

.REMA

#### IDENTIFICATION

PRODUCT ID: AC-T777A-MC  
PRODUCT TITLE: CZTKFA TK25 FRT END FUNC 02  
PRODUCT DATE: MARCH, 1984  
DEPARTMENT: TAPE DIAGNOSTIC ENGINEERING  
AUTHOR: DICE SYSTEMS, INC.

COPYRIGHT (C) 1984 BY  
DIGITAL EQUIPMENT CORPORATION,  
WESTBORO, MASSACHUSETTS.  
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

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

TABLE OF CONTENTS

|     |                                            |
|-----|--------------------------------------------|
| 1.0 | ABSTRACT                                   |
| 2.0 | REQUIREMENTS                               |
| 2.1 | HARDWARE REQUIREMENTS                      |
| 2.2 | SOFTWARE REQUIREMENTS                      |
| 2.3 | PREREQUISITES                              |
| 3.0 | OPERATING INSTRUCTIONS - OPERATOR COMMANDS |
| 3.1 | OPERATOR COMMANDS                          |
| 3.2 | HARDWARE PARAMETERS                        |
| 3.3 | SOFTWARE PARAMETERS                        |
| 4.0 | OPERATING INSTRUCTIONS - SAMPLE PRINTOUTS  |
| 4.1 | SUCCESSFUL RUN EXAMPLES                    |
| 4.2 | ERROR MESSAGES                             |
| 5.0 | PROGRAM RUN TIMES                          |
| 5.1 | RUN TIME - CZTKF                           |
| 6.0 | TEST DESCRIPTIONS - CZTKF                  |
| 6.1 | TEST 1 - INITIALIZE #4                     |
| 6.2 | TEST 2 - OFF-LINE REJECT/REWIND TEST       |
| 6.3 | TEST 3 - BASIC WRITE DATA TEST             |
| 6.4 | TEST 4 - BASIC READ DATA TEST              |
| 6.5 | TEST 5 - MANUAL INTERVENTION               |
| 6.6 | TEST 6 - CONFIGURATION TYPEOUT TEST        |
| 6.7 | TEST 7 - SCOPE LOOPS                       |

## 1.0 ABSTRACT

THIS IS A PDP-11/LSI RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF AN TK25 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11 SYSTEM (Q-BUS OR UNIBUS). THE PROGRAM HAS BEEN DIVIDED INTO FOUR MAJOR PIECES: CZTKE, CZTKF, CZTKG, CZTKH. SUCCESSFUL RUN EXAMPLES, AND TEST DESCRIPTIONS HAVE BEEN PROVIDED FOR EACH PROGRAM.

THE PROGRAMS PROVIDE ERROR MESSAGES WHICH IDENTIFY FAILING CONDITIONS, AND AID IN DEVICE REPAIR. REFERENCE THE FOLLOWING DIGITAL EQUIPMENT DOCUMENTS:

1. CIQPMAO XXDP. PROGRAMMER'S MANUAL; DOCUMENT NUMBER AC-S296A-AC;  
DATE: 14 JULY 1980.

## 1.1 REVISION HISTORY

NEW RELEASE APRIL 1984

## 2.0 REQUIREMENTS

### 2.1 HARDWARE REQUIREMENTS

PDP-11 FAMILY PROCESSOR WITH 32K WORDS OF MEMORY  
TK25 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)  
CAUTION: DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY  
(28K USEABLE I.E. 4K FOR I/O PAGE)

#### 2.1.1 OPTIONAL HARDWARE -

FOUR TK25 CONTROLLERS PER PDP-11, ONE  
DRIVE PER CONTROLLER

### 2.2 SOFTWARE REQUIREMENTS

PDP-11 DIAGNOSTIC SUPERVISOR (CIQPMAS VERSION 34 OR LATER)  
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

### 2.3 PREREQUISITES

FUNCTIONAL PDP-11/LSI FAMILY CENTRAL PROCESSOR AND MEMORY  
FUNCTIONAL CONSOLE TERMINAL  
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR

### 3.0 OPERATING INSTRUCTIONS - OPERATOR COMMANDS

#### 3.1 OPERATOR COMMANDS

THE TK25 DIAGNOSTICS ARE PDP-11 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAMS.  
ALL LOADING AND RUN TIME INSTRUCTIONS CAN BE REFERENCED IN THE PDP-11  
PROGRAMMER'S MANUAL "CIQPMAD XXDP" PROGRAMMER'S MANUAL NUMBER AC-S296A-AC.

BOOT THE DIAGNOSTIC XXDP+ MEDIA (OPERATOR RESPONSES ARE UNDERLINED)

CHMDLEO XXDP+ DL MONITOR  
BOOTED VIA UNIT 0  
28K NON-UNIBUS SYSTEM

ENTER DATE <DD-MMM-YY>: 29-JAN-82

RESTART ADDRESS: 152010 -----  
THIS IS XXDP+ TYPE "H" OR "H/L" FOR HELP.

.R CZTKFA

-----  
CZTKFA.BIC

DRS-EO  
CZTKF-A-0  
CZTKFA TK-25 FRT END FUNC 02 UNIT IS TK25  
RSTRT ADR 147642  
DR>START/FLAG:PNT:HOE

-----  
THE ABOVE COMMAND WILL START THE DIAGNOSTIC. THE COMMAND HAS TWO  
SWITCHES ON WHICH ARE "PRINT EACH TEST NUMBER AS EXECUTED" AND "HALT ON  
ERROR".

### 3.2 HARDWARE PARAMETERS

AFTER INITIAL STARTING OF THE PROGRAM (START COMMAND TO THE DIAGNOSTIC SUPERVISOR), THE PROGRAM WILL ISSUE THE "CHANGE HW?" QUESTION TO ASK IF THE HARDWARE PARAMETERS ARE TO BE CHANGED (BY THE OPERATOR).

ON A "N" (NO) RESPONSE TO THE QUESTION, THE PROGRAM WILL USE IT'S DEFAULT HARDWARE PARAMETER VALUES. IT WILL DEFAULT TO ONE UNIT SELECTED (UNIT 0), THE DEFAULT TSBA/TSDB WILL BE 172522 AND THE INTERRUPT VECTOR WILL BE 224.

ON A "Y" (YES) RESPONSE TO THE QUESTION, THE FOLLOWING QUESTIONS WILL THEN BE ASKED TO ALLOW THE OPERATOR TO SELECT THE UNITS TO BE TESTED. A VALUE, IF PRESENT, LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ONLY IF A CARRIAGE RETURN IS TYPED AS A RESPONSE. A "(D)" IN A QUESTION INDICATES THAT A DECIMAL NUMBER IS REQUIRED AS A RESPONSE. AN "(O)" INDICATES AN OCTAL NUMBER IS BEING SOLICITED. AN "(L)" THAT A LOGICAL RESPONSE IS TO BE MADE: "Y" FOR YES, "N" FOR NO.

UNITS (D) ? < ENTER THE NUMBER OF CONTROLLERS  
PRESENT TO BE TESTED >

UNIT 0

DEVICE ADDRESS (O) 172522 ? <ENTER THE ADDRESS OF THE  
TSBA/TSDB REGISTER >

VECTOR (O) 224 ?      <ENTER ADDRESS OF INTERRUPT  
VECTOR >

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE " UNITS ?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING.

### 3.3 SOFTWARE PARAMETERS

THE  
FOLLOWING QUESTIONS ARE ASKED ON A START, RESTART, OR CONTINUE. THEY  
ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES.

CHANGE SW (L) ? < TYPE "Y" TO CAUSE THE FOLLOWING  
QUESTIONS TO BE ASKED.>

INHIBIT ITERATIONS (L) N ? < TYPE "Y" TO PREVENT MULTIPLE  
ITERATIONS OF CERTAIN TESTS.  
THIS CAUSES EACH TEST PASS TO  
RUN AS QUICKLY AS POSSIBLE.  
ONLY QUICK-RUNNING LOGIC  
TESTS USE MULTIPLE ITERATIONS.>

ENABLE CONTROLLER RAM DUMP ON ERROR (L) N? < TYPE "Y" TO DUMP  
SELECTED RAM CONTENTS IN THE  
CONTROLLER MODULE.>

#### 4.0 OPERATING INSTRUCTIONS - SAMPLE PRINTOUTS

#### 4.1 SUCCESSFUL RUN EXAMPLES

##### 4.1.1 SUCCESSFUL RUN EXAMPLE - CZTKFA -

```
TST: 001 INITIALIZATION #2 TEST
TST: 002 OFF-LINE REJECT AND REWIND TEST
TST: 003 BASIC WRITE TEST
TST: 004 BASIC READ DATA (FORWARD AND REVERSE) TEST
TST: 005 STAND-ALONE MANUAL INTERVENTION NOT EXECUTED TEST
TST: 006 STAND-ALONE CONFIGURATION TIMEOUT NOT EXECUTED TEST
TST: 007 STAND-ALONE SCOPE LOOPS NOT EXECUTED TEST
CZTKF EOP      1
      0 TOTAL ERRS
```

NOTE: PROGRAM NOW STARTS OVER AGAIN AT TEST 1

#### 4.2 OPERATING INSTRUCTIONS - SAMPLE ERROR MESSAGE

##### ERROR MESSAGE EXAMPLE

TST: 003 BASIC WRITE TEST  
CZTKF WRD ERR 00303 ON UNIT 00 TST 003 SUB 001 PC:026500  
TSSR INCORRECT AFTER WRITE COMMAND, MORE BITS SET THAN SSR

TSSR=000000  
TERMINATION CLASS CODE = NORMAL TERMINATION  
\*\*\*\*\*REPLACE CONTROLLER\*\*\*\*\*  
PACKET ADDRESS=030510  
PACKET WORD #0=140005  
PACKET WORD #1=053470  
PACKET WORD #2=000000  
PACKET WORD #3=000024

MESSAGE BUFFER ADDRESS=030400  
MESSAGE BUFFER CONTENTS:  
MESSAGE BUFFER HEADER    =100020  
DATA FIELD LENGTH        =000012  
RESIDUAL BYTE COUNTER    =000000  
XSTAT0 CONTENTS          =000300  
XSTAT1 CONTENTS          =000000  
XSTAT2 CONTENTS          =101000  
XSTAT3 CONTENTS          =000000

## 5.0 PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAMS ARE LISTED BELOW. THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A PDP-11/23 (LSI) PROCESSOR WITH LA-120 CONSOLE.

THE PROGRAMS RUN IN NON-ITERATIVE MODE. EACH TEST IS RUN ONCE, WITH NO ITERATIONS. THEREFOR, THE DEFAULT MODE (NORMALLY ITERATIVE) AND THE NON-ITERATIVE MODE TIMES ARE IDENTICAL.

### 5.1 RUN TIMES - CZTKF

| TEST<br>NUMBER | N/I<br>SECS. | DEF<br>SECS. |
|----------------|--------------|--------------|
| 1              | 2            | 2            |
| 2              | 2            | 2            |
| 3              | 41           | 41           |
| 4              | 70           | 70           |
| 5              | N/A          | N/A          |
| 6              | N/A          | N/A          |
| 7              | N/A          | N/A          |

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 4 IN ONE COMMAND:

|         |                  |
|---------|------------------|
| Q.V.    | 1 MIN 55 SECONDS |
| DEFAULT | 1 MIN 55 SECONDS |

L1

## 6.0 TEST DESCRIPTIONS - CZTKF

### 6.1 TEST 1 - INITIALIZATION #2

\*\*\*\*\*  
\* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S \*  
\* CONTROLLER \*  
\*\*\*\*\*

THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE CONTROLLER TO ITS  
INITIALIZED STATE FROM VARIOUS CONDITONS.

## 6.2 TEST 2 - OFF LINE REJECT AND REWIND TEST

THIS TEST VERIFIES BASIC TAPE MOTION COMMAND DECODING AND BASIC OPERATION OF THE REWIND POSITIONING COMMAND. IT DOES NOT NECESSARILY DEMONSTRATE THAT THE TRANSPORT CAN BE REWOUND FROM AN ARBITRARY POSITION ON THE TAPE. SUBSEQUENT TESTS IMPLICITLY CHECK THE OPERATION OF THE REWIND COMMAND SINCE THEY MUST TYPICALLY REWIND THE TAPE IN IN THE NORMAL COURSE OF THEIR TEST SEQUENCES. THE TEST CONSISTS OF THE FOLLOWING THREE SUBTESTS:

## 6.2.1 TEST 2, SUBTEST 1: -

THIS SUBTEST VERIFIES THAT A REWIND COMMAND, WITH THE CLEAR VOLUME CHECK (CVC) BIT CLEAR IS REJECTED IF THE VOLUME CHECK (VCK) FLAG IS SET.

## 6.2.2 TEST 2, SUBTEST 2: -

THIS SUBTEST VERIFIES THAT A REWIND COMMAND WITH A CVC=1 CLEARS VCK AND RETURNS PROPER STATUS IN THE MESSAGE BUFFER.

### 6.3 TEST 3 - BASIC WRITE TEST

\*\*\*\*\*  
\* NOTE: THIS TEST MUST HAVE A GOOD MAGTAPE IN THE DRIVE. ANY \*  
\* TAPE ERRORS WILL BE DISPLAYED AS TAPE STATUS ALERT \*  
\*\*\*\*\*

THIS TEST VERIFIES THAT THE WRITE DATA (NEXT) COMMAND OPERATES CORRECTLY, UP TO THE POINT OF CHECKING THAT THE DATA WAS ACTUALLY WRITTEN ONTO THE TAPE CORRECTLY. THE TESTING IN THIS TEST IS LIMITED TO VERIFYING THAT THE COMMAND WAS TERMINATED CORRECTLY WITH THE CORRECT REGISTER, BUFFER, AND RAM CONTENTS.

#### 6.3.1 TEST 3, SUBTEST 1: -

THIS SUBTEST VERIFIES THAT A WRITE COMMAND (ANY VALID MODE CODE) WITH THE CLEAR VOLUME CHECK (CVC) BIT CLEAR IS REJECTED IF THE VOLUME CHECK (VCK) FLAG IS SET. ALL VALID MODE CODES ARE CHECKED (WRITE DATA, WRITE RETRY).

#### 6.3.2 TEST 3, SUBTEST 2: -

THIS SUBTEST VERIFIES THAT WRITE DATA COMMANDS WITH CVC=1 AND THE SWAP BYTES (SWB) BIT CLEAR OPERATES PROPERLY. THE BYTE COUNT (RECORD SIZE) VARIES FROM 20 THROUGH 64K IN VARYING INCREMENTS (DEPENDING ON WHETHER OR NOT THE DIAGNOSTIC IS RUNNING ON THE LONG VERIFICATION MODE). THE TAPE IS NOT REWOUND BETWEEN SUCCESSIVE RECORDS BUT IS REWOUND AFTER THE FINAL RECORD IS WRITTEN. AN INCREMENTING COUNT PATTERN IS SUPPLIED IN THE DATA BUFFER. AFTER EACH BLOCK IS WRITTEN, THE TTSR AND THE TSBA REGISTERS AND THE MESSAGE BUFERS ARE CHECKED.

#### 6.3.3 TEST 3, SUBTEST 3: -

THIS SUBTEST VERIFIES THAT WRITE DATA COMMANDS WITH CVC=1 AND THE SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2. THE RESULTS SHOULD BE THE SAME.

#### 6.3.4 TEST 3, SUBTEST 4: -

THIS SUBTEST VERIFIES THAT A WRITE COMMAND WITH AN ILLEGAL BUFFER ADDRESS IS REJECTED WITH THE PROPER ERROR STATUS AND THAT TAPE DOES NOT MOVE.

6.3.5    TEST 3, SUBTEST 5: -

THIS SUBTEST VERIFIES THAT A WRITE DATA COMMAND SPECIFYING A DATA BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH THE PROPER ERROR STATUS WITHOL MOVING TAPE. THIS TEST IS SKIPPED IF NONEXISTANT MEMORY CAN NOT BE ADDRESSED.

6.3.6    TEST 3, SUBTEST 6: -

THIS SUBTEST VERIFIES THAT A WRITE DATA COMMAND SPECIFYING A DATA BUFFER STARTING IN EXISTANT MEMORY BUT RUNNING INTO NONEXISTANT MEMORY TERMINATES WITH THE PROPER ERROR STATUS. A LARGE ENOUGH RECORD SIZE IS SPECIFIED SUCH THAT TAPE IS ACTUALLY MOVED AND WRITTEN.

#### 6.4 TEST 4 - BASIC READ DATA TEST (FORWARD AND REVERSE)

\*\*\*\*\*  
\* NOTE: THIS TAPE MUST HAVE A GOOD MAGTAPE IN THE DRIVE \*  
\* ANY TAPE ERRORS WILL BE DISPLAYED AS TAPE STATUS ALERT \*  
\*\*\*\*\*

THIS TEST VERIFIES THAT THE READ FORWARD AND READ REVERSE COMMANDS OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN DATA BUFFER BOUNDARIES, RECORD SIZES AND BYTE SWAP CONTROL VARIABLES ARE USED. THE TEST FURTHER VERIFIES THE WRITE DATA COMMAND BY ACTUALLY READING AND VERIFYING WRITTEN DATA. ALSO TESTED ARE PROPER TERMINATIONS ON EXCEPTIONAL OR ERROR CONDITIONS: RECORD LENGTH LONG, RECORD LENGTH SHORT, READ REVERSE AT BOT, ILLEGAL DATA BUFFER ADDRESSES, AND NONEXISTANT DATA BUFFER ADDRESSES.

##### 6.4.1 TEST 4, SUBTEST 1: -

THIS SUBTEST VERIFIES THAT THE READ FORWARD COMMAND WITH SWB=0 OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN WRITTEN WITH A SERIES OF TEST RECORDS VARYING IN LENGTH AND DATA CONTENT. THE TAPE IS THEN REWOUND AGAIN AND THE RECORD READ SEQUENTIALLY AND RESULTS (STATUS, DATA, ETC.) VERIFIED. THE BYTE COUNT ON EACH READ FORWARD COMMAND IS SET TO THE LENGTH OF THE EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD OCCUR.

##### 6.4.2 TEST 4, SUBTEST 2: -

THIS SUBTEST VERIFIES THAT THE READ DATA COMMANDS WITH CVC=1 AND THE SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2. THE RESULTS, EXCEPT FOR RAM CONTENTS SHOULD BE THE SAME.

##### 6.4.3 TEST 4, SUBTEST 3: -

THIS SUBTEST VERIFIES THAT A READ FORWARD COMMAND READING A RECORD LONGER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH (RL) BIT SET.

##### 6.4.4 TEST 4, SUBTEST 4: -

THIS SUBTEST VERIFIES THAT A READ FORWARD COMMAND READING A RECORD SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH SHORT (RLS) BIT SET. IT IS VERIFIED THAT THE RESIDUAL BYTE COUNT (RBPCH) IN THE MESSAGE BUFFER CONTAINS THE PROPER NONZERO VALUE (E.G. THE DIFFERENCE BETWEEN THE ACTUAL BYTE COUNT AND THE ACTUAL RECORD LENGTH).

6.4.5 TEST 4, SUBTEST 5: -

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN WRITTEN WITH A SERIES OF TEST RECORDS VARYING IN LENGTH AND DATA CONTENT. THE TAPE IS THEN READ IN REVERSE SEQUENTIALLY AND THE RESULTS (STATUS, DATA, ETC.) VERIFIED. THE BYTE COUNT ON EACH READ REVERSE COMMAND IS SET TO THE LENGTH OF THE EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD OCCUR.

6.4.6 TEST 4, SUBTEST 6: -

THIS SUBTEST VERIFIES THAT THE READ DATA COMMANDS WITH CVC=1 AND THE SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2. THE RESULTS EXCEPT FOR RAM CONTENTS SHOULD BE THE SAME.

6.4.7 TEST 4, SUBTEST 7: -

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND, READING A RECORD LONGER THAN THE SPECIFIED BYTE COUNT, CAUSES A TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH LONG (RLL) BIT SET.

6.4.8 TEST 4, SUBTEST 8: -

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND SPECIFYING A DATA BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH THE PROPER ERROR STATUS WITHOUT MOVING THE TAPE.

6.4.9 TEST 4, SUBTEST 9: -

THIS SUBTEST VERIFIES THAT ILLEGAL BUFFER ADDRESSES CAUSE A FUNCTION REJECT TERMINATION WITH ILLEGAL ADDRESS (ILA) ERROR BIT SET.

6.4.10 TEST 4, SUBTEST 10: -

THIS SUBTEST VERIFIES THAT A DATA BUFFER ADDRESS, REFERENCING NONEXISTANT MEMORY, CAUSES RECOVERABLE ERROR TERMINATION (TC=4), WITH THE NXM BIT SET IN THE TSSR, AND THAT THE TAPE IS ULTIMATELY POSITIONED PROPERLY.

## 6.4.11 TEST 4, SUBTEST 11: -

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND ISSUED WHILE THE TAPE IS AT BOT RESULTS IN A FUNCTION REJECT TERMINATION WITH THE NONEXECUTABLE FUNCTION (NEF) ERROR BIT SET.

## 6.4.12 TEST 4, SUBTEST 12: -

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND ISSUED WHILE THE TAPE IS POSITIONED BEFORE THE FIRST RECORD ON TAPE (BUT NOT AT BOT) RESULTS IN TAPE STATUS ALERT.

## 6.5 TEST 5 - MANUAL INTERVENTION

THIS TEST MUST BE STARTED AS FOLLOWS:

AT THE DIAGNOSTIC SUPERVISOR PROMPT "DR>"  
TYPE "START/FLAG:PNT/TEST:5/PASS:1"

-----  
THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST") THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE. THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR ARE BY TYPING <CTRL-C> OR SELECTING CODE 4. SELECTION CODES AND SUBROUTINES ARE:

| CODE | ROUTINE                                  |
|------|------------------------------------------|
| 0    | DISPLAY THIS MENU                        |
| 1    | REWIND AND UNLOAD COMMAND TEST           |
| 2    | WRITE PROTECT TEST                       |
| 3    | FRONT PANEL ON-LINE/OFF-LINE SWITCH TEST |
| 4    | RETURN TO THE DIAGNOSTIC SUPERVISOR      |

ENTER MENU SELECTION: (0) ?

EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:

SELECTION 0 - PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.

SELECTION 1 - THIS ROUTINE INSTRUCTS THE OPERATOR TO PLACE THE DRIVE ON-LINE AND AT OR BEYOND BOT. THE TEST WILL THEN ISSUE THE REWIND AND UNLOAD COMMAND. IT WILL ALSO TELL THE OPERATOR IF THE DRIVE ENDED UP ON-LINE OR OFF-LINE.

SELECTION 2 - THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED. THEN WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED. UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS WRITE-PROTECTED. IF THE APPROPRIATE TERMINATION IS NOT RECEIVED, AN ERROR IS REPORTED.

SELECTION 3 - THIS TEST CHECKS THAT THE PROGRAM CAN READ THE SENSE OF THE FRONT PANEL "ON-LINE" BUTTON/LIGHT. THE PROGRAM CHECKS THE STATE OF THE DRIVE (ON-LINE OR OFF-LINE) AND PRINTS A MESSAGE TO NOTIFY THE OPERATOR I.E. "DRIVE IS NOW OFF-LINE" OR "DRIVE IS NOW ON-LINE".

SELECTION 4 - THIS WILL RETURN THE PROGRAM TO THE DIAGNOSTIC SUPERVISOR PROMPT. NOTE: IF THE OPERATOR FAILED TO SELECT A PASS COUNT OF ONE, THE PROGRAM WILL LOOP UNTIL STOPPED WITH A CONTROL C.

## 6.6 TEST 6 - CONFIGURATION TYPEOUT

THIS TEST MUST BE STARTED AS FOLLOWS:

AT THE DIAGNOSTIC SUPERVISOR PROMPT "DR>"  
TYPE "START/FLAG:PNT/TEST:6/PASS:1"

-----

THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL THE CONFIGURATION OF THE CONTROLLER MODULE AND THE TK25 SUBSYSTEM. SPECIFICALLY, THE FOLLOWING INFORMATION IS PRINTED:

1. MICROCODE REVISION LEVEL OF THE CONTROLLER.
2. NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
3. UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED) OF EACH CONNECTED TRANSPORT.

THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES THE ACTUAL CONFIGURATION AT HAND. IF, FOR EXAMPLE, THE PROGRAM INDICATES THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT, THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE CAUSE (BAD CABLE, FAULTY UNIT SELECT DECODING IN THE TRANSPORT, ETC.). [SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD" MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]

THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE SELECTED (E.G. FROM THE INITIAL STARTUP DIALOGUE), THE ROUTINE WILL BE REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.

## 6.7 TEST 7 - SCOPE LOOPS

THIS TEST MUST BE STARTED AS FOLLOWS:

AT THE DIAGNOSTIC SUPERVISOR PROMPT "DR>"  
TYPE "START/FLAG:PNT/TEST:7/PASS:1"

-----

THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE LOOPS"  
USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH THE CONTROLLER MODULE.  
THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL "LOOP ON ERROR" OR "LOOP ON  
TEST (SUBTEST)" FACILITIES DON'T SEEM TO ALLOW THE OPERATOR TO ZERO IN A  
PROBLEM IN THE EARLY TESTS (I.E. THE HARDWARE MAY NOT BE RESPONDING TO A  
REGISTER ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS  
SELECTED BY THE PROGRAM MATCHES THE THE CONFIGURATION SET UP IN THE  
HARDWARE DIP SWITCHES). THE FOLLOWING MENU OF SCOPE LOOPS IS AVAILABLE:

| CODE | SCOPE LOOP                              |
|------|-----------------------------------------|
| 0    | HELP. PRINT THIS MENU                   |
| 1    | TSBA READ ACCESS                        |
| 2    | TSSR READ ACCESS                        |
| 3    | INITIALIZE (TSSR WRITE ACCESS)          |
| 4    | TSDB HIGH BYTE WRITE ACCESS             |
| 5    | TSDB LOW BYTE WRITE ACCESS              |
| 6    | TSDB MAINTENANCE-MODE WORD WRITE ACCESS |
| 7    | EXIT (RETURN TO SUPERVISOR)             |

FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS THE OPERATOR  
FOR THE DATA TO BE WRITTEN. TYPING <RETURN> CAUSES AN EXIT FROM THE SCOPE  
LOOP BACK TO THE MENU.

```
687
688
694      .SBTTL  PROGRAM HEADER
695      .MCALL  SVC
696      SVC
697      .ENABLE LC
703      .MLIST  BEX,CND
704      .ENABL  AMA,ABS
705      . = 2000
706      BGNMOD  TUV2A
707      TUV2A::
708
709      ;**
710      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
711      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
712      ;--
713
714      POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT,BGNSETUP
715      HEADER  CZTKF,A,0,655,,0
716      L$NAME::
717      .ASCII  /C/
718      .ASCII  /Z/
719      .ASCII  /T/
720      .ASCII  /K/
721      .ASCII  /F/
722      .BYTE   0
723      .BYTE   0
724      .BYTE   0
725      L$REV::
726      .ASCII  /A/
727      L$DEPO::
728      .ASCII  /0/
729      L$UNIT::
730      .WORD   T$PTHV
731      L$TIML::
732      .WORD   655,
733      L$HPCP::
734      .WORD   L$HARD
735      L$SPCP::
736      .WORD   L$SOFT
737      L$HPTP::
738      .WORD   L$HW
739      L$SPTP::
740      .WORD   L$SW
741      L$LADP::
742      .WORD   L$LAST
743      L$STA::
744      .WORD   0
745      L$CO::
746      .WORD   0
747      L$DTYP::
748      .WORD   0
749      L$APT::
750      .WORD   0
751      L$DTP::
752      .WORD   L$DISPATCH
```

J2

CZTKFA TK-25 FRT END FUNC 02  
PROGRAM HEADER

MACRO M1200 20-APR-84 08:12 PAGE 22-1

SEQ 22

|        |        |           |                   |                                  |
|--------|--------|-----------|-------------------|----------------------------------|
| 002042 |        | L\$PRIO:: |                   | ;DIAGNOSTIC RUN PRIORITY         |
| 002042 | 000000 |           | .WORD 0           |                                  |
| 002044 |        | L\$ENVI:: |                   | ;FLAGS DESCRIBE HOW IT WAS SETUP |
| 002044 | 000000 |           | .WORD 0           |                                  |
| 002046 |        | L\$EXP1:: |                   | ;EXPANSION WORD                  |
| 002046 | 000000 |           | .WORD 0           |                                  |
| 002050 |        | L\$MREV:: |                   | ;SVC REV AND EDIT #              |
| 002050 | 003    |           | .BYTE C\$REVISION |                                  |
| 002051 | 003    |           | .BYTE C\$EDIT     |                                  |
| 002052 |        | L\$EF::   |                   | ;DIAG. EVENT FLAGS               |
| 002052 | 000000 |           | .WORD 0           |                                  |
| 002054 | 000000 |           | .WORD 0           |                                  |
| 002056 |        | L\$SPC::  |                   |                                  |
| 002056 | 000000 |           | .WORD 0           |                                  |
| 002060 |        | L\$DEVP:: |                   | ; POINTER TO DEVICE TYPE LIST    |
| 002060 | 003334 |           | .WORD L\$DVTYP    |                                  |
| 002062 |        | L\$REPP:: |                   | ;PTR. TO REPORT CODE             |
| 002062 | 023052 |           | .WORD L\$RPT      |                                  |
| 002064 |        | L\$EXP4:: |                   |                                  |
| 002064 | 000000 |           | .WORD 0           |                                  |
| 002066 |        | L\$EXP5:: |                   |                                  |
| 002066 | 000000 |           | .WORD 0           |                                  |
| 002070 |        | L\$AUT::  |                   | ;PTR. TO ADD UNIT CODE           |
| 002070 | 022544 |           | .WORD L\$AU       |                                  |
| 002072 |        | L\$DUT::  |                   | ;PTR. TO DROP UNIT CODE          |
| 002072 | 022642 |           | .WORD L\$DU       |                                  |
| 002074 |        | L\$LUN::  |                   | ;LUN FOR EXERCISERS TO FILL      |
| 002074 | 000000 |           | .WORD 0           |                                  |
| 002076 |        | L\$DESP:: |                   | ;POINTER TO DIAG. DESCRIPTION    |
| 002076 | 003342 |           | .WORD L\$DESC     |                                  |
| 002100 |        | L\$LOAD:: |                   | ;GENERATE SPECIAL AUTOLOAD EMT   |
| 002100 | 104035 |           | EMT E\$LOAD       |                                  |
| 002102 |        | L\$ETP::  |                   | ;POINTER TO ERRtbl               |
| 002102 | 000000 |           | .WORD 0           |                                  |
| 002104 |        | L\$ICP::  |                   | ;PTR. TO INIT CODE               |
| 002104 | 021764 |           | .WORD L\$INIT     |                                  |
| 002106 |        | L\$CCP::  |                   | ;PTR. TO CLEAN-UP CODE           |
| 002106 | 023024 |           | .WORD L\$CLEAN    |                                  |
| 002110 |        | L\$ACP::  |                   | ;PTR. TO AUTO CODE               |
| 002110 | 022750 |           | .WORD L\$AUTO     |                                  |
| 002112 |        | L\$PRT::  |                   | ;PTR. TO PROTECT TABLE           |
| 002112 | 021754 |           | .WORD L\$PROT     |                                  |
| 002114 |        | L\$TEST:: |                   | ;TEST NUMBER                     |
| 002114 | 000000 |           | .WORD 0           |                                  |
| 002116 |        | L\$DLY::  |                   | ;DELAY COUNT                     |
| 002116 | 000000 |           | .WORD 0           |                                  |
| 002120 |        | L\$HIME:: |                   | ;PTR. TO HIGH MEM                |
| 002120 | 000000 |           | .WORD 0           |                                  |

```

716                                     .SBTTL  DEFAULT HARDWARE P-TABLE
717
718                                     ;**
719                                     ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
720                                     ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
721                                     ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
722                                     ;**
723 002122      BGNHW      DFPTBL      ;DEFAULT HARD-P-TABLE
      002122      000003      .WORD      L10000-L$HW/2
      002124
      002124
724
725 002124      172522      .WORD      172522      ; 2ND (OF 2) REGISTERS.
726 002126      000224      .WORD      224      ; INTERRUPT VECTOR
727 002130      000240      .WORD      PRI05      ; INTERRUPT PRIORITY.
728 002132      ENDPHW
      002132      L10000:

```

L2

CZTKFA TK-25 FRT END FUNC #2  
SOFTWARE P-TABLE

MACRO M1200 20-APR-84 08:12 PAGE 24

SEQ 24

```
730          .SBTTL  SOFTWARE P-TABLE
731
732          ;**
733          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
734          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
735          ;--
736 002132      BGNSW   SFPTBL
002132      .WORD   L10001-L$SW/2
002134
002134      L$SW::
SFPTBL::
737
738 002134      000000      TRANSTST::      .WORD   0          ;ENABLE RAM DUMP
739 002136      000000      NOITS::      .WORD   0          ; INHIBIT ITERATION OPTION.
740          ; ... 0 = ITERATE.
741          ; ...NZ = INHIBIT ITERATE.
742 002140      000031      LERRMAX::      .WORD   25.        ; LOCAL (PER TEST) ERROR LIMIT
743 002142      000310      GERRMAX::      .WORD   200.       ; GLOBAL (PER UNIT) ERROR LIMIT
744 002144          ENDSW
002144      L10001:
745
```

748

755

760

766

767

768

769

770

771

772

773

774

775

779 002144

.SBTTL GLOBAL EQUATES SECTION

.SBTTL GLOBAL EQUATES SECTION

; ++  
; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT  
; ARE USED IN MORE THAN ONE TEST.  
; --

EQUALS ; GET STANDARD EQUATES.

;  
; BIT DIFINITIONS

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

|        |        |       |
|--------|--------|-------|
| 001000 | BIT9== | BIT09 |
| 000400 | BIT8== | BIT08 |
| 000200 | BIT7== | BIT07 |
| 000100 | BIT6== | BIT06 |
| 000040 | BIT5== | BIT05 |
| 000020 | BIT4== | BIT04 |
| 000010 | BIT3== | BIT03 |
| 000004 | BIT2== | BIT02 |
| 000002 | BIT1== | BIT01 |
| 000001 | BIT0== | BIT00 |

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

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

;  
; PRIORITY LEVEL DEFINITIONS

```

000340      PRI07== 340
000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
;
; OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

780
781 002144      .SBTTL      KT11      ;DEFINE MEMORY MANAGEMENT REGISTERS
;KT11 MEMORY MANAGEMENT DEFINITIONS
;KT11 VECTOR ADDRESS
000250      MMVEC= 250
;KT11 STATUS REGISTER ADDRESSES
177572      SR0= 177572
177574      SR1= 177574
177576      SR2= 177576
172516      SR3= 172516
;IF NB
;*USER "I" PAGE DESCRIPTOR REGISTERS
UIPDR0= 177600
UIPDR1= 177602
UIPDR2= 177604
UIPDR3= 177606
UIPDR4= 177610
UIPDR5= 177612
UIPDR6= 177614
UIPDR7= 177616
;IF NB
;*USER "D" PAGE DESCRIPTOR REGISTERS
UDPDR0= 177620
UDPDR1= 177622
UDPDR2= 177624
UDPDR3= 177626
UDPDR4= 177630
UDPDR5= 177632
UDPDR6= 177634
UDPDR7= 177636
;ENOC
;*USER "I" PAGE ADDRESS REGISTERS

```

```
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
.IF NB
; *USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
.ENDC
.ENDC
.IF NB
; *SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216
.IF NB
; *SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
SDPDR0= 172220
SDPDR1= 172222
SDPDR2= 172224
SDPDR3= 172226
SDPDR4= 172230
SDPDR5= 172232
SDPDR6= 172234
SDPDR7= 172236
.ENDC
; *SUPERVISOR "I" PAGE ADDRESS REGISTERS
SIPAR0= 172240
SIPAR1= 172242
SIPAR2= 172244
SIPAR3= 172246
SIPAR4= 172250
SIPAR5= 172252
SIPAR6= 172254
SIPAR7= 172256
.IF NB
; *SUPERVISOR "D" PAGE ADDRESS REGISTERS
SDPAR0= 172260
SDPAR1= 172262
SDPAR2= 172264
SDPAR3= 172266
SDPAR4= 172270
```

```

SDPAR5= 172272
SDPAR6= 172274
SDPAR7= 172276
.ENDC
.ENDC
; *KERNEL "I" . AGE DESCRIPTOR REGISTERS
172300 KIPDR0= 172300
172302 KIPDR1= 172302
172304 KIPDR2= 172304
172306 KIPDR3= 172306
172310 KIPDR4= 172310
172312 KIPDR5= 172312
172314 KIPDR6= 172314
172316 KIPDR7= 172316
; IF NB
; *KERNEL "D" PAGE
DESCRIPTOR REGISTERS
KDPDR0= 172320
KDPDR1= 172322
KDPDR2= 172324
KDPDR3= 172326
KDPDR4= 172330
KDPDR5= 172332
KDPDR6= 172334
KDPDR7= 172336
.ENDC
; *KERNEL "I" PAGE ADDRESS REGISTERS
172340 KIPAR0= 172340
172342 KIPAR1= 172342
172344 KIPAR2= 172344
172346 KIPAR3= 172346
172350 KIPAR4= 172350
172352 KIPAR5= 172352
172354 KIPAR6= 172354
172356 KIPAR7= 172356
; IF NB
; *KERNEL "D" PAGE ADDRESS REGISTERS
KDPAR0= 172360
KDPAR1= 172362
KDPAR2= 172364
KDPAR3= 172366
KDPAR4= 172370
KDPAR5= 172372
KDPAR6= 172374
KDPAR7= 172376
.ENDC

```

CZTKEA IK 25 FRI END EUNC 02 MACRO  
TR-25 REGISTER AND PACKET DEFINITIONS

```

786          .SBTTL  TK-25 REGISTER AND PACKET DEFINITIONS
787
788          |
789          | SOME GENERAL EQUATES.
790          |
791
792          000004      ERRVEC==      4              | POINTER TO ERROR VECTOR FOR BUS TIME OUT.
793          000060      TTIVEC==      60             | INTERRUPT VECTOR FOR CONSOLE INPUT
794          177560      TTICSR==     177560          | BUS ADDRESS OF CONSOLE INPUT
795          177562      TTIBFR==     177562          | CONSOLE INPUT DATA BUFFER
796
797          |*
798          |BIT DEFINITIONS FOR TSSR REGISTER
799          |*
800
801          100000      SC=      BIT15              |SPECIAL CONDITION
802          040000      BIE=      BIT14             |BUS INTERFACE ERROR
803          020000      SCE=      BIT13             |SANITY CHECK ERROR
804          010000      RMR=      BIT12             |MODIFICATION REFUSED
805          004000      NXH=      BIT11             |NONEXISTANT MEMORY ERROR
806          002000      NBA=      BIT10             |NEED BUFFER ADDRESS
807          001400      HIADDR=    BIT9!BIT8         |EXTENDED ADDRESS BITS
808          000200      SSR=      BIT7              |SUB SYSTEM READY
809          000100      OFL=      BIT6              |OFF LINE BIT
810          000060      FATERR=    BIT4!BIT5         |FATAL TERMINATION ERROR CODES
811          000016      TERCLS=    BIT3!BIT2!BIT1     |TERMINATION CODES
812
813          |*
814          |
815          |BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
816          |(XST0)
817          |
818          |*
819          |
820
821          100000      XSOTMK=    BIT15             |TAPE MARK DETECTED
822          040000      XSORLS=    BIT14             |RECORD LENGTH SHORT
823          020000      XSOLET=    BIT13             |LOGICAL END OF TAPE
824          010000      XSORLL=    BIT12             |RECORD LENGTH LONG
825          004000      XSOWLE=    BIT11             |WRITE LOCK ERROR
826          002000      XSONEF=    BIT10             |NON EXECUTABLE FUNCTION
827          001000      XSILC=     BIT9              |ILLEGAL COMMAND
828          000400      XSOLA=     BIT8              |ILLEGAL ADDRESS
829          000200      XSOMOT=    BIT7              |TAPE IN MOTION
830          000100      XSOMNL=    BIT6              |TRANSPORT ON LINE
831          000040      XSOLIE=    BIT5              |INTERRUPT ENABLE
832          000020      XSQVCK=    BIT4              |VOLUME CHECK BIT
833          000010      XSOPED=    BIT3              |PHASE ENCODED DRIVE
834          000004      XSOWLK=    BIT2              |WRITE LOCKED
835          000002      XSOSBT=    BIT1              |BEGINNING OF TAPE
836          000001      XSOEOT=    BIT0              |END OF TAPE
837
838          |*
839          |BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
840          |(XST1)
841          |
842          |*

```

```

843      100000      X1.DLT = BIT15      ;DATA LATE
844      040000      X1.SPARE= BIT14      ;NOT USED
845      020000      X1.COR = BIT13      ;CORRECTABLE DATA ERROR
846      017375      X1.MBZ = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
847      000400      X1.RBP = BIT8       ;READ BUS PARITY ERROR
848      000002      X1.UNC = BIT1       ;UNCORRECTABLE DATA OR HARD ERROR
849
850      ;+
851      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
852      ;(XST2)
853      ;-
854      100000      X2.OPM = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
855      040000      X2.RCE = BIT14      ;RAI CHECKSUM ERROR
856      035400      X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8 ;NOT USED BY TK-25 (ALWAYS=0)
857      002000      X2.WCF = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
858      000200      X2.EXTF = BIT7      ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
859      000100      X2.BUFE = BIT6      ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
860      000077      X2.REV = 000077    ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
861      000007      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
862
863      ;+
864      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
865      ;(XST3)
866      ;-
867      177400      X3.MDE = 177400    ;MICRO-DIAGNOSTIC ERROR CODE
868      000200      X3.SPARE= BIT7      ;NOT USED BY TK-25
869      000100      X3.OPI = BIT6      ;OPERATION INCOMPLETE
870      000040      X3.REV = BIT5      ;REVERSE
871      000020      X3.TRF = BIT4      ;TRANSPORT RESPONSE FAILURE
872      000010      X3.DCK = BIT3      ;DENSITY CHECK
873      000006      X3.MBZ = BIT2+BIT1  ;NOT USED ALWAYS 0
874      000001      X3.RIB = BIT0      ;REVERSE INTO BOT
875
876      ;+
877      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
878      ;(XST4)
879      ;-
880      100000      X4.HSP = BIT15      ;HIGH SPEED
881      040000      X4.RCE = BIT14      ;RETRY COUNT EXCEEDED
882      020000      X4.TSM = BIT13      ;TRANSPORT SPECIAL MODE
883      017400      X4.MBZ = BIT12+BIT11+BIT10+BIT9+BIT8 ;NOT USED ALWAYS 0
884      000377      X4.WRC = 000377    ;WRITE RETRY COUNT FIELD
885
886
887      ;+
888      ;
889      ;TSSR TERMINATION CODES (BIT 0-2)
890      ;
891      ;-
892
893      000006      TSREJ= 3+2          ;COMMAND REJECTED
894      000006      UNREC= 6           ;UNRECOVERABLE ERROR
895
896      ;+
897      ;
898      ;DEVICE REGISTER OFFSETS
899      ;

```

```

900      ;
901      ;
902      177776      TSBA== -2
903      177776      TSBAL== -2
904      177776      TSDB== -2      ;TSDB/TSBA REGISTER
905      177776      TSDBL== -2      ;TSDB/TSBA REGISTER
906      177777      TSBAH== -1
907      177777      TSDBH== -1      ;TSDB/TSBA REGISTER HIGH BYTE
908      000000      TSSR== 0      ;TSSR REGISTER
909      000001      TSSRH== 1      ;TSSR REGISTER HIGH BYTE
910
911      ;+
912      ; TSDB ADDRESS BIT DEFINITIONS
913      ;
914      000003      A1716 = BIT1+BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
915
916      ;+
917      ; COMMAND DEFINITIONS
918      ;
919      000017      P.GETSTAT = 17      ;GET STATUS
920      000013      P.INIT = 13      ;INITIALIZE
921      000012      P.CONTROL = 12      ;CONTROL COMMANDS
922      000011      P.FORMAT = 11      ;FORMAT
923      000010      P.POSITION = 10      ;POSITION
924      000006      P.WRITSUB = 6      ;SUBSYSTEM WRITE
925      000005      P.WRITE = 5      ;WRITE
926      000004      P.WRCHAR = 4      ;WRITE CHARACTERISTICS
927      000001      P.READ = 1      ;READ
928
929      ;+
930      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
931      ;
932      100000      P.ACK = BIT15      ;BUFFER AVAIL FOR CONTROLLER
933      040000      P.CVC = BIT14      ;CLEAR VOLUME CHECK
934      020000      P.OPP = BIT13      ;REVERSE SEQUENCE OF DATA BITS
935      010000      P.SWB = BIT12      ;SWAP BYTES IN MEMORY
936      007400      P.MODE = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
937      000200      P.IE = BIT7      ;INTERRUPT ENABLE
938      000140      P.FMT = BIT6!BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
939      00003      P.CMD = 37      ;MAJOR COMMAND FIELD
940
941      ;+
942      ; CONTROL COMMAND MODE CODES
943      ;
944      000000      PC.RELEASE = 0*256. ;RELEASE BUFFER
945      000400      PC.REWIND = 1*256. ;REWIND
946      001000      PC.NOOP = 2*256. ;NO-OP
947      002000      PC.IEREW = 4*256. ;REWIND IMMEDIATE INTERRUPT
948      002400      PC.ERASE = 5*256. ;SECURITY ERASE
949
950      ;+
951      ; CONTROLLER RAM DEFINITIONS
952      ;
953      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
954      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
955      000020      RMPKTBEG = 20      ;COMMAND PACKET BEGIN RAM ADDRESS
956      000027      RMPKTEND = 27      ;COMMAND PACKET END RAM ADDRESS
957      000104      RMMSGBEG = 104      ;MESSAGE BUFFER BEGIN RAM ADDRESS

```

G3

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 27-3  
TK-25 REGISTER AND PACKET DEFINITIONS

SEQ 32

```

957      000117      RMMSGEND= 117      ;MESSAGE BUFFER END RAM ADDRESS
958      ;+
959      ;
960      ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
961      ;
962      ;-
963
964      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
965      000010      XST1== 8.      ;EXTENDED STATUS REGISTER 1 (WORD 5)
966      000012      XST2== 10.      ;EXTENDED STATUS REGISTER 2 (WORD 6)
967      000014      XST3== 12.      ;EXTENDED STATUS REGISTER 3 (WORD 7)
968      000016      XST4== 14.      ;EXTENDED STATUS REGISTER 4 (WORD 8)
969
970
971      ;+
972      ;
973      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
974      ;
975      ;-
976
977      000002      PKLOW = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
978      000004      PKHI = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
979      000006      PKBCNT = 6      ;NUMBER OF BYTES IN DATA PACKET
980
981      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
982
983      ;+
984      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
985      ;-
986      000000      BSELO = 0      ;BYTE 0
987      000001      BSEL1 = 1      ;BYTE 1
988      000002      SEL2 = 2      ;WORD 2
989      000004      SELDATA = 4      ;WORD 3
990
991      ;+
992      ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
993      ;-
994      000000      PW.NOP = 0      ;NO-OP
995      000001      PW.RDRAM = 1      ;READ RAM
996      000002      PW.WTRAM = 2      ;WRITE RAM
997      000003      PW.RFIFO = 3      ;READ FIFO
998      000004      PW.WFIFO = 4      ;WRITE FIFO
999      000005      PW.RDSTAT = 5      ;READ STATUS
1000      000006      PW.WCTL = 6      ;WRITE TAPE CONTROL
1001      000007      PW.WFMT = 7      ;WRITE TAPE FORMAT
1002      000010      PW.WMISC = 10      ;WRITE MISCELLANEOUS
1003      000011      PW.WNPR = 11      ;WRITE NPR CONTROL
1004      000020      PW.D22 = 20      ;DO MICROTEST 22
1005      000021      PW.D11 = 21      ;DO MICROTEST 11
1006      000022      PW.D13 = 22      ;DO MICROTEST 13
1007      000023      PW.NO1311 = 23      ;DISABLE MICROTEST 11 AND 13
1008      000024      PW.RDEXT = 24      ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPQ
RTS
1009
1010      ;+
1011      ;BSEL1 CODES FOR WRITE TAPE CONTROL
1012      ;-
1013      000200      WC.IFAD = BIT7      ;IFAD - FORMATTER ADDRESS

```

H3

CZTKEA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 27-4  
TK-25 REGISTER AND PACKET DEFINITIONS

SEQ 33

|      |        |                                                    |             |                                                |
|------|--------|----------------------------------------------------|-------------|------------------------------------------------|
| 1014 | 000100 | WC.IOTAD                                           | = BIT6      | ;ITADO - TRANSPORT ADDRESS BIT 0               |
| 1015 | 000040 | WC.I1TAD                                           | = BIT5      | ;ITAD1 - TRANSPORT ADDRESS BIT 1               |
| 1016 | 000020 | WC.I5RESV                                          | = BIT4      | ;IRESV5 - RESERVED #5                          |
| 1017 | 000010 | WC.IREW                                            | = BIT3      | ;IREW - REWIND                                 |
| 1018 | 000004 | WC.IRWU                                            | = BIT2      | ;IRWU - REWIND AND UNLOAD                      |
| 1019 | 000002 | WC.IFEN                                            | = BIT1      | ;IFEN - FORMATTER ENABLE                       |
| 1020 | 000001 | WC.IGO                                             | = BIT0      | ;GO                                            |
| 1021 |        |                                                    |             |                                                |
| 1022 |        |                                                    |             |                                                |
| 1023 |        | ;+ BSEL1 CODES FOR WRITE FORMAT                    |             |                                                |
| 1024 |        |                                                    |             |                                                |
| 1025 | 000200 | WF.IHISP                                           | = BIT7      | ;IHISP - HIGH SPEED                            |
| 1026 | 000100 | WF.IWRT                                            | = BIT6      | ;IWRT - WRITE                                  |
| 1027 | 000040 | WF.IREV                                            | = BIT5      | ;IREV - REVERSE                                |
| 1028 | 000020 | WF.IWFM                                            | = BIT4      | ;IWFM - WRITE FILE MARK                        |
| 1029 | 000010 | WF.IEDIT                                           | = BIT3      | ;IEDIT - EDIT                                  |
| 1030 | 000004 | WF.IERASE                                          | = BIT2      | ;IERASE - ERASE                                |
| 1031 | 000002 | WF.I3RESV                                          | = BIT1      | ;IRESV3 - RESERVED #3                          |
| 1032 | 000001 | WF.I4RESV                                          | = BIT0      | ;IRESV4 - RESERVED #4                          |
| 1033 |        |                                                    |             |                                                |
| 1034 |        |                                                    |             |                                                |
| 1035 |        |                                                    |             |                                                |
| 1036 |        | ;+ BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND  |             |                                                |
| 1037 |        |                                                    |             |                                                |
| 1038 | 000200 | MS.EXT                                             | = BIT7      | ;INVERT SENSE OF EXTENDED FEATURES SWITCH      |
| 1039 | 000020 | MS.RSFIFO                                          | = BIT4      | ;RESET FIFO AND INPUT PARITY ERRORR            |
| 1040 | 000010 | MS.RSTAPE                                          | = BIT3      | ;RESET TAPE STATUS IN 2 FLIP-FLOPS             |
| 1041 | 000006 | MS.ATTN                                            | = BIT2!BIT1 | ;ATTENTION TRIGGER FIELD                       |
| 1042 | 000001 | MS.RSD                                             | = BIT0      | ;RESET TIMER A,B THEN DELAY TIMES IN SEL2      |
| 1043 |        |                                                    |             |                                                |
| 1044 |        | ;+ MS.ATTN SUBCODES                                |             |                                                |
| 1045 |        |                                                    |             |                                                |
| 1046 | 000000 | MSA.NOP                                            | = 0*2       | ;NO-OP (NOTHING TRIGGERED)                     |
| 1047 | 000002 | MSA.VOL                                            | = 1*2       | ;SIMULATE ON-LINE/OFF-LINE TRANSITION          |
| 1048 | 000004 | MSA.NRAM                                           | = 2*2       | ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54) |
| 1049 | 000006 | MSA.FRAME                                          | = 3*2       | ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)     |
| 1050 |        |                                                    |             |                                                |
| 1051 |        | ;+ WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS |             |                                                |
| 1052 |        |                                                    |             |                                                |
| 1053 | 000200 | NP.IR                                              | = BIT7      | ;INTERRUPT REQUEST (0-1 TRANSITION)            |
| 1054 | 000100 | NP.OUT                                             | = BIT6      | ;TAPE DATA DIRECTION OUT (0= IN)               |
| 1055 | 000040 | NP.LOOP                                            | = BIT5      | ;ENABLE TRANSPORT LOOPBACK                     |
| 1056 | 000020 | NP.WRP                                             | = BIT4      | ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)   |
| 1057 |        |                                                    |             |                                                |
| 1058 |        | ;+ READ STATUS MESSAGE BUFFER BIT DEFINITIONS      |             |                                                |
| 1059 |        |                                                    |             |                                                |
| 1060 |        |                                                    |             |                                                |
| 1061 | 000200 | S2.DIM                                             | = BIT7      | ;WORD #9 BYTE 2 DATA IN MISS                   |
| 1062 | 000100 | S2.ILW                                             | = BIT6      | ;ILW H                                         |
| 1063 | 000040 | S2.OUTRDY                                          | = BIT5      | ;OUT RDY H                                     |
| 1064 | 000020 | S2.INRDY                                           | = BIT4      | ;IN RDY H                                      |
| 1065 | 000010 | S2.ATIMR                                           | = BIT3      | ;TIMER A FLAG H                                |
| 1066 | 000004 | S2.BTIMR                                           | = BIT2      | ;TIMER B FLAG H                                |
| 1067 | 000003 | S2.UNDEF                                           | = BIT1,BIT0 | ;(UNDEFINED)                                   |
| 1068 | 100000 | S1.PARIN                                           | = BIT15     | ;WORD #8 BYTE 1 PARIN H                        |
| 1069 | 040000 | S1.I2RESV                                          | = BIT14     | ;IRESV2                                        |
| 1070 | 020000 | S1.I1RESV                                          | = BIT13     | ;IRESV1                                        |

|      |        |           |                                          |   |                                  |
|------|--------|-----------|------------------------------------------|---|----------------------------------|
| 1071 | 010000 | S1.IEOT   | = BIT12                                  | : | IEGT L                           |
| 1072 | 004000 | S1.IIDENT | = BIT11                                  | : | IIDENT H                         |
| 1073 | 002000 | S1.ICER   | = BIT10                                  | : | ICER H                           |
| 1074 | 001000 | S1.IFMK   | = BIT9                                   | : | IFMK H                           |
| 1075 | 000400 | S1.IHER   | = BIT8                                   | : | IHER H                           |
| 1076 | 000200 | S0.ISPEED | = BIT7                                   | : | WORD #8 BYTE 0 ISPEED H          |
| 1077 | 000100 | S0.IRDY   | = BIT6                                   | : | IRDY L                           |
| 1078 | 000040 | S0.IONL   | = BIT5                                   | : | IONL L                           |
| 1079 | 000020 | S0.ILDP   | = BIT4                                   | : | ILDP L                           |
| 1080 | 000010 | S0.IDBY   | = BIT3                                   | : | IDBY L                           |
| 1081 | 000004 | S0.IRWD   | = BIT2                                   | : | IRWD L                           |
| 1082 | 000002 | S0.IFBY   | = BIT1                                   | : | IFBY L                           |
| 1083 | 000001 | S0.IFPT   | = BIT0                                   | : | IFPT L                           |
| 1084 |        | :         |                                          | : |                                  |
| 1085 |        | :         | SPECIAL KEYBOARD STUFF FOR MOVER PROGRAM | : |                                  |
| 1086 | 177560 | TKS       | =177560                                  | : | ;KEYBOARD STATUS REGISTER        |
| 1087 | 177562 | TKB       | =177562                                  | : | ;KEYBOARD DATA REGISTER          |
| 1088 | 177564 | TPS       | =177564                                  | : | ;CONSOLE PRINTER STATUS REGISTER |
| 1089 | 177566 | TPB       | =177566                                  | : | ;CONSOLE PRINTER DATA REGISTER   |
| 1090 | 007776 | HIMEM     | =007776                                  | : | ;HIGH MEMORY MASK VALUE          |
| 1091 |        | :         | CONTROLLER DEFINITIONS                   | : |                                  |
| 1092 |        | :         |                                          | : |                                  |
| 1093 |        | :         |                                          | : |                                  |
| 1094 | 174400 | CSR       | =174400                                  | : | ;STATUS AND CONTROL REGISTER     |
| 1095 | 174402 | BAR       | =174402                                  | : | ;DL ADDRESS REGISTER             |
| 1096 | 174404 | DAR       | =174404                                  | : | ;PLATTER ADDRESS                 |
| 1097 | 174406 | MPR       | =174406                                  | : | ;MULTIPURPOSE REGISTER           |
| 1098 |        | :         |                                          | : |                                  |
| 1099 |        | :         |                                          | : |                                  |
| 1100 |        | :         |                                          | : |                                  |
| 1101 |        | :         |                                          | : |                                  |
| 1102 |        | :         |                                          | : |                                  |
| 1103 |        | :         | CONTROLLER COMMANDS                      | : |                                  |
| 1104 |        | :         |                                          | : |                                  |
| 1105 |        | :         |                                          | : |                                  |
| 1106 | 000004 | DLGETS    | =4                                       | : | ;GET STATUS COMMAND              |
| 1107 | 000006 | SEEK      | =6                                       | : | ;SEEK TRACK AND HEAD SELECT      |
| 1108 | 000010 | DLRHD     | =10                                      | : | ;READ SECTOR HEADER              |
| 1109 | 000014 | READ      | =14                                      | : | ;READ COMMAND                    |
| 1110 | 000016 | DLRDNH    | =16                                      | : | ;READ SECTOR NO HEADER CHECK     |
| 1111 |        | :         |                                          | : |                                  |
| 1112 |        | :         |                                          | : |                                  |
| 1113 |        | :         |                                          | : |                                  |
| 1114 |        | :         |                                          | : |                                  |
| 1115 |        | :         |                                          | : |                                  |
| 1116 |        | :         |                                          | : |                                  |
| 1117 | 000001 | READY     | =1                                       | : | ;DRIVE READY BIT IN STATUS REG.  |
| 1118 | 000013 | DLSR      | =13                                      | : | ;STATUS AND RESET                |
| 1119 | 177730 | DLERR     | =177730                                  | : | ;MASK FOR COVER OPEN             |
| 1120 | 000006 | DLUN      | =6                                       | : | ;HEADS UNLOADED                  |
| 1121 | 000177 | DLCYL     | =000177                                  | : | ;MASK FOR CYLINDER ADDRESS       |
| 1122 | 100200 | DLDNR     | =100200                                  | : | ;DONE SET OR ERROR SET BITS      |
| 1123 |        | :         |                                          | : |                                  |
| 1124 |        | :         |                                          | : |                                  |
| 1125 |        | :         |                                          | : |                                  |
| 1126 |        | :         |                                          | : |                                  |
| 1127 | 177560 | TTICSR    | = 177560                                 | : | ;KEYBOARD INPUT STATUS           |

J3

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 27-6  
TK-25 REGISTER AND PACKET DEFINITIONS

SEQ 35

1128 177562  
1129 177564  
1130 177566  
1131

TTIBFR = 177562  
TTOCSR = 177564  
TTOBFR = 177566

;KEYBOARD DATA REGISTER  
;CONSOLE PRINTER STATUS REGISTER  
;CONSOLE PRINTER DATA REGISTER

```
1133          .SBTTL  SPECIAL MACROS AND OPDEFS.
1134
1135
1136      ;+
1137      ;SAVE GENERAL REGS 1 TO 5
1138      ;-
1139
1140          .MACRO  SAVREG
1141          JSR     R5,REGSAV
1142          .ENDM
1143
1144      ;+
1145      ; MACRO TO FORCE AN ERROR
1146      ;-
1147      .MACRO  FORCEERROR      TAG,NOTSSR
1148      .NLIST
1149      .IIF NDF LISTALL, .NLIST
1150      .LIST
1151      .IF B NOTSSR
1152          MOV     TSSR(R5),R1          ;READ TSSR
1153      .ENDC
1154          MOV     FORCER,FORCER        ;IS FORCER SET? (LEAVE C BIT ALONE)
1155          BNE     TAG                  ;BR IF YES
1156      .NLIST
1157      .IIF NDF LISTALL, .LIST
1158      .LIST
1159      .ENDM
1160
1161      ;+
1162      ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1163      ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1164      ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1165      ; FORCER TO 177777
1166      ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1167      ;-
1168      .MACRO  FORCEEXIT      TAG
1169      .NLIST
1170      .IIF NDF LISTALL, .NLIST
1171      .LIST
1172          MOV     FORCER,FORCER        ;IS FORCER NEGATIVE?
1173          BMI     TAG                  ;BR IF YES
1174      .NLIST
1175      .IIF NDF LISTALL, .LIST
1176      .LIST
1177      .ENDM
1178      ;+
1179      ; MACRO TO INCREMENT ERROR COUNTS
1180      ;-
1181      .MACRO  NEXT,ERRNO
1182      .NLIST
1183      .IIF NDF LISTALL, .NLIST
1184          ERRNO=ERRNO+1
1185      .IIF NDF LISTALL, .LIST
1186      .LIST
1187      .ENDM
1188
1189      ;+
```

L3

CZTKFA TK-25 FRT END FUNC #2  
SPECIAL MACROS AND OPDEFS.

MACRO M1200 20-APR-84 08:12 PAGE 28-1

SEQ 37

```

1190          ;MACRO TO PERFORM XOR
1191          ; -
1192
1193          .MACRO XOR      A,B
1194          MOV      A,-(SP)
1195          BIC      B,(SP)
1196          BIC      A,B
1197          BIS      (SP)+,B
1198          .ENDM
1199
1200          000000          EN=0          ; INITIALIZE ERROR NUMBER
1201          .SBTTL  FORCER  - FORCE ERROR FLAG
1202
1203          ;
1204          ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
1205          ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
1206          ;
1207
1208 002144 000000  FORCER::      0          ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
1209                                ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
1210                                ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
1211
1212
1213

```

1215 .SBTTL GLOBAL DATA SECTION

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226 002146 000000

1227 002150 000000

1228 002152 000000

1229 002154 000000

1230 002156 000224

1231 002160 000200

1232 002162 000000

1233 002164 000000

1234 002166 000000

1235 002170 000000

1236 002172 000000

1237 002174 000000

1238 002176 000000

1239 002200 000000

1240 002202 000000

1241 002204 000000

1242 002206 000000

1243 002246 000000

1244 002250 000000

1245 002252 000000

1246 002254 000000

1247 002256 000000

1248 002260 000000

1249 002262 000000

1250 002264 000000

1251 002266

1252 002432

1253 002576

1254 002716 000000

\*\*\*

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

---

;

; THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.  
; SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.

;

|            |       |       |                                                 |
|------------|-------|-------|-------------------------------------------------|
| EPPTSW::   | .WORD | 0     | ; PRINT SWITCH                                  |
| UNITN::    | .WORD | 0     | ; UNIT # UNDER TEST.                            |
| QVP::      | .WORD | 0     | ; QUICK VERIFY FLAG.                            |
| CSRADDR::  | .WORD | 0     | ; ADDRESS OF CSR FOR CURRENT DEVICE             |
| IVEC::     | .WORD | 224   | ; INTERRUPT VECTOR                              |
| IPRI::     | .WORD | PRI04 | ; INTERRUPT PRIORITY.                           |
| TSTCNT::   | .WORD | 0     | ; NUMBER OF TESTS RUN IN THIS PASS              |
| LOOPCNT::  | .WORD | 0     | ; REMAINING ITERATION COUNT FOR TEST            |
| DEVCNT::   | .WORD | 0     | ; NUMBER OF DEVICE UNDER TEST                   |
| FATFLG::   | .WORD | 0     | ; SET IF FATAL ERROR IS DETECTED IN TEST        |
| INTRECV::  | .WORD | 0     | ; SET IF TAPE INTERRUPT WAS RECEIVED            |
| BENBSW::   | .WORD | 0     | ; BUFFER ENABLE SWITCH SW 0=OFF;1=ON            |
| EXPD::     | .WORD | 0     | ; EXPECTED RAM DATA FOR PRAMPKT ROUTINE         |
| RECV::     | .WORD | 0     | ; RECEIVED RAM DATA FOR PRAMPKT ROUTINE         |
| ERRHI::    | .WORD | 0     | ; HIGH ADDRESS MEMORY ERROR                     |
| ERRLO::    | .WORD | 0     | ; LOW ADDRESS MEMORY ERROR                      |
| RAMDATA::  | .BLKW | 16.   | ; DATA READ FROM RAM PACKET OR MESSAGE BUF AREA |
| RAMSIZ::   | .WORD | 0     | ; RAM DATA SIZE FOR PRAMPKT ROUTINE             |
| RCVHIADD:: | .WORD | 0     | ; RECEIVED BUFFER HIGH ADDRESS                  |
| RCVLOADD:: | .WORD | 0     | ; RECEIVED BUFFER LOW ADDRESS                   |
| COUNT::    | .WORD | 0     | ; TEST COUNT PATTERN                            |
| DATA::     | .WORD | 0     | ; TEST DATA                                     |
| TSTFLAG::  | .WORD | 0     | ; TEST FLAG WORD                                |
| TSTPTR::   | .WORD | 0     | ; TSTBLK POINTER                                |
| PRMNO::    | .WORD | 0     | ; PRINT ROUTINE TEMP                            |
| EXPMSG::   | .BLKB | 100.  | ; EXPECTED MESSAGE BUFFER DATA                  |
| RECMMSG::  | .BLKB | 100.  | ; RECEIVED MESSAGE BUFFER DATA                  |
| TMPBFR::   | .BLKB | 80.   | ; TEMPORARY STORAGE FOR PRINT                   |
| MESBFA::   | .WORD | 0     | ; STORES ADDRESS OF MESSAGE BUFFER FOR ERR PRT  |

```

1256          .SBTTL  TSTBLK  - TEST DATA TABLE
1257
1258          ;+
1259          ; THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
1260          ;
1261          ; IN SEQUENCE THE DATA IS:
1262          ;
1263          ;
1264          ;     ALL ZEROS
1265          ;     ALL ONES
1266          ;     WALKING ONES
1267          ;     WALKING ZEROS
1268          ;     ALTERNATING ONES AND ZEROS
1269          ;
1270          ; -
1271
1272 002720
1273 002720 000000
1274 002722 177777
1275 002724 000001
1276 002726 000002
1277 002730 000004
1278 002732 000010
1279 002734 000020
1280 002736 000040
1281 002740 000100
1282 002742 000200
1283 002744 000400
1284 002746 001000
1285 002750 002000
1286 002752 004000
1287 002754 010000
1288 002756 020000
1289 002760 040000
1290 002762 100000
1291 002764 177776
1292 002766 177775
1293 002770 177773
1294 002772 177767
1295 002774 177757
1296 002776 177737
1297 003000 177677
1298 003002 177577
1299 003004 177377
1300 003006 176777
1301 003010 175777
1302 003012 173777
1303 003014 167777
1304 003016 157777
1305 003020 137777
1306 003022 077777
1307 003024 125252
1308 003026 052525
1309          003030

          TSTBLK::
          .WORD  0                      ;ALL ZEROS
          .WORD  177777                  ;ALL ONES
          .WORD  BIT0                     ;DATA FOR WALKING ONES
          .WORD  BIT1
          .WORD  BIT2
          .WORD  BIT3
          .WORD  BIT4
          .WORD  BIT5
          .WORD  BIT6
          .WORD  BIT7
          .WORD  BIT8
          .WORD  BIT9
          .WORD  BIT10
          .WORD  BIT11
          .WORD  BIT12
          .WORD  BIT13
          .WORD  BIT14
          .WORD  BIT15
          .WORD  +CBIT0                   ;DATA FOR WALKING ZEROS
          .WORD  +CBIT1
          .WORD  +CBIT2
          .WORD  +CBIT3
          .WORD  +CBIT4
          .WORD  +CBIT5
          .WORD  +CBIT6
          .WORD  +CBIT7
          .WORD  +CBIT8
          .WORD  +CBIT9
          .WORD  +CBIT10
          .WORD  +CBIT11
          .WORD  +CBIT12
          .WORD  +CBIT13
          .WORD  +CBIT14
          .WORD  +CBIT15
          .WORD  125252                   ;ALTERNATING ONES, ZEROS
          .WORD  052525                   ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE

          TBLEND**

```

```

1311          .SBTTL GLOBAL ENVIRONMENT STORAGE
1312
1313          ; STORAGE FOR DEVICE REGISTERS
1314
1315 003030 000000 100000 000000 DUMMY 0,100000,0,0          ; DUMMY DEVICE REGISTERS...
1316 003040 000000 000000 000000      0,0,0,0,0,0,0,0,0    ; ...FOR MULTI-UNIT CHECKOUT.
1317
1318
1319
1320 003060 000000          DUFLG::      .WORD 0          ; "DROPPED UNIT" FLAG.
1321          ; INHIBITS CODE IN "CLEAN-UP".
1322 003062 000000          NODEV::      .WORD 0          ; FLAG TO SAY NO DEVICE.
1323
1324 003064 000000          TEMP1::      .WORD 0          ; SOME TEMP LOCATIONS.
1325 003066 000000          TEMP2::      .WORD 0
1326 003070 000000          XXCOMM::     .WORD 0          ; XXDP, COMM BLOCK POINTER.
1327 003072 000000          FREE::       .WORD 0          ; 1ST FREE MEMORY ADDRESS...
1328 003074 000000          FRESIZ::     .WORD 0          ; ...AND SIZE (IN WORDS).
1329 003076 000000          FREEHI::     .WORD 0          ; LAST WORD IN FREE SPACE
1330 003100 000000          KTFLG::      .WORD 0          ; KT11, MEM AVAIL FLAG
1331          ; - .WORD 0 = <24K OR NO KT
1332          ; - NZ = >24K AND KT.
1333 003102 000000          KIENABLE::     .WORD 0          ; SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
1334 003104 002000          PST32W::      .WORD 2000      ; 32W BLOCK ADDRESS FOR 32K START
1335 003106 000000          SIFLAG::      .WORD 0
1336 003110 000000          BADDAT::      .WORD 0          ; ACTUAL DATA
1337 003112 000000          GDDAT::       .WORD 0          ; EXPECTED DATA
1338 003114 000000          LOOPFL::      .WORD 0
1339 003116          CTAB::               .WORD 0          ; CONFIGURATION TABLES.
1340 003116 000000          CTABM::       .WORD 0          ; CONFIG WORK.
1341 003120 000000          .WORD 0
1342 003122 000000          .WORD 0
1343 003124 000000          .WORD 0
1344 003126 177777          .WORD -1          ; END OF MEM TABLE.
1345 003130          CTABE::
1346          ; ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX;
1347          ;
1348          ; 0 = UNIT NOT TESTED
1349          ; 100000 = UNIT ONLINE, NO ERRORS
1350          ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
1351          ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
1352          ; 160001 = UNIT DROPPED, NOT IDLE AT START
1353          ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
1354          ;
1355 003130          ERTABL:               .BLKW 64,
1356 003330 000000          ERTABE:       .WORD 0
1357
1358 003332 000000          SKIPT: .WORD 0          ; 1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

```

C4

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 32

SEQ 41

```

1360 .SBTTL GLOBAL TEXT MESSAGES
1361
1362 ;++
1363 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1364 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1365 ; MORE THAN ONE TEST.
1366 ;--
1367
1368 ;+
1369 ; NAMES OF DEVICES SUPPORTED
1370 ;-
1371
1372
1373 003334 DEVTYP <TK-25>
003334 L#DVTYP:
003334 .ASCIZ /TK-25/
      124 113 055 .EVEN
1374
1375 ;+
1376 ; TEST DESCRIPTION
1377 ;-
1378 003342 DESCRIPT <CZTKFA TK-25 FRT END FUNC #2>
003342 L#DESC:
003342 .ASCIZ /CZTKFA TK-25 FRT END FUNC #2/
      103 132 124 .EVEN
1379
1380 ;+
1381 ; BIT TO ASCII CONVERSION FOR TSSR REGISTER
1382 ;-
1383 003400 003440 003443 003447 TSSRBIT: .WORD 1#,2#,3#,4#,5#,6#,7#,8#
1384 003420 003501 003505 003511 .WORD 9#,10#,11#,12#,13#,14#,15#,16#
1385 003440 123 103 000 1#: .ASCIZ 'SC'
1386 003443 102 111 105 2#: .ASCIZ 'BIE'
1387 003447 123 103 105 3#: .ASCIZ 'SCE'
1388 003453 122 115 122 4#: .ASCIZ 'RMR'
1389 003457 116 130 115 5#: .ASCIZ 'NXM'
1390 003463 116 102 101 6#: .ASCIZ 'NBA'
1391 003467 102 111 124 7#: .ASCIZ 'BIT9'
1392 003474 102 111 124 8#: .ASCIZ 'BIT8'
1393 003501 123 123 122 9#: .ASCIZ 'SSR'
1394 003505 117 106 114 10#: .ASCIZ 'OFL'
1395 003511 102 111 124 11#: .ASCIZ 'BIT5'
1396 003516 102 111 124 12#: .ASCIZ 'BIT4'
1397 003523 102 111 124 13#: .ASCIZ 'BIT3'
1398 003530 102 111 124 14#: .ASCIZ 'BIT2'
1399 003535 102 111 124 15#: .ASCIZ 'BIT1'
1400 003542 102 111 124 16#: .ASCIZ 'BIT0'
1401 .EVEN
1402 003550 124 123 123 SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
1403 003603 124 123 123 SFMERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
1404 003636 040 040 116 NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
1405 003675 045 101 040 NXR: .ASCIZ /#A ADDRESS: #06#
1406 003716 045 101 040 TSSX: .ASCII /#A TSBA,TSSR EXP'D: #06#A,#06#N/
1407 003756 045 101 040 TSSX: .ASCIZ /#A TSBA,TSSR REC'D: #06#A,#06#N/
1408 004015 045 116 045 FUSI: .ASCII /#N#A/
1409 004021 040 040 125 USI: .ASCIZ / UNEXPECTED INTERRUPT/
1410 004050 040 040 111 NSI: .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/

```

D4

CZIKFA IK-25 FBI END FUNC 02  
GLOBAL TEXT MESSAGES

MACRO M1200 20-APR-84 08:12 PAGE 32-1

SEQ 42

|      |        |     |     |     |          |                                                                             |
|------|--------|-----|-----|-----|----------|-----------------------------------------------------------------------------|
| 1411 | 004113 | 045 | 116 | 045 | FNOINTR: | .ASCII /#N#A/                                                               |
| 1412 | 004117 | 040 | 040 | 116 | NOINTR:  | .ASCIIZ / NO INTERRUPT WAS GENERATED/                                       |
| 1413 | 004154 | 040 | 040 | 111 | IFALT:   | .ASCIIZ / INTERRUPT FAULT/                                                  |
| 1414 | 004176 | 045 | 101 | 040 | INTX:    | .ASCIIZ /#A CPU PC: #06#A TSBA: #06/                                        |
| 1415 | 004233 | 040 | 040 | 042 | NOINIT:  | .ASCIIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/                          |
| 1416 | 004305 | 040 | 040 | 042 | NSINIT:  | .ASCIIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/                            |
| 1417 | 004355 | 040 | 040 | 042 | BRINIT:  | .ASCIIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/                            |
| 1418 |        |     |     |     |          |                                                                             |
| 1419 | 004425 | 000 |     |     | NUL:     | .ASCIIZ //                                                                  |
| 1420 | 004426 | 045 | 116 | 000 | NULCR:   | .ASCIIZ /#N/                                                                |
| 1421 | 004431 | 045 | 101 | 040 | EXPGOT:  | .ASCIIZ /#A EXP'D: #06#A, REC'D: #06/                                       |
| 1422 | 004465 | 045 | 116 | 045 | EXPGT2:  | .ASCIIZ /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/                       |
| 1423 | 004541 | 045 | 101 | 040 | DUAD12:  | .ASCIIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ, EXP'D: #06#A, REC'D: #06/ |
| 1424 | 004643 | 122 | 101 | 115 | PKTRAM:  | .ASCIIZ 'RAM Contents Do Not Match Packet Sent'                             |
| 1425 | 004711 | 040 | 040 | 103 | SCME:    | .ASCIIZ / CONFIG DOESN'T MATCH MFG. MASTER/                                 |
| 1426 | 004754 | 127 | 122 | 111 | WRTHSG:  | .ASCIIZ 'WRITE CHARACTERISTICS Failed'                                      |
| 1427 | 005011 | 124 | 123 | 123 | WTERR:   | .ASCIIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'        |
| 1428 | 005104 | 124 | 123 | 123 | RDEPR:   | .ASCIIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'         |
| 1429 |        |     |     |     |          | .EVEN                                                                       |
| 1430 |        |     |     |     |          |                                                                             |
| 1431 |        |     |     |     |          |                                                                             |
| 1432 |        |     |     |     |          |                                                                             |

1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442 005176  
005176  
1443 005176  
005176 013746 003062  
005202 012746 003675  
005206 012746 000002  
005212 010600  
005214 104415  
005216 062706 000006  
1444 005222 004737 005230  
1445 005226  
005226  
005226 104423  
1446  
1447  
1448  
1449  
1450  
1451  
1452 005230 005727  
1453 005232 000000  
1454 005234 001402  
1455 005236 004777 177770  
1456 005242  
005242 012746 004426  
005246 012746 000001  
005252 010600  
005254 104415  
005256 062706 000004  
1457 005262 000207

## .SBTTL GLOBAL ERROR REPORT SECTION

\*\*\*  
; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX  
; CALLS THAT ARE USED IN MORE THAN ONE TEST.  
; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.  
---

BGNMSG NXRRERR ;NON-EXISTANT DEVICE REGISTER.  
NXRRERR: ;  
PRINTX #NXRX,NODEV ;NODEV = NEXM ADDRESS.  
MOV NODEV,-(SP)  
MOV #NXRX,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C#PNTX  
ADD #6,SP  
JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.  
ENDMSG  
L10002:  
TRAP C#MSG

;  
; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)  
; TO ANY OF THE ABOVE ERROR SIGNATURES.  
;

EXTEND: TST (PC)+  
EXTA: 0 ; 0 = NO EXTENSION.  
BEQ 1#  
JSR PC,EXTA ; APPEND EXTENSION TEXT.  
1#: PRINTX #NULCR ; PRINT A BLANK LINE  
MOV #NULCR,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C#PNTX  
ADD #4,SP  
RTS PC

```

1460          .SBTTL  PRITSSR - PRINT TSSR CONTENTS
1461
1462          ;*
1463          ;
1464          ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
1465          ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
1466          ;BY A MESSAGE PRINTING ROUTINE
1467          ;
1468          ;INPUTS:
1469          ;
1470          ;      R1      CONTENTS OF TSSR
1471          ;
1472          ;SUBORDINATE ROUTINES:
1473          ;
1474          ;      CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
1475          ;
1476          ;-
1477
1478          PRITSSR:
1479          SAVREG          ;SAVE GENERAL REGISTERS
1480          MOV             R1,R4          ;SAVE THE TSSR CONTENTS
1481          PRINTB          ;PRINT THE CONTENTS OF TSSR
1482          MOV             R4,-(SP)
1483          MOV             #TSSRFOR,R4
1484          MOV             #2,-(SP)
1485          MOV             SP,R0
1486          TRAP            C#PNTB
1487          ADD             #6,SP
1488          MOV             R4,R0
1489          JSR             PC,CHKAMB
1490          BCS             5#
1491          PRINTX          #AMBTSSR
1492          MOV             #AMBTSSR,-(SP)
1493          MOV             #1,-(SP)
1494          MOV             SP,R0
1495          TRAP            C#PNTX
1496          ADD             #4,SP
1497          MOV             R4,R3
1498          PIC             #HIADDR!FATERR!TERCLS,R3
1499          BEQ             20#
1500          MOV             #TMPBFR,R2
1501          MOV             #TSSRBIT,R1
1502          TST             R3
1503          BEQ             10#
1504          RCU             15#
1505          CLC
1506          ROL             R3
1507          BCC             13#
1508          MOV             (R1),R0
1509          MOV             (R0),-(R2)+
1510          BNE             11#
1511          MOV             #'-1(R2)
1512          MOV             (R1)+
1513          TST             (R1)+
1514          BR              10#
1515          CLRB            -(R2)
1516          PRINTX          #TSSDEF,#TMPBFR
1517          MOV             #TMPBFR,-(SP)
1518          MOV             #TSSDEF,-(SP)
1519
1520          ;CONTENTS OF TSSR
1521          ;CLEAR ALL MULTIPLE BIT FIELDS
1522          ;NO BITS ARE SET
1523          ;TEMPORARY ASCII BUFFER
1524          ;ASCII EQUIVALENT OF BITS
1525          ;REMAINING BITS TO CONVERT
1526          ;BRANCH WHEN ALL ARE DONE
1527          ;CLEAR CARRY FOR SHIFT
1528          ;SHIFT NEXT BIT TO CARRY
1529          ;BRANCH IF BIT NOT SET
1530          ;POINTER TO BIT DEFINITION
1531          ;MOVE ASCII TO BUFFER
1532          ;MOVE ALL BITS
1533          ;INSERT A COMMA TO TERMINATE
1534          ;POINT TO NEXT DESCRIPTION
1535          ;GET THE REMAINING BITS
1536          ;TERMINATE THE LINE
1537          ;PRINT THE BIT DEFINITIONS

```

G4

CZTKFA TK-25 FRT END CONC #2  
PRITSSR - PRINT TSSR CONTENTS

MACRO M1200 20-APR-84 08:12 PAGE 35-1

SEQ 45

|        |        |        |                     |        |               |                                                           |
|--------|--------|--------|---------------------|--------|---------------|-----------------------------------------------------------|
| 005430 | 012746 | 000002 |                     | MOV    | #2,-(SP)      |                                                           |
| 005434 | 010600 |        |                     | MOV    | SP,R0         |                                                           |
| 005436 | 104415 |        |                     | TRAP   | C#PNTX        |                                                           |
| 005440 | 062706 | 000006 |                     | ADD    | #6,SP         |                                                           |
| 1504   |        |        |                     |        |               |                                                           |
| 1505   | 005444 | 010403 | 20#:                | MOV    | R4,R3         | ;GET THE TSSR CONTENTS                                    |
| 1506   | 005446 | 042703 |                     | BIC    | #+CTERCLS,R3  | ;CLEAR ALL BUT TERMINATION                                |
| 1507   | 005452 | 016303 |                     | MOV    | TCOCOD(R3),R3 | ;GET THE TERMINATION CODE MEANING                         |
| 1508   | 005456 |        |                     | PRINTX | #TCOASC,R3    | ;PRINT THE TERMINATION CODE                               |
|        | 005456 | 010346 |                     | MOV    | R3,-(SP)      |                                                           |
|        | 005460 | 012746 |                     | MOV    | #TCOASC,-(SP) |                                                           |
|        | 005464 | 012746 |                     | MOV    | #2,-(SP)      |                                                           |
|        | 005470 | 010600 |                     | MOV    | SP,R0         |                                                           |
|        | 005472 | 103415 |                     | TRAP   | C#PNTX        |                                                           |
|        | 005474 | 062706 |                     | ADD    | #6,SP         |                                                           |
| 1509   | 005500 | 010403 |                     | MOV    | R4,R3         | ;TSSR CONTENTS AGAIN                                      |
| 1510   | 005502 | 042703 |                     | BIC    | #+CFATERR,R3  | ;CLEAR ALL BUT FATAL TERMINATION                          |
| 1511   | 005506 | 001421 |                     | BEQ    | 25#           | ;DON'T PRINT IF ZERO                                      |
| 1512   | 005510 | 006203 |                     | ASR    | R3            |                                                           |
| 1513   | 005512 | 006203 |                     | ASR    | R3            |                                                           |
| 1514   | 005514 | 006203 |                     | ASR    | R3            |                                                           |
| 1515   | 005516 | 015303 |                     | MOV    | TSFCOD(R3),R3 | ;ALINE TERMINATION CODE FOR INDEX                         |
| 1516   | 005522 |        |                     | PRINTX | #TFCASC,R3    | ;GET THE FATAL TERMINATION CODE                           |
|        | 005522 | 010346 |                     | MOV    | R3,-(SP)      | ;PRINT THE FATAL TERMINATION CODE                         |
|        | 005524 | 012746 |                     | MOV    | #TFCASC,-(SP) |                                                           |
|        | 005530 | 012746 |                     | MOV    | #2,-(SP)      |                                                           |
|        | 005534 | 010600 |                     | MOV    | SP,R0         |                                                           |
|        | 005536 | 104415 |                     | TRAP   | C#PNTX        |                                                           |
|        | 005540 | 062706 |                     | ADD    | #6,SP         |                                                           |
| 1517   | 005544 | 012737 | 000031 002170       | MOV    | #25,FATFLG    | ;DROP THIS UNIT AFTER ERROR                               |
| 1518   | 005552 | 010403 | 25#:                | MOV    | R4,R3         | ;GET TSSR CONTENTS                                        |
| 1519   | 005554 | 042703 |                     | BIC    | #+CHIADDR,R3  | ;CLEAR ALL BUT EXTENDED ADDRESS                           |
| 1520   | 005560 | 001411 |                     | BEQ    | 30#           | ;DON'T PRINT IF ZERO                                      |
| 1521   | 005562 |        |                     | PRINTX | #TEXASC,R3    | ;PRINT THE EXTENDED ADDRESS BITS                          |
|        | 005562 | 010346 |                     | MOV    | R3,-(SP)      |                                                           |
|        | 005564 | 012746 |                     | MOV    | #TEXASC,-(SP) |                                                           |
|        | 005570 | 012746 |                     | MOV    | #2,-(SP)      |                                                           |
|        | 005574 | 010600 |                     | MOV    | SP,R0         |                                                           |
|        | 005576 | 104415 |                     | TRAP   | C#PNTX        |                                                           |
|        | 005600 | 062706 |                     | ADD    | #6,SP         |                                                           |
| 1522   | 005604 | 022704 | 000210 30#:         | CMP    | #100210,R4    | ;CHECK FOR MEDIA ERROR                                    |
| 1523   | 005610 | 001003 |                     | BNE    | 31#           | ;BR, IF PROBABLY NOT TAPE ERROR                           |
| 1524   | 005612 | 012737 | 006026 002146       | MOV    | #EPRT3,EPRTSW | ; "PROBABLY MEDIA RELETED ERROR - BAD TAPE"               |
| 1525   | 005620 | 005737 | 002146 31#:         | TST    | EPRTSW        | ;CHECK FOR THE SWITCH EMPTY                               |
| 1526   | 005624 | 001003 |                     | BNE    | 310#          | ;BR, IF SWITCH IS NOT EMPTY                               |
| 1527   | 005626 | 012737 | 005762 002146       | MOV    | #EPRT1,EPRTSW | ;SET SWITCH TO DEFAULT                                    |
| 1528   | 005634 | 013737 | 002146 005644 310#: | MOV    | EPRTSW,32#*2  | ;PUT REAL SWITCHABLE MESSAGE IN PLACE                     |
| 1529   | 005642 |        | 32#:                | PRINTB | #EPRT1        | ;PRINT THE ERROR MESSAGE                                  |
|        | 005642 | 012746 |                     | MOV    | #EPRT1,-(SP)  |                                                           |
|        | 005646 | 012746 |                     | MOV    | #1,-(SP)      |                                                           |
|        | 005652 | 010600 |                     | MOV    | SP,R0         |                                                           |
|        | 005654 | 104415 |                     | TRAP   | C#PNTB        |                                                           |
|        | 005656 | 062706 |                     | ADD    | #4,SP         |                                                           |
| 1530   | 005662 | 012737 | 005762 002146       | MOV    | #EPRT1,EPRTSW | ;RESET TO NORMAL ERROR POINTER                            |
| 1531   | 005670 | 000207 |                     | RTS    | PC            | ;RETURN TO CALLER                                         |
| 1532   | 005672 | 045    | 116 045             | EPRT2: | .ASCIZ        | '#NMA *****CHECK CONTROLLER, CABLES AND TRANSPORT*****#S' |
| 1533   | 005762 | 045    | 116 045             | EPRT1: | .ASCIZ        | '#NMA *****REPLACE CONTROLLER*****#S'                     |

```

1534 006026      045      116      045 EPRT3: .ASCIZ 'NNA *****POSSIBLE MEDIA RELATED ERROR - BAD TAPE*****S'
1535
1536 006117      045      116      045 TSSRFOR: .ASCIZ 'NNA TSSR = #06'
1537 006137      045      116      045 TEXASC: .ASCIZ 'NNA Extended Address Bits = #06'
1538 006200      045      116      045 TCOASC: .ASCIZ 'NNA Termination Class Code = #T'
1539 006211      045      116      045 TFCASC: .ASCIZ 'NNA Fatal Termination Class Code = #T'
1540 006310      045      116      045 TSSDEF: .ASCIZ 'NNA TSSR Bits Set: #T'
1541 006337      045      116      045 AMBTSSR: .ASCIZ 'NNA TSSR Contents Are Ambiguous'
1542
1543 006400 006420 006443 006471 TCUCOD: .WORD 1#,2#,3#,4#,5#,6#,7#,8#
1544 006420      116      157      162 1#: .ASCIZ 'Normal Termination'
1545 006443      124      145      162 2#: .ASCIZ 'Termination Condition'
1546 006471      124      141      160 3#: .ASCIZ 'Tape Status Alert'
1547 006513      106      165      156 4#: .ASCIZ 'Function Reject'
1548 006533      122      145      143 5#: .ASCIZ 'Recoverable Error - Tape Position One Record Down'
1549 006615      122      145      143 6#: .ASCIZ 'Recoverable Error - Tape Was Not Moved'
1550 006664      125      156      162 7#: .ASCIZ 'Unrecoverable Error'
1551 006710      106      141      164 8#: .ASCIZ 'Fatal Controller Error'
1552
1553
1554 006740 006750 007004 007015 TSFCOD: .WORD 1#,2#,3#,4#
1555 006750      111      156      164 1#: .ASCIZ 'Internal Diagnostic Failure'
1556 007004      120      145      163 2#: .ASCIZ 'Reserved'
1557 007015      102      165      163 3#: .ASCIZ 'Bus Interface or Sanity Check Error'
1558 007061      122      145      163 4#: .ASCIZ 'Reserved'
1559

```

```

1561 .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
1562
1563
1564 ; THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
1565 ; THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
1566
1567 ; INPUT:
1568
1569 ; R0      NUMBER OF WORDS IN PACKET
1570 ; R3      HIGH ORDER COMMAND PACKET ADDRESS
1571 ; R4      ADDRESS OF COMMAND PACKET
1572
1573 ; NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
1574
1575
1576 007072 PRIPKT::
1577 007072 SAVREG          ;SAVE THE REGISTERS
1578 007076 010005 MOV      R0,R5          ;SAVE NO. OF WORDS IN PACKET
1579 007100 005737 003102 TST      KTENABLE      ;ABOVE 28K UNDER TEST?
1580 007104 001001 BNE      10$          ;BR IF YES
1581 007106 005003 CLR      R3              ;SET HIGH ORDER ADDRESS TO 0
1582 007110 010301 10$: MOV      R3,R1          ;COPY HIGH ORDER ADDRESS
1583 007112 010400 MOV      R4,R0          ;GET LOWER ADDRESS
1584 007114 006100 ROL      R0              ;SHIFT BIT 15 INTO C BIT
1585 007116 006101 ROL      R1              ;AND INTO HIGH ORDER.
1586 007120 PRINTB    #PKTADD,R1,R4          ;PRINT PACKET ADDRESS
1587 007120 010445 MOV      R4,-(SP)
1588 007122 010146 MOV      R1,-(SP)
1589 007124 012746 007276 MOV      #PKTADD,-(SP)
1590 007130 012746 000003 MOV      #3,-(SP)
1591 007134 010600 MOV      SP,R0
1592 007136 104414 TRAP     C#PNTB
1593 007140 062706 030010 ADD      #10,SP
1594 007144 010300 15$: MOV      R1,R3          ;GET HIGH ORDER ADDRESS
1595 007146 001404 BEQ      20$          ;BR IF NOT ABOVE 28K.
1596 007150 010401 MOV      R1,R3          ;GET LOW ORDER ADDRESS
1597 007152 004757 020270 JSR      #MAP          ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
1598 007156 010004 MOV      R1,R4          ;GET RETURNED PAR6 ADDRESS BIAS
1599 007160 005001 20$: CLR      R1              ;SAVE WORD NUMBER
1600 007162 012402 25$: MOV      (R4),R2          ;GET PACKET CONTENTS
1601 007164 PRINTB    #PKTFRM,R1,R2          ;PRINT THE DATA
1602 007164 010275 MOV      R2,-(SP)
1603 007166 010146 MOV      R1,-(SP)
1604 007170 012746 007240 MOV      #PKTFRM,-(SP)
1605 007174 012746 000003 MOV      #3,-(SP)
1606 007200 010600 MOV      SP,R0
1607 007202 104414 TRAP     C#PNTB
1608 007204 062706 030010 ADD      #10,SP
1609 007210 005201 INC      R1              ;NEXT WORD NUMBER
1610 007212 020105 CMP      R1,R5          ;DONE ALL PACKET WORDS?
1611 007214 002762 BLT      25$          ;LOOP TILL ALL DONE
1612 007216 PRINTB    #PKTNEW          ;JUST A COUPLE NEW LINES
1613 007216 012746 007333 MOV      #PKTNEW,-(SP)
1614 007222 012746 000001 MOV      #1,-(SP)
1615 007226 010600 MOV      SP,R0
1616 007230 104414 TRAP     C#PNTB
1617 007232 062706 000004 ADD      #4,SP

```

J4

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 36-1  
PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

SEQ 48

|      |        |        |     | RTS | PC             | RETURN                      |
|------|--------|--------|-----|-----|----------------|-----------------------------|
| 1599 | 007236 | 000207 |     |     |                |                             |
| 1600 |        |        |     |     |                |                             |
| 1601 | 007240 | 045    | 116 | 045 | PKTFRM: .ASCIZ | 'NNA Packet Word 0D1A = 06' |
| 1602 | 007276 | 045    | 116 | 045 | PKTADD: .ASCIZ | 'NNA Packet Address = 0105' |
| 1603 |        |        |     |     |                |                             |
| 1604 | 007333 | 045    | 116 | 045 | PKTNEW: .ASCIZ | 'NNA '                      |
| 1605 |        |        |     |     | .EVEN          |                             |
| 1606 |        |        |     |     |                |                             |

```

1608 .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
1609
1610
1611
1612 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
1613 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
1614
1615 ;INPUTS:
1616
1617 ;      R1      RECEIVED DATA
1618 ;      R2      EXPECTED DATA
1619
1620 ;OUTPUT:
1621
1622 ;      R0      XOR OF EXPECTED/RECEIVED DATA
1623
1624
1625
1626 PRIBXOR::
1627     SAVREG                      ;SAVE THE REGISTERS
1628     MOV      R2,R3              ;EXPECTED DATA
1629     XOR      R1,R3              ;FORM THE EXCLUSIVE OR
1630     MOV      #C<377>,R0        ;BYTE MASK
1631     BIC      R0,R1              ;SAVE LOW BYTE RECV
1632     BIC      R0,R2              ;SAVE LOW BYTE EXPD
1633     BIC      R0,R3              ;SAVE LOW BYTE XOR
1634     PRINTB   #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
1635     MOV      R3,-(SP)
1636     MOV      R1,-(SP)
1637     MOV      R2,-(SP)
1638     MOV      #XORBFOR,-(SP)
1639     MOV      #4,-(SP)
1640     MOV      SP,R0
1641     TRAP     C#PNTB
1642     ADD      #12,SP
1643     MOV      R3,R0              ;R0 HAS XOR ON RETURN
1644     RTS      PC                ;RETURN TO CALLER
1645
1646     .ASCIZ   '###A EXPD: #03#A RECV: #03#A XOR: #03'
1647     ,EVEN

```

```

1642                                .SBTTL  PRI XOR  - PRINT EXPD, RECV AND XOR
1643
1644                                ;+
1645                                ;
1646                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
1647                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINE
1648                                ;
1649                                ;INPUTS:
1650                                ;
1651                                ;          R1      RECEIVED DATA
1652                                ;          R2      EXPECTED DATA
1653                                ;
1654                                ;OUTPUT:
1655                                ;
1656                                ;          R0      XOR OF EXPECTED/RECEIVED DATA
1657                                ;
1658                                ;-
1659
1660 007474                                PRI XOR::
1661 007474                                SAVREG                                ;SAVE THE REGISTERS
1662 007500 010203                                MOV      R2,R3                                ;EXPECTED DATA
1663 007502                                XOR      R1,R3                                ;FORM THE EXCLUSIVE OR
1664 007512                                PRINTB   XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
1665                                MOV      R3,-(SP)
1666                                MOV      R1,-(SP)
1667                                MOV      R2,-(SP)
1668                                MOV      XORFOR,-(SP)
1669                                MOV      #4,-(SP)
1670                                MOV      SP,R0
1671                                TRAP     C$PNTB
1672                                ADD      #12,SP
1673                                MOV      R3,R0                                ;R0 HAS XOR ON RETURN
1674                                RTS      PC                                ;RETURN TO CALLER
1675
1676 007544 045 116 045 XORFOR: .ASCIZ 'NNNA EXPD: #06NA RECV: #06NA XOR: #06'
1677                                .EVEN

```

```

1671 .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
1672
1673 ;+
1674 ;
1675 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
1676 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1677 ;
1678 ;INPUTS:
1679 ;
1680 ;      RO      OCTAL VALUE TO CONVERT
1681 ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
1682 ;
1683 ;-
1684
1685 007612 PRIEQU:
1686 007612     SAVREG                                ;SAVE THE REGISTERS
1687 007616 000207     RTS      PC                    ;RETURN TO CALLER
1688
1689
1690
1691 .SBTTL PRIRAM - PRINT RAM ADDRESS
1692
1693 ;+
1694 ;
1695 ;PRINT CONTROLLER RAM ADDRESS.
1696 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1697 ;
1698 ;INPUTS:
1699 ;
1700 ;      R4      RAM ADDRESS
1701 ;
1702 ;-
1703 PRIRAM:
1704 007620     SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
1705 007624     PRINTB  RAMFOR,R4                    ;PRINT RAM ADDRESS IN ERROR
1706 007624 010446     MOV      R4,-(SP)
1707 007626 012746 007650     MOV      RAMFOR,-(SP)
1708 007632 012746 000002     MOV      #2,-(SP)
1709 007636 010600     MOV      SP,R0
1710 007640 104414     TRAP     C#PNTB
1711 007642 062706 000006     ADD      #6,SP
1712 007646 000207     RTS      PC                    ;RETURN
1713
1714 007650 045 116 045 RAMFOR: .ASCIZ 'NOA CONTROLLER RAM ADDRESS = #06'
1715 .EVEN
1716
1717 .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
1718
1719 ;+
1720 ;
1721 ;PRINT MEMORY ADDRESS
;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
;
; IMPLICIT INPUTS
;
;      ERRHI    - HIGH ORDER ADDRESS
;      ERRLO    - LOW ORDER ADDRESS

```

N4

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 39-1  
PRIADD - PRINT MEMORY ERROR ADDRESS

SEQ 52

```

1722
1723
1724 007712
1725 007712
1726 007716 013700 002202
1727 007722 013701 002204
1728 007726 010102
1729 007730 006101
1730 007732 006100
1731 007734
      007734 010246
      007736 010046
      007740 012746 007762
      007744 012746 000003
      007750 010600
      007752 104414
      007754 062706 000010
1732 007760 000207
1733
1734 007762 045 116 045 PRIA0: .ASCIZ 'NMA MEMORY ERROR ADDRESS = #01#05'
1735 .EVEN
1736
1737
1738 .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750 010026
1751 010026
1752 010032 013700 002202
1753 010036 013701 002204
1754 010042 010102
1755 010044 006101
1756 010046 006100
1757 010050
      010050 010246
      010052 010046
      010054 012746 010076
      010060 012746 000003
      010064 010600
      010066 104414
      010070 062706 000010
1758 010074 000207
1759
1760 010076 045 116 045 PRITO: .ASCIZ 'NMA MEMORY TEST ADDRESS = #01#05'
1761 .EVEN
1762
1763
1764

```

```

;
;-
PRIADD:
  SAVREG
  MOV ERRHI,R0 ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV ERRLO,R1 ;GET HIGH ADDRESS
  MOV R1,R2 ;GET LOW ADDRESS
  ROL R1 ;COPY LOW ADDRESS
  ROL R0 ;SHIFT BIT 15 TO C BIT
  PRINTB @PRIA0,R0,R2 ;SHIFT INTO HIGH ORDER
  ;PRINT MEMORY ADDRESS IN ERROR
  MOV R2,-(SP)
  MOV R0,-(SP)
  MOV @PRIA0,-(SP)
  MOV @3,-(SP)
  MOV SP,R0
  TRAP C:PNTB
  ADD @10,SP
  RTS PC ;RETURN

;
;-
PRIA0: .ASCIZ 'NMA MEMORY ERROR ADDRESS = #01#05'
.EVEN

.SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
;
;-
PRITADD:
  SAVREG
  MOV ERRHI,R0 ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV ERRLO,R1 ;GET HIGH ADDRESS
  MOV R1,R2 ;GET LOW ADDRESS
  ROL R1 ;COPY LOW ADDRESS
  ROL R0 ;SHIFT BIT 15 TO C BIT
  PRINTB @PRITO,R0,R2 ;SHIFT INTO HIGH ORDER
  ;PRINT MEMORY ADDRESS IN ERROR
  MOV R2,-(SP)
  MOV R0,-(SP)
  MOV @PRITO,-(SP)
  MOV @3,-(SP)
  MOV SP,R0
  TRAP C:PNTB
  ADD @10,SP
  RTS PC ;RETURN

;
;-
PRITO: .ASCIZ 'NMA MEMORY TEST ADDRESS = #01#05'
.EVEN

```

B5

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 40  
 SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 53

```

1766          .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
1767
1768          ;
1769          ;ROUTINE TO ISSUE A SPACE RECORDS
1770          ;COMMAND (FORWARD OR REVERSE)
1771          ;
1772          ;INPUT:
1773          ;
1774          ;
1775          ;R3      NUMBER OF RECORDS TO BE SPACED OVER
1776          ;          BIT15 CONTROLS DIRECTION
1777          ;          BIT15 = 0 IS FORWARD
1778          ;          BIT15 = 1 IS REVERSE
1779          ;R5      FIRST DEVICE UNIBUS ADDRESS
1780          ;
1781          ;REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
1782          ;
1783          ;OUTPUT:
1784          ;
1785          ;CARRY    SET - SPACE RECORDS COMMAND OK
1786          ;          CLR - SPACE RECORDS FAILED
1787          ;
1788          ;
1789          ;R0      THE CONTENTS OF R4 IS MOVED TO R0
1790          ;
1791          ;
1792          ;IMPLICIT OUTPUT:
1793          ;
1794          ;TAPE HAS BEEN MOVED
1795          ;
1796          ;SIDE EFFECTS:
1797          ;
1798          ;
1799          ;
1800
1801          SPACE::
1802          SAVREG
1803          MOV     #500.,SDELAY
1804          MOV     #140010,801
1805          TST     R3
1806          BMI     51
1807          MOV     R3,901
1808          BR      101
1809          BIC     #BIT15,R3
1810          MOV     R3,901
1811          BIS     #BIT15,801
1812          MOV     #801,R4
1813          MOV     R4,TSDB(R5)
1814          JSR     PC,WAITF
1815          BCS     201
1816          DELAY   250
1817          MOV     #250,(PC),
1818          .WORD   0
1819          MOV     L#DLY,(PC),
1820          .WORD   0
1821          DEC     -6(PC)
1822          BNE     ,-4
1823
1824          ;SAVE THE GENERAL REGISTERS
1825          ;SET UP DELAY
1826          ;SET UP COMMAND, SPACE FORWARD
1827          ;CHECK FOR DIRECTION
1828          ;BR, IF REVERSE INDICATED
1829          ;LOAD UP NUMBER OF RECORDS TO SPACE
1830          ;GO DO COMMAND
1831          ;CLEAR DIRECTION BIT
1832          ;LOAD UP NUMBER OF RECORDS TO SPACE
1833          ;SET REVERSE BIT IN COMMAND PACKET
1834          ;SET UP R4 WITH PACKET ADDRESS
1835          ;SEND OUT COMMAND
1836          ;WAIT FOR SSR
1837          ;BR, IF SSR IS SET AND OK
1838          ;DELAY ABOUT .25 SECONDS

```

C5

CZIKFA TK-25 FBT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 40-1  
SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 54

|      |        |        |        |     |     |             |                              |
|------|--------|--------|--------|-----|-----|-------------|------------------------------|
|      | 010250 | 005367 | 177756 |     | DEC | -22(PC)     |                              |
|      | 010254 | 001367 |        |     | BNE | .-20        |                              |
| 1817 | 010256 | 005337 | 010330 |     | DEC | SDELAY      | ;BUMP DELAY COUNTER DOWN     |
| 1818 | 010262 | 001356 |        |     | BNE | 15;         | ;BR, IF MORE DELAY           |
| 1819 | 010264 | 000411 |        |     | BR  | 60;         | ;BR IF TROUBLE CARRY = CLEAR |
| 1820 | 010266 | 016501 | 000000 | 20; | MOV | TSSR(R5),R1 | ;READ TSSR                   |
| 1821 | 010272 | 012702 | 000200 |     | MOV | *SSR,R2     | ;SET UP EXPECTED             |
| 1822 | 010276 | 020201 |        | 25; | CMF | R2,R1       | ;ARE THEY OK                 |
| 1823 | 010300 | 001401 |        |     | BEQ | 40;         | ;BR, IF EQUAL = OK           |
| 1824 | 010302 | 000402 |        |     | BR  | 60;         | ;TROUBLE EXIT                |
| 1825 | 010304 | 000261 |        | 40; | SEC |             | ;SET CARRY NO TROUBLE        |
| 1826 | 010306 | 000401 |        |     | BR  | 70;         | ;EXIT                        |
| 1827 | 010310 | 000241 |        | 60; | CLC |             | ;CARRY CLEAR = ERROR         |
| 1828 | 010312 |        |        | 70; |     |             |                              |
| 1829 | 010312 | 010400 |        |     | MOV | R4,R0       | ;PASS PACKET ADDRESS         |
| 1830 | 010314 | 000207 |        |     | RTS | PC          | ;RETURN                      |

D5

CZIKLA TK-25 EBT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 41  
SPACE SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 55

```

1832
1833
1834
1835
1836
1838 010316
1840
1841
1842 010320 000000
1843
1844 010322 000000
1845 010324 000000
1846 010326 000000
1847 010330 000000
1848

```

E5

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 42  
WRCHNR - WRITE CHARACTERISTICS COMMAND

SEQ 56

```

1850          .SBTTL WRCHNR - WRITE CHARACTERISTICS COMMAND
1851
1852          ;*
1853          ;
1854          ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1855          ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1856          ;
1857          ;INPUT:
1858          ;
1859          ;      R4      ADDRESS OF PACKET FROM TEST
1860          ;      R5      FIRST DEVICE UNIBUS ADDRESS
1861          ;      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1862          ;
1863          ;OUTPUT:
1864          ;
1865          ;      R0      TSSR CONTENTS
1866          ;      CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
1867          ;              CLR - WRITE CHARACTERISTICS FAILED
1868          ;
1869          ;IMPLICIT OUTPUT:
1870          ;
1871          ;      MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1872          ;      SOFTWARE SWITCHES SET AS FOLLOWS:
1873          ;              BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1874          ;
1875          ;
1876          ;SIDE EFFECTS:
1877          ;
1878          ;
1879          ;-
1880
1881 010332 WRCHNR::
1882 010332          SAVREG          ;SAVE THE GENERAL REGISTERS
1883 010336 005037 002174          CLR      BENBSW          ;CLEAR BUFFER ENABLE SWITCH
1884 010342 010465 177776 10$:   MOV      R4,TSD8(R5)      ;SEND OUT COMMAND
1885 010346 004737 017236          JSR      PC,CHKTSSR      ;WAIT FOR SSR
1886 010352 103401          BCS      20$          ;BR, IF SSR IS SET AND OK
1887 010354 000423          BR       60$          ;BR IF TROUBLE CARRY = CLEAR
1888 010356 016501 000000 20$:   MOV      TSSR(R5),R1      ;READ TSSR
1889 010362 012702 000200          MOV      @SSR,R2        ;SET UP EXPECTED
1890 010366 032701 000100          BIT      @OFL,R1        ;WAS OFF LINE SET IN TSSR
1891 010372 001402          BEQ      25$          ;BR, IF NO OFL SET
1892 010374 052702 000100          BIS      @OFL,R2        ;MAKE THEM LOOK ALIKE
1893 010400 020201 25$:   CMP      R2,R1          ;ARE THEY OK
1894 010402 001401          BEQ      40$          ;BR, IF EQUAL = OK
1895 010404 000407          BR       60$          ;TROUBLE EXIT
1896 010406 062704 000010 40$:   ADD      @8.,R4          ;POINT TO WRT CHARA DATA PACKET
1897 010412 011403          MOV      (R4),R3          ;GET ADDRESS OF MESSAGE BUFFER
1898 010414 010337 002716          MOV      R3,MESBFA      ;STORE FOR PRINT ROUTINES
1899 010420 000261          SEC          ;SET CARRY NO TROUBLE
1900 010422 000401          BR       70$          ;EXIT
1901 010424 000241 60$:   CLC          ;CARRY CLEAR = ERROR
1902 010426 016500 000000 70$:   MOV      TSSR(R5),R0      ;RETURN TSSR CONTENTS
1903 010432 000207          RTS      PC              ;RETURN
1904
1905

```

F5

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 43  
 REWIND - POSITION TAPE (REWIND) COMMAND

SEQ 57

```

1907 .SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
1908
1909
1910
1911 ; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
1912 ;
1913 ; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
1914 ; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
1915 ; SSR TO SET IN THE TSSR
1916 ;
1917 ; CALLING SEQUENCE:
1918 ;
1919 ; DO A SOFT INIT
1920 ; DO A WRITE CHARACTERISTICS
1921 ; JSR PC,REWIND
1922
1923 ; INPUT:
1924 ;
1925 ; R5 FIRST DEVICE UNIBUS ADDRESS
1926 ;
1927 ;
1928 ; OUTPUT
1929 ;
1930 ; R0 THE CONTENTS OF R4 IS PASSED TO R0
1931 ;
1932 ;
1933 ;
1934 ; -
1935 REWIND::
1936 SAVREG
1937 MOV #RWPACK,R4 ;SAVE R1-R5 UNTIL NEXT RETURN
1938 MOV R4,TSD8(R5) ;GET PACKET ADDRESS
1939 MOV #360,R3 ;SEND PACKET ADDRESS TO EXECUTE
1940 JSR PC,WAITF ;ENOUGH TIME FOR 2400' REEL TO REWIND
1941 BCS 20$ ;WAIT FOR SSR TO SET
1942 DELAY 250. ;LEAVE WHEN SSR IS SET
1943 MOV #250,.(PC)+ ;WAIT FOR .25 SECONDS
1944 .WORD 0
1945 MOV L#DLY,(PC)+
1946 .WORD 0
1947 DEC -6(PC)
1948 BNE .-4
1949 DEC -22(PC)
1950 BNE .-20
1951 DEC R3 ;BUMP COUNTER DOWN
1952 BNE 10$ ;KEEP GOING
1953 CLC ;CLEAR CARRY TO SET ERROR
1954 MOV R4,R0 ;PASS THE PACKET ADDRESS
1955 RTS PC ;RETURN
1956 .BLKB 10-<.-TUV2A&7>
1957 RWPACK:
1958 .WORD 102010 ;POSITION COMMAND (REWIND)
1959 .WORD 0 ;NOT USED

```

```

1955          .SBTTL  CKRAM  - COMPARE RAM TO I/O PACKET
1956
1957          ;+
1958          ;
1959          ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1960          ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1961          ;
1962          ;INPUT:
1963          ;
1964          ;      R4      ADDRESS OF THE COMMAND PACKET
1965          ;      R5      FIRST DEVICE UNIBUS ADDRESS
1966          ;
1967          ;OUTPUT:
1968          ;
1969          ;      CARRY    SET - RAM MATCHES PACKET
1970          ;              CLR - RAM DOES NOT MATCH PACKET
1971          ;
1972          ;IMPLICIT OUTPUT:
1973          ;
1974          ;      THE TABLE RAMDATA IS FILLED WITH THE
1975          ;      DATA HELD IN RAM.
1976          ;      RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1977          ;
1978          ;SIDE EFFECTS:
1979          ;
1980          ;
1981          ;-
1982
1983 010534      CKRAM::
1984 010534          SAVREG          ;SAVE THE GENERAL REGISTERS
1985 010540      MOV      #RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
1986 010544      MOV      #RMPKTBEGR2 ;BYTE ADDRESS OF FIRST RAM DATA
1987 010550      CLR      R3          ;CLEAR THE ERROR FLAG
1988 010552      JSR      PC,CHKTSSR  ;WAIT FOR SSR
1989 010556      JSR      PC,CHKTSSR  ;WAIT FOR SSR TO SET
1990 010562      MOV      R2,TSDBH(R5) ;SELECT NEXT RAM ADDRESS
1991 010566      JSR      PC,CHKTSSR  ;WAIT FOR SSR TO SET
1992 010572      MOV      R1,TSBAL(R5),(R1) ;READ THE RAM DATA
1993 010576      CMP      (R1)+,(R4)+ ;COMPARE TO EXPECTED
1994 010600      BEQ      20$         ;BRANCH IF OK
1995 010602      INC      R3          ;SET ERROR FLAG
1996 010604      INC      R2          ;ADDRESS OF NEXT RAM LOCATION
1997 010606      CMP      R2,#RMPKTEND ;REACHED END YET ?
1998 010612      BLE      10$         ;BRANCH TILL ALL READ
1999 010614      TST      R3          ;WAS AN ERROR FOUND ?
2000 010616      BEQ      30$         ;BRANCH IF NOT
2001 010620      CLC              ;CLEAR CARRY TO SHOW ERROR
2002 010622      BR       50$         ;AND EXIT
2003 010624      SEC              ;SHOW GOOD COMPARE
2004 010626      MOV      #8,RAMSIZ  ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
2005 010634      RTS      PC          ;RETURN
2006

```

H5

CZTKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 45  
RAMER - READ AND DISPLAY SELECTED RAM

SEQ 59

```

2008 .SBTTL RAMER - READ AND DISPLAY SELECTED RAM
2009 ;*
2010 ;
2011 ;ROUTINE TO READ THE SELECTED RAM LOCATIONS
2012 ;
2013 ;INPUT:
2014 ;
2015 ;       R5       FIRST DEVICE UNIBUS ADDRESS
2016 ;       CONSOLE WILL ALSO BE PRINTED TO
2017 ;
2018 ;IMPLICIT OUTPUT:
2019 ;
2020 ;       THE TABLE RAMDATA IS FILLED WITH THE
2021 ;       DATA HELD IN RAM.
2022 ;
2023 ;SIDE EFFECTS:
2024 ;
2025 ;
2026 ;-
2027
2028 RAMER::
2029     SAVREG          ;SAVE THE GENERAL REGISTERS
2030     MOV             RAMR5H,R5      ;RESET R5 TO FIRST DEVICE REGISTER
2031     MOV             #RAMDATA,R1    ;ADDRESS TO SAVE THE RAM DATA
2032     MOV             RAMHLD,R2      ;BYTE ADDRESS OF THE FIRST RAM DATA
2033     MOV             RAMSIZ,R3      ;SET THE SIZE OF THE READ UP
2034     JSR             PC,CHKTSSR     ;WAIT FOR THE SSR TO SET
2035     MOV             R2,TSDBH(R5)    ;SELECT NEXT RAM ADDRESS
2036     JSR             PC,CHKTSSR     ;WAIT FOR SSR TO SET
2037     MOV             TSDBH(R5),(R1) ;READ THE RAM DATA
2038     ADD             #1,R2          ;ADDRESS OF THE NEXT RAM LOCATION
2039     SOB             R3,10$         ;NUMBER OF LOCATIONS COUNTER
2040     MOV             RAMSIZ,R4      ;GET THE RAM SIZE
2041     MOV             RAMHLD,R2      ;GET THE STARTING RAM ADDRESS
2042     ADD             R2,R4          ;CALCULATE THE END ADDRESS
2043     SUB             #1,R4          ;CORRECT VALUE OF PRINTOUT
2044     PRINTX          #RAMIOP,R2,R4  ;RAM ADDRESS = 10 - 17, ETC.
2045     MOV             R4,-(SP)
2046     MOV             R2,-(SP)
2047     MOV             #RAMIOP,-(SP)
2048     MOV             #3,-(SP)
2049     MOV             SP,R0
2050     TRAP            C:PNTX
2051     ADD             #10,SP
2052     MOV             #RAMDATA,R1    ;ADDRESS OF WHERE RAM DATA IS
2053     MOV             RAMSIZ,R3      ;THE SIZE OF THE RAM FIELD READ
2054     CLR             R4             ;NO EXTRA DATA LEFT OVER
2055     MOV             (R1),R4        ;PICK UP BYTE OF RAM DATA
2056     BIC             #177400,R4     ;GET RID OF SIGN EXTEND
2057     PRINTX          #RAMPD,R4      ;"010 211 111 222 377 000 123 134 ETC."
2058     MOV             R4,-(SP)
2059     MOV             #RAMPD,-(SP)
2060     MOV             #2,-(SP)
2061     MOV             SP,R0
2062     TRAP            C:PNTX
2063     ADD             #6,SP
2064     SOB             R3,30$         ;LOOP UNTIL ALL PRINTED

```

```

2052 011016 000207          504:   RTS      PC          ;RETURN
2053
2054 011020 000000          RAMHLD: .WORD  0          ;RAM ADDR HOLDER 1ST ADDRESS
2055 011022 000000          RAMR5H: .WORD  0          ;HOLDS R5 FOR LATER
2056 011024      045      116      045  RAMIOP: .ASCIIZ 'NNA Ram Address (Octal) = %03NA - %03N'
2057 011075      045      101      040  RAMPD: .ASCIIZ 'NA %03NA '
2058
2059

```

J5

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 46  
CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

SEQ 61

```

2061          .SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
2062          ;+
2063          ;
2064          ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
2065          ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
2066          ;
2067          ;INPUT:
2068          ;
2069          ;      R4      ADDRESS OF THE CHARACTERISTICS DATA
2070          ;      R5      FIRST DEVICE UNIBUS ADDRESS
2071          ;
2072          ;OUTPUT:
2073          ;
2074          ;      CARRY   SET - RAM MATCHES PACKET
2075          ;              CLR - RAM DOES NOT MATCH PACKET
2076          ;
2077          ;IMPLICIT OUTPUT:
2078          ;
2079          ;      THE TABLE RAMDATA IS FILLED WITH THE
2080          ;      DATA HELD IN RAM.
2081          ;      RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
2082          ;
2083          ;SIDE EFFECTS:
2084          ;
2085          ;
2086          ;-
2087
2088 011110      CKRAM2:
2089 011110      SAVREG          ;SAVE THE GENERAL REGISTERS
2090 011114      MOV            #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
2091 011120      MOV            #RMCHBEG,R2      ;BYTE ADDRESS OF FIRST RAM DATA
2092 011124      CLR            R3              ;CLEAR THE ERROR FLAG
2093 011126      JSR            PC,CHKTSSR      ;WAIT FOR SSR
2094 011132      JSR            PC,CHKTSSR      ;WAIT FOR SSR TO SET
2095 011136      MOV            R2,TSDBH(R5)    ;SELECT NEXT RAM ADDRESS
2096 011142      JSR            PC,CHKTSSR      ;WAIT FOR SSR TO SET
2097 011146      MOV            R1,TSBAL(R5),(R1) ;READ THE RAM DATA
2098 011152      CMP            (R1)+,(R4)+    ;COMPARE TO EXPECTED
2099 011154      BEQ            20$            ;BRANCH IF OK
2100 011156      INC            R3              ;SET ERROR FLAG
2101 011160      INC            R2              ;ADDRESS OF NEXT RAM LOCATION
2102 011162      MOV            #8.,RAMSIZ     ;ASSUME NORMAL NOT SET
2103 011170      CMP            R2,#RMCHEND-2  ;REACHED END YET ?
2104 011174      BLE            10$            ;BRANCH TILL ALL READ
2105 011176      IST            R3              ;WAS AN ERROR FOUND ?
2106 011200      BEQ            30$            ;BRANCH IF NOT
2107 011202      CLC              ;CLEAR CARRY TO SHOW ERROR
2108 011204      BR            50$            ;AND EXIT
2109 011206      SEC              ;SHOW GOOD COMPARE
2110 011210      RTS            PC             ;RETURN
2111

```

K5

CZIKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 06:12 PAGE 47  
CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

SEQ 62

```

2113          .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
2114          ;+
2115          ;
2116          ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
2117          ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2118          ;ERROR PRINT ROUTINES.
2119          ;
2120          ;INPUT:
2121          ;
2122          ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2123          ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
2124          ;      R2      EXPD MESSAGE BUFFER ADDRESS
2125          ;OUTPUT:
2126          ;
2127          ;      CARRY   SET - MESSAGE BUFFERS MATCH
2128          ;              CLR -MESSAGE BUFFERS DON'T MATCH
2129          ;
2130          ;IMPLICIT OUTPUT:
2131          ;
2132          ;      EXPMSG      BUFFER IS SET TO EXPD DATA
2133          ;      RECVMSG     BUFFER IS SET TO RECV DATA
2134          ;      RCVHIADD    SET TO HIGH ORDER ADDRESS OF RECV
2135          ;      RCVLOADD    SET TO LOW ORDER ADDRESS OF RECV
2136          ;
2137          ;-
2138          CKMSG::
2139          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2140          MOV             R0,RCVHIADD    ;SAVE RECV HIGH ADDRESS
2141          MOV             R1,RCVLOADD    ;SAVE RECV LOW ADDRESS
2142          TST             KTENABLE       ;TESTING ABOVE 28K?
2143          BEQ             10$           ;BR IF NO
2144          JSR             PC,SETMAP      ;RETURN ADDRESS BIASED TO PAR6 IN R0
2145          MOV             R0,R1         ;GET RETURNED ADDRESS BIASED TO PAR6
2146          10$:          CLR             R4
2147          CLR             R3           ;WORD IN BUFFER
2148          MOV             R2,R5        ;CLEAR ERROR SEEN FLAG
2149          15$:          MOV             (R2),EXPMSG(R4) ;GET EXPD BUFFER ADDRESS
2150          MOV             (R1),RECVMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2151          CMP             (R2)+,(R1)+   ;SAVE RECV FOR ERROR REPORT
2152          BEQ             25$           ;EXPD EQUAL RECV?
2153          INC             R3           ;BR IF YES
2154          25$:          ADD             02,R4    ;SET ERROR SEEN FLAG
2155          CMP             R4,014        ;POINT TO NEXT WORD ADDRESS
2156          BLE             15$           ;DONE FIRST 7 WORDS?
2157          BIT             0X2,EXTF,XST2(R5) ;BR IF NO
2158          BEQ             50$           ;IS EXTENDED FEATURES SET IN EXPD?
2159          CMP             R4,016        ;BR IF NO
2160          BLE             15$           ;DONE EXTENDED FEATURES WORD?
2161          50$:          TST             R3
2162          BEQ             55$           ;BR IF NO
2163          CLC              ;ANY ERRORS SEEN?
2164          BR              60$           ;SET FAILURE
2165          55$:          SEC              ;
2166          60$:          RTS             PC    ;SET SUCCESS
2167          ;RETURN

```

L5

```

2169 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
2170 ;*
2171 ;
2172 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
2173 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2174 ;ERROR PRINT ROUTINES.
2175 ;
2176 ;INPUT:
2177 ;
2178 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2179 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
2180 ; R2 EXPD MESSAGE BUFFER ADDRESS
2181 ; R3 NUMBER OF BYTES TO COMPARE
2182 ;
2183 ;OUTPUT:
2184 ;
2185 ; CARRY SET - MESSAGE BUFFERS MATCH
2186 ; CLR - MESSAGE BUFFERS DON'T MATCH
2187 ;
2188 ;IMPLICIT OUTPUT:
2189 ;
2190 ; EXPMSG BUFFER IS SET TO EXPD DATA
2191 ; RECMSG BUFFER IS SET TO RECV DATA
2192 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2193 ; RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
2194 ;
2195 ;-
2196 CKMSG2:
2197 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2198 CMP R3, #RECVMSG-EXPMSG, #000 ;IS COUNT ABOVE MAX ALLOWED?
2199 BLE 5$ ;BR IF NO
2200 MOV #RECVMSG-EXPMSG, R3, #000
2201 PRINTF #DEBUGMSG, #000
2202 MOV #DEBUGMSG, -(SP)
2203 MOV #1, -(SP)
2204 MOV SP, R0
2205 TRAP C$PRINTF
2206 ADD #4, SP
2207 5$: MOV R0, RCVHIADD ;SAVE RECV HIGH ADDRESS
2208 MOV R1, RCVLOADD ;SAVE RECV LOW ADDRESS
2209 TST K$ENABLE ;TESTING ABOVE 28K?
2210 BEQ 10$ ;BR IF NO
2211 JSR PC, SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
2212 MOV R0, R1 ;GET RETURNED ADDRESS BIASED TO PAR6
2213 10$: CLR R4 ;WORD IN BUFFER
2214 CLR R5 ;CLEAR ERROR SEEN FLAG
2215 15$: MOVB (R2), EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2216 MOVB (R1), RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
2217 CMPB (R2)+, (R1)+ ;EXPD EQUAL RECV?
2218 BEQ 25$ ;BR IF YES
2219 INC R5 ;SET ERROR SEEN FLAG
2220 25$: ADD #1, R4 ;POINT TO NEXT BYTE
2221 CMP R4, R3 ;DONE ALL BYTES?
2222 BGE 50$ ;BR IF YES
2223 BR 15$ ;DO NEXT BYTE
2224 50$: TST R5 ;ANY ERRORS SEEN?
2225 BEQ 55$ ;BR IF NO

```

M5

CZIKFA TK-25 FRI END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 48-1  
CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

SEQ 64

```

2221 011454 000241          CLC          ;SET FAILURE
2222 011456 000401          BR          60$          ;
2223 011460 000261          55$: SEC          ;SET SUCCESS
2224 011462 000207          60$: RTS          PC          ;RETURN
2225
2226 011464 120 122 117 DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;000
2227 011554 045 116 045 FERCM: .ASCII /NMA ***/
2228 011565 040 040 124 ERCM: .ASCIZ / TSSR ERROR CODE REC'D * /
2229 011620 056 056 056 SIMSG: .ASCIZ /... AFTER DOING SOFT INIT/
2230 011653 124 105 123 TINERR: .ASCIZ /TEST: .../
2231          .EVEN

```

N5

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 49  
 CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

SEQ 65

```

2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249 011666
      011666
2250 011666 004737 005264
2251 011672 004737 020154
2252 011676
      011676
      011676 104423
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265 011700
      011700
2266 011700 004737 005264
2267 011704 012700 000004
2268 011710 004737 007072
2269 011714 013700 002716
2270 011720 005001
2271 011722 004737 014062
2272 011726
      011726
      011726 104423
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283

;+
;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
;
;INPUT:
;
;      R1      CONTENTS OF TSSR AT ERROR
;
;SIDE EFFECTS:
;
;      EXECUTES DROP UNIT TO CEASE TESTING
;-

      BGNMSG   SFMSG
SFMSG:
      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
      JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
      ENDMMSG
L10003:
      TRAP     C#MSG

;+
;PRINT ROUTINE TO PRINT THE CONTENTS OF
;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
;
;INPUTS:
;
;      R1      TSSR CONTENTS
;      R4      ADDRESS OF COMMAND PACKET
;-

      BGNMSG   PKTSSR
PKTSSR:
      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
      MOV      #4,R0           ;NO. OF WORDS IN PACKET
      JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
      MOV      MESBFA,R0      ;ADDRESS OF MESSAGE BUFFER
      CLR      R1             ;ASSUME NO HIGH MEMORY
      JSR      PC,PRMESS      ;PRINT THE MESSAGE BUFFER ALSO
      ENDMMSG
L10004:
      TRAP     C#MSG

;+
;PRINT ROUTINE TO PRINT THE CONTENTS OF
;TSSR AND A GET STATUS COMMAND PACKET.
;
;INPUTS:
;
;      R1      TSSR CONTENTS
;      R4      ADDRESS OF COMMAND PACKET
;-

```

B6

SEQ 66

```

2284
2285 011730          BGNMSG  PKTGETS
      011730          PKTGETS:
2286 011730 004737 005264      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2287 011734 012700 000002      MOV      #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
2288 011740 004737 007072      JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
2289 011744          ENDMMSG
      011744          L10005:
      011744 104423      TRAP      C#MSG

2290
2291
2292
2293          ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
2294          ;
2295          ;INPUTS:
2296          ;
2297          ;      R1      TSSR CONTENTS
2298          ;      R4      ADDRESS OF COMMAND PACKET
2299          ;
2300          ;
2301 011746          BGNMSG  SFFMSG
      011746          SFFMSG:
2302 011746 004737 005264      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2303 011752          ENDMMSG
      011752          L10006:
      011752 104423      TRAP      C#MSG

2304
2305
2306          .SBTTL  PKTMES - PRINT TSSR AND MESSAGE BUFFER
2307          ;
2308          ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
2309          ;BUFFER FOR ERROR REPORTS
2310          ;
2311          ;INPUTS:
2312          ;
2313          ;      R1      CONTENTS OF TSSR
2314          ;      R2      LOW ORDER MESSAGE BUFFER
2315          ;      R3      HIGH ORDER MESSAGE BUFFER ADDRESS
2316          ;      NOTE: R3 IS IGNORED IF KENABLE FLAG IS CLEAR
2317          ;
2318          ;
2319 011754          BGNMSG  PKTMES
      011754          PKTMES:
2320 011754 004737 005264      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR
2321 011760 010200          MOV      R2,R0          ;LOW ORDER ADDRESS
2322 011762 010301          MOV      R3,R1          ;HIGH ORDER ADDRESS
2323 011764 004737 014062      JSR      PC,PRMESS      ;PRINT THE MESSAGE BUFFER
2324 011770          ENDMMSG
      011770          L10007:
      011770 104423      TRAP      C#MSG

2325

```

C6

CZIKEA TK-25 FBT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 50

SEQ 67

```

2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339 011772
      011772
2340 011772 004737 010026
2341 011776 016501 000000
2342 012002 004737 005264
2343 012006
      012006
      012006 104423
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358 012010
      012010
2359 012010 012700 000007
2360 012014 004737 015422
2361 012020
      012020
      012020 104423
2362
2363

```

```

      .SBTTL ADDSSR - PRINT TEST ADDRESS AND TSSR
      **
      ;PRINT ROUTINE TO PRINT THE CONTENTS OF
      ;TSSR AND A MEMORY TEST ADDRESS
      ;
      ;INPUTS:
      ;
      ;      R5      FIRST DEVICE UNIBUS ADDRESS
      ;      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
      ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
      ;
      ;
      BGNMSG ADDSSR
ADDSSR:
      JSR      PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
      MOV      TSSR(R5),R1     ;GET CURRENT TSSR
      JSR      PC,PRITSSR     ;PRINT THE CONTENTS OF TSSR REGISTER
      ENDMSG
L10010:
      TRAP     C#MSG

      .SBTTL MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
      **
      ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
      ;
      ;IMPLICIT INPUTS:
      ;
      ;      EXPMSG - EXPECTED MESSAGE BUFFER
      ;      RECMG - RECEIVED MESSAGE BUFFER
      ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
      ;      RCVLOAD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
      ;
      ;
      BGNMSG MSGEXP
MSGEXP:
      MOV      #7,R0           ;ASSUME NO EXT FEATURES
      JSR      PC,PRMSGEXP     ;PRINT EXPD/RCV MESSAGE BUFFERS
      ENDMSG
L10011:
      TRAP     C#MSG

```



E6

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 52  
MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

SEQ 69

```

2388          .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
2389          ;*
2390          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2391          ;
2392          ;
2393          ;IMPLICIT INPUTS:
2394          ;
2395          ;
2396          ;   EXPMSG - EXPECTED MESSAGE BUFFER
2397          ;   RECMSG - RECEIVED MESSAGE BUFFER
2398          ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2399          ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2400          ;
2401          BGNMSG MSGSTAT
2402          MSGSTAT:
2403          10$: MOV     #STATCOD,R1      ;ASCII ADDRESS TABLE
2404          BEQ     20$                ;DONE ALL MSG LINES?
2405          PRINTX  R0                  ;BR IF YES
2406          MOV     R0,-(SP)            ;PRINT STATUS BIT NAMES
2407          MOV     #1,-(SP)
2408          MOV     SP,R0
2409          TRAP    C#PNTX
2410          ADD     #4,SP
2411          BR      10$                ;DO ANOTHER MSG LINE
2412          20$: MOV     #10.,R0        ;NUMBER OF WORDS IN A READ STATUS BUFFER
2413          JSR     PC,PRMSGEXP        ;PRINT EXPD/RCV MESSAGE BUFFERS
2414          ENOMSG
2415          L10013: TRAP    C#MSG
2416          STATCOD: .WORD 1$,2$,3$,4$,5$,6$,0
2417          1$: .ASCIZ 'ANNA Tape Bus Signals in Word #8:'
2418          2$: .ASCIZ 'ANNA PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
2419          3$: .ASCIZ 'ANNA IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
2420          4$: .ASCIZ 'ANNA IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
2421          5$: .ASCIZ 'ANNA Tape Bus Signals in Word #9:'
2422          6$: .ASCIZ 'ANNA DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
2423          .EVEN
2424          .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
2425          ;*
2426          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2427          ;
2428          ;
2429          ;IMPLICIT INPUTS:
2430          ;
2431          ;
2432          ;   EXPMSG - EXPECTED MESSAGE BUFFER
2433          ;   RECMSG - RECEIVED MESSAGE BUFFER
2434          ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2435          ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2436          ;
2437          BGNMSG MSGLOOP
2438          MSGLOOP:
2439          MOV     #LOOPCOD,R1      ;ASCII ADDRESS TABLE

```

F6

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 52-1  
MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

SEQ 70

```

2436 012722 012100          10$: MOV      (R1)+,R0          ;DONE ALL MSG LINES?
2437 012724 001410          BEQ      20$                  ;BR IF YES
2438 012726          PRINTX  R0                          ;PRINT STATUS BIT NAMES
      012726 010046          MOV      R0,-(SP)
      012730 012746 000001    MOV      #1,-(SP)
      012734 010600          MOV      SP,R0
      012736 104415          TRAP     C$PNTX
      012740 062706 000004    ADD      #4,SP
2439 012744 000766          BR       10$                  ;DO ANOTHER MSG LINE
2440 012746 012700 000012    20$: MOV      #10,R0          ;NUMBER OF WORDS IN A READ STATUS BUFFER
2441 012752 004737 015422    JSR      PC,PRMSGEXP        ;PRINT EXPD/RECV MESSAGE BUFFERS
2442 012756          ENDMSG
      012756          L10014:
      012756 104423          TRAP     C$MSG
2443
2444 012760 013000 013053 013152 LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
2445 013000          045      116      045 1$:ASCIZ 'NNA Tape Bus Loopback Signals in Word #8:'
2446 013053          045      116      045 2$:ASCIZ 'NNA PARERR<15> IRESV2<14> IRESV1<13>'
2447 013152          045      116      045 3$:ASCIZ 'NNA IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
2448 013251          045      116      045 4$:ASCIZ 'NNA IWM =>IFM<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
2449 013350          045      116      045 5$:ASCIZ 'NNA ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDOP <04>'
2450 013447          045      116      045 6$:ASCIZ 'NNA IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
2451 013546          045      116      045 7$:ASCIZ 'NNA IGO =>IFPT<00>'
2452          .EVEN
2453

```

G6

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 53  
MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

SEQ 71

```

2455          .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
2456          ;+
2457          ;
2458          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
2459          ;
2460          ;
2461          ;IMPLICIT INPUTS:
2462          ;
2463          ;      EXPMSG - EXPECTED MESSAGE BUFFER
2464          ;      RECMMSG - RECEIVED MESSAGE BUFFER
2465          ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2466          ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2467          ;
2468          BGNMSG MSGSUB
2469          MSGSUB:
2470          MOV     #10, R0          ;SIZE OF WRITE SUBSYSTEM BUFFER
2471          JSR     PC, PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
2472          ENDMMSG
2473          L10015:
2474          TRAP     C#MSG
2475
2476          .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
2477          ;+
2478          ;
2479          ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
2480          ;
2481          ;IMPLICIT INPUTS:
2482          ;
2483          ;      ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
2484          ;      ERRLO - MEMORY ERROR LOW ORDER ADDRESS
2485          ;      EXP - EXPECTED DATA
2486          ;      RECV - RECEIVED DATA
2487          ;
2488          ;
2489          BGNMSG MEMADD
2490          MEMADD:
2491          JSR     PC, PRIADD        ;PRINT MEMORY ADDRESS IN ERROR
2492          MOV     EXPD, R1          ;GET EXPD DATA
2493          MOV     RECV, R2         ;GET RECEIVED DATA
2494          JSR     PC, PRI XOR      ;PRINT EXPD/RECV
2495          ENDMMSG
2496          L10016:
2497          TRAP     C#MSG

```

H6

CZTKEA TK-25 FBT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 54

SEQ 72

```

2497          .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
2498          ;*
2499          ;
2500          ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2501          ;WHEN THE RAM DATA DOES NOT MATCH.
2502          ;
2503          ;INPUTS:
2504          ;
2505          ;      R4      POINTER TO COMMAND PACKET
2506          ;
2507          ;IMPLICIT INPUTS:
2508          ;
2509          ;      RAMDATA  DATA AS READ FROM THE RAM
2510          ;      RAMSIZ   NUMBER OF BYTES IN PACKET
2511          ;                  IF RAMSIZ=0 THEN DEFAULT TO 8.
2512          ;
2513          ;IMPLICIT OUTPUTS:
2514          ;
2515          ;      RAMSIZ   SET TO 0
2516          ;
2517          ;
2518          PRAMPKT:
2519          ;      SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2520          ;      MOV      #RAMDATA,R1      ;DATA FROM THE RAM
2521          ;      CLR      R2              ;INIT BYTE NUMBER
2522          5$:  ;      CMPB     (R1)+,(R4)+    ;COMPARE EXPECTED, RECEIVED
2523          ;      BNE      7$              ;BR IF NO MATCH
2524          7$:  ;      MOVB     -1(R1),R5      ;GET RECV RAM DATA
2525          ;      MOVB     -1(R4),R3      ;GET EXPD PACKET DATA
2526          ;      XOR      R5,R3          ;XOR EXPD/RECV
2527          ;      BIC      #177400,R3      ;LOW BYTE ONLY
2528          ;      MOVB     -1(R1),RECV      ;GET RECEIVED RAM DATA
2529          ;      MOVB     -1(R4),EXPD      ;GET EXPECTED RAM DATA
2530          ;      PRINTB    #RAMASC,R2,RECV,EXPD,R3
2531          ;      MOV      R3,-(SP)
2532          ;      MOV      EXPD,-(SP)
2533          ;      MOV      RECV,-(SP)
2534          ;      MOV      R2,-(SP)
2535          ;      MOV      #RAMASC,-(SP)
2536          ;      MOV      #5,-(SP)
2537          ;      MOV      SP,R0
2538          ;      TRAP     C#PNTB
2539          ;      ADD      #14,SP
2540          10$: ;      INC      R2              ;UPDATE BYTE COUNT
2541          ;      TST      RAMSIZ          ;DEFAULT TO 8.?
2542          ;      BEQ      15$              ;BR IF YES
2543          ;      CMP      R2,RAMSIZ        ;DONE ALL BYTES?
2544          ;      BLE      5$              ;BR IF NO
2545          ;      BR       25$              ;
2546          15$: ;      CMP      R2,#8.        ;DONE DEFAULT NUMBER OF BYTES?
2547          20$: ;      BLT      5$              ;BR IF NO
2548          25$: ;      CLR      RAMSIZ        ;SET DEFAULT RAMSIZ
2549          ;      RTS      PC              ;RETURN
2550          ;
2551          ;      RAMASC: .ASCIZ '##A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
2552          ;      .EVEN

```

```

2544 .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
2545 ;+
2546 ;
2547 ;THIS ROUTINE PRINTS THE CONTENTS OF
2548 ;THE 7 WORD MESSAGE BUFFER RETURNED BY THE
2549 ;TK-25.
2550 ;
2551 ;INPUT:
2552 ;
2553 ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
2554 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
2555 ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
2556 ;
2557 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
2558 ;
2559 ;-
2560
2561 014062 PRMESS: SAVREG ;SAVE THE REGISTERS
2562 014062 MOV R5,RAMR5H ;SAVE DEVICE REGISTER POINTER
2563 014066 010537 011022 MOV R0,R5 ;SAVE LOW ORDER ADDRESS
2564 014072 010005 TST KTENABLE ;ADDRESS ABOVE 28K?
2565 014074 005737 003102 BNE 10$ ;BR IF YES
2566 014100 001001 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
2567 014102 005001 10$: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
2568 014104 010103 ROL R0 ;SHIFT BIT15 TO C BIT
2569 014106 006100 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2570 014110 006101 PRINTX #PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
2571 014112 010546 MOV R5,-(SP)
2572 014114 010146 MOV R1,-(SP)
2573 014116 012746 014714 MOV #PROASC,-(SP)
2574 014122 012746 000003 MOV #3,-(SP)
2575 014126 010600 MOV SP,R0
2576 014130 104415 TRAP C#PNTX
2577 014132 062706 000010 ADD #10,SP
2578 014136 022715 177777 CMP #177777,(R5) ;MESSAGE BUFFER FULL OF ONES
2579 014142 001010 BNE 15$ ;BR IF BUFFER IS PROBABLY OKAY
2580 014144 PRINTX #MESBFN ;"MESSAGE BUFFER PROBABLY NOT VALID"
2581 014144 012746 014634 MOV #MESBFN,-(SP)
2582 014150 012746 000001 MOV #1,-(SP)
2583 014154 010600 MOV SP,R0
2584 014156 104415 TRAP C#PNTX
2585 014160 062706 000004 ADD #4,SP
2586 014164 012746 014764 15$: PRINTX #PRIASC ;PRINT HEADER FOR CONTENTS
2587 014164 012746 000001 MOV #PRIASC,-(SP)
2588 014170 012746 000001 MOV #1,-(SP)
2589 014174 010600 MOV SP,R0
2590 014176 104415 TRAP C#PNTX
2591 014200 062706 000004 ADD #4,SP
2592 014204 005004 CLR R4 ;NUMBER OF THE NEXT WORD
2593 014206 010501 MOV R5,R1 ;COPY LOW ORDER ADDRESS
2594 014210 010300 MOV R3,R0 ;COPY HIGH ORDER ADDRESS
2595 014212 001403 BEQ 20$ ;BR IF NOT ABOVE 28K
2596 014214 004737 020270 JSR PC,SETHAP ;SETUP PAR ADDRESS IN R0
2597 014220 010005 MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
2598 014222 20$: PRINTX #MESHEA,(R5) ;PRINT "MESSAGE BUFFER HEADER ="
2599 014222

```

J6

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 55-1  
 PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

SEQ 74

|      |        |        |               |        |                |                                |
|------|--------|--------|---------------|--------|----------------|--------------------------------|
|      | 014222 | 012546 |               | MOV    | (R5)+, -(SP)   |                                |
|      | 014224 | 012746 | 015017        | MOV    | #MESHEA, -(SP) |                                |
|      | 014230 | 012746 | 000002        | MOV    | #2, -(SP)      |                                |
|      | 014234 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014236 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014240 | 062706 | 000006        | ADD    | #6, SP         |                                |
| 2584 | 014244 |        |               | PRINTX | #DATAFL, (R5)+ | ;PRINT "DATA FIELD LENGTH"     |
|      | 014244 | 012546 |               | MOV    | (R5)+, -(SP)   |                                |
|      | 014246 | 012746 | 015064        | MOV    | #DATAFL, -(SP) |                                |
|      | 014252 | 012746 | 000002        | MOV    | #2, -(SP)      |                                |
|      | 014256 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014260 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014262 | 062706 | 000006        | ADD    | #6, SP         |                                |
| 2585 | 014266 |        |               | PRINTX | #RBPORA, (R5)+ | ;PRINT "RESIDUAL BYTE COUNTER" |
|      | 014266 | 012546 |               | MOV    | (R5)+, -(SP)   |                                |
|      | 014270 | 012746 | 015131        | MOV    | #RBPORA, -(SP) |                                |
|      | 014274 | 012746 | 000002        | MOV    | #2, -(SP)      |                                |
|      | 014300 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014302 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014304 | 062706 | 000006        | ADD    | #6, SP         |                                |
| 2586 | 014310 |        |               | PRINTX | #XSOCN, (R5)+  | ;PRINT "XSTAT0 CONTENTS"       |
|      | 014310 | 012546 |               | MOV    | (R5)+, -(SP)   |                                |
|      | 014312 | 012746 | 015176        | MOV    | #XSOCN, -(SP)  |                                |
|      | 014316 | 012746 | 000002        | MOV    | #2, -(SP)      |                                |
|      | 014322 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014324 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014326 | 062706 | 000006        | ADD    | #6, SP         |                                |
| 2587 | 014332 |        |               | PRINTX | #XS1CON, (R5)+ | ;PRINT "XSTAT1 CONTENTS"       |
|      | 014332 | 012546 |               | MOV    | (R5)+, -(SP)   |                                |
|      | 014334 | 012746 | 015243        | MOV    | #XS1CON, -(SP) |                                |
|      | 014340 | 012746 | 000002        | MOV    | #2, -(SP)      |                                |
|      | 014344 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014346 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014350 | 062706 | 000006        | ADD    | #6, SP         |                                |
| 2588 | 014354 |        |               | PRINTX | #XS2CON, (R5)+ | ;PRINT "XSTAT2 CONTENTS"       |
|      | 014354 | 012546 |               | MOV    | (R5)+, -(SP)   |                                |
|      | 014356 | 012746 | 015310        | MOV    | #XS2CON, -(SP) |                                |
|      | 014362 | 012746 | 000002        | MOV    | #2, -(SP)      |                                |
|      | 014366 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014370 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014372 | 062706 | 000006        | ADD    | #6, SP         |                                |
| 2589 | 014376 |        |               | PRINTX | #XS3CON, (R5)+ | ;PRINT "XSTAT3 CONTENTS"       |
|      | 014376 | 012546 |               | MOV    | (R5)+, -(SP)   |                                |
|      | 014400 | 012746 | 015355        | MOV    | #XS3CON, -(SP) |                                |
|      | 014404 | 012746 | 000002        | MOV    | #2, -(SP)      |                                |
|      | 014410 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014412 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014414 | 062706 | 000006        | ADD    | #6, SP         |                                |
| 2590 | 014420 | 022737 | 000001 002134 | CMP    | #1, TRANSTST   | ;CHECK FOR DUMP                |
| 2591 | 014426 | 001042 |               | BNE    | 50\$           | ;BR, IF NO DUMP REQUIRED       |
| 2592 | 014430 |        |               | PRINTX | #RAMFHR        |                                |
|      | 014430 | 012746 | 014536        | MOV    | #RAMFHR, -(SP) |                                |
|      | 014434 | 012746 | 000001        | MOV    | #1, -(SP)      |                                |
|      | 014440 | 010600 |               | MOV    | SP, R0         |                                |
|      | 014442 | 104415 |               | TRAP   | C#PNTX         |                                |
|      | 014444 | 062706 | 000004        | ADD    | #4, SP         |                                |
| 2593 | 014450 | 012737 | 000010 002246 | MOV    | #8, RAMSIZ     | ;RAM FIELD IS 8 BYTES LONG     |

K6

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 55-2  
 PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

SEQ 75

```

2594 014456 012737 000020 011020      MOV      #20,RAMHLD      ;FIELD STARTS AT 20 OCTAL (10 HEX)
2595 014464 004737 010636              JSR      PC,RAMER      ;READ AND PRINT THEM
2596 014470 012737 000040 011020      MOV      #40,RAMHLD      ;FIELD STARTS AT 40 OCTAL (20 HEX)
2597 014476 004737 010636              JSR      PC,RAMER      ;READ AND PRINT THEM
2598 014502 012737 000060 011020      MOV      #60,RAMHLD      ;FIELD STARTS AT 60 OCTAL (30 HEX)
2599 014510 004737 010636              JSR      PC,RAMER      ;READ AND PRINT THEM
2600 014514 012737 000020 002246      MOV      #16.,RAMSIZ     ;RAM FIELD IS SIXTEEN BYTES LONG
2601 014522 012737 000100 011020      MOV      #100,RAMHLD    ;FIELD STARTS AT 100 OCTAL (40 HEX)
2602 014530 004737 010636              JSR      PC,RAMER      ;READ AND PRINT THEM
2603 014534 000207              SO#; RTS      PC      ;RETURN
2604 014536      045      116      045 RAMFHR: .ASCIZ  '###A ***** SPECIAL CONTROLLER RAM MEMORY DUMP *****'
2605 014634      045      116      045 MESBFN: .ASCIZ  '###A MESSAGE BUFFER CONTENTS PROBABLY NOT VALID'
2606 014714      045      116      045 PROASC: .ASCIZ  '###A Message buffer Address = #01#05'
2607 014761      045      116      045 PRIASC: .ASCIZ  '###A Message Buffer Contents:'
2608
2609 015017      045      116      045 MESHEA: .ASCIZ  '###A Message Buffer Header      = #06'
2610 015064      045      116      045 DATAFL: .ASCIZ  '###A Data Field Length      = #06'
2611 015131      045      116      045 RBPCRA: .ASCIZ  '###A Residual Byte Counter    = #06'
2612 015176      045      116      045 XS0CON: .ASCIZ  '###A XSTAT0 Contents      = #06'
2613 015243      045      116      045 X1CON: .ASCIZ  '###A XSTAT1 Contents      = #06'
2614 015310      045      116      045 XS2CON: .ASCIZ  '###A XSTAT2 Contents      = #06'
2615 015355      045      116      045 XS3CON: .ASCIZ  '###A XSTAT3 Contents      = #06'
2616                      .EVEN

```

```

2618 .SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
2619 ;+[B
2620 ;
2621 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
2622 ;
2623 ; R0 - NUMBER OF WORDS IN BUFFER
2624 ;
2625 ;IMPLICIT INPUTS:
2626 ;
2627 ; EXPMSG - EXPECTED MESSAGE BUFFER
2628 ; RECMMSG - RECEIVED MESSAGE BUFFER
2629 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2630 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2631 ;-
2632 PRMSGEXP::
2633 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2634 MOV R0,R5 ;SAVE NUMBER OF WORDS
2635 MOV RCVLOADD,R0 ;GET RECV LOW ADDRESS
2636 MOV R0,R4 ;COPY LOW ADDRESS
2637 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
2638 ROL R0 ;SHIFT BIT15 TO C BIT
2639 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2640 PRINTX #PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
2641 015446 010446 MOV R4,-(SP)
2642 015450 010146 MOV R1,-(SP)
2643 015452 012746 015602 MOV #PRMSG0,-(SP)
2644 015456 012746 000003 MOV #3,-(SP)
2645 015462 010600 MOV SP,R0
2646 015464 104415 TRAP C#PNTX
2647 015466 062706 000010 ADD #10,SP
2648 015472 PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
2649 015472 012746 015647 MOV #PRMSG1,-(SP)
2650 015476 012746 000001 MOV #1,-(SP)
2651 015502 010600 MOV SP,R0
2652 015504 104415 TRAP C#PNTX
2653 015506 062706 000004 ADD #4,SP
2654 015512 005004 CLR R4 ;NUMBER OF THE CURRENT WORD
2655 015514 012701 002266 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2656 015520 012702 002432 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
2657 015524 011100 20$: MOV (R1),R0 ;GET EXPD
2658 015526 011203 MOV (R2),R3 ;GET RECV
2659 015530 XOR R0,R3 ;XOR EXPD/RECV
2660 015540 PRINTX #PRMSG2,R4,(R1)+,(R2)+,R3
2661 015540 010346 MOV R3,-(SP)
2662 015542 012246 MOV (R2)+,-(SP)
2663 015544 012146 MOV (R1)+,-(SP)
2664 015546 010446 MOV R4,-(SP)
2665 015550 012746 015705 MOV #PRMSG2,-(SP)
2666 015554 012746 000005 MOV #5,-(SP)
2667 015560 010600 MOV SP,R0
2668 015562 104415 TRAP C#PNTX
2669 015564 062706 000014 ADD #14,SP
2670 015570 005204 INC R4 ;NUMBER OF THE NEXT
2671 015572 020405 CMP R4,R5 ;DONE ALL YET?
2672 015574 002001 BGE 50$ ;BR IF YES
2673 015576 000752 BR 20$ ;DO ANOTHER
2674 015600 000207 50$: RTS PC ;RETURN

```

M6

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 56-1  
PRMSGEXP . PRINT EXPD/RECV MESSAGE BUFFERS

SEQ 77

2654  
2655 015602 045 116 045 PRMSG0: .ASCIZ 'N#A Message Buffer Address = #01#05'  
2656 015647 045 116 045 PRMSG1: .ASCIZ 'N#A Message Buffer Contents:'  
2657 015705 045 116 045 PRMSG2: .ASCIZ 'N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06'  
2658 .EVEN  
2659

N6

CZTKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 57  
PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

SEQ 78

```

2661 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
2662 ;*
2663 ;
2664 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
2665 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
2666 ;
2667 ; RO - NUMBER OF BYTES IN BUFFER
2668 ;
2669 ;IMPLICIT INPUTS:
2670 ;
2671 ; EXPMSG - EXPECTED MESSAGE BUFFER
2672 ; RECMSG - RECEIVED MESSAGE BUFFER
2673 ;
2674 015772 PRBYTEXP::
2675 015772 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2676 015776 010005 MOV R0,R5 ;SAVE NUMBER OF BYTES
2677 016000 005037 002264 CLR PRMNO ;INIT ERROR COUNT
2678 016004 005004 CLR R4 ;NUMBER OF THE CURRENT BYTE
2679 016006 012701 002266 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2680 016012 012702 002432 MOV #RECMSG,R2 ;GET RECV BUFFER ADDRESS
2681 016016 111100 20$: MOV8 (R1),R0 ;GET EXPD BYTE
2682 016020 042700 177400 BIC #C<377>,R0 ;CLEAR UPPER BYTE
2683 016024 110037 016340 MOV8 R0,PRBEXP ;SAVE FOR ERROR REPORT
2684 016030 111203 MOV8 (R2),R3 ;GET RECV BYTE
2685 016032 042703 177400 BIC #C<377>,R3 ;CLEAR UPPER BYTE
2686 016036 110337 016342 MOV8 R3,PRBREC ;FOR ERROR REPORT
2687 016042 XOR R0,R3 ;XOR EXPD/RECV
2688 016052 122122 CMP8 (R1)+,(R2)+ ;EXPD = RECV?
2689 016054 001431 BEQ 30$ ;BR IF YES
2690 016056 005237 002264 INC PRMNO ;UPDATE ERROR COUNT
2691 016062 023727 002264 000010 CMP PRMNO,#8 ;PRINTED 8?
2692 016070 101023 BHI 30$ ;BR IF YES
2693 016072 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
016072 010346 MOV R3,-(SP)
016074 013746 016342 MOV PRBREC,-(SP)
016100 013746 016340 MOV PRBEXP,-(SP)
016104 010446 MOV R4,-(SP)
016106 012746 016206 MOV #PRBMSG,-(SP)
016112 012746 000005 MOV #5,-(SP)
016116 010600 MOV SP,R0
016120 104415 TRAP C#PNTX
016122 062706 000014 ADD #14,SP
2694 016126 FORCEEXIT 50$ ;000
2695 016136 000404 BR 35$ ;000
2696 016140 30$: FOWCERROR 27$,NOTSSR ;000
2697 016140 35$: ;000
2698 016150 INC R4 ;NUMBER OF THE NEXT
2699 016150 005204 CMP R4,R5 ;DONE ALL YET?
2700 016152 020405 BGE 50$ ;BR IF YES
2701 016154 002001 BR 20$ ;DO ANOTHER
2702 016156 000717 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
2703 016160 013746 002264 MOV PRMNO,-(SP)
016164 012746 016273 MOV #PRBTOT,-(SP)
016170 012746 000002 MOV #2,-(SP)
016174 010600 MOV SP,R0
016176 104415 TRAP C#PNTX

```

B7

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 57-1  
PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

SEQ 79

|      |        |        |        |     |         |        |                                                    |
|------|--------|--------|--------|-----|---------|--------|----------------------------------------------------|
| 2704 | 016200 | 062706 | 000005 |     | ADD     | #5,SP  |                                                    |
|      | 016204 | 000207 |        |     | RTS     | PC     | ,RETURN                                            |
| 2705 |        |        |        |     |         |        |                                                    |
| 2706 | 016206 | 045    | 116    | 045 | PRBMSG: | .ASCIZ | '#N#A BYTE #D2#A EXPD: #03#A RECV: #03#A XOR: #03' |
| 2707 | 016273 | 045    | 116    | 045 | PRBTOT: | .ASCIZ | '#N#A NUMBER OF BYTES IN ERROR = #D2'              |
| 2708 |        |        |        |     |         | .EVEN  |                                                    |
| 2709 | 016340 | 000000 |        |     | PRBEXP: | .WORD  | 0                                                  |
| 2710 | 016342 | 000000 |        |     | PRBREC: | .WORD  | 0                                                  |
| 2711 |        |        |        |     |         |        |                                                    |

C7

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 58  
EXPREC - PRINT EXPD/RECV WORD DATA

SEQ 80

2713  
2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729

016344  
016344  
016344 004737 007474  
016350  
016350  
016350 104423

```
.SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
**
PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
INPUTS:
      R1      RECEIVED DATA
      R2      EXPECTED DATA
--
      BGNMSG  EXPREC
EXPREC:: JSR   PC,PRIXOR      ;PRINT THE DATA
          ENDMSG
L10017:  TRAP   C#MSG
```

D7

CZTKEA TK-25 FRT END FUNC #2 MACRO M1200 20 APR-84 08:12 PAGE 59

SEQ 81

```

2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744 016352
      016352
2745 016352 004737 007344
2746 016356
      016356
      016356 104423
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771 016360
      016360
2772 016360 004737 013630
2773 016364
      016364
      016364 104423
2774
2775
2776
2777
2778
2779
2780
2781

```

```

      .SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
      ;*
      ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
      ;
      ;INPUTS:
      ;
      ;      R1      RECEIVED DATA BYTE
      ;      R2      EXPECTED DATA BYTE
      ;
      ;-
      BGNMSG EXPBREC
EXPBREC::
      JSR      PC,PRI8XOR      ;PRINT THE DATA
      ENDMMSG
L10020:
      TRAP     C#MSG

      .SBTTL RAMERR - PRINT RAM AND PACKET DATA
      ;*
      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
      ;
      ;INPUTS:
      ;
      ;      R4      POINTER TO COMMAND PACKET
      ;
      ;IMPLICIT INPUTS:
      ;
      ;      RAMDATA      DATA AS READ FROM THE RAM
      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
      ;                  IF RAMSIZ=0 THEN DEFAULT TO 8.
      ;
      ;IMPLICIT OUTPUTS:
      ;
      ;      RAMSIZ      SET TO 0
      ;
      ;-
      BGNMSG RAMERR
RAMERR::
      JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
      ENDMMSG
L10021:
      TRAP     C#MSG

      .SBTTL RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
      ;*
      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
      ;
      ;INPUTS:

```

E7

CZTKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 59-1  
RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA

SEQ 82

```

2782
2783      R4      POINTER TO COMMAND PACKET
2784
2785      IMPLICIT INPUTS:
2786
2787      RAMDATA   DATA AS READ FROM THE RAM
2788      RAMSIZ    NUMBER OF BYTES IN PACKET
2789              IF RAMSIZ=0 THEN DEFAULT TO 8.
2790      ERRHI     HIGH ORDER TEST ADDRESS
2791      ERRLO     LOW ORDER TEST ADDRESS
2792
2793      IMPLICIT OUTPUTS:
2794
2795      RAMSIZ    SET TO 0
2796
2797
2798      BGNMSG    RAMTADD
2799      RAMTADD:  JSR      PC,PRITADD      ;PRINT TEST ADDRESS
2800              JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
2801              ENDMMSG
2802      L10022:    TRAP      C#MSG
2803
2804      .SBTTL    RAMEXP - PRINT RAM EXPD/RECV DATA
2805
2806      ;
2807      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2808      ;
2809      ;INPUTS:
2810      ;
2811      R1        RECEIVED DATA
2812      R2        EXPECTED DATA
2813      R4        CONTROLLER RAM ADDRESS
2814
2815      ;
2816      BGNMSG    RAMEXP
2817      RAMEXP:   BIC      #PC<377>,R1    ;SAVE EXPD RAM DATA BYTE
2818              BIC      #PC<377>,R2    ;SAVE EXPD RAM DATA BYTE
2819              JSR      PC,PRIRAM      ;PRINT THE RAM ADDRESS
2820              JSR      PC,PRIXOR      ;PRINT THE DATA
2821              ENDMMSG
2822      L10023:    TRAP      C#MSG
2823
2824      .SBTTL    TIMEXP - PRINT TIMER A,B AND EXP/REC
2825
2826      ;
2827      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2828      ;AND TIMER A,B HEADER MESSAGE
2829      ;
2830      ;INPUTS:
2831      ;
2832      R1        RECEIVED DATA
2833      R2        EXPECTED DATA

```

F7

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 59-2  
 TIMEXP - PRINT TIMER A,B AND EXP/REC

SEQ 83

```

2833
2834
2835 016422          BGNMSG  TIMEXP
      016422          TIMEXP:
2836 016422          PRINTX  #TIMSGO      ;PRINT HEADER
      016422 012746 016450      MOV      #TIMSGO, -(SP)
      016426 012746 000001      MOV      #1, -(SP)
      016432 010600      MOV      SP, R0
      016434 104415      TRAP      C:PNTX
      016436 062706 000004      ADD      #4, SP
2837 016442 004737 007474      JSR      PC, PRI XOR      ;PRINT THE DATA
2838 016446          ENDMMSG
      016446          L10024:
      016446 104423      TRAP      C:MSG
2839
2840
2841 016450          045      116      045  TIMSGO: .ASCIZ  'NWA TIMER A STATUS IS IN BIT 3NWA TIMER B STATUS IS IN BIT 2'
2842          .EVEN

```

G7

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 60  
 BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

SEQ 84

```

2844 .SBTTL BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
2845
2846 ;+
2847 ;
2848 ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2849 ;
2850 ;INPUTS:
2851 ;
2852 ; R1 CONTENTS OF TSSR
2853 ; R2 DATA WRITTEN (8 BITS)
2854 ;
2855 ;-
2856
2857 016550 BGNMSG BADSSR
2858 016550 BADSSR::
2859 016550 010246 MOV R2, -(SP) ;SAVE DATA TRANSFERRED
2859 016552 C42702 177400 BIC #177400, R2 ;GET JUST ONE BYTE
2860 016556 010246 PRINTB #XFERASC, R2
2860 016556 010246 MOV R2, -(SP)
2860 016560 012746 016610 MOV #XFERASC, -(SP)
2860 016564 012746 000002 MOV #2, -(SP)
2860 016570 010600 MOV SP, R0
2860 016572 104414 TRAP C#PNTB
2860 016574 062706 000006 ADD #6, SP
2861 016600 012602 MOV (SP)+, R2 ;RESTORE R2
2862 016602 004737 005264 JSR PC, PRITSSR ;DECODE TSSR CONTENTS
2863 016606 ENDMMSG
2863 016606 L10025:
2863 016606 104423 TRAP C#MSG
2864 016610 045 116 045 XFERASC: .ASCIZ '#N#A Data Transferred = #03'
2865

```

```
2867          .SBTTL  GLOBAL SUBROUTINES SECTION
2868
2869          ;++
2870          ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2871          ; THAT ARE USED IN MORE THAN ONE TEST.
2872          ;--
2873
2874          .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER
2875
2876          ;+
2877          ;
2878          ; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2879          ; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
2880          ; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2881          ; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2882          ;
2883          ; INPUTS:
2884          ;
2885          ;     R5      ADDRESS OF FIRST REGISTER
2886          ;
2887          ; OUTPUTS:
2888          ;
2889          ;     R0      CONTENTS OF TSSR, IF ERROR
2890          ;     CARRY   SET IF INIT WAS OKAY
2891          ;              CLEAR IF FATAL ERROR
2892          ;
2893          ; CALLING SEQUENCE:
2894          ;
2895          ;     MOV     @ADDRESS,R5
2896          ;     JSR     PC,SOFINIT
2897          ;     BCS     CONTINUE
2898          ;     ERRDF                     ;REPORT FATAL ERROR
2899          ;
2900          ;-
2901
2902          SOFINIT::
2903          SAVREG                     ; SAVE THE REGISTERS
2904          MOV     @0,TSSR(R5)        ; DO THE INIT.
2905          JSR     PC,WAITF            ; WAIT FOR SSR
2906          MOV     TSSR(R5),R0        ; GET THE TSSR REGISTER
2907          MOV     R0,R4              ; START SETUP OF EXPECTED TSSR
2908          BIC     @C<HIADDR!OFL>,R4 ; CLEAR OUT UNUSED BITS
2909          BIS     @SSR!NBA,R4        ; R4 HAS EXPECTED CONTENTS
2910          CMP     R4,R0              ; ONLY EXPECTED BITS SET ?
2911          BEQ     5$                 ; BRANCH IF OKAY
2912          CLC                          ; CLEAR THE CARRY FOR ERROR
2913          BR      10$                ; GO TO EXIT
2914          5$: SEC                     ; SET THE CARRY BIT
2915          10$: RTS PC                ; RETURN TO CALLER
```

```

2917          .SBTTL  CHKAMB - CHECK TSSR FOR AMBIGUITY
2918
2919          ;+
2920          ;
2921          ; THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2922          ; FOR AMBIGUITY
2923          ;
2924          ; INPUT:
2925          ;
2926          ;      R0      CONTENTS OF TSSR
2927          ;
2928          ; OUTPUT:
2929          ;
2930          ;      R0      CONTENTS OF TSSR
2931          ;
2932          ;      CARRY   SET - NO AMBIGUITY
2933          ;              CLR - AMBIGUOUS CONTENTS
2934          ;
2935          ; -
2936
2937          CHKAMB:
2938          SAVREG          ;SAVE THE GENERAL REGISTERS
2939          MOV             R0,R4          ;CONTENTS OF TSSR
2940          BIT             #SC,R0        ;IS BIT 15 SET ?
2941          BNE             5$            ;BRANCH IF YES
2942          BIT             #C<NBA!OFL!SSR!HIADDR>,R0 ;ANY OTHER BITS SET ?
2943          BNE             40$          ;MUST BE AN ERROR
2944          BR              45$          ;RETURN WITH SUCCESS
2945          5$:            BIT             #SSR,R0        ;IS READY BIT SET ?
2946          BNE             10$          ;BRANCH IF READY BIT IS SET.
2947          BIT             #BIT5,R0     ;IS FATAL ERROR BIT SET ?
2948          BEQ             40$          ;ERROR IF NOT
2949          BIC             #C<TERCLS,R4   ;CLEAR ALL BUT TERMINATION CODE
2950          CMP             R4,#16        ;ALL THREE BITS MUST BE SET
2951          BNE             40$          ;ERROR IF NOT SET
2952          BR              45$          ;OK IF ALL ARE SET
2953          10$:           BIT             #BIT5,R0        ;IS FATAL ERROR BIT SET ?
2954          BEQ             45$          ;ERROR IF BIT IS SET WITH SSR
2955          BIT             #BIT2!BIT1,R0 ;IS THIS A FUNCTION REJECT
2956          BNE             45$          ;BR, IF TSSR IS OK
2957          40$:           CLC              ;AMBIGUOUS CONTENTS
2958          BR              50$
2959          45$:           SEC              ;SHOW SUCCESS - NO AMBIGUITY
2960          50$:           RTS             PC             ;RETURN TO CALLER
2961
2917 016714
2918 016714
2919 016714
2920 016714
2921 016720 010004
2922 016722 032700 100000
2923 016726 001004
2924 016730 032700 174077
2925 016734 001023
2926 016736 000424
2927 016740 032700 000200
2928 016744 001011
2929 016746 032700 000040
2930 016752 001414
2931 016754 042704 177761
2932 016760 020427 000016
2933 016764 001007
2934 016766 000410
2935 016770 032700 000040
2936 016774 001405
2937 016776 032700 000006
2938 017002 001002
2939 017004 000241
2940 017006 000401
2941 017010 000261
2942 017012 000207
2943 017014

```

J7

SEQ 87

```

2963          .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2964          ;
2965          ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2966          ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2967          ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2968          ;
2969          ;
2970          ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2971          ;
2972          000200          IOKCKIN=BIT7          ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2973          000001          IOKSTP=BIT0          ; EXPECT "STOP" INTERRUPT.
2974          ;
2975          ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2976 017014          000          INTMASK: .BYTE 0
2977          ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2978 017015          000          INTFLAG: .BYTE 0
2979          ;
2980          ; SAVED INTERRUPT VECTOR:
2981 017016          000000          INTVEC: .WORD 0
2982          ; SAVE CPU PC
2983 017020          000000          INTCPC: .WORD 0
2984          ;
2985          ; SUBROUTINE TO ENABLE INTERRUPTS:
2986 017022          010046          ENAINT: MOV R0,-(SP)          ; SAVE R0
2987 017024          013700          002156          MOV IVEC,R0          ; GET POINTER TO VECTORS
2988 017030          012720          017066          MOV @INTR,(R0)+          ; SET UP INTERRUPT VECTOR
2989 017034          012720          000340          MOV @PRI07,(R0)+
2990 017040          012600          MOV (SP)+,R0          ; RESTORE R0
2991 017042          011646          MOV (SP),-(SP)
2992 017044          012766          000000          000002          MOV #0,2(SP)          ; SET CPU TO LEVEL 0
2993 017052          000002          RTI
2994          ;
2995          ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2996 017054          011646          DSBINT: MOV (SP),-(SP)
2997 017056          012766          000340          000002          MOV @PRI07,2(SP)
2998 017064          000002          RTI
2999

```

1K7

CZIKFA TK-25 FRI END FUNC #2  
INTR - INTERRUPT HANDLERS

MACRO M1200 20-APR-84 08:12 PAGE 64

SEQ 88

```

3001          .SBTTL  INTR      - INTERRUPT HANDLERS
3002
3003 017066      BGNSRV  INTR          ;DEFINE INTERRUPT ENTRY
      017066
3004 017066 012737 000001 002172 INTR:: MOV    #1,INTRECV    ;SET FLAG TO SHOW INTERRUPT RECEIVED
3005 017074 105037 017015          CLR    INTFLAG        ;CLEAR FLAG TO SAY WE GOT INTERRUPT
3006 017100 132737 000001 017014          BIT    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
3007 017106 001003          BNE    1$                ;BR IF YES
3008 017110 152737 000001 017015          BISB   #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
3009
3010          ;SAVE REGISTERS, MSG BUFFER, ETC.
3011 017116      1$:
3012 017116          ENDSRV
      017116      L10026:
      017116 000002          RTI
3013
3014

```

L7

```

3016 .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
3017 ;
3018 ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
3019 ;
3020 ; INPUTS:
3021 ;
3022 ; R5 ADDRESS OF FIRST DEVICE REGISTER
3023 ;
3024 ; OUTPUTS:
3025 ;
3026 ; R0 CONTENTS OF LAST TSSR READ
3027 ; CARRY SET - READY BIT SET
3028 ; CLR - TIMEOUT WAITING FOR READY
3029 ;
3030 WAITF:: BREAK ; DO A SUPVSR BREAK FIRST.
3031 017120 104422 TRAP C#BRK
3032 017122 012746 177776 MOV #177776,-(SP) ;BIG MSEC TIMER
3033 017126 012727 000001 DELAY 1 ;DELAY 100US
3034 017132 000000 MOV #1,(PC)+
3035 017134 013727 002116 .WORD 0
3036 017140 000000 MOV L#DLY,(PC)+
3037 017142 005367 177772 .WORD 0
3038 017146 001375 DEC -6(PC)
3039 017150 005367 177756 BNE .-4
3040 017154 001367 DEC -22(PC)
3041 017156 016500 000000 BNE .-20
3042 017162 105700 2$: MOV TSSR(R5),R0 ;READ THE TSSR REGISTER
3043 017164 100421 TSTB R0 ;TEST FOR READY BIT SET
3044 017166 012727 000001 BMI 3$ ; EXIT ON STOP FLAG.
3045 017172 000000 DELAY 1 ; WAIT 100 USEC
3046 017174 013727 002116 MOV #1,(PC)+
3047 017200 000000 .WORD 0
3048 017202 005367 177772 MOV L#DLY,(PC)+
3049 017206 001375 .WORD 0
3050 017210 005367 177756 DEC -6(PC)
3051 017214 001367 BNE .-4
3052 017216 104422 DEC -22(PC)
3053 017220 005316 BNE .-20
3054 017222 001355 BREAK ; DO A SUPVSR BREAK FIRST.
3055 017224 000241 TRAP C#BRK
3056 017226 000401 DEC (SP) ;REDUCE DELAY COUNT
3057 017230 000261 BNE 2$ ;RETRY UNTIL TIMER EXPIRES
3058 017232 005326 CLC ; C = 0, CONTROLLER STILL RUNNING...
3059 017234 000207 BR 4$ ;...OR HUNG-UP AFTER 300 MSEC.
3060 3$: SEC ; C = 1, CONTROLLER IS STOPPED.
3061 4$: DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
3062 RTS PC

```

```

3047          .SBTTL  CHKTSSR - CHECK TSSR FOR READY
3048
3049          ;+
3050          ; THIS ROUTINE WAITS FOR READY IN THE TSSR
3051          ; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
3052
3053          ; INPUT:
3054          ;
3055          ;       R5      ADDRESS OF CSR REGISTERS
3056
3057          ; OUTPUT:
3058          ;
3059          ;       R0      CONTENTS OF TSSR
3060          ;       CARRY   SET - OKAY
3061          ;              CLR - NOT READY AMBIGUOUS, OR SC SET
3062          ;
3063          ; -
3064
3065          CHKTSSR:
3066          JSR    PC, WAITF      ; WAIT FOR READY
3067          BCC    20$           ; BRANCH IF TIME OUT
3068          JSR    PC, CHKAMB     ; TSSR AMBIGUOUS?
3069          BCC    10$           ; BR IF YES
3070          EIT    *SC, R0       ; SPECIAL CONDITION SET?
3071          BEQ    15$           ; BR IF NO
3072          BIT    *(<SCE!BIE!RMR!NXM>, R0, ANY ERROR BITS SET?
3073          BEQ    15$           ; BR IF NO
3074          CLC                    ; SET FAILURE
3075          BR     20$           ;
3076          SEC                    ; SET SUCCESS
3077          RTS    PC           ; RETURN TO CALLER
3078
10$:
15$:
20$:

```

|      |        |        |        |
|------|--------|--------|--------|
| 3047 | 017236 | 004737 | 017120 |
| 3048 | 017236 | 103014 |        |
| 3049 | 017242 | 004737 | 016714 |
| 3050 | 017250 | 103006 |        |
| 3051 | 017252 | 032700 | 100000 |
| 3052 | 017256 | 001405 |        |
| 3053 | 017260 | 032700 | 074000 |
| 3054 | 017264 | 001402 |        |
| 3055 | 017266 | 000241 |        |
| 3056 | 017270 | 000401 |        |
| 3057 | 017272 | 000261 |        |
| 3058 | 017274 | 000207 |        |

N7

CZTKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 67  
XNXM - CHECK FOR NONEXISTENT MEMORY

SEQ 91

```

3080 .SBTTL XNXM - CHECK FOR NONEXISTENT MEMORY
3081 ;*
3082 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
3083 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
3084 ; "C" = 0, ALL ADDRESSES OK.
3085 ;
3086 ; CALL: MOV ADR1,R1
3087 ; MOV ADR2,R2
3088 ; JSR PC,NXM
3089 ; RETURN
3090 ; TEST "C" AND PROCEED.
3091 017276 012737 017330 000004 XNXM: MOV #2$,R04 ; SET BUSERR VECTOR.
3092 017304 012737 000200 000006 MOV #PRI04,R06
3093 017312 005003 CLR R3 ; FLAG.
3094 017314 005711 1$: TST (R1) ; TEST THE ADDRESS(ES).
3095 ; IF ANY TRAP, CONTINUE AT 2$.
3096 017316 020102 CMP R1,R2 ; OTHERWISE, CONTINUE HERE.
3097 017320 001407 BEQ 3$ ; BR IF FINISHED (NO NEXM'S).
3098 017322 062701 000002 ADD #2,R1 ; SET NEXT ADDRESS...
3099 017326 000772 BR 1$ ; ...AND CONTINUE.
3100
3101 017330 005103 2$: COM R3 ; GOT ONE, SET FLAG...
3102 017332 012716 017340 MOV #3$,(SP)
3103 017336 000002 RTI ; ...AND DISMISS INTERRUPT...
3104 017340 3$: CLRVEC #4 ; ...AND GIVE BACK THE VECTOR.
3105 017344 012700 000004 MOV #4,R0
3106 017346 104436 TRAP C$CVEC
3107 017350 001401 TST R3 ; DID WE CATCH ONE ??
3108 017352 000261 BEQ .+4 ; NO, "C" = 0, SKIP NEXT.
3109 017354 000207 SEC ; YES, "C" = 1, (R1) = NEXM ADDR.
3110 RTS PC
3111
3112 .SBTTL TSTLOOP - CHECK ITERATION COUNT
3113 ;*
3114 ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
3115 ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
3116 ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
3117 ;
3118 ; CALL: LOOPTO ARG
3119 ;
3120 ;
3121 017356 TSTLOOP: TST NOITS ; ITERATIONS INHIBITED?
3122 017356 005737 002136 BNE 1$ ; YES.
3123 017362 001006 TST QVP ; NO.
3124 017364 005737 002152 BMI 1$ ; LOOPS DISALLOWED IN QUICK PASS.
3125 017370 100403 DEC LOOPCNT ; BUMP LOOP COUNTER.
3126 017372 005337 002164 BNE 2$
3127 017376 001002
3128 017400 000241 1$: CLC ; LOOP DISALLOWED, OR DONE.
3129 017402 000401 BR 3$
3130 017404 000261 2$: SEC ; LOOP ENABLED.
3131 017406 000207 3$: RTS PC

```

```

3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161 017410
3162 017410 010046
3163 017412 005037 003106
3164 017416 005037 017656
3165 017422 005037 005232
3166 017426 105037 017014
3167 017432 013700 002150
3168 017436 006300
3169 017440 005737 003062
3170 017444 001430
3171 017446 100010
3172 017450 052760 160000 003130
3173 017456
    017456 104455
    017460 000001
    017462 003636
    017464 005176
3174 017466 000407
3175 017470 052760 160001 003130 31:
3176 017476
    017476 104455
    017500 000002
    017502 004233
    017504 000000
3177 017506 012737 177777 003060 21:
3178 017514
    017514 013700 002150
    017520 104451
3179 017522

```

```

      .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
      *
      PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
      INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
      IN THE CURRENT RUN SEQUENCE.
      CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
      INPUT:
      R0      POINTER TO TEST ID ASCIZ STRING
      OUTPUT:
      R5      ADDRESS OF FIRST DEVICE REGISTER
      IMPLICIT OUTPUTS:
      TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
      SIDE EFFECTS:
      INTERRUPT LEVEL IS RAISED TO LEVEL OF
      THE DEVICE UNDER TEST
      -
TSTSETUP::
      MOV     R0, -(SP)      ; SAVE THE TEST ID MESSAGE
      CLR     SIFLAG         ; CLEAR "SOFT INIT" FLAG
      CLR     ERRK           ; CLEAR LOCAL ERROR COUNTER.
      CLR     EXTA           ; CLEAR ERROR EXTENSION FLAG.
      CLRB    INTMASK        ; CLEAR INTERRUPT MASK (CHECK ERROR)
      MOV     UNITN,R0        ; GET THE UNIT NUMBER,
      ASL     R0             ; ... AND MAKE IT A WORD OFFSET.
      TST     NODEV          ; DID STARTUP FIND THE DEVICE?
      BEQ     41             ; BR IF YES
      BPL     31             ; BR IF NOT IDLE
      BIS     @160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
      ERDF    1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
      TRAP    C1ERDF
      .WORD   1
      .WORD   NXR
      .WORD   NXRERR
      BR      21
      BIS     @160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
      ERDF    2,NOINIT        ; DEVICE NOT IDLE
      TRAP    C1ERDF
      .WORD   2
      .WORD   NOINIT
      .WORD   0
      MOV     @-1,DUFLG        ; DROP THE UNIT
      DODU    UNITN
      MOV     UNITN,R0
      TRAP    C1DODU
      DOCLN
      ; ABORT THE PASS

```

C8

CZIKFA JK-25 FRI END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 68-1  
TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

SEQ 93

|      |        |        |        |     |        |             |                                      |
|------|--------|--------|--------|-----|--------|-------------|--------------------------------------|
|      | 017522 | 104444 |        |     | TRAP   | C#DCLN      |                                      |
| 3180 | 017524 | 000423 |        |     | BR     | 5#          |                                      |
| 3181 |        |        |        |     |        |             |                                      |
| 3182 | 017526 |        |        | 4#: | RFLAGS | RO          | ; GET THE OPERATOR FLAGS.            |
|      | 017526 | 104421 |        |     | TRAP   | C#RFLA      |                                      |
| 3183 | 017530 | 032700 | 001000 |     | BIT    | #PNT,RO     | ; PRINT THE TEST NUMBERS?            |
| 3184 | 017534 | 001412 |        |     | BEQ    | 1#          | ; BR IF NO                           |
| 3185 | 017536 | 011600 |        |     | MOV    | (SP),RO     | ; GET THE ID MESSAGE                 |
| 3186 | 017540 |        |        |     | PRINTF | #TNAM,RO    | ; DISPLAY THE TEST ID                |
|      | 017540 | 010046 |        |     | MOV    | RO,-(SP)    |                                      |
|      | 017542 | 012746 | 017604 |     | MOV    | #TNAM,-(SP) |                                      |
|      | 017546 | 012746 | 000002 |     | MOV    | #2,-(SP)    |                                      |
|      | 017552 | 010600 |        |     | MOV    | SP,RO       |                                      |
|      | 017554 | 104417 |        |     | TRAP   | C#PNTF      |                                      |
|      | 017556 | 062706 | 000006 |     | ADD    | #6,SP       |                                      |
| 3187 | 017562 | 005237 | 002162 | 1#: | INC    | TSTCNT      | ; BUMP TEST COUNTER.                 |
| 3188 | 017566 |        |        |     | SETPRI | IPRI        | ; PRIORITY THAT OF DEVICE            |
|      | 017566 | 013700 | 002160 |     | MOV    | IPRI,RO     |                                      |
|      | 017572 | 104441 |        |     | TRAP   | C#SPRI      |                                      |
| 3189 | 017574 | 005726 |        | 5#: | TST    | (SP),       | ; FIX UP THE STACK                   |
| 3190 | 017576 | 013705 | 002154 |     | MOV    | CSRADDR,R5  | ; ADDRESS OF TSV REGISTERS ON UNIBUS |
| 3191 | 017602 | 000207 |        |     | RTS    | PC          |                                      |
| 3192 | 017604 | 045    | 123    | 045 | TNAM;  | .ASCIZ      | '#S#T#A Test'                        |
| 3193 |        |        |        |     |        | .EVEN       |                                      |

```

3195
3196
3197
3198
3199
3200 017620
      017620 104421
3201 017622 030027 020000
3202 017626 001412
3203 017630
      017630 013746 017656
      017634 012746 017660
      017640 012746 000002
      017644 010600
      017646 104417
      017650 062706 000006
3204 017654 000207
3205
3206 017656 000000
3207 017660 045 101 040
3208 017677 105 122 122
3209
3210
3211
3212
3213
3214
3215 017744 005237 017656
3216 017750 017046
3217 017752 013700 002150
3218 017756 006300
3219 017760 062700 003130
3220 017764 005210
3221 017766 032710 007777
3222 017772 001001
3223 017774 005310
3224 017776 012600
3225 020000 000207
3226
3227 020002 010046
3228 020004 013700 002150
3229 020010 006300
3230 020012 016000 003130
3231 020016 042700 170000
3232 020022 020037 002142
3233 020026 103004
3234 020030 023737 017656 002140
3235 020036 103417
3236 020040
      020040 104421
3237 020042 032700 000040
3238 020046 001013
3239 020050 012737 177777 003060
3240 020056
      020056 104455
      020060 000004
      020062 017677

      .SBTTL TSTEND - PRINT ERRORS RECEIVED
      ;
      ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
      ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
      ;
TSTEND: RFLAGS RO
      TRAP C#RFLA
      BIT RO,#IER
      BEQ 1# ; BR IF "IER" NOT SET.
      PRINTF #ESUM,ERRK ; PRINT ERROR COUNT.
      MOV ERRK,-(SP)
      MOV #ESUM,-(SP)
      MOV #2,-(SP)
      MOV SP,RO
      TRAP C#PNTF
      ADD #6,SP
1#: RTS PC

ERRK: 0 ; LOCAL ERROR COUNT.
ESUM: .ASCIZ /#A #D#A ERRORS/
EMAXDU: .ASCIZ /ERROR LIMIT REACHED -- DROPPING UNIT/
      .EVEN

      .SBTTL INCERK - INCREMENT LOCAL ERROR COUNT
      ;
      ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
      ;
INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
      MOV RO,-(SP) ; SAVE RO
      MOV UNITN,RO ; GET UNIT NUMBER,
      ASL RO ; ... AND MAKE IT A WORD OFFSET.
      ADD #ERTABL,RO ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
      INC (RO) ; INCREMENT THE DEVICE ERROR COUNT
      BIT #7777,(RO) ; DID WE OVERFLOW THE FIELD?
      BNE 1# ; BR IF NO.
      DEC (RO) ; YES -- BACK IT UP TO 7777.
1#: MOV (SP)+,RO ; RESTORE RO
      RTS PC ; RETURN TO CALLER.

CKEMAX: MOV RO,-(SP) ; SAVE RO
      MOV UNITN,RO ; GET UNIT NUMBER
      ASL RO ; ... AND MAKE IT A WORD OFFSET
      MOV ERTABL(RO),RO ; GET ERROR TABLE ENTRY
      BIC #170000,RO ; EXTRACT ERROR COUNT FIELD
      CMP RO,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
      BHS 1# ; BR IF YES
      CMP ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
      BLO 2# ; BR IF NO
1#: RFLAGS RO ; GET OPERATOR FLAGS
      TRAP C#RFLA
      BIT #IDU,RO ; IS DROPPING INHIBITED?
      BNE 2# ; BR IF YES.
      MOV #-1,D#FLG ; NO -- DROP THE UNIT
      ERDF 4,EMAXDU
      TRAP C#ERDF
      .WORD 4
      .WORD EMAXDU

```

```

3241 020064 000000          .WORD 0
      020066          DODU UNITN
      020066 013700 002150 MOV UNITN,R0
      020072 104451 TRAP C#DODU
3242 020074          DOCLN
      020074 104444 TRAP C#DCLN
3243 020076 012600 2$: MOV (SP)+,R0      ; RESTORE R0
3244 020100 000207 RTS PC                ; RETURN TO CALLER
3245          .SBTTL FATCHK - INC FATAL ERRORS AND CHECK FOR LIMIT
3246          ;
3247          ;
3248          ; CHECK FATAL COUNTER, AFTER INC, FOR MORE THAN 25
3249          ; ERRORS AND IF OVER CALL UNIT DROP ROUTINE
3250          ;
3251          ;
3252          FATCHK:
3253 020102          SAVREG
3254 020106 013701 002150 MOV UNITN,R1      ;BETTER SAVE THE REGISTERS
3255 020112 006301      ASL R1                ;PICK UP THE UNIT NUMBER
3256 020114 062761 000001 003130 ADD #1,ERTABL(R1) ;MAKE IT INTO A BYTE OFFSET
3257 020122 005237 002170 INC FATFLG        ;ADD 1 TO THE PROPER UNIT'S ERROR COUNTER
3258 020126 023727 002170 000031 CMP FATFLG,#25 ;BUMP FATAL ERROR COUNTER
3259 020134 002406      BLT 9$                ;CHECK AGAINST 25
3260 020136          RFLAGS R0                ;BR, IF LESS THAN 25 ERRORS
      020136 104421 TRAP C#RFLA              ;READ THE FLAGS INTO R0
3261 020140 032700 040000 BIT #BIT14,R0      ;BR, IF LOOP ON ERROR IS SET
3262 020144 001002      BNE 9$                ;OTHERWISE NEVER BE ABLE TO SCOPE ETC.
3263 020146 004737 020154 JSR PC,CKDROP      ;DROP UNIT IF ALLOWED
3264 020152 000207 9$: RTS PC                ;RETURN ETC.
3265          ;
3266          ;
3267          ;

```

F8

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 70  
 CKDROP - CHECK IF UNIT SHOULD BE DROPPED

SEQ 96

```

3269                                     .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
3270
3271                                     ;+
3272                                     ; CHECK IF UNIT SHOULD BE DROPPED
3273 020154 010046                                     ;-
3274 020156                                     CKDROP: MOV      RO,-(SP)
3275 020166                                     FORCERROR 1$,NOTSSR
3276 020170 032700 000040                                     RFLAGS RO
3277 020174 001010                                     TRAP  C$RFLA
3278 020176 011600                                     BIT   $IDU,RO
3279 020200 012737 177777 003060                                     BNE   1$
3280 020206 013700 002150                                     MOV   (SP),RO
3281 020212 104451                                     MOV   $-1,DUFLG
3282 020214 104444                                     DODU   UNITN
3283 020216 012600                                     MOV   UNITN,RO
3284 020220 000207                                     TRAP  C$DODU
3285                                     DOCLN                                     ;ABORT THE PASS
3286                                     TRAP  C$DCLN
3287                                     1$: MOV   (SP)+,RO
3288                                     RTS    PC
3289
3290                                     .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
3291                                     ;
3292                                     ; SUBROUTINE - DETERMINE CONFIGURATION OF TK-25 SYSTEM.
3293 020222 004737 016644                                     ;
3294 020226 000207                                     CONFIG:
3295                                     JSR    PC,SOFINIT
3296                                     RTS    PC
3297

```

G8

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 71  
 KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT

SEQ 97

```

3299          .SBTTL KTON,KTOFF          - ENABLE/DISABLE MEMORY MANAGEMENT
3300
3301          ; SUBROUTINE - ENABLE MEM MGT.
3302          ;
3303 020230 005737 003100          KTON:  TST      KTFLG          ; GOT KT?
3304 020234 001403              BEQ      1$              ; NO.
3305 020236 012737 000001 177572      MOV      #1,SRO          ; YES. ENABLE KT11.
3306 020244 000207              1$:    RTS      PC
3307
3308
3309
3310          ; SUBROUTINE - DISABLE MEM MGT.
3311          ;
3312          ;
3313 020246 005737 003100          KTOFF: TST      KTFLG          ; GOT KT11?
3314 020252 001405              BEQ      1$              ; NO.
3315 020254 000240              NOP
3316 020256 000240              NOP
3317 020260 012737 000000 177572      MOV      #0,SRO          ; DISABLE KT.
3318 020266 000207              1$:    RTS      PC
3319
3320

```

```

3322          .SBTTL  SETMAP - SETUP PAR6 MAPPING
3323
3324          ;
3325          ;
3326          ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
3327          ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
3328          ; IS RETURNED BIASED TO PAR6.
3329          ;
3330          ; INPUTS:
3331          ;
3332          ;      R0      HIGH ORDER ADDRESS BITS
3333          ;      R1      LOW ORDER ADDRESS BITS
3334          ;
3335          ; OUTPUTS:
3336          ;
3337          ;      R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
3338          ;      CARRY   SET IF SUCCESS
3339          ;               CLR IF ERROR
3340          ;
3341          SETMAP:
3342          SAVREG          ;SAVE R1-R4 UNTIL NEXT RETURN
3343          TST             ;SYSTEM HAVE ABOVE 28K?
3344          BEQ             ;BR IF NO
3345          MOV             ;SAVE LOW ORDER BITS
3346          .REPT          6
3347          ASR             ;CONVERT WORD ADDRESS TO 32W BLOCKS
3348          ROR             ;MAKE IT DOUBLE PRECISION
3349          .ENDR
3350          BIC             ;ALINE FOR LOWER 4K BOUNDARY
3351          CMP             ;HIGHER THAN EXISTING MEMORY?
3352          BHS             ;BR IF YES
3353          MOV             ;SETUP MAPPING REGISTER PAR6
3354          BIC             ;SETUP DISPLACEMENT IN PAGE
3355          ADD             ;ADD IN PAR6 BIAS
3356          MOV             ;RETURN IN R0
3357          SEC             ;SET SUCCESS
3358          BR              15$
3359          CLC             ;SET FAILURE
3360          RTS             ;RETURN
3361          PC

```

```

3363 .SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
3364 ;*
3365 ; FILL MEMORY WITH A BACKGROUND PATTERN
3366 ;
3367 ; INPUTS:
3368 ;
3369 ;     R0 = BACKGROUND PATTERN
3370 ;     FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
3371 ;     KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
3372 ;
3373 ; OUTPUTS:
3374 ;
3375 ;     NONE
3376 ;
3377 ;
3378 ; FILLMEM:
3379 ; SAVREG
3380 ; JSR PC,KTOFF ;SAVE R1-R5 UNTIL NEXT RETURN
3381 ; MOV R0,R3 ;DISABLE KT.
3382 ; MOV FREE,R1 ;COPY TEST PATTERN
3383 ; MOV FRESIZ,R2 ;GET FIRST FREE LOCATION
3384 10$: MOV R3,(R1)+ ;SIZE OF FREE SPACE BELOW 28K.
3385 ; DEC R2 ;STORE A BACKGROUND WORD
3386 ; BGT 10$ ;DOSE ALL MEMORY IN FREE SPACE?
3387 ; TST KTFLG ;BR IF NO
3388 ; BEQ 55$ ; GOT KT?
3389 ; JSR PC,KTON ; NO. GET OUT.
3390 ; CLR R0 ; YES. ENABLE KT.
3391 ; MOV PST32W,R1 ;HIGH ORDER ADDRESS START
3392 ; .REPT 6 ;GET >28K START ADDRESS (IN 32W BLOCKS)
3393 ; CLC
3394 ; ROL R1 ;CLEAR C BIT
3395 ; ROL R0 ;CONVERT BLOCKS TO WORDS
3396 ; .ENDR ;MAKE IT DOUBLE PRECISION
3397 ; JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
3398 30$: MOV R3,(R0)+ ;STORE TEST PATTERN IN >28K ADDRESS
3399 ; CMP R0,#160000 ;END OF PAR6 MAPPING AREA?
3400 ; BLO 30$ ;BR IF NO
3401 ; SUB #20000,R0 ;BACKUP INTO PAR6 MAPPING BEGIN
3402 ; ADD #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
3403 ; CMP #KIPAR6,KTFLG ;END OF MEMORY?
3404 ; BEQ 50$ ;BR IF YES
3405 ; JMP 30$ ;KEEP GOING ON ETC.
3406 50$: JSR PC,KTOFF ;DISABLE KT.
3407 55$: RTS PC
3408
3409

```

```

3411 .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
3412 ;
3413 ; COMPARE MEMORY WITH A BACKGROUND PATTERN
3414 ;
3415 ; INPUTS:
3416 ;
3417 ; RO = BACKGROUND PATTERN
3418 ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
3419 ; KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
3420 ;
3421 ; OUTPUTS:
3422 ;
3423 ; CARRY - SET IF NO ERROR
3424 ; CARRY - CLR IF ERROR
3425 ;
3426 ; IMPLICIT OUTPUTS:
3427 ;
3428 ; ERRHI - ERROR HIGH ADDRESS
3429 ; ERRLO - ERROR LOW ADDRESS
3430 ; EXPD - EXPECTED DATA
3431 ; RECV - RECEIVED DATA
3432 ;
3433 CMPMEM:
3434 SAVREG
3435 MOV R0,R3 ;SAVE R1-R5 UNTIL NEXT RETURN
3436 JSR PC,KTOFF ;COPY TEST PATTERN
3437 MOV FREE,R1 ;DISABLE KT.
3438 MOV FRESIZ,R2 ;GET FIRST FREE LOCATION
3439 10$: CMP R3,(R1) ;SIZE OF FREE SPACE BELOW 28K.
3440 BEQ 15$ ;FREE SPACE LOCATION EQUAL TO EXPD?
3441 MOV R1,ERRLO ;BR IF YES
3442 CLR ERRHI ;SAVE ADDRESS IN ERROR
3443 MOV R3,EXPD ;NO HIGH ADDRESS
3444 MOV (R1),RECV ;SAVE EXPD FOR ERROR REPORT
3445 BR 50$ ;SAVE RECV FOR ERROR REPORT
3446 15$: TST (R1)+ ;POINT TO NEXT ADDRESS
3447 DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
3448 BGT 10$ ;BR IF NO
3449 TST KTFLG ;GOT KT?
3450 BEQ 55$ ;NO. GET OUT.
3451 JSR PC,K10N ;YES. ENABLE KT.
3452 CLR R0 ;HIGH ORDER ADDRESS START
3453 MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
3454 .REPT 6
3455 ROL R1 ;CONVERT BLOCKS TO WORDS
3456 ROL R0 ;MAKE IT DOUBLE PRECISION
3457 .ENDR
3458 BIC #177,R1 ;ALINE 4K BOUNDARY
3459 MOV R0,-(SP) ;SAVE HIGH ORDER
3460 MOV R1,-(SP) ;SAVE LOW ORDER
3461 JSR PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
3462 MOV R0,R4 ;COPY ADDRESS BIASED TO PAR6
3463 MOV (SP)+,R1 ;RESTORE LOW ORDER IN NON PAR6 FORMAT
3464 MOV (SP)+,R0 ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
3465 30$: CMP R0,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
3466 BEQ 32$ ;BR IF YES
3467 MOV R0,ERRHI ;SAVE HIGH ORDER IN ERROR

```

K8

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 74-1  
 CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

SEQ 101

|      |        |        |        |        |      |                |                                |
|------|--------|--------|--------|--------|------|----------------|--------------------------------|
| 3468 | 020740 | 010137 | 002204 |        | MOV  | R1,ERRLO       | SAVE LOW ORDER IN ERROR        |
| 3469 | 020744 | 010337 | 002176 |        | MOV  | R3,EXPD        | SAVE EXPD FOR ERROR REPORT     |
| 3470 | 020750 | 011437 | 002200 |        | MOV  | (R4),RECV      | SAVE RECV FOR ERROR REPORT     |
| 3471 | 020754 | 000421 |        |        | BR   | 50\$           |                                |
| 3472 | 020756 | 062701 | 000002 | 32\$:  | ADD  | #2,R1          | UPDATE NON PAR6 ADDRESS        |
| 3473 | 020762 | 005500 |        |        | ADC  | R0             | MAKE IT DOUBLE PRECISION ADD   |
| 3474 | 020764 | 062704 | 000002 |        | ADD  | #2,R4          | UPDATE PAR FORMAT ADDRESS      |
| 3475 | 020770 | 020427 | 160000 |        | CMP  | R4,#160000     | END OF PAR6 MAPPING AREA?      |
| 3476 | 020774 | 103755 |        |        | BLO  | 30\$           | BR IF NO                       |
| 3477 | 020776 | 162704 | 020000 |        | SUB  | #20000,R4      | BACKUP INTO PAR6 MAPPING BEGIN |
| 3478 | 021002 | 062737 | 000200 | 172354 | ADD  | #200,8#KIPAR6  | POINT TO NEXT 4K BLOCK >28K.   |
| 3479 | 021010 | 023737 | 172354 | 003100 | CMP  | 8#KIPAR6,KTFLG | END OF MEMORY?                 |
| 3480 | 021016 | 101744 |        |        | BLOS | 30\$           | BR IF NO                       |
| 3481 | 021020 | 004737 | 020246 | 50\$:  | JSR  | PC,KTOFF       | TURN OFF MEMORY MAPPING        |
| 3482 | 021024 | 000241 |        |        | CLC  |                | SET FAILURE                    |
| 3483 | 021026 | 000403 |        |        | BR   | 60\$           |                                |
| 3484 | 021030 | 004737 | 020246 | 55\$:  | JSR  | PC,KTOFF       | TURN OFF MEMORY MAPPING        |
| 3485 | 021034 | 000261 |        |        | SEC  |                | SET SUCCESS                    |
| 3486 | 021036 | 000207 |        | 60\$:  | RTS  | PC             |                                |
| 3487 |        |        |        |        |      |                |                                |

L8

CZTKFA TK-25 FRT END FUNC #2  
REGSAV - SAVE R1-R5 ON STACK

MACRO M1200 20-APR-84 08:12 PAGE 75

SEQ 102

```

3489          .SBTTL REGSAV - SAVE R1-R5 ON STACK
3490          ;+
3491          ;
3492          ;ROUTINE TO
3493          ;SAVE R1 THROUGH R5 ON THE STACK
3494          ;
3495          ;CALLING SEQUENCE:
3496          ;
3497          ;      JSR      R5,REGSAV
3498          ;
3499          ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
3500          ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
3501          ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
3502          ;REGISTERS.
3503          ;
3504          ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
3505          ;CALLED VIA A JSR PC INSTRUCTION
3506          ;
3507          ;-
3508
3509 021040      REGSAV:
3510 021040      BREAK          ;LOOK FOR CNTL C
3511 021040      TRAP          C#BRK
3512 021042      MOV          R4,-(SP)
3513 021044      MOV          R3,-(SP)
3514 021046      MOV          R2,-(SP)
3515 021050      MOV          R1,-(SP)
3516 021052      MOV          R5,-(SP)
3517 021054      MOV          10.(SP),R5
3518 021060      JSR          PC,0(SP)+
3519 021062      MOV          (SP)+,R1
3520 021064      MOV          (SP)+,R2
3521 021066      MOV          (SP)+,R3
3522 021070      MOV          (SP)+,R4
3523 021072      MOV          (SP)+,R5
3524 021074      BREAK
3525 021074      TRAP          C#BRK
3526 021076      RTS          PC          ;LOOK FOR CNTL C

```

000012

```

3527 .SBTTL GETPAT - GET 8 BIT PATTERN FROM OPERATOR
3528 ;+
3529 ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
3530 ;
3531 ;INPUTS:
3532 ;
3533 ;    NONE.
3534 ;
3535 ;OUTPUTS:
3536 ;
3537 ;    R0    OCTAL NUMBER FROM THE OPERATOR
3538 ;
3539 ;CALLING SEQUENCE:
3540 ;
3541 ;    JSR    PC,GETPAT
3542 ;
3543 ;-
3544
3545 GETPAT::
3546 021100 SAVREG ;SAVE THE GENERAL REGISTERS
3547 021100 1$: GMANID DATASC,PATDAT,0,377,0,377,NO
3548 021104 TRAP C$GMAN
3549 021104 104443 BR 10000$
3550 021106 000406 .WORD PATDAT
3551 021110 021134 .WORD T$CODE
3552 021112 000022 .WORD DATASC
3553 021114 021136 .WORD 377
3554 021116 000377 .WORD T$LOLIM
3555 021120 000000 .WORD T$HILIM
3556 021122 000377 10000$: BNCOMPLETE 1$ ;RETRY IF ERROR
3557 021124 BCC 1$
3558 021124 103367 MOV PATDAT,R0 ;DATA PATTERN FROM OPERATOR
3559 021126 013700 021134 RTS PC ;RETURN TO CALLER
3560 021132 000207
3561
3562 ;+
3563 ;LOCAL DATA AREA
3564 ;-
3565
3566 021134 000000 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
3567 021136 105 116 124 DATASC: .ASCIZ 'ENTER DATA PATTERN'
3568 .EVEN
3569

```

N8

(ZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 77  
 (ETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

SEQ 104

```

3561 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
3562 ;*
3563 ;ROUTINE TO ISSUE A MENU AND GET
3564 ;THE OPERATOR'S RESPONSE.
3565 ;
3566 ;INPUTS:
3567 ;
3568 ; R0 ADDRESS OF ASCIZ STRING OF MENU
3569 ; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
3570 ;
3571 ;OUTPUTS:
3572 ;
3573 ; R0 NUMBER OF THE OPERATOR'S SELECTION
3574 ;
3575 ;-
3576 GETSEL::
3577 SAVREG ;SAVE GENERAL REGISTERS
3578 MOV R0,R2 ;SAVE THE MENU ADDRESS
3579 MOV R2,R3 ;START OF MENU STRING
3580 TST (R3) ;END OF ASCII ?
3581 BEQ 3$ ;BRANCH IF ALL LINES DISPLAYED
3582 PRINTF #SELASC,(R3)+ ;DISPLAY THE MENU
      MOV (R3)+,-(SP)
      MOV #SELASC,-(SP)
      MOV #2,-(SP)
      MOV SP,R0
      TRAP C$PNTF
      ADD #6,SP
      BR 2$
3583 2$:
3584 3$: GMANID MENASC,MENRES,D,-1,0,-1,NO
      TRAP C$GMAN
      BR 10001$
      .WORD MENRES
      .WORD T$CODE
      .WORD MENASC
      .WORD -1
      .WORD T$LOLIM
      .WORD T$HILIM
3585 10001$: BNCOMPLETE 1$ ;RETRY IF ERROR
      BCC 1$
      MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
      CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
      BLOS 5$ ;BRANCH IF OK
      PRINTF #MENERR ;DISPLAY ERROR MESSAGE
      MOV #MENERR,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C$PNTF
      ADD #4,SP
      BR 1$ ;RETRY
3590 1$: RT$ PC ;RETURN TO CALLER
3591 5$:
3592 045 MENERR: .ASCIZ '*** Menu Selection Too Large ***'
3593 045 SELASC: .ASCIZ 'N#T'
3594 164 MENASC: .ASCIZ 'Enter Menu Selection: '
3595 .EVEN
3596 MENRES: .WORD 0

```

B9

C2TKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 78  
CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

SEQ 105

```

3598                                     .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
3599                                     ;*
3600                                     ;
3601                                     ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
3602                                     ;
3603                                     ;INPUT:
3604                                     ;
3605                                     ;      NONE.
3606                                     ;
3607                                     ;OUTPUT:
3608                                     ;
3609                                     ;      CARRY    0      MANUAL INTERVENTION NOT ALLOWED
3610                                     ;              1      MANUAL INTERVENTION IS OK
3611                                     ;
3612                                     ;SIDE EFFECTS:
3613                                     ;
3614                                     ;      A MESSAGE IS DISPLAYED WARNING THAT TEST IS
3615                                     ;      NOT EXECUTED IF MANUAL INTERVENTION IS NOT
3616                                     ;      ALLOWED.
3617                                     ;
3618                                     ;-
3619
3620 021404                               CHKMAN:
3621 021404                               SAVREG                               ;SAVE THE REGISTERS
3622 021410                               MANUAL                           ;SEE IF MANUAL INTERVENTION OK
3623 021410 104450                       TRAP    C#MANI
3624 021412                               BCOMPLETE 11                      ;BRANCH IF ALLOWED
3625 021412 103411                       BCS     11
3626 021414                               PRINTF  #NOMAN                     ;PRINT THE WARNING MESSAGE
3627 021414 012746 021440                 MOV     #NOMAN, -(SP)
3628 021420 012746 000001                 MOV     #1, -(SP)
3629 021424 010600                         MOV     SP, R0
3630 021426 104417                         TRAP    C#PRINTF
3631 021430 062706 000004                 ADD     #4, SP
3632 021434 000241                         CLC
3633 021436 000207                         11:  RTS     PC                      ;CLEAR CARRY FOR ERROR
3634                                     ;RETURN
3635
3636 021440 045 116 045 NOMAN: .ASCIZ  '### Manual Intervention not Allowed - Test Aborted ###'
3637                                     .even

```

C9

EZTKEA TK-25-FBT-END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 79

SEQ 106

```

3631          .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
3632
3633          ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3634          ;
3635 021534      ENVIRN: MEMORY  R0
                 TRAP        C0MEM
3636 021534      104431          MOV    R0,FREE          ; GET 1ST FREE ADDRESS...
3637 021536 010037 003072      ADD    #2,FREE
3638 021542 062737 000002 003072  MOV    (R0),FRESIZ    ; ...AND WORD COUNT.
3639 021550 011037 003074      SUB    #4,FRESIZ
3640 021554 162737 000004 003074  MOV    L#UNIT,R2      ; GET NUMBER OF UNITS
3641 021562 013702 002012      SUB    #7,FRESIZ        ; TAKE AWAY 7 WORDS PER UNIT
3642 021566 162737 000007 003074 104: DEC    R2
3643 021574 005302          BNE     104
3644 021576 001373          MOV    FREE,R0              ;GET FIRST FREE ADDRESS
3645 021600 013700 003072      MOV    FRESIZ,R0        ;POINT TO LAST FREE ADDRESS
3646 021604 063700 003074      ADD    FRESIZ,R0
3647 021610 162700 000002      SUB    #2,R0            ;BACKUP 1 WORD
3648 021614 010037 003076      MOV    R0,FREEHI        ;STORE LAST FREE ADDRESS
3649 021620 000207          RTS     PC                  ;RETURN

```

```

3651                                     .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
3652                                     ;*
3653                                     ;
3654                                     ;ROUTINE TO INIT KT-11
3655                                     ;
3656                                     ;-
3657
3658 021622                                     KTINIT:
3659 021622 005037 003100                     CLR      KTFLG          ; INIT >28K MEMORY FLAG
3660 021626 005037 003102                     CLR      KTENABLE       ; INIT TEST >28K FLAG
3661 021632 023727 002120 001577             CMP      L#HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
3662 021640 101444                             BLOS     9#              ; NO.
3663 021642 013700 000004                     MOV      @ERRVEC,R0      ; SAVE OLD ERR VEC PTR.
3664 021646 012737 021740 000004             MOV      #2#,@ERRVEC    ; SET ERR VEC PTR.
3665 021654 005737 177572                     TST      @SRO          ; GOT KT11?
3666 021660 000240                             NOP                      ; (TRAP IF NO).
3667 021662 013737 002120 003100             MOV      L#HIME,KTFLG    ; YES. SET KT FLAG.
3668 021670 042737 000177 003100             BIC      #177,KTFLG
3669 021676 010037 000004                     MOV      R0,@ERRVEC    ; RESTORE OLD ERR VEC PTR.
3670 021702 005000                             CLR      R0          ; R0 = AR DATA.
3671 021704 012701 172340                     MOV      @KIPAR0,R1      ; R1 = KI REGS PTR.
3672 021710 012761 077406 177740 1#         MOV      #77406,-40(R1) ; SET DESCRIPTOR REG.
3673 021716 010021                             MOV      R0,(R1)+      ; SET KIPAR REG.
3674 021720 062700 000200                     ADD      #200,R0        ; BUMP AR DATA BY "4K".
3675 021724 020027 002000                     CMP      R0,#2000       ; AT "I/O"?
3676 021730 001367                             BNE      1#              ; NO.
3677 021732 012741 177600                     MOV      #177600,-(R1)    ; YES. SET KTPAR7 FOR I/O.
3678 021736 000405                             BR       9#
3679
3680 021740 012716 021746                     2#         MOV      #6#,(SP)    ; SET UP RETURN
3681 021744 000002                             RTI                      ; RTI TO NEXT LOCATION
3682
3683 021746 010037 000004                     6#         MOV      R0,@ERRVEC    ; RESTORE OLD ERR VEC PTR.
3684
3685 021752 000207                     9#         RTS      PC
3694
3695
3701

```

E9

CZTKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 81  
PROTECTION TABLE

SEQ 108

```
3703 .SBTTL PROTECTION TABLE
3704 021754 BGNPROT
      021754 L$PROT::
3705 021754 177777 177777 177777 .WORD -1, -1, -1, -1 ;NO DEVICE PROTECTION REQUIRED.
3706 021764 ENDPROT
3707
```

```

3709          .SBTTL  INITIALIZE SECTION
3710
3711          ;**
3712          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3713          ;AT THE BEGINNING OF EACH PASS.
3714
3715          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
3716          ;IF "CONTINUE", NOTHING IS REQUIRED.
3717
3718          ;--
3719          ;+
3720          ;INSERT TEMPORARY JUMP TO ODT
3721          ;-
3722          BGNINIT
3723          L$INIT::
3724          40$:
3725          MOV     #EPRT1,EPRTSW      ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
3726          CLR     SIFLAG             ;CLEAR "SOFT INIT" FLAG
3727          CLR     KTENABLE           ;CLEAR TEST ABOVE 28K FLAG
3728          CLR     RAMSIZ             ;CLEAR RAM SIZE FOR RAMERR ROUTINE
3729          READDEF #EF.CONTINUE
3730          MOV     #EF.CONTINUE,R0
3731          TRAP    C$REFG
3732          BNCOMPLETE 1$
3733          BCC     1$
3734          CMP     UNITN,L$UNIT       ;UNIT IN RANGE?
3735          BHIS    4$                ;BR IF NO.
3736          TST     DUFLG             ;DROPPED UNIT?
3737          BMI     NXTU              ;BR IF YES
3738          MOV     UNITN,R1
3739          ASL     R1
3740          TST     ERTABL(R1)
3741          BEQ     SETU
3742          BIT     #BIT14,ERTABL(R1) ;DROPPED?
3743          BNE     NXTU
3744          EXIT     INIT              ;DO NOTHING IF "CONTINUE".
3745          TRAP    C$EXIT
3746          .WORD   L10030-.
3747          1$:
3748          READDEF #EF.NEW
3749          MOV     #EF.NEW,R0
3750          TRAP    C$REFG
3751          BNCOMPLETE NXTU            ;TAKE NEXT UNIT IF NOT NEW PASS.
3752          BCC     NXTU
3753          READDEF #EF.START
3754          MOV     #EF.START,R0
3755          TRAP    C$REFG
3756          BCOMPLETE 2$
3757          BCS     2$
3758          READDEF #EF.RESTART
3759          MOV     #EF.RESTART,R0
3760          TRAP    C$REFG
3761          BNCOMPLETE 31$
3762          BCC     31$
3763          2$:
3764          BRESET
3765          TRAP    C$RESET
3766          CLR     TSTCNT             ;NUMBER OF TESTS RUN IN PASS

```

```

3750 022122 005037 002170          CLR    FATFLG      ;RESET FLAG TO ZERO "FATAL ERRORS"
3751 022126 005037 003332          CLR    SKIPT       ;CLEAR THE SUBTEST "SKIPPER"
3752 022132          20$:          MOV    # -1,QVP      ;...QUICK VERIFY...
3753 022132 012737 177777 002152  JSR    PC,ENVIRN    ;SET ENVIRONMENT.
3754 022140 004737 021534          JSR    PC,KTINIT    ;INITIALIZE KT MEMORY MANAGEMENT
3755 022144 004737 021622          MOV    #ERTABL,R0
3756 022150 012700 003130          CLR    (R0)+       ;CLEAR THE ERROR TABLE
3757 022154 005020          30$:    CMP    R0,#ERTABE
3758 022156 020027 003330          BLO    30$
3759 022162 103774          BR      4/
3760 022164 000404          31$:    CLR    R,VP
3761 022166 005037 002152          JMP    PASRPT       ;GO REPORT THE STATUS
3762 022172 000137 022242
3763
3764 022176          4$:          NEWPAS: MOV    # -1,UNITN    ;INIT UNIT NUMBER...
3765 022176 012737 177777 002150  CLR    DEVCNT     ;CLEAR COUNT OF DEVICES RUNNING
3766 022204 005037 002166          NXTU:  BREAK
3767 022210          TRAP    C#BRK
3768 022212 005237 002150          INC    UNITN
3769 022216 023737 002150 002012  CMP    UNITN,L#UNIT  ;...AND SET NEXT UNIT NUMBER.
3770 022224 103423          BLO    SETU
3771 022226 012737 177777 003060  MOV    # -1,DUFLG
3772 022234 000401          BR      11$
3773 022236          DOCLN
3774 022240 000240          11$:    TRAP    C#DCLN
3775 022242          NOP
3776 022242 023727 002012 000001  PASRPT: CMP    L#UNIT,#1    ;HOW MANY UNITS SELECTED?
3777 022250 101752          BLOS    NEWPAS      ;BR IF ONLY 1
3778 022252 005737 002166          TST    DEVCNT     ;ARE ANY STILL RUNNING?
3779 022256 001747          BEQ    NEWPAS      ;BR IF NO
3780 022260          RFLAGS    RO
3781 022262 032700 000100          TRAP    C#RFLA
3782 022266 001343          BIT     #ISR,R0
3783          BNE    NEWPAS      ;SHOULD WE PRINT STATISTICS
3784 022270          DORPT
3785 022272 104424          TRAP    C#DRPT
3786 022274 000741          BR      NEWPAS
3787
3788 022274          10$:          SETU:  GPHARD  UNITN,R0    ;GET UNIT N P-TABLE POINTER.
3789 022274 013700 002150          MOV    UNITN,R0
3790 022300 104442          TRAP    C#GPHRD
3791 022302          BNCOMPLETE NXTU
3792 022302 103342          BCC    NXTU
3793 022304 005037 003060          CLR    DUFLG
3794 022310 005237 002166          INC    DEVCNT
3795 022314 012001          MOV    (R0)+,R1
3796 022316 010137 002154          MOV    R1,CSRADDR
3797          ;GET 1ST REGISTER ADDRESS.
3798          ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
3799 022322 012001          MOV    (R0)+,R1
3800 022324 011002          MOV    (R0),R2
3801 022326 010237 002160          MOV    R2,IPRI
3802 022332 010137 002156          MOV    R1,IVEC
3803 022336 012721 017066          MOV    #INTR,(R1),
3804          ;GET VECTOR ADDRESS.
3805          ;GET INTERRUPT PRIORITY
3806          ;SET INTERRUPT PRIORITY.
3807          ;SET INTERRUPT VECTOR POINTER...
3808          ;...VECTOR...

```

```
3800 022342 010221          MOV     R2,(R1)+      ;...AND PRIORITY.
3801
3802 022344          1$:
3803          ;          TST     QVP              ;1ST PASS ??
3804          ;          BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.
3805
3806          ;
3807          ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
3808          ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
3809          ;
3810 022344 013701 002150      MOV     UNITN,R1
3811 022350 006301          ASL     R1
3812 022352 052761 100000 003130  BIS     #BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
3813 022360 005037 005232      CLR     EXTA          ;CLEAR ERROR EXTENSION FLAG.
3814 022364 023727 002012 000001  CMP     L$UNIT,#1      ;ARE WE TESTING MULTIPLE UNITS?
3815 022372 101416          BLOS    10$          ;BR IF NO.
3816 022374          RFLAGS    R0              ;YES -- GET OPERATOR FLAGS.
3817 022376 032700 001000      TRAP    C$RFLA
3818 022402 001412          BIT     #PNT,R0        ;SHOULD WE PRINT UNIT #?
3819 022404          BEQ     10$          ;BR IF NOT.
3820          PRINTF    #PUNIT,UNITN          ;PRINT THE UNIT #
3821          MOV     UNITN,-(SP)
3822          MOV     #PUNIT,-(SP)
3823          MOV     #2,-(SP)
3824          MOV     SP,R0
3825          TRAP    C$PNIF
3826          ADD     #6,SP
3827          10$:
3828          CLR     NODEV
3829          MOV     CSRADDR,R1          ;ADDRESS OF FIRST REGISTER
3830          MOV     R1,R2              ;START OF REGISTERS
3831          ADD     #TSSR,R2          ;ADDRESS OF TSSR REGISTER
3832          JSR     PC,XNXM          ;TEST BOTH CONTROLLER REGISTERS...
3833          BCC     2$              ;...AND BR IF ALL OK.
3834          MOV     R1,NODEV          ;FLAG DEVICE AS NON-EXISTENT
3835          MOV     #-1,DUFLG        ;DROP THIS UNIT.
3836          2$:
3837          ;
3838          ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
3839          5$:
3840          SETPRI    #PRI00          ;ENABLE INTERRUPTS.
3841          MOV     #PRI00,R0
3842          TRAP    C$SPRI
3843          ENDINIT
3844          L10030:
3845          TRAP    C$INIT
3846          045 116 045 PUNIT: .ASCIZ /#N#N#A***** TESTING UNIT #D2#A *****/
3847          .EVEN
```

```

3839                                     .SBTTL  ADD AND DROP UNITS SECTIONS
3840
3841                                     ;++
3842                                     ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3843                                     ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
3844                                     ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
3845                                     ;--
3846 022544                                     BGNAU
3847 022544                                     L$AU::
3848 022546 010001                               MOV     R0,R1           ; GET UNIT TO BE ADDED (R0)
3849 022550 006301                               ASL      R1              ; MAKE IT A WORD INDEX
3850 022556 052761 100000 003130                BIS      #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
3851 022564 042761 040000 003130                BIC      #40000,ERTABL(R1) ; CLEAR THE "DROPPED" BIT
3852 022564 010046                               PRINTF   #1$,R0
3853 022566 012746 022612                               MOV     R0,-(SP)
3854 022572 012746 000002                               MOV     #1$,-(SP)
3855 022576 010600                               MOV     #2,-(SP)
3856 022600 104417                               MOV     SP,R0
3857 022602 062706 000006                               TRAP    C$PNTF
3858 022606 000167                               ADD      #6,SP
3859 022610 000026                               EXIT     AU
3860 022612 045      116      045 1$:          .WORD    J$JMP
3861 022612 045      116      045 1$:          .WORD    L10031-2-.
3862 022612 045      116      045 1$:          .ASCIZ   /#N$A UNIT #D$A ADDED/
3863 022612 045      116      045 1$:          .EVEN
3864
3865                                     ENDAU                                     ; UNUSED.
3866 022640                                     L10031:
3867 022640 104452                               TRAP     C$AU
3868
3869                                     ;++
3870                                     ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3871                                     ; TO BE REMOVED FROM THE TEST LIST.
3872                                     ;
3873                                     ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
3874                                     ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
3875                                     ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
3876                                     ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
3877                                     ; WHICH ARE STILL ACTIVE.
3878                                     ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
3879                                     ;--
3880 022642                                     BGNDU
3881 022642                                     L$DU::
3882 022642 012737 177777 003060                MOV     #-1,DUFLG
3883 022650 010001                               MOV     R0,R1
3884 022652 006301                               ASL      R1
3885 022654 052761 140000 003130                BIS      #140000,ERTABL(R1) ; SAY DROPPED
3886 022662 000240 000240 000240                240,240,240 ; ??????????
3887 022670 010046                               PRINTF   #1$,R0
3888 022672 012746 022716                               MOV     R0,-(SP)
3889 022676 012746 000002                               MOV     #1$,-(SP)
3890 022702 010600                               MOV     #2,-(SP)
3891 022704 104417                               MOV     SP,R0
3892 022706 062706 000006                               TRAP    C$PNTF
3893 022712 000167                               ADD      #6,SP
3894 022712 000030                               EXIT     DU
3895 022714 000030                               .WORD    J$JMP
3896 022714 000030                               .WORD    L10032-2-.

```

```

3876 022716      045      116      045 1$: ,ASCIZ /NMA UNIT DMA DROPPED/
3877                                     ,EVEN
3878 022746                                     ENDDU
      022746                                     L10032: TRAP C$DU
      022746 104453
3879
3880                                     ;**
3881                                     ; AUTO-DROP CODE SECTION.
3882 022750                                     ;--
      022750                                     BGNAU19
3883 022750 012703 000550                                     L$AUTO:;
3884 022754 004737 017120 10$: MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
3885 022760 103420 BCS 20$ ;WAIT FOR SSR TO SET
3886 022762 DELAY 250. ;LEAVE WHEN SSR IS SET
      022762 012727 000372 MOV #250.,(PC)+ ;WAIT FOR .25 SECONDS
      022766 000000 .WORD 0
      022770 013727 002116 MOV L$DLY,(PC)+
      022774 000000 .WORD 0
      022776 005367 177772 DEC -6(PC)
      023002 001375 BNE .-4
      023004 005367 177756 DEC -22(PC)
      023010 001367 BNE .-20
3887 023012 005303 DEC R3 ;BUMP COUNTER DOWN
3888 023014 001357 BNE 10$ ;KEEP GOING
3889 023016 004737 020154 JSR PC,CKDROP ;TRY AND DROP UNIT
3890 023022
3891 023022 20$: ENDAUTO ; UNUSED.
      023022 L10033:
      023022 104461 TRAP C$AUTO

```

```

3893                                     .SBTTL  CLEAN-UP AND REPORT CODING SECTIONS
3894
3895
3896                                     ;**
3897                                     ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
3898                                     ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
3899                                     ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
3900                                     ;--
3900 023024                                     BGNCLN
3901 023024                                     L$CLEAN::
3902 023030 005737 003060                     TST     DUFLG      ; "DROPPED" FLAG IS SET ON...
3903                                     BMI     1$          ; ...AND GROSS CONTROLLER FAULT...
3904                                     ; ...DON'T TRY TO XCT CLEANUP CODE.
3905 023032 013705 002154                     MOV     CSRADDR,R5    ; ADDRESS OF TSV REGISTERS ON UNIBUS
3906 023036 012765 000000 000000             MOV     #0,ISSR(R5)    ; DO SOFT INIT
3907 023044 004737 017120                     JSR     PC,WAITF
3908 023050                                     1$:
3909 023050                                     2$:
3909 023050                                     L10034:
3909 023050 104412                             TRAP     C$CLEAN
3910
3911                                     ;**
3912                                     ; THE REPORT CODING SECTION CONTAINS THE
3913                                     ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3914                                     ;--
3914 023052                                     BGNRPT
3915 023052                                     L$RPT::
3915 023052 012746 023314                     PRINTS  #DEVSUM
3915 023056 012746 000001                     MOV     #DEVSUM, -(SP)
3915 023062 010600                             MOV     #1, -(SP)
3915 023064 104416                             MOV     SP,R0
3915 023066 062706 000004                     TRAP     C$PNTS
3916 023072 010246                             ADD     #4,SP
3917 023074 010346                             MOV     R2, -(SP)
3918 023076 010446                             MOV     R3, -(SP)
3919 023100 012704 003130                     MOV     R4, -(SP)
3920 023104 005003                             MOV     #ERTABL,R4    ; GET START OF ERROR TABLE.
3921 023106 011402                             CLR     R3        ; CLEAR UNIT NUMBER
3922 023110 001467                             1$: MOV     (R4),R2    ; GET ERROR TABLE ENTRY & TEST IT.
3923 023112 100066                             BEQ     4$        ; ZERO IF UNIT NOT RUN
3924 023114 032702 040000                     BPL     4$
3925 023120 001015                             BIT     #BIT14,R2    ; WAS UNIT DROPPED?
3926 023122 042702 170000                     BNE     2$        ; BR IF YES
3927 023126 010246                             BIC     #C7777,R2    ; GET ERROR COUNT FIELD
3927 023130 010346                             PRINTS  #DEVONL,R3,R2 ; PRINT
3927 023132 012746 023351                     MOV     R2, -(SP)
3927 023136 012746 000003                     MOV     R3, -(SP)
3927 023142 010600                             MOV     #DEVONL, -(SP)
3927 023144 104416                             MOV     SP,R0
3927 023146 062706 000010                     TRAP     C$PNTS
3928 023152 000446                             ADD     #10,SP
3929 023154 020227 160000                     BR      4$
3930 023160 001012                             2$: CMP     R2,#160000 ; WAS UNIT NON-EXISTENT?
3931 023162 010346                             BNE     3$        ; BR IF NO
3931 023164 012746 023421                     PRINTS  #DEVNXR,R3
3931 023164 012746 023421                     MOV     R3, -(SP)
3931 023164 012746 023421                     MOV     #DEVNXR, -(SP)

```

L9

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 84-1  
CLEAN-UP AND REPORT CODING SECTIONS

SEQ 115

|      |        |        |        |         |         |               |                                                     |
|------|--------|--------|--------|---------|---------|---------------|-----------------------------------------------------|
|      | 023170 | 012746 | 000002 |         | MOV     | #2,-(SP)      |                                                     |
|      | 023174 | 010600 |        |         | MOV     | SP,R0         |                                                     |
|      | 023176 | 104416 |        |         | TRAP    | C#PNTS        |                                                     |
|      | 023200 | 062706 | 000006 |         | ADD     | #6,SP         |                                                     |
| 3932 | 023204 | 000431 |        |         | BR      | 4#            |                                                     |
| 3933 | 023206 | 020227 | 160001 | 3#:     | CMP     | R2,#160001    | ; WAS UNIT NOT READY AT STARTUP?                    |
| 3934 | 023212 | 001012 |        |         | BNE     | 30#           | ; BR IF NO.                                         |
| 3935 | 023214 |        |        |         | PRINTS  | #DEVNRD,R3    |                                                     |
|      | 023214 | 010346 |        |         | MOV     | R3,-(SP)      |                                                     |
|      | 023216 | 012746 | 023503 |         | MOV     | #DEVNRD,-(SP) |                                                     |
|      | 023222 | 012746 | 000002 |         | MOV     | #2,-(SP)      |                                                     |
|      | 023226 | 010600 |        |         | MOV     | SP,R0         |                                                     |
|      | 023230 | 104416 |        |         | TRAP    | C#PNTS        |                                                     |
|      | 023232 | 062706 | 000006 |         | ADD     | #6,SP         |                                                     |
| 3936 | 023236 | 000414 |        |         | BR      | 4#            |                                                     |
| 3937 | 023240 | 042702 | 170000 | 30#:    | BIC     | #C7777,R2     |                                                     |
| 3938 | 023244 |        |        |         | PRINTS  | #DEVDR0,R3,R2 |                                                     |
|      | 023244 | 010246 |        |         | MOV     | R2,-(SP)      |                                                     |
|      | 023246 | 010346 |        |         | MOV     | R3,-(SP)      |                                                     |
|      | 023250 | 012746 | 023564 |         | MOV     | #DEVDR0,-(SP) |                                                     |
|      | 023254 | 012746 | 000003 |         | MOV     | #3,-(SP)      |                                                     |
|      | 023260 | 010600 |        |         | MOV     | SP,R0         |                                                     |
|      | 023262 | 104416 |        |         | TRAP    | C#PNTS        |                                                     |
|      | 023264 | 062706 | 000010 |         | ADD     | #10,SP        |                                                     |
| 3939 | 023270 | 062704 | 000002 | 4#:     | ADD     | #2,R4         |                                                     |
| 3940 | 023274 | 005203 |        |         | INC     | R3            |                                                     |
| 3941 | 023276 | 020427 | 003330 |         | CMP     | R4,#ERTABE    |                                                     |
| 3942 | 023302 | 103701 |        |         | BLO     | 1#            |                                                     |
| 3943 | 023304 | 012604 |        |         | MOV     | (SP)+,R4      |                                                     |
| 3944 | 023306 | 012603 |        |         | MOV     | (SP)+,R3      |                                                     |
| 3945 | 023310 | 012602 |        |         | MOV     | (SP)+,R2      |                                                     |
| 3946 | 023312 |        |        |         | ENDRPT  |               | ; UNUSED.                                           |
|      | 023312 |        |        | L10035: |         |               |                                                     |
|      | 023312 | 104425 |        |         | TRAP    | C#RPT         |                                                     |
| 3947 |        |        |        |         |         |               |                                                     |
| 3948 |        |        |        |         |         |               |                                                     |
| 3949 | 023314 | 045    | 116    | 045     | DEVSUM: | .ASCIZ        | /##ADEVICE STATUS SUMMARY:##N/                      |
| 3950 | 023351 | 045    | 101    | 040     | DEVONL: | .ASCIZ        | /##A UNIT #D3##A ONLINE, ERRORS = #D##N/            |
| 3951 | 023421 | 045    | 101    | 040     | DEVNXR: | .ASCIZ        | /##A UNIT #D3##A DROPPED, NON-EXISTENT REGISTER##N/ |
| 3952 | 023503 | 045    | 101    | 040     | DEVNRD: | .ASCIZ        | /##A UNIT #D3##A DROPPED, NOT READY AT STARTUP##N/  |
| 3953 | 023564 | 045    | 101    | 040     | DEVDR0: | .ASCIZ        | /##A UNIT #D3##A DROPPED, ERRORS = #D##N/           |
| 3954 |        |        |        |         |         | .EVEN         |                                                     |
| 3957 |        |        |        |         |         |               |                                                     |
| 3958 |        |        |        |         |         |               |                                                     |
| 3959 |        |        |        |         |         |               |                                                     |
| 3966 |        |        |        |         |         |               |                                                     |
| 3972 |        |        |        |         |         |               |                                                     |
| 3980 |        |        |        |         |         |               |                                                     |

```

3983          .SBTTL TEST 1: INITIALIZE #2 TEST
3984          ;+
3985          ;
3986          ;THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE
3987          ;CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITIONS
3988          ;
3989          ;-
3990          BGNTST
3991          023634      005037  002170      CLR    FATFLG      ;CLEAR FATAL ERROR FLAG
3992          023640      005037  003100      CLR    KTFLG      ;HOLD OFF KT11
3993          023644      012737  005762  002146  MOV    #EPRT1,EPRTSW ;SET UP PRIMARY ERROR MESSAGE
3994
3995          ;
3996          ;
3997          ;TEST 1
3998          ;
3999          ;-
4000
4001          4006 023652  004737  017054      JSR    PC,DSBINT      ;DISABLE INTERRUPTS
4002          4007 023656  012700  024516      MOV    #TST21ID,R0    ;ASCII MESSAGE TO IDENTIFY TEST
4003          4008 023662  004737  017410      JSR    PC,TSTSETUP    ;DO INITIAL TEST SETUP
4004          4009 023666  012737  000002  002164  MOV    #2,LOOPCNT    ;PERFORM 2 ITERATIONS
4005          4010 023674
4006          4011 023674  004737  024540      JSR    PC,T21REST      ;SET COMMAND PACKET
4007          4012 023700  004737  024630      JSR    PC,T21RT2      ;SET UP OTHER COMMAND PACKET
4008          4013
4009          4014          ;*****
4010          4015          ;
4011          4016          ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
4012          4017          ;
4013          4018          ;*****
4014          4019
4015          4020 023704  012737  176750  024276  MOV    #65000.,T21DLY    ;SET DELAY ROUTINE
4016          4021 023712  004737  016644      11$: JSR    PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
4017          4022 023716  103426      BCS    20$      ;BR IF INIT WAS OK
4018          4023 023720      DELAY    250      ;DELAY FOR A REWIND TO FINISH
4019          023720  012727  000250      MOV    #250,(PC)+
4020          023724  000000      .WORD    0
4021          023726  013727  002116      MOV    L$DLY,(PC)+
4022          023732  000000      .WORD    0
4023          023734  005367  177772      DEC    -6(PC)
4024          023740  001375      BNE    -4
4025          023742  005367  177756      DEC    -22(PC)
4026          023746  001367      BNE    -20
4027          4024 023750  005337  024276      DEC    T21DLY      ;BUMP COUNTER DOWN
4028          4025 023754  001356      BNE    11$      ;BR, IF MORE TIME TO GO
4029          4026 023756  004737  020102      JSR    PC,FATCHK    ;INC AND CHECK FOR MORE THAN 25 ERRORS
4030          4030 023762  010001      MOV    R0,R1      ;CONTENTS OF TSSR REGISTER
4031          4031 023764      ERROF    ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
4032          023764  104455      TRAP    C$ERDF
4033          023766  000145      .WORD    101
4034          023770  003550      .WORD    SFIERR
4035          023772  011666      .WORD    SFIMSG
4036          4032 023774      20$:
4037          4033 023774  012704  024150      MOV    #T21PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS

```

```

4034
4035
4036
4037
4038
4039
4040
4041 024000 004737 010332
4042 024004 103407
4043 024006 004737 020102
4047 024012 010001
4048 024014
    024014 104456
    024016 000146
    024020 004754
    024022 011666
4049 024024
4050 024024 012765 001000 000000 23$:
4051 024032 004737 017120
4052 024036 016501 000000
4053 024042 005002
4054 024044 032701 000100
4055 024050 001402
4056 024052 052702 000100
4057 024056 052702 002200
4058 024062 032701 000100 24$:
4059 024066 001012
4060 024070 020102 35$:
4061 024072 001406
4062 024074 004737 020102
4066 024100
    024100 104456
    024102 000147
    024104 024375
    024106 016344
4067 024110 37$:
    024110 104406
4068 024112 000406
4069 024114
4073 024114 38$:
    024114 104455
    024116 000150
    024120 024475
    024122 016344
4074 024124 004737 020154
4075 024130 004737 017356 40$:
4076 024134 103002
4077 024136 000137 023674
4078 024142 63$:
    024142 104432
    024144 000526

```

\*\*\*\*\*  
 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)  
 ;\*\*\*\*\*  
 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS  
 BCS 23\$ ;BR, IF COMMAND ISSUED OK  
 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS  
 MOV R0,R1 ;SAVE CONTENTS OF TSSR  
 ERRHRD ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED  
 TRAP C\$ERHRD  
 .WORD 102  
 .WORD WRTMSG  
 .WORD SFMSG  
 MOV #0,TSSR(R5) ;ISSUE A SOFT INITIALIZE  
 JSR PC,WAITF ;WAIT FOR JUST THE SSR BIT TO SET  
 MOV TSSR(R5),R1 ;READ THE TSSR BACK  
 CLR R2 ;SET UP EXPECTED CONTENTS IN R2  
 BIT #0FL,R1 ;CHECK FOR OFF LINE SET (NOT ERROR)  
 BEQ 24\$ ;BR, IF 0FL IS NOT SET  
 BIS #0FL,R2 ;IT WAS SET SO SET IN EXPECTED  
 BIS #SSR:NBA,R2 ;R2 HAS EXPECTED CONTENTS  
 BIT #0FL,R1 ;IS OFF LINE BIT SET  
 BNE 38\$ ;BR, IF DRIVE IS OFF LINE  
 CMP R1,R2 ;EXPECTED (R2) = RECEIVED (R1)  
 BEQ 37\$ ;BR, IF THEY ARE EQUAL (OK)  
 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS  
 ERRHRD ERRNO,T21AM3,EXPREC ;"ERROR TRYING TO INIT AFTER WRITE MISC.  
 TRAP C\$ERHRD  
 .WORD 103  
 .WORD T21AM3  
 .WORD EXPREC  
 CKLOOP ;LOOP IF SELECTED  
 TRAP C\$CLP1  
 BR 40\$ ;SKIP OVER OFF-LINE STUFF  
 ERRDF ERRNO,T21OFL,EXPREC ;DRIVE IS OFF LINE  
 TRAP C\$ERDF  
 .WORD 104  
 .WORD T21OFL  
 .WORD EXPREC  
 JSR PC,CKDROP ;TRY AND DROP UNIT  
 JSR PC,TSTLOOP ;DO WE NEED TO ITERATE TEST  
 BCC 63\$ ;BR, IF NO LOOP REQUIRED  
 JMP T21LOOP ;EXECUTE AGAIN  
 EXIT TST ;ALL DONE THIS TEST  
 TRAP C\$EXIT  
 .WORD L10036-.

B10

CZTRFA TK-25 FRT END FUNC #2  
TEST 1: INITIALIZE #2 TEST

MACRO M1200 20-APR-84 08:12 PAGE 87

SEQ 118

4080  
 4081  
 4082  
 4084 024146  
 4086 024150  
 4087 024150 100004  
 4088 024152 024160  
 4089 024154 000000  
 4090 024156 000012  
 4091 024160  
 4092 024160 024170  
 4093 024162 000000  
 4094 024164 000024  
 4095 024166 000000  
 4096 024170  
 4097  
 4098  
 4099  
 4101 024252  
 4103 024260  
 4104 024260 100206  
 4105 024262 024270  
 4106 024264 000000  
 4107 024266 000006  
 4108  
 4109  
 4110 024270  
 4111 024270 000  
 4112 024271 000  
 4113 024272 000000  
 4114 024274 000000  
 4115 024276 000000  
 4116  
 4117  
 4118

```

**
;LOCAL STORAGE FOR THIS TEST
;
;
;BLKB 10-<.-TUV2A&7>
T21PACKET:
;WORD 100004
;WORD T21DATA
;WORD 0
;WORD 10.
T21DATA:
;WORD T21BFR
;WORD 0
;WORD 20.
;WORD 0
T21BFR: .BLKW 25.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
;BLKB 10-<.-TUV2A&7>
T21PK2:
;WORD 100206
;WORD T21BF2
;WORD 0
;WORD 6.
;EVEN
T21BF2:
T21BS0: .BYTE 0
T21BS1: .BYTE 0
T21S2: .WORD 0
T21S3: .WORD 0
T21DLY: .WORD 0

```

```

;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH, ACK
;ADDRESS OF CHARACTERISTICS BLOCK

```

```

;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER

```

```

;LENGTH OF MESSAGE BUFFER

```

```

;MESSAGE BUFFER

```

```

;WRITE SUB SYS MEM COMMAND, IE AND ACK
;ADDRESS OF SELECT BLOCK DATA

```

```

;SIZE OF DATA PACKET

```

```

;BSELO AREA --- "COMMAND" BYTE
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA
;DELAY COUNTER

```

C10

C2TKFA TK 25 FRT END FUNC #2  
TEST 1: INITIALIZE #2 TEST

MACRO M1200 20-APR-84 08:12 PAGE 88

SEQ 119

```

4120
4121
4122          ;*
4123          ;LOCAL TEXT MESSAGES FOR TEST
4124          ;*
4125 024300      127      122      111  T21SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
4126 024375      124      123      123  T21AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
4127 024475      104      162      151  T21OFL: .ASCIZ 'Drive is OFFLINE'
4128 024516      111      156      151  T21ID:  .ASCIZ 'Initialization #2'
4129          .EVEN
4130          ;*
4131          ;
4132          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4133          ;WRITE SUBSYSTEM MEMORY COMMAND
4134          ;
4135          ;*
4136
4137 024540      T21REST:
4138 024540          SAVREG
4139 024544      012701  024150          MOV      #T21PACKET,R1          ;SAVE THE REGISTERS
4140 024550      012721  100004          MOV      #100004,(R1)+      ;START OF THE PACKET
4141 024554      012721  024160          MOV      #T21DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,
4142 024560      005021          CLR      (R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
4143 024562      012721  000010          MOV      #8,(R1)+      ;EXTENDED ADDRESS
4144 024566      012721  024170          MOV      #T21BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
4145 024572      005021          CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
4146 024574      012721  000024          MOV      #20,(R1)+      ;LENGTH OF MESSAGE BUFFER
4147 024600      005021          CLR      (R1)+
4148 024602      005011          CLR      (R1)
4149 024604      012702  000020          MOV      #20,R2          ;NUMBER OF LOCATIONS TO BE CLEARED
4150 024610      012762  177777  024170  64#  MOV      #177777,T21BFR(R2) ;ALL ONES TO MESSAGE BUFFER
4151 024616      005742          TST      -(R2)          ;NEXT LOCATION
4152 024620      020227  000000          CMP      R2,#0          ;CHECK R2 FOR ZERO
4153 024624      001371          BNE      64#          ;BR. IF NOT AT ZERO YET
4154 024626      000207          RTS      PC          ;RETURN
4155
4156
4157 024630      T21RT2:
4158 024630          SAVREG
4159 024634      012701  024260          MOV      #T21PK2,R1          ;SAVE THE REGISTERS
4160 024640      012721  100206          MOV      #100206,(R1)+      ;START OF THE PACKET
4161 024644      012721  024270          MOV      #T21BF2,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4162 024650      005021          CLR      (R1)+      ;ADDRESS OF DATA BLOCK
4163 024652      012721  000006          MOV      #6,(R1)+      ;EXTENDED ADDRESS
4164 024656      005021          CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES
4165 024660      012701  024270          MOV      #T21BF2,R1          ;ADDRESS OF DATA FOR WRT SUB SYS MEM
4166 024664      005021          CLR      (R1)+
4167 024666      005011          CLR      (R1)
4168 024670      000207          RTS      PC          ;RETURN
4169 024672          ENDTST
          024672
          024672  104401

```

L10036: TRAP C#ETST

D10

CZIKFA TK-25 FRT END FUNC 22 MACRO M1200 20-APR-84 08:12 PAGE 89  
TEST 2: OFF-LINE AND REJECT REWIND

SEQ 120

```

4171                                     .SBTTL TEST 2: OFF-LINE AND REJECT REWIND
4172                                     ;*
4173                                     ;
4174                                     ;THIS TEST VERIFIES BASIC TAPE-MOTION COMMAND DECODING AND BASIC
4175                                     ;OPERATION OF THE REWIND POSITIONING COMMAND. IT DOES NOT
4176                                     ;NECESSARILY DEMONSTRATE THAT THE TRANSPORT C.N BE REWOUND FROM AN
4177                                     ;ARBITRARY POSITION ON THE TAPE. SUBSEQUENT TESTS IMPLICITLY
4178                                     ;CHECK THE OPERATION OF THE REWIND COMMAND SINCE THEY MUST
4179                                     ;TYPICALLY REWIND THE TAPE IN THE NORMAL COURSE OF THEIR TEST
4180                                     ;SEQUENCES. THE TEST CONSISTS OF THE FOLLOWING ONE SUBTEST
4181                                     ;
4182                                     ;
4183                                     ;-
4184                                     BGNTST
4185 024674 005037 002170                                     CLR     FATFLG                                     ;CLEAR FATAL ERROR FLAG
4186 024700 005037 003100                                     CLR     KTFLG                                     ;HOLD OFF KT11
4187 024704 012737 005762 002146                               MOV     @EPRT1,EPRTSW                               ;SET UP PRIMARY ERROR MESSAGE
4188 024712 004737 017054                                     JSR     PC,DSBINT                                   ;DISABLE INTERRUPTS
4189 024716 012700 026032                                     MOV     @TST22ID,R0                                ;ASCII MESSAGE TO IDENTIFY TEST
4190 024722 004737 017410                                     JSR     PC,TSTSETUP                                ;DO INITIAL TEST SETUP
4191 024726 012737 000002 002164                               MOV     @2,LOOPCNT                                ;PERFORM 2 ITERATIONS
4192
4193
4194
4195
4196                                     ;*
4197                                     ;

```



| Address | Offset | Hex    | Label  | Operation | Comments            |             |
|---------|--------|--------|--------|-----------|---------------------|-------------|
| 4252    | 025060 | 004737 | 020102 | JSR       | PC,FATCHK           |             |
| 4256    | 025064 |        |        | ERRHRD    | ERRNO,T22RWJ,EXPREC |             |
|         | 025064 | 104456 |        |           |                     |             |
|         | 025066 | 000313 |        |           |                     |             |
|         | 025070 | 025704 |        |           |                     |             |
|         | 025072 | 016344 |        |           |                     |             |
| 4257    | 025074 |        | 80\$:  | CKLOOP    |                     |             |
|         | 025074 | 104406 |        |           |                     |             |
| 4258    | 025076 | 012703 | 025222 | MOV       | #T22BFR,R3          |             |
| 4259    | 025102 | 016301 | 000006 | MOV       | XST0(R3),R1         |             |
| 4260    | 025106 | 010102 |        | MOV       | R1,R2               |             |
| 4261    | 025110 | 042702 | 000020 | BIC       | #BIT4,R2            |             |
| 4262    | 025114 | 020102 |        | CMP       | R1,R2               |             |
| 4263    | 025116 | 001406 |        | BEQ       | 90\$                |             |
| 4264    | 025120 | 004737 | 020102 | JSR       | PC,FATCHK           |             |
| 4268    | 025124 |        |        | ERRHRD    | ERRNO,T22VCK,EXPREC |             |
|         | 025124 | 104456 |        |           |                     |             |
|         | 025126 | 000314 |        |           |                     |             |
|         | 025130 | 025757 |        |           |                     |             |
|         | 025132 | 016344 |        |           |                     |             |
| 4269    | 025134 |        | 90\$:  |           |                     |             |
| 4270    | 025134 |        |        | ENDSUB    |                     |             |
|         | 025134 |        |        |           |                     |             |
|         | 025134 | 104403 |        |           |                     |             |
| 4271    | 025136 | 023727 | 002170 | 000031    | CMP                 | FATFLG,#25. |
| 4272    | 025144 | 002402 |        |           | BLT                 | 999\$       |
| 4273    | 025146 | 004737 | 020154 |           | JSR                 | PC,CKDROP   |
| 4274    | 025152 |        |        | 999\$:    |                     |             |
| 4275    | 025152 | 004737 | 017356 |           | JSR                 | PC,TSTLOOP  |
| 4276    | 025156 | 103002 |        |           | BCC                 | 163\$       |
| 4277    | 025160 | 000137 | 024734 |           | JMP                 | T22LOOP     |
| 4278    | 025164 |        |        | 163\$:    | EXIT                | TST         |
|         | 025164 | 104432 |        |           |                     |             |
|         | 025166 | 001036 |        |           |                     |             |

G10

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 91  
 TEST 2: OFF-LINE AND REJECT REWIND

SEQ 123

|      |        |        |                             |                                       |                                               |
|------|--------|--------|-----------------------------|---------------------------------------|-----------------------------------------------|
| 4280 |        | ;      |                             |                                       |                                               |
| 4281 |        | ;      | LOCAL STORAGE FOR THIS TEST |                                       |                                               |
| 4282 |        | ;      |                             |                                       |                                               |
| 4284 | 025170 |        | .BLKB                       | 10-<.-TUV2A&7>                        |                                               |
| 4286 | 025200 |        | T22PACKET:                  |                                       | ; COMMAND PACKET FOR TEST                     |
| 4287 | 025200 | 100204 | .WORD                       | 100204                                | ; WRITE CHARACTERISTICS COMMAND, WITH IE, ACK |
| 4288 | 025202 | 025210 | .WORD                       | T22DATA                               | ; ADDRESS OF CHARACTERISTICS BLOCK            |
| 4289 | 025204 | 000000 | .WORD                       | 0                                     |                                               |
| 4290 | 025206 | 000012 | .WORD                       | 10.                                   | ; STARTING VALUE OF BLOCK SIZE                |
| 4291 | 025210 |        | T22DATA:                    |                                       | ; CHARACTERISTICS DATA BLOCK                  |
| 4292 | 025210 | 025222 | .WORD                       | T22BFR                                | ; ADDRESS OF MESSAGE BUFFER                   |
| 4293 | 025212 | 000000 | .WORD                       | 0                                     |                                               |
| 4294 | 025214 | 000024 | .WORD                       | 20.                                   | ; LENGTH OF MESSAGE BUFFER                    |
| 4295 | 025216 | 000000 | .WORD                       | 0                                     |                                               |
| 4296 | 025220 | 000007 | .WORD                       | 7                                     | ; SELECT DRIVE 7                              |
| 4297 | 025222 |        | T22BFR: .BLKW               | 25.                                   | ; MESSAGE BUFFER                              |
| 4298 |        |        | ;                           |                                       |                                               |
| 4299 |        |        | ;                           | WRITE SUBSYSTEM MEMORY COMMAND PACKET |                                               |
| 4300 |        |        | ;                           |                                       |                                               |
| 4302 | 025304 |        | .BLKB                       | 10-<.-TUV2A&7>                        |                                               |
| 4304 | 025310 |        | T22PK2:                     |                                       |                                               |
| 4305 | 025310 | 100206 | .WORD                       | 100206                                | ; WRITE SUB SYS MEM COMMAND, IE AND ACK       |
| 4306 | 025312 | 025320 | .WORD                       | T22BF2                                | ; ADDRESS OF SELECT BLOCK DATA                |
| 4307 | 025314 | 000000 | .WORD                       | 0                                     |                                               |
| 4308 | 025316 | 000006 | .WORD                       | 6.                                    | ; SIZE OF DATA PACKET                         |
| 4309 |        |        |                             |                                       |                                               |
| 4310 |        |        | .EVEN                       |                                       |                                               |
| 4311 | 025320 |        | T22BF2:                     |                                       |                                               |
| 4312 | 025320 | 000    | T22BS0: .BYTE               | 0                                     | ; BSEL0 AREA                                  |
| 4313 | 025321 | 000    | T22BS1: .BYTE               | 0                                     | ; BSEL1 AREA                                  |
| 4314 | 025322 | 000000 | T22S2: .WORD                | 0                                     | ; SEL 2 AREA                                  |
| 4315 | 025324 | 000000 | T22S3: .WORD                | 0                                     | ; DATA AREA                                   |
| 4316 |        |        | ;                           |                                       |                                               |
| 4317 |        |        | ;                           |                                       |                                               |
| 4318 |        |        | .EVEN                       |                                       |                                               |
| 4319 |        |        | ;                           | TAPE MOTION PACKET COMMAND VALUES     |                                               |
| 4320 | 025326 | 100201 | T22RD: .WORD                | 100201                                | ; READ TAPE FORWARD                           |
| 4321 | 025330 | 100205 | T22WRT: .WORD               | 100205                                | ; WRITE TAPE FORWARD                          |
| 4322 | 025332 | 100210 | T22POS: .WORD               | 100210                                | ; POSITION TAPE                               |
| 4323 | 025334 | 100211 | T22FOR: .WORD               | 100211                                | ; FORMAT TAPE                                 |
| 4324 | 025336 | 177777 | .WORD                       | 177777                                | ; END OF DATA                                 |
| 4325 |        |        |                             |                                       |                                               |
| 4326 |        |        |                             |                                       |                                               |

H10

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 92  
TEST 2: OFF-LINE AND REJECT REWIND

SEQ 124

```

4328
4329
4330      ;+
4331      ;LOCAL TEXT MESSAGES FOR TEST
4332      ;-
4333 025340      127      122      111  T22SSR: .ASCIZ 'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
4334 025435      124      123      123  T22AM3: .ASCIZ 'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
4335 025535      104      162      151  T22OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
4336 025610      124      123      123  T22TM: .ASCIZ 'TSSR Incorrect After Tape Motion Command To Off-Line Device'
4337 025704      124      123      123  T22RWJ: .ASCIZ 'TSSR Not Correct After REWIND With VCK Set'
4338 025757      103      126      103  T22VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
4339 026032      117      146      146  T22ID: .ASCIZ 'Off-Line And Reject Rewind'
4340      .EVEN
4341
4342      ;+
4343      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4344      ;WRITE SUBSYSTEM MEMORY COMMAND
4345      ;-
4346
4347
4348 026066      T22REST:
4349 026066      SAVREG
4350 026072      012701  025200      MOV      #T22PACKET,R1      ;SAVE THE REGISTERS
4351 026076      012721  100204      MOV      #100204,(R1)+      ;START OF THE PACKET
4352 026102      012721  025210      MOV      #T22DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4353 026106      005021      CLR      (R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
4354 026110      012721  000012      MOV      #10.,(R1)+      ;EXTENDED ADDRESS
4355 026114      012721  025222      MOV      #T22BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
4356 026120      005021      CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
4357 026122      012721  000024      MOV      #20.,(R1)+      ;LENGTH OF MESSAGE BUFFER
4358 026126      005021      CLR      (R1)+
4359 026130      012711  000007      MOV      #7,(R1)      ;SELECT DRIVE SEVEN
4360 026134      012702  000020      MOV      #20,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
4361 026140      012762  177777  025222  64+: MOV      #177777,T22BFR(R2)      ;ALL ONES TO MESSAGE BUFFER
4362 026146      005742      TST      -(R2)      ;BUMP R2 DOWN
4363 026150      020227  000000      CMP      R2,#0      ;IS R2 AT ZERO YET
4364 026154      001371      BNE      64+      ;KEEP GOING UNTIL DONE
4365 026156      000207      RTS      PC      ;RETURN
4366
4367
4368 026160      T22RT2:
4369 026160      SAVREG
4370 026164      012701  025310      MOV      #T22PK2,R1      ;SAVE THE REGISTERS
4371 026170      012721  100206      MOV      #100206,(R1)+      ;START OF THE PACKET
4372 026174      012721  025320      MOV      #T22BF2,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4373 026200      005021      CLR      (R1)+      ;ADDRESS OF DATA BLOCK
4374 026202      012721  000006      MOV      #6.,(R1)+      ;EXTENDED ADDRESS
4375 026206      005021      CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES
4376 026210      012701  025320      MOV      #T22BF2,R1      ;POINT TO DATA SEL AREA
4377 026214      005021      CLR      (R1)+
4378 026216      005011      CLR      (R1)
4379 026220      005011      CLR      (R1)
4380 026222      000207      RTS      PC      ;LAST LOC TO BE CLEARED
4381 026224      ENDTST      ;RETURN
      026224
      026224  104401

```

L10037: TRAP C\$ETST

```

4383          .SBTTL  TEST  3: BASIC WRITE
4384          ;+
4385          ;
4386          ;THIS TEST VERIFIES THAT THE WRITE DATA (NEXT) COMMAND OPERATES
4387          ;PROPERLY, UP TO THE POINT OF CHECKING THAT THE DATA WAS ACTUALLY
4388          ;WRITTEN ONTO THE TAPE CORRECTLY. CHECKING IN THIS TEST IS
4389          ;LIMITED TO VERIFYING THAT THE COMMAND TERMINATED CORRECTLY WITH
4390          ;THE CORRECT REGISTER, MESSAGE BUFFER AND RAM CONTENTS.
4391          ;
4392          ;THE TEST CONSISTS OF THE FOLLOWING 5 SUBTESTS
4393          ;
4394          ;
4395          ;
4396          ;-
4397          026226          BGNTST
4398          026226          T3::
4398 026226 005037 002170    CLR    FATFLG          ;CLEAR FATAL ERROR FLAG
4399 026232 005037 003100    CLR    KTFLG          ;HOLD OFF KT11
4400 026235 012737 005762    MOV    #EPT1,EPTSW    ;SET UP PRIMARY ERROR MESSAGE
4401 026244 005037 003102    CLR    KTENABLE        ;TURN OFF KT11
4402 026250 004737 020246    JSR    PC,KTOFF        ;TURN OFF KT11
4407 026254 004737 017054    JSR    PC,DSBINT       ;DISABLE INTERRUPTS
4408 026260 012700 031761    MOV    #TST23ID,R0    ;ASCII MESSAGE TO IDENTIFY TEST
4409 026264 004737 017410    JSR    PC,TSTSETUP    ;DO INITIAL TEST SETUP
4410 026270 012737 000001    MOV    #1,LOOPCNT      ;PERFORM 1 ITERATIONS
4411          ;+
4412          ;

```



K10

CZTKFA TK-25 FRT END FUNC #2  
TEST 3: BASIC WRITE

MACRO M1200 20-APR-84 08:12 PAGE 94-1

SEQ 127

```

4471 026372 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
4472 026374          ERRHRD  ERRNO,WRTMSG,SFMSG    ;WRITE CHARACTERISTICS FAILED
                                TRAP    C$ERRHRD
                                .WORD   302
                                .WORD   WRTMSG
                                .WORD   SFMSG
4473 026404          23$:    CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
4474
4475          ;*****
4476          ;
4477          ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
4478          ;
4479          ;*****
4480
4481 026406 004737 010434      JSR    PC,REWIND      ;CALL THE TAPE REWIND
4482 026412 012703 000024      MOV    #20.,R3        ;STARTING RECORD SIZE
4483 026416 013737 003072 030512 65$:    MOV    FREE,T23WB    ;STARTING WRITE BUFFER ADDRESS
4484
4485          ;*****
4486          ;
4487          ;WRITE DATA,CVC=1,ACK COMMAND
4488          ;
4489          ;*****
4490
4491 026424 012737 140005 030510      MOV    #140005,T23PK3    ;WRITE DATA,CVC=1,ACK COMMAND
4492 026432 012704 030510      MOV    #T23PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4493 026436 010300              MOV    R3,R0            ;SET PATTERN IN CORRECT REGISTER
4494 026440 004737 020374      JSR    PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
4495 026444 010337 030516      MOV    R3,T23S2        ;SET UP RECORD SIZE IN PACKET
4496 026450 010465 177776      MOV    R4,TSDB(R5)      ;ISSUE COMMAND
4497 026454 004737 017120      JSR    PC,WAITF        ;WAIT FOR SSR TO SET
4498 026460 016501 000000      MOV    TSSR(R5),R1      ;GET TSSR CONTENTS
4499 026464 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED
4500 026470 020102              CMP    R1,R2          ;ARE THEY EQUAL
4501 026472 001406              BEQ    80$            ;BR. IF OK
4502 026474 004737 020102      JSR    PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
4507 026500          ERRHRD  ERRNO,WRTERR,PKTSSR    ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERRHRD
                                .WORD   303
                                .WORD   WRTERR
                                .WORD   PKTSSR
                                .WORD   011700
4508 026510          80$:    CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
4509 026512 016501 177776      MOV    TSBA(R5),R1      ;GET TSBA CONTENTS
4510 026516 012702 030510      MOV    #T23PK3,R2      ;SET UP EXPECTED
4511 026522 020102              CMP    R1,R2          ;ARE THEY EQUAL
4512 026524 001406              BEQ    90$            ;BR. IF TSBA IS CORRECT
4513 026526 004737 020102      JSR    PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
4517 026532          ERRHRD  ERRNO,T23BA,EXPREC    ;TSBA WAS NOT CORRECT AFTER WRITE DATA
                                TRAP    C$ERRHRD
                                .WORD   304
                                .WORD   T23BA
                                .WORD   EXPREC
                                .WORD   016344
4518 026542          90$:    CKLOOP                ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                .WORD   104406
4519 026544 062703 001750      115$:    ADD    #1000.,R3    ;NEXT RECORD SIZE/DATA PATTERN

```

L10

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 94-2  
TEST 3: BASIC WRITE

SEQ 128

|      |        |        |        |        |        |                     |         |                                        |
|------|--------|--------|--------|--------|--------|---------------------|---------|----------------------------------------|
| 4520 | 026550 | 020337 | 030520 |        | CMP    | R3,T23RSZ           |         | ;IS R3 OVER MAX RECORD SIZE            |
| 4521 | 026554 | 002005 |        |        | BGE    | 120\$               |         | ;IF RECORD SIZE IS TOO BIG QUIT        |
| 4522 | 026556 | 020327 | 177776 |        | CMP    | R3,#65534.          |         | ;END OF SUBTEST MAX RECORD SIZE        |
| 4523 | 026562 | 001402 |        |        | BEQ    | 120\$               |         | ;BR, IF COMPLETED                      |
| 4524 | 026564 | 000137 | 026416 |        | JMP    | 65\$                |         | ;DO MORE RECORDS                       |
| 4525 | 026570 |        |        | 120\$: |        |                     |         |                                        |
| 4526 | 026570 | 004737 | 032132 |        | JSR    | PC,T23RT3           |         | ;RESTORE PACKET                        |
| 4527 | 026574 | 004737 | 032070 |        | JSR    | PC,T23RT2           |         | ;CLEAN UP PACKET                       |
| 4528 | 026600 | 004737 | 010434 |        | JSR    | PC,REWIND           |         | ;ISSUE REWIND COMMAND WITH WAIT        |
| 4529 | 026604 | 103407 |        |        | BCS    | 130\$               |         | ;BR, IF TSSR IS OK (GOOD)              |
| 4530 | 026606 | 010001 |        |        | MOV    | R0,R1               |         | ;SAVE TSSR CONTENTS                    |
| 4531 | 026610 | 004737 | 020102 |        | JSR    | PC,FATCHK           |         | ;INC AND CHECK FOR MORE THAN 25 ERRORS |
| 4535 | 026614 |        |        |        | ERRHRD | ERRNO,T23RWN,PKTSSR |         | ;TSSR IS INCORRECT AFTER REWIND        |
|      | 026614 | 104456 |        |        |        |                     | TRAP    | C\$ERHRD                               |
|      | 026616 | 000461 |        |        |        |                     | .WORD   | 305                                    |
|      | 026620 | 031111 |        |        |        |                     | .WORD   | T23RWN                                 |
|      | 026622 | 011700 |        |        |        |                     | .WORD   | PKTSSR                                 |
| 4536 | 026624 |        |        | 130\$: |        |                     |         |                                        |
| 4537 | 026624 |        |        |        | ENDSUB |                     |         | ; >>>>>>>>> END SUBTEST >>>>>>>>>      |
|      | 026624 |        |        |        |        |                     | L10042: |                                        |
|      | 026624 | 104403 |        |        |        |                     | TRAP    | C\$ESUB                                |
| 4538 | 026626 | 023727 | 002170 | 000031 | CMP    | FATFLG,#25.         |         | ;IS ERROR COUNT AT 25                  |
| 4539 | 026634 | 002402 |        |        | BLT    | 999\$               |         | ;BR, IF LESS THAN 25                   |
| 4540 | 026636 | 004737 | 020154 |        | JSR    | PC,CKDROP           |         | ;TRY TO DROP THE UNIT                  |
| 4541 | 026642 |        |        | 999\$: |        |                     |         |                                        |



```

4596      ;WRITE DATA,CVC=1,ACK,SWB COMMAND
4597      ;
4598      ;*****
4599
4600 026742 012737 150005 030510      MOV      #150005,T23PK3      ;WRITE DATA,CVC=1,ACK,SWB COMMAND
4601 026750 012704 030510      MOV      #T23PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4602 026754 010300      MOV      R3,R0      ;SET PATTERN IN CORRECT REGISTER
4603 026756 004737 020374      JSR      PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
4604 026762 010337 030516      MOV      R3,T23SZ      ;SET UP RECORD SIZE IN PACKET
4605 026766 010465 177776      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
4606 026772 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
4607 026776 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
4608 027002 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
4609 027006 020102      CMP      R1,R2      ;ARE THEY EQUAL
4610 027010 001406      BEQ      80$      ;BR, IF OK
4611 027012 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
4615 027016      ERRHRD      ERRNO,WRterr,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERRHRD
                                .WORD      308
                                .WORD      WRterr
                                .WORD      PKTSSR
                                TRAP      C$CLP1
4616 027026      80$:      CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD      309
                                .WORD      T23BA
                                .WORD      EXPREC
4617 027030      104406      MOV      TSBA(R5),R1      ;GET TSBA CONTENTS
4618 027034 012702 030510      MOV      #T23PK3,R2      ;SET UP EXPECTED
4619 027040 020102      85$:      CMP      R1,R2      ;ARE THEY EQUAL
4620 027042 001406      BEQ      90$      ;BR, IF TSBA IS CORRECT
4621 027044 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
4625 027050      ERRHRD      ERRNO,T23BA,EXPREC      ;TSBA WAS NOT CORRECT AFTER WRITE DATA
                                TRAP      C$ERRHRD
                                .WORD      309
                                .WORD      T23BA
                                .WORD      EXPREC
                                TRAP      C$CLP1
4626 027060      90$:      CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD      310
                                .WORD      T23WSS
                                .WORD      PKTSSR
                                TRAP      C$CLP1
4627 027062 020327 007376      CMP      R3,#7376      ;ONLY CHECK RAM UNTIL ITS FULL
4628 027066 002057      BGE      115$      ;IT WRAPS AROUND ETC.
4629 027070 004737 032070      JSR      PC,T23RT2      ;MAKE SURE PACKET AND DATA ARE CLEAN
4630 027074 012737 000400 030524      MOV      #256.,T23S2      ;STARTING RAM ADDRESS
4631 027102 112737 000000 030522      MOV      #0,T23BS0      ;STOP INTERNAL TUV05 DIAGNOSTICS
4632 027110 112737 000000 030523      MOV      #0,T23BS1      ;SIZE OF RAM READ
4633 027116 012704 030470      MOV      #T23PK2,R4      ;SET R4 WITH PACKET ADDRESS
4634 027122 010465 177776      MOV      R4,TSDB(R5)      ;ISSUE WRITE SUB SYS MEM COMMAND
4635 027126 004737 017236      JSR      PC,CHKTSSR      ;CHECK TSSR AND WAIT FOR SSR TO SET
4636 027132 103407      BCS      92$      ;BR, IF NO ERRORS IN TSSR
4637 027134 010001      MOV      R0,R1      ;SAVE TSSR
4638 027136 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
4642 027142      ERRHRD      ERRNO,T23WSS,PKTSSR      ;TSSR BAD AFTER WRITE SUB SYS MEM
                                TRAP      C$ERRHRD
                                .WORD      310
                                .WORD      T23WSS
                                .WORD      PKTSSR
                                TRAP      C$CLP1
4643 027152      92$:      CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD      310
                                .WORD      T23WSS
                                .WORD      PKTSSR
                                TRAP      C$CLP1
4644 027154 004737 032070      JSR      PC,T23RT2      ;MAKE SURE PACKET AND DATA ARE CLEAN
4645 027160 012737 000400 030524      MOV      #256.,T23S2      ;STARTING RAM ADDRESS
4646 027166 112737 000001 030522      MOV      #1,T23BS0      ;READ RAM COMMAND FOR WRITE SUB SYS M.

```

|      |        |        |        |        |        |                     |                                         |       |         |
|------|--------|--------|--------|--------|--------|---------------------|-----------------------------------------|-------|---------|
| 4647 | 027174 | 112737 | 000002 |        | MOV    | #2,T23B51           | ;SIZE OF RAM READ                       |       |         |
| 4648 | 027202 | 012704 | 030470 |        | MOV    | #T23PK2,R4          | ;SET R4 WITH PACKET ADDRESS             |       |         |
| 4649 | 027206 | 010465 | 177776 | 95#:   | MOV    | R4,TSD8(R5)         | ;ISSUE WRITE SUB SYS MEM CMD (READ RAM) |       |         |
| 4650 | 027212 | 004737 | 017236 |        | JSR    | PC,CHKTSSR          | ;CHECK TSSR AND WAIT FOR SSR TO SET     |       |         |
| 4651 | 027216 | 103403 |        |        | BCS    | 115#                | ;BR, IF NO ERRORS IN TSSR               |       |         |
| 4652 | 027220 | 010001 |        |        | MOV    | R0,R1               | ;SAVE TSSR                              |       |         |
| 4653 | 027222 | 004737 | 020102 |        | JSR    | PC,FATCHK           | ;INC AND CHECK FOR MORE THAN 25 ERRORS  |       |         |
| 4657 | 027226 | 062703 | 001750 | 115#:  | ADD    | #1000,,R3           | ;NEXT RECORD SIZE/DATA PATTERN          |       |         |
| 4658 | 027232 | 020337 | 030520 |        | CMP    | R3,T23RSZ           | ;IS R3 OVER MAX RECORD SIZE             |       |         |
| 4659 | 027236 | 002005 |        |        | BGE    | 120#                | ;IF RECORD SIZE IS TOO BIG QUIT         |       |         |
| 4660 | 027240 | 020327 | 177776 |        | CMP    | R3,#65534.          | ;END OF SUBTEST MAX RECORD SIZE         |       |         |
| 4661 | 027244 | 001402 |        |        | BEQ    | 120#                | ;BR, IF COMPLETED                       |       |         |
| 4662 | 027246 | 000137 | 026734 |        | JMP    | 65#                 | ;DO MORE RECORDS                        |       |         |
| 4663 | 027252 |        |        | 120#:  |        |                     |                                         |       |         |
| 4664 | 027252 | 004737 | 032070 |        | JSR    | PC,T23RT2           | ;CLEAN UP PACKET                        |       |         |
| 4665 | 027256 | 004737 | 010434 |        | JSR    | PC,REWIND           | ;ISSUE REWIND COMMAND WITH WAIT         |       |         |
| 4666 | 027262 | 103407 |        |        | BCS    | 130#                | ;BR, IF TSSR IS OK (GOOD)               |       |         |
| 4667 | 027264 | 010001 |        |        | MOV    | R0,R1               | ;SAVE TSSR CONTENTS                     |       |         |
| 4668 | 027266 | 004737 | 020102 |        | JSR    | PC,FATCHK           | ;INC AND CHECK FOR MORE THAN 25 ERRORS  |       |         |
| 4672 | 027272 |        |        |        | ERRHRD | ERRNO,T23RWN,PKTSSR | ;TSSR IS INCORRECT AFTER REWIND         |       |         |
|      | 027272 | 104456 |        |        |        |                     |                                         | TRAP  | C#ERHRD |
|      | 027274 | 000470 |        |        |        |                     |                                         | .WORD | 312     |
|      | 027276 | 031111 |        |        |        |                     |                                         | .WORD | T23RWN  |
|      | 027300 | 011700 |        |        |        |                     |                                         | .WORD | PKTSSR  |
| 4673 | 027302 |        |        | 130#:  |        |                     |                                         |       |         |
| 4674 | 027302 |        |        |        | ENDSUB |                     | ;***** END SUBTEST *****                |       |         |
|      | 027302 |        |        |        |        |                     | L10043;                                 |       |         |
| 4675 | 027304 | 104403 |        |        |        |                     |                                         | TRAP  | C#ESUB  |
| 4676 | 027304 | 023727 | 002170 | 000031 | CMP    | FATFLG,#25,         | ;IS ERROR COUNT AT 25                   |       |         |
| 4676 | 027312 | 002402 |        |        | BLT    | 999#                | ;BR, IF LESS THAN 25                    |       |         |
| 4677 | 027314 | 004737 | 020154 |        | JSR    | PC,CKDROP           | ;TRY TO DROP THE UNIT                   |       |         |
| 4678 | 027320 |        |        | 999#:  |        |                     |                                         |       |         |



|      |        |        |        |        |       |        |                      |                                        |          |
|------|--------|--------|--------|--------|-------|--------|----------------------|----------------------------------------|----------|
| 4733 |        |        |        |        |       |        |                      |                                        |          |
| 4734 | 027412 | 012737 | 104405 | 030510 | 23#:  | MOV    | #104405,T23PK3       | ;WRITE DATA, ACK, ILLEGAL BITS         |          |
| 4735 | 027420 | 013737 | 003072 | 030512 |       | MOV    | FREE,T23WB           | ;SET UP WRITE BUFFER ADDRESS           |          |
| 4736 | 027426 | 062737 | 000001 | 030512 |       | ADD    | #1,T23WB             | ;MAKE ADDRESS ODD (ILLEGAL)            |          |
| 4737 | 027434 | 012737 | 000400 | 030516 |       | MOV    | #256,,T23SZ          | ;SET UP BUFFER SIZE                    |          |
| 4738 | 027442 | 012704 | 030510 |        |       | MOV    | #T23PK3,R4           | ;R4 = POINTER TO PACKET                |          |
| 4739 | 027446 | 010465 | 177776 |        |       | MOV    | R4,TSD8(R5)          | ;ISSUE COMMAND                         |          |
| 4740 | 027452 | 004737 | 017120 |        |       | JSR    | PC,WAITF             | ;WAIT FOR SSR TO SET                   |          |
| 4741 | 027456 | 016501 | 000000 |        |       | MOV    | TSSR(R5),R1          | ;GET TSSR CONTENTS                     |          |
| 4742 | 027462 | 012702 | 100206 |        |       | MOV    | #SSR!SC!BIT1!BIT2,R2 | ;SET UP EXPECTED                       |          |
| 4743 | 027466 | 020102 |        |        |       | CMP    | R1,R2                | ;ARE THEY EQUAL                        |          |
| 4744 | 027470 | 001406 |        |        |       | BEG    | 80#                  | ;BR, IF OK ESP. FUNCTION REJECT        |          |
| 4745 | 027472 | 004737 | 020102 |        |       | JSR    | PC,FATCHK            | ;INC AND CHECK FOR MORE THAN 25 ERRORS |          |
| 4749 | 027476 |        |        |        |       | ERRHRD | ERRNO,T23TM,PKTSSR   | ;TSSR INCORRECT AFTER WRITE COMMAND    |          |
|      | 027476 | 104456 |        |        |       |        |                      | TRAP                                   | C\$ERHRD |
|      | 027500 | 000473 |        |        |       |        |                      | .WORD                                  | 315      |
|      | 027502 | 030736 |        |        |       |        |                      | .WORD                                  | T23TM    |
|      | 027504 | 011700 |        |        |       |        |                      | .WORD                                  | PKTSSR   |
| 4750 | 027506 |        |        |        | 80#:  | CKLOOP |                      | ;LOOP IF SELECTED                      |          |
|      | 027506 | 104406 |        |        |       |        |                      | TRAP                                   | C\$CLP1  |
| 4751 | 027510 |        |        |        |       | ENDSUB |                      | ; >>>>>>>>> END SUBTEST >>>>>>>>       |          |
|      | 027510 |        |        |        |       |        |                      | L10044;                                |          |
|      | 027510 | 104403 |        |        |       |        |                      | TRAP                                   | C\$ESUB  |
| 4752 | 027512 | 023727 | 002170 | 000031 |       | CMP    | FATFLG,#25,          | ;IS ERROR COUNT AT 25                  |          |
| 4753 | 027520 | 002402 |        |        |       | BLT    | 999#                 | ;BR, IF LESS THAN 25                   |          |
| 4754 | 027522 | 004737 | 020154 |        |       | JSR    | PC,CKDROP            | ;TRY TO DROP THE UNIT                  |          |
| 4755 | 027526 |        |        |        | 999#: |        |                      |                                        |          |

```

4757      ;
4758      ;
4759      ;TEST 3, SUBTEST 4
4760      ;
4761      ;VERIFIES THAT A WRITE DATA COMMAND SPECIFYING A DATA
4762      ;BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH
4763      ;THE PROPER ERROR STATUS WITHOUT MOVING TAPE
4764      ;
4765      ;
4766      ;
4767      ;
4768      ;-
4769      027526      BGNSUB                                ;>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>
                                     T3.4:
4770      027526      104402      003100                  TRAP      C$BSUB
4771      027530      005737      BEQ      KTFLG           ;CHECK FOR KT-11
4772      ;          10$              ;BR, IF NO KT-11
4773      ;
4774      ;          DIALOG PATCH FOR NXM
4775      ;
4776      027534      000240      NOP                      ;REMOVE WHEN NXM IS FIXED
4777      027536      000137      JMP      90$             ;SKIP THIS TEST IF KT-11
4778      027542      004737      JSR      PC,T23REST      ;SET COMMAND PACKET
4779      027546      004737      JSR      PC,T23RT3       ;RESTORE PACKET
4780      027552      004737      JSR      PC,T23RT2       ;SET UP OTHER COMMAND PACKET
4801      ;*****
4802      ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
4803      ;*****
4804      027556      004737      JSR      PC,SOFINIT        ;DO INITIALIZE ON CONTROLLER
4805      027562      103407      BCS      20$             ;BR IF INIT WAS OK
4806      027564      004737      JSR      PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
4807      027570      010001      MOV      R0,R1            ;CONTENTS OF TSSR REGISTER
4808      027572      004737      ERDFF      ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
4809      027574      104455      TRAP      C$ERDF         .WORD    316
4810      027576      003550      .WORD    SFIERR         .WORD    SFIMSG
4811      027600      011666
4812      027602      012704      20$:      MOV      @T23PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4813      ;*****
4814      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
4815      ;*****
4816      027606      004737      JSR      PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
4817      027612      103407      BCS      23$             ;BR, IF COMMAND ISSUED OK
4818      027614      004737      JSR      PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
4819      027620      010001      MOV      R0,R1            ;SAVE CONTENTS OF TSSR
4820      027622      104456      WRHRRD      ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
4821      027624      000475      TRAP      C$WRHRRD       .WORD    317

```

SEQ 135

| PC | PC+1 | PC+2 | PC+3 | PC+4 | PC+5 | PC+6 | PC+7 | PC+8 | PC+9 | PC+10 | PC+11 | PC+12 | PC+13 | PC+14 | PC+15 | PC+16 | PC+17 | PC+18 | PC+19 | PC+20 | PC+21 | PC+22 | PC+23 | PC+24 | PC+25 | PC+26 | PC+27 | PC+28 | PC+29 | PC+30 | PC+31 | PC+32 | PC+33 | PC+34 | PC+35 | PC+36 | PC+37 | PC+38 | PC+39 | PC+40 | PC+41 | PC+42 | PC+43 | PC+44 | PC+45 | PC+46 | PC+47 | PC+48 | PC+49 | PC+50 | PC+51 | PC+52 | PC+53 | PC+54 | PC+55 | PC+56 | PC+57 | PC+58 | PC+59 | PC+60 | PC+61 | PC+62 | PC+63 | PC+64 | PC+65 | PC+66 | PC+67 | PC+68 | PC+69 | PC+70 | PC+71 | PC+72 | PC+73 | PC+74 | PC+75 | PC+76 | PC+77 | PC+78 | PC+79 | PC+80 | PC+81 | PC+82 | PC+83 | PC+84 | PC+85 | PC+86 | PC+87 | PC+88 | PC+89 | PC+90 | PC+91 | PC+92 | PC+93 | PC+94 | PC+95 | PC+96 | PC+97 | PC+98 | PC+99 | PC+100 | PC+101 | PC+102 | PC+103 | PC+104 | PC+105 | PC+106 | PC+107 | PC+108 | PC+109 | PC+110 | PC+111 | PC+112 | PC+113 | PC+114 | PC+115 | PC+116 | PC+117 | PC+118 | PC+119 | PC+120 | PC+121 | PC+122 | PC+123 | PC+124 | PC+125 | PC+126 | PC+127 | PC+128 | PC+129 | PC+130 | PC+131 | PC+132 | PC+133 | PC+134 | PC+135 | PC+136 | PC+137 | PC+138 | PC+139 | PC+140 | PC+141 | PC+142 | PC+143 | PC+144 | PC+145 | PC+146 | PC+147 | PC+148 | PC+149 | PC+150 | PC+151 | PC+152 | PC+153 | PC+154 | PC+155 | PC+156 | PC+157 | PC+158 | PC+159 | PC+160 | PC+161 | PC+162 | PC+163 | PC+164 | PC+165 | PC+166 | PC+167 | PC+168 | PC+169 | PC+170 | PC+171 | PC+172 | PC+173 | PC+174 | PC+175 | PC+176 | PC+177 | PC+178 | PC+179 | PC+180 | PC+181 | PC+182 | PC+183 | PC+184 | PC+185 | PC+186 | PC+187 | PC+188 | PC+189 | PC+190 | PC+191 | PC+192 | PC+193 | PC+194 | PC+195 | PC+196 | PC+197 | PC+198 | PC+199 | PC+200 | PC+201 | PC+202 | PC+203 | PC+204 | PC+205 | PC+206 | PC+207 | PC+208 | PC+209 | PC+210 | PC+211 | PC+212 | PC+213 | PC+214 | PC+215 | PC+216 | PC+217 | PC+218 | PC+219 | PC+220 | PC+221 | PC+222 | PC+223 | PC+224 | PC+225 | PC+226 | PC+227 | PC+228 | PC+229 | PC+230 | PC+231 | PC+232 | PC+233 | PC+234 | PC+235 | PC+236 | PC+237 | PC+238 | PC+239 | PC+240 | PC+241 | PC+242 | PC+243 | PC+244 | PC+245 | PC+246 | PC+247 | PC+248 | PC+249 | PC+250 | PC+251 | PC+252 | PC+253 | PC+254 | PC+255 | PC+256 | PC+257 | PC+258 | PC+259 | PC+260 | PC+261 | PC+262 | PC+263 | PC+264 | PC+265 | PC+266 | PC+267 | PC+268 | PC+269 | PC+270 | PC+271 | PC+272 | PC+273 | PC+274 | PC+275 | PC+276 | PC+277 | PC+278 | PC+279 | PC+280 | PC+281 | PC+282 | PC+283 | PC+284 | PC+285 | PC+286 | PC+287 | PC+288 | PC+289 | PC+290 | PC+291 | PC+292 | PC+293 | PC+294 | PC+295 | PC+296 | PC+297 | PC+298 | PC+299 | PC+300 | PC+301 | PC+302 | PC+303 | PC+304 | PC+305 | PC+306 | PC+307 | PC+308 | PC+309 | PC+310 | PC+311 | PC+312 | PC+313 | PC+314 | PC+315 | PC+316 | PC+317 | PC+318 | PC+319 | PC+320 | PC+321 | PC+322 | PC+323 | PC+324 | PC+325 | PC+326 | PC+327 | PC+328 | PC+329 | PC+330 | PC+331 | PC+332 | PC+333 | PC+334 | PC+335 | PC+336 | PC+337 | PC+338 | PC+339 | PC+340 | PC+341 | PC+342 | PC+343 | PC+344 | PC+345 | PC+346 | PC+347 | PC+348 | PC+349 | PC+350 | PC+351 | PC+352 | PC+353 | PC+354 | PC+355 | PC+356 | PC+357 | PC+358 | PC+359 | PC+360 | PC+361 | PC+362 | PC+363 | PC+364 | PC+365 | PC+366 | PC+367 | PC+368 | PC+369 | PC+370 | PC+371 | PC+372 | PC+373 | PC+374 | PC+375 | PC+376 | PC+377 | PC+378 | PC+379 | PC+380 | PC+3 |
|----|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------|
|----|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------|



H11

CZIKFA TK-25 FRT END FUNC #2  
TEST 3: BASIC WRITE

MACRO M1200 20-APR-84 08:12 PAGE 98-1

SEQ 137

```

4910 030110          ERRHRD  ERRNO,WRTMSG,SFMSG      ;WRITE CHARACTERISTICS FAILED
      030110 104456          TRAP  C$ERRHRD
      030112 000500          .WORD 320
      030114 004754          .WORD WRTMSG
      030116 011666          .WORD SFMSG

4911
4912
4913
4914
4915
4916
4917
4918 030120 012737 000000 030514 23$:  MOV  #0,T23WB+2      ;HIGH ORDER ADDRESS BITS ETC.
4919 030126 012737 140005 030510 24$:  MOV  #140005,T23PK3    ;WRITE DATA, ACK,CVC=1
4920 030134 013701 003076          MOV  FREEHI,R1          ;HIGHEST ADDRESS
4921 030140 162701 000100          SUB   #100,R1            ;SET ADDRESS A LITTLE LOWER
4922 030144 010137 030512          MOV  R1,T23WB            ;LOAD INTO THE PACKET
4923 030150 012737 175000 030516      MOV  #64000.,T23SZ      ;SET UP BUFFER SIZE (64K BYTES)
4924 030156 012704 030510          MOV  #T23PK3,R4          ;R4 = POINTER TO PACKET
4925 030162 010465 177776          MOV  R4,TSD8(R5)         ;ISSUE COMMAND
4926 030166 004737 017120          JSR   PC,WAITF           ;WAIT FOR SSR TO SET
4927 030172 016501 000000          MOV  TSSR(R5),R1         ;GET TSSR CONTENTS
4928 030176 012702 104210          MOV  #SCINXM!SSR!BIT3,R2  ;SET UP EXPECTED
4929 030202 020102          CMP  R1,R2                      ;ARE THEY EQUAL
4930 030204 001415          BEQ  80$                        ;BR, IF OK ESP. FUNCTION REJECT
4931 030206 062737 000001 030514      ADD  #1,T23WB+2        ;START CUTTING THE HIGH ADDRESS BITS DOWN
4932 030214 022737 000100 030514      CMP  #100,T23WB+2
4933 030222 001341          BNE  24$                        ;IF NOT AT ZERO, KEEP TRYING
4934 030224 004737 020102          JSR   PC,FATCHK          ;INC AND CHECK FOR MORE THAN 25 ERRORS
4938 030230          ERRHRD  ERRNO,T23TMA,PKTSSR          ;TSSR INCORRECT AFTER WRITE COMMAND
      030230 104456          TRAP  C$ERRHRD
      030232 000501          .WORD 321
      030234 031025          .WORD T23TMA
      030236 011700          .WORD PKTSSR

4939 030240          80$:  CKLOOP                          ;LOOP IF SELECTED
      030240 104406          TRAP  C$CLP1
4940 030242 004737 032070          JSR   PC,T23RT2          ;CLEAN UP PACKET
4941 030246 004737 032132          JSR   PC,T23RT3          ;RESTORE PACKET
4942
4943
4944
4945
4946
4947
4948 030252 004737 010434          JSR   PC,REWIND          ;CALL THE TAPE REWIND
4949
4950
4951
4952 030256 012737 102010 030470      MOV  #102010,T23PK2    ;REWIND (POSITION) COMMAND
4953 030264 012704 030470          MOV  #T23PK2,R4          ;LOAD R4 WITH PACKET ADDRESS
4954 030270 010465 177776          MOV  R4,TSD8(R5)         ;ISSUE REWIND COMMAND
4955 030274 004737 017236          JSR   PC,CHKTSSR         ;WAIT FOR SSR TO SET
4956 030300 103407          BCS  85$                        ;BR, IF TSSR IS OK (GOOD)
4957 030302 010001          MOV  R0,R1                      ;SAVE TSSR CONTENTS
4958 030304 004737 020102          JSR   PC,FATCHK          ;INC AND CHECK FOR MORE THAN 25 ERRORS
4962 030310          ERRHRD  ERRNO,T23RWN,PKTSSR          ;TSSR IS INCORRECT AFTER REWIND
      030310 104456          TRAP  C$ERRHRD

```

| Address | Op Code | Op 1 | Op 2 | Op 3 | Op 4 | Op 5 | Op 6 | Op 7 | Op 8 | Op 9 | Op 10 | Op 11 | Op 12 | Op 13 | Op 14 | Op 15 | Op 16 | Op 17 | Op 18 | Op 19 | Op 20 | Op 21 | Op 22 | Op 23 | Op 24 | Op 25 | Op 26 | Op 27 | Op 28 | Op 29 | Op 30 | Op 31 | Op 32 | Op 33 | Op 34 | Op 35 | Op 36 | Op 37 | Op 38 | Op 39 | Op 40 | Op 41 | Op 42 | Op 43 | Op 44 | Op 45 | Op 46 | Op 47 | Op 48 | Op 49 | Op 50 | Op 51 | Op 52 | Op 53 | Op 54 | Op 55 | Op 56 | Op 57 | Op 58 | Op 59 | Op 60 | Op 61 | Op 62 | Op 63 | Op 64 | Op 65 | Op 66 | Op 67 | Op 68 | Op 69 | Op 70 | Op 71 | Op 72 | Op 73 | Op 74 | Op 75 | Op 76 | Op 77 | Op 78 | Op 79 | Op 80 | Op 81 | Op 82 | Op 83 | Op 84 | Op 85 | Op 86 | Op 87 | Op 88 | Op 89 | Op 90 | Op 91 | Op 92 | Op 93 | Op 94 | Op 95 | Op 96 | Op 97 | Op 98 | Op 99 | Op 100 | Op 101 | Op 102 | Op 103 | Op 104 | Op 105 | Op 106 | Op 107 | Op 108 | Op 109 | Op 110 | Op 111 | Op 112 | Op 113 | Op 114 | Op 115 | Op 116 | Op 117 | Op 118 | Op 119 | Op 120 | Op 121 | Op 122 | Op 123 | Op 124 | Op 125 | Op 126 | Op 127 | Op 128 | Op 129 | Op 130 | Op 131 | Op 132 | Op 133 | Op 134 | Op 135 | Op 136 | Op 137 | Op 138 | Op 139 | Op 140 | Op 141 | Op 142 | Op 143 | Op 144 | Op 145 | Op 146 | Op 147 | Op 148 | Op 149 | Op 150 | Op 151 | Op 152 | Op 153 | Op 154 | Op 155 | Op 156 | Op 157 | Op 158 | Op 159 | Op 160 | Op 161 | Op 162 | Op 163 | Op 164 | Op 165 | Op 166 | Op 167 | Op 168 | Op 169 | Op 170 | Op 171 | Op 172 | Op 173 | Op 174 | Op 175 | Op 176 | Op 177 | Op 178 | Op 179 | Op 180 | Op 181 | Op 182 | Op 183 | Op 184 | Op 185 | Op 186 | Op 187 | Op 188 | Op 189 | Op 190 | Op 191 | Op 192 | Op 193 | Op 194 | Op 195 | Op 196 | Op 197 | Op 198 | Op 199 | Op 200 | Op 201 | Op 202 | Op 203 | Op 204 | Op 205 | Op 206 | Op 207 | Op 208 | Op 209 | Op 210 | Op 211 | Op 212 | Op 213 | Op 214 | Op 215 | Op 216 | Op 217 | Op 218 | Op 219 | Op 220 | Op 221 | Op 222 | Op 223 | Op 224 | Op 225 | Op 226 | Op 227 | Op 228 | Op 229 | Op 230 | Op 231 | Op 232 | Op 233 | Op 234 | Op 235 | Op 236 | Op 237 | Op 238 | Op 239 | Op 240 | Op 241 | Op 242 | Op 243 | Op 244 | Op 245 | Op 246 | Op 247 | Op 248 | Op 249 | Op 250 | Op 251 | Op 252 | Op 253 | Op 254 | Op 255 | Op 256 | Op 257 | Op 258 | Op 259 | Op 260 | Op 261 | Op 262 | Op 263 | Op 264 | Op 265 | Op 266 | Op 267 | Op 268 | Op 269 | Op 270 | Op 271 | Op 272 | Op 273 | Op 274 | Op 275 | Op 276 | Op 277 | Op 278 | Op 279 | Op 280 | Op 281 | Op 282 | Op 283 | Op 284 | Op 285 | Op 286 | Op 287 | Op 288 | Op 289 | Op 290 | Op 291 | Op 292 | Op 293 | Op 294 | Op 295 | Op 296 | Op 297 | Op 298 | Op 299 | Op 300 | Op 301 | Op 302 | Op 303 | Op 304 | Op 305 | Op 306 | Op 307 | Op 308 | Op 309 | Op 310 | Op 311 | Op 312 | Op 313 | Op 314 | Op 315 | Op 316 | Op 317 | Op 318 | Op 319 | Op 320 | Op 321 | Op 322 | Op 323 | Op 324 | Op 325 | Op 326 | Op 327 | Op 328 | Op 329 | Op 330 | Op 331 | Op 332 | Op 333 | Op 334 | Op 335 | Op 336 | Op 337 | Op 338 | Op 339 | Op 340 | Op 341 | Op 342 | Op 343 | Op 344 | Op 345 | Op 346 | Op 347 | Op 348 | Op 349 | Op 350 | Op 351 | Op 352 | Op 353 | Op 354 | Op 355 | Op 356 | Op 357 | Op 358 | Op 359 | Op 360 | Op 361 | Op 362 | Op 363 | Op 364 | Op 365 | Op 366 | Op 367 | Op 368 | Op 369 | Op 370 | Op 371 | Op 372 | Op 373 | Op 374 | Op 375 | Op 376 | Op 377 | Op 378 | Op 379 | Op 380</ |
|---------|---------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|
|---------|---------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|

|      |        |        |                             |                                       |
|------|--------|--------|-----------------------------|---------------------------------------|
| 4977 |        | ;      |                             |                                       |
| 4978 |        | ;      | LOCAL STORAGE FOR THIS TEST |                                       |
| 4979 |        | ;      |                             |                                       |
| 4981 | 030354 |        | .BLKB                       | 10-<.-TUV2A&7>                        |
| 4983 | 030360 |        | T23PACKET:                  |                                       |
| 4984 | 030360 | 100004 | .WORD                       | 100004                                |
| 4985 | 030362 | 030370 | .WORD                       | T23DATA                               |
| 4986 | 030364 | 000000 | .WORD                       | 0                                     |
| 4987 | 030366 | 000010 | .WORD                       | 8.                                    |
| 4988 | 030370 |        | T23DATA:                    |                                       |
| 4989 | 030370 | 030400 | .WORD                       | T23BFR                                |
| 4990 | 030372 | 000000 | .WORD                       | 0                                     |
| 4991 | 030374 | 000012 | .WORD                       | 10.                                   |
| 4992 | 030376 | 000000 | .WORD                       | 0                                     |
| 4993 | 030400 |        | T23BFR: .BLKW               | 25.                                   |
| 4994 |        |        | ;                           |                                       |
| 4995 |        |        | ;                           | WRITE SUBSYSTEM MEMORY COMMAND PACKET |
| 4996 |        |        | ;                           |                                       |
| 4998 | 030462 |        | .BLKB                       | 10-<.-TUV2A&7>                        |
| 5000 | 030470 |        | T23PK2:                     |                                       |
| 5001 | 030470 | 100006 | .WORD                       | 100006                                |
| 5002 | 030472 | 030522 | .WORD                       | T23BF2                                |
| 5003 | 030474 | 000000 | .WORD                       | 0                                     |
| 5004 | 030476 | 000006 | .WORD                       | 6.                                    |
| 5005 |        |        |                             |                                       |
| 5007 | 030500 |        | .BLKB                       | 10-<.-TUV2A&7>                        |
| 5009 | 030510 |        | T23PK3:                     |                                       |
| 5010 | 030510 | 100005 | .WORD                       | 100005                                |
| 5011 | 030512 | 000000 | T23WB: .WORD                | 0                                     |
| 5012 | 030514 | 000000 | .WORD                       | 0                                     |
| 5013 | 030516 | 000000 | T23SZ: .WORD                | 0                                     |
| 5014 |        |        | .EVEN                       |                                       |
| 5015 |        |        | ;                           |                                       |
| 5016 | 030520 | 000000 | T23RSZ: .WORD               | 0                                     |
| 5017 |        |        | ;                           |                                       |
| 5018 |        |        | ;                           |                                       |
| 5019 | 030522 |        | T23BF2:                     |                                       |
| 5020 | 030522 | 010    | T23BS0: .BYTE               | 10                                    |
| 5021 | 030523 | 200    | T23BS1: .BYTE               | 200                                   |
| 5022 | 030524 | 000000 | T23S2: .WORD                | 0                                     |
| 5023 | 030526 | 000000 | T23S3: .WORD                | 0                                     |
| 5024 |        |        | ;                           |                                       |
| 5025 |        |        | ;                           |                                       |
| 5026 |        |        | .EVEN                       |                                       |
| 5027 |        |        | ;                           |                                       |
| 5028 |        |        | ;                           | TAPE MOTION PACKET COMMAND VALUES     |
| 5029 | 030530 | 100005 | T23WD: .WORD                | 100005                                |
| 5030 | 030532 | 100405 | T23WDR: .WORD               | 100405                                |
| 5031 | 030534 | 102005 | T23CON: .WORD               | 102005                                |
| 5032 | 030536 | 177777 | .WORD                       | 177777                                |
| 5033 |        |        |                             |                                       |
| 5034 |        |        |                             |                                       |

  

|   |                                         |
|---|-----------------------------------------|
| ; | COMMAND PACKET FOR TEST                 |
| ; | WRITE CHARACTERISTICS COMMAND, WITH ACK |
| ; | ADDRESS OF CHARACTERISTICS BLOCK        |
| ; | STARTING VALUE OF BLOCK SIZE            |
| ; | CHARACTERISTICS DATA BLOCK              |
| ; | ADDRESS OF MESSAGE BUFFER               |
| ; | LENGTH OF MESSAGE BUFFER                |
| ; | MESSAGE BUFFER                          |
| ; |                                         |
| ; | WRITE SUB SYS MEM COMMAND, AND ACK      |
| ; | ADDRESS OF SELECT BLOCK DATA            |
| ; | SIZE OF DATA PACKET                     |
| ; |                                         |
| ; | WRITE COMMAND, AND ACK                  |
| ; | ADDRESS OF WRITE BUFFER                 |
| ; | SIZE OF BUFFER (EXTENT)                 |
| ; |                                         |
| ; | LARGEST TAPE RECORD IN BYTES            |
| ; |                                         |
| ; | BSEL0 AREA                              |
| ; | BSEL1 AREA                              |
| ; | SEL 2 AREA                              |
| ; | DATA AREA                               |
| ; |                                         |
| ; | WRITE DATA (NEXT)                       |
| ; | WRITE DATA RETRY                        |
| ; | WRITE CONTINUOUS                        |
| ; | END OF DATA                             |

```

5036
5037
5038      ;+
5039      ;LOCAL TEXT MESSAGES FOR TEST
5040      ;-
5041 030540      127      122      111 T23SSR: .ASCIIZ 'WRITE Command Not Accepted'
5042 030573      105      117      124 T23ET:  .ASCIIZ 'EOT Not Found In 12000 4k Writes. (Use Shorter Tape)'
5043 030660      127      122      111 T23EOT: .ASCIIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5044 030736      124      123      123 T23TM:  .ASCIIZ 'TSSR Not Correct After WRITE Command Reject Due To NXM'
5045 031025      124      123      123 T23TMA: .ASCIIZ 'TSSR Not Correct After WRITE To Non-Existent Memory'
5046 031111      122      145      167 T23RWN: .ASCIIZ 'Rewind (POSITION) Command Not Accepted'
5047 031160      122      101      115 T23RNC: .ASCIIZ 'RAM Error. Correct Data Pattern Not In Ram'
5048 031233      124      123      123 T23AM3: .ASCIIZ 'TSSR Init. Failed After WRITE Command'
5049 031301      104      162      151 T23OFL: .ASCIIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
5050 031354      124      123      123 T23WDD: .ASCIIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
5051 031443      124      123      123 T23WDC: .ASCIIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
5052 031545      103      126      103 T23VCK: .ASCIIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
5053 031620      124      123      102 T238A:  .ASCIIZ 'TSBA Not Correct After WRITE DATA Command'
5054 031672      127      122      111 T23WSS: .ASCIIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5055 031761      102      141      163 T23ID:  .ASCIIZ 'Basic Write'
5056
5057      .EVEN
5058
5059      ;+
5060      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5061      ;WRITE SUBSYSTEM MEMORY COMMAND
5062      ;-
5063
5064 031776      T23REST:
5065 031776      SAVREG
5066 032002      012701      030360      MOV      #T23PACKET,R1      ;SAVE THE REGISTERS
5067 032006      012721      100004      MOV      #100004,(R1)+      ;START OF THE PACKET
5068 032012      012721      030370      MOV      #T23DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK
5069 032016      005021      CLR      (R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
5070 032020      012721      000012      MOV      #10.,(R1)+      ;EXTENDED ADDRESS
5071 032024      012721      030400      MOV      #T23BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
5072 032030      005021      CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
5073 032032      012721      000024      MOV      #20.,(R1)+      ;LENGTH OF MESSAGE BUFFER
5074 032036      005021      CLR      (R1)+
5075 032040      012711      000000      MOV      #0,(R1)      ;SELECT DRIVE ZERO
5076 032044      012702      000030      MOV      #24.,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
5077 032050      012762      177777      030400      64$: MOV      #177777,T23BFR(R2)      ;ALL ONES TO MESSAGE BUFFER
5078 032056      005742      TST      -(R2)      ;BUMP DOWN TO NEXT LOCATION
5079 032060      020227      000000      CMP      R2,#0      ;R2 AT ZERO YET
5080 032064      001371      BNE      64$      ;KEEP GOING UNTIL DONE
5081 032066      000207      RTS      PC      ;RETURN
5082
5083
5084 032070      T23RT2:
5085 032070      SAVREG
5086 032074      012701      030470      MOV      #T23PK2,R1      ;SAVE THE REGISTERS
5087 032100      012721      100006      MOV      #100006,(R1)+      ;START OF THE PACKET
5088 032104      012721      030522      MOV      #T23BF2,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK
5089 032110      005021      CLR      (R1)+      ;ADDRESS OF DATA BLOCK
5090 032112      012721      000006      MOV      #6.,(R1)+      ;EXTENDED ADDRESS
5091 032116      012701      030522      MOV      #T23BF2,R1      ;SIZE OF DATA BLOCK IN BYTES
5092 032122      005021      CLR      (R1)+      ;POINT TO DATA SEL AREA

```

L11

CZTKFA TK 25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 100-1  
TEST 3: BASIC WRITE

SEQ 141

5093 032124 005021  
5094 032126 005011  
5095 032130 000207  
5096 032132  
5097 032132  
5098 032136 012701 030510  
5099 032142 012721 100005  
5100 032146 005021  
5101 032150 005021  
5102 032152 005011  
5103 032154 000207  
5104 032156  
032156  
032156 104401

T23RT3:

CLH (R1)+  
CLR (R1)  
RTS PC  
SAVREG  
MOV #T23PK3,R1  
MOV #100005,(R1)+  
CLR (R1)+  
CLR (R1)+  
CLR (R1)  
RTS PC  
ENDTST

;RETURN

;SAVE THE REGISTERS  
;START OF THE PACKET  
;WRITE TAPE, WITH ACK  
;ADDRESS OF DATA BLOCK  
;EXTENDED ADDRESS  
;SIZE OF DATA BLOCK  
;RETURN

L10041:

TRAP

C\$ETST

M11

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 102  
 TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 142

```

5107 .SBTTL TEST 4: BASIC READ DATA (FORWARD AND REVERSE)
5108 ;*
5109 ;
5110 ;THIS TEST VERIFIES THAT THE READ FORWARD AND READ REVERSE
5111 ;COMMANDS OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN
5112 ;DATA BUFFER BOUNDARIES, RECORD SIZES (UP TO 64K BYTES IF MEMORY
5113 ;SPACE IS AVAILIABLE), AND BYTE-SWAP CONTROL ARE USED. THIS TEST
5114 ;OF COURSE, FURTHER VERIFIES THE WRITE DATA COMMAND BY ACTUALLY
5115 ;READING AND VERIFYING WRITTEN DATA. ALSO TESTED ARE PROPER
5116 ;TERMINATIONS ON EXCEPTIONAL OR ERROR CONDITIONS: RECORD LENGTH
5117 ;LONG, RECORD LENGTH SHORT, READ REVERSE AT BOT, ILLEGAL DATA
5118 ;BUFFER ADDRESSES, ILLEGAL CODES IN THE MODE FIELD OF THE BASIC
5119 ;READ COMMAND, AND DATA BUFFERS IN NON-EXISTANT MEMORY. THE TEST
5120 ;
5121 ;
5122 ;THE TEST CONSISTS OF THE FOLLOWING 12 SUBTESTS
5123 ;
5124 ;
5125 ;
5126 ;-
5127 BGNTST
5128 032160 005037 002170 CLR FATFLG ;CLEAR FATAL ERROR FLAG
5129 032160 005037 003100 CLR KTFLG ;HOLD OFF KT11
5130 032170 012737 005762 002146 MOV #EPRT1,EPRTSW ;SET UP PRIMARY ERROR MESSAGE
5131 032176 005037 003102 CLR KTENABLE ;TURN OFF KT11
5132 032202 004737 020246 JSR PC,KTOFF ;TURN KT11 OFF
5137 032206 012700 043172 MOV #TST24ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
5138 032212 004737 017410 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
5139 032216 012737 000001 002164 MOV #1,LOOPCNT ;PERFORM 1 ITERATIONS
5140 ;
5141 032224 ;T24LOOP:
5142 ;

```



B12

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 103-1  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 144

```

5200 032322          ERRHRD  ERRNO,WRTMSG,SFMSG      ;WRITE CHARACTERISTIC FAILED
      032322 104456          TRAP                  C1ERRRD
      032324 000622          .WORD                  402
      032326 004754          .WORD                  WRTMSG
      032330 011666          .WORD                  SFMSG
5201 032332          241:    CKLOOP                  ;LOOP IF SELECTED
      032332 104406          TRAP                  C1CLP1
5202
5203 ;*****
5204 ;
5205 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5206 ;
5207 ;*****
5208
5209 032334 004737 010434      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
5210 032340 103407          BCS      301             ;BR, IF NO PROBLEM
5211 032342 010001          MOV      R0,R1            ;SAVE TSSR
5212 032344 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5216 032350          ERRHRD  ERRNO,T24RWN,PKTSSR      ;REWIND NOT ACCEPTED
      032350 104456          TRAP                  C1ERRRD
      032352 000623          .WORD                  403
      032354 042036          .WORD                  T24RWN
      032356 011700          .WORD                  PKTSSR
5217 032360          301:    CKLOOP                  ;LOOP IF SELECTED
      032360 104406          TRAP                  C1CLP1
5218
5219 ;*****
5220 ;
5221 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5222 ;
5223 ;*****
5224
5225 032362 013701 040576      MOV      T24BFR+6,R1      ;PICK UP XSTO
5226 032366 010102          MOV      R1,R2            ;SET UP EXPECTED
5227 032370 052702 000002      BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
5228 032374 020102          CMP      R1,R2            ;DOES EXP = REC D
5229 032376 001406          BEQ      401             ;BR, IF EQUAL (OK)
5230 032400 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5234 032404          ERRHRD  ERRNO,T24BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      032404 104456          TRAP                  C1ERRRD
      032406 000624          .WORD                  404
      032410 041553          .WORD                  T24BOT
      032412 015344          .WORD                  EXPREC
5235 032414          401:    CKLOOP                  ;LOOP IF SELECTED
      032414 104406          TRAP                  C1CLP1
5236 032416 012703 000400      MOV      #256,R3        ;RECORD SIZE
5237 032422 013737 003072 040702  MOV      FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
5238
5239 ;*****
5240 ;
5241 ;WRITE DATA,CVC=1,ACK COMMAND
5242 ;
5243 ;*****
5244
5245 032430 012737 140005 040700  MOV      #140005,T24PK3      ;WRITE DATA,CVC=1,ACK COMMAND
5246 032436 012704 040700      MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
5247 032442          651:

```

C12

CZIKFA TK-25 ERI END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 103-2  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 145

|      |        |        |        |        |                            |           |                                           |
|------|--------|--------|--------|--------|----------------------------|-----------|-------------------------------------------|
| 5248 | 032442 | 010300 |        | MOV    | R3,R0                      |           | ;SET PATTERN IN CORRECT REGISTER          |
| 5249 | 032444 | 004737 | 020374 | JSR    | PC,FILLMEM                 |           | ;FILL MEMORY WITH RECORD SIZE             |
| 5250 | 032450 | 010337 | 040706 | MOV    | R3,T24SZ                   |           | ;SET UP RECORD SIZE IN PACKET             |
| 5251 | 032454 | 010465 | 177776 | MOV    | R4,TSD8(R5)                |           | ;ISSUE COMMAND                            |
| 5252 | 032460 | 004737 | 017120 | JSR    | PC,WAITF                   |           | ;WAIT FOR SSR TO SET                      |
| 5253 | 032464 | 016501 | 000000 | MOV    | TSSR(R5),R1                |           | ;GET TSSR CONTENTS                        |
| 5254 | 032470 | 012702 | 000200 | MOV    | #SSR,R2                    |           | ;SET UP EXPECTED                          |
| 5255 | 032474 | 020102 |        | CMP    | R1,R2                      |           | ;ARE THEY EQUAL                           |
| 5256 | 032476 | 001406 |        | BEQ    | 75#                        |           | ;BR, IF OK                                |
| 5257 | 032500 | 004737 | 020102 | JSR    | PC,FATCHK                  |           | ;INC AND CHECK FOR MORE THAN 25 ERRORS    |
| 5261 |        |        |        |        |                            |           | ;SOFT ERROR, REALLY CHECKING THE          |
| 5262 |        |        |        |        |                            |           | ;READ COMMAND                             |
| 5263 | 032504 |        |        |        | ERRSOFT ERRNO,WRERR,PKTSSR |           | ;TSSR INCORRECT AFTER WRITE DATA          |
|      | 032504 | 104457 |        |        |                            | TRAP      | C#ERSOFT                                  |
|      | 032506 | 000625 |        |        |                            | .WORD     | 405                                       |
|      | 032510 | 005011 |        |        |                            | .WORD     | WRERR                                     |
|      | 032512 | 011700 |        |        |                            | .WORD     | PKTSSR                                    |
| 5264 | 032514 |        |        | 75#:   | CKLOOP                     |           | ;LOOP IF SELECTED                         |
|      | 032514 | 104406 |        |        |                            | TRAP      | C#CLP1                                    |
| 5265 | 032516 | 005723 |        | TST    | (R3)+                      |           | ;BUMP RECORD SIZE                         |
| 5266 | 032520 | 022703 | 000414 | CMP    | #268,R3                    |           | ;END OF RECORD YET                        |
| 5267 | 032524 | 001346 |        | BNE    | 65#                        |           | ;BR, IF MORE RECORDS TO WRITE             |
| 5268 | 032526 |        |        | 80#:   | CKLOOP                     |           | ;LOOP IF SELECTED                         |
|      | 032526 | 104406 |        |        |                            | TRAP      | C#CLP1                                    |
| 5269 | 032530 |        |        | 120#:  |                            |           |                                           |
| 5270 |        |        |        |        |                            |           |                                           |
| 5271 |        |        |        |        |                            |           |                                           |
| 5272 |        |        |        |        |                            |           |                                           |
| 5273 |        |        |        |        |                            |           |                                           |
| 5274 |        |        |        |        |                            |           |                                           |
| 5275 |        |        |        |        |                            |           |                                           |
| 5276 |        |        |        |        |                            |           |                                           |
| 5277 | 032530 | 012703 | 000012 | MOV    | #10,R3                     |           | ;SPECIAL MULTIPLE REWIND                  |
| 5278 | 032534 | 004737 | 010434 | 125#:  | JSR                        | PC,REWIND | ;ISSUE REWIND COMMAND                     |
| 5279 | 032540 | 103407 |        | BOS    | 130#                       |           | ;BR, IF NO PROBLEM                        |
| 5280 | 032542 | 010001 |        | MOV    | R0,R1                      |           | ;SAVE TSSR                                |
| 5281 | 032544 | 004737 | 020102 | JSR    | PC,FATCHK                  |           | ;INC AND CHECK FOR MORE THAN 25 ERRORS    |
| 5285 | 032550 |        |        | ERRHRD | ERRNO,T24RWN,PKTSSR        |           | ;REWIND NOT ACCEPTED                      |
|      | 032550 | 104456 |        |        |                            | TRAP      | C#ERRHRD                                  |
|      | 032552 | 000626 |        |        |                            | .WORD     | 406                                       |
|      | 032554 | 042036 |        |        |                            | .WORD     | T24RWN                                    |
|      | 032556 | 011700 |        |        |                            | .WORD     | PKTSSR                                    |
| 5286 | 032560 |        |        | 130#:  | CKLOOP                     |           | ;LOOP IF SELECTED                         |
|      | 032560 | 104406 |        |        |                            | TRAP      | C#CLP1                                    |
| 5287 |        |        |        |        |                            |           |                                           |
| 5288 |        |        |        |        |                            |           |                                           |
| 5289 |        |        |        |        |                            |           |                                           |
| 5290 |        |        |        |        |                            |           |                                           |
| 5291 |        |        |        |        |                            |           |                                           |
| 5292 |        |        |        |        |                            |           |                                           |
| 5293 |        |        |        |        |                            |           |                                           |
| 5294 | 032562 | 013701 | 040576 | MOV    | T24BFR+6,R1                |           | ;PICK UP XSTO                             |
| 5295 | 032566 | 010102 |        | MOV    | R1,R2                      |           | ;SET UP EXPECTED                          |
| 5296 | 032570 | 052702 | 000002 | BIS    | #BIT1,R2                   |           | ;SET BOT BIT IN EXPECTED                  |
| 5297 | 032574 | 020102 |        | CMP    | R1,R2                      |           | ;DOES EXP = REC'D                         |
| 5298 | 032576 | 001407 |        | BEQ    | 140#                       |           | ;BR, IF EQUAL (OK)                        |
| 5299 | 032600 | 077323 |        | SOS    | R3,125#                    |           | ;DO ANOTHER REWIND BEFORE REPORTING ERROR |

D12

CZIKFA, TK-25, FRT, END, FUNC #2, MACRO M1200, 20-APR-84 08:12 PAGE 103-3  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 146

```

5300 032602 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5304 032606      ERRHRD      ERRNO,T24BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      032606 104456      TRAP      C$ERHRD
      032610 000627      .WORD      407
      032612 041553      .WORD      T24BOT
      032614 016344      .WORD      EXPREC
5305 032616      140$: CKLOOP      ;LOOP IF SELECTED
      032616 104406      TRAP      C$CLP1
5306 032620 012703 000400      MOV      #256.,R3      ;RECORD SIZE
5307 032624 013737 003072 040702      MOV      FREE,T24RB      ;STARTING READ BUFFER ADDRESS
5308
5309      ;*****
5310      ;
5311      ;READ DATA,CVC=1,ACK COMMAND
5312      ;
5313      ;*****
5314
5315 032632 012737 140001 040700      MOV      #140001,T24PK3      ;READ DATA,CVC=1,ACK COMMAND
5316 032640 012704 040700      165$: MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
5317 032644 010337 040706      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
5318 032650 010465 177776      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
5319 032654 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5320 032660 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
5321 032664 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
5322 032670 020102      CMP      R1,R2      ;ARE THEY EQUAL
5323 032672 001406      BEQ      170$      ;BR, IF OK
5324 032674 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5328 032700      ERRHRD      ERRNO,RDERR,PKTSSR      ;TSSR INCORRECT AFTER READ DATA
      032700 104456      TRAP      C$ERHRD
      032702 000630      .WORD      408
      032704 005104      .WORD      RDERR
      032706 011700      .WORD      PKTSSR
5329 032710      170$: CKLOOP      ;LOOP IF SELECTED
      032710 104406      TRAP      C$CLP1
5330 032712 013702 003072      MOV      FREE,R2      ;GET BUFFER ADDRESS
5331 032716 010304      MOV      R3,R4      ;CURRENT RECORD SIZE
5332 032720 162704 000400      SUB      #256.,R4      ;FIRST LOCATION IN BUFFER
5333 032724 060204      173$: ADD      R2,R4      ;GET LOCATION IN BUFFER (ADDRESS)
5334 032726 021403      CMP      (R4),R3      ;CHECK DATA READ (R3=DATA ALSO)
5335 032730 001410      BEQ      180$      ;BR, IF ALL IS WELL
5336 032732 011401      MOV      (R4),R1      ;RECD DATA
5337 032734 010302      MOV      R3,R2      ;EXPECTED DATA
5338 032736 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5342 032742      ERRHRD      ERRNO,T24DTA,EXPREC      ;DATA READ NOT = WRITTEN
      032742 104456      TRAP      C$ERHRD
      032744 000631      .WORD      409
      032746 041620      .WORD      T24DTA
      032750 016344      .WORD      EXPREC
5343 032752      180$: CKLOOP      ;LOOP IF SELECTED
      032752 104406      TRAP      C$CLP1
5344 032754 005724      TST      (R4)+      ;BUMP TO NEXT LOCATION
5345 032756 160204      SUB      R2,R4      ;GET BACK TO CORRECT SIZE
5346 032760 020403      CMP      R4,R3      ;END OF RECORD YET
5347 032762 001360      BNE      173$      ;BR, IF NOT AT END OF RECORD
5348 032764 005723      TST      (R3)+      ;BUMP RECORD SIZE
5349 032766 022703 000412      CMP      #266.,R3      ;END OF RECORD YET
5350 032772 001322      BNE      165$      ;BR, IF MORE RECORDS TO WRITE

```

CZTKFA TK-25 FRT END FUNC 42 MACRO M1200 20-APR-84 08:12 PAGE 103-4  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 147

[illegible]



G12

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 104-1  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 149

```

5411 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5412 ;
5413 ;*****
5414 ;
5415 033110 004737 010434 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5416 033114 103407 BCS 30$ ;BR, IF NO PROBLEM
5417 033116 010001 MOV R0,R1 ;SAVE TSSR
5418 033120 004737 020102 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
5422 033124 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
; TRAP C$ERHRD
; .WORD 412
; .WORD T24RWN
; .WORD PKTSSR
5423 033134 104406 30$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
5424 ;
5425 ;*****
5426 ;
5427 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
5428 ;
5429 ;*****
5430 ;
5431 033136 013701 040576 MOV T24BFR+6,R1 ;PICK UP XST0
5432 033142 010102 MOV R1,R2 ;SET UP EXPECTED
5433 033144 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
5434 033150 020102 CMP R1,R2 ;DOES EXP = REC'D
5435 033152 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5436 033154 004737 020102 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
5440 033160 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
; TRAP C$ERHRD
; .WORD 413
; .WORD T24BOT
; .WORD EXPREC
5441 033170 104406 40$: CKLOOP ;LOOP IF SELECTED
; TRAP C$CLP1
5442 033172 012703 000400 MOV #256.,R3 ;RECORD SIZE
5443 033176 013737 003072 040702 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
5444 ;
5445 ;*****
5446 ;
5447 ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
5448 ;
5449 ;*****
5450 ;
5451 033204 012737 150005 040700 MOV #150005,T24PK3 ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
5452 033212 012704 040700 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5453 033216 65$: MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
5454 033216 010300 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
5455 033220 004737 020374 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
5456 033224 010337 040706 MOV R4,TSD8(R5) ;ISSUE COMMAND
5457 033230 010465 177776 JSR PC,WAITF ;WAIT FOR SSR TO SET
5458 033234 004737 017120 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5459 033240 016501 000000 MOV #SSR,R2 ;SET UP EXPECTED
5460 033244 012702 000200 CMP R1,R2 ;ARE THEY EQUAL
5461 033250 020102 BEQ 75$ ;BR, IF OK
5462 033252 001406 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
5463 033254 004737 020102

```

H12

CZTKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 104-2  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 150

```

5467
5468
5469 033260          ERRSOFT ERRNO,WRTErr,PKTSSR      ;SOFT ERROR, REALLY CHECKING THE
;READ COMMAND
;TSSR INCORRECT AFTER WRITE DATA
          033260 104457                                TRAP C$ERSOFT
          033262 000636                                .WORD 414
          033264 005011                                .WORD WRTERR
          033266 011700                                .WORD PKTSSR
5470 033270          75$: CKLOOP                      ;LOOP IF SELECTED
          033270 104406                                TRAP C$CLP1
5471 033272 005723          TST (R3),
5472 033274 022703 000414    CMP #268.,R3            ;BUMP RECORD SIZE
5473 033300 001346          BNE 65$                    ;END OF RECORD YET
5474 033302          80$: CKLOOP                      ;BR, IF MORE RECORDS TO WRITE
          033302 104406                                ;LOOP IF SELECTED
          033304          120$:                          TRAP C$CLP1
5475 033304
5476
5477 ;*****
5478 ;
5479 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5480 ;
5481 ;*****
5482
5483 033304 004737 010434    JSR PC,REWIND              ;CALL TAPE REWIND COMMAND
5484 033310 103407          BCS 130$                    ;BR, IF NO PROBLEM
5485 033312 010001          MOV R0,R1                    ;SAVE TSSR
5486 033314 004737 020102    JSR PC,FATCHK              ;INC AND CHECK FOR MORE THAN 25 ERRORS
5490 033320          ERRHRD ERRNO,T24RWN,EXPREC          ;REWIND NOT ACCEPTED
          033320 104456                                TRAP C$ERHRD
          033322 000637                                .WORD 415
          033324 042036                                .WORD T24RWN
          033326 016344                                .WORD EXPREC
5491 033330          130$: CKLOOP                      ;LOOP IF SELECTED
          033330 104406                                TRAP C$CLP1
5492
5493 ;*****
5494 ;
5495 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5496 ;
5497 ;*****
5498
5499 033332 013701 040576    MOV T24BFR+6,R1            ;PICK UP XSTO
5500 033336 010102          MOV R1,R2                    ;SET UP EXPECTED
5501 033340 052702 000002    BIS #BIT1,R2              ;SET BOT BIT IN EXPECTED
5502 033344 020102          CMP R1,R2                    ;DOES EXP = REC'D
5503 033346 001406          BEQ 140$                    ;BR, IF EQUAL (OK)
5504 033350 004737 020102    JSR PC,FATCHK              ;INC AND CHECK FOR MORE THAN 25 ERRORS
5508 033354          ERRHRD ERRNO,T24BOT,EXPREC          ;TAPE NOT AT BOT AFTER REWIND
          033354 104456                                TRAP C$ERHRD
          033356 000640                                .WORD 416
          033360 041553                                .WORD T24BOT
          033362 016344                                .WORD EXPREC
5509 033364          140$: CKLOOP                      ;LOOP IF SELECTED
          033364 104406                                TRAP C$CLP1
5510 033366 012703 000400    MOV #256.,R3              ;RECORD SIZE
5511 033372 013737 003072 040702    MOV FREE,T24RB      ;STARTING READ BUFFER ADDRESS
5512
5513 ;*****

```

|      |        |        |        |        |        |                              |                     |  |                                       |
|------|--------|--------|--------|--------|--------|------------------------------|---------------------|--|---------------------------------------|
| 5514 |        |        |        |        |        |                              |                     |  |                                       |
| 5515 |        |        |        |        |        | READ DATA,IE,ACK,SWB COMMAND |                     |  |                                       |
| 5516 |        |        |        |        |        |                              |                     |  |                                       |
| 5517 |        |        |        |        |        | *****                        |                     |  |                                       |
| 5518 |        |        |        |        |        |                              |                     |  |                                       |
| 5519 | 033400 | 012737 | 110001 | 040700 |        | MOV                          | #110001,T24PK3      |  | READ DATA,IE,ACK,SWB COMMAND          |
| 5520 | 033406 | 012704 | 040700 |        | 165\$: | MOV                          | #T24PK3,R4          |  | SET UP R4 WITH PACKET ADDRESS         |
| 5521 | 033412 | 010337 | 040706 |        |        | MOV                          | R3,T24SZ            |  | SET UP RECORD SIZE IN PACKET          |
| 5522 | 033416 | 010465 | 177776 |        |        | MOV                          | R4,TSD8(R5)         |  | ISSUE COMMAND                         |
| 5523 | 033422 | 004737 | 017120 |        |        | JSR                          | PC,WAITF            |  | WAIT FOR SSR TO SET                   |
| 5524 | 033426 | 016501 | 000000 |        |        | MOV                          | TSSR(R5),R1         |  | GET TSSR CONTENTS                     |
| 5525 | 033432 | 012702 | 000200 |        |        | MOV                          | #SSR,R2             |  | SET UP EXPECTED                       |
| 5526 | 033436 | 020102 |        |        |        | CMP                          | R1,R2               |  | ARE THEY EQUAL                        |
| 5527 | 033440 | 001406 |        |        |        | BEQ                          | 170\$               |  | BR, IF OK                             |
| 5528 | 033442 | 004737 | 020102 |        |        | JSR                          | PC,FATCHK           |  | INC AND CHECK FOR MORE THAN 25 ERRORS |
| 5532 | 033446 |        |        |        |        | ERRHRD                       | ERRNO,RDERR,PKTSSR  |  | TSSR INCORRECT AFTER READ DATA        |
|      | 033446 | 104456 |        |        |        |                              |                     |  | TRAP C\$ERHRD                         |
|      | 033450 | 000641 |        |        |        |                              |                     |  | .WORD 417                             |
|      | 033452 | 005104 |        |        |        |                              |                     |  | .WORD RDERR                           |
|      | 033454 | 011700 |        |        |        |                              |                     |  | .WORD PKTSSR                          |
| 5533 | 033456 |        |        |        | 170\$: | CKLOOP                       |                     |  | LOOP IF SELECTED                      |
|      | 033456 | 104406 |        |        |        |                              |                     |  | TRAP C\$CLP1                          |
| 5534 | 033460 | 013702 | 003072 |        |        | MOV                          | FREE,R2             |  | GET BUFFER ADDRESS                    |
| 5535 | 033464 | 010304 |        |        |        | MOV                          | R3,R4               |  | CURRENT RECORD SIZE                   |
| 5536 | 033466 | 162704 | 000400 |        |        | SUB                          | #256.,R4            |  | FIRST LOCATION IN BUFFER              |
| 5537 | 033472 | 060204 |        |        | 173\$: | ADD                          | R2,R4               |  | GET LOCATION IN BUFFER (ADDRESS)      |
| 5538 | 033474 | 021403 |        |        |        | CMP                          | (R4),R3             |  | CHECK DATA READ (R3=DATA ALSO)        |
| 5539 | 033476 | 001410 |        |        |        | BEQ                          | 180\$               |  | BR, IF ALL IS WELL                    |
| 5540 | 033500 | 011401 |        |        |        | MOV                          | (R4),R1             |  | RECD DATA                             |
| 5541 | 033502 | 010302 |        |        |        | MOV                          | R3,R2               |  | EXPECTED DATA                         |
| 5542 | 033504 | 004737 | 020102 |        |        | JSR                          | PC,FATCHK           |  | INC AND CHECK FOR MORE THAN 25 ERRORS |
| 5546 | 033510 |        |        |        |        | ERRHRD                       | ERRNO,T24DTA,EXPREC |  | DATA READ NOT = WRITTEN               |
|      | 033510 | 104456 |        |        |        |                              |                     |  | TRAP C\$ERHRD                         |
|      | 033512 | 000642 |        |        |        |                              |                     |  | .WORD 418                             |
|      | 033514 | 041620 |        |        |        |                              |                     |  | .WORD T24DTA                          |
|      | 033516 | 016344 |        |        |        |                              |                     |  | .WORD EXPREC                          |
| 5547 | 033520 |        |        |        | 180\$: | CKLOOP                       |                     |  | LOOP IF SELECTED                      |
|      | 033520 | 104406 |        |        |        |                              |                     |  | TRAP C\$CLP1                          |
| 5548 | 033522 | 005724 |        |        |        | TST                          | (R4),               |  | BUMP TO NEXT LOCATION                 |
| 5549 | 033524 | 160204 |        |        |        | SUB                          | R2,R4               |  | SET SIZE TO CORRECT VALUE             |
| 5550 | 033526 | 020403 |        |        |        | CMP                          | R4,R3               |  | END OF RECORD YET                     |
| 5551 | 033530 | 001360 |        |        |        | BNE                          | 173\$               |  | BR, IF NOT AT END OF RECORD           |
| 5552 | 033532 | 005723 |        |        |        | TST                          | (R3),               |  | BUMP RECORD SIZE                      |
| 5553 | 033534 | 022703 | 000412 |        |        | CMP                          | #266.,R3            |  | END OF RECORD YET                     |
| 5554 | 033540 | 001322 |        |        |        | BNE                          | 165\$               |  | BR, IF MORE RECORDS TO WRITE          |
| 5555 | 033542 |        |        |        | 190\$: | CKLOOP                       |                     |  | LOOP IF SELECTED                      |
|      | 033542 | 104406 |        |        |        |                              |                     |  | TRAP C\$CLP1                          |
| 5556 | 033544 |        |        |        |        | ENDSUB                       |                     |  | >>>>>>>>> END SUBTEST >>>>>>>>        |
|      | 033544 |        |        |        |        |                              |                     |  | L10051:                               |
|      | 033544 | 104403 |        |        |        |                              |                     |  | TRAP C\$ESUB                          |
| 5557 | 033546 | 023727 | 002170 | 000031 |        | CMP                          | FATFLG,#25.         |  | IS ERROR COUNT AT 25                  |
| 5558 | 033554 | 002402 |        |        |        | BLT                          | 999\$               |  | BR, IF LESS THAN 25                   |
| 5559 | 033556 | 004737 | 020154 |        |        | JSR                          | PC,CKDROP           |  | TRY TO DROP THE UNIT                  |
| 5560 | 033562 |        |        |        | 999\$: |                              |                     |  |                                       |



K12

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 105-1  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 153

```

5614 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5615 ;
5616 ;*****
5617 ;
5618 033656 004737 010434 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
5619 033662 103407 BCS 30$ ;BR, IF NO PROBLEM
5620 033664 010001 MOV R0,R1 ;SAVE TSSR
5621 033666 004737 020102 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
5625 033672 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      033672 104456 TRAP C$ERHRD
      033674 000645 .WORD 421
      033676 042036 .WORD T24RWN
      033700 011700 .WORD PKTSSR
5626 033702 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      033702 104406
5627 ;*****
5628 ;
5629 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5630 ;
5631 ;*****
5632 ;
5633 ;
5634 033704 013701 040576 MOV T24BFR+6,R1 ;PICK UP XSTO
5635 033710 010102 MOV R1,R2 ;SET UP EXPECTED
5636 033712 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
5637 033716 020102 CMP R1,R2 ;DOES EXP = REC'D
5638 033720 001406 BEQ 40$ ;BR, IF EQUAL (OK)
5639 033722 004737 020102 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
5643 033726 ERRHRD ERRNO,T24BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      033726 104456 TRAP C$ERHRD
      033730 000646 .WORD 422
      033732 041553 .WORD T24BOT
      033734 016344 .WORD EXPREC
5644 033736 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
      033736 104406
5645 033740 012703 001000 MOV #512,R3 ;RECORD SIZE
5646 033744 013737 003072 040702 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
5647 ;*****
5648 ;
5649 ;WRITE DATA,ACK,CVC=1 COMMAND
5650 ;
5651 ;*****
5652 ;
5653 ;
5654 033752 012737 140005 040700 MOV #140005,T24PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
5655 033760 012704 040700 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
5656 033764 65$:
5657 033764 010337 040706 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
5658 033770 010465 177776 MOV R4,TSD8(R5) ;ISSUE COMMAND
5659 033774 004737 017120 JSR PC,WAITF ;WAIT FOR SSR TO SET
5660 034000 016501 000000 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
5661 034004 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
5662 034010 020102 CMP R1,R2 ;ARE THEY EQUAL
5663 034012 001406 BEQ 75$ ;BR, IF OK
5664 034014 004737 020102 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
5668 ;SOFT ERROR, REALLY CHECKING THE
5669 ;READ DATA COMMAND

```

```

5670 034020          ERRSOFT ERRNO,WRterr,PKTSSR      ;TSSR INCORRECT AFTER WRITE DATA
      034020 104457
      034022 000647
      034024 005011
      034026 011700
5671 034030          75$: CKLOOP                      ;LOOP IF SELECTED
      034030 104406
5672 034032          120$:
5673
5674
5675
5676
5677
5678
5679
5680 034032 004737 010434      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
5681 034036 103407          BCS      130$                ;BR, IF NO PROBLEM
5682 034040 010001          MOV      R0,R1                ;SAVE TSSR
5683 034042 004737 020102      JSR      PC,FATCHK          ;INC AND CHECK FOR MORE THAN 25 ERRORS
5687 034046          ERRHRD ERRNO,T24RWN,PKTSSR          ;REWIND NOT ACCEPTED
      034046 104456
      034050 000650
      034052 042036
      034054 011700
5688 034056          130$: CKLOOP                      ;LOOP IF SELECTED
      034056 104406
5689
5690
5691
5692
5693
5694
5695
5696 034060 013701 040576      MOV      T24BFR+6,R1        ;PICK UP XST0
5697 034064 010102          MOV      R1,R2                ;SET UP EXPECTED
5698 034066 052702 000002      BIS      #BIT1,R2          ;SET BOT BIT IN EXPECTED
5699 034072 020102          CMP      R1,R2                ;DOES EXP = REC'D
5700 034074 001406          BEQ      140$                ;BR, IF EQUAL (OK)
5701 034076 004737 020102      JSR      PC,FATCHK          ;INC AND CHECK FOR MORE THAN 25 ERRORS
5705 034102          ERRHRD ERRNO,T24BOT,EXPREC          ;TAPE NOT AT BOT AFTER REWIND
      034102 104456
      034104 000651
      034106 041553
      034110 016344
5706 034112          140$: CKLOOP                      ;LOOP IF SELECTED
      034112 104406
5707 034114 012703 000400      MOV      #256,R3            ;RECORD SIZE
5708 034120 013737 003072 040702  MOV      FREE,T24RB        ;STARTING READ BUFFER ADDRESS
5709
5710
5711
5712
5713
5714
5715
5716 034126 012737 140001 040700  MOV      #140001,T24PK3      ;READ DATA,ACK,CVC=1 COMMAND
5717 034134 012704 040700      155$: MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS

```

M12

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 105-3  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 155

```

5718 034140 010337 040706      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
5719 034144 010465 177776      MOV      R4,TSD8(R5)    ;ISSUE COMMAND
5720 034150 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5721 034154 016501 000000      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
5722 034160 012702 100204      MOV      #SSR!SC!BIT2,R2  ;SET UP EXPECTED
5723 034164 020102              CMP      R1,R2      ;ARE THEY EQUAL
5724 034166 001406              BEQ      170$      ;BR, IF OK
5725 034170 004737 020102      JSR      PC,FATCHK    ;INC AND CHECK FOR MORE THAN 25 ERRORS
5729 034174              ERRHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERRHRD
                                .WORD     426
                                .WORD     T24TRL
                                .WORD     PKTSSR
5730 034204              170$:  CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
5731 034204 104406
5732              ;*****
5733              ;
5734              ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
5735              ;
5736              ;*****
5737
5738 034206 013701 040576      MOV      T24BFR+6,R1    ;GET MESSAGE BUFFER
5739 034212 010102              MOV      R1,R2      ;SET UP EXPECTED
5740 034214 052702 010000      BIS      #BIT12,R2    ;SET THE RLL BIT IN EXPECTED
5741 034220 020102              CMP      R1,R2      ;ARE THEY EQUAL
5742 034222 001406              BEQ      180$      ;BR, IF EQUAL (ALL IS WELL)
5743 034224 004737 020102      JSR      PC,FATCHK    ;INC AND CHECK FOR MORE THAN 25 ERRORS
5747 034230              ERRHRD  ERRNO,T24LON,EXPREC ;THE RLL BIT WAS NOT SET IN XST0
                                TRAP      C$ERRHRD
                                .WORD     427
                                .WORD     T24LON
                                .WORD     EXPREC
5748 034240              180$:  ENDSUB
5749 034240              ;>>>>>>>>> END SUBTEST >>>>>>>>>
                                L10052:
                                TRAP      C$ESUB
5750 034242 023727 002170 000031  CMP      FATFLG,#25,    ;IS ERROR COUNT AT 25
5751 034250 002402              BLT      999$      ;BR, IF LESS THAN 25
5752 034252 004737 020154      JSR      PC,CKDROP    ;TRY TO DROP THE UNIT
5753 034256              999$:

```



```

5808 034350      241:  CKLOOP                      ;LOOP IF SELECTED
      034350 104406                                TRAP  C1CLP1
5809
5810 ;*****
5811 ;
5812 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5813 ;
5814 ;*****
5815
5816 034352 004737 010434      JSR  PC,REWIND          ;CALL TAPE REWIND COMMAND
5817 034356 103407            BCS  301              ;BR, IF NO PROBLEM
5818 034360 010001            MOV  R0,R1              ;SAVE TSSR
5819 034362 004737 020102      JSR  PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
5823 034366            ERRHRD  ERRNO,T24RWN,PKTSSR    ;REWIND NOT ACCEPTED
      034366 104456                                TRAP  C1ERRRD
      034370 000656                                .WORD  430
      034372 042036                                .WORD  T24RWN
      034374 011700                                .WORD  PKTSSR
5824 034376      301:  CKLOOP                      ;LOOP IF SELECTED
      034376 104406                                TRAP  C1CLP1
5825 034400 012703 000400      MOV  #256,R3          ;RECORD SIZE
5826 034404 013737 003072 040702  MOV  FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
5827
5828 ;*****
5829 ;
5830 ;WRITE DATA,ACK,CVC-1 COMMAND
5831 ;
5832 ;*****
5833
5834 034412 012737 140005 040700      MOV  #140005,T24PK3    ;WRITE DATA,ACK,CVC-1 COMMAND
5835 034420 012704 040700      MOV  #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
5836 034424
5837 034424 010337 040706      651:  MOV  R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
5838 034430 010465 177776      MOV  R4,TSD8(R5)        ;ISSUE COMMAND
5839 034434 004737 017120      JSR  PC,WAITF          ;WAIT FOR SSR TO SET
5840 034440 016501 000000      MOV  TSSR(R5),R1        ;GET TSSR CONTENTS
5841 034444 012702 000200      MOV  #SSR,R2          ;SET UP EXPECTED
5842 034450 020102            CMP  R1,R2              ;ARE THEY EQUAL
5843 034452 001406            BEQ  751              ;BR, IF OK
5844 034454 004737 020102      JSR  PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
5848
5849
5850 034460            ERRSOF  ERRNO,WTERR,PKTSSR      ;SOFT ERROR, REALLY CHECKING THE
      034460 104457                                ;READ DATA COMMAND
      034462 000657                                ;TSSR INCORRECT AFTER WRITE DATA
      034464 005011                                TRAP  C1ERSOF
      034466 011700                                .WORD  431
      034470 011700                                .WORD  WTERR
      034470 104406                                .WORD  PKTSSR
5851 034470      751:  CKLOOP                      ;LOOP IF SELECTED
      034470 104406                                TRAP  C1CLP1
5852 034472
5853
5854 ;*****
5855 ;
5856 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5857 ;
5858 ;*****
5859

```

C13

C2TKFA TK-25 FRI END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 106-2  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 158

```

5860 034472 004737 010434      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
5861 034476 103407              BCS      1301          ;BR, IF NO PROBLEM
5862 034500 010001              MOV      R0,R1      ;SAVE TSSR
5863 034502 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5867 034506              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
      034506 104456              TRAP      C#ERRHRD
      034510 000660              .WORD    432
      034512 042036              .WORD    T24RWN
      034514 011700              .WORD    PKTSSR
5868 034516              1301:  CKLOOP              ;LOOP IF SELECTED
      034516 104406              TRAP      C#CLP1
5869 034520 012703 001000      MOV      #512.,R3      ;RECORD SIZE
5870 034524 013737 003072 040702  MOV      FREE,T24RB      ;STARTING READ BUFFER ADDRESS
5871
5872      ;*****
5873      ;
5874      ;READ DATA,ACK,CVC=1 COMMAND
5875      ;
5876      ;*****
5877
5878 034532 012737 140001 040700      MOV      #140001,T24PK3      ;READ DATA,ACK,CVC=1 COMMAND
5879 034540 012704 040700      1651:  MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
5880 034544 010337 040706      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
5881 034550 010465 177776      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
5882 034554 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
5883 034560 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
5884 034564 012702 100204      MOV      #SSR!SC!BIT2,R2      ;SET UP EXPECTED
5885 034570 020102      LMP      R1,R2      ;ARE THEY EQUAL
5886 034572 001406      BEQ      1701          ;BR, IF OK
5887 034574 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5391 034600              ERRHRD  ERRNO,T24TRL,EXPREC      ;TSSR INCORRECT AFTER READ DATA
      034600 104456              TRAP      C#ERRHRD
      034602 000661              .WORD    433
      034604 043104              .WORD    T24TRL
      034606 016344              .WORD    EXPREC
5892 034610              1701:  CKLOOP              ;LOOP IF SELECTED
      034610 104406              TRAP      C#CLP1
5893
5894      ;*****
5895      ;
5896      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
5897      ;
5898      ;*****
5899
5900 034612 013701 040576      MOV      T24BFR+6,R1      ;GET MESSAGE BUFFER
5901 034616 010102      MOV      R1,R2      ;SET UP EXPECTED
5902 034620 052702 040000      BIS      #BIT14,R2      ;SET THE RLS BIT IN EXPECTED
5903 034624 020102      CMP      R1,R2      ;ARE THEY EQUAL
5904 034626 001406      BEQ      1801          ;OR, IF EQUAL (ALL IS WELL)
5905 034630 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
5909 034634              ERRHRD  ERRNO,T24LOP,EXPREC      ;THE RLL BIT WAS NOT SET IN XSTO
      034634 104456              TRAP      C#ERRHRD
      034636 000662              .WORD    434
      034640 042734              .WORD    T24LOP
      034642 016344              .WORD    EXPREC
5910 034644
5911 034644 013701 040574      1801:  MOV      T24BFR+4,R1      ;PICK UP RESIDUAL BYTE COUNTER

```

D13

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 159

|  |      |        |        |        |        |        |                     |  |                                        |
|--|------|--------|--------|--------|--------|--------|---------------------|--|----------------------------------------|
|  | 5912 | 034650 | 012702 | 000400 |        | MOV    | #256.,R2            |  | ;THIS SHOULD BE THE DIFFERENCE         |
|  | 5913 | 034654 | 020102 |        |        | CMP    | R1,R2               |  | ;IS THE DIFFERENCE CORRECT             |
|  | 5914 | 034656 | 001406 |        |        | BEQ    | 190\$               |  | ;BR, IF CORRECT                        |
|  | 5915 | 034660 | 004737 | 020102 |        | JSR    | PC,FATCHK           |  | ;INC AND CHECK FOR MORE THAN 25 ERRORS |
|  | 5919 | 034664 |        |        |        | ERRHRD | ERRNO,T24PBP,EXPREC |  | ;RBPCR NOT CORRECT                     |
|  |      | 034664 | 104456 |        |        |        |                     |  | TRAP C#ERHRD                           |
|  |      | 034666 | 000663 |        |        |        |                     |  | .WORD 435                              |
|  |      | 034670 | 043016 |        |        |        |                     |  | .WORD T24PBP                           |
|  |      | 034672 | 016344 |        |        |        |                     |  | .WORD EXPREC                           |
|  | 5920 | 034674 |        |        | 190\$: | CKLOOP |                     |  | ;LOOP IF SELECTED                      |
|  |      | 034674 | 104406 |        |        |        |                     |  | TRAP C#CLP1                            |
|  | 5921 | 034676 |        |        |        | ENDSUB |                     |  | ; >>>>>>>> END SUBTEST >>>>>>>>        |
|  |      | 034676 |        |        |        |        |                     |  | L10053:                                |
|  |      | 034676 | 104403 |        |        |        |                     |  | TRAP C#ESUB                            |
|  | 5922 | 034700 | 023727 | 002170 | 000031 | CMP    | FATFLG,#25.         |  | ;IS ERROR COUNT AT 25                  |
|  | 5923 | 034706 | 002402 |        |        | BLT    | 999\$               |  | ;BR, IF LESS THAN 25                   |
|  | 5924 | 034710 | 004737 | 020154 |        | JSR    | PC,CKDROP           |  | ;TRY TO DROP THE UNIT                  |
|  | 5925 | 034714 |        |        | 999\$: |        |                     |  |                                        |



F13

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 107-1  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 161

```

5980 035006      104406      24$: CKLOOP                      ;LOOP IF SELECTED
                                TRAP      C$CLP1
5981 035006
5982
5983 ;*****
5984 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
5985 ;
5986 ;*****
5987
5988 035010 004737 010434      JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
5989 035014 103407          BCS      30$              ;BR, IF NO PROBLEM
5990 035016 010001          MOV      R0,R1              ;SAVE TSSR
5991 035020 004737 020102      JSR      PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
5995 035024          ERRHRD  ERRNO,T24RWN,PKTSSR        ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD     438
                                .WORD     T24RWN
                                .WORD     PKTSSR
5996 035034      104406      30$: CKLOOP                      ;LOOP IF SELECTED
                                TRAP      C$CLP1
5997 035036 012703 000400      MOV      #256.,R3          ;RECORD SIZE
5998 035042 013737 003072 040702  MOV      FREE,T24R8      ;STARTING WRITE BUFFER ADDRESS
5999
6000 ;*****
6001 ;
6002 ;WRITE DATA,ACK,CVC-1 COMMAND
6003 ;
6004 ;*****
6005
6006 035050 012737 140005 040700  MOV      #140005,T24PK3      ;WRITE DATA,ACK,CVC-1 COMMAND
6007 035056 012704 040700      MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6008 035062
6009 035062 010300      65$: MOV      R3,R0              ;SET PATTERN IN CORRECT REGISTER
6010 035064 004737 020374      JSR      PC,FILLMEM        ;FILL MEMORY WITH RECORD SIZE
6011 035070 010337 040706      MOV      R3,T24SZ          ;SET UP RECORD SIZE IN PACKET
6012 035074 010465 177776      MOV      R4,T24B(R5)        ;ISSUE COMMAND
6013 035100 004737 017120      JSR      PC,WAITF          ;WAIT FOR SSR TO SET
6014 035104 016501 000000      MOV      TSSR(R5),R1        ;GET TSSR CONTENTS
6015 035110 012702 000200      MOV      #SSR,R2          ;SET UP EXPECTED
6016 035114 020102          CMP      R1,R2              ;ARE THEY EQUAL
6017 035116 001406          BEQ      75$              ;BR, IF OK
6018 035120 004737 020102      JSR      PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
6022
6023 ;SOFT ERROR, REALLY CHECKING THE
6024 035124          ERRSOFT ERRNO,WRterr,PKTSSR          ;READ DATA COMMAND
                                ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERSOFT
                                .WORD     439
                                .WORD     WRterr
                                .WORD     PKTSSR
6025 035134      104406      75$: CKLOOP                      ;LOOP IF SELECTED
                                TRAP      C$CLP1
6026 035136 005723          TST      (R3),              ;BUMP RECORD SIZE
6027 035140 022703 000414      CMP      #268.,R3          ;END OF RECORD YET
6028 035144 001346          BNE      65$              ;BR, IF MORE RECORDS TO WRITE
6029 035146      104406      80$: CKLOOP                      ;LOOP IF SELECTED
                                TRAP      C$CLP1
6030 035150 005743          TST      -(R3)              ;SET BACK TO 512.

```





```

6134 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6135 ;
6136 ;*****
6137 ;
6138 035436 004737 010434 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
6139 035442 103407 BCS 30$ ;BR, IF NO PROBLEM
6140 035444 010001 MOV R0,R1 ;SAVE TSSR
6141 035446 004737 020102 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
6145 035452 ERRHRD ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
        035452 104456 TRAP C$ERHRD
        035454 000674 .WORD 444
        035456 042036 .WORD T24RWN
        035460 011700 .WORD PKTSSR
6146 035462 30$: CKLOOP ;LOOP IF SELECTED
        035462 104406 TRAP C$CLP1
6147 035464 012703 000400 MOV #256.,R3 ;RECORD SIZE
6148 035470 013737 003072 040702 MOV FREE,T24RB ;STARTING WRITE BUFFER ADDRESS
6149 ;
6150 ;*****
6151 ;
6152 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
6153 ;
6154 ;*****
6155 ;
6156 035476 012737 150005 040700 MOV #150005,T24PK3 ;WRITE DATA,ACK,CVC=1,SWB COMMAND
6157 035504 012704 040700 MOV #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6158 035510 65$:
6159 035510 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
6160 035512 004737 020374 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
6161 035516 010337 040706 MOV R3,T24SZ ;SET UP RECORD SIZE IN PACKET
6162 035522 010465 177776 MOV R4,TSD8(R5) ;ISSUE COMMAND
6163 035526 004737 017120 JSR PC,WAITF ;WAIT FOR SSR TO SET
6164 035532 016501 000000 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
6165 035536 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
6166 035542 020102 CMP R1,R2 ;ARE THEY EQUAL
6167 035544 001406 BEQ 75$ ;BR, IF OK
6168 035546 004737 020102 JSR PC,FATCHK ;INC AND CHECK FOR MORE THAN 25 ERRORS
6172 ;
6173 ;SOFT ERROR, REALLY CHECKING THE
6174 035552 ERRSOFT ERRNO,WRERR,PKTSSR ;READ DATA COMMAND
        035552 104457 ;TSSR INCORRECT AFTER WRITE DATA
        035554 000675 TRAP C$ERSOFT
        035556 005011 .WORD 445
        035560 011700 .WORD WRERR
        035562 011700 .WORD PKTSSR
6175 035562 75$: CKLOOP ;LOOP IF SELECTED
        035562 104406 TRAP C$CLP1
6176 035564 005723 TST (R3), ;BUMP RECORD SIZE
6177 035566 022703 000414 CMP #268.,R3 ;END OF RECORD YET
6178 035572 001346 BNE 65$ ;BR, IF MORE RECORDS TO WRITE
6179 035574 80$: CKLOOP ;LOOP IF SELECTED
        035574 104406 TRAP C$CLP1
6180 035576 005743 TST -(R3) ;SET RECORD SIZE BACK TO 512.
6181 035600 013737 003072 040702 MOV FREE,T24RB ;STARTING READ BUFFER ADDRESS
6182 ;
6183 ;*****
6184 ;
6185 ;READ REVERSE DATA,ACK,SWB COMMAND

```

J13

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 108-2  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 165

```

6186
6187
6188
6189 035606 012737 110401 040700      MOV      #110401,T24PK3      ;READ REVERSE DATA,ACK,SWB COMMAND
6190 035614 012704 040700      165$: MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6191 035620 010337 040706      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
6192 035624 010465 177776      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
6193 035630 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6194 035634 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
6195 035640 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
6196 035644 020102      CMP      R1,R2      ;ARE THEY EQUAL
6197 035646 001406      BEQ      170$      ;BR, IF OK
6198 035650 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6202 035654      ERRHRD      ERRNO,T24WDC,EXPREC      ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERHRD
                                .WORD      446
                                .WORD      T24WDC
                                .WORD      EXPREC
6203 035664      170$: CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
6204 035666 013702 003072      MOV      FREE,R2      ;GET BUFFER ADDRESS
6205 035672 010304      MOV      R3,R4      ;CURRENT RECORD SIZE
6206 035674 162704 000400      SUB      #256.,R4      ;FIRST LOCATION IN BUFFER
6207 035700 060204      173$: ADD      R2,R4      ;SET POINTER TO FRAME (WORD)
6208 035702 021403      CMP      (R4),R3      ;CHECK DATA READ (R3=DATA ALSO)
6209 035704 001410      BEQ      180$      ;BR, IF ALL IS WELL
6210 035706 011401      MOV      (R4),R1      ;RECD DATA
6211 035710 010302      MOV      R3,R2      ;EXPECTED DATA
6212 035712 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6216 035716      ERRHRD      ERRNO,T24DTA,EXPREC      ;DATA READ NOT = WRITTEN
                                TRAP      C$ERHRD
                                .WORD      447
                                .WORD      T24DTA
                                .WORD      EXPREC
6217 035726      180$: CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
6218 035730 005724      TST      (R4),      ;BUMP TO NEXT LOCATION
6219 035732 160204      SUB      R2,R4      ;GET RID OF BASE ADDRESS
6220 035734 020403      CMP      R4,R3      ;END OF RECORD YET
6221 035736 001360      BNE      173$      ;BR, IF NOT AT END OF RECORD
6222 035740 005743      TST      -(R3)      ;BUMP RECORD SIZE
6223 035742 022703 000400      CMP      #256.,R3      ;END OF RECORD YET
6224 035746 001322      BNE      165$      ;BR, IF MORE RECORDS TO WRITE
6225 035750      190$: CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
6226 035752      104406      ENDSUB      ;>>>>>>>>> END SUBTEST >>>>>>>>
                                L10055:
                                TRAP      C$ESUB
6227 035754 023727 002170 000031      CMP      FATFLG,#25,      ;IS ERROR COUNT AT 25
6228 035762 002402      BLT      999$      ;BR, IF LESS THAN 25
6229 035764 004737 020154      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
6230 035770      999$:

```

```
6232      ;+  
6233      ;  
6234      ;TEST 4, SUBTEST 7  
6235      ;  
6236      ;VERIFIES THAT A READ REVERSE COMMAND READING A RECORD  
6237      ;LONGER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE  
6238      ;STATUS ALERT TERMINATION WITH THE RECORD LENGTH LONG  
6239      ;(RLL) BIT SET.  
6240      ;  
6241      ;  
6242      ;-  
6243      ;-  
6244      035770          BGNSUB                                ;>>>>>>>>> BEGIN SUBTEST >>>>>>>>  
        035770                                          T4.7:  
        035770    104402          TRAP              C$BSUB  
6245      035772    004737    043374          JSR       PC,T2ART3           ;SET UP OTHER COMMAND PACKET  
6246      035776    004737    043240          JSR       PC,T2AREST         ;SET COMMAND PACKET  
6247      036002    004737    043332          JSR       PC,T2ART2           ;SET UP OTHER COMMAND PACKET  
6248      ;  
6249      ;*****  
6250      ;  
6251      ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR  
6252      ;  
6253      ;*****  
6254      ;  
6255      036006    004737    016644          JSR       PC,SOFINIT           ;DO INITIALIZ ON CONTROLLER  
6256      036012    103407                    BCS       20$                ;BR IF INIT WAS OK  
6257      036014    004737    020102          JSR       PC,FATCHK            ;INC AND CHECK FOR MORE THAN 25 ERRORS  
6261      036020    010001                    MOV       R0,R1               ;CONTENTS OF TSSR REGISTER  
6262      036022                    ERDFF     ERNO,SFIERR,SFMSG             ;FATAL ERROR TSSR WAS NOT OK  
        036022    104455                                  TRAP              C$ERDF  
        036024    000700                                  .WORD              448  
        036026    003550                                  .WORD              SFIERR  
        036030    011666                                  .WORD              SFMSG  
6263      036032    20$:                   MOV       #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS  
6264      036032    012704    040550                                     ;  
6265      ;  
6266      ;*****  
6267      ;  
6268      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)  
6269      ;  
6270      ;*****  
6271      ;  
6272      036036    004737    010332          JSR       PC,WRTCHR            ;ISSUE WRITE CHARACTERISTICS  
6273      036042    103407                    BCS       24$                ;BR, IF COMMAND ISSUED OK  
6274      036044    004737    020102          JSR       PC,FATCHK            ;INC AND CHECK FOR MORE THAN 25 ERRORS  
6278      036050    010001                    MOV       R0,R1               ;SAVE CONTENTS OF TSSR  
6279      036052          ERHRD     ERNO,WRTMSG,SFMSG                     ;WRITE CHARACTERISTIC FAILED  
        036052    104456                                  TRAP              C$ERHRD  
        036054    000701                                  .WORD              449  
        036056    004754                                  .WORD              WRTMSG  
        036060    011666                                  .WORD              SFMSG  
6280      036062    24$:   CKLOOP                               ;LOOP IF SELECTED  
        036062    104406                                  TRAP              C$CLP1  
6281      ;  
6282      ;*****  
6283      ;
```

L13

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 109-1  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 167

```

6284      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6285      ;
6286      ;*****
6287      ;
6288 036064 004737 010434      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
6289 036070 004737 017236      JSR      PC,CHKTSSR      ;SEE HOW TSSR IS
6290 036074 103407              BCS      30$      ;BR, IF NO PROBLEM
6291 036076 010001              MOV      R0,R1      ;SAVE TSSR
6292 036100 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6296 036104              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
        036104 104456              TRAP      C1ERRHRD
        036106 000702              .WORD    450
        036110 042036              .WORD    T24RWN
        036112 011700              .WORD    PKTSSR
6297 036114              30$:      CKLOOP      ;LOOP IF SELECTED
        036114 104406              TRAP      C1CLP1
6298 036116 012703 001000      MOV      #512.,R3      ;RECORD SIZE
6299 036122 013737 003072 040702  MOV      FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
6300
6301      ;*****
6302      ;
6303      ;WRITE DATA,ACK,CVC-1 COMMAND
6304      ;
6305      ;*****
6306      ;
6307 036130 012737 140005 040700      MOV      #140005,T24PK3      ;WRITE DATA,ACK,CVC-1 COMMAND
6308 036136 012704 040700      MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6309 036142
6310 036142 010337 040706      65$:      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
6311 036146 010465 177776      MOV      R4,TSDB(R5)      ;ISSUE COMMAND
6312 036152 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6313 036156 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
6314 036162 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
6315 036166 020102              CMP      R1,R2      ;ARE THEY EQUAL
6316 036170 001406              BEQ      75$      ;BR, IF OK
6317 036172 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6321
6322      ;SOFT ERROR, REALLY CHECKING THE
6323 036176              ERRSOFT  ERRNO,WRTErr,PKTSSR ;READ DATA COMMAND
        036176 104457              ;TSSR INCORRECT AFTER WRITE DATA
        036200 000703              TRAP      C1ERRSOFT
        036202 005011              .WORD    451
        036204 011700              .WORD    WRTErr
        036206              .WORD    PKTSSR
6324 036206              75$:      CKLOOP      ;LOOP IF SELECTED
        036206 104406              TRAP      C1CLP1
6325 036210 012703 000400      MOV      #256.,R3      ;SIZE OF RECORD
6326 036214 013737 003072 040702  MOV      FREE,T24RB      ;STARTING READ BUFFER ADDRESS
6327
6328      ;*****
6329      ;
6330      ;READ DATA,ACK COMMAND
6331      ;
6332      ;*****
6333      ;
6334 036222 012737 100401 040700      165$:      MOV      #100401,T24PK3      ;READ DATA,ACK COMMAND
6335 036230 012704 040700      MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6336 036234 010337 040706      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET

```

CZIKFA TK-25 FRT END FUNC #2  
TEST 4: BASIC READ DATA (FOR

MACRO M1200 20-APR-84 08:12 PAGE 109-2

TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

|      |        |        |        |               |        |                     |                                        |       |          |
|------|--------|--------|--------|---------------|--------|---------------------|----------------------------------------|-------|----------|
| 6337 | 036240 | 010465 | 177776 |               | MOV    | R4,TSD8(R5)         | ;ISSUE COMMAND                         |       |          |
| 6338 | 036244 | 004737 | 017120 |               | JSR    | PC,WAITF            | ;WAIT FOR SSR TO SET                   |       |          |
| 6339 | 036250 | 016501 | 000000 |               | MOV    | TSSR(R5),R1         | ;GET TSSR CONTENTS                     |       |          |
| 6340 | 036254 | 012702 | 100204 |               | MOV    | #SSR!SC!BIT2,R2     | ;SET UP EXPECTED                       |       |          |
| 6341 | 036260 | 020102 |        |               | CMP    | R1,R2               | ;ARE THEY EQUAL                        |       |          |
| 6342 | 036262 | 001406 |        |               | BEQ    | 170\$               | ;BR, IF OK                             |       |          |
| 6343 | 036264 | 004737 | 020102 |               | JSR    | PC,FATCHK           | ;INC AND CHECK FOR MORE THAN 25 ERRORS |       |          |
| 6347 | 036270 |        |        |               | ERRHRD | ERRNO,T24TRL,EXPREC | ;TSSR INCORRECT AFTER READ DATA        |       |          |
|      | 036270 | 104456 |        |               |        |                     |                                        | TRAP  | C\$ERHRD |
|      | 036272 | 000704 |        |               |        |                     |                                        | .WORD | 452      |
|      | 036274 | 043104 |        |               |        |                     |                                        | .WORD | T24TRL   |
|      | 036276 | 016344 |        |               |        |                     |                                        | .WORD | EXPREC   |
| 6348 | 036300 |        |        | 170\$: CKLOOP |        |                     | ;LOOP IF SELECTED                      |       |          |
|      | 036300 | 104406 |        |               |        |                     |                                        | TRAP  | C\$CLP1  |
| 6349 |        |        |        |               |        |                     |                                        |       |          |
| 6350 |        |        |        |               |        |                     |                                        |       |          |
| 6351 |        |        |        |               |        |                     |                                        |       |          |
| 6352 |        |        |        |               |        |                     |                                        |       |          |
| 6353 |        |        |        |               |        |                     |                                        |       |          |
| 6354 |        |        |        |               |        |                     |                                        |       |          |
| 6355 |        |        |        |               |        |                     |                                        |       |          |
| 6356 | 036302 | 013701 | 040576 |               | MOV    | T24BFR+6,R1         | ;GET MESSAGE BUFFER (XSTO)             |       |          |
| 6357 | 036306 | 010102 |        |               | MOV    | R1,R2               | ;SET UP EXPECTED                       |       |          |
| 6358 | 036310 | 052702 | 010000 |               | BIS    | #BIT12,R2           | ;SET THE RLL BIT IN EXPECTED           |       |          |
| 6359 | 036314 | 020102 |        |               | CMP    | R1,R2               | ;ARE THEY EQUAL                        |       |          |
| 6360 | 036316 | 001406 |        |               | BEQ    | 180\$               | ;BR, IF EQUAL (ALL IS WELL)            |       |          |
| 6361 | 036320 | 004737 | 020102 |               | JSR    | PC,FATCHK           | ;INC AND CHECK FOR MORE THAN 25 ERRORS |       |          |
| 6365 | 036324 |        |        |               | ERRHRD | ERRNO,T24LON,EXPREC | ;THE RLL BIT WAS NOT SET IN XSTO       |       |          |
|      | 036324 | 104456 |        |               |        |                     |                                        | TRAP  | C\$ERHRD |
|      | 036326 | 000705 |        |               |        |                     |                                        | .WORD | 453      |
|      | 036330 | 042652 |        |               |        |                     |                                        | .WORD | T24LON   |
|      | 036332 | 016344 |        |               |        |                     |                                        | .WORD | EXPREC   |
| 6366 | 036334 |        |        | 180\$:        |        |                     |                                        |       |          |
| 6367 | 036334 |        |        | ENDSUB        |        |                     | ;>>>>>>>> END SUBTEST >>>>>>>>         |       |          |
|      | 036334 |        |        |               |        |                     | L10056:                                |       |          |
|      | 036334 | 104403 |        |               |        |                     |                                        | TRAP  | C\$ESUB  |
| 6368 | 036336 | 023727 | 002170 | 000031        | CMP    | FATELG,#25,         | ;IS ERROR COUNT AT 25                  |       |          |
| 6369 | 036344 | 002402 |        |               | BLT    | 999\$               | ;BR, IF LESS THAN 25                   |       |          |
| 6370 | 036346 | 004737 | 020154 |               | JSR    | PC,CKDROP           | ;TRY TO DROP THE UNIT                  |       |          |
| 6371 | 036352 |        |        | 999\$:        |        |                     |                                        |       |          |
| 6372 |        |        |        | IT            |        |                     |                                        |       |          |



|      |        |        |        |        |                           |                                         |       |          |
|------|--------|--------|--------|--------|---------------------------|-----------------------------------------|-------|----------|
|      | 036460 | 004754 |        |        |                           |                                         | .WORD | WRMSG    |
|      | 036462 | 011666 |        |        |                           |                                         | .WORD | SFIMSG   |
| 6429 | 036464 |        |        | 241:   | CKLOOP                    | I LOOP IF SELECTED                      |       |          |
|      | 036464 | 104406 |        |        |                           |                                         | TRAP  | CICLP1   |
| 6430 |        |        |        |        |                           |                                         |       |          |
| 6431 |        |        |        |        |                           |                                         |       |          |
| 6432 |        |        |        |        |                           |                                         |       |          |
| 6433 |        |        |        |        |                           |                                         |       |          |
| 6434 |        |        |        |        |                           |                                         |       |          |
| 6435 |        |        |        |        |                           |                                         |       |          |
| 6436 |        |        |        |        |                           |                                         |       |          |
| 6437 | 036466 | 004737 | 010434 |        | JSR PC,REWIND             | I CALL TAPE REWIND COMMAND              |       |          |
| 6438 | 036472 | 004737 | 017236 |        | JSR PC,CHKTSSR            | I SEE HOW TSSR IS                       |       |          |
| 6439 | 036476 | 103407 |        |        | BCS 301                   | I BR, IF NO PROBLEM                     |       |          |
| 6440 | 036500 | 010001 |        |        | MOV R0,R1                 | I SAVE TSSR                             |       |          |
| 6441 | 036502 | 004737 | 020102 |        | JSR PC,FATCHK             | I INC AND CHECK FOR MORE THAN 25 ERRORS |       |          |
| 6445 | 036506 |        |        |        | ERRHRD ERNO,T24RWN,PKTSSR | I REWIND NOT ACCEPTED                   |       |          |
|      | 036506 | 104456 |        |        |                           |                                         | TRAP  | C1ERHRD  |
|      | 036510 | 000710 |        |        |                           |                                         | .WORD | 456      |
|      | 036512 | 042036 |        |        |                           |                                         | .WORD | T24RWN   |
|      | 036514 | 011700 |        |        |                           |                                         | .WORD | PKTSSR   |
| 6446 | 036516 |        |        | 301:   | CKLOOP                    | I LOOP IF SELECTED                      |       |          |
|      | 036516 | 104406 |        |        |                           |                                         | TRAP  | CICLP1   |
| 6447 | 036520 | 012703 | 000400 |        | MOV #256.,R3              | I RECORD SIZE                           |       |          |
| 6448 | 036524 | 013737 | 003072 | 040702 | MOV FREF,T24RB            | I STARTING WRITE BUFFER ADDRESS         |       |          |
| 6449 |        |        |        |        |                           |                                         |       |          |
| 6450 |        |        |        |        |                           |                                         |       |          |
| 6451 |        |        |        |        |                           |                                         |       |          |
| 6452 |        |        |        |        |                           |                                         |       |          |
| 6453 |        |        |        |        |                           |                                         |       |          |
| 6454 |        |        |        |        |                           |                                         |       |          |
| 6455 |        |        |        |        |                           |                                         |       |          |
| 6456 | 036532 | 012737 | 140005 | 040700 | MOV #140005,T24PK3        | I WRITE DATA,ACK,CVC=1 COMMAND          |       |          |
| 6457 | 036540 | 012704 | 040700 |        | MOV #T24PK3,R4            | I SET UP R4 WITH PACKET ADDRESS         |       |          |
| 6458 | 036544 |        |        | 651:   |                           |                                         |       |          |
| 6459 | 036544 | 010337 | 040706 |        | MOV R3,T24SZ              | I SET UP RECORD SIZE IN PACKET          |       |          |
| 6460 | 036550 | 010465 | 177776 |        | MOV R4,TSD8(R5)           | I ISSUE COMMAND                         |       |          |
| 6461 | 036554 | 004737 | 017120 |        | JSR PC,WAITF              | I WAIT FOR SSR TO SET                   |       |          |
| 6462 | 036560 | 016501 | 000000 |        | MOV TSSR(R5),R1           | I GET TSSR CONTENTS                     |       |          |
| 6463 | 036564 | 012702 | 000200 |        | MOV #SSR,R2               | I SET UP EXPECTED                       |       |          |
| 6464 | 036570 | 020102 |        |        | CMP R1,R2                 | I ARE THEY EQUAL                        |       |          |
| 6465 | 036572 | 001406 |        |        | BEQ 751                   | I BR, IF OK                             |       |          |
| 6466 | 036574 | 004737 | 020102 |        | JSR PC,FATCHK             | I INC AND CHECK FOR MORE THAN 25 ERRORS |       |          |
| 6470 | 036600 |        |        |        | ERRSOFT ERNO,WTERR,PKTSSR | I TSSR INCORRECT AFTER WRITE DATA       |       |          |
|      | 036600 | 104457 |        |        |                           |                                         | TRAP  | C1ERSOFT |
|      | 036602 | 000711 |        |        |                           |                                         | .WORD | 457      |
|      | 036604 | 005011 |        |        |                           |                                         | .WORD | WTERR    |
|      | 036606 | 011700 |        |        |                           |                                         | .WORD | PKTSSR   |
| 6471 | 036610 |        |        | 751:   | CKLOOP                    | I LOOP IF SELECTED                      |       |          |
|      | 036610 | 104406 |        |        |                           |                                         | TRAP  | CICLP1   |
| 6472 | 036612 | 012703 | 000400 |        | MOV #256.,R3              | I RECORD SIZE                           |       |          |
| 6473 | 036616 | 012701 | 160000 |        | MOV #160000,R1            | I START POSSIBLE NXM ADDRESS            |       |          |
| 6474 | 036622 | 012702 | 177776 |        | MOV #177776,R2            | I END POSSIBLE NXM ADDRESS              |       |          |
| 6475 | 036626 | 004737 | 017276 |        | JSR PC,NXNM               | I CALL NXM FINDER ROUTINE               |       |          |
| 6476 | 036632 | 103402 |        |        | BCS 761                   | I BR IF NXM ADDRESS FOUND               |       |          |
| 6477 | 036634 | 000137 | 037010 |        | JMP 1801                  | I JUMP OVER CAN'T FIND NXM              |       |          |
| 6478 | 036640 | 010137 | 040702 | 761:   | MOV R1,T24RB              | I STARTING READ REVERSE BUFFER ADDRESS  |       |          |

C14

CZIKFA TK 25 ERI END UNC #2 MACRO M1200 20-APR-84 08:12 PAGE 110-2  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 171

```

6479
6480
6481
6482
6483
6484
6485
6486 036644 013737 040726 040704      MOV      T24DLY,T24RB.2      ;GET BITS 16 AND 17
6487 036652 012737 100401 040700      MOV      #100401,T24PK3    ;READ REVERSE DATA,ACK COMMAND
6488 036660 012704 040700      MOV      #T24PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
6489 036664 012737 000400 040706      MOV      #256.,T24SZ      ;SET UP RECORD SIZE IN PACKET
6490 036672 010465 177776      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
6491 036676 004737 017120      JSR      PC,WAITF         ;WAIT FOR SSR TO SET
6492 036702 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
6493 036706 012702 104210      MOV      #SSRINXHI$C!BIT5,R2 ;SET UP EXPECTED
6494 036712 020102      CMP      R1,R2           ;ARE THEY EQUAL
6495 036714 001417      BEQ      1701            ;BR, IF OK
6496 036716 062737 000001 040726      ADD      #1,T24DLY        ;NEXT BUNCH OF MEMORY
6497 036724 022737 000100 040726      CMP      #100,T24DLY      ;TOO MUCH MEMORY
6498 036732 001402      BEQ      1681            ;BR IF OVER
6499 036734 000137 036374      JMP      101             ;TRY AGAIN
6500 036740 004737 020102      JSR      PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
6504 036744      ERRHRD  ERRNO,T24NXH,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C1ERRHRD
                                .WORD     458
                                .WORD     T24NXH
                                .WORD     PKTSSR
                                TRAP      C1CLP1
036744 104456
036746 000712
036750 041141
036752 011700
6505 036754      CKLOOP      ;LOOP IF SELECTED
036754 104406
6506
6507
6508
6509
6510
6511
6512
6513 036756 013701 040576      MOV      T24BFR.6,R1      ;GET MESSAGE BUFFER
6514 036762 010102      MOV      R1,R2           ;SET UP EXPECTED
6515 036764 052702 040000      BIS      #BIT14,R2        ;SET THE RLS BIT IN EXPECTED
6516 036770 020102      CMP      R1,R2           ;ARE THEY EQUAL
6517 036772 001406      BEQ      1801            ;BR, IF EQUAL (ALL IS WELL)
6518 036774 004737 020102      JSR      PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
6522 037000      ERRHRD  ERRNO,T24LOP,EXPREC ;THE RLL BIT WAS NOT SET IN XSTO
                                TRAP      C1ERRHRD
                                .WORD     459
                                .WORD     T24LOP
                                .WORD     EXPREC
037000 104456
037002 000713
037004 042734
037006 016344
6523 037010
6524 037010      END SUB      ;END SUBTEST
037010
037010 104403
6525 037012 023727 002170 000031      CMP      FATELG,#25,      ;IS ERROR COUNT AT 25
6526 037020 002402      BLT      9991            ;BR, IF LESS THAN 25
6527 037022 004737 020154      JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
6528 037026      9991;

```



E14

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 111-1  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 173

```

6582 ;*****
6583 ;
6584 ;LEGAL MODE,ACK,CVC=1,READ COMMAND
6585 ;
6586 ;*****
6587
6588 037136 012737 140001 040700      MOV      #140001,T24PK3      ;LEGAL MODE,ACK,CVC=1,READ COMMAND
6589 037144 012704 040700      MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6590 037150 012737 000400 040706      MOV      #256.,T24SZ      ;SET UP RECORD SIZE IN PACKET
6591 037156 010465 177776      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
6592 037162 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR:BIT1!BIT2 TO SET
6593 037166 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
6594 037172 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2      ;SET UP EXPECTED
6595 037176 020102      CMP      R1,R2      ;ARE THEY EQUAL
6596 037200 001406      BEQ      75$      ;BR, IF OK
6597 037202 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6601 037206      ERRHRD      ERRNO,T24WDG,PKTSSR      ;TSSR INCORRECT AFTER READ DATA
        037206 104456      TRAP      C!ERHRD
        037210 000716      .WORD      462
        037212 041052      .WORD      T24WDG
        037214 011700      .WORD      PKTSSR
6602 037216      75$:      CKLOOP      ;LOOP IF SELECTED      TRAP      C!CLP1
        037216 104406
6603 ;*****
6604 ;
6605 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
6606 ;
6607 ;*****
6608 ;
6609
6610 037220 013701 040576      MOV      T24BFR+6,R1      ;GET MESSAGE BUFFER
6611 037224 010102      MOV      R1,R2      ;SET UP EXPECTED
6612 037226 052702 000400      BIS      #BIT8,R2      ;SET THE ILA BIT IN EXPECTED
6613 037232 020102      CMP      R1,R2      ;ARE THEY EQUAL
6614 037234 001406      BEQ      180$      ;BR, IF EQUAL (ALL IS WELL)
6615 037236 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6619 037242      ERRHRD      ERRNO,T24ILA,EXPREC      ;THE ILA BIT WAS NOT SET IN XSTO
        037242 104456      TRAP      C!ERHRD
        037244 000717      .WORD      463
        037246 041302      .WORD      T24ILA
        037250 016344      .WORD      EXPREC
6620 037252      180$:      ENDSUB      ;***** END SUBTEST *****
6621 037252      L10060:      TRAP      C!ESUB
        037252 104403
6622 037254 023727 002170 000031      CMP      FATELG,#25.      ;IS ERROR COUNT AT 25
6623 037262 002402      BLT      999$      ;BR, IF LESS THAN 25
6624 037264 004737 020154      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
6625 037270      999$:

```



G14

CZTKFA TK-25 FRT END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 112-1  
 TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 175

|        |        |        |        |        |        |                     |                                        |
|--------|--------|--------|--------|--------|--------|---------------------|----------------------------------------|
| 037372 | 104456 |        |        |        |        | TRAP                | C\$ERHRD                               |
| 037374 | 000721 |        |        |        |        | .WORD               | 465                                    |
| 037376 | 004754 |        |        |        |        | .WORD               | WRTMSG                                 |
| 037400 | 011666 |        |        |        |        | .WORD               | SFIMSG                                 |
| 6684   | 037402 |        |        | 24\$:  | CKLOOP |                     | ;LOOP IF SELECTED                      |
|        | 037402 | 104406 |        |        |        | TRAP                | C\$CLP1                                |
| 6685   | 037404 | 012701 | 160000 |        | MOV    | #160000,R1          | ;START POSSIBLE NX' ADDRESS            |
| 6686   | 037410 | 012702 | 177776 |        | MOV    | #177776,R2          | ;END POSSIBLE NXM ADDRESS              |
| 6687   | 037414 | 004737 | 017276 |        | JSR    | PC,XNXM             | ;CALL NXM FINDER ROUTINE               |
| 6688   | 037420 | 103402 |        |        | BCS    | 76\$                | ;BR IF WE FOUND NXM                    |
| 6689   | 037422 | 000137 | 037544 |        | JMP    | 180\$               | ;JUMP OVER, CAN'T FIND NXM             |
| 6690   | 037426 | 010137 | 040702 |        | 76\$:  | MOV                 | R1,T249B                               |
| 6691   |        |        |        |        |        |                     | ;STARTING READ BUFFER ADDRESS          |
| 6692   |        |        |        |        |        |                     | *****                                  |
| 6693   |        |        |        |        |        |                     | ;READ, ACK, CVC-1, COMMAND             |
| 6694   |        |        |        |        |        |                     | *****                                  |
| 6695   |        |        |        |        |        |                     |                                        |
| 6696   |        |        |        |        |        |                     |                                        |
| 6697   | 037432 | 013737 | 040726 | 040704 | MOV    | T24DLY,T24RB+2      | ;MEMORY BITS 16 AND 17                 |
| 6698   | 037440 | 012737 | 140001 | 040700 | MOV    | #140001,T24PK3      | ;READ, ACK, CVC-1, COMMAND             |
| 6699   | 037446 | 012704 | 040700 |        | MOV    | #T24PK3,R4          | ;SET UP R4 WITH PACKET ADDRESS         |
| 6700   | 037452 | 012737 | 000400 | 040706 | MOV    | #256.,T24SZ         | ;SET UP RECORD SIZE IN PACKET          |
| 6701   | 037460 | 010465 | 177776 |        | MOV    | R4,TSD8(R5)         | ;ISSUE COMMAND                         |
| 6702   | 037464 | 004737 | 017120 |        | JSR    | FJ,WAITE            | ;WAIT FOR SSR!BIT1!BIT2 TO SET         |
| 6703   | 037470 | 016501 | 000000 |        | MOV    | TSSR(R5),R1         | ;GET TSSR CONTENTS                     |
| 6704   | 037474 | 012702 | 104210 |        | MOV    | #SSR!SC!NXM!BIT3,R2 | ;SET UP EXPECTED                       |
| 6705   | 037500 | 020102 |        |        | CMP    | R1,R2               | ;ARE THEY EQUAL                        |
| 6706   | 037502 | 001417 |        |        | BEQ    | 75\$                | ;BR, IF OK                             |
| 6707   | 037504 | 062737 | 000001 | 040726 | ADD    | #1,T24DLY           | ;NEXT BUNCH OF MEMORY                  |
| 6708   | 037512 | 022737 | 000100 | 040726 | CMP    | #100,T24DLY         | ;TOO MUCH MEMORY                       |
| 6709   | 037520 | 001402 |        |        | BEQ    | 50\$                | ;END OF LINE                           |
| 6710   | 037522 | 000137 | 037312 |        | JMP    | 10\$                | ;TRY AGAIN                             |
| 6711   | 037526 | 004737 | 020102 |        | 50\$:  | JSR                 | PC,FATCHK                              |
| 6715   | 037532 |        |        |        | ERRHRD | ERRNO,T24NXM,PKTSSR | ;INC AND CHECK FOR MORE THAN 25 ERRORS |
|        | 037532 | 104456 |        |        |        |                     | ;TSSR INCORRECT AFTER READ DATA        |
|        | 037534 | 000722 |        |        |        |                     | TRAP                                   |
|        | 037536 | 041141 |        |        |        |                     | .WORD                                  |
|        | 037540 | 011700 |        |        |        |                     | .WORD                                  |
| 6716   | 037542 |        |        |        |        |                     | .WORD                                  |
|        | 037542 | 104406 |        |        | 75\$:  | CKLOOP              | ;LOOP IF SELECTED                      |
| 6717   | 037544 |        |        |        |        |                     | TRAP                                   |
| 6718   | 037544 |        |        |        | 80\$:  |                     | C\$CLP1                                |
|        | 037544 |        |        |        | 180\$: | ENDSUB              |                                        |
|        | 037544 | 104403 |        |        |        |                     | ;***** END SUBTEST *****               |
| 6719   | 037546 | 023727 | 002170 | 000031 |        |                     | L10061:                                |
| 6720   | 037554 | 002402 |        |        | CMP    | FATFLG,#25,         | TRAP                                   |
| 6721   | 037556 | 004737 | 020154 |        | BLT    | 999\$               | C\$ESUB                                |
| 6722   | 037562 |        |        |        | JSR    | PC,CKDROP           | ;IS ERROR COUNT AT 25                  |
|        |        |        |        |        |        |                     | ;BR, IF LESS THAN 25                   |
|        |        |        |        |        |        |                     | ;TRY TO DROP THE UNIT                  |
|        |        |        |        |        | 999\$: |                     |                                        |



```

6776      ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6777      ;
6778      ;*****
6779
6780 037656 004737 010434      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
6781 037662 004737 017236      JSR      PC,CHKTSSR      ;SEE HOW TSSR IS
6782 037666 103407              BCS      30$      ;BR, IF NO PROBLEM
6783 037670 010001              MOV      R0,R1      ;SAVE TSSR
6784 037672 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6788 037676              ERRHRD  ERRNO,T24RWN,PKTSSR ;REWIND NOT ACCEPTED
              TRAP      C1ERRHRD
              .WORD      469
              .WORD      T24RWN
              .WORD      PKTSSR
6789 037706              30$:    CKLOOP      ;LOOP IF SELECTED
              TRAP      C1CLP1
              .WORD      469
6790 037710 012703 000400      MOV      #256.,R3      ;RECORD SIZE
6791 037714 013737 003072 040702  MOV      FREE,T24RB      ;STARTING WRITE BUFFER ADDRESS
6792
6793      ;*****
6794      ;
6795      ;READ REVERSE DATA,ACK COMMAND
6796      ;
6797      ;*****
6798
6799 037722 012737 100401 040700      MOV      #100401,T24PK3      ;READ REVERSE DATA,ACK COMMAND
6800 037730 012704 040700      MOV      #T24PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
6801 037734
6802 037734 010337 040706      65$:    MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
6803 037740 010465 177776      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
6804 037744 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6805 037750 016501 000000      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
6806 037754 012702 100206      MOV      #SSR!SC!BIT1!BIT2,R2 ;SET UP EXPECTED
6807 037760 020102              CMP      R1,R2      ;ARE THEY EQUAL
6808 037762 001406              BEQ      75$      ;BR, IF OK
6809 037764 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6813 037770              ERRHRD  ERRNO,T24WDE,PKTSSR ;TSSR INCORRECT AFTER READ DATA
              TRAP      C1ERRHRD
              .WORD      470
              .WORD      T24WDE
              .WORD      PKTSSR
6814 040000              75$:    CKLOOP      ;LOOP IF SELECTED
              TRAP      C1CLP1
              .WORD      470
              .WORD      104406
6815
6816      ;*****
6817      ;
6818      ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6819      ;
6820      ;*****
6821
6822 040002 013701 040576      MOV      T24BFR+6,R1      ;GET MESSAGE BUFFER
6823 040006 010102              MOV      R1,R2      ;SET UP EXPECTED
6824 040010 052702 002000      BIS      #BIT10,R2      ;SET THE NEF BIT IN EXPECTED
6825 040014 020102              CMP      R1,R2      ;ARE THEY EQUAL
6826 040016 001406              BEQ      180$      ;BR, IF EQUAL (ALL IS WELL)
6827 040020 004737 020102      JSR      PC,FATCHK      ;INC AND CHECK FOR MORE THAN 25 ERRORS
6831 040024              ERRHRD  ERRNO,T24NEF,EXPREC ;THE RLL BIT WAS NOT SET IN XST0

```

J14

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 113-2  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 178

```
040024 104456
040026 0C0727
040030 040730
040032 016344
6832 040034 180$:
6833 040034 ENDSUB
040034
040034 104403
6834 040036 023727 002170 000031 CMP FATFLG, #25.
6835 040044 002402 BLT 999$
6836 040046 004737 020154 JSR PC, CKDROP
6837 040052 999$:
```

```
TRAP C$ERHND
.WORD 471
.WORD T24NEF
.WORD EXPREC
```

```
; >>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
L10062:
```

```
TRAP C$ESUB
; IS ERROR COUNT AT 25
; BR, IF LESS THAN 25
; TRY TO DROP THE UNIT
```



```

6891                                     ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6892                                     ;
6893                                     ;*****
6894                                     ;
6895 040146 004737 010434                JSR      PC,REWIND                ;CALL TAPE REWIND COMMAND
6896 040152 004737 017236                JSR      PC,CHKTSSR                ;SEE HOW TSSR IS
6897 040156 103407                        BCS      30$                    ;BR, IF NO PROBLEM
6898 040160 010001                        MOV      R0,R1                    ;SAVE TSSR
6899 040162 004737 020102                JSR      PC,FATCHK                ;INC AND CHECK FOR MORE THAN 25 ERRORS
6903 040166                                ERRHRD  ERRNO,T24RWN,PKTSSR        ;REWIND NOT ACCEPTED
                                040166 104456                                TRAP      C$ERRHRD
                                040170 000732                                .WORD    474
                                040172 042036                                .WORD    T24RWN
                                040174 011700                                .WORD    PKTSSR
6904 040176                                30$:  CKLOOP                        ;LOOP IF SELECTED
                                040176 104406                                TRAP      C$CLP1
6905 040200 012703 000400                MOV      0256.,R3                ;RECORD SIZE
6906 040204 013737 003072 040702        MOV      FREE,T24RB            ;STARTING WRITE BUFFER ADDRESS
6907
6908                                     ;*****
6909                                     ;
6910                                     ;WRITE DATA,ACK,CVC=1 COMMAND
6911                                     ;
6912                                     ;*****
6913
6914 040212 012737 140005 040700        MOV      0140005,T24PK3        ;WRITE DATA,ACK,CVC=1 COMMAND
6915 040220 012704 040700                MOV      0T24PK3,R4            ;SET UP R4 WITH PACKET ADDRESS
6916 040224                                65$:
6917 040224 010337 040706                MOV      R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
6918 040230 010465 177776                MOV      R4,TSD8(R5)            ;ISSUE COMMAND
6919 040234 004737 017120                JSR      PC,WAITF                ;WAIT FOR SSR TO SET
6920 040240 016501 000000                MOV      TSSR(R5),R1          ;GET TSSR CONTENTS
6921 040244 012702 000200                MOV      0SSR,R2                ;SET UP EXPECTED
6922 040250 020102                        CMP      R1,R2                ;ARE THEY EQUAL
6923 040252 001406                        BEQ      75$                    ;BR, IF OK
6924 040254 004737 020102                JSR      PC,FATCHK                ;INC AND CHECK FOR MORE THAN 25 ERRORS
6928                                     ;SOFT ERROR, REALLY CHECKING THE
6929                                     ;READ REVERSE DATA COMMAND
6930                                ERRSOFT ERRNO,WRERR,PKTSSR        ;TSSR INCORRECT AFTER READ DATA
                                040260 104457                                TRAP      C$ERRSOFT
                                040262 000733                                .WORD    475
                                040264 005011                                .WORD    WRERR
                                040266 011700                                .WORD    PKTSSR
6931 040270                                75$:  CKLOOP                        ;LOOP IF SELECTED
                                040270 104406                                TRAP      C$CLP1
6932 040272 012703 000400                MOV      0256.,R3                ;RECORD SIZE
6933 040276 013737 003072 040702        MOV      FREE,T24RB            ;STARTING READ BUFFER ADDRESS
6934
6935                                     ;*****
6936                                     ;
6937                                     ;READ REVERSE DATA,ACK COMMAND
6938                                     ;
6939                                     ;*****
6940
6941 040304 012737 100401 040700        MOV      0100401,T24PK3        ;READ REVERSE DATA,ACK COMMAND
6942 040312 012704 040700                MOV      0T24PK3,R4            ;SET UP R4 WITH PACKET ADDRESS
6943 040316 010337 040706                MOV      R3,T24SZ                ;SET UP RECORD SIZE IN PACKET

```

M14

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 114-2  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 181

```

6944 040322 010465 177776      MOV      R4,TSD8(R5)      ;ISSUE COMMAND
6945 040326 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6946 040332 016501 000000      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
6947 040336 012702 0C0200      MOV      #SSR,R2      ;SET UP EXPECTED
6948 040342 020102              CMP      R1,R2      ;ARE THEY EQUAL
6949 040344 001406              BEQ      170$      ;BR, IF OK
6950 040346 004737 020102      JSR      PC,FATCHK    ;INC AND CHECK FOR MORE THAN 25 ERRORS
6954 040352              ERRHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERRHRD
                                .WORD    476
                                .WORD    T24TRL
                                .WORD    PKTSSR
                                TRAP      C$CLP1
040352 104456
040354 000734
040356 043104
040360 011700
6955 040362              170$: CKLOOP      ;LOOP IF SELECTED
040362 104406              TRAP      C$CLP1
6956 040364 012703 000400      MOV      #256,R3      ;RECORD SIZE
6957 040370 013737 003072 040702 MOV      FREE,T24RB    ;STARTING READ BUFFER ADDRESS
6958
6959 ;*****
6960 ;
6961 ;READ REVERSE DATA,ACK COMMAND
6962 ;
6963 ;*****
6964
6965 040376 012737 100401 040700      MOV      #100401,T24PK3 ;READ REVERSE DATA,ACK COMMAND
6966 040404 012704 040700      195$: MOV      #T24PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
6967 040410 010337 040706      MOV      R3,T24SZ      ;SET UP RECORD SIZE IN PACKET
6968 040414 010465 177776      MOV      R4,TSD8(R5)    ;ISSUE COMMAND
6969 040420 004737 017120      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
6970 040424 016501 000000      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
6971 040430 012702 100204      MOV      #SSR!SC!BIT2,R2 ;SET UP EXPECTED
6972 040434 020102              CMP      R1,R2      ;ARE THEY EQUAL
6973 040436 001406              BEQ      200$      ;BR, IF OK
6974 040440 004737 020102      JSR      PC,FATCHK    ;INC AND CHECK FOR MORE THAN 25 ERRORS
6978 040444              ERRHRD  ERRNO,T24TRL,PKTSSR ;TSSR INCORRECT AFTER READ DATA
                                TRAP      C$ERRHRD
                                .WORD    477
                                .WORD    T24TRL
                                .WORD    PKTSSR
                                TRAP      C$CLP1
040444 104456
040446 000735
040450 043104
040452 011700
6979 040454              200$: CKLOOP      ;LOOP IF SELECTED
040454 104406              TRAP      C$CLP1
6980 040456 013701 040604      MOV      T24BFR+14,R1   ;GET MESSAGE BUFFER (XST3)
6981 040462 010102              MOV      R1,R2      ;SET UP EXPECTED
6982 040464 052702 000001      BIS      #BIT0,R2      ;SET THE RIB BIT IN EXPECTED
6983 040470 020102              CMP      R1,R2      ;ARE THEY EQUAL
6984 040472 001406              BEQ      210$      ;BR, IF EQUAL (ALL IS WELL)
6985 040474 004737 020102      JSR      PC,FATCHK    ;INC AND CHECK FOR MORE THAN 25 ERRORS
6989 040500              ERRHRD  ERRNO,T24LOR,EXPREC ;THE RIB BIT WAS NOT SET IN XST0
                                TRAP      C$ERRHRD
                                .WORD    478
                                .WORD    T24LOR
                                .WORD    EXPREC
040500 104456
040502 000736
040504 041002
040506 016344
6990 040510              210$: ENDSUB      ;***** END SUBTEST *****
6991 040510
040510
040510 104403
6992 040512 023727 002170 000031      CMP      FATFLG,#25     ;IS ERROR COUNT AT 25
6993 040520 002402              BLT      999$          ;BR, IF LESS THAN 25

```







015

CZIKFA IK-25 LRI END FUNC 02 MACRO M1200 20-APR-84 08:12 PAGE 117-1  
TEST 4: BASIC READ DATA (FORWARD AND REVERSE)

SEQ 185

|                           |         |               |                                    |
|---------------------------|---------|---------------|------------------------------------|
| 7123 043330 000207        | RTS     | PC            | ;RETURN                            |
| 7124                      |         |               |                                    |
| 7125                      |         |               |                                    |
| 7126 043332               | T24RT2: |               |                                    |
| 7127 043332               | SAVREG  |               | ;SAVE THE REGISTER'S               |
| 7128 043336 012701 040660 | MOV     | #T24PK2,R1    | ;START OF THE PACKET               |
| 7129 043342 012721 100206 | MOV     | #100206,(R1)+ | ;WRITE SUBSYSTEM MEM. WITH ACK, IE |
| 7130 043346 012721 040710 | MOV     | #T24BF2,(R1)+ | ;ADDRESS OF DATA BLOCK             |
| 7131 043352 005021        | CLR     | (R1)+         | ;EXTENDED ADDRESS                  |
| 7132 043354 012721 000006 | MOV     | #6,(R1)+      | ;SIZE OF DATA BLOCK IN BYTES       |
| 7133 043360 005021        | CLR     | (R1)+         |                                    |
| 7134 043362 012701 040710 | MOV     | #T24BF2,R1    | ;POINT TO DATA SEL AREA            |
| 7135 043366 005021        | CLR     | (R1)+         |                                    |
| 7136 043370 005011        | CLR     | (R1)          |                                    |
| 7137 043372 000207        | RTS     | PC            | ;RETURN                            |
| 7138 043374               | T24RT3: |               |                                    |
| 7139 043374               | SAVREG  |               | ;SAVE THE REGISTERS                |
| 7140 043400 012701 040700 | MOV     | #T24PK3,R1    | ;START OF THE PACKET               |
| 7141 043404 012721 000000 | MOV     | #0,(R1)+      | ;CLEAR AREA OUT                    |
| 7142 043410 012721 000000 | MOV     | #0,(R1)+      | ;ADDRESS OF DATA BLOCK             |
| 7143 043414 005021        | CLR     | (R1)+         | ;EXTENDED ADDRESS                  |
| 7144 043416 012711 000000 | MOV     | #0,(R1)       | ;SIZE OF DATA BLOCK IN BYTES       |
| 7145 043422 000207        | RTS     | PC            | ;RETURN                            |
| 7146 043424               | ENDTSY  |               |                                    |
| 043424                    |         |               |                                    |
| 043424 104401             |         |               |                                    |
|                           |         |               | L10047: TRAP C#ETST                |

7148  
7149  
7150  
7151  
7152  
7153  
7154  
7155  
7156  
7157  
7158  
7159  
7160  
7161  
7162  
7163  
7164  
7165  
7166  
7167  
7168  
7169  
7170  
7171  
7172  
7173  
7174  
7175  
7176  
7177  
7178  
7179  
7180  
7181  
7182  
7183  
7184  
7185  
7186  
7187  
7188  
7189  
7190  
7191  
7192  
7193  
7194  
7195  
7196  
7197  
7198  
7199  
7200  
7201  
7202  
7203  
7204

```
.SBTTL TEST 5: MANUAL INTERVENTION

THIS TEST MUST BE STARTED AS FOLLOWS:
AT THE DIAGNOSTIC SUPERVISOR PROMPT "DR>"
TYPE "START/FLAG:PNT/TEST:5/PASS:1"

-----
THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
ARE BY TYPING <CTRL-C> OR SELECTING CODE 4.
SELECTION CODES AND SUBROUTINES ARE:

CODE    ROUTINE
0        HELP, PRINTS THIS MENU.
1        REWIND AND UNLOAD COMMAND TEST
2        WRITE-PROTECT TEST
3        FRONT PANEL ON-LINE SWITCH TEST
4        EXIT (RETURN TO SUPERVISOR)

EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:

SELECTION 0 - PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.

SELECTION 1 - THIS ROUTINE INSTRUCTS THE OPERATOR TO PLACE THE DRIVE ON-LINE
AND AT OR BEYOND BOT. THE TEST WILL THEN ISSUE THE REWIND AND
UNLOAD COMMAND. IT WILL ALSO TELL THE OPERATOR IF THE DRIVE
ENDED UP ON-LINE OR OFF-LINE.

SELECTION 2 - THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
WRITE-PROTECTED. IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
AN ERROR IS REPORTED.

SELECTION 3 - THIS TEST CHECKS THAT THE PROGRAM CAN READ THE SENSE
OF THE FRONT PANEL "ON-LINE" BUTTON/LIGHT. THE PROGRAM CHECKS THE
```

```

7205      ; STATE OF THE DRIVE (ON-LINE OR OFF-LINE) AND PRINTS A MESSAGE TO
7206      ; NOTIFY THE OPERATOR I.E. "DRIVE IS NOW OFF-LINE" OR "DRIVE IS NOW ON-LINE".
7207      ;
7208      ;
7209      ; SELECTION 4 - THIS WILL RETURN THE PROGRAM TO THE DIAGNOSTIC
7210      ; SUPERVISOR PROMPT, NOTE: IF THE OPERATOR FAILED TO SELECT A PASS
7211      ; COUNT OF ONE, THE PROGRAM WILL LOOP UNTIL STOPPED WITH A CONTROL C.
7212      ;
7213      ;
7214      ;
7215      ;
7216      043426      BGNSTST
7217      043426      005037      002170      CLR      FATFLG      ; CLEAR FATAL ERROR FLAG
7218      043432      005037      003100      CLR      KI1FLG      ; HOLD OFF KI11
7223      043436      005737      002162      TST      TSTCNT      ; IS THIS THE FIRST TEST
7224      043442      001403      BEQ      214      ; BR, IF FIRST TEST
7225      043444      012700      045400      MOV      #T38NE,R0      ; "TEST NOT EXECUTED"
7226      043450      000402      BR      34      ; JUMP IF NOT FIRST TEST
7227      043452      214:      MOV      #T38ID,R0      ; TEST ID MESSAGE
7228      043452      012700      046466      34:      JSR      PC,TSTSETUP      ; DO THE COMMON SETUP
7229      043456      004737      017410      JSR      PC,CHKMAN      ; IS MANUAL INTERVENTION ALLOWED?
7230      043462      004737      021404      BCS      194      ; BR, IF MANUAL INTER ALLOWED
7231      043466      103402      JMP      644      ; JUMP IF NOT ALLOWED
7232      043470      000137      044600      CMP      #1,TSTCNT      ; CHECK MIGHT HAVE TO LEAVE
7233      043474      022737      000001      002162      194:      BEQ      224      ; BR, IF YOU DON'T HAVE TO
7234      043502      001402      JMP      644      ; WASN'T FIRST TEST IN SEQUENCE
7235      043504      000137      044600      224:      CLR      FATFLG      ; CLEAR THE FATAL ERROR FLAG
7236      043510      005037      002170      24:      MOV      #65000.,T38DLY      ; SET UP DELAY COUNTER
7240      043510      005037      002170      044612      54:      JSR      PC,SOFINIT      ; DO A SOFT INIT
7241      043514      012737      176750      BCS      234      ; BRANCH IF OK
7242      043522      004737      016644      MOV      R0,R1      ; CONTENTS OF TSSR REGISTER
7243      043526      103427      BIT      #SSR,R1      ; CHECK FOR YSSR SET
7244      043530      010001      BNE      234      ; KEEP GOING IF NOT SET
7245      043532      032701      000200      DELAY      250      ; CALL DELAY ROUTINE
7246      043536      001023      MOV      #250,(PC)+      ;
7247      043540      012727      000250      .WORD      0      ;
7248      043544      000000      MOV      L#DLY,(PC)+      ;
7249      043546      013727      002116      .WORD      0      ;
7250      043552      000000      DEC      -6(PC)      ;
7251      043554      005367      177772      BNE      -4      ;
7252      043560      001375      DEC      -22(PC)      ;
7253      043562      005367      177756      BNE      -20      ;
7254      043566      001367      DEC      T38DLY      ; BUMP COUNTER DOWN
7255      043570      005337      044612      BNE      54      ; BR, IF MORE TIME LEFT
7256      043574      001352      ERNDF      ERRNO,SFIERR,SFIMSG      ; REPORT FATAL ERROR
7257      043576      104455      TRAP      C1ERDF      ;
7258      043600      000765      .WORD      501      ;
7259      043602      003550      .WORD      SFIERR      ;
7260      043604      011666      .WORD      SFIMSG      ;
7261      043606      012700      046512      234:      MOV      #MIMENU,R0      ; MENU OF MANUAL INTERVENTIONS
7262      043612      012701      000004      MOV      #4,R1      ; MAXIMUM ALLOWED SELECTION
7263      043616      004737      021162      JSR      PC,GETSEL      ; GO GET THE OPERATORS SELECTION
7264      043622      010004      MOV      R0,R4      ; GET NUMBER FROM ROUTINE
7265      043624      006304      ASL      R4      ; CONVERT TO WORD OFFSET

```

G15

C2TKFA TK-25 FRT END FUNC 32  
TEST 5: MANUAL INTERVENTION

MACRO M1200 20-APR-84 08:12 PAGE 118-2

SEQ 188

|      |        |        |               |       |        |                       |       |                                   |
|------|--------|--------|---------------|-------|--------|-----------------------|-------|-----------------------------------|
| 7256 | 043626 | 000174 | 043632        |       | JMP    | 86\$(R4)              |       | JUMP TO PROPER LOOP               |
| 7257 | 043632 | 043510 |               | 61:   | .WORD  | 21                    |       | RETYPE THE MENU                   |
| 7258 | 043634 | 043644 |               |       | .WORD  | 201                   |       | 1 REWIND AND UNLOAD COMMAND TEST  |
| 7259 | 043636 | 044176 |               |       | .WORD  | 251                   |       | 2 WRITE PROTECT                   |
| 7260 | 043640 | 044456 |               |       | .WORD  | 5001                  |       | 3 FRONT PANEL ON-LINE SWITCH TEST |
| 7261 | 043642 | 044600 |               |       | .WORD  | 631                   |       | 4 LEAVE THE TEST                  |
| 7262 |        |        |               |       |        |                       |       |                                   |
| 7263 |        |        |               |       |        |                       |       |                                   |
| 7264 | 043644 |        |               | 201:  | PRINTF | @T38MS4               |       | TELL'EM WHAT TO DO                |
|      | 043644 | 012746 | 046026        |       |        |                       | MOV   | @T38MS4, -(SP)                    |
|      | 043650 | 012746 | 000001        |       |        |                       | MOV   | @1, -(SP)                         |
|      | 043654 | 010600 |               |       |        |                       | MOV   | SP, R0                            |
|      | 043656 | 104417 |               |       |        |                       | TRAP  | C1PNTF                            |
|      | 043660 | 062706 | 000004        |       |        |                       | ADD   | @4, SP                            |
| 7265 | 043664 | 004737 | 016644        | 2221: | JSR    | PC, SOFINIT           |       | DO SOFT INIT OF CONTROLLER        |
| 7266 | 043670 | 103405 |               |       | BCS    | 3001                  |       | BR IF SOFT INIT = OK              |
| 7270 | 043672 | 010001 |               |       | MOV    | R0, R1                |       | SAVE CONTENTS OF TSSR             |
| 7271 | 043674 |        |               |       | ERRDF  | ERRNO, SFIERR, SFIMSG |       | DEVICE FATAL ERROR DURING INIT    |
|      | 043674 | 104455 |               |       |        |                       | TRAP  | C1ERDF                            |
|      | 043676 | 000766 |               |       |        |                       | .WORD | 502                               |
|      | 043700 | 003550 |               |       |        |                       | .WORD | SFIERR                            |
|      | 043702 | 011666 |               |       |        |                       | .WORD | SFIMSG                            |
| 7272 | 043704 |        |               | 3001: |        |                       |       |                                   |
| 7273 | 043704 | 012704 | 045320        |       | MOV    | @T38PK2, R4           |       | SUBROUTINE NEEDS PACKET ADDRESS   |
| 7274 | 043710 | 004737 | 010332        |       | JSR    | PC, WRTCHR            |       | ISSUE WRITE CHARACTERISTICS       |
| 7275 | 043714 | 103405 |               |       | BCS    | 3101                  |       | BR, IF COMMAND ISSUED OK          |
| 7279 | 043716 | 010001 |               |       | MOV    | R0, R1                |       | SAVE CONTENTS OF TSSR             |
| 7280 | 043720 |        |               |       | ERRHRD | ERRNO, WRTMSG, SFIMSG |       | WRITE CHARACTERISTICS FAILED      |
|      | 043720 | 104456 |               |       |        |                       | TRAP  | C1ERHRD                           |
|      | 043722 | 000767 |               |       |        |                       | .WORD | 503                               |
|      | 043724 | 004754 |               |       |        |                       | .WORD | WRTMSG                            |
|      | 043726 | 011666 |               |       |        |                       | .WORD | SFIMSG                            |
| 7281 | 043730 |        |               | 3101: |        |                       |       |                                   |
| 7282 | 043730 | 012737 | 000031 044612 |       | MOV    | @25, T38DLY           |       | SET UP FOR A LONG WAIT            |
| 7283 | 043736 | 036527 | 000000 000100 | 3111: | BIT    | TSSR(R5), @OFL        |       | IS DRIVE OFF-LINE                 |
| 7284 | 043744 | 001431 |               |       | BEQ    | 3151                  |       | BR, IF DRIVE IS ON-LINE           |
| 7285 | 043746 |        |               |       | DELAY  | 250                   |       | DELAY ABOUT .25 SEC               |
|      | 043746 | 012727 | 000250        |       |        |                       | MOV   | @250, (PC)+                       |
|      | 043752 | 000000 |               |       |        |                       | .WORD | 0                                 |
|      | 043754 | 013727 | 002116        |       |        |                       | MOV   | L1DLY, (PC)+                      |
|      | 043760 | 000000 |               |       |        |                       | .WORD | 0                                 |
|      | 043762 | 005367 | 177772        |       |        |                       | DEC   | -6(PC)                            |
|      | 043766 | 001375 |               |       |        |                       | BNE   | -4                                |
|      | 043770 | 005367 | 177756        |       |        |                       | DEC   | -22(PC)                           |
|      | 043774 | 001367 |               |       |        |                       | BNE   | -20                               |
| 7286 | 043776 | 005337 | 044612        |       | DEC    | T38DLY                |       | BUMP LONG DELAY COUNTER DOWN      |
| 7287 | 044002 | 001355 |               |       | BNE    | 3111                  |       | BR, IF MORE LONG DELAY TO GO      |
| 7288 | 044004 |        |               |       | PRINTF | @T38OFL               |       | "DRIV" IS NOW OFF-LINE"           |
|      | 044004 | 012746 | 045772        |       |        |                       | MOV   | @T38OFL, -(SP)                    |
|      | 044010 | 012746 | 000001        |       |        |                       | MOV   | @1, -(SP)                         |
|      | 044014 | 010600 |               |       |        |                       | MOV   | SP, R0                            |
|      | 044016 | 104417 |               |       |        |                       | TRAP  | C1PNTF                            |
|      | 044020 | 062706 | 000004        |       |        |                       | ADD   | @4, SP                            |
| 7289 | 044024 | 000137 | 043664        |       | JMP    | 2221                  |       | STAY HERE FOREVER, WITH MESSAGE   |
| 7290 | 044030 |        |               | 3151: | PRINTF | @T38MS5               |       | "DRIVE SHOULD NOW REWIND AND GO   |
|      | 044030 | 012746 | 046105        |       |        |                       | MOV   | @T38MS5, -(SP)                    |
|      | 044034 | 012746 | 000001        |       |        |                       | MOV   | @1, -(SP)                         |

| Address | Offset | Hex    | Dec    | Label  | Instruction | Comment             | Register/Value    |
|---------|--------|--------|--------|--------|-------------|---------------------|-------------------|
| 044040  | 010600 |        |        |        | MOV         | SP,R0               |                   |
| 044042  | 104417 |        |        |        | TRAP        | C\$PNTF             |                   |
| 044044  | 062706 | 000004 |        |        | ADD         | 04,SP               |                   |
| 7291    | 044050 | 012704 | 045350 |        | MOV         | 0T38PK3,R4          |                   |
| 7292    | 044054 | 010465 | 177776 |        | MOV         | R4,TSD8(R5)         |                   |
| 7293    | 044060 | 004737 | 017120 |        | JSR         | PC,WAITF            |                   |
| 7294    | 044064 | 016501 | 000000 |        | MOV         | TSSR(R5),R1         |                   |
| 7295    | 044070 | 012702 | 000300 | 320\$: | MOV         | 0SSR!OFL,R2         |                   |
| 7296    | 044074 | 020201 |        |        | CMP         | R2,R1               |                   |
| 7297    | 044076 | 001404 |        |        | BEQ         | 350\$               |                   |
| 7301    | 044100 |        |        |        | ERRHRD      | ERRNO,T38SST,PKTSSR |                   |
|         | 044100 | 104456 |        |        |             |                     | TRAP C\$ERRHRD    |
|         | 044102 | 000770 |        |        |             |                     | .WORD 504         |
|         | 044104 | 046240 |        |        |             |                     | .WORD T38SST      |
|         | 044106 | 011700 |        |        |             |                     | .WORD PKTSSR      |
| 7302    | 044110 |        |        | 350\$: | CKLOOP      |                     | SET TRAP C\$CLP1  |
|         | 044110 | 104406 |        |        |             |                     |                   |
| 7303    | 044112 | 016501 | 000000 |        | MOV         | TSSR(R5),R1         |                   |
| 7304    | 044116 | 032701 | 000100 |        | BIT         | 0OFL,R1             |                   |
| 7305    | 044122 | 001011 |        |        | BNE         | 380\$               |                   |
| 7306    | 044124 |        |        |        | PRINTF      | 0T38ONL             |                   |
|         | 044124 | 012746 | 045676 |        |             |                     | MOV 0T38ONL,-(SP) |
|         | 044130 | 012746 | 000001 |        |             |                     | MOV 01,-(SP)      |
|         | 044134 | 010600 |        |        |             |                     | MOV SP,R0         |
|         | 044136 | 104417 |        |        |             |                     | TRAP C\$PNTF      |
|         | 044140 | 062706 | 000004 |        |             |                     | ADD 04,SP         |
| 7307    | 044144 | 000410 |        |        | BR          | 390\$               |                   |
| 7308    | 044146 |        |        | 380\$: | PRINTF      | 0T38OFL             |                   |
|         | 044146 | 012746 | 045772 |        |             |                     | MOV 0T38OFL,-(SP) |
|         | 044152 | 012746 | 000001 |        |             |                     | MOV 01,-(SP)      |
|         | 044156 | 010600 |        |        |             |                     | MOV SP,R0         |
|         | 044160 | 104417 |        |        |             |                     | TRAP C\$PNTF      |
|         | 044162 | 062706 | 000004 |        |             |                     | ADD 04,SP         |
| 7309    | 044166 | 005037 | 002172 | 390\$: | CLR         | INTRECV             |                   |
| 7310    | 044172 | 000137 | 043510 |        | JMP         | 2\$                 |                   |
| 7311    | 044176 |        |        | 25\$:  | GMANIL      | T38MSG,T38DAT,-1,NO |                   |
|         | 044176 | 104443 |        |        |             |                     | TRAP C\$GMAN      |
|         | 044200 | 000404 |        |        |             |                     | BR 10000\$        |
|         | 044202 | 047042 |        |        |             |                     | .WORD T38DAT      |
|         | 044204 | 000120 |        |        |             |                     | .WORD T\$CODE     |
|         | 044206 | 046370 |        |        |             |                     | .WORD T38MSG      |
|         | 044210 | 177777 |        |        |             |                     | .WORD -1          |
|         | 044212 |        |        |        |             |                     | 10000\$:          |
| 7312    | 044212 |        |        |        | BNCOMPLETE  | 25\$                |                   |
|         | 044212 | 103371 |        |        |             |                     | BCC 25\$          |
| 7313    | 044214 | 005737 | 047042 |        | TST         | T38DAT              |                   |
| 7314    | 044220 | 001002 |        |        | BNE         | 27\$                |                   |
| 7315    | 044222 | 000137 | 043510 |        | JMP         | 2\$                 |                   |
| 7316    | 044226 |        |        | 27\$:  |             |                     |                   |
| 7317    | 044226 | 004737 | 016644 |        | JSR         | PC,SOFINIT          |                   |
| 7318    | 044232 | 103405 |        |        | BCS         | 400\$               |                   |
| 7322    | 044234 | 010001 |        |        | MOV         | R0,R1               |                   |
| 7323    | 044236 |        |        |        | ERRDF       | ERRNO,SFIERR,SFIMSG |                   |
|         | 044236 | 104455 |        |        |             |                     | TRAP C\$ERRDF     |
|         | 044240 | 000771 |        |        |             |                     | .WORD 505         |
|         |        |        |        |        |             |                     |                   |

| Address | Offset | Hex    | Label  | Instruction | Comment                              |
|---------|--------|--------|--------|-------------|--------------------------------------|
| 7324    | 044246 | 104406 | 400\$: | CKLOOP      | LOOP IF SELECTED                     |
|         | 044246 | 104406 |        |             | TRAP C\$CLP1                         |
| 7325    | 044250 | 012704 | 045320 | MOV         | \$T38PK2,R4                          |
| 7326    | 044254 | 004737 | 010332 | JSR         | PC,WRTCHR                            |
| 7327    | 044260 | 103405 |        | BCS         | 410\$                                |
| 7331    | 044262 | 010001 |        | MOV         | R0,R1                                |
| 7332    | 044264 |        |        | ERRHRD      | ERRNO,WRTMSG,SFMSG                   |
|         | 044264 | 104456 |        |             | WRITE CHARACTERISTICS FAILED         |
|         | 044266 | 000772 |        |             | TRAP C\$ERHRD                        |
|         | 044270 | 004754 |        |             | .WORD 506                            |
|         | 044272 | 011666 |        |             | .WORD WRTMSG                         |
|         |        |        |        |             | .WORD SFMSG                          |
| 7333    | 044274 |        | 410\$: | CKLOOP      | LOOP IF SELECTED                     |
|         | 044274 | 104406 |        |             | TRAP C\$CLP1                         |
| 7334    | 044276 | 013701 | 044614 | MOV         | T38BFR+6,R1                          |
| 7335    | 044302 | 010102 |        | MOV         | R1,R2                                |
| 7336    | 044304 | 052702 | 000004 | BIS         | \$BIT2,R2                            |
| 7337    | 044310 | 020102 |        | CMP         | R1,R2                                |
| 7338    | 044312 | 001406 |        | BEQ         | 430\$                                |
| 7342    | 044314 |        |        | ERRHRD      | ERRNO,T38WRL,EXPREC                  |
|         | 044314 | 104456 |        |             | "WRITE LOCKED BIT IS NOT SET ETC."   |
|         | 044316 | 000773 |        |             | TRAP C\$ERHRD                        |
|         | 044320 | 045541 |        |             | .WORD 507                            |
|         | 044322 | 016344 |        |             | .WORD T38WRL                         |
| 7343    | 044324 | 000137 | 043510 | JMP         | 2\$                                  |
| 7344    | 044330 |        | 430\$: | CKLOOP      | BECAUSE OF ERROR GO BACK TO MENU     |
|         | 044330 | 104406 |        |             | LOOP IF SELECTED                     |
| 7345    | 044332 |        |        | PRINTF      | \$T38WOK                             |
|         | 044332 | 012746 | 046430 |             | PRINT "DRIVE IS WRITE PROTECTED"     |
|         | 044336 | 012746 | 000001 |             | MOV \$T38WOK,-(SP)                   |
|         | 044342 | 010600 |        |             | MOV \$1,-(SP)                        |
|         | 044344 | 104417 |        |             | MOV SP,R0                            |
|         | 044346 | 062706 | 000004 |             | TRAP C\$PNIF                         |
| 7346    | 044352 | 017737 | 136514 | MOV         | \$FREE,T38WR                         |
| 7347    | 044360 | 012704 | 045370 | MOV         | \$T38PK4,R4                          |
| 7348    | 044364 | 010465 | 177776 | MOV         | R4,TSD8(R5)                          |
| 7349    | 044370 | 004737 | 017120 | JSR         | PC,WAITF                             |
| 7350    | 044374 | 016501 | 000000 | MOV         | TSSR(R5),R1                          |
| 7351    | 044400 | 012702 | 100206 | MOV         | \$SCSSR!BIT1!BIT2,R2                 |
| 7352    | 044404 | 020102 |        | CMP         | R1,R2                                |
| 7353    | 044406 | 001404 |        | BEQ         | 440\$                                |
| 7357    | 044410 |        |        | ERRHRD      | ERRNO,T38WRT,PKTSSR                  |
|         | 044410 | 104456 |        |             | "TSSR INCORRECT AFTER WRITE COMMAND" |
|         | 044412 | 000774 |        |             | TRAP C\$ERHRD                        |
|         | 044414 | 045455 |        |             | .WORD 508                            |
|         | 044416 | 011700 |        |             | .WORD T38WRT                         |
|         |        |        |        |             | .WORD PKTSSR                         |
| 7358    | 044420 |        | 440\$: | CKLOOP      | LOOP ON ERROR, IF FLAG SET           |
|         | 044420 | 104406 |        |             | TRAP C\$CLP1                         |
| 7359    | 044422 | 013701 | 044644 | MOV         | T38BFR+6,R1                          |
| 7360    | 044426 | 010102 |        | MOV         | R1,R2                                |
| 7361    | 044430 | 052702 | 004000 | BIS         | \$BIT11,R2                           |
| 7362    | 044434 | 020102 |        | CMP         | R1,R2                                |
| 7363    | 044436 | 001404 |        | BEQ         | 450\$                                |
| 7367    | 044440 |        |        | ERRHRD      | ERRNO,T38WLE,EXPREC                  |
|         | 044440 | 104456 |        |             | "WRITE LOCK ERROR BIT NOT SET"       |
|         | 044442 | 000775 |        |             | TRAP C\$ERHRD                        |
|         | 044444 | 045602 |        |             | .WORD 509                            |
|         | 044446 | 016344 |        |             | .WORD T38WLE                         |
|         |        |        |        |             | .WORD EXPREC                         |

|      |        |        |        |         |        |                     |                                 |       |                                       |
|------|--------|--------|--------|---------|--------|---------------------|---------------------------------|-------|---------------------------------------|
| 7368 | 044450 |        |        | 450\$:  | CKLOOP |                     | ;LOOP IF SELECTED               |       |                                       |
|      | 044450 | 104406 |        |         |        |                     |                                 | TRAP  | C#CLP1                                |
| 7369 | 044452 | 000137 | 043510 |         | JMP    | 2\$                 | ;GO BACK TO MENU                |       |                                       |
| 7370 |        |        |        |         |        |                     |                                 |       |                                       |
| 7371 | 044456 |        |        | 500\$:  | PRINTF | @T38MS6             | ;TELL'EM WHAT TO DO             |       |                                       |
|      | 044456 | 012746 | 046162 |         |        |                     |                                 | MOV   | @T38MS6,-(SP)                         |
|      | 044462 | 012746 | 000001 |         |        |                     |                                 | MOV   | @1,-(SP)                              |
|      | 044466 | 010600 |        |         |        |                     |                                 | MOV   | SP,R0                                 |
|      | 044470 | 104417 |        |         |        |                     |                                 | TRAP  | C#PNTF                                |
|      | 044472 | 062706 | 000004 |         |        |                     |                                 | ADD   | @4,SP                                 |
| 7372 | 044476 | 004737 | 016644 | 510\$:  | JSR    | PC,SOFINIT          | ;DO SOFT INIT OF CONTROLLER     |       |                                       |
| 7373 | 044502 | 103405 |        |         | BCS    | 520\$               | ;BR IF SOFT INIT = OK           |       |                                       |
| 7377 | 044504 | 010001 |        |         | MOV    | R0,R1               | ;SAVE CONTENTS OF TSSR          |       |                                       |
| 7378 | 044506 |        |        |         | ERRDF  | ERRNO,SFIERR,SFIMSG | ;DEVICE FATAL ERROR DURING INIT |       |                                       |
|      | 044506 | 104455 |        |         |        |                     |                                 | TRAP  | C#ERDF                                |
|      | 044510 | 000776 |        |         |        |                     |                                 | .WORD | 510                                   |
|      | 044512 | 003550 |        |         |        |                     |                                 | .WORD | SFIERR                                |
|      | 044514 | 011666 |        |         |        |                     |                                 | .WORD | SFIMSG                                |
| 7379 | 044516 |        |        | 520\$:  |        |                     |                                 |       |                                       |
| 7380 | 044516 | 016503 | 000000 |         | MOV    | TSSR(R5),R3         | ;IS DRIVE OFF-LINE              |       |                                       |
| 7381 | 044522 | 030327 | 000100 |         | BIT    | R3,@OFL             | ;LOOK AT THE OFF-LINE BIT       |       |                                       |
| 7382 | 044526 | 001412 |        |         | BEQ    | 550\$               | ;BR, IF DRIVE IS ON-LINE        |       |                                       |
| 7383 | 044530 |        |        |         | PRINTF | @T38OFL             | ; "DRIVE IS NOW OFF-LINE"       |       |                                       |
|      | 044530 | 012746 | 045772 |         |        |                     |                                 | MOV   | @T38OFL,-(SP)                         |
|      | 044534 | 012746 | 000001 |         |        |                     |                                 | MOV   | @1,-(SP)                              |
|      | 044540 | 010600 |        |         |        |                     |                                 | MOV   | SP,R0                                 |
|      | 044542 | 104417 |        |         |        |                     |                                 | TRAP  | C#PNTF                                |
|      | 044544 | 062706 | 000004 |         |        |                     |                                 | ADD   | @4,SP                                 |
| 7384 | 044550 | 000137 | 043510 |         | JMP    | 2\$                 | ;RETURN TO MENU                 |       |                                       |
| 7385 |        |        |        |         |        |                     |                                 |       |                                       |
| 7386 |        |        |        |         |        |                     |                                 |       |                                       |
| 7387 | 044554 |        |        | 550\$:  | PRINTF | @T38ONM             | ; "DRIVE IS NOW ON-LINE"        |       |                                       |
|      | 044554 | 012746 | 045740 |         |        |                     |                                 | MOV   | @T38ONM,-(SP)                         |
|      | 044560 | 012746 | 000001 |         |        |                     |                                 | MOV   | @1,-(SP)                              |
|      | 044564 | 010600 |        |         |        |                     |                                 | MOV   | SP,R0                                 |
|      | 044566 | 104417 |        |         |        |                     |                                 | TRAP  | C#PNTF                                |
|      | 044570 | 062706 | 000004 |         |        |                     |                                 | ADD   | @4,SP                                 |
| 7388 | 044574 | 000137 | 043510 |         | JMP    | 2\$                 | ;RETURN TO MENU                 |       |                                       |
| 7389 |        |        |        |         |        |                     |                                 |       |                                       |
| 7390 | 044600 |        |        | 63\$:   |        |                     |                                 |       |                                       |
| 7391 | 044600 |        |        | 64\$:   | EXIT   | TST                 | ;LEAVE TEST                     |       |                                       |
|      | 044600 | 104432 |        |         |        |                     |                                 | TRAP  | C#EXIT                                |
|      | 044602 | 002752 |        |         |        |                     |                                 | .WORD | L10064-                               |
| 7392 |        |        |        |         |        |                     |                                 |       |                                       |
| 7393 |        |        |        |         |        |                     |                                 |       |                                       |
| 7394 |        |        |        |         |        |                     |                                 |       |                                       |
| 7395 |        |        |        |         |        |                     |                                 |       |                                       |
| 7396 |        |        |        |         |        |                     |                                 |       |                                       |
| 7397 |        |        |        |         |        |                     |                                 |       |                                       |
| 7398 |        |        |        |         |        |                     |                                 |       |                                       |
| 7399 |        |        |        |         |        |                     |                                 |       |                                       |
| 7400 |        |        |        |         |        |                     |                                 |       |                                       |
| 7401 |        |        |        |         |        |                     |                                 |       |                                       |
| 7402 |        |        |        |         |        |                     |                                 |       |                                       |
| 7403 | 044604 | 000000 |        | TTION2: |        | .WORD 0             |                                 |       | ;WORD SET IF SUPERVISOR TTI INTER OFF |
| 7404 | 044606 | 000000 |        | TVSAV2: |        | .WORD 0             |                                 |       | ;SAVE TTI VECTOR                      |
| 7405 | 044610 | 000000 |        | TPSAV2: |        | .WORD 0             |                                 |       | ;SAVE TTI PRIORITY                    |

```

7406
7407 044612 000000      T38DLY: .WORD 0          ;DELAY COUNTER FOR TEST
7409 044614             .BLKB 10-<,-TUV2A67>
7411 044620             T38PACKET: ;COMMAND PACKET FOR TEST
7412 044620 140006       .WORD 140006      ;WRITE SUBSYSTEM MEM. CMD. ACK,CVC=1
7413 044622 044630       .WORD T38TAD      ;ADDRESS OF CHARACTERISTICS BLOCK
7414 044624 000000       .WORD 0
7415 044626 000012       .WORD 10.         ;STARTING VALUE OF BLOCK SIZE
7416 044630             T38TAD: ;CHARACTERISTICS DATA BLOCK
7417 044630 000         T38BS0: .BYTE 0      ;BSEL0 BYTE
7418 044631 000         T38BS1: .BYTE 0      ;BSEL1 BYTE
7419 044632 000000       T38BS2: .WORD 0      ;BSEL1 WORD
7420 044634 000000       .WORD 0          ;DATA
7421 044636             T38BFR: .BLKW 150.    ;MESSAGE BUFFER
7422 045312 000000       T38EB: .WORD         ;END OF BUFFER ADDRESS
7423
7424
7426 045314             .BLKB 10-<,-TUV2A67>
7428 045320             T38PK2: ;COMMAND PACKET FOR TEST
7429 045320 140004       .WORD 140004      ;WRITE CHARA. MEM. CMND., ACK,CVC=1
7430 045322 045330       .WORD T38DTA      ;ADDRESS OF SELECT DATA BLOCK
7431 045324 000000       .WORD 0
7432 045326 000012       .WORD 10.         ;STARTING VALUE OF BLOCK SIZE
7433
7434
7435 045330             T38DTA: ;SELECT DATA BLOCK
7436 045330 044636       .WORD T38BFR      ;ADDRESS OF MESSAGE BUFFER
7437 045332 000000       .WORD 0
7438 045334 000400       .WORD 256.        ;LENGTH OF MESSAGE BUFFER
7439 045336 000000       T38EAI: .WORD 0      ;EAI BIT WORD
7441 045340             .BLKB 10-<,-TUV2A67>
7443 045350 140412       T38PK3: .WORD 140412 ;REWIND AND UNLOAD COMMAND
7444 045352 000000       .WORD 0          ;NOT USED
7445 045354 000000       .WORD 0          ;NOT USED
7446 045356 000000       .WORD 0          ;NOT USED
7447 045360 000000       .WORD 0          ;NOT USED
7448
7449 ;WRITE TAPE PACKET
7450 ;
7452 045362             .BLKB 10-<,-TUV2A67>
7454 045370 140005       T38PK4: .WORD 140005 ;WRITE, ACK, CVC=1 COMMAND
7455 045372 000000       T38WR: .WORD 0      ;ADDRESS OF WRITE BUFFER
7456 045374 000000       .WORD 0          ;MORE ADDRESS OF WRITE BUFFER
7457 045376 000400       T38SIZ: .WORD 256.  ;SIZE OF RECORD
7458
7459
7460
7461
7462 ;
7463 ;LOCAL TEXT MESSAGES FOR TEST
7464 ;
7465
7466
7467
7468
7469
7470 045400 123 164 141 T38NE: .ASCIIZ 'Stand-alone Manual Intervention Not Executed'

```

```

7471 045455      124      123      123 T38WRT: .ASCIZ 'ISSR Not Correct After WRITE, With WRITE PROTECT On'
7472 045541      127      122      111 T38WRL: .ASCIZ 'WRITE LOCKED Bit Not Set In XST0'
7473 045602      127      122      111 T38WLE: .ASCIZ 'WRITE LOCK ERROR Bit Not Set In XST0, After Attempted WRITE'
7474 045676      045      116      045 T38ONL: .ASCIZ 'NMA ERROR Drive Is Still ON-LINE'
7475 045740      045      116      045 T38ONM: .ASCIZ 'NMA Drive Is Now ON-LINE'
7476 045772      045      116      045 T38OFL: .ASCIZ 'NMA Drive Is Now OFF-LINE'
7477 046026      045      116      045 T38MS4: .ASCIZ 'NMA Set Drive To On-line and At Or Beyond BOT'
7478 046105      045      116      045 T38MS5: .ASCIZ 'NMA Drive Should Now Rewind and Go Off-line'
7479
7480 046162      045      116      045 T38MS6: .ASCIZ 'NMA Front Panel On-line/Off-line Switch Test'
7481 046240      103      157      156 T38SST: .ASCIZ 'Contents Of ISSR Incorrect After REWIND And RELEASE'
7482 046324      045      116      045 T38MS2: .ASCIZ 'NMA Type RETURN To Return To Menu'
7483 046370      111      163      040 T38MSG: .ASCIZ 'Is Write-Protected Tape Mounted'
7484 046430      045      116      045 T38WOK: .ASCIZ 'NMA Drive Is Write Protected'
7485 046466      115      141      156 T38ID: .ASCIZ 'Manual Intervention'
7486
7487 046512      046532      046604      046632 MIMENU: .WORD 1$,2$,5$,6$,7$
7488 046524      046776      047041      000000 .WORD 9$,10$,0
7489
7490 046532      012      123      105 1$: .ASCIZ '<12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
7491 046604      012      011      060 2$: .ASCIZ '<12>' 0 Display This Menu'
7492 046632      011      061      011 5$: .ASCIZ ' 1 Rewind and Unload Command Test'
7493 046674      011      062      011 6$: .ASCIZ ' 2 Write Protect Test'
7494 046722      011      063      011 7$: .ASCIZ ' 3 Front Panel On-line/Off-line Switch Test'
7495 046776      011      064      011 9$: .ASCIZ ' 4 Return to Diagnostic Supervisor'
7496 047041      000      10$: .ASCIZ ''
7497
7498
7499
7500
7501
7502
7503 047042      000000 T38DAT: .WORD 0 ;LOGICAL RESPONSE TO QUESTION
7504 047044 T38REST:
7505 047044 SAVREG ;SAVE THE REGISTERS
7506 047050 012701 044620 MOV #T38PACKET,R1 ;START OF THE PACKET
7507 047054 012721 140206 MOV #140206,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,CVC*1
7508 047060 012721 044630 MOV #T38TAD,(R1)+ ;ADDRESS OF DATA BLOCK
7509 047064 005021 CLR (R1)+ ;EXTENDED ADDRESS
7510 047066 012721 000006 MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
7511 047072 005021 CLR (R1)+ ;CLEAR BSEL0 AND BSEL1
7512 047074 005021 CLR (R1)+ ;CLEAR SEL2
7513 047076 005011 CLR (R1) ;CLEAR DATA AREA
7514 047100 000207 RTS PC ;RETURN
7515
7516
7517
7518
7519
7520
7521
7522
7523
7524
7525
7526
7527

```

;  
 ;LOCAL STORAGE FOR THIS TEST  
 ;  
 ;  
 ;THIS ROUTINE PRINTS THE CONTENTS OF  
 ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE  
 ;TUV-05.  
 ;  
 ;INPUT:  
 ;  
 ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER  
 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER  
 ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR

M15

CZTKFA TK-25 FRT END FUNC 22  
TEST 5: MANUAL INTERVENTION

MACRO M1200 20-APR-84 08:12 PAGE 118-8

SEQ 194

```

7528
7529
7530
7531
7532 047102      T38MBP:
7533 047102      SAVREG
7534 047106 010005      MOV R0,R5      ;SAVE THE REGISTERS
7535 047 10 005737 003102      TST KTENABLE      ;SAVE LOW ORDER ADDRESS
7536 047.14 001001      BNE 910$      ;ADDRESS ABOVE 28K?
7537 047116 005001      CLR R1      ;BR IF YES
7538 047120 010103      MOV R1,R3      ;SET HIGH ORDER ADDRESS TO 0
7539 047122 006100      ROL R0      ;SAVE HIGH ORDER ADDRESS
7540 047124 006101      ROL R1      ;SHIFT BIT15 TO C BIT
7541 047126      PRINTX 0T38AS0,R1,R5      ;SHIFT TO HIGH ORDER FOR PRINTOUT
7542 047126 010546      ;PRINT MESSAGE BUFFER ADDRESS
7543 047130 010146      MOV R5,-(SP)
7544 047132 012746 047404      MOV R1,-(SP)
7545 047136 012746 000003      MOV 0T38AS0,-(SP)
7546 047142 010600      MOV 03,-(SP)
7547 047144 104415      MOV SP,R0
7548 047146 062706 000010      TRAP C:PNTX
7549 047152      ADD 10,SP
7550 047152      PRINTX 0T38AS1      ;PRINT HEADER FOR CONTENTS
7551 047152 012746 047451      MOV 0T38AS1,-(SP)
7552 047156 012746 000001      MOV 01,-(SP)
7553 047162 010600      MOV SP,R0
7554 047164 104415      TRAP C:PNTX
7555 047166 062706 000004      ADD 04,SP
7556 047172 010501      MOV R5,R1      ;COPY LOW ORDER ADDRESS
7557 047174 010300      MOV R3,R0      ;COPY HIGH ORDER ADDRESS
7558 047176 001403      BEQ 913$      ;BR IF NOT ABOVE 28K
7559 047200 004737 020270      JSR PC,SETMAP      ;SETUP PAR ADDRESS IN R0
7560 047204 010005      MOV R0,R5      ;GET PAR FORMAT ADDRESS ABOVE 28K
7561 047206 010537 047552      913$: MOV R5,T38CNT      ;HOLD ADDRESS
7562 047212 011504      911$: MOV (R5),R4      ;GET BUFFER ENTRY
7563 047214 022704 125252      CMP 0125252,R4      ;CHECK FOR NO LOAD CONDITION
7564 047220 001417      BEQ 912$      ;BR, IF BUFFER WASN'T LOADED
7565 047222 010403      MOV R4,R3      ;MAKE COPY
7566 047224 042704 170377      BIC 0170377,R4      ;ONLY BITS 11,10,9 AND 8 ARE SAVED
7567 047230 000241      CLC      ;CLEAR CARRY
7568 047232 006003      ROR R4      ;11 TO 10 BIT POSITION
7569 047234 006003      ROR R4      ;10 TO 9 BIT POSITION
7570 047236 006004      ROR R4      ;9 TO 8 BIT POSITION
7571 047240 006004      ROR R4      ;8 TO 7 BIT POSITION
7572 047242 042703 177760      BIC 0177760,R3      ;ONLY BITS 3,2,1 AND 0 ARE SAVED
7573 047246 060403      ADD R4,R3      ;"OR'EM TOGETHER
7574 047250 010325      MOV R3,(R5)+      ;PUT BACK IN BUFFER
7575 047252 020527 045312      CMP R5,0T38EB      ;END OF BUFFER YET
7576 047256 001355      BNE 911$      ;BR, IF NOT AT END YET
7577 047260 013705 047552      912$: MOV T38CNT,R5      ;PUT ADDRESS BACK
7578 047264 012704 000001      MOV 01,R4      ;START BYTE NUMBER AT ONE
7579 047270      915$: PRINTX 0T38ASN,R4,(R5)+      ;PRT MEM BUFFER W/NEWLINE
7580 047270 012546      MOV (R5)+,-(SP)
7581 047272 010446      MOV R4,-(SP)
7582 047274 012746 047526      MOV 0T38ASN,-(SP)
7583 047300 012746 000003      MOV 03,-(SP)
7584 047304 010600      MOV SP,R0
7585 047306 104415      TRAP C:PNTX

```

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

```

16::
;CLEAR FATAL ERROR FLAG
;HOLD OF RT11
;IS THIS FIRST TEST IN SEQUENCE ?
;BR. IF FIRST TEST
;"TEST NOT EXECUTED"
;JUMP OUT OF TEST IF NOT
;TEST ID MESSAGE
;DO THE COMMON SETUP
;IS MANUAL INTERVENTION ALLOWED?
;BR. IF MANUAL INTERVENTION ALLOWED
;JUMP TO OUT IF NOT
;IS THIS THE FIRST TEST IN SEQ
;BR. IF FIRST TEST
;JUMP IF IT WASN'T

;DO SOFT INIT OF CONTROLLER
;BR IF SOFT INIT = OK

```

| Address | Hex    | Dec    | Label  | Op     | Opnd                | Comment                            |
|---------|--------|--------|--------|--------|---------------------|------------------------------------|
| 7652    | 047646 | 010001 |        | MOV    | R0,R1               | SAVE CONTENTS OF TSSR              |
| 7653    | 047650 | 104455 |        | ERRDF  | ERRNO,SFIERR,SFIMSG | DEVICE FATAL ERROR DURING INIT     |
|         | 047652 | 001131 |        |        |                     | TRAP C#ERDF                        |
|         | 047654 | 003550 |        |        |                     | .WORD 601                          |
|         | 047656 | 011666 |        |        |                     | .WORD SFIERR                       |
| 7654    | 047660 |        | 251:   | CKLOOP |                     | .WORD SFIMSG                       |
|         | 047660 | 104406 |        |        |                     | LOOP IF SELECTED                   |
| 7655    | 047662 | 012704 | 050730 | MOV    | #T39PK2,R4          | TRAP C#CLP1                        |
| 7656    | 047666 | 004737 | 010332 | JSR    | PC,WRTCHR           | SUBROUTINE NEEDS PACKET ADDRESS    |
| 7657    | 047672 | 103405 |        | BCS    | 501                 | ISSUE WRITE CHARACTERISTICS        |
| 7661    | 047674 | 010001 |        | MOV    | R0,R1               | IF COMMAND ISSUED OK               |
| 7662    | 047676 |        |        | ERRHRD | ERRNO,WRTMSG,SFIMSG | SAVE CONTENTS OF TSSR              |
|         | 047676 | 104456 |        |        |                     | WRITE CHARACTERISTICS FAILED       |
|         | 047700 | 001132 |        |        |                     | TRAP C#ERHRD                       |
|         | 047702 | 004754 |        |        |                     | .WORD 602                          |
|         | 047704 | 011666 |        |        |                     | .WORD WRTMSG                       |
| 7663    | 047706 |        | 501:   | CKLOOP |                     | .WORD SFIMSG                       |
|         | 047706 | 104406 |        |        |                     | LOOP IF SELECTED                   |
| 7664    | 047710 | 013701 | 050252 | MOV    | T39BFR+4,R1         | TRAP C#CLP1                        |
| 7665    | 047714 | 042701 | 177700 | BIC    | #177700,R1          | PICK UP THE RES. BYTE CNTR AREA    |
| 7666    | 047720 | 010137 | 051406 | MOV    | R1,T39RL            | ONLY LEAVE MICROCODE REV LEVEL     |
| 7667    | 047724 |        |        | PRINTX | #T39MCL,T39RL       | LOAD UP REV LEVEL                  |
|         | 047724 | 013746 | 051406 |        |                     | "MICROCODE REVISION LEVEL =000XXX" |
|         | 047730 | 012746 | 051323 |        |                     | MOV T39RL,-(SP)                    |
|         | 047734 | 012746 | 000002 |        |                     | MOV #T39MCL,-(SP)                  |
|         | 047740 | 010600 |        |        |                     | MOV #2,-(SP)                       |
|         | 047742 | 104415 |        |        |                     | MOV SP,R0                          |
|         | 047744 | 062706 | 000006 |        |                     | TRAP C#PNTX                        |
| 7668    | 047750 | 004737 | 016644 | JSR    | PC,SOFINIT          | ADD #6,SP                          |
| 7669    | 047754 | 103405 |        | BCS    | 1401                | DO SOFT INIT OF CONTROLLER         |
| 7673    | 047756 | 010001 |        | MOV    | R0,R1               | IF SOFT INIT = OK                  |
| 7674    | 047760 |        |        | ERRDF  | ERRNO,SFIERR,SFIMSG | SAVE CONTENTS OF TSSR              |
|         | 047760 | 104455 |        |        |                     | DEVICE FATAL ERROR DURING INIT     |
|         | 047762 | 001133 |        |        |                     | TRAP C#ERDF                        |
|         | 047764 | 003550 |        |        |                     | .WORD 603                          |
|         | 047766 | 011666 |        |        |                     | .WORD SFIERR                       |
| 7675    | 047770 |        | 1401:  | CKLOOP |                     | .WORD SFIMSG                       |
|         | 047770 | 104406 |        |        |                     | LOOP IF SELECTED                   |
| 7676    | 047772 | 012704 | 050730 | MOV    | #T39PK2,R4          | TRAP C#CLP1                        |
| 7677    | 047776 | 004737 | 010332 | JSR    | PC,WRTCHR           | SUBROUTINE NEEDS PACKET ADDRESS    |
| 7678    | 050002 | 103405 |        | BCS    | 1501                | ISSUE WRITE CHARACTERISTICS        |
| 7682    | 050004 | 010001 |        | MOV    | R0,R1               | IF COMMAND ISSUED OK               |
| 7683    | 050006 |        |        | ERRHRD | ERRNO,WRTMSG,SFIMSG | SAVE CONTENTS OF TSSR              |
|         | 050006 | 104456 |        |        |                     | WRITE CHARACTERISTICS FAILED       |
|         | 050010 | 001134 |        |        |                     | TRAP C#ERHRD                       |
|         | 050012 | 004754 |        |        |                     | .WORD 604                          |
|         | 050014 | 011666 |        |        |                     | .WORD WRTMSG                       |
| 7684    | 050016 |        | 1501:  | CKLOOP |                     | .WORD SFIMSG                       |
|         | 050016 | 104406 |        |        |                     | LOOP IF SELECTED                   |
| 7685    | 050020 | 005037 | 002150 | CLR    | UNITN               | TRAP C#CLP1                        |
| 7686    |        |        |        |        |                     | SET TO DRIVE 0                     |
| 7687    | 050024 | 016501 | 000000 | MOV    | TSSR(R5),R1         | GET TSSR STATUS                    |
| 7688    | 050030 | 032701 | 000100 | BIT    | #OFL,R1             | CHECK FOR OFF-LINE                 |
| 7689    | 050034 | 001414 |        | BEQ    | 2001                | IF DRIVE IS ON-LINE                |
| 7690    | 050036 |        |        | PRINTX | #T39OF2,UNITN       | "DRIVE NUMBER XX IS OFF LINE"      |
|         | 050036 | 013746 | 002150 |        |                     | MOV UNITN,-(SP)                    |

```

,DELAY: COUNTER FOR TEST
,COMMAND PACKET FOR TEST
,WRITE SUBSYSTEM MEM. CMD. ACK,CVC=1
,ADDRESS OF CHARACTERISTICS BLOCK
,STARTING VALUE OF BLOCK SIZE
,CHARACTERISTICS DATA BLOCK
,BSLO BYTE

```

LOCAL STORAGE FOR THIS TEST

F16

CZTKFA TK-25 FRT END FUNC #2  
TEST 6: CONFIGURATION TYPEOUT

MACRO M1200 20-APR-84 08:12 PAGE 119-4

SEQ 200

```

7783 051410 000000          T39DAT: .WORD 0          ;LOGICAL RESPONSE TO QUESTION
7784 051412          T39REST:
7785 051412          SAVREG          ;SAVE THE REGISTERS
7786 051416 012701 050230      MOV     #T39PACKET,R1      ;START OF THE PACKET
7787 051422 012721 140006      MOV     #140006,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
7788 051426 012721 050240      MOV     #T39TAD,(R1)+      ;ADDRESS OF DATA BLOCK
7789 051432 005021          CLR     (R1)+      ;EXTENDED ADDRESS
7790 051434 012721 000006      MOV     #6,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
7791 051440 005021          CLR     (R1)+      ;CLEAR 9SELO AND BSEL1
7792 051442 005021          CLR     (R1)+      ;CLEAR SEL2
7793 051444 005011          CLR     (R1)        ;CLEAR DATA AREA
7794 051446 000207          RTS     PC          ;RETURN
7795
7796
7797          ;LOCAL TEXT MESSAGES FOR TEST
7798          ;
7799
7800 051450          103      157      156  TST39ID:      .ASCIZ 'Configuration Typeout'
7801          .EVEN
7802 051476          ENDTST
7803 051476 104401          L10065:          TRAP      C$ETST

```

```

7805          .SBTTL  TEST 7: SCOPE LOOPS
7806
7807
7808          ; THIS TEST MUST BE STARTED AS FOLLOWS:
7809
7810          ; AT THE DIAGNOSTIC SUPERVISOR PROMPT "DR>"
7811          ; TYPE "START/FLAG:PNT/TEST:7/PASS:1"
7812          ;
7813          ; -----
7814
7815
7816
7817
7818          ; THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE
7819          ; LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH
7820          ; THE CONTROLLER MODULE.  THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL
7821          ; "LOOP ON ERROR" OR "LOOP ON TEST (SUBTEST)" FACILITIES DON'T
7822          ; SEEM TO ALLOW THE OPERATOR TO ZERO IN A PROBLEM IN THE EARLY
7823          ; TESTS (I.E., THE HARDWARE MAY NOT BE RESPONDING TO A REGISTER
7824          ; ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS
7825          ; SELECTED BY THE PROGRAM MATCHES THE CONFIGURATION SET UP IN THE
7826          ; HARDWARE DIP SWITCHES).  THE FOLLOWING MENU OF SCOPE LOOPS ARE
7827          ; AVAILABLE:
7828
7829          ;
7830          ; CODE      SCOPE LOOP
7831          ;
7832          ; 0        HELP. DISPLAY THIS MENU.
7833          ; 1        TSBA READ ACCESS
7834          ; 2        TSSR READ ACCESS
7835          ; 3        INITIALIZE (TSSR WRITE ACCESS)
7836          ; 4        TSD8 HIGH BYTE WRITE ACCESS
7837          ; 5        TSD8 LOW BYTE WRITE ACCESS
7838          ; 6        TSSR BYTE WRITE (SELF-TEST)
7839          ; 7        RETURN TO DIAGNOSTIC SUPERVISOR
7840
7841          ;
7842          ; FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS
7843          ; THE OPERATOR FOR THE DATA TO BE WRITTEN.  TYPING <RETURN> CAUSES
7844          ; AN EXIT FROM THE SCOPE LOOP BACK TO MENU LEVEL.
7845          ;
7846
7847          051500      BGNTST
7848          051500
7849          051500      005037      002170      CLR      FATFLG
7850          051504      005037      003100      CLR      KTFLG
7851          051510      005737      002162      TST      TSTCNT
7852          051514      001403      BEQ      1001
7853          051516      012700      052711      MOV      #T4ONE,RO
7854          051522      000402      BR      1001
7855          051524      012700      052756      1001:  MOV      #TST40ID,RO
7856          051530      004737      017410      1001:  JSR      PC,TSTSETUP
7857          051534      004737      021404      JSR      PC,CHKMAN
7858          051540      103402      BCS      1021
7859          051542      000137      052174      JMP      641
7860          051544      022737      000001      002162      1021:  CMP      #1,TSTCNT
7861          051554      001402      BEQ      21
7862          051556      000137      052174      JMP      641

```

```

T7::
;CLEAR FATAL ERROR FLAG
;HOLD OF KT11
;IS THIS FIRST TEST IN RUN ?
;CONTINUE TEST IF FIRST IN RUN
;"TEST NOT EXECUTED"
;JUST EXIT IF NOT
;TEST ID MESSAGE
;DO THE COMMON SETUP
;SEE IF MANUAL INTERVENTION ALLOWED
;CARRY SET IF INTERVENTION ALLOWED
;EXIT IF NO MANUAL INTERVENTION
;WAS THIS THE FIRST TEST IN SEQ
;BR, IF IT WAS
;JUMP TO END OF TEST

```

| Address | Offset | Hex    | Label  | Instruction | Comment             |
|---------|--------|--------|--------|-------------|---------------------|
| 7865    | 051562 | 004737 | 016644 | 2\$: JSR    | PC,SOFINIT          |
| 7866    | 051566 | 103405 |        | BCS         | 5\$                 |
| 7867    | 051570 | 010001 |        | MOV         | R0,R1               |
| 7871    | 051572 |        |        | ERRDF       | ERRNO,SFIERR,SFIMSG |
|         | 051572 | 104455 |        |             |                     |
|         | 051574 | 001275 |        |             |                     |
|         | 051576 | 003550 |        |             |                     |
|         | 051600 | 011666 |        |             |                     |
| 7872    | 051602 | 012700 | 052206 | 5\$: MOV    | #SCMENU,R0          |
| 7873    | 051606 | 012701 | 000010 | MOV         | #8,R1               |
| 7874    | 051612 | 004737 | 021162 | JSR         | PC,GETSEL           |
| 7875    | 051616 | 005700 |        | TST         | R0                  |
| 7876    | 051620 | 001760 |        | BEQ         | 2\$                 |
| 7877    | 051622 | 010004 |        | 3\$: MOV    | R0,R4               |
| 7878    | 051624 |        |        | SETPRI      | #PRI07              |
|         | 051624 | 012700 | 000340 |             |                     |
|         | 051630 | 104441 |        |             |                     |
| 7879    | 051632 | 005037 | 052200 | CLR         | TTION               |
| 7880    | 051636 | 032737 | 000100 | 177560      | BIT                 |
| 7881    | 051644 | 001005 |        | BNE         | 4\$                 |
| 7882    | 051646 | 005237 | 052200 | INC         | TTION               |
| 7883    | 051652 | 052737 | 000100 | 177560      | BIS                 |
| 7884    | 051660 | 012701 | 000060 | 4\$: MOV    | #TTIVEC,R1          |
| 7885    | 051664 | 011137 | 052202 | MOV         | (R1),TVECSAV        |
| 7886    | 051670 | 012721 | 052106 | MOV         | #60,(R1),           |
| 7887    | 051674 | 011137 | 052204 | MOV         | (R1),TPRISAV        |
| 7888    | 051700 | 012711 | 000340 | MOV         | #PRI07,(R1)         |
| 7889    | 051704 |        |        | SETPRI      | #PRI00              |
|         | 051704 | 012700 | 000000 |             |                     |
|         | 051710 | 104441 |        |             |                     |
| 7890    | 051712 | 006304 |        | ASL         | R4                  |
| 7891    | 051714 | 000174 | 051720 | 6\$: JMP    | #6,(R4)             |
| 7892    | 051720 | 051562 |        | .WORD       | 2\$                 |
| 7893    | 051722 | 051740 |        | .WORD       | 10\$                |
| 7894    | 051724 | 051746 |        | .WORD       | 15\$                |
| 7895    | 051726 | 051754 |        | .WORD       | 20\$                |
| 7896    | 051730 | 051770 |        | .WORD       | 25\$                |
| 7897    | 051732 | 052004 |        | .WORD       | 30\$                |
| 7898    | 051734 | 052020 |        | .WORD       | 35\$                |
| 7899    | 051736 | 052174 |        | .WORD       | 65\$                |
| 7900    |        |        |        |             |                     |
| 7901    |        |        |        |             |                     |
| 7902    | 051740 |        |        | 10\$: MOV   | TSBA(R5),R0         |
| 7903    | 051740 | 016500 | 177776 | 12\$: BR    | 12\$                |
| 7904    | 051744 | 000775 |        |             |                     |
| 7905    |        |        |        |             |                     |
| 7906    | 051746 |        |        | 15\$: MOV   | TSSR(R5),R0         |
| 7907    | 051746 | 016500 | 000000 | 18\$: BR    | 18\$                |
| 7908    | 051752 | 000775 |        |             |                     |
| 7909    |        |        |        |             |                     |
| 7910    |        |        |        |             |                     |
| 7911    | 051754 | 004737 | 021100 | 20\$: JSR   | PC,GETPAT           |
| 7912    | 051760 | 010001 |        | MOV         | R0,R1               |
| 7913    | 051762 | 010165 | 000000 | 22\$: MOV   | R1,TSSR(R5)         |
| 7914    | 051766 | 000775 |        | BR          | 22\$                |
| 7915    |        |        |        |             |                     |
| 7916    |        |        |        |             |                     |

TEST 7: SCOPE LOOPS

|             |        |        |        |       |                     |                                         |
|-------------|--------|--------|--------|-------|---------------------|-----------------------------------------|
| 7917 051770 |        | 25:    |        |       |                     |                                         |
| 7918 051770 | 004737 | 021100 |        | JSR   | PC,GETPAT           | ;READ THE DATA PATTERN                  |
| 7919 051774 | 010001 |        |        | MOV   | R0,R1               | ;DATA PATTERN FOR LOOP                  |
| 7920 051776 | 110165 | 177777 | 27:    | MOVB  | R1,TSDBH(R5)        | ;WRITE THE DATA TO TSDB, HIGH BYTE      |
| 7921 052002 | 000775 |        |        | BR    | 27:                 | ;LOOP UNTIL STOPPED                     |
| 7922        |        |        |        |       |                     |                                         |
| 7923        |        |        |        |       |                     |                                         |
| 7924 052004 |        |        | 30:    |       |                     |                                         |
| 7925 052004 | 004737 | 021100 |        | JSR   | PC,GETPAT           | ;READ THE DATA PATTERN                  |
| 7926 052010 | 010001 |        |        | MOV   | R0,R1               | ;DATA PATTERN FOR LOOP                  |
| 7927 052012 | 110165 | 177776 | 32:    | MOVB  | R1,TSDB(R5)         | ;WRITE DATA TO TSSR, LOW BYTE           |
| 7928 052016 | 000775 |        |        | BR    | 32:                 | ;LOOP UNTIL HALTED BY OPERATOR          |
| 7929        |        |        |        |       |                     |                                         |
| 7930 052020 | 004737 | 021100 | 35:    | JSR   | PC,GETPAT           | ;PICK UP THE PATTERN FROM OPR           |
| 7931 052024 | 010001 |        |        | MOV   | R0,R1               | ;STORE IN R1                            |
| 7932 052026 | 110165 | 000000 | 37:    | MOVB  | R1,TSSR(R5)         | ;WRITE BYTE TO TSSR, THIS STARTS MDIAG  |
| 7933 052032 |        |        |        | DELAY | 250                 | ;WAIT 2500US                            |
|             | 052032 | 012727 |        |       |                     | MOV #250,(PC)+                          |
|             | 052036 | 000000 |        |       |                     | .WORD 0                                 |
|             | 052040 | 013727 |        |       |                     | MOV L\$DLY,(PC)+                        |
|             | 052044 | 000000 |        |       |                     | .WORD 0                                 |
|             | 052046 | 005367 | 177772 |       |                     | DEC -6(PC)                              |
|             | 052052 | 001375 |        |       |                     | BNE -4                                  |
|             | 052054 | 005367 | 177756 |       |                     | DEC -22(PC)                             |
|             | 052060 | 001367 |        |       |                     | BNE -20                                 |
| 7934 052062 | 036527 | 000000 | 000200 | BIT   | TSSR(R5),#SSR       | ;CHECK FOR READY SET, IF A TRAP OCCURS  |
| 7935        |        |        |        |       |                     | ;THE TSSR WAS "NOT BACK TO THE BUS"     |
| 7936        |        |        |        |       |                     | ;IN TIME.                               |
| 7937 052070 | 001356 |        |        | BNE   | 37:                 | ;BR, IF SSR WAS SET (GOOD) KEEP LOOPING |
| 7941 052072 |        |        |        | ERRDF | ERRNO,T4ONSR,SFIMSG | ;PRINT DEVICE FATAL ERROR MESSAGE       |
|             | 052072 | 104455 |        |       |                     | TRAP C\$ERDF                            |
|             | 052074 | 001276 |        |       |                     | .WORD 702                               |
|             | 052076 | 052772 |        |       |                     | .WORD T4ONSR                            |
|             | 052100 | 011666 |        |       |                     | .WORD SFIMSG                            |
| 7942 052102 | 000137 | 000200 |        | JMP   | 200                 | ;GO TO SUPERVISOR ETC.                  |
| 7943        |        |        |        |       |                     |                                         |
| 7944        |        |        |        |       |                     |                                         |
| 7945        |        |        |        |       |                     |                                         |
| 7946        |        |        |        |       |                     |                                         |
| 7947        |        |        |        |       |                     |                                         |
| 7948        |        |        |        |       |                     |                                         |
| 7949 052106 | 010046 |        | 60:    | MOV   | R0,-(SP)            | ;SAVE WORK REGISTER                     |
| 7950 052110 | 113700 | 177562 |        | MOVB  | #TTIBFR,R0          | ;GET THE OPERATOR INPUT                 |
| 7951 052114 | 042700 | 000200 |        | BIC   | #200,R0             | ;STRIP OFF PARITY BIT                   |
| 7952 052120 | 122700 | 000015 |        | CMPB  | #15,R0              | ;IS IT A CARRIAGE RETURN ?              |
| 7953 052124 | 001021 |        |        | BNE   | 61:                 | ;JUST EXIT IF NOT                       |
| 7954 052126 | 012766 | 051562 | 000002 | MOV   | #21,2(SP)           | ;RETURN TO MASTER MENU                  |
| 7955 052134 | 005066 | 000004 |        | CLR   | 4(SP)               | ;FORCE PRIORITY ZERO                    |
| 7956 052140 | 013737 | 052202 | 000060 | MOV   | IVECSAV,#TTIVEC     | ;RESTORE SUPERVISOR VECTOR              |
| 7957 052146 | 013737 | 052204 | 000062 | MOV   | TPRISAV,#TTIVEC+2   | ;RESTORE SUPERVISOR PRIORITY            |
| 7958 052154 | 005737 | 052200 |        | TSI   | ITION               | ;ARE SUPERVISOR INTERRUPTS ENABLED ?    |
| 7959 052160 | 001403 |        |        | BEQ   | 61:                 | ;BRANCH IF YES                          |
| 7960 052162 | 042737 | 000100 | 177560 | BIC   | #100,#TTICSR        | ;TURN OFF ITI INTERRUPTS                |
| 7961 052170 | 012600 |        | 61:    | MOV   | (SP)+,R0            | ;RESTORE REGISTER                       |
| 7962 052172 | 000002 |        |        | RTI   |                     | ;RETURN FROM INTERRUPT                  |
| 7963        |        |        |        |       |                     |                                         |
| 7964 052174 |        |        | 64:    |       |                     |                                         |

```

7965 052174      63$:
7966 052174      65$:  EXIT  TST                ;EXIT THE TEST
      052174 104432
      052176 000664                TRAP  C$EXIT
                                   .WORD  L10066-.
7967
7968
7969      ;
7970      ;LOCAL STORAGE FOR THIS TEST
7971      ;
7972 052200 000000      TTION:      .WORD  0                ;WORD SET IF SUPERVISOR ITI INTER OFF
7973 052202 000000      TVECSAV:   .WORD  0                ;SAVE ITI VECTOR
7974 052204 000000      TPRISAV:   .WORD  0                ;SAVE ITI PRIORITY
7975
7976
7977      ;
7978      ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
7979      ;
7980
7981      .EVEN
7982 052206 052236 052311 052337  SCMENU: .WORD  1$,2$,3$,4$,5$,6$
7983 052222 052510 052546 052605      .WORD  7$,8$,10$,11$,12$,0
7984
7985
7986 052236      012      123      105  1$:  .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
7987 052311      012      011      060  2$:  .ASCIZ  <12>' 0      Display This Menu'
7988 052337      011      061      011  3$:  .ASCIZ  ' 1      TSBA Read Access'
7989 052363      011      062      011  4$:  .ASCIZ  ' 2      TSSR Read Access'
7990 052407      011      063      011  5$:  .ASCIZ  ' 3      Initialize (TSSR Write Access)'
7991 052451      011      064      011  6$:  .ASCIZ  ' 4      TSDB High Byte Write Access'
7992 052510      011      065      011  7$:  .ASCIZ  ' 5      TSDB Low Byte Write Access'
7993 052546      011      066      011  8$:  .ASCIZ  ' 6      TSSR Write Byte (Self Test)'
7994 052605      011      067      011 10$:  .ASCIZ  ' 7      Return to Diagnostic Supervisor'
7995 052650      000      11$:  .ASCIZ  ''
7996 052651      124      171      160 12$:  .ASCIZ  'Type RETURN To Stop Scope Loops'
7997 052711      123      164      141 T4ONE: .ASCIZ  'Stand-alone Scope Loops Not Executed'
7998 052756      123      143      157 T40ID:  .ASCIZ  'Scope Loops'
7999 052772      123      123      122 T40NSR: .ASCIZ  'SSR Failed To Set After TSSR Write Byte And 10ms Delay'
8000      .EVEN
8001 053062      053062      104401      ENDTST
                                   L10066:  TRAP  C$ETST
8007
8012
8018
8019
8020
8021      .SBTTL  HARDWARE PARAMETER CODING SECTION
8022
8023      ;
8024      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
8025      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
8026      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8027      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
8028      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8029      ; WITH THE OPERATOR.
8030      ;
8031 053064      BGNHRD

```

K16

CZTKFA TK-25 FRT END FUNC 42 MACRO M1200 20-APR-84 08:12 PAGE 120-4  
HARDWARE PARAMETER CODING SECTION

SEQ 205

```

      053064 000015          .WORD L10067-L$HARD/2
      053066          ' $HARD;;
8032
8033 053066          GPRMA  HPM1,0,0,160000,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
      053066 000031          .WORD  T$CODE
      053070 053120          .WORD  HPM1
      053072 160000          .WORD  T$LOLIM
      053074 177776          .WORD  T$HILIM
8034 053076          GPRMA  HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
      053076 001031          .WORD  T$CODE
      053100 053147          .WORD  HPM2
      053102 000000          .WORD  T$LOLIM
      053104 000776          .WORD  T$HILIM
8035 053106          GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
      053106 002032          .WORD  T$CODE
      053110 053173          .WORD  HPM3
      053112 000340          .WORD  340
      053114 000000          .WORD  T$LOLIM
      053116 000007          .WORD  T$HILIM
8036 053120          ENDMRD
      053120          .EVEN
      053120          L10067:
8037 053120          104      105      126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSSR) '
8038 053147          111      116      124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
8039 053173          111      116      124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
8040
8041          .EVEN

```

L16

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 121  
SOFTWARE PARAMETER CODING SECTION

SEQ 206

```

8043          .SBTTL  SOFTWARE PARAMETER CODING SECTION
8044
8045          ;++
8046          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
8047          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8048          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8049          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8050          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8051          ; WITH THE OPERATOR.
8052          ;--
8053 053224      BGNSFT
8054          053224 000006      .WORD L10070-L$SOFT/2
8055          053226      L$SOFT::
8056          053226 000130      GPRM1  SPM1,0,-1,YES      ;GET RAM DUMP FLAG
8057          053230 053242      .WORD  T$CODE
8058          053232 177777      .WORD  SPM1
8059          053234      GPRM1  SPM4,2,-1,YES      ; GET ITERATION CONTROL.
8060          053234 001130      .WORD  T$CODE
8061          053236 053306      .WORD  SPM4
8062          053240 177777      .WORD  -1
8063          053242      GPRM6  SPM6,4,D,7777,0,7777,YES      ; GET LOCAL ERROR LIMIT
8064          053242      GPRM7  SPM7,6,D,7777,0,7777,YES      ; GET GLOBAL ERROR LIMIT
8065          053242      ENDSFT
8066          053242      L10070:
8067          053242      .EVEN
8068          105      116      101  SPM1:  .ASCIZ  'ENABLE CONTROLLER RAM DUMP ON ERROR'
8069          111      116      110  SPM4:  .ASCIZ  'INHIBIT ITERATIONS'
8070          120      105      122  SPM6:  .ASCIZ  'PER TEST ERROR LIMIT'
8071          120      105      122  SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT'
8072          053366      .EVEN
8073          053366      .SBTTL  PATCH AREA
8074          ;+
8075          ;DISPATCH TABLE
8076          ;
8077          ; *** MOVE TO FRONT OF PROGRAM FOR RELEASE ***
8078          ;--
8079          DISPATCH      TESTNO
8080          053416      .WORD  7
8081          053420      L$DISPATCH::
8082          053420 023654      .WORD  T1
8083          053422 024674      .WORD  T2
8084          053424 026226      .WORD  T3
8085          053426 032160      .WORD  T4
8086          053430 043426      .WORD  T5
8087          053432 047556      .WORD  T6
8088          053434 051500      .WORD  T7
8089          ;
8090          ; FINALLY A GENEROUS PATCH AREA.
8091          ;
8092          ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT?" HACK
8093          ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
8094          ;

```

CZTKFA TK-25 FRT END FUNC #2  
PATCH AREA

MACRO M1200 20-APR-84 08:12 PAGE 121-1

SEQ 207

8081  
 8082 053436  
 8083  
 8084  
 8085  
 8086  
 8087 053436  
  
 053436 053454  
 053440 000005  
 053442  
 8088  
 8089  
 8090  
 8091  
 8092 053442  
 8093 053442  
 053442 000000  
 053444 000003  
 053446  
 8094 053446 172522  
 8095 053450 000224  
 8096 053452 000240  
 8097 053454  
 053454  
 8098 053454  
 8099  
 8100 000001

```

PATCH::
      .EVEN
      .IF      NZ,,&377
      .-..!377+1
      .ENDC
      LASTAD      ;SET LAST USED ADDRESS.
      .EVEN
      .WORD T$FREE
      .WORD T$SIZE
L$LAST::
      .SBTTL  HARD CODED P-TABLE
      ;++
      ;--
      BGNSETUP      1
      BGNPTAB
      .WORD      0
      .WORD      L10073-.,/2-1
L10071:
      .WORD      172522
      .WORD      224
      .WORD      PRI05
      ENDP TAB
L10073:
      ENDSETUP
      .END

```

USER DOCUMENTATION ....B1  
 USER DOCUMENTATION ....C1  
 USER DOCUMENTATION ....D1  
 USER DOCUMENTATION ....E1  
 USER DOCUMENTATION ....F1  
 USER DOCUMENTATION ....G1  
 USER DOCUMENTATION ....H1  
 USER DOCUMENTATION ....I1  
 USER DOCUMENTATION ....J1  
 USER DOCUMENTATION ....K1  
 USER DOCUMENTATION ....L1  
 USER DOCUMENTATION ....M1  
 USER DOCUMENTATION ....N1

SPACE - SPACE RECO....B5  
 SPACE - SPACE RECO....C5  
 SPACE - SPACE RECO....D5  
 WRCHR - WRITE CHAR....E5  
 REWIND - POSITION T....F5  
 CKRAM - COMPARE RA....G5  
 RAMER - READ AND DIS....H5  
 RAMER - READ AND DIS....I5  
 CKRAM2 - COMPARE RA....J5  
 CKMSG - COMPARE WR....K5  
 CKMSG2 - COMPARE EX....L5  
 CKMSG2 - COMPARE EX....M5  
 CKMSG2 - COMPARE EX....N5

CHKMAN - CHECK MANU....B9  
 ENVIRN - SETUP FREE....C9  
 KINIT - SETUP KT11....D9  
 PROTECTION TABLE ....E9  
 INITIALIZE SECTION ....F9  
 INITIALIZE SECTION ....G9  
 INITIALIZE SECTION ....H9  
 ADD AND DROP UNITS S....I9  
 ADD AND DROP UNITS S....J9  
 CLEAN-UP AND REPORT ....K9  
 CLEAN-UP AND REPORT ....L9  
 TEST 1: INITIALIZE ....M9  
 TEST 1: INITIALIZE ....N9

TEST 4: BASIC READ ....B13  
 TEST 4: BASIC READ ....C13  
 TEST 4: BASIC READ ....D13  
 TEST 4: BASIC READ ....E13  
 TEST 4: BASIC READ ....F13  
 TEST 4: BASIC READ ....G13  
 TEST 4: BASIC READ ....H13  
 TEST 4: BASIC READ ....I13  
 TEST 4: BASIC READ ....J13  
 TEST 4: BASIC READ ....K13  
 TEST 4: BASIC READ ....L13  
 TEST 4: BASIC READ ....M13  
 TEST 4: BASIC READ ....N13

USER DOCUMENTATION ....B2  
 USER DOCUMENTATION ....C2  
 USER DOCUMENTATION ....D2  
 USER DOCUMENTATION ....E2  
 USER DOCUMENTATION ....F2  
 USER DOCUMENTATION ....G2  
 USER DOCUMENTATION ....H2  
 USER DOCUMENTATION ....I2  
 PROGRAM HEADER ....J2  
 DEFAULT HARDWARE P-T....K2  
 SOFTWARE P-TABLE ....L2  
 SOFTWARE P-TABLE ....M2  
 GLOBAL EQUATES SECTI....N2

CKMSG2 - COMPARE EX....B6  
 ADDSSR - PRINT TEST....C6  
 FIFEXP - PRINT FIFO....D6  
 MSGSTAT - PRINT STAT....E6  
 MSGLOOP - PRINT LOOP....F6  
 MSGSUB - PRINT WRITE....G6  
 PRAMPKT - PRINT RAM ....H6  
 PRMESS - PRINT CONT....I6  
 PRMESS - PRINT CONT....J6  
 PRMESS - PRINT CONT....K6  
 PRMSGEXP - PRINT EXP....L6  
 PRMSGEXP - PRINT EXP....M6  
 PRBYTEXP - PRINT ERR....N6

TEST 1: INITIALIZE ....B10  
 TEST 1: INITIALIZE ....C10  
 TEST 2: OFF-LINE AN....D10  
 TEST 2: OFF-LINE AN....E10  
 TEST 2: OFF-LINE AN....F10  
 TEST 2: OFF-LINE AN....G10  
 TEST 2: OFF-LINE AN....H10  
 TEST 3: BASIC WRITE....I10  
 TEST 3: BASIC WRITE....J10  
 TEST 3: BASIC WRITE....K10  
 TEST 3: BASIC WRITE....L10  
 TEST 3: BASIC WRITE....M10  
 TEST 3: BASIC WRITE....N10

TEST 4: BASIC READ ....B14  
 TEST 4: BASIC READ ....C14  
 TEST 4: BASIC READ ....D14  
 TEST 4: BASIC READ ....E14  
 TEST 4: BASIC READ ....F14  
 TEST 4: BASIC READ ....G14  
 TEST 4: BASIC READ ....H14  
 TEST 4: BASIC READ ....I14  
 TEST 4: BASIC READ ....J14  
 TEST 4: BASIC READ ....K14  
 TEST 4: BASIC READ ....L14  
 TEST 4: BASIC READ ....M14  
 TEST 4: BASIC READ ....N14

MEMORY MANAGEMENT DE....B3  
 MEMORY MANAGEMENT DE....C3  
 TK-25 REGISTER AND P....D3  
 TK-25 REGISTER AND P....E3  
 TK-25 REGISTER AND P....F3  
 TK-25 REGISTER AND P....G3  
 TK-25 REGISTER AND P....H3  
 TK-25 REGISTER AND P....I3  
 TK-25 REGISTER AND P....J3  
 SPECIAL MACROS AND O....K3  
 SPECIAL MACROS AND O....L3  
 GLOBAL DATA SECTION ....M3  
 TSTBLK - TEST DATA ....N3

PRBYTEXP - PRINT ERR....B7  
 EXPREC - PRINT EXPD....C7  
 EXPBREC - PRINT EXPD....D7  
 RAMTADD - PRINT TEST....E7  
 TIMEXP - PRINT TIME....F7  
 BACSSR - PRINT TSSR....G7  
 GLOBAL SUBROUTINES S....H7  
 CHKAMB - CHECK TSSR....I7  
 ENAINT,DSBINT - ENAB....J7  
 INTR - INTERRUPT ....K7  
 WAITF - WAIT FOR S....L7  
 CHKTTSSR - CHECK TSSR....M7  
 XNXM - CHECK FOR ....N7

TEST 3: BASIC WRITE....B11  
 TEST 3: BASIC WRITE....C11  
 TEST 3: BASIC WRITE....D11  
 TEST 3: BASIC WRITE....E11  
 TEST 3: BASIC WRITE....F11  
 TEST 3: BASIC WRITE....G11  
 TEST 3: BASIC WRITE....H11  
 TEST 3: BASIC WRITE....I11  
 TEST 3: BASIC WRITE....J11  
 TEST 3: BASIC WRITE....K11  
 TEST 3: BASIC WRITE....L11  
 TEST 4: BASIC READ ....M11  
 TEST 4: BASIC READ ....N11

TEST 4: BASIC READ ....B15  
 TEST 4: BASIC READ ....C15  
 TEST 4: BASIC READ ....D15  
 TEST 5: MANUAL INTER....E15  
 TEST 5: MANUAL INTER....F15  
 TEST 5: MANUAL INTER....G15  
 TEST 5: MANUAL INTER....H15  
 TEST 5: MANUAL INTER....I15  
 TEST 5: MANUAL INTER....J15  
 TEST 5: MANUAL INTER....K15  
 TEST 5: MANUAL INTER....L15  
 TEST 5: MANUAL INTER....M15  
 TEST 5: MANUAL INTER....N15

GLOBAL ENVIRONMENT S....B4  
 GLOBAL TEXT MESSAGES....C4  
 GLOBAL TEXT MESSAGES....D4  
 GLOBAL ERROR REPORT ....E4  
 PRITSSR - PRINT TSSR....F4  
 PRITSSR - PRINT TSSR....G4  
 PRITSSR - PRINT TSSR....H4  
 PRIPKT - PRINT THE ....I4  
 PRIPKT - PRINT THE ....J4  
 PRIBXOR - PRINT EXPD....K4  
 PRIBXOR - PRINT EXPD....L4  
 PRIQU - PRINT BIT ....M4  
 PRIADD - PRINT MEMO....N4

TSTLOOP - CHECK ITER....B8  
 TSTSETUP - PRINT TES....C8  
 TSTEND - PRINT ERRO....D8  
 INCERK - INCREMENT ....E8  
 CKDROP - CHECK IF U....F8  
 KTON,KTOFF - EN....G8  
 SETMAP - SETUP PAR6....H8  
 FILLMEM - FILL MEMOR....I8  
 CMPMEM - COMPARE ME....J8  
 CMPMEM - COMPARE ME....K8  
 REGSAV - SAVE R1-R5....L8  
 GETPAT - GET 3 BIT ....M8  
 GETSEL - ISSUE MENU....N8

TEST 4: BASIC READ ....B12  
 TEST 4: BASIC READ ....C12  
 TEST 4: BASIC READ ....D12  
 TEST 4: BASIC READ ....E12  
 TEST 4: BASIC READ ....F12  
 TEST 4: BASIC READ ....G12  
 TEST 4: BASIC READ ....H12  
 TEST 4: BASIC READ ....I12  
 TEST 4: BASIC READ ....J12  
 TEST 4: BASIC READ ....K12  
 TEST 4: BASIC READ ....L12  
 TEST 4: BASIC READ ....M12  
 TEST 4: BASIC READ ....N12

TEST 6: CONFIGURATIO....B16  
 TEST 6: CONFIGURATIO....C16  
 TEST 6: CONFIGURATIO....D16  
 TEST 6: CONFIGURATIO....E16  
 TEST 6: CONFIGURATIO....F16  
 TEST 7: SCOPE LOOPS ....G16  
 TEST 7: SCOPE LOOPS ....H16  
 TEST 7: SCOPE LOOPS ....I16  
 TEST 7: SCOPE LOOPS ....J16  
 HARDWARE PARAMETER C....K16  
 SOFTWARE PARAMETER C....L16  
 PATCH AREA ....M16

|        |        |        |       |        |        |        |        |        |        |        |        |        |        |        |        |        |   |  |
|--------|--------|--------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---|--|
| ADDSSR | 011772 | G      | CIAU  | =      | 000052 | DEBUG  | 011464 | FATFLG | 002170 | G      | HIMEM  | =      | 007776 |        |        |        |   |  |
| ADR    | =      | 000020 | G     | CIAUTO | =      | 000061 | DEVcnt | 002166 | G      | FERCH  | 011554 | MOE    | =      | 100000 | G      |        |   |  |
| AMBTSS | 006337 |        | CIBRK | =      | 000022 | DEVDR0 | 023564 | FIFEXP | 012022 | G      | HPM1   | 053120 |        |        |        |        |   |  |
| ASSEMB | =      | 000010 |       | CIBSEG | =      | 000004 | DEVNR0 | 023503 | FIF1MS | 012074 | HPM2   | 053147 |        |        |        |        |   |  |
| A1716  | =      | 000003 |       | CIBSUB | =      | 000002 | DEVNKR | 023421 | FIF2MS | 012143 | HPM3   | 053173 |        |        |        |        |   |  |
| BAD0AT | 003110 | G      |       | CICEFG | =      | 000045 | DEVONL | 023351 | FILLME | 020374 | IBE    | =      | 010000 | G      |        |        |   |  |
| BADSSR | 016550 | G      |       | CICLCK | =      | 000062 | DEVSUM | 023314 | FNOINT | 004113 | IDU    | =      | 000040 | G      |        |        |   |  |
| BAR    | =      | 174402 |       | CICLEA | =      | 000012 | JFPTBL | 002124 | G      | FORCER | 002144 | IER    | =      | 020000 | G      |        |   |  |
| BENBSW | 002174 | G      |       | CICLOS | =      | 000035 | DIAGMC | =      | 000000 | G      | FREE   | 003072 | IFALT  | 004154 |        |        |   |  |
| BIE    | =      | 040000 |       | CICLP1 | =      | 000006 | DLCYL  | =      | 000177 | G      | FREEH  | 003076 | INCERK | 017744 |        |        |   |  |
| BIT0   | =      | 000001 | G     | CICVEC | =      | 000036 | DLNER  | =      | 100200 | G      | FRESIZ | 003074 | INTCPC | 017020 |        |        |   |  |
| BIT00  | =      | 000001 | G     | CIDCLN | =      | 000044 | DLERR  | =      | 177730 | G      | FUSI   | 004015 | INTFLA | 017015 |        |        |   |  |
| BIT01  | =      | 000002 | G     | CIDODU | =      | 000051 | DLGETS | =      | 000004 | G      | FIAU   | =      | 000015 | INTMAS | 017014 |        |   |  |
| BIT02  | =      | 000004 | G     | CIDRPT | =      | 000024 | DLRDND | =      | 000010 | G      | FIAUTO | =      | 000020 | INTR   | 017066 | G      |   |  |
| BIT03  | =      | 000010 | G     | CIDU   | =      | 000053 | DLRDND | =      | 000016 | G      | FIBGN  | =      | 000040 | INTREC | 002172 | G      |   |  |
| BIT04  | =      | 000020 | G     | CIEDIT | =      | 000003 | DLR    | =      | 000013 | G      | FICLEA | =      | 000007 | INTVEC | 017016 |        |   |  |
| BIT05  | =      | 000040 | G     | CIERDF | =      | 000055 | DLUN   | =      | 000006 | G      | FIDU   | =      | 000016 | INTX   | 004176 |        |   |  |
| BIT06  | =      | 000100 | G     | CIERNR | =      | 000056 | DSBINT | 017054 | FEND   | =      | 000041 | IOCKI  | =      | 000200 |        |        |   |  |
| BIT07  | =      | 000200 | G     | CIERRO | =      | 000060 | DUAD12 | 004541 | FHARD  | =      | 000004 | IOKSTP | =      | 000001 |        |        |   |  |
| BIT08  | =      | 000400 | G     | CIERSF | =      | 000054 | DUFLG  | 003060 | G      | FIMW   | =      | 000013 | IPRI   | 002160 | G      |        |   |  |
| BIT09  | =      | 001000 | G     | CIERSO | =      | 000057 | DUMMY  | 003030 | FINIT  | =      | 000006 | ISR    | =      | 000100 | G      |        |   |  |
| BIT1   | =      | 000002 | G     | CIESCA | =      | 000010 | EF.CON | =      | 000036 | G      | FJMP   | =      | 000050 | IVEC   | 002156 | G      |   |  |
| BIT10  | =      | 002000 | G     | CIESEG | =      | 000005 | EF.PWR | =      | 000035 | G      | FIMOD  | =      | 000000 | IXE    | =      | 004000 | G |  |
| BIT11  | =      | 004000 | G     | CIESUB | =      | 000003 | EF.NEW | =      | 000034 | G      | FIMSG  | =      | 000011 | IAU    | =      | 000041 |   |  |
| BIT12  | =      | 010000 | G     | CJETST | =      | 000001 | EF.RES | =      | 000037 | G      | FIPROT | =      | 000021 | IAUTO  | =      | 000041 |   |  |
| BIT13  | =      | 020000 | G     | CJEXIT | =      | 000032 | EF.STA | =      | 000040 | G      | FIPWR  | =      | 000017 | IICLN  | =      | 000041 |   |  |
| BIT14  | =      | 040000 | G     | CJGETB | =      | 000026 | EMAXDU | 017677 | FIRPT  | =      | 000012 | IDU    | =      | 000041 |        |        |   |  |
| BIT15  | =      | 100000 | G     | CJGETW | =      | 000027 | EN     | =      | 000000 | G      | FISEG  | =      | 000003 | IHRD   | =      | 000041 |   |  |
| BIT2   | =      | 000004 | G     | CJGMAN | =      | 000043 | ENAIN  | 017022 | FISOFT | =      | 000005 | IINIT  | =      | 000041 |        |        |   |  |
| BIT3   | =      | 000010 | G     | CJGPHR | =      | 000042 | ENVIRN | 021534 | FISRV  | =      | 000010 | IIMOD  | =      | 000040 |        |        |   |  |
| BIT4   | =      | 000020 | G     | CJGPLO | =      | 000030 | EPRTSW | 002146 | FISUB  | =      | 000002 | IIMSG  | =      | 000041 |        |        |   |  |
| BIT5   | =      | 000040 | G     | CJGPRI | =      | 000040 | EPRT1  | 005762 | FISW   | =      | 000014 | IIPROT | =      | 000040 |        |        |   |  |
| BIT6   | =      | 000100 | G     | CJINIT | =      | 000011 | EPRT2  | 005672 | FITEST | =      | 000001 | IIPTAB | =      | 000041 |        |        |   |  |
| BIT7   | =      | 000200 | G     | CJINLP | =      | 000020 | EPRT3  | 006026 | GDDAT  | =      | 003112 | IIPWR  | =      | 000041 |        |        |   |  |
| BIT8   | =      | 000400 | G     | CJMANI | =      | 000050 | ERRC   | 011565 | GERRMA | 002142 | G      | IIRPT  | =      | 000041 |        |        |   |  |
| BIT9   | =      | 001000 | G     | CJMEM  | =      | 000031 | ERRHI  | 002202 | G      | GETPAT | 021100 | G      | IISEG  | =      | 000041 |        |   |  |
| BOE    | =      | 000400 | G     | CJMSG  | =      | 000023 | ERRK   | 017656 | G      | GETSEL | 021162 | G      | IISetu | =      | 000041 |        |   |  |
| BRINIT | 004355 |        |       | CJOPEN | =      | 000034 | ERRLO  | 002204 | G      | GICNT0 | =      | 000200 | IISFT  | =      | 000041 |        |   |  |
| BSELO  | =      | 000000 |       | CJPNTF | =      | 000014 | ERRNO  | =      | 001276 | G      | GIDELM | =      | 000372 | IISRV  | =      | 000041 |   |  |
| BSEL1  | =      | 000001 |       | CJPNTF | =      | 000017 | ERRVEC | =      | 000004 | G      | GIDISP | =      | 000003 | IISUB  | =      | 000041 |   |  |
| CHKAMB | 016714 |        |       | CJPNTS | =      | 000016 | ERTABE | 003330 | G      | GEXCP  | =      | 000000 | IITST  | =      | 000041 |        |   |  |
| CHKMAN | 021404 | G      |       | CJPNTX | =      | 000015 | ERTABL | 003130 | G      | GIMILI | =      | 000002 | JUMP   | =      | 000167 |        |   |  |
| CHKTSS | 017236 |        |       | CJQIO  | =      | 000377 | ESUM   | 017660 | G      | GILOLI | =      | 000001 | KIPAR0 | =      | 172340 |        |   |  |
| CKDROP | 020154 |        |       | CJROBU | =      | 000007 | EVL    | =      | 000004 | G      | GINO   | =      | 000000 | KIPAR1 | =      | 172342 |   |  |
| CKEMAX | 020002 |        |       | CJREFG | =      | 000047 | EXBCNT | =      | 000010 | G      | GIOFFS | =      | 000400 | KIPAR2 | =      | 172344 |   |  |
| CKMSG  | 011212 | G      |       | CJRESE | =      | 000033 | EXPBRE | 016352 | G      | GIOFSI | =      | 000376 | KIPAR3 | =      | 172346 |        |   |  |
| CKMSG2 | 011332 | G      |       | CJREVI | =      | 000003 | EXP0   | 002176 | G      | GIPRMA | =      | 000001 | KIPAR4 | =      | 172350 |        |   |  |
| CKRAM  | 010534 | G      |       | CJRFIA | =      | 000021 | EXPGOT | 004431 | G      | GIPRMD | =      | 000002 | KIPAR5 | =      | 172352 |        |   |  |
| CKRAM2 | 011110 | G      |       | CJRPT  | =      | 000025 | EXPGT2 | 004465 | G      | GIPRML | =      | 000000 | KIPAR6 | =      | 172354 |        |   |  |
| CMEMEM | 020560 |        |       | CJSEFG | =      | 000046 | EXPMG  | 002266 | G      | GIRADA | =      | 000140 | KIPAR7 | =      | 172356 |        |   |  |
| CONFIG | 020222 |        |       | CJSPRI | =      | 000041 | EXPREC | 016344 | G      | GIRADB | =      | 000000 | KIPDR0 | =      | 172300 |        |   |  |
| COUNT  | 002254 | G      |       | CJSVEC | =      | 000037 | EXTA   | 005232 | G      | GIRADD | =      | 000040 | KIPDR1 | =      | 172302 |        |   |  |
| CSR    | =      | 174400 |       | CJTPRI | =      | 000013 | EXTEND | 005230 | G      | GIRADL | =      | 000120 | KIPDR2 | =      | 172304 |        |   |  |
| CSRADD | 002154 | G      |       | DAR    | =      | 174404 | EEND   | =      | 002100 | G      | GIRADO | =      | 000020 | KIPDR3 | =      | 172306 |   |  |
| CTAB   | 003116 | G      |       | DATA   | =      | 002256 | ELOAD  | =      | 000035 | G      | GIXFER | =      | 000004 | KIPDR4 | =      | 172310 |   |  |
| CTABE  | 003130 | G      |       | DATAFL | =      | 015064 | FATCHK | 020102 | G      | GYES   | =      | 000010 | KIPDR5 | =      | 172312 |        |   |  |
| CTABM  | 003116 | G      |       | DATASC | =      | 021136 | FATERR | =      | 000060 | G      | HIADDR | =      | 001400 | KIPDR6 | =      | 172314 |   |  |

C1

CZTKFA TK-25 FRT END FUNC #2    MACRO M1200    20-APR-84 08:12    PAGE 121-3

SEQ 209

|                  |                  |                 |                  |                 |
|------------------|------------------|-----------------|------------------|-----------------|
| KIPDR7= 172316   | L\$SOFT 053226 G | L10061 037544   | O1BGNR= 000001   | PROASC 014714   |
| KTENAB 003102 G  | L\$SPC 002056 G  | L10062 040034   | O1BGNS= 000001   | PRIASC 014761   |
| KTFLG 003100 G   | L\$SPCP 002020 G | L10063 040510   | O1DU = 000001    | PST32W 003104 G |
| KTINIT 021622    | L\$SPTP 002024 G | L10064 047554   | O1ERRT= 000000   | PUNIT 022476    |
| KTOFF 020246     | L\$STA 002030 G  | L10065 051476   | O1GNSW= 000001   | PW.D11= 000021  |
| KTON 020230      | L\$SW 002134 G   | L10066 053062   | O1POIN= 000001   | PW.D13= 000022  |
| LERRMA 002140 G  | L\$TEST 002114 G | L10067 053120   | O1SETU= 000001   | PW.D22= 000020  |
| LISTAL= 000001   | L\$TIML 002014 G | L10070 053242   | PASRPT 022242    | PW.NOP= 000000  |
| LOE = 040000 G   | L\$UNIT 002012 G | L10071 053446   | PATCH 053436 G   | PW.NO1= 000023  |
| LOOPCN 002164 G  | L10000 002132    | L10073 053454   | PATDAT 071134    | PW.RDE= 000024  |
| LOOPCO 012760    | L10001 002144    | MENADD 013606 G | PC.ERA= 002400   | PW.RDR= 000001  |
| LOOPFL 003114 G  | L10002 005226    | MENASC 021353   | PC.IER= 002000   | PW.RDS= 000005  |
| LOT = 000010 G   | L10003 011676    | MENERR 021300   | PC.NOO= 001000   | PW.RFI= 000003  |
| L\$ACP 002110 G  | L10004 011726    | MENRES 021402   | PC.REL= 000000   | PW.WCT= 000006  |
| L\$APT 002036 G  | L10005 011744    | MESBFA 002716 G | PC.REW= 00C400   | PW.WFI= 000004  |
| L\$AU 022544 G   | L10006 011752    | MESBFN 014634   | PKBCNT= 000006   | PW.WFM= 000007  |
| L\$AUT 002070 G  | L10007 011770    | MESHEA 015017   | PKHI = 000004    | PW.WMI= 000010  |
| L\$AUTO 002750 G | L10010 012006    | MIMENJ 046512   | PKLOW = 000002   | PW.WNP= 000011  |
| L\$CCP 002106 G  | L10011 012020    | MMVEC = 000250  | PKTADD 007276    | PW.WTR= 000002  |
| L\$CLEA 023024 G | L10012 012072    | MPR = 174406    | PKTFRM 007240    | P.ACK = 100000  |
| L\$CO 002032 G   | L10013 012242    | MSA.FR= 000006  | PKTGET 011730 G  | P.CMD = 000037  |
| L\$DEPO 002011 G | L10014 012756    | MSA.NO= 000000  | PKTMES 011754 G  | P.CONT= 000012  |
| L\$DESC 003342 G | L10015 013604    | MSA.NR= 000004  | PKTNEW 007333    | P.CVC = 040000  |
| L\$DESP 002076 G | L10016 013626    | MSA.VO= 000002  | PKTRAM 004643 G  | P.FMT = 000140  |
| L\$DEVP 002060 G | L10017 016350    | MSGEXP 012010 G | PKTSSR 011700 G  | P.FORM= 000011  |
| L\$DISP 053420 G | L10020 016356    | MSGLOO 012716 G | PNT = 001000 G   | P.GETS= 000017  |
| L\$DLY 002116 G  | L10021 016364    | MSGSTA 012202 G | PRAMPK 013630    | P.IE = 000200   |
| L\$DTP 002040 G  | L10022 016376    | MSGSUB 013574 G | PRBEXP 016340    | P.INIT= 000013  |
| L\$DTYP 002034 G | L10023 016420    | MS.ATT= 000006  | PRBMSG 016206    | P.MODE= 007400  |
| L\$DU 022642 G   | L10024 016446    | MS.EXT= 000200  | PRBREC 016342    | P.OPP = 020000  |
| L\$DUT 002072 G  | L10025 016606    | MS.RSD= 000001  | PRBTOT 016273    | P.POSI= 000010  |
| L\$DVTY 003334 G | L10026 017116    | MS.RSF= 000020  | PRBYTE 015772 G  | P.READ= 000001  |
| L\$EF 002052 G   | L10030 022474    | MS.RST= 000010  | PRI = 002000 G   | P.SWB = 010000  |
| L\$ENVI 002044 G | L10031 022640    | NBA = 002000    | PRIADD 007712    | P.WRIT= 000005  |
| L\$ETP 002102 G  | L10032 022746    | NEWPAS 022176   | PRIAO 007762     | P.WRTC= 000004  |
| L\$EXP1 002046 G | L10033 023022    | NODEV 003062 G  | PRIBX0 007344 G  | P.WRTS= 000006  |
| L\$EXP4 002064 G | L10034 023050    | NOINIT 004233   | PRIEQU 007612    | QVP 002152 G    |
| L\$EXP5 002066 G | L10035 023312    | NOINTR 004117   | PRIPKT 007072 G  | RAMASC 013776   |
| L\$HARD 053066 G | L10036 024672    | NOITS 002136 G  | PRIRAM 007620    | RAMDAT 002206 G |
| L\$HIME 002120 G | L10037 026224    | NOMAN 021440    | PRITAD 010026    | RAMER 010636 G  |
| L\$HPCP 002016 G | L10040 025134    | NP.IR = 000200  | PRITSS 005264    | RAMERR 016360 G |
| L\$HPTP 002022 G | L10041 032156    | NP.LOO= 000040  | PRITO 010076     | RAMEXP 016400 G |
| L\$HW 002124 G   | L10042 026624    | NP.OUT= 000100  | PRI XOR 007474 G | RAMFHR 014536   |
| L\$ICP 002104 G  | L10043 027302    | NP.WRP= 000020  | PRI00 = 000000 G | RAMFOR 007650   |
| L\$INIT 021764 G | L10044 027510    | NSI 004050      | PRI01 = 000040 G | RAMHLD 011020   |
| L\$LADP 002026 G | L10045 027776    | NSINIT 004305   | PRI02 = 000100 G | RAMIOP 011024   |
| L\$LAST 053442 G | L10046 030320    | NUL 004425      | PRI03 = 000140 G | RAMPD 011075    |
| L\$LOAD 002100 G | L10047 043424    | NULCR 004426    | PRI04 = 000200 G | RAMRSH 011022   |
| L\$LUN 002074 G  | L10050 032776    | NXM = 004000    | PRI05 = 000240 G | RAMSIZ 002246 G |
| L\$PREV 002050 G | L10051 033544    | NXR 003636      | PRI06 = 000300 G | RAMTAD 016366 G |
| L\$NAME 002000 G | L10052 034240    | NXRERR 005176 G | PRI07 = 000340 G | RBPCRA 015131   |
| L\$PRIO 002042 G | L10053 034676    | NXRXX 003675    | PRMESS 014062    | RCVHIA 002250 G |
| L\$PROT 021754 G | L10054 035324    | NXTU 022210     | PRMNO 002264 G   | RCVLOA 002252 G |
| L\$PRT 002112 G  | L10055 035752    | O L = 000100    | PRMSGE 015422 G  | RDERR 005104    |
| L\$REPP 002062 G | L10056 036334    | ONEFIL= 000000  | PRMSG0 015602    | READ = 000014   |
| L\$REV 002010 G  | L10057 037010    | O1APTS= 000000  | PRMSG1 015647    | READY = 000001  |
| L\$RPT 023052 G  | L10060 037252    | O1AU = 000001   | PRMSG2 015705    | RECMG 002432 G  |

D1

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 121-4

SEQ 210

|         |        |   |        |        |        |        |        |        |        |        |
|---------|--------|---|--------|--------|--------|--------|--------|--------|--------|--------|
| RECV    | 002200 | G | S1.ICE | 002000 | TST21I | 024516 | T#PTA  | 010072 | T23ET  | 030573 |
| REGSAV  | 021040 |   | S1.IEO | 010000 | TST22I | 026032 | T#RPT  | 010035 | T23L00 | 026276 |
| REWIND  | 010434 | G | S1.IFM | 001000 | TST23I | 031761 | T#SOF  | 010070 | T23OFL | 031301 |
| RMCHBE  | 000167 |   | S1.IHE | 000400 | TST24I | 043172 | T#SRV  | 010026 | T23PAC | 030360 |
| RMCHEN  | 000200 |   | S1.IID | 004000 | TST39I | 051450 | T#SUB  | 010063 | T23PK2 | 030470 |
| RMMSGB  | 000104 |   | S1.IIR | 020000 | TST40I | 052756 | T#SW   | 010001 | T23PK3 | 030510 |
| RMMSGE  | 000117 |   | S1.I2R | 040000 | TTIBFR | 177562 | T#TES  | 010066 | T23RES | 031776 |
| RMPKTB  | 000020 |   | S1.PAR | 100000 | TTICSR | 177560 | T1     | 023634 | T23RNC | 031160 |
| RMPKTE  | 000027 |   | S2.ATI | 000010 | TTION  | 052200 | T2     | 024674 | T23RSZ | 030520 |
| RMR     | 010000 |   | S2.BTI | 000004 | TTION2 | 044604 | T2.1   | 024734 | T23RT2 | 032070 |
| RWPACK  | 010530 |   | S2.DIM | 000200 | TTIVEC | 000060 | T21AM3 | 024375 | T23RT3 | 032132 |
| SC      | 100000 |   | S2.ILW | 000100 | TTIBFR | 177566 | T21BFR | 024170 | T23RWN | 031111 |
| SCE     | 020000 |   | S2.INR | 000020 | TTICSR | 177564 | T21BF2 | 024270 | T23SSR | 030540 |
| SCME    | 004711 |   | S2.OUT | 000040 | TUV2A  | 002000 | T21BS0 | 024270 | T23SZ  | 030516 |
| SCMENU  | 052206 |   | S2.UND | 000003 | TVECSA | 052202 | T21BS1 | 024271 | T23S2  | 030524 |
| SDELAY  | 010330 |   | TBLEND | 003030 | TVSAV2 | 044606 | T21DAT | 024160 | T23S3  | 030526 |
| SEEK    | 000006 |   | TCOASC | 006200 | T#ARGC | 000001 | T21DLY | 024276 | T23TM  | 030736 |
| SELASC  | 021346 |   | TCOCOD | 006400 | T#CODE | 001130 | T21L00 | 023674 | T23TMA | 031025 |
| SELDAT  | 000004 |   | TEMP1  | 003064 | T#ERRN | 001276 | T21OFL | 024475 | T23VCK | 031545 |
| SEL2    | 000002 |   | TEMP2  | 003066 | T#EXCP | 000000 | T21PAC | 024150 | T23WB  | 030512 |
| SETHAP  | 020270 |   | TERCLS | 000016 | T#FLAG | 000040 | T21PK2 | 024260 | T23WD  | 030530 |
| SETU    | 022274 |   | TESTNO | 000007 | T#FREE | 053454 | T21RES | 024540 | T23WDC | 031443 |
| SFFMSG  | 011746 | G | TEXASC | 006137 | T#GMAN | 000000 | T21RT2 | 024630 | T23WDO | 031354 |
| SFHERR  | 003603 |   | TFCASC | 006241 | T#HILI | 000007 | T21SSR | 024300 | T23WDR | 030532 |
| SFIERR  | 003550 |   | TIMEXP | 016422 | T#LAST | 000001 | T21S2  | 024272 | T23WSS | 031672 |
| SFIMSG  | 011666 | G | YIMSGO | 016450 | T#LOLI | 000000 | T21S3  | 024274 | T24AM3 | 042160 |
| SFPTBL  | 002134 | G | TINERR | 011653 | T#LSYM | 010000 | T22AM3 | 025435 | T24BA  | 042512 |
| SIFLAG  | 003106 | G | TKB    | 177562 | T#LTNO | 000007 | T22BFR | 025222 | T24BFR | 040570 |
| SIMSG   | 011620 |   | TKS    | 177560 | T#NEST | 000000 | T22BF2 | 025320 | T24BF2 | 040710 |
| SKIP    | 030320 |   | TMPBFR | 002576 | T#NSC  | 000000 | T22BS0 | 025320 | T24BOT | 041553 |
| SKIPT   | 003332 |   | TNAM   | 017604 | T#NS1  | 000005 | T22BS1 | 025321 | T24BS0 | 040710 |
| SOFINI  | 016644 | G | TPB    | 177566 | T#NS2  | 000002 | T22DAT | 025210 | T24BS1 | 040711 |
| SPACE   | 010140 | G | TPRISA | 052204 | T#PCNT | 000000 | T22FOR | 025334 | T24CON | 040722 |
| SPM1    | 053242 |   | TPS    | 177564 | T#PTAB | 010072 | T22L00 | 024734 | T24DAT | 040560 |
| SPM4    | 053306 |   | TPSAV2 | 044610 | T#PTHV | 000001 | T22OFL | 025535 | T24DLY | 040726 |
| SPM6    | 053336 |   | TRANST | 002134 | T#PTNU | 000001 | T22PAC | 025200 | T24DTA | 041620 |
| SPM7    | 053366 |   | TSBA   | 177776 | T#SAVL | 177777 | T22PK2 | 025310 | T24EOT | 041706 |
| SRO     | 177572 |   | TSBAH  | 177777 | T#SEGL | 177777 | T22POS | 025332 | T24ILA | 041302 |
| SR1     | 177574 |   | TSBAL  | 177776 | T#SIZE | 000005 | T22RD  | 025326 | T24LON | 042652 |
| SR2     | 177570 |   | TSDB   | 177776 | T#SUBN | 000000 | T22RES | 026066 | T24LOO | 032224 |
| SR3     | 172516 |   | TSDBH  | 177777 | T#TAGL | 177777 | T22R12 | 026100 | T24LOP | 042734 |
| SSR     | 000200 |   | TSDBL  | 177776 | T#TAGN | 010074 | T22RWJ | 025704 | T24LOQ | 041366 |
| STATCO  | 012244 |   | TSFCOD | 006740 | T#TEMP | 000010 | T22SSR | 025340 | T24LOR | 041002 |
| SVCGBL  | 000000 |   | TSREJ  | 000006 | T#TEST | 000007 | T22S2  | 025322 | T24NEF | 040730 |
| SVCINS  | 000000 |   | TSSDEF | 006310 | T#TSTM | 177777 | T22S3  | 025324 | T24NXM | 041141 |
| SVCSUB  | 000001 |   | TSSR   | 000000 | T#TSTS | 000001 | T22TM  | 025610 | T24OFL | 042225 |
| SVCTAG  | 000000 |   | TSSRBI | 003400 | T#AU   | 010031 | T22VCK | 025757 | T24PAC | 040550 |
| SVCTST  | 000001 |   | TSSRFO | 006117 | T#AUT  | 010033 | T22WRT | 025330 | T24PBP | 043016 |
| T#LSYM  | 010000 |   | TSSRM  | 000001 | T#CLE  | 010034 | T23AM3 | 031233 | T24PK2 | 040660 |
| TJ IDB  | 000010 |   | TSSX   | 003716 | T#DAT  | 010073 | T23BA  | 031620 | T24PK3 | 040700 |
| SO. IFB | 000002 |   | TSTBLK | 002720 | T#DU   | 010032 | T23BFR | 030400 | T24RB  | 040702 |
| SO. IFP | 000001 |   | TSTCNT | 002162 | T#HAR  | 010067 | T23BF2 | 030522 | T24RES | 043240 |
| SO. ILD | 000020 |   | TSTEND | 017620 | T#HW   | 010000 | T23BS0 | 030522 | T24RN  | 040716 |
| SO. ION | 000000 |   | TSTFLA | 002260 | T#INI  | 010030 | T23BS1 | 030523 | T24RNC | 042105 |
| SO. IRD | 000000 |   | TSTLOO | 017356 | T#MSG  | 010025 | T23CON | 030534 | T24RT2 | 043332 |
| SO. IRW | 000004 |   | TSTPTR | 002262 | T#PC   | 000001 | T23DAT | 030370 | T24RT3 | 043374 |
| SO. ISP | 000200 |   | TSTSET | 017410 | T#PRO  | 010027 | T23EOT | 030660 | T24RWN | 042036 |

E1

CZTKFA TK-25 FRT END FUNC #2 MACRO M1200 20-APR-84 08:12 PAGE 121-5

SEQ 211

|        |          |        |        |         |          |         |          |         |          |
|--------|----------|--------|--------|---------|----------|---------|----------|---------|----------|
| T24SSR | 041447   | T38MS2 | 046324 | T39PK4  | 050770   | WC.I1T= | 000040   | XSOWLE= | 004000   |
| T24SZ  | 040706   | T38MS4 | 046026 | T39RES  | 051412   | WC.I3R= | 000020   | XSOWLK= | 000004   |
| T24S2  | 040712   | T38MS5 | 046105 | T39RL   | 051406   | WF.IED= | 000010   | XS1CON  | 015243   |
| T24S3  | 040714   | T38MS6 | 046162 | T39SIZ  | 050776   | WF.IER= | 000004   | XS2CON  | 015310   |
| T24TM  | 041763   | T38NE  | 045400 | T39TAD  | 050240   | WF.IHI= | 000200   | XS3CON  | 015355   |
| T24TRL | 043104   | T38OFL | 045772 | T39WPN  | 051244   | WF.IRE= | 000040   | XXCOMM  | 003070 G |
| T24VCL | 042437   | T38ONL | 045676 | T39WR   | 050772   | WF.IWF= | 000020   | X#ALWA= | 000000   |
| T24WB  | 040702   | T38ONM | 045740 | T39WRT  | 051171   | WF.IWR= | 000100   | X#FALS= | 000040   |
| T24WDC | 042366   | T38PAC | 044620 | T4      | 032160 G | WF.I3R= | 000002   | X#OFFS= | 000400   |
| T24WDD | 042300   | T38PK2 | 045320 | T4.1    | 032224   | WF.I4R= | 000001   | X#TRUE= | 000020   |
| T24WDE | 041501   | T38PK3 | 045350 | T4.10   | 037270   | WRCHR   | 010332 G | X1.COR= | 020000   |
| T24WDF | 041225   | T38PK4 | 045370 | T4.11   | 037562   | WRTERR  | 005011   | X1.DLT= | 100000   |
| T24WDG | 041052   | T38RES | 047044 | T4.12   | 040052   | WRTMSG  | 004754   | X1.MBZ= | 017375   |
| T24WDR | 040720   | T38SIZ | 045376 | T4.2    | 033014   | XFERAS  | 016610   | X1.RBP= | 000400   |
| T24WSS | 042563   | T38SST | 046240 | T4.3    | 033562   | XNXM    | 017276   | X1.SPA= | 040000   |
| T3     | 026226 G | T38TAD | 044630 | T4.4    | 034256   | XORBFO  | 007426   | X1.UNC= | 000002   |
| T3.1   | 026276   | T38WLE | 045602 | T4.5    | 034714   | XORFOR  | 007544   | X2.BUF= | 000100   |
| T3.2   | 026642   | T38WOK | 046430 | T4.6    | 035342   | XST0 =  | 000006 G | X2.EXT= | 000200   |
| T3.3   | 027320   | T38WR  | 045372 | T4.7    | 035770   | XST1 =  | 000010 G | X2.OPM= | 100000   |
| T3.4   | 027526   | T38WRL | 045541 | T4.8    | 036352   | XST2 =  | 000012 G | X2.RCE= | 040000   |
| T3.5   | 030014   | T38WRT | 045455 | T4.9    | 037026   | XST3 =  | 000014 G | X2.REV= | 000077   |
| T38ASC | 047507   | T39BFR | 050246 | T4ONE   | 052711   | XST4 =  | 000016 G | X2.SPA= | 035400   |
| T38ASN | 047526   | T39BSO | 050240 | T4ONSR  | 052772   | XS0BOT= | 000002   | X2.UNI= | 000007   |
| T38ASO | 047404   | T39BS1 | 050241 | T5      | 043426 G | XS0CON  | 015176   | X2.WCF= | 002000   |
| T38AS1 | 047451   | T39BS2 | 050242 | T6      | 047556 G | XS0EOT= | 000001   | X3.DCK= | 000010   |
| T38BFR | 044636   | T39DAT | 051410 | T7      | 051500 G | XS0IE = | 000040   | X3.MBZ= | 000006   |
| T38BSO | 044630   | T39DLY | 050224 | UAM =   | 000200 G | XS0ILA= | 000400   | X3.MDE= | 177400   |
| T38BS1 | 044631   | T39DTA | 050740 | UNITN   | 002150 G | XS0ILC= | 001000   | X3.OPI= | 000100   |
| T38BS2 | 044632   | T39EAI | 050746 | UNREC = | 000706   | XS0LET= | 020000   | X3.REV= | 000040   |
| T38CNT | 047552   | T39MCL | 051323 | USI     | 004021   | XS0MOT= | 000200   | X3.RIB= | 000001   |
| T38DAT | 047042   | T39NE  | 051003 | WAITF   | 017120 G | XS0NEF= | 002000   | X3.SPA= | 000200   |
| T38DLY | 044612   | T39NFL | 051000 | WC.IFA= | 000200   | XS0ONL= | 000100   | X3.TRF= | 000020   |
| T38DTA | 045330   | T39OF2 | 051062 | WC.IFE= | 000002   | XS0PED= | 000010   | X4.HSP= | 100000   |
| T38EAI | 045336   | T39ON2 | 051126 | WC.IGO= | 000001   | XS0RLL= | 010000   | X4.MBZ= | 017400   |
| T38EB  | 045312   | T39PAC | 050230 | WC.IRE= | 000010   | XS0RLS= | 040000   | X4.RCE= | 040000   |
| T38ID  | 046466   | T39PK2 | 050730 | WC.IRW= | 000004   | XS0TMK= | 100000   | X4.TSM= | 020000   |
| T38MBP | 047102   | T39PK3 | 050760 | WC.IOT= | 000100   | XS0VCK= | 000020   | X4.WRC= | 000377   |
| T38MSG | 046370   |        |        |         |          |         |          |         |          |

. ABS. 053454 000  
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 29424 WORDS ( 115 PAGES)  
DYNAMIC MEMORY: 20060 WORDS ( 77 PAGES)  
ELAPSED TIME: 00:30:09  
CZTKFA.BIC,CZTKFA/-SP=SVC/ML,CZTKFA

|        |       |        |
|--------|-------|--------|
| SYMBOL | TABLE | ....B1 |
| SYMBOL | TABLE | ....C1 |
| SYMBOL | TABLE | ....D1 |
| SYMBOL | TABLE | ....E1 |