737 000001 002322      CMP      #1,BUFNUM      ;IS THERE ONLY 1 XMIT & RECEIVE BUFFER?
3801 013522 001004                      BNE      20$           ;IF NOT, BR
3802 013524 012737 037777 002320      MOV      #37777,BUFSIZ ;EACH BUFFER IS 16K BYTES
3803 013532 000525                      BR       60$
3804 013534                      20$:
3805 013534 022737 000007 002322      CMP      #7,BUFNUM      ;ARE THERE 7 XMIT & RECEIVE (14 TOTAL BUFFER)?
3806 013542 001004                      BNE      21$           ;IF NOT - MUST BE 64 BUFFERS
3807 013544 012737 004000 002320      MOV      #4000,BUFSIZ  ;EACH BUFFER IS 2K BYTES
3808 013552 000515                      BR       60$
3809 013554                      21$:
3810 013554 012737 000376 002320      MOV      #376,BUFSIZ   ;EACH BUFFER IS 254. BYTES.
3811 013562 000511                      BR       60$
3812
3813 013564                      29$:
3814 013564 005037 177572                      CLR      @#177572      ;TURN OFF MEMORY MANAGEMENT
3815 013570                      30$:
3816 013570 005037 002302                      CLR      MMANAG       ;FLAG THAT MEMORY MANAGEMENT NOT USED.
3817 013574 012700 000004                      CLRVEC   #4           ;RESTORE TRAP 4.
(3) 013574 012700 000004
(3) 013600 104436
3818 013602                      MEMORY R2            ;FIND THE FREE MEMORY AVAILABLE BETWEEN
(3) 013602 104431
(3) 013604 010002
3819
3820 013606 021227 002000                      CMP      @R2,#2000     ;THE DIAGNOSTIC AND THE DRS (SUPERVISOR).
3821
3822
3823 013612 003406                      BLE      35$           ;IS THERE AT LEAST 1K WORDS? (NOTE: CONTENTS
3824 013614 010237 003230                      MOV      R2,XMTBUF     ;OF THE RETURNED ADDRESS OF THE START OF FREE
3825 013620 011200                      MOV      @R2,R0        ;MEMORY CONTAIN THE AMOUNT OF AVAILABLE MEM.)
3826 013622 042700 000001                      BIC      #BIT0,R0      ;IF NOT AT LEAST 1K, USE DEFAULT BUFFER.
3827 013626 000405                      BR       40$           ;USE THE FREE MEMORY BUFFER.
3828 013630                      35$:
3829 013630 012737 004230 003230      MOV      #BIGBUF,XMTBUF ;SAVE THE WORD SIZE OF THE BUFFER.
3830 013636 012700 002000      MOV      #2000,R0      ;START WITH AN EVEN # OF WORDS.
3831 013642                      40$:
3832 013642 013737 003230 003630      MOV      XMTBUF,RCVBUF ;CALCULATE THE RECEIVE BUFFER ADDRESS
3833 013650 060037 003630      ADD      R0,RCVBUF     ;AS STARTING IN THE 2ND HALF OF THE BUFFER.
3834 013654 010001                      MOV      R0,R1         ;BUFFER SIZE IN WORDS.
3835 013656 022737 000001 002322      CMP      #1,BUFNUM     ;ARE WE SETTING UP 1 RECEIVE AND XMIT BUFFER?
3836 013664 001415                      BEQ      47$           ;IF YES - R1 = BYTE SIZE FOR BOTH BUFFERS.
3837 013666 022737 000007 002322      CMP      #7,BUFNUM     ;ARE WE SETTING UP 7 RCV & 7 XMIT BUFFERS?
3838 013674 001004                      BNE      45$           ;IF NOT WE MUST NEED 64 RCV & 64 XMIT BUFFERS.
3839 013676 006201                      ASR      R1            ;R1 = # BYTES IN THE BUFFERS/8
3840 013700 006201                      ASR      R1
3841 013702 006201                      ASR      R1
3842 013704 000405                      BR       47$
3843 013706                      45$:
3844 013706 012704 000007      MOV      #7,R4         ;DIVIDE BYTES BY 128.
3845 013712                      46$:
3846 013712 006201                      ASR      R1            ;SHIFT RIGHT 7 TIMES
3847 013714 005304                      DEC      R4
3848 013716 001375                      BNE      46$
3849 013720                      47$:
3850 013720 010137 002320      MOV      R1,BUFSIZ     ;SAVE THE BUFFER SIZE IN BYTES.
3851 013724 162737 000002 002320      SUB      #2,BUFSIZ     ;ADJUST BUFFER SIZE BECAUSE WE
    
```





3908 014124 005037 002340  
 3909 014130 005037 002366  
 3910 014134 000207  
 3911  
 3912  
 3913  
 3914  
 3915  
 3916  
 3917  
 3918  
 3919  
 3920  
 3921  
 3922  
 3923  
 3924  
 3925  
 3926  
 3927  
 3928  
 3929  
 3930  
 3931  
 3932  
 3933  
 3934  
 3935  
 3936 014136  
 3937 014136 011637 002366  
 3938 014142 162737 000004 002366  
 3939 014150 012737 000001 002370  
 3940 014156 013737 002322 002324  
 3941 014164 013737 002322 002326  
 3942 014172 013737 002322 002330  
 3943 014200 013737 002322 002332  
 3944 014206 005037 002346  
 3945 014212 005037 002350  
 3946 014216 005037 002272  
 3947  
 3948 014222 012702 003630  
 3949 014226 012703 003230  
 3950 014232 012704 003630  
 3951 014236 012705 003230  
 3952 014242  
 (3) 014242 012700 000200  
 (3) 014246 104441  
 3953  
 3954 014250 013737 002314 002316  
 3955 014256 112777 000143 165746  
 3956 014264  
 3957 014264 012701 001000  
 3958 014270  
 3959 014270  
 (3) 014270 104422  
 3960

CLR FLAG ;RESTORE FLAG USED IN TRAP VECTOR.  
 CLR SUBRPC ;CLEAR PC.  
 RETURN

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

SUBROUTINE \$INOUT

FUNCTION - TO MANAGE THE INTERRUPT FROM BASE IN  
 TO BA/CC OUT IN THE INTERRUPT TESTS 15-19

ENTRY CONDITIONS - BUFNUM = # OF RCV AND XMIT BUFFERS  
 ALL BUFFERS SET UP IN THE \$BUFFS SUBROUTINE.  
 WAIT3 = # OF OUTER LOOP TIMEOUT COUNTERS.  
 THIS VALUE IS DETERMINED BY THE BAUD  
 RATE IN THE INIT. SECTION OF CODE.

EXIT CONDITIONS -

REGISTERS - R0 - R5 DESTROYED

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

\$INOUT:

MOV (SP),SUBRPC ;SAVE THE PC AFTER THE CALL TO \$LOOP  
 SUB #4,SUBRPC ;BACKUP TO THE PC OF THE ACTUAL CALL.  
 MOV #1,NESTPC ;FLAG THAT ANY SUBROUTINE USED WILL BE NESTED.  
 MOV BUFNUM,INRCV ;# OF BA/CC IN RECEIVES  
 MOV BUFNUM,INXMIT ;# OF BA/CC IN TRANSMITS  
 MOV BUFNUM,OUTRCV ;# OF BA/CC OUT RECEIVES  
 MOV BUFNUM,OUTXMT ;# OF BA/CC OUT TRANSMITS  
 CLR INFLAG ;CLEAR INPUT BA/CC FLAG  
 CLR OUTFLG ;CLEAR OUTPUT BA/CC FLAG  
 CLR START ;CLEAR FLAG TO SHOW START UP NOT DONE (SET  
 ;AFTER CONTROL IN)  
 MOV #RCVBUF,R2 ;ADDR OF RCV. BUFFER TABLE (FOR INPUT)  
 MOV #XMTBUF,R3 ;ADDR OF XMIT BUFFER TABLE (FOR INPUT)  
 MOV #RCVBUF,R4 ;ADDR OF RCV. BUFFER TABLE (OUTPUT CHECKING)  
 MOV #XMTBUF,R5 ;ADDR OF XMIT BUFFER TABLE (OUTPUT CHECKING)  
 SETPRI #PRI04 ;SET THE PRIORITY TO LEVEL 4 TO ALLOW THE  
 ;DMR TO INTERRUPT AT LEVEL 5  
 MOV WAIT3,WAIT4 ;TIMEOUT COUNTER DETERMINED BY BAUD RATE.  
 MOV #IESET!RQI!BASEI,ABSELO ;FIRST COMMAND - BASE IN.  
 8\$: MOV #1000,R1 ;INNER LOOP COUNTER  
 10\$: BREAK ;OPERATOR INTERRUPT ENABLE. CALL TO  
 ;THE SUPERVISOR TO ALLOW CONSOLE INTERRUPT

```

3961
3962
3963 014272 005737 002346
3964 014276 001403
3965 014300 005737 002350
3966 014304 001026
3967 014306
3968 014306
(2) 014306 012727 000001
(2) 014312 000000
(2) 014314 013727 002116
(2) 014320 000000
(2) 014322 005367 177772
(2) 014326 001375
(2) 014330 005367 177756
(2) 014334 001367
3969 014336 005301
3970 014340 001353
3971 014342 005337 002316
3972 014346 001346
3973 014350
(4) 014350 104455
(5) 014352 000002
(5) 014354 020025
(5) 014356 014562
3974
3975
3976 014360 000453
3977 014362
3978
3979 014362 012700 003630
3980 014366 012701 003230
3981 014372 013702 002322
3982 014376 005737 002302
3983 014402 001403
3984 014404 012737 000001 177572
3985 014412
3986 014412 012003
3987 014414 012104
3988 014416 011005
3989 014420 022021
3990 014422 001412
3991 014424 005737 002302
3992 014430 001402
3993 014432 005037 177572
3994 014436
3995 014436
(4) 014436 104455
(5) 014440 000014
(5) 014442 020241
(5) 014444 016160
3996 014446 000420
3997 014450
3998 014450 122324
3999 014452 001005
4000 014454 005305

; (NOTE: INFLAG AND OUTFLG SET IN THE INTERRUPT
; SERVICE ROUTINES)
; ARE THE INPUTS DONE? (INISR DONE?)
; IF NOT KEEP CHECKING.
; ARE THE OUTPUTS DONE? (OUTISR DONE?)
; IF YES EXIT WAIT LOOP.

12$:
DELAY 1
; WAIT 100 MICROSECONDS.

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

DEC R1
BNE 10$
DEC WAIT4
BNE 8$
ERRDF 2,EMG2,ERRG1
; CONTINUE IN LOOP UNTIL R1 = 0.
; DECREMENT OUTER LOOP COUNTER
; IF NOT DONE - GO THROUGH INNER LOOP AGAIN.
; TIMEOUT MESSAGE.

TRAP C$ERDF
.WORD 2
.WORD EMG2
.WORD ERRG1

; ALSO PRINT # OF BUFFERS NOT COMPLETE.

20$:
BR 60$
; EXIT

MOV #RCVBUF,R0
MOV #XMTBUF,R1
MOV BUFNUM,R2
TST MMANAG
BEQ 40$
MOV #1,@#177572
; RECEIVE BUFFER POINTER TABLE ADDRESS.
; TRANSMIT BUFFERS
; # OF RCV. AND XMIT BUFFERS.
; ARE THE BUFFERS MEMORY MANAGED?
; IF YES - PROCEED.
; TURN ON MEMORY MANAGEMENT

40$:
MOV (R0)+,R3
MOV (R1)+,R4
MOV @R0,R5
CMP (R0)+,(R1)+
BEQ 45$
TST MMANAG
BEQ 41$
CLR @#177572
; ADDRESS OF A RECEIVE BUFFER.
; ADDRESS OF A TRANSMIT BUFFER.
; CHARACTER COUNT.
; ARE THE CHARACTER COUNTS THE SAME?
; IF YES - PROCEED.
; IS MEMORY MANAGEMENT TURNED ON?
; IF NOT - SKIP TURN OFF.
; TURN OFF MEMORY MANAGEMENT.

41$:
ERRDF 12,EMG12,ERRG10
TRAP C$ERDF
.WORD 12
.WORD EMG12
.WORD ERRG10

45$:
BR 60$
; EXIT

CMPB (R3)+,(R4)+
BNE 50$
DEC R5
; ARE THE CHARACTERS THE SAME?
; IF NOT - ERROR EXIT
; CHECK ALL THE CHARACTERS

```

```

4001 014456 001374      BNE      45$
4002 014460 005302      DEC      R2          ;CHECK ALL THE BUFFERS.
4003 014462 001353      BNE      40$
4004 014464 000411      BR       60$
4005 014466
4006 014466 005737 002302      50$:    TST      MMANAG      ;IS MEMORY MANAGEMENT TURNED ON?
4007 014472 001402      BEQ      51$          ;IF NOT - SKIP TURN OFF.
4008 014474 005037 177572      CLR      @#177572    ;TURN OFF MEMORY MANAGEMENT.
4009 014500
4010 014500      51$:    ERRDF   15,EMG15,ERRG12
      (4) 014500 104455
      (5) 014502 000017      TRAP    C$ERDF
      (5) 014504 020341      .WORD  15
      (5) 014506 016244      .WORD  EMG15
4011 014510      60$:    TST      MMANAG      ;IS MEMORY MANAGEMENT TURNED ON?
4012 014510 005737 002302      BEQ      61$          ;IF NOT - SKIP TURN OFF.
4013 014514 001402      CLR      @#177572    ;TURN OFF MEMORY MANAGEMENT.
4014 014516 005037 177572
4015 014522
4016 014522 042777 000120 165502      61$:    BIC      #IESET!IECLR,@SELO ;DISABLE BOTH INPUT INTERRUPTS
4017 014530 042777 000100 165476      BIC      #IEO,@SEL2  ;DISABLE OUTPUT INTERRUPT
4018
4019 014536      SHUTDN      ;SHUT DOWN THE DMR
      (1)
      (1) 014536 004737 012560      JSR      PC, $HALT  ;**** MACRO EXPANSION ****
      (1)
      ;DMR HALT ROUTINE.
4020 014542      SETPRI   #PRI07    ;****
      ;RETURN PROCESSOR PRIORITY TO 7
      (3) 014542 012700 000340      MOV     #PRI07,R0
      (3) 014546 104441      TRAP   C$SPRI
4021 014550 005037 002370
4022 014554 005037 002366      CLR     NESTPC      ;CLEAR NESTED FLAG.
4023 014560 000207      CLR     SUBRPC      ;CLEAR PC.
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035      .SBTTL  GLOBAL ERROR REPORT REPORT SECTION
4036      ;////////////////////////////////////////////////////
4037      ;/ THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
4038      ;/ THAT ARE USED IN MORE THAN ONE TEST.
4039      ;////////////////////////////////////////////////////
4040      .EVEN
4041
4042 014562      BGNMSG  ERRG1
      (3) 014562
4043 014562      PRINTB #FMG3,SUBRPC ;PC THAT SUBROUTINE WAS CALLED.
      (8) 014562 013746 002366      ERRG1::
      (7) 014566 012746 016362      MOV    SUBRPC,-(SP)
      (6) 014572 012746 000002      MOV    #FMG3,-(SP)
      MOV    #2,-(SP)
    
```

(3)	014576	010600				MOV	SP,R0
(4)	014600	104414				TRAP	C\$PNTB
(4)	014602	062706	000006			ADD	#6,SP
4044	014606				PRINTB	#FMG1,@SEL0,@SEL2 ;PRINT SEL0 AND SEL2 CONTENTS.	
(9)	014606	017746	165422			MOV	@SEL2,-(SP)
(8)	014612	017746	165414			MOV	@SEL0,-(SP)
(7)	014616	012746	016276			MOV	#FMG1,-(SP)
(6)	014622	012746	000003			MOV	#3,-(SP)
(3)	014626	010600				MOV	SP,R0
(4)	014630	104414				TRAP	C\$PNTB
(4)	014632	062706	000010			ADD	#10,SP
4045	014636				PRINTB	#FMG2,@SEL4,@SEL6 ;PRINT SEL4 AND SEL2 CONTENTS.	
(9)	014636	017746	165376			MOV	@SEL6,-(SP)
(8)	014642	017746	165370			MOV	@SEL4,-(SP)
(7)	014646	012746	016330			MOV	#FMG2,-(SP)
(6)	014652	012746	000003			MOV	#3,-(SP)
(3)	014656	010600				MOV	SP,R0
(4)	014660	104414				TRAP	C\$PNTB
(4)	014662	062706	000010			ADD	#10,SP
4046	014666				PRINTB	#FMG21,BUFNUM ;# OF BUFFERS	
(8)	014666	013746	002322			MOV	BUFNUM,-(SP)
(7)	014672	012746	017545			MOV	#FMG21,-(SP)
(6)	014676	012746	000002			MOV	#2,-(SP)
(3)	014702	010600				MOV	SP,R0
(4)	014704	104414				TRAP	C\$PNTB
(4)	014706	062706	000006			ADD	#6,SP
4047	014712				PRINTB	#FMG22,BUFSIZ ;BUFFER SIZE	
(8)	014712	013746	002320			MOV	BUFSIZ,-(SP)
(7)	014716	012746	017625			MOV	#FMG22,-(SP)
(6)	014722	012746	000002			MOV	#2,-(SP)
(3)	014726	010600				MOV	SP,R0
(4)	014730	104414				TRAP	C\$PNTB
(4)	014732	062706	000006			ADD	#6,SP
4048	014736	005437	002324			NEG	INRCV ;NEGATE BUFFER VALUES
4049	014742	005437	002326			NEG	INXMIT
4050	014746	005437	002330			NEG	OUTRCV
4051	014752	005437	002332			NEG	OUTXMT
4052	014756	063737	002322	002324		ADD	BUFNUM,INRCV ;CALCULATE BUFFERS ASSIGNED.
4053	014764	063737	002322	002326		ADD	BUFNUM,INXMIT
4054	014772	063737	002322	002330		ADD	BUFNUM,OUTRCV ;CALCULATE BUFFERS RECEIVED.
4055	015000	063737	002322	002332		ADD	BUFNUM,OUTXMT
4056	015006				PRINTB	#FMG23,INRCV,INXMIT	
(9)	015006	013746	002326			MOV	INXMIT,-(SP)
(8)	015012	013746	002324			MOV	INRCV,-(SP)
(7)	015016	012746	017652			MOV	#FMG23,-(SP)
(6)	015022	012746	000003			MOV	#3,-(SP)
(3)	015026	010600				MOV	SP,R0
(4)	015030	104414				TRAP	C\$PNTB
(4)	015032	062706	000010			ADD	#10,SP
4057	015036				PRINTB	#FMG24,OUTRCV,OUTXMT	
(9)	015036	013746	002332			MOV	OUTXMT,-(SP)
(8)	015042	013746	002330			MOV	OUTRCV,-(SP)
(7)	015046	012746	017733			MOV	#FMG24,-(SP)
(6)	015052	012746	000003			MOV	#3,-(SP)
(3)	015056	010600				MOV	SP,R0
(4)	015060	104414				TRAP	C\$PNTB

```

(4) 015062 062706 000010
4058 015066          ENDMSG          ADD      #10,SP
(3) 015066          L10002:          TRAP     C$MSG
(3) 015066 104423
4059
4060
4061 015070          BGNMSG  ERRG2
(3) 015070          .
4062 015070 005737 002366          TST      SUBRPC          ;IS THE ERROR IN A SUBROUTINE?
4063 015074 001412          BEQ      10$              ;IF NOT, DON'T PRINT SUBR. PC
4064 015076          PRINTB  #FMG3,SUBRPC      ;PC THAT SUBROUTINE WAS CALLED.
(8) 015076 013746 002366          MOV      SUBRPC,-(SP)
(7) 015102 012746 016362          MOV      #FMG3,-(SP)
(6) 015106 012746 000002          MOV      #2,-(SP)
(3) 015112 010600          MOV      SP,R0
(4) 015114 104414          TRAP    C$PNTB
(4) 015116 062706 000006          ADD      #6,SP
4065 015122          10$:
4066 015122          PRINTB  #FMG1,@SEL0,@SEL2 ;PRINT SEL0 AND SEL2 CONTENTS.
(9) 015122 017746 165106          MOV      @SEL2,-(SP)
(8) 015126 017746 165100          MOV      @SEL0,-(SP)
(7) 015132 012746 016276          MOV      #FMG1,-(SP)
(6) 015136 012746 000003          MOV      #3,-(SP)
(3) 015142 010600          MOV      SP,R0
(4) 015144 104414          TRAP    C$PNTB
(4) 015146 062706 000010          ADD      #10,SP
4067 015152          PRINTB  #FMG2,@SEL4,@SEL6 ;PRINT SEL4 AND SEL2 CONTENTS.
(9) 015152 017746 165062          MOV      @SEL6,-(SP)
(8) 015156 017746 165054          MOV      @SEL4,-(SP)
(7) 015162 012746 016330          MOV      #FMG2,-(SP)
(6) 015166 012746 000003          MOV      #3,-(SP)
(3) 015172 010600          MOV      SP,R0
(4) 015174 104414          TRAP    C$PNTB
(4) 015176 062706 000010          ADD      #10,SP
4068 015202          ENDMSG
(3) 015202          L10003:          TRAP     C$MSG
(3) 015202 104423
4069
4070 015204          BGNMSG  ERRG3
(3) 015204          .
4071 015204 005737 002366          TST      SUBRPC          ;IS THE ERROR IN A SUBROUTINE?
4072 015210 001412          BEQ      10$              ;IF NOT, DON'T PRINT SUBR. PC
4073 015212          PRINTB  #FMG3,SUBRPC      ;PC THAT SUBROUTINE WAS CALLED.
(8) 015212 013746 002366          MOV      SUBRPC,-(SP)
(7) 015216 012746 016362          MOV      #FMG3,-(SP)
(6) 015222 012746 000002          MOV      #2,-(SP)
(3) 015226 010600          MOV      SP,R0
(4) 015230 104414          TRAP    C$PNTB
(4) 015232 062706 000006          ADD      #6,SP
4074 015236          10$:
4075 015236          PRINTB  #FMG1,@SEL0,@SEL2 ;PRINT SEL0 AND SEL2 CONTENTS.
(9) 015236 017746 164772          MOV      @SEL2,-(SP)
(8) 015242 017746 164764          MOV      @SEL0,-(SP)
(7) 015246 012746 016276          MOV      #FMG1,-(SP)
(6) 015252 012746 000003          MOV      #3,-(SP)
(3) 015256 010600          MOV      SP,R0
  
```

Line	Address	Code	Label	Comment	Trap	Target
(4)	015260	104414			TRAP	C\$PNTB
(4)	015262	062706	000010		ADD	#10,SP
4076	015266	032777	100000	164736		
4077	015274	001043				
4078	015276	122777	000001	164740		
4079	015304	001011				
4080	015306					
(7)	015306	012746	016434		MOV	#FMG4,-(SP)
(6)	015312	012746	000001		MOV	#1,-(SP)
(3)	015316	010600			MOV	SP,R0
(4)	015320	104414			TRAP	C\$PNTB
(4)	015322	062706	000004		ADD	#4,SP
4081	015326	000455				
4082	015330					
4083	015330	122777	000002	164706		
4084	015336	001011				
4085	015340					
(7)	015340	012746	016465		MOV	#FMG5,-(SP)
(6)	015344	012746	000001		MOV	#1,-(SP)
(3)	015350	010600			MOV	SP,R0
(4)	015352	104414			TRAP	C\$PNTB
(4)	015354	062706	000004		ADD	#4,SP
4086	015360	000440				
4087	015362					
4088	015362					
(7)	015362	012746	016465		MOV	#FMG5,-(SP)
(6)	015366	012746	000001		MOV	#1,-(SP)
(3)	015372	010600			MOV	SP,R0
(4)	015374	104414			TRAP	C\$PNTB
(4)	015376	062706	000004		ADD	#4,SP
4089	015402	000427				
4090	015404					
4091	015404	105777	164634			
4092	015410	001010				
4093	015412					
(7)	015412	012746	017473		MOV	#FMG19,-(SP)
(6)	015416	012746	000001		MOV	#1,-(SP)
(3)	015422	010600			MOV	SP,R0
(4)	015424	104414			TRAP	C\$PNTB
(4)	015426	062706	000004		ADD	#4,SP
4094	015432					
4095	015432	122777	000100	164604		
4096	015440	001010				
4097	015442					
(7)	015442	012746	017516		MOV	#FMG20,-(SP)
(6)	015446	012746	000001		MOV	#1,-(SP)
(3)	015452	010600			MOV	SP,R0
(4)	015454	104414			TRAP	C\$PNTB
(4)	015456	062706	000004		ADD	#4,SP
4098	015462					
4099	015462					
(3)	015462					
(3)	015462	104423				
4100						
4101						
4102	015464					

12\$:

15\$:

20\$:

25\$:

30\$:  
ENDMSG

L10004:

BGNMSG ERRG4

```

(3) 015464
4103 015464 005737 002366
4104 015470 001412
4105 015472
(8) 015472 013746 002366
(7) 015476 012746 016362
(6) 015502 012746 000002
(3) 015506 010600
(4) 015510 104414
(4) 015512 062706 000006
4106 015516
4107 015516 105737 002633
4108 015522 001003
4109 015524 105737 002636
4110 015530 001416
4111 015532
4112 015532
(9) 015532 005046
(9) 015534 153716 002636
(8) 015540 005046
(8) 015542 153716 002633
(7) 015546 012746 016557
(6) 015552 012746 000003
(3) 015556 010600
(4) 015560 104414
(4) 015562 062706 000010
4113 015566
4114 015566 105737 002635
4115 015572 001003
4116 015574 105737 002640
4117 015600 001416
4118 015602
4119 015602
(9) 015602 005046
(9) 015604 153716 002640
(8) 015610 005046
(8) 015612 153716 002635
(7) 015616 012746 016630
(6) 015622 012746 000003
(3) 015626 010600
(4) 015630 104414
(4) 015632 062706 000010
4120 015636
4121 015636 105737 002634
4122 015642 001003
4123 015644 105737 002637
4124 015650 001416
4125 015652
4126 015652
(9) 015652 005046
(9) 015654 153716 002637
(8) 015660 005046
(8) 015662 153716 002634
(7) 015666 012746 016701
(6) 015672 012746 000003
(3) 015676 010600
ERRG4::
;IS THE ERROR IN A SUBROUTINE?
;IF NOT, DON'T PRINT SUBR. PC
;PC THAT SUBROUTINE WAS CALLED.
MOV SUBRPC, -(SP)
MOV #FMG3, -(SP)
MOV #2, -(SP)
MOV SP, R0
TRAP C$PNTB
ADD #6, SP
10$:
TSTB BASE+3 ;ONLY PRINT NON-ZERO VALUES
BNE 11$
TSTB BASE+6
BEQ 12$
11$:
PRINTB #FMG7, <B, BASE+3>, <B, BASE+6>
CLR -(SP)
BISB BASE+6, (SP)
CLR -(SP)
BISB BASE+3, (SP)
MOV #FMG7, -(SP)
MOV #3, -(SP)
MOV SP, R0
TRAP C$PNTB
ADD #10, SP
12$:
TSTB BASE+5
BNE 13$
TSTB BASE+8.
BEQ 14$
13$:
PRINTB #FMG8, <B, BASE+5>, <B, BASE+8.>
CLR -(SP)
BISB BASE+8., (SP)
CLR -(SP)
BISB BASE+5, (SP)
MOV #FMG8, -(SP)
MOV #3, -(SP)
MOV SP, R0
TRAP C$PNTB
ADD #10, SP
14$:
TSTB BASE+4
BNE 15$
TSTB BASE+7
BEQ 16$
15$:
PRINTB #FMG9, <B, BASE+4>, <B, BASE+7>
CLR -(SP)
BISB BASE+7, (SP)
CLR -(SP)
BISB BASE+4, (SP)
MOV #FMG9, -(SP)
MOV #3, -(SP)
MOV SP, R0
    
```





(6)	016062	012746	000001						
(3)	016066	010600							MOV #1,-(SP)
(4)	016070	104414							MOV SP,R0
(4)	016072	062706	000004						TRAP C\$PNTB
4147	016076			PRINTB	#FMG13,@SEL4,@SEL6	;ACTUAL BA/CC			ADD #4,SP
(9)	016076	017746	164136						MOV @SEL6,-(SP)
(8)	016102	017746	164130						MOV @SEL4,-(SP)
(7)	016106	012746	017073						MOV #FMG13,-(SP)
(6)	016112	012746	000003						MOV #3,-(SP)
(3)	016116	010600							MOV SP,R0
(4)	016120	104414							TRAP C\$PNTB
(4)	016122	062706	000010						ADD #10,SP
4148	016126			PRINTB	#FMG14,-4(R5),-2(R5)	;EXPECTED BA/CC			
(9)	016126	016546	177776						MOV -2(R5),-(SP)
(8)	016132	016546	177774						MOV -4(R5),-(SP)
(7)	016136	012746	017147						MOV #FMG14,-(SP)
(6)	016142	012746	000003						MOV #3,-(SP)
(3)	016146	010600							MOV SP,R0
(4)	016150	104414							TRAP C\$PNTB
(4)	016152	062706	000010						ADD #10,SP
4149	016156			ENDMSG					
(3)	016156								
(3)	016156	104423						L10007:	TRAP C\$MSG
4150									
4151									
4152	016160			BGNMSG	ERRG10				
(3)	016160								
4153	016160				PRINTB	#FMG16,-2(R0),-2(R1)	;RCV CC & XMIT CC	ERRG10::	
(9)	016160	016146	177776						MOV -2(R1),-(SP)
(8)	016164	016046	177776						MOV -2(R0),-(SP)
(7)	016170	012746	017252						MOV #FMG16,-(SP)
(6)	016174	012746	000003						MOV #3,-(SP)
(3)	016200	010600							MOV SP,R0
(4)	016202	104414							TRAP C\$PNTB
(4)	016204	062706	000010						ADD #10,SP
4154	016210			ENDMSG					
(3)	016210								
(3)	016210	104423						L10010:	TRAP C\$MSG
4155									
4156	016212			BGNMSG	ERRG11				
(3)	016212								
4157	016212				PRINTB	#FMG17,-4(R0),-4(R1)	;RCV BUFFER & XMIT BUFFER	ERRG11::	
(9)	016212	016146	177774						MOV -4(R1),-(SP)
(8)	016216	016046	177774						MOV -4(R0),-(SP)
(7)	016222	012746	017331						MOV #FMG17,-(SP)
(6)	016226	012746	000003						MOV #3,-(SP)
(3)	016232	010600							MOV SP,R0
(4)	016234	104414							TRAP C\$PNTB
(4)	016236	062706	000010						ADD #10,SP
4158	016242			ENDMSG					
(3)	016242								
(3)	016242	104423						L10011:	TRAP C\$MSG
4159									
4160	016244			BGNMSG	ERRG12				
(3)	016244								
4161	016244	005303			DEC R3			ERRG12::	
									;BACKUP TO RECEIVE ADDRESS

```

4162 016246 005304
4163 016250
(9) 016250 010446
(8) 016252 010346
(7) 016254 012746 017404
(6) 016260 012746 000003
(3) 016264 010600
(4) 016266 104414
(4) 016270 062706 000010
4164 016274
(3) 016274
(3) 016274 104423
4165
4166
4167
4168
4169
4170
4171 016276 040445 042523 030114 FMG1: .ASCIZ /%ASELO: %06%A SEL2: %06%/
      016304 020072 047445 022466
      016312 020101 042523 031114
      016320 020072 047445 022466
      016326 000116
4172 016330 040445 042523 032114 FMG2: .ASCIZ /%ASEL4: %06%A SEL6: %06%/
      016336 020072 047445 022466
      016344 020101 042523 033114
      016352 020072 047445 022466
      016360 000116
4173 016362 040445 051105 047522 FMG3: .ASCIZ /%AERROR IN SUBROUTINE CALLED AT PC: %06%/
      016370 020122 047111 051440
      016376 041125 047522 052125
      016404 047111 020105 040503
      016412 046114 042105 040440
      016420 020124 041520 020072
      016426 047445 022466 000116
4174 016434 040445 050103 020125 FMG4: .ASCIZ /%ACPU MICROTTEST FAILED%/
      016442 044515 051103 052117
      016450 051505 020124 040506
      016456 046111 042105 047045
      016464 000
4175 016465 045 046101 027125 FMG5: .ASCIZ /%ALU. MICROTTEST FAILED%/
      016472 046440 041511 047522
      016500 042524 052123 043040
      016506 044501 042514 022504
      016514 000116
4176 016516 040445 047516 051040 FMG6: .ASCIZ /%AND RUN - MASTER CLEAR FAILED%/
      016524 047125 026440 046440
      016532 051501 042524 020122
      016540 046103 040505 020122
      016546 040506 046111 042105
      016554 047045 000
4177 016557 045 047101 045501 FMG7: .ASCIZ /%ANAKS-NO BUFFER RCV: %D3%A SENT: %D3%/
      016564 026523 047516 041040
      016572 043125 042506 020122
      016600 051040 053103 020072
      016606 042045 022463 020101

```

DEC R4 ;BACKUP TO TRANSMIT ADDRESS  
 PRINTB #FMG18,R3,R4 ;PRINT OUT ADDRESS

MOV R4,-(SP)  
 MOV R3,-(SP)  
 MOV #FMG18,-(SP)  
 MOV #3,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #10,SP

L10012:

TRAP C\$MSG

ENDMSG

	016614	042523	052116	020072	
	016622	042045	022463	000116	
4178	016630	040445	040516	051513	FMG8: .ASCIZ /%ANAKS-BAD DATA RCV: %D3%A SENT: %D3%N/
	016636	041055	042101	042040	
	016644	052101	020101	020040	
	016652	041522	035126	022440	
	016660	031504	040445	051440	
	016666	047105	035124	022440	
	016674	031504	047045	000	
4179	016701	045	047101	045501	FMG9: .ASCIZ /%ANAKS-BAD HEADER RCV: %D3%A SENT: %D3%N/
	016706	026523	040502	020104	
	016714	042510	042101	051105	
	016722	051040	053103	020072	
	016730	042045	022463	020101	
	016736	042523	052116	020072	
	016744	042045	022463	000116	
4180	016752	040445	042522	051520	FMG10: .ASCIZ /%AREPS-RCV: %D3%A SENT: %D3%N/
	016760	051055	053103	020072	
	016766	042045	022463	020101	
	016774	042523	052116	020072	
	017002	042045	022463	000116	
4181	017010	040445	046530	052111	FMG11: .ASCIZ /%AXMIT BACC OUT COMMAND%N/
	017016	041040	041501	020103	
	017024	052517	020124	047503	
	017032	046515	047101	022504	
	017040	000116			
4182	017042	040445	041522	020126	FMG12: .ASCIZ /%ARCV BACC OUT COMMAND%N/
	017050	040502	041503	047440	
	017056	052125	041440	046517	
	017064	040515	042116	047045	
	017072	000			
4183	017073	045	040501	052103	FMG13: .ASCIZ /%AACTUAL ADDR. %06%A ACTUAL COUNT %D5%N/
	017100	040525	020114	020040	
	017106	042101	051104	020056	
	017114	047445	022466	020101	
	017122	041501	052524	046101	
	017130	041440	052517	052116	
	017136	020040	022440	032504	
	017144	047045	000		
4184	017147	045	042501	050130	FMG14: .ASCIZ /%AEXPECTED ADDR. %06%A EXPECTED COUNT %D5%N/
	017154	041505	042524	020104	
	017162	042101	051104	020056	
	017170	047445	022466	020101	
	017176	054105	042520	052103	
	017204	042105	041440	052517	
	017212	052116	022440	032504	
	017220	047045	000		
4185	017223	045	042501	050130	FMG15: .ASCIZ /%AEXPECTED ADDR. %06%N/
	017230	041505	042524	020104	
	017236	042101	051104	020056	
	017244	047445	022466	000116	
4186	017252	040445	041522	020126	FMG16: .ASCIZ /%ARCV CHAR. COUNT %D5%A XMIT CHAR. COUNT %D5%N/
	017260	044103	051101	020056	
	017266	047503	047125	020124	
	017274	042045	022465	020101	
	017302	046530	052111	041440	

	017310	040510	027122	041440	
	017316	052517	052116	022440	
	017324	032504	047045	000	
4187	017331	045	051101	053103	FMG17: .ASCIZ /%ARCV BUFFER AT %06%A XMIT BUFFER AT %06%N/
	017336	041040	043125	042506	
	017344	020122	052101	022440	
	017352	033117	040445	054040	
	017360	044515	020124	052502	
	017366	043106	051105	040440	
	017374	020124	047445	022466	
	017402	000116			
4188	017404	040445	040504	040524	FMG18: .ASCIZ /%ADATA DIFFERS AT RCV ADDR. %06%A AND XMIT ADDR. %06%N/
	017412	042040	043111	042506	
	017420	051522	040440	020124	
	017426	041522	020126	042101	
	017434	051104	020056	047445	
	017442	022466	020101	047101	
	017450	020104	046530	052111	
	017456	040440	042104	027122	
	017464	022440	033117	047045	
	017472	000			
4189	017473	045	042101	053105	FMG19: .ASCIZ /%ADEVICE NOT DMR%N/
	017500	041511	020105	047516	
	017506	020124	046504	022522	
	017514	000116			
4190	017516	040445	044515	051103	FMG20: .ASCIZ /%AMICROTEST DISABLED%N/
	017524	052117	051505	020124	
	017532	044504	040523	046102	
	017540	042105	047045	000	
4191	017545	045	051501	040524	FMG21: .ASCIZ /%ASTATUS OF BUFFERS %N%NUMBER OF BUFFERS:%D3%N/
	017552	052524	020123	043117	
	017560	041040	043125	042506	
	017566	051522	022440	022516	
	017574	047101	046525	042502	
	017602	020122	043117	041040	
	017610	043125	042506	051522	
	017616	022472	031504	047045	
	017624	000			
4192	017625	045	041101	043125	FMG22: .ASCIZ /%ABUFFER SIZE: %D5%N/
	017632	042506	020122	044523	
	017640	042532	020072	042045	
	017646	022465	000116		
4193	017652	040445	047111	020040	FMG23: .ASCIZ /%AIN - RCV ASSIGNED:%D3%A XMIT ASSIGNED:%D3%N/
	017660	020055	041522	020126	
	017666	051501	044523	047107	
	017674	042105	022472	031504	
	017702	040445	020040	054040	
	017710	044515	020124	051501	
	017716	044523	047107	042105	
	017724	022472	031504	047045	
	017732	000			
4194	017733	045	047501	052125	FMG24: .ASCIZ /%AOUT - RCV RETURNED:%D3%A XMIT RETURNED:%D3%N/
	017740	026440	051040	053103	
	017746	051040	052105	051125	
	017754	042516	035104	042045	
	017762	022463	020101	020040	

	017770	046530	052111	051040		
	017776	052105	051125	042516		
	020004	035104	042045	022463		
	020012	000116				
4195						
4196						
4197	020014	044524	042515	047440	EMG1:	.ASCIZ /TIME OUT/
	020022	052125	000			
4198	020025	124	046511	020105	EMG2:	.ASCIZ /TIME OUT - DURING INTERRUPT EXERCISE/
	020032	052517	020124	020055		
	020040	052504	044522	043516		
	020046	044440	052116	051105		
	020054	052522	052120	042440		
	020062	042530	041522	051511		
	020070	000105				
4199	020072	054105	042520	052103	EMG8:	.ASCIZ /EXPECTED CONTROL OUT - NOT RECEIVED/
	020100	042105	041440	047117		
	020106	051124	046117	047440		
	020114	052125	026440	047040		
	020122	052117	051040	041505		
	020130	044505	042526	000104		
4200	020136	047125	054105	042520	EMG9:	.ASCIZ /UNEXPECTED CONTROL OUT/
	020144	052103	042105	041440		
	020152	047117	051124	046117		
	020160	047440	052125	000		
4201	020165	105	051122	051117	EMG10:	.ASCIZ /ERROR - MULTIPLE XMIT/
	020172	026440	046440	046125		
	020200	044524	046120	020105		
	020206	046530	052111	000123		
4202	020214	052502	043106	051105	EMG11:	.ASCIZ /BUFFER ADDRESS ERROR/
	020222	040440	042104	042522		
	020230	051523	042440	051122		
	020236	051117	000			
4203						
4204	020241	103	040510	040522	EMG12:	.ASCIZ /CHARACTER COUNT ERROR/
	020246	052103	051105	041440		
	020254	052517	052116	042440		
	020262	051122	051117	000		
4205	020267	105	051122	051117	EMG13:	.ASCIZ /ERROR - MULTIPLE RCVS/
	020274	026440	046440	046125		
	020302	044524	046120	020105		
	020310	041522	051526	000		
4206	020315	122	041505	044505	EMG14:	.ASCIZ /RECEIVED EXTRA DATA/
	020322	042526	020104	054105		
	020330	051124	020101	040504		
	020336	040524	000			
4207	020341	104	052101	020101	EMG15:	.ASCIZ /DATA ERROR/
	020346	051105	047522	000122		
4208	020354	047125	054105	042520	EMG16:	.ASCIZ /UNEXPECTED HALT RECEIVED/
	020362	052103	042105	044040		
	020370	046101	020124	042522		
	020376	042503	053111	042105		
	020404	000				
4209	020405	103	047117	051124	EMG17:	.ASCIZ /CONTROL IN PROBLEM - IN INTERRUPT ROUTINE/
	020412	046117	044440	020116		
	020420	051120	041117	042514		

020426 020115 020055 047111  
 020434 044440 052116 051105  
 020442 052522 052120 051040  
 020450 052517 044524 042516  
 020456 000  
 4210 020457 123 052520 044522  
 020464 052517 020123 042122  
 020472 020117 047111 042524  
 020500 051122 050125 000124

EMG18: .ASCIZ /SPURIOUS RDO INTERRUPT/

.EVEN

4211

4212

4213

4214

4215

4216

4217

4218

4219

4220

4221

4222

4223

4224

4225

4226

4227

4228

4229

4230

4231

4232

4233

4234

4235

4236

4237

4238

4239

4240

4241

4242

4243

4244

4245

4246

4247

4248

4249

4250

4251

4252

4253

.SBTTL LOAD DEVICE PROTECTION TABLE

://  
 :/ THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE  
 :/ PROTECTED FROM TESTING. IF DESIRED.  
 ://

BGNPROT

L\$PROT::

.WORD -1  
 .WORD -1  
 .WORD -1

:DON'T CHECK CSR ADDRESS  
 :DON'T CHECK MASSBUS UNIT NUMBER  
 :DON'T CHECK DRIVE NUMBER

ENDPROT

.SBTTL INITIALIZE SECTION

://  
 :/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED  
 :/ AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.  
 ://

BGNINIT

L\$INIT::

SETPRI #PRI07

:SET DIAGNOSTIC PRIORITY = 7

MOV SP,PSTACK  
 CLR SUBRPC  
 CLR ERROR  
 CLR RESUME  
 CLR DMCMD

MOV #PRI07,RO  
 TRAP C\$SPRI  
 :STORE BASE LEVEL PROGRAM STACK POINTER  
 :CLEAR STORAGE WORD FOR SUBROUTINE PC CALL  
 :CLEAR ERROR FLAGS  
 :CLEAR FLAG USED TO ALLOW BASE IN - RESUME.  
 :CLEAR FLAG USED TO INDICATE DMC MODE

```

4254 020546 005037 002372          CLR      CLRNO          ;CLEAR WORD USED TO RUN MICRO TESTS ON
4255                                     ;EVERY OTHER MASTER CLEAR.
4256 020552 005737 002264          TST      FRSTIM         ;IS THIS THE TIME THROUGH AFTER LOAD?
4257 020556 001005                   BNE      1$             ;IF NOT - ERROR TRAP VECTOR ALREADY SAVED
4258 020560 012737 000001 002264   MOV      #1,FRSTIM     ;FLAG THAT WE'VE BEEN THRU THE 1ST TIME
4259 020566 005037 002266          CLR      FRSPAS        ;CLEAR COUNTER FOR # OF PASSES AFTER LOAD
4260
4261 020572                               1$:
4262 020572                               CLRVEC  #4             ;ENSURE VECTOR 4 IS IN NORMAL STATE.
(3) 020572 012700 000004          MOV      #4,R0
(3) 020576 104436          TRAP    C$CVEC
4263
4264 020600          READEF  #EF.START     ;IS THIS JUST STARTED?
(3) 020600 012700 000040          MOV      #EF.START,R0
(3) 020604 104447          TRAP    C$REFG
4265 020606          BCOMPLETE STARST     ;IF YES - BRANCH.
(2) 020606 103416          BCS     STARST
4266 020610          READEF  #EF.RESTART   ;IS THIS A RESTART ?
(3) 020610 012700 000037          MOV      #EF.RESTART,R0
(3) 020614 104447          TRAP    C$REFG
4267 020616          BCOMPLETE STARST     ;IF YES - BRANCH.
(2) 020616 103412          BCS     STARST
4268 020620          READEF  #EF.NEW       ;IS THIS A NEW PASS?
(3) 020620 012700 000035          MOV      #EF.NEW,R0
(3) 020624 104447          TRAP    C$REFG
4269 020626          BCOMPLETE NEWST      ;IF YES - BRANCH
(2) 020626 103410          BCS     NEWST
4270 020630          READEF  #EF.CONTINUE  ;IS THIS A CONTINUATION?
(3) 020630 012700 000036          MOV      #EF.CONTINUE,R0
(3) 020634 104447          TRAP    C$REFG
4271 020636          BNCOMPLETE GETPRM    ;IF NOT - GET PARAMETERS
(2) 020636 103013          BCC     GETPRM
4272 020640 000137 021510          JMP      END           ;OTHERWISE - DON'T INITIALIZE.
4273
4274 020644          STARST:
4275 020644 005037 002270          CLR      STARES       ;CLEAR THE FLAG TO SHOW START/RESTART.
4276
4277 020650          NEWST:
4278 020650 012737 177777 002362   MOV      #-1,LOGDEV    ;INITIALIZE LOGICAL UNIT NUMBER.
4279 020656 005237 002266          INC      FRSPAS        ;INCREMENT # OF PASSES AFTER LOAD.
4280 020662 005237 002270          INC      STARES       ;INCREMENT # OF PASSES SINCE START/RESTART.
4281 020666          GETPRM:
4282 020666 005237 002362          INC      LOGDEV        ;NEXT LOGICAL UNIT TO BE TESTED
4283 020672 023737 002362 002012   CMP      LOGDEV,LSUNIT ;IS THE MAXIMUM UNIT # EXCEEDED?
4284 020700 002363          BGE     NEWST         ;IF YES - DO A NEW START
4285 020702          GPHARD  LOGDEV,R1    ;GET THE P-TABLE POINTER INTO R1
(3) 020702 013700 002362          MOV      LOGDEV,R0
(3) 020706 104442          TRAP    C$GPHRD
(3) 020710 010001          MOV      R0,R1
4286 020712          BNCOMPLETE GETPRM    ;IF NOT AVAILABLE, GET THE NEXT ONE
(2) 020712 103365          BCC     GETPRM
4287 020714 012137 002252          MOV      (R1)+,WTYPE   ;MICROPROCESSOR TYPE
4288 020720 011100          MOV      (R1),R0      ;SAVE THE ADDRESS
4289 020722 032700 000007          BIT      #7,R0        ;DOES THIS DEVICE ADDRESS END IN NON-ZERO?
4290 020726 001414          BEQ     10$           ;IF NOT - OK (76XXX0)
4291 020730 042711 000007          BIC     #7,(R1)       ;MAKE IT 76XXX0
  
```

```

4292 020734          PRINTB #FINIT1,(R1),R0 ;INFORM THE USER
(9) 020734 010046
(8) 020736 011146
(7) 020740 012746 021512
(6) 020744 012746 000003
(3) 020750 010600
(4) 020752 104414
(4) 020754 062706 000010
4293 020760          10$:
4294 020760 011137 002232      MOV (R1),CSR ;CSR ADDRESS
4295 020764 011137 002242      MOV (R1),BSEL1
4296 020770 005237 002242      INC BSEL1 ;HIGH BYTE ADDRESS OF CSR
4297 020774 011137 002234      MOV (R1),SEL2
4298 021000 062737 000002 002234  ADD #2,SEL2 ;CONTROL OUT REGISTER ADDRESS
4299 021006 011137 002244      MOV (R1),BSEL3
4300 021012 062737 000003 002244  ADD #3,BSEL3 ;HIGH BYTE OF SEL2
4301 021020 011137 002236      MOV (R1),SEL4
4302 021024 062737 000004 002236  ADD #4,SEL4 ;PORT REG (SEL 4) ADDRESS
4303 021032 011137 002246      MOV (R1),BSEL5
4304 021036 062737 000005 002246  ADD #5,BSEL5 ;HIGH BYTE OF SEL4
4305 021044 011137 002240      MOV (R1),SEL6
4306 021050 062737 000006 002240  ADD #6,SEL6 ;PORT REG (SEL 6) ADDRESS
4307 021056 012137 002250      MOV (R1)+,BSEL7
4308 021062 062737 000007 002250  ADD #7,BSEL7 ;HIGH BYTE OF SEL6
4309 021070 011100
4310 021072 032700 000007      MOV (R1),R0 ;GET VECTOR
4311 021076 001414
4312 021100 042711 000007      BIT #7,R0 ;DOES THIS VECTOR END IN NON-ZERO?
4313 021104          BEQ 11$ ;IF NOT - OK (XX0)
(9) 021104 010046          BIC #7,(R1) ;MAKE IT XX0
(8) 021106 011146          PRINTB #FINIT2,(R1),R0 ;INFORM THE USER
(7) 021110 012746 021605
(6) 021114 012746 000003
(3) 021120 010600
(4) 021122 104414
(4) 021124 062706 000010
4314 021130          11$:
4315 021130 011137 002226      MOV (R1),DMRVEC ;RCV. VECTOR
4316 021134 011137 002230      MOV (R1),DMTVEC ;TRANSMIT VECTOR
4317 021140 011100      MOV (R1),R0 ;RCV. VECTOR
4318 021142 105060 000003      CLRB 3(R0) ;CLEAR HI BYTE OF PSW FOR RCV. VECTOR.
4319 021146 105060 000007      CLRB 7(R0) ;CLEAR HI BYTE OF PSW FOR XMIT. VECTOR.
4320
4321 ;THIS WILL ENSURE THAT WE DON'T PICK
4322 ;UP ANY UNEXPECTED BITS IN PROCESSORS
4323 ;WHICH USE BITS 11-15 OF THE PSW. IE
4324 ;IF BIT 11 IS SET IN AN 11/70 ANOTHER
4325 021152 062737 000004 002230  ADD #4,DMTVEC ;REGISTER SET MAY BE USED.
4326 ;ADJUST XMIT VECTOR
4327 ;SET UP ISRS FOR DMR. INTERRUPTS ENABLED IN
4328 ;TESTS 15-19.
4329
4330 SETVEC DMRVEC,#INISR,#PRI05 ;INPUT ISR
(7) 021160 012746 000240
(6) 021164 012746 021772
(5) 021170 013746 002226

```

```

MOV R0,-(SP)
MOV (R1),-(SP)
MOV #FINIT1,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP

```

```

MOV R0,-(SP)
MOV (R1),-(SP)
MOV #FINIT2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP

```

```

MOV #PRI05,-(SP)
MOV #INISR,-(SP)
MOV DMRVEC,-(SP)

```



```

(4) 021174 012746 000003
(3) 021200 104437
(2) 021202 062706 000010
4331 021206                SETVEC DMTVEC,#OUTISR,#PRI05 ;OUTPUT ISR
(7) 021206 012746 000240
(6) 021212 012746 023054
(5) 021216 013746 002230
(4) 021222 012746 000003
(3) 021226 104437
(2) 021230 062706 000010
4332
4333 021234 062701 000014      ADD    #14,R1          ;INCR. P-TABLE POINTER.
4334 021240 012137 002254      MOV    (R1)+,DMTURN   ;TURNAROUND
4335
4336
4337 021244 013700 002224      MOV    SPEED,R0      ;GET THE SOFTWARE P-TABLE VALUE GIVEN
4338                               ;BY THE USER
4339
4340
4341 021250                13$:
4342 021250 012701 000002      MOV    #2,R1          ;GET FIRST TIMER VALUE
4343 021254 012702 000012      MOV    #10.,R2       ;GET SECOND TIMER VALUE
4344 021260                14$:
4345 021260 006301          ASL    R1              ; TIMER VALUES X 2
4346 021262 006302          ASL    R2
4347 021264 005300          DEC    R0              ; DEC SPEED VARIABLE
4348 021266 001374          BNE    14$            ; CONTINUE UNTIL DONE INCREASING WAIT VALUES
4349
4350 021270 010137 002310      MOV    R1,WAIT1       ;SAVE TIMER VALUE FOR $WAIT
4351 021274 006201          ASR    R1              ;HALF THAT VALUE
4352 021276 006201          ASR    R1              ;HALF IT AGAIN.
4353 021300 010137 002312      MOV    R1,WAIT2       ;SAVE TIMER VALUE FOR $MSCLR AND $CLRQI
4354                               ;SUBROUTINES.
4355 021304 010237 002314      MOV    R2,WAIT3       ;TIMER VALUE FOR $INOUT SUBROUTINE.
4356
4357
4358 021310 012737 000333 002304  MOV    #333,AX3       ;CHECK TURNAROUND.
4359                               ;MASK FOR AX3-15 - BIT CLEARED WILL
4360 021316 022737 000001 002254  CMP    #1,DMTURN      ;BE THE INTERFACE SELECTED.
4361 021324 001004          BNE    20$            ;IS V.35 REQUESTED?
4362 021326 042737 000020 002304  BIC    #BIT4,AX3      ;IF NOT - CONTINUE
4363 021334 000427          BR     30$            ;SELECT V.35
4364 021336                20$:
4365 021336 022737 000002 002254  CMP    #2,DMTURN      ;IS INTEGRAL REQUESTED?
4366 021344 001004          BNE    22$            ;IF NOT - CONTINUE.
4367 021346 042737 000010 002304  BIC    #BIT3,AX3      ;SELECT INTEGRAL MODEM.
4368 021354 000417          BR     30$
4369 021356                22$:
4370 021356 022737 000003 002254  CMP    #3,DMTURN      ;IS EIA REQUESTED?
4371 021364 001004          BNE    25$            ;IF NOT - CONTINUE.
4372 021366 042737 000100 002304  BIC    #BIT6,AX3      ;SELECT EIA(XYZ).
4373 021374 000407          BR     30$
4374 021376                25$:
4375 021376 022737 000004 002254  CMP    #4,DMTURN      ;IS RS422 REQUESTED?
4376 021404 001007          BNE    35$            ;IF NOT, DON'T ALLOW INTERFACE CHANGE.
4377 021406 042737 000200 002304  BIC    #BIT7,AX3      ;SELECT RS422.

```

```
4378 021414 30$:
4379 021414 012737 000001 002262 MOV #1,INFACE ;SET FLAG THAT ALLOWS INTERFACE CHANGE.
4380 021422 000404 BR 40$
4381 021424 35$:
4382 021424 005037 002262 CLR INFACE ;CLEAR FLAG - NO INTERFACE CHANGE.
4383 021430 005037 002304 CLR AX3 ;CLEAR AX3 BITS
4384 021434 40$:
4385 021434 022737 000001 002254 CMP #1,DMTURN ;IS THIS V.35?
4386 021442 001414 BEQ 45$ ;IF YES - SET WRITE MAINT. BIT
4387 021444 022737 000003 002254 CMP #3,DMTURN ;IS THIS EIA?
4388 021452 001410 BEQ 45$ ;IF YES - SET WRITE MAINT. BIT
4389 021454 022737 000006 002254 CMP #6,DMTURN ;IS THIS LOCAL LOOP?
4390 021462 001404 BEQ 45$ ;IF YES - SET WRITE MAINT. BIT.
4391 021464 022737 000007 002254 CMP #7,DMTURN ;IS THIS REMOTE LOOP?
4392 021472 001004 BNE 50$ ;IF NOT - CLEAR MAINT. BIT FLAG
4393 021474 45$:
4394 021474 012737 000001 002306 MOV #1,WMAINT ;SET FLAG TO WRITE MODEM MAINTENANCE BITS.
4395 021502 000402 BR END
4396 021504 50$:
4397 021504 005037 002306 CLR WMAINT ;CLEAR FLAG - DON'T WRITE MAINT. 1 OR 2.
4398 021510 END:
4399 021510 ENDINIT
(3) 021510 L10014:
(3) 021510 104411 TRAP C$INIT
4400 021512 040445 025052 053440 FINIT1: .ASCIZ /%A** WARNING - WILL ASSUME DMR ADDRESS %06% (NOT %06%)%N/
021520 051101 044516 043516
021526 026440 053440 046111
021534 020114 051501 052523
021542 042515 042040 051115
021550 040440 042104 042522
021556 051523 022440 033117
021564 040445 024040 047516
021572 020124 047445 022466
021600 024501 047045 000
4401 021605 045 025101 020052 FINIT2: .ASCIZ /%A** WARNING - WILL ASSUME DMR VECTOR %03% (NOT %03%)%N/
021612 040527 047122 047111
021620 020107 020055 044527
021626 046114 040440 051523
021634 046525 020105 046504
021642 020122 042526 052103
021650 051117 020040 047445
021656 022463 020101 047050
021664 052117 022440 031517
021672 040445 022451 000116
4402 .EVEN
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412 .SBTTL AUTO DROP UNIT SECTION
4413
```

```

4414 :////////////////////////////////////////////////////////////////////
4415 :// THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
4416 :// WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
4417 :////////////////////////////////////////////////////////////////////
4418
4419 021700 BGNAUTO
(3) 021700
4420
4421 021700 SETVEC #4,#NOXMEM,#PRI07 ;SET UP NON -EXISTENT MEMORY TRAP VECTOR.
(7) 021700 012746 000340
(6) 021704 012746 023512
(5) 021710 012746 000004
(4) 021714 012746 000003
(3) 021720 104437
(2) 021722 062706 000010
4422 021726 005037 002340
4423 021732 005777 160274
4424
4425
4426
4427
4428
4429
4430
4431
4432 021736 CLRVEC #4 ;RETURN VECTOR 04 TO NORMAL STATE
(3) 021736 012700 000004
(3) 021742 104436
4433 021744 005737 002340
4434 021750 001406
4435 021752
(3) 021752 013700 002362
(3) 021756 104451
4436 021760
(3) 021760 104444
4437 021762 005037 002340
4438 021766
4439 021766
(3) 021766
(3) 021766 104461
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
    
```

.SBTTL CLEANUP CODING SECTION

```

:////////////////////////////////////////////////////////////////////
:// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED AT THE
:// END OF THE TEST SEQUENCE ON A PARTICULAR UNIT. THIS SECTION IS REQUIRED
    
```

```
4456      :/ EVEN IF IT IS A NULL CLEANUP
4457      :////////////////////////////////////////////////////////////////////
4458
4459 021770  BGNCLN
(3) 021770
4460
4461 021770  ENDCLN
(3) 021770
(3) 021770 104412
4462
4463
4464
4465
4466
4467
4468
4469
4470      .SBTTL GLOBAL INTERRUPT HANDLING ROUTINES
4471
4472      :////////////////////////////////////////////////////////////////////
4473      :/ THE INTERRUPT HANDLING SECTION CONTAINS CODING REQUIRED TO USE
4474      :/ THE 'SETVEC' MACRO. NOTE EVERY INTERRUPT ROUTINE SHOULD SAVE
4475      :/ AND RESTORE R0.
4476      :////////////////////////////////////////////////////////////////////
4477 021772  BGNSRV INISR      ;INPUT INTERRUPT SERVICE ROUTINE
(3) 021772
4478 021772 010046      MOV R0,-(SP)      ;SAVE R0
4479 021774 010146      MOV R1,-(SP)      ;SAVE R1
4480 021776 017701 160230  MOV @SELO,R1      ;SAVE THE CONTROL IN COMMAND.
4481 022002 042701 177760  BIC #177760,R1    ;CLEAR ALL BUT THE COMMAND BITS (0-3)
4482 022006 032777 000200 160216 BIT #RDI,@SELO    ;IS RDI SET
4483 022014 001002      BNE 1$           ;IF YES - PROCESS INPUT COMMAND.
4484 022016 000137 022526  JMP NEXT        ;ISSUE NEXT INPUT COMMAND.
4485
4486      :*****
4487      : PROCESS INPUT COMMAND
4488
4489      :*****
4490 022022 1$:
4491 022022 022701 000004  CMP #BACCR,R1    ;IS THIS A RCV. BA/CC?
4492 022026 001533      BEQ 29$         ;BR IF YES.
4493 022030 022701 000000  CMP #BACCT,R1    ;IS THIS A XMIT. BA/CC?
4494 022034 001537      BEQ 30$         ;BR IF YES.
4495 022036 022701 000003  CMP #BASEI,R1    ;IS THIS A BASE IN?
4496 022042 001461      BEQ 20$         ;BR IF YES.
4497 022044 022701 000001  CMP #CNTRL,R1    ;IS THIS A CONTROL IN?
4498 022050 001444      BEQ 15$         ;BR IF YES.
4499 022052 022701 000005  CMP #WMODEM,R1   ;IS THIS A WRITE MODEM?
4500 022056 001417      BEQ 10$         ;BR IF YES.
4501 022060 022701 000015  CMP #INTER,R1    ;IS THIS AN INTERFACE WRITE.
4502 022064 001410      BEQ 5$          ;BR IF YES.
4503 022066 022701 000002  CMP #HLT,R1      ;IS THIS A HALT?
4504 022072 001572      BEQ 70$         ;EXIT - IF YES (NOTHING TO SET UP)
4505 022074 104455      ERRDF 17,EMG17,ERRG2 ;PROBLEM IF IT'S NOT ONE OF THE ABOVE.
(4) 022074 104455
(5) 022076 000021
                                TRAP C$ERDF
                                .WORD 17
```

```

(5) 022100 020405
(5) 022102 015070
4506 022104 000565
4507
4508 022106
4509
4510
4511
4512 022106 113777 002304 160134
4513
4514
4515 022114 000561
4516
4517
4518 022116
4519
4520
4521
4522 022116 022737 000006 002254
4523 022124 001007
4524 022126 042777 000004 160104
4525 022134 052777 000010 160076
4526 022142 000546
4527 022144
4528 022144 042777 000010 160066
4529 022152 052777 000004 160060
4530 022160 000537
4531 022162
4532
4533
4534
4535 022162 005737 002300
4536 022166 001404
4537 022170 012777 000400 160042
4538 022176 000530
4539 022200
4540 022200 005077 160034
4541 022204 000525
4542 022206
4543
4544
4545
4546 022206 012777 002630 160022
4547
4548 022214 005737 002276
4549 022220 001004
4550 022222 012777 000522 160010
4551 022230 000402
4552 022232
4553 022232 005077 160002
4554 022236
4555 022236 005737 002272
4556 022242 001004
4557 022244 052777 000100 157762
4558 022252 000406
4559 022254

```

```

        BR      70$      ;EXIT
5$:
      ;
      ; WRITE AX3-15
      ;
      MOV      AX3,@BSEL7    ;WRITE NECESSARY AX3-15 INTERFACE.
                          ;AX3 HAS BEEN DETERMINED IN THE INIT
                          ;CODE.
        BR      70$
10$:
      ;
      ; MODEM WRITE
      ;
      CMP      #6,DMTURN    ;IS LOCAL MODEM LOOPBACK DESIRED?
      BNE     11$          ;BR IF NOT
      BIC     #MAINT2,@SEL6 ;ENSURE REMOTE LOOPBACK IS CLEAR.
      BIS     #MAINT1,@SEL6 ;SET MAINTENANCE 1 BIT
      BR      70$
11$:
      BIC     #MAINT1,@SEL6 ;ENSURE LOCAL LOOPBACK IS CLEAR.
      BIS     #MAINT2,@SEL6 ;SET MAINTENANCE 2 BIT.
      BR      70$          ;CLEAR RQI
15$:
      ;
      ; CONTROL IN
      ;
      TST     MNTMDE        ;IS MAINTENANCE MODE REQUESTED
      BEQ     17$          ;BR IF NOT
      MOV     #MAINT,@SEL6 ;REQUEST MAINT. MODE
      BR      70$
17$:
      CLR     @SEL6        ;FULL DUPLEX - NON-MAINT. MODE.
      BR      70$
20$:
      ;
      ; BASE IN
      ;
      MOV     #BASE,@SEL4   ;BASE TABLE ADDRESS.
      ;
      TST     DMCMDMDE      ;ARE WE IN DMC MODE?
      BNE     22$          ;BR IF YES
      MOV     #DMR,@SEL6   ;DMR MODE.
      BR      23$         ;CHECK LOOPBACK.
22$:
      CLR     @SEL6        ;DMC MODE
23$:
      TST     START        ;IS THIS THE FIRST BASE IN?
      BNE     24$          ;IF NOT - SET RESUME.
      BIS     #IEO,@SEL2   ;ON FIRST BASE IN SET RDO INT.ENABLE.
      BR      25$
24$:

```

.WORD EMG17  
 .WORD ERRG2

4560	022254	052777	010000	157756	BIS	#RES,@SEL6	:SET RESUME
4561	022262	012737	177777	002352	MOV	#-1,RESFLG	:FLAG THAT THIS IS A BASE IN RESUME COMMAND
4562							:(THIS WILL BE USED LATER IN THIS ISR TO
4563							:DECIDE WHAT THE NEXT COMMAND WILL BE)
4564	022270				25\$:		
4565	022270	005737	002254		TST	DMTURN	:IS INTERNAL LOOPACK REQUESTED?
4566	022274	001004			BNE	27\$	:BR IF NOT - CLEAR LU LOOP
4567	022276	052777	004000	157726	BIS	#LPLU,@SEL0	:SET THE LINE UNIT LOOPBACK BIT
4568	022304	000465			BR	70\$	:CLEAR RQI AND EXIT.
4569	022306				27\$:		
4570	022306	042777	004000	157716	BIC	#LPLU,@SEL0	:CLEAR LINE UNIT LOOPBACK (CONNECTOR OR
4571							:CABLE)
4572	022314	000461			BR	70\$	:CLEAR RQI AND EXIT
4573							
4574					:		
4575					:	BA/CC IN RCV	
4576					:		
4577	022316				29\$:		
4578	022316	005337	002324		DEC	INRCV	:DECREMENT COUNTER
4579	022322	012277	157710		MOV	(R2)+,@SEL4	:RCV BUFFER ADDRESS
4580	022326	012277	157706		MOV	(R2)+,@SEL6	:RCV CHARACTER COUNT
4581	022332	000406			BR	40\$	
4582					:		
4583					:	BA/CC IN XMIT	
4584					:		
4585	022334				30\$:		
4586	022334	005337	002326		DEC	INXMIT	:DECREMENT COUNTER
4587	022340	012377	157672		MOV	(R3)+,@SEL4	:XMIT BUFFER ADDRESS.
4588	022344	012377	157670		MOV	(R3)+,@SEL6	:XMIT CHARACTER COUNT.
4589	022350				40\$:		
4590	022350	005737	002302		TST	MMANAG	:ARE THE BUFFERS MEMORY MANAGED?
4591	022354	001441			BEQ	70\$	:IF NOT SKIP CONVERTING VIRTUAL ADDR
4592							:TO PHYSICAL ADDR.
4593	022356	052777	040000	157654	BIS	#BIT14,@SEL6	:SET BIT 16 OF PHYSICAL ADDRESS (I.E.
4594							:VIRTUAL ADDR 60000 = PHYSICAL ADDR 200000
4595	022364	010246			MOV	R2,-(SP)	:SAVE R2 (NEXT RCV BUFFER ADDRESS)
4596	022366	017702	157644		MOV	@SEL4,R2	:SAVE THE VIRTUAL ADDRESS.
4597	022372	042777	160000	157636	BIC	#160000,@SEL4	:CLEAR BITS CORRESPONDING TO THE PAGE #
4598							:IN THE VIRTUAL ADDRESS.
4599	022400	042702	017777		BIC	#17777,R2	:SAVE ONLY THE PAGE # IN THE SAVED ADDR.
4600	022404	022702	060000		CMP	#60000,R2	:IS THIS PAGE 3?
4601	022410	001421			BEQ	44\$	:IF YES, PHYSICAL ADDRESS CALCULATED
4602	022412	022702	100000		CMP	#100000,R2	:IS THIS PAGE 4?
4603	022416	001004			BNE	41\$	:IF NOT SEE IF IT'S PAGE 4 OR 5
4604	022420	052777	020000	157610	BIS	#BIT13,@SEL4	:SET BIT FOR PHYSICAL ADDR (I.E. VIRTUAL
4605							:ADDR 100000 = PHYSICAL ADDR. 220000
4606	022426	000412			BR	44\$	
4607	022430				41\$:		
4608	022430	022702	120000		CMP	#120000,R2	:IS THIS PAGE 4?
4609	022434	001004			BNE	42\$	:IF NOT, MUST BE PAGE 5.
4610	022436	052777	040000	157572	BIS	#BIT14,@SEL4	:SET BIT FOR PHYSICAL ADDR (I.E. VIRTUAL
4611							:ADDR 120000 = PHYSICAL ADDR. 240000
4612	022444	000403			BR	44\$	
4613	022446				42\$:		
4614	022446	052777	060000	157562	BIS	#BIT14!BIT13,@SEL4	:SET BIT FOR PHYSICAL ADDR (I.E. VIRTUAL
4615							:ADDR 140000 = PHYSICAL ADDR. 260000

```

4616 022454          44$:
4617 022454 012602      MOV    (SP)+,R2      ;RESTORE R2 (NEXT RCV BUFFER ADDRESS)
4618 022456 000400      BR     70$          ;CLEAR RQI AND EXIT
4619
4620
4621 022460          70$:
4622 022460 010137 002356  MOV    R1, LAST      ;SAVE THE INPUT COMMAND (USED
4623                                ;TO DETERMINE NEXT INPUT COMMAND)
4624 022464 005737 002276  TST    DMCMD E      ;ARE WE IN DMC MODE?
4625 022470 001011      BNE    80$          ;IF YES - DON'T USE IECLR
4626                                ;NOTE: INTERRUPT CAPABILITY FOR RQI
4627                                ;CLEAR IS ONLY AVAILABLE IN DMR MODE.
4628 022472 012601      MOV    (SP)+,R1      ;RESTORE R1
4629 022474 012600      MOV    (SP)+,R0      ;RESTORE R0
4630 022476 052777 000020 157526  BIS    #IECLR,@SELO ;SET INTERRUPT ENABLE FOR RDI CLEAR.
4631 022504 042777 000040 157520  BIC    #RQI,@SELO   ;CLEAR RQI - INT. GENERATED WHEN RDI
4632                                ;CLEARS IN RESPONSE.
4633 022512 000002      RTI                    ;RETURN AND WAIT FOR RQI CLEAR INTERRUPT.
4634
4635 022514          80$:
4636 022514 042777 000020 157510  BIC    #IECLR,@SELO ;ENSURE INTERRUPT ENABLE FOR RDI CLEAR IS CLR.
4637 022522      CALL  $CLRQI      ;CLEAR RQI AND WAIT FOR RDI TO CLEAR.
4638                                ;*****
4639                                ;
4640                                ; RDI CLEAR - DETERMINE NEXT INPUT COMMAND.
4641                                ;
4642                                ;*****
4643 022526          NEXT:
4644 022526 022737 000002 002356  CMP    #HLT, LAST    ;WAS THE LAST COMMAND A HALT?
4645 022534 001015      BNE    110$          ;IF NOT - PROCEED.
4646 022536 005737 002274  TST    RESUME        ;ARE WE TESTING RESUME?
4647 022542 001541      BEQ    170$          ;IF NOT, DON'T ISSUE ANOTHER COMMAND.
4648 022544 005737 002346  TST    INFLAG        ;INPUT BUFFER DONE?
4649 022550 001403      BEQ    106$          ;IF NOT, BASE IN.
4650 022552 005737 002350  TST    OUTFLG        ;OUTPUT BUFFER DONE?
4651 022556 001133      BNE    170$          ;IF YES, DON'T ISSUE ANOTHER COMMAND.
4652 022560          106$:
4653 022560 112777 000143 157444  MOVB   #IESET!RQI!BASEI,@SELO ;ISSUE A BASE IN.
4654 022566 000527      BR     170$          ;EXIT
4655 022570          110$:
4656 022570 005737 002276  TST    DMCMD E      ;ARE WE IN DMC MODE?
4657 022574 001032      BNE    130$          ;IF YES - DON'T BOTHER CHECKING MODEM
4658                                ;WRITE AND AX3-15 WRITE COMMANDS
4659 022576 022737 000003 002356  CMP    #BASEI, LAST  ;WAS THE LAST COMMAND A BASE IN ?
4660 022604 001405      BEQ    115$          ;IF YES - SEE IF INTER. OR M. WRITE IS NEEDED?
4661 022606 022737 000015 002356  CMP    #INTER, LAST  ;WAS THE LAST COMMAND AN AX3-15 WRITE?
4662 022614 001413      BEQ    117$          ;
4663 022616 000421      BR     130$          ;KEEP CHECKING FOR NEXT COMMAND.
4664 022620          115$:
4665 022620 005737 002262  TST    INFACE        ;IS AN AX3-15 WRITE NEEDED?
4666 022624 001407      BEQ    117$          ;BR IF NOT
4667 022626 005737 002272  TST    START        ;WAS CONTROL IN ISSUED?
4668 022632 001004      BNE    117$          ;IF YES - NO NEED TO REWRITE AX3-15. THIS
4669                                ;SHOULD HAVE BEEN DONE ON THE 1ST BASE IN.
4670 022634 112777 000155 157370  MOVB   #IESET!RQI!INTER,@BSELO ;ISSUE AN AX3-15 WRITE COMMAND.
4671 022642 000501      BR     170$
    
```

```

4672 022644          117$:
4673 022644 005737 002306      TST      WMAINT          ;WRITE MAINT 1 OR 2?
4674 022650 001404          BEQ      130$          ;IF NOT - SKIP WRITE MODEM COMMAND.
4675 022652 112777 000145 157352  MOVVB   #IESET!RQI!WMODEM,@BSELO ;ISSUE A MODEM WRITE COMMAND
4676 022660 000472          BR       170$
4677 022662          130$:
4678 022662 005737 002272      TST      START          ;WAS A CONTROL IN ISSUED?
4679 022666 001006          BNE     150$          ;IF YES - SKIP
4680 022670 005237 002272      INC     START          ;SET FLAG.
4681 022674 112777 000141 157330  MOVVB   #IESET!RQI!CNTRL,@BSELO ;ISSUE A CONTROL IN
4682 022702 000461          BR       170$
4683 022704          150$:
4684 022704 005737 002324      TST      INRCV          ;ARE ALL THE BA/CC IN RCVS DONE?
4685 022710 001424          BEQ     160$          ;IF YES - BR TO SEE IF XMITS DONE.
4686 022712 005737 002274      TST      RESUME         ;IS A TEST OF RESUME REQUESTED?
4687 022716 001415          BEQ     153$          ;BR IF NOT.
4688 022720 032737 000001 002324  BIT     #BIT0,INRCV    ;IS THIS AN ODD COUNT?
4689 022726 001411          BEQ     153$          ;BR IF NOT.
4690 022730 005737 002352      TST      RESFLG         ;WAS THE LAST COMMAND A BASE IN RESUME?
4691 022734 001004          BNE     152$          ;IF YES, ISSUE BA/CC
4692          ;HALT - TO TEST RESUME. NOTE: THIS WILL
4693          ;OCCUR ONLY WHEN RESUME IS REQUESTED,
4694          ;FOLLOWING EVERY OTHER BA/CC
4695          ;COMMAND (NEVER FOLLOWING A RESUME)
4696 022736 112777 000142 157266  MOVVB   #IESET!RQI!HLT,@BSELO ;HALT IT
4697 022744 000440          BR       170$
4698 022746          152$:
4699 022746 005037 002352      CLR     RESFLG         ;CLEAR FLAG.
4700 022752          153$:
4701 022752 112777 000144 157252  MOVVB   #IESET!RQI!BACCR,@BSELO ;ISSUE A BA/CC IN RCV. COMMAND.
4702 022760 000432          BR       170$
4703 022762          160$:
4704 022762 005737 002326      TST      INXMIT         ;ARE ALL THE BA/CC IN XMITS DONE?
4705 022766 001424          BEQ     165$          ;IF YES, SET THE FLAG
4706 022770 005737 002274      TST      RESUME         ;IS A TEST OF RESUME REQUESTED?
4707 022774 001415          BEQ     163$          ;BR IF NOT.
4708 022776 032737 000001 002326  BIT     #BIT0,INXMIT    ;IS THIS AN ODD COUNT?
4709 023004 001411          BEQ     163$          ;BR IF NOT.
4710 023006 005737 002352      TST      RESFLG         ;WAS THE LAST COMMAND A BASE IN RESUME?
4711 023012 001004          BNE     162$          ;IF YES, ISSUE BA/CC
4712          ;HALT - TO TEST RESUME. NOTE: THIS WILL
4713          ;OCCUR ONLY WHEN RESUME IS REQUESTED,
4714          ;FOLLOWING EVERY OTHER BA/CC
4715          ;COMMAND (NEVER FOLLOWING A RESUME)
4716 023014 112777 000142 157210  MOVVB   #IESET!RQI!HLT,@BSELO ;HALT IT
4717 023022 000411          BR       170$
4718 023024          162$:
4719 023024 005037 002352      CLR     RESFLG         ;CLEAR BASE IN RESUME FLAG.
4720 023030          163$:
4721 023030 112777 000140 157174  MOVVB   #IESET!RQI!BACCT,@BSELO ;ISSUE A BA/CC IN XMIT COMMAND.
4722 023036 000403          BR       170$
4723 023040          165$:
4724 023040 012737 177777 002346  MOV     #-1,INFLAG     ;FLAG THAT ALL BA/CC INS DONE.
4725
4726 023046          170$:
4727 023046 012601          MOV     (SP)+,R1      ;RESTORE R1
    
```





```

4769
4770 023222 001005          BNE      21$          ;ADDRESS RANGE: 200000 - 277776)
4771 023224          ERRDF  11,EMG11,ERRG2 ;PROCEED - IF BIT SET.
(4) 023224 104455
(5) 023226 000013          TRAP   C$ERDF
(5) 023230 020214          .WORD  11
(5) 023232 015070          .WORD  EMG11
4772 023234 000505          .WORD  ERRG2
4773 023236
4774 023236 042777 140000 156774 21$: BR      60$
4775 023244 017702 156766          BIC     #BIT15!BIT14,@SEL6 ;CLEAR THE EXTENDED ADDRESS BITS.
4776 023250 042702 017777          MOV     @SEL4,R2          ;SAVE BITS 0-15 OF THE PHYSICAL ADDRESS.
4777 023254 042777 160000 156754          BIC     #17777,R2          ;SAVE ONLY PAGE ADDRESS BITS.
4778          BIC     #160000,@SEL4 ;CLEAR PAGE ADDRESS BITS IN SEL4
4779 023262 005702          TST     R2          ;DETERMINE PAGE # FOR VIRTUAL ADDRESS.
4780 023264 001004          BNE     22$          ;IS THIS PAGE 3?
4781 023266 052777 060000 156742          BIS     #60000,@SEL4      ;IF NOT CHECK FOR OTHER PAGES
4782 023274 000421          BR      40$          ;SET BITS FOR PAGE 3.
4783 023276
4784 023276 022702 020000          22$: CMP     #20000,R2          ;IS THIS PAGE 4?
4785 023302 001004          BNE     23$          ;IF NOT - KEEP CHECKING
4786 023304 052777 100000 156724          BIS     #100000,@SEL4    ;SET BITS FOR PAGE 4.
4787 023312 000412          BR      40$
4788 023314
4789 023314 022702 040000          23$: CMP     #40000,R2          ;IS THIS PAGE 5?
4790 023320 001004          BNE     24$          ;IF NOT - MUST BE PAGE 6
4791 023322 052777 120000 156706          BIS     #120000,@SEL4    ;SET BITS FOR PAGE 5.
4792 023330 000403          BR      40$
4793 023332
4794 023332 052777 140000 156676          24$: BIS     #140000,@SEL4    ;SET BITS FOR PAGE 6.
4795 023340          40$:
4796 023340 032777 000004 156666          BIT     #RCV,@SEL2        ;IS THIS A RECV. BUFFER?
4797 023346 001023          BNE     50$          ;IF YES - PROCESS THE BUFFER.
4798 023350 005337 002332          DEC     OUTXMT          ;DECREMENT BA/CC OUT XMIT.
4799 023354 022577 156656          CMP     (R5)+,@SEL4      ;IS THE XMIT BUFFER ADDRESS CORRECT?
4800 023360 001406          BEQ     41$          ;IF YES - PROCEED.
4801 023362 005725          TST     (R5)+          ;INCR. POINTER FOR ERROR MESSAGE.
4802 023364          ERRDF  11,EMG11,ERRG8 ;IF NOT - ERROR
(4) 023364 104455
(5) 023366 000013          TRAP   C$ERDF
(5) 023370 020214          .WORD  11
(5) 023372 016056          .WORD  EMG11
4803 023374 000425          .WORD  ERRG8
4804 023376
4805 023376 022577 156636          41$: BR      60$          ;EXIT ROUTINE
4806 023402 001422          CMP     (R5)+,@SEL6      ;IS THE CHAR. COUNT CORRECT?
4807 023404          BEQ     60$          ;IF OK - EXIT ROUTINE.
(4) 023404 104455          ERRDF  12,EMG12,ERRG8 ;IF NOT - ERROR
(5) 023406 000014          TRAP   C$ERDF
(5) 023410 020241          .WORD  12
(5) 023412 016056          .WORD  EMG12
4808 023414 000415          .WORD  ERRG8
4809 023416
4810 023416 005337 002330          50$: BR      60$          ;EXIT
4811 023422 022477 156610          DEC     OUTRCV          ;DECREMENT BA/CC OUT RCV
4812 023426 001406          CMP     (R4)+,@SEL4      ;IS THE RCV BUFFER ADDRESS CORRECT?
          BEQ     51$          ;IF OK - PROCEED
    
```

```

4813 023430          ERRDF  11,EMG11,ERRG7
(4) 023430 104455
(5) 023432 000013          TRAP  CSERDF
(5) 023434 020214          .WORD  11
(5) 023436 015760          .WORD  EMG11
4814 023440 005724          TST   (R4)+      ;UPDATE POINTER
4815 023442 000402          BR    60$        ;EXIT ROUTINE
4816 023444          51$:
4817 023444 017724 156570    MOV   @SEL6,(R4)+ ;CHANGE THE CHARACTER COUNT TO WHAT
4818                                ;WAS RECEIVED.
4819 023450          60$:
4820 023450 005737 002332    TST   OUTXMT     ;HAVE ALL THE XMITTS BEEN DONE?
4821 023454 001011          BNE   65$        ;IF NOT, CONTINUE
4822 023456 005737 002330    TST   OUTRCV     ;HAVE ALL THE RECEIVES BEEN DONE?
4823 023462 001006          BNE   65$        ;IF NOT, CONTINUE
4824 023464          61$:
4825 023464 042777 000100 156542 BIC   #IEO,@SEL2 ;CLEAR THE OUTPUT INTERRUPT
4826 023472 012737 177777 002350 MOV   #-1,OUTFLG ;FLAG AS DONE.
4827 023500          65$:
4828 023500 042777 000207 156526 BIC   #RDO!CMD,@SEL2 ;CLEAR THE RDO BIT.
4829 023506 012600          MOV   (SP)+,RO   ;RESTORE RO
4830 023510          ENDSRV
(3) 023510
(2) 023510 000002          L10020:
4831                                RTI
4832          ;*****
4833          ;*****
4834
4835 023512          BGNSRV NOXMEM
(3) 023512          NOXMEM::
4836
4837 023512 012737 000001 002340 MOV   #1,FLAG     ;SET FLAG IF MEMORY ADDRESSED IS NON-EXISTENT.
4838
4839 023520          ENDSRV
(3) 023520          L10021:
(2) 023520 000002          RTI
4840
4841
4842
4843
4844
4845
4846          .SBTTL  DROP UNIT SECTION
4847
4848          ;////////////////////////////////////
4849          ;// THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4850          ;// TO NO LONGER BE TESTED.
4851          ;////////////////////////////////////
4852
4853 023522          BGNDU
(3) 023522          LSDU::
4854
4855 023522          BRESET      ;ISSUE UNIBUS RESET TO CLEAN UP
(3) 023522 104433          (RAP  CSRESET
4856 023524          PRINTF #FMDROP,LOGDEV
(8) 023524 013746 002362          MOV   LOGDEV,-(SP)
    
```

(7) 023530 012746 023552  
 (6) 023534 012746 000002  
 (3) 023540 010600  
 (4) 023542 104417  
 (4) 023544 062706 000006

MOV #FMDROP, -(SP)  
 MOV #2, -(SP)  
 MOV SP, R0  
 TRAP C\$PNTF  
 ADD #6, SP

4857  
 4858 023550 ENDDU  
 (3) 023550  
 (3) 023550 104453

L10022: TRAP C\$DU

4859  
 4860  
 4861 023552 047045 040445 047125 FMDROP: .ASCIZ /%N%AUNIT %D2%A DROPPED/  
 023560 052111 022440 031104  
 023566 040445 042040 047522  
 023574 050120 042105 000

4862 023602 .EVEN

4863  
 4864  
 4865  
 4866  
 4867 .SBTTL TEST 1 - DMR CSR VERIFICATION

4868  
 4869  
 4870  
 4871  
 4872  
 4873  
 4874  
 4875  
 4876  
 4877  
 4878  
 4879  
 4880  
 4881  
 4882  
 4883  
 4884  
 4885  
 4886  
 4887  
 4888  
 4889  
 4890  
 4891  
 4892  
 4893  
 4894  
 4895

```

*****
* TEST 1 - DMR-11
* VERIFY THAT ADDRESSING THE 4 UNIBUS CSRS DOES NOT CAUSE A NON-
* EXISTENT MEMORY TRAP.
*
* THE DMR IS AN NPR DEVICE RESIDING ON A UNIBUS. COMMUNICATION
* BETWEEN THE MAIN CPU AND THE DMR IS ACCOMPLISHED THROUGH A
* SET OF FOUR 16-BIT UNIBUS CONTROL AND STATUS REGISTERS (CSRS).
* THE FOUR REGISTERS ARE ASSIGNED ADDRESSES IN THE I/O PAGE
* FLOATING ADDRESS SPACE: 76XXX0 - 76XXX6
*
* NOTE: THIS TEST IS REDUNDANT IN THAT STATIC LOGIC TESTS SHOULD
* HAVE BEEN RUN BEFORE THESE FREE-RUNNING TESTS WERE STARTED, AND
* THEY SHOULD HAVE DETECTED ANY CSR ADDRESSING PROBLEMS.
* BUT JUST IN CASE THOSE STATIC TESTS AREN'T RUN, WE'LL BE SAFE.
*****
  
```

4885 023602  
 (3) 023602

BGNTST

T1::

4887 023602 SETVEC #4, #LOCATE, #PRI07 ;SET UP NON -EXISTENT MEMORY TRAP VECTOR.

(7) 023602 012746 000340  
 (6) 023606 012746 023724  
 (5) 023612 012746 000004  
 (4) 023616 012746 000003  
 (3) 023622 104437  
 (2) 023624 062706 000010  
 4888 023630 005037 002340  
 4889 023634 005001

MOV #PRI07, -(SP)  
 MOV #LOCATE, -(SP)  
 MOV #4, -(SP)  
 MOV #3, -(SP)  
 TRAP C\$SVEC  
 ADD #10, SP

CLR FLAG ;FLAG USED IN THE TRAP ROUTINE.  
 CLR R1 ;USE REGISTER TO REMEMBER WHICH OF THE  
 ;4 CSRS WE ARE ADDRESSING.

```

*****
: IF ADDRESSING ANY ONE OF THE CSRS RESULTS IN A TRAP TO VECTOR 04, THE TRAP
: WILL REPORT THE ERROR (SEE INTERRUPT ROUTINE 'LOCATE'). OTHERWISE THE
: MEMORY REFERENCE IS UNEVENTFUL AND THE DEVICE IS READY FOR FURTHER TESTS
  
```

```

4896                                     ;:*****
4897                                     ;:*****
4898 023636 005777 156370                TST    @SELO                ;TEST THE CSR AT 76XXX0
4899 023642 012701 000002                MOV    #2,R1                ;SAVE THE OFFSET OF THE NEXT CSR
4900 023646 005777 156362                TST    @SEL2                ;TEST THE CSR AT 76XXX2
4901 023652 012701 000004                MOV    #4,R1                ;SAVE THE OFFSET OF THE NEXT CSR
4902 023656 005777 156354                TST    @SEL4                ;TEST THE CSR AT 76XXX4
4903 023662 012701 000006                MOV    #6,R1                ;SAVE THE OFFSET OF THE NEXT CSR
4904 023666 005777 156346                TST    @SEL6                ;TEST THE CSR AT 76XXX6
4905 023672 005737 002340                TST    FLAG                ;WAS THERE A TRAP?
4906 023676 001406                BEQ    10$                  ;IF NOT - EXIT.
4907 023700                DODU    LOGDEV                ;DROP THE DEVICE
    (3) 023700 013700 002362                MOV    LOGDEV,R0
    (3) 023704 104451                TRAP   C$DODU
4908 023706                DOCLN                ;DO CLEAN UP - FORCE BACK TO INIT CODE.
    (3) 023706 104444                TRAP   C$DCLN
4909 023710 005037 002340                CLR    FLAG                ;RESTORE THE FLAG.
4910 023714                10$:
4911 023714                CLRVEC #4                ;RETURN VECTOR 04 TO NORMAL STATE
    (3) 023714 012700 000004                MOV    #4,R0
    (3) 023720 104436                TRAP   C$CVEC
4912
4913 023722                ENDTST
    (3) 023722                L10023:
    (3) 023722 104401                TRAP   C$SETST
4914
4915
4916 023724                BGNSRV LOCATE                ;INTERRUPT SERVICE ROUTINE
    (3) 023724                LOCATE::
4917 023724 010046                MOV    R0,-(SP)                ;SAVE R0
4918 023726 005737 002340                TST    FLAG                ;HAVE WE HAD AT LEAST 1 PREVIOUS TRAP?
4919 023732 001006                BNE    10$                  ;IF YES, DON'T BOTHER DECLARING ANOTHER
4920
4921 023734                ERRDF 6,EMTO                ;DEVICE FATAL ERROR
    (4) 023734 104455                ;NON-EXISTENT DEVICE ERROR
    (5) 023736 000006                TRAP   C$ERDF
    (5) 023740 024002                .WORD 6
    (5) 023742 000000                .WORD EMTO
    (5) 023742 000000                .WORD 0
4922 023744 005237 002340                10$: INC    FLAG                ;SET THE FLAG
4923 023750                10$:
4924 023750                PRINTX #FMT0,R1,CSR(R1) ;PRINT THE CSR THAT DOESN'T RESPOND.
    (9) 023750 016146 002232                MOV    CSR(R1),-(SP)
    (8) 023754 010146                MOV    R1,-(SP)
    (7) 023756 012746 024040                MOV    #FMT0,-(SP)
    (6) 023762 012746 000003                MOV    #3,-(SP)
    (3) 023766 010600                MOV    SP,R0
    (4) 023770 104415                TRAP   C$PNTX
    (4) 023772 062706 000010                ADD    #10,SP
4925 023776 012600                ENDSRV MOV    (SP)+,R0                ;RESTORE R0
    (3) 024000                L10024:
    (2) 024000 000002                RTI
4927
4928 024002 051503 020122 042101 EMT0: .ASCIZ /CSR ADDRESSING ERROR - TRAP 4/
    024010 051104 051505 044523
    024016 043516 042440 051122
    
```

024024 051117 026440 052040  
 024032 040522 020120 000064  
 4929 024040 051445 022463 041501  
 024046 051123 024040 042523  
 024054 022514 030504 040445  
 024062 020051 052101 022440  
 024070 033117 040445 042040  
 024076 042517 020123 047516  
 024104 020124 042522 050123  
 024112 047117 022504 000116

FMT0: .ASCIZ /%S3%ACSR (SEL%D1%A) AT %06%A DOES NOT RESPOND%N/

4930  
4931  
4932  
4933  
4934  
4935  
4936  
4937  
4938  
4939  
4940  
4941  
4942  
4943  
4944  
4945  
4946  
4947  
4948  
4949  
4950  
4951  
4952  
4953  
4954  
4955  
4956  
4957  
4958  
4959  
4960  
4961  
4962  
4963  
4964  
4965

.EVEN

.SBTTL TEST 2 - ROM CHECK

```

*****
*          TEST 2 - DMR-11
* ROM CRC/CCITT - CHECK ROM POSITION AND CALCULATE CRC/CCITT. THE
* LAST 4 BYTES CONTAIN INFORMATION ABOUT THE ROM TO CHECK. THE 1ST
* OF THESE BYTES CONTAINS THE ASCII VERSION NUMBER. THE 2ND BYTE
* CONTAINS THE ROM NUMBER. THE 3RD AND 4TH BYTES CONTAIN A NEGATIVE
* CRC/CCITT WORD FOR THE ROM.
*          CHIP ADDRESS RANGE
*          LOCATION  CHIP NO.      BYTE    ADDRESS RANGE
*          E03      0            LOW     0000 - 1777
*          E02      1            HIGH    0000 - 1777
*          E04      2            LOW     2000 - 3777
*          E01      3            HIGH    2000 - 3777
*          E05      4            LOW     4000 - 5777
*          E14      5            HIGH    4000 - 5777
*
***** IMPORTANT !!!!!!!!!!!!! *****
* FOR THIS TEST TO RUN CORRECTLY, ENSURE THAT SWITCH 1 AT LOCATION
* E85 ON THE M8207 IS ON. IF THIS SWITCH IS OFF, BSEL1 WILL BE
* LOCKED OUT AND THE MAINTENANCE FEATURES WILL NOT BE ENABLED.
*****
*          SUBTEST 1 - ON THE FIRST PASS PRINT THE VERSION # IN EACH ROM
*          SUBTEST 2 - GENERATE THE CRC-CCITT IN EACH ROM AND COMPARE IT
*                      IT AGAINST THE CRC BLASTED IN THE ROM
*          SUBTEST 3 - COMPARE THE ROM # BLASTED IN THE ROM AGAINST THE
*                      EXPECTED ROM #.
*****
    
```

4966 024120  
 (3) 024120  
 4967 024120  
 (3) 024120  
 (3) 024120 104402  
 4968 024122 022737 000001 002270  
 4969 024130 001061  
 4970  
 4971 024132 005004  
 4972 024134 012705 000001

```

BGNTST
BGNSUB
T2::
T2.1: TRAP C$BSUB
CMP #1,STARES ;IS THIS THE FIRST PASS?
BNE 5$ ;IF NOT - SKIP THIS SUBROUTINE.
CLR R4 ;GET VERSION # FROM EACH ROM AND PRINT IT OUT
MOV #1,R5 ;# OF THE 1ST ROM
;# OF NEXT ROM
    
```

```

4973 024140 012737 001774 002404      MOV    #1774,ROMADR    ;ADDRESS OF BYTE CONTAINING # IN ROMS 0 & 1
4974 024146      PRINTB #FMT1,LOGDEV    ;MICROCODE VERSION
(8) 024146 013746 002362      MOV    LOGDEV,-(SP)
(7) 024152 012746 025074      MOV    #FMT1,-(SP)
(6) 024156 012746 000002      MOV    #2,-(SP)
(3) 024162 010600      MOV    SP,R0
(4) 024164 104414      TRAP  C$PNTB
(4) 024166 062706 000006      ADD    #6,SP
4975 024172      1$:
4976 024172      CALL  $ROMO           ;GET ROM CONTENTS.
4977 024176 117737 156036 025430      MOVB  @BSEL6,REV1     ;SAVE THE ASCII REVISION # OF THE ROM
4978 024204 117737 156040 025432      MOVB  @BSEL7,REV2     ;SAVE THE REV. # OF THE NEXT ROM
4979      :PRINT
4980 024212      PRINTB #FMT2,R4,#REV1,R5,#REV2
(11) 024212 012746 025432      MOV    #REV2,-(SP)
(10) 024216 010546      MOV    R5,-(SP)
(9) 024220 012746 025430      MOV    #REV1,-(SP)
(8) 024224 010446      MOV    R4,-(SP)
(7) 024226 012746 025143      MOV    #FMT2,-(SP)
(6) 024232 012746 000005      MOV    #5,-(SP)
(3) 024236 010600      MOV    SP,R0
(4) 024240 104414      TRAP  C$PNTB
(4) 024242 062706 000014      ADD    #14,SP
4981
4982 024246 022705 000005      CMP    #5,R5           ;ARE WE DONE?
4983 024252 001410      BEQ   5$              ;IF YES - EXIT
4984 024254 062704 000002      ADD    #2,R4           ;INCR. ROM NUMBERS
4985 024260 062705 000002      ADD    #2,R5
4986 024264 062737 002000 002404      ADD    #2000,ROMADR    ;ADDRESS OF BYTES CONTAINING NEXT ROM REV #S.
4987 024272 000737      BR    1$
4988
4989      5$:
4990 024274      ENDSUB
(3) 024274      L10026:
(3) 024274 104403      TRAP  C$ESUB
4991
4992
4993 024276      BGNSUB
(3) 024276      T2.2:
(3) 024276 104402      TRAP  C$BSUB
4994 024300 005037 002340      CLR    FLAG           ;USE THE FLAG TO MARK WHEN AN ERRDF
4995      :HAS BEEN DETECTED IN THIS TEST.
4996 024304 005004      CLR    R4             ;START CRC CHECK WITH ROM 0
4997      :R4 IS THE ROM #. THE LOCATION FOR THE
4998      :ROM IS CONTAINED IN THE TABLE 'ROMLOC'.
4999 024306 005037 002404      CLR    ROMADR         ;BEGIN AT ROM ADDRESS 0
5000
5001 024312      10$:
5002 024312 012737 177777 002374      MOV    #-1,LOCRC      ;INITIALIZE CRC WORD FOR THE LOW BYTE
5003      :CALCULATION.
5004 024320 012737 177777 002376      MOV    #-1,HICRC      ;INIT. CRC WORD FOR THE HIGH BYTE.
5005 024326 012701 001000      MOV    #1000,R1       ;COUNTER FOR LOOP TO READ THE ROM CONTENTS
5006      :AND CALCULATE THE CRC - THE COUNTER IS 512.
5007      :BECAUSE 2 ADDRESS LOCATIONS ARE READ FOR EACH
5008      :PASS (I.E. THE ROMS ARE 1K X 8 BITS)
5009
    
```

```

5010
5011
5012
5013
5014 024332          20$:
5015 024332
5016 024336 117737 155676 002400
5017 024344 117737 155700 002402
5018 024352 005237 002404
5019 024356
5020 024362 117737 155652 002401
5021 024370 117737 155654 002403
5022
5023
5024 024376 005237 002404
5025 024402 005301
5026 024404 001443
5027
5028
5029
5030
5031 024406 012703 000020
5032 024412          25$:
5033 024412 000241
5034 024414 006037 002374
5035 024420 006037 002400
5036
5037
5038 024424 102011
5039 024426 012702 102010
5040 024432 043702 002374
5041 024436 042737 102010 002374
5042 024444 050237 002374
5043 024450          30$:
5044 024450 000241
5045 024452 006037 002376
5046 024456 006037 002402
5047
5048 024462 102011
5049
5050 024464 012702 102010
5051 024470 043702 002376
5052 024474 042737 102010 002376
5053 024502 050237 002376
5054 024506          35$:
5055 024506 005303
5056 024510 001340
5057 024512 000707
5058 024514          40$:
5059
5060
5061
5062
5063
5064 024514 005137 002374
5065 024520 023737 002374 002400
    
```

: BECAUSE A ROM OUT WILL OUTPUT THE ROM CONTENTS (I.E. 16 BITS)  
 : THIS ROUTINE WILL CALCULATE/CHECK THE CRC 2 ROMS AT A TIME.  
 :  
 : GET THE ROM CONTENTS  
 : SAVE THE LOW BYTE OF THE ROM CONTENTS.  
 : SAVE THE HIGH BYTE OF THE ROM CONTENTS.  
 : INCREMENT THE ROM ADDRESS POINTER  
 : GET THE CONTENTS OF THE NEXT ROM ADDRESS  
 : SAVE THE NEXT LOW BYTE.  
 : SAVE THE NEXT HIGH BYTE.  
 : NOTE: AT THIS POINT LOWORD IS A WORD WHICH  
 : HAS 2 CONSECUTIVE LOW BYTES OF ROM CONTENTS.  
 : INCREMENT THE ROM ADDRESS POINTER  
 : ARE WE FINISHED WITH THESE 2 ROMS?  
 : IF YES, CHECK CRC  
 :  
 : CRC/CCITT CALCULATION = CONVERT THE WORD (LOWORD & HIWORD) TO  
 : A SERIAL STREAM FOR CALCULATION.  
 :  
 : 16 BITS TO CONSIDER  
 : CLEAR THE CARRY  
 : ROTATE BIT 0 INTO THE CARRY BIT  
 : ROTATE BIT 0 INTO C AND THE OLD C INTO BIT 15  
 : ARE THE BITS 15 & BITS 0 THE SAME?  
 : IF YES (V IS CLEAR), DON'T DO THE CRC  
 : NOTE: V IS THE EXCLUSIVE OR OF BIT 0 & BIT 15.  
 : CRC/CCITT POLYNOMIAL  
 :  
 : CLEAR THE CARRY  
 : ROTATE BIT 0 INTO C  
 : ROTATE OLD C INTO BIT 15 (SIGN) & BIT 0 INTO C  
 : ARE THE BITS 0 OF HICRC & HIWORD THE SAME?  
 : IF YES (V IS CLEAR), DON'T DO THE CRC.  
 : NOTE: V IS THE EXCLUSIVE OR OF BIT 0 & BIT 15.  
 : CRC/CCITT POLYNOMIAL  
 : DO ALL 16 BITS  
 : GET THE CONTENTS OF THE NEXT 2 ROM ADDRESSES.  
 :  
 : AT THIS POINT WE'VE READ THE CONTENTS AND CALCULATED THE CRC FOR  
 : 2 ROM ROMS (ONE LOW BYTE & ONE HIGH BYTE). ALSO WE'VE READ THE  
 : CRC BLASTED INTO THE LAST 2 BYTES OF THE ROM (IN LOWORD/HIWORD)  
 :  
 : COMPLEMENT THE CALCULATED CRC  
 : IS THE CRC IN ROM THE SAME AS THE





```

5096 024720          BGNSUB
(3) 024720
(3) 024720 104402          T2.3: TRAP C$BSUB
5097 024722 005037 002340 CLR FLAG ;CLEAR FLAG
5098 024726 005004          CLR R4 ;BEGIN AT ROM 0
5099 024730 012737 001775 002404 MOV #1775,ROMADR ;ADDRESS OF BYTE CONTAINING ROM #
5100 024736          10$:
5101 024736          CALL $ROMO ;GET ROM CONTENTS
5102 024742 117701 155272 MOVB @BSEL6,R1 ;SAVE THE CONTENTS OF THE LOW BYTE
5103          ;FOR ROMS 0,2,4
5104 024746 000402          BR 17$
5105 024750          15$:
5106 024750 117701 155274 MOVB @BSEL7,R1 ;SAVE THE CONTENTS OF THE HIGH BYTE
5107          ;FOR ROMS 1,3,5
5108 024754          17$:
5109 024754 042701 177760 BIC #^C17,R1 ;CONVERT THE ASCII BYTE TO AN OCTAL WORD.
5110 024760 020104          CMP R1,R4 ;IS THIS THE EXPECTED ROM #
5111 024762 001427          BEQ 20$ ;IF YES - OK.
5112 024764 005737 002340 TST FLAG ;HAS AN ERRDF ALREADY BEEN DECLARED (REMEMBER
5113          ;WE'RE IN A LOOP)
5114 024770 001007          BNE 18$ ;IF YES, DON'T BOTHER WITH ANOTHER ERRDF.
5115 024772 012737 000001 002340 MOV #1,FLAG ;FLAG THAT ERRDF HAS BEEN DETECTED.
5116 025000          ERRDF 7,EMT2 ;ROM ERROR
(4) 025000 104455          TRAP C$ERDF
(5) 025002 000007          .WORD 7
(5) 025004 025402          .WORD EMT2
(5) 025006 000000          .WORD 0
5117 025010          18$:
5118 025010          PRINTB #FMT4,<B,ROMLOC(R4)>,R1,R4
(10) 025010 010446          MOV R4,-(SP)
(9) 025012 010146          MOV R1,-(SP)
(8) 025014 005046          CLR -(SP)
(8) 025016 156416 025421 BISB ROMLOC(R4),(SP)
(7) 025022 012746 025304 MOV #FMT4,-(SP)
(6) 025026 012746 000004 MOV #4,-(SP)
(3) 025032 010600          MOV SP,R0
(4) 025034 104414          TRAP C$PNTB
(4) 025036 062706 000012 ADD #12,SP
5119 025042          20$:
5120 025042 022704 000005 CMP #5,R4 ;DID WE FINISH THE LAST ROM?
5121 025046 001410          BEQ 30$ ;IF YES - SKIP TO THE END
5122 025050 005204          INC R4 ;POINT TO THE NEXT ROM #
5123 025052 032704 000001 BIT #BIT0,R4 ;IS THIS AN ODD #
5124 025056 001334          BNE 15$ ;IF YES GO BACK AND READ THE HIGH BYTE
5125
5126 025060 062737 002000 002404 ADD #2000,ROMADR ;INCR. ADDRESS POINTER TO NEXT ROM #.
5127 025066 000723          BR 10$
5128 025070          30$:
5129 025070          ENDSUB
(3) 025070          L10030: TRAP C$ESUB
(3) 025070 104403
5130
5131 025072          ENDTST          L10025: TRAP C$ETST
(3) 025072
(3) 025072 104401
5132 025074 047045 040445 044515 FMT1: .ASCIZ /%N%AMICROCODE REVISION IN UNIT%D3%A:%N/
    
```

```
025102 051103 041517 042117
025110 020105 042522 044526
025116 044523 047117 044440
025124 020116 047125 052111
025132 042045 022463 035101
025140 047045 000
5133 025143 045 051101 046517 FMT2: .ASCIZ /%AROM%D2%A - REV. %T%N%AROM%D2%A - REV. %T%N/
025150 042045 022462 020101
025156 020055 042522 027126
025164 022440 022524 022516
025172 051101 046517 042045
025200 022462 020101 020055
025206 042522 027126 022440
025214 022524 000116
5134 025220 040445 047522 022515 FMT3: .ASCIZ /%AROM%D2%A: CALCUATED CRC =%06%A CRC IN ROM =%06%N/
025226 031104 040445 020072
025234 040503 041514 040525
025242 042524 020104 051103
025250 020103 022475 033117
025256 040445 020040 051103
025264 020103 047111 051040
025272 046517 036440 047445
025300 022466 000116
5135 025304 040445 022505 031104 FMT4: .ASCIZ /%AE%D2%A IS ROM %D1%A (SHOULD BE ROM %D1%A)%N/
025312 040445 044440 020123
025320 047522 020115 042045
025326 022461 020101 051450
025334 047510 046125 020104
025342 042502 051040 046517
025350 022440 030504 040445
025356 022451 000116
5136
5137 025362 051103 026503 041503 EMT1: .ASCIZ /CRC-CCITT ERROR/
025370 052111 020124 051105
025376 047522 000122
5138 025402 047514 040503 044524 EMT2: .ASCIZ /LOCATION ERROR/
025410 047117 042440 051122
025416 051117 000
5139
5140 025421 003 002 004 ROMLOC: .BYTE 3,2,4,1,5,14. ;ROM 0 = ROM LOCATION 3 ETC.
025424 001 005 016
5141 025430 .EVEN
5142 025432 000000 REV1: .WORD 0 ;ASCII VALUE OF THE REV. NUMBER
5143 025432 000000 REV2: .WORD 0 ;ASCII VALUE OF THE REV. NUMBER
5144
5145
5146
5147
5148
5149
5150 .SBTTL TEST 3 - MASTER CLEAR AND MICROTST
5151
5152 ;*****
5153 ;* TEST 3 - DMR-11
5154 ;* MASTER CLEAR
5155 ;* THIS TEST WILL ISSUE 2 MASTER CLEARS. EACH CALL TO THE MASTER
```

5156  
 5157  
 5158  
 5159  
 5160  
 5161  
 5162  
 5163  
 5164  
 5165  
 5166  
 5167  
 5168  
 5169  
 5170  
 5171  
 5172  
 5173  
 5174  
 5175  
 5176  
 5177  
 5178 025434  
 (3) 025434  
 5179 025434  
 (1)  
 (1) 025434 004737 011064  
 (1)  
 5180  
 5181 025440  
 (3) 025440 104410  
 (3) 025442 000072  
 5182 025444 105777 154574  
 5183 025450 001011  
 5184 025452  
 (7) 025452 012746 017473  
 (6) 025456 012746 000001  
 (3) 025462 010600  
 (4) 025464 104414  
 (4) 025466 062706 000004  
 5185 025472 000420  
 5186 025474  
 5187 025474 117701 154544  
 5188 025500  
 (1)  
 (1) 025500 004737 011064  
 (1)  
 5189  
 5190 025504  
 (3) 025504 104410  
 (3) 025506 000026  
 5191 025510 117702 154530  
 5192 025514 060102  
 5193  
 5194  
 5195

```

:* CLEAR ROUTINE WILL ENSURE THAT THE RUN BIT WILL BE SET. ALSO
:* THE MASTER CLEAR WILL CAUSE THE DIAGNOSTIC MICROTSTES TO BE
:* RUN WHEN THE MICRODIAGNOSTIC BIT (BIT 13 IN SEL0) IS CORRECTLY
:* SET OR CLEARED. BECAUSE THE RUNNING OF MICROTSTES DEPENDS ON THE
:* EXCLUSIVE OR OF THE HARDWARE SWITCH 10 ON E134 OF THE M8203 AND
:* THE MICRODIAGNOSTIC BIT, WE CAN'T KNOW WHETHER THE SETTING OR
:* CLEARING OF BIT 13 WILL RESULT IN THE RUNNING OF MICROTSTES.
:* THEREFORE THE MASTER CLEAR SUBROUTINE WILL TOGGLE (I.E. SET
:* BIT 13 ONLY ON EVERY OTHER MASTER CLEAR) THE SOFTWARE BIT.
:* THIS WILL ENSURE THAT REGARDLESS OF THE POSITION OF THE
:* HARDWARE SWITCH, MICROTSTES WILL BE RUN EVERY OTHER MASTER CLEAR.
:* WHEN RUNNING THIS TEST, WE EXPECT TO ADD THE RESULTS OF BSEL3
:* AFTER EACH MASTER CLEAR.
:* BSEL3 = 100 - MICROTSTES DISABLED
:* BSEL3 = 200 - MICROTSTES RUN SUCCESSFULLY
:* IF THE RESULT OF THE 2 MASTER CLEARS IS NOT 300, AN ERROR IS
:* REPORTED.
:*
:* ADDITIONALLY THIS ROUTINE WILL REPORT WHENEVER THE RESULT OF
:* BSEL3 IS 0. THIS WILL MEAN THAT THE DEVICE IS NOT A DMR
:* (I.E. DMC)
    
```

\*\*\*\*\*  
 BGNTST

```

T3::
CLEAR                ;MACRO FOR MASTER CLEAR
                    ;**** MACRO EXPANSION ****
                    ;ISSUE A DMR MASTER CLEAR
                    ;****

JSR PC, $MSCLR

ESCAPE TST          ;IF ERROR, BR TO TEST END.

TSTB @BSEL3         ;IS THERE A DMR RESPONSE?
BNE 1$
PRINTB #FMG19       ;REPORT DEVICE NOT DMR.

MOV #FMG19, -(SP)
MOV #1, -(SP)
MOV SP, R0
TRAP C$PNTB
ADD #4, SP

1$:
BR 5$

MOVB @BSEL3, R1     ;SAVE THE RESULT OF THE FIRST MASTER CLEAR.
CLEAR              ;MASTER CLEAR AGAIN.
                    ;**** MACRO EXPANSION ****
                    ;ISSUE A DMR MASTER CLEAR
                    ;****

JSR PC, $MSCLR

ESCAPE TST          ;IF ERROR, BR TO TEST END.

MOVB @BSEL3, R2     ;SAVE THE RESULTS OF THE SECOND MASTER CLEAR
ADD R1, R2          ;ADD THE RESULTS OF THE 2 CLEARS
                    ;NOTE: ONE SHOULD BE 100 - MICRO TESTS NOT
                    ;ENABLED AND ONE SHOULD BE 200 - MICRO TESTS
                    ;SUCCESSFULLY RUN.

TRAP C$ESCAPE
WORD L10031-
    
```

```

5196 025516 122702 000300      CMPB   #300,R2      ;WAS THE MICROTEST COMPLETED?
5197 025522 001404      BEQ    5$          ;IF YES - OK
5198 025524      ERRDF  3,EMT3,ERRG3 ;MICROTEST NOT COMPLETED
(4) 025524 104455
(5) 025526 000003      TRAP   C$ERDF
(5) 025530 025536      .WORD 3
(5) 025532 015204      .WORD EMT3
5199 025534      .WORD ERRG3
5200 025534      5$:
(3) 025534      ENDTST
(3) 025534 104401      L10031:
5201      TRAP   C$ETST
5202 025536 044515 051103 052117 EMT3: .ASCIZ /MICROTEST NOT COMPLETED/
    025544 051505 020124 047516
    025552 020124 047503 050115
    025560 042514 042524 000104
5203      .EVEN
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224 025566
(3) 025566
5225 025566
(3) 025566
(3) 025566 104402
5226 025570
(1)
(1) 025570 004737 011064
(1)
5227
5228
5229 025574
(3) 025574 104410
(3) 025576 000244
5230
5231 025600
(1)
(1) 025600 004737 011262
(1) 025604 000000
    
```

```

.SBTTL      TEST 4 - BASE IN COMMAND
*****
*          TEST 4 - DMR-11
* BASE IN COMMANDS
* SUBTEST 1 - ISSUE A BASE IN - DMR MODE.
*           ENSURE THAT THE DMR MODE BIT (BIT 4) IS SET IN
*           THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
*           MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
* SUBTEST 2 - ISSUE A BASE IN - DMC MODE.
*           ENSURE THAT THE DMC MODE BIT (BIT 4) IS CLEAR IN
*           THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
*           MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
*****
BGNTST
BGNSUB
T4::
T4.1:
TRAP   C$BSUB
CLEAR      ;MACRO FOR MASTER CLEAR COMMAND
          ;**** MACRO EXPANSION ****
          ;ISSUE A DMR MASTER CLEAR
          ;****
          ;****
ESCAPE TST ;IF ERROR, BR TO TEST END
          TRAP   C$ESCAPE
          .WORD  L10032-
BASEIN 0,BASE,DMR ;BASE IN COMMAND WITH NO MAINTENANCE,
          ;BASE=BASE TABLE ADDRESS, AND DMR-11 MODE
          ;**** MACRO EXPANSION ****
          ;CALL BASE IN ROUTINE
          ;MAINTENANCE MODE BITS TO SET IN BSEL1
    
```

```

(1) 025606 002630          .WORD  BASE  :BASE TABLE ADDRESS
(1) 025610 000522          .WORD  DMR   :MODE
(1)                                     :*****          *****
5232
5233 025612          ESCAPE TST          ;IF ERROR, BR TO TEST END
(3) 025612 104410          TRAP    C$ESCAPE
(3) 025614 000226          .WORD  L10032-.
5234 025616          SHUTDN
(1)
(1) 025616 004737 012560    JSR    PC, $HALT          ;***** MACRO EXPANSION *****
(1)                                     ;DMR HALT ROUTINE.
5235 025622 132737 000020 002722    BITB  #BIT4,BASE+ISP7    ;*****          *****
5236                                     ;SEE IF THE DMR MODE BIT IS SET IN THE
5237                                     ;DMR SCRATCH PAD REGISTER 7 (BASE TABLE
5238 025630 001004          BNE    10$              ;LOCATION CONTAINS AN IMAGE OF SP7)
5239 025632          ERRDF  20,EMT4          ;OK IF SET - BR
(4) 025632 104455          TRAP    C$ERDF
(5) 025634 000024          .WORD  20
(5) 025636 026264          .WORD  EMT4
(5) 025640 000000          .WORD  0
5240 025642          10$:
5241
5242                                     ;CHECK MESSAGE EXCHANGE VALUES
5243 025642 105737 002672    TSTB  BASE+R            ;IN THE BASE TABLE.
5244 025646 001015          BNE    20$              ; #R (MESSAGE RECEIVED) = 0?
5245 025650 105737 002673    TSTB  BASE+N            ;ERROR IF NON ZERO
5246 025654 001012          BNE    20$              ; #N (MESSAGE TRANSMITTED) = 0?
5247 025656 105737 002674    TSTB  BASE+A            ;ERROR IF NON ZERO
5248 025662 001007          BNE    20$              ; #A (MESSAGE ACKNOWLEDGED) = 0?
5249 025664 122737 000001 002675    CMPB  #1,BASE+T        ; #T (NEXT MESSAGE # TRANSMITTED) = 1?
5250 025672 001003          BNE    20$              ;ERROR IF NOT EQUAL TO 1.
5251 025674 105737 002676    TSTB  BASE+X            ; #X (LAST MESSAGE TRANSMITTED) = 0?
5252 025700 001404          BEQ    30$
5253 025702          20$:
5254 025702          ERRDF  20,EMT5,ERRT1
(4) 025702 104455          TRAP    C$ERDF
(5) 025704 000024          .WORD  20
(5) 025706 026311          .WORD  EMT5
(5) 025710 026044          .WORD  ERRT1
5255 025712          30$:
5256 025712          ENDSUB
(3) 025712          L10033:
(3) 025712 104403          TRAP    C$ESUB
5257
5258 025714          BGNSUB
(3) 025714          T4.2:
(3) 025714 104402          TRAP    C$BSUB
5259 025716          CLEAR          ;MACRO FOR MASTER CLEAR COMMAND
(1)                                     ;***** MACRO EXPANSION *****
(1) 025716 004737 011064    JSR    PC, $MSCLR        ;ISSUE A DMR MASTER CLEAR
(1)                                     ;*****          *****
5260
5261 025722          ESCAPE TST          ;IF ERROR, BR TO TEST END
(3) 025722 104410          TRAP    C$ESCAPE
(3) 025724 000116          .WORD  L10032-.
5262
    
```

```

5263                                     ;BASE IN COMMAND WITH NO MAINTENANCE
5264 025726 BASEIN 0,BASE,0                ;AND DMC MODE.
(1)                                     ;**** MACRO EXPANSION ****
(1) 025726 004737 011262 JSR PC,$BASEI    ;CALL BASE IN ROUTINE
(1) 025732 000000                .WORD 0    ;MAINTENANCE MODE BITS TO SET IN BSEL1
(1) 025734 002630                .WORD BASE ;BASE TABLE ADDRESS
(1) 025736 000000                .WORD 0    ;MODE
(1)                                     ;****                ****
5265
5266 025740 ESCAPE TST                    ;IF ERROR, BR TO TEST END
(3) 025740 104410                TRAP C$ESCAPE
(3) 025742 000100                .WORD L10032-.
5267 025744 SHUTDN
(1)
(1) 025744 004737 012560 JSR PC,$HALT ;**** MACRO EXPANSION ****
(1)                                     ;DMR HALT ROUTINE.
5268 025750 132737 000020 002722 BITB #BIT4,BASE+ISP7 ;****                ****
5269                                     ;SEE IF THE DMR MODE BIT IS CLEAR IN THE
5270                                     ;DMR SCRATCH PAD REGISTER 7 (BASETABLE
5271 025756 001404 BEQ 10$                ;LOCATION CONTAINS AN IMAGE OF SP7)
5272 025760 ERRDF 20,EMT6                ;OK IF CLEAR - BR
(4) 025760 104455                TRAP C$ERDF
(5) 025762 000024                .WORD 20
(5) 025764 026357                .WORD EMT6
(5) 025766 000000                .WORD 0
5273 025770 10$:
5274                                     ;CHECK MESSAGE EXCHANGE VALUES
5275                                     ;IN THE BASE TABLE.
5276 025770 105737 002672 TSTB BASE+R    ; #R (MESSAGE RECEIVED) = 0?
5277 025774 001015 BNE 20$        ;ERROR IF NON ZERO
5278 025776 105737 002673 TSTB BASE+N    ; #N (MESSAGE TRANSMITTED) = 0?
5279 026002 001012 BNE 20$        ;ERROR IF NON ZERO
5280 026004 105737 002674 TSTB BASE+A    ; #A (MESSAGE ACKNOWLEDGED) = 0?
5281 026010 001007 BNE 20$        ;ERROR IF NON ZERO
5282 026012 122737 000001 002675 CMPB #1,BASE+T ; #T (NEXT MESSAGE # TRANSMITTED) = 1?
5283 026020 001003 BNE 20$        ;ERROR IF NOT EQUAL TO 1.
5284 026022 105737 002676 TSTB BASE+X    ; #X (LAST MESSAGE TRANSMITTED) = 0?
5285 026026 001404 BEQ 30$
5286 026030 20$:
5287 026030 ERRDF 20,EMT5,ERRT1
(4) 026030 104455                TRAP C$ERDF
(5) 026032 000024                .WORD 20
(5) 026034 026311                .WORD EMT5
(5) 026036 026044                .WORD ERRT1
5288 026040 30$:
5289 026040 ENDSUB
(3) 026040 L10034:
(3) 026040 104403 TRAP C$ESUB
5290
5291 026042 ENDTST
(3) 026042 L10032:
(3) 026042 104401 TRAP C$ETST
5292
5293 026044 BGNMSG ERRT1
(3) 026044
5294 026044 105737 002672 TSTB BASE+R    ;IS #R = 0?
  
```

5295	026050	001413		BEQ	1\$		:OK - IF ZERO		
5296	026052			PRINTB	#FMT5,<B,BASE+R>		:PRINT #R		
(8)	026052	005046						CLR	-(SP)
(8)	026054	153716	002672					BISB	BASE+R,(SP)
(7)	026060	012746	026423					MOV	#FMT5,-(SP)
(6)	026064	012746	000002					MOV	#2,-(SP)
(3)	026070	010600						MOV	SP,R0
(4)	026072	104414						TRAP	C\$PNTB
(4)	026074	062706	000006					ADD	#6,SP
5297	026100			1\$:					
5298	026100	105737	002673	TSTB	BASE+N		:IS #N = 0?		
5299	026104	001413		BEQ	2\$		:OK - IF ZERO		
5300	026106			PRINTB	#FMT6,<B,BASE+N>		:PRINT #N		
(8)	026106	005046						CLR	-(SP)
(8)	026110	153716	002632					BISB	BASE+2,(SP)
(7)	026114	012746	026454					MOV	#FMT6,-(SP)
(6)	026120	012746	000002					MOV	#2,-(SP)
(3)	026124	010600						MOV	SP,R0
(4)	026126	104414						TRAP	C\$PNTB
(4)	026130	062706	000006					ADD	#6,SP
5301	026134			2\$:					
5302									
5303	026134	105737	002674	TSTB	BASE+A		:IS #A = 0?		
5304	026140	001413		BEQ	3\$		:OK - IF ZERO		
5305	026142			PRINTB	#FMT7,<B,BASE+A>		:PRINT #A		
(8)	026142	005046						CLR	-(SP)
(8)	026144	153716	002674					BISB	BASE+A,(SP)
(7)	026150	012746	026505					MOV	#FMT7,-(SP)
(6)	026154	012746	000002					MOV	#2,-(SP)
(3)	026160	010600						MOV	SP,R0
(4)	026162	104414						TRAP	C\$PNTB
(4)	026164	062706	000006					ADD	#6,SP
5306	026170			3\$:					
5307	026170	122737	000001	CMPB	#1,BASE+T		:IS #T = 1?		
5308	026176	001413	002675	BEQ	4\$		:OK - IF ONE		
5309	026200			PRINTB	#FMT8,<B,BASE+T>		:PRINT #T		
(8)	026200	005046						CLR	-(SP)
(8)	026202	153716	002675					BISB	BASE+T,(SP)
(7)	026206	012746	026536					MOV	#FMT8,-(SP)
(6)	026212	012746	000002					MOV	#2,-(SP)
(3)	026216	010600						MOV	SP,R0
(4)	026220	104414						TRAP	C\$PNTB
(4)	026222	062706	000006					ADD	#6,SP
5310	026226			4\$:					
5311	026226	105737	002676	TSTB	BASE+X		:IS #X = 0?		
5312	026232	001413		BEQ	5\$		:OK - IF ZERO		
5313	026234			PRINTB	#FMT9,<B,BASE+X>		:PRINT #X		
(8)	026234	005046						CLR	-(SP)
(8)	026236	153716	002676					BISB	BASE+X,(SP)
(7)	026242	012746	026601					MOV	#FMT9,-(SP)
(6)	026246	012746	000002					MOV	#2,-(SP)
(3)	026252	010600						MOV	SP,R0
(4)	026254	104414						TRAP	C\$PNTB
(4)	026256	062706	000006					ADD	#6,SP
5314	026262			5\$:					
5315	026262			ENDMSG					



L10035: TRAP C\$MSG

(3)	026262					
(3)	026262	104423				
5316						
5317	026264	046504	020122	047515	EMT4:	.ASCIZ /DMR MODE BIT NOT SET/
	026272	042504	041040	052111		
	026300	047040	052117	051440		
	026306	052105	000			
5318	026311	104	041504	050115	EMT5:	.ASCIZ /DDCMP MESSAGE VARIABLE(S) NOT CORRECT/
	026316	046440	051505	040523		
	026324	042507	053040	051101		
	026332	040511	046102	024105		
	026340	024523	047040	052117		
	026346	041440	051117	042522		
	026354	052103	000			
5319	026357	104	041515	046440	EMT6:	.ASCIZ /DMC MODE - DMR MODE BIT NOT CLEARED/
	026364	042117	020105	020055		
	026372	046504	020122	047515		
	026400	042504	041040	052111		
	026406	047040	052117	041440		
	026414	042514	051101	042105		
	026422	000				
5320						
5321	026423	045	021501	020122	FMT5:	.ASCIZ /%A#R (MSG. RCVD) = %D3%N/
	026430	046450	043523	020056		
	026436	041522	042126	020051		
	026444	020075	042045	022463		
	026452	000116				
5322	026454	040445	047043	024040	FMT6:	.ASCIZ /%A#N (MSG. XMIT) = %D3%N/
	026462	051515	027107	054040		
	026470	044515	024524	036440		
	026476	022440	031504	047045		
	026504	000				
5323	026505	045	021501	020101	FMT7:	.ASCIZ /%A#A (MSG. ACK) = %D3%N/
	026512	046450	043523	020056		
	026520	041501	024513	020040		
	026526	020075	042045	022463		
	026534	000116				
5324	026536	040445	052043	024040	FMT8:	.ASCIZ /%A#T (NEXT MSG TO XMIT) = %D3%N/
	026544	042516	052130	046440		
	026552	043523	052040	020117		
	026560	046530	052111	020051		
	026566	020040	036440	022440		
	026574	031504	047045	000		
5325	026601	045	021501	020130	FMT9:	.ASCIZ /%A#X (LAST COMPLETED XMIT) = %D3%N/
	026606	046050	051501	020124		
	026614	047503	050115	042514		
	026622	042524	020104	046530		
	026630	052111	020051	020075		
	026636	042045	022453	000116		

5326  
 5327  
 5328  
 5329  
 5330  
 5331  
 5332

.EVEN

```

5333 .SBTTL TEST 5 - DMR COMMANDS
5334
5335 :*****
5336 :* TEST 5 - DMR-11
5337 :* DMR COMMANDS
5338 :* SUBTEST 1 - ISSUE AN ENABLE EXTENDED ERROR COMMAND AND CHECK THAT
5339 :* THE EXT. ENABLE BIT IS SET IS SCRATCH PAD 13. THEN
5340 :* DISABLE EXTENDED ERROR AND CHECK THAT THE ENABLE BIT
5341 :* IS CLEAR.
5342 :* SUBTEST 2 - SET REP/SEL TIMER VALUE AND SET THE DMR THRESHOLD
5343 :* VALUES. CHECK THAT THE VALUES ARE CORRECT IN
5344 :* THE BASE TABLE AFTER HALTING THE DMR.
5345 :*
5346 :*
5347 :*****
5348 026644 BGNTST
5349 (3) 026644 BGNSUB T5::
5350 (3) 026644 104402 T5.1: TRAP C$BSUB
5351 (1) 026646 004737 011064 CLEAR ;MACRO FOR MASTER CLEAR COMMAND
5352 (1) 026646 004737 011064 JSR PC, $MSCLR ;**** MACRO EXPANSION ****
5353 (1) 026646 004737 011064 ;ISSUE A DMR MASTER CLEAR
5354 (1) 026646 004737 011064 ;****
5355 (1) 026652 104410 ESCAPE TST ;IF ERROR, BR TO TEST END
5356 (3) 026652 104410 TRAP C$ESCAPE
5357 (3) 026654 000330 .WORD L10036-.
5358 026656 BASEIN ;BASE IN COMMAND WITH LINE UNIT LOOP,
5359 (1) 026656 004737 011262 JSR PC, $BASEI ;**** MACRO EXPANSION ****
5360 (1) 026662 004000 .WORD LPLU ;CALL BASE IN ROUTINE WITH DEFAULTS
5361 (1) 026664 002630 .WORD BASE ;SET LINE UNIT LOOP
5362 (1) 026666 000522 .WORD DMR ;BASE TABLE ADDRESS
5363 (1) 026666 000522 ;DMR-11 MODE
5364 (1) 026666 000522 ;****
5365 (1) 026670 104410 ESCAPE TST ;IF ERROR, BR TO TEST END
5366 (3) 026670 104410 TRAP C$ESCAPE
5367 (3) 026672 000312 .WORD L10036-.
5368 026674 DMRIN EXERR ;ENABLE EXTENDED ERROR NOTIFICATION
5369 (1) 026674 004737 012064 JSR PC, $DMRIN ;**** MACRO EXPANSION ****
5370 (1) 026700 000006 .WORD EXERR ;CALL DMR MODE INPUT ROUTINE
5371 (1) 026702 000000 .WORD 0 ;INPUT COMMAND
5372 (1) 026704 000000 .WORD 0 ;NO SEL4
5373 (1) 026704 000000 .WORD 0 ;NO SEL6
5374 (1) 026704 000000 ;****
5375 (1) 026706 104410 ESCAPE TST ;IF ERROR, BR TO TEST END
5376 (3) 026706 104410 TRAP C$ESCAPE
5377 (3) 026710 000274 .WORD L10036-.
5378 026712 SHUTDN ;HALT THE DMR
5379 (1) 026712 004737 012560 JSR PC, $HALT ;**** MACRO EXPANSION ****
5380 (1) 026712 004737 012560 ;DMR HALT ROUTINE.
5381 (1) 026712 004737 012560 ;****
5382 026716 ESCAPE TST ;IF ERROR, BR TO TEST END

```

```

(3) 026716 104410
(3) 026720 000264
5362 026722 132737 000001 002726 BITB #BIT0,BASE+ISP13 ;CHECK EXT ENABLE BIT IN THE BASE TABLE.
5363 ;IMAGE OF SCRATCH PAD 13.
5364 026730 001005 BNE 10$ ;BIT SET - OK.
5365 026732 104455 ERRDF 24,EMT7 ;ERROR EXT ENABLE CLEAR
(4) 026732 104455 TRAP C$ERDF
(5) 026734 000030 .WORD 24
(5) 026736 027514 .WORD EMT7
(5) 026740 000000 .WORD 0
5366 026742 000430 BR 20$
5367 026744 10$: BASEIN LPLU,BASE,RES!DMR ;BASE IN COMMAND WITH RESUME SET.
5368 026744 (1) ;**** MACRO EXPANSION ****
(1) 026744 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE
(1) 026750 004000 .WORD LPLU ;MAINTENANCE MODE BITS TO SET IN BSEL1
(1) 026752 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 026754 010522 .WORD RES!DMR ;MODE
(1) ;****
5369 DMRIN DXERR ;DISABLE EXTENDED ERROR NOTIFICATION.
5370 026756 (1) ;**** MACRO EXPANSION ****
(1) 026756 004737 012064 JSR PC, $DMRIN ;CALL DMR MODE INPUT ROUTINE
(1) 026762 000007 .WORD DXERR ;INPUT COMMAND
(1) 026764 000000 .WORD 0 ;NO SEL4
(1) 026766 000000 .WORD 0 ;NO SEL6
(1) ;****
5371 ESCAPE TST ;IF ERROR, BR TO TEST END
5372 026770 (3) 026770 104410 TRAP C$ESCAPE
(3) 026772 000212 .WORD L10036-.
5373 026774 SHUTDN ;HALT THE DMR
(1) ;**** MACRO EXPANSION ****
(1) 026774 004737 012560 JSR PC, $HALT ;DMR HALT ROUTINE.
(1) ;****
5374 027000 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 027000 104410 TRAP C$ESCAPE
(3) 027002 000202 .WORD L10036-.
5375 027004 132737 000001 002726 BITB #BIT0,BASE+ISP13 ;CHECK EXT ENABLE BIT IN THE BASE TABLE.
5376 ;IMAGE OF SCRATCH PAD 13.
5377 027012 001404 BEQ 20$ ;IF CLEAR OK
5378 027014 104455 ERRDF 24,EMT7 ;ERROR EXT ENABLE SET
(4) 027014 104455 TRAP C$ERDF
(5) 027016 000030 .WORD 24
(5) 027020 027514 .WORD EMT7
(5) 027022 000000 .WORD 0
5379 027024 20$: ENDSUB
5380 027024 (3) 027024 104403 L10037: TRAP C$ESUB
(3) 027024 104403
5381 BGNSUB
5382 027026 (3) 027026 104402 T5.2: TRAP C$SUB
(3) 027026 104402
5383 027030 CLEAR ;MACRO FOR MASTER CLEAR COMMAND
(1) ;**** MACRO EXPANSION ****

```

```

(1) 027030 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
(1) ;*****
5384
5385 027034 ESCAPE TST ;IF ERROR, BR TO TEST END
(3) 027034 104410 TRAP C$ESCAPE
(3) 027036 000146 .WORD L10036-.
5386
5387 027040 BASEIN ;BASE IN COMMAND WITH LINE UNIT LOOP,
(1) ;***** MACRO EXPANSION *****
(1) 027040 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 027044 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 027046 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 027050 000522 .WORD DMR ;DMR-11 MODE
(1) ;*****
5388
5389 027052 ESCAPE TST ;IF ERROR, BR TO TEST END
(3) 027052 104410 TRAP C$ESCAPE
(3) 027054 000130 .WORD L10036-.
5390 027056 DMRIN TIMER,0,54 ;SET REP/SELECT TIMER VALUE
(1) ;***** MACRO EXPANSION *****
(1) 027056 004737 012064 JSR PC, $DMRIN ;CALL DMR MODE INPUT ROUTINE
(1) 027062 000012 .WORD TIMER ;INPUT COMMAND
(1) 027064 000000 .WORD 0 ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
(1) 027066 000054 .WORD 54 ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
(1) ;*****
5391
5392 027070 ESCAPE TST ;IF ERROR, BR TO TEST END
(3) 027070 104410 TRAP C$ESCAPE
(3) 027072 000112 .WORD L10036-.
5393
5394 ;SET THRESHOLD VALUES AS FOLLOWS:
5395 ;BSEL4 = NAKS RECEIVED (3)
5396 ;BSEL5 = NAKS TRANSMITTED (13)
5397 ;BSEL6 = REP/SEL SENT (15)
5398 ;BSEL7 = NO BUFFFER (4)
5399 027074 DMRIN THRESH,5403,2015
(1) ;***** MACRO EXPANSION *****
(1) 027074 004737 012064 JSR PC, $DMRIN ;CALL DMR MODE INPUT ROUTINE
(1) 027100 000013 .WORD THRESH ;INPUT COMMAND
(1) 027102 005403 .WORD 5403 ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
(1) 027104 002015 .WORD 2015 ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
(1) ;*****
5400
5401 027106 ESCAPE TST ;IF ERROR, BR TO TEST END
(3) 027106 104410 TRAP C$ESCAPE
(3) 027110 000074 .WORD L10036-.
5402 027112 SHUTDN ;HALT THE DMR.
(1) ;***** MACRO EXPANSION *****
(1) 027112 004737 012560 JSR PC, $HALT ;DMR HALT ROUTINE.
(1) ;*****
5403 027116 ESCAPE TST ;IF ERROR, BR TO TEST END
(3) 027116 104410 TRAP C$ESCAPE
(3) 027120 000064 .WORD L10036-.
5404 027122 122737 000054 002705 CMPB #54,BASE+PRETIM ;CHECK REP/SEL TIME IN BASE TABLE.
5405 027130 001020 BNE 10$ ;IF NOT 54, BR TO ERROR.
5406 027132 122737 000015 002714 CMPB #15,BASE+TH3L ;CHECK REP. THRESH. IN BASE TABLE.

```

```

5407 027140 001014      BNE      10$      ;IF NOT 15, BR TO ERROR.
5408 027142 122737 000003 002710  CMPB    #3,BASE+TH1L ;CHECK NAK RCVD. THRESH. IN BASE TABLE.
5409 027150 001010      BNE      10$      ;IF NOT 3, BR TO ERROR.
5410 027152 122737 000013 002712  CMPB    #13,BASE+TH2L ;CHECK NAK SENT THRESH. IN BASE TABLE.
5411 027160 001004      BNE      10$      ;IF NOT 13, BR TO ERROR
5412 027162 122737 000004 002716  CMPB    #4,BASE+TH4L ;CHECK NO BUF. THRESH. IN BASE TABLE.
5413 027170 001404      BEQ      20$      ;IF 4, ALL CHECKS OK - EXIT
5414 027172
5415 027172      10$:      ERRDF    24,EMT8,ERRT3
(4) 027172 104455
(5) 027174 000030
(5) 027176 027545
(5) 027200 027206
5416 027202
5417 027202      20$:      ENDSUB
(3) 027202
(3) 027202 104403
5418 027204      ENDTST
(3) 027204
(3) 027204 104401
5419
5420 027206      BGNMSG  ERRT3
(3) 027206
5421 027206      PRINTB #FMG1,@SELO,@SEL2 ;PRINT SELO AND SEL2
(9) 027206 017746 153022
(8) 027212 017746 153014
(7) 027216 012746 016276
(6) 027222 012746 000003
(3) 027226 010600
(4) 027230 104414
(4) 027232 062706 000010
5422 027236      PRINTB #FMT11,<B,BASE+ISP13> ;PRINT OUT THE IMAGE OF SCRATCH PAD 13.
(8) 027236 005046
(8) 027240 153716 002726
(7) 027244 012746 027602
(6) 027250 012746 000002
(3) 027254 010600
(4) 027256 104414
(4) 027260 062706 000006
5423 027264 122737 000054 002705  CMPB    #54,BASE+PRETIM ;IS REP/SEL TIME OK?
5424 027272 001413      BEQ      1$      ;BR IF OK
5425 027274      PRINTB #FMT12,<B,BASE+PRETIM> ;PRINT IT OUT.
(8) 027274 005046
(8) 027276 153716 002705
(7) 027302 012746 027633
(6) 027306 012746 000002
(3) 027312 010600
(4) 027314 104414
(4) 027316 062706 000006
5426 027322
5427 027322 122737 000003 002710  1$:      CMPB    #3,BASE+TH1L ;IS NAK RCVD OK?
5428 027330 001413      BEQ      2$      ;BR IF OK.
5429 027332      PRINTB #FMT13,<B,BASE+TH1L> ;PRINT IT OUT
(8) 027332 005046
(8) 027334 153716 002710
(7) 027340 012746 027670
    
```

```

TRAP    C$ERDF
.WORD   24
.WORD   EMT8
.WORD   ERRT3

L10040: TRAP    C$ESUB

L10036: TRAP    C$ETST

ERRT3::
MOV     @SEL2,-(SP)
MOV     @SELO,-(SP)
MOV     #FMG1,-(SP)
MOV     #3,-(SP)
MOV     SP,R0
TRAP    C$PNTB
ADD     #10,SP
CLR     -(SP)
BISB   BASE+ISP13,(SP)
MOV     #FMT11,-(SP)
MOV     #2,-(SP)
MOV     SP,R0
TRAP    C$PNTB
ADD     #6,SP

CLR     -(SP)
BISB   BASE+PRETIM,(SP)
MOV     #FMT12,-(SP)
MOV     #2,-(SP)
MOV     SP,R0
TRAP    C$PNTB
ADD     #6,SP

CLR     -(SP)
BISB   BASE+TH1L,(SP)
MOV     #FMT13,-(SP)
    
```

(6)	027344	012746	000002					MOV	#2,-(SP)
(3)	027350	010600						MOV	SP,R0
(4)	027352	104414						TRAP	C\$PNTB
(4)	027354	062706	000006					ADD	#6,SP
5430	027360								
5431	027360	122737	000013	002712	2\$:				
5432	027366	001413				CMPB	#13,BASE+TH2L		;IS NAK SENT OK?
5433	027370					BEQ	3\$		;BR IF OK.
(8)	027370	005046				PRINTB	#FMT14,<B,BASE+TH2L>		;PRINT IT OUT
(8)	027372	153716	002712					CLR	-(SP)
(7)	027376	012746	027725					BISB	BASE+TH2L,(SP)
(6)	027402	012746	000002					MOV	#FMT14,-(SP)
(3)	027406	010600						MOV	#2,-(SP)
(4)	027410	104414						MOV	SP,R0
(4)	027412	062706	000006					TRAP	C\$PNTB
5434	027416							ADD	#6,SP
5435	027416	122737	000015	002714	3\$:				
5436	027424	001413				CMPB	#15,BASE+TH3L		;IS REP LEVEL OK?
5437	027426					BEQ	4\$		;BR IF OK.
(8)	027426	005046				PRINTB	#FMT15,<B,BASE+TH3L>		;PRINT IT OUT
(8)	027430	153716	002714					CLR	-(SP)
(7)	027434	012746	027762					BISB	BASE+TH3L,(SP)
(6)	027440	012746	000002					MOV	#FMT15,-(SP)
(3)	027444	010600						MOV	#2,-(SP)
(4)	027446	104414						MOV	SP,R0
(4)	027450	062706	000006					TRAP	C\$PNTB
5438	027454							ADD	#6,SP
5439	027454	122737	000004	002716	4\$:				
5440	027462	001413				CMPB	#4,BASE+TH4L		;IS NO BUFFER LEVEL OK?
5441	027464					BEQ	5\$		;BR IF OK.
(8)	027464	005046				PRINTB	#FMT16,<B,BASE+TH4L>		;PRINT IT OUT
(8)	027466	153716	002716					CLR	-(SP)
(7)	027472	012746	030017					BISB	BASE+TH4L,(SP)
(6)	027476	012746	000002					MOV	#FMT16,-(SP)
(3)	027502	010500						MOV	#2,-(SP)
(4)	027504	104414						MOV	SP,R0
(4)	027506	062706	000006					TRAP	C\$PNTB
5442	027512				5\$:			ADD	#6,SP
5443	027512				ENDMSG				
(3)	027512								
(3)	027512	104423						L10041:	TRAP
5444									C\$MSG
5445									
5446	027514	054105	027124	042440	EMT7:		.ASCIZ	/EXT. ERROR BIT INCORRECT/	
	027522	051122	051117	041040					
	027530	052111	044440	041516					
	027536	051117	042522	052103					
	027544	000							
5447	027545	104	051115	046440	EMT8:		.ASCIZ	/DMR MODE INPUT COMMAND ERROR/	
	027552	042117	020105	047111					
	027560	052520	020124	047503					
	027566	046515	047101	020104					
	027574	051105	047522	000122					
5448									
5449	027602	040445	046511	043501	FMT11:		.ASCIZ	/%AIMAGE OF SP 13 = %D3%N/	
	027610	020105	043117	051440					

027616	020120	031461	036440	
027624	022440	031504	047045	
027632	000			
5450	027633	045	051101	050105 FMT12: .ASCIZ /%AREP-SEL TIME VALUE = %D3%N/
	027640	051455	046105	052040
	027646	046511	020105	040526
	027654	052514	020105	020075
	027662	042045	022463	000116
5451	027670	040445	040516	020113 FMT13: .ASCIZ /%ANAK RCVD THRESHOLD = %D3%N/
	027676	041522	042126	052040
	027704	051110	051505	047510
	027712	042114	036440	022440
	027720	031504	047045	000
5452	027725	045	047101	045501 FMT14: .ASCIZ /%ANAK SENT THRESHOLD = %D3%N/
	027732	051440	047105	020124
	027740	044124	042522	044123
	027746	046117	020104	020075
	027754	042045	022463	000116
5453	027762	040445	042522	020120 FMT15: .ASCIZ /%AREP SENT THRESHOLD = %D3%N/
	027770	042523	052116	052040
	027776	051110	051505	047510
	030004	042114	036440	022440
	030012	031504	047045	000
5454	030017	045	047101	020117 FMT16: .ASCIZ /%ANO BUFFER THRESHOLD = %D3%N/
	030024	052502	043106	051105
	030032	052040	051110	051505
	030040	047510	042114	036440
	030046	022440	031504	047045
	030054	000		
5455		030056		

.EVEN

5455  
5456  
5457  
5458  
5459  
5460  
5461  
5462  
5463  
5464  
5465  
5466  
5467  
5468  
5469  
5470  
5471  
5472  
5473  
5474  
5475  
5476  
5477  
5478  
5479  
5480  
(3)

030056  
030056

.SBTTL

TEST 6 - CONTROL IN COMMAND

```
*****  
:* TEST 6 - DMR-11  
:* CONTROL IN COMMAND TEST -  
:* SUBTEST 1 - CONTROL IN, FULL DUPLEX, DDCMP MODE. ENSURE THAT  
:* THE HALF-DUPLEX BIT IS CLEAR IN THE MODEM STATUS WORD,  
:* ALSO ENSURE THAT DDCMP MODE BIT IS SET IN SCRATCH PAD 7.  
:* SUBTEST 2 - CONTROL IN, HALF DUPLEX. ENSURE THAT THE HALF DUPLEX  
:* BIT IS SET.  
:* SUBTEST 3 - CONTROL IN, MAINTENANCE MODE. ENSURE THAT MAINT. MODE  
:* BIT IS SET IN SCRATCH PAD 7.  
:* SUBTEST 4 - CONTROL IN USING SELECTED LOOPBACK. ISSUE A CONTROL IN  
:* USING THE USER SELECTED LOOPBACK. IF THE LOOPBACK IS  
:* NOT CORRECT, DMR RUN MODE ACKNOWLEDGE WILL NOT BE  
:* RECEIVED.  
:*  
:*  
*****  
BGNTST
```

T6::

```

5481 030056          BGNSUB
(3) 030056          T6.1:
(3) 030056 104402          TRAP  C$BSUB
5482 030060          CLEAR          ;MACRO FOR MASTER CLEAR
(1)          ;**** MACRO EXPANSION ****
(1) 030060 004737 011064 JSR      PC, $MSCLR          ;ISSUE A DMR MASTER CLEAR
(1)          ;****
5483          ESCAPE TST          ;IF ERROR, BR TO TEST END.
5484 030064          TRAP  C$ESCAPE
(3) 030064 104410          .WORD  L10042-.
(3) 030066 000404          BASEIN          ;MACRO FOR BASE IN COMMAND
5485 030070          ;**** MACRO EXPANSION ****
(1)          ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 030070 004737 011262 JSR      PC, $BASEI          ;SET LINE UNIT LOOP
(1) 030074 004000          .WORD  LPLU          ;BASE TABLE ADDRESS
(1) 030076 002630          .WORD  BASE          ;DMR-11 MODE
(1) 030100 000522          .WORD  DMR          ;****
5486          ESCAPE TST          ;IF ERROR, BR TO TEST END.
5487 030102          TRAP  C$ESCAPE
(3) 030102 104410          .WORD  L10042-.
(3) 030104 000366          CNTRIN          ;MACRO FOR CONTROL IN (FULL DUPLEX)
5488 030106          ;**** MACRO EXPANSION ****
(1)          ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 030106 004737 011516 JSR      PC, $CNTIN          ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1) 030112 000000          .WORD  0          ;****
5489          ESCAPE TST          ;IF ERROR, BR TO TEST END.
5490 030114          TRAP  C$ESCAPE
(3) 030114 104410          .WORD  L10042-.
(3) 030116 000354          BIS      #RQI!RMODEM,@SELO ;SET RQI AND READ MODEM COMMAND
5491 030120 052777 000057 152104 WAIT     RDI          ;WAIT FOR RDI TO BE SET
5492 030126          ;**** MACRO EXPANSION ****
(1)          ;CALL WAIT ROUTINE
(1) 030126 004737 010266 JSR      PC, $WAIT          ;FLAG THAT WE'RE WAITING FOR RDI
(1) 030132 000000          .WORD  0          ;****
5493 030134 032777 000020 152074 BIT      #BIT4,@SEL4          ;IS THE HDX BIT SET IN MODEM STATUS REG?
5494 030142 001404 BEQ      10$          ;OK - IF BIT CLEAR
5495 030144          ERRDF 21,EMT9          ;ERROR HDX BIT SET
(4) 030144 104455          TRAP  C$ERDF
(5) 030146 000025          .WORD  21
(5) 030150 030474          .WORD  EMT9
(5) 030152 000000          .WORD  0
5496 030154          10$:
5497 030154          WAIT     RQI          ;CLEAR RQI AND WAIT FOR RDI TO CLEAR.
(1)          ;**** MACRO EXPANSION ****
(1) 030154 004737 010702 JSR      PC, $CLRQI          ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
(1)          ;****
5498 030160          SHUTDN          ;HALT DMR
(1)          ;**** MACRO EXPANSION ****
(1) 030160 004737 012560 JSR      PC, $HALT          ;DMR HALT ROUTINE.
(1)          ;****
5499 030164          ESCAPE TST          ;IF ERROR, EXIT.
(3) 030164 104410          TRAP  C$ESCAPE
(3) 030166 000304          .WORD  L10042-.

```



```

5500 030170 132737 000020 002722 BITB #BIT4,BASE+ISP7 ;IS THE DDCMP RUN BIT SET IN IMAGE OF SP 7.
5501 030176 001004 BNE 20$
5502 030200 ERRDF 21,EMT10 ;ERROR DDCMP RUN BIT NOT SET
(4) 030200 104455 TRAP C$ERDF
(5) 030202 000025 .WORD 21
(5) 030204 030524 .WORD EMT10
(5) 030206 000000 .WORD 0
5503 030210 20$:
5504 030210 ENDSUB
(3) 030210
(3) 030210 104403 L10043: TRAP C$ESUB
5505
5506 030212 BGNSUB
(3) 030212
(3) 030212 104402 T6.2: TRAP C$BSUB
5507 030214 BASEIN LPLU,BASE,RES!DMR ;BASE IN WITH RESUME.
(1) ;**** MACRO EXPANSION ****
(1) 030214 004737 011262 JSR PC, $BASE1 ;CALL BASE IN ROUTINE
(1) 030220 004000 .WORD LPLU ;MAINTENANCE MODE BITS TO SET IN BSEL1
(1) 030222 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 030224 010522 .WORD RES!DMR ;MODE
;****
5508
5509 030226 CNTRIN HDX ;CONTROL IN COMMAND WITH HDX.
(1) ;**** MACRO EXPANSION ****
(1) 030226 004737 011516 JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE
(1) 030232 002000 .WORD HDX ;SEL6 - (DUPLEX, MODE)
;****
5510
5511 030234 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 030234 104410 TRAP C$ESCAPE
(3) 030236 000234 .WORD L10042-.
5512 030240 052777 000057 151764 BIS #RQI!RMODEM,@SELO ;SET RQI AND READ MODEM COMMAND
5513 030246 WAIT RDI ;WAIT FOR RDI TO BE SET
(1) ;**** MACRO EXPANSION ****
(1) 030246 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 030252 000000 .WORD 0 ;FLAG THAT WE'RE WAITING FOR RDI
;****
5514 030254 032777 000020 151754 BIT #BIT4,@SEL4 ;IS THE HDX BIT SET IN MODEM STATUS REG?
5515 030262 001004 BNE 10$ ;OK - IF BIT SET
5516 030264 ERRDF 21,EMT11 ;ERROR HDX BIT CLEAR.
(4) 030264 104455 TRAP C$ERDF
(5) 030266 000025 .WORD 21
(5) 030270 030552 .WORD EMT11
(5) 030272 000000 .WORD 0
5517 030274 10$:
5518 030274 SHUTDN ;HALT THE DMR.
(1) ;**** MACRO EXPANSION ****
(1) 030274 004737 012560 JSR PC, $HALT ;DMR HALT ROUTINE.
;****
5519
5520 030300 ENDSUB
(3) 030300
(3) 030300 104403 L10044: TRAP C$ESUB
5521
5522 030302 BGNSUB

```

```

(3) 030302
(3) 030302 104402
5523 030304 CLEAR ;MACRO FOR MASTER CLEAR
(1) ;**** MACRO EXPANSION ****
(1) 030304 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
(1) ;****
5524 ;****
5525 030310 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 030310 104410 TRAP C$ESCAPE
(3) 030312 000160 .WORD L10042-.
5526 030314 BASEIN ;MACRO FOR BASE IN COMMAND
(1) ;**** MACRO EXPANSION ****
(1) 030314 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 030320 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 030322 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 030324 000522 .WORD DMR ;DMR-11 MODE
(1) ;****
5527 ;****
5528 030326 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 030326 104410 TRAP C$ESCAPE
(3) 030330 000142 .WORD L10042-.
5529 030332 CNTRIN MAINT ;MACRO FOR CONTROL IN (MAINT. MODE)
(1) ;**** MACRO EXPANSION ****
(1) 030332 004737 011516 JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE
(1) 030336 000400 .WORD MAINT ;SEL6 - (DUPLEX, MODE)
(1) ;****
5530 ;****
5531 030340 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 030340 104410 TRAP C$ESCAPE
(3) 030342 000130 .WORD L10042-.
5532 030344 SHUTDN ;HALT
(1) ;**** MACRO EXPANSION ****
(1) 030344 004737 012560 JSR PC, $HALT ;DMR HALT ROUTINE.
(1) ;****
5533 030350 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 030350 104410 TRAP C$ESCAPE
(3) 030352 000120 .WORD L10042-.
5534 030354 132737 000002 002722 BITB #BIT1,BASE+ISP7 ;IS THE MAINTENANCE BIT SET IN IMAGE OF SP 7.
5535 030362 001004 BNE 10$
5536 030364 ERRDF 21,EMT12 ;ERROR - MAINT. BIT NOT SET.
(4) 030364 104455 TRAP C$ERDF
(5) 030366 000025 .WORD 21
(5) 030370 030606 .WORD EMT12
(5) 030372 000000 .WORD 0
5537 030374 10$:
5538 030374 ENDSUB
(3) 030374 L10045:
(3) 030374 104403 TRAP C$ESUB
5539 ;
5540 030376 BGNSUB
(3) 030376 T6.4:
(3) 030376 104402 TRAP C$BSUB
5541 ;
5542 030400 CLEAR ;MACRO FOR MASTER CLEAR
(1) ;**** MACRO EXPANSION ****
(1) 030400 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
    
```

```

(1)
5543                                     ;****          ****
5544 030404                               ESCAPE TST                               ;IF ERROR, BR TO TEST END.
(3) 030404 104410                               TRAP C$ESCAPE
(3) 030406 000064                               .WORD L10042-.
5545 030410 005737 002254                     TST DMTURN                               ;IS INTERNAL LOOPBACK REQUESTED?
5546 030414 001004                               BNE 1$                               ;IF NOT, BR
5547 030416 052737 004000 030440             BIS #LPLU,100$                          ;SET LINE UNIT LOOPBACK.
5548 030424 000403                               BR 2$
5549 030426                                     1$:
5550 030426 042737 004000 030440             BIC #LPLU,100$                          ;CLEAR LINE UNIT LOOPBACK.
5551 030434                                     2$:
5552 030434                               CALL $BASEI                               ;BASE IN COMMAND.
5553 030440 000000                               .WORD 0                               ;MAINTENANCE BITS (L. U. LOOPBACK?)
5554 030442 002630                               .WORD BASE                             ;BASE TABLE ADDRESS.
5555 030444 000522                               .WORD DMR                             ;DMR MODE.
5556 030446                               ESCAPE TST                               ;IF ERROR, BR TO TEST END.
(3) 030446 104410                               TRAP C$ESCAPE
(3) 030450 000022                               .WORD L10042-.
5557 030452                               CALL $LOOP                               ;EXTENDED DMR COMMAND TO SET MAINT. BITS
5558                                     ;IF NEEDED. THIS WILL ALLOW MODEM LOOPBACK
5559                                     ;IF THE USER REQUESTED IT.
5560 030456                               ESCAPE TST                               ;IF ERROR, BR TO TEST END.
(3) 030456 104410                               TRAP C$ESCAPE
(3) 030460 000012                               .WORD L10042-.
5561 030462                               CNTRIN                                   ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1)                                     ;**** MACRO EXPANSION ****
(1) 030462 004737 011516                     JSR PC, $CNTIN                          ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 030466 000000                               .WORD 0                               ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1)                                     ;****          ****
5562 030470                               ENDSUB
(3) 030470                                     L10046:
(3) 030470 104403                               TRAP C$ESUB
5563
5564
5565
5566 030472                               ENDTST
(3) 030472                                     L10042:
(3) 030472 104401                               TRAP C$ETST
5567
5568 030474 042110 020130 044502             EMT9: .ASCIZ /HDX BIT SET WHEN IN FDX/
030502 020124 042523 020124
030510 044127 047105 044440
5569 030516 020116 042106 000130             EMT10: .ASCIZ /DDCMP RUN BIT NOT SET/
030524 042104 046503 020120
030532 052522 020116 044502
030540 020124 047516 020124
030546 042523 000124
5570 030552 042110 020130 044502             EMT11: .ASCIZ /HDX BIT NOT SET WHEN IN HDX/
030560 020124 047516 020124
030566 042523 020124 044127
030574 047105 044440 020116
030602 042110 000130
5571 030606 040515 047111 027124             EMT12: .ASCIZ /MAINT. MODE BIT NOT SET/
030614 046440 042117 020105
030622 044502 020124 047516

```

030630 020124 042523 000124

5572 .EVEN  
 5573  
 5574  
 5575  
 5576  
 5577  
 5578  
 5579

.SBTTL TEST 7 - MODEM WRITE COMMAND

5580  
 5581  
 5582  
 5583  
 5584  
 5585  
 5586  
 5587  
 5588  
 5589  
 5590  
 5591  
 5592

```

:*****
:* TEST 7 - DMR-11
:* MODEM WRITE COMMAND
:* SUBTEST 1 - WRITE DATA PATTERNS INTO THE MODEM WRITE REGISTER.
:* ENSURE THAT ON THE NEXT MODEM READ THAT THE
:* MICROCODE RETURNS THE PATTERN WRITTEN INTO BSEL6.
:* SUBTEST 2 - ATTEMPT TO WRITE BOTH THE HALF-DUPLEX BIT AND THE
:* RTS HOLD BIT. THE MICROCODE SHOULD NOT ALLOW THIS
:* TO HAPPEN. WHEN READING THE MODEM STATUS, ONLY
:* THE HALF-DUPLEX SHOULD BE SET.
:*****
    
```

5593 030636  
 (3) 030636

BGNTST

5594  
 5595 030636  
 (3) 030636  
 (3) 030636 104402

5596 030640  
 (1)  
 (1) 030640 004737 011064  
 (1)

5597  
 5598 030644  
 (3) 030644 104410  
 (3) 030646 000232

5599 030650  
 (1)  
 (1) 030650 004737 011262  
 (1) 030654 004000  
 (1) 030656 002630

(1) 030660 000522  
 (1)  
 5600

5601 030662  
 (3) 030662 104410  
 (3) 030664 000214

5602  
 5603 030666 012701 000005  
 5604 030672 012702 031102  
 5605 030676

5606 030676 012237 030712  
 5607 030702 004737 012064  
 5608 030706 000005  
 5609 030710 000377  
 5610 030712 000000

BGNSUB

```

T7::
T7.1:
CLEAR          ;MACRO FOR MASTER CLEAR
                ;**** MACRO EXPANSION ****
                ;ISSUE A DMR MASTER CLEAR
                ;****
JSR PC, $MSCLR ;IF ERROR, BR TO TEST END.
                ;IF ERROR, BR TO TEST END.
                TRAP C$BSUB
ESCAPE TST     ;BASE IN COMMAND.
                ;**** MACRO EXPANSION ****
                ;CALL BASE IN ROUTINE WITH DEFAULTS
                ;SET LINE UNIT LOOP
                ;BASE TABLE ADDRESS
                ;DMR-11 MODE
                ;****
                TRAP C$ESCAPE
                .WORD L10047-.
BASEIN
JSR PC, $BASEI ;BASE IN COMMAND.
                ;**** MACRO EXPANSION ****
                ;CALL BASE IN ROUTINE WITH DEFAULTS
                ;SET LINE UNIT LOOP
                ;BASE TABLE ADDRESS
                ;DMR-11 MODE
                ;****
                TRAP C$ESCAPE
                .WORD L10047-.
                .WORD LPLU
                .WORD BASE
                .WORD DMR
                ;COUNTER
                ;PATTERN TO WRITE INTO MODEM
10$: MOV #5,R1
     MOV #MODEM,R2
     MOV (R2)+,15$ ;WRITE PATTERN
     JSR PC,$DMRIN ;ISSUE DMR MODE COMMAND
                ;WRITE MODEM COMMAD
                .WORD WMODEM
                .WORD 377 ;CLEAR ALL BITS IN BSEL6
                .WORD 0 ;SET THE BITS IN BSEL6 (FROM PATTERN)
15$:
    
```



```

(1) 031030 004737 010266      JSR    PC, $WAIT      ;CALL WAIT ROUTINE
(1) 031034 000000              .WORD  0              ;FLAG THAT WE'RE WAITING FOR RDI
(1)                               ;*****
5637 031036              ESCAPE TST            ;IF ERROR, EXIT TEST.
(3) 031036 104410              TRAP   C$ESCAPE
(3) 031040 000040              .WORD  L10047-.
5638
5639 031042 122777 000020 151170  CMPB   #20,@BSEL6     ;IS ONLY HDX SET?
5640 031050 001406              BEQ    10$            ;IF YES - OK
5641 031052 012703 000021      MOV    #21,R3         ;SAVE THE PATTERN FOR THE ERROR MESSAGE.
5642 031056              ERRDF  22,EMT13,ERRT2
(4) 031056 104455              TRAP   C$ERDF
(5) 031060 000026              .WORD  22
(5) 031062 031146              .WORD  EMT13
(5) 031064 031114              .WORD  ERRT2
5643 031066              10$:
5644 031066              WAIT   RQI           ;CLEAR RQI AND WAIT FOR RDI TO CLEAR.
(1)                               ;***** MACRO EXPANSION *****
(1) 031066 004737 010702      JSR    PC, $CLRQI     ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
(1)                               ;*****
5645 031072              SHUTDN
(1)                               ;***** MACRO EXPANSION *****
(1) 031072 004737 012560      JSR    PC, $HALT      ;DMR HALT ROUTINE.
(1)                               ;*****
5646
5647 031076              ENDSUB
(3) 031076              L10051:
(3) 031076 104403              TRAP   C$ESUB
5648
5649 031100              ENDTST
(3) 031100              L10047:
(3) 031100 104401              TRAP   C$ETST
5650
5651 031102 000000 000376 000001  MODEM: .WORD  0,376,1,252,357 ;PATTERN TO WRITE INTO MODEM
(3) 031110 000252 000357
5652
5653 031114              BGNMSG ERRT2
(3) 031114              PRINTB #FMT19,R3,<B,@BSEL6>
5654 031114              ERRT2::
(9) 031114 005046              CLR    -(SP)
(9) 031116 157716 151116      BISB  @BSEL6,(SP)
(8) 031122 010346              MOV    R3,-(SP)
(7) 031124 012746 031172      MOV    #FMT19,-(SP)
(6) 031130 012746 000003      MOV    #3,-(SP)
(3) 031134 010600              MOV    SP,R0
(4) 031136 104414              TRAP  C$PNTB
(4) 031140 062706 000010      ADD   #10,SP
5655 031144              ENDMSG
(3) 031144              L10052:
(3) 031144 104423              TRAP   C$MSG
5656
5657
5658 031146 051127 052111 020105  EMT13: .ASCIZ  /WRITE MODEM ERROR /
      031154 047515 042504 020115
      031162 051105 047522 020122
      031170 000
    
```

5659 031172  
 5660  
 5661 031172 040445 051127 052117  
 031200 020105 047111 041040  
 031206 042523 033114 020072  
 031214 047445 022463 020101  
 031222 046440 042117 046505  
 031230 043040 051117 040515  
 031236 020124 047111 041040  
 031244 042523 033114 020072  
 031252 047445 022463 000116

.EVEN

FMT19: .ASCIZ /%AWROTE IN BSEL6: %03%A MODEM FORMAT IN BSEL6: %03%N/

5662  
 5663  
 5664  
 5665  
 5666  
 5667  
 5668  
 5669  
 5670  
 5671  
 5672  
 5673  
 5674  
 5675  
 5676  
 5677  
 5678  
 5679  
 5680  
 5681

.EVEN

.SBTTL TEST 8 - NO BUFFER ERROR

```

:*****
:*          TEST 8 - DMR-11
:* SUBTEST 1 - TRANSMIT A BUFFER THREE TIMES WIHOUT ASSIGNING A
:*              RECEIVE BUFFER. BY ASSIGNING A NO BUFFER THRESHOLD
:*              OF THREE, ENSURE THAT A NO BUFFER ERROR IS RECEIVED
:*              AFTER THE THIRD THRANSMISSION.
:* SUBTEST 2 - TRANSMIT A BUFFER WITHOUT A RECEIVE BUFFER.
:*              ASSIGN THE NAKS THRESHOLD OF 3 AND A NO BUFFER
:*              THRESHOLD OF 7. CHECK THAT THE NAKS ERROR COUNT IS
:*              THREE AFTER SHUTDOWN.
:*****
  
```

BGNTST

5682 031260  
 (3) 031260  
 5683 031260  
 (3) 031260  
 (3) 031260 104402  
 5684 031262  
 (1)  
 (1) 031262 004737 011064  
 (1)  
 5685  
 5686 031266  
 (3) 031266 104410  
 (3) 031270 000416  
 5687 031272  
 (1)  
 (1) 031272 004737 011262  
 (1) 031276 004000  
 (1) 031300 002630  
 (1) 031302 000522  
 (1)  
 5688  
 5689 031304  
 (3) 031304 104410  
 (3) 031306 000400  
 5690 031310

BGNSUB

T8::

T8.1:

CLEAR

```

;MACRO FOR MASTER CLEAR
;**** MACRO EXPANSION ****
;ISSUE A DMR MASTER CLEAR
;****
  
```

TRAP C\$BSUB

ESCAPE TST

;IF ERROR, BR TO TEST END.

TRAP C\$ESCAPE  
 .WORD L10053-

BASEIN

```

;MACRO FOR BASE IN COMMAND
;**** MACRO EXPANSION ****
;CALL BASE IN ROUTINE WITH DEFAULTS
;SET LINE UNIT LOOP
;BASE TABLE ADDRESS
;DMR-11 MODE
;****
  
```

ESCAPE TST

;IF ERROR, BR TO TEST END.

TRAP C\$ESCAPE  
 .WORD L10053-

CNTRIN MAINT

;MACRO FOR CONTROL IN (FULL DUPLEX AND MAINT)

```

(1)
(1) 031310 004737 011516 JSR PC, $CONTIN ;**** MACRO EXPANSION ****
(1) 031314 000400 .WORD MAINT ;CALL CONTROL IN ROUTINE
(1) ;SEL6 - (DUPLEX, MODE)
5691 ;****
5692 031316 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 031316 104410 TRAP C$ESCAPE
(3) 031320 000366 .WORD L10053-.
5693
5694 ;SET THRESHOLDS:
5695 ;NAKS RCVD = 377
5696 ;NAKS SENT = 377
5697 ;REP SENT = 377
5698 ;NO BUFFER = 3
5699 031322 DMRIN THRESH,177777,1777
(1)
(1) 031322 004737 012064 JSR PC, $DMRIN ;**** MACRO EXPANSION ****
(1) 031326 000013 .WORD THRESH ;CALL DMR MODE INPUT ROUTINE
(1) 031330 177777 .WORD 177777 ;INPUT COMMAND
(1) 031332 001777 .WORD 1777 ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
(1) ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
5700 ;****
5701 031334 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 031334 104410 TRAP C$ESCAPE
(3) 031336 000350 .WORD L10053-.
5702 031340 012700 000003 MOV #3,R0 ;SET UP A COUNTER
5703 031344 1$: BACCIT ;BA/CC IN COMMAND FOR TRANSMIT
5704 031344 (1) ;**** MACRO EXPANSION ****
(1) 031344 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 031350 000040 .WORD RQI!BACC ;BA/CC IN TRANSMIT COMMAND
(1) 031352 002512 .WORD TBUF ;TRANSMIT BUFFER ADDRESS
(1) 031354 000044 .WORD TCOUNT ;TRANSMIT CHARACTER COUNT
5705 ;****
5706 031356 WAIT RDO ;WAIT FOR RDO TO BE SET
(1) ;**** MACRO EXPANSION ****
(1) 031356 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 031362 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
(1) ;****
5707 031364 ESCAPE TST ;IF RDO NOT SET, BR TO TEST END.
(3) 031364 104410 TRAP C$ESCAPE
(3) 031366 000320 .WORD L10053-.
5708 031370 005300 DEC R0 ;DEC COUNTER
5709 031372 001404 BEQ 10$ ;TRANSMIT FOR 3 TIMES.
5710 031374 042777 000207 150632 BIC #RDO!CMD,@SEL2 ;CLEAR BACC OUT TRANSMIT.
5711 031402 000760 BR 1$ ;TRANSMIT AGAIN
5712 031404 10$:
5713 031404 032777 000001 150622 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
5714 031412 001005 BNE 20$ ;IF YES, PROCEED.
5715 031414 ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT NOT RECEIVED.
(4) 031414 104455 TRAP C$ERDF
(5) 031416 000010 .WORD 8
(5) 031420 020072 .WORD EMG8
(5) 031422 015070 .WORD ERRG2
5716 031424 000410 BR 30$ ;EXIT
  
```



```

5717 031426          20$:
5718 031426 032777 000004 150604 BIT #NOBFR,@SEL6 ;IS THE NO BUFFER FLAG SET?
5719 031434 001004 BNE 30$ ;IF YES - OK, PROCEED.
5720 031436 ERRDF 9,EMG9,ERRG2 ;WE'RE NOT GETTING EXPECTED RESULT
(4) 031436 104455 TRAP C$ERDF
(5) 031440 000011 .WORD 9
(5) 031442 020136 .WORD EMG9
(5) 031444 015070 .WORD ERRG2
5721          ;(EITHER CONTROL OUT OR NOBUF/NAKS)
5722 031446          30$:
5723 031446 042777 000207 150560 BIC #RDO!CMD,@SEL2 ;CLEAR CONTROL OUT
5724 031454 WAIT RDO ;EXPECT ANOTHER BACC OUT.
(1) ;**** MACRO EXPANSION ****
(1) 031454 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 031460 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
(1) ;****
5725 031462 ESCAPE TST ;IF ERROR, BR TO END.
(3) 031462 104410 TRAP C$ESCAPE
(3) 031464 000222 .WORD L10053-.
5726 031466 042777 000207 150540 BIC #RDO!CMD,@SEL2 ;CLEAR BACC OUT.
5727 031474 SHUTDN ;HALT DMR
(1) ;**** MACRO EXPANSION ****
(1) 031474 004737 012560 JSR PC, $HALT ;DMR HALT ROUTINE.
(1) ;****
5728 031500          50$:
5729 031500 ENDSUB
(3) 031500          L10054:
(3) 031500 104403 TRAP C$ESUB
5730          BGNSUB
5731 031502          T8.2:
(3) 031502 104402 TRAP C$BSUB
5732 031504 CLEAR ;MACRO FOR MASTER CLEAR
(1) ;**** MACRO EXPANSION ****
(1) 031504 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
(1) ;****
5733          ESCAPE TST ;IF ERROR, BR TO TEST END.
5734 031510          TRAP C$ESCAPE
(3) 031510 104410 .WORD L10053-.
(3) 031512 000174
5735 031514 BASEIN ;MACRO FOR BASE IN COMMAND
(1) ;**** MACRO EXPANSION ****
(1) 031514 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 031520 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 031522 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 031524 000522 .WORD DMR ;DMR-11 MODE
(1) ;****
5736          ESCAPE TST ;IF ERROR, BR TO TEST END.
5737 031526          TRAP C$ESCAPE
(3) 031526 104410 .WORD L10053-.
(3) 031530 000156
5738 031532 CNTRIN ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1) ;**** MACRO EXPANSION ****
(1) 031532 004737 011516 JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 031536 000000 .WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1) ;****

```

```

5739
5740 031540          ESCAPE TST          ;IF ERROR, BR TO TEST END.
(3) 031540 104410
(3) 031542 000144          TRAP      C$ESCAPE
                          .WORD      L10053-.
5741
5742                          ;SET THRESHOLDS:
5743                          ;NAKS RCVD = 3
5744                          ;NAKS SENT = 3
5745                          ;REP SENT = 377
5746                          ;NO BUFFER = 7
5747 031544          DMRIN THRESH,1403,3777
(1)
(1) 031544 004737 012064      JSR PC, $DMRIN          ;**** MACRO EXPANSION ****
(1) 031550 000013              .WORD THRESH          ;CALL DMR MODE INPUT ROUTINE
(1) 031552 001403              .WORD 1403            ;INPUT COMMAND
(1) 031554 003777              .WORD 3777            ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
                          ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
                          ;****
5748
5749 031556          ESCAPE TST          ;IF ERROR, BR TO TEST END.
(3) 031556 104410
(3) 031560 000126          TRAP      C$ESCAPE
5750 031562          BACCIT              ;BA/CC IN COMMAND FOR TRANSMIT
                          ;**** MACRO EXPANSION ****
(1) 031562 004737 012274      JSR PC, $BACC          ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 031566 000040              .WORD RQI!BACCT       ;BA/CC IN TRANSMIT COMMAND
(1) 031570 002512              .WORD TBUF            ;TRANSMIT BUFFER ADDRESS
(1) 031572 000044              .WORD TCOUNT         ;TRANSMIT CHARACTER COUNT
                          ;****
5751 031574          10$: WAIT RDO          ;WAIT FOR RDO TO BE SET
5752 031574          JSR PC, $WAIT        ;**** MACRO EXPANSION ****
(1) 031574 004737 010266      .WORD 1              ;CALL WAIT ROUTINE
(1) 031600 000001              ;FLAG THAT WE'RE WAITING FOR RDO
                          ;****
5753 031602          ESCAPE TST          ;IF RDO NOT SET, BR TO TEST END.
(3) 031602 104410
(3) 031604 000102          TRAP      C$ESCAPE
5754 031606 032777 000001 150420      BIT #CNTRL,@SEL2     ;IS THIS A CONTROL OUT?
5755 031614 001005          BNE 20$              ;IF YES, PROCEED.
5756 031616          ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT NOT RECEIVED.
(4) 031616 104455          TRAP      C$ERDF
(5) 031620 000010          .WORD      8
(5) 031622 020072          .WORD      EMG8
(5) 031624 015070          .WORD      ERRG2
5757 031626 000410          BR 30$              ;EXIT
5758 031630
5759 031630 032777 000004 150402      20$: BIT #NOBFR,@SEL6 ;IS THE NO BUFFER FLAG SET?
5760 031636 001004          BNE 30$              ;IF YES - OK, PROCEED.
5761 031640          ERRDF 9,EMG9,ERRG2 ;WE'RE NOT GETTING EXPECTED RESULT
(4) 031640 104455          TRAP      C$ERDF
(5) 031642 000011          .WORD      9
(5) 031644 020136          .WORD      EMG9
(5) 031646 015070          .WORD      ERRG2
5762
5763 031650          30$: SHUTDN          ;(EITHER CONTROL OUT OR NOBUF/NAKS)
5764 031650

```

```

(1)
(1) 031650 004737 012560 JSR PC, $HALT ;**** MACRO EXPANSION ****
(1) ;DMR HALT ROUTINE.
5765 031654 123727 002633 000003 CMPB BASE+3,#3 ;****
5766 031662 001004 BNE 35$ ;NAKS REC. - NO BUFFER = 3?
5767 031664 123727 002636 000003 CMPB BASE+6,#3 ;IF NOT ERROR
5768 031672 001404 BEQ 40$ ;NAKS SENT - NO BUFFER = 3?
5769 031674 35$: ;IF OK - SKIP.
5770 031674 ERRDF 23,EMT20,ERRT4
(4) 031674 104455 TRAP C$ERDF
(5) 031676 000027 .WORD 23
(5) 031700 031746 .WORD EMT20
(5) 031702 031710 .WORD ERRT4
5771
5772 031704 40$:
5773 031704 ENDSUB
(3) 031704 L10055:
(3) 031704 104403 TRAP C$ESUB
5774 031706 ENDTST L10053:
(3) 031706 104401 TRAP C$ETST
5775
5776
5777 031710 BGNMSG ERRT4
(3) 031710 ERRT4::
5778 031710 PRINTB #FMG7,<B,BASE+3>,<B,BASE+6>
(9) 031710 005046 CLR -(SP)
(9) 031712 153716 002636 BISB BASE+6,(SP)
(8) 031716 005046 CLR -(SP)
(8) 031720 153716 002633 BISB BASE+3,(SP)
(7) 031724 012746 016557 MOV #FMG7,-(SP)
(6) 031730 012746 000003 MOV #3,-(SP)
(3) 031734 010600 MOV SP,R0
(4) 031736 104414 TRAP C$PNTB
(4) 031740 062706 000010 ADD #10,SP
5779 031744 ENDMSG L10056:
(3) 031744 104423 TRAP C$MSG
5780
5781 031746 040516 051513 042440 EMT20: .ASCIZ /NAKS ERROR/
031754 051122 051117 000 .EVEN
5782
5783
5784
5785
5786
5787
5788 .SBTTL TEST 9 - NON-EXISTENT MEMORY ERROR
5789
5790 ;*****
5791 ;* TEST 9 - DMR-11
5792 ;* NON-EXISTENT MEMORY (NXM) ERROR CHECK
5793 ;* PERFORM DMR COMMANDS USING NXM ADDRESSES; VERIFY THAT NXM ERROR IS
5794 ;* REPORTED IN EACH OF THE FOLLOWING SUBTESTS:
5795 ;* SUBTEST 1 - BASE IN RESUME COMMAND - BASE TABLE ADDRESS IS NXM
5796 ;* SUBTEST 2 - BA/CC IN RECEIVE COMMAND - BA/CC IN ADDRESS IS NXM
  
```

```

5797      ;* SUBTEST 3 - BA/CC IN TRANSMIT COMMAND - BA/CC IN ADDRESS IS NXM
5798      ;*
5799      ;*****
5800 031762  BGNTST
      (3) 031762
5801 031762  BGNSUB
      (3) 031762 104402
5802 031764  CLEAR          ;MASTER CLEAR MACRO
      (1) 031764 004737 011064 JSR PC, $MSCLR      ;**** MACRO EXPANSION ****
      (1)                                     ;ISSUE A DMR MASTER CLEAR
      (1)                                     ;****
5803      ESCAPE TST          ;IF ERROR, BR TO TEST END
5804 031770  BASEIN         ;BASE IN COMMAND - DMR MODE
      (3) 031770 104410      ;**** MACRO EXPANSION ****
      (3) 031772 000500      ;CALL BASE IN ROUTINE WITH DEFAULTS
5805 031774  JSR PC, $BASEI  ;SET LINE UNIT LOOP
      (1) 031774 004737 011262 .WORD LPLU
      (1) 032000 004000      ;BASE TABLE ADDRESS
      (1) 032002 002630      ;DMR-11 MODE
      (1) 032004 000522      ;****
5806      ESCAPE TST          ;IF ERROR, BR TO TEST END
5807 032006  SHUTDN         ;HALT
      (3) 032006 104410      ;**** MACRO EXPANSION ****
      (3) 032010 000462      ;DMR HALT ROUTINE.
5808 032012  JSR PC, $HALT   ;****
      (1) 032012 004737 012560 ESCAPE TST          ;IF ERROR, BR TO TEST END.
      (1)
5809 032016  MOV #CNTRL,ERROR ;THIS FLAG WILL INHIBIT CONTROL OUT
      (3) 032016 104410      ;ERROR REPORTING - BECAUSE WE EXPECT ONE.
      (3) 032020 000452      ;BASE IN RESUME COMMAND WITH NXM BASE TABLE.
5810 032022 012737 000001 002360 BASEIN 0,160000,BIT15!BIT14!RES!DMR
5811      ;**** MACRO EXPANSION ****
5812      JSR PC, $BASEI
      (1) 032030 004737 011262 .WORD 0
      (1) 032034 000000      ;CALL BASE IN ROUTINE
      (1) 032036 160000      ;MAINTENANCE MODE BITS TO SET IN BSEL1
      (1) 032040 150522      ;BASE TABLE ADDRESS
      (1)                                     .WORD BIT15!BIT14!RES!DMR ;MODE
5815      ;****
5816 032042  WAIT RDO        ;WAIT FOR RDO TO BE SET
      (1) 032042 004737 010266 JSR PC, $WAIT
      (1) 032046 000001      ;**** MACRO EXPANSION ****
      (1)                                     ;CALL WAIT ROUTINE
      (1)                                     ;FLAG THAT WE'RE WAITING FOR RDO
5817 032050  BIT #CNTRL,@SEL2 ;IS THERE A CONTROL OUT REPORTED ?
      (4) 032050 032777 000001 150156 BNE 10$
      (5) 032060 104455      ;IF YES, PROCEED.
      (5) 032062 000010      ;EXPECTED CONTROL OUT
                                     TRAP C$ERDF
                                     .WORD 8

```

```

(5) 032064 020072 .WORD EMG8
(5) 032066 015070 .WORD ERRG2
5820 032070 000410 BR 20$ ;EXIT
5821 032072 10$:
5822 032072 032777 000400 150140 BIT #NXM,@SEL6 ;IS THE NXM FLAG SET?
5823 032100 001004 BNE 20$ ;IF YES - ERROR REPORTED CORRECTLY
5824 032102 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 032102 104455 TRAP C$ERDF
(5) 032104 000011 .WORD 9
(5) 032106 020136 .WORD EMG9
(5) 032110 015070 .WORD ERRG2
5825 032112 20$:
5826 032112 042777 000207 150114 BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS
5827 032120 005037 002360 CLR ERROR ;ALLOW ERROR REPORTING
5828 032124 ENDSUB
(3) 032124 L10060:
(3) 032124 104403 TRAP C$ESUB
5829
5830 032126 BGNSUB
(3) 032126 T9.2:
(3) 032126 104402 TRAP C$BSUB
5831 032130 CLEAR ;MACRO FOR MASTER CLEAR
(1) ;**** MACRO EXPANSION ****
(1) 032130 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
(1) ;****
5832
5833 032134 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032134 104410 TRAP C$ESCAPE
(3) 032136 000334 .WORD L10057-.
5834 032140 BASEIN ;MACRO FOR BASE IN COMMAND
(1) ;**** MACRO EXPANSION ****
(1) 032140 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 032144 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 032146 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 032150 000522 .WORD DMR ;DMR-11 MODE
(1) ;****
5835
5836 032152 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032152 104410 TRAP C$ESCAPE
(3) 032154 000316 .WORD L10057-.
5837 032156 CNTRIN ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1) ;**** MACRO EXPANSION ****
(1) 032156 004737 011516 JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 032162 000000 .WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1) ;****
5838
5839 032164 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032164 104410 TRAP C$ESCAPE
(3) 032166 000304 .WORD L10057-.
5840 032170 012737 000001 002360 MOV #CNTRL,ERROR ;INHIBIT CONTROL OUT ERROR REPORTING AGAIN.
5841
5842 ;BA/CC IN REC. COMMAND WITH NXM
5843 ;ADDR = 760000 AND A CHARACTER COUNT = 3.
5844 032176 BACCIR 160000,BIT15!BIT14!RCOUNT
(1) ;**** MACRO EXPANSION ****
(1) 032176 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE

```

```

(1) 032202 000044 .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
(1) 032204 160000 .WORD 160000 ;BUFFER ADDRESS BITS 0-15
(1) 032206 140044 .WORD BIT15!BIT14!RCOUNT ;BA BITS 16/17 AND CHAR. COUNT
(1) ;****
5845 ;****
5846 032210 BACCIT ;BA/CC IN XMIT
(1) ;**** MACRO EXPANSION ****
(1) 032210 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 032214 000040 .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
(1) 032216 002512 .WORD TBUF ;TRANSMIT BUFFER ADDRESS
(1) 032220 000044 .WORD TCOUNT ;TRANSMIT CHARACTER COUNT
(1) ;****
5847 ;****
5848 032222 WAIT RDO ;WAIT FOR RDO
(1) ;**** MACRO EXPANSION ****
(1) 032222 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 032226 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
(1) ;****
5849 032230 032777 000001 147776 BIT #CNTRL,@SEL2 ;IS THERE A CONTROL OUT REPORTED ?
5850 032236 001005 BNE 10$ ;IF YES, PROCEED.
5851 032240 ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
(4) 032240 104455 TRAP C$ERDF
(5) 032242 000010 .WORD 8
(5) 032244 020072 .WORD EMG8
(5) 032246 015070 .WORD ERRG2
5852 032250 000410 BR 20$ ;EXIT
5853 032252 10$:
5854 032252 032777 000400 147760 BIT #NXM,@SEL6 ;IS THE NXM FLAG SET?
5855 032260 001004 BNE 20$ ;IF YES - ERROR REPORTED CORRECTLY
5856 032262 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 032262 104455 TRAP C$ERDF
(5) 032264 000011 .WORD 9
(5) 032266 020136 .WORD EMG9
(5) 032270 015070 .WORD ERRG2
5857
5858 032272 20$:
5859 032272 042777 000207 147734 BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
5860 032300 005037 002360 CLR ERROR ;ENABLE ERROR REPORTING
5861 032304 ENDSUB
(3) 032304 L10061:
(3) 032304 104403 TRAP C$ESUB
5862
5863 032306 BGNSUB
(3) 032306 T9.3:
(3) 032306 104402 TRAP C$BSUB
5864 032310 CLEAR ;MACRO FOR MASTER CLEAR
(1) ;**** MACRO EXPANSION ****
(1) 032310 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
(1) ;****
5865
5866 032314 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032314 104410 TRAP C$ESCAPE
(3) 032316 000154 .WORD L10057-.
5867 032320 BASEIN ;MACRO FOR BASE IN COMMAND
(1) ;**** MACRO EXPANSION ****
(1) 032320 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
  
```

```

(1) 032324 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 032326 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 032330 000522 .WORD DMR ;DMR-11 MODE
(1) ;**** ;****
5868
5869 032332 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032332 104410 TRAP C$ESCAPE
(3) 032334 000136 .WORD L10057-.
5870 032336 CNTRIN ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1) ;**** MACRO EXPANSION ****
(1) 032336 004737 011516 JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 032342 000000 .WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1) ;**** ;****
5871
5872 032344 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032344 104410 TRAP C$ESCAPE
(3) 032346 000124 .WORD L10057-.
5873 032350 BACCIR ;BA/CC IN RCV
(1) ;**** MACRO EXPANSION ****
(1) 032350 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 032354 000044 .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
(1) 032356 002562 .WORD RBUF ;RECEIVE BUFFER
(1) 032360 000044 .WORD RCOUNT ;RECEIVE CHARACTER COUNT
(1) ;**** ;****
5874
5875 032362 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032362 104410 TRAP C$ESCAPE
(3) 032364 000106 .WORD L10057-.
5876 032366 012737 000001 002360 MOV #CNTRL,ERROR ;INHIBIT CONTROL OUT ERROR REPORTING AGAIN.
5877
5878 ;BA/CC IN XMIT COMMAND WITH NXM BUFFER
5879 ;ADDRESS (760000) AND A CHAR. COUNT = 1
5880 032374 BACCIT 160000,BIT15!BIT14!1
(1) ;**** MACRO EXPANSION ****
(1) 032374 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE
(1) 032400 000040 .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
(1) 032402 160000 .WORD 160000 ;BUFFER ADDRESS BITS 0-15
(1) 032404 140001 .WORD BIT15!BIT14!1 ;BA BITS 16 & 17 AND CHAR. COUNT
(1) ;**** ;****
5881
5882 032406 WAIT RDO ;WAIT FOR RDO TO BE SET.
(1) ;**** MACRO EXPANSION ****
(1) 032406 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 032412 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
(1) ;**** ;****
5883 032414 032777 000001 147612 BIT #CNTRL,@SEL2 ;IS THERE A CONTROL OUT REPORTED ?
5884 032422 001005 BNE 10$ ;IF YES, PROCEED.
5885 032424 ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
(4) 032424 104455 TRAP C$ERDF
(5) 032426 000010 .WORD 8
(5) 032430 020072 .WORD EMG8
(5) 032432 015070 .WORD ERRG2
5886 032434 000410 BR 20$ ;EXIT
5887 032436 10$:
5888 032436 032777 000400 147574 BIT #NXM,@SEL6 ;IS THE NXM FLAG SET?
5889 032444 001004 BNE 20$ ;IF YES - ERROR REPORTED CORRECTLY

```

```

5890 032446 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 032446 104455 TRAP C$ERDF
(5) 032450 000011 .WORD 9
(5) 032452 020136 .WORD EMG9
(5) 032454 015070 .WORD ERRG2
5891 032456 20$:
5892 032456 042777 000207 147550 BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
5893 032464 005037 002360 CLR ERROR ;DON'T INHIBIT CONTROL OUT ERRORS
5894 032470 ENDSUB
(3) 032470 L10062:
(3) 032470 104403 TRAP C$ESUB
5895
5896 032472 ENDTST
(3) 032472 L10057:
(3) 032472 104401 TRAP C$ETST
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911

```

.SBTTL TEST 10 - TIME OUT ERROR

```

:*****
:* TEST 10 - DMR-11
:* TIME OUT - FORCE A TIMEOUT AND VERIFY THAT THE ERROR IS REPORTED
:*
:*****
BGNTST

```

```

5912 032474 T10::
(3) 032474 CLEAR ;MACRO FOR MASTER CLEAR
5913 032474 JSR PC, $MSCLR ;**** MACRO EXPANSION ****
(1) ;ISSUE A DMR MASTER CLEAR
(1) 032474 004737 011064 ;****
(1)
5914
5915 032500 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032500 104410 TRAP C$ESCAPE
(3) 032502 000172 .WORD L10063-
5916 032504 BASEIN ;MACRO FOR BASE IN COMMAND
(1) ;**** MACRO EXPANSION ****
(1) 032504 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 032510 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 032512 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 032514 000522 .WORD DMR ;DMR-11 MODE
(1) ;****
5917
5918 ;SET THRESHOLD VALUES AS FOLLOWS:
5919 ;BSEL4 = NAKS RECEIVED (377)
5920 ;BSEL5 = NAKS TRANSMITTED (377)
5921 ;BSEL6 = REP/SEL SENT (1)
5922 ;BSEL7 = NO BUFFFER (377)
5923 032516 DMRIN THRESH,177777,177401
(1)
(1) 032516 004737 012064 JSR PC, $DMRIN ;**** MACRO EXPANSION ****
;CALL DMR MODE INPUT ROUTINE

```



```

(1) 032522 000013          .WORD THRESH :INPUT COMMAND
(1) 032524 177777          .WORD 177777  :SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
(1) 032526 177401          .WORD 177401  :SEL6 VALUE (OR BITS TO SET IN BSEL6)
(1)                                     :*****
5924                                     :*****
5925 032530                ESCAPE TST          ;IF ERROR, BR TO TEST END
(3) 032530 104410          TRAP C$ESCAPE
(3) 032532 000142          .WORD L10063-.
5926 032534                DMRIN TIMER,0,1      ;SET REP/SEL TIMER TO MINIMUM (100 MSECS)
(1)                                     :***** MACRO EXPANSION *****
(1) 032534 004737 012064  JSR PC, $DMRIN      ;CALL DMR MODE INPUT ROUTINE
(1) 032540 000012          .WORD TIMER      ;INPUT COMMAND
(1) 032542 000000          .WORD 0           ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
(1) 032544 000001          .WORD 1           ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
(1)                                     :*****
5927                                     :*****
5928 032546                ESCAPE TST          ;IF ERROR, BR TO TEST END.
(3) 032546 104410          TRAP C$ESCAPE
(3) 032550 000124          .WORD L10063-.
5929 032552                CNTRIN                ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1)                                     :***** MACRO EXPANSION *****
(1) 032552 004737 011516  JSR PC, $CNTIN      ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 032556 000000          .WORD 0           ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1)                                     :*****
5930                                     :*****
5931 032560                ESCAPE TST          ;IF ERROR, BR TO TEST END.
(3) 032560 104410          TRAP C$ESCAPE
(3) 032562 000112          .WORD L10063-.
5932                                     :*****
5933                                     :*****
5934 032564                DMRIN WMODEM,0,BIT4 ;BLIND THE RECEIVER BY GOING INTO HDX.
(1)                                     ;USE WRITE MODEM COMMAND TO SET HALF DUPLEX.
(1) 032564 004737 012064  JSR PC, $DMRIN      ;***** MACRO EXPANSION *****
(1) 032570 000005          .WORD WMODEM      ;CALL DMR MODE INPUT ROUTINE
(1) 032572 000000          .WORD 0           ;INPUT COMMAND
(1) 032574 000020          .WORD BIT4        ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
(1)                                     ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
(1)                                     :*****
5935                                     :*****
5936 032576                BACCIT                ;BA/CC IN XMIT BUFFER
(1)                                     :***** MACRO EXPANSION *****
(1) 032576 004737 012274  JSR PC, $BACC        ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 032602 000040          .WORD RQI!BACCT   ;BA/CC IN TRANSMIT COMMAND
(1) 032604 002512          .WORD TBUF        ;TRANSMIT BUFFER ADDRESS
(1) 032606 000044          .WORD TCOUNT     ;TRANSMIT CHARACTER COUNT
(1)                                     :*****
5937                                     :*****
5938 032610                ESCAPE TST          ;IF ERROR, EXIT
(3) 032610 104410          TRAP C$ESCAPE
(3) 032612 000062          .WORD L10063-.
5939 032614                WAIT RDO                ;WAIT FOR THE READY OUT.
(1)                                     :***** MACRO EXPANSION *****
(1) 032614 004737 010266  JSR PC, $WAIT        ;CALL WAIT ROUTINE
(1) 032620 000001          .WORD 1           ;FLAG THAT WE'RE WAITING FOR RDO
(1)                                     :*****
5940 032622                ESCAPE TST          ;IF ERROR, EXIT.
(3) 032622 104410          TRAP C$ESCAPE
    
```

```

(3) 032624 000050
5941 032626 032777 000001 147400 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT .WORD L10063-.
5942 032634 001005 BNE 10$ ;IF YES, PROCEED.
5943 032636 ERRDF 8,EMG8,ERRG2 ;EXPECTED A CONTROL OUT.
(4) 032636 104455 TRAP C$ERDF
(5) 032640 000010 .WORD 8
(5) 032642 020072 .WORD EMG8
(5) 032644 015070 .WORD ERRG2
5944 032646 000410 BR 20$ ;EXIT
5945 032650 10$:
5946 032650 032777 000002 147362 BIT #TOUT,@SEL6 ;WAS THE TIME OUT REPORTED?
5947 032656 001004 BNE 20$ ;IF YES, EXIT
5948 032660 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED ERROR.
(4) 032660 104455 TRAP C$ERDF
(5) 032662 000011 .WORD 9
(5) 032664 020136 .WORD EMG9
(5) 032666 015070 .WORD ERRG2
5949 032670 20$:
5950 032670 SHUTDN
(1) (1) 032670 004737 012560 JSR PC, $HALT ;**** MACRO EXPANSION ****
;DMR HALT ROUTINE.
;****
5951
5952 032674 ENDTST
(3) 032674 L10063:
(3) 032674 104401 TRAP C$ETST
5953
5954
5955
5956
5957 .SBTTL TEST 11 - MESSAGE TOO LONG ERROR
5958
5959 ;*****
5960 ;* TEST 11 - DMR-11
5961 ;* MESSAGE TOO LONG - TRANSMIT A MESSAGE THAT IS TOO LONG FOR THE
5962 ;* RECEIVE BUFFER AND VERIFY THAT THE 'TOO LONG' ERROR IS RECEIVED.
5963 ;*
5964 ;*****
5965 032676 BGNTST
(3) 032676 T11::
5966 032676 CLEAR ;MACRO FOR MASTER CLEAR
(1) (1) 032676 004737 011064 JSR PC, $MSCLR ;**** MACRO EXPANSION ****
;ISSUE A DMR MASTER CLEAR
;****
5967
5968 032702 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 032702 104410 TRAP C$ESCAPE
(3) 032704 000150 .WORD L10064-.
5969 032706 BASEIN ;MACRO FOR BASE IN COMMAND
(1) (1) 032706 004737 011262 JSR PC, $BASEI ;**** MACRO EXPANSION ****
;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 032712 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 032714 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 032716 000522 .WORD DMR ;DMR-11 MODE
;****
5970

```

```

5971 032720          ESCAPE TST          ;IF ERROR, BR TO TEST END.
(3) 032720 104410
(3) 032722 000132          TRAP C$ESCAPE
5972 032724          CNTRIN          ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1)          ;**** MACRO EXPANSION ****
(1) 032724 004737 011516 JSR PC, $CNTIN      ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 032730 000000          .WORD 0          ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1)          ;****          ****
5973
5974 032732          ESCAPE TST          ;IF ERROR, BR TO TEST END.
(3) 032732 104410          TRAP C$ESCAPE
(3) 032734 000120          .WORD L10064-.
5975 032736          BACCIR RBUF,RCOUNT/2 ;SET UP THE RECEIVE BUFFER WITH 1/2 BUF. SPACE
(1)          ;**** MACRO EXPANSION ****
(1) 032736 004737 012274 JSR PC, $BACC      ;CALL BA/CC IN ROUTINE
(1) 032742 000044          .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
(1) 032744 002562          .WORD RBUF      ;BUFFER ADDRESS BITS 0-15
(1) 032746 000022          .WORD RCOUNT/2 ;BA BITS 16/17 AND CHAR. COUNT
(1)          ;****          ****
5976
5977 032750 012737 000001 002360 MOV #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT ERROR
5978          ;REPORTING BECAUSE WE ARE INTENTIONALLY
5979          ;CAUSING ONE IN THIS TEST.
5980 032756          BACCIT          ;BA/CC IN XMIT COMMAND
(1)          ;**** MACRO EXPANSION ****
(1) 032756 004737 012274 JSR PC, $BACC      ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 032762 000040          .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
(1) 032764 002512          .WORD TBUF      ;TRANSMIT BUFFER ADDRESS
(1) 032766 000044          .WORD TCOUNT    ;TRANSMIT CHARACTER COUNT
(1)          ;****          ****
5981 032770          10$:
5982 032770          WAIT RDO          ;WAIT FOR RDO TO BE SET
(1)          ;**** MACRO EXPANSION ****
(1) 032770 004737 010266 JSR PC, $WAIT      ;CALL WAIT ROUTINE
(1) 032774 000001          .WORD 1          ;FLAG THAT WE'RE WAITING FOR RDO
(1)          ;****          ****
5983 032776          ESCAPE TST          ;IF RDO NOT SET, BR TO TEST END.
(3) 032776 104410          TRAP C$ESCAPE
(3) 033000 000054          .WORD L10064-.
5984 033002 032777 000001 147224 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
5985 033010 001005          BNE 20$          ;IF YES, PROCEED
5986 033012          ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT.
(4) 033012 104455          TRAP C$ERDF
(5) 033014 000010          .WORD 8
(5) 033016 020072          .WORD EMG8
(5) 033020 015070          .WORD ERRG2
5987 033022 000410          BR 40$          ;EXIT
5988 033024          20$:
5989 033024 032777 000020 147206 BIT #TOLONG,@SEL6 ;IS THE TOO LONG BIT SET?
5990 033032 001004          BNE 40$          ;IF YES, TEST OK - FINISH UP.
5991 033034          30$:
5992 033034          ERRDF 9,EMG9,ERRG2 ;WE'RE NOT GETTING EXPECTED RESULT
(4) 033034 104455          TRAP C$ERDF
(5) 033036 000011          .WORD 9
(5) 033040 020136          .WORD EMG9
(5) 033042 015070          .WORD ERRG2

```

```

5993
5994 033044
5995 033044 005037 002360
5996 033050
(1)
(1) 033050 004737 012560
(1)
5997
5998
5999
6000 033054
(3) 033054
(3) 033054 104401
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020 033056
(3) 033056
6021 033056
(3) 033056
(3) 033056 104402
6022
6023 033060
(1)
(1) 033060 004737 011064
(1)
6024
6025 033064
(1)
(1) 033064 004737 011262
(1) 033070 004000
(1) 033072 002630
(1) 033074 000522
(1)
6026
6027 033076 012737 000001 002360
6028
6029
6030 033104
(1)
    
```

```

40$: CLR ERROR ;RESTORE ERROR FLAG TO NORMAL STATE.
      SHUTDN ;HALT THE DMR.
      ;**** MACRO EXPANSION ****
      JSR PC, $HALT ;DMR HALT ROUTINE.
      ;****
    
```

```

ENDTST
L10064: TRAP C$ETST
    
```

```

.SBTTL TEST 12 - PROCEDURE ERRORS
:*****
:* TEST 12 - DMR-11
:* PROCEDURE ERRORS -
:* THE FOLLOWING SHOULD CAUSE THE DMR-11 TO HALT AND RESPOND WITH
:* A PROCEDURE ERROR:
:* SUBTEST 1 - A SECOND BASE IN COMMAND
:* SUBTEST 2 - A CONTROL IN BEFORE A BASE IN
:* SUBTEST 3 - A BA/CC IN BEFORE A BASE IN
:* SUBTEST 4 - A BA/CC IN RCV WITH A BUFFER LENGTH OF 0
:* SUBTEST 5 - A BA/CC IN XMIT. WITH A BUFFER LENGTH OF 0
:*
:*****
BGNTST
BGNSUB
T12::
T12.1: TRAP C$BSUB
    
```

```

CLEAR ;MASTER CLEAR MACRO
      ;**** MACRO EXPANSION ****
      JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
      ;****
BASEIN
      ;**** MACRO EXPANSION ****
      JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
      .WORD LPLU ;SET LINE UNIT LOOP
      .WORD BASE ;BASE TABLE ADDRESS
      .WORD DMR ;DMR-11 MODE
      ;****
MOV #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT ERROR
      ;REPORTING BECAUSE WE ARE INTENTIONALLY
      ;CAUSING ONE IN THIS TEST.
BASEIN ;SECOND BASE IN
      ;**** MACRO EXPANSION ****
    
```

```

(1) 033104 004737 011262      JSR    PC, $BASEI      ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 033110 004000              .WORD  LPLU           ;SET LINE UNIT LOOP
(1) 033112 002630              .WORD  BASE           ;BASE TABLE ADDRESS
(1) 033114 000522              .WORD  DMR            ;DMR-11 MODE
(1)                               ;*****
6031                               ;*****
6032 033116              WAIT   RDO              ;WAIT FOR RDO TO BE SET
(1)                               ;***** MACRO EXPANSION *****
(1) 033116 004737 010266      JSR    PC, $WAIT       ;CALL WAIT ROUTINE
(1) 033122 000001              .WORD  1              ;FLAG THAT WE'RE WAITING FOR RDO
(1)                               ;*****
6033 033124              ESCAPE TST              ;IF RDO NOT SET, BR TO TEST END.
(3) 033124 104410              TRAP  C$ESCAPE
(3) 033126 000632              .WORD  L10065-
6034 033130 032777 000001 147076  BIT    #CNTRL,@SEL2    ;IS THIS A CONTROL OUT?
6035 033136 001005              BNE    10$             ;IF YES, PROCEED.
6036 033140              ERRDF  8,EMG8,ERRG2    ;EXPECTED CONTROL OUT
(4) 033140 104455              TRAP  C$ERDF
(5) 033142 000010              .WORD  8
(5) 033144 020072              .WORD  EMG8
(5) 033146 015070              .WORD  ERRG2
6037 033150 000410              BR     15$             ;EXIT
6038 033152              10$:
6039 033152 032777 001000 147060  BIT    #HALTC,@SEL6    ;IS THE HALT - PROCEDURE ERROR BIT SET?
6040 033160 001004              BNE    15$             ;IF YES - ERROR REPORTED CORRECTLY
6041 033162              ERRDF  9,EMG9,ERRG2    ;UNEXPECTED CONTROL OUT RECEIVED
(4) 033162 104455              TRAP  C$ERDF
(5) 033164 000011              .WORD  9
(5) 033166 020136              .WORD  EMG9
(5) 033170 015070              .WORD  ERRG2
6042 033172              15$:
6043 033172 042777 000207 147034  BIC    #RDO!CMD,@SEL2  ;CLEAR RDO AND THE COMMAND BITS
6044 033200 005037 002360      CLR    ERROR           ;RESTORE FLAG
6045 033204              ENDSUB
(3) 033204              L10066:
(3) 033204 104403              TRAP  C$ESUB
6046 033206              BGNSUB
(3) 033206              T12.2:
(3) 033206 104402              TRAP  C$BSUB
6048 033210              CLEAR
6049 033210              (1)
(1) 033210 004737 011064      JSR    PC, $MSCLR      ;MASTER CLEAR MACRO
(1)                               ;***** MACRO EXPANSION *****
(1)                               ;ISSUE A DMR MASTER CLEAR
(1)                               ;*****
6050 033214 012737 000001 002360  MOV    #CNTRL,ERROR    ;THIS FLAG WILL DISABLE ANY CONTROL OUT ERROR
6051 033214 012737 000001 002360  ;REPORTING BECAUSE WE ARE INTENTIONALLY
6052 033214 012737 000001 002360  ;CAUSING ONE IN THIS TEST.
6053 033222 005037 002260      CLR    DMRFLG          ;CLEAR FLAG THAT IS SET IN BASEIN IN ORDER
6054 033222 005037 002260  ;TO FLAG THAT A CONTROL OUT-DMR RUN MODE
6055 033222 005037 002260  ;COMMAND IS EXPECTED (THIS FLAG WAS SET IN
6056 033222 005037 002260  ;THE PREVIOUS SUBTEST BASEIN)
6057 033222 005037 002260  ;CONTROL IN
6058 033226              (1)
(1) 033226 004737 011516      JSR    PC, $CNTIN     ;***** MACRO EXPANSION *****
(1)                               ;CALL CONTROL IN ROUTINE WITH DEFAULT
    
```

```

(1) 033232 000000 .WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1) ;****
6059
6060 033234 WAIT RDO ;WAIT FOR RDO TO BE SET
(1) ;**** MACRO EXPANSION ****
(1) 033234 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 033240 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
(1) ;****
6061 033242 ESCAPE TST ;IF RDO NOT SET, BR TO TEST END.
(3) 033242 104410 TRAP C$ESCAPE
(3) 033244 000514 .WORD L10065-.
6062 033246 032777 000001 146760 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
6063 033254 001005 BNE 10$ ;IF YES - PROCEED.
6064 033256 ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
(4) 033256 104455 TRAP C$ERDF
(5) 033260 000010 .WORD 8
(5) 033262 020072 .WORD EMG8
(5) 033264 015070 .WORD ERRG2
6065 033266 000410 BR 15$ ;EXIT
6066 033270 10$:
6067 033270 032777 001000 146742 BIT #HALTC,@SEL6 ;IS THE HALT - PROCEDURE ERROR BIT SET?
6068 033276 001004 BNE 15$ ;IF YES - ERROR REPORTED CORRECTLY
6069 033300 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 033300 104455 TRAP C$ERDF
(5) 033302 000011 .WORD 9
(5) 033304 020136 .WORD EMG9
(5) 033306 015070 .WORD ERRG2
6070 033310 15$:
6071 033310 042777 000207 146716 BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
6072 033316 005037 002360 CLR ERROR ;RESTORE FLAG
6073 033322 ENDSUB
(3) 033322 L10067:
(3) 033322 104403 TRAP C$ESUB
6074
6075 033324 BGNSUB
(3) 033324 T12.3:
(3) 033324 104402 TRAP C$BSUB
6076
6077 033326 CLEAR ;MASTER CLEAR MACRO
(1) ;**** MACRO EXPANSION ****
(1) 033326 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
(1) ;****
6078
6079 033332 012737 000001 002360 MOV #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT ERROR
6080 ;REPORTING BECAUSE WE ARE INTENTIONALLY
6081 ;CAUSING ONE IN THIS TEST.
6082 033340 BACCIR ;BA/CC IN RCV. COMMAND
(1) ;**** MACRO EXPANSION ****
(1) 033340 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 033344 000044 .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
(1) 033346 002562 .WORD RBUF ;RECEIVE BUFFER
(1) 033350 000044 .WORD RCOUNT ;RECEIVE CHARACTER COUNT
(1) ;****
6083
6084 033352 WAIT RDO ;WAIT FOR RDO TO BE SET
(1) ;**** MACRO EXPANSION ****
  
```

```

(1) 033352 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 033356 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
(1) ESCAPE TST ;*****
6085 033360 ESCAPE TST ;IF RDO NOT SET, BR TO TEST END.
(3) 033360 104410 TRAP C$ESCAPE
(3) 033362 000376 .WORD L10065-.
6086 033364 032777 000001 146642 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
6087 033372 001005 BNE 10$ ;IF YES - PROCEED.
6088 033374 ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
(4) 033374 104455 TRAP C$ERDF
(5) 033376 000010 .WORD 8
(5) 033400 020072 .WORD EMG8
(5) 033402 015070 .WORD ERRG2
6089 033404 000410 BR 15$ ;EXIT
6090 033406 10$: BIT #HALTC,@SEL6 ;IS THE HALT - PROCEDURE ERROR BIT SET?
6091 033406 032777 001000 146624 BNE 15$ ;IF YES - ERROR REPORTED CORRECTLY
6092 033414 001004 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 033416 104455 TRAP C$ERDF
(5) 033420 000011 .WORD 9
(5) 033422 020136 .WORD EMG9
(5) 033424 015070 .WORD ERRG2
6094 033426 15$: BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
6095 033426 042777 000207 146600 CLR ERROR ;RESTORE FLAG
6096 033434 005037 002360 ENDSUB
(3) 033440 L10070: TRAP C$ESUB
(3) 033440 104403 BGNSUB
6098 033442 T12.4: TRAP C$BSUB
(3) 033442 104402 CLEAR ;MASTER CLEAR
6100 033444 JSR PC, $MSCLR ;***** MACRO EXPANSION *****
(1) ;ISSUE A DMR MASTER CLEAR
(1) ;*****
6101 ESCAPE TST ;IF ERROR, EXIT.
(3) 033450 TRAP C$ESCAPE
(3) 033452 104410 .WORD L10065-.
6103 033454 BASEIN ;BASE IN COMMAND
(1) ;***** MACRO EXPANSION *****
(1) 033454 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 033460 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 033462 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 033464 000522 .WORD DMR ;DMR-11 MODE
(1) ;*****
6104 ESCAPE TST ;IF ERROR, EXIT.
(3) 033466 TRAP C$ESCAPE
(3) 033470 104410 .WORD L10065-.
6106 033472 BACCIR ;ASSIGN A BA/CC IN RECEIVE BUFFER
(1) ;***** MACRO EXPANSION *****
(1) 033472 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 033476 000044 .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
(1) 033500 002562 .WORD RBUF ;RECEIVE BUFFER
  
```

```

(1) 033502 000044          .WORD RCOUNT ;RECEIVE CHARACTER COUNT
(1)                               :*****
6107                               :*****
6108 033504                ESCAPE TST          ;IF ERROR, EXIT.
(3) 033504 104410                TRAP C$ESCAPE
(3) 033506 000252                .WORD L10065-.
6109 033510 012737 000001 002360 MOV #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT
6110                               ;ERROR REPORTING BECAUSE WE ARE INTENTIONALLY
6111                               ;CAUSING ONE.
6112 033516                BACCIT TBUF,0 ;ASSIGN A BA/CC IN XMIT BUFFER LENGTH = 0.
(1)                               ;***** MACRO EXPANSION *****
(1) 033516 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE
(1) 033522 000040                .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
(1) 033524 002512                .WORD TBUF ;BUFFER ADDRESS BITS 0-15
(1) 033526 000000                .WORD 0 ;BA BITS 16 & 17 AND CHAR. COUNT
6113                               ;*****
6114 033530                WAIT RDO ;WAIT FOR RDO TO BE SET
(1)                               ;***** MACRO EXPANSION *****
(1) 033530 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 033534 000001                .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
6115                               ;*****
6115 033536                ESCAPE TST          ;IF RDO NOT SET, BR TO TEST END.
(3) 033536 104410                TRAP C$ESCAPE
(3) 033540 000220                .WORD L10065-.
6116 033542 032777 000001 146464 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
6117 033550 001005                BNE 10$ ;IF YES - PROCEED.
6118 033552                ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
(4) 033552 104455                TRAP C$ERDF
(5) 033554 000010                .WORD 8
(5) 033556 020072                .WORD EMG8
(5) 033560 015070                .WORD ERRG2
6119 033562 000410                BR 15$ ;EXIT
6120 033564                10$:
6121 033564 032777 001000 146446 BIT #HALTC,@SEL6 ;IS THE HALT - PROCEDURE ERROR BIT SET?
6122 033572 001004                BNE 15$ ;IF YES - ERROR REPORTED CORRECTLY
6123 033574                ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 033574 104455                TRAP C$ERDF
(5) 033576 000011                .WORD 9
(5) 033600 020136                .WORD EMG9
(5) 033602 015070                .WORD ERRG2
6124 033604                15$:
6125 033604 042777 000207 146422 BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
6126 033612 005037 002360 CLR ERROR ;RESTORE FLAG
6127 033616                ENDSUB
(3) 033616                L10071:
(3) 033616 104403                TRAP C$ESUB
6128                               ;
6129 033620                BGNSUB
(3) 033620                T12.5:
(3) 033620 104402                TRAP C$BSUB
6130 033622                CLEAR ;MASTER CLEAR
(1)                               ;***** MACRO EXPANSION *****
(1) 033622 004737 011064 JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
(1)                               ;*****
6131
    
```



```

6132 033626          ESCAPE TST          ;IF ERROR, EXIT.
(3) 033626 104410
(3) 033630 000130          TRAP C$ESCAPE
6133 033632          BASEIN          ;BASE IN COMMAND
(1) (1) 033632 004737 011262 JSR PC, $BASEI ;**** MACRO EXPANSION ****
(1) 033636 004000          ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 033640 002630          ;SET LINE UNIT LOOP
(1) 033642 000522          ;BASE TABLE ADDRESS
(1) (1) (1) (1) (1) (1) ;DMR-11 MODE
(1) (1) (1) (1) (1) (1) ;****

6134
6135 033644          ESCAPE TST          ;IF ERROR, EXIT.
(3) 033644 104410          TRAP C$ESCAPE
(3) 033646 000112          ;WORD L10065-.
6136 033650 012737 000001 002360 MOV #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT
6137 (1) (1) (1) (1) (1) (1) ;ERROR REPORTING BECAUSE WE ARE INTENTIONALLY
6138 (1) (1) (1) (1) (1) (1) ;CAUSING ONE.
6139 033656          BACCIR RBUF,0      ;ASSIGN A BA/CC IN REC. BUFFER LENGTH = 0
(1) (1) 033656 004737 012274 JSR PC, $BACC ;**** MACRO EXPANSION ****
(1) 033662 000044          ;CALL BA/CC IN ROUTINE
(1) 033664 002562          ;BA/CC IN RECEIVE COMMAND
(1) 033666 000000          ;BUFFER ADDRESS BITS 0-15
(1) (1) (1) (1) (1) (1) ;BA BITS 16/17 AND CHAR. COUNT
(1) (1) (1) (1) (1) (1) ;****

6140
6141 033670          WAIT RDO          ;WAIT FOR RDO TO BE SET
(1) (1) 033670 004737 010266 JSR PC, $WAIT ;**** MACRO EXPANSION ****
(1) 033674 000001          ;CALL WAIT ROUTINE
(1) (1) (1) (1) (1) (1) ;FLAG THAT WE'RE WAITING FOR RDO
(1) (1) (1) (1) (1) (1) ;****

6142 033676          ESCAPE TST          ;IF RDO NOT SET, BR TO TEST END.
(3) 033676 104410          TRAP C$ESCAPE
(3) 033700 000060          ;WORD L10065-.
6143 033702 032777 000001 146324 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
6144 033710 001005          BNE 10$          ;IF YES - PROCEED.
6145 033712          ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
(4) 033712 104455          TRAP C$ERDF
(5) 033714 000010          ;WORD 8
(5) 033716 020072          ;WORD EMG8
(5) 033720 015070          ;WORD ERRG2
6146 033722 000410          BR 15$          ;EXIT
6147 033724          10$:
6148 033724 032777 001000 146306 BIT #HALTC,@SEL6 ;IS THE HALT - PROCEDURE ERROR BIT SET?
6149 033732 001004          BNE 15$          ;IF YES - ERROR REPORTED CORRECTLY
6150 033734          ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 033734 104455          TRAP C$ERDF
(5) 033736 000011          ;WORD 9
(5) 033740 020136          ;WORD EMG9
(5) 033742 015070          ;WORD ERRG2
6151 033744          15$:
6152 033744 042777 000207 146262 BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
6153 033752 005037 002360 CLR ERROR ;RESTORE FLAG
6154 033756          ENDSUB
(3) 033756          L10072:
(3) 033756 104403          TRAP C$ESUB
6155
    
```

6156 033760 ENDTST  
 (3) 033760  
 (3) 033760 104401 L10065: TRAP C\$ETST

6157  
 6158  
 6159  
 6160  
 6161  
 6162  
 6163  
 6164  
 6165  
 6166  
 6167  
 6168  
 6169  
 6170  
 6171

.SBTTL TEST 13 - DATA TEST

```

:*****
:*          TEST 13 - DMR-11
:* FREE RUNNING FLAG MODE DATA TEST
:* TRANSMIT A MESSAGE AND VERIFY THE RECEIVED DATA IS CORRECT.
:* IN THIS TEST NO INTERRUPTS ARE USED AND THE LINE UNIT IS IN
:* INTERNAL (TTL) LOOPBACK. THIS TEST IS THE FIRST TEST IN WHICH
:* THE DMR IS USED IN A DATA TRANSMISSION MODE.
:*****
    
```

```

6172 033762 BGNTST
(3) 033762
6173 033762 013700 000044          MOV    RCOUNT,R0          ;BYTE COUNT FOR RECEIVE BUFFER
6174 033766 062700 000002          ADD    #2,R0              ;2 ADDITIONAL BYTES AT END OF BUFFER ARE
6175                                     ;USED FOR DELIMITOR
6176 033772 012701 002562          MOV    #RBUF,R1          ;ADDRESS OF RECEIVE BUFFER
6177 033776          .10$: CLRB  (R1)+              ;CLEAR A BYTE IN THE BUFFER
6178 033776 105021          DEC    R0                ;CONTINUE - UNTIL ENTIRE BUFFER DONE
6179 034000 005300          BNE   10$
6180 034002 001375
6181
6182 034004 005037 002510          CLR    TFLAG             ;CLEAR TRANSMIT FLAG
6183 034010 005037 002560          CLR    RFLAG            ;CLEAR RECEIVER FLAG
6184 034014          CLEAR                ;MACRO FOR MASTER CLEAR
(1)                                     ;**** MACRO EXPANSION ****
(1) 034014 004737 011064          JSR    PC,$MSCLR         ;ISSUE A DMR MASTER CLEAR
(1)                                     ;****
6185
6186 034020          ESCAPE TST                ;IF ERROR, BR TO TEST END.
(3) 034020 104410
(3) 034022 000432          TRAP  .WORD C$ESCAPE
6187 034024          BASEIN                ;MACRO FOR BASE IN COMMAND
(1)                                     ;**** MACRO EXPANSION ****
(1) 034024 004737 011262          JSR    PC,$BASEI        ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 034030 004000          .WORD LPLU              ;SET LINE UNIT LOOP
(1) 034032 002630          .WORD BASE              ;BASE TABLE ADDRESS
(1) 034034 000522          .WORD DMR               ;DMR-11 MODE
(1)                                     ;****
6188
6189 034036          ESCAPE TST                ;IF ERROR, BR TO TEST END.
(3) 034036 104410          TRAP  .WORD C$ESCAPE
(3) 034040 000414          .WORD L10073-
6190 034042          CNTRIN                ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1)                                     ;**** MACRO EXPANSION ****
(1) 034042 004737 011516          JSR    PC,$CNTIN        ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 034046 000000          .WORD 0                 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
(1)                                     ;****
6191 034050          ESCAPE TST                ;IF ERROR, BR TO TEST END.
    
```

```

(3) 034050 104410
(3) 034052 000402 TRAP C$ESCAPE
6192 .WORD L10073-.
6193 034054 BACCIR ;BUFFER ADDRESS/CHARACTER COUNT REC. IN
(1) ;**** MACRO EXPANSION ****
(1) 034054 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 034060 000044 .WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
(1) 034062 002562 .WORD RBUF ;RECEIVE BUFFER
(1) 034064 000044 .WORD RCOUNT ;RECEIVE CHARACTER COUNT
(1) ;****
6194 ESCAPE TST ;IF ERROR (I.E. RDI NOT SET), ESCAPE
6195 034066 104410 TRAP C$ESCAPE
(3) 034066 000364 .WORD L10073-.
6196 034072 BACCIT ;BUFFER ADDRESS/CHARACTER COUNT XMIT. IN
(1) ;**** MACRO EXPANSION ****
(1) 034072 004737 012274 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
(1) 034076 000040 .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
(1) 034100 002512 .WORD TBUF ;TRANSMIT BUFFER ADDRESS
(1) 034102 000044 .WORD TCOUNT ;TRANSMIT CHARACTER COUNT
(1) ;****
6198 ESCAPE TST ;IF ERROR (I.E. RDI NOT SET), ESCAPE
6199 034104 104410 TRAP C$ESCAPE
(3) 034104 000346 .WORD L10073-.
6200 20$:
6201 034110 WAIT RDO ;WAIT FOR RDO
6202 034110 (1) ;**** MACRO EXPANSION ****
(1) 034110 004737 010266 JSR PC, $WAIT ;CALL WAIT ROUTINE
(1) 034114 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
(1) ;****
6203 034116 BERROR 52$ ;IF ERROR - RDO NOT SET, END TEST
(2) 034116 103552 BCS 52$
6204 034120 032777 000001 146106 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT COMMAND?
6205 034126 001405 BEQ 25$ ;IF NOT - PROCEED
6206 034130 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
(4) 034130 104455 TRAP C$ERRDF
(5) 034132 000011 .WORD 9
(5) 034134 020136 .WORD EMG9
(5) 034136 015070 .WORD ERRG2
6207 034140 000541 BR 52$
6208 034142 25$:
6209 034142 032777 000004 146064 BIT #RCV,@SEL2 ;TRANSMIT OR RECEIVE?
6210 034150 001035 BNE 40$ ;BR FOR RECEIVE
6211 ;CHECK TRANSMIT
6212
6213
6214 034152 005737 002510 TST TFLAG ;IS THIS THE FIRST TRANSMIT DONE?
6215 034156 001405 BEQ 30$ ;YES - OK
6216 034160 ERRDF 10,EMG10,ERRG2 ;ERROR MULTIPLE TRANSMITS
(4) 034160 104455 TRAP C$ERRDF
(5) 034162 000012 .WORD 10
(5) 034164 020165 .WORD EMG10
(5) 034166 015070 .WORD ERRG2

```

```

6217 034170 000525 BR 52$
6218 034172
6219 034172 012737 177777 002510 30$: MOV #-1,TFLAG ;FLAG THAT TRANSMIT CHECK IS DONE.
6220 034200 022777 002512 146030 CMP #TBUF,@SEL4 ;TRANSMIT BUFFER ADDRESS CORRECT?
6221 034206 001405 BEQ 32$ ;YES - PROCEED
6222 034210 ERRDF 11,EMG11,ERRG2 ;BUFFER ADDRESS ERROR
(4) 034210 104455 TRAP C$ERDF
(5) 034212 000013 .WORD 11
(5) 034214 020214 .WORD EMG11
(5) 034216 015070 .WORD ERRG2
6223 034220 000511 BR 52$
6224 034222
6225 034222 022777 000044 146010 32$: CMP #TCOUNT,@SEL6 ;COUNT CORRECT ?
6226 034230 001470 BEQ 50$ ;YES - PROCEED
6227 034232 ERRDF 12,EMG12,ERRG2 ;CHARACTER COUNT ERROR
(4) 034232 104455 TRAP C$ERDF
(5) 034234 000014 .WORD 12
(5) 034236 020241 .WORD EMG12
(5) 034240 015070 .WORD ERRG2
6228 034242 000500 BR 52$
6229
6230 ;CHECK RECEIVE
6231
6232 034244
6233 034244 005737 002560 40$: TST RFLAG ;IS THIS THE FIRST RECEIVE DONE ?
6234 034250 001405 BEQ 41$ ;YES - PROCEED
6235 034252 ERRDF 13,EMG13,ERRG2 ;MULTIPLE RECEIVES
(4) 034252 104455 TRAP C$ERDF
(5) 034254 000015 .WORD 13
(5) 034256 020267 .WORD EMG13
(5) 034260 015070 .WORD ERRG2
6236 034262 000470 BR 52$
6237 034264
6238 034264 012737 177777 002560 41$: MOV #-1,RFLAG ;FLAG THAT RECEIVE CHECK HAS BEEN DONE.
6239 034272 022777 002562 145736 CMP #RBUF,@SEL4 ;IS THE RECEIVE BUFFER ADDRESS CORRECT?
6240 034300 001405 BEQ 43$ ;YES - PROCEED
6241 034302 ERRDF 11,EMG11,ERRG2 ;BUFFER ADDRESS ERROR
(4) 034302 104455 TRAP C$ERDF
(5) 034304 000013 .WORD 11
(5) 034306 020214 .WORD EMG11
(5) 034310 015070 .WORD ERRG2
6242 034312 000454 BR 52$
6243 034314
6244 034314 022777 000044 145716 43$: CMP #RCOUNT,@SEL6 ;IS THE BUFFER COUNT CORRECT?
6245 034322 001405 BEQ 44$ ;YES - PROCEED
6246 034324 ERRDF 12,EMG12,ERRG2 ;CHARACTER COUNT ERROR
(4) 034324 104455 TRAP C$ERDF
(5) 034326 000014 .WORD 12
(5) 034330 020241 .WORD EMG12
(5) 034332 015070 .WORD ERRG2
6247
6248 034334 000443 BR 52$
6249 034336
6250 034336 012700 000044 44$: MOV #RCOUNT,R0 ;SET UP FOR DATA CHECK (CHARCATER COUNT)
6251 034342 012701 002512 MOV #TBUF,R1 ;GOOD DATA POINTER
6252 034346 012702 002562 MOV #RBUF,R2 ;RECEIVE DATA POINTER
    
```

```

6253 034352
6254 034352 122122
6255 034354 001011
6256 034356 005300
6257 034360 001374
6258 034362 005712
6259
6260 034364 001412
6261 034366
(4) 034366 104455
(5) 034370 000016
(5) 034372 020315
(5) 034374 015070
6262 034376 000422
6263 034400
6264 034400
(4) 034400 104455
(5) 034402 000017
(5) 034404 020341
(5) 034406 015070
6265 034410 000415
6266
6267
6268
6269 034412
6270 034412 042777 000213 145614
6271 034420 005737 002560
6272 034424 001002
6273 034426 000137 034110
6274 034432
6275 034432 005737 002510
6276 034436 001002
6277 034440 000137 034110
6278 034444
6279 034444
(1)
(1) 034444 004737 012560
(1)
6280
6281 034450
6282
6283 034454
(3) 034454
(3) 034454 104401
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
    
```

```

45$:
    CMPB (R1)+,(R2)+ ;IS THE DATA THE SAME ?
    BNE 46$ ;IF NOT, BRANCH TO DATA ERROR MESSAGE
    DEC R0 ;CONTINUE CHECKING UNTIL DONE WITH BUFFER.
    BNE 45$
    TST @R2 ;THIS SHOULD BE 0 - REMEMBER WE CLEARED
    ;2 EXTRA BYTES DURING BUFFER INIT.
    BEQ 50$ ;IF OK - PROCEED
    ERRDF 14,EMG14,ERRG2 ;RECEIVED EXTRA DATA
    TRAP C$ERDF
    .WORD 14
    .WORD EMG14
    .WORD ERRG2

46$:
    BR 52$
    ERRDF 15,EMG15,ERRG2 ;DATA ERROR
    TRAP C$ERDF
    .WORD 15
    .WORD EMG15
    .WORD ERRG2

50$:
    BIC #RDO+RCV+CMD,@SEL2 ;CLEAR RDO, RCV & COMMAND BITS (0,1)
    TST RFLAG ;IS THE RECEIVE DONE ? (IF DONE, FLAG = -1)
    BNE 51$ ;YES - SEE IF TRANSMIT DONE
    JMP 20$ ;NO - GO BACK AND DO IT.

51$:
    TST TFLAG ;IS THE TRANSMIT DONE ?
    BNE 52$ ;YES - BR TO SHUTDOWN
    JMP 20$ ;NO - DO IT

52$:
    SHUTDN ;SHUTDOWN DMR
    JSR PC,$HALT ;**** MACRO EXPANSION ****
    ;DMR HALT ROUTINE.
    ;****

CALL $ERROR ;CHECK BASE TABLE AND REPORT ANY SOFT ERRORS

ENDTST
L10073:
    TRAP C$ETST
    
```

.SBTTL TEST 14 - EXTENDED ADDRESSING DATA TEST

```

:*****
:*
:* TEST 14 - DMR-11
:* IN THIS TEST - SEE IF WE HAVE MEMORY MANAGEMENT, IF SO SEE IF WE
:* HAVE THE MEMORY TO CHECK BITS 16 & 17 IN SEL6. THIS WILL ALLOW
    
```

6296  
6297  
6298  
6299  
6300  
6301 034456  
(3) 034456  
6302  
6303  
6304 034456  
(7) 034456 012746 000340  
(6) 034462 012746 023512  
(5) 034466 012746 000004  
(4) 034472 012746 000003  
(3) 034476 104437  
(2) 034500 062706 000010  
6305 034504 005037 002340  
6306 034510 005737 177572  
6307 034514  
(3) 034514 012700 000004  
(3) 034520 104436  
6308 034522 005737 002340  
6309  
6310  
6311 034526 001404  
6312 034530 005037 002340  
6313 034534 000137 036044  
6314 034540  
6315  
6316  
6317  
6318 034540 023727 002120 002200  
6319 034546 002002  
6320 034550 000137 036044  
6321 034554  
6322 034554  
(3) 034554 012700 000340  
(3) 034560 104441  
6323  
6324  
6325 034562 012701 172300  
6326 034566 012700 000010  
6327 034572  
6328 034572 012721 077406  
6329  
6330 034576 005300  
6331 034600 001374  
6332 034602 012701 172340  
6333 034606 005011  
6334 034610 012761 000200 000002  
6335 034616 012761 000400 000004  
6336 034624 012761 000600 000006  
6337 034632 012761 001000 000010  
6338 034640 012761 002000 000012  
6339 034646 012761 004000 000014  
6340 034654 012761 007600 000016

:\* US TO TRANSFER DATA USING THOSE EXTENDED ADDRESSING BITS. AS IN  
:\* TEST 13 THE TEST IS NON-INTERRUPT AND INTERNAL (TTL) LOOPBACK IS  
:\* USED.  
:\*  
:\*\*\*\*\*  
:BGNTST

T14::  
.ENABL LSB ;ENABLE LOCAL BLOCK - NEEDED BECAUSE OF  
;USE OF SYMBOLIC LABELS 'RSEL4' ETC.  
SETVEC #4,#NOXMEM,#PRI07 ;SET UP TRAP VECTOR 4  
MOV #PRI07,-(SP)  
MOV #NOXMEM,-(SP)  
MOV #4,-(SP)  
MOV #3,-(SP)  
TRAP C\$SVEC  
ADD #10,SP  
CLR FLAG ;CLEAR FLAG - SET IF TRAP TO 4.  
TST @#177572 ;ADDRESS MEMORY MANAGEMENT REGISTER.  
CLRVEC #4 ;RESTORE TRAP VECTOR 4.  
MOV #4,RO  
TRAP C\$CVEC  
TST FLAG ;IS THE FLAG STILL CLEARED?  
;NOTE: THE FLAG WILL BE SET BY TRAP 4  
;IF THERE IS NO MEMORY MANAGEMENT.  
;IF FLAG IS CLEARED, PROCEED WITH TEST.  
BEQ 10\$ ;RESTORE FLAG  
CLR FLAG  
JMP 85\$ ;EXIT - CAN'T TEST WITHOUT MEM. MANAG.  
10\$:  
;NOTE: L\$HIMEM IS SIZE OF TOTAL MEMORY IN  
;PAGE ADDRESS REGISTER FORM - DETERMINED BY  
;BY DIAGNOSTIC SUPERVISOR AT STARTUP.  
CMP L\$HIMEM,#2200 ;DO WE HAVE ENOUGH MEMORY TO ADDRESS BIT 16?  
BGE 15\$ ;IF YES - PROCEED WITH TEST  
JMP 85\$ ;IF NOT - EXIT  
15\$:  
SETPRI #PRI07 ;MAKE SURE WE ARE IN KERNEL MODE.  
MOV #PRI07,RO  
TRAP C\$SPRI  
;SETTING PRI SHOULD ALSO CLEAR BITS 14 & 15  
;IN PSW WHICH PLACES PROCESSOR IN KERNEL MODE.  
MOV #172300,R1 ;GET ADDRESS OF KERNEL PDR REG 0  
MOV #8.,RO ;GOING TO WRITE PDR REG 0-7  
20\$:  
MOV #77406,(R1)+ ;WRITE BITS FOR THE FOLLOWING PAGE DESCRIPTION  
;READ/WRITE ACCESS, 128. BLOCK PAGE LENGTH.  
;WRITE ALL PDRS  
DEC RO  
BNE 20\$  
MOV #172340,R1 ;GET ADDRESS OF KERNAL PAR 0  
CLR (R1) ;PAR 0, ADDR 0 - 17776  
MOV #200,2(R1) ;PAR 1, ADDR 20000 - 37776  
MOV #400,4(R1) ;PAR 2, ADDR 40000 - 57776  
MOV #600,6(R1) ;PAR 3, ADDR 60000 - 77776  
MOV #1000,10(R1) ;PAR 4, ADDR 100000 - 117776  
MOV #2000,12(R1) ;PAR 5, ADDR 200000 - 217776  
MOV #4000,14(R1) ;PAR 6, ADDR 400000 - 417776  
MOV #7600,16(R1) ;PAR 7, ADDR 160000 - 177776 (I/O PAGE)

```

6341
6342 034662 012703 000100      MOV    #64,R3      ;COUNTER FOR OUTER LOOP OF TEST PATTERN GEN.
6343 034666 012704 120000      MOV    #120000,R4  ;USE VIRTUAL ADDRESS TO MAP TO PAR 5
6344                                     ;GENERATE A TEST PATTERN IN THE 1ST 4K
6345                                     ;BYTES OF PAR 5 (VIRTUAL ADDR 120000 - 127776)
6346 034672 005037 002340      CLR    FLAG        ;ENSURE FLAG IS CLEARED
6347 034676                                     SETVEC #4,#NOXMEM,#PRI07 ;SET UP TRAP VECTOR 4 (WILL SET FLAG)
(7) 034676 012746 000340                                     MOV    #PRI07,-(SP)
(6) 034702 012746 023512                                     MOV    #NOXMEM,-(SP)
(5) 034706 012746 000004                                     MOV    #4,-(SP)
(4) 034712 012746 000003                                     MOV    #3,-(SP)
(3) 034716 104437                                     TRAP   C$SVEC
(2) 034720 062706 000010                                     ADD    #10,SP
6348 034724 012737 000001 177572 MOV    #1,@#177572 ;ENABLE MEMORY MANAGEMENT
6349 034732                                     30$:
6350 034732 012701 000040      MOV    #32,R1      ;COUNTER FOR INNER LOOP OF TEST PATTERN GEN.
6351 034736 012702 002410      MOV    #32,R2      ;ADDRESS FOR 32. WORD TEST PATTERN
6352 034742                                     31$:
6353 034742 012224      MOV    (R2)+,(R4)+ ;WRITE TEST PATTERN INTO 4K BYTES
6354                                     ;(PHYSICAL ADDRESS 200000 - 207776)
6355 034744 005737 002340      TST    FLAG        ;NXM TRAP 4?
6356 034750 001014      BNE    33$        ;IF YES - EXIT
6357 034752 005301      DEC    R1          ;DO THE INNER LOOP 32. TIMES
6358 034754 001372      BNE    31$
6359 034756 005303      DEC    R3          ;DO THE OUTER LOOP 128. TIMES
6360 034760 001364      BNE    30$
6361 034762 012701 004000      MOV    #4000,R1   ;COUNTER TO CLEAR THE NEXT 4K BYTES.
6362 034766                                     32$:
6363 034766 005024      CLR    (R4)+      ;CLEAR OUT THE ENTIRE PAR
6364                                     ;(PHYSICAL ADDRESS 210000 - 217776)
6365 034770 005737 002340      TST    FLAG        ;NXM TRAP 4?
6366 034774 001002      BNE    33$        ;IF YES - EXIT
6367 034776 005301      DEC    R1
6368 035000 001372      BNE    32$
6369 035002                                     33$:
6370 035002 005037 177572      CLR    @#177572   ;TURN OFF MEMORY MANAGEMENT
6371 035006 012700 000004      CLRVEC #4         ;RESTORE TRAP 4 TO SUPERVISOR
(3) 035006 012700 000004                                     MOV    #4,R0
(3) 035012 104436                                     TRAP   C$CVEC
6372 035014 005737 002340      TST    FLAG        ;WAS THIS AN ERROR EXIT
6373 035020 001417      BEQ    34$        ;IF NOT, PROCEED.
6374 035022      ERRDF 19,EMT22
(4) 035022 104455                                     TRAP   C$ERDF
(5) 035024 000023      .WORD 19
(5) 035026 036046      .WORD EMT22
(5) 035030 000000      .WORD 0
6375 035032      PRINTB #FMT25,R4
(8) 035032 010446                                     MOV    R4,-(SP)
(7) 035034 012746 036104                                     MOV    #FMT25,-(SP)
(6) 035040 012746 000002                                     MOV    #2,-(SP)
(3) 035044 010600      MOV    SP,R0
(4) 035046 104414      TRAP   C$PNTB
(4) 035050 062706 000006      ADD    #6,SP
6376 035054 000137 036044      JMP    85$
6377 035060      34$:
6378 035060      CLEAR           ;MACRO FOR MASTER CLEAR
  
```

```

(1)
(1) 035060 004737 011064 JSR PC, $MSCLR ;**** MACRO EXPANSION ****
(1) ;ISSUE A DMR MASTER CLEAR
6379 ;****
6380 035064 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 035064 104410 TRAP C$ESCAPE
(3) 035066 000756 .WORD L10074-.
6381 035070 BASEIN ;MACRO FOR BASE IN COMMAND
(1) ;**** MACRO EXPANSION ****
(1) 035070 004737 011262 JSR PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 035074 004000 .WORD LPLU ;SET LINE UNIT LOOP
(1) 035076 002630 .WORD BASE ;BASE TABLE ADDRESS
(1) 035100 000522 .WORD DMR ;DMR-11 MODE
6382 ;****
6383 035102 ESCAPE TST ;IF ERROR, BR TO TEST END.
(3) 035102 104410 TRAP C$ESCAPE
(3) 035104 000740 .WORD L10074-.
6384 035106 CNTRIN ;MACRO FOR CONTROL IN (FULL DUPLEX)
(1) ;**** MACRO EXPANSION ****
(1) 035106 004737 011516 JSR PC, $CNTIN ;CALL CONTROL IN ROUTINE WITH DEFAULT
(1) 035112 000000 .WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
6385 035114 ESCAPE TST ;****
(3) 035114 104410 ;IF ERROR, BR TO TEST END.
(3) 035116 000726 TRAP C$ESCAPE
6386 .WORD L10074-.
6387 035120 005037 002510 CLR TFLAG ;CLEAR TRANSMIT FLAG
6388 035124 005037 002560 CLR RFLAG ;CLEAR RECEIVE FLAG
6389 035130 005037 002342 CLR SFLAG ;CLEAR SECOND LOOP FLAG
6390 ;IF SFLAG = 0, THEN THIS IS A TEST OF BIT 16
6391 ;IF SFLAG = -1, THEN THIS IS A TEST OF BIT 17
6392 035134 012737 010000 035170 MOV #10000,RSEL4 ;RECEIVE BUFFER ADDRESS (BITS 0-15)
6393 035142 012737 050000 035172 MOV #BIT14!10000,RSEL6 ;REC BUFFER ADDR BIT 16 SET AND 4K
6394 ;BYTE RECEIVE CHARACTER COUNT
6395 035150 005037 035206 CLR TSEL4 ;TRANSMIT BUFFER ADDRESS (BITS 0-15)
6396 035154 012737 050000 035210 MOV #BIT14!10000,TSEL6 ;XMIT BUFFER ADDR BIT 16 SET AND 4K
6397 ;BYTE XMIT CHARACTER COUNT
6398 035162 35$: CALL $BACC ;ISSUE THE BUFFER ADDR/ CHAR COUNT COMMAND
6399 035162 .WORD RQI!BACCR ;COMMAND FOR BA/CC IN RECEIVE
6400 035166 000044 .WORD 0 ;BUFFER ADDRESS BITS 0-15
6401 035170 000000 RSEL4: .WORD 0 ;BUFFER ADDR BIT 16 + CHAR. COUNT
6402 035172 000000 RSEL6: .WORD 0 ;IF ERROR, END TEST
6403 035174 ESCAPE TST
(3) 035174 104410 TRAP C$ESCAPE
(3) 035176 000646 .WORD L10074-.
6404
6405 035200 CALL $BACC ;ISSUE THE BUFFER ADDR/ CHAR COUNT COMMAND
6406 035204 000040 .WORD RQI!BACCT ;COMMAND FOR BA/CC IN TRANSMIT
6407 035206 000000 TSEL4: .WORD 0 ;BUFFER ADDRESS BITS 0-15
6408 035210 000000 TSEL6: .WORD 0 ;BUFFER ADDR BIT 16 + CHAR. COUNT
6409 035212 ESCAPE TST ;IF ERROR, END TEST
(3) 035212 104410 TRAP C$ESCAPE
(3) 035214 000630 .WORD L10074-.
6410 035216 40$: WAIT RDO ;WAIT FOR RDO TO BE SET
6411 035216
    
```



```
(1)
(1) 035216 004737 010266 JSR PC, $WAIT ;**** MACRO EXPANSION ****
(1) 035222 000001 .WORD 1 ;CALL WAIT ROUTINE
(1) ;FLAG THAT WE'RE WAITING FOR RDO
6412 035224 ESCAPE TST ;****
(3) 035224 104410 ;IF RDO NOT SET BEFORE TIMEOUT, END TEST
(3) 035226 000616 TRAP C$ESCAPE
6413 .WORD L10074-.
6414 035230 032777 000001 144776 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT COMMAND?
6415 035236 001406 BEQ 50$ ;NO - PROCEED
6416 035240 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT.
(4) 035240 104455 TRAP C$ERDF
(5) 035242 000011 .WORD 9
(5) 035244 020136 .WORD EMG9
(5) 035246 015070 .WORD ERRG2
6417 035250 000137 036034 JMP 80$ ;EXIT
6418 035254 50$: BIT #RCV,@SEL2 ;IS THIS A TRANSMIT OR RECEIVE?
6419 035254 032777 000004 144752 BNE 60$ ;BR FOR RECEIVE
6420 035262 001040 TST TFLAG ;IS THIS THE 1ST TRANSMIT DONE
6421 035264 005737 002510 BEQ 55$ ;IF YES, PROCEED
6422 035270 001406 ERRDF 10,EMG10,ERRG2 ;MULTIPLE TRANSMITS
(4) 035272 104455 TRAP C$ERDF
(5) 035274 000012 .WORD 10
(5) 035276 020165 .WORD EMG10
(5) 035300 015070 .WORD ERRG2
6424 035302 000137 036034 JMP 80$ ;EXIT
6425 035306 55$: MOV #-1,TFLAG ;FLAG THAT THE TRANSMIT IS DONE.
6426 035306 012737 177777 002510 CMP TSEL4,@SEL4 ;IS THE BUFFER ADDRESS CORRECT?
6427 035314 023777 035206 144714 BEQ 56$ ;IF OK, PROCEED WITH CHECK.
6428 035322 001406 ERRDF 11,EMG11,ERRG2 ;BUFFER ADDRESS ERROR
(4) 035324 104455 TRAP C$ERDF
(5) 035326 000013 .WORD 11
(5) 035330 020214 .WORD EMG11
(5) 035332 015070 .WORD ERRG2
6430 035334 000137 036034 JMP 80$ ;EXIT
6431 035340 56$: CMP TSEL6,@SEL6 ;IS THE CHAR. COUNT CORRECT?
6432 035340 023777 035210 144672 BEQ 70$ ;IF OK, PROCEED
6433 035346 001502 ERRDF 12,EMG12,ERRG2 ;CHARACTER COUNT ERROR - OR EXT MEM PROBLEM
(4) 035350 104455 TRAP C$ERDF
(5) 035352 000014 .WORD 12
(5) 035354 020241 .WORD EMG12
(5) 035356 015070 .WORD ERRG2
6435 035360 000137 036034 JMP 80$ ;EXIT
6436 035364 60$: TST RFLAG ;IS THIS THE 1ST RECEIVE DONE
6437 035364 005737 002560 BEQ 61$ ;IF YES, PROCEED
6438 035370 001406 ERRDF 13,EMG13,ERRG2 ;MULTIPLE RECEIVES
(4) 035372 104455 TRAP C$ERDF
(5) 035374 000015 .WORD 13
(5) 035376 020267 .WORD EMG13
(5) 035400 015070 .WORD ERRG2
6440 035402 000137 036034 JMP 80$ ;EXIT
6441 035406 61$:
```

```
6442 035406 012737 177777 002560 MOV # -1,RFLAG ;FLAG THAT THE RECEIVE IS DONE.
6443 035414 023777 035170 144614 CMP RSEL4,@SEL4 ;IS THE BUFFER ADDRESS CORRECT?
6444 035422 001405 BEQ 62$ ;IF OK, PROCEED WITH CHECK.
6445 035424 ERRDF 11,EMG11,ERRG2 ;BUFFER ADDRESS ERROR
(4) 035424 104455 TRAP C$ERDF
(5) 035426 000013 .WORD 11
(5) 035430 020214 .WORD EMG11
(5) 035432 015070 .WORD ERRG2
6446 035434 000577 BR 80$ ;EXIT
6447 035436 62$: CMP RSEL6,@SEL6 ;IS THE CHAR. COUNT CORRECT?
6448 035436 023777 035172 144574 BEQ 63$ ;IF OK, PROCEED
6449 035444 001404 ERRDF 12,EMG12,ERRG2 ;CHARACTER COUNT ERROR
(4) 035446 104455 TRAP C$ERDF
(5) 035450 000014 .WORD 12
(5) 035452 020241 .WORD EMG12
(5) 035454 015070 .WORD ERRG2
6451 035456 63$: TST SFLAG ;WHICH EXTENDED ADDRESS ARE WE CHECKING?
6452 035456 005737 002342 BNE 65$ ;BR FOR BIT 17 CHECK
6453 035462 001007 MOV #4000,R0 ;IN THE TEST OF BIT 16 WE SENT 4K BYTES
6454 035464 012700 004000 MOV #120000,R1 ;THE XMIT BUFFER BEGAN AT THIS VIRTUAL ADDR
6455 035470 012701 120000 ;WHICH WILL MAP TO 200000
6456 ;THE REC BUF. MAPS TO 210000
6457 035474 012702 130000 MOV #130000,R2 ;GO COMPARE THE XMIT AND REC BUFFERS
6458 035500 000406 BR 66$
6459 035502 65$: MOV #10000,R0 ;IN THE TEST OF BIT 17 WE SENT 8K BYTES
6460 035502 012700 010000 MOV #120000,R1 ;THE XMIT BUFFER MAPS TO 200000
6461 035506 012701 120000 MOV #140000,R2 ;THE REC BUF. MAPS TO 400000
6462 035512 012702 140000
6463 035516 66$: MOV #1,@#177572 ;TURN MEMORY MANAGEMENT BACK ON.
6464 035516 012737 000001 177572
6465 035524 67$: CMP (R1)+,(R2)+ ;IS THE DATA THE SAME?
6466 035524 022122 BNE 68$ ;IF NOT THERE IS A DATA ERROR.
6467 035526 001003 DEC R0 ;CHECK ENTIRE BUFFER
6468 035530 005300 BNE 67$
6469 035532 001374 BR 70$
6470 035534 000407
6471 035536 68$: CLR @#177572 ;TURN MEMORY MANAGEMENT OFF.
6472 035536 005037 177572 ERRDF 15,EMG15,ERRG2 ;DATA ERROR
(4) 035542 104455 TRAP C$ERDF
(5) 035544 000017 .WORD 15
(5) 035546 020341 .WORD EMG15
(5) 035550 015070 .WORD ERRG2
6474 035552 000530 BR 80$ ;EXIT
6475 035554 70$: CLR @#177572 ;TURN MEMORY MANAGEMENT OFF.
6476 035554 005037 177572 BIC #RDO+RCV+CMD,@SEL2 ;CLEAR RDO, RCV & COMMAND BITS (0,1)
6477 035560 042777 000213 144446 TST RFLAG ;IS THE RECEIVE DONE ? (IF DONE, FLAG = -1)
6478 035566 005737 002560 BNE 71$ ;YES - SEE IF TRANSMIT DONE
6479 035572 001002 JMP 40$ ;NO - GO BACK AND DO IT.
6480 035574 000137 035216
6481 035600 71$: TST TFLAG ;IS THE TRANSMIT DONE ?
6482 035600 005737 002510 BNE 72$ ;YES - SEE IF THERE IS MORE
6483 035604 001002 JMP 40$ ;NO - DO IT
6484 035606 000137 035216
6485 035612 72$:
```

```

6486 035612 005737 002342      TST      SFLAG      ;HAVE WE ALREADY TESTED BIT 17
6487 035616 001106              BNE      80$        ;IF SO - END OF TEST
6488
6489 035620 012737 177777 002342      MOV      #-1,SFLAG   ;FLAG SO WE DON'T COME THIS WAY AGAIN.
6490 035626 023727 002120 004200      CMP      L$HIMEM,#4200 ;IS THERE ENOUGH MEMORY TO TEST BIT 17?
6491 035634 002477              BLT      80$        ;IF NOT - END OF TEST.
6492 035636 005037 002510      CLR      TFLAG      ;CLEAR FLAGS FOR NEXT TEST
6493 035642 005037 002560      CLR      RFLAG
6494
6495      ;
6496      ;SET UP TO TEST BIT 17, IF THERE IS ENOUGH MEMORY.
6497      ;THIS TEST WILL TRANSMIT 8K BYTES STARTING AT PHYSICAL ADDRESS 200000
6498      ;TO PHYSICAL ADDRESS 400000. THE TRANSMITTED BUFFER STILL CONTAINS
6499      ;THE TEST PATTERN GENERATED IN THE BIT 16 TEST.
6500 035646 005037 035170      CLR      RSEL4      ;RECEIVE BUFFER ADDRESS (BITS 0-15)
6501 035652 012737 120000 035172      MOV      #BIT15!20000,RSEL6 ;REC BUFFER ADDR BIT 17 SET AND 8K
6502              ;BYTE RECEIVE CHARACTER COUNT
6503 035660 005037 035206      CLR      TSEL4      ;TRANSMIT BUFFER ADDRESS (BITS 0-15)
6504 035664 012737 060000 035210      MOV      #BIT14!20000,TSEL6 ;XMIT BUFFER ADDR BIT 16 SET AND 8K
6505              ;BYTE XMIT CHARACTER COUNT
6506 035672 012701 010000      MOV      #10000,R1   ;COUNTER TO CLEAR 8K BYTES
6507 035676 012704 140000      MOV      #140000,R4  ;VIRTUAL ADDRESS THAT WILL MAP INTO PAR 6
6508              ;WITH THE PHYSICAL ADDRESS 400000
6509 035702 005037 002340      CLR      FLAG       ;ENSURE FLAG IS CLEAR
6510 035706      SETVEC  #4,#NOXMEM,#PRI07 ;SET UP TRAP TO VECTOR 4 (WILL SET FLAG)
(7) 035706 012746 000340      MOV      #PRI07,-(SP)
(6) 035712 012746 023512      MOV      #NOXMEM,-(SP)
(5) 035716 012746 000004      MOV      #4,-(SP)
(4) 035722 012746 000003      MOV      #3,-(SP)
(3) 035726 104437              TRAP     C$SVEC
(2) 035730 062706 000010      ADD     #10,SP
6511 035734 012737 000001 177572      MOV     #1,@#177572 ;TURN ON MEMORY MANAGEMENT
6512 035742              74$:
6513 035742 005024              CLR     (R4)+      ;CLEAR 400000 - 417776
6514 035744 005737 002340      TST     FLAG       ;DOES A NXM TRAP 4 OCCUR?
6515 035750 001002              BNE     75$        ;IF YES, EXIT
6516 035752 005300              DEC     R0
6517 035754 001372              BNE     74$
6518 035756              75$:
6519 035756 005037 177572      CLR     @#177572   ;TURN OFF MEMORY MANAGEMENT
6520 035762      CLRVEC #4         ;RESTORE TRAP 4
(3) 035762 012700 000004      MOV     #4,R0
(3) 035766 104436              TRAP   C$CVEC
6521 035770 005737 002340      TST     FLAG       ;WAS THIS AN ERROR EXIT?
6522 035774 001002              BNE     76$        ;IF YES - REPORT ERROR
6523 035776 000137 035162      JMP     35$        ;START THE SECOND TEST
6524 036002              76$:
6525 036002      ERRDF 19,EMT22
(4) 036002 104455              TRAP   C$ERDF
(5) 036004 000023              .WORD 19
(5) 036006 036046              .WORD EMT22
(5) 036010 000000              .WORD 0
6526 036012      PRINTB #FMT25,R4
(8) 036012 010446              MOV     R4,-(SP)
(7) 036014 012746 036104      MOV     #FMT25,-(SP)
(6) 036020 012746 000002      MOV     #2,-(SP)

```

```

(3) 036024 010600
(4) 036026 104414
(4) 036030 062706 000006
6527 036034
6528 036034
(1)
(1) 036034 004737 012560
(1)
6529 036040
6530
6531 036044
6532
6533 036044
(3) 036044
(3) 036044 104401
6534
6535
6536 036046 040503 023516 020124
036054 042101 051104 051505
036062 020123 054105 042524
036070 042116 042105 046440
036076 046505 051117 000131
6537 036104 040445 042515 047515
036112 054522 040440 042104
036120 042522 051523 022440
036126 033117 020045 047504
036134 051505 047040 052117
036142 051040 051505 047520
036150 042116 026440 052040
036156 040522 020120 022464
036164 000116

MOV SP,RO
TRAP C$PNTB
ADD #6,SP

80$: SHUTDN ;SHUTDOWN DMR
;**** MACRO EXPANSION ****
JSR PC, $HALT ;DMR HALT ROUTINE.
;****
CALL $ERROR ;CHECK BASE TABLE AND REPORT ANY SOFT ERRORS

85$: .DSABL LSB ;DISABLE LOCAL SYMBOL BLOCK

ENDTST

L10074:
TRAP C$ETST

EMT22: .ASCIZ /CAN'T ADDRESS EXTENDED MEMORY/

FMT25: .ASCIZ /%MEMORY ADDRESS %06% DOES NOT RESPOND - TRAP 4%/

.EVEN

```

6538  
6539  
6540  
6541  
6542  
6543  
6544  
6545  
6546  
6547  
6548  
6549  
6550  
6551  
6552  
6553  
6554  
6555  
6556  
6557  
6558  
6559  
6560  
6561  
6562

```

.SBTTL TEST 15 - DMC MODE (RESUME) INTERRUPT TEST

:*****
:* TEST 15 - DMR-11
:* RESUME BASE IN - DMC MODE
:* ** WILL NOT RUN IF MODEM LOOPBACK IS SELECTED **
:* IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
:* TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
:* FOLLOWING MANNER:
:* BASE IN
:* CONTROL IN
:* HALT - BASE IN RESUME
:* 2 BA/CC IN RECEIVE
:* HALT - BASE IN RESUME
:* 2 BA/CC IN RECEIVE
:* HALT - BASE IN RESUME
:* 2 BA/CC IN RECEIVE
:* HALT - BASE IN RESUME
:* 1 BA/CC IN RECEIVE
:* HALT - BASE IN RESUME

```

```

6563 : * 2 BA/CC IN TRANSMIT
6564 : * HALT - BASE IN RESUME
6565 : * 2 BA/CC IN TRANSMIT
6566 : * HALT - BASE IN RESUME
6567 : * 2 BA/CC IN TRANSMIT
6568 : * HALT - BASE IN RESUME
6569 : * 1 BA/CC IN TRANSMIT
6570 : * HALT - BASE IN RESUME
6571 : *
6572 : *
6573 : *
6574 : *
6575 : *
6576 : *
6577 : *
6578 : *
6579 : *
6580 : *
6581 : *
6582 : *
6583 : *
6584 : *
6585 : *
6586 : *
6587 : *
  
```

ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST THE RECEIVE/TRANSMIT TABLE.

THE BUFFERS ARE DETERMINED IN THE SUBROUTINE \$BUFFS. THIS SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING HIERARCHY:

- A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
- B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER THAN 2K BYTES, USE THAT MEMORY
- C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.

\*\*\*\*\*  
 BGNTST

```

6588 036166          TST      WMAINT      ;DO WE NEED TO WRITE MODEM
(3) 036166          ;MAINTENACE 1 OR 2?
6589 036166 005737 002306      BNE      40$        ;IF YES WE CAN'T RUN THIS TEST
6590                                ;(NOTE: CAN'T WRITE MODEM IN DMC MODE)
6591 036172 001036      MOV      #7,BUFNUM   ;# OF RCV & XMIT BUFFERS.
6592                                ;FLAG SET TO REQUEST USE OF RESUME.
6593 036174 012737 000007 002322  MOV      #1,RESUME  ;FLAG SET TO REQUEST DMC MODE.
6594 036202 012737 000001 002274  MOV      #1,DMCMDE ;FLAG NOT TO REQUEST MAINTENANCE MODE.
6595 036210 012737 000001 002276  CLR      MNTMDE
6596 036216 005037 002300
6597
6598 036222          CALL     $BUFFS    ;DETERMINE 7 RCV & 7 XMIT BUFFERS
6599
6600 036226          CLEAR
(1)                                ;MASTER CLEAR
(1) 036226 004737 011064      JSR      PC, $MSCLR ;**** MACRO EXPANSION ****
(1)                                ;ISSUE A DMR MASTER CLEAR
6601                                ;****
6602 036232          ESCAPE  TST      ;IF ERROR, EXIT TEST
(3) 036232 104410
(3) 036234 000034          TRAP   C$ESCAPE
6603                                .WORD  L10075-
6604
6605                                ;ISSUE A DMR MODE BASEIN
6606 036236          BASEIN  ;IN DMR MODE, IF A INTERFACE IS REQUIRED
(1)                                ;TO BE WRITTEN - IT WILL BE DONE.
(1) 036236 004737 011262      JSR      PC, $BASEI ;**** MACRO EXPANSION ****
(1) 036242 004000          ;CALL BASE IN ROUTINE WITH DEFAULTS
(1) 036244 002630          ;SET LINE UNIT LOOP
(1) 036246 000522          ;BASE TABLE ADDRESS
                                ;DMR-11 MODE
                                ;****
  
```

```

6607
6608 036250          ESCAPE TST          ;IF ERROR, EXIT TEST
(3) 036250 104410
(3) 036252 000016          TRAP      C$ESCAPE
6609 036254          SHUTDN              ;HALT          .WORD      L10075-.
(1)
(1) 036254 004737 012560 JSR      PC, $HALT ;**** MACRO EXPANSION ****
(1)                      ;DMR HALT ROUTINE.
6610 036260          ESCAPE TST          ;****          ****
(3) 036260 104410          ;IF ERROR, EXIT TEST.
(3) 036262 000006          TRAP      C$ESCAPE
6611                      ;AFTER SETTING INTERFACE OR LOOPBACK MAINT.
6612                      ;(CAN'T BE DONE IN DMC MODE), START IN
6613                      ;DMC MODE.
6614
6615
6616 036264          CALL      $INOUT      ;THIS ROUTINE WILL MANAGE ALL THE DMR
6617                      ;COMMANDS ISSUED IN THE INTERRUPT ROUTINES
6618                      ;(FROM BASE IN UNTIL SHUT DOWN). BESIDES
6619                      ;CONTROLLING THE SOFTWARE TIMEOUT, THIS
6620                      ;ROUTINE WILL ALSO CHECK THAT BUFFER
6621                      ;CHARACTER COUNTS AND ADDRESSES ARE CORRECT
6622                      ;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS
6623
6624 036270          40$:
6625 036270          ENDTST
(3) 036270
(3) 036270 104401          L10075: TRAP      C$ETST
6626
6627
6628
6629
6630
6631
6632
6633
6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652
6653
    
```

```

.SBTTL          TEST 16 - DMR MODE (RESUME) INTERRUPT TEST
:*****
:*          TEST 16 - DMR-11
:* RESUME BASE IN - DMR MODE
:* IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
:* TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
:* FOLLOWING MANNER:
:*          BASE IN
:*          CONTROL IN
:*          HALT -          BASE IN RESUME
:*          2 BA/CC IN RECEIVE
:*          HALT -          BASE IN RESUME
:*          2 BA/CC IN RECEIVE
:*          HALT -          BASE IN RESUME
:*          2 BA/CC IN RECEIVE
:*          HALT -          BASE IN RESUME
:*          1 BA/CC IN RECEIVE
:*          HALT -          BASE IN RESUME
:*          2 BA/CC IN TRANSMIT
:*          HALT -          BASE IN RESUME
:*          2 BA/CC IN TRANSMIT
    
```

```

6654      :*      HALT      -      BASE IN RESUME
6655      :*      2 BA/CC IN TRANSMIT
6656      :*      HALT      -      BASE IN RESUME
6657      :*      1 BA/CC IN TRANSMIT
6658      :*      HALT      -      BASE IN RESUME
6659      :*
6660      :*      ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
6661      :*      THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
6662      :*      THE RECEIVE/TRANSMIT TABLE.
6663      :*
6664      :*      THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
6665      :*      SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
6666      :*      SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL
6667      :*      ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
6668      :*      HIERARCHY:
6669      :*      A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
6670      :*      B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
6671      :*      THAN 2K BYTES, USE THAT MEMORY
6672      :*      C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
6673      :*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
6674      :*
6675      :*
6676      :*
6677      :*      *****
    
```

```

6677 036272      BGNTST
(3) 036272
6678 036272 012737 000007 002322      MOV      #7,BUFNUM      ;# OF RCV & XMIT BUFFERS.
6679 036300 012737 000001 002274      MOV      #1,RESUME     ;FLAG SET TO REQUEST USE OF RESUME.
6680 036306 005037 002276      CLR      DMCMD         ;FLAG CLEARED - DMR MODE.
6681 036312 005037 002300      CLR      MNTMDE        ;FLAG NOT TO REQUEST MAINTENANCE MODE.
6682
6683 036316      CALL     $BUFFS        ;DETERMINE 7 RCV & 7 XMIT BUFFERS
6684
6685 036322      CLEAR
(1)
(1) 036322 004737 011064      JSR      PC, $MSCLR     ;MASTER CLEAR
(1)                                     ;**** MACRO EXPANSION ****
6686                                     ;ISSUE A DMR MASTER CLEAR
6687                                     ;****
6687 036326      ESCAPE  TST
(3) 036326 104410                                     ;IF ERROR, EXIT TEST
(3) 036330 000006                                     TRAP    C$ESCAPE
6688                                               .WORD   L10076-
6689 036332      CALL     $INOUT
6690                                               ;THIS ROUTINE WILL MANAGE ALL THE DMR
6691                                               ;COMMANDS ISSUED IN THE INTERRUPT ROUTINES
6692                                               ;(FROM BASE IN UNTIL SHUT DOWN). BESIDES
6693                                               ;CONTROLLING THE SOFTWARE TIMEOUT, THIS
6694                                               ;ROUTINE WILL ALSO CHECK THAT BUFFER
6695                                               ;CHARACTER COUNTS AND ADDRESSES ARE CORRECT
6696                                               ;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS
6697
6698
6699 036336      ENDTST
(3) 036336
(3) 036336 104401      L10076:  TRAP    C$ETST
6700
6701
    
```

6702  
6703  
6704  
6705  
6706  
6707  
6708  
6709  
6710  
6711  
6712  
6713  
6714  
6715  
6716  
6717  
6718  
6719  
6720  
6721  
6722  
6723  
6724  
6725  
6726  
6727  
6728  
6729  
6730  
6731  
6732  
6733  
6734  
6735  
6736  
6737  
6738  
6739  
6740  
6741  
6742  
6743  
6744  
6745  
6746  
6747  
6748  
6749  
6750  
6751

```
.SBTTL          TEST 17 - DMR MODE INTERRUPT EXERCISE
:*****
:*              TEST 17 - DMR-11
:* INTERRUPT DRIVEN EXERCISE
:* IN THIS TEST 64 BUFFERS WILL BE TRANSMITTED AND RECEIVED
:*
:* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* 64 RECEIVE AND 64 TRANSMIT BUFFERS. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*****
```

```
BGNTST
(3) 036340 012737 000100 002322
036340 012737 000100 002322
036346 005037 002274
036352 005037 002276
036356 005037 002300
036362
036366
(1) 036366 004737 011064
(1)
(1)
036372 104410
(3) 036372 104410
(3) 036374 000012
036376
036402
036406
```

```
MOV      #64.,BUFNUM      ;# OF RCV & XMIT BUFFERS.
CLR      RESUME           ;FLAG CLEARED IN ORDER NOT TO USE RESUME.
CLR      DMCMD            ;FLAG CLEARED TO ALLOW DMR MODE.
CLR      MNTMDE          ;FLAG NOT TO REQUEST MAINTENANCE MODE.
CALL     $BUFFS          ;DETERMINE 64 RCV & 64 XMIT BUFFERS
CLEAR    ;MASTER CLEAR
JSR     PC, $MSCLR       ;**** MACRO EXPANSION ****
                        ;ISSUE A DMR MASTER CLEAR
                        ;****
ESCAPE  TST              ;IF ERROR, EXIT TEST
TRAP    C$ESCAPE
        .WORD    L10077-.
CALL    $INOUT           ;THIS ROUTINE WILL MANAGE ALL THE DMR
                        ;COMMANDS ISSUED IN THE INTERRUPT ROUTINES
                        ;(FROM BASE IN UNTIL SHUT DOWN). BESIDES
                        ;CONTROLLING THE SOFTWARE TIMEOUT, THIS
                        ;ROUTINE WILL ALSO CHECK THAT BUFFER
                        ;CHARACTER COUNTS AND ADDRESSES ARE CORRECT
                        ;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS
CALL    $ERROR           ;CHECK BASE TABLE FOR SOFT ERRORS
```

ENDTST



L10077: TRAP C\$ETST

(3) 036406  
 (3) 036406 104401  
 6752  
 6753  
 6754  
 6755  
 6756  
 6757  
 6758  
 6759  
 6760  
 6761  
 6762  
 6763  
 6764  
 6765  
 6766  
 6767  
 6768  
 6769  
 6770  
 6771  
 6772  
 6773  
 6774  
 6775  
 6776  
 6777  
 6778  
 6779 036410  
 (3) 036410  
 6780  
 6781 036410 012737 000001 002322  
 6782  
 6783 036416 005037 002274  
 6784 036422 005037 002276  
 6785 036426 005037 002300  
 6786  
 6787 036432  
 6788  
 6789 036436  
 (1)  
 (1) 036436 004737 011064  
 (1)  
 6790  
 6791 036442  
 (3) 036442 104410  
 (3) 036444 000012  
 6792  
 6793 036446  
 6794  
 6795  
 6796  
 6797  
 6798  
 6799

```
.SBTTL          TEST 18 - DMR MODE LARGE MESSAGE
:*****
:*              TEST 18 - DMR-11
:* LARGE MESSAGE
:* IN THIS MODE TRANSMIT AND RECEIVE 1 LARGE BUFFER
:*
:* THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* ONE RECEIVE AND ONE TRANSMIT BUFFER. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*
:*****
BGNTST
:*****
T18::
MOV    #1,BUFNUM      ;# OF RCV & XMIT BUFFERS.
CLR    RESUME         ;FLAG CLEARED IN ORDER NOT TO USE RESUME.
CLR    DMCMD          ;FLAG CLEARED TO ALLOW DMR MODE.
CLR    MNTMDE         ;FLAG NOT TO REQUEST MAINTENANCE MODE.
CALL   $BUFFS        ;DETERMINE 1 RCV & 1 XMIT BUFFER
CLEAR  ;MASTER CLEAR
:**** MACRO EXPANSION ****
JSR    PC, $MSCLR    ;ISSUE A DMR MASTER CLEAR
:****
ESCAPE TST           ;IF ERROR, EXIT TEST
TRAP   C$ESCAPE     ;TRAP
        .WORD       L10100-
CALL   $INOUT        ;THIS ROUTINE WILL MANAGE ALL THE DMR
:COMMANDS ISSUED IN THE INTERRUPT ROUTINES
:(FROM BASE IN UNTIL SHUT DOWN). BESIDES
:CONTROLLING THE SOFTWARE TIMEOUT, THIS
:ROUTINE WILL ALSO CHECK THAT BUFFER
:CHARACTER COUNTS AND ADDRESSES ARE CORRECT
:AND THAT THE DATA IS CORRECT IN THOSE BUFFERS
```

```

6800
6801 036452          CALL    $ERROR          ;CHECK BASE TABLE FOR SOFT ERRORS
6802                                     ;NOTE: NORMALLY ANY NON-ZERO ERROR COUNT IS
6803                                     ;REPORTED; HOWEVER IN THIS TEST A REP COUNT
6804                                     ;OF 1 IS ALLOWED, BECAUSE AT LOW BAUD RATES
6805                                     ;WE WOULD EXPECT 1 REP.
6806
6807
6808 036456          ENDTST
    (3) 036456
    (3) 036456 104401          L10100: TRAP C$ETST
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835 036460
    (3) 036460
6836 036460 012737 000001 002322
6837
6838 036466 005037 002274
6839 036472 005037 002276
6840 036476 012737 000001 002300
6841
6842 036504
6843
6844 036510
    (1)
    (1) 036510 004737 011064
    (1)
6845
6846 036514          ESCAPE TST          ;IF ERROR, EXIT TEST
    (3) 036514 104410
    (3) 036516 000012          TRAP C$ESCAPE
6847                                     .WORD L10101-
```

```

.SBTTL          TEST 19 - DMR MAINTENANCE MODE MESSAGE
:*****
:*          TEST 19 - DMR-11
:* MAINTENANCE MODE OPERATION
:*
:* THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* ONE RECEIVE AND ONE TRANSMIT BUFFER. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*
:*****
BGNTST
```

T19::

```

MOV    #1,BUFNUM          ;# OF RCV & XMIT BUFFERS.
CLR    RESUME             ;DON'T ALLOW RESUME
CLR    DMCMD             ;FLAG CLEARED TO ALLOW DMR MODE.
MOV    #1,MNTMDE         ;FLAG SET TO REQUEST MAINTENANCE MODE.
CALL   $BUFFS            ;DETERMINE 1 RCV & 1 XMIT BUFFER
CLEAR  ;MASTER CLEAR
;**** MACRO EXPANSION ****
JSR    PC, $MSCLR        ;ISSUE A DMR MASTER CLEAR
;****
;****

```

6848 036520 CALL \$INOUT ;THIS ROUTINE WILL MANAGE ALL THE DMR  
6849 ;COMMANDS ISSUED IN THE INTERRUPT ROUTINES  
6850 ;(FROM BASE IN UNTIL SHUT DOWN). BESIDES  
6851 ;CONTROLLING THE SOFTWARE TIMEOUT, THIS  
6852 ;ROUTINE WILL ALSO CHECK THAT BUFFER  
6853 ;CHARACTER COUNTS AND ADDRESSES ARE CORRECT  
6854 ;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS  
6855

6856 036524 CALL \$ERROR ;CHECK BASE TABLE FOR SOFT ERRORS  
6857  
6858

6859 036530 ENDTST  
(3) 036530 L10101: TRAP C\$ETST  
(3) 036530 104401

.SBTTL HARDWARE PARAMETER CODING SECTION

\*\*\*\*\*  
: THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS  
: THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE  
: MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE  
: INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE  
: MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS  
: WITH THE OPERATOR.  
\*\*\*\*\*

6878 036532 BGNHRD  
(3) 036532 000015  
(3) 036534 .WORD L10102-L\$HARD/2  
L\$HARD::

6880 036534 GPRMA P1,2,0,160000,177776,YES  
(4) 036534 001031 .WORD T\$CODE  
(4) 036536 036566 .WORD P1  
(4) 036540 160000 .WORD T\$LLOLIM  
(4) 036542 177776 .WORD T\$HILIM

6881 036544 GPRMA P2,4,0,0,776,YES  
(4) 036544 002031 .WORD T\$CODE  
(4) 036546 036604 .WORD P2  
(4) 036550 000000 .WORD T\$LLOLIM  
(4) 036552 000776 .WORD T\$HILIM

6882 036554 GPRMD P3,20,0,7,0,7,YES  
(4) 036554 010032 .WORD T\$CODE  
(4) 036556 036625 .WORD P3  
(4) 036560 000007 .WORD 7  
(4) 036562 000000 .WORD T\$LLOLIM  
(4) 036564 000007 .WORD T\$HILIM

6883  
6884 036566 ENDHRD  
(2)  
(3) 036566 L10102: .EVEN

6885						
6886	036566	051503	020122	042101	P1:	.ASCIZ /CSR ADDRESS: /
	036574	051104	051505	035123		
	036602	000040				
6887	036604	042526	052103	051117	P2:	.ASCIZ /VECTOR ADDRESS: /
	036612	040440	042104	042522		
	036620	051523	020072	000		
6888	036625	124	051505	020124	P3:	.ASCII /TEST CONFIGURATION -/<CR><LF>
	036632	047503	043116	043511		
	036640	051125	052101	047511		
	036646	020116	006455	012		
6889	036653	040	030040	036440		.ASCII / 0 = INTERNAL (NO CONNECTOR)/<CR><LF>
	036660	044440	052116	051105		
	036666	040516	020114	047050		
	036674	020117	047503	047116		
	036702	041505	047524	024522		
	036710	005015				
6890	036712	020040	020061	020075		.ASCII / 1 = H3254 - V.35 (NOTE: MODE 1-4 ALLOWS/<CR><LF>
	036720	031510	032462	020064		
	036726	020055	027126	032463		
	036734	020040	020040	020040		
	036742	047050	052117	035105		
	036750	020040	047515	042504		
	036756	030440	032055	040440		
	036764	046114	053517	006523		
	036772	012				
6891	036773	040	031040	036440		.ASCII / 2 = H3254 - INTEGRAL PROGRAM INTERFACE SELECTION)/
	037000	044040	031063	032065		
	037006	026440	044440	052116		
	037014	043505	040522	020114		
	037022	020040	051120	043517		
	037030	040522	020115	047111		
	037036	042524	043122	041501		
	037044	020105	042523	042514		
	037052	052103	047511	024516		
6892	037060	005015	020040	020063		.ASCII <CR><LF>/ 3 = H3255 - RS232C/<57>/423/<CR><LF>
	037066	020075	031510	032462		
	037074	020065	020055	051522		
	037102	031462	041462	032057		
	037110	031462	005015			
6893						
6894	037114	020040	020064	020075		.ASCII / 4 = H3255 - RS422/<CR><LF>
	037122	031510	032462	020065		
	037130	020055	051522	031064		
	037136	006462	012			
6895	037141	040	032440	036440		.ASCII / 5 = CABLE AND SW PACK INTERFACE SELECTED/<CR><LF>
	037146	041440	041101	042514		
	037154	040440	042116	051440		
	037162	020127	040520	045503		
	037170	044440	052116	051105		
	037176	040506	042503	051440		
	037204	046105	041505	042524		
	037212	006504	012			
6896	037215	040	020040	020040		.ASCII / (V.35-H3250, INTEGRAL-BC55A-10, /
	037222	024040	027126	032463		
	037230	044055	031063	030065		

037236 020054 047111 042524  
037244 051107 046101 041055  
037252 032503 040465 030455  
037260 026060  
6897 037262 051040 031123 031063  
037270 026503 031510 032462  
037276 020054 051522 031064  
037304 027463 031064 026462  
037312 031510 032462 024461  
037320 005015  
6898 037322 020052 042523 042514  
037330 052103 052040 042510  
037336 043040 046117 047514  
037344 044527 043516 047440  
037352 046116 020131 043111  
037360 052040 042510 046440  
037366 042117 046505 051440  
037374 050125 047520 052122  
037402 020123 047514 050117  
037410 040502 045503 025040  
6899 037416 005015 020040 020066  
037424 020075 047514 040503  
037432 020114 047514 050117  
037440 005015  
6900 037442 020040 020067 020075  
037450 042522 047515 042524  
037456 046040 047517 006520  
037464 000012

.ASCII / RS232C-H325, RS423/<57>/422-H3251)/<CR><LF>

.ASCII /\* SELECT THE FOLLOWING ONLY IF THE MODEM SUPPORTS LOOPBACK \*/

.ASCII <CR><LF>/ 6 = LOCAL LOOP/<CR><LF>

.ASCIZ / 7 = REMOTE LOOP/<CR><LF>

.EVEN

.SBTTL SOFTWARE PARAMETER CODING SECTION

\*\*\*\*\*  
: THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS  
: THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE  
: MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE  
: INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE  
: MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS  
: WITH THE OPERATOR.  
:\*\*\*\*\*

6918 037466  
(3) 037466 000005  
(3) 037470  
6919  
6920 037470  
(4) 037470 000032  
(4) 037472 037502  
(4) 037474 000007  
(4) 037476 000001  
(4) 037500 000005  
6921

BGNSFT

.WORD L10103-L\$SOFT/2  
L\$SOFT::

GPRMD S1,0,0,7,1,5,YES

.WORD T\$CODE  
.WORD S1  
.WORD 7  
.WORD T\$LOLIM  
.WORD T\$HILIM

6922 037502  
 (2)  
 (3) 037502  
 6923  
 6924 037502 042523 042514 052103  
 037510 041101 042514 050040  
 037516 047522 051107 046501  
 037524 046040 047517 020120  
 037532 044524 042515 047455  
 037540 052125 053040 051101  
 037546 040511 046102 006505  
 037554 012  
 6925 037555 133 042522 042506  
 037562 020122 047524 046040  
 037570 051511 044524 043516  
 037576 033040 031456 030456  
 037604 056463 020040 024040  
 037612 040515 036530 035465  
 037620 046440 047111 030475  
 037626 020051 000  
 037632  
 6926  
 6927  
 6928  
 6929 037632  
 6930 037732 037732  
 6931 037732 000240  
 6932 037734 000240  
 6933 037736 000240  
 6934  
 6935 037740  
 6936  
 6937 037740  
 (2)  
 (4) 037740 000000  
 (4) 037742 000000  
 (3) 037744  
 6938 000001

ENDSFT

L10103: .EVEN

S1: .ASCII /SELECTABLE PROGRAM LOOP TIME-OUT VARIABLE/<CR><LF>

.ASCIZ /[REFER TO LISTING 6.3.13] (MAX=5; MIN=1) /

.EVEN

\*\*\*\*\* PATCH AREA \*\*\*\*\*

PATCH:  
 . = +100  
 NOP  
 NOP  
 NOP

\*\*\*\*\*  
 ÉNDMOD

LASTAD

.EVEN  
 .WORD 0  
 .WORD 0

L\$LAST::  
 .END











FMG1	016276	4044	4066	4075	4171#	5421									
FMG10	016752	4133	4180#												
FMG11	017010	4146	4181#												
FMG12	017042	4140	4182#												
FMG13	017073	4141	4147	4183#											
FMG14	017147	4148	4184#												
FMG15	017223	4142	4185#												
FMG16	017252	4153	4186#												
FMG17	017331	4157	4187#												
FMG18	017404	4163	4188#												
FMG19	017473	4093	4189#	5184											
FMG2	016330	4045	4067	4172#											
FMG20	017516	4097	4190#												
FMG21	017545	4046	4191#												
FMG22	017625	4047	4192#												
FMG23	017652	4056	4193#												
FMG24	017733	4057	4194#												
FMG3	016362	4043	4064	4073	4105	4173#									
FMG4	016434	4080	4174#												
FMG5	016465	4085	4088	4175#											
FMG6	016516	4176#													
FMG7	016557	4112	4177#	5778											
FMG8	016630	4119	4178#												
FMG9	016701	4126	4179#												
FMS1	010600	3011	3034#												
FMS2	010641	3020	3035#												
FMS3	011742	3333	3352#												
FMT0	024040	4924	4929#												
FMT1	025074	4974	5132#												
FMT11	027602	5422	5449#												
FMT12	027633	5425	5450#												
FMT13	027670	5429	5451#												
FMT14	027725	5433	5452#												
FMT15	027762	5437	5453#												
FMT16	030017	5441	5454#												
FMT19	031172	5654	5661#												
FMT2	025143	4980	5133#												
FMT25	036104	6375	6526	6537#											
FMT3	025220	5074	5086	5134#											
FMT4	025304	5118	5135#												
FMT5	026423	5296	5321#												
FMT6	026454	5300	5322#												
FMT7	026505	5305	5323#												
FMT8	026536	5309	5324#												
FMT9	026601	5313	5325#												
FRSPAS	002266	2526#	4259*	4279*											
FRSTIM	002264	2525#	4256	4258*											
F\$AU =	000015	2209#													
F\$AUTO=	000020	2209#	4419	4439											
F\$BGN =	000040	2209#	2212	4042	4061	4070	4102	4139	4145	4152	4156	4160	4223	4246	
			4419	4459	4477	4735	4835	4853	4885	4913	4916	4966	4967	4990	4993
			5094	5096	5129	5131	5178	5181	5190	5200	5224	5225	5229	5233	5256
			5258	5261	5266	5289	5291	5293	5348	5349	5352	5356	5359	5361	5372
			5374	5380	5382	5385	5389	5392	5401	5403	5417	5418	5420	5480	5481
			5484	5487	5490	5499	5504	5506	5511	5520	5522	5525	5528	5531	5533
			5538	5540	5544	5556	5560	5562	5566	5593	5595	5598	5601	5611	5615



























\$WAIT	010266	2975#	3234	3266	3316	3331	3407	3474	3584	3588	5492	5513	5614	5636
		5706	5724	5752	5816	5848	5882	5939	5982	6032	6060	6084	6114	6141
		6202	6411											
.	= 037744	2203#	2642#	2650#	2657#	2662#	2664#	2669#	2999	3088	3180	3557#	3605#	3968
		4862#	5141#	5181	5190	5229	5233	5261	5266	5352	5356	5359	5361	5372
		5374	5385	5389	5392	5401	5403	5455#	5484	5487	5490	5499	5511	5525
		5528	5531	5533	5544	5556	5560	5598	5601	5611	5615	5623	5634	5637
		5659#	5686	5689	5692	5701	5707	5725	5734	5737	5740	5749	5753	5782#
		5804	5807	5809	5833	5836	5839	5866	5869	5872	5875	5915	5925	5928
		5931	5938	5940	5968	5971	5974	5983	6033	6061	6085	6102	6105	6108
		6115	6132	6135	6142	6186	6189	6191	6195	6199	6380	6383	6385	6403
		6409	6412	6602	6608	6610	6687	6738	6791	6846	6926#	6930#		







MSCHEC	2118#	2209#													
MSCNTO	2182#	2209#	6880#	6881#	6882#	6920#									
MSCOUN	2066#	2209#	3011#	3020#	3333#	4043#	4044#	4045#	4046#	4047#	4056#	4057#	4064#	4066#	4067#
	4073#	4075#	4080#	4085#	4088#	4093#	4097#	4105#	4112#	4119#	4126#	4133#	4140#	4141#	4142#
	4146#	4147#	4148#	4153#	4157#	4163#	4292#	4313#	4856#	4924#	4974#	4980#	5074#	5086#	5118#
	5184#	5296#	5300#	5305#	5309#	5313#	5421#	5422#	5425#	5429#	5433#	5437#	5441#	5654#	5778#
	6375#	6526#													
MSDATA	1867#	2209#	2251#	2687#	2692#										
MSDECR	2029#	2209#	2314#	2329#	4058#	4068#	4099#	4135#	4143#	4149#	4154#	4158#	4164#	4229#	4399#
	4439#	4461#	4730#	4830#	4839#	4858#	4913#	4926#	4990#	5094#	5129#	5131#	5200#	5256#	5289#
	5291#	5315#	5380#	5417#	5418#	5443#	5504#	5520#	5538#	5562#	5566#	5628#	5647#	5649#	5655#
	5729#	5773#	5774#	5779#	5828#	5861#	5894#	5896#	5952#	6000#	6045#	6073#	6097#	6127#	6154#
	6156#	6283#	6533#	6625#	6699#	6751#	6808#	6859#	6884#	6922#	6935#				
MSDEFA	2170#	2209#	6880#	6881#	6882#	6920#									
MSENDE	2074#	2209#	2314#	2329#	4058#	4068#	4099#	4135#	4143#	4149#	4154#	4158#	4164#	4399#	4439#
	4461#	4730#	4830#	4839#	4858#	4913#	4926#	4990#	5094#	5129#	5131#	5200#	5256#	5289#	5291#
	5315#	5380#	5417#	5418#	5443#	5504#	5520#	5538#	5562#	5566#	5628#	5647#	5649#	5655#	5729#
	5773#	5774#	5779#	5828#	5861#	5894#	5896#	5952#	6000#	6045#	6073#	6097#	6127#	6154#	6156#
	6283#	6533#	6625#	6699#	6751#	6808#	6859#	6884#	6922#	6935#					
MSERRI	1649#	2209#	3003#	3014#	3091#	3183#	3338#	3343#	3546#	3592#	3597#	3973#	3995#	4010#	4505#
	4740#	4751#	4762#	4771#	4802#	4807#	4813#	4921#	5072#	5084#	5116#	5198#	5239#	5254#	5272#
	5287#	5365#	5378#	5415#	5495#	5502#	5516#	5536#	5620#	5642#	5715#	5720#	5756#	5761#	5770#
	5819#	5824#	5851#	5856#	5885#	5890#	5943#	5948#	5986#	5992#	6036#	6041#	6064#	6069#	6088#
	6093#	6118#	6123#	6145#	6150#	6206#	6216#	6222#	6227#	6235#	6241#	6246#	6261#	6264#	6374#
	6416#	6423#	6429#	6434#	6439#	6445#	6450#	6473#	6525#						
MSESCA	2006#	2209#	5181#	5190#	5229#	5233#	5261#	5266#	5352#	5356#	5359#	5361#	5372#	5374#	5385#
	5389#	5392#	5401#	5403#	5484#	5487#	5490#	5499#	5511#	5525#	5528#	5531#	5533#	5544#	5556#
	5560#	5598#	5601#	5611#	5615#	5623#	5634#	5637#	5686#	5689#	5692#	5701#	5707#	5725#	5734#
	5737#	5740#	5749#	5753#	5804#	5807#	5809#	5833#	5836#	5839#	5866#	5869#	5872#	5875#	5915#
	5925#	5928#	5931#	5938#	5940#	5968#	5971#	5974#	5983#	6033#	6061#	6085#	6102#	6105#	6108#
	6115#	6132#	6135#	6142#	6186#	6189#	6191#	6195#	6199#	6380#	6383#	6385#	6403#	6409#	6412#
	6602#	6608#	6610#	6687#	6738#	6791#	6846#								
MSESCS	2010#	2209#	5181#	5190#	5229#	5233#	5261#	5266#	5352#	5356#	5359#	5361#	5372#	5374#	5385#
	5389#	5392#	5401#	5403#	5484#	5487#	5490#	5499#	5511#	5525#	5528#	5531#	5533#	5544#	5556#
	5560#	5598#	5601#	5611#	5615#	5623#	5634#	5637#	5686#	5689#	5692#	5701#	5707#	5725#	5734#
	5737#	5740#	5749#	5753#	5804#	5807#	5809#	5833#	5836#	5839#	5866#	5869#	5872#	5875#	5915#
	5925#	5928#	5931#	5938#	5940#	5968#	5971#	5974#	5983#	6033#	6061#	6085#	6102#	6105#	6108#
	6115#	6132#	6135#	6142#	6186#	6189#	6191#	6195#	6199#	6380#	6383#	6385#	6403#	6409#	6412#
	6602#	6608#	6610#	6687#	6738#	6791#	6846#								
MSEXCP	2101#	2209#	6880#	6881#	6882#	6920#									
MSEXIT	2014#	2209#													
MSEXSE	2022#	2209#													
MSEX TJ	2018#	2209#													
MSGEN	2038#	2209#	2251#	2277#	2300#	2314#	2325#	2329#	2687#	2692#	4042#	4058#	4061#	4068#	4070#
	4099#	4102#	4135#	4139#	4143#	4145#	4149#	4152#	4154#	4156#	4158#	4160#	4164#	4223#	4246#
	4399#	4419#	4439#	4459#	4461#	4477#	4730#	4735#	4830#	4835#	4839#	4853#	4858#	4885#	4913#
	4916#	4926#	4966#	4967#	4990#	4993#	5094#	5096#	5129#	5131#	5178#	5200#	5224#	5225#	5256#
	5258#	5289#	5291#	5293#	5315#	5348#	5349#	5380#	5382#	5417#	5418#	5420#	5443#	5480#	5481#
	5504#	5506#	5520#	5522#	5538#	5540#	5562#	5566#	5593#	5595#	5628#	5630#	5647#	5649#	5653#
	5655#	5682#	5683#	5729#	5731#	5773#	5774#	5777#	5779#	5800#	5801#	5828#	5830#	5861#	5863#
	5894#	5896#	5912#	5952#	5965#	6000#	6020#	6021#	6045#	6047#	6073#	6075#	6097#	6099#	6127#
	6129#	6154#	6156#	6172#	6283#	6301#	6533#	6588#	6625#	6677#	6699#	6727#	6751#	6779#	6808#
	6835#	6859#	6878#	6884#	6918#	6922#	6937#								
MSGENB	1938#	2209#													
MSGETS	2035#	2209#	2314#	2329#	4058#	4068#	4099#	4135#	4143#	4149#	4154#	4158#	4164#	4229#	4399#
	4439#	4461#	4730#	4830#	4839#	4858#	4913#	4926#	4990#	5094#	5129#	5131#	5200#	5256#	5289#

	5291#	5315#	5380#	5417#	5418#	5443#	5504#	5520#	5538#	5562#	5566#	5628#	5647#	5649#	5655#
	5729#	5773#	5774#	5779#	5828#	5861#	5894#	5896#	5952#	6000#	6045#	6073#	6097#	6127#	6154#
MSGETT	6156#	6283#	6533#	6625#	6699#	6751#	6808#	6859#	6884#	6922#	6935#				
	1877#	2209#	5181#	5190#	5229#	5233#	5261#	5266#	5352#	5356#	5359#	5361#	5372#	5374#	5385#
	5389#	5392#	5401#	5403#	5484#	5487#	5490#	5499#	5511#	5525#	5528#	5531#	5533#	5544#	5556#
	5560#	5598#	5601#	5611#	5615#	5623#	5634#	5637#	5686#	5689#	5692#	5701#	5707#	5725#	5734#
	5737#	5740#	5749#	5753#	5804#	5807#	5809#	5833#	5836#	5839#	5866#	5869#	5872#	5875#	5915#
	5925#	5928#	5931#	5938#	5940#	5968#	5971#	5974#	5983#	6033#	6061#	6085#	6102#	6105#	6108#
	6115#	6132#	6135#	6142#	6186#	6189#	6191#	6195#	6199#	6380#	6383#	6385#	6403#	6409#	6412#
MSGNGB	6602#	6608#	6610#	6687#	6738#	6791#	6846#								
	1902#	2209#	2212#	2251#	2277#	2300#	2325#	2687#	2692#	4042#	4061#	4070#	4102#	4139#	4145#
	4152#	4156#	4160#	4223#	4246#	4419#	4459#	4477#	4735#	4835#	4853#	4916#	5293#	5420#	5653#
MSGNIN	5777#	6878#	6918#	6937#											
	2049#	2209#	2251#	2277#	2300#	2325#	2687#	2692#	2996#	2999#	3003#	3011#	3014#	3020#	3085#
	3088#	3091#	3177#	3180#	3183#	3235#	3247#	3267#	3317#	3332#	3333#	3338#	3343#	3409#	3475#
	3546#	3585#	3587#	3589#	3592#	3597#	3742#	3752#	3817#	3818#	3952#	3959#	3968#	3973#	3995#
	4010#	4020#	4043#	4044#	4045#	4046#	4047#	4056#	4057#	4058#	4064#	4066#	4067#	4068#	4073#
	4075#	4080#	4085#	4088#	4093#	4097#	4099#	4105#	4112#	4119#	4126#	4133#	4135#	4140#	4141#
	4142#	4143#	4146#	4147#	4148#	4149#	4153#	4154#	4157#	4158#	4163#	4164#	4248#	4262#	4264#
	4265#	4266#	4267#	4268#	4269#	4270#	4271#	4285#	4286#	4292#	4313#	4330#	4331#	4399#	4421#
	4432#	4435#	4436#	4439#	4461#	4505#	4730#	4740#	4751#	4762#	4771#	4802#	4807#	4813#	4830#
	4839#	4855#	4856#	4858#	4887#	4907#	4908#	4911#	4913#	4921#	4924#	4926#	4967#	4974#	4980#
	4990#	4993#	5072#	5074#	5084#	5086#	5094#	5096#	5116#	5118#	5129#	5131#	5181#	5184#	5190#
	5198#	5200#	5225#	5229#	5233#	5239#	5254#	5256#	5258#	5261#	5266#	5272#	5287#	5289#	5291#
	5296#	5300#	5305#	5309#	5313#	5315#	5349#	5352#	5356#	5359#	5361#	5365#	5372#	5374#	5378#
	5380#	5382#	5385#	5389#	5392#	5401#	5403#	5415#	5417#	5418#	5421#	5422#	5425#	5429#	5433#
	5437#	5441#	5443#	5481#	5484#	5487#	5490#	5495#	5499#	5502#	5504#	5506#	5511#	5516#	5520#
	5522#	5525#	5528#	5531#	5533#	5536#	5538#	5540#	5544#	5556#	5560#	5562#	5566#	5595#	5598#
	5601#	5611#	5615#	5620#	5623#	5628#	5630#	5634#	5637#	5642#	5647#	5649#	5654#	5655#	5683#
	5686#	5689#	5692#	5701#	5707#	5715#	5720#	5725#	5729#	5731#	5734#	5737#	5740#	5749#	5753#
	5756#	5761#	5770#	5773#	5774#	5778#	5779#	5801#	5804#	5807#	5809#	5819#	5824#	5828#	5830#
	5833#	5836#	5839#	5851#	5856#	5861#	5863#	5866#	5869#	5872#	5875#	5885#	5890#	5894#	5896#
	5915#	5925#	5928#	5931#	5938#	5940#	5943#	5948#	5952#	5968#	5971#	5974#	5983#	5986#	5992#
	6000#	6021#	6033#	6036#	6041#	6045#	6047#	6061#	6064#	6069#	6073#	6075#	6085#	6088#	6093#
	6097#	6099#	6102#	6105#	6108#	6115#	6118#	6123#	6127#	6129#	6132#	6135#	6142#	6145#	6150#
	6154#	6156#	6186#	6189#	6191#	6195#	6199#	6203#	6206#	6216#	6222#	6227#	6235#	6241#	6246#
	6261#	6264#	6283#	6304#	6307#	6322#	6347#	6371#	6374#	6375#	6380#	6383#	6385#	6403#	6409#
	6412#	6416#	6423#	6429#	6434#	6439#	6445#	6450#	6473#	6510#	6520#	6525#	6526#	6533#	6602#
	6608#	6610#	6625#	6687#	6699#	6738#	6751#	6791#	6808#	6846#	6859#	6878#	6880#	6881#	6882#
MSGNLS	6884#	6918#	6920#	6922#	6937#										
MSGNSU	1913#	2209#													
	1898#	2209#	4967#	4993#	5096#	5225#	5258#	5349#	5382#	5481#	5506#	5522#	5540#	5595#	5630#
	5683#	5731#	5801#	5830#	5863#	6021#	6047#	6075#	6099#	6129#					
MSGNTA	1890#	2209#	2314#	2329#	4058#	4068#	4099#	4135#	4143#	4149#	4154#	4158#	4164#	4399#	4439#
	4461#	4730#	4830#	4839#	4858#	4913#	4926#	4990#	5094#	5129#	5131#	5200#	5256#	5289#	5291#
	5315#	5380#	5417#	5418#	5443#	5504#	5520#	5538#	5562#	5566#	5628#	5647#	5649#	5655#	5729#
	5773#	5774#	5779#	5828#	5861#	5894#	5896#	5952#	6000#	6045#	6073#	6097#	6127#	6154#	6156#
	6283#	6533#	6625#	6699#	6751#	6808#	6859#	6884#	6922#						
MSGNTE	1894#	2209#	4885#	4966#	5178#	5224#	5348#	5480#	5593#	5682#	5800#	5912#	5965#	6020#	6172#
	6301#	6588#	6677#	6727#	6779#	6835#									
MSHAPT	1739#	2209#	2251#												
MSHNAP	1824#	2209#	2251#												
MSINCR	2026#	2209#	2212#	2300#	2325#	2996#	3003#	3011#	3014#	3020#	3085#	3091#	3177#	3183#	3333#
	3338#	3343#	3546#	3592#	3597#	3742#	3752#	3817#	3818#	3952#	3959#	3973#	3995#	4010#	4020#
	4042#	4043#	4044#	4045#	4046#	4047#	4056#	4057#	4058#	4061#	4064#	4066#	4067#	4068#	4070#
	4073#	4075#	4080#	4085#	4088#	4093#	4097#	4099#	4102#	4105#	4112#	4119#	4126#	4133#	4135#







XFERT 1620# 2209#

. ABS. 037744 000

ERRORS DETECTED: 0

CZDMIA.BIN,CZDMIA.SEQ/CRF/NL:TOC=SVC34R.MLB,CZDMIA.P11  
RUN-TIME: 118 144 13 SECONDS  
RUN-TIME RATIO: 507/277=1.8  
CORE USED: 17K (33 PAGES)