

DMP - 11
DMV - 11

DMP/V - 11 FCTNL TST #1
CZDMTAAO

AH - E238A - MC
FICHE 1 OF 1

NOV 1980
COPYRIGHT © 1980
MADE IN USA



.TITLE CZDMTAO DMP/V-11 FCTNL TST #1
.REM 8

IDENTIFICATION

PRODUCT CODE: AC-E237A-MC
PRODUCT NAME: CZDMTAO DMP/V-11 FUNCTIONAL TEST 1
PRODUCT DATE: AUG, 1980
MAINTAINER: DIAGNOSTICS MERRIMACK CC: 38P

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	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

HISTORY

REV	DATE	REASON
---	---	-----
1	18-AUG-80	INITIAL RELEASE DMP ONLY

TABLE OF 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
- 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
 - 6.3.2 RESTART COMMAND
 - 6.3.3 CONTINUE COMMAND
 - 6.3.4 PROCEED COMMAND
 - 6.3.5 ADD COMMAND
 - 6.3.6 DROP COMMAND
 - 6.3.7 PRINT COMMAND
 - 6.3.8 DISPLAY COMMAND
 - 6.3.9 FLAGS COMMAND
 - 6.3.1 ZFLAGS COMMAND
 - 6.3.1 CONTROL CHARACTERS
 - 6.3.1 HARDWARE PARAMETERS
 - 6.3.1 SOFTWARE PARAMETERS
 - 6.3.1 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
- 7.0 TEST DESCRIPTIONS
- 8.0 ERROR INFORMATION
 - 8.1 ERROR REPORTING

* NOTE * THIS PROGRAM HAS BEEN TESTED FOR DMP ONLY

1.0 INTRODUCTION

THE DMP AND DMV OPTIONS ARE COMMUNICATION OPTIONS THAT IMPLEMENT THE DDCMP PROTOCOL IN A MULTIDROP ENVIRONMENT. THE DMP IS USED WITH UNIBUS SYSTEMS WHILE THE DMV IS A Q BUS OPTION. THE PURPOSE OF THIS FUNCTIONAL TEST IS TO VERIFY AND EXERCISE THE MICROCODE USED IN THIS OPTION. THIS IS DONE BY PERFORMING THE FOLLOWING TESTS.
CSR ADDRESSING TESTS, ROM VERIFICATION BY CRC TESTS, RUNNING MICRO DIAGNOSTICS, RUNNING INTERFACE DIAGS. (DMP ONLY), CHECKS FOR RDO AND RDI, CHECKS FOR VARIOUS PROCEDURE ERRORS, MODE DEFINITION CHECKS, TEST FOR ALL CONTROL IN COMMANDS AND TESTS FOR ALL CONTROL AND INFORMATION OUT COMMANDS, TRANSMIT, AND RECEIVE MESSAGE TESTS OF VARIOUS LENGTHS, TO AND FROM VARIOUS BUFFERS.

THE FUNCTIONAL DIAGNOSTIC TEST WILL PROVIDE EXTENSIVE TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS, SWITCH OPTIONS, AND ABILITY TO 'LOCK' ONTO INTERMITTENT ERRORS. IN ADDITION TESTS WILL BE DESIGNED AND STRUCTURED TO ACHIEVE MAXIMUM FAULT RESOLUTION AND FACILITATE REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.

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

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

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DMP/DMV-11 FUNCTIONAL TESTS:

FOR DMP:
PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70
DMP-11

FOR DMV:
LSI-11/03,
DMV-11

FOR BOTH:
16K MEMORY
CONSOLE TERMINAL

3.0 PRELIMINARY PROGRAM REQUIREMENTS

FOR DMP:
THE M8207 STATIC DIAGNOSTICS AND THE M8203 STATIC DIAGNOSTICS SHOULD BE RUN BEFORE RUNNING THIS FUNCTIONAL DIAG.

FOR DMV:
THE 8053/54 STATIC LOGIC TESTS AND THE 5930 STATIC LOGIC TESTS PARTS 1 AND 2.

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

THE TOTAL TIME REQUIRED TO RUN THE DMP-11 FUNCTIONAL TESTS IS ABOUT 120 SECONDS PER PASS FOR EACH UNIT.

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

IT IS USED IN TX AND RX TESTS.

4.7 MEMORY PARITY OPTION

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

4.8 ERROR LOGGING

THE NUMBER OF ERRORS WHICH HAVE OCCURRED ON EACH DEVICE UNDER TEST SINCE THE LAST START OR RESTART COMMAND IS KEPT IN AN ERROR LOG. THIS LOG MAY BE PRINTED BY USING THE 'PRINT' COMMAND (SEE SECTION 6.3.8).

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 WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

- A) LOAD AND START THE DIAGNOSTIC USING THE RUN COMMAND
- B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION PROMPT (DR)
- 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 THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED:

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CZDMT-A-0
DMP/V-11 FUNCTIONAL DIAG.
UNIT IS DMP-11 OR DMV-11
DR>
```

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

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

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

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

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS.

THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

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

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURRENCE 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 DIALOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

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

6.3.3 CONTINUE COMMAND

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

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

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

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

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

6.3.3.3 EFFECT OF CONTINUE COMMAND

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

6.3.4 PROCEED COMMAND

```
*****  
PROCEED)/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>

CZDMTAD DMP/V-11 PCTNL TST #1 MACY11 30A(1052) 12-AUG-80 15:53 ^{N 1} PAGE 14
CZDMTA.REL 12-AUG-80 15:52

SEQ 0013

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- INITIAL DIALOGUE (SEE 6.2), HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

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

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 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.

DEVICE CSR ADDRESS : (0) 160170 ?
DEVICE VECTOR ADDRESS : (0) 300 ?
DEVICE PRIORITY LEVEL : (0) 5 ?
TURNAROUND TYPE -(0=H3254H3255,1=CABLE,2=MOD LOC,3=MOD REM,4=NONE) (0) 0 ?
PLEASE SELECT BAUD RATE: TYPE '0' FOR 2, 4K; '1' FOR 4 8K;
'2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K;
'6' FOR 500K; OR '7' FOR 1 MEG BAUDS (0) 4 ?

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY THE DMP/V-11 FUNCTIONAL TEST

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 1
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 21
<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 16 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM 'UNIT XX' AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS A 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).

TEST DESCRIPTIONS

7.0

7.1 ADDRESS TEST (TEST-1)

VERIFIES THAT ALL ADDRESSES IN THE MCPU RESPOND.
THIS TEST IS USED TO VERIFY THAT THE OPTION
IS AT THE ADDRESS THE USER THINKS IT IS ON.

7.2 ROM VERIFICATION TESTS (TESTS 2-9) (2-7 DMP ONLY+++8-9 DMV ONLY)

THIS SERIES OF TESTS VERIFIES THAT ALL ROMS
ARE IN PLACE AND THAT THE THE CONTENTS ARE
CORRECT BY DOING A CRC CALCULATION ON THE
ROM CONTENTS. THE TEST ALSO PRINTS THE REV AND
ROM NUMBER OF THE ROM ON THE FIRST PASS OF THE
TEST.

7.3 INITIALIZATION TEST (TEST 10)

THIS TEST DOES A MASTER CLEAR TO THE DEVICE
AND WAITS FOR THE MICRO-DIAGNOSTICS TO COMPLETE
IF MICRO DIAGS FAIL TO COMPLETE THEN A TIME
OUT ERROR WILL BE REPORTED.

7.4 INTERFACE DIAGNOSTICS(DMP ONLY) (TEST 11)

THIS TEST RUNS ADDITIONAL MICRODIAGNOSTIC CODE IN THE
DMP THAT CHECKS OUT THE INTERRUPT LOGIC AND THE
NPR LOGIC.

7.5 RDI REMAINS SET TEST (TEST 12)

THIS TEST SETS RDI, WAITS FOR RDI TO SET, ISSUES
A 'NO REQUEST' CONTROL IN AND LOOKS FOR RDI TO
REMAIN SET.

7.6 TEST FOR RDO SETTING (TEST 13)

THIS TEST DOES A CONTROL IN COMMAND OF 'READ MODEM'
AND EXPECTS RDC TO SET WITH AN INFORMATION OUT CODE
OF RETURN MODEM STATUS.

7.7 CHECK FOR PROCEDURE ERROR 100 (TEST 14)

THIS TEST ISSUES A MASTER CLEAR WAITS FOR RUN TO
SET THEN ISSUES A CONTROL IN COMMAND AND EXPECTS
A PROCEDURE ERROR OF 100 'NON MODE DEFINITION COMMAND
AFTER A MASTER CLEAR'. UNLESS MODE HAS BEEN DEFINED
IN THE SWITCHES THEN LOOK FOR INFORMATION OUT.

7.8 CHECK FOR PROCEDURE ERROR 104 (TEST 15)

THIS TEST ISSUES A MASTER CLEAR, MODE DEFINITION, FOLLOWED BY A MODE DEF. COMMAND DEFINING A DIFFERENT TYPE OF MODE. THE TEST LOOKS FOR A PROCEDURE ERROR OF 104 "ILLEGAL MODE CHANGE".

7.9 TEST MODE CHANGE OF DUPLEX PORTION OF MODE (TEST 16)

THIS TEST ISSUES A MASTER CLEAR, MODE DEFINITION SEQUENCE (CONTROL STATION/FULL DUPLEX). THE TEST THEN ISSUES A MODE DEF. COMMAND TO CHANGE TO HALF DUPLEX. THEN THE TESTS WAITS AND MAKES SURE NO PROCEDURE ERROR OCCURS.

7.10 TEST FOR MAX TRIBS TO BE ESTABLISHED. (TEST 17)

THIS TEST ESTABLISHES MAX TRIBS THEN ATTEMPTS TO ESTABLISH MAX+1 TRIBS AND CHECKS FOR A PROCEDURE ERROR 114, "ATTEMPT TO ESTABLISH MORE THEN MAXIMUM NUMBER OF TRIBS". THE TEST THEN TRIES TO ESTABLISH A TRIB THAT HAS ALREADY BEEN ESTABLISHED AND CHECKS FOR A PROCEDURE ERROR OF 116 "ATTEMPT TO ESTABLISH ALREADY ESTABLISHED TRIB".

NOTE: MAX TRIBS FOR DMP = 32
MAX TRIBS FOR DMV = 12

7.11 READ/WRITE TRIBUTARY STATUS SLOTS TEST (TEST 18)

THIS TEST WRITES EACH TSS SLOT WITH VARIOUS DATA PATTERNS THEN READS THAT SLOT TO BE SURE THAT THE CORRECT OUTPUT COMMAND AND DATA IS RETURNED. THE SLOTS THAT ARE WRITTEN ARE TRIB STATUS SLOTS 30 THRU 37. THE DATA PATTERNS USED ARE: 0,125252,052525,0,-1,377,177400,562:OCTAL.

7.12 TESTS FOR PROCEDURE ERROR 132 (TEST 19-20)

THESE TESTS CHECK THAT A PROCEDURE ERROR OF 132 "ATTEMPT TO WRITE INTO A RESERVED AREA OF THE TRIBUTARY STATUS SLOTS" IS PRODUCED WHEN A WRITE TSS COMMAND IS ISSUED FOR ADDRESS 4. A READ/CLEAR TSS COMMAND IS ISSUED FOR ADDRESS 6.

7.13 TEST FOR READ/CLEAR COMMAND (TEST 21)

THIS TEST ISSUES A READ CLEAR COMMAND TO TRIBUTARY STATUS SLOT 7 AND MAKES SURE THAT NO ERRORS OCCUR.

7.14 TESTS FOR GLOBAL STATUS SLOTS (TEST 22)

THIS TEST READS ALL THE GLOBAL STATUS SLOTS THEN WRITES ALL THE GLOBAL SLOTS USING THE ADDRESSES AS DATA THEN READS THEM BACK AND MAKES SURE THE DATA IS CORRECT. THIS TEST ALSO CHECKS FOR THE LIMITS ON THE WRITE TSS COMMAND BY MAKING SURE A

PROCEDURE ERROR OCCURES WHEN THE LIMITS ARE EXCEEDED. THIS TEST ALSO CHECKS THE READ/CLEAR COMMAND TO A GLOBAL STATUS SLOT.

7.15 HALT TRIB COMMAND TESTS (TEST 23)

THIS TEST CHECKS THE HALT TRIB COMMAND BY DOING THE FOLLOWING: MASTER CLEAR; MODE DEF; ESTABLISH TRIB; ISTRT TRIB; QUE UP REC BUFFER; ISSUE HALT TRIB COMMAND; CHECK FOR OUTPUT OF REC BUFFER UNUSED; CHECK FOR SECOND OUTPUT OF BUFFER RETURNED COMPLETE. THE TEST THEN ISSUES A SECOND HALT TRIB COMMAND AND CHECKS THAT AFTER A DELAY NO CONTROL OUT OCCURS

7.16 KILL TRIB COMMAND TESTS (TEST 24)

THIS TEST CHECKS THE KILL TRIB COMMAND BY DOING THE FOLLOWING: MASTER CLEAR; MODE DEF.; ESTABLISH TRIB; READ TSS SLOT 1 AND COMPARE FOR GOOD ADDRESS; PUT TRIB IN MAINT STATE; ISSUE KILL TRIB; CHECK FOR PROCEDURE ERROR 112 'KILL TO UNHALTED TRIB'; HALT TRIB; KILL TRIB; READ TSS SLOT 1 AND CHECK FOR PROCEDURE ERROR 106 'NON GLOBAL CONTROL IN COMMAND TO UNESTABLISHED TRIB'.

7.17 CHECK FOR PROCEDURE ERROR 102 (TEST 25)

THIS TEST ISSUES ILLEGAL TYPE CODES OF 7 6 5 AND 3 AND CHECKS THAT EACH ONE PRODUCES A PROCEDURE ERROR 102 'ILLEGAL TYPE CODE USED IN AN INPUT COMMAND'.

7.18 CHECK FOR PROCEDURE ERROR OF 110 (TEST 26)

THIS TEST ISSUES A MASTER CLEAR; MODE DEF; FOLLOWED BY AN ISTRT TO TRIB ADDRESS OF ZERO. IT THEN CHECKS FOR A PROCEDURE ERROR OF 110 'ATTEMPT TO PERFORM A NON-GLOBAL COMMAND FOR TRIBUTARY ADDRESS OF 0'.

7.19 CHECK FOR PROCEDURE ERROR OF 120 (TEST 27)

THIS TEST ISSUES A CONTROL IN WITH A REQUEST KEY OF 7 AND ALSO A CONTROL IN WITH A REQUEST KEY OF 17 THEN IT CHECKS THAT BOTH CASES GIVE PROCEDURE ERROR 120 'ILLEGAL REQUEST KEY ON CONTROL IN.'

7.20 CHECK FOR PROCEDURE ERROR OF 134 (TEST 28)

THIS TEST ISSUES A MASTER CLEAR, MODE DEF, AND ESTABLISH TRIB SEQUENCE, FOLLOWED BY AN ATTEMPT TO USE A RESERVED BIT IN BSEL 7 THEN CHECKS THAT THIS PRODUCES A PROCEDURE ERROR OF 134 'ATTEMPT TO USE RESERVED BIT IN BSEL 7 ON CONTROL IN'

7.21 LATCH/UNLATCH POLL CHECK (TEST 29)

THIS TEST CHECKS THE LATCH AND UNLATCH POLL COMMANDS BY DOING THE FOLLOWING SEQUENCE OF COMMANDS:

MASTER CLEAR; MODE DEF; ESTABLISH TRIB; LATCH POLL
TO DEAD STATE; READ TSS SLOT 2 AND CHECK THAT DEAD
BIT IS ON; UNLATCH POLL; READ TSS SLOT 2; CHECK THAT ACTIVE
BIT IS ON.

7.22 SHORT MESSAGE SENDING TEST (TEST 30)

THIS TEST SENDS A 4 BYTE MESSAGE FROM AN EVEN TRANSMIT
BUFFER TO AN EVEN REC BUFFER IN DDCMP FORMAT CONFIGURED
AS A MULTIPOINT CONTROL STATION FULL DUPLEX. THE TEST
CHECKS THAT REC BUFFERS ARE RETURNED AND DATA IS CORRECT
AND THAT THE NEXT OUTPUT IS TRANSMIT BUFFER RETURNED.
THIS TEST IS ALWAYS DONE IN TTL LOOPBACK MODE.

7.23 CHECK FOR PROCEDURE ERROR 122 (TEST 31)

THIS TEST CHECKS FOR A PROCEDURE ERROR OF 122 BY
PERFORMING THE FOLLOWING: MASTER CLEAR; MODE DEF;
ESTABLISH BUFFER; CHECK FOR ERROR 122 'ATTEMPT
TO ASSIGN A BUFFER FOR AN UNESTABLISHED TRIB'.

7.24 CHECK FOR PROCEDURE ERROR 124 (TEST 32)

THIS TEST CHECKS FOR A PROCEDURE ERROR OF 124 BY
PERFORMING THE FOLLOWING: MASTER CLEAR; MODE DEF;
ESTABLISH TRIB; ESTABLISH BUFFER; CHECK FOR ERROR
124 'ATTEMPT TO ASSIGN A BUFFER FOR A HALTED TRIB'.

7.25 CHECK FOR PROCEDURE ERROR 126 (TEST 33)

THIS TEST CHECKS FOR A PROCEDURE ERROR OF 126 BY
PERFORMING THE FOLLOWING: MASTER CLEAR; MODE DEF
ESTABLISH TRIB; PUT TRIB IN MAINT STATE; ESTABLISH
BUFFER WITH 0 BYTE COUNT; LOOK FOR ERROR 126
'ATTEMPT TO ASSIGN A BUFFER WITH A BYTE COUNT OF 0'.

7.26 CHECK FOR PROCEDURE ERROR 130 (TEST 34)

THIS TEST CHECKS FOR A PROCEDURE ERROR OF 130 BY
PERFORMING THE FOLLOWING: MASTER CLEAR; MODE DEF;
ESTABLISH TX BUFFER TO TRIB; CHECK FOR ERROR
130 'ATTEMPT TO ASSIGN A TRANSMIT BUFFER FOR TRIB
0'.

7.27 TRANSMIT/RECEIVE 256 BYTES, PTP, DDCMP (TEST 35)

THIS TEST WILL TRANSMIT A BUFFER OF 256 BYTES, STARTING
ON AN EVEN BYTE BOUNDARY TO A REC BUFFER STARTING ON AN
EVEN BYTE BOUNDARY. THE MODE DEFINED IS POINT TO POINT
FULL DUPLEX, DDCMP PROTOCOL. IF THERE IS EXTERNAL LOOP
BACK THEN THE TEST WILL BE DONE OVER THAT LOOPBACK; ELSE
THE LOOPBACK WILL BE SET TO INTERNAL (TTL).

7.28 TRANSMIT/RECEIVE 255 BYTES, MTP, DDCMP (TEST 36)

THIS TEST WILL TRANSMIT A BUFFER OF 255 BYTES STARTING

ON AN EVEN BYTE ADDRESS TO A REC BUFFER STARTING ON AN ODD BYTE ADDRESS. THE MODE IS FULL DUPLEX CONTROL STATION MULTIPOINT, DDCMP PROTOCOL. THE DATA IS COMPARED FOR CORRECTNESS. THE TEST IS DONE WITH INTERNAL LOOPBACK UNLESS EXTERNAL LOOP BACK EXISTS.

7.29 READ/WRITE MODEM REGISTER TESTS (TEST 37)

THIS TEST WRITES THE MODEM REGS OVER THE VARIOUS INTERFACES WITH A PATTERN OF 100. THE MODEM REGS ARE THEN READ AND COMPARED FOR CORRECTNESS.

* N O T E ----- THIS TEST ONLY RUNS IF LOOPBACK CONNECTORS ARE ATTACHED

7.30 TEST OF MEM EXTENSION BITS. (TESTS 38-40)

THESE THREE TESTS CHECK THE ABILITY OF THE DEVICE TO DO TRANSFERS TO UPPER MEMORY IF IT EXISTS. THE TRANSFERS ARE DONE BY TRANSMITTING AND REC. A MESSAGE. (TTL LOOPBACK MODE ONLY). THE THREE TESTS ARE DONE FOR BIT 16, BIT 17 AND BITS 16 AND 17.

* N O T E ----- THIS TEST USES MEMORY ONLY IF IT EXISTS

7.31 TEST FOR TX/RX 257 BYTES (TEST 41)

THIS TEST TRANSMITS A MESSAGE OF 257 BYTES FROM A TRANSMIT BUFFER STARTING WITH AN ODD BYTE BOUNDRY TO A RECEIVE BUFFER STARTING ON AN ODD BYTE BOUNDRY IN DDCMP MODE, POINT TO POINT.

7.32 TEST FOR TX/RX 1 BYTE (TEST 42)

THIS TEST TX'S AND REC'S A 1 BYTE MESSAGE FROM AN ODD TX BUFFER TO AN EVEN RX BUFFER IN MAINT MODE, MULTIPOINT CONTROL STATION.

7.33 POLLING STATE TESTS (TEST 43)

THIS TEST CHECKS THE DEGRADING OF THE POLLING STATES FROM ACTIVE TO INACTIVE TO POTENTIALLY DEAD TO DEAD THE SEQUENCE THAT IS EXECUTED IS AS FOLLOWS:

MASTER CLEAR, MODE DEF(FULL DUP CONTROL STATION), SET POLL DELAY(GSS ADD 37), ESTABLISH TRIB, SET SELECTION TIMER(TSS ADD 36), SET NUMBER OF NO DATA MESSAGES TO INACTIVE TO 10 AND THE NUMBER OF TIME OUTS TO POTENTIALLY DEAD TO 4, ISTRY TRIB, WAIT FOR RUN STATE, READ TSS (ADD 2), CHECK FOR INACTIVE BIT, LOOP UNTIL INACTIVE OR TIME OUT, READ THE SELECTION TIMER(TSS 11), COMPARE IT TO 10, CHANGE MODE TO HALF DUPLEX, WAIT FOR TSS SLOT 2 TO INDICATE POT. DEAD, READ SELECTION TIMER(TSS 16), COMPARE IT TO 4, WAIT FOR

CONTROL OUT INDICATING DEAD TRIB,READ SELECTION TIMER(TSS 16)
COMPARE IT TO 10.

8.0 ERROR INFORMATION

8.1 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 EXAMPLES PROVIDE TYPICAL ERROR REPORTS:

CZDMT DVC FTL ERR 00024 ON UNIT 00 TST 004 SUB 000 PC: 016170
ERROR IN ROM E04 READ = 177777 ; CALCULATED = 017327

FOR ALL OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

```

1082      &
1083      002000      ;      . =2000
1084
1085
1086
1087      000200      DRUN== 200
1088
1089
1090
1091
1092
1093
1094      000001      $LSTIN= 1
1095      000001      $LSTTAG= 1
1096      000000      SVCINS= 0      ; LIST INSTRUCTIONS, SHIFTED RIGHT
1097      000000      SVCTST= 0     ; LIST TEST TAGS, SHIFTED RIGHT
1098      000000      SVCSUB= 0     ; LIST SUBTEST TAGS, SHIFTED RIGHT
1099      000000      SVCGBL= 0     ; LIST GLOBAL TAGS, SHIFTED RIGHT
1100      000000      SVCTAG= 0     ; LIST OTHER TAGS, SHIFTED RIGHT
1101
1102      ;      CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1103      ;      TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1104      ;      SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1105      ;      CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1106
```

CZDMTAD DMP/V-11 + CTNL TST #1 MACY11 3GA(1052) 12-AUG-80 15:53 L 2
CZDMTA.REL 12-AUG-80 15:52 PAGE 25

SEQ 0024

1107

1108
1109
1110
1111
1112
1113
1114
1115
1116 002000
1117 002000 103
1118 002001 132
1119 002002 104
1120 002003 115
1121 002004 124
1122 002005 000
1123 002006 000
1124 002007 000
1125 002010
1126 002010 101
1127 002011
1128 002011 060
1129 002012
1130 002012 000000
1131 002014
1132 002014 003410
1133 002016
1134 002016 031622
1135 002020
1136 002020 000000
1137 002022
1138 002022 002262
1139 002024
1140 002024 000000
1141 002026
1142 002026 036746
1143 002030
1144 002030 000000
1145 002032
1146 002032 000000
1147 002034
1148 002034 000000
1149 002036
1150 002036 000000
1151 002040
1152 002040 002132
1153 002042
1154 002042 000000
1155 002044
1156 002044 000000
1157 002046
1158 002046 000000
1159 002050
1160 002050 003
1161 002051 003
1162 002052
1163 002052 000000

```
.SBTTL PROGRAM HEADER
:++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--

LSNAME::          ;DIAGNOSTIC NAME
      .ASCII /C/
      .ASCII /Z/
      .ASCII /D/
      .ASCII /M/
      .ASCII /T/
      .BYTE 0
      .BYTE 0
      .BYTE 0

LSREV::           ;REVISION LEVEL
      .ASCII /A/

LSDEPO::          ;0
      .ASCII /O/

LSUNIT::          ;NUMBER OF UNITS
      .WORD 0

LSTIML::          ;LONGEST TEST TIME
      .WORD 1800.

LSHPCP::          ;POINTER TO H.W. QUES.
      .WORD LSHARD

LSSPLP::          ;POINTER TO S.W. QUES.
      .WORD 0

LSHPTP::          ;PTR. TO DEF. H.W. PTABLE
      .WORD LSHW

LSSPTP::          ;PTR. TO S.W. PTABLE
      .WORD 0

LSLADP::          ;DIAG. END ADDRESS
      .WORD L$LAST

LSSTA::           ;RESERVED FOR APT STATS
      .WORD 0

LSCO::            ;DIAGNOSTIC TYPE
      .WORD 0

$DTYP::           ;DIAGNOSTIC TYPE
      .WORD 0

LSAPT::           ;APT EXPANSION
      .WORD 0

LSDTP::           ;PTR. TO DISPATCH TABLE
      .WORD L$DISPATCH

L$PRIO::          ;DIAGNOSTIC RUN PRIORITY
      .WORD 0

L$ENVI::          ;FLAGS DESCRIBE HOW IT WAS SETUP
      .WORD 0

L$EXP1::          ;EXPANSION WORD
      .WORD 0

L$MREV::          ;SVC REV AND EDIT #
      .BYTE C$REVISION
      .BYTE C$EDIT

L$EF::            ;DIAG. EVENT FLAGS
      .WORD 0
```

1164	002054	000000			
1165	002056		LS\$PC::	.WORD	0
1166	002056	000000			
1167	002060		LS\$DEVP::	.WORD	0
1168	002060	002560			; POINTER TO DEVICE TYPE LIST
1169	002062		LS\$REPP::	.WORD	LS\$DVTYP
1170	002062	000000			;PTR. TO REPORT CODE
1171	002064		LS\$EXP4::	.WORD	0
1172	002064	000000			
1173	002066		LS\$EXPS::	.WORD	0
1174	002066	000000			
1175	002070		LS\$AUT::	.WORD	0
1176	002070	013672			;PTR. TO ADD UNIT CODE
1177	002072		LS\$DUT::	.WORD	LS\$AU
1178	002072	013666			;PTR. TO DROP UNIT CODE
1179	002074		LS\$LUN::	.WORD	LS\$DU
1180	002074	000000			;LUN FOR EXERCISERS TO FILL
1181	002076		LS\$DESP::	.WORD	0
1182	002076	002602			;POINTER TO DIAG. DESCRIPTION
1183	002100		LS\$LOAD::	.WORD	LS\$DESC
1184	002100	104035			;GENERATE SPECIAL AUTOLOAD EMT
1185	002102		LS\$ETP::	EMT	ES\$LOAD
1186	002102	000000			;POINTER TO ERR TBL
1187	002104		LS\$ICP::	.WORD	0
1188	002104	013206			;PTR. TO INIT CODE
1189	002106		LS\$CCP::	.WORD	LS\$INIT
1190	002106	013662			;PTR. TO CLEAN-UP CODE
1191	002110		LS\$ACP::	.WORD	LS\$CLEAN
1192	002110	013620			;PTR. TO AUTO CODE
1193	002112		LS\$PRT::	.WORD	LS\$AUTO
1194	002112	002122			;PTR. TO PROTECT TABLE
1195	002114		LS\$TEST::	.WORD	LS\$PROT
1196	002114	000000			;TEST NUMBER
1197	002116		LS\$DLY::	.WORD	0
1198	002116	000000			;DELAY COUNT
1199	002120		LS\$HIME::	.WORD	0
1200	002120	000000			;PTR. TO HIGH MEM
1201					
1202	002122		LS\$PROT::	.WORD	-1
1203	002122	177777			
1204	002124	177777			
1205	002126	177777			
1206					

1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214 002130 000053
 1215 002132
 1216 002132 013674
 1217 002134 014054
 1218 002136 014406
 1219 002140 014740
 1220 002142 015272
 1221 002144 015624
 1222 002146 016156
 1223 002150 016510
 1224 002152 016724
 1225 002154 017140
 1226 002156 017324
 1227 002160 020456
 1228 002162 020570
 1229 002164 021000
 1230 002166 021244
 1231 002170 021454
 1232 002172 021706
 1233 002174 022140
 1234 002176 022424
 1235 002200 022532
 1236 002202 022640
 1237 002204 022754
 1238 002206 023502
 1239 002210 024130
 1240 002212 024556
 1241 002214 024672
 1242 002216 024756
 1243 002220 025204
 1244 002222 025432
 1245 002224 025760
 1246 002226 026054
 1247 002230 026174
 1248 002232 026342
 1249 002234 026530
 1250 002236 026646
 1251 002240 026750
 1252 002242 027044
 1253 002244 027456
 1254 002246 027720
 1255 002250 030162
 1256 002252 030424
 1257 002254 030522
 1258 002256 030620
 1259
 1260
 1261
 1262

.SBTTL DISPATCH TABLE

```

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

```

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

CZDMTAD DMP/V-11 + CTNL TST #1 MACY11 30A(1052) 12-AUG-80 15:53 ^{C 3} PAGE 29
CZDMTA.REL 12-AUG-80 15:52 DISPATCH TABLE

SEQ 0028

1263
1264

1265
1266
1267
1268
1269
1270
1271
1272
1273 002260 000014
1274 002262
1275 002262
1276
1277 002262 000000
1278 002264 160170
1279 002266 000300
1280 002270 005000
1281 002272 000003
1282 002274 000056
1283 002276 000000
1284 002300 000000
1285 002302 000004
1286
1287
1288 002304 000004
1289
1290
1291 002306 000000
1292 002310 000002
1293
1294 002312
1295
1296
1297
1298
1299

.SBTTL DEFAULT HARDWARE P-TABLE

:/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
:/ THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
:/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.

.WORD L10001-LSHW/2
LSHW::
DFPTBL::

.WORD 0
.WORD 160170
.WORD 300
.WORD 5000
.WORD 3
.WORD 056
.WORD 000
.WORD 000
.WORD 4

: HARDWARE TYPE
: M8200,4,7 CSR UNIBUS ADDRESS
: M8200,4,7 INTERRUPT VECTOR
: M8200,4,7 INTERRUPT PRIORITY LEVEL - 5
: LINE UNIT = M8203
: SWITCH PACK #1 (REG 11)
: SWITCH PACK #2 (REG 15)
: SWITCH PACK #3 (REG 16)
: H3251&H3252 USED
: 0= LOOPBACK CABLE,2= TEST CONNECTOR
: 4= NONE
: CONTAINS BAUD RATE 4=56K BAUD DEFAULT
: 0=2.4K , 1=4.8K , 2=9.6K , 3=19.2K , 4=56K
: 5=250K , 6=500K , 7=1 MEG BAUD
: DUMMY WORD FOR RUN
: 1=INTEGRAL ;2=EIA;3=V.35;4=422

L10001:

1300
1301
1302
1303
1304
1305
1306
1307 002312 000000
1308 002314
1309 002314
1310
1311
1312 002314
1313
1314
1315
1316
1317
1318

.SBTTL SOFTWARE P-TABLE
:////////////////////
:// THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
:// PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
:////////////////////
.WORD L10002-L\$SW/2
L\$SW::
SFPTBL::

L10002:

1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374

.SBTTL GLOBAL EQUATES SECTION

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

: BIT DEFINITIONS

:
: BIT15== 100000
: BIT14== 40000
: BIT13== 20000
: BIT12== 10000
: BIT11== 4000
: BIT10== 2000
: BIT09== 1000
: BIT08== 400
: BIT07== 200
: BIT06== 100
: BIT05== 40
: BIT04== 20
: BIT03== 10
: BIT02== 4
: BIT01== 2
: BIT00== 1
:
: BIT9== BIT09
: BIT8== BIT08
: BIT7== BIT07
: BIT6== BIT06
: BIT5== BIT05
: BIT4== BIT04
: BIT3== BIT03
: BIT2== BIT02
: BIT1== BIT01
: BIT0== BIT00

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

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

: PRIORITY LEVEL DEFINITIONS

:
: PRI07== 340
: PRI06== 300
: PRI05== 240

1375 000200 PRI04== 200
1376 000140 PRI03== 140
1377 000100 PRI02== 100
1378 000040 PRI01== 40
1379 000000 PRI00== 0

1380
1381 ; OPERATOR FLAG BITS
1382 ;
1383 000004 EVL== 4
1384 000010 LOT== 10
1385 000020 ADR== 20
1386 000040 IPU== 40
1387 000100 ISR== 100
1388 000200 UAM== 200
1389 000400 BOE== 400
1390 001000 PNT== 1000
1391 002000 PRI== 2000
1392 004000 IXE== 4000
1393 010000 IBE== 10000
1394 020000 IER== 20000
1395 040000 LOE== 40000
1396 100000 HOE== 100000

1397
1398
1399
1400
1401 : *****
1402 : * PROGRAM EVENT FLAG DEFINITIONS
1403 : *****

1404
1405
1406
1407
1408
1409 : *****
1410 : * MAINTENANCE REGISTER - BSEL1
1411 : *****
1412 000200 RUN = BIT7
1413 000100 MCLR = BIT6
1414 000020 STEPLU = BIT4
1415 000010 LULOOP = BIT3
1416 000004 ROMO = BIT2
1417 000002 ROMI = BIT1
1418 000001 STEPMP = BIT0

1419
1420
1421 : *****
1422 : OTHER BIT DEFINITIONS
1423 : *****
1424
1425 000200 RQI =200
1426 000020 RDI =020
1427 000200 RDO =200
1428
1429
1430

CZDMTAD DMP/V-11 PCTNL TST #1 MACY11 30A(1052) 12-AUG-80 15:53 H 3 PAGE 34
CZDMTA.REL 12-AUG-80 15:52 GLOBAL EQUATES SECTION

SEQ 0033

1431
1432

```

1433 .SBTTL GLOBAL DATA SECTION
1434
1435 :////////////////////
1436 :/ THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1437 :/ IN MORE THAN ONE TEST.
1438 :////////////////////
1439
1440 :*****
1441 :* MISCELLANEOUS STORAGE
1442 :*****
1443 002314 000000 SAVE4: .WORD 0 ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
1444 002316 000000 SAVE6: .WORD 0
1445 002320 000000 PSTACK: .WORD 0
1446 002322 000000 SUBRPC: .WORD 0
1447 002324 000000 ERROR1: .WORD 0
1448 002326 000000 FRSTIM: .WORD 0
1449 002330 000000 LOGDEV: .WORD 0
1450 002332 000000 IFLAG: .WORD 0
1451 002334 000000 $GDDAT: .WORD 0 ;GOOD AND BAD DATA STORAGE
1452 002336 000000 $BDDAT: .WORD 0
1453 002340 000000 COUNT: .WORD 0
1454 002342 000000 REG: .WORD 0
1455 002344 000000 STARES: .WORD 0 ;INDICATES PASSES
1456 002346 000000 DEVMAP: .WORD 0
1457 002350 000000 DEVPTR: .WORD 0
1458 002352 000000 FRSPAS: .WORD 0
1459 002354 000000 MODINT: .WORD 0 ;MODEM INTERFACE SELECTION
1460 002356 000000 TRIBN: .WORD 0 ;POINTS TO CURRENT TRIP NUMBER.
1461 002360 000040 TRIBMX: .WORD 32. ;MAXIMUM NUMBER OF TRIBS
1462 002362 000000 TRIBH: .WORD 0 ;VALUE OF HIGHEST TRIB USED
1463
1464 002364 000000 ROMN: .WORD 0 ;CURRENT ROM AND USED FOR TEST #
1465 002366 000000 ROMN1: .WORD 0 ;CURRENT ROM NUMBER
1466 002370 000000 WORDT: .WORD 0 ;CURRENT ROM CONTENTS.
1467 002372 177777 CWORD: .WORD -1 ;CURRENT CRC CAL.
1468 002374 000000 ERRWRD: .WORD 0 ;ERROR OCCURRED.
1469 002376 000000 CADDR: .WORD 0 ;CURRENT ROM ADDR.
1470 002400 000000 ERRADD: .WORD 0 ;PC OF ERROR
1471 002402 000000 PERR: .WORD 0 ;PROCEDURE ERROR CHECKED
1472 002404 000000 TSSADD: .WORD 0 ;WORD FOR TSS ADD
1473 002406 003406 TYLST: .WORD 3406
1474 002410 002403 .WORD 2403
1475 002412
1476 002412 000000 TXADD: .WORD 0 ;TX BUFF ADDRESS
1477 002414 000000 RXADD: .WORD 0 ;RX BUFF ADDRESS
1478 002416 000000 RXCC: .WORD 0 ;RX CHAR COUNT
1479 002420 000000 TXCC: .WORD 0 ;TX CHAR COUNT
1480 002422 000000 CODEW: .WORD 0 ;LOCATION FOR ERROR CODES
1481 002424 000000 GENWRD: .WORD 0 ;USED FOR MAINT STATE AND EX MEM
1482 002426 000000 CRCCAL: .WORD 0 ;TEMP FOR CRC
1483 002430 000000 ROMADD: .WORD 0 ;ROM ADDRESS
1484 002432 000000 ROMBK1: .WORD 0
1485 002434 000000 ROMBK2: .WORD 0 ;ROM ADDRESS FOR DMV
1486
1487 :***** CURRENT DEVICE PARAMETERS *****
1488 002436 BSELO:
  
```

```
1489 002436          SEL0:
1490 002436 160170    MPCSR: .WORD 160170          ;POINTER TO M8200,4,7 CSR'S
1491 002440 160171    BSEL1: .WORD 160171          ;POINTER TO BSEL1
1492 002442          SEL2:
1493 002442 160172    BSEL2: .WORD 160172
1494 002444 160173    BSEL3: .WORD 160173
1495 002446          BSEL4:
1496 002446 160174    SEL4: .WORD 160174          ;POINTER TO SEL4
1497 002450 160175    BSEL5: .WORD 160175
1498 002452          BSEL6:
1499 002452 160176    SEL6: .WORD 160176
1500 002454 160177    ESEL7: .WORD 160177
1501 002456          KMRVEC:
1502 002456 000300    MPIVEC: .WORD 300          ;M8200,4,7 INPUT INTERRUPT VECTOR
1503 002460          KMTVEC:
1504 002460 000304    MPOVEC: .WORD 304          ;M8200,4,7 OUTPUT INTERRUPT VECTOR
1505 002462 000000    SPEEDM: .WORD 0          ;SPEED OF LINE UNIT
1506 002464          KMRLVL:
1507 002464          KMTLVL:
1508 002464 000240    MPRIOR: .WORD 240          ;M8200,4,7 DEVICE PRIORITY
1509 002466 000000    OPTYP: .WORD 0          ;OPTION TYPE
1510 002470 000000    IFTYP: .WORD 0          ;INTERFACE TYPE
1511 002472 000000    TSTCON: .WORD 0          ;TEST CONNECTOR INDICATOR
1512 002474 000000    RETADR: .WORD 0          ;SUBR ERROR RETURN ADDRESS
1513 002476 000000    REDBYT: .WORD 0          ;LO BYTE CONTAINS BYTE READ FROM LU REG
1514 002500 000000    WRIBYT: .WORD 0          ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
1515 002502 000000    AXNUM: .WORD 0          ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
1516 002504 000000    DISILO: .WORD 0          ;CONTAINS CURRENT STATE OF DISSI IN BITS
1517
1518          ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
1519 002506 000      REDDAT: .BYTE 0
1520 002507 000          .BYTE 0
1521 002510 000          .BYTE 0
1522 002511 000          .BYTE 0
1523 002512 000          .BYTE 0
1524 002513 000          .BYTE 0
1525 002514 000          .BYTE 0
1526 002515 000          .BYTE 0
1527
1528          ;***** GENERAL PURPOSE SCRATCH STORAGE *****
1529 002516 000000    REG0: .WORD 0
1530 002520 000000    REG1: .WORD 0
1531 002522 000000    REG2: .WORD 0
1532 002524 000000    REG3: .WORD 0
1533 002526 000000    REG4: .WORD 0
1534 002530 000000    REG5: .WORD 0
1535 002532 000000    REG6: .WORD 0
1536 002534 000000    REG7: .WORD 0
1537
1538          ;***** SCRATCH STORAGE FOR MESSAGE REPORTING *****
1539 002536 000000    $TMP0: .WORD 0
1540 002540 000000    TMP0: .WORD 0
1541 002542 000000    TMP1: .WORD 0
1542 002544 000000    TMP2: .WORD 0
1543 002546 000000    TMP3: .WORD 0
1544 002550 000000    TMP4: .WORD 0
```

1545	002552	000000	TMP5:	.WORD	0
1546	002554	000000	TMP6:	.WORD	0
1547	002556	000000	TMP7:	.WORD	0
1548					

1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588

002560				
002560	046504	026520	030461	
002566	047440	020122	046504	
002574	026526	030461	000	
	002602			
002602				
002602	046504	020120	051117	
002610	042040	053115	030455	
002616	020061	052506	041516	
002624	044524	047117	046101	
002632	042040	040511	027107	
002640	000			
	002642			

```
.SBTTL GLOBAL TEXT SECTION
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
: THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: MORE THAN ONE TEST.
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:*****
: * NAMES OF DEVICES SUPPORTED BY PROGRAM
:*****
LSDVTYP::
      .ASCIZ /DMP-11 OR DMV-11/
      .EVEN
LSDESC::
      .ASCIZ /DMP OR DMV-11 FUNCTIONAL DIAG./
      .EVEN
:
: FORMAT STATEMENTS USED IN PRINT CALLS
:
```

1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642
 1643

.SBTTL GLOBAL SUBROUTINES

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

..+
 : FUNCTIONAL DESCRIPTION: WRDO.. WAITS FOR READY OUT
 : FIRST SAVE THE CALLING ADDRESS
 : IN ERRADD. THEN SEE IF TIME OUT OCCURED
 : IF TIME OUT EXIT ROUTINE..ELSE SEE IF
 : READY OUT SET IF READY OUT SET EXIT
 : ROUTINE. IF NOT THEN WAIT A WHILE
 : THEN SEE IF READY IN SET. IF READY IN
 : IS SET REPORT ERROR AND EXIT ROUTINE.
 : IF NOT READY IN THEN GO BACK AND CHECK
 : FOR TIME OUT.
 : NOTE: CAN BE ENTERED AT WRDO1 IF CALLING
 : ADDRESS FROM R5 DOES NOT NEED TO BE SAVED.

: INPUTS: R5=ADDRESS FROM WHERE ROUTINE WAS CALLED

: OUTPUTS: ERRWRD= -1 IF ERROR OCCURED IN ROUTINE.

: SUBORDINATE ROUTINES USED:
 : TOUT - TIME OUT ROUTINE
 : WAIT50 - SHORT DELAY ROUTINE

: CALLING SEQUENCE:
 : JSR R5,WRDO

..--

```

WRDO:  MOV    R5,ERRADD      ;STORE ERROR ADD. AWAY
WRDO1: JSR    R5,TOUT       ; GO TO TIME OUT ROUTINE
      TST    ERRWRD        ;CHECK IF ERROR
      BMI    WRDOE         ;EXIT NOW
      BIT    #RDO,@BSEL2   ;RDO SET?
      BNE    WRDOE         ;EXIT IF RDO IS SET
      JSR    PC,WAIT50     ;ELSE DELAY A LITTLE
      BIT    #RDI,@BSEL2   ;THEN SEE IF RDI IS SET
      BEQ    WRDO1        ;IF NOT THEN GO BACK TO START
      ; ERROR -UNEXPECTED RDI SET

      TRAP   C$ERDF
      .WORD  1
      .WORD  MEF14
      .WORD  ERR26
      DEC   ERRWRD
      CLRB  @BSEL2        ;CLEAR RDO
WRDOE: RTS    R5         ;EXIT
  
```

```

002642 010537 002400
002646 004537 005016
002652 005737 002374
002656 100422
002660 032777 000200 177554
002666 001016
002670 004737 004172
002674 032777 000020 177540
002702 001761
002704 104455
002706 000001
002710 011271
002712 007536
002714 005337 002374
002720 105077 177516
002724 000205
  
```


1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689

```

:++
: FUNCTIONAL DESCRIPTION: WRDI - WAIT FOR READY IN
: THIS ROUTINE FIRST SAVES THE CALLING ADDRESS
: IN ERRADD, UNLESS ENTERED AT WRDI1.
: THEN CHECK FOR TIME OUT IF TIME OUT REPORT
: ERROR AND EXIT. IF NOT TIME OUT CHECK FOR
: READY IN. IF READY IN EXIT IF NOT READY IN
: DELAY A LITTLE AND CHECK FOR READY OUT. IF
: READY OUT REPORT ERROR AND EXIT ROUTINE.
: IF NOT READY OUT GO BACK AND CHECK FOR TIME OUT.

: INPUTS:      R5= CALLING ADDRESS
: OUTPUTS:     ERRWRD= -1 IF ERROR OCCURED IN ROUTINE
: SUBORDINATE ROUTINES USED:
:             TOUT- TIME OUT
:             WAIT50- DELAY A LITTLE
: CALLING SEQUENCE:
:             JSR      R5,WRDI      ;OR
:             JSR      R5,WRDI1
:--
  
```

```

1666 002726 010537 002400 WRDI:  MOV      R5,ERRADD      ;STORE AWAY ERROR ADD.
1667 002732 004537 005016 WRDI1: JSR      R5,TOUT        ;GO TO TIME OUT
1668 002736 005737 002374      TST      ERRWRD          ;IF ERROR EXIT
1669 002742 100422          BMI      WRDIE           ;
1670 002744 032777 000020 177470 1$:  BIT      #20,@BSEL2      ;RDI SET?
1671 002752 001016          BNE      WRDIE           ;YES-EXIT
1672 002754 004737 004172      JSR      PC,WAIT50       ;SHORT DELAY
1673 002760 032777 000200 177454  BIT      #200,@BSEL2    ;RDYO SET?
1674 002766 001761          BEQ      WRDI1          ;NO-LOOP.
1675
1676                                ;RDO INSTEAD OF RDI
1677
1678 002770 104455          TRAP     C$ERDF
1679 002772 000002          .WORD   2
1680 002774 011336          .WORD   MEF15
1681 002776 007536          .WORD   ERR26
1682 003000 005337 002374      DEC      ERRWRD          ;SET ERROR OCCURRED
1683 003004 105077 177432      CLRB    @BSFL2          ;CLEAR RDYO
1684 003010 000205      WRDIE: RTS      R5      ;EXIT.
  
```

1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1740
 1741
 1742
 1743
 1744
 1745

```

  : **
  : FUNCTIONAL DESCRIPTION: WFPE - WAIT FOR PROCEDURE ERROR
  : FIRST SAVE CALLING ADDRESS IN ERRADD.
  : THEN WAIT FOR READY OUT, IF ERROR FROM
  : WRDO ROUTINE EXIT THIS ROUTINE. ELSE
  : GET CONTROL KEY FROM BSEL2 IF NOT CONTROL
  : OUT REPORT ERROR AND EXIT. ELSE CHECK THAT
  : CONTROL OUT CODE IS SAME AS IN PERR. IF
  : EQUAL THEN EXIT ELSE REPORT ERROR AND EXIT.
  :
  : INPUTS:      R5= CALLING ADDRESS
  :              PERR = PROCEDURE ERROR EXPECTED.
  :
  : OUTPUTS:     ERRWRD= -1 IF ERROR OCCURED IN ROUTINE
  :
  : SUBORDINATE ROUTINES USED:
  :              WRDO1 - WAIT FOR READY OUT
  :
  : CALLING SEQUENCE:
  :              JSR      R5,WFPE
  : --
  
```

```

WFPE:  MOV      R5,ERRADD      ;STORE OFF ERROR ADDRESS
      JSR      R5,WRDO1      ;WAIT FOR READY OUT
      TST     ERRWRD
      BMI     20$           ; IF ERROR OCCURRED IN SR
      ; EXIT THIS SR.
      MOVB    @BSEL2,$BDDAT
      BIC     #'C<7>,$BDDAT ; STRIP DATA TO CONTROL KEY
      CMPB    #01,$BDDAT
      BEQ     10$           ; GO TO 10 IF CONTROL OUT
      ; ELSE REPORT ERROR
      MOV     #01,$GDDAT    ;SET GOOD DATA TO 01
      MOV     #M18F,CODEW   ;SET UP CODE WORD
      ;ERROR NOT CONTROL OUT
      TRAP    C$ERDF
      .WORD   3
      .WORD   EROIC
      .WORD   ERR27
      DEC     ERRWRD        ;SET ERROR OCCURRED
      BR      20$          ;AND EXIT SUBROUTINE
10$:  MOVB    @BSEL6,$BDDAT  ;MOVE ERROR CODE TO BDDAT
      CMPB    PERR,$BDDAT  ;IS IT WHAT IT SHOULD BE
      BEQ     20$          ; IF SO GO TO 20
      MOV     PERR,$GDDAT  ;PUT EXPECTED IN GOOD DATA
      MOV     #M13F,CODEW  ;SET UP ERROR WORD
      ;ERROR BAD ERROR CODE RETURNED
      TRAP    C$ERDF
      .WORD   4
      .WORD   EROIC
      .WORD   ERR27
      DEC     ERRWRD        ;SET ERROR INDICATOR
20$:  RTS      R5           ;RETURN TO CALLER
  
```

1746
 1747
 1748
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1780

```

  **
  : FUNCTIONAL DESCRIPTION: CONTIN - CONTROL IN ROUTINE
  : THIS ROUTINE SAVES THE CALLING ADDRESS IN R5.
  : THEN SETS RQI AND WAITS FOR RDI TO BE RETURNED BY
  : THE DMP/V-11. IF WRDI REPORTS ERROR EXIT TEST. ELSE
  : LOAD BSEL 3 WITH TRIB NUMBER FROM TRIBN,CLEAR THE RQI
  : BIT,MOV DATA FROM R4 TO SEL4,DATA FROM R3 TO SEL6, AND
  : THEN ISSURE CONTROL IN AND EXIT ROUTINE.
  :
  : INPUTS: R4 = SEL4 DATA
  : R3 = SEL6 DATA
  : TRIBN = TRIBUTARY NUMBER.
  : R5 = CALLING ADDRESS
  :
  : OUTPUTS: ERRWRD = -1 IF ERROR REPORTED IN THIS OR ANY SUBODINATE
  : SUB ROUTINE.
  : SUBORDINATE ROUTINES USED:
  : WRDI1 - WAIT FOR READY IN.
  : CALLING SEQUENCE:
  : JSR R5,CONTIN
  :--
  
```

```

CONTIN: MOV R5,ERRADD ;SET UP ERROR ADDRESS
        BIS #RQI,@BSELO ;SET REQUEST
        JSR R5,WRDI1 ;GO WAIT FOR RDI
        TST ERRWRD
        BMI 43$ ;EXIT IF ERROR OCCURRED
        MOVB TRIBN,@BSEL3 ;SET TRIBN
        BIC #RQI,@BSELO ;CLEAR REQUEST
        MOV R4,@BSEL4 ;SET DATA
        MOV R3,@BSEL6 ;SET REQUEST TYPE
        MOVB #01,@BSEL2 ;DO CONTROL IN
43$: RTS R5 ;RETURN TO CALLER
  
```

```

003156 010537 002400
003162 052777 000200 177246
003170 004537 002732
003174 005737 002374
003200 100415
003202 113777 002356 177234
003210 042777 000200 177220
003216 010477 177224
003222 010377 177224
003226 112777 000001 177206
003234 000205
  
```

1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833

```

**
FUNCTIONAL DESCRIPTION:      GETOUT - GET OUTPUT CODE

THIS SUB-ROUTINE WAITS FOR RDO(REPORTS ERROR IF
RDI OR TIME OUT);CHECKS THAT OUTPUT COMMAND TYPE
IS THE SAME AS THE VALUE IN $GDDAT(REPORTS ERROR
IF NOT);THEN CHECKS THAT TRIB NUMBER IN BSEL3 IS
EQUAL TO THE VALUE IN TRIBN(REPORTS ERROR IF NOT
THEN RETURNS TO CALLER.

INPUTS:      $GDDAT = OUTPUT COMMAND TYPE EXPECTED
             TRIBN  = TRIBUTARY ADDRESS EXPECTED
             R5     = ADDRESS OF CALLING ROUTINE

OUTPUTS:     ERRWRD = -1 IF ERROR OCCURED

SUBORDINATE ROUTINES USED:
             WRD01 - WAIT FOR READY OUT

CALLING SEQUENCE:
             JSR   R5,GETOUT
--

```

```

GETOUT: MOV     R5,ERRADD      ;STORE OFF ERROR ADD.
        JSR     R5,WRD01     ; GO WAIT FOR READY OUT
        TST     ERRWRD
        BMI     20$         ;EXIT IF ERROR OCCURRED
        MOVB    @BSEL2,$BDDAT ;GET COMMAND TYPE TO BDDAT
        BIC     #'C<7>,$BDDAT ;STRIP IT TO JUST COMMAND TYPE
        CMP     $BDDAT,$GDDAT ;IS IT THE RIGHT VALUE??
        BEQ     10$         ;IF YES GO TO 10
                           ;ELSE REPORT ERROR
        MOV     #M28F,COEW
        TRAP    C$ERDF
        .WORD   5
        .WORD   EROIC
        .WORD   ERR27
        DEC     ERRWRD
        BR      20$         ;AND EXIT ON ERROR

10$:   MOV     TRIBN,$GDDAT   ;SET UP GDDAT FOR GOOD TRIBN
        MOVB    @BSEL3,$BDDAT ;GET TRIB NUMBER RETURNED
        CMP     $BDDAT,$GDDAT ;ARE THEY THE SAME???
        BEQ     20$         ;IF YES GO TO 20
                           ;ELSE REPORT ERROR

        TRAP    C$ERDF
        .WORD   6
        .WORD   MEF18A
        .WORD   ERR18
        DEC     ERRWRD
        RTS     R5         ;RETURN TO CALLER
20$:

```

```

003236 010537 002400
003242 004537 002646
003246 005737 002374
003252 100444
003254 117737 177162 002336
003262 042737 177770 002336
003270 023737 002336 002334
003276 001412
003300 012737 011257 002422
003306 104455
003310 000005
003312 010756
003314 007574
003316 005337 002374
003322 000420
003324 013737 002356 002334 10$:
003332 117737 177106 002336
003340 023737 002336 002334
003346 001406
003350 104455
003352 000006
003354 011512
003356 007116
003360 005337 002374
003364 000205
20$:

```

1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849 003366 010537 002400
1850 003372 017737 177054 002336
1851 003400 042737 177600 002336
1852 003406 023737 002336 002334
1853 003414 001411
1854
1855 003416 012737 011217 002422
1856 003424 104455
1857 003426 000007
1858 003430 010756
1859 003432 007574
1860 003434 005337 002374
1861 003440 000205
1862

..**
FUNCTIONAL DESCRIPTION: GETRKY - GET RETURN KEY VALUE
THIS ROUTINE GETS THE VALUE OF THE RETURN KEY
FROM BSEL6 AND COMPARES IT TO THE VALUE IN
\$GDDAT. IF EQUAL EXIT IF NOT EQUAL REPORT
ERROR AND EXIT.
: INPUTS: R5 = ADDRESS OF CALLER
\$GDDAT = VALUE OF EXPECTED RETURN KEY
: OUTPUTS: ERRWRD = -1 IF ERROR OCCURS
: CALLING SEQUENCE:
JSR R5,GETRKY
..--
GETRKY: MOV R5,ERRADD ; STORE OFF ERROR ADDRESS
MOV @BSEL6,\$BDDAT ; GET RETURN KEY FROM BSEL6
BIC #^C<177>,\$BDDAT ; STRIP TO VALID BITS
CMP \$BDDAT,\$GDDAT ; ARE THE VALUES EQUAL
BEQ 10\$; IF YES GO TO 10
; ELSE ERROR.....
MOV #M12F,CODEW ; SET UP CODE WORD
TRAP C\$ERDF
.WORD 7
.WORD EROIC
.WORD ERR27
DEC ERRWRD ; SET ERROR OCCURRED
10\$: RTS R5 ; RETURN TO CALLER

1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892

```
..**  
: FUNCTIONAL DESCRIPTION: GETDAT - GET DATA CODE  
: THIS ROUTINE GETS THE DATA CODE FROM BSEL4  
: AND COMPARES IIT TO THE VALUE IN $GDDAT  
: IF EQUAL EXIT ELSE REPORT ERROR AND EXIT  
:  
: INPUTS: R5 = ADDRESS OF CALLER  
: $GDDAT = VALUE OF EXPECTED DATA  
:  
: OUTPUTS: ERRWRD = -1 IF ERROR OCCURED  
: CALLING SEQUENCE:  
: JSR R5,GETDAT  
:--
```

```
003442 010537 002400 GETDAT: MOV R5,ERRADD ;STORE OFF ERROR ADDRESS  
003446 017737 176774 002336 MOV @BSEL4,$BDDAT ;GET DATA  
003454 023737 002336 002334 CMP $BDDAT,$GDDAT ;COMPARE GOOD AND BAD  
003462 001411 BEQ 10$ ;IF OK GO TO 10  
;ELSE ERROR  
;ERROR BAD DATA CODE  
003464 012737 011264 002422 MOV #M30F,CODEW  
003472 104455 TRAP C$ERDF  
003474 000010 .WORD 8  
003476 010756 .WORD EROIC  
003500 007574 .WORD ERR27  
003502 005337 002374 DEC ERRWRD ;SET ERROR OCCURRED  
003506 000205 10$: RTS R5 ;RETURN TO CALLER
```

1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907 003510 010537 002400
1908 003514 017737 176732 002336
1909 003522 042737 140000 002336
1910 003530 023737 002336 002334
1911 003536 001406
1912
1913
1914
1915 003540 104455
1916 003542 000011
1917 003544 011403
1918 003546 007116
1919 003550 005337 002374
1920 003554 000205
1921
1922

```
..**  
..FUNCTIONAL DESCRIPTION: GETCC - GET CHARACTER COUNT  
..THIS ROUTINE GETS THE CHAR. COUNT FROM  
..BSEL6 AND BSEL7 AND THEN COMPARES IT TO  
..THE VALUE IN $GDDAT  
..IF EQUAL EXIT ELSE REPORT ERROR AND EXIT  
..  
..INPUTS: R5 = ADDRESS OF CALLER  
.. $GDDAT = VALUE OF EXPECTED DATA  
..  
..OUTPUTS: ERRWRD = -1 IF ERROR OCCURED  
.. CALLING SEQUENCE:  
.. JSR R5,GETCC  
..--  
GETCC: MOV R5,ERRADD ;STORE OFF RETURN ADDRESS  
MOV @BSEL6,$BDDAT ;GET BSEL6 AND 7 TO BDDAT  
BIC #140000,$BDDAT ;STRIP TO CHAR COUNT  
CMP $BDDAT,$GDDAT ;COMPARE  
BEQ 10$ ;IF OK GO TO 10  
..  
;ESLE REPORT ERROR BAD CHAR COUNT  
TRAP C$ERDF  
.WORD 9  
.WORD MEF16A  
.WORD ERR18  
DEC ERRWRD ;SET ERROR OCCURRED  
10$: RTS R5 ;RETURN TO CALLER
```


1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978

003556 010537 002400
003562 017737 176660 002336
003570 023737 002336 002334
003576 001406
003600 104455
003602 000012
003604 011446
003606 007116
003610 005337 002374
003614 000205

003616 010537 002400
003622 017737 176624 002336
003630 042737 177400 002336
003636 023737 002336 002334
003644 001411
003646 012737 011250 002422
003654 104455
003656 000013
003660 010756
003662 007574
003664 005337 002374
003670 000205

FUNCTIONAL DESCRIPTION: GETBA - GET BUFFER ADDRESS
THIS ROUTINE GETS THE BUFFER ADDRESS FROM
SEL4 AND THEN COMPARES IT TO
THE VALUE IN \$GDDAT
IF EQUAL EXIT ELSE REPORT ERROR AND EXIT
:
INPUTS: R5 = ADDRESS OF CALLER
\$GDDAT = VALUE OF EXPECTED BUFFER ADDRESS
:
OUTPUTS: ERRWRD = -1 IF ERROR OCCURED
CALLING SEQUENCE:
JSR R5,GETBA

GETBA: MOV R5,ERRADD ;STORE OFF ERROR ADDRESS
MOV @BSEL4,\$BDDAT ;GET ADDRESS OUTPUT
CMP \$BDDAT,\$GDDAT ;ARE THEY EQUAL
BEQ 10\$;IF SO GO TO 10
;ELSE ERROR
TRAP C\$ERDF
.WORD 10
.WORD MEF17A
.WORD ERR18
10\$: DEC ERRWRD ;SET ERROR OCCURRED
RTS R5 ;RETURN TO CALLER

FUNCTIONAL DESCRIPTION: GETOC - GET OUTPUT CODE
THIS ROUTINE GETS THE OUTPUT CODE FROM
BSEL6 AND THEN COMPARES IT TO
THE VALUE IN \$GDDAT
IF EQUAL EXIT ELSE REPORT ERROR AND EXIT
:
INPUTS: R5 = ADDRESS OF CALLER
\$GDDAT = VALUE OF EXPECTED OUTPUT CODE
:
OUTPUTS: ERRWRD = -1 IF ERROR OCCURED
CALLING SEQUENCE:
JSR R5,GETOC

GETOC: MOV R5,ERRADD ;STORE OFF ERROR ADDRESS
MOV @BSEL6,\$BDDAT ;GET OUTPUT FROM BSEL6
BIC #^C<377>,\$BDDAT ;STRIP TO VALID BITS
CMP \$BDDAT,\$GDDAT ;ARE THEY EQUAL
BEQ 10\$;IF SO GO TO 10
;ELSE ERROR
MOV #M18F,CODEW
TRAP C\$ERDF
.WORD 11
.WORD EROIC
.WORD ERR27
10\$: DEC ERRWRD ;SET ERROR OCCURRED
RTS R5 ;RETURN TO CALLER

1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034

003672
003672 004537 002642
003676 005737 002374
003702 100524
003704 117737 176532 002336
003712 042737 177770 002336
003720 122737 000000 002336
003726 001450
003730 122737 000001 002336
003736 001012
003740 012737 000302 002334
003746 117737 176500 002336
003754 123737 002334 002336
003762 001412
003764 012737 011232 002422
003772 104455
003774 000014
003776 010756
004000 007574
004002 005337 002374
004006 000462
004010
004010 005237 177572
004014 012737 004144 000004
004022 005737 120000
004026 005037 177572
004032 104455
004034 000015
004036 000000
004040 007224
004042 005337 002374
004046 000442

FUNCTIONAL DESCRIPTION: MEMEX - MEMEORY EXTENSTION CODE
THIS ROUTINE CHECKS THAT IS USED WITH THE
MEMORY EXTENTION TESTS. THE ROUTINE FIRST
CHECKS FOR A CONTROL OUT.. IF THE CONTROL
OUT IS A REC COMPLETE IT COMPARES THE FIRST
DATA WORD ON THE EXTENTIONS PAGE IF GOOD
THEN EXIT IF BAD REPORT ERROR AND EXIT.
IF CONTROL OUT IS NON EXISTENT MEMORY THEN
CHECK TO BE SURE MEMORY IS NON-EXISITENT
IF MEMORY EXIST THEN PRINT ERROR AND
EXIT ROUTINE.
INPUTS: R5 = CALLING ADDRESS
OUTPUTS: ERRWRD = -1 IF ERROR OCCURED
SUBORDINATE ROUTINES USED:
WRDO = WAIT FOR READY OUT
CALLING SEQUENCE:
JSR R4, MEMEX
--

MEMEX:
EXMEM: JSR R5, WRDO ;WAIT FOR READY OUT
TST ERRWRD
BMI EXMEMX
MOVB @BSEL2, \$BDDAT ;
BIC #^C<7>, \$BDDAT ;STRIP TO TYPE CODE
CMPB #0, \$BDDAT ;IS IT REC COMP
BEQ EXMEMA ; IF YES THEN GO TO A
CMPB #1, \$BDDAT ;IF NOT IS IT CONTROL OUT
BNE EXMEMB ;IF NOT GO TO B
MOV #302, \$GDDAT
MOVB @BSEL6, \$BDDAT
CMPB \$GDDAT, \$BDDAT ;IS IT NON EXISTENT MEM?
BEQ EXMEMC ;IF YES GO TO C
EXMEMB: MOV #M13F, CODEW
TRAP C\$ERDF
.WORD 12
.WORD EROIC
.WORD ERR27
DEC ERRWRD ;SET ERROR FLAG
BR EXMEMX ;PRINT ERROR AND FXIT
EXMEMC: ;IF ERROR IS NON EXISTENT
INC 177572 ;INCREMENT MM
MOV #METB, 4 ;SET UP TRAP
TST @#120000 ;TEST THE NON EXISTENT
CLR 177572 ;CLEAR MM
TRAP C\$ERDF
.WORD 13
.WORD 0
.WORD ERR20
DEC ERRWRD
BR EXMEMX ;PRINT ERROR AND EXIT TEST

```
2035
2036                ; GET HERE IF BUFFER RETURNED OK
2037
2038 004050 005237 177572      EXMEMA: INC 177572          ;ENABLE MM
2039 004054 012737 004130 000004  MOV #METC,4        ;SET UP TRAP
2040 004062 013737 032544 002334  MOV MR1+2,$GDDAT  ;GET FIRST WORD FROM NEW PAGE
2041 004070 013737 120000 002336  MOV 120000,$BDDAT ;AND FIRST RX WORD
2042 004076 005037 177572      CLR 177572          ;DISABLE MM
2043 004102 023737 002334 002336  CMP $GDDAT,$BDDAT ;COMPARE DATA
2044 004110 001421              BEQ EXMEMX          ;EXIT IF GOOD
2045 004112 104455              TRAP C$ERDF
2046 004114 000016              .WORD 14
2047 004116 000000              .WORD 0
2048 004120 007256              .WORD ERR21
2049 004122 005337 002374      DEC ERRWRD
2050 004126 000412              BR EXMEMX          ;AND EXIT
2051 004130 005037 177572      METC: CLR 177572   ;DISABLE MM
2052 004134 104455              TRAP C$ERDF
2053 004136 000017              .WORD 15
2054 004140 000000              .WORD 0
2055 004142 007340              .WORD ERR22
2056 004144 005037 177572      METB: CLR 177572   ;DISABLE MM
2057 004150 062716 000004      META: ADD #4,(SP)
2058 004154 013737 002314 000004  EXMEMX: MOV SAVE4,4
2059 004162 013737 002316 000006  MOV SAVE6,6        ;RESTORE TRAPS
2060 004170 000205              RTS R5
2061
```

2062
 2063
 2064
 2065
 2066
 2067
 2068
 2069
 2070
 2071
 2072 004172 010146
 2073 004174 012701 000310
 2074 004200 005737 002466
 2075 004204 001402
 2076 004206 062701 000310
 2077 004212 005301
 2078 004214 001376
 2079 004216 012601
 2080 004220 000207
 2081
 2082
 2083
 2084
 2085
 2086
 2087
 2088
 2089
 2090
 2091
 2092
 2093
 2094
 2095 004222 005077 176210
 2096 004226 113777 002377 176206
 2097 004234 004537 004274
 2098 004240 121053
 2099 004242 042737 000377 004262
 2100 004250 153737 002376 004262
 2101 004256 004537 004274
 2102 004262 100000
 2103 004264 052777 002000 176144
 2104 004272 000207

```

:++
: FUNCTIONAL DESCRIPTION:   WAIT50 - WAIT 50 MICOR SES
:                           THIS ROUTINE COUNTS DOWN R1 FROM 200 TO 0
:                           IF DMP AND FROM 400 TO 0 IF DMV. THIS
:                           IS USED AS A DELAY ROUTINE
:
: CALLING SEQUENCE:
:   JSR      PC,WAIT50
:--
  
```

```

WAIT50: MOV      R1,-(SP)      ;SAVE R1
        MOV      #200.,R1    ;INIT COUNTER
        TST      OPTYP
        BEQ      3$         ;IF DMP GO TO 3
        ADD      #200.,R1    ;ELSE DOUBLE UP TIMER FOR DMV
3$:     DEC      R1          ;DECREMENT COUNTER
        BNE      3$         ;BR IF NOT DONE YET
        MOV      (SP)+,R1    ;RESTORE R1
        RTS      PC         ;RETURN
  
```

```

:++
: FUNCTIONAL DESCRIPTION:   GWORD - GET WORD
:                           THIS ROUTINE READS A WORD FROM THE M8207 ROM.
:
: INPUTS:      CADDR = ADDRESS TO BE READ
:
: OUTPUTS:     SEL6 = DATA READ
:
: SUBORDINATE ROUTINES USED:
:   ROMCLK - ROUTINE TO ISSUE CLOCKS TO ROM CIRCUIT
:
: CALLING SEQUENCE:
:   JSR      PC,GWORD
:--
  
```

```

GWORD: CLR      @SELO        ;INIT
        MOVB    CADDR+1,@SEL2 ;NOW HIGH BYTE OF ADDRESS
        JSR     R5,ROMCLK
        .WORD   121053
        B!C    #377,1$      ;MOV IBUS* 2 TO OBUS* 13
        BISB   CADDR,1$     ;STRIP ADDR FLIED.
        JSR     R5,ROMCLK    ;ADD IN IMM ADDR.
        .WORD   100000
        BIS    #2000,@SELO  ;GO DO BRANCH.
        RTS    PC          ;BRANCH EXT PUTS ADDR. IN PCREG.
        ;SET READ ENABLE.
        ;EXIT.
  
```

2105
 2106
 2107
 2108
 2109
 2110
 2111
 2112
 2113
 2114
 2115
 2116
 2117
 2118
 2119 004274
 2120 004274 152777 000002 176136
 2121 004302 012577 176144
 2122 004306 152777 000003 176124
 2123 004314 142777 000007 176116
 2124 004322 000205
 2125
 2126
 2127
 2128
 2129
 2130
 2131
 2132
 2133
 2134
 2135
 2136
 2137
 2138
 2139
 2140
 2141 004324 010146
 2142 004326 010246
 2143 004330 012702 000020
 2144 004334 000241
 2145 004336 006037 002372
 2146 004342 006037 002370
 2147 004346 102011
 2148 004350 012701 102010
 2149 004354 043701 002372
 2150 004360 042737 102010 002372
 2151 004366 050137 002372
 2152 004372 005302
 2153 004374 003357
 2154 004376 012602
 2155 004400 012601
 2156 004402 000207
 2157

```

:++
: FUNCTIONAL DESCRIPTION: ROMCLK - ROM CLOCK ROUTINE
: THIS ROUTINE ISSUES A SINGE STEP TO THE
: M8207.
: INPUTS: R5 - POINTS TO INSTRUCTION TO BE STEPPED
: RETURN: RETURN IS TO WORD FOLLWOING INSTRUCTION
: CALLING SEQUENCE:
: JSR R5,ROMCLK
: .WORD INSTR ;INSTURCITON TO EXECUTE
:--
  
```

```

ROMCLK:
BISB #2,@BSEL1 ;SET ROMI
MOV (R5)+,@BSEL6 ;SET INSTRUCTION.
BISB #3,@BSEL1 ;CLOCK INSTR.
BICB #7,@BSEL1 ;CLEAR.
RTS R5
  
```

```

:++
: FUNCTIONAL DESCRIPTION: CRCR - CRC CALUCULATE ROUTINE
: THIS ROUTINE TAKES 16 BITS OF DATA FROM WORDT
: AND CONVERTS THEM INTO PART OF THE SERIAL STREAM
: THAT IS BEING USED TO CALCULATE A CRC-CCITT WORD.
: INPUTS: WORDT - WORD TO CALUCLATE ON
: IMPLICIT INPUTS:
: WORDT - MUST BE A -1 FIRST TIME CALLED
: OUTPUTS: CWORD - 16 BIT CALUCLATED WORD
: CALLING SEQUENCE:
: JSR PC,CRCR
:--
  
```

```

CRCR:
MOV R1,-(SP)
MOV R2,-(SP)
MOV #16.,R2
10$:
CLC
ROR CWORD
ROR WORDT
BVC 20$
MOV #102010,R1
BIC CWORD,R1
BIC #102010,CWORD
BIS R1,CWORD
20$:
DEC R2
BGT 10$
MOV (SP)+,R2
MOV (SP)+,R1
RTS PC
  
```

2158
 2159
 2160 : **
 2161 : FUNCTIONAL DESCRIPTION: MINITR - MASTER CLEAR ROUTINE
 2162 : THIS ROUTINE ISSUES A MASTER CLEAR TO THE DEVICE
 2163 : IF OPTION IS AN 8206 IT ALSO SETS THE RUN BIT.
 2164 :
 2165 : SUBORDINATE ROUTINES USED:
 2166 : WAIT50 - SHORT DELAY ROUTINE
 2167 : CALLING SEQUENCE:
 2168 : JSR PC,MINITR
 2169 :--

2170 004404 112777 000100 176026 MINITR: MOVB #100,@BSEL1 ;SET MASTER CLEAR.
 2171 004412 022737 000004 002466 CMP #04,OPTYP ;IS THIS 8206
 2172 004420 001003 BNE MIN1R ;BRANCH IF NOT
 2173 004422 112777 000200 176010 MOVB #200,@BSEL1 ;SET RUN
 2174 004430 000240 MIN1R: NOP
 2175 004432 004737 004172 JSR PC,WAIT50 ;SHORT DELAY.
 2176 004436 000207 RTS PC ;RETURN.

2177 : **
 2178 : FUNCTIONAL DESCRIPTION: MINITS - MASTER CLEAR AND INIT
 2179 : THIS ROUTINE ISSUES A MASTER CLEAR, WAITS FOR THE
 2180 : RUN BIT TO SET, CHECKS FOR GOOD COMPLETION OF MICRO
 2181 : DIAGNOSTICS AND ISSUES THE MODE DEFINITION.
 2182 : IF ENTERED AT MINIT1 - SET MODE TO FULL DUPLEX POINT
 2183 : TO POINT
 2184 : IF ENTERED AT MINITS - SET MODE TO FULL DUPLEX CONTROL
 2185 : IF ENTERED AT MINTR - SET MODE TO VALUE IN AXNUM
 2186 :
 2187 : OUTPUTS: ERRWRD = -1 IF ERROR OCCURS.

2188 :
 2189 : IMPLICIT OUTPUTS:
 2190 : DMP EXITS WITH MODE DEFINED
 2191 :
 2192 : SUBORDINATE ROUTINES USED:
 2193 : TOUT - TIME OUT ROUTINE
 2194 : WAIT50 - SHORT DELAY ROUTINE
 2195 : WRDO - WAIT FOR READY OUT
 2196 :
 2197 : CALLING SEQUENCE:
 2198 : JSR PC,MINITS ;OR MINIT1 OR MINTR
 2199 :--

2200 004440 012737 000003 002502 MINIT1: MOV #03,AXNUM
 2201 004446 000403 BR MINTR
 2202 004450 012737 000005 002502 MINITS: MOV #05,AXNUM
 2203 004456 112777 000100 175754 MINTR: MOVB #100,@BSEL1 ;SET MASTER CLEAR.
 2204 004464 022737 000004 002466 CMP #4,OPTYP ;IS THIS 8206
 2205 004472 001003 BNE MIN2R ;SKIP IF NOT
 2206 004474 112777 000200 175736 MOVB #200,@BSEL1 ;SET RUN BIT
 2207 004502 000240 MIN2R: NOP
 2208 004504 000240 NOP
 2209 004506 012737 004526 002400 MOV #ERLB7,ERRADD ;SET ERROR ADDRESS
 2210 004514 004537 005016 TLB7: JSR R5,TOUT
 2211 004520 005737 002374 TST ERRWRD
 2212 004524 100533 BMI MINTE ;EXIT IF ERROR
 2213 004526 ERLB7:

```

2214 004526 004737 004172      4$: JSR PC, WAIT50
2215 004532 005777 175700      TST @BSEL0 ;NOW WAIT FOR RUN TO ACTUALLY SET.
2216 004536 100366      BPL TLB7
2217 004540 012737 004560 002400  MOV #ERLB8, ERRADD ;SET ERROR ADDRESS
2218 004546 004537 005016      JSR R5, TOUT ;CHECK TIME OUT
2219 004552 005737 002374      TST ERRWRD
2220 004556 100516      BMI MINTE ;EXIT IF ERROR
2221 004560      ERLB8:
2222 004560 122777 000305 175664 5$: CMPB #305, @BSEL6 ;GOOD END TO MICRO DIAG?
2223 004566 001404      BEQ 2$
2224 004570 122777 000264 175654  CMPB #264, @BSEL6 ;LINE UNIT FAILURE.?
2225 004576 001363      BNE TLB8 ;NO-STAY IN LOOP.
2226 ;YES, CATCH THE PROBLEM LATER.
2227 004600 012737 000077 002334 2$: MOV #77, $GDDAT
2228 004606 032737 000003 002466  BIT #3, OPTYP ;IS THIS DMV
2229 004614 001403      BEQ 3$ ;IF NOT GO TO 3
2230 004616 012737 000033 002334  MOV #33, $GDDAT
2231 004624 023777 002334 175614 3$: CMP $GDDAT, @BSEL4
2232 004632 001407      BEQ 4$ ;IF CORRECT OPTYP CONTINUE
2233 004634 104457      TRAP CSERSOFT
2234 004636 000025      .WORD 21
2235 004640 012000      .WORD MEF32
2236 004642 007672      .WORD ERR32
2237 004644 005337 002374      DEC ERRWRD
2238 004650 000461      BR MINTE ;EXIT TEST
2239 004652 032737 000003 002466 4$: BIT #3, OPTYP ;IS THIS DMV
2240 004660 001423      BEQ 6$ ;IF NOT GO TO 6
2241 004662 112777 000301 175550  MOVB #301, @BSEL1
2242 004670 004537 002642      JSR R5, WRDO ;WAIT FOR READY OUT
2243 004674 012777 000020 175544  MOV #20, @BSEL4
2244 004702 012777 000006 175532  MOV #06, @BSEL2
2245 004710 004537 005016      60$: JSR R5, TOUT ;DO MAINT LOOP FOR TTL LOOPBACK
2246 004714 005737 002374      TST ERRWRD
2247 004720 100435      BMI MINTE
2248 004722 005777 175510      TST @SELO
2249 004726 100370      BPL 60$
2250 004730 152777 000010 175502 6$: BISB #BIT3, @BSEL1 ;SET LINE UNIT LOOP.-
2251 ;THIS ALLOWS US TO SET THE MODE.
2252 004736 052777 000200 175472      BIS #RQ1, @BSEL0
2253 004744 012737 004764 002400  MOV #ERLB9, ERRADD ;SET ERROR ADDRESS
2254 004752 004537 005016      TLB9: JSR R5, TOUT
2255 004756 005737 002374      TST ERRWRD
2256 004762 100414      BMI MINTE ;EXIT IF ERROR
2257 004764 032777 000020 175450  ERLB9: BIT #20, @BSEL2 ;WAIT FOR RDI.
2258 004772 001767      BEQ TLB9
2259 004774 013777 002502 175450  MOV AXNUM, @BSEL6 ;:SET UP FOR F/D CONTROL STATION
2260 005002 105077 175430      CLWB @BSEL0 ;NO MORE REQUESTS.
2261 005006 012777 000002 175426  MOV #2, @BSEL2 ;START.
2262 005014 000207      MINTE: RTS PC
2263

```

2264
2265
2266
2267
2268
2269
2270
2271
2272
2273 005016 020537 005064
2274 005022 001011
2275 005024 005237 005062
2276 005030 001012
2277 005032 104455
2278 005034 000020
2279 005036 010664
2280 005040 007536
2281 005042 005337 002374
2282 005046 005037 005062
2283 005052 010537 005064
2284 005056
2285 005056 104422
2286 005060 000205
2287
2288 005062 000000
2289
2290 005064 000000
2291

```

:++
: FUNCTIONAL DESCRIPTION:      TOUT  - TIME OUT ROUTINE
:                             THIS ROUTINE INC COUNTT LOCATION EVERY
:                             TIME IT IS CALLED IF COUNTT OVERFLOWS THEN
:                             TIME OUT IS REPORTED AND THE ROUTINE IS EXITED.
:
: CALLING SEQUENCE:
: JSR      R5,TOUT
:--
TOUT:   CMP      R5,LA5R5
        BNE     TOUTE
        INC     COUNTT
        BNE     TOUTEX
        TRAP    C$ERDF
        .WORD   16
        .WORD   MEF7
        .WORD   ERR26
        DEC     ERRWRD
TOUTE:  CLR     COUNTT
        MCV     R5,LA5R5      ;SAVE CURRENT PC.
TOUTEX: TRAP    C$BRK
        RTS     R5          ;EXIT
COUNTT: 0
LASTR5: 0
;NUMBERS OF TIMES IN THIS ROUTINE FROM
;SAME CALLING LOCATION
;LAST CALLING LOCATION.
```


2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306 005066
2307 005066 013746 002436
2308 005072 013746 002330
2309 005076 012746 005160
2310 005102 012746 000003
2311 005106 010600
2312 005110 104415
2313 005112 062706 000010
2314 005116 017746 175330
2315 005122 017746 175320
2316 005126 017746 175310
2317 005132 017746 175300
2318 005136 012746 005245
2319 005142 012746 000005
2320 005146 010600
2321 005150 104415
2322 005152 062706 000014
2323 005156 000207
2324 005160 047045 040445 040506
2325 005166 046111 047111 020107
2326 005174 041515 052520 044440
2327 005202 020123 047125 052111
2328 005210 021440 047445 022462
2329 005216 020101 044127 051517
2330 005224 020105 042101 051104
2331 005232 051505 020123 051511
2332 005240 022440 033117 000
2333 005245 045 022516 051501
2334 005252 046105 036460 047445
2335 005260 022466 020101 042523
2336 005266 031114 022475 033117
2337 005274 040445 051440 046105
2338 005302 036464 047445 022466
2339 005310 020101 042523 033114
2340 005316 022475 033117 047045
2341 005324 000
2342 005326
2343

++
FUNCTIONAL DESCRIPTION: STAND - PRINT STANDARD REGS
THIS ROUTINE PRINTS THE UNIT NUMBER AND
CSR ADDRESS OF THE FAILING UNIT AS WELL AS THE
CONTENTS OF ALL THE CSR REGS.
THE ERROR MMSG ROUTINES USE THIS SUBROUTNE
IMPLICIT INPUTS:
CSRS' - THE CSR ARE EXPECTED TO CONTAIN USEFUL DATA
LOGDEV - THE LOGICAL DEVICE NUMBER
SELO - ADDRESS OF THIS UNIT
CALLING SEQUENCE:
JSR PC,STAND
--

STAND:
MOV SELO,-(SP)
MOV LOGDEV,-(SP)
MOV #CFM1,-(SP)
MOV #3,-(SP)
MOV SP,RO
TRAP C\$PNTX
ADD #10,SP
MOV @SEL6,-(SP)
MOV @SEL4,-(SP)
MOV @SEL2,-(SP)
MOV @SELO,-(SP)
MOV #CFM2,-(SP)
MOV #5,-(SP)
MOV SP,RO
TRAP C\$PNTX
ADD #14,SP
RTS PC
CFM1: .ASCIZ '%N%AFAILING MCPU IS UNIT #X02XA WHOSE ADDRESS IS X06%'
CFM2: .ASCIZ '%N%ASELO=X06XA SEL2=X06XA SEL4=X06XA SEL6=X06XN%'
.EVEN

2344
 2345
 2346
 2347
 2348
 2349
 2350
 2351
 2352
 2353
 2354
 2355
 2356 005326
 2357 005326 005037 002430
 2358 005332 012737 177777 002372
 2359 005340 112777 000301 175072
 2360 005346 004537 002642
 2361 005352 005737 002374
 2362 005356 100470
 2363 005360 012777 033022 175060
 2364 005366 013777 002430 175056
 2365 005374 112777 000003 175040
 2366 005402 004537 002642
 2367 005406 005737 002374
 2368 005412 100452
 2369 005414 005002
 2370 005416 016237 033022 002370
 2371 005424 004737 004324
 2372 005430 022702 000376
 2373 005434 062702 000002
 2374 005440 001366
 2375 005442 023737 002432 002430
 2376 005450 001004
 2377 005452 016237 033022 002426
 2378 005460 000405
 2379 005462 016237 033022 002370
 2380 005470 004737 004324
 2381 005474 023737 002434 002430
 2382 005502 001404
 2383 005504 062737 000400 002470
 2384 005512 000722
 2385 005514 005137 002372
 2386 005520 023737 002372 002426
 2387 005526 001404
 2388 005530 104455
 2389 005532 000021
 2390 005534 000000
 2391 005536 007432
 2392 005540
 2393 005540
 2394 005540 000205

++
 : FUNCTIONAL DESCRIPTION: RMVRT - VERIFY ROM CONTENTS
 : THIS ROUTINE READS DMV ROMS USING 256 BYTE READS
 : AND CALCULATES CRC.

: SUBORDINATE ROTUINES:
 : WRDO - WAIT FOR READY OUT
 : CRCR - CALCULATE CRC
 : CALLING SEQUENCE:
 : JSR R5,RMVRT
 :--

```

RMVRT:
  CLR ROMADD ;CLEAR ROM ADD
  MOV #-1,CWORD ;INIT CRC WORD
  MOVB #301,@BSEL1 ;MAINT LOOP
  JSR R5,WRDO
  TST ERRWRD
  BMI RMVEX ;EXIT IF ERROR
  RMVXX: MOV #RECBU1,@BSEL4
  MOV ROMADD,@BSEL6 ;SET UP BOTH ADDRESSES
  MOVB #03,@BSEL2 ;DO BLOCK READ OF 256 BYTES
  JSR R5,WRDO
  TST ERRWRD
  BMI RMVEX ;EXIT IF ERROR
  CLR R2
  RMVYY: MOV RECBU1(R2),WORDT ;GET WORD
  JSR PC,CRCR ;CALC CRC
  CMP #254.,R2 ;IS IT LAST WORD
  ADD #2,R2
  BNE RMVYY ;IF NOT GO BACK
  CMP ROMBK1,ROMADD ;IS THIS THE CRC WORD
  BNE RMVZZ
  MOV RECBU1(R2),CRCCAL ;STORE OFF CALCULATED
  BR RMVAA ;THEN CONTINUE
  RMVZZ: MOV RECBU1(R2),WORDT
  JSR PC,CRCR
  RMVAA: CMP ROMBK2,ROMADD ;IS THIS THE LAST BLOCK
  BEQ RMVBB
  ADD #256.,ROMADD ;IF NOT ADD TO ADD
  BR RMVXX ;AND GO TO ANOTHER
  RMVBB: COM CWORD
  CMP CWORD,CRCCAL ;COMPARE
  BEQ RMVCC
  TRAP C$ERDF
  .WORD 17
  .WORD 0
  .WORD ERR24
  RMVEX:
  RMVCC:
    RTS R5 ;RETURN TO CALLER
  
```

2395
 2396
 2397
 2398
 2399
 2400
 2401
 2402
 2403
 2404
 2405
 2406
 2407
 2408
 2409
 2410
 2411
 2412
 2413
 2414
 2415
 2416
 2417
 2418
 2419
 2420
 2421
 2422
 2423
 2424
 2425
 2426
 2427
 2428
 2429
 2430
 2431
 2432
 2433
 2434
 2435
 2436
 2437
 2438
 2439
 2440
 2441
 2442
 2443
 2444
 2445
 2446
 2447
 2448
 2449
 2450

++
 FUNCTIONAL DESCRIPTION: TXRXSR - TRANSMIT RECEIVE SUBROUTINE
 THIS ROUTINE DOES IS USED BY ALL TESTS THAT TRANSMIT
 AND RECIEVE DATA. THE FIRST PART OF THE ROUTINE VERIFIES
 THE OPERATOR INPUTS AND MAKES SURE THAT INTERFACE
 SELECTION CORRESPOND TO SELECTED BAUD RATES.
 THE SECOND PART OF THE ROUTINE FORM "'TXRX3'" ON DOES THE
 FOLLOWING. ESTABLIS TRIBUTARY. THEN EITHER DO ISTRT OR
 MAINT STATE DEPENDING ON FLAG. IF ISTRT THEN CHECK FOR
 RUN STATE IF MAINT STATE THEN GO TO NEXT STEP. NEXT QUE
 REC AND TRANSMIT BUFFERS THEN WAIT FOR OUTPUT. IF MEMORY
 MANAGEMENT EXIT TEST. IF ISTRT LOOK FOR REC COMPLETED
 FOLLOWED BY TX COMPLETED. IF MAINT LOOK FOR TRANSMIT FIRST.

INPUTS: TXADD - ADDRESS OF TRANSMIT BUFFER
 TXCC - CHAR COUNT OF TX
 RXADD - ADDRESS OF REC BUFFER
 RXCC - CHAR COUNT OF REC BUFFER
 GENWRD - FLAG WORD IF BIT 15 SET-MAINT MODE
 IF BIT 14 SET THEN MEMORY MGT.

SUBORDINATE ROUTINES USED:
 WAIT50 - SHORT DELAY
 CONTIN - CONTROL IN ROUTINE
 GETOUT - GET CONTROL OUT CODE
 GETOC - GET OUTPUT CODE
 WRDI - WAIT FOR READY IN
 GETCC - GET CHAR COUNT
 GETBA - GET BUFFER ADDRESS

CALLING SEQUENCE:
 JSR R5, TXRXSR ;IF NOT INTERFACE CHECK
 ;CALL AT JSR R5, TXRX3

--

```

2431 005542 022737 000004 002472 TXRXSR: CMP #4, ISTRCON
2432 005550 001002 BNE 10$ ;IF INTERNAL LOOP
2433 005552 000137 006154 JMP TXRX3 ;GO TO 3
;JUMP TO TXRX 3 IF INTERNAL LOOP
2437 005556 004737 004172 10$: JSR PC, WAIT50 ;WAIT FOR A WHILE
2438 005562 142777 000010 174650 BICB #BIT3, @BSEL1 ;CLEAR LU LOOP IF NOT INTERNAL
2439 005570 013737 002356 002502 MOV TRIBN, AXNUM ;SAVE TRIBN.
2440 005576 005037 002356 CLR TRIBN ;MAKE TRIBN 0
; IF INTERGRAL MODEM MAKE SURE OPTION
; IS NOT 8053 AND SPEED IS 56K OR HIGHER
; IF NOT INTERGRAL MODEM GO TO NEXT CHECK IF EIA
2446 005602 022737 000001 002470 CMP #1, IFTYP ;IS THIS INTEGRAL MODEM
2447 005610 001014 BNE XYZTC ;IF NOT THEN GO CHECK FOR EIA
2448 005612 022737 000001 002466 CMP #1, OPTYP ;IS THIS 8053 (DMV NO INTEGRAL)
2449 005620 001453 BEQ BADIF ;IF SO PRINT BAD INTERFACE MESSG.
2450 005622 023727 002462 000004 CMP SPEEDM, #4 ;IS THIS 56 K OR HIGHER
  
```

```

2451 005630 103464          BLO      BADBR          ;IF NOT PRINT BAD BAUD RATE
2452
2453                          ; GET HERE IF EVERYTHING OK FOR INTEGRAL
2454
2455 005632 012704 000323    MOV      #323,R4        ;LOAD R4 WITH INTERFACE TYPE
2456 005636 000137 006062    JMP      SETIF          ; AND GO SET IT
2457
2458                          ;IF THIS IS EIA THEN CHECK THAT OPTION IS NOT
2459                          ; M8054 AND THAT SPEED IS 56K OR LOWER
2460                          ;IF NOT EIA CHECK IF V.35
2461
2462 005642 022737 000002 002470 YV2TC: CMP      #2,IF1YP        ;IS THIS EIA INTERFACE
2463 005650 001014          BNE     V35TC          ;IF NOT GO CHECK FOR V.35 TYPE
2464 005652 022737 000002 002466    CMP      #2,OPTYP        ;IS THIS 8054 (DMV INTEGRAL MODEM)
2465 005660 001433          BEQ     BADIF          ;IF SO PRINT BAD INTERFACE
2466 005662 023727 002462 000004    CMP      SPEEDM,#4      ;IS THIS HIGHER THAN 56K
2467 005670 101044          BHI     BADBR          ;IF SO PRINT BAD BAUD RATE
2468
2469                          ; GET HERE IF EVERYTHING OK FOR EIA
2470
2471 005672 012704 000233    MOV      #233,R4        ;LOAD R4 WITH INTERFACE TYPE
2472 005676 000137 006062    JMP      SETIF          ;AND GO SET IT.
2473
2474                          ;IF V.35 CHECK THAT OPTION IS NOT 8064
2475                          ;IF NOT V.35 MUST BE 422
2476
2477 005702 022737 000003 002470 V35TC: CMP      #3,IF1YP        ;IS THIS V.35 INTERFACE
2478 005710 001010          BNE     T422C          ;IF NOT IT MUST BE 422
2479 005712 022737 000002 002466    CMP      #2,OPTYP        ;IS THIS 8064 (DMV INTEGRAL)
2480 005720 001413          BEQ     BADIF          ;IF SO PRINT BAD INTERFACE
2481
2482                          ; GET HERE IF EVERYTHING OK FOR V.35
2483
2484 005722 012704 000313    MOV      #313,R4        ;LOAD R4 WITH INTERFACE TYPE AND
2485 005726 000137 006062    JMP      SETIF          ;GO SET IT
2486
2487                          ;IF OPTION TYPE IS DMV THEN ERROR ELSE OK FOR 422
2488
2489 005732 032737 000003 002466 1422C: BIT      #3,OPTYP        ;IS THIS DMV
2490 005740 001003          BNE     BADIF          ;IF SO PRINT BAD INTERFACE
2491
2492                          ; GET HERE IF EVERYTHING OK FOR 422
2493
2494 005742 012704 000133    MOV      #133,R4        ;LOAD UP INTERFACE AND SET IT
2495 005746 000445          BR      SETIF
2496
2497                          ;PRINTS BAD INTERFACE AND EXITS
2498
2499 005750 022737 000001 002344 BADIF: CMP      #1,STARES
2500 005756 001037          BNE     TABEN
2501 005760 012746 007736    MOV      #BADIFM,-(SP)
2502 005764 012746 000001    MOV      #1,-(SP)
2503 005770 010600          MOV     SP,R0
2504 005772 104417          TRAP   C$PNTF
2505 005774 062706 000004    ADD     #4,SP
2506 006000 000414          BR      TABM          ;PRINT 1ST PART OF MESSAGE

```



```

2563 006216 100432          BMI      TXRXA          :EXIT IF ERROR OCCURRED
2564
2565                      :CHECK FOR RUN STATE
2566
2567 006220 012737 000001 002334  MOV      #01,$GDDAT      :CHECK FOR CONTROL OUT
2568 006226 004537 003236          JSR      R5,GETOUT      : AND CORRECT TRIBN
2569 006232 005737 002374          TST      ERRWRD          :
2570 006236 100422          BMI      TXRXA          :EXIT IF ERROR OCCURRED
2571 006240 012737 000024 002334  MOV      #24,$GDDAT      :CHECK FOR RUN STATE
2572 006246 004537 003616          JSR      R5,GETOC      : IN OUTPUT CODE
2573 006252 005737 002374          TST      ERRWRD          :
2574 006256 100412          BMI      TXRXA          :EXIT IF ERROR OCCURRED
2575 006260 042777 000200 174154  BIC      #RDO,@BSEL2     :CLEAR OUTPUT
2576 006266 000407          BR       TXRXC          :AND GO TO 20
2577
2578                      : PUT TRIB IN M'INT STATE
2579
2580 006270 012703 000004          TXRXB: MOV      #04,R3      :PUT TRIB IN MAINT STATE
2581 006274 004537 003156          JSR      R5,CONTIN      :
2582                      : TIME OUT OR READY ERRORS REPORT THIS PC
2583 006300 005737 002374          TXRXA: TST      ERRWRD      :
2584 006304 100567          BMI      TXRXEN          :EXIT IF ERROR OCCURRED
2585
2586                      :QUEUE REC BUFFER
2587
2588 006306 052777 000200 174122  TXRXC: BIS      #RQI,@BSEL0 :SET REQUEST
2589 006314 004537 002726          JSR      R5,WRDI        :
2590                      : TIME OUT OR READY ERROR REPORT THIS PC
2591 006320 005737 002374          TST      ERRWRD          :
2592 006324 100557          BMI      TXRXEN          :EXIT IF ERROR OCCURRED
2593 006326 013777 002414 174112  MOV      RXADD,@BSEL4    :SET ADDRESS
2594 006334 013777 002416 174110  MOV      RXCC,@BSEL6     :SET CHAR COUNT
2595 006342 113777 002356 174074  MOVB    TRIBN,@BSEL3    :SET TRIBN
2596 006350 112777 000000 174064  MOVB    #0,@BSEL2      :SET RX BUFFER IN
2597
2598                      :QUEUE TX BUFFER
2599
2600 006356 004537 002726          JSR      R5,WRDI        :WAIT FOR READY
2601                      : TIME OUT OR READY ERROR REPORT THIS PC
2602 006362 005737 002374          TST      ERRWRD          :
2603 006366 100536          BMI      TXRXEN          :EXIT IF ERROR OCCURRED
2604 006370 013777 002412 174050  MOV      TXADD,@BSEL4    :SET TX ADD
2605 006376 013777 002420 174046  MOV      TXCC,@BSEL6     :SET CHAR COUNT
2606 006404 113777 002356 174032  MOVB    TRIBN,@BSEL3    :SET TRIBN
2607 006412 042777 000200 174016  BIC      #RQI,@BSEL0    :CLEAR REQUEST
2608 006420 112777 000004 174014  MOVB    #04,@BSEL2     :SET UP TX BUFFER
2609 006426 032737 040000 002424  BIT      #BIT14,GENWRD  :
2610 006434 001113          BNE     TXRXEN          :IF MM GO TO RETURN
2611 006436 005737 002424          TST      GENWRD          :
2612 006442 100013          BPL     TXRXG          :GO AHEAD IF NOT MAINT STATE
2613
2614                      :CHECK FOR RX COMPLETED
2615
2616 006444 012737 000004 002334  MOV      #04,$GDDAT      :GET OUTPUT CODE
2617 006452 004537 003236          JSR      R5,GETOUT      :
2618 006456 005737 002374          TST      ERRWRD          :

```

```

2619 006462 100500          BMI      TXRXEN      ;EXIT IF ERROR
2620 006464 042777 000200 173750    BIC      #RDO,@BSEL2 ;CLEAR READY OUT
2621 006472 012737 000000 002334 TXRXG:  MOV      #0,$GDDAT ;
2622 006500 004537 003236          JSR      R5,GETOUT ;CHECK FOR RX RETURN'
2623 006504 005737 002374          TST      ERRWRD ;
2624 006510 100465          BMI      TXRXEN      ;EXIT IF ERROR OCCURRED
2625 006512 013737 002420 002334    MOV      TXCC,$GDDAT ;
2626 006520 004537 003510          JSR      R5,GETCC ;IS THE CHAR COUNT CORRECT
2627 006524 005737 002374          TST      ERRWRD ;
2628 006530 100455          BMI      TXRXEN      ;EXIT IF ERROR OCCURRED
2629 006532 013737 002414 002334    MOV      RXADD,$GDDATT ;
2630 006540 004537 003556          JSR      R5,GETBA ;IS THE BUFFER ADD RIGHT
2631 006544 005737 002374          TST      ERRWRD ;
2632 006550 100445          BMI      TXRXEN      ;EXIT IF ERROR OCCURRED
2633 006552 013703 002414          MOV      RXADD,R3 ;
2634 006556 013701 002412          MOV      TXADD,R1 ;SET UP ADDRESS
2635 006562 005004          CLR      R4 ;CLEAR R4
2636 006564          25$:
2637 006564 112337 002336          MOVB     (R3)+,$BDDAT ;GET BYTE OF RX
2638 006570 112137 002334          MOVB     (R1)+,$GDDAT ;GET BYTE OF TX
2639 006574 123737 002336 002334    CMPB     $BDDAT,$GDDAT ;ARE THEY THE SAME
2640 006602 001411          BEQ      30$ ;IF SO GO TO 30
2641 006604 005204          INC      R4 ;MAKE COUNT RIGHT
2642 006606 104455          TRAP     C$ERDF ;
2643 006610 000022          .WORD   18 ;
2644 006612 011607          .WORD   MEF19A ;
2645 006614 007164          .WORD   ERR19 ;
2646 006616 005337 002374          DEC      ERRWRD ;
2647 006622 005304          DEC      R4 ;MAKE COUNT RIGHT
2648 006624 000417          BR       TXRXEN      ;EXIT IF ERROR
2649 006626 005204          30$:  INC      R4 ;BUMP TO NEXT BYTE
2650 006630 020437 002420          CMP      R4,TXCC ;ARE WE DONE?
2651 006634 103753          BLO     25$ ;IF NOT GO BACK
2652 006636 005737 002424          TST      GENWRD ;TEST FOR MAINT STATE
2653 006642 100410          BMI      TXRXEN      ;RETURN TO CALLER IF MAINT STATE
2654 006644 042777 000200 173570    BIC      #RDO,@BSEL2 ;CLEAR OUTPUT
2655 006652 012737 000004 002334    MOV      #4,$GDDAT ;
2656 006660 004537 003236          JSR      R5,GETOUT ;CHECK FOR TX BUFF COMP
2657 006664 000205          TXRXEN: RTS      R5
2658

```

2659
2660
2661
2662
2663
2664
2665
2666 006666
2667 006666 013746 002436
2668 006672 012746 010145
2669 006676 012746 000002
2670 006702 010600
2671 006704 104414
2672 006706 062706 000006
2673 006712
2674 006712 104423
2675
2676
2677
2678
2679 006714
2680 006714 017746 173532
2681 006720 012746 010355
2682 006724 012746 000002
2683 006730 010600
2684 006732 104414
2685 006734 062706 000006
2686 006740 004737 005066
2687 006744
2688 006744 104423
2689
2690 006746
2691 006746 013746 002336
2692 006752 013746 002334
2693 006756 012746 010513
2694 006762 012746 000003
2695 006766 010600
2696 006770 104414
2697 006772 062706 000010
2698 006776 004737 005066
2699 007002
2700 007002 104423
2701
2702 007004
2703 007004 013746 002336
2704 007010 013746 002334
2705 007014 012746 010606
2706 007020 012746 000003
2707 007024 010600
2708 007026 104414
2709 007030 062706 000010
2710 007034 004737 005066
2711 007040
2712 007040 104423
2713
2714

.SBTTL GLOBAL ERROR REPORT SECTION

:/ THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
:/ THAT ARE USED IN MORE THAN ONE TEST.

ERR1::
MOV SELO,-(SP)
MOV #MEF1,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #6,SP

L10003:
TRAP C\$MSG

:FAILING CODE

ERR3::
MOV @BSEL6,-(SP)
MOV #MEF3,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #6,SP
JSR PC,STAND

L10004:
TRAP C\$MSG

ERR5::
MOV \$BDDAT,-(SP)
MOV \$GDDAT,-(SP)
MOV #MEF5,-(SP)
MOV #3,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #10,SP
JSR PC,STAND

L10005:
TRAP C\$MSG

ERR6::
MOV \$BDDAT,-(SP)
MOV \$GDDAT,-(SP)
MOV #MEF6,-(SP)
MOV #3,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #10,SP
JSR PC,STAND

L10006:
TRAP C\$MSG

:PRINT FAILED TO SET RDI

2715					
2716	007042			ERR9::	
2717	007042	012746	011130		MOV #MRDI,-(SP)
2718	007046	012746	000001		MOV #1,-(SP)
2719	007052	010600			MOV SP,RO
2720	007054	104414			TRAP C\$PNTB
2721	007056	062706	000004		ADD #4,SP
2722	007062	004737	005066		JSR PC,STAND
2723	007066			L10007:	
2724	007066	104423			TRAP C\$MSG
2725					
2726					;PRINT FAILED TO SET RDO
2727					
2728	007070			ERR10::	
2729	007070	012746	011154		MOV #MRDO,-(SP)
2730	007074	012746	000001		MOV #1,-(SP)
2731	007100	010600			MOV SP,RO
2732	007102	104414			TRAP C\$PNTB
2733	007104	062706	000004		ADD #4,SP
2734	007110	004737	005066		JSR PC,STAND
2735	007114			L10010:	
2736	007114	104423			TRAP C\$MSG
2737					
2738					
2739					;PRINTS GOOD AND BAD DATA AND
2740					;FAILING PC ADDRS AND STANARD REGS
2741					
2742	007116			ERR18::	
2743	007116	013746	002400		MOV ERRADD,-(SP)
2744	007122	012746	011061		MOV #MFPC,-(SP)
2745	007126	013746	002336		MOV \$BDDAT,-(SP)
2746	007132	013746	002334		MOV \$GDDAT,-(SP)
2747	007136	012746	011551		MOV #MEF18,-(SP)
2748	007142	012746	000005		MOV #5,-(SP)
2749	007146	010600			MOV SP,RO
2750	007150	104414			TRAP C\$PNTB
2751	007152	062706	000014		ADD #14,SP
2752	007156	004737	005066		JSR PC,STAND
2753	007162			L10011:	
2754	007162	104423			TRAP C\$MSG
2755					
2756					;DATA COMPARE ERROR
2757					
2758	007164			ERR19::	
2759	007164	013746	002336		MOV \$BDDAT,-(SP)
2760	007170	013746	002334		MOV \$GDDAT,-(SP)
2761	007174	010446			MOV R4,-(SP)
2762	007176	012746	011642		MOV #MEF19,-(SP)
2763	007202	012746	000004		MOV #4,-(SP)
2764	007206	010600			MOV SP,RO
2765	007210	104414			TRAP C\$PNTB
2766	007212	062706	000012		ADD #12,SP
2767	007216	004737	005066		JSR PC,STAND
2768	007222			L10012:	
2769	007222	104423			TRAP C\$MSG
2770					

2771 007224
2772 007224 013746 002536
2773 007230 012746 012045
2774 007234 012746 000002
2775 007240 010600
2776 007242 104414
2777 007244 062706 000006
2778 007250 004737 005066
2779 007254
2780 007254 104423
2781
2782 007256
2783 007256 012746 012170
2784 007262 012746 000001
2785 007266 010600
2786 007270 104414
2787 007272 062706 000004
2788 007276 013746 002336
2789 007302 013746 002334
2790 007306 013746 002536
2791 007312 012746 012257
2792 007316 012746 000004
2793 007322 010600
2794 007324 104414
2795 007326 062706 000012
2796 007332 004737 005066
2797 007336
2798 007336 104423
2799
2800 007340
2801 007340 012746 012341
2802 007344 012746 000001
2803 007350 010600
2804 007352 104414
2805 007354 062706 000004
2806 007360 013746 002536
2807 007364 012746 012426
2808 007370 012746 000002
2809 007374 010600
2810 007376 104414
2811 007400 062706 000006
2812 007404 012746 012516
2813 007410 012746 000001
2814 007414 010600
2815 007416 104414
2816 007420 062706 000004
2817 007424 004737 005066
2818 007430
2819 007430 104423
2820
2821 007432
2822 007432 013746 002372
2823 007436 013746 002370
2824 007442 013746 002364
2825 007446 012746 012602
2826 007452 012746 000004

ERR20::
MOV \$TMPO,-(SP)
MOV #TFM20,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #6,SP
JSR PC,STAND
L10013:
TRAP C\$MSG
ERR21::
MOV #TFM21,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #4,SP
MOV \$BDDAT,-(SP)
MOV \$GDDAT,-(SP)
MOV \$TMPO,-(SP)
MOV #TFM2A,-(SP)
MOV #4,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #12,SP
JSR PC,STAND
L10014:
TRAP C\$MSG
ERR22::
MOV #TFM22,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #4,SP
MOV \$TMPO,-(SP)
MOV #TFM22A,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #6,SP
MOV #TFM22B,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C\$PNTB
ADD #4,SP
JSR PC,STAND
L10015:
TRAP C\$MSG
ERR24::
MOV CWORD,-(SP)
MOV WORDT,-(SP)
MOV ROMN,-(SP)
MOV #TFM24,-(SP)
MOV #4,-(SP)

2827	007456	010600		MOV	SP,RO
2828	007460	104414		TRAP	C\$PNTB
2829	007462	062706	G00012	ADD	#12,SP
2830	007466	004737	G05066	JSR	PC,STAND
2831	007472			L10016:	
2832	007472	104423		TRAP	C\$MSG
2833					
2834	007474			ERR25::	
2835	007474	013746	002370	MOV	WORDT,-(SP)
2836	007500	013746	002366	MOV	ROMN1,-(SP)
2837	007504	013746	002364	MOV	ROMN,-(SP)
2838	007510	012746	012670	MOV	#TFM25,-(SP)
2839	007514	012746	000004	MOV	#4,-(SP)
2840	007520	010600		MOV	SP,RO
2841	007522	104414		TRAP	C\$PNTB
2842	007524	062706	000012	ADD	#12,SP
2843	007530	004737	005066	JSR	PC,STAND
2844	007534			L10017:	
2845	007534	104423		TRAP	C\$MSG
2846					
2847					;PRINTS FAILING PC ADDRESS AND STANARD REGS
2848					
2849	007536			ERR26::	
2850	007536	013746	002400	MOV	ERRADD,-(SP)
2851	007542	012746	011061	MOV	#MFPC,-(SP)
2852	007546	012746	011053	MOV	#MEF1A,-(SP)
2853	007552	012746	000003	MOV	#3,-(SP)
2854	007556	010600		MOV	SP,RO
2855	007560	104414		TRAP	C\$PNTB
2856	007562	062706	000010	ADD	#10,SP
2857	007566	004737	005066	JSR	PC,STAND
2858	00757.			L10020:	
2859	007572	104423		TRAP	C\$MSG
2860					
2861					;PRINTS FALILING PC ADDRESS AND
2862					;CODE IN ERROR FROM CODEW AND
2863					;STANDARD REGISTERS
2864					
2865	007574			ERR27::	
2866	007574	013746	002400	MOV	ERRADD,-(SP)
2867	007600	012746	011061	MOV	#MFPC,-(SP)
2868	007604	012746	011053	MOV	#MEF1A,-(SP)
2869	007610	012746	000003	MOV	#3,-(SP)
2870	007614	010600		MOV	SP,RO
2871	007616	104414		TRAP	C\$PNTB
2872	007620	062706	000010	ADD	#10,SP
2873	007624	013746	002336	MOV	\$BDDAT,-(SP)
2874	007630	013746	002334	MOV	\$GDDAT,-(SP)
2875	007634	012746	011200	MOV	#MGB,-(SP)
2876	007640	013746	002422	MOV	CODEW,-(SP)
2877	007644	012746	011006	MOV	#MEF11,-(SP)
2878	007650	012746	000005	MOV	#5,-(SP)
2879	007654	010600		MOV	SP,RO
2880	007656	104414		TRAP	C\$PNTB
2881	007660	062706	000014	ADD	#14,SP
2882	007664	004737	005066	JSR	PC,STAND

2883	007670				L10021:	
2884	007670	104423			TRAP	C\$MSG
2885						
2886						;PRINTS THE STANDARD REGS
2887						
2888	007672				ERR32::	
2889	007672	012746	007720		MOV	#BASER,-(SP)
2890	007676	012746	000001		MOV	#1,-(SP)
2891	007702	010600			MOV	SP,R0
2892	007704	104414			TRAP	C\$PNTB
2893	007706	062706	000004		ADD	#4,SP
2894	007712	004737	005066		JSR	PC,STAND
2895	007716				L10022:	
2896	007716	104423			TRAP	C\$MSG
2897						
2898	007720	040445	040502	044523	BASER:	.ASCIZ '%ABASIC ERROR'
2899	007726	020103	051105	047522		
2900	007734	000122				
2901	007736	047045	040445	044440	BADIFM:	.ASCIZ '%N%A INCORRECT INTERFACE FOR OPTION SELECTED ''
2902	007744	041516	051117	042522		
2903	007752	052103	044440	052116		
2904	007760	051105	040506	042503		
2905	007766	043040	051117	047440		
2906	007774	052120	047511	020116		
2907	010002	042523	042514	052103		
2908	010010	042105	000040			
2909	010014	047045	040445	044440	BADBRM:	.ASCIZ '%N%A INCORRECT BAUD RATE FOR INTERFACE SELECTED''
2910	010022	041516	051117	042522		
2911	010030	052103	041040	052501		
2912	010036	020104	040522	042524		
2913	010044	043040	051117	044440		
2914	010052	052116	051105	040506		
2915	010060	042503	051440	046105		
2916	010066	041505	042524	000104		
2917	010074	047045	040445	025040	TESTAB:	.ASCIZ '%N%A ***** SUBTEST %02XA ABORTED ***** ''
2918	010102	025052	025052	020052		
2919	010110	052523	052102	051505		
2920	010116	020124	047445	022462		
2921	010124	020101	041101	051117		
2922	010132	042524	020104	025052		
2923	010140	025052	020052	000		
2924						
2925	010145	045	020101	042101	MEF1:	.ASCIZ '%A ADDRESSING PROBLEM UNIT ADDRESS %06%N''
2926	010152	051104	051505	044523		
2927	010160	043516	050040	047522		
2928	010166	046102	046505	052440		
2929	010174	044516	020124	042101		
2930	010202	051104	051505	020123		
2931	010210	047445	022466	000116		
2932	010216	042101	051104	051505	EMTO:	.ASCIZ '/ADDRESS ERROR -TRAP 4/'
2933	010224	020123	051105	047522		
2934	010232	020122	052055	040522		
2935	010240	020120	000064			
2936	010244	051445	022463	041501	FMT0:	.ASCIZ '/%S3%ACSR (SEL%01XA) DOES NOT RESPOND%N/'
2937	010252	051123	024040	042523		
2938	010260	022514	030504	040445		

2939	010266	020051	047504	051505	
2940	010274	047040	052117	051040	
2941	010302	051505	047520	042116	
2942	010310	047045	000		
2943	010313	111	052116	051105	MEF3A: .ASCIZ /INTERNAL DMP-11 DIAGNOSITC FAILED/
2944	010320	040516	020114	046504	
2945	010326	026520	030461	042040	
2946	010334	040511	047107	051517	
2947	010342	052111	020103	040506	
2948	010350	046111	042105	000	
2949	010355	045	052101	051505	MEF3: .ASCII '%ATEST CODE- %03'
2950	010362	020124	047503	042504	
2951	010370	020055	047445	063	
2952	010375	045	022516	044501	MEF4: .ASCIZ /%N%INTERNAL DMP-11-LINE UNIT TEST FAILURE/
2953	010402	052116	051105	040516	
2954	010410	020114	046504	026520	
2955	010416	030461	046055	047111	
2956	010424	020105	047125	052111	
2957	010432	052040	051505	020124	
2958	010440	040506	046111	051125	
2959	010446	000105			
2960	010450	047111	042524	043122	MEFC: .ASCIZ /INTERFACE MICRO-DIAGNOSTIC FAILURE/
2961	010456	041501	020105	044515	
2962	010464	051103	026517	044504	
2963	010472	043501	047516	052123	
2964	010500	041511	043040	044501	
2965	010506	052514	042522	000	
2966	010513	045	020101	052515	MEF5: .ASCII '%A MULTIPORT RAM WRITE/READ ERROR%N''
2967	010520	052114	050111	051117	
2968	010526	020124	040522	020115	
2969	010534	051127	052111	027505	
2970	010542	042522	042101	042440	
2971	010550	051122	051117	047045	
2972	010556	040445	043440	047517	.ASCIZ '%A GOOD= %03%A BAD= %03''
2973	010564	036504	022440	031517	
2974	010572	040445	041040	042101	
2975	010600	020075	047445	000063	
2976					
2977	010606	040445	047040	051120	MEF6: .ASCII '%A NPR TRANSFER TEST''
2978	010614	052040	040522	051516	
2979	010622	042506	020122	042524	
2980	010630	052123			
2981	010632	047045	040445	043440	.ASCIZ '%N%A GOOD= %06%A BAD= %06''
2982	010640	047517	036504	022440	
2983	010646	033117	040445	041040	
2984	010654	042101	020075	047445	
2985	010662	000066			
2986					
2987	010664	047125	054105	042520	MEF7: .ASCIZ /UNEXPECTED TEST LOOP HANG/
2988	010672	052103	042105	052040	
2989	010700	051505	020124	047514	
2990	010706	050117	044040	047101	
2991	010714	000107			
2992	010716	046504	020120	047111	MEF8: .ASCIZ /DMP INTERRUPTED TO WRONG VECTOR/
2993	010724	042524	051122	050125	
2994	010732	042524	020104	047524	

2995	010740	053440	047522	043516		
2996	010746	053040	041505	047524		
2997	010754	000122				
2998	010756	047111	047503	051122	EROIC.	.ASCIZ /INCORRECT CODE RETURNED/
2999	010764	041505	020124	047503		
3000	010772	042504	051040	052105		
3001	011000	051125	042516	000104		
3002	011006	047045	052045	040445	MEF11:	.ASCIZ 'XNXTXA CODE INCORRECTXNXTXN%06XS3%06'
3003	011014	041440	042117	020105		
3004	011022	047111	047503	051122		
3005	011030	041505	022524	022516		
3006	011036	022524	022516	033117		
3007	011044	051445	022463	033117		
3008	011052	000				
3009	011053	045	022524	033117	MEF1A:	.ASCIZ '%T%06'
3010	011060	000				
3011	011061	106	044501	052514	MFPC:	.ASCIZ 'FAILURE AT PC '
3012	011066	042522	040440	020124		
3013	011074	041520	000040			
3014						
3015	011100	047522	020115	052506	MRFT:	.ASCIZ 'ROM FUNCTION TEST ERROR'
3016	011106	041516	044524	047117		
3017	011114	052040	051505	020124		
3018	011122	051105	047522	000122		
3019	011130	040445	042122	020111	MRDI:	.ASCIZ '%ARDI FAILED TO SET'
3020	011136	040506	046111	042105		
3021	011144	052040	020117	042523		
3022	011152	000124				
3023	011154	040445	042122	020117	MRDO:	.ASCIZ '%ARDO FAILED TO SET'
3024	011162	040506	046111	042105		
3025	011170	052040	020117	042523		
3026	011176	000124				
3027	011200	047507	042117	020040	MGB:	.ASCIZ 'GOOD BAD'
3028	011206	020040	020040	041040		
3029	011214	042101	000			
3030	011217	122	052105	051125	M12F:	.ASCIZ 'RETURN KEY'
3031	011224	020116	042513	000131		
3032	011232	051105	047522	000122	M13F:	.ASCIZ 'ERROR'
3033	011240	042122	000117		MFRO:	.ASCIZ 'RDO'
3034	011244	042122	000111		MFRI:	.ASCIZ 'RDI'
3035	011250	052517	050124	052125	M18F:	.ASCIZ 'OUTPUT'
3036	011256	000				
3037	011257	124	050131	000105	M28F:	.ASCIZ 'TYPE'
3038	011264	040504	040524	000	M30F:	.ASCIZ 'DATA'
3039	011271	122	044504	051440	MEF14:	.ASCIZ 'RDI SET WHEN EXPECTING RDO TO BE SET''
3040	011276	052105	053440	042510		
3041	011304	020116	054105	042520		
3042	011312	052103	047111	020107		
3043	011320	042122	020117	047524		
3044	011326	041040	020105	042523		
3045	011334	000124				
3046	011336	042122	020117	042527	MEF15:	.ASCIZ 'RDO WET WHEN EXPECTING RDI TO BE SET''
3047	011344	020124	044127	047105		
3048	011352	042440	050130	041505		
3049	011360	044524	043516	051040		
3050	011366	044504	052040	020117		

3051	011374	042502	051440	052105	
3052	011402	000			
3053	011403	111	041516	051117	MEF16A: .ASCIZ /INCORRECT CHARACTER COUNT RETURNED/
3054	011410	042522	052103	041440	
3055	011416	040510	040522	052103	
3056	011424	051105	041440	052517	
3057	011432	052116	051040	052105	
3058	011440	051125	042516	000104	
3059	011446	047111	047503	051122	MEF17A: .ASCIZ /INCORRECT REC BUFFER ADDR. RETURNED/
3060	011454	041505	020124	042522	
3061	011462	020103	052502	043106	
3062	011470	051105	040440	042104	
3063	011476	027122	051040	052105	
3064	011504	051125	042516	000104	
3065	011512	047111	047503	051122	MEF18A: .ASCIZ /INCORRECT TRIB NUMBER RETURNED/
3066	011520	041505	020124	051124	
3067	011526	041111	047040	046525	
3068	011534	042502	020122	042522	
3069	011542	052524	047122	042105	
3070	011550	000			
3071	011551	045	043501	047517	MEF18: .ASCII '%AGOOD= %03%A BAD= %03''
3072	011556	036504	022440	031517	
3073	011564	040445	041040	042101	
3074	011572	020075	047445	063	
3075	011577	045	022516	022524	.ASCIZ '%N%T%06''
3076	011604	033117	000		
3077	011607	115	051505	040523	MEF19A: .ASCIZ /MESSAGE DATA COMPARE ERROR/
3078	011614	042507	042040	052101	
3079	011622	020101	047503	050115	
3080	011630	051101	020105	051105	
3081	011636	047522	000122		
3082	011642	040445	044103	051101	MEF19: .ASCII '%A CHARACTER# %03%A SENT CODE''
3083	011650	041501	042524	021522	
3084	011656	022440	031517	040445	
3085	011664	051440	047105	020124	
3086	011672	047503	042504		
3087	011676	022440	031517	040445	.ASCIZ '' %03%A RECEIVED CODES %03''
3088	011704	051040	041505	044505	
3089	011712	042526	020104	047503	
3090	011720	042504	020123	047445	
3091	011726	000063			
3092	011730	042122	020117	046111	MEF30: .ASCIZ /RDO ILLEGALLY SET/
3093	011736	042514	040507	046114	
3094	011744	020131	042523	000124	
3095	011752	047522	020115	042526	MEF31: .ASCIZ /ROM VERSION INCORRECT/
3096	011760	051522	047511	020116	
3097	011766	047111	047503	051122	
3098	011774	041505	000124		
3099	012000	053440	047522	043516	MEF32: .ASCIZ /WRONG OPTION TYPE SELECTED IN TABLE/
3100	012006	047440	052120	047511	
3101	012014	020116	054524	042520	
3102	012022	051440	046105	041505	
3103	012030	042524	020104	047111	
3104	012036	052040	041101	042514	
3105	012044	000			
3106	012045	045	042101	050115	TFM20: .ASCII '%ADMP RETURNED NON-EXISTENT MEM ERR FOR ADD'

3107	012052	051040	052105	051125
3108	012060	042516	020104	047516
3109	012066	026516	054105	051511
3110	012074	042524	052116	046440
3111	012102	046505	042440	051122
3112	012110	043040	051117	040440
3113	012116	042104		
3114	012120	022440	031117	040445
3115	012126	030060	030060	030060
3116	012134	047045	040445	046455
3117	012142	046505	051117	020131
3118	012150	047504	051505	047040
3119	012156	052117	042440	044530
3120	012164	052123	000041	
3121	012170	040445	040504	040524
3122	012176	042440	051122	051117
3123	012204	044440	020116	051124
3124	012212	047101	043123	051105
3125	012220	052040	020117	042522
3126	012226	042503	053111	020105
3127	012234	052502	043106	051105
3128	012242	040440	020124	042101
3129	012250	051104	051505	020123
3130	012256	000		
3131	012257	045	031117	040445
3132	012264	030060	030060	030060
3133	012272	047045	040445	040504
3134	012300	040524	051440	047105
3135	012306	020124	020075	047445
3136	012314	066		
3137	012315	045	026101	042040
3138	012322	052101	020101	042522
3139	012330	042103	036440	022440
3140	012336	033117	000	
3141	012341	045	042101	050115
3142	012346	043040	044501	042514
3143	012354	020104	047524	042040
3144	012362	052105	041505	020124
3145	012370	047516	026516	054105
3146	012376	051511	042524	052116
3147	012404	046440	046505	051117
3148	012412	026131	040440	042104
3149	012420	042522	051523	000040
3150	012426	047445	022462	030101
3151	012434	030060	030060	022460
3152	012442	022516	020101	020127
3153	012450	020101	020122	020116
3154	012456	020111	020116	020107
3155	012464	050040	051101	051524
3156	012472	047440	020106	044124
3157	012500	051511	042040	040511
3158	012506	047107	051517	044524
3159	012514	000103		
3160	012516	040445	046440	054501
3161	012524	044040	053101	020105
3162	012532	042502	047105	022440

.ASCIZ ' %02%A0000000%N%A-MEMORY DOES NOT EXIST!'

TFM21: .ASCIZ '%ADATA ERROR IN TRANSFER TO RECEIVE BUFFER AT ADDRESS '

TFM2A: .ASCII '%02%A0000000%N%ADATA SENT = %06'

.ASCIZ '%A, DATA RECD = %06'

TFM22: .ASCIZ '%ADMP FAILED TO DETECT NON-EXISTENT MEMORY, ADDRESS '

TFM22A: .ASCIZ '%02%A0000000%N%A W A R N I N G PARTS OF THIS DIAGNOSTIC'

TFM22B: .ASCIZ '%A MAY HAVE BEEN %N%DESTROYED BY THE DMP TRANSFER'

3163	012540	022516	042101	051505	
3164	012546	051124	054517	042105	
3165	012554	041040	020131	044124	
3166	012562	020105	046504	020120	
3167	012570	051124	047101	043123	
3168	012576	051105	000041		
3169	012602	040445	051105	047522	TFM24: .ASCIZ '%AERROR IN ROM EX02%A READ = %06%A ; CALCULATED = %06''
3170	012610	020122	047111	051040	
3171	012616	046517	042440	047445	
3172	012624	022462	020101	042522	
3173	012632	042101	036440	022440	
3174	012640	033117	040445	035440	
3175	012646	041440	046101	052503	
3176	012654	040514	042524	020104	
3177	012662	020075	047445	000066	
3178	012670	040445	051105	047522	TFM25: .ASCII '%AERROR IN ROM EX02%A, ''
3179	012676	020122	047111	051040	
3180	012704	046517	042440	047445	
3181	012712	022462	026101	040	
3182	012717	123	047510	046125	.ASCIZ ''SHOULD BE ROM NO.: %01%A, NO. READ IS: %01''
3183	012724	020104	042502	051040	
3184	012732	046517	047040	027117	
3185	012740	020072	022440	030517	
3186	012746	040445	020054	047516	
3187	012754	020056	042522	042101	
3188	012762	044440	035123	022440	
3189	012770	030517	000		
3190	012773	045	022516	022524	DFMT4: .ASCIZ '/%N%T%N%T%N/'
3191	013000	022516	022524	000116	
3192	013006	047445	022463	032523	DFMT5: .ASCIZ '/%03%S5%03%S5%03%S5%03%N/'
3193	013014	047445	022463	032523	
3194	013022	047445	022463	032523	
3195	013030	047445	022463	000116	
3196	013036	051445	022464	031517	DFMT6: .ASCIZ '/%S4%03%S5%03%S5%03%S5%03%N/'
3197	013044	051445	022465	031517	
3198	013052	051445	022465	031517	
3199	013060	051445	022465	031517	
3200	013066	047045	000		
3201	013071	045	022524	000116	DFMT9: .ASCIZ '/%T%N/'
3202	013076	047045	040445	052440	ROMMSG: .ASCIZ '%N%A UNIT NUMBER. %D5%A ROM NUMBER IS: %T%A REV. NO. IS: %T''
3203	013104	044516	020124	052516	
3204	013112	041115	051105	020072	
3205	013120	042045	022465	020101	
3206	013126	047522	020115	052516	
3207	013134	041115	051105	044440	
3208	013142	035123	022440	022524	
3209	013150	020101	051040	053105	
3210	013156	020056	047516	020056	
3211	013164	051511	020072	052045	
3212	013172	000			
3213		013174			.EVEN
3214	013174	000	000		ROMNO: .BYTE 0.0
3215	013176	000	000		REVNO: .BYTE 0.0
3216					.EVEN

3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238

.SBTTL REPORT CODING SECTION

:++
: THE REPORT CODING SECTION CONTAINS THE
: 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
:--

013200

L\$RPT::

013200 000167
013202 000000

.WORD JSJMP
.WORD L10023-2-

013204
013204 104425

L10023:
TRAP C\$RPT

```
.SBTTL INITIALIZE SECTION

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

L$INIT::

3239
3240
3241
3242
3243
3244
3245
3246 013206
3247
3248 013206 010637 002320      MOV      SP,PSTACK      ;SAVE BASE-LEVEL STACK POINTER
3249 013212 005037 002322      CLR      SUBRPC        ;CLEAR SUBR CALL PC
3250 013216 005037 002324      CLR      ERROR1       ;CLEAR ERROR FLAGS
3251 013222 005737 002326      TST      FRSTIM        ;SEE IF FIRST TIME THROUGH AFTER LOAD
3252 013226 001007                BNE      6$            ;BR IF NOT
3253 013230 013737 000004 002314  MOV      @#4,SAVE4      ;SAVE ERROR TRAP VECTOR
3254 013236 013737 000006 002316  MOV      @#6,SAVE6
3255 013244 000406                BR       9$
3256 013246 013737 002314 000004 6$:     MOV      SAVE4,@#4      ;RESTORE ERROR TRAP VECTOR
3257 013254 013737 002316 000006  MOV      SAVE6,@#6
3258 013262 012737 000001 002326 9$:     MOV      #1,FRSTIM     ;MARK FLAG FOR NEXT TIME THROUGH
3259
;SEE IF PROGRAM JUST STARTED, BR IF YES
3260 013270 012700 000040      MOV      #EF.START,RO
3261 013274 104447                TRAP     C$REFG
3262 013276 103415                BCS     STARST
3263
;SEE IF PROGRAM JUST RESTARTED, BR IF YES
3264 013300 012700 000037      MOV      #EF.RESTART,RO
3265 013304 104447                TRAP     C$REFG
3266 013306 103411                BCS     STARST
3267
;SEE IF THIS IS A NEW PASS, BR IF YES
3268 013310 012700 000035      MOV      #EF.NEW,RO
3269 013314 104447                TRAP     C$REFG
3270 013316 103411                BCS     NEWST
3271
;SEE IF PROGRAM WAS JUST CONTINUED
3272 013320 012700 000036      MOV      #EF.CONTINUE,RO
3273 013324 104447                TRAP     C$REFG
3274 013326 103533                BCS     ENDIT
3275 013330 000416                BR       GETPRM
3276 013332
STARST:
3277 013332 005037 002344      CLR      STARES        ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
3278
;CLEAR DEVICE MAP
3279 013336 005037 002346      CLR      DEVMAP
3280 013342
NEWST:
3281 013342 012737 177777 002330  MOV      #-1,LOGDEV     ;RESET LOGICAL DEVICE TO -1
3282 013350 005237 002352                INC      FRSPAS        ;INCREMENT NO. OF PASSES AFTER LOAD
3283 013354 005237 002344                INC      STARES        ;INCREMENT NO. OF PASSES SINCE STA OR RES
3284 013360 012737 000001 002350  MOV      #BIT0,DEVPTR   ;INIT DEVICE MAP BIT POINTER
3285
; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
; CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
3286
GETPRM:
3287 013366
3288 013366 005237 002330                INC      LOGDEV        ;INCREMENT LOGICAL DEVICE NUMBER
3289 013372 023737 002330 002012  CMP      LOGDEV,L$UNIT ;SEE IF MAXIMUM UNIT NO. EXCEEDED
3290 013400 002360                BGE     NEWST         ;BR IF YES....
3291 013402 013700 002330                MOV      LOGDEV,RO
3292 013406 104442                TRAP     C$GPHRD
3293 013410 010001                MOV      RO,R1
3294 013412 103403                BCS     10$
```

3295	013414	006337	002350		ASL	DEVPTR	:SHIFT DEVICE MAP BIT POINTER
3296	013420	000762			BR	GETPRM	:SKIP THIS DEVICE
3297	013422			10\$:			
3298	013422	053737	002350	002346	BIS	DEVPTR,DEVMAP	:SHIFT DEVICE MAP BIT POINTER
3299	013430	006337	002350		ASL	DEVPTR	:
3300	013434	012137	002466		MOV	(R1)+,OPTYP	:SET THE OPTION TYPE
3301	013440	011137	002436		MOV	(R1),MPCSR	:STORE POINTER TO M8200,4,7 CSR'S
3302	013444	011137	002440		MOV	(R1),BSEL1	
3303	013450	005237	002440		INC	BSEL1	:GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
3304	013454	011137	002446		MOV	(R1),SEL4	
3305	013460	062737	000004	002446	ADD	#4,SEL4	:GET POINTER TO SEL4
3306	013466	011137	002442		MOV	(R1),SEL2	
3307	013472	062737	000002	002442	ADD	#2,SEL2	
3308	013500	011137	002444		MOV	(R1),BSEL3	
3309	013504	062737	000003	002444	ADD	#3,BSEL3	
3310	013512	011137	002450		MOV	(R1),BSEL5	
3311	013516	062737	000005	002450	ADD	#5,BSEL5	
3312	013524	011137	002454		MOV	(R1),BSEL7	
3313	013530	062737	000007	002454	ADD	#7,BSEL7	
3314	013536	012137	002452		MOV	(R1)+,SEL6	
3315	013542	062737	000006	002452	ADD	#6,SEL6	:STORE POINTER TO SEL6
3316	013550	011137	002456		MOV	(R1),MPIVEC	:GET M8200,4,7 INPUT INTRPT VECTOR
3317	013554	012137	002460		MOV	(R1)+,MPOVEC	
3318	013560	062737	000004	002460	ADD	#4,MPOVEC	:GET M8200,4,7 OUTPUT INTRPT VECTOR
3319	013566	012137	002464		MOV	(R1)+,MPRIOR	:GET M8200,4,7 DEVICE PRIORITY
3320	013572	062701	000010		ADD	#10,R1	:POINT TO TEST CON
3321	013576	012137	002472		MOV	(R1)+,TSTCON	:GET TEST CONNECTOR INDICATOR
3322	013602	012137	002462		MOV	(R1)+,SPEEDM	:GET SPEED
3323	013606	012137	002470		MOV	(R1)+,IFTYP	
3324	013612	012137	002470		MOV	(R1)+,IFTYP	:FIRST TIME SKIP RUN WORD;THEN LOAD
3325							:INTERFACE TYPE
3326	013616			ENDIT:			
3327	013616			L10024:			
3328	013616	104411			TRAP	C\$INIT	
3329							
3330							
3331	013620			L\$AUTO::			
3332	013620	004737	004404		JSR	PC,MINITR	:INITIALIZE
3333	013624	005037	002340		CLR	COUNT	:+COUNTER
3334							
3335	013630	004737	004172	1\$:	JSR	PC,WAIT50	:STALL
3336	013634	005777	166576		TST	@SELO	:HAS IT STARTED?
3337	013640	100406			BMI	4\$	
3338	013642	105337	002340		DECB	COUNT	:TIME UP?
3339	013646	001370			BNE	1\$:NO LOOP
3340	013650	013700	002330		MOV	LOGDEV,R0	
3341	013654	104451			TRAP	C\$DODU	
3342							
3343	013656			4\$:			
3344	013656	104433			TRAP	C\$RESET	
3345	013660			L10025:			
3346	013660	104461			TRAP	C\$AUTO	
3347							

3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364

.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.  
:/
```

L\$CLEAN::
TRAP C\$RESET

L10026:
TRAP C\$CLEAN

013662
013662 104433

013664
013664 104412

3365
3366
3367
3368
3369
3370
3371
3372 013666
3373
3374 013666 104433
3375 013670
3376 013670 104453
3377
3378
3379
3380
3381

.SBTTL DROP UNIT SECTION
:////////////////////
:// THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
:// TO NO LONGER BE TESTED.
:////////////////////
L\$DU: :
:ISSUE UNIBUS RESET TO CLEAN UP
TRAP C\$RESET
L10027: TRAP C\$DU

3382
3383
3384
3385
3386
3387
3388
3389
3390 013672
3391 013672
3392 013672 104452
3393

.SBTTL ADD UNIT SECTION

:/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
:/ TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
:/ 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
:/

LSAU::
L'0030: TRAP CSAU

CZDMTAD DMP/V-11 PCTNL TST #1 MACY11 30A(1052) 12-AUG-80 15:53 M 6
CZDMTA.REL 12-AUG-80 15:52 ADD UNIT SECTION PAGE 78

SEQ 0077

3394
3395
3396
3397
3398

3399
3400
3401
3402
3403
3404
3405
3406
3407 013674
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419 013674
3420 013674 012746 000340
3421 013700 012746 014006
3422 013704 012746 000004
3423 013710 012746 000003
3424 013714 104437
3425 013716 062706 000010
3426 013722 005037 014004
3427 013726 005001
3428 013730 005777 166502
3429 013734 012701 000002
3430 013740 005777 166476
3431 013744 012701 000004
3432 013750 005777 166472
3433 013754 012701 000006
3434 013760 005777 166466
3435 013764 005737 014004
3436 013770 001401
3437 013772 104444
3438 013774
3439 013774 012700 000004
3440 014000 104436
3441
3442 014002
3443 014002 104401
3444
3445 014004 000000
3446
3447 014006
3448 014006 005737 014004
3449 014012 001006
3450 014014 104455
3451 014016 000023
3452 014020 010216
3453 014022 006666
3454 014024 005237 014004

.SBTTL HARDWARE TESTS

.SBTTL ;***** TEST 1 *****
.SBTTL * ADDRES TEST-VERIFY THAT ALL MCPU ADDRESSES RESPOND
ZZ
;*ECB
*
* THIS IS THE VERY FIRST TEST IN NORMAL SEQUENCE
* IT IS USED TO VERIFY THAT DMP OR DMV-11 UNDER TEST, RESPONDS
* TO THE ADDRESS THAT YOU THINK IT IS AT. ON DMP FAILURE CHECK
* ADDRESS SWITCHES ON THE M8207 MICRO-CPU. WITH LITTLE
* DOUBT, THIS FAILURE CAN ONLY BE ATTRIBUTED TO THE M8207 BOARD.
* NOTE:8207 IS DMP ONLY.....
*
*
*-

.SBTTL ;***** TEST 1 *****

T1: :
MOV #PRI07,-(SP)
MOV #ECBINT,-(SP)
MOV #4,-(SP)
MOV #3,-(SP)
TRAP C\$SVEC
ADD #10,SP
CLR JMO ;CLEAR FLAG
CLR R1
TST @SELO ;TEST CSR 0
MOV #2,R1 ;SAVE OFFSET FOR NEXT CSR
TST @SEL2 ;TEST CSR 2
MOV #4,R1 ;SAVE OFFSET
TST @SEL4 ;TEST CSR 4
MOV #6,R1 ;SAVE OFFSET
TST @SEL6 ;TEST CSR 6
TST JMO ;WAS THERE A NXM TRAP
BEQ 10\$;IF NOT EXIT CLEANLY
TRAP C\$DCLN
10\$:
MOV #4,R0
TRAP C\$CVEC
L10031:
TRAP C\$ETST
JMO: .WORD 0 ;FLAG FOR O'CONNOR CODE
ECBINT: :
TST JMO ;HAVE WE HAD AT LEAST 1 TRAP
BNE 10\$
TRAP C\$ERDF
.WORD 19
.WORD EMT0
.WORD ERR1
INC JMO ;SET FLAG

3455 014030
3456 014030 010146
3457 014032 012746 010244
3458 014036 012746 000002
3459 014042 010600
3460 014044 104415
3461 014046 062706 000006
3462 014052
3463 014052 000002
3464
3465
3466

10\$:
MOV R1,-(SP)
MOV #FMT0,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #6,SP
L10032:
RTI

```
3467 .SBTTL :***** TEST 2 *****
3468 .SBTTL *DMP ONLY VERIFY CONTENTS OF ROM 3
3469 014054 ZZ
3470 :* THIS TEST DONE FOR DMP ONLY
3471 :*
3472 :* IN THIS TEST WE'LL VERIFY THE CONTENTS OF ROM 3
3473 :*
3474 .SBTTL :***** TEST 2 *****
3475 ;:-CROMT-
3476 014054 T2::
3477 014054 022737 000000 002466 CMP #0,OPTYP ;IS THIS AN 8207 DMP
3478 014062 001150 BNE 60$ ;IF NOT END.....
3479 014064 012737 000003 002364 MOV #3,ROMN ;ROM NUMBER
3480 014072 012737 000000 002376 MOV #0,CADDR ;GET STARTING ADDR.
3481
3482 014100 012737 177777 002372 MOV #-1,CWORD ;INIT CRC WORD.
3483
3484 014106 004737 004222 10$: JSR PC,GWORD ;GET FIRST BYTE.
3485 014112 117737 166334 002370 MOVB @SEL6,WORDT ;STORE FIRST BYTE.
3486 014120 005237 002376 INC CADDR ;UPDATE ADDR.
3487 014124 004737 004222 JSR PC,GWORD ;GET NEXT BYTE.
3488 014130 117737 166316 002371 MOVB @SEL6,WORDT+1 ;STORE IN HIGH BYTE OF WORDT
3489 014136 005237 002376 INC CADDR ;UPDATE ADDR.
3490 014142 023727 002376 004000 CMP CADDR,#3777+1 ;AT END?
3491 014150 001403 BEQ 20$ ;YES,EXIT LOOP.
3492
3493 014152 004737 004324 JSR PC,CRCR ;NO-CALCULATE CRC ON THIS WORD.
3494 014156 000753 BR 10$ ;LOOP.
3495
3496 014160 005137 002372 20$: COM CWORD ;STORED CRC WORD IS COMPLEMENT.
3497 014164 023737 002372 002370 CMP CWORD,WORDT ;EQUAL?
3498 014172 001404 BEQ 30$
3499
3500 ;ROM CRC WORD BAD.
3501 014174 104455 TRAP C$ERDF
3502 014176 000024 .WORD 20
3503 014200 000000 .WORD 0
3504 014202 007432 .WORD ERR24
3505 014204 012737 003775 002376 30$: MOV #3777-2,CADDR ;SET ROM NUMBER ADDRESS
3506 014212 012737 000060 002366 MOV #60,ROMN1 ;ROM NUMBER
3507 014220 004737 004222 JSR PC,GWORD ;READ ROM NUMBER
3508 014224 117737 166222 002370 MOVB @SEL6,WORDT ;STORE BYTE
3509 014232 123737 002366 002370 CMPB ROMN1,WORDT ;GOOD?
3510 014240 001404 BEQ 40$
3511 014242 104455 TRAP C$ERDF
3512 014244 000025 .WORD 21
3513 014246 000000 .WORD 0
3514 014250 007474 .WORD ERR25
3515 014252
3516 014252 022737 000001 002344 40$: CMP #1,STARES ;IS THIS FIRST PASS
3517 014260 001031 BNE 50$ ;IF NOT THEN GO TO 50
3518 014262 113737 002370 013174 MOVB WORDT,ROMNO ;PUT ROM NO IN PRINT CONDITION
3519 014270 012737 003774 002376 MOV #3777-3,CADDR
3520 014276 004737 004222 JSR PC,GWORD ;READ REV NO.
3521 014302 117737 166144 013176 MOVB @SEL6,REVNO ;STORE BYTE
3522 014310 012746 013176 MCV #RFVNO,-(SP)
```

```

3523 014314 012746 013174      MOV      #ROMNO,-(SP)
3524 014320 013746 002330      MOV      LOGDEV,-(SP)
3525 014324 012746 013076      MOV      #ROMMSG,-(SP)
3526 014330 012746 000004      MOV      #4,-(SP)
3527 014334 010600      MOV      SP,R0
3528 014336 104417      TRAP     C$PNTF
3529 014340 062706 000012      ADD      #12,SP
3530 014344 012737 003773 002376 50$:      MOV      #3777-4,CADDR      ;GET VERSION
3531 014352 004737 004222      JSR      PC,GWORD          ;READ IT
3532 014356 117737 166070 002370      MOV      @SEL6,WORDT
3533 014364 122737 000131 002370      CMPB     #131,WORDT
3534 014372 001404      BEQ      60$
3535 014374 104455      TRAP     C$ERDF
3536 014376 000026      .WORD   22
3537 014400 011752      .WORD   MEF31
3538 014402 007672      .WORD   ERR32
3539 014404      60$:
3540 014404      L10033:
3541 014404 104401      TRAP     C$ETST
  
```

```
3542 .SBTTL ;***** TEST 3 *****
3543 .SBTTL ;DMP ONLY VERIFY CONTENTS OF ROM 2
3544 014406 22
3545 ;* THIS TEST DONE FOR DMP ONLY
3546 ;*
3547 ;* IN THIS TEST WE'LL VERIFY THE CONTENTS OF ROM 2
3548 ;*
3549 .SBTTL ;***** TEST 3 *****
3550 ;-CROMT-
3551 014406 T3::
3552 014406 022737 000000 002466 CMP #0,OPTYP ;IS THIS AN 8207 DMP
3553 014414 001150 BNE 60$ ;IF NOT END.....
3554 014416 012737 000002 002364 MOV #2,ROMN ;ROM NUMBER
3555 014424 012737 000000 002376 MOV #0,CADDR ;GET STARTING ADDR.
3556
3557 014432 012737 177777 002372 MOV #-1,CWORD ;INIT CRC WORD.
3558
3559 014440 004737 004222 10$: JSR PC,GWORD ;GET FIRST BYTE.
3560 014444 117737 166004 002370 MOVB @BSEL7,WORDT ;STORE FIRST BYTE.
3561 014452 005237 002376 INC CADDR ;UPDATE ADDR.
3562 014456 004737 004222 JSR PC,GWORD ;GET NEXT BYTE.
3563 014462 117737 165766 002371 MOVB @BSEL7,WORDT+1 ;STORE IN HIGH BYTE OF WORDT.
3564 014470 005237 002376 INC CADDR ;UPDATE ADDR.
3565 014474 023727 002376 004000 CMP CADDR,#3777+1 ;AT END?
3566 014502 001403 BEQ 20$ ;YES,EXIT LOOP.
3567
3568 014504 004737 004324 JSR PC,CRCR ;NO-CALCULATE CRC ON THIS WORD.
3569 014510 000753 BR 10$ ;LOOP.
3570
3571 014512 005137 002372 20$: COM CWORD ;STORED CRC WORD IS COMPLEMENT.
3572 014516 023737 002372 002370 CMP CWORD,WORDT ;EQUAL?
3573 014524 001404 BEQ 30$
3574
3575 ;ROM CRC WORD BAD.
3576 014526 104455 TRAP C$ERDF
3577 014530 000024 .WORD 20
3578 014532 000000 .WORD 0
3579 014534 007432 .WORD ERR24
3580 014536 012737 003775 002376 30$: MOV #3777-2,CADDR ;SET ROM NUMBER ADDRESS
3581 014544 012737 000061 002366 MOV #61,ROMN1 ;ROM NUMBER
3582 014552 004737 004222 JSR PC,GWORD ;READ ROM NUMBER
3583 014556 117737 165672 002370 MOVB @BSEL7,WORDT ;STORE BYTE
3584 014564 123737 002366 002370 CMPB ROMN1,WORDT ;GOOD?
3585 014572 001404 BEQ 40$
3586 014574 104455 TRAP C$ERDF
3587 014576 000025 .WORD 21
3588 014600 000000 .WORD 0
3589 014602 007474 .WORD ERR25
3590 014604 40$:
3591 014604 022737 000001 002344 CMP #1,STARES ;IS THIS FIRST PASS
3592 014612 001031 BNE 50$ ;IF NOT THEN GO TO 50
3593 014614 113737 002370 013174 MOVB WORDT,ROMNO ;PUT ROM NO IN PRINT CONDITION
3594 014622 012737 003774 002376 MOV #3777-3,CADDR
3595 014630 004737 004222 JSR PC,GWORD ;READ REV NO.
3596 014634 117737 165614 013176 MOVB @BSEL7,REVNO ;STORE BYTE
3597 014642 012746 013176 MOV #REVNO,-(SP)
```

3598	014646	012746	013174			MOV	#ROMNO,-(SP)		
3599	014652	013746	002330			MOV	LOGDEV,-(SP)		
3600	014656	012746	013076			MOV	#ROMMSG,-(SP)		
3601	014662	012746	000004			MOV	#4,-(SP)		
3602	014666	010600				MOV	SP,R0		
3603	014670	104417				TRAP	C\$PNTF		
3604	014672	062706	000012			ADD	#12,SP		
3605	014676	012737	003773	002376	50\$:	MOV	#3777-4,CADDR		:GET VERSION
3606	014704	004737	004222			JSR	PC,GWORD		:READ IT
3607	014710	117737	165540	002370		MOVB	@BSEL7,WORDT		
3608	014716	122737	000131	002370		CMPB	#131,WORDT		
3609	014724	001404				BEQ	60\$		
3610	014726	104455				TRAP	C\$ERDF		
3611	014730	000026				.WORD	22		
3612	014732	011752				.WORD	MEF31		
3613	014734	007672				.WORD	ERR32		
3614	014736				60\$:				
3615	014736				L10034:				
3616	014736	104401				TRAP	C\$ETST		

```
3617 .SBTTL :***** TEST 4 *****
3618 .SBTTL *DMP ONLY VERIFY CONTENTS OF ROM 4
3619 014740 ZZ
3620 :* THIS TEST DONE FOR DMP ONLY
3621 :*
3622 :* IN THIS TEST WE'LL VERIFY THE CONTENTS OF ROM 4
3623 :*
3624 .SBTTL :***** TEST 4 *****
3625 :*-CROMT-
3626 014740 T4::
3627 014740 022737 000000 002466 CMP #0,OPTYP ;IS THIS AN 8207 DMP
3628 014746 001150 BNE 60$ ;IF NOT END.....
3629 014750 012737 000004 002364 MOV #4,ROMN ;ROM NUMBER
3630 014756 012737 004000 002376 MOV #4000,CADDR ;GET STARTING ADDR.
3631
3632 014764 012737 177777 002372 MOV #-1,CWORD ;INIT CRC WORD.
3633
3634 014772 004737 004222 10$: JSR PC,GWORD ;GET FIRST BYTE.
3635 014776 117737 165450 002370 MOVB @SEL6,WORDT ;STORE FIRST BYTE.
3636 015004 005237 002376 INC CADDR ;UPDATE ADDR.
3637 015010 004737 004222 JSR PC,GWORD ;GET NEXT BYTE.
3638 015014 117737 165432 002371 MOVB @SEL6,WORDT+1 ;STORE IN HIGH BYTE OF WORDT
3639 015022 005237 002376 INC CADDR ;UPDATE ADDR.
3640 015026 023727 002376 010000 CMP CADDR,#7777+1 ;AT END?
3641 015034 001403 BEQ 20$ ;YES,EXIT LOOP.
3642
3643 015036 004737 004324 JSR PC,CRCR ;NO-CALCULATE CRC ON THIS WORD.
3644 015042 000753 BR 10$ ;LOOP.
3645
3646 015044 005137 002372 20$: COM CWORD ;STORED CRC WORD IS COMPLEMENT.
3647 015050 023737 002372 002370 CMP CWORD,WORDT ;EQUAL?
3648 015056 001404 BEQ 30$
3649
3650 ;ROM CRC WORD BAD.
3651 015060 104455 TRAP C$ERDF
3652 015062 000024 .WORD 20
3653 015064 000000 .WORD 0
3654 015066 007432 .WORD ERR24
3655 015070 012737 007775 002376 30$: MOV #7777-2,CADDR ;SET ROM NUMBER ADDRESS
3656 015076 012737 000062 002366 MOV #62,ROMN1 ;ROM NUMBER
3657 015104 004737 004222 JSR PC,GWORD ;READ ROM NUMBER
3658 015110 117737 165336 002370 MOVB @SEL6,WORDT ;STORE BYT.
3659 015116 123737 002366 002370 CMPB ROMN1,WORDT ;GOOD?
3660 015124 001404 BEQ 40$
3661 015126 104455 TRAP C$ERDF
3662 015130 000025 .WORD 21
3663 015132 000000 .WORD 0
3664 015134 007474 .WORD ERR25
3665 015136
3666 015136 022737 000001 002344 40$: CMP #1,STARES ;IS THIS FIRST PASS
3667 015144 001031 BNE 50$ ;IF NOT THEN GO TO 50
3668 015146 113737 002370 013174 MOVB WORDT,ROMNO ;PUT ROM NO IN PRINT CONDITION
3669 015154 012737 007774 002376 MOV #7777-3,CADDR
3670 015162 004737 004222 JSR PC,GWORD ;READ REV NO.
3671 015166 117737 165260 013176 MOVB @SEL6,REVNO ;STORE BYTE
3672 015174 012746 013176 MOV #REVNO,-(SP)
```

3673	015200	012746	013174			MOV	#ROMNO,-(SP)	
3674	015204	013746	002330			MOV	LOGDEV,-(SP)	
3675	015210	012746	C13076			MOV	#ROMMSG,-(SP)	
3676	015214	012746	000004			MOV	#4,-(SP)	
3677	015220	010600				MOV	SP,R0	
3678	015222	104417				TRAP	C\$PNTF	
3679	015224	062706	000012			ADD	#12,SP	
3680	015230	012737	007773	002376	50\$:	MOV	#7777-4,CADDR	;GET VERSION
3681	015236	004737	004222			JSR	PC,GWORD	;READ IT
3682	015242	117737	165204	002370		MOVB	@SEL6,WORDT	
3683	015250	122737	000131	002370		CMPB	#131,WORDT	
3684	015256	001404				BEQ	60\$	
3685	015260	104455				TRAP	C\$ERDF	
3686	015262	000026				.WORD	22	
3687	015264	011752				.WORD	MEF31	
3688	015266	007672				.WORD	ERR32	
3689	015270				60\$:			
3690	015270				L10035:			
3691	015270	104401				TRAP	C\$ETST	


```
3692 .SBTTL :***** TEST 5 *****
3693 .SBTTL *DMP ONLY VERIFY CONTENTS OF ROM 1
3694 015272 ZZ
3695 :* THIS TEST DONE FOR DMP ONLY
3696 :*
3697 :* IN THIS TEST WE'LL VERIFY THE CONTENTS OF ROM 1
3698 :*
3699 .SBTTL :***** TEST 5 *****
3700 : -CROMT-
3701 015272 T5::
3702 015272 022737 000000 002466 CMP #0,OPTYP ;IS THIS AN 8207 DMP
3703 015300 001150 BNE 60$ ;IF NOT END.....
3704 015302 012737 000001 002364 MOV #1,ROMN ;ROM NUMBER
3705 015310 012737 004000 002376 MOV #4000,CADDR ;GET STARTING ADDR.
3706
3707 015316 012737 177777 002372 MOV #-1,CWORD ;INIT CRC WORD.
3708
3709 015324 004737 004222 10$: JSR PC,GWORD ;GET FIRST BYTE.
3710 015330 117737 165120 002370 MOV @BSEL7,WORDT ;STORE FIRST BYTE.
3711 015336 005237 002376 INC CADDR ;UPDATE ADDR.
3712 015342 004737 004222 JSR PC,GWORD ;GET NEXT BYTE.
3713 015346 117737 165102 002371 MOV @BSEL7,WORDT+1 ;STORE IN HIGH BYTE OF WORDT.
3714 015354 005237 002376 INC CADDR ;UPDATE ADDR.
3715 015360 023727 002376 010000 CMP CADDR,#7777+1 ;AT END?
3716 015366 001403 BEQ 20$ ;YES,EXIT LOOP.
3717
3718 015370 004737 004324 JSR PC,CRCR ;NO-CALCULATE CRC ON THIS WORD.
3719 015374 000753 BR 10$ ;LOOP.
3720
3721 015376 005137 002372 20$: COM CWORD ;STORED CRC WORD IS COMPLEMENT.
3722 015402 023737 002372 002370 CMP CWORD,WORDT ;EQUAL?
3723 015410 001404 BEQ 30$
3724
3725 ;ROM CRC WORD BAD.
3726 015412 104455 TRAP C$ERDF
3727 015414 000024 .WORD 20
3728 015416 000000 .WORD 0
3729 015420 007432 .WORD ERR24
3730 015422 012737 007775 002376 30$: MOV #7777-2,CADDR ;SET ROM NUMBER ADDRESS
3731 015430 012737 000063 002366 MOV #63,ROMN1 ;ROM NUMBER
3732 015436 004737 004222 JSP PC,GWORD ;READ ROM NUMBER
3733 015442 117737 165006 002370 MOV @BSEL7,WORDT ;STORE BYTE
3734 015450 123737 002366 002370 CMPB ROMN1,WORDT ;GOOD?
3735 015456 001404 BEQ 40$
3736 015460 104455 TRAP C$ERDF
3737 015462 000025 .WORD 21
3738 015464 000000 .WORD 0
3739 015466 007474 .WORD ERR25
3740 015470 40$:
3741 015470 022737 000001 002344 CMP #1,STARES ;IS THIS FIRST PASS
3742 015476 001031 BNE 50$ ;IF NOT THEN GO TO 50
3743 015500 113737 002370 013174 MOVB WORDT,ROMNO ;PUT ROM NO IN PRINT CONDITION
3744 015506 012737 007774 002376 MOV #7777-3,CADDR
3745 015514 004737 004222 JSR PC,GWORD ;READ REV NO.
3746 015520 117737 164730 013176 MOV @BSEL7,REVNO ;STORE BYTE
3747 015526 012746 013176 MOV #REVNO,-(SP)
```

CZDMTAD DMP/V-11 rCTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 88

TEST 5

SEQ 0087

```

3748 015532 012746 013174      MOV      #ROMNO,-(SP)
3749 015536 013746 002330      MOV      LGGDEV,-(SP)
3750 015542 012746 013076      MOV      #ROMMSG,-(SP)
3751 015546 012746 000004      MOV      #4,-(SP)
3752 015552 010600      MOV      SP,R0
3753 015554 104417      TRAP     C$PNTF
3754 015556 062706 000012      ADD      #12,SP
3755 015562 012737 007773 002376 50$:      MOV      #7777-4,CADDR      :GET VERSION
3756 015570 004737 004222      JSR      PC,GWORD          :READ IT
3757 015574 117737 164654 002370      MOVB     @BSEL7,WORDT
3758 015602 122737 000131 002370      CMPB     #131,WORDT
3759 015610 001404      BEQ      60$
3760 015612 104455      TRAP     C$ERDF
3761 015614 000026      .WORD   22
3762 015616 011752      .WORD   MEF31
3763 015620 007672      .WORD   ERR32
3764 015622      60$:
3765 015622      L10036:
3766 015622 104401      TRAP     C$ETST

```

```
3767 .SBTTL :***** TEST 6 *****
3768 .SBTTL *DMP ONLY VERIFY CONTENTS OF ROM 5
3769 015624 ZZ
3770 :* THIS TEST DONE FOR DMP ONLY
3771 :*
3772 :* IN THIS TEST WE'LL VERIFY THE CONTENTS OF ROM 5
3773 :*
3774 .SBTTL :***** TEST 6 *****
3775 :*-CROMT-
3776 015624 T6::
3777 015624 022737 000000 002466 CMP #0,OPTYP ;IS THIS AN 8207 DMP
3778 015632 001150 BNE 60$ ;IF NOT END.....
3779 015634 012737 000005 002364 MOV #5,ROMN ;ROM NUMBER
3780 015642 012737 010000 002376 MOV #10000,CADDR ;GET STARTING ADDR.
3781
3782 015650 012737 177777 002372 MOV #-1,CWORD ;INIT CRC WORD.
3783
3784 015656 004737 004222 10$: JSR PC,GWORD ;GET FIRST BYTE.
3785 015662 117737 164564 002370 MOVB @SEL6,WORDT ;STORE FIRST BYTE.
3786 015670 005237 002376 INC CADDR ;UPDATE ADDR.
3787 015674 004737 004222 JSR PC,GWORD ;GET NEXT BYTE.
3788 015700 117737 164546 002371 MOVB @SEL6,WORDT+1 ;STORE IN HIGH BYTE OF WORDT
3789 015706 005237 002376 INC CADDR ;UPDATE ADDR.
3790 015712 023727 002376 014000 CMP CADDR,#13777+1 ;AT END?
3791 015720 001403 BEQ 20$ ;YES,EXIT LOOP.
3792
3793 015722 004737 004324 JSR PC,CRCR ;NO-CALCULATE CRC ON THIS WORD.
3794 015726 000753 BR 10$ ;LOOP.
3795
3796 015730 005137 002372 20$: COM CWORD ;STORED CRC WORD IS COMPLEMENT.
3797 015734 023737 002372 002370 CMP CWORD,WORDT ;EQUAL?
3798 015742 001404 BEQ 30$
3799
3800 ;ROM CRC WORD BAD.
3801 015744 104455 TRAP C$ERDF
3802 015746 000024 .WORD 20
3803 015750 000000 .WORD 0
3804 015752 007432 .WORD ERR24
3805 015754 012737 013775 002376 30$: MOV #13777-2,CADDR ;SET ROM NUMBER ADDRESS
3806 015762 012737 000064 002366 MOV #64,ROMN1 ;ROM NUMBER
3807 015770 004737 004222 JSR PC,GWORD ;READ ROM NUMBER
3808 015774 117737 164452 002370 MOVB @SEL6,WORDT ;STORE BYTE
3809 016002 123737 002366 002370 CMPB ROMN1,WORDT ;GOOD?
3810 016010 001404 BEQ 40$
3811 016012 104455 TRAP C$ERDF
3812 016014 000025 .WORD 21
3813 016016 000000 .WORD 0
3814 016020 007474 .WORD ERR25
3815 016022
3816 016022 022737 000001 002344 40$: CMP #1,STARES ;IS THIS FIRST PASS
3817 016030 001031 BNE 50$ ;IF NOT THEN GO TO 50
3818 016032 113737 002370 013174 MOVB WORDT,ROMNO ;PUT ROM NO IN PRINT CONDITION
3819 016040 012737 013774 002376 MOV #13777-3,CADDR
3820 016046 004737 004222 JSR PC,GWORD ;READ REV NO.
3821 016052 117737 164374 013176 MOVB @SEL6,REVNO ;STORE BYTE
3822 016060 012746 013176 MOV #REVNO,-(SP)
```

```

3823 016064 012746 013174      MOV      #ROMNO,-(SP)
3824 016070 013746 002330      MOV      LOGDEV,-(SP)
3825 016074 012746 013076      MOV      #ROMMSG,-(SP)
3826 016100 012746 000004      MOV      #4,-(SP)
3827 016104 010600              MOV      SP,R0
3828 016106 104417              TRAP     C$PNTF
3829 016110 062706 000012      ADD      #12,SP
3830 016114 012737 013773 002376 50$:  MOV      #13777-4,CADDR      ;GET VERSION
3831 016122 004737 004222              JSR      PC,GWORD          ;READ IT
3832 016126 117737 164320 002370      MOV      @SEL6,WORDT
3833 016134 122737 000131 002370      CMPB    #131,WORDT
3834 016142 001404              BEQ      60$
3835 016144 104455              TRAP     C$ERDF
3836 016146 000026              .WORD   22
3837 016150 011752              .WORD   MEF31
3838 016152 007672              .WORD   ERR32
3839 016154              60$:
3840 016154              L10037:
3841 016154 104401              TRAP     C$ETST
  
```

```
3842 .SBTTL :***** TEST 7 *****
3843 .SBTTL *DMP ONLY VERIFY CONTENTS OF ROM 14
3844 016156 ZZ
3845 :* THIS TEST DONE FOR DMP ONLY
3846 :*
3847 :* IN THIS TEST WE'LL VERIFY THE CONTENTS OF ROM 14
3848 :*
3849 .SBTTL :***** TEST 7 *****
3850 :-CROMT-
3851 016156 T7::
3852 016156 022737 000000 002466 CMP #0,OPTYP ;IS THIS AN 8207 DMP
3853 016164 001150 BNE 60$ ;IF NOT END.....
3854 016166 012737 000014 002364 MOV #14,ROMN ;ROM NUMBER
3855 016174 012737 010000 002376 MOV #10000,CADDR ;GET STARTING ADDR.
3856
3857 016202 012737 177777 002372 MOV #-1,CWORD ;INIT CRC WORD.
3858
3859 016210 004737 004222 10$: JSR PC,GWORD ;GET FIRST BYTE.
3860 016214 117737 164234 002370 MOV @BSEL7,WORDT ;STORE FIRST BYTE.
3861 016222 005237 002376 INC CADDR ;UPDATE ADDR.
3862 016226 004737 004222 JSR PC,GWORD ;GET NEXT BYTE.
3863 016232 117737 164216 002371 MOV @BSEL7,WORDT+1 ;STORE IN HIGH BYTE OF WORDT.
3864 016240 005237 002376 INC CADDR ;UPDATE ADDR.
3865 016244 023727 002376 014000 CMP CADDR,#13777+1 ;AT END?
3866 016252 001403 BEQ 20$ ;YES,EXIT LOOP.
3867
3868 016254 004737 004324 JSR PC,CRCR ;NO-CALCULATE CRC ON THIS WORD.
3869 016260 000753 BR 10$ ;LOOP.
3870
3871 016262 005137 002372 20$: COM CWORD ;STORED CRC WORD IS COMPLEMENT.
3872 016266 023737 002372 002370 CMP CWORD,WORDT ;EQUAL?
3873 016274 001404 BEQ 30$
3874
3875 ;ROM CRC WORD BAD.
3876 016276 104455 TRAP C$ERDF
3877 016300 000024 .WORD 20
3878 016302 000000 .WORD 0
3879 016304 007432 .WORD ERR24
3880 016306 012737 013775 002376 30$: MOV #13777-2,CADDR ;SET ROM NUMBER ADDRESS
3881 016314 012737 000065 002366 MOV #65,ROMN1 ;ROM NUMBER
3882 016322 004737 004222 JSR PC,GWORD ;READ ROM NUMBER
3883 016326 117737 164122 002370 MOV @BSEL7,WORDT ;STORE BYTE
3884 016334 123737 002366 002370 CMPB ROMN1,WORDT ;GOOD?
3885 016342 001404 BEQ 40$
3886 016344 104455 TRAP C$ERDF
3887 016346 000025 .WORD 21
3888 016350 000000 .WORD 0
3889 016352 007474 .WORD ERR25
3890 016354 40$:
3891 016354 022737 000001 002344 CMP #1,STARES ;IS THIS FIRST PASS
3892 016362 001031 BNE 50$ ;IF NOT THEN GO TO 50
3893 016364 113737 002370 013174 MOVB WORDT,ROMNO ;PUT ROM NO IN PRINT CONDITION
3894 016372 012737 013774 002376 MOV #13777-3,CADDR
3895 016400 004737 004222 JSR PC,GWORD ;READ REV NO.
3896 016404 117737 164044 013176 MOV @BSEL7,REVNO ;STORE BYTE
3897 016412 012746 013176 MOV #REVNO,-(SP)
```

CZDMTAD DMP/V-11 FCTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 92

***** TEST 7 *****

SEQ 0091

3898	016416	012746	013174			MOV	#ROMNO,-(SP)	
3899	016422	013746	002330			MOV	LOGDEV,-(SP)	
3900	016426	012746	013076			MOV	#ROMMSG,-(SP)	
3901	016432	012746	000004			MOV	#4,-(SP)	
3902	016436	010600				MOV	SP,R0	
3903	016440	104417				TRAP	C\$PNTF	
3904	016442	062706	000012			ADD	#12,SP	
3905	016446	012737	013773	002376	50\$:	MOV	#13777-4,CADDR	;GET VERSION
3906	016454	004737	004222			JSR	PC,GWORD	;READ IT
3907	016460	117737	163770	002370		MOVB	@BSEL7,WORDT	
3908	016466	122737	000131	002370		CMPB	#131,WORDT	
3909	016474	001404				BEQ	60\$	
3910	016476	104455				TRAP	C\$ERDF	
3911	016500	000026				.WORD	22	
3912	016502	011752				.WORD	MEF31	
3913	016504	007672				.WORD	ERR32	
3914	016506				60\$:			
3915	016506				L10040:			
3916	016506	104401				TRAP	C\$ETST	

3917
3918
3919 016510
3920
3921
3922
3923
3924
3925
3926
3927 016510
3928 016510 032737 000003 002466
3929 016516 001501
3930 016520 012737 000000 002366
3931 016526 012737 000000 002432
3932 016534 012737 000000 002434
3933 016542 004537 005326
3934 016546 005037 002374
3935 016552 104410
3936 016554 000146
3937 016556 022737 000001 002344
3938 016564 001056
3939
3940 016566 012737 000000 002432
3941 016574 012737 000000 002434
3942 016602 004537 005326
3943 016606 005037 002374
3944 016612 104410
3945 016614 000106
3946 016616 012702 000000
3947 016622 116237 033022 013174
3948 016630 012702 000000
3949 016634 116237 033022 013176
3950 016642 023737 013174 002366
3951 016650 001406
3952 016652 104455
3953 016654 000027
3954 016656 000000
3955 016660 007474
3956
3957 016662 104410
3958 016664 000036
3959 016666
3960 016666 012746 013176
3961 016672 012746 013174
3962 016676 013746 002330
3963 016702 012746 013076
3964 016706 012746 000004
3965 016712 010600
3966 016714 104417
3967 016716 062706 000012
3968
3969
3970
3971 016722
3972 016722

.SBTTL :***** TEST 8 *****
.SBTTL * ROM VERIFY ROM 1 DMV
ZZ
:
:
:
* THIS TEST IS USED TO VERIFY THE CONTENTS OF ROM 1
* THIS TEST IS NOT DONE FOR DMP
*-
.SBTTL :***** TEST 8 *****
T8: :
BIT #3,OPTYP ;IS THIS DMV
BEQ RDVEX ;IF NOT EXIT
MOV #0,ROMN1 ;SET UP ROM NUMBER
MOV #0,ROMBK1 ;SET ADD OF CRC BLOCK
MOV #0,ROMBK2 ;AND ADDRESS OF BLOCK
JSR R5,RMVRT ;GO TO SUB ROUTINE
CLR ERRWRD
TRAP C\$ESCAPE
.WORD L10041-
CMP #1,STARES ;IS IT FIRST PASS
BNE RDVEX ;IF NOT EXIT
MOV #0,ROMBK1 ;READ BLOCK WITH # AND REV
MOV #0,ROMBK2
JSR R5,RMVRT ;
CLR ERRWRD
TRAP C\$ESCAPE
.WORD L10041-
MOV #0,R2 ;SET UP ROM NUMBER OFFSET
MOVB RECBU1(R2),ROMNO
MOV #0,R2 ;SET UP REV NUMBER OFFSET
MOVB RECBU1(R2),REVNO
CMP ROMNO,ROMN1 ;COMPARE ROM NUMBER
BEQ 10\$
TRAP C\$ERDF
.WORD 23
.WORD 0
.WORD ERR25
TRAP C\$ESCAPE
.WORD L10041-
10\$:
MOV #REVNO,-(SP)
MOV #ROMNO,-(SP)
MOV LOGDEV,-(SP)
MOV #ROMMSG,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C\$PNTF
ADD #12,SP

RDVEX:
L10041:

CZDMTAD DMP/V-11 P CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 94
;***** TEST 8 *****

C 8

SEQ 0093

3973 016722 104401

TRAP CSETST


```
3974 .SBTTL :***** TEST 9 *****
3975 .SBTTL * ROM VERIFY ROM 2 DMV ONLY
3976 016724 ZZ
3977 :
3978 :
3979 : * THIS IS THE TEST THAT VERIFIES THE CONTENTS OF ROM 2
3980 : * OF THE DMV OPTION. THIS TEST IS NOT RUN FOR DMP
3981 :
3982 .SBTTL :***** TEST 9 *****
3983 19:
3984 016724 032737 000003 002466 BIT #3,OPTYP ;IS THIS DMV
3985 016732 001501 BEQ RDVFX2 ;IF NOT EXIT
3986 016734 012737 000000 002366 MOV #0,ROMN1 ;PUT ROM NUMBER IN LOC
3987
3988 016742 012737 000000 002432 MOV #0,ROMBK1 ;SET UP ADD OF CRC BLOCK
3989 016750 012737 000000 002434 MOV #0,ROMBK2 ;SET UP ADD OF LAST BLOCK
3990 016756 004537 005326 JSR R5,RMVRT ;GO READ AND CALC.
3991 016762 005037 002374 CLR ERRWRD
3992 016766 104410 TRAP C$ESCAPE
3993 016770 000146 .WORD L10042-
3994 016772 022737 000001 002344 CMP #1,STARES
3995 017000 001056 BNE RDVEX2 ;IF NOT FIRST PASS EXIT
3996
3997 017002 012737 000000 002432 MOV #0,ROMBK1
3998 017010 012737 000000 002434 MOV #0,ROMBK2
3999 017016 004537 005326 JSR R5,RMVRT ;READ BLOCK WITH REVNO.
4000 017022 005037 002374 CLR ERRWRD
4001 017026 104410 TRAP C$ESCAPE
4002 017030 000106 .WORD L10042-
4003 017032 012702 000000 MOV #0,R2 ;SET R2 TO REVNO OFFSET
4004 017036 116237 033022 013176 MOVB RECBU1(R2),REVNO
4005 017044 012702 000000 MOV #0,R2 ;SET R2 TO ROMNO OFFSET
4006 017050 116237 033022 013174 MOVB RECBU1(R2),ROMNO
4007 017056 123737 013174 002366 CMPB ROMNO,ROMN1
4008 017064 001406 BEQ 10$
4009 017066 104455 TRAP C$ERDF
4010 017070 000030 .WORD 24
4011 017072 000000 .WORD 0
4012 017074 007474 .WORD ERR25
4013 017076 104410 TRAP C$ESCAPE
4014 017100 000036 .WORD L10042-
4015 017102
4016 017102 012746 013176 10$: MOV #REVNO,-(SP)
4017 017106 012746 013174 MOV #ROMNO,-(SP)
4018 017112 013746 002330 MOV LOGDEV,-(SP)
4019 017116 012746 013076 MOV #ROMMSG,-(SP)
4020 017122 012746 000004 MOV #4,-(SP)
4021 017126 010600 MOV SP,R0
4022 017130 104417 TRAP C$PNTF
4023 017132 062706 000012 ADD #12,SP
4024
4025
4026 017136 RDVEX2:
4027 017136 L10042:
4028 017136 104401 TRAP C$ETST
4029
```

4030
4031
4032
4033 017140
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083 017140
4084
4085 017140 012737 000000 002340

.SBTTL :***** TEST 10 *****
.SBTTL *INITIALIZATION TEST (INTERNAL DIAGNOSTICS)

ZZ

* IN THIS TEST WE'LL START OUT BY SETTING THE MASTER CLEAR BIT (BIT 14 OF SELO)
* THE LOGIC CLEARS AND STARTS THE MICRO DIAGNOSTICS. IF THE MICRO-DIAGNOSTICS
* PASS, THE RUN BIT (BIT15 OF SELO) WILL SET.
* IF THE RUN BIT FAILS TO SET WITHIN 300 MILLI-SEC, IT
* PROBABLY MEANS THAT MICRO DIAGNOSTICS HAVE DETECTED AN
* ERROR AND THE TEST CODE IS IN BSFL6

DMP

TEST CODE	TEST ENTERED
143	BRANCH TEST
135	BRANCH EXTENDED TESTS
125,252,0	IBUS/OBUS TESTS
123	SCRATCH PAD TEST
151	ALU TESTS
222	MAIN MEMORY DATA TEST
132	MAIN MEMORY DUAL ADDRESS TEST
264	LINE UNIT TESTS
305	TESTS COMPLETE
	DMV TEST
101	BRANCH TEST
102	INTERNAL REG TEST
103	LOAD AND STORE INSTR.
104	COMPARE INSTR. TEST
105	INC/DEC INSTR.
106	SHIFT AND ROTATE INSTR.
107	LOGIC INSTR.
110	ADC,SBC,SED,CLD INSTRU.
111	STACK PUSH,PULL INSTR.
112	SUBROUTINE INSTR.
113	SCRATCH PAD,CSR,AND NPR
114	..
115	FALSE INT TEST
116	RAM DATA AND ADDRESS
117	RAM ALTERNATING TEST
120	INDEX INDIRECT TEST
121	LINE UNIT TEST

* NOTE THE RUN BIT WILL BE SET EVEN IF THE LINE UNIT
* TEST FAILS. TEST CODE MUST BE CHECKED TO FIND ERROR.
* THESE CODES ARE SET UPON ENTRY OF EACH TEST
* ONE SHOULD NOT BE DEPENDENT ON A BAD DMP-DMV MODULE
* TO PASS A CORRECT TEST CODE. IF THIS TEST FAILS, YOU
* SHOULD RUN THE REPAIR LEVEL DIAGNOSTIC

NOTE

* IF THIS TEST FAILS, CHECK SW7 OF SP#1 TO SEE IF RUN IS ENABLED.

.SBTTL :***** TEST 10 *****

T10::

MOV #0,COUNT ;CLEAR COUNTER

4086 017146 004737 004404
4087 017152
4088 017152 004737 004172
4089 017156 005777 163254
4090 017162 100411
4091 017164 005337 002340
4092 017170 001370
4093
4094
4095
4096 017172 104455
4097 017174 000031
4098 017176 010313
4099 017200 006714
4100 017202 104410
4101 017204 000116
4102 017206 122777 000305 163236 20\$
4103 017214 001420
4104 017216 122777 000264 163226
4105 017224 001406
4106
4107 017226 104455
4108 017230 000032
4109 017232 010313
4110 017234 006714
4111
4112
4113 017236 104410
4114 017240 000062
4115
4116 017242
4117 017242 104455
4118 017244 000033
4119 017246 010375
4120 017250 007672
4121 017252 104410
4122 017254 000046
4123
4124 017256
4125 017256 112777 000200 163152
4126 017264 012737 017306 002400
4127 017272
4128 017272 004537 005016
4129 017276 005037 002374
4130 017302 104410
4131 017304 000016
4132 017306
4133
4134
4135
4136 017306 032777 000020 163126 47\$
4137 017314 001766
4138 017316
4139 017316 104410
4140 017320 000002
4141 017322

10\$: JSR PC,MINITR
JSR PC,WAIT50
TST @SELO ;TEST DONE?
RMI 20\$;YES TEST FOR ERROR
DEC CJUNT ;UPDATE COUNT IF NOT TOO LONG
BNE 10\$;IN THIS WAIT LOOP, GO BACK
;INTERNAL DIAG FAILED
TRAP C\$ERDF
.WORD 25
.WORD MEF3A
.WORD ERR3
TRAP C\$ESCAPE
.WORD L10043-
CMPB #305,@BSEL6 ;LEGAL TEST COMPLETE CODE?
BEQ 40\$
CMPB #264,@BSEL6 ;LINE UNIT TEST FAILURE?
BEQ 30\$
TRAP C\$ERDF
.WORD 26
.WORD MEF3A
.WORD ERR3
;UNKNOWN ERROR WHILE INITIALLING
;DMP-11
TRAP C\$ESCAPE
.WORD L10043-
30\$: TRAP C\$ERDF
.WORD 27
.WORD MEF4
.WORD ERR32
TRAP C\$ESCAPE
.WORD L10043-
;MODULE FAULT
40\$: MOVB #R01,@BSEL0 ;SET R01 AND THEN WAIT FOR RDI TO SET.
MOV #ERL01,ERRADD ;SET UP ERROR ADD.
TLB1: JSR R5,TOUT
CLR ERRWRD
TRAP C\$ESCAPE
.WORD L10043-
ERL01: ;:TIME OUT ERROR REPORTS THIS ADDRESS
;:TIME OUT ERROR REPORTS THIS ADDRESS
47\$: BIT #RDI,@BSEL2 ;DID RDI SET?
BEQ TLB1
50\$: TRAP C\$ESCAPE
.WORD L10043-
L10043:

CZDMTAD DMP/V-11 r CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 ^{G 8} PAGE 98
;***** TEST 10 *****

SEQ 0097

4142 017322 104401

TRAP CBETST

```

4143
4144
4145 .SBTTL :***** TEST 11 *****
4146 .SBTTL * MICRO-DIAGNOSTIC-INTERFACE TESTING DMP ONLY
4147 017324 ??
4148 * DMP ONLY THIS TEST...
4149 * THIS TEST WILL EXERCISE THE MICRO-CPU'S INTERFACE TO THE PDP-11
4150 * WE FIRST START THE MCPU. NEXT WE GIVE THE COMMAND THAT
4151 * TAKES US TO THE INTERFACE DIAGNOSTIC CODE. ONCE THIS CODE IS
4152 * STARTED, WE MUST GO THROUGH ALL TESTS. THEREFORE, YOU WILL NOTICE
4153 * FIVE DISTINCT TESTS PERFORMED
4154 * AT THE END OF THIS TEST, THE MICRO-CODE IS LISTED.
4155 * VARIOUS SCOPE POINTS DO EXIST IF YOU NEED THEM. IT IS NOT
4156 * COMMON PRACTICE TO USE THEM, HOWEVER, WHERE SOME USE OF THEM
4157 * COULD BE MADE, THEY ARE NOTED.
4158
4159 .SBTTL :***** TEST 11 *****
4160
4161
4162 017324 T11::
4163
4164 017324 032737 000003 002466 BIT #3,OPTYP ;IS THIS DMV
4165 017332 001402 BEQ 10$ ;IF NOT GO TO 10;ELSE
4166 017334 000137 020454 JMP EXMDT ;EXIT TEST
4167 017340 10$:
4168 017340 T9BG:
4169 017340 004737 004450 JSR PC,MINITS
4170
4171 * JUMP TO END OF TEST IF ERROR
4172
4173 017344 005037 002374 CLR ERRWRD
4174 017350 104410 TRAP C$ESCAPE
4175 017352 001102 .WORD L10044-.
4176 017354 105077 163074 CLR @BSEL7
4177 017360 112777 000200 163050 MOV @DRUN,@BSEL0 ;REQUEST INTERFACE DIAGNOSTICS
4178 017366 25$:
4179 017366 004537 002726 JSR R5,WRDI ;WAIT FOR RDI TO SET
4180
4181 017372 005037 002374 CLR ERRWRD ;CLEAR ERROR
4182 017376 104410 TRAP C$ESCAPE
4183 017400 001054 .WORD L10044-.
4184
4185 * TIME OUT OR READY ERROR REPORTS THIS
4186 * ADDRESS AS FAILING PC
4187
4188 017402 105077 163030 CLR @BSEL0 ;NO MORE REQUESTS.
4189 017406 112777 000022 163036 MOV #22,@BSEL6 ;DIAGNOSTIC CODE.
4190 017414 105077 163034 CLR @BSEL7 ;CLEAR BSEL7
4191 017420 112777 000001 163014 MOV #1,@BSEL2 ;START.!
4192
4193 017426 TLB2:
4194 017426 012737 017450 002400 MOV #ERLB2,ERRADD ;SET UP ERROR ADDRESS
4195 017434 004537 005016 JSR R5,TOUT
4196 017440 005037 002374 LLR ERRWRD
4197 017444 104410 TRAP C$ESCAPE
4198 017446 001006 .WORD L10044-.
  
```

```

4199 017450          ERLB2:
4200                : : : : :
4201                : TIME OUT ERROR REPORTS THIS ADDRESS
4202                : : : : :
4203 017450 122777 000377 162776 26$: (MPB #377,@BSEL7 ;LOOK FOR SYNC OF CODE 377 IN LAS' REG
4204 017456 001363                BNE TLB2 ;IF 'HANG' OCCURS HERE THEN ITS POSSIBLE
4205                ;THAT EITHER THE DATA PATHS ARE BAD OR
4206                ;THAT THE MCPU FAILED TO START
4207
4208 017460 012737 000377 002334      MOV #377,$GDDAT ;EXPECT 377 BACK FROM ALL REGS
4209 017466 013701 002436      MOV BSEL0,R1 ;EXPECT REG 1 (MAINTENANCE)
4210 017472 012737 000000 002342      MOV #0,REG
4211 017500 111137 002336 4$: MOVB (R1),$BDDAT ;READ REG, EXPECT 377
4212 017504 123737 002336 002334      CMPB $BDDAT,$GDDAT ;OK?
4213 017512 001412                BEQ 5$ ;YES-CONTINUE
4214 017514 022737 000001 002342      CMP #1,REG ;NO ERROR? (EXCEPT REG 1)
4215 017522 001406                BEQ 5$ ;IF REG 1, SKIP
4216
4217 017524 104455                TRAP C$ERDF
4218 017526 000034                .WORD 28
4219 017530 010450                .WORD MEFC
4220 017532 006746                .WORD ERR5
4221 017534 104410                TRAP C$ESCAPE
4222 017536 000716                .WORD L10044-.
4223 017540 005237 002342 5$: INC REG ;UPDATE REGISTER #
4224 017544 005201                INC R1 ;AND ADDRESS
4225 017546 023727 002342 000010      CMP REG,#10 ;DONE ALL REGS?
4226 017554 001351                BNE 4$
4227 017556 105077 162654          CLRB @BSEL0 ;CAUSES MCPU TO EXIT TSTA
4228
4229
4230                ;TEST B
4231
4232
4233 017562          TLB3:
4234 017562 012737 017604 002400      MOV #ERLB3,EKRADD ;SET ERROR ADDRESS
4235 017570 004537 005016          JSR R5,TOUT
4236 017574 005037 002374          CLR ERRWRD
4237 017600 104410                TRAP C$ESCAPE
4238 017602 000652                .WORD L10044-.
4239 017604          ERLB3:
4240                : : : : :
4241                : TIME OUT ERROR REPORTS THIS ADDRESS
4242                : : : : :
4243 017604 105777 162644 27$: TSTB @BSEL7 ;LOOK FOR A ZERO IN BSEL7
4244 017610 001364                BNE TLB3
4245
4246 017612 005037 002334          CLR $GDDAT ;EXPECT ALL ZEROS EXCEPT SBEL1
4247 017616 013701 002436          MOV BSEL0,R1 ;GET ADDR OF MCPU.
4248 017622 012737 000000 002342      MOV #0,REG
4249 017630 005037 002336          CLR $BDDAT
4250 017634 111137 002336 7$: MOVB (R1),$BDDAT ;READ REG
4251 017640 001412                BEQ 8$ ;IF ZERO-CONTINUE
4252 017642 022737 000001 002342      CMP #1,REG ;IF REG #1 CONTINUE
4253 017650 001406                BEQ 8$
4254
  
```

```

4255 017652 104455 TRAP C$ERDF
4256 017654 000035 .WORD 29
4257 017656 010450 .WORD MEFC
4258 017660 006746 .WORD ERR5
4259 017662 104410 TRAP C$ESCAPF
4260 017664 000570 .WORD L10044-.
4261 017666 005237 002342 8$: INC REG ;UPDATE REGISTER #
4262 017672 005201 INC R1 ;AND ADDRESS
4263 017674 122737 000010 002342 CMPB #10,REG ;DONE ALL REGS (0-7)?
4264 017702 001354 BNE 7$ ;NO-DO NEXT ONE
4265
4266 017704 000404 BR 9$ ;REPLACE THIS INSTRUCTION WITH CODE 240
4267 ;(NOP) IF YOU WITH TO COOP IN
4268 ;TESTS A&B
4269 017706 112777 000200 162522 MOVB #200,@BSELO ;ALL MICRO-CODE TO LOOP
4270 017714 000611 BR T9BG ;LOOP
4271
4272 017716 112777 000377 162512 9$: MOVB #377,@BSELO ;TELL MICRO-CODE TO EXIT TEST B,
4273 ;PROCEED TO TEST C.
4274
4275
4276 ;TEST C
4277
4278 017724 017746 162534 MOV @KMLVL,-(SP)
4279 017730 012746 020364 MOV #INTC0,-(SP)
4280 017734 013746 002456 MOV KMRVEC,-(SP)
4281 017740 012746 000003 MOV #3,-(SP)
4282 017744 104437 TRAP C$SVEC
4283 017746 062706 000010 ADD #10,SP
4284 ;INTERRUPT VECTOR
4285
4286 017752 017746 162506 MOV @KMTLVL,-(SP)
4287 017756 012746 020400 MOV #INTC4,-(SP)
4288 017762 013746 002460 MOV KMTVEC,-(SP)
4289 017766 012746 000003 MOV #3,-(SP)
4290 017772 104437 TRAP C$SVEC
4291 017774 062706 000010 ADD #10,SP
4292 ;ILLEGAL INTERRUPT TO WRONG VECTOR
4293
4294 020000 005037 002332 CLR IFLAG
4295 020004 112777 000377 162432 MOVB #377,@BSEL3 ;TELL MICRO-CODE TO FORCE INTERRUPT
4296 020012 012700 000000 MOV #0,R0
4297 020016 104441 TRAP C$SPRI
4298
4299 020020 TLB4:
4300 020020 012737 020042 002400 MOV #ERLB4,ERRADD ;SET UP ERROR ADDRESS
4301 020026 004537 005016 JSR R5,TOUT
4302 020032 005037 002374 CLR ERRWRD
4303 020036 104410 TRAP C$ESCAPE
4304 020040 000414 .WORD L10044-.
4305 020042 ERLB4:
4306 ;:TIME OUT REPORTS THIS ADDRESS
4307 ;:
4308 ;:
4309 020042 005737 002332 28$: TST IFLAG ;IFLAG=1 SET BY INTERRUPT SERVICE ROUTINE
4310 020046 001764 BEQ TLB4 ;LOOP UNIT DONE

```

:NOTE: IF HANGS HERE, MCPU FAILS TO
 :GENERATE INTERRUPT TO PDP-11.

```

4311
4312
4313
4314
4315          :TEST   D
4316
4317 020050 017746 162410      MOV    @KMRVLV,-(SP)
4318 020054 012746 020420      MOV    #INTD0,-(SP)
4319 020060 013746 002456      MOV    KMRVEC,-(SP)
4320 020064 012746 000003      MOV    #3,-(SP)
4321 020070 104437              TRAP   C$SVEC
4322 020072 062706 000010      ADD    #10,SP
4323 020076 017746 162362      MOV    @KMTLVL,-(SP)
4324 020102 012746 020440      MOV    #INTD4,-(SP)
4325 020106 013746 002460      MOV    KMTVEC,-(SP)
4326 020112 012746 000003      MOV    #3,-(SP)
4327 020116 104437              TRAP   C$SVEC
4328 020120 062706 000010      ADD    #10,SP
4329
4330 020124 005037 002332      CLR    IFLAG          ;NO INTERRUPT INDICATOR
4331 020130 012700 000000      MOV    #0,R0
4332 020134 104441              TRAP   C$SPRI
4333 020136 105077 162302      CLR    @BSEL3        ;TELL MCPU TO INTERRUPT
4334
4335 020142          TLB5:
4336 020142 012737 020164 002400  MOV    #ERLB5,ERRADD ;SET UP ERROR ADDRESS
4337 020150 004537 005016      JSR    R5,TOUT
4338 020154 005037 002374      CLR    ERRWRD
4339 020160 104410              TRAP   C$ESCAPE
4340 020162 000272          .WORD  L10044-
4341 020164          ERLB5:
4342          :
4343          : TIME OUT REPORTS THIS ADDRESS
4344          :
4345 020164 005737 002332      29$:  TST    IFLAG          ;DID MCPU INTERRUPT (IFLAG NOT 0)?
4346 020170 001764          BEQ    TLB5          ;NO - LOOP
4347          ;NOTE: IF PROGRAM 'HANGS' HERE, MCPU
4348          ;FAILED TO INTERRUPT TO VECTOR XX4
4349          ;WE KNOW MCPU IS ABLE TO INTERRUPT
4350          ;TO XX0 (TEST C)
4351
4352 020172 013700 002456      MOV    KMRVEC,R0
4353 020176 104436              TRAP   C$CVEC
4354 020200 013700 002460      MOV    KMTVEC,R0
4355 020204 104436              TRAP   C$CVEC
4356
4357
4358          ;TEST E NPR TEST
4359
4360
4361 020206 012701 033002      MOV    #DATLST,R1    ;GET DATA LIST
4362 020212 152777 000010 162220  BIC    #BIT3,@BSEL1 ;SET INDICATOR THAT WE ARE STILL IN TEST.
4363
4364 020220 011137 002334          TLB6A: MOV    (R1),%GDDAT    ;GET NEXT PATTERN
4365 020224 010177 162216      MOV    R1,@BSEL4    ;SET NPR IN ADDR
4366 020230 012777 002336 162214  MOV    #BDDAT,@BSEL6 ;SET NPR OUT ADDR
  
```



```

4367 020236 105077 162174          CLRB    @BSELO          ;TELL MCPU TO DO NPRS
4368
4369 020242
4370 020242 012737 020264 002400  TLB6:  MOV    #ERLB6,ERRADD ;SET ERROR ADDRESS
4371 020250 004537 005016          JSR    R5,TOUT
4372 020254 005037 002374          CLR    ERRWRD
4373 020260 104410          TRAP   C$ESCAPE
4374 020262 000172          .WORD L10044-.
4375 020264          ERLB6:
4376
4377          : TIME OUT ERROR REPORTS THIS ADDRESS
4378
4379 020264 132777 000010 162146 30$:  BITB   #BIT3,@BSEL1 ;DID WE ACCIDENTILY ESCAPE THIS TEST???
4380 020272 001006          BNE    135$
4381 020274 104455          TRAP   C$ERDF
4382 020276 000036          .WORD 30
4383 020300 010664          .WORD MEF7
4384 020302 007672          .WORD ERR32
4385
4386 020304 104410          TRAP   C$ESCAPE          ;UNKNOWN MCPU ERROR CAUSED ABORT OF TEST.
4387 020306 000146          .WORD L10044-.
4388 020310          135$:
4389
4390 020310 122777 000377 162120  CMPB   #377,@BSELO ;WHEN MCPU DONE, IT PUTS 377 INTO BSELO
4391 020316 001351          BNE    TLB6             ;IF WE 'HANG' HERE, MCPU FAILS TO DO
4392
4393
4394 020320 023737 002334 002336  CMP    $GDDAT,$BDDAT ;NPRED FRO PATTERN LIST TO $BDDAT
4395
4396 020326 001406          BEQ    14$             ;DID XFER OCCUR SUCCESSFULLY?
4397
4398 020330 104455          TRAP   C$ERDF
4399 020332 000037          .WORD 31
4400 020334 010450          .WORD MEF7
4401 020336 007004          .WORD ERR6
4402 020340 104410          TRAP   C$ESCAPE
4403 020342 000112          .WORD L10044-.
4404 020344 022721 000562 14$:  CMP    #562,(R1)+ ;IS IT THE LAST PATTERN (562) IS TERM)?
4405 020350 001323          BNE    TLB6A
4406 020352 112777 000200 162056  MOVB   #200,@BSELO ;TELL MCPU TO EXIT TEST
4407 020360 104432          TRAP   C$EXIT
4408 020362 000072          .WORD L10044-.
4409
4410          INTC0::
4411 020364 013700 000006          MOV    6,R0
4412 020370 104441          TRAP   C$SPRI
4413 020372 005237 002332          INC    IFLAG
4414 020376          L10045:
4415 020376 000002          RTI
4416
4417          INTC4::
4418 020400 013700 000006          MOV    6,R0
4419 020404 104441          TRAP   C$SPRI
4420 020406 104455          TRAP   C$ERDF
4421 020410 000040          .WORD 32
4422 020412 010716          .WORD MEF8
  
```

4423	020414	007672				
4424	020416			L10046:	.WORD	ERR32
4425	020416	000002			RTI	
4426						
4427	020420			INTD0::		
4428	020420	013700	000006		MOV	6,RO
4429	020424	104441			TRAP	C\$SPRI
4430	020426	104455			TRAP	C\$ERDF
4431	020430	000041			.WORD	33
4432	020432	010716			.WORD	MEF8
4433	020434	007672			.WORD	ERR32
4434	020436			L10047:		
4435	020436	000002			RTI	
4436						
4437	020440			INTD4::		
4438	020440	013700	000006		MOV	6,RO
4439	020444	104441			TRAP	C\$SPRI
4440	020446	005237	002332		INC	IFLAG
4441	020452			L10050:		
4442	020452	000002			RTI	
4443						
4444	020454			EXMDT:		
4445						
4446	020454			L10044:		
4447	020454	104401			TRAP	C\$ETST
4448						
4449						

;CORRECT VECTOR (XX4)

```
4450 .SBTTL ***** TEST 12 *****
4451 .SBTTL RDI REMAINS SET TEST
4452 020456 ZZ
4453
4454 *ROM FUNCTION TEST IN THIS TEST, WE'RE GOING TO SET RDI, GET A
4455 * RDI, DO A CONTROL IN COMMAND WITH A REQUEST
4456 * KEY OF 00 (NO REQUEST). NEXT WE'LL WAIT
4457 * FOR RDI TO SET AGAIN SINCE RDI WAS
4458 * LEFT SET
4459 *
4460 .SBTTL ***** TEST 12 *****
4461 T12::
4462
4463 020456 004737 004450 JSR PC,MINITS ;INITIALIZE & START MCPU
4464
4465 020462 005037 J02374 CLR ERRWRD
4466 020466 104410 TRAP C$ESCAPE
4467 020470 000076 .WORD L10051-.
4468
4469 : JUMP TO END OF TEST IF ERROR
4470 :
4471
4472 020472 7$:
4473
4474 020472 052777 000200 161736 BIS #RDI,@BSEL0 ;SET RDI
4475
4476 020500 10$:
4477 020500 004537 002726 JSR R5,WRDI ;WAIT FOR RDI TO SET
4478
4479 020504 005037 002374 CLR ERRWRD ;CLEAR ERROR
4480 020510 104410 TRAP C$ESCAPE
4481 020512 000054 .WORD L10051-.
4482
4483 : TIME OUT - OR READY ERROR REPORTS
4484 : THIS ADDRESS
4485 :
4486
4487 020514 005077 161732 20$: CLR @BSEL6 ;CLEAR RDI, ISSUE REQUEST OF NO REQUEST
4488 ;THIS SHOULD CAUSE RDI TO SET AGAIN
4489 ;SINCE RDI HAD REMAINED SET
4490 020520 112777 000001 161714 MOVB #1,@BSEL2 ;START
4491
4492 020526 004737 004172 JSR PC,WAIT50 ;WAIT THIS SHORT TIME SO THAT THE
4493 020532 004737 004172 JSR PC,WAIT50 ;DMP MICRO-CODE MAY RESET 'RDI' IF
4494 020536 004737 004172 JSR PC,WAIT50
4495 020542 J04737 004172 JSR PC,WAIT50
4496 ;IT IS GOING TO
4497
4498 020546 032777 000020 161666 BIT #RDI,@BSEL2 ;IS RDI SET?
4499 020554 001004 BNE 30$
4500
4501 020556 104455 TRAP C$ERDF
4502 020560 000042 .WORD 34
4503 020562 011100 .WORD MRF1
4504 020564 007042 .WORD ERR9
4505 ;COMPLETING A 'NO REQUEST' CONTROL
```

CZDMTAD DMP/V-11 + CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 106

B 9

;***** TEST 12 *****

SEQ 0105

4506
4507
4508 020566
4509 020566
4510 020566 104401

;IN COMMAND

308:
L10051:
TRAP CSETST

```

4511
4512 .SBTTL :***** TEST 13 *****
4513 .SBTTL *ROM FUNC TEST. VERIFY RDO SETS
4514 020570 ZZ
4515 :*ROM FUNC IN THIS TEST WE'LL DO A CONTROL IN WITH
4516 :* READ MODEM AS THE REQUEST KEY. WE'LL MAKE
4517 :* SURE THAT RDO SETS. WE SHOULD GET A
4518 :* RETURN KEY OF 10 'RETURN MODEM'
4519 :*
4520 .SBTTL :***** TEST 13 *****
4521 020570 T13::
4522
4523 020570 004737 004450 JSR PC,MINITS ;INIT & START MCPU
4524
4525 020574 005037 002374 CLR ERRWRD
4526 020600 104410 TRAP C$ESCAPE
4527 020602 000174 .WORD L10052-.
4528
4529 : JUMP TO END OF TEST IF ERROR
4530 :
4531
4532 020604 052777 000200 161624 BIS #RQ1,@BSELO ;SET REQUEST IN
4533 020612 004537 002726 JSR R5,WRDI ;WAIT FOR RDI TO SET
4534
4535 020616 005037 002374 CLR ERRWRD ;CLEAR ERROR
4536 020622 104410 TRAP C$ESCAPE
4537 020624 000152 .WORD L10052-.
4538
4539 : TIME OUT OR READY ERROR REPORTS
4540 : THIS ADDRESS AS FAILING PC
4541 :
4542 020626 042777 000200 161602 20$: BIC #RQ1,@BSELO ;DROP REQUEST
4543 020634 012777 000020 161610 MOV #20,@BSEL6 ;READ MODEM
4544 020642 112777 000001 161572 MOVB #1,@BSEL2
4545
4546 020650 004737 004172 JSR PC,WAIT50 ;STALL
4547 020654 004737 004172 JSR PC,WAIT50
4548
4549 020660 032777 000200 161554 BIT #RDO,@BSEL2 ;DID 'RDO' SET?
4550 020666 001006 BNE 30$
4551
4552 020670 104455 TRAP C$ERDF
4553 020672 000043 .WORD 35
4554 020674 011100 .WORD MRFT
4555 020676 007070 .WORD ERR10
4556 :REQUEST FOR READ MODEM
4557 020700 104410 TRAP C$ESCAPE
4558 020702 000074 .WORD L10052-.
4559
4560 020704 117737 161532 002336 30$: MOVB @BSEL2,$BDDAT ;NOW GET CSR AND
4561 020712 042737 177770 002336 BIC #^C<?>,$BDDAT ;STRIP FOR
4562 020720 012737 000002 002334 MOV #2,$GDDAT ;TYPE CODE OF INFORMATION OUT
4563 020726 023737 002334 002336 CMP $GDDAT,$BDDAT
4564 020734 001411 BEQ 40$
4565 020736 012737 011257 002422 MOV #M28F,CODEW
4566 020744 104455 TRAP C$ERDF
  
```

CZDMTAD DMP/V-11 CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 108
;***** TEST 13 *****

D 9

SEQ 0107

4567	020746	000044				.WORD	36		
4568	020750	010756				.WORD	EROIC		
4569	020752	007574				.WORD	ERR27		
4570									; IN RDO
4571	020754	104410				TRAP	C\$ESCAPE		
4572	020756	000020				.WORD	L10052-		
4573									
4574	020760				40\$:				
4575	020760	112737	000010	002334		MOVB	#10, \$GDDAT		; SHOULD=10 'RETURN MODEM'
4576	020766	004537	003366			JSR	R5,GETRKY		; GO GET AND CHECK RETURN KEY
4577									
4578	020772	005037	002374			CLR	ERRWRD		
4579	020776				L10052:				
4580	020776	104401				TRAP	C\$ETST		

```
4581 .SBTTL :***** TEST 14 *****
4582 .SBTTL * NON-MODE DEF AFTER MC PROCEEDURE ERR CHECK
4583 021000 ZZ
4584 :
4585 :
4586 : THIS TEST CHECKS FOR PROCEDURE ERROR WHEN
4587 : NON-MODE DEFINITION IS DONE AFTER MC
4588 :
4589 :
4590 .SBTTL :***** TEST 14 *****
4591 021000 T14::
4592
4593 021000 112777 000100 161432 MOVB #100,@BSEL1 ;MASTER CLEAR
4594 021006 022737 000004 002466 CMP #4,OPTYP ;8206
4595 021014 001003 BNE TLB10 ;IF NOT GO TO TLB10
4596 021016 112777 000200 161414 MOVB #200,@BSEL1 ;SET RUN 8206
4597
4598 021024 TLB10:
4599 021024 004537 005016 JSR R5,TOUT
4600 021030 005037 002374 CLR ERRWRD
4601 021034 104410 TRAP C$ESCAPE
4602 021036 000204 .WORD L10053-
4603 021040 005777 161372 ERLB10: TST @BSEL0
4604 021044 100367 BPL TLB10 ;LOOP IF NOT RUN
4605 021046 052777 000200 161362 BIS #RQI,@BSEL0 ;SET REQUEST
4606 021054 004537 002726 JSR R5,WRDI ;WAIT FOR RDI TO SET
4607
4608 021060 CLR ERRWRD ;CLEAR ERROR
4609 021064 104410 TRAP C$ESCAPE
4610 021066 000154 .WORD L10053-
4611 021070 105077 161342 CLRB @BSEL0 ;CLEAR REQUEST
4612 021074 105077 161344 CLRB @BSEL3 ;MAKE TRIB ADD 0
4613 021100 112777 000052 161344 MOVB #52,@BSEL6 ;READ TSS
4614 021106 112777 000001 161326 MOVB #01,@BSEL2 ; EXECUTE CONTROL IN
4615 021114 004537 002642 JSR R5,WRDO ;WAIT FOR RDO
4616 021120 005037 002374 CLR ERRWRD
4617 021124 104410 TRAP C$ESCAPE
4618 021126 000114 .WORD L10053-
4619 021130 117737 161306 002336 MOVB @BSEL2,$BDDAT
4620 021136 042737 177770 002336 BIC #^C<7>,$BDDAT ;STRIP TO COMMAND CODE
4621 021144 022737 000002 002336 CMP #2,$BDDAT ;IS IT INFO OUT
4622 021152 001433 BEQ T14EX ;IF YES EXIT TEST
4623 021154 022737 000001 002336 CMP #01,$BDDAT ;IF NOT IS IT CONTROL OUT
4624 021162 001411 BEQ T14A ;
4625 021164 012737 011250 002422 MOV #M18F,CODEW
4626 021172 104455 TRAP C$ERDF
4627 021174 000045 .WORD 37
4628 021176 010756 .WORD EROIC
4629 021200 007574 .WORD ERR27
4630 021202 104410 TRAP C$ESCAPE
4631 021204 000036 .WORD L10053-
4632 021206 117737 161240 002336 T14A: MOVB @BSEL6,$BDDAT
4633 021214 022737 000100 002336 CMP #100,$BDDAT ;IS IT 100
4634 021222 001407 BEQ T14EX ;IF SO END TEST ELSE ERROR
4635 021224 012737 011232 002422 MOV #M13F,CODEW
4636 021232 104455 TRAP C$ERDF
```

CZDMTAD DMP/V-11 PCTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 F 9 PAGE 110
;..... TEST 14

SEQ 0109

4637 021234 000046
4638 021236 010756
4639 021240 007574
4640 021242
4641 021242
4642 021242 104401
4643

.WORD 38
.WORD EROIC
.WORD ERR27

T14EX:
L10053:
TRAP CSETST


```

4644
4645
4646
4647 021244
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658 021244
4659
4660 021244 004737 004450
4661
4662
4663 021250 005037 002374
4664 021254 104410
4665 021256 000174
4666
4667
4668
4669 021260 004737 004172
4670
4671 021264 004737 004172
4672 021270 142777 000010 161142
4673 021276 022737 000004 002472
4674 021304 001027
4675 021306 032777 000200 161126
4676 021314 001423
4677 021316 012737 000304 002334
4678 021324 117737 161122 002336
4679 021332 023737 002336 002334
4680 021340 001406
4681 021342 104455
4682 021344 000047
4683 021346 011730
4684 021350 007672
4685 021352 104432
4686 021354 000076
4687 021356
4688 021356 042777 000200 161056
4689 021364 052777 000200 161044
4690 021372 004537 002726
4691
4692 021376 005037 002374
4693 021402 104410
4694 021404 000046
4695 021406 043777 000200 161022
4696 021414 012777 000007 161030
4697 021422 012777 000002 161012
4698 021430 012737 000104 002402
4699

```

```

.SBTTL :***** TEST 15 *****
.SBTTL * MODE DEF,MODE DEF PROCEDURE ERROR
ZZ
:
:
:
:
: THIS TEST CHECKS THAT AFTER THE SEQUENCE OF
: MASTER CLEAR MODE DEF FOLLOWED BY MODE DEF
: DIFFERENT TYPE PRODUCES A PROCEDURE ERROR OF
: OCTAL 104
:
:
:
.SBTTL :***** TEST 15 *****
115::
JSR PC,MINITS ;DO MC,MODE DEF(CONT STA/FD)

CLR ERRWRD
TRAP C$ESCAPE
.WORD L10054-.
:
: JUMP TO END OF TEST IF ERROR
:
JSR PC,WAIT50 ;WAIT A WHILE TO BE SURE MODEM READY
;IS SET

JSR PC,WAIT50
BICB #BIT3,@BSEL1 ;CLEAR LU LOOP
CMP #4,TSTCON ;IS THIS NO LOOPBACK
BNE 20$ ;IF LOOPBACK GO TO 20
BIT #RDO,@BSEL2 ;IS RDO SET?
BEQ 20$ ;IF NOT GO TO 20
MOV #304,$GDDAT
MOV @BSEL6,$BDDAT ;IF YES.IS IT 304
CMP $BDDAT,$GDDAT
BEQ 25$ ;IF EQUAL GO TO 25
TRAP C$ERDF
.WORD 39
.WORD MEF30
.WORD ERR32
TRAP C$EXIT
.WORD L10054-.

25$:
BIC #RDO,@BSEL2 ;CLEAR RDO
20$:
BIS #RQ1,@BSEL0 ;SET REQUEST
JSR R5,WRDI ;WAIT FOR RDI TO SET

CLR ERRWRD ;CLEAR ERROR
TRAP C$ESCAPE
.WORD L10054-.
BIC RQ1,@BSEL0 ; CLEAR REQUEST
MOV #7,@BSEL6 ;SET MODE FOR TRIB/FD
MOV #02,@BSEL2 ;DO MODE DEF
MOV #104,PERR ;SET PROCEEDURE ERROR OF
; 104 TO BE CHECKED

```

4700	021436	004537	003012	JSR	R5,WFPE	;WAIT FOR PROCEDURE ERROR
4701						
4702	021442	005037	002374	CLR	ERRWRD	
4703	021446	104410		TRAP	C\$ESCAPE	
4704	021450	000002		.WORD	L10054-	
4705					
4706				: ESCAPE TEST IF ERROR		
4707					
4708						; ERROR,OR TIME OUT.
4709						
4710	021452					
4711	021452	104401		L10054:	TRAP C\$ETST	

```

4712
4713
4714
4715 021454
4716
4717
4718
4719
4720
4721
4722
4723 021454
4724
4725 021454 004737 004450
4726 021460 004737 004172
4727
4728 021464 005037 002374
4729 021470 104410
4730 021472 000212
4731
4732
4733
4734
4735 021474 032737 000001 002472
4736 021502 001402
4737 021504 012704 000104
4738 021510 012703 000021
4739 021514 004537 003156
4740
4741 021520 005737 002374
4742 021524 100467
4743
4744 021526 142777 000010 160704
4745
4746 021534 004737 004172
4747 021540 004737 004172
4748 021544 022737 000004 002472
4749 021552 001020
4750 021554 032777 000200 160660
4751 021562 001414
4752 021564 012737 000304 002402
4753 021572 004537 003012
4754
4755 021576 005037 002374
4756 021602 104410
4757 021604 000100
4758
4759
4760
4761 021606 042777 000200 160626
4762 021614 052777 000200 160614
4763 021622 004537 002726
4764
4765 021626 005037 002374
4766 021632 104410
4767 021634 000050

.SBTTL ***** TEST 16 *****
.SBTTL * MODE DEF ,MODE DEF CHANGE DUPLEX ONLY.
ZZ
*
*
* THIS CHECKS THAT YOU CAN CHANGE THE DUPLEX PORTION
* OF A MODE DEF
*
.SBTTL ***** TEST 16 *****
T16::
JSR PC,MINITS ;MC,MODE DEF(CONT/FD)
JSR PC,WAIT50 ;DELAY
CLR ERRWRD
TRAP C$ESCAPE
.WORD L10055-.
; JUMP TO END OF TEST IF ERROR

BIT #BIT0,TSTCON ;IS IT REMOTE MODEM
BEQ 1$ ;IF NOT THEN GO TO 1A
MOV #104,R4
MOV #21,R3
JSR R5,CONTIN ; WRITE MODEM WITH CORRECT
; TYPE OF LOOP CODE

TST ERRWRD
BMI 10$ ;EXIT IF ERROR

BICB #BIT3,@BSEL1 ;CLEAR LU LOOP

JSR PC,WAIT50 ;WAIT A WHILE
JSR PC,WAIT50
CMP #4,TSTCON
BNE 20$ ;IF LOOPBACK GO TO 20
BIT #RDO,@BSEL2 ;ELSE SEE IF READY OUT
BEQ 20$ ;IF NOT GO TO 20
MOV #304,PERR
JSR R5,WFPE ;WAIT FOR PROCEDURE ERROR

CLR ERRWRD
TRAP C$ESCAPE
.WORD L10055-.
; ESCAPE TEST IF ERROR

BIC #RDO,@BSEL2 ;CLEAR OUTPUT
BIS #RQ1,@BSEL0 ;SET REQUEST
JSR R5,WRDI ;WAIT FOR RDI TO SET

CLR ERRWRD ;CLEAR ERROR
TRAP C$ESCAPE
.WORD L10055-.
    
```

4768
4769
4770
4771 021636 043777 000200 160572
4772 021644 112777 000004 160600
4773 021652 112777 000002 160562
4774 021660 004737 004172
4775 021664 032777 000200 160550
4776 021672 001404
4777 021674 104455
4778 021676 000050
4779 021700 011730
4780 021702 007672
4781 021704
4782 021704
4783 021704 104401
4784

TIME OUT OR READY ERRORS REPORT THIS PC

BIC R01,@BSELO ;NO MORE REQUESTS
MOVB #04,@BSEL6 ;CONT/FD FOR MODE
MOVB #02,@BSEL2 ;DO MODE DEF
JSR PC,WAIT50 ;DELAY A WHILE
BIT #RDO,@BSEL2 ;IS RDO SET
BEQ 10\$;BRANCH IF NOT
TRAP C\$ERDF
.WORD 40
.WORD MEF30
.WORD ERR32

10\$:
L10055:
TRAP C\$ETST

```
4785 .SBTTL :***** TEST 17 *****
4786 .SBTTL :ROM FUNC. TEST. VERIFY THAT MAX TRIBS CAN BE ESTABLISHED
4787 021706 ZZ
4788 :ROM FUNCTION TEST-ESTABLISHING TRIBS-
4789 :* THIS TEST WILL ESTABLISH MAX TRIBS
4790 :* THEN TRY TO ESTABLISH MAX+1 TRIBS
4791 :* AND CHECK FOR PROCEDURE ERROR.
4792 :* THE TEST ALSO CHECKS FOR PROCEDURE
4793 :* ERROR WHEN TRYING TO ESTABLISH AN
4794 :* ALREADY ESTABLISHED TRIB.
4795 :*
4796 .SBTTL :***** TEST 17 *****
4797
4798 021706 T17::
4799 021706 012737 000040 002360 MOV #32.,TRIBMX ;SET MAX TRIB TO 32
4800 021714 032737 000003 002466 BIT #3,OPTYP ;IS THIS DMV
4801 021722 001403 BEQ AX ;IF NOT BRANCH
4802 021724 012737 000014 002360 MOV #12.,TRIBMX ;ELSE SET THE MAX TO 12
4803 021732
4804
4805 021732 112737 000161 002356 MOVB #161,TRIBN ;NUMBER OF TRIBUTARY,
4806 021740 112737 000161 002362 MOVB #161,TRIBH ;START TRIB HIGH AT SAME AS TRIBN
4807 021746 063737 002360 002362 ADD TRIBMX,TRIBH ;ADD MAX NUMBER OF TRIBS TO TRIBH
4808
4809 021754 004737 004450 JSR PC,MINITS ;INITIALIZE
4810
4811 021760 005037 002374 CLR ERRWRD
4812 021764 104410 TRAP C$ESCAPE
4813 021766 000150 .WORD L10056-.
4814
4815 : JUMP TO END OF TEST IF ERROR
4816 :*****
4817 021770 30$:
4818 021770 112703 000001 MOVB #01,R3 ;SET ESTABLISH TRIB
4819 021774 004537 003156 JSR R5,CONTIN ;
4820 :*****
4821 : READY OR TIME OUT ERRORS REPORT THIS PC
4822 :*****
4823
4824 022000 005037 002374 CLR ERRWRD
4825 022004 104410 TRAP C$ESCAPE
4826 022006 000130 .WORD L10056-.
4827 :*****
4828 : JUMP TO END OF TEST IF ERROR
4829 :*****
4830 022010 005237 002356 INC TRIBN ;UPDATE TRIB#
4831 022014 023737 002362 002356 CMP TRIBH, TRIBN ;ONLY ALLOW MAX TRIBS TO BE SET
4832 022022 001362 BNE 30$
4833
4834 022024 37$:
4835 022024 112703 000001 MOVB #01, R3 ;ESTABLISH MAX +1 TRIBS
4836 022030 004537 003156 JSR R5,CONTIN ; DO IT
4837 :*****
4838 : READY OR TIME OUT ERRORS REPORT THIS PC
4839 :*****
4840
```

```

4841 022034 005037 002374      CLR      ERRWRD
4842 022040 104410              TRAP     C$ESCAPE
4843 022042 000074              .WORD   L10056-.
4844                               :
4845                               : JUMP TO END OF TEST IF ERROR
4846                               :
4847                               :
4848 022044 012737 000114 002402  MOV      #114, PERR      ;SHOULD READ 114, PROCEDURE ERROR
4849                               : TRYING TO ESTABLISH MAX+1 TRIBUTARIES
4850                               :
4851 022052 004537 003012      JSR      R5,WFPE        ;WAIT FOR PROCEDURE ERROR
4852                               :
4853 022056 005037 002374      CLR      ERRWRD
4854 022062 104410              TRAP     C$ESCAPE
4855 022064 000052              .WORD   L10056-.
4856                               :
4857                               : ESCAPE TEST IF ERROR
4858                               :
4859 022066                               60$:
4860 022066 042777 000200 160346  BIC      #RDO,@BSEL2    ;CLEAR RDO
4861 022074 005337 002356      DEC      TRIBN         ;DEC TRIB NUMBER
4862 022100 112703 000001      MOVB    #01, R3        ;SET ESTABLISH TRIB
4863 022104 004537 003156      JSR      R5,CONTIN     ;DO IT
4864                               :
4865                               : READY OR TIME OUT ERRORS REPORT THIS PC
4866                               :
4867                               :
4868 022110 005037 002374      CLR      ERRWRD
4869 022114 104410              TRAP     C$ESCAPE
4870 022116 000020              .WORD   L10056-.
4871                               :
4872                               : JUMP TO END OF TEST IF ERROR
4873                               :
4874                               :
4875 022120 112737 000116 002402  MOVB    #116, PERR     ;SHOULD BE PROCEDURE ERROR
4876                               : OF 116 ESTABLISH ALREADY
4877                               : ESTABLISHED TRIB.
4878 022126 004537 003012      JSR      R5,WFPE        ;GO CHECK FOR PROCEDURE ERROR
4879 022132 005037 002374      CLR      ERRWRD
4880 022136                               70$:
4881 022136                               L10056:
4882 022136 104401              TRAP     C$ETST
4883

```

```
4884      .SBTTL      :***** TEST 18 *****
4885      .SBTTL      * READ/WRITE TSS TEST
4886      022140      ZZ
4887      *
4888      *
4889      * THIS TEST CHECKS THAT A TRIB STATUS SLOT CAN
4890      * BE WRITTEN AND READ
4891      *
4892      *
4893      .SBTTL      :***** TEST 18 *****
4894      022140      T18::
4895      022140      012737 000030 002404      MOV      #30,TSSADD      ;START ADD AT 30
4896      022146      005002      NEWSLT: CLR      R2              ;CLEAR R2
4897
4898      022150      004737 004450      NEWPAT: JSR      PC,MINITS      ;MASTER CLEAR MODE DEF
4899
4900      022154      005037 002374      CLR      ERRWRD
4901      022160      104410      TRAP     C$ESCAPE
4902      022162      000240      .WORD   L10057-.
4903      :
4904      : JUMP TO END OF TEST IF ERROR
4905      :
4906
4907      022164      012737 000055 002356      MOV      #55,TRIBN      ;PUT 55 IN TRIB NUMBER
4908      022172      012703 000001      MOV      #01,R3        ;THIS WILL ESTABLISH
4909      022176      004537 003156      JSR      R5,CONTIN     ; A TRIB
4910      :
4911      : TIME OUT AND READY ERRORS REPORT THIS PC
4912      :
4913
4914      022202      005037 002374      CLR      ERRWRD
4915      022206      104410      TRAP     C$ESCAPE
4916      022210      000212      .WORD   L10057-.
4917      :
4918      : JUMP TO END OF TEST IF ERROR
4919      :
4920
4921      022212      016204 033002      MOV      DATLST(R2),R4 ;PATTERN TO BE WRITTEN
4922      022216      013703 002404      MOV      TSSADD,R3     ;WRITE TO TSS
4923      022222      052703 000200      BIS      #BIT7,R3      ;SET THE WRITE BIT
4924      022226      004537 003156      JSR      R5,CONTIN     ; GO DO IT!!!
4925      :
4926      : TIME OUT AND READY ERRORS REPORT THIS PC
4927      :
4928
4929      022232      005037 002374      CLR      ERRWRD
4930      022236      104410      TRAP     C$ESCAPE
4931      022240      000162      .WORD   L10057-.
4932      :
4933      : JUMP TO END OF TEST IF ERROR
4934      :
4935
4936      022242      013703 002404      MOV      TSSADD,R3     ;SET UP TO READ SLOT
4937      022246      052703 000040      BIS      #BIT5,R3      ;SET THE READ BIT
4938      022252      004537 003156      JSR      R5,CONTIN     ;DO CONTROL IN
4939      :
```

```

4940 ; TIME OUT AND READY ERRORS REPORT THIS PC
4941 ;*****
4942
4943 022256 005037 002374 CLR ERRWRD
4944 022262 104410 TRAP C$ESCAPE
4945 022264 000136 .WORD L10057-.
4946 ;*****
4947 ; JUMP TO END OF TEST IF ERROR
4948 ;*****
4949 022266 012737 000002 002334 MOV #02,$GDDAT ; COMPARE FOR A INFO OUT
4950 022274 004537 003236 JSR R5,GETOUT ; CHECK FOR INFO OUT AND
4951 ; CORRECT TRIB NO. IF ERROR
4952 ; REPORT THIS PC.
4953 022300 005037 002374 CLR ERRWRD
4954 022304 104410 TRAP C$ESCAPE
4955 022306 000114 .WORD L10057-.
4956 ;*****
4957 ; JUMP TO END OF TEST IF ERROR
4958 ;*****
4959
4960 022310 013737 002404 002334 MOV TSSADD,$GDDAT ;MOVE EXPECTED ADDRESS TO GDDAT
4961 022316 052737 000040 002334 BIS #BIT5,$GDDAT ;SET THE READ TSS BIT IN EXPECTED
4962 022324 004537 003366 JSR R5,GETRKY ; GO CHECK FOR GOOD RETURN KEY
4963
4964 022330 005037 002374 CLR ERRWRD
4965 022334 104410 TRAP C$ESCAPE
4966 022336 000064 .WORD L10057-.
4967 ;*****
4968 ; JUMP TO END OF TEST IF ERROR
4969 ;*****
4970
4971 022340 016237 033002 002334 30$: MOV DATLST(R2),$GDDAT ;MOVE EXPECTED PATTERN
4972 022346 004537 003442 JSR R5,GETDAT ;GET DATA RETURNED.
4973 ; IF ERROR REPORT THIS PC.
4974 022352 005037 002374 CLR ERRWRD
4975 022356 104410 TRAP C$ESCAPE
4976 022360 000042 .WORD L10057-.
4977 ;*****
4978 ; JUMP TO END OF TEST IF ERROR
4979 ;*****
4980
4981 022362 022762 000562 033002 CMP #562,DATLST(R2) ;ARE WE DONE WITH PATTERN
4982 022370 001404 BEQ 50$ ;IF SO DO NEXT SLOT
4983 022372 062702 000002 ADD #2,R2 ;BUMP LIST POINTER
4984 022376 000137 022150 JMP NEWPAT ;GO BACK FOR THIS PATTERN.
4985 022402 022737 000037 002404 50$: CMP #37,TSSADD ;IS THIS THE LAST SLOT
4986 022410 001404 BEQ 60$ ;IF SO END TEST
4987 022412 005237 002404 INC TSSADD ;ELSE BUMP ADD
4988 022416 000137 022146 JMP NEWSLT ; AND DO NEXT SLOT
4989 022422 60$:
4990
4991 022422 L10057:
4992 022422 104401 TRAP C$ETST
4993
  
```



```
4994
4995
4996
4997 022424
4998
4999
5000
5001
5002
5003
5004
5005 022424
5006
5007 022424 004737 004450
5008
5009 022430 005037 002374
5010 022434 104410
5011 022436 000072
5012
5013
5014
5015
5016 022440 012737 000022 002356
5017 022446 012703 000001
5018 022452 004537 003156
5019
5020
5021
5022
5023 022456 005037 002374
5024 022462 104410
5025 022464 000044
5026
5027
5028
5029
5030 022466 012703 000204
5031 022472 004537 003156
5032
5033
5034
5035
5036 022476 005037 002374
5037 022502 104410
5038 022504 000024
5039
5040
5041
5042
5043 022506 012737 000132 002402
5044 022514 004537 003012
5045
5046 022520 005037 002374
5047 022524 104410
5048 022526 000002
5049

.SBTTL ***** TEST 19 *****
.SBTTL *WRITE RESERVED AREA OF TSS. P.E. 132
ZZ
:*
:*
:* THIS TEST CHECKS FOR PROCEDURE ERROR
:* ON WRITING TO ILLEGAL SLOT
:*
.SBTTL ***** TEST 19 *****
T19::
JSR PC,MINITS ;MASTER CLEAR MODE DEF
CLR ERRWRD
TRAP C$ESCAPE
.WORD L10060-.
*****
: JUMP TO END OF TEST IF ERROR
*****
MOV #22,TRIBN ;SET TRIB NUMBER TO 22
MOV #01,R3
JSR R5,CONTIN ;ESTABLISH TRIB
*****
: TIME OUT OR READY ERRORS REPORT THIS PC
*****
CLR ERRWRD
TRAP C$ESCAPE
.WORD L10060-.
*****
: JUMP TO END OF TEST IF ERROR
*****
MOV #204,R3 ;SEL6=204
JSR R5,CONTIN ;WRITE TSS (ILLEGAL)
*****
: TIME OUT OR READY ERROR REPORT THIS PC
*****
CLR ERRWRD
TRAP C$ESCAPE
.WORD L10060-.
*****
: JUMP TO END OF TEST IF ERROR
*****
MOV #132,PERR ;CHECK FOR PROCEDURE ERROR
JSR R5,WfPE ;WAIT FOR PROCEDURE ERROR
CLR ERRWRD
TRAP C$ESCAPE
.WORD L10060-.
*****
```

5050 : ESCAPE TEST IF ERROR
5051 : : : : : :
5052 : : : : : :
5053 022530 L10060:
5054 022530 104401 TRAP C\$ETST
5055

```

5056
5057 .SBTTL ;***** TEST 20 *****
5058 .SBTTL *READ CLEAR WRONG ADD P.E.132
5059 022532 ZZ
5060 :
5061 :
5062 : * THIS TEST CHECKS FOR PROCEDURE ERROR
5063 : * FOR RD/CLR TSS WRONG ADD(132)
5064 :
5065 :
5066 .SBTTL ;***** TEST 20 *****
5067 022532 TPO::
5068
5069 022532 004737 004450 JSR PC,MINITS ;MASTER CLEAR MODE DEF
5070
5071 022536 005037 002374 CLR ERRWRD
5072 022542 104410 TRAP C$ESCAPE
5073 022544 000072 .WORD L10061-.
5074 :
5075 : JUMP TO END OF TEST IF ERROR
5076 :
5077
5078 022546 012737 000077 002356 MOV #77,TRIBN ;MAKE TRIBN 77
5079 022554 012703 000001 MOV #01,R3 ;ESTABLISH TRIB
5080 022560 004537 003156 JSR R5,CONTIN
5081 :
5082 : TIME OUT OR READY ERROR REPORTS THIS PC
5083 :
5084
5085 022564 005037 002374 CLR ERRWRD
5086 022570 104410 TRAP C$ESCAPE
5087 022572 000044 .WORD L10061-.
5088 :
5089 : JUMP TO END OF TEST IF ERROR
5090 :
5091
5092 022574 012703 000106 MOV #106,R3
5093 022600 004537 003156 JSR R5,CONTIN ;READ/CLEAR ADD 6
5094 :
5095 : TIME OUT OR READY ERRORS REPORT THIS PC
5096 :
5097
5098 022604 005037 002374 CLR ERRWRD
5099 022610 104410 TRAP C$ESCAPE
5100 022612 000024 .WORD L10061-.
5101 :
5102 : JUMP TO END OF TEST IF ERROR
5103 :
5104
5105 022614 012737 000132 002402 MOV #132,PERR ;SET PROCEDURE ERROR TO
5106 : ;BE CHECKED TO 132
5107 022622 004537 003012 JSR R5,WPE ;WAIT FOR PROCEDURE ERROR
5108
5109 022626 005037 002374 CLR ERRWRD
5110 022632 104410 TRAP C$ESCAPE
5111 022634 000002 .WORD L10061-.
  
```

5112
5113
5114
5115
5116 022636
5117 022636 104401

.....
: ESCAPE TEST IF ERROR
:.....

L10061:
TRAP C\$ETST

```
5118
5119
5120 .SBTTL :***** TEST 21 ***
5121 .SBTTL *READ/CLEAR TSS
5122 022640 7Z
5123 :*
5124 :*
5125 :* THIS TEST THAT READ CLEAR WORKS
5126 :*
5127 :*-
5128 .SBTTL :***** TEST 21 *****
5129 022640 T?1::
5130
5131 022640 004737 004450 JSR PC,MINITS ;MASTER CLEAR MODE-DEF
5132
5133 022644 005037 002374 CLR ERRWRD
5134 022650 104410 TRAP C$ESCAPE
5135 022652 000100 .WORD L10062-.
5136 :*****
5137 : JUMP TO END OF TEST IF ERROR
5138 :*****
5139
5140 022654 012737 000003 002356 MOV #03,TRIBN ;SET TRIB NUMBER
5141 022662 012703 000001 MOV #01,R3
5142 022666 004537 003156 JSR R5,CONTIN ;ESTABLISH TRIB
5143 :*****
5144 : TIME OUT OR READY ERRORS REPORT THIS PC
5145 :*****
5146
5147 022672 005037 002374 CLR ERRWRD
5148 022676 104410 TRAP C$ESCAPE
5149 022700 000052 .WORD L10062-.
5150 :*****
5151 : JUMP TO END OF TEST IF ERROR
5152 :*****
5153
5154 022702 012703 000107 MOV #107,R3
5155 022706 004537 003156 JSR R5,CONTIN ;READ/CLEAR ADD 7
5156 :*****
5157 : TIME OUT OR READY ERRORS REPORT THIS PC
5158 :*****
5159
5160 022712 005037 002374 CLR ERRWRD
5161 022716 104410 TRAP C$ESCAPE
5162 022720 000032 .WORD L10062-.
5163 :*****
5164 : JUMP TO END OF TEST IF ERROR
5165 :*****
5166
5167 022722 004737 004172 JSR PC,WAIT50
5168 022726 004737 004172 JSR PC,WAIT50 ;DELAY
5169 022732 032777 000200 157502 BIT #RDO,@BSEL2
5170 022740 001004 BNE 108 ;IF RDO THEN END
5171 :FALSE ERROR
5172 022742 104455 TRAP C$ERDF
5173 022744 000051 .WORD 41
```

CZDMTAD DMP/V-11 PCTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 124

;***** TEST 21 *****

SEQ 0123

5174 022746 011100
5175 022750 007070
5176
5177 022752
5178 022752
5179 022752 104401

.WORD MRF1
.WORD ERR10

10\$:
L10062:

TRAP C\$ETST

```
5180 .SBTTL :***** TEST 22 *****
5181 .SBTTL *GLOBAL STATUS SLOT TESTS
5182 022754 ZZ
5183 *
5184 * THIS TEST CHECKS THAT GLOBAL STATUS
5185 * SLOTS RESPOND TO COMMANDS
5186 *
5187 *
5188 .SBTTL :***** TEST 22 *****
5189 022754 T22::
5190
5191 022754 T22.1:
5192 022754 104402 TRAP C$BSUB
5193 :*****
5194 : READ ALL SLOTS TEST :
5195 :*****
5196
5197 022756 005037 002404 CLR TSSADD ;CLEAR ADD
5198 022752 005037 002356 CLR TRIBN ;MAKE TRIB #0(GLOBAL COMM)
5199 022766
5200 022766 004737 004450 JSR PC,MINITS ;MASTER CLEAR INIT
5201
5202 022772 005037 002374 CLR ERRWRD
5203 022776 104410 TRAP C$ESCAPE
5204 023000 000114 .WORD L10064-.
5205 :*****
5206 : JUMP TO END OF TEST IF ERROR
5207 :*****
5208
5209 023002 013703 002404 MOV TSSADD,R3
5210 023006 052703 000040 BIS #BIT5,R3 ;SET UP READ GSS COMMAND
5211 023012 004537 003156 JSR R5,CONTIN ;GO DO IT
5212
5213 023016 005037 002374 CLR ERRWRD
5214 023022 104410 TRAP C$ESCAPE
5215 023024 000070 .WORD L10064-.
5216 :*****
5217 : JUMP TO END OF TEST IF ERROR
5218 :*****
5219
5220 023026 012737 000002 002334 MOV #2,$GDDAT ;
5221 023034 004537 003236 JSR R5,GETOUT ;CHECK CORRECT TYPE AND
5222 ; TRIB NO.
5223
5224 023040 005037 002374 CLR ERRWRD
5225 023044 104410 TRAP C$ESCAPE
5226 023046 000046 .WORD L10064-.
5227 :*****
5228 : ESCAPE SUB IF ERROR
5229 :*****
5230
5231 023050 10$:
5232 023050 013737 002404 002334 MOV TSSADD,$GDDAT
5233 023056 052737 000040 002334 BIS #BIT5,$GDDAT ;SET THE READ TSS BIT
5234 023064 004537 003366 JSR R5,GETRKY ;CHECK RETURN KEY
5235
```

```

5236 023070 005037 002374          CLR   ERRWRD
5237 023074 104410          TRAP  C$ESCAPE
5238 023076 000016          .WORD L10064-.
5239                               :
5240                               : GO TO END OF SUB IF ERROR
5241                               :
5242                               :
5243                               :
5244 023100 005237 002404          30$: INC   TSSADD      ;BUMP ADDRESS
5245 023104 022737 000040 002404  CMP   #40,TSSADD  ; ARE WE ALL DONE
5246 023112 001325          BNE   5$         ; IF NOT GO BACK
5247                               ;ELSE END SUBTEST
5248 023114          L10064:
5249 023114 104403          TRAP  C$ESUB
5250 023116          122.2:
5251 023116 104402          TRAP  C$BSUB
5252                               :
5253                               : WRITE ALL SLOTS TEST ;
5254                               :
5255                               :
5256 023120 012737 000034 002404          MOV   #34,TSSADD  ;START WITH FIRST WRITABLE ADD
5257 023126 005037 002356          CLR   TRIBN      ;AND TRIBN AT ZERO
5258 023132 004737 004450          40$: JSR   PC,MINITS ;MASTER CLEAR INT
5259                               :
5260 023136 005037 002374          CLR   ERRWRD
5261 023142 104410          TRAP  C$ESCAPE
5262 023144 000226          .WORD L10065-.
5263                               :
5264                               : JUMP TO END OF SUB IF ERROR
5265                               :
5266                               :
5267 023146 013703 002404          45$: MOV   TSSADD,R3
5268 023152 052703 000200          BIS   #BIT7,R3   ;WRITE TSS(GLOBAL BECAUSE TRIBN=0)
5269 023156 013704 002404          MOV   TSSADD,R4  ;PUT IN ADD FOR DATA
5270 023162 004537 003156          JSR   R5,CONTIN  ;DO IT
5271                               :
5272 023166 005037 002374          CLR   ERRWRD
5273 023172 104410          TRAP  C$ESCAPE
5274 023174 000176          .WORD L10065-.
5275                               :
5276                               : JUMP TO END OF SUB IF ERROR
5277                               :
5278                               :
5279 023176 005237 002404          INC   TSSADD      ;BUMP ADD
5280 023202 022737 000040 002404  CMP   #40,TSSADD  ;DONE ALL
5281 023210 001356          BNE   45$        ;NO GO FINISH!!!
5282 023212 012703 000227          MOV   #227,R3
5283 023216 013704 002404          MOV   TSSADD,R4
5284 023222 004537 003156          JSR   R5,CONTIN  ; TRY TO WRITE BAD ADDRESS
5285                               :
5286 023226 005037 002374          CLR   ERRWRD
5287 023232 104410          TRAP  C$ESCAPE
5288 023234 000136          .WORD L10065-.
5289                               :
5290                               : JUMP TO END OF SUB IF ERROR
5291                               :

```



```

5292 023236 012737 000132 002402      MOV    #132,PERR
5293 023244 004537 003012      JSR    R5,WFPD      ;WAIT FOR PROCEDURE ERROR
5294
5295 023250 005037 002374      CLR    ERRWRD
5296 023254 104410      TRAP  C$ESCAPE
5297 023256 000222      .WORD L10063-.
5298
5299      :.:.:.:.
5300      :ESCAPE TEST IF ERROR
5301      :.:.:.:.
5302 023260 042777 000200 157154 GSSREP: BIC    #RDO,@BSEL2      ;CLEAR OUTPUT
5303 023266 005337 002404      DEC    TSSADD      ;GET TSSADD BACK TO MAY
5304 023272 013703 002404      MOV    TSSADD,R3
5305 023276 052703 000040      BIS    #BIT5,R3      ;SET READ BIT
5306 023302 004537 003156      JSR    R5,CONTIN    ;READ TSS
5307
5308 023306 005037 002374      CLR    ERRWRD
5309 023312 104410      TRAP  C$ESCAPE
5310 023314 000056      .WORD L10065-.
5311
5312      :.:.:.:.
5313      :JUMP TO END OF SUB IF ERROR
5314      :.:.:.:.
5315 023316 012737 000002 002334      MOV    #2,$GDDAT
5316 023324 004537 003236      JSR    R5,GETOUT    ;CHECK FOR INFOR. OUT AND
5317
5318 023330 005037 002374      CLR    ERRWRD
5319      :.:.:.
5320 023334 104410      TRAP  C$ESCAPE
5321 023336 000034      .WORD L10065-.
5322
5323      :.:.:.
5324      :JUMP TO END OF SUB IF ERROR
5325 023340 013737 002404 002334 60$: MOV    TSSADD,$GDDAT
5326 023346 004537 003442      JSR    R5,GETDAT    ;CHECK FOR GOOD DATA
5327
5328 023352 005037 002374      CLR    ERRWRD
5329 023356 104410      TRAP  C$ESCAPE
5330 023360 000012      .WORD L10065-.
5331
5332      :.:.:.
5333      :JUMP TO END OF SUB IF ERROR
5334      :.:.:.
5335 023362 022737 000034 002404 70$: CMP    #34,TSSADD      ;ARE WE ALL DONE
5336 023370 001333      BNE    GSSREP      ;GO BACK IF NOT
5337
5338      L10065:
5339 023372 104403      TRAP  C$ESUB
5340 023374      T22.3:
5341 023374 104402      TRAP  C$BSUB
5342
5343      :.:.:.
5344      :READ CLEAR SLOT TEST :
5345      :.:.:.
5346 023376 004737 004450      JSR    PC,MINITS    ;MASTER CLEAR MODE DEF
5347
  
```

```

5348 023402 005037 002374      CLR      ERRWRD
5349 023406 104410              TRAP     C$ESCAPE
5350 023410 000066              .WORD   L10066-.
5351                               :JUMP TO END OF SUB IF ERROR
5352                               :*****
5353                               :*****
5354                               :*****
5355 023412 005037 002356      CLR      TRIBN
5356 023416 012703 000117      MOV     #117,R3
5357 023422 004537 003156      JSR     R5,CONTIN      ;READ CLEAR SLOT
5358                               :*****
5359 023426 005037 002374      CLR      ERRWRD
5360 023432 104410              TRAP     C$ESCAPE
5361 023434 000042              .WORD   L10066-.
5362                               :JUMP TO END OF SUB IF ERROR
5363                               :*****
5364                               :*****
5365                               :*****
5366 023436 012737 000002 002334  MOV     #02,$GDDAT
5367 023444 004537 003236      JSR     R5,GETOUT      ;CHECK FOR INFO OUT
5368                               :AND CORRECT TRIBN.
5369 023450 005037 002374      CLR      ERRWRD
5370 023454 104410              TRAP     C$ESCAPE
5371 023456 000020              .WORD   L10066-.
5372                               :JUMP TO END OF SUB IF ERROR
5373                               :*****
5374                               :*****
5375                               :*****
5376 023460 012737 000117 002334  MOV     #117,$GDDAT
5377 023466 004537 003366      JSR     R5,GETRKY      ;CHECK FOR CORRECT RETURN KEY
5378                               :*****
5379 023472 005037 002374      CLR      ERRWRD
5380 023476 104403              L10066: TRAP     C$ESUB
5381 023476 104403              L10063: TRAP     C$ESUB
5382 023500 104401              TRAP     C$ETST
5383 023500 104401
5384
    
```

5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440

023502

023502

023502 004737 004450
023506 005037 002374
023512 104410
023514 000412
023516 012737 000036
023,24 012703 000001
023530 004537 003156

023534 005037 002374
023540 104410
023542 000364
023544 012703 000003
023550 004537 003156

023554 005037 002374
023560 104410
023562 000344
023564 012737 000001
023572 004537 003236

023576 005037 002374
023602 104410
023604 000322
023606 012737 000024
023614 004537 003616
023620 005037 002374
023624 104410
023626 000300

023630 042777 000200 156604 20\$:
023636 052777 000200 156572
023644 004537 002726

.SBTTL ***** TEST 23 *****
.SBTTL *HALT TRIB COMMAND TEST

.SBTTL ***** TEST 23 *****
T3: :

```
JSR PC,MINITS ;MASTER CLEAR -MODE DEF
CLR ERRWRD ;IF ERROR GO TO END TEST
TRAP C$ESCAPE
.WORD L10067-.
MOV #36,TRIBN ;SET TRIBN
MOV #01,R3
JSR R5,CONTIN ;ESTABLISH TRIB

;IF TIME OUT OR READY ERRORS THE PROGRAM WILL
;REPORT THIS PC AS FAILING PC

CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
TRAP C$ESCAPE
.WORD L10067-.
MOV #03,R3
JSR R5,CONTIN ;ISTRIB TRIB

;IF TIME OUT OR READY ERRORS THE PROGRAM WILL
;REPORT THIS PC AS FAILING PC
CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
TRAP C$ESCAPE
.WORD L10067-.
MOV #01,$GDDAT ;CHECK
JSR R5,GETOUT ;FOR CONTROL OUT AND
;CORRECT TRIBN
;IF ERROR REPORT
;THIS PC AND ESCAPE TEST
; JUMP TO END OF TEST IF ERROR

CLR ERRWRD
TRAP C$ESCAPE
.WORD L10067-.

; QUE REC BUFFER WITH 100 DECIMAL LOCATIONS

BIC #RDO,@BSEL2 ;CLEAR READY OUT
BIS #RQ1,@BSELO ;SET REQUEST IN
;WAIT FOR READY IN
JSR R5,WRDI ;WAIT FOR RDI TO SET
```

```

5441 023650 005037 002374 CLR ERRWRD ;CLEAR ERROR
5442 023654 104410 TRAP C$ESCAPE
5443 023656 000250 .WORD L10067-
5444 023660 042777 000200 156550 BIC #RQ1,@BSEL0 ;CLEAR REQUEST IN
5445 023666 012777 033022 156552 MOV #RECBU1,@BSEL4 ;
5446 023674 012777 000144 156550 MOV #100.,@BSEL6
5447 023702 113777 002356 156534 MOVVB TRIBN,@BSEL3
5448 023710 105077 156526 CLRB @BSEL2 ;QUE UP BUFF
5449 023714 012703 000005 MOV #05,R3
5450 023720 004537 003156 JSR R5,CONTIN ;HALT TRIB
5451 ;IF TIME OUT OR READY ERRORS THE PROGRAM WILL
5452 ;REPORT THIS PC AS FAILING PC
5453
5454 023724 005037 002374 CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
5455 023730 104410 TRAP C$ESCAPE
5456 023732 000174 .WORD L10067-
5457 023734 012737 000003 002334 MOV #03,$GDDAT
5458 023742 004537 003236 JSR R5,GETOUT ;CHECK FOR BUFFER UNUSED
5459 ;AND CORRECT TRIBN.
5460 023746 005037 002374 CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
5461 023752 104410 TRAP C$ESCAPE
5462 023754 000152 .WORD L10067-
5463 023756 012737 000144 002334 MOV #100.,$GDDAT
5464 023764 004537 003510 JSR R5,GETCC ;CHECK FOR GOOD CHAR.COUNT
5465 023770 005037 002374 CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
5466 023774 104410 TRAP C$ESCAPE
5467 023776 000130 .WORD L10067-
5468 024000 012737 033022 002334 MOV #RECBU1,$GDDAT
5469 024006 004537 003556 JSR R5,GETBA ;CHECK FOR GOOD BUFF. ADD.
5470 024012 005037 002374 CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
5471 024016 104410 TRAP C$ESCAPE
5472 024020 000106 .WORD L10067-
5473 024022 042777 000200 156412 23$: BIC #RDO,@BSEL2 ;CLEAR OUTPUT AND LOOK FOR NEXT
5474 024030 012737 000002 002334 MOV #2,$GDDAT
5475 024036 004537 003236 JSR R5,GETOUT ;NEXT GET INFO OUT
5476 024042 005037 002374 CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
5477 024046 104410 TRAP C$ESCAPE
5478 024050 000056 .WORD L10067-
5479 024052 012737 000020 002334 MOV #20,$GDDAT
5480 024060 004537 003366 JSR R5,GETRKY ;CHECK FOR GOOD RETURN KEY
5481 ; BUFFER RETURN COMPLETE
5482 024064 005037 002374 CLR ERRWRD ; JUMP TO END OF TEST IF ERROR
5483 024070 104410 TRAP C$ESCAPE
5484 024072 000034 .WORD L10067-
5485 024074 042777 000200 156340 25$: BIC #RDO,@BSEL2 ;CLEAR OUTPUT
5486 024102 004737 004172 JSR PC,WAIT50
5487 024106 004737 004172 JSR PC,WAIT50 ;WAIT A WHILE
5488 024112 012703 000000 MOV #0,R3 ;DO A NO REQUEST
5489 024116 004537 003156 JSR R5,CONTIN
5490 024122 005037 002374 CLR ERRWRD ;IF ERROR OCCURS THE SECOND
5491 ;HALT TRIB CAUSED AN OUTPUT
5492 ;AND SHOULD NOT HAVE.
5493 024126 L10067:
5494 024126 104401 TRAP C$ETST
  
```

```

5495
5496 .SBTTL ;***** TEST 24 *****
5497 .SBTTL *KILL TRIB TESTS
5498 024130 ZZ
5499 ;*
5500 ;* THIS TEST CHECKS THE KILL TRIB FUNCTIONS
5501 ;*-
5502 .SBTTL ;***** TEST 24 *****
5503 T24::
5504
5505 024130 004737 004450 JSR PC,MINITS
5506
5507 024134 005037 002374 CLR ERRWRD
5508 024140 104410 TRAP C$ESCAPE
5509 024142 000412 .WORD L10070-.
5510 ;*
5511 ; JUMP TO END OF TEST IF ERROR
5512 ;*
5513
5514 024144 012737 000143 002356 MOV #143,TRIBN ;SET TRIB NUMBER
5515 024152 012703 000001 MOV #01,R3
5516 024156 004537 003156 JSR R5,CONTIN ;ESTABLISH TRIB
5517 ;*
5518 ; TIME OUT OR READY ERRORS REPORT THIS P
5519 ;*
5520
5521 024162 005037 002374 CLR ERRWRD
5522 024166 104410 TRAP C$ESCAPE
5523 024170 000364 .WORD L10070-.
5524 ;*
5525 ; JUMP TO END OF TEST IF ERROR
5526 ;*
5527
5528 024172 012703 000041 MOV #41,R3
5529 024176 004537 003156 JSR R5,CONTIN ;READ TRIB STATUS SLOT 1
5530 ;*
5531 ; TIME OUT OR READY ERRORS REPORT THIS PC
5532 ;*
5533
5534 024202 005037 002374 CLR ERRWRD
5535 024206 104410 TRAP C$ESCAPE
5536 024210 000344 .WORD L10070-.
5537 ;*
5538 ; JUMP TO END OF TEST IF ERROR
5539 ;*
5540
5541 024212 012737 000002 002334 MOV #02,$GDDAT ;SET TYPE FOR INFO OUT
5542 024220 004537 003236 JSR R5,GETOUT ;CHECK FOR INFO OUT AND
5543 ;CORRECT TRIBN.
5544
5545 024224 005037 002374 CLR ERRWRD
5546 024230 104410 TRAP C$ESCAPE
5547 024232 000322 .WORD L10070-.
5548 ;*
5549 ; JUMP TO END OF TEST IF ERROR
5550 ;*
  
```

```

5551
5552 024234 000377 156206      20$: SWAB @BSEL4 ;SWAB BYTES
5553 024240 117737 156202 002336 MOV @BSEL4,$BDDAT ;MOVE TRIB ADD TO BDDAT
5554 024246 113737 002356 002334 MOVB TRIBN,$GDDAT ;MOVE TRIB NUMBER TO GDDAT
5555 024254 123737 002334 002336 CMPB $GDDAT,$BDDAT ;COMPARE
5556 024262 001407 BEQ 30$ ;IF OK GO TO 30
5557 ;ELSE ERROR
5558 024264 012737 011264 002422 MOV #M30F,CODEW
5559 024272 104455 TRAP C$ERDF
5560 024274 000052 .WORD 42
5561 024276 010756 .WORD EROIC
5562 024300 007574 .WORD ERR27
5563
5564 024302 042777 000200 156132 30$: BIC #RDO,@BSEL2 ;CLEAR OUTPUT
5565 024310 012703 000004 MOV #04,R3
5566 024314 004537 003156 JSR R5,CONTIN ;MAINT STATE TRIB
5567 ;*****
5568 ;TIME OUT OR READY ERRORS REPORT THIS PC
5569 ;*****
5570
5571 024320 005037 002374 CLR ERRWRD
5572 024324 104410 TRAP C$ESCAPE
5573 024326 000226 .WORD L10070-.
5574 ;*****
5575 ; JUMP TO END OF TEST IF ERROR
5576 ;*****
5577
5578 024330 012703 000002 MOV #02,R3
5579 024334 004537 003156 JSR R5,CONTIN ;KILL TRIB
5580 ;*****
5581 ; TIME OUT OR READY ERRORS REPORT THIS PC
5582 ;*****
5583
5584 024340 005037 002374 CLR ERRWRD
5585 024344 104410 TRAP C$ESCAPE
5586 024346 000206 .WORD L10070-.
5587 ;*****
5588 ; JUMP TO END OF TEST IF ERROR
5589 ;*****
5590
5591 024350 012737 000112 002402 MOV #112,PERR ;CHECK FOR KILL TO UNHALTED
5592 024356 004537 003012 JSR R5,WFPE ;WAIT FOR PROCEDURE ERROR
5593
5594 024362 005037 002374 CLR ERRWRD
5595 024366 104410 TRAP C$ESCAPE
5596 024370 000164 .WORD L10070-.
5597 ;*****
5598 ; ESCAPE TEST IF ERROR
5599 ;*****
5600 024372 042777 000200 156042 BIC #RDO,@BSEL2 ;CLEAR OUTPUT
5601 024400 012703 000005 MOV #05,R3
5602 024404 004537 003156 JSR R5,CONTIN ;HALT TRIB
5603 ;*****
5604 ; TIME OUT OR READY ERRORS REPORT THIS PC
5605 ;*****
5606

```

```

CZDMTAAO DMP/V-11 PCTNL 1ST #1 MACY11 30A(1052) 12-AUG-80 15:53 PAGE 133 C 11
CZDMTA.REL 12-AUG-80 15:52 ;***** TEST 24 ***** SEQ 0132 C
5607 024410 005037 002374 CLR ERRWRD
5608 024414 104410 TRAP C$ESCAPE
5609 024416 000136 .WORD L10070-.
5610 : JUMP TO END OF TEST IF ERROR
5611 : JUMP TO END OF TEST IF ERROR
5612 : JUMP TO END OF TEST IF ERROR
5613 : JUMP TO END OF TEST IF ERROR
5614 024420 012737 000002 002334 MOV #2,$GDDAT
5615 024426 004537 003236 JSR R5,GETOUT ;CHECK FOR INFO OUT
5616 ;AND CORRECT PC
5617
5618 024432 005037 002374 CLR ERRWRD
5619 024436 104410 TRAP C$ESCAPE
5620 024440 000114 .WORD L10070-.
5621 : JUMP TO END OF TEST IF ERROR
5622 : JUMP TO END OF TEST IF ERROR
5623 : JUMP TO END OF TEST IF ERROR
5624 : JUMP TO END OF TEST IF ERROR
5625 024442 012737 000020 002334 MOV #20,$GDDAT
5626 024450 004537 003366 JSR R5,GETRKY ;CHECK FOR GOOD RETURN KEY
5627
5628
5629 024454 005037 002374 CLR ERRWRD
5630 024460 104410 TRAP C$ESCAPE
5631 024462 000072 .WORD L10070-.
5632 : JUMP TO END OF TEST IF ERROR
5633 : JUMP TO END OF TEST IF ERROR
5634 : JUMP TO END OF TEST IF ERROR
5635 : JUMP TO END OF TEST IF ERROR
5636 024464 042777 000200 155750 2$ BIC #RDO,@BSEL2 ;CLEAR OUTPUT
5637 024472 012703 000002 MOV #02,R3 ;KILL TRIB
5638 024476 004537 003156 JSR R5,CONTIN
5639 : JUMP TO END OF TEST IF ERROR
5640 : TIME OUT OR READY ERRORS REPORT THIS PC
5641 : JUMP TO END OF TEST IF ERROR
5642 : JUMP TO END OF TEST IF ERROR
5643 024502 005037 002374 CLR ERRWRD
5644 024506 104410 TRAP C$ESCAPE
5645 024510 000044 .WORD L10070-.
5646 : JUMP TO END OF TEST IF ERROR
5647 : JUMP TO END OF TEST IF ERROR
5648 : JUMP TO END OF TEST IF ERROR
5649 : JUMP TO END OF TEST IF ERROR
5650 024512 012703 000041 MOV #41,R3 ;READ SLOT 1
5651 024516 004537 003156 JSR R5,CONTIN
5652 : JUMP TO END OF TEST IF ERROR
5653 : TIME OUT OR READY ERRORS REPORT THIS PC
5654 : JUMP TO END OF TEST IF ERROR
5655 : JUMP TO END OF TEST IF ERROR
5656 024522 005037 002374 CLR ERRWRD
5657 024526 104410 TRAP C$ESCAPE
5658 024530 000024 .WORD L10070-.
5659 : JUMP TO END OF TEST IF ERROR
5660 : JUMP TO END OF TEST IF ERROR
5661 : JUMP TO END OF TEST IF ERROR
5662 : JUMP TO END OF TEST IF ERROR

```

CZDMTAD DMP/V-11 PCTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 134

;***** TEST 24 *****

SEQ 0133

5663	024532	012737	000106	002402	MOV	#106,PERR	;CHECK FOR PROCEDURE 106 ERROR
5664	024540	004537	003012		JSR	R5,WFPE	;WAIT FOR PROCEDURE ERROR
5665							
5666	024544	005037	002374		CLR	ERRWRD	
5667	024550	104410			TRAP	C\$ESCAPE	
5668	024552	000002			.WORD	L10070-	
5669						
5670					;	ESCAPE TEST IF ERROR	
5671						
5672							
5673	024554			L10070:			
5674	024554	104401			TRAP	C\$ETST	
5675							
5676							


```

5677
5678
5679
5680 024556
5681
5682
5683
5684
5685
5686
5687
5688
5689 024556
5690 024556 012702 002406
5691
5692 024562 004737 004450
5693
5694 024566 005037 002374
5695 024572 104410
5696 024574 000074
5697
5698
5699
5700
5701 024576 052777 000200 155632
5702 024604 004537 002726
5703
5704 024610 005037 002374
5705 024614 104410
5706 024616 000052
5707 024620 042777 000200 155610
5708 024626 112277 155610
5709 024632 012737 000102 002402
5710 024640 004537 003012
5711
5712 024644 005037 002374
5713 024650 104410
5714 024652 000016
5715
5716
5717
5718 024654 042777 000200 155560
5719 024662 022702 002412
5720 024666 001335
5721
5722
5723 024670
5724 024670 104401
5725

```

.SBTTL :***** TEST 25 *****
 .SBTTL *CHECK FOR PROCEDURE ERROR 102
 ZZ
 *
 * THIS TEST CHECKS THAT ILLEGAL TYPE CODES
 * ON INPUT COMMANDS WILL PRODUCE PROCEDURE
 * ERRORS.
 *
 *-----
 .SBTTL :***** TEST 25 *****
 T25::
 MOV #TYLST,R2 ;SET R2 TO START OF LIST
 JSR PC,MINITS ;MASTER CLEAR-MODE DEF
 CLR ERRWRD
 TRAP C\$ESCAPE
 .WORD L10071-
 :
 : JUMP TO END OF TEST IF ERROR
 :
 :
 BIS #RQI,@BSELO ;SET REQUEST
 JSR R5,WRDI ;WAIT FOR RDI TO SET
 CLR ERRWRD ;CLEAR ERROR
 TRAP C\$ESCAPE
 .WORD L10071-
 BIC #RQI,@BSELO ;CLEAR REQUEST
 MOVB (R2)+,@BSEL2 ;DO FIRST BAD CODE
 MOV #102,PERR
 JSR R5,WPE ;WAIT FOR PROCEDURE ERROR
 CLR ERRWRD
 TRAP C\$ESCAPE
 .WORD L10071-
 :
 : ESCAPE TEST IF ERROR
 :
 :
 BIC #RDI,@BSEL2 ;CLEAR READY OUT
 CMP #TYEND,R2 ;IS IT END
 BNE 10\$;IF NOT GO BACK
 L10071:
 TRAP C\$ETST

```
5726  
5727 .SBTTL ;***** TEST 26 *****  
5728 .SBTTL * CHECK FOR PROCEDURE ERROR 110  
5729 024672 ZZ  
5730 *  
5731 *  
5732 * THIS TEST CHECKS FOR THE PROCEDURE ERROR  
5733 * NON-GLOBAL COMMAND TO TRIB ADDRESS OF 0  
5734 *  
5735 *  
5736 .SBTTL ;***** TEST 26 *****  
5737 024672 T26::  
5738  
5739 024672 004737 004450 JSR PC,MINITS ;MASTER CLEAR -MODE-DEF  
5740  
5741 024676 005037 002374 CLR ERRWRD  
5742 024702 104410 TRAP C$ESCAPE  
5743 024704 000050 .WORD L10072-.  
5744 :  
5745 : JUMP TO END OF TEST IF ERROR  
5746 :  
5747 :  
5748 024706 005037 002356 CLR TRIBN ;MAKE TRIB ADDRESS 0  
5749 024712 012703 000001 MOV #01,R3  
5750 024716 004537 003156 JSR R5,CONTIN ;TRY TO DO ISTRT  
5751  
5752 024722 005037 002374 CLR ERRWRD  
5753 024726 104410 TRAP C$ESCAPE  
5754 024730 000024 .WORD L10072-.  
5755 :  
5756 : JUMP TO END OF TEST IF ERROR  
5757 :  
5758 :  
5759 024732 012737 000110 002402 MOV #110,PERR ;CHECK FOR PE OF 110  
5760 024740 004537 003012 JSR R5,WPE ;WAIT FOR PROCEDURE ERROR  
5761  
5762 024744 005037 002374 CLR ERRWRD  
5763 024750 104410 TRAP C$ESCAPE  
5764 024752 000002 .WORD L10072-.  
5765 :  
5766 : ESCAPE TEST IF ERROR  
5767 :  
5768 :  
5769 024754 L10072: TRAP C$ETST  
5770 024754 104401
```

5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826

024756

024756

024756 004737 004450

024762 005037 002374

024766 104410

024770 000212

024772 012737 000003 002356

025000 012703 000001

025004 004537 003156

025010 005037 002374

025014 104410

025016 000164

025020 012703 000007

025024 004537 003156

025030 005037 002374

025034 104410

025036 000144

025040 012737 000120 002402

025046 004537 003012

025052 005037 002374

025056 104410

025060 000122

.SBTTL :***** TEST 27 *****
.SBTTL * CHECKS FOR PROCEDURE ERROR 120

ZZ

THIS TEST ISSUES A CONTROL IN WITH A REQUEST
KEY OF 7 AND LOOKS FOR A PROCEDURE ERROR OF
ILLEGAL REQUEST KEY ON CONTROL IN (120)

.SBTTL :***** TEST 27 *****
I27::

JSR PC,MINITS ;MASTER CLEAR - MODE DEF

CLR ERRWRD

TRAP C\$ESCAPE

.WORD L10073-

: JUMP TO END OF TEST IF ERROR

MOV #3,TRIBN

MOV #01,R3

JSR R5,CONTIN ;ESTABLISH TRIB

: TIME OUT OR READY ERRORS REPORT HERE

CLR ERRWRD

TRAP C\$ESCAPE

.WORD L10073-

: JUMP TO END OF TEST IF ERROR

MOV #07,R3

JSR R5,CONTIN ;DO CONTROL IN WITH KEY OF 07

CLR ERRWRD

TRAP C\$ESCAPE

.WORD L10073-

: JUMP TO END OF TEST IF ERROR

MOV #120,PERR ;LOOK FOR ERROR

JSR R5,W\$PE ;WAIT FOR PROCEDURE ERROR

CLR ERRWRD

TRAP C\$ESCAPE

.WORD L10073-

: ESCAPE TEST IF ERROR

```

5827
5828 025062 042777 000200 155352      BIC      #RDO,@BSEL2      ;CLEAR RDO
5829 025070 012703 000017              MOV      #17,R3           ;MOV 17 TO KEY WORD
5830 025074 004537 003156              JSR      R5,CONTIN       ;DO CONTROL IN WITH KEY OF 17
5831
5832 025100 005037 002374              CLR      ERRWRD
5833 025104 104410              TRAP    C$ESCAPE
5834 025106 000074              .WORD   L10073-.
5835
5836              ;*****
5837              ; JUMP TO END OF TEST IF ERROR
5838              ;*****
5839 025110 012737 000120 002402      MOV      #120,PERR       ;LOOK FOR ERROR
5840 025116 004537 003012              JSR      R5,WFPE         ;WAIT FOR PROCEDURE ERROR
5841
5842 025122 005037 002374              CLR      ERRWRD
5843 025126 104410              TRAP    C$ESCAPE
5844 025130 000052              .WORD   L10073-.
5845
5846              ;*****
5847              ; ESCAPE TEST IF ERROR
5848              ;*****
5848 025132 042777 000200 155302      BIC      #RDO,@BSEL2      ;CLEAR OUTPUT
5849 025140 005003              CLR      R3
5850 025142 004537 003156              JSR      R5,CONTIN       ;DO A NO REQUEST
5851
5852 025146 005037 002374              CLR      ERRWRD
5853 025152 104410              TRAP    C$ESCAPE
5854 025154 000026              .WORD   L10073-.
5855
5856              ;*****
5857              ; JUMP TO END OF TEST IF ERROR
5858              ;*****
5859 025156 004737 004172              JSR      PC,WAIT50       ;THEN DELAY
5860 025162 032777 000200 155252      BIT      #RDO,@BSEL2
5861 025170 001404              BEQ     10$              ;IF NOT SET THEN END
5862              ;ELSE ERROR
5863 025172 104455              TRAP    C$ERDF
5864 025174 000053              .WORD   43
5865 025176 011730              .WORD   MEF30
5866 025200 007672              .WORD   ERR32
5867 025202
5868 025202
5869 025202 104401              TRAP    C$ETST
5870

```

10\$:
L10073:

```

5871 .SBTTL ;***** TEST 28 *****
5872 .SBTTL * CHECK FOR PROCEDURE ERROR 134
5873 025204 ZZ
5874
5875
5876
5877
5878
5879
5880 .SBTTL ;***** TEST 28 *****
5881 025204 T28::
5882
5883 025204 004737 004450 JSR PC,MINITS ;MASTER CLEAR MODE DEF
5884
5885 025210 005037 002374 CLR ERRWRD
5886 025214 104410 TRAP C$ESCAPE
5887 025216 000212 .WORD L10074-.
5888
5889
5890
5891
5892 025220 012737 000005 002356 MOV #5,TRIBN
5893 025226 012703 000001 MOV #1,R3
5894 025232 004537 003156 JSR R5,CONTIN ;ESTABLISH TRIB
5895
5896
5897
5898
5899 025236 005037 002374 CLR ERRWRD
5900 025242 104410 TRAP C$ESCAPE
5901 025244 000164 .WORD L10074-.
5902
5903
5904
5905
5906 025246 012703 100000 MOV #100000,R3 ;SET BIT 7
5907 025252 004537 003156 JSR R5,CONTIN
5908
5909 025256 005037 002374 CLR ERRWRD
5910 025262 104410 TRAP C$ESCAPE
5911 025264 000144 .WORD L10074-.
5912
5913
5914
5915
5916 025266 012737 000134 002402 MOV #134,PERR
5917 025274 004537 003012 JSR R5,WFPE ;WAIT FOR PROCEDURE ERROR
5918
5919 025300 005037 002374 CLR ERRWRD
5920 025304 104410 TRAP C$ESCAPE
5921 025306 000122 .WORD L10074-.
5922
5923
5924
5925 025310 042777 000200 155124 BIC #RDO,ABSEL2 ;CLEAR OUTPUT
5926 025316 012703 046000 MOV #46000,R3 ;SET 6 3 AND2
  
```

THIS TEST CHECKS FOR PROCEDURE ERROR OF USING RESERVED BITS IN BSEL7 ON CONTROL IN

```

5927 025322 004537 003156 JSR R5,CONTIN ;DO CONTROL IN
5928
5929 025326 005037 002374 CLR ERRWRD
5930 025332 104410 TRAP C$ESCAPE
5931 025334 000074 .WORD L10074-.
5932
5933 ;*****
5934 ; JUMP TO END OF TEST IF ERROR
5935 ;*****
5936 025336 012737 000134 002402 MOV #134,PERR
5937 025344 004537 003012 JSR R5,WFPE ;WAIT FOR PROCEDURE ERROR
5938
5939 025350 005037 002374 CLR ERRWRD
5940 025354 104410 TRAP C$ESCAPE
5941 025356 000052 .WORD L10074-.
5942
5943 ;*****
5944 ; ESCAPE TEST IF ERROR
5945 ;*****
5945 025360 005003 CLR R3
5946 025362 042777 000200 155052 BIC #RDO,@BSEL2 ;CLEAR OUTPUT
5947 025370 004537 003156 JSR R5,CONTIN ;DO CONTROL IN
5948
5949 025374 005037 002374 CLR ERRWRD
5950 025400 104410 TRAP C$ESCAPE
5951 025402 000026 .WORD L10074-.
5952
5953 ;*****
5954 ; JUMP TO END OF TEST IF ERROR
5955 ;*****
5956 025404 004737 004172 JSR PC,WAIT50 ;WAIT A WHILE
5957 025410 032777 000200 155024 BIT #RDO,@BSEL2 ;IS RDO SET NOW
5958 025416 001404 BEQ 10$ ;IF NOT THEN GO ON
5959 ;ELSE ERROR
5960 025420 104455 TRAP C$ERDF
5961 025422 000054 .WORD 44
5962 025424 011730 .WORD MEF30
5963 025426 007672 .WORD ERR32
5964 025430
5965 025430 10$:
5966 025430 104401 L10074: TRAP C$ETST
5967
  
```

```
5968 .SBTTL :***** TEST 29 *****
5969 .SBTTL *LATCH - UNLATCH POLL CHECK
5970 025432 ZZ
5971 :
5972 :
5973 :
5974 : THIS TEST CHECKS THE LATCH - UNLATCH POLL
5975 : COMMANDS. FIRST LATCH TRIB IN DEAD STATE
5976 : MAKE SURE ITS DEAD. THEN UNLATCH AND MAKE
5977 : SURE THAT IT GOES ACTIVE.
5978 :
5979 .SBTTL :***** TEST 29 *****
5980 025432 T29::
5981 :
5982 025432 004737 004450 JSR PC,MINITS ;MASTER CLEAR MODE DEF
5983 :
5984 025436 005037 002374 CLR ERRWRD
5985 025442 104410 TRAP C$ESCAPE
5986 025444 000312 .WORD L10075-.
5987 :
5988 : JUMP TO END OF TEST IF ERROR
5989 :
5990 :
5991 025446 012737 000027 002356 MOV #27,TRIBN
5992 025454 012703 000001 MOV #01,R3 ;ESTABLISH TRIB
5993 025460 004537 003156 JSR R5,CONTIN
5994 :
5995 : TIME OUT OR READY ERROR REPORT THIS PC
5996 :
5997 :
5998 025464 005037 002374 CLR ERRWRD
5999 025470 104410 TRAP C$ESCAPE
6000 025472 000264 .WORD L10075-.
6001 :
6002 : JUMP TO END OF TEST IF ERROR
6003 :
6004 :
6005 025474 012704 000003 MOV #3,R4
6006 025500 012703 020000 MOV #20000,R3 ;LATCH POLL DEAD
6007 025504 004537 003156 JSR R5,CONTIN
6008 :
6009 : TIME OUT OR READY ERRORS REPORT THIS PC
6010 :
6011 :
6012 025510 005037 002374 CLR ERRWRD
6013 025514 104410 TRAP C$ESCAPE
6014 025516 000240 .WORD L10075-.
6015 :
6016 : JUMP TO END OF TEST IF ERROR
6017 :
6018 :
6019 025520 012703 000042 MOV #42,R3
6020 025524 004537 003156 JSR R5,CONTIN ;READ SLOT 2 TSS
6021 :
6022 : TIME OUT OR READY ERRORS REPORT THIS PC
6023 :
```

```

6024
6025 025530 005037 002374 CLR ERRWRD
6026 025534 104410 TRAP C$ESCAPE
6027 025536 000220 .WORD L10075-.
6028
6029 : JUMP TO END OF TEST IF ERROR
6030 :
6031
6032 025540 012737 000002 002334 MOV #02,$GDDAT
6033 025546 004537 003236 JSR R5,GETOUT
6034
6035 : CHECK FOR INFO OUT AND CORRECT TRIBN
6036 : IF ERROR REPORT THIS PC
6037 :
6038
6039 025552 005037 002374 CLR ERRWRD
6040 025556 104410 TRAP C$ESCAPE
6041 025560 000176 .WORD L10075-.
6042
6043 : JUMP TO END OF TEST IF ERROR
6044 :
6045
6046 025562 012737 000042 002334 MOV #42,$GDDAT
6047 025570 004537 003366 JSR R5,GETRKY
6048
6049 : CHECK FOR CORRECT RETURN KEY
6050 :
6051
6052 025574 005037 002374 CLR ERRWRD
6053 025600 104410 TRAP C$ESCAPE
6054 025602 000154 .WORD L10075-.
6055
6056 : JUMP TO END OF TEST IF ERROR
6057 :
6058
6059 025604 012737 100220 002334 MOV #100220,$GDDAT ;
6060 025612 004537 003442 JSR R5,GETDAT
6061
6062 : CHECK FOR DEAD STATE
6063 :
6064
6065 025616 005037 002374 CLR ERRWRD
6066 025622 104410 TRAP C$ESCAPE
6067 025624 000132 .WORD L10075-.
6068
6069 : JUMP TO END OF TEST IF ERROR
6070 :
6071
6072 025626 042777 000200 154606 BIC #RDO,@BSEL2 ;CLEAR OUTPUT
6073 025634 012703 010000 MOV #10000,R3
6074 025640 004537 003156 JSR R5,CONTIN ;UNLATCH POLL
6075
6076 : TIME OUT OR READY ERRORS REPORT THIS PC
6077 :
6078
6079 025644 005037 002374 CLR ERRWRD
  
```



```

6080 025650 104410 TRAP C$ESCAPE
6081 025652 000104 .WORD L10075-.
6082
6083 : JUMP TO END OF TEST IF ERROR
6084 : JUMP TO END OF TEST IF ERROR
6085
6086 025654 012703 000042 MOV #42,R3
6087 025660 004537 003156 JSR R5,CONTIN ;READ TSS SLOT 2
6088
6089 : TIME OUT OR READY ERRORS REPORT THIS PC
6090 : JUMP TO END OF TEST IF ERROR
6091
6092 025664 005037 002374 CLR ERRWRD
6093 025670 104410 TRAP C$ESCAPE
6094 025672 000064 .WORD L10075-.
6095
6096 : JUMP TO END OF TEST IF ERROR
6097 : JUMP TO END OF TEST IF ERROR
6098
6099 025674 012737 000002 002334 MOV #02,$GDDAT ;
6100 025702 004537 003236 JSR R5,GETOUT
6101
6102 : CHECK FOR INFORMATION OUT AND CORRECT TRIBN
6103 : JUMP TO END OF TEST IF ERROR
6104
6105 025706 005037 002374 CLR ERRWRD
6106 025712 104410 TRAP C$ESCAPE
6107 025714 000042 .WORD L10075-.
6108
6109 : JUMP TO END OF TEST IF ERROR
6110 : JUMP TO END OF TEST IF ERROR
6111
6112 025716 012737 000042 002334 MOV #42,$GDDAT
6113 025724 004537 003366 JSR R5,GETRKY
6114
6115 : CHECK FOR CORRECT RETURN KEY
6116 : JUMP TO END OF TEST IF ERROR
6117
6118 025730 005037 002374 CLR ERRWRD
6119 025734 104410 TRAP C$ESCAPE
6120 025736 000020 .WORD L10075-.
6121
6122 : JUMP TO END OF TEST IF ERROR
6123 : JUMP TO END OF TEST IF ERROR
6124
6125 025740 012737 000600 002334 MOV #600,$GDDAT
6126 025746 004537 003442 JSR R5,GETDAT
6127
6128 : CHECK FOR ACTIVE STATE
6129 : JUMP TO END OF TEST IF ERROR
6130 025752 005037 002374 CLR ERRWRD
6131 025756 L10075: TRAP C$SETST
6132 025756 104401
6133

```

6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171

025760

025760

025760

025766

025774

026002

026010

026014

026022

026026

026032

026034

026036

026042

026046

026050

026052

026052

012737
012737
012737
012737
005037
012737
004737

005037
104410
000016

001750
033022
032542
000004
002424
000003
004450

002374

002416
002414
002412
002420
002356

.SBTTL ***** TEST 30 *****
.SBTTL SHORT MESSAGE SENDING TEST, WITH INTERNAL LOOPBACK

7Z
:*
:* THIS TEST SENDS A 4 BYTE MESSAGE FROM AN EVEN TRANSMIT
:* BUFFER TO AN EVEN REC BUFFER IN DDCMP FORMAT CONFIGURED
:* AS A MULTIPOINT CONTROL STATION FULL DUPLEX. THE TEST
:* CHECKS THAT REC BUFFERS ARE RETURNED AND THAT THE DATA
:* IS CORRECT. IT ALSO CHECKS THAT THE NEXT OUTPUT COMMAND
:* IS A TX BUFFER COMPLETE.

.SBTTL ***** TEST 30 *****

T30::

MOV #1000,RXCC ;SET UP RX CC
MOV #RECBU1,RXADD ;SET UP RX ADD
MOV #MR1,TXADD ;SET UP TX ADD
MOV #4,TXCC ;SET UP TX COUNT
CLR GENWRD ;CLEAR GEN WORD
MOV #3,TRIBN ;SET TRIB ADDRESS
JSR PC,MINITS ;INITIALIZE

CLR ERRWRD
TRAP C\$ESCAPE
.WORD L10076-
.....
: JUMP TO END OF TEST IF ERROR
.....

JSR R5,TXRX3 ;TRANSMIT AND REC.
CLR ERRWRD
TRAP C\$ESCAPE
.WORD L10076-

L10076:

TRAP C\$ETST

```
6172 .SBTTL ;***** TEST 31 *****
6173 .SBTTL * PROCEEDURE ERROR 122 CHECK
6174 026054 ZZ
6175 :
6176 :
6177 : * THIS TEST CHECKS FOR PROCEDURE ERROR 122
6178 : * ESTABLISH BUFFER FOR UNESTABLISHED TRIB.
6179 :
6180 .SBTTL ;***** TEST 31 *****
6181 026054 T31::
6182
6183 026054 004737 004450 JSR PC,MINITS ;MASTER CLEAR MODE DEF
6184
6185 026060 005037 002374 CLR ERRWRD
6186 026064 104410 TRAP C$ESCAPE
6187 026066 000104 .WORD L10077-.
6188 :
6189 : JUMP TO END OF TEST IF ERROR
6190 :
6191
6192 026070 052777 000200 154340 BIS #RQ1,@BSELO ;SET REQUEST FOR INPUT
6193 026076 004537 002726 JSR R5,WRDI ;WAIT FOR RDI TO SET
6194
6195 026102 005037 002374 CLR ERRWRD ;CLEAR ERROR
6196 026106 104410 TRAP C$ESCAPE
6197 026110 000062 .WORD L10077-.
6198 026112 042777 000200 154316 BIC #RQ1,@BSELO ;CLEAR REQUEST
6199 026120 112777 000003 154316 MOVB #03,@BSEL3
6200 026126 012777 000010 154316 MOV #10,@BSEL6 ;SET CC
6201 026134 012777 033022 154304 MOV #RECBU1,@BSEL4 ;SET BA
6202 026142 112777 000000 154272 MOVB #0,@BSEL2 ;ESTABLISH BUFFER
6203 026150 012737 000122 002402 MOV #122,PERR ;WAIT
6204 026156 004537 003012 JSR R5,WFPE ;WAIT FOR PROCEDURE ERROR
6205
6206 026162 005037 002374 CLR ERRWRD
6207 026166 104410 TRAP C$ESCAPE
6208 026170 000002 .WORD L10077-.
6209 :
6210 : ESCAPE TEST IF ERROR
6211 :
6212 026172 104401 L10077:
6213 026172 104401 TRAP C$ETST
6214
```

```

6215 .SBTTL ;***** TEST 32 *****
6216 .SBTTL *PROCEDURE ERROR 124 CHECK
6217 026174 ZZ
6218 *
6219 *
6220 * TH1: TEST CHECKS FOR PROCEDURE ERROR 124
6221 * ESTABLISHING BUFFER FOR HALTED TRIB
6222 *
6223 .SBTTL ;***** TEST 32 *****
6224 026174 T32::
6225
6226 026174 004737 004450 JSR PC,MINITS ;MASTER CLEAR MODE DEF
6227
6228 026200 005037 002374 CLR ERRWRD
6229 026204 104410 TRAP C$ESCAPE
6230 026206 000132 .WORD L10100-.
6231
6232 ; JUMP TO END OF TEST IF ERROR
6233 ;
6234
6235 026210 012737 000047 002356 MOV #47,TRIBN
6236 026216 012703 000001 MOV #01,R3
6237 026222 004537 003156 JSR R5,CONTIN ;ESTABLISH TRIB
6238
6239 ; TIME OUT AND READY ERRORS REPORT THIS PC
6240 ;
6241
6242 026226 005037 002374 CLR ERRWRD
6243 026232 104410 TRAP C$ESCAPE
6244 026234 000104 .WORD L10100-.
6245
6246 ; JUMP TO END OF TEST IF ERROR
6247 ;
6248
6249 026236 052777 000200 154172 BIS #RQI,@BSELO ;SET REQUEST
6250 026244 004537 002726 JSR R5,WRDI ;WAIT FOR RDI TO SET
6251
6252 026250 005037 002374 CLR ERRWRD ;CLEAR ERROR
6253 026254 104410 TRAP C$ESCAPE
6254 026256 000062 .WORD L10100-.
6255 026260 042777 000200 154150 BIC #RQI,@BSELO ;CLEAR REQUEST
6256 026266 012777 000010 154156 MOV #10,@BSEL6 ;SET CC
6257 026274 012777 033022 154144 MOV #RECBU1,@BSEL4 ;SET BA
6258 026302 113777 002356 154134 MOVB TRIBN,@BSEL3 ;SET TRIB NO.
6259 026310 112777 000000 154124 MOVB #0,@BSEL2 ;ESTABLISH BUFFER
6260 026316 012737 000124 002402 MOV #124,PERR ;WAIT FOR ERROR
6261 026324 004537 003012 JSR R5,WFPE ;WAIT FOR PROCEDURE ERROR
6262
6263 026330 005037 002374 CLR ERRWRD
6264 026334 104410 TRAP C$ESCAPE
6265 026336 000002 .WORD L10100-.
6266
6267 ; ESCAPE TEST IF ERROR
6268 ;
6269
6270 026340 104401 L10100: TRAP C$ETST
    
```

6271
6272
6273
6274

6275
6276
6277 026342
6278
6279
6280
6281
6282
6283
6284
6285 026342
6286
6287 026342 004737 004450
6288
6289 026346 005037 002374
6290 026352 104410
6291 026354 000152
6292
6293
6294
6295
6296 026356 012737 000074 002356
6297 026364 012703 000001
6298 026370 004537 003156
6299
6300 026374 005037 002374
6301 026400 104410
6302 026402 000124
6303
6304
6305
6306
6307 026404 012703 000004
6308 026410 004537 003156
6309
6310 026414 005037 002374
6311 026420 104410
6312 026422 000104
6313
6314
6315
6316
6317 026424 052777 000200 154004
6318 026432 004537 002726
6319
6320 026436 005037 002374
6321 026442 104410
6322 026444 000062
6323 026446 042777 000200 153762
6324 026454 012777 000000 153770
6325 026462 012777 033022 153756
6326 026470 113777 002356 153746
6327 026476 112777 000000 153736
6328 026504 012737 000126 002402
6329 026512 004537 003012
6330

```
.SBTTL ***** TEST 33 *****  
.SBTTL *PROCEDURE ERROR #126 CHECK  
ZZ  
*  
*  
* THIS TEST CHECKS FOR A PROCEDURE ERROR OF #126  
* ASSIGNING A BUFFER WITH A ZERO BYTE COUNT  
*  
*  
.SBTTL ***** TEST 33 *****  
T33::  
JSR PC,MINITS ;MASTER CLEAR MODE DEF  
  
CLR ERRWRD  
TRAP C$ESCAPE  
.WORD L10101-  
:  
: JUMP TO END OF TEST IF ERROR  
:  
:  
  
MOV #74,TRIBN  
MOV #01,R3  
JSR R5,CONTIN ; ESTABLISH TRIB  
  
CLR ERRWRD  
TRAP C$ESCAPE  
.WORD L10101-  
:  
: JUMP TO END OF TEST IF ERROR  
:  
:  
  
MOV #04,R3  
JSR R5,CONTIN ;MAINT STATE  
  
CLR ERRWRD  
TRAP C$ESCAPE  
.WORD L10101-  
:  
: JUMP TO END OF TEST IF ERROR  
:  
:  
  
BIS #RQI,@BSELO ;SET REQUEST  
JSR R5,WRDI ;WAIT FOR RDI TO SET  
  
CLR ERRWRD ;CLEAR ERROR  
TRAP C$ESCAPE  
.WORD L10101-  
BIC #RQI,@BSELO ;CLEAR REQUEST  
MOV #0,@BSEL6 ;0 BYTES  
MOV #RECBU1,@BSEL4 ;BA  
MOVB TRIBN,@BSEL3 ;SET TRIBN  
MOVB #0,@BSEL2 ;SET BUFFER  
MOV #126,PERR ;WAIT FOR ERROR  
JSR R5,WFPF ;WAIT FOR PROCEDURE ERROR
```

6331 026516 005037 002374
6332 026522 104410
6333 026524 000002
6334
6335
6336
6337
6338 026526
6339 026526 104401
6340

CLR ERRWRD
TRAP C\$ESCAPE
.WORD L10101-
: : : : :
: ESCAPE TEST IF ERROR
: : : : :
L10101:
TRAP C\$ETST

```
6341
6342
6343
6344 026530
6345
6346
6347
6348
6349
6350
6351
6352 026530
6353
6354 026530 004737 004450
6355
6356 026534 005037 002374
6357 026540 104410
6358 026542 000102
6359
6360
6361
6362
6363 026544 052777 000200 153664
6364
6365 026552 004537 002726
6366
6367 026556 005037 002374
6368 026562 104410
6369 026564 000060
6370 026566 042777 000200 153642
6371 026574 105077 153644
6372 026600 012777 000010 153644
6373 026606 012777 032542 153632
6374 026614 012777 000004 153620
6375
6376 026622 012737 000130 002402
6377 026630 004537 003012
6378
6379 026634 005037 002374
6380 026640 104410
6381 026642 000002
6382
6383
6384
6385 026644
6386 026644 104401
6387
6388
6389
6390
6391
6392
6393
6394 026646
6395
6396

.SBTTL :***** TEST 34 *****
.SBTTL *CHECK FOR PROCEDURE ERROR 130
ZZ
*
*
* THIS TEST CHECKS FOR A PROCEDURE ERROR OF 130
* ASSIGNING TRANSMIT BUFFER FOR TRIB 0
*
*
.SBTTL :***** TEST 34 *****
T34::
JSR PC,MINITS ;MASTER CLEAR MODE DEF
CLR ERRWRD
TRAP C$ESCAPE
.WORD L10102-
:
: JUMP TO END OF TEST IF ERROR
:
BIS #RQ1,@BSEL0 ;SET REQUEST
;WAIT FOR READY
JSR R5,WRDI ;WAIT FOR RDI TO SET
CLR ERRWRD ;CLEAR ERROR
TRAP C$ESCAPE
.WORD L10102-
BIC #RQ1,@BSEL0 ;CLEAR REQUEST.
CLRB @BSEL3 ;MAKE TRIB NO. 0
MOV #10,@BSEL6 ;MAKE CC 10
MOV #MR1,@BSEL4 ;MAKE ADD MR1
MOV #4,@BSEL2 ;CLEAR RDI AND ESTAB
;TRANSMIT BUFFER
MOV #130,PERR ;WAIT FOR PROCEDURE ERROR
JSR R5,W$PE ;WAIT FOR PROCEDURE ERROR
CLR ERRWRD
TRAP C$ESCAPE
.WORD L10102-
:
: ESCAPE TEST IF ERROR
:
L10102:
TRAP C$ETST

.SBTTL :***** TEST 35 *****
.SBTTL * TRANSMIT REC 256,PTP,DDCMP
ZZ
*
*
```



```
6397      ; * THIS TEST WILL TRANSMIT 256 BYTE MESSAGE
6398      ; * DDCMP PROTOCOL
6399      ; * THIS WILL BE DONE EXTERNAL LOOPBACK IF
6400      ; * IT EXISTS ELSE INTERNAL LOOPBACK WILL
6401      ; * BE USED
6402      ; *
6403      ; *
6404      ; *-----
6405      ; ***** TEST 35 *****
6405 026646      .SBTTL
6406 026646 012737 000035 002366      I35::      MOV      #35,ROMN1      ;SET UP TEST NUMBER
6407
6408 026654 012737 001750 002416      MOV      #1000.,RXCC      ;SET REC BUFFER FOR 1000BYTES
6409 026662 012737 033022 002414      MOV      #RECBU1,RXADD      ;SET UP BUFF ADD
6410 026670 012737 000400 002420      MOV      #256.,TXCC      ;SET UP RX CHAR COUNT
6411 026676 012737 032542 002412      MOV      #MR1,TXADD      ;AND ADDRESS
6412 026704 005037 002424      CLR      GENWRD      ;CLEAR GEN WORD
6413 026710 012737 000001 002356      MOV      #01,TRIBN      ;SET THE TRIB TO 01
6414 026716 004737 004440      JSR      PC,INIT1      ;MASTER CLEAR-MODF DEF
6415
6416
6417 026722 005037 002374      CLR      ERRWRD
6418 026726 104410      TRAP     C$ESCAPE
6419 026730 000016      .WORD   L10103-.
6420
6421      ; *****
6422      ; JUMP TO END OF TEST IF ERROR
6423      ; *****
6424 026732 004537 005542      JSR      R5,TXRXSR      ;GO TRANSMIT RX AND CHECK
6425
6426 026736 005037 002374      CLR      ERRWRD
6427 026742 104410      TRAP     C$ESCAPE
6428 026744 000002      .WORD   L10103-.
6429
6430      ; *****
6431      ; JUMP TO END OF TEST IF ERROR
6432      ; *****
6433 026746      L10103:
6434 026746 104401      TRAP     C$EST1
6435      .SBTTL      ; ***** TEST 36 *****
6436      .SBTTL      ; * TX AND RX,255 BYTES,EVEN TX,ODD RX,DDCMP,MULTIPOINT
6437 026750      ZZ
6438      ; *
6439      ; *
6440      ; * THIS TEST WILL TRANSMIT A MESSAGE OF 255
6441      ; * BYTES FROM AN EVEN TX START ADD TO AN ODD
6442      ; * REC START ADD. IN DDCMP MODE MULTI POINT
6443      ; * CONTROL STATION.
6444      ; *
6445      ; *
6446      ; *-----
6447      ; ***** TEST 36 *****
6447 026750      .SBTTL
6448 026750 012737 000377 002420      I36::      MOV      #255.,TXCC      ;SET UP TRANSMIT CHAR COUNT
6449 026756 012737 032542 002412      MOV      #MR1,TXADD      ;SET UP TRANSMIT ADD
6450 026764 012737 000764 002416      MOV      #500.,RXCC      ;SET UP REC CHAR COUNT
6451 026772 012737 033023 002414      MOV      #RECBU1+1,RXADD      ;SET UP REC ADD
6452 027000 012737 000075 002356      MOV      #75,TRIBN      ;SET UP TRIB NO.
```

```

6453 027006 005037 002424          CLR    GENWRD          ;CLEAR THE GENWRD
6454 027012 004737 004450          JSR    PC,MINITS      ;MASTER CLEAR MODE DEF
6455                                     ;MULTI POINT CONTROL
6456
6457 027016 005037 002374          CLR    ERRWRD
6458 027022 104410          TRAP  C$ESCAPE
6459 027024 000016          .WORD L10104-.
6460                                     ;*****
6461                                     ; JUMP TO END OF TEST IF ERROR
6462                                     ;*****
6463
6464 027026 004537 006154          208: JSR    R5,TRRX3      ;GO TRANSMIT AND REC
6465                                     ; AND CHECK DATA.
6466
6467 027032 005037 002374          CLR    ERRWRD
6468 027036 104410          TRAP  C$ESCAPE
6469 027040 000002          .WORD L10104-.
6470                                     ;*****
6471                                     ; JUMP TO END OF TEST IF ERROR
6472                                     ;*****
6473
6474 027042          L10104:
6475 027042 104401          TRAP  C$ETST
6476
  
```

```

6477 .SBTTL ;***** TEST 37 *****
6478 .SBTTL *READ/WRITE MODEM TESTS
6479 027044 ZZ
6480 :
6481 :
6482 :
6483 : * THIS TEST WILL SELECT EACH OF THE 4 MODEM
6484 : * INTERFACES AND WRITE AND READ THEM. THIS IS
6485 : * ONLY DONE IF CONNECTORS ARE PRESENT
6486 :
6487 .SBTTL ;***** TEST 37 *****
6488 027044 T37::
6489
6490 027044 005002 CLR R2 ;CLEAR R2
6491 027046 022737 000000 002472 CMP #0,TSTCON ;IS THIS WITH TEST CONN LOOPBACK
6492 027054 001135 BNE MODEX ;IF NOT GO TO END
6493 027056 016237 027430 002502 MODEB: MOV DUPTYP(R2),AXNUM
6494 027064 004737 004456 JSR PC,MINTR ;MASTER CLEAR MODF DEF
6495 027070 004737 004172 JSR PC,WAIT50 ;DELAY
6496
6497 027074 005037 002374 CLR ERRWRD
6498 027100 104410 TRAP C$ESCAPE
6499 027102 000352 .WORD L10105-.
6500 :
6501 :
6502 :
6503 :
6504 :
6505 :
6506 027112 142777 000010 153320 BICB #BIT3,@BSEL1
6507
6508 027120 005037 002356 MODEA: CLR TRIBN ;MAKE TRIB NO. = 0
6509 027124 016204 027354 MOV MODTYP(R2),R4 ;SELECT TYPE OF INTERFACE
6510 027130 012703 000023 MOV #23,R3 ;SELECT INTERFACE
6511 027134 004537 003156 JSR R5,CONTIN
6512 :
6513 : TIME OUT OR READY ERRORS REPORT THIS PC
6514 :
6515 :
6516 027140 005037 002374 CLR ERRWRD
6517 027144 104410 TRAP C$ESCAPE
6518 027146 000306 .WORD L10105-.
6519 :
6520 :
6521 :
6522 :
6523 :
6524 :
6525 :
6526 :
6527 :
6528 :
6529 027164 005037 002374 CLR ERRWRD
6530 027170 104410 TRAP C$ESCAPE
6531 027172 000262 .WORD L10105-.
6532 :
  
```

```
6533          : JUMP TO END OF TEST IF ERROR
6534          :*****:
6535
6536 027174 012703 000020      MOV    #20,R3
6537 027200 004537 003156      JSR    R5,CONTIN      ;READ MODEM
6538          :*****:
6539          : TIME OUT OR READY ERRORS REPORT THIS PC
6540          :*****:
6541
6542 027204 005037 002374      CLR    ERRWRD
6543 027210 104410              TRAP   C$ESCAPE
6544 027212 000742              .WORD L10105-.
6545          :*****:
6546          : JUMP TO END OF TEST IF ERROR
6547          :*****:
6548
6549 027214 012737 000002 002334  MOV    #02,$GDDAT
6550 027222 004537 003236      JSR    R5,GETOUT
6551          :*****:
6552          : CHECK FOR INFORMATION OUT AND CORRECT TRIBN
6553          :*****:
6554
6555 027226 005037 002374      CLR    ERRWRD
6556 027232 104410              TRAP   C$ESCAPE
6557 027234 000220              .WORD L10105-.
6558          :*****:
6559          : JUMP TO END OF TEST IF ERROR
6560          :*****:
6561
6562 027236 012737 000010 002334  MOV    #10,$GDDAT
6563 027244 004537 003366      JSR    R5,GETRKY
6564          :*****:
6565          : CHECK FOR CORRECT RETURN KEY MODEM STATUS
6566          :*****:
6567
6568 027250 005037 002374      CLR    ERRWRD
6569 027254 104410              TRAP   C$ESCAPE
6570 027256 000176              .WORD L10105-.
6571          :*****:
6572          : JUMP TO END OF TEST IF ERROR
6573          :*****:
6574 027260 005037 002334      CLR    $GDDAT
6575 027264 005037 002336      CLR    $BDDAT
6576 027270 117737 153152 002336  MOVB  @BSEL4,$BDDAT
6577 027276 116237 027404 002334  MOVB  MODOUT(R2),$GDDAT
6578 027304 123737 002336 002334  CMPB  $BDDAT,$GDDAT
6579 027312 001411              BEQ   10$
6580 027314 012737 011264 002422  MOV    #M30F,CODEW
6581 027322 104455              TRAP   C$ERDF
6582 027324 000055              .WORD 45
6583 027326 010756              .WORD EROIC
6584 027330 007574              .WORD ERR27
6585 027332 104410              TRAP   C$ESCAPE
6586 027334 000120              .WORD L10105-.
6587 027336 062702 000002      10$:  ADD    #2,R2
6588
```


6639
6640
6641 027456
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652
6653
6654
6655 027456
6656
6657 027456 012737 004150 000004
6658 027464 005037 000006
6659 027470 012737 000002 002536
6660 027476 012746 000002
6661 027502 012705 027712
6662
6663 027506 005737 177572
6664
6665
6666 027512 062716 000002
6667 027516 012700 000000
6668 027522 104441
6669 027524 012701 172300
6670 027530 012702 000010
6671 027534
6672 027534 012721 077406
6673 027540 005302
6674 027542 001374
6675 027544 012701 172340
6676 027550 005011
6677 027552 012761 000200 000002
6678 027560 012761 000400 000004
6679 027566 012761 000600 000006
6680 027574 012761 001000 000010
6681 027602 012761 007600 000016
6682 027610 012761 002000 000012
6683 027616 005037 002424
6684 027622 012737 000033 002356
6685 027630 052737 040000 002424
6686 027636 012737 177776 002414
6687 027644 012737 001750 002416
6688 027652 012737 032542 002412
6689 027660 012737 001000 002420
6690 027666 004737 004450
6691 027672 004537 006154
6692 027676 005037 002374
6693 027702 104410
6694 027704 000012

```
.SBTTT ***** TEST 38 *****  
.SBTTT TEST OF MEM EXTENSION BIT 16, ADDRESS 200000  
ZZ  
:  
:  
* THIS TEST WE'LL TRY TRANSMITTING A MESSAGE  
* TO VIRTUAL ADDRESS 200000 THAT IS BIT 16 SET  
* IF MEMORY MANAGEMENT AND/OR SUFFICIENT MEMORY IS NOT  
* AVAILABLE FOR MESSAGE STORAGE, WE WILL EXPECT THE DMP-11  
* TO RETURN A NON-EXISTENT MEMORY ERROR FOR THE BUFFER.  
* IF ENOUGH MEMORY EXISTS, WE'LL MAKE SURE THE TRANSFER TAKES  
* PLACE PROPERLY.  
:  
:  
*  
*  
.SBTTT ***** TEST 38 *****  
T38: :  
MOV #META,4 ;SET UP TRAP FOR NO MEM  
CLR 6  
MOV #2,$TMP0  
MOV #2,-(SP) ;DUMMY MOVE TO STACK  
MOV #RET16,R5 ;SET UP R5 FOR RETURN IF TRAP  
; TAKES U TO SR.  
TST @#177572 ;DOES MEM MANAGEMENT EXIST  
; IF NOT TRAP TO META  
; ELSE CONTINUE  
; FIX THE STACK  
ADD #2,(SP)  
MOV #0,R0  
TRAP C$SPRI  
MOV #172300,R1 ;GET ADDRESS OF KERNEL PDR REQD/  
MOV #8.,R2 ; DO 8 TIMES  
10$: MOV #77406,(R1)+  
DEC R2  
BNE 10$  
MOV #172340,R1  
CLR (R1)  
MOV #200,2(R1)  
MOV #400,4(R1)  
MOV #600,6(R1)  
MOV #1000,10(R1)  
MOV #7600,16(R1)  
MOV #2000,12(R1)  
CLR GENWRD  
MOV #33,TRIBN ;MAP MEM CLEAR GEN AND SET UP TRIB  
BIS #BIT14,GENWRD ;SET MM BIT  
MOV #177776,RXADD ;SET UP RX ADD  
MOV #1000.,RXCC ;SET UP RX COUNT  
MOV #MR1,TXADD ;SET UP TRANSMIT COUNT  
MOV #512.,TXCC ;SET UP TRANSMIT COUNT  
JSR PC,MINITS ; MASTER CLEAR MODE DEF  
JSR R5,TXRX3 ;GO TRANSMIT AND RX  
CLR ERRWRD ; IF ERROR GO TO END  
TRAP C$ESCAPE  
.WORD L10106-
```

CZDMTAD DMP/V-11 PCTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 N 12
PAGE 157

;***** TEST 38 *****

SEQ 0156

6695 027706 004537 003672
6696 027712 005037 002374
6697 027716
6698 027716 104401

RET16: JSR R5, MEMEX ;CHECK MEM EXTENSION
CLR ERRWRD ;CLEAR ERROR WORD
L10106:
TRAP CSETST

```
6699 .SBTTL :***** TEST 39 *****
6700 .SBTTL TEST OF MEM EXTENSION BIT 17, ADDRESS 400000
6701 027720 ZZ
6702 :
6703 :
6704 : * THIS TEST WE'LL TRY TRANSMITTING A MESSAGE
6705 : * TO VIRTUAL ADDRESS 400000 THAT IS BIT 17 SET
6706 : * IF MEMORY MANAGEMENT AND/OR SUFFICIENT MEMORY IS NOT
6707 : * AVAILABLE FOR MESSAGE STORAGE, WE WILL EXPECT THE DMP-11
6708 : * TO RETURN A NON-EXISTENT MEMORY ERROR FOR THE BUFFER.
6709 : * IF ENOUGH MEMORY EXISTS,WE'LL MAKE SURE THE TRANSFER TAKES
6710 : * PLACE PROPERLY.
6711 :
6712 :
6713 :
6714 .SBTTL :***** TEST 39 *****
6715 027720 T39::
6716
6717 027720 012737 004150 000004 MOV #META,4 ;SET UP TRAP FOR NO MEM
6718 027726 005037 000006 CLR 6
6719 027732 012737 000004 002536 MOV #4,$TMP0
6720 027740 012746 000004 MOV #4,-(SP) ;DUMMY MOVE ON STACK IF TRAP
6721 027744 012705 030154 MOV #RET17,R5 ;SET UP R5 FOR RETURN IF TRAP IS
6722 : ; TO SR.EXMEM.
6723 027750 005737 177572 TST @#177572 ;DOES MEM MANAGEMENT EXIST
6724 : ;IF NOT TRAP TO META
6725 : ; ELSE CONTINUE
6726 : ;FIX THE STACK
6726 027754 062716 000002 ADD #2,(SP)
6727 027760 012700 000000 MOV #0,R0
6728 027764 104441 TRAP C$SPRI
6729 027766 012701 172300 MOV #172300,R1 ;GET ADDRESS OF KERNEL PDR REQD/
6730 027772 012702 000010 MOV #8.,R2 ; DO 8 TIMES
6731 027776
6732 027776 012721 077406 10$: MOV #77406,(R1)+
6733 030002 005302 DEC R2
6734 030004 001374 BNE 10$
6735 030006 012701 172340 MOV #172340,R1
6736 030012 005011 CLR (R1)
6737 030014 012761 000200 000002 MOV #200,2(R1)
6738 030022 012761 000400 000004 MOV #400,4(R1)
6739 030030 012761 000600 000006 MOV #600,6(R1)
6740 030036 012761 001000 000010 MOV #1000,10(R1)
6741 030044 012761 007600 000016 MOV #7600,16(R1)
6742 030052 012761 004000 000012 MOV #4000,12(R1)
6743 030060 005037 002424 CLR GENWRD
6744 030064 012737 000033 002356 MOV #33,TRIBN ;MAP MEM CLEAR GEN AND SET UP TRIB
6745 030072 052737 040000 002424 BIS #BIT14,GENWRD ;SET MM BIT
6746 030100 012737 177776 002414 MOV #177776,RXADD ;SET UP RX ADD
6747 030106 012737 041750 002416 MOV #41750,RXCC ;SET UP RX COUNT (1000 DECIMAL)
6748 030114 012737 032542 002412 MOV #MR1,TXADD ;SET UP TRANSMIT COUNT
6749 030122 012737 001000 002420 MOV #512.,TXCC ;SET UP TRANSMIT COUNT
6750 030130 004737 004450 JSR PC,MINITS ; MASTER CLEAR MODE DEF
6751 030134 004537 006154 JSR R5,TXRX3 ;GO TRANSMIT AND RX
6752 030140 005037 002374 CLR ERRWRD ; IF ERROR GO TO END
6753 030144 104410 TRAP C$ESCAPE
6754 030146 000012 .WORD L10107-
```


6755 030150 004537 003672
6756 030154 005037 002374
6757 030160
6758 030160 104401
6759

RET17: JSR R5, MEMEX ;CHECK MEM EXTENSION
CLR ERRWRD ;CLEAR ERROR WORD
L10107:
TRAP CSETST

6760
6761
6762 030162
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776 030162
6777
6778 030162 012737 004150 000004
6779 030170 005037 000006
6780 030174 012737 000006 002536
6781 030202 012746 000006
6782
6783 030206 012705 030416
6784
6785 030212 005737 177572
6786
6787
6788 030216 062716 000002
6789 030222 012700 000000
6790 030226 104441
6791 030230 012701 172300
6792 030234 012702 000010
6793 030240
6794 030240 012721 077406
6795 030244 005302
6796 030246 001374
6797 030250 012701 172340
6798 030254 005011
6799 030256 012761 000200 000002
6800 030264 012761 000400 000004
6801 030272 012761 000600 000006
6802 030300 012761 001000 000010
6803 030306 012761 007600 000016
6804 030314 012761 006000 000012
6805 030322 005037 002424
6806 030326 012737 000033 002356
6807 030334 052737 040000 002424
6808 030342 012737 177776 002414
6809 030350 012737 101750 002416
6810 030356 012737 032542 002412
6811 030364 012737 001000 002420
6812 030372 004737 004450
6813 030376 004537 006154
6814 030402 005037 002374
6815 030406 104410

```
.SBTTL ***** TEST 40 *****  
.SBTTL TEST OF MEM EXTENSION BIT 16 AND 17, ADDRESS 600000  
ZZ  
*  
*  
* THIS TEST WE'LL TRY TRANSMITTING A MESSAGE  
* TO VIRTUAL ADDRESS 600000 THAT IS BIT 16 AND 17 SET  
* IF MEMORY MANAGEMENT AND/OR SUFFICIENT MEMORY IS NOT  
* AVAILABLE FOR MESSAGE STORAGE, WE WILL EXPECT THE DMP-11  
* TO RETURN A NON-EXISTENT MEMORY ERROR FOR THE BUFFER.  
* IF ENOUGH MEMORY EXISTS, WE'LL MAKE SURE THE TRANSFER TAKES  
* PLACE PROPERLY.  
*  
*  
.SBTTL ***** TEST 40 *****  
T40::  
MOV #META,4 ;SET UP TRAP FOR NO MEM  
CLR 6  
MOV #6,$TMP0  
MOV #6,-(SP) ;DUMMY MOVE TO STACK IF  
;TRAP TAKES U TO SR.  
MOV #RET18,R5 ;SET UP R5 FOR RETURN  
TST @#177572 ;DOES MEM MANAGEMENT EXIST  
;IF NOT TRAP TO META  
;ELSE CONTINUE  
ADD #2,(SP) ;FIX STACK  
MOV #0,R0  
TRAP C$SPRI  
MOV #172300,R1 ;GET ADDRESS OF KERNEL PDR REQD/  
MOV #8.,R2 ;DO 8 TIMES  
10$:  
MOV #77406,(R1)+  
DEC R2  
BNE 10$  
MOV #172340,R1  
CLR (R1)  
MOV #200,2(R1)  
MOV #400,4(R1)  
MOV #600,6(R1)  
MOV #1000,10(R1)  
MOV #7600,16(R1)  
MOV #6000,12(R1)  
CLR GENWRD  
MOV #33,TRIBN ;MAP MEM CLEAR GEN AND SET UP TRIB  
BIS #BIT14,GENWRD ;SET MM BIT  
MOV #177776,RXADD ;SET UP RX ADD  
MOV #101750,RXCC ;SET UP RX COUNT (1000 DECIMAL)  
MOV #MR1,IXADD ;SET UP TRANSMIT COUNT  
MOV #512.,IXCC ;SET UP TRANSMIT COUNT  
JSR PC,MINITS ;MASTER CLEAR MODE DEF  
JSR R5,IXRX3 ;GO TRANSMIT AND RX  
CLR ERRWRD ;IF ERROR GO TO END  
TRAP C$ESCAPE
```

6816 030410 000012
6817 030412 004537 003672
6818 030416 005037 002374
6819 030422
6820 030422 104401
6821
6822
6823
6824
6825
6826 030424
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836 030424
6837
6838 030424 012737 000041 002366
6839
6840 030432 005037 002424
6841 030436 012737 000401 002420
6842 030444 012737 032543 002412
6843 030452 012737 033023 002414
6844 030460 012737 001750 002416
6845 030466 012737 000001 002356
6846 030474 004737 004440
6847 030500 005037 002374
6848 030504 104410
6849 030506 000012
6850
6851
6852
6853 030510 004537 005542
6854 030514 005037 002374
6855 030520
6856 030520 104401
6857

RET18: .WORD L10110-
L10110: JSR R5, MEMEX ;CHECK MEM EXTENSION
CLR ERRWRD ;CLEAR ERROR WORD
TRAP CSETST

.SBTTL ;***** TEST 41 *****
.SBTTL *TX AND RX 257 BYTES, ODD TX, ODD RX, DDCMP, POINT TO POINT
ZZ
*
* THIS TEST WILL TRANSMIT A MESSAGE OF 257 BYTES
* FROM A TX BUFFER STARTING WITH AN ODD BYTE TO
* A RECEIVE BUFFER STARTING WITH AN ODD BYTE IN
* DDCMP MODE, POINT TO POINT
*-

.SBTTL ;***** TEST 41 *****
I41: :

MOV #41, ROMN1 ;SET UP TEST NUMBER
CLR GENWRD ;CLEAR THE GEN WORD
MOV #257., TXCC ;SET UP TRANSMIT CHAR COUNT
MOV #MR1+1, TXADD ;SET UP ADD FOR TX
MOV #RECBU1+1, RXADD ;SET UP RX
MOV #1000., RXCC ;SET UP RX COUNT 1000 DECIMAL
MOV #01, TRIBN
JSR PC, MINIT1 ;MASTER CLEAR
CLR ERRWRD
TRAP C\$ESCAPE
.WORD L10111-
: : : : :
:ESCAPE TEST IF ERROR
: : : : :
20\$: JSR R5, TXRXSR ;GO DO IT
CLR ERRWRD
L10111: TRAP CSETST

```
6858 .SBTTL :***** TEST 42 *****
6859 .SBTTL *TX AND RX 1 BYTE,ODD TX,EVEN RX,MAINT,MULTIPOINT
6860 030522 ZZ
6861 :*
6862 :*
6863 :* THIS TEST TRANSMITS AND RECEIVES 1 BYTE MESSAGE
6864 :* FROM AND ODD TRANSMIT BUFFER TO AN EVEN RX BUFFER
6865 :* IN MAINTAINCE MODE,MULTIPOINT
6866 :*
6867 :*-
6868 .SBTTL :***** TEST 42 *****
6869 030522 T42::
6870
6871 030522 012737 000001 002356 MOV #01,TRIBN
6872 030530 012737 000001 002420 MOV #01,TXCC
6873 030536 012737 032543 002412 MOV #MR1+1,TXADD
6874 030544 012737 000764 002416 MOV #500.,RXCC
6875 030552 012737 033022 002414 MOV #RECBUI,RXADD
6876 030560 005037 002424 CLR GENWRD
6877 030564 052737 100000 002424 BIS #BIT15,GENWRD ;SET UP TX RX AND MAINT STATE
6878 030572 004737 004450 JSR PC,MINITS ;MASTER CLEAR MULTIPOINT
6879 030576 005037 002374 CLR ERRWRD
6880 030602 104410 TRAP C$ESCAPE
6881 030604 000C12 .WORD L10112-.
6882
6883 :*****
6884 : ESCAPE TEST IF ERROR
6885 030606 004537 006154 20$: JSR R5,TXRX3 ;GO DO IT
6886 030612 005037 002374 CLR ERRWRD ;CLEAR ERROR WORD
6887 030616
6888 030616 104401 L10112: TRAP C$ETST
```

6889
 6890
 6891 030620
 6892
 6893
 6894
 6895
 6896
 6897
 6898
 6899 030620
 6900
 6901 030620 005037 002424
 6902 030624 004737 004450
 6903 030630 005037 002374
 6904 030634 104410
 6905 030636 000760
 6906
 6907
 6908
 6909 030640 005037 002356
 6910 030644 012703 000237
 6911 030650 012704 000100
 6912 030654 004537 003156
 6913 030660 005037 002374
 6914 030664 104410
 6915 030666 000730
 6916
 6917
 6918
 6919
 6920 030670 012737 000034 002356
 6921 030676 012703 000001
 6922 030702 004537 003156
 6923 030706 005037 002374
 6924 030712 104410
 6925 030714 000702
 6926
 6927
 6928
 6929 030716 012703 000236
 6930 030722 012704 002000
 6931 030726 004537 003156
 6932 030732 005037 002374
 6933 030736 104410
 6934 030740 000656
 6935
 6936
 6937
 6938 030742 012704 002010
 6939 030746 012703 000234
 6940 030752 004537 003156
 6941 030756 005037 002374
 6942 030762 104410
 6943 030764 000632
 6944

```

.SBTTL ;***** TEST 43 *****
.SBTTL POLLING STATE TESTS
ZZ
;*
;* THIS TEST CHECKS THE DEGRADING OF THE POLLING
;* STATES FROM ACTIVE TO INACTIVE TO POTEN. DEAD
;* TO DEAD.
.SBTTL ;***** TEST 43 *****
T43::
BEGPOL: CLR GENWRD ;CLEAR FLAG WORD
        JSR PC,MINITS ;MASTER CLEAR MODE DEF(FD/CS/MP)
        CLR ERRWRD
        TRAP C$ESCAPE
        .WORD L10113-.
        :::::::::::
        ;SET POLL DELAY
        :::::::::::
        CLR TRIBN
        MOV #237,R3
        MOV #100,R4
        JSR R5,CONTIN
        CLR ERRWRD
        TRAP C$ESCAPE
        .WORD L10113-.
        :::::::::::
        ; ESTABLISH TRIB
        :::::::::::
        MOV #34,TRIBN ;SET TRIB NO.
        MOV #01,R3
        JSR R5,CONTIN
        CLR ERRWRD
        TRAP C$ESCAPE
        .WORD L10113-.
        :::::::::::
        ; SET SELECTION TIMER TO 1 SEC
        :::::::::::
        MOV #236,R3
        MOV #2000,R4
        JSR R5,CONTIN
        CLR ERRWRD
        TRAP C$ESCAPE
        .WORD L10113-.
        :::::::::::
        ; WRITE NDM > INACTIVE AND #TO > PDEAD
        :::::::::::
10$: MOV #2010,R4
        MOV #234,R3
        JSR R5,CONTIN ;WRITE TSS SLOT
        CLR ERRWRD
        TRAP C$ESCAPE
        .WORD L10113-.
        :::::::::::
    
```

6945							: ISTRT TRIB
6946						
6947	030766	012703	000003				MOV #03,R3
6948	030772	004537	003156				JSR R5,CONTIN
6949	030776	005037	002374				CLR ERRWRD
6950	031002	104410					TRAP C\$ESCAPE
6951	031004	000612					.WORD L10113-
6952						
6953							: WAIT FOR RUN STATE
6954						
6955	031006	012737	000001	002334			MOV #01,\$GDDAT
6956	031014	004537	003236				JSR R5,GETOUT
6957	031020	005037	002374				CLR ERRWRD
6958	031024	104410					TRAP C\$ESCAPE
6959	031026	000570					.WORD L10113-
6960	031030	012737	000024	002334			MOV #24,\$GDDAT
6961	031036	004537	003616				JSR R5,GETOC
6962	031042	005037	002374				CLR ERRWRD
6963	031046	104410					TRAP C\$ESCAPE
6964	031050	000546					.WORD L10113-
6965						
6966							: READ TSS SLOT WITH POLL STATUS
6967						
6968	031052	042777	000200	151362	20\$:		BIC #RDO,@BSEL2 ;CLEAR RDO
6969	031060	012703	000042				MOV #42,R3
6970	031064	004537	003156				JSR R5,CONTIN ;READ TSS SLOT 2
6971	031070	005037	002374				CLR ERRWRD
6972	031074	104410					TRAP C\$ESCAPE
6973	031076	000520					.WORD L10113-
6974	031100	012737	000002	002334			MOV #02,\$GDDAT
6975	031106	004537	003236				JSR R5,GETOUT ;GET INFO OUT
6976	031112	005037	002374				CLR ERRWRD
6977	031116	104410					TRAP C\$ESCAPE
6978	031120	000476					.WORD L10113-
6979	031122	032777	001000	151316			BIT #BIT9,@BSEL4 ;IS IT INACTIVE
6980	031130	001750					BEQ 20\$;IF NOT GO BACK
6981						
6982							: GET HERE WHEN STATE GOES TO INACTIVE
6983						
6984	031132	042777	000200	151302			BIC #RDO,@BSEL2 ;CLEAR OUTPUT
6985						
6986							: READ # OF SELECTION INTERVALS
6987						
6988	031140	012703	000051				MOV #51,R3
6989	031144	004537	003156				JSR R5,CONTIN
6990	031150	005037	002374				CLR ERRWRD
6991	031154	104410					TRAP C\$ESCAPE
6992	031156	000440					.WORD L10113-
6993	031160	012737	000002	002334			MOV #02,\$GDDAT
6994	031166	004537	003236				JSR R5,GETOUT
6995						
6996							: MAKE SURE #OF SELC. INTV IS CORRECT
6997						
6998	031172	012737	000010	002334			MOV #10,\$GDDAT
6999	031200	004537	003442				JSR R5,GETDAT
7000						

```

7001          : GET HERE IN INACTIVE STATE
7002          : .....:
7003 031204   40$:
7004 031204 042777 000200 151230 BIC   #RDO,@BSEL2   ;CLEAR OUTPUT
7005          : .....:
7006          : CHANGE MODE TO HALF DUPLEX
7007          : .....:
7008 031212 052777 000200 151216 BIS   #RQ1,@BSELO
7009 031220 004537 002726 JSR   R5,WRDI       ;WAIT FOR RDI TO SET
7010          : .....:
7011 031224 005037 002374 CLR   ERRWRD       ;CLEAR ERROR
7012 031230 104410 TRAP  C$ESCAPE
7013 031232 000364 .WORD L10113-.
7014 031234 042777 000200 151174 BIC   #RQ1,@BSELO
7015 031242 012777 000004 151202 MOV   #04,@BSEL6
7016 031250 112777 000002 151164 MOVB  #02,@BSEL2
7017 031256 042777 000200 151156 50$: BIC   #RDO,@BSEL2
7018          : .....:
7019          : .....:
7020          : READ POLL STATUS SLOT
7021          : .....:
7022          : .....:
7023 031264 012703 000042 MOV   #42,R3
7024 031270 004537 003156 JSR   R5,CONTIN
7025 031274 005037 002374 CLR   ERRWRD
7026 031300 104410 TRAP  C$ESCAPE
7027 031302 000314 .WORD L10113-.
7028 031304 012737 000002 002334 MOV   #02,$GDDAT
7029          : .....:
7030 031312 004537 003236 JSR   R5,GETOUT
7031          : .....:
7032          : IS THE STATE POTN. DEAD??
7033          : .....:
7034 031316 032777 010000 151122 BIT   #BIT12,@BSEL4
7035 031324 001754 BEQ   50$
7036          : .....:
7037          : IF NOT GO BACK TO 50
7038          : .....:
7039          : .....:
7040          : .....:
7041          : IF SO READ THE SELECTION TIMER
7042          : .....:
7043          : .....:
7044 031326 012703 000056 MOV   #56,R3
7045 031332 042777 000200 151102 BIC   #RDO,@BSEL2
7046 031340 004537 003156 JSR   R5,CONTIN
7047 031344 005037 002374 CLR   ERRWRD
7048 031350 104410 TRAP  C$ESCAPE
7049 031352 000244 .WORD L10113-.
7050 031354 012737 000002 002334 MOV   #02,$GDDAT
7051 031362 004537 003236 JSR   R5,GETOUT
7052 031366 005037 002374 CLR   ERRWRD
7053 031372 104410 TRAP  C$ESCAPE
7054 031374 000222 .WORD L10113-.
7055          : .....:
7056          : COMPARE SELECTION TIME OUTS WITH # WRITTEN

```

```

7057
7058
7059 031376 012737 001004 002334      MOV    #1004,$GDDAT
7060 031404 004537 003442              JSR    R5,GETDAT
7061
7062      ; WAIT FOR TRIB TO POST SELECT. THRESH. ERROR
7063
7064 031410 042777 000200 151024      BIC    #RDO,@BSEL2      ;CLEAR OUTPUT CODE
7065
7066 031416 012737 000001 002334      MOV    #01,$GDDAT
7067 031424 004537 003236              JSR    R5,GETOUT
7068 031430 005037 002374              CLR    ERRWRD
7069 031434 104410                      TRAP   C$ESCAPE
7070 031436 000160                      .WORD L10113-
7071 031440 012737 000006 002334      MOV    #6,$GDDAT
7072 031446 004537 003616              JSR    R5,GETOC
7073 031452 005037 002374              CLR    ERRWRD
7074 031456 104410                      TRAP   C$ESCAPE
7075 031460 000136                      .WORD L10113-
7076 031462 042777 000200 150752      BIC    #RDO,@BSEL2
7077
7078      ; NOW WAIT FOR TRIB TO POST DEAD STATUS
7079
7080 031470 012737 000001 002334      MOV    #01,$GDDAT
7081 031476 004537 003236              JSR    R5,GETOUT
7082 031502 005037 002374              CLR    ERRWRD
7083 031506 104410                      TRAP   C$ESCAPE
7084 031510 000106                      .WORD L10113-
7085 031512 012737 000022 002334      MOV    #22,$GDDAT
7086 031520 004537 003616              JSR    R5,GETOC
7087
7088 031524 005037 002374              CLR    ERRWRD
7089 031530 104410                      TRAP   C$ESCAPE
7090 031532 000064                      .WORD L10113-
7091
7092
7093      ; NOW READ SELECTION TIMER AND
7094      ; SEE IF IT IS EQUAL TO 10
7095
7096
7097 031534 042777 000200 150700      BIC    #RDO,@BSEL2
7098 031542 012703 000056              MOV    #56,R3
7099 031546 004537 003156              JSR    R5,CONTIN
7100 031552 005037 002374              CLR    ERRWRD
7101 031556 104410                      TRAP   C$ESCAPE
7102 031560 000036                      .WORD L10113-
7103 031562 012737 000002 002334      MOV    #02,$GDDAT
7104 031570 004537 003236              JSR    R5,GETOUT
7105 031574 005037 002374              CLR    ERRWRD
7106 031600 104410                      TRAP   C$ESCAPE
7107 031602 000014                      .WORD L10113-
7108 031604 012737 001010 002334      MOV    #1010,$GDDAT
7109 031612 004537 003442              JSR    R5,GETDAT
7110 031616
7111 031616 104401      L10113: TRAP   C$ETST
7112

```


CZDMTAD DMP/V-11 PCTNL TST #1 MACY11 30A(1052) 12-AUG-80 15:53 PAGE 167
CZDMTA.REL 12-AUG-80 15:52 ----- END OF HARDWARE TESTS -----

K 13

SEQ 0166

7113
7114
7115
7116
7117

.SBTTL ----- END OF HARDWARE TESTS -----

7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128
7129
7130
7131
7132 031620 000041
7133 031622
7134 031622 000032
7135 031624 031724
7136 031626 000007
7137 031630 000000
7138 031632 000007
7139 031634 001031
7140 031636 032016
7141 031640 160000
7142 031642 177776
7143 031644 002031
7144 031646 032044
7145 031650 000000
7146 031652 000674
7147 031654 003032
7148 031656 032075
7149 031660 007000
7150 031662 000004
7151 031664 000007
7152 031666 010032
7153 031670 032126
7154 031672 000007
7155 031674 000000
7156 031676 000004
7157 031700 011032
7158 031702 032232
7159 031704 000007
7160 031706 000000
7161 031710 000007
7162 031712 013032
7163 031714 032443
7164 031716 000007
7165 031720 000000
7166 031722 000007
7167
7168
7169
7170 031724
7171
7172 031724 042523 042514 052103
7173 031732 047440 052120 047511

.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.
://////

```

```

        .WORD L10114-L$HARD/2
L$HARD::

```

```

        .WORD T$CODE
        .WORD OPTYPM
        .WORD 7
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD ADDRES
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD VECTOR
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD PRIRTY
        .WORD 7000
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD LOOPBK
        .WORD 7
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD SPEDM
        .WORD 7
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD IFTYPM
        .WORD 7
        .WORD T$LLOLIM
        .WORD T$HILIM

```

.EVEN

L10114:

OPTYPM: .ASCIZ /SELECT OPTION TYPE (0=8207'DMP',1=8053'DMV',2=8064'DMV):/

7174	031740	020116	054524	042520	
7175	031746	024040	030040	034075	
7176	031754	030062	023467	046504	
7177	031762	023520	030454	034075	
7178	031770	032460	023463	046504	
7179	031776	023526	031054	034075	
7180	032004	033060	023464	046504	
7181	032012	024526	000072		
7182	032016	042504	044526	042503	ADDRESS: .ASCIZ /DEVICE CSR ADDRESS : /
7183	032024	041440	051123	040440	
7184	032032	042104	042522	051523	
7185	032040	035040	000040		
7186	032044	042504	044526	042503	VECTOR: .ASCIZ /DEVICE VECTOR ADDRESS : /
7187	032052	053040	041505	047524	
7188	032060	020122	042101	051104	
7189	032066	051505	020123	020072	
7190	032074	000			
7191	032075	104	053105	041511	PRIORTY: .ASCIZ /DEVICE PRIORITY LEVEL : /
7192	032102	020105	051120	047511	
7193	032110	044522	054524	046040	
7194	032116	053105	046105	035040	
7195	032124	000040			
7196	032126	052524	047122	051101	LOOPBK: .ASCII /TURNAROUND TYPE -/
7197	032134	052517	042116	052040	
7198	032142	050131	020105	055	
7199	032147	050	036460	031510	.ASCIZ /(0=H3254&H3255,1=CABLE,2=MOD LOC,3=MOD REM,4-NONE)/
7200	032154	032462	023064	031510	
7201	032162	032462	026065	036461	
7202	032170	040503	046102	026105	
7203	032176	036462	047515	020104	
7204	032204	047514	026103	036463	
7205	032212	047515	020104	042522	
7206	032220	026115	036464	047516	
7207	032226	042516	000051		
7208	032232	046120	040505	042523	SPEDEM: .ASCII 'PLEASE SELECT BAUD RATE;TYPE '0' FOR 2.4K; '1' FOR 4.8K;''
7209	032240	051440	046105	041505	
7210	032246	020124	040502	042125	
7211	032254	051040	052101	035505	
7212	032262	054524	042520	023440	
7213	032270	023460	043040	051117	
7214	032276	031040	032056	035513	
7215	032304	023440	023461	043040	
7216	032312	051117	032040	034056	
7217	032320	035513			
7218	032322	005015	031047	020047	.ASCII<15><12>'2' FOR 9.6K; '3' FOR 19.2K; '' FOR 56K; '5' FOR 250K;''
7219	032330	047506	020122	027071	
7220	032336	045466	020073	031447	
7221	032344	020047	047506	020122	
7222	032352	034461	031056	035513	
7223	032360	023440	023464	043040	
7224	032366	051117	032440	045466	
7225	032374	020073	032447	020047	
7226	032402	047506	020122	032462	
7227	032410	045460	073		
7228	032413	015	047412	020122	.ASCIZ<15><12>'OR '6' FOR 500K BAUDS''
7229	032420	033047	020047	047506	

CZDMTAD DMP/V-11 FC1NL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 170
HARDWARE PARAMETER CODING SECTION

SEQ 0169

7230	032426	020122	030065	045460
7231	032434	041040	052501	051504
7232	032442	000		
7233	032443	123	046105	041505
7234	032450	020124	047111	042524
7235	032456	043122	041501	020105
7236	032464	054524	042520	024040
7237	032472	036461	047111	042524
7238	032500	043522	046101	031054
7239	032506	042475	040511	031454
7240	032514	053075	031456	026065
7241	032522	036464	031064	024462
7242	032530	000072		
7243				
7244				
7245				
7246				
7247				

IFTYPM: .ASCIZ /SELECT INTERFACE TYPE (1=INTERGAL,2=EIA,3=V.35,4=422):/

.EVEN

.SBTTL SOFTWARE PARAMETER CODING SECTION

```
7248
7249
7250
7251 :////////////////////
7252 :// THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
7253 :// THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
7254 :// MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7255 :// INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
7256 :// MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7257 :// WITH THE OPERATOR.
7258 :////////////////////
7259
7260 032532 000000 .WORD L10115-L$SOFT/2
7261 032534 L$SOFT::
7262
7263 .EVEN
7264 032534 L10115:
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274 :***** PATCH AREA FOR DEBUG *****
7275 032534 PATCH:
7276 032534 000240 NOP
7277 032536 000240 NOP
7278 032540 000240 NOP
7279 :*****
7280
7281
7282 032542 041101 000103 MR1: .ASCIZ "ABC"
7283 032546 MR1E:
7284
7285 032546 047516 020127 051511 MR12: .ASCII "NOW IS THE TIME FOR ALL GOOD PEOPLE TO COME TO THE"
7286 032554 052040 042510 052040
7287 032562 046511 020105 047506
7288 032570 020122 046101 020114
7289 032576 047507 042117 050040
7290 032604 047505 046120 020105
7291 032612 047524 041440 046517
7292 032620 020105 047524 052040
7293 032626 042510
7294 032630 044124 020105 052521 .ASCII "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG'S"
7295 032636 041511 020113 051102
7296 032644 053517 020116 047506
7297 032652 020130 052512 050115
7298 032660 042105 047440 042526
7299 032666 020122 044124 020105
7300 032674 040514 054532 042040
7301 032702 043517 051447
7302 032706 040502 045503 020056 .ASCII "BACK. LOPS ARE TOPS!"
7303 032714 047514 051520 040440
```

7304	032722	042522	052040	050117	
7305	032730	020523			
7306	032732	041101	042103	043105	.ASCII 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
7307	032740	044107	045111	046113	
7308	032746	047115	050117	051121	
7309	032754	052123	053125	054127	
7310	032762	055131			
7311	032764	041501	043505	045511	.ASCII 'ACEGIKMOQSUWY'
7312	032772	047515	051521	053525	
7313	033000	131			
7314					
7315		033002			.EVEN
7316	033002				MR12E:
7317					
7318	033002	000000			DATLST: 0
7319	033004	125252			125252
7320	033006	052525			052525
7321	033010	000000			0
7322	033012	177777			-1
7323	033014	000377			377
7324	033016	177400			177400
7325	033020	000562			562
7326					
7327	033022	001750			RECBU1: .BLKW 1000.
7328					
7329					.EVEN
7330	036742	000000			.WORD 0
7331	036744	000000			.WORD 0
7332	036746				L\$LAST::
7333					
7334		000001			.END

ADDRES	032016	7140	7182#													
ADR =	000020 G	1385#														
ASSEMB=	000010	1084														
AXNUM	002502	1515#	2200*	2202*	2259	2439*	2544	6493*								
BADBR	006002	2451	2467	2510#												
BADBRM	010014	2512	2909#													
BADIF	005750	2449	2465	2480	2490	2499#										
BADIFM	007736	2501	2901#													
BASER	007720	2889	2898#													
BEGPOL	030624	6902#														
BIT0 =	000001 G	1358#	1418	2575	3284	4735										
BIT00 =	000001 G	1347#	1358													
BIT01 =	000002 G	1346#	1357													
BIT02 =	000004 G	1345#	1356													
BIT03 =	000010 G	1344#	1355													
BIT04 =	000020 G	1343#	1354													
BIT05 =	000040 G	1342#	1353													
BIT06 =	000100 G	1341#	1352													
BIT07 =	000200 G	1340#	1351													
BIT08 =	000400 G	1339#	1350													
BIT09 =	001000 G	1338#	1349													
BIT1 =	000002 G	1357#	1417	2532												
BIT10 =	002000 G	1337#														
BIT11 =	004000 G	1336#														
BIT12 =	010000 G	1335#	7034													
BIT13 =	020000 G	1334#														
BIT14 =	040000 G	1333#	2609	6685	6745	6807										
BIT15 =	100000 G	1332#	2554	6877												
BIT2 =	000004 G	1356#	1416													
BIT3 =	000010 G	1355#	1415	2250	2438	4362	4379	4672	4744	6506						
BIT4 =	000020 G	1354#	1414													
BIT5 =	000040 G	1353#	4937	4961	5210	5233	5305									
BIT6 =	000100 G	1352#	1413													
BIT7 =	000200 G	1351#	1412	4923	5268											
BIT8 =	000400 G	1350#														
BIT9 =	001000 G	1349#	6979													
BOE =	000400 G	1389#														
BSELO	002436	1488#	1770*	1775*	2215	2252*	2260*	2588*	2607*	4125*	4177*	4188*	4209	4227*		
		4247	4269*	4272*	4367*	4390	4406*	4474*	4532*	4542*	4603	4605*	4611*	4689*		
		4695*	4762*	4771*	5437*	5444*	5701*	5707*	6192*	6198*	6249*	6255*	6317*	6323*		
		6363*	6370*	7008*	7014*											
BSEL1	002440	1491#	2120*	2122*	2123*	2170*	2173*	2203*	2206*	2241*	2250*	2359*	2438*	3302*		
		3303*	4362*	4379	4593*	4596*	4672*	4744*	6506*							
BSEL2	002442	1493#	1627	1630	1640*	1670	1673	1683*	1715	1778*	1809	2005	2244*	2257		
		2261*	2365*	2575*	2596*	2608*	2620*	2654*	4136	4191*	4490*	4498	4544*	4549		
		4560	4614*	4619	4675	4688*	4697*	4750	4761*	4773*	4775	4860*	5169	5302*		
		5436*	5448*	5473*	5485*	5564*	5600*	5636*	5708*	5718*	5828*	5848*	5860	5925*		
		5946*	5957	6072*	6202*	6259*	6327*	6374*	6968*	6984*	7004*	7016*	7017*	7045*		
		7064*	7076*	7097*												
BSEL3	002444	1494#	1774*	1823	2595*	2606*	3308*	3309*	4295*	4333*	4612*	5447*	6199*	6258*		
		6326*	6371*													
BSEL4	002446	1495#	1776*	1879	1939	2231	2243*	2363*	2593*	2604*	4365*	5445*	5552*	5553		
		6201*	6257*	6325*	6373*	6576	6919	7034								
BSEL5	002450	1497#	3310*	3311*												
BSEL6	002452	1498#	1731	1777*	1850	1908	1966	2012	2121*	2222	2224	2259*	2364*	2594*		
		2605*	2680	4102	4104	4189*	4366*	4487*	4543*	4613*	4632	4678	4696*	4772*		

BSEL7	002454	5446*	6200*	6256*	6324*	6372*	7015*							
		1500#	3312*	3313*	3560	3563	3583	3596	3607	3710	3713	3733	3746	3757
CADDR	002376	3860	3863	3883	3896	3907	4176*	4190*	4203	4243				
		1469#	2096	2100	3480*	3486*	3489*	3490	3505*	3519*	3530*	3555*	3561*	3564*
		3565	3580*	3594*	3605*	3630*	3636*	3639*	3640	3655*	3669*	3680*	3705*	3711*
		3714*	3715	3730*	3744*	3755*	3780*	3786*	3789*	3790	3805*	3819*	3830*	3855*
		3861*	3864*	3865	3880*	3894*	3905*							
CFM1	005160	2309	2324#											
CFM2	005245	2318	2333#											
CODEW	002422	1480#	1721*	1735*	1814*	1855*	1885*	1971*	2015*	2876	4565*	4625*	4635*	5558*
		6580*												
CONTIN	003156	1769#	2529	2539	2549	2560	2581	4739	4819	4836	4863	4909	4924	4938
		5018	5031	5080	5093	5142	5155	5211	5270	5284	5306	5357	5404	5413
		5450	5489	5516	5529	5566	5579	5602	5638	5651	5750	5747	5810	5830
		5850	5894	5907	5927	5947	5993	6007	6020	6074	6087	6237	6298	6308
		6511	6524	6537	6912	6922	6931	6940	6948	6970	6989	7024	7046	7099
COUNT	002340	1453#	3333*	3338*	4085*	4091*								
COUNTT	005062	2275*	2282*	2288#										
CRCCAL	002426	1482#	2377*	2386										
CRCR	004324	2141#	2371	2380	3493	3568	3643	3718	3793	3868				
CWORD	002372	1467#	2145*	2149	2150*	2151*	2358*	2385*	2386	2822	3482*	3496*	3497	3557*
		3571*	3572	3632*	3646*	3647	3707*	3721*	3722	3782*	3796*	3797	3857*	3871*
		3872												
CSAU =	U00052	1084#	3392											
CSAUTO=	000061	1084#	3346											
CSBRK =	000022	1084#	2285											
CSBSEG=	000004	1084#												
CSBSUB=	000002	1084#	5192	5251	5341									
CSCEFG=	000045	1084#												
CSCLCK=	000062	1084#												
CSCLEA=	000012	1084#	3359											
CSCLOS=	000035	1084#												
CSCLP1=	000006	1084#												
CSCVEC=	000036	1084#	3440	4353	4355									
CSDCLM=	000044	1084#	3437											
CSDODU=	000051	1084#	3341											
CSDRPT=	000024	1084#												
CSDU =	000053	1084#	3376											
CSEDIT=	000003	1084#	1161											
CSERDF=	000055	1084#	1635	1678	1725	1739	1815	1827	1856	1886	1915	1943	1972	2016
		2029	2045	2052	2277	2388	2642	3450	3501	3511	3535	3576	3586	3610
		3651	3661	3685	3726	3736	3760	3801	3811	3835	3876	3886	3910	3952
		4009	4096	4107	4117	4217	4255	4381	4398	4420	4430	4501	4552	4566
		4626	4636	4681	4777	5172	5559	5863	5960	6581				
CSERHR=	000056	1084#												
CSERRO=	000060	1084#												
CSERSF=	000054	1084#												
CSERSO=	000057	1084#	2233											
CSESCA=	000010	1084#	3935	3944	3957	3992	4001	4013	4100	4113	4121	4130	4139	4174
		4182	4197	4221	4237	4259	4303	4339	4373	4386	4402	4466	4480	4526
		4536	4557	4571	4601	4609	4617	4630	4664	4693	4703	4729	4756	4766
		4812	4825	4842	4854	4869	4901	4915	4930	4944	4954	4965	4975	5010
		5024	5037	5047	5072	5086	5099	5110	5134	5148	5161	5203	5214	5225
		5237	5261	5273	5287	5296	5309	5320	5329	5349	5360	5370	5400	5410
		5418	5426	5431	5442	5455	5461	5466	5471	5477	5483	5508	5522	5535
		5546	5572	5585	5595	5608	5619	5630	5644	5657	5667	5695	5705	5713

ERR9 = 007042 G	2716#	4504																		
EVL = 000004 G	1383#																			
EXMDT 020454	4166	4444#																		
EXMEM 003672	2002#																			
EXMEMA 004050	2008	2038#																		
EXMEMB 003764	2010	2015#																		
EXMEMC 004010	2014	2022#																		
EXMEMX 004154	2004	2021	2034	2044	2050	2058#														
ESEND = 002100	1084#																			
ESLOAD= 000035	1084#	1184																		
FMT0 010244	2936#	3457																		
FRSPAS 002352	1458#	3282*																		
FRSTIM 002326	1448#	3251	3258*																	
FSAU = 000015	1084#	3390	3391																	
FSAUTO= 000020	1084#	3331	3345																	
F\$BGN = 000040	1084#	1092	1202	2666	2679	2690	2702	2716	2728	2742	2758	2771	2782							
		2800	2821	2834	2849	2865	2888	3225	3230	3246	3331	3355	3372	3390						
		3419	3442	3447	3476	3540	3551	3615	3626	3690	3701	3765	3776	3840						
		3851	3915	3927	3935	3944	3957	3972	3983	3992	4001	4013	4027	4083						
		4100	4113	4121	4130	4139	4141	4162	4174	4182	4197	4221	4237	4259						
		4303	4339	4373	4386	4402	4407	4410	4417	4427	4437	4466	4461	4466						
		4480	4509	4521	4526	4536	4557	4571	4579	4591	4601	4609	4617	4630						
		4641	4658	4664	4685	4693	4703	4710	4723	4729	4756	4766	4782	4798						
		4812	4825	4842	4854	4869	4881	4894	4901	4915	4930	4944	4954	4965						
		4975	4991	5005	5010	5024	5037	5047	5053	5067	5072	5086	5099	5110						
		5116	5129	5134	5148	5161	5178	5189	5191	5203	5214	5225	5237	5248						
		5250	5261	5273	5287	5296	5309	5320	5329	5338	5340	5349	5360	5370						
		5380	5382	5396	5400	5410	5418	5426	5431	5442	5455	5461	5466	5471						
		5477	5483	5493	5503	5508	5522	5535	5546	5572	5585	5595	5608	5619						
		5630	5644	5657	5667	5673	5689	5695	5705	5713	5723	5737	5742	5753						
		5763	5769	5784	5789	5803	5813	5823	5833	5843	5853	5868	5881	5886						
		5900	5910	5920	5930	5940	5950	5965	5980	5985	5999	6013	6026	6040						
		6053	6066	6080	6093	6106	6119	6131	6148	6159	6167	6169	6181	6186						
		6196	6207	6212	6224	6229	6243	6253	6264	6269	6285	6290	6301	6311						
		6321	6332	6338	6352	6357	6368	6380	6385	6405	6418	6427	6433	6447						
		6458	6468	6474	6488	6498	6517	6530	6543	6556	6569	6585	6592	6632						
		6655	6693	6697	6715	6753	6757	6776	6815	6819	6836	6848	6855	6869						
		6880	6887	6899	6904	6914	6924	6933	6942	6950	6958	6963	6972	6977						
		6991	7012	7026	7048	7053	7069	7074	7083	7089	7101	7106	7110	7132						
		7260																		
F\$CLEA= 000007	1084#	3355	3358																	
F\$DU = 000016	1084#	3372	3375																	
F\$END = 000041	1084#	1092	2675	2689	2701	2713	2725	2737	2755	2770	2781	2799	2820							
		2833	2846	2860	2885	2897	3228	3234	3329	3347	3360	3377	3393	3419						
		3442	3444	3464	3476	3540	3542	3551	3615	3617	3626	3690	3692	3701						
		3765	3767	3776	3840	3842	3851	3915	3917	3927	3935	3944	3957	3972						
		3974	3983	3992	4001	4013	4027	4029	4083	4100	4113	4121	4130	4139						
		4141	4143	4162	4174	4182	4197	4221	4237	4259	4303	4339	4373	4386						
		4402	4407	4416	4426	4436	4443	4446	4448	4461	4466	4480	4509	4511						
		4521	4526	4536	4557	4571	4579	4581	4591	4601	4609	4617	4630	4641						
		4643	4658	4664	4685	4693	4703	4710	4712	4723	4729	4756	4766	4782						
		4784	4798	4812	4825	4842	4854	4869	4881	4883	4894	4901	4915	4930						
		4944	4954	4965	4975	4991	4993	5005	5010	5024	5037	5047	5053	5055						
		5067	5072	5086	5099	5110	5116	5118	5129	5134	5148	5161	5178	5180						
		5189	5191	5203	5214	5225	5237	5248	5250	5261	5273	5287	5296	5309						
		5320	5329	5338	5340	5349	5360	5370	5380	5382	5384	5396	5400	5410						

G\$LOLI=	000001	1084#																		
G\$NO	= 000000	1084#																		
G\$OFFS=	000400	1084#	7134	7139	7143	7147	7152	7157	7162											
G\$OF SI=	000376	1084#	7134	7139	7143	7147	7152	7157	7162											
G\$PRMA=	000001	1084#	7139	7143																
G\$PRMD=	000002	1084#	7134	7147	7152	7157	7162													
G\$PRML=	000000	1084#																		
G\$RADA=	000140	1084#																		
G\$RADB=	000000	1084#																		
G\$RADD=	000040	1084#																		
G\$RADL=	000120	1084#																		
G\$RADO=	000020	1084#	7134	7139	7143	7147	7152	7157	7162											
G\$XFER=	000004	1084#																		
G\$YES =	000010	1084#	7134	7139	7143	7147	7152	7157	7162											
HELP =	000000	1083#	1115	1202	1260	1584	3227	3231												
HOE =	100000	G	1396#																	
IBE =	010000	G	1393#																	
IDU =	000040	G	1386#																	
IER =	020000	G	1394#																	
I FLAG	002332	1450#	4294*	4309	4330*	4345	4413*	4440*												
I FTYP	002470	1510#	2446	2462	2477	3323*	3324*													
I FTYPM	032443	7163	7233#																	
INTCO	020364	G	4279	4410#																
INTC4	020400	G	4287	4417#																
INTDO	020420	G	4318	4427#																
INTD4	020440	G	4324	4437#																
ISR =	000100	G	1387#																	
IXE =	004000	G	1392#																	
ISAU =	000041	1084#	3390#	3393#																
ISAUTO=	000041	1084#	3331#	3347#																
ISCLN =	000041	1084#	3355#	3360#																
ISDU =	000041	1084#	3372#	3377#																
ISHRD =	000041	7132#	7171#																	
ISINIT=	000041	1084#	3246#	3329#																
ISMOD =	000040	1084#	1092#																	
ISMSG =	000041	1084#	2666#	2675#	2679#	2689#	2690#	2701#	2702#	2713#	2716#	2725#	2728#	2737#						
			2742#	2755#	2758#	2770#	2771#	2781#	2782#	2799#	2800#	2820#	2821#	2833#	2834#					
			2846#	2849#	2860#	2865#	2885#	2888#	2897#											
ISPROT=	000040	1084#	1202#																	
ISPTAB=	000041	1084#																		
ISPWR =	000041	1084#																		
ISRPT =	000041	1084#	3225#	3234#																
ISSEG =	000041	1084#	3419	3476	3551	3626	3701	3776	3851	3927	3983	4083	4162	4461						
			4521	4591	4658	4723	4798	4894	5005	5067	5129	5189	5191	5250	5340					
			5396	5503	5689	5737	5784	5881	5980	6148	6181	6224	6285	6352	6405					
			6447	6488	6655	6715	6776	6836	6869	6899										
ISSETU=	000041	1084#																		
ISSFT =	000041	7260#	7265#																	
ISSRV =	000041	1084#	3447#	3464#	4410#	4416#	4417#	4426#	4427#	4436#	4437#	4443#								
ISSUB =	000041	1084#	3419	3476	3551	3626	3701	3776	3851	3927	3983	4083	4162	4461						
			4521	4591	4658	4723	4798	4894	5005	5067	5129	5189	5191#	5203	5214					
			5225	5237	5248#	5250#	5261	5273	5287	5309	5320	5329	5338#	5340#	5349					
			5360	5370	5380#	5382#	5396	5503	5689	5737	5784	5881	5980	6148	6181					
			6224	6285	6352	6405	6447	6488	6655	6715	6776	6836	6869	6899						
ISTST	000041	1084#	3419#	3442#	3444#	3476#	3540#	3542#	3551#	3615#	3617#	3626#	3690#	3692#						
			3701#	3765#	3767#	3776#	3840#	3842#	3851#	3915#	3917#	3927#	3935	3944	3957					

CZDMTAD DMP/V-11 - CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 182
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0180

L\$ETP	002102	G	1185#		
L\$EXP1	002046	G	1157#		
L\$EXP4	002064	G	1171#		
L\$EXP5	002066	G	1173#		
L\$HARD	031622	G	1134	7132	7133#
L\$HIME	002120	G	1199#		
L\$HPCP	002016	G	1133#		
L\$HPTP	002022	G	1137#		
L\$HW	002262	G	1138	1273	1274#
L\$ICP	002104	G	1187#		
L\$INIT	013206	G	1188	3246#	
L\$LADP	002026	G	1141#		
L\$LAST	036746	G	1142	7332#	
L\$LOAD	002100	G	1183#		
L\$LUN	002074	G	1179#		
L\$MREV	002050	G	1159#		
L\$NAME	002000	G	1116#		
L\$PRIO	002042	G	1153#		
L\$PROT	002122	G	1194	1202#	
L\$PRT	002112	G	1193#		
L\$REPP	002062	G	1169#		
L\$REV	002010	G	1125#		
L\$RPT	013200	G	3225#		
L\$SOFT	032534	G	7260	7261#	
L\$SPC	002056	G	1165#		
L\$SPCP	002020	G	1135#		
L\$SPTP	002024	G	1139#		
L\$STA	002030	G	1143#		
L\$SW	002314	G	1307	1308#	
L\$TEST	002114	G	1195#		
L\$TIML	002014	G	1131#		
L\$UNIT	002012	G	1129#	3289	
L10001	002312		1273	1294#	
L10002	002314		1307	1312#	
L10003	006712		2673#		
L10004	006744		2687#		
L10005	007002		2699#		
L10006	007040		2711#		
L10007	007066		2723#		
L10010	007114		2735#		
L10011	007162		2753#		
L10012	007222		2768#		
L10013	007254		2779#		
L10014	007336		2797#		
L10015	007430		2818#		
L10016	007472		2831#		
L10017	007534		2844#		
L10020	007572		2858#		
L10021	007670		2883#		
L10022	007716		2895#		
L10023	013204		3229	3232#	
L10024	013616		3327#		
L10025	013660		3345#		
L10026	013664		3358#		
L10027	013670		3375#		
L10030	013672		3391#		

CZDMTAD DMP/V-11 r CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 186
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0184

RETADR	002474	1512#												
RET16	027712	6661	6696#											
RET17	030154	6721	6756#											
RET18	030416	6783	6818#											
REVNO	013176	3215#	3521*	3522	3596*	3597	3671*	3672	3746*	3747	3821*	3822	3896*	3897
		3949*	3960	4004*	4016									
RMVAA	005474	2378	2381#											
RMVBB	005514	2382	2385#											
RMVCC	005540	2387	2393#											
RMVEX	005540	2362	2368	2392#										
RMVRT	005326	2356#	3933	3942	3990	3999								
RMVXX	005360	2363#	2384											
RMVYY	005416	2370#	2374											
RMVZZ	005462	2376	2379#											
ROMADD	002430	1483#	2357*	2364	2375	2381	2383*							
ROMBK1	002432	1484#	2375	3931*	3940*	3988*	3997*							
ROMBK2	002434	1485#	2381	3932*	3941*	3989*	3998*							
ROMCLK	004274	2097	2101	2119#										
ROMI =	000002	1417#												
ROMMSG	013076	3202#	3525	3600	3675	3750	3825	3900	3963	4019				
ROMN	002364	1464#	2824	2837	3479*	3554*	3629*	3704*	3779*	3854*				
ROMNO	013174	3214#	3518*	3523	3593*	3598	3668*	3673	3743*	3748	3818*	3823	3893*	3898
		3947*	3950	3961	4006*	4007	4017							
ROMN1	002366	1465#	2518	2836	3506*	3509	3581*	3584	3656*	3659	3731*	3734	3806*	3809
		3881*	3884	3930*	3950	3986*	4007	6406*	6838*					
ROMO =	000004	1416#												
RQ1 =	000200	1425#	1770	1775	2252	2588	2607	4125	4474	4532	4542	4605	4689	4695
		4762	4771	5437	5444	5701	5707	6192	6198	6249	6255	6317	6323	6363
		6370	7008	7014										
RUN =	000200	1412#												
RXADD	002414	1477#	2593	2629	2633	6150*	6409*	6451*	6686*	6746*	6808*	6843*	6875*	
RXCC	002416	1478#	2594	6149*	6408*	6450*	6687*	6747*	6809*	6844*	6874*			
SAVE4	002314	1443#	2058	3253*	3256									
SAVE6	002316	1444#	2059	3254*	3257									
SELO	002436	1489#	2095*	2103*	2248	2307	2317	2667	3336	3428	4089			
SEL2	002442	1492#	2096*	2316	3306*	3307*	3430							
SEL4	002446	1496#	2315	3304*	3305*	3432								
SEL6	002452	1499#	2314	3314*	3315*	3434	3485	3488	3508	3521	3532	3635	3638	3658
		3671	3682	3785	3788	3808	3821	3832						
SETIF	006062	2456	2472	2485	2495	2528#								
SFPTBL	002314	1309#												
SPEDM	032232	7158	7208#											
SPEEDM	002462	1505#	2450	2466	3322*									
STAND	005066	2306#	2686	2698	2710	2722	2734	2752	2767	2778	2796	2817	2830	2843
		2857	2882	2894										
STARES	002344	1455#	2499	2510	3277*	3283*	3516	3591	3666	3741	3816	3891	3937	3994
STARST	013332	3262	3266	3276#										
STEPLU=	000020	1414#												
STEPMP=	000001	1418#												
SUBRPC	002322	1446#	3249*											
SVCGBL=	000000	1084#	1092	1099#	1116	1117	1125	1126	1127	1128	1129	1130	1131	1132
		1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145
		1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158
		1159	1160	1162	1163	1165	1166	1167	1168	1169	1170	1171	1172	1173
		1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186
		1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199

SVCINS= 000000

1200	1202	1203	1215	1216	1274	1275	1276	1308	1309	1310	1565	1566
1571	1572	2666	2667	2679	2680	2690	2691	2702	2703	2716	2717	2728
2729	2742	2743	2758	2759	2771	2772	2782	2783	2800	2801	2821	2822
2834	2835	2849	2850	2865	2866	2888	2889	3225	3226	3246	3247	3331
3332	3355	3356	3372	3373	3390	3391	3447	3448	4410	4411	4417	4418
4427	4428	4437	4438	7133	7134	7261	7262	7332#	7333			
1084#	1096#	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127
1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140
1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153
1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166
1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179
1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192
1193	1194	1195	1196	1197	1198	1199	1200	1201	1214	1215	1216	1217
1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230
1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243
1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256
1257	1258	1259	1273	1274	1307	1308	1566	1569	1570	1572	1578	1579
1635	1636	1637	1638	1639	1678	1679	1680	1681	1682	1725	1726	1727
1728	1729	1739	1740	1741	1742	1743	1815	1816	1817	1818	1819	1827
1828	1829	1830	1831	1856	1857	1858	1859	1860	1886	1887	1888	1889
1890	1915	1916	1917	1918	1919	1943	1944	1945	1946	1947	1972	1973
1974	1975	1976	2016	2017	2018	2019	2020	2029	2030	2031	2032	2033
2045	2046	2047	2048	2049	2052	2053	2054	2055	2056	2233	2234	2235
2236	2237	2277	2278	2279	2280	2281	2285	2286	2317	2308	2309	2310
2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323
2388	2389	2390	2391	2392	2501	2502	2503	2504	2505	2506	2512	2513
2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2642	2643
2644	2645	2646	2667	2668	2669	2670	2671	2672	2673	2674	2675	2680
2681	2682	2683	2684	2685	2686	2688	2689	2691	2692	2693	2694	2695
2696	2697	2698	2700	2701	2703	2704	2705	2706	2707	2708	2709	2710
2712	2713	2717	2718	2719	2720	2721	2722	2724	2725	2729	2730	2731
2732	2733	2734	2736	2737	2743	2744	2745	2746	2747	2748	2749	2750
2751	2752	2754	2755	2759	2760	2761	2762	2763	2764	2765	2766	2767
2769	2770	2772	2773	2774	2775	2776	2777	2778	2780	2781	2783	2784
2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2798
2799	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812
2813	2814	2815	2816	2817	2819	2820	2822	2823	2824	2825	2826	2827
2828	2829	2830	2832	2833	2835	2836	2837	2838	2839	2840	2841	2842
2843	2845	2846	2850	2851	2852	2853	2854	2855	2856	2857	2859	2860
2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878
2879	2880	2881	2882	2884	2885	2889	2890	2891	2892	2893	2894	2896
2897	3228	3229	3230	3233	3234	3260	3261	3262	3263	3264	3265	3266
3267	3268	3269	3270	3271	3272	3273	3274	3275	3291	3292	3293	3294
3295	3328	3329	3340	3341	3342	3344	3345	3346	3347	3356	3357	3359
3360	3374	3375	3376	3377	3392	3393	3420	3421	3422	3423	3424	3425
3426	3437	3438	3439	3440	3441	3443	3444	3450	3451	3452	3453	3454
3456	3457	3458	3459	3460	3461	3462	3463	3464	3501	3502	3503	3504
3505	3511	3512	3513	3514	3515	3522	3523	3524	3525	3526	3527	3528
3529	3530	3535	3536	3537	3538	3539	3541	3542	3576	3577	3578	3579
3580	3586	3587	3588	3589	3590	3597	3598	3599	3600	3601	3602	3603
3604	3605	3610	3611	3612	3613	3614	3616	3617	3651	3652	3653	3654
3655	3661	3662	3663	3664	3665	3672	3673	3674	3675	3676	3677	3678
3679	3680	3685	3686	3687	3688	3689	3691	3692	3726	3727	3728	3729
3730	3736	3737	3738	3739	3740	3747	3748	3749	3750	3751	3752	3753
3754	3755	3760	3761	3762	3763	3764	3766	3767	3801	3802	3803	3804
3805	3811	3812	3813	3814	3815	3822	3823	3824	3825	3826	3827	3828

3829	3830	3835	3836	3837	3838	3839	3841	3842	3876	3877	3878	3879
3880	3886	3887	3888	3889	3890	3897	3898	3899	3900	3901	3902	3903
3904	3905	3910	3911	3912	3913	3914	3916	3917	3935	3936	3937	3944
3945	3946	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962
3963	3964	3965	3966	3967	3968	3973	3974	3992	3993	3994	4001	4002
4003	4009	4010	4011	4012	4013	4014	4015	4016	4017	4018	4019	4020
4021	4022	4023	4024	4028	4029	4096	4097	4098	4099	4100	4101	4102
4107	4108	4109	4110	4111	4113	4114	4115	4117	4118	4119	4120	4121
4122	4123	4130	4131	4132	4139	4140	4141	4142	4143	4174	4175	4176
4182	4183	4184	4197	4198	4199	4217	4218	4219	4220	4221	4222	4223
4237	4238	4239	4255	4256	4257	4258	4259	4260	4261	4278	4279	4280
4281	4282	4283	4284	4286	4287	4288	4289	4290	4291	4292	4296	4297
4298	4303	4304	4305	4317	4318	4319	4320	4321	4322	4323	4324	4325
4326	4327	4328	4329	4331	4332	4333	4339	4340	4341	4352	4353	4354
4355	4356	4373	4374	4375	4381	4382	4383	4384	4385	4386	4387	4388
4398	4399	4400	4401	4402	4403	4404	4407	4408	4409	4411	4412	4413
4415	4416	4418	4419	4420	4421	4422	4423	4424	4425	4426	4428	4429
4430	4431	4432	4433	4434	4435	4436	4438	4439	4440	4442	4443	4447
4448	4466	4467	4468	4480	4481	4482	4501	4502	4503	4504	4505	4510
4511	4526	4527	4528	4536	4537	4538	4552	4553	4554	4555	4556	4557
4558	4559	4566	4567	4568	4569	4570	4571	4572	4573	4580	4581	4601
4602	4603	4609	4610	4611	4617	4618	4619	4620	4627	4628	4629	4630
4631	4632	4636	4637	4638	4639	4640	4642	4643	4664	4665	4666	4681
4682	4683	4684	4685	4686	4687	4693	4694	4695	4703	4704	4705	4711
4712	4729	4730	4731	4756	4757	4758	4766	4767	4768	4777	4778	4779
4780	4781	4783	4784	4812	4813	4814	4825	4826	4827	4842	4843	4844
4854	4855	4856	4869	4870	4871	4882	4883	4901	4902	4903	4915	4916
4917	4930	4931	4932	4944	4945	4946	4954	4955	4956	4965	4966	4967
4975	4976	4977	4992	4993	5010	5011	5012	5024	5025	5026	5037	5038
5039	5047	5048	5049	5054	5055	5072	5073	5074	5086	5087	5088	5099
5100	5101	5110	5111	5112	5117	5118	5134	5135	5136	5148	5149	5150
5161	5162	5163	5172	5173	5174	5175	5176	5179	5180	5192	5193	5203
5204	5205	5214	5215	5216	5225	5226	5227	5237	5238	5239	5249	5250
5251	5252	5261	5262	5263	5273	5274	5275	5287	5288	5289	5296	5297
5298	5309	5310	5311	5320	5321	5322	5329	5330	5331	5339	5340	5341
5342	5349	5350	5351	5360	5361	5362	5370	5371	5372	5381	5382	5383
5384	5400	5401	5402	5410	5411	5412	5418	5419	5420	5426	5427	5428
5431	5432	5433	5442	5443	5444	5455	5456	5457	5461	5462	5463	5466
5467	5468	5471	5472	5473	5477	5478	5479	5483	5484	5485	5494	5495
5508	5509	5510	5522	5523	5524	5535	5536	5537	5546	5547	5548	5559
5560	5561	5562	5563	5572	5573	5574	5585	5586	5587	5595	5596	5597
5608	5609	5610	5619	5620	5621	5630	5631	5632	5644	5645	5646	5657
5658	5659	5667	5668	5669	5674	5675	5695	5696	5697	5705	5706	5707
5713	5714	5715	5724	5725	5742	5743	5744	5753	5754	5755	5763	5764
5765	5770	5771	5789	5790	5791	5803	5804	5805	5813	5814	5815	5823
5824	5825	5833	5834	5835	5843	5844	5845	5853	5854	5855	5863	5864
5865	5866	5867	5869	5870	5886	5887	5888	5900	5901	5902	5910	5911
5912	5920	5921	5922	5930	5931	5932	5940	5941	5942	5950	5951	5952
5960	5961	5962	5963	5964	5966	5967	5985	5986	5987	5999	6000	6001
6013	6014	6015	6026	6027	6028	6040	6041	6042	6053	6054	6055	6066
6067	6068	6080	6081	6082	6093	6094	6095	6106	6107	6108	6119	6120
6121	6132	6133	6159	6160	6161	6167	6168	6169	6170	6171	6186	6187
6188	6196	6197	6198	6207	6208	6209	6213	6214	6229	6230	6231	6243
6244	6245	6253	6254	6255	6264	6265	6266	6270	6271	6290	6291	6292
6301	6302	6303	6311	6312	6313	6321	6322	6323	6332	6333	6334	6339
6340	6357	6358	6359	6368	6369	6370	6380	6381	6382	6386	6387	6418

CZDMTAD DMP/V-11 PCTL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 190
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0188

TLB4	020020	4299#	4310											
TLB5	020142	4335#	4346											
TLB6	020242	4369#	4391											
TLB6A	020220	4364#	4405											
TLB7	004514	2210#	2216											
TLB8	004546	2218#	2225											
TLB9	004752	2254#	2258											
TMP0	002540	1540#												
TMP1	002542	1541#												
TMP2	002544	1542#												
TMP3	002546	1543#												
TMP4	002550	1544#												
TMP5	002552	1545#												
TMP6	002554	1546#												
TMP7	002556	1547#												
TOUT	005016	1624	1667	2210	2218	2245	2254	2273#	4128	4195	4235	4301	4337	4371
		4599												
TOUTE	005046	2274	2282#											
TOUTEX	005056	2276	2284#											
TRIBH	002362	1462#	4806*	4807*	4831									
TRIBMX	002360	1461#	4799*	4802*	4807									
TRIBN	002356	1460#	1774	1822	2439	2440*	2544*	2595	2606	4805*	4830*	4831	4861*	4907*
		5016*	5078*	5140*	5198*	5257*	5355*	5402*	5447	5514*	5554	5748*	5795*	5892*
		5991*	6154*	6235*	6258	6296*	6326	6413*	6452*	6508*	6684*	6744*	6806*	6845*
		6871*	6909*	6920*										
TSSADD	002404	1472#	4895*	4922	4936	4960	4985	4987*	5197*	5209	5232	5244*	5245	5256*
		5267	5269	5279*	5280	5283	5303*	5304	5325	5335				
TSTCON	002472	1511#	2431	2532	2535	3321*	4673	4735	4748	6491				
TXADD	002412	1476#	2604	2634	6151*	6411*	6449*	6688*	6748*	6810*	6842*	6873*		
TXCC	002420	1479#	2605	2625	2650	6152*	6410*	6448*	6689*	6749*	6811*	6841*	6872*	
TXRXA	006304	2531	2542	2552	2563	2570	2574	2584#						
TXRXB	006270	2555	2580#											
TXRXC	006306	2576	2588#											
TXRXEN	006664	2524	2584	2592	2603	2610	2619	2624	2628	2632	2648	2653	2657#	
TXRXG	006472	2612	2621#											
TXRXSR	005542	2431#	6424	6853										
TXRX1	006100	2532#												
TXRX1A	006130	2536	2538#											
TXRX2	006146	2533	2544#											
TXRX3	006154	2433	2548#	6165	6464	6691	6751	6813	6885					
TYEND	002412	1475#	5719											
TYLST	002406	1473#	5690											
TSARGC=	000004	1117#	1118#	1119#	1120#	1121#	1122#	2307#	2313	2314#	2322	2501#	2505	2512#
		2516	2518#	2523	2667#	2672	2680#	2685	2691#	2697	2703#	2709	2717#	2721
		2729#	2733	2743#	2751	2759#	2766	2772#	2777	2783#	2787	2788#	2795	2801#
		2805	2806#	2811	2812#	2816	2822#	2829	2835#	2842	2850#	2856	2866#	2872
		2873#	2881	2889#	2893	3456#	3461	3522#	3529	3597#	3604	3672#	3679	3747#
		3754	3822#	3829	3897#	3904	3960#	3967	4016#	4023				
TSBERRN=	000055	7134#	7139#	7143#	7147#	7152#	7157#	7162#						
		1084#	1636#	1679#	1726#	1740#	1816#	1828#	1857#	1887#	1916#	1944#	1973#	2017#
		2030#	2046#	2053#	2234#	2278#	2389#	2643#	3451#	3502#	3512#	3536#	3577#	3587#
		3611#	3652#	3662#	3686#	3727#	3737#	3761#	3802#	3812#	3836#	3877#	3887#	3911#
		3953#	4010#	4097#	4108#	4118#	4218#	4256#	4382#	4399#	4421#	4431#	4502#	4553#
		4567#	4627#	4637#	4682#	4778#	5173#	5560#	5864#	5961#	6582#			
TSBEXCP=	000000	7134#	7139#	7143#	7147#	7152#	7157#	7162#	7167					
TSFLAG=	000040	3228#	3230	3935#	3944#	3957#	3992#	4001#	4013#	4100#	4113#	4121#	4130#	4139#

CZDMTAD DMP/V-11 r CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 192
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0190

TSSSEGL= 177777
TSSUBN= 000000

1084#	1084#	3419#	3476#	3551#	3626#	3701#	3776#	3851#	3927#	3983#	4083#	4162#	4461#
4521#	4591#	4658#	4723#	4798#	4894#	5005#	5067#	5067#	5129#	5189#	5191#	5250#	5340#
5396#	5503#	5689#	5737#	5784#	5881#	5980#	6148#	6148#	6181#	6224#	6285#	6352#	6405#
6447#	6488#	6655#	6715#	6776#	6836#	6869#	6899#						

TSTAGL= 177777
TSTAGN= 010116

1084#	1084#	1202#	1273#	1307#	2666#	2679#	2690#	2702#	2716#	2728#	2742#	2758#	2771#
2782#	2800#	2821#	2834#	2849#	2865#	2888#	3225#	3225#	3246#	3331#	3355#	3372#	3390#
3420#	3447#	3477#	3552#	3627#	3702#	3777#	3852#	3852#	3928#	3984#	4084#	4163#	4410#
4417#	4427#	4437#	4462#	4522#	4592#	4659#	4724#	4724#	4799#	4895#	5006#	5068#	5130#
5190#	5192#	5251#	5341#	5397#	5504#	5690#	5738#	5738#	5785#	5882#	5981#	6149#	6182#
6225#	6286#	6353#	6406#	6448#	6489#	6656#	6716#	6716#	6777#	6837#	6870#	6900#	7132#

TSTEMP= 000005

7260#	1206#	1216#	1217#	1218#	1219#	1220#	1221#	1222#	1223#	1224#	1225#	1226#	1227#
1228#	1229#	1230#	1231#	1232#	1233#	1234#	1235#	1236#	1237#	1238#	1239#	1240#	
1241#	1242#	1243#	1244#	1245#	1246#	1247#	1248#	1249#	1250#	1251#	1252#	1253#	
1254#	1255#	1256#	1257#	1258#	1259#	1294#	1312#	2673#	2687#	2699#	2711#	2723#	
2735#	2753#	2768#	2779#	2797#	2818#	2831#	2844#	2858#	2883#	2895#	3228#	3229#	
3232#	3327#	3345#	3358#	3375#	3391#	3442#	3462#	3540#	3615#	3690#	3765#	3840#	
3915#	3935#	3936#	3944#	3945#	3957#	3958#	3972#	3992#	3993#	4001#	4002#	4013#	
4014#	4027#	4100#	4101#	4113#	4114#	4121#	4122#	4130#	4131#	4139#	4140#	4141#	
4174#	4175#	4182#	4183#	4197#	4198#	4221#	4222#	4237#	4238#	4259#	4260#	4303#	
4304#	4339#	4340#	4373#	4374#	4386#	4387#	4402#	4403#	4407#	4408#	4414#	4424#	
4434#	4441#	4446#	4466#	4467#	4480#	4481#	4509#	4526#	4527#	4536#	4537#	4557#	
4558#	4571#	4572#	4579#	4601#	4602#	4609#	4610#	4617#	4618#	4630#	4631#	4641#	
4664#	4665#	4685#	4686#	4693#	4694#	4703#	4704#	4710#	4729#	4730#	4756#	4757#	
4766#	4767#	4782#	4812#	4813#	4825#	4826#	4842#	4843#	4854#	4855#	4869#	4870#	
4881#	4901#	4902#	4915#	4916#	4930#	4931#	4944#	4945#	4954#	4955#	4965#	4966#	
4975#	4976#	4991#	5010#	5011#	5024#	5025#	5037#	5038#	5047#	5048#	5053#	5072#	
5073#	5086#	5087#	5099#	5100#	5110#	5111#	5116#	5134#	5135#	5148#	5149#	5161#	
5162#	5178#	5203#	5204#	5214#	5215#	5225#	5226#	5237#	5238#	5248#	5261#	5262#	
5273#	5274#	5287#	5288#	5296#	5297#	5309#	5310#	5320#	5321#	5329#	5330#	5338#	
5349#	5350#	5360#	5361#	5370#	5371#	5380#	5382#	5400#	5401#	5410#	5411#	5418#	
5419#	5426#	5427#	5431#	5432#	5442#	5443#	5455#	5456#	5461#	5462#	5466#	5467#	
5471#	5472#	5477#	5478#	5483#	5484#	5493#	5508#	5509#	5522#	5523#	5535#	5536#	
5546#	5547#	5572#	5573#	5585#	5586#	5595#	5596#	5608#	5609#	5619#	5620#	5630#	
5631#	5644#	5645#	5657#	5658#	5667#	5668#	5673#	5695#	5696#	5705#	5706#	5713#	
5714#	5723#	5742#	5743#	5753#	5754#	5763#	5764#	5769#	5789#	5790#	5803#	5804#	
5813#	5814#	5823#	5824#	5833#	5834#	5843#	5844#	5853#	5854#	5868#	5886#	5887#	
5900#	5901#	5910#	5911#	5920#	5921#	5930#	5931#	5940#	5941#	5950#	5951#	5965#	
5985#	5986#	5999#	6000#	6013#	6014#	6026#	6027#	6040#	6041#	6053#	605#	6066#	
6067#	6080#	6081#	6093#	6094#	6106#	6107#	6119#	6120#	6131#	6159#	6160#	6167#	
6168#	6169#	6186#	6187#	6196#	6197#	6207#	6208#	6212#	6229#	6230#	6243#	6244#	
6253#	6254#	6264#	6265#	6269#	6290#	6291#	6301#	6302#	6311#	6312#	6321#	6322#	
6332#	6333#	6338#	6357#	6358#	6368#	6369#	6380#	6381#	6385#	6418#	6419#	6427#	
6428#	6433#	6458#	6459#	6468#	6469#	6474#	6498#	6499#	6517#	6518#	6530#	6531#	
6543#	6544#	6556#	6557#	6569#	6570#	6585#	6586#	6592#	6593#	6632#	6693#	6694#	
6697#	6753#	6754#	6757#	6815#	6816#	6819#	6848#	6849#	6855#	6880#	6881#	6887#	
6904#	6905#	6914#	6915#	6924#	6925#	6933#	6934#	6942#	6943#	6950#	6951#	6958#	
6959#	6963#	6964#	6972#	6973#	6977#	6978#	6991#	6992#	7012#	7013#	7026#	7027#	
7048#	7049#	7053#	7054#	7069#	7070#	7074#	7075#	7083#	7084#	7089#	7090#	7101#	
7102#	7106#	7107#	7110#	7134#	7139#	7143#	7147#	7152#	7157#	7162#	7169#	7263#	
1084#	3405#	3418#	3419#	3467#	3474#	3476#	3542#	3549#	3551#	3617#	3624#	3626#	
3692#	3699#	3701#	3767#	3774#	3776#	3842#	3849#	3851#	3917#	3926#	3927#	3974#	
3982#	3983#	4031#	4082#	4083#	4145#	4159#	4162#	4450#	4460#	4461#	4512#	4520#	
4521#	4581#	4590#	4591#	4645#	4657#	4658#	4713#	4722#	4723#	4785#	4796#	4798#	

*\$TEST= 000053

	4884	4893	4894#	4995	5004	5005#	5057	5066	5067#	5120	5128	5129#	5180
	5188	5189#	5191	5250	5340	5386	5395	5396#	5496	5502	5503#	5678	5688
	5689#	5727	5736	5737#	5773	5783	5784#	5871	5880	5881#	5968	5979	5980#
	6136	6146	6148#	6172	6180	6181#	6215	6223	6224#	6275	6284	6285#	6342
	6351	6352#	6392	6404	6405#	6435	6446	6447#	6477	6487	6488#	6639	6654
	6655#	6699	6714	6715#	6760	6775	6776#	6824	6835	6836#	6858	6868	6869#
	6889	6898	6899#	7333									
TSTSTM= 17777	1084#	1635	1678	1725	1739	1815	1827	1856	1886	1915	1943	1972	2016
	2029	2045	2052	2233	2277	2285	2312	2321	2388	2504	2515	2522	2642
	2671	2674	2684	2688	2696	2700	2708	2712	2720	2724	2732	2736	2750
	2754	2765	2769	2776	2780	2786	2794	2798	2804	2810	2815	2819	2828
	2832	2841	2845	2855	2859	2871	2880	2884	2892	2896	3233	3261	3265
	3269	3273	3292	3328	3341	3344	3346	3356	3359	3374	3376	3392	3424
	3437	3440	3443	3450	3460	3501	3511	3528	3535	3541	3576	3586	3603
	3610	3616	3651	3661	3678	3685	3691	3726	3736	3753	3760	3766	3801
	3811	3828	3835	3841	3876	3886	3903	3910	3916	3935	3944	3952	3957
	3966	3973	3992	4001	4009	4013	4022	4028	4096	4100	4107	4113	4117
	4121	4130	4139	4142	4174	4182	4197	4217	4221	4237	4255	4259	4282
	4290	4297	4303	4321	4327	4332	4339	4353	4355	4373	4381	4386	4398
	4402	4407	4412	4419	4420	4429	4430	4439	4447	4466	4480	4501	4510
	4526	4536	4552	4557	4566	4571	4580	4601	4609	4617	4626	4630	4636
	4642	4664	4681	4685	4693	4703	4711	4729	4750	4766	4777	4783	4812
	4825	4842	4854	4869	4882	4901	4915	4930	4944	4954	4965	4975	4992
	5010	5024	5037	5047	5054	5072	5086	5099	5110	5117	5134	5148	5161
	5172	5179	5192	5203	5214	5225	5237	5249	5251	5261	5273	5287	5296
	5309	5320	5329	5339	5341	5349	5360	5370	5381	5383	5400	5410	5418
	5426	5431	5442	5455	5461	5466	5471	5477	5483	5494	5508	5522	5535
	5546	5559	5572	5585	5595	5608	5619	5630	5644	5657	5667	5674	5695
	5705	5713	5724	5742	5753	5763	5770	5789	5803	5813	5823	5833	5843
	5853	5863	5869	5886	5900	5910	5920	5930	5940	5950	5960	5966	5985
	5999	6013	6026	6040	6053	6066	6080	6093	6106	6119	6132	6159	6167
	6170	6186	6196	6207	6213	6229	6243	6253	6264	6270	6290	6301	6311
	6321	6332	6339	6357	6368	6380	6386	6418	6427	6434	6458	6468	6475
	6498	6517	6530	6543	6556	6569	6581	6585	6592	6633	6668	6693	6698
	6728	6753	6758	6790	6815	6820	6848	6856	6880	6888	6904	6914	6924
	6933	6942	6950	6958	6963	6972	6977	6991	7012	7026	7048	7053	7069
	7074	7083	7089	7101	7106	7111							
TSTSTS= 000001	1084#	3420#	3477#	3552#	3627#	3702#	3777#	3852#	3928#	3984#	4084#	4163#	4462#
	4522#	4592#	4659#	4724#	4799#	4895#	5006#	5068#	5130#	5190#	5397#	5504#	5690#
	5738#	5785#	5882#	5981#	6149#	6182#	6225#	6286#	6353#	6406#	6448#	6489#	6656#
	6716#	6777#	6837#	6870#	6900#								
TSSAU = 010030	3390#	3391											
TSSAUT= 010025	3331#	3345											
TSSCLE= 010026	3355#	3358											
TSSDU = 010027	3372#	3375											
TSSHAR= 010114	7132#	7170											
TSSHW = 010001	1273#	1294											
TSSINI= 010024	3246#	3327											
TSSMSG= 010022	2666#	2673	2679#	2687	2690#	2699	2702#	2711	2716#	2723	2728#	2735	2742#
	2753	2758#	2768	2771#	2779	2782#	2797	2800#	2818	2821#	2831	2834#	2844
	2849#	2858	2865#	2883	2888#	2895							
TSSPRO= 010000	1202#												
TSSRPT= 010023	3225#	3228	3232										
TSSSOF= 010115	7260#	7264											
TSSSRV= 010050	3447#	3462	4410#	4414	4417#	4424	4427#	4434	4437#	4441			
TSSSUB= 010066	5192#	5203	5214	5225	5237	5248	5251#	5261	5273	5287	5309	5320	5329

TSSW = 010002
 TSTES= 010113

5338	5341#	5349	5360	5370	5380										
1307#	1312														
3420#	3442	3477#	3540	3552#	3615	3627#	3690	3702#	3765	3777#	3840	3852#			
3915	3928#	3935	3944	3957	3972	3984#	3992	4001	4013	4027	4084#	4100			
4113	4121	4130	4139	4141	4163#	4174	4182	4197	4221	4237	4259	4303			
4339	4373	4386	4402	4407	4446	4462#	4466	4480	4509	4522#	4526	4536			
4557	4571	4579	4592#	4601	4609	4617	4630	4641	4659#	4664	4685	4693			
4703	4710	4724#	4729	4756	4766	4782	4799#	4812	4825	4842	4854	4869			
4881	4895#	4901	4915	4930	4944	4954	4965	4975	4991	5006#	5010	5024			
5037	5047	5053	5068#	5072	5086	5099	5110	5116	5130#	5134	5148	5161			
5178	5190#	5296	5382	5397#	5400	5410	5418	5426	5431	5442	5455	5461			
5466	5471	5477	5483	5493	5504#	5508	5522	5535	5546	5572	5585	5595			
5608	5619	5630	5644	5657	5667	5673	5690#	5695	5705	5713	5723	5738#			
5742	5753	5763	5769	5785#	5789	5803	5813	5823	5833	5843	5853	5868			
5882#	5886	5900	5910	5920	5930	5940	5950	5965	5981#	5985	5999	6003			
6026	6040	6053	6066	6080	6093	6106	6119	6131	6149#	6159	6167	6169			
6182#	6186	6196	6207	6212	6225#	6229	6243	6253	6264	6269	6286#	6290			
6301	6311	6321	6332	6338	6353#	6357	6368	6380	6385	6406#	6418	6427			
6433	6448#	6458	6468	6474	6489#	6498	6517	6530	6543	6556	6569	6585			
6592	6632	6656#	6693	6697	6716#	6753	6757	6777#	6815	6819	6837#	6848			
6855	6870#	6880	6887	6900#	6904	6914	6924	6933	6942	6950	6958	6963			
6972	6977	6991	7012	7026	7048	7053	7069	7074	7083	7089	7101	7106			
7110															

T1 013674 G
 T10 017140 G
 T11 017324 G
 T12 020456 G
 T13 020570 G
 T14 021000 G
 T14A 021206
 T14EX 021242
 T15 021244 G
 T16 021454 G
 T17 021706 G
 T18 022140 G
 T19 022424 G
 T2 014054 G
 T20 022532 G
 T21 022640 G
 T22 022754 G
 T22.1 022754
 T22.2 023116
 T22.3 023374
 T23 023502 G
 T24 024130 G
 T25 024556 G
 T26 024672 G
 T27 024756 G
 T28 025204 G
 T29 025432 G
 T3 014406 G
 T30 025760 G
 T31 026054 G
 T32 026174 G
 T33 026342 G
 T34 026530 G

1216 3419#
 1225 4083#
 1226 4162#
 1227 4461#
 1228 4521#
 1229 4591#
 4624 4632#
 4622 4634 4640#
 1230 4658#
 1231 4723#
 1232 4798#
 1233 4894#
 1234 5005#
 1217 3476#
 1235 5067#
 1236 5129#
 1237 5189#
 5191#
 5250#
 5340#
 1238 5396#
 1239 5503#
 1240 5689#
 1241 5737#
 1242 5784#
 1243 5881#
 1244 5980#
 1218 3551#
 1245 6148#
 1246 6181#
 1247 6224#
 1248 6285#
 1249 6352#

T35	026646	G	1250	6405#																		
T36	026750	G	1251	6447#																		
T37	027044	G	1252	6488#																		
T38	027456	G	1253	6655#																		
T39	027720	G	1254	6715#																		
T4	014740	G	1219	3626#																		
T40	030162	G	1255	6776#																		
T41	030424	G	1256	6836#																		
T42	030522	G	1257	6869#																		
T422C	005732		2478	2489#																		
T43	030620	G	1258	6899#																		
T5	015272	G	1220	3701#																		
T6	015624	G	1221	3776#																		
T7	016156	G	1222	3851#																		
T8	016510	G	1223	3927#																		
T9	016724	G	1224	3983#																		
T9BG	017340		4168#	4270																		
UAM =	000200	G	1388#																			
VECTOR	032044		7144	7186#																		
V35TC	005702		2463	2477#																		
WAIT50	004172		1629	1672	2072#	2175	2214	2437	3335	4088	4492	4493	4494	4495	4546							
			4547	4669	4671	4726	4746	4747	4774	5167	5168	5486	5487	5859	5956							
			6495																			
WFPE	003012		1710#	4700	4753	4851	4878	5044	5107	5293	5592	5664	5710	5760	5820							
			5840	5917	5937	6204	6261	6329	6377													
WORDT	002370		1466#	2146*	2370*	2379*	2823	2835	3485*	3488*	3497	3508*	3509	3518	3532*							
			3533	3560*	3563*	3572	3583*	3584	3593	3607*	3608	3635*	3638*	3647	3658*							
			3659	3668	3682*	3683	3710*	3713*	3722	3733*	3734	3743	3757*	3758	3785*							
			3788*	3797	3808*	3809	3818	3832*	3833	3860*	3863*	3872	3883*	3884	3893							
			3907*	3908																		
WRD!	002726		1666#	2589	2600	4179	4477	4533	4606	4690	4763	5439	5702	6193	6250							
			6318	6365	7009																	
WRDIE	003010		1669	1671	1684#																	
WRDI1	002732		1667#	1674	1771																	
WRDO	002642		1623#	2002	2242	2360	2366	4615														
WRDOE	002724		1626	1628	1641#																	
WRDO1	002646		1624#	1631	1711	1806																
WRIBYT	002500		1514#																			
XX	021732		4801	4803#																		
XYZTC	005642		2447	2462#																		
XSALWA=	000000		1084#																			
XSFALS=	000040		1084#																			
XSOFFS=	000400		1084#																			
XSTRUE=	000020		1084#																			
SBDDAT	002336		1452#	1715*	1716*	1717	1731*	1732	1809*	1810*	1811	1823*	1824	1850*	1851*							
			1852	1879*	1880	1908*	1909*	1910	1939*	1940	1966*	1967*	1968	2005*	2006*							
			2007	2009	2012*	2013	2041*	2043	2637*	2639	2691	2703	2745	2759	2788							
			2873	4211*	4212	4249*	4250*	4366	4394	4560*	4561*	4563	4619*	4620*	4621							
			4623	4632*	4633	4678*	4679	5553*	5555	6575*	6576*	6578										
SGNDAT	002334		1451#	1720*	1734*	1811	1822*	1824	1852	1880	1910	1940	1968	2011*	2013							
			2040*	2043	2227*	2230*	2231	2567*	2571*	2616*	2621*	2625*	2629*	2638*	2639							
			2655*	2692	2704	2746	2760	2789	2874	4208*	4212	4246*	4364*	4394	4562*							
			4563	4575*	4677*	4679	4949*	4960*	4961*	4971*	5220*	5232*	5233*	5315*	5325*							
			5366*	5376*	5420*	5428*	5457*	5463*	5468*	5474*	5479*	5541*	5554*	5555	5614*							
			5625*	6032*	6046*	6059*	6099*	6112*	6125*	6549*	6562*	6574*	6577*	6578	6955*							
			6960*	6974*	6993*	6998*	7028*	7050*	7059*	7066*	7071*	7080*	7085*	7103*	7108*							

GPRML	1#	1084#													
HEADER	1#	1084#	1116												
INLOOP	1#	1084#													
IOSETU	1#	1084#													
IOSTAR	1#	1084#													
KT11	1#	1084#													
LASTAD	1#	1084#	7329												
MANUAL	1#	1084#													
MEMORY	1#	1084#													
MSBYTE	1#	1084#	1116#	1122	1123	1124									
MSCHEC	1#	1084#	3228#	4407#	4685#	6592#									
MSCNTO	1#	1084#	7134#	7139#	7143#	7147#	7152#	7157#	7162#						
MSCOUN	1#	1084#	2307#	2314#	2501#	2512#	2518#	2667#	2680#	2691#	2703#	2717#	2729#	2743#	2759#
	2772#	2783#	2788#	2801#	2806#	2812#	2822#	2835#	2850#	2866#	2873#	2889#	3456#	3522#	3597#
	3672#	3747#	3822#	3897#	3960#	4016#									
MSDATA	1#	1084#	1116#	1125	1127	1129	1131	1133	1135	1137	1139	1141	1143	1145	1147
	1149	1151	1153	1155#	1157	1159	1162	1165	1167	1169	1171	1173	1175	1177	1179
	1181	1183	1185	1187	1189	1191	1193	1195	1197	1199	1565#	1571#			
MSDECR	1#	1084#	1206#	1294#	1312#	2673#	2687#	2699#	2711#	2723#	2735#	2753#	2768#	2779#	2797#
	2818#	2831#	2844#	2858#	2883#	2895#	3232#	3327#	3345#	3358#	3375#	3391#	3442#	3462#	3540#
	3615#	3690#	3765#	3840#	3915#	3972#	4027#	4141#	4414#	4424#	4434#	4441#	4446#	4509#	4579#
	4641#	4710#	4782#	4881#	4991#	5053#	5116#	5178#	5248#	5338#	5380#	5382#	5493#	5573#	5723#
	5769#	5868#	5965#	6131#	6169#	6212#	6269#	6338#	6385#	6433#	6474#	6632#	6697#	6757#	6819#
	6855#	6887#	7110#	7169#	7263#										
MSDEFA	1#	1084#	7134#	7139#	7143#	7147#	7152#	7157#	7162#						
MSENDE	1#	1084#	1294#	1312#	2673#	2687#	2699#	2711#	2723#	2735#	2753#	2768#	2779#	2797#	2818#
	2831#	2844#	2858#	2883#	2895#	3232#	3327#	3345#	3358#	3375#	3391#	3442#	3462#	3540#	3615#
	3690#	3765#	3840#	3915#	3972#	4027#	4141#	4414#	4424#	4434#	4441#	4446#	4509#	4579#	4641#
	4710#	4782#	4881#	4991#	5053#	5116#	5178#	5248#	5338#	5380#	5382#	5493#	5573#	5723#	5769#
	5848#	5965#	6131#	6169#	6212#	6269#	6338#	6385#	6433#	6474#	6632#	6697#	6757#	6819#	6855#
	6887#	7110#	7169#	7263#											
MSERRI	1#	1084#	1635#	1678#	1725#	1739#	1815#	1827#	1856#	1886#	1915#	1943#	1972#	2016#	2029#
	2045#	2052#	2233#	2277#	2388#	2642#	3450#	3501#	3511#	3535#	3576#	3586#	3610#	3651#	3661#
	3685#	3726#	3736#	3760#	3801#	3811#	3835#	3876#	3886#	3910#	3952#	4009#	4096#	4107#	4117#
	4217#	4255#	4381#	4398#	4420#	4430#	4501#	4552#	4566#	4626#	4636#	4681#	4777#	5172#	5559#
	5863#	5960#	6581#												
MSESCA	1#	1084#	3935#	3936	3944#	3945	3957#	3958	3992#	3993	4001#	4002	4013#	4014	4100#
	4101	4113#	4114	4121#	4122	4130#	4131	4139#	4140	4174#	4175	4182#	4183	4197#	4198
	4221#	4222	4237#	4238	4259#	4260	4303#	4304	4339#	4340	4373#	4374	4386#	4387	4402#
	4403	4466#	4467	4480#	4481	4526#	4527	4536#	4537	4557#	4558	4571#	4572	4601#	4602
	4609#	4610	4617#	4618	4630#	4631	4664#	4665	4693#	4694	4703#	4704	4729#	4730	4756#
	4757	4766#	4767	4812#	4813	4825#	4826	4842#	4843	4854#	4855	4869#	4870	4901#	4902
	4915#	4916	4930#	4931	4944#	4945	4954#	4955	4965#	4966	4975#	4976	5010#	5011	5024#
	5025	5037#	5038	5047#	5048	5072#	5073	5086#	5087	5099#	5100	5110#	5111	5134#	5135
	5148#	5149	5161#	5162	5203#	5204	5214#	5215	5225#	5226	5237#	5238	5261#	5262	5273#
	5274	5287#	5288	5296#	5297	5309#	5310	5320#	5321	5329#	5330	5349#	5350	5360#	5361
	5370#	5371	5400#	5401	5410#	5411	5418#	5419	5426#	5427	5431#	5432	5442#	5443	5455#
	5456	5461#	5462	5466#	5467	5471#	5472	5477#	5478	5483#	5484	5508#	5509	5522#	5523
	5535#	5536	5546#	5547	5572#	5573	5585#	5586	5595#	5596	5608#	5609	5619#	5620	5630#
	5631	5644#	5645	5657#	5658	5667#	5668	5695#	5696	5705#	5706	5713#	5714	5742#	5743
	5753#	5754	5763#	5764	5789#	5790	5803#	5804	5813#	5814	5823#	5824	5833#	5834	5843#
	5844	5853#	5854	5886#	5887	5900#	5901	5910#	5911	5920#	5921	5930#	5931	5940#	5941
	5950#	5951	5985#	5986	5999#	6000	6013#	6014	6026#	6027	6040#	6041	6053#	6054	6066#
	6067	6080#	6081	6093#	6094	6106#	6107	6119#	6120	6159#	6160	6167#	6168	6186#	6187
	6196#	6197	6207#	6208	6229#	6230	6243#	6244	6253#	6254	6264#	6265	6290#	6291	6301#
	6302	6311#	6312	6321#	6322	6332#	6333	6357#	6358	6368#	6369	6380#	6381	6418#	6419

MSESCS	6427#	6428	6458#	6459	6468#	6469	6498#	6499	6517#	6518	6530#	6531	6543#	6544	6556#
	6557	6569#	6570	6585#	6586	6693#	6694	6753#	6754	6815#	6816	6848#	6849	6880#	6881
	6904#	6905	6914#	6915	6924#	6925	6933#	6934	6942#	6943	6950#	6951	6958#	6959	6963#
	6964	6972#	6973	6977#	6978	6991#	6992	7012#	7013	7026#	7027	7048#	7049	7053#	7054
	7069#	7070	7074#	7075	7083#	7084	7089#	7090	7101#	7102	7106#	7107			
	1#	1084#	3935#	3944#	3957#	3992#	4001#	4013#	4100#	4113#	4121#	4130#	4139#	4174#	4182#
	4197#	4221#	4237#	4259#	4303#	4339#	4373#	4386#	4402#	4466#	4480#	4526#	4536#	4557#	4571#
	4601#	4609#	4617#	4630#	4664#	4693#	4703#	4729#	4756#	4766#	4812#	4825#	4842#	4854#	4869#
	4901#	4915#	4930#	4944#	4954#	4965#	4975#	5010#	5024#	5037#	5047#	5072#	5086#	5099#	5110#
	5134#	5148#	5161#	5203#	5214#	5225#	5237#	5261#	5273#	5287#	5296#	5309#	5320#	5329#	5349#
	5360#	5370#	5400#	5410#	5418#	5426#	5431#	5442#	5455#	5461#	5466#	5471#	5477#	5483#	5508#
	5522#	5535#	5546#	5572#	5585#	5595#	5608#	5619#	5630#	5644#	5657#	5667#	5695#	5705#	5713#
	5742#	5753#	5763#	5789#	5803#	5813#	5823#	5833#	5843#	5853#	5886#	5900#	5910#	5920#	5930#
	5940#	5950#	5985#	5999#	6013#	6026#	6040#	6053#	6066#	6080#	6093#	6106#	6119#	6159#	6167#
	6186#	6196#	6207#	6229#	6243#	6253#	6264#	6290#	6301#	6311#	6321#	6332#	6357#	6368#	6380#
	6418#	6427#	6458#	6468#	6498#	6517#	6530#	6543#	6556#	6569#	6585#	6693#	6753#	6815#	6848#
	6880#	6904#	6914#	6924#	6933#	6942#	6950#	6958#	6963#	6972#	6977#	6991#	7012#	7026#	7048#
	7053#	7069#	7074#	7083#	7089#	7101#	7106#								
MSEXCP	1#	1084#	7134#	7139#	7143#	7147#	7152#	7157#	7162#						
MSEXIT	1#	1084#	3228#	4407#	4408	4685#	4686	6592#	6593						
MSEXSE	1#	1084#	3228#	4407#	4685#	6592#									
MSEXTJ	1#	1084#	3228#	3229	4407#	4685#	6592#								
MSGEN	1#	1084#	1092#	1116#	1125#	1127#	1129#	1131#	1133#	1135#	1137#	1139#	1141#	1143#	1145#
	1147#	1149#	1151#	1153#	1155#	1157#	1159#	1162#	1165#	1167#	1169#	1171#	1173#	1175#	1177#
	1179#	1181#	1183#	1185#	1187#	1189#	1191#	1193#	1195#	1197#	1199#	1202#	1215#	1274#	1275#
	1294#	1308#	1309#	1312#	1565#	1571#	2666#	2673#	2679#	2687#	2690#	2699#	2702#	2711#	2716#
	2723#	2728#	2735#	2742#	2753#	2758#	2768#	2771#	2779#	2782#	2797#	2800#	2818#	2821#	2831#
	2834#	2844#	2849#	2858#	2865#	2883#	2888#	2895#	3225#	3232#	3246#	3327#	3331#	3345#	3355#
	3358#	3372#	3375#	3390#	3391#	3419#	3442#	3447#	3462#	3476#	3540#	3551#	3615#	3626#	3690#
	3701#	3765#	3776#	3840#	3851#	3915#	3927#	3972#	3983#	4027#	4083#	4141#	4162#	4410#	4414#
	4417#	4424#	4427#	4434#	4437#	4441#	4446#	4461#	4509#	4521#	4579#	4591#	4641#	4658#	4710#
	4723#	4782#	4798#	4881#	4894#	4991#	5005#	5053#	5067#	5116#	5129#	5178#	5189#	5191#	5248#
	5250#	5338#	5340#	5380#	5382#	5396#	5493#	5503#	5673#	5689#	5723#	5737#	5769#	5784#	5868#
	5881#	5965#	5980#	6131#	6148#	6169#	6181#	6212#	6224#	6269#	6285#	6338#	6352#	6385#	6405#
	6433#	6447#	6474#	6488#	6632#	6655#	6697#	6715#	6757#	6776#	6819#	6836#	6855#	6869#	6887#
	6899#	7110#	7133#	7170#	7261#	7264#	7332#								
MSGENB	1#	1084#													
MSGETS	1#	1084#	1206#	1294#	1312#	2673#	2687#	2699#	2711#	2723#	2735#	2753#	2768#	2779#	2797#
	2818#	2831#	2844#	2858#	2883#	2895#	3232#	3327#	3345#	3358#	3375#	3391#	3442#	3462#	3540#
	3615#	3690#	3765#	3840#	3915#	3972#	4027#	4141#	4414#	4424#	4434#	4441#	4446#	4509#	4579#
	4641#	4710#	4782#	4881#	4991#	5053#	5116#	5178#	5248#	5338#	5380#	5382#	5493#	5673#	5723#
	5769#	5868#	5965#	6131#	6169#	6212#	6269#	6338#	6385#	6433#	6474#	6632#	6697#	6757#	6819#
	6855#	6887#	7110#	7169#	7263#										
MSGETT	1#	1084#	3228#	3935#	3944#	3957#	3992#	4001#	4013#	4100#	4113#	4121#	4130#	4139#	4174#
	4182#	4197#	4221#	4237#	4259#	4303#	4339#	4373#	4386#	4402#	4407#	4466#	4480#	4526#	4536#
	4557#	4571#	4601#	4609#	4617#	4630#	4664#	4685#	4693#	4703#	4729#	4756#	4766#	4812#	4825#
	4842#	4854#	4869#	4901#	4915#	4930#	4944#	4954#	4965#	4975#	5010#	5024#	5037#	5047#	5072#
	5086#	5099#	5110#	5134#	5148#	5161#	5203#	5214#	5225#	5237#	5261#	5273#	5287#	5296#	5309#
	5320#	5329#	5349#	5360#	5370#	5400#	5410#	5418#	5426#	5431#	5442#	5455#	5461#	5466#	5471#
	5477#	5483#	5508#	5522#	5535#	5546#	5572#	5585#	5595#	5608#	5619#	5630#	5644#	5657#	5667#
	5695#	5705#	5713#	5742#	5753#	5763#	5789#	5803#	5813#	5823#	5833#	5843#	5853#	5886#	5900#
	5910#	5920#	5930#	5940#	5950#	5985#	5999#	6013#	6026#	6040#	6053#	6066#	6080#	6093#	6106#
	6119#	6159#	6167#	6186#	6196#	6207#	6229#	6243#	6253#	6264#	6290#	6301#	6311#	6321#	6332#
	6357#	6368#	6380#	6418#	6427#	6458#	6468#	6498#	6517#	6530#	6543#	6556#	6569#	6585#	6592#
	6693#	6753#	6815#	6848#	6880#	6904#	6914#	6924#	6933#	6942#	6950#	6958#	6963#	6972#	6977#
	6991#	7012#	7026#	7048#	7053#	7069#	7074#	7083#	7089#	7101#	7106#				

MSGNGB	1#	1084#	1092#	1116#	1125#	1127#	1129#	1131#	1133#	1135#	1137#	1139#	1141#	1143#	1145#
	1147#	1149#	1151#	1153#	1155#	1157#	1159#	1162#	1165#	1167#	1169#	1171#	1173#	1175#	1177#
	1179#	1181#	1183#	1185#	1187#	1189#	1191#	1193#	1195#	1197#	1199#	1202#	1214#	1215	1273#
	1274	1275	1307#	1308	1309	1565#	1571#	2666#	2679#	2690#	2702#	2716#	2728#	2742#	2758#
	2771#	2782#	2800#	2821#	2834#	2849#	2865#	2888#	3225#	3246#	3331#	3355#	3372#	3390#	3447#
	4410#	4417#	4427#	4437#	7132#	7133	7260#	7261	7329#	7332					
MSGNIN	1#	1084#	1116#	1117	1118	1119	1120	1121	1122#	1123#	1124#	1125#	1126	1127#	1128
	1129#	1130	1131#	1132	1133#	1134	1135#	1136	1137#	1138	1139#	1140	1141#	1142	1143#
	1144	1145#	1146	1147#	1148	1149#	1150	1151#	1152	1153#	1154	1155#	1156	1157#	1158
	1159#	1160	1161	1162#	1163	1164#	1165#	1166	1167#	1168	1169#	1170	1171#	1172	1173#
	1174	1175#	1176	1177#	1178	1179#	1180	1181#	1182	1183#	1184	1185#	1186	1187#	1188
	1189#	1190	1191#	1192	1193#	1194	1195#	1196	1197#	1198	1199#	1200	1214#	1216#	1217#
	1218#	1219#	1220#	1221#	1222#	1223#	1224#	1225#	1226#	1227#	1228#	1229#	1230#	1231#	1232#
	1233#	1234#	1235#	1236#	1237#	1238#	1239#	1240#	1241#	1242#	1243#	1244#	1245#	1246#	1247#
	1248#	1249#	1250#	1251#	1252#	1253#	1254#	1255#	1256#	1257#	1258#	1273#	1307#	1565#	1566
	1569	1571#	1572	1578	1635#	1636#	1637#	1638#	1678#	1679#	1680#	1681#	1725#	1726#	1727#
	1728#	1739#	1740#	1741#	1742#	1815#	1816#	1817#	1818#	1827#	1828#	1829#	1830#	1856#	1857#
	1858#	1859#	1886#	1887#	1888#	1889#	1915#	1916#	1917#	1918#	1943#	1944#	1945#	1946#	1972#
	1973#	1974#	1975#	2016#	2017#	2018#	2019#	2029#	2030#	2031#	2032#	2045#	2046#	2047#	2048#
	2052#	2053#	2054#	2055#	2233#	2234#	2235#	2236#	2277#	2278#	2279#	2280#	2285#	2307#	2308#
	2309#	2310#	2311	2312#	2313	2314#	2315#	2316#	2317#	2318#	2319#	2320	2321#	2322	2388#
	2389#	2390#	2391#	2501#	2502#	2503	2504#	2505	2512#	2513#	2514	2515#	2516	2518#	2519#
	2520#	2521	2522#	2523	2642#	2643#	2644#	2645#	2667#	2668#	2669#	2670	2671#	2672	2674#
	2680#	2681#	2682#	2683	2684#	2685	2688#	2691#	2692#	2693#	2694#	2695	2696#	2697	2700#
	2703#	2704#	2705#	2706#	2707	2708#	2709	2712#	2717#	2718#	2719	2720#	2721	2724#	2729#
	2730#	2731	2732#	2733	2736#	2743#	2744#	2745#	2746#	2747#	2748#	2749	2750#	2751	2754#
	2759#	2760#	2761#	2762#	2763#	2764	2765#	2766	2769#	2772#	2773#	2774#	2775	2776#	2777
	2780#	2783#	2784#	2785	2786#	2787	2788#	2789#	2790#	2791#	2792#	2793	2794#	2795	2798#
	2801#	2802#	2803	2804#	2805	2806#	2807#	2808#	2809	2810#	2811	2812#	2813#	2814	2815#
	2816	2819#	2822#	2823#	2824#	2825#	2826#	2827	2828#	2829	2832#	2835#	2836#	2837#	2838#
	2839#	2840	2841#	2842	2845#	2850#	2851#	2852#	2853#	2854	2855#	2856	2859#	2866#	2867#
	2868#	2869#	2870	2871#	2872	2873#	2874#	2875#	2876#	2877#	2878#	2879	2880#	2881	2884#
	2889#	2890#	2891	2892#	2893	2896#	3228#	3229#	3233#	3260#	3261#	3262#	3264#	3265#	3266#
	3268#	3269#	3270#	3272#	3273#	3274#	3291#	3292#	3293#	3294#	3328#	3340#	3341#	3344#	3346#
	3356#	3359#	3374#	3376#	3392#	3420#	3421#	3422#	3423#	3424#	3425	3437#	3439#	3440#	3443#
	3450#	3451#	3452#	3453#	3456#	3457#	3458#	3459	3460#	3461	3462#	3463	3501#	3502#	3503#
	3504#	3511#	3512#	3513#	3514#	3522#	3523#	3524#	3525#	3526#	3527	3528#	3529	3535#	3536#
	3537#	3538#	3541#	3576#	3577#	3578#	3579#	3586#	3587#	3588#	3589#	3597#	3598#	3599#	3600#
	3601#	3602	3603#	3604	3610#	3611#	3612#	3613#	3616#	3651#	3652#	3653#	3654#	3661#	3662#
	3663#	3664#	3672#	3673#	3674#	3675#	3676#	3677	3678#	3679	3685#	3686#	3687#	3688#	3691#
	3726#	3727#	3728#	3729#	3736#	3737#	3738#	3739#	3747#	3748#	3749#	3750#	3751#	3752	3753#
	3754	3760#	3761#	3762#	3763#	3766#	3801#	3802#	3803#	3804#	3811#	3812#	3813#	3814#	3822#
	3823#	3824#	3825#	3826#	3827	3828#	3829	3835#	3836#	3837#	3838#	3841#	3876#	3877#	3878#
	3879#	3886#	3887#	3888#	3889#	3897#	3898#	3899#	3900#	3901#	3902	3903#	3904	3910#	3911#
	3912#	3913#	3916#	3935#	3936#	3944#	3945#	3952#	3953#	3954#	3955#	3957#	3958#	3960#	3961#
	3962#	3963#	3964#	3965	3966#	3967	3973#	3992#	3993#	4001#	4002#	4009#	4010#	4011#	4012#
	4013#	4014#	4016#	4017#	4018#	4019#	4020#	4021	4022#	4023	4028#	4096#	4097#	4098#	4099#
	4100#	4101#	4107#	4108#	4109#	4110#	4113#	4114#	4117#	4118#	4119#	4120#	4121#	4122#	4130#
	4131#	4139#	4140#	4142#	4174#	4175#	4182#	4183#	4197#	4198#	4217#	4218#	4219#	4220#	4221#
	4222#	4237#	4238#	4255#	4256#	4257#	4258#	4259#	4260#	4278#	4279#	4280#	4281#	4282#	4283
	4286#	4287#	4288#	4289#	4290#	4291	4296#	4297#	4303#	4304#	4317#	4318#	4319#	4320#	4321#
	4322	4323#	4324#	4325#	4326#	4327#	4328	4331#	4332#	4339#	4340#	4352#	4353#	4354#	4355#
	4373#	4374#	4381#	4382#	4383#	4384#	4386#	4387#	4398#	4399#	4400#	4401#	4402#	4403#	4407#
	4408#	4411#	4412#	4414#	4415	4418#	4419#	4420#	4421#	4422#	4423#	4424#	4425	4428#	4429#
	4430#	4431#	4432#	4433#	4434#	4435	4438#	4439#	4441#	4442	4447#	4466#	4467#	4480#	4481#
	4501#	4502#	4503#	4504#	4510#	4526#	4527#	4536#	4537#	4552#	4553#	4554#	4555#	4557#	4558#

CZDMTAD DMP/V-11 P CTNL TST #1
CZDMTA.REL 12-AUG-80 15:52

MACY11 30A(1052) 12-AUG-80 15:53 PAGE 203
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0200

4566#	4567#	4568#	4569#	4571#	4572#	4580#	4601#	4602#	4609#	4610#	4617#	4618#	4626#	4627#	
4628#	4629#	4630#	4631#	4636#	4637#	4638#	4639#	4642#	4664#	4665#	4681#	4682#	4683#	4684#	
4685#	4686#	4693#	4694#	4703#	4704#	4711#	4729#	4730#	4756#	4757#	4766#	4767#	4777#	4778#	
4779#	4780#	4783#	4812#	4813#	4825#	4826#	4842#	4843#	4854#	4855#	4869#	4870#	4882#	4901#	
4902#	4915#	4916#	4930#	4931#	4944#	4945#	4954#	4955#	4965#	4966#	4975#	4976#	4992#	5010#	
5011#	5024#	5025#	5037#	5038#	5047#	5048#	5054#	5072#	5073#	5086#	5087#	5099#	5100#	5110#	
5111#	5117#	5134#	5135#	5148#	5149#	5161#	5162#	5172#	5173#	5174#	5175#	5179#	5192#	5203#	
5204#	5214#	5215#	5225#	5226#	5237#	5238#	5249#	5251#	5261#	5262#	5273#	5274#	5287#	5288#	
5296#	5297#	5309#	5310#	5320#	5321#	5329#	5330#	5339#	5341#	534	5350#	5360#	5361#	5370#	
5371#	5381#	5383#	5400#	5401#	5410#	5411#	5418#	5419#	5426#	5427#	5431#	5432#	5442#	5443#	
5455#	5456#	5461#	5462#	5466#	5467#	5471#	5472#	5477#	5478#	5483#	5484#	5494#	5508#	5509#	
5522#	5523#	5535#	5536#	5546#	5547#	5559#	5560#	5561#	5562#	5572#	5573#	5585#	5586#	5595#	
5596#	5608#	5609#	5619#	5620#	5630#	5631#	5644#	5645#	5657#	5658#	5667#	5668#	5674#	5695#	
5696#	5705#	5706#	5713#	5714#	5724#	5742#	5743#	5753#	5754#	5763#	5764#	5770#	5789#	5790#	
5803#	5804#	5813#	5814#	5823#	5824#	5833#	5834#	5843#	5844#	5853#	5854#	5863#	5864#	5865#	
5866#	5869#	5886#	5887#	5900#	5901#	5910#	5911#	5920#	5921#	5930#	5931#	5940#	5941#	5950#	
5951#	5960#	5961#	5962#	5963#	5966#	5985#	5986#	5999#	6000#	6013#	6014#	6026#	6027#	6040#	
6041#	6053#	6054#	6066#	6067#	6080#	6081#	6093#	6094#	6106#	6107#	6119#	6120#	6132#	6159#	
6160#	6167#	6168#	6170#	6186#	6187#	6196#	6197#	6207#	6208#	6213#	6229#	6230#	6243#	6244#	
6253#	6254#	6264#	6265#	6270#	6290#	6291#	6301#	6302#	6311#	6312#	6321#	6322#	6332#	6333#	
6339#	6357#	6358#	6368#	6369#	6380#	6381#	6386#	6418#	6419#	6427#	6428#	6434#	6458#	6459#	
6468#	6469#	6475#	6498#	6499#	6517#	6518#	6530#	6531#	6543#	6544#	6556#	6557#	6569#	6570#	
6581#	6582#	6583#	6584#	6585#	6586#	6592#	6593#	6633#	6667#	6668#	6693#	6694#	6698#	6727#	
6728#	6753#	6754#	6758#	6789#	6790#	6815#	6816#	6820#	6848#	6849#	6856#	6880#	6881#	6888#	
6904#	6905#	6914#	6915#	6924#	6925#	6933#	6934#	6942#	6943#	6950#	6951#	6958#	6959#	6963#	
6964#	6972#	6973#	6977#	6978#	6991#	6992#	7012#	7013#	7026#	7027#	7048#	7049#	7053#	7054#	
7069#	7070#	7074#	7075#	7083#	7084#	7089#	7090#	7101#	7102#	7106#	7107#	7111#	7132#	7134#	
7135	7136	7137	7138	7139#	7140	7141	7142	7143#	7144	7145	7146	7147#	7148	7149	
7150	7151	7152#	7153	7154	7155	7156	7157#	7158	7159	7160	7161	7162#	7163	7164	
7165	7166	7169#	7260#	7263#	7329#	7330#	7331#								
MSGNLS	1#	1084#													
MSGNSU	1#	1084#	5191#	5250#	5340#										
MSGNTA	1#	1084#	1294#	1312#	2673#	2687#	2699#	2711#	2723#	2735#	2753#	2768#	2779#	2797#	2818#
	2831#	2844#	2858#	2883#	2895#	3232#	3327#	3345#	3358#	3375#	3391#	3442#	3462#	3540#	3615#
	3690#	3765#	3840#	3915#	3972#	4027#	4141#	4414#	4424#	4434#	4441#	4446#	4509#	4579#	4641#
	4710#	4782#	4881#	4991#	5053#	5116#	5178#	5248#	5338#	5380#	5382#	5493#	5673#	5723#	5769#
	5868#	5965#	6131#	6169#	6212#	6269#	6338#	6385#	6433#	6474#	6632#	6697#	6757#	6819#	6855#
	6887#	7110#	7169#	7170	7263#	7264									
MSGNTE	1#	1084#	3419#	3476#	3551#	3626#	3701#	3776#	3851#	3927#	3983#	4083#	4162#	4461#	4521#
	4591#	4658#	4723#	4798#	4894#	5005#	5067#	5129#	5189#	5396#	5503#	5689#	5737#	5784#	5881#
	5980#	6148#	6181#	6224#	6285#	6352#	6405#	6447#	6488#	6655#	6715#	6776#	6836#	6869#	6899#
MSHAPT	1#	1084#	1116#												
MSHNAP	1#	1084#	1116#	1155											
MSINCR	1#	1084#	1092#	1202#	1273#	1307#	1635#	1678#	1725#	1739#	1815#	1827#	1856#	1886#	1915#
	1943#	1972#	2016#	2029#	2045#	2052#	2233#	2277#	2285#	2312#	2321#	2388#	2504#	2515#	2522#
	2642#	2666#	2671#	2674#	2679#	2684#	2688#	2690#	2696#	2700#	2702#	2708#	2712#	2716#	2720#
	2724#	2728#	2732#	2736#	2742#	2750#	2754#	2758#	2765#	2769#	2771#	2776#	2780#	2782#	2786#
	2794#	2798#	2800#	2804#	2810#	2815#	2819#	2821#	2828#	2832#	2834#	2841#	2845#	2849#	2855#
	2859#	2865#	2871#	2880#	2884#	2888#	2892#	2896#	3225#	3233#	3246#	3261#	3265#	3269#	3273#
	3292#	3328#	3331#	3341#	3344#	3346#	3355#	3356#	3359#	3372#	3374#	3376#	3390#	3392#	3419#
	3420#	3424#	3437#	3440#	3443#	3447#	3450#	3460#	3476#	3477#	3501#	3511#	3528#	3535#	3541#
	3551#	3552#	3576#	3586#	3603#	3610#	3616#	3626#	3627#	3651#	3661#	3678#	3685#	3691#	3701#
	3702#	3726#	3736#	3753#	3760#	3766#	3776#	3777#	3801#	3811#	3828#	3835#	3841#	3851#	3852#
	3876#	3886#	3903#	3910#	3916#	3927#	3928#	3935#	3944#	3952#	3957#	3966#	3973#	3983#	3984#
	3992#	4001#	4009#	4013#	4022#	4028#	4083#	4084#	4096#	4100#	4107#	4113#	4117#	4121#	4130#
	4139#	4142#	4162#	4163#	4174#	4182#	4197#	4217#	4221#	4237#	4255#	4259#	4282#	4290#	4297#

	4303#	4321#	4327#	4332#	4339#	4353#	4355#	4373#	4381#	4386#	4398#	4402#	4407#	4410#	4412#
	4417#	4419#	4420#	4427#	4429#	4430#	4437#	4439#	4447#	4461#	4462#	4466#	4480#	4501#	4510#
	4521#	4522#	4526#	4536#	4552#	4557#	4566#	4571#	4580#	4591#	4592#	4601#	4609#	4617#	4626#
	4630#	4636#	4642#	4658#	4659#	4664#	4681#	4685#	4693#	4703#	4711#	4723#	4724#	4729#	4756#
	4766#	4777#	4783#	4798#	4799#	4812#	4825#	4842#	4854#	4869#	4882#	4894#	4895#	4901#	4915#
	4930#	4944#	4954#	4965#	4975#	4992#	5005#	5006#	5010#	5024#	5037#	5047#	5054#	5067#	5068#
	5072#	5086#	5099#	5110#	5117#	5129#	5130#	5134#	5148#	5161#	5172#	5179#	5189#	5190#	5191#
	5192#	5203#	5214#	5225#	5237#	5249#	5250#	5251#	5261#	5273#	5287#	5296#	5309#	5320#	5329#
	5339#	5340#	5341#	5349#	5360#	5370#	5381#	5383#	5396#	5397#	5400#	5410#	5418#	5426#	5431#
	5442#	5455#	5461#	5466#	5471#	5477#	5483#	5494#	5503#	5504#	5508#	5522#	5535#	5546#	5559#
	5572#	5585#	5595#	5608#	5619#	5630#	5644#	5657#	5667#	5674#	5689#	5690#	5695#	5705#	5713#
	5724#	5737#	5738#	5742#	5753#	5763#	5770#	5784#	5785#	5789#	5803#	5813#	5823#	5833#	5843#
	5853#	5863#	5869#	5881#	5882#	5886#	5900#	5910#	5920#	5930#	5940#	5950#	5960#	5966#	5980#
	5981#	5985#	5999#	6013#	6026#	6040#	6053#	6066#	6080#	6093#	6106#	6119#	6132#	6148#	6149#
	6159#	6167#	6170#	6181#	6182#	6186#	6196#	6207#	6213#	6224#	6225#	6229#	6243#	6253#	6264#
	6270#	6285#	6286#	6290#	6301#	6311#	6321#	6332#	6339#	6352#	6353#	6357#	6368#	6380#	6386#
	6405#	6406#	6418#	6427#	6434#	6447#	6448#	6458#	6468#	6475#	6488#	6489#	6498#	6517#	6530#
	6543#	6556#	6569#	6581#	6585#	6592#	6633#	6655#	6656#	6668#	6693#	6698#	6715#	6716#	6728#
	6753#	6758#	6776#	6777#	6790#	6815#	6820#	6836#	6837#	6848#	6856#	6869#	6870#	6880#	6888#
	6899#	6900#	6904#	6914#	6924#	6933#	6942#	6950#	6958#	6963#	6972#	6977#	6991#	7012#	7026#
	7048#	7053#	7069#	7074#	7083#	7089#	7101#	7106#	7111#	7132#	7260#				
MSIOSE	1#	1084#													
MSLDRO	1#	1084#	3260#	3264#	3268#	3272#	3291#	3340#	3439#	4296#	4331#	4352#	4354#	4411#	4418#
	4428#	4438#	6667#	6727#	6789#										
MSMASK	1#	1084#													
MSMCHI	1#	1084#													
MSMCLO	1#	1084#													
MSMSK1	1#	1084#													
MSPOP	1#	1084#	1206#	1294#	1312#	2673#	2687#	2699#	2711#	2723#	2735#	2753#	2768#	2779#	2797#
	2818#	2831#	2844#	2858#	2883#	2895#	3232#	3327#	3345#	3358#	3375#	3391#	3442#	3462#	3540#
	3615#	3690#	3765#	3840#	3915#	3972#	4027#	4141#	4414#	4424#	4434#	4441#	4446#	4509#	4579#
	4641#	4710#	4782#	4881#	4991#	5053#	5116#	5178#	5248#	5338#	5380#	5382#	5493#	5673#	5723#
	5769#	5868#	5965#	6131#	6169#	6212#	6269#	6338#	6385#	6433#	6474#	6632#	6697#	6757#	6819#
	6855#	6887#	7110#	7169#	7263#										
MSPRIN	1#	1084#	2307#	2314#	2501#	2512#	2518#	2667#	2680#	2691#	2703#	2717#	2729#	2743#	2759#
	2772#	2783#	2788#	2801#	2806#	2812#	2822#	2835#	2850#	2866#	2873#	2889#	3456#	3522#	3597#
	3672#	3747#	3822#	3897#	3960#	4016#									
MSPUSH	1#	1084#	1092#	1202#	1273#	1307#	2666#	2679#	2690#	2702#	2716#	2728#	2742#	2758#	2771#
	2782#	2800#	2821#	2834#	2849#	2865#	2888#	3225#	3246#	3331#	3355#	3372#	3390#	3419#	3420#
	3447#	3476#	3477#	3551#	3552	3626#	3627	3701#	3702	3776#	3777	3851#	3852	3927#	3928
	3983#	3984	4083#	4084	4162#	4163	4410#	4417#	4427#	4437#	4461#	4462	4521#	4522	4591#
	4592	4658#	4659	4723#	4724	4798#	4799	4894#	4895	5005#	5006	5067#	5068	5129#	5130
	5189#	5190	5191#	5192	5250#	5251	5340#	5341	5396#	5397	5503#	5504	5689#	5690	5737#
	5738	5784#	5785	5881#	5882	5980#	5981	6148#	6149	6181#	6182	6224#	6225	6285#	6286
	6352#	6353	6405#	6406	6447#	6448	6488#	6489	6655#	6656	6715#	6716	6776#	6777	6836#
	6837	6869#	6870	6899#	6900	7132#	7260#								
MSPUT	1#	1084#	2307#	2314#	2501#	2512#	2518#	2667#	2680#	2691#	2703#	2717#	2729#	2743#	2759#
	2772#	2783#	2788#	2801#	2806#	2812#	2822#	2835#	2850#	2866#	2873#	2889#	3420#	3456#	3522#
	3597#	3672#	3747#	3822#	3897#	3960#	4016#	4278#	4286#	4317#	4323#				
MSPUT1	1#	1084#	2307#	2308	2309	2310	2314#	2315	2310	2317	2318	2319	2501#	2502	2512#
	2513	2518#	2519	2520	2667#	2668	2669	2680#	2681	2682	2691#	2692	2693	2694	2703#
	2704	2705	2706	2717#	2718	2729#	2730	2743#	2744	2745	2746	2747	2748	2759#	2760
	2761	2762	2763	2772#	2773	2774	2783#	2784	2788#	2789	2790	2791	2792	2801#	2802
	2806#	2807	2808	2812#	2813	2822#	2823	2824	2825	2826	2835#	2836	2837	2838	2839
	2850#	2851	2852	2853	2866#	2867	2868	2869	2873#	2874	2875	2876	2877	2878	2889#
	2890	3420#	3421	3422	3423	3456#	3457	3458	3522#	3523	3524	3525	3526	3597#	3598

	3599	3600	3601	3672#	3673	3674	3675	3676	3747#	3748	3749	3750	3751	3822#	3823
	3824	3825	3826	3897#	3898	3899	3900	3901	3960#	3961	3962	3963	3964	4016#	4017
	4018	4019	4020	4278#	4279	4280	4281	4286#	4287	4288	4289	4317#	4318	4319	4320
	4323#	4324	4325	4326											
MSRADI	1#	1084#	7134#	7139#	7143#	7147#	7152#	7157#	7162#						
MSRBRO	1#	1084#													
MSRNRO	1#	1084#	3291#	3293											
MSSETS	1#	1084#	1092#	1202#	1273#	1307#	2666#	2679#	2690#	2702#	2716#	2728#	2742#	2758#	2771#
	2782#	2800#	2821#	2834#	2849#	2865#	2888#	3225#	3246#	3331#	3355#	3372#	3390#	3420#	3447#
	3477#	3552#	3627#	3702#	3777#	3852#	3928#	3984#	4084#	4163#	4410#	4417#	4427#	4437#	4462#
	4522#	4592#	4659#	4724#	4799#	4895#	5006#	5068#	5130#	5190#	5192#	5251#	5341#	5397#	5504#
	5690#	5738#	5785#	5882#	5981#	6149#	6182#	6225#	6286#	6353#	6406#	6448#	6489#	6656#	6716#
	6777#	6837#	6870#	6900#	7132#	7260#									
MSSTAR	1#	1084#													
MS SVC	1#	1084#	1635	1678	1725	1739	1815	1827	1856	1886	1915	1943	1972	2016	2029
	2045	2052	2233	2277	2285#	2307#	2312	2314#	2321	2388	2501#	2504	2512#	2515	2518#
	2522	2642	2667#	2671	2673#	2674	2680#	2684	2687#	2688	2691#	2696	2699#	2700	2703#
	2708	2711#	2712	2717#	2720	2723#	2724	2729#	2732	2735#	2736	2743#	2750	2753#	2754
	2759#	2765	2768#	2769	2772#	2776	2779#	2780	2783#	2786	2788#	2794	2797#	2798	2801#
	2804	2806#	2810	2812#	2815	2818#	2819	2822#	2828	2831#	2832	2835#	2841	2844#	2845
	2850#	2855	2858#	2859	2866#	2871	2873#	2880	2883#	2884	2889#	2892	2895#	2896	3228#
	3232#	3233	3260#	3261	3264#	3265	3268#	3269	3272#	3273	3291#	3292	3327#	3328	3340#
	3341	3344#	3345#	3346	3356#	3358#	3359	3374#	3375#	3376	3391#	3392	3420#	3424	3437#
	3439#	3440	3442#	3443	3450	3456#	3460	3501	3511	3522#	3528	3535	3540#	3541	3576
	3586	3597#	3603	3610	3615#	3616	3651	3661	3672#	3678	3685	3690#	3691	3726	3736
	3747#	3753	3760	3765#	3766	3801	3811	3822#	3828	3835	3840#	3841	3876	3886	3897#
	3903	3910	3915#	3916	3935#	3944#	3952	3957#	3960#	3966	3972#	3973	3992#	4001#	4009
	4013#	4016#	4022	4027#	4028	4096	4100#	4107	4113#	4117	4121#	4130#	4139#	4141#	4142
	4174#	4182#	4197#	4217	4221#	4237#	4255	4259#	4278#	4282	4286#	4290	4296#	4297	4303#
	4317#	4321	4323#	4327	4331#	4332	4339#	4352#	4353	4354#	4355	4373#	4381	4386#	4398
	4402#	4407#	4411#	4412	4418#	4419	4420	4428#	4429	4430	4438#	4439	4446#	4447	4466#
	4480#	4501	4509#	4510	4526#	4536#	4552	4557#	4566	4571#	4579#	4580	4601#	4609#	4617#
	4626	4630#	4636	4641#	4642	4664#	4681	4685#	4693#	4703#	4710#	4711	4729#	4756#	4766#
	4777	4782#	4783	4812#	4825#	4842#	4854#	4869#	4881#	4882	4901#	4915#	4930#	4944#	4954#
	4965#	4975#	4991#	4992	5010#	5024#	5037#	5047#	5053#	5054	5072#	5086#	5099#	5110#	5116#
	5117	5134#	5148#	5161#	5172	5178#	5179	5191#	5192	5203#	5214#	5225#	5237#	5248#	5249
	5250#	5251	5261#	5273#	5287#	5296#	5309#	5320#	5329#	5338#	5339	5340#	5341	5349#	5360#
	5370#	5380#	5381	5382#	5383	5400#	5410#	5418#	5426#	5431#	5442#	5455#	5461#	5466#	5471#
	5477#	5483#	5493#	5494	5508#	5522#	5535#	5546#	5559	5572#	5585#	5595#	5608#	5619#	5630#
	5644#	5657#	5667#	5673#	5674	5695#	5705#	5713#	5723#	5724	5742#	5753#	5763#	5769#	5770
	5789#	5803#	5813#	5823#	5833#	5843#	5853#	5863	5868#	5869	5886#	5900#	5910#	5920#	5930#
	5940#	5950#	5960	5965#	5966	5985#	5999#	6013#	6026#	6040#	6053#	6066#	6080#	6093#	6106#
	6119#	6131#	6132	6159#	6167#	6169#	6170	6186#	6196#	6207#	6212#	6213	6229#	6243#	6253#
	6264#	6269#	6270	6290#	6301#	6311#	6321#	6332#	6338#	6339	6357#	6368#	6380#	6385#	6386
	6418#	6427#	6433#	6434	6458#	6468#	6474#	6475	6498#	6517#	6530#	6543#	6556#	6569#	6581
	6585#	6592#	6632#	6633	6667#	6668	6693#	6697#	6698	6727#	6728	6753#	6757#	6758	6789#
	6790	6815#	6819#	6820	6848#	6855#	6856	6880#	6887#	6888	6904#	6914#	6924#	6933#	6942#
	6950#	6958#	6963#	6972#	6977#	6991#	7012#	7026#	7048#	7053#	7069#	7074#	7083#	7089#	7101#
	7106#	7110#	7111												
MSLAB	1#	1084#	1635#	1678#	1725#	1739#	1815#	1827#	1856#	1886#	1915#	1943#	1972#	2016#	2029#
	2045#	2052#	2233#	2277#	2285#	2312#	2321#	2388#	2504#	2515#	2522#	2642#	2671#	2674#	2684#
	2688#	2696#	2700#	2708#	2712#	2720#	2724#	2732#	2736#	2750#	2754#	2765#	2769#	2776#	2780#
	2786#	2794#	2798#	2804#	2810#	2815#	2819#	2828#	2832#	2841#	2845#	2855#	2859#	2871#	2880#
	2884#	2892#	2896#	3233#	3261#	3265#	3269#	3273#	3292#	3328#	3341#	3344#	3346#	3356#	3359#
	3374#	3376#	3392#	3424#	3437#	3440#	3443#	3450#	3460#	3501#	3511#	3528#	3535#	3541#	3576#
	3586#	3603#	3610#	3616#	3651#	3661#	3678#	3685#	3691#	3726#	3736#	3753#	3760#	3766#	3801#

	3811#	3828#	3835#	3841#	3876#	3886#	3903#	3910#	3916#	3935#	3944#	3952#	3957#	3966#	3975#
	3992#	4001#	4009#	4013#	4022#	4028#	4096#	4100#	4107#	4113#	4117#	4121#	4130#	4139#	4142#
	4174#	4182#	4197#	4217#	4221#	4237#	4255#	4259#	4282#	4290#	4297#	4303#	4321#	4327#	4332#
	4339#	4353#	4355#	4373#	4381#	4386#	4398#	4402#	4407#	4412#	4419#	4420#	4429#	4430#	4439#
	4447#	4466#	4480#	4501#	4510#	4526#	4536#	4552#	4557#	4566#	4571#	4580#	4601#	4609#	4617#
	4626#	4630#	4636#	4642#	4664#	4681#	4685#	4693#	4703#	4711#	4729#	4756#	4766#	4777#	4783#
	4812#	4825#	4842#	4854#	4869#	4882#	4901#	4915#	4930#	4944#	4954#	4965#	4975#	4992#	5010#
	5024#	5037#	5047#	5054#	5072#	5086#	5099#	5110#	5117#	5134#	5148#	5161#	5172#	5179#	5192#
	5203#	5214#	5225#	5237#	5249#	5251#	5261#	5273#	5287#	5296#	5309#	5320#	5329#	5339#	5341#
	5349#	5360#	5370#	5381#	5383#	5400#	5410#	5418#	5426#	5431#	5442#	5455#	5461#	5466#	5471#
	5477#	5483#	5494#	5508#	5522#	5535#	5546#	5559#	5572#	5585#	5595#	5608#	5619#	5630#	5644#
	5657#	5667#	5674#	5695#	5705#	5713#	5724#	5742#	5753#	5763#	5770#	5789#	5803#	5813#	5823#
	5833#	5843#	5853#	5863#	5869#	5886#	5900#	5910#	5920#	5930#	5940#	5950#	5960#	5966#	5985#
	5999#	6013#	6026#	6040#	6053#	6066#	6080#	6093#	6106#	6119#	6132#	6159#	6167#	6170#	6186#
	6196#	6207#	6213#	6229#	6243#	6253#	6264#	6270#	6290#	6301#	6311#	6321#	6332#	6339#	6357#
	6368#	6380#	6386#	6418#	6427#	6434#	6458#	6468#	6475#	6498#	6517#	6530#	6543#	6556#	6569#
	6581#	6585#	6592#	6633#	6668#	6693#	6698#	6728#	6753#	6758#	6790#	6815#	6820#	6848#	6856#
	6880#	6888#	6904#	6914#	6924#	6933#	6942#	6950#	6958#	6963#	6972#	6977#	6991#	7012#	7026#
	7048#	7053#	7069#	7074#	7083#	7089#	7101#	7106#	7111#						
MSTSTL	1#	1084#	1635#	1678#	1725#	1739#	1815#	1827#	1856#	1886#	1915#	1943#	1972#	2016#	2029#
	2045#	2052#	2233#	2277#	2285#	2312#	2321#	2388#	2504#	2515#	2522#	2642#	2671#	2674#	2684#
	2688#	2696#	2700#	2708#	2712#	2720#	2724#	2732#	2736#	2750#	2754#	2765#	2769#	2776#	2780#
	2786#	2794#	2798#	2804#	2810#	2815#	2819#	2828#	2832#	2841#	2845#	2855#	2859#	2871#	2880#
	2884#	2892#	2896#	3233#	3261#	3265#	3269#	3273#	3292#	3328#	3341#	3344#	3346#	3356#	3359#
	3374#	3376#	3392#	3424#	3437#	3440#	3443#	3450#	3460#	3501#	3511#	3528#	3535#	3541#	3576#
	3586#	3603#	3610#	3616#	3651#	3661#	3678#	3685#	3691#	3726#	3736#	3753#	3760#	3766#	3801#
	3811#	3828#	3835#	3841#	3876#	3886#	3903#	3910#	3916#	3935#	3944#	3952#	3957#	3966#	3973#
	3992#	4001#	4009#	4013#	4022#	4028#	4096#	4100#	4107#	4113#	4117#	4121#	4130#	4139#	4142#
	4174#	4182#	4197#	4217#	4221#	4237#	4255#	4259#	4282#	4290#	4297#	4303#	4321#	4327#	4332#
	4339#	4353#	4355#	4373#	4381#	4386#	4398#	4402#	4407#	4412#	4419#	4420#	4429#	4430#	4439#
	4447#	4466#	4480#	4501#	4510#	4526#	4536#	4552#	4557#	4566#	4571#	4580#	4601#	4609#	4617#
	4626#	4630#	4636#	4642#	4664#	4681#	4685#	4693#	4703#	4711#	4729#	4756#	4766#	4777#	4783#
	4812#	4825#	4842#	4854#	4869#	4882#	4901#	4915#	4930#	4944#	4954#	4965#	4975#	4992#	5010#
	5024#	5037#	5047#	5054#	5072#	5086#	5099#	5110#	5117#	5134#	5148#	5161#	5172#	5179#	5192#
	5203#	5214#	5225#	5237#	5249#	5251#	5261#	5273#	5287#	5296#	5309#	5320#	5329#	5339#	5341#
	5349#	5360#	5370#	5381#	5383#	5400#	5410#	5418#	5426#	5431#	5442#	5455#	5461#	5466#	5471#
	5477#	5483#	5494#	5508#	5522#	5535#	5546#	5559#	5572#	5585#	5595#	5608#	5619#	5630#	5644#
	5657#	5667#	5674#	5695#	5705#	5713#	5724#	5742#	5753#	5763#	5770#	5789#	5803#	5813#	5823#
	5833#	5843#	5853#	5863#	5869#	5886#	5900#	5910#	5920#	5930#	5940#	5950#	5960#	5966#	5985#
	5999#	6013#	6026#	6040#	6053#	6066#	6080#	6093#	6106#	6119#	6132#	6159#	6167#	6170#	6186#
	6196#	6207#	6213#	6229#	6243#	6253#	6264#	6270#	6290#	6301#	6311#	6321#	6332#	6339#	6357#
	6368#	6380#	6386#	6418#	6427#	6434#	6458#	6468#	6475#	6498#	6517#	6530#	6543#	6556#	6569#
	6581#	6585#	6592#	6633#	6668#	6693#	6698#	6728#	6753#	6758#	6790#	6815#	6820#	6848#	6856#
	6880#	6888#	6904#	6914#	6924#	6933#	6942#	6950#	6958#	6963#	6972#	6977#	6991#	7012#	7026#
	7048#	7053#	7069#	7074#	7083#	7089#	7101#	7106#	7111#						
MSWORD	1#	1084#	1155#	1164	1214#	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225
	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240
	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255
	1256	1257	1258	1635#	1636	1637	1638	1678#	1679	1680	1681	1725#	1726	1727	1728
	1739#	1740	1741	1742	1815#	1816	1817	1818	1827#	1828	1829	1830	1856#	1857	1858
	1859	1886#	1887	1888	1889	1915#	1916	1917	1918	1943#	1944	1945	1946	1972#	1973
	1974	1975	2016#	2017	2018	2019	2029#	2030	2031	2032	2045#	2046	2047	2048	2052#
	2053	2054	2055	2233#	2234	2235	2236	2277#	2278	2279	2280	2388#	2389	2390	2391
	2642#	2643	2644	2645	3228#	3450#	3451	3452	3453	3501#	3502	3503	3504	3511#	3512
	3513	3514	3535#	3536	3537	3538	3576#	3577	3578	3579	3586#	3587	3588	3589	3610#
	3611	3612	3613	3651#	3652	3653	3654	3661#	3662	3663	3664	3685#	3686	3687	3688

	3726#	3727	3728	3729	3736#	3737	3738	3739	3760#	3761	3762	3763	3801#	3802	3803
	3804	3811#	3812	3813	3814	3835#	3836	3837	3838	3876#	3877	3878	3879	3886#	3887
	3888	3889	3910#	3911	3912	3913	3952#	3953	3954	3955	4009#	4010	4011	4012	4096#
	4097	4098	4099	4107#	4108	4109	4110	4117#	4118	4119	4120	4217#	4218	4219	4220
	4255#	4256	4257	4258	4381#	4382	4383	4384	4398#	4399	4400	4401	4407#	4420#	4421
	4422	4423	4430#	4431	4432	4433	4501#	4502	4503	4504	4552#	4553	4554	4555	4566#
	4567	4568	4569	4626#	4627	4628	4629	4636#	4637	4638	4639	4681#	4682	4683	4684
	4685#	4777#	4778	4779	4780	5172#	5173	5174	5175	5559#	5560	5561	5562	5863#	5864
	5865	5866	5960#	5961	5962	5963	6581#	6582	6583	6584	6592#	7134#	7139#	7143#	7147#
	7152#	7157#	7162#	7330	7331										
MSXFER	1#	1084#													
NEWTST	1085#	3405	3467	3542	3617	3692	3767	3842	3917	3974	4031	4145	4450	4512	4581
	4645	4713	4785	4884	4995	5057	5120	5180	5386	5496	5678	5727	5773	5871	5968
	6136	6172	6215	6275	6342	6392	6435	6477	6639	6699	6760	6824	6858	6889	
OPEN	1#	1084#													
POINTE	1#	1084#	1114												
PRINTB	1#	1084#	2667	2680	2691	2703	2717	2729	2743	2759	2772	2733	2788	2801	2806
	2812	2822	2835	2850	2866	2873	2889								
PRINTF	1#	1084#	2501	2512	2517	3522	3597	3672	3747	3822	3897	3960	4016		
PRINTS	1#	1084#													
PRINTX	1#	1084#	2306	2314	3455										
READBU	1#	1084#													
READEF	1#	1084#	3260	3264	3268	3272									
RFLAGS	1#	1084#													
SETPRI	1#	1084#	4296	4331	4411	4418	4428	4438	6667	6727	6789				
SETVEC	1#	1084#	3420	4278	4286	4317	4323								
SLASH	1#	1084#													
STARS	1#	1084#													
SVC	1#	1084#													
TOR	159#	4127	4195	4235	4301	4337	4371	4598							
WFE	1668#	4700	4753	4851	5044	5107	5293	5592	5664	5710	5760	5820	5840	5917	5937
	6204	6261	6329	6377											
WFR	1686#	4179	4477	4533	4606	4690	4763	5439	5702	6193	6250	6318	6365	7009	
WFRO	1687#														
XFER	1#	1084#	3228#	4407#	4685#	6592#									
XFERF	1#	1084#													
XFERT	1#	1084#													
ZZ	3405#	3407	3467#	3469	3542#	3544	3617#	3619	3692#	3694	3767#	3769	3842#	3844	3917#
	3919	3974#	3976	4031#	4033	4145#	4147	4450#	4452	4512#	4514	4581#	4583	4645#	4647
	4713#	4715	4785#	4787	4884#	4886	4995#	4997	5057#	5059	5119#	5122	5180#	5182	5386#
	5388	5496#	5498	5678#	5680	5727#	5729	5773#	5775	5871#	5873	5968#	5970	6135#	6138
	6172#	6174	6215#	6217	6275#	6277	6342#	6344	6391#	6394	6435#	6437	6477#	6479	6639#
	6641	6699#	6701	6760#	6762	6824#	6826	6858#	6860	6889#	6891				

. ABS. 036746 000

ERRORS DETECTED: 0

CZDMTA/I,CZDMTA.SEQ/CRF/SOL=SVC34R.MLB,CZDMTA.REL
 RUN-TIME: 32 40 5 SECONDS
 RUN-TIME RATIO: 115/77=1.4
 CORE USED: 20K (39 PAGES)