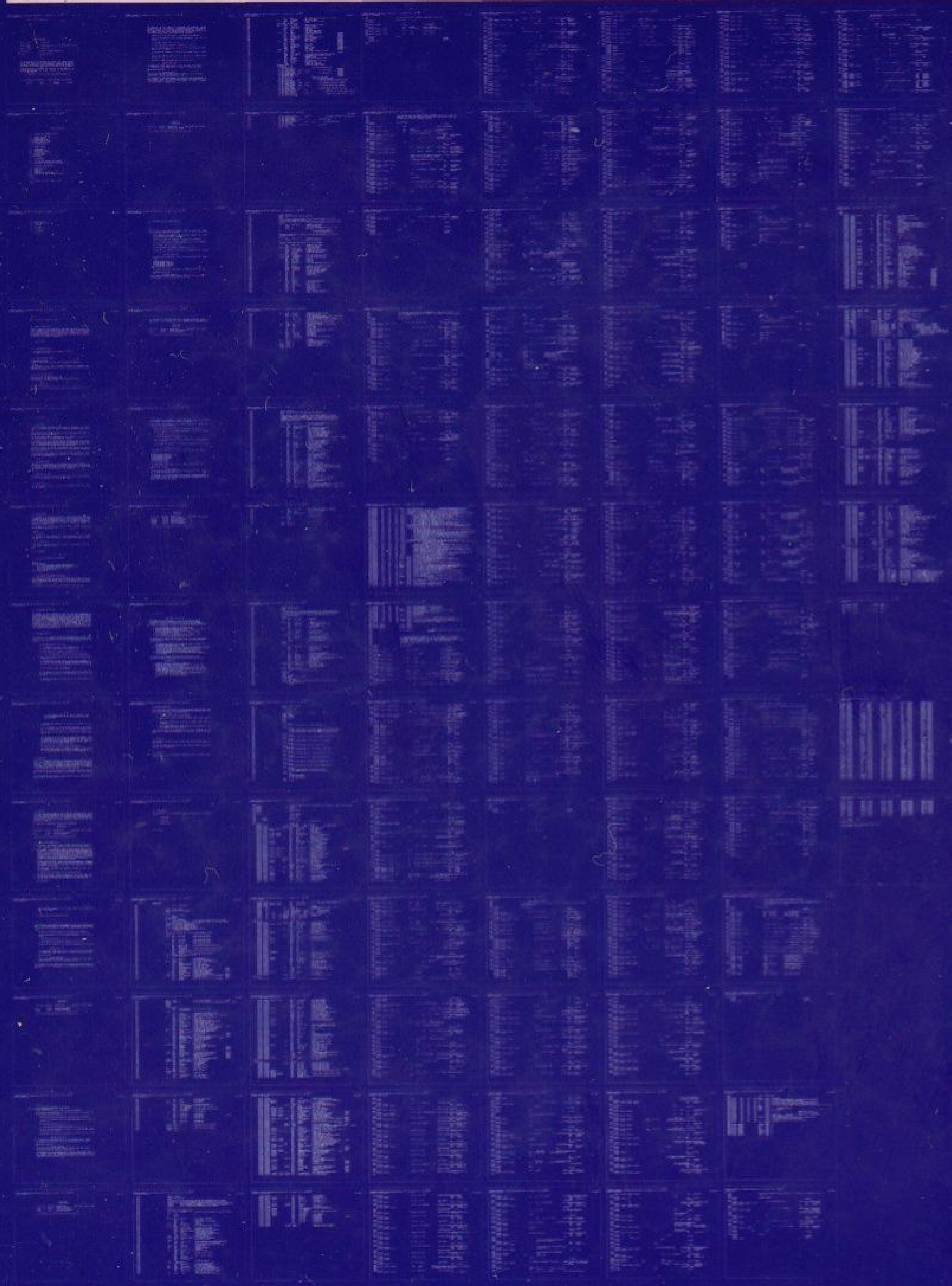
MCV11-DA

DATA RETENTION DIAG
CVMEMA0

AH-S919A-MC
FICHE 1 OF 1

OCT 1981



```
                .SBTTL USER DOCUMENTATION
                                                    .REM &

                    IDENTIFICATION
                    --------------

        PRODUCT CODE:           AC-S918A-MC

        PRODUCT NAME:           CVMEMAO NON-VOLATILE DATA RETENTION DIAGNOSTIC

        PRODUCT DATE:           5-AUGUST-81

        MAINTAINER:             STORAGE SYSTEMS DIAGNOSTICS

        AUTHOR:                 KENNETH LANGLAIS
```

TABLE OF CONTENTS

1.0 OVERVIEW OF DIAGNOSTIC PRODUCT

1.1 PRODUCT DESCRIPTION

MEMORY DIAGNOSTICS WAS CONTRACTED BY TOEM MARKETING GROUP TO GIVE DIGITAL CUSTOMERS, FINAL ASSEMBLY AND TEST (F.A. AND T.), AND FIELD SERVICE A MEANS OF VERIFYING THAT THE NON-VOLATILE MEMORY AND IT'S BATTERIES ARE OPERATIONAL. THIS DIAGNOSTIC WILL TEST NON-VOLATILE MEMORY IN ANY QBUS SYSTEM PROVIDING THE RESTRICTION CRITERIA SET FORTH IN SECTION 2 OF THIS MANUAL IS ADHERED TO.

1.2 PRODUCT USERS

1.2.1 ENGINEERING USAGE -

- DESIGN VERIFICATION OF BATTERY BACKUP CIRCUITRY
- AVAILABLE FOR DMT USE

1.2.2 MANUFACTURING USAGE -

VOLUME MANUFACTURING
A MEANS OF VERIFYING DATA DETECION OF NON-VOLATILE MEMORIES.

F A AND T
THIS DIAGNOSTIC WILL GIVE F.A. AND T. A MEANS OF CHECKING THE NON-VOLATILITY OF CMOS RAMS AND THE ABILITY OF THE BATTERIES TO MAINTAIN DATA RETENTION.

1.2.3 FIELD SERVICE USAGE -

VERIFICATION OF CUSTOMER INSTALLATION
SERVICE CALLS: FAILURE ISOLATION, REPAIR AND VERIFICATION
VERIFICATION OF ECO INSTALLATION

1.2.4 CUSTOMER USAGE -

MEANS OF VERIFYING DATA RETENTION OF NON-VOLATILE MEMORY AND IT'S BATTERIES.

2.0 PRODUCT GOALS

2.1 ASSUMPTIONS

IT WILL BE ASSUMED THAT PRIOR TO EXECUTING THIS DIAGNOSTIC ALL APPROPRIATE CPU, MEMORY, AND PERIPHERAL DIAGNOSTICS HAVE BEEN SUCCESSFULLY RUN.

THE DIAGNOSTIC WILL REPORT ANY BAD LOCATIONS FOUND WITHIN ANY AREA OF MEMORY THAT NON-VOLATILE EXISTS.

THE OPERATOR MUST HAVE THE KNOWLEDGE OF THE STARTING ADDRESS OF A NON-VOLATILE MEMORY WITHIN THE OPERATOR'S SYSTEM.

THIS DIAGNOSTIC TREATS ALL MEMORY MODULES CONTAINING ROM, NON-BATTERY BACKED UP MOS RAM, AND NON-BATTERY BACKED UP CMOS RAM AS VOLATILE MEMORY. ALL MEMORY MODULES CONTAINING CORE, BATTERY BACKED UP MOS RAM, AND BATTERY BACKED UP CMOS RAM AS NON-VOLATILE MEMORY.

THE Q-BUS CPU MUST HAVE EITHER A HALT/ENABLE SWITCH OR THE CPU MODULE MUST BE STRAPPED TO HALT WHEN POWERED UP (WIRE JUMPER CONFIGURATION MUST BE W5-IN, W6-OUT)

THE OPERATOR OF THIS DIAGNOSTIC IS ASSUMED TO HAVE THE MEANS OF VERIFYING THAT THE MEMORY MAP THIS DIAGNOSTIC WILL PRODUCE CORRECTLY CONFIGURES THE SYSTEM UNDER TEST.

2.2 PERFORMANCE GOALS

THE GOAL OF THIS DIAGNOSTIC IS TO DETERMINE THE DATA RETENTION OF ALL NON-VOLATILE MEMORY CONTAINED WITHIN A GIVEN SYSTEM. THIS PROGRAM WILL REQUIRE INFORMATION FROM THE OPERATOR DURING IT'S EXECUTION. THE INFORMATION WILL BE GATHERED USING A MENU TECHNIQUE.

2.3 COMPATIBILITY GOALS

THE INITIAL DIAGNOSTIC IS EXPECTED TO RUN ON ANY QBUS SYSTEM PROVIDING THAT THE TRAP LOCATIONS HAVE READ/WRITE ACCESS.

2.4 FAILSOFT GOALS

ANY PARITY ERROR CAUSED BY ACCESSING VOLATILE MEMORY AFTER POWER DOWN WILL BE IGNORED.

2.5 RESTRICTIONS

THIS DIAGNOSTIC IS DESIGNED TO RUN ON ALL QBUS SYSTEM EXCEPT FOR THOSE SYSTEMS THAT DO NOT ALLOW READ/WRITE ACCESS IN THE TRAP VECTOR SECTION OF MEMORY (ADDRESS 0 THROUGH 400 OCTAL). THE SYSTEM MUST HAVE A TERMINAL IN ORDER FOR THE OPERATOR TO ANSWER THE QUESTIONS THE DIAGNOSTIC WILL ASK. THE OPERATOR MUST KNOW AT WHICH LOCATIONS IN MEMORY THE NON-VOLATILE MEMORY EXISTS.

THIS DIAGNOSTIC IS TO ONLY WORK WITH CPU'S THA- SUPPPORT POWER-DOWN DATA RETENTION CAPABILITY.

2.6 NON-GOALS

THIS DIAGNOSTIC'S ONLY PURPOSE IS TO CHECK THAT NON-VOLATILE IS WORKING AND THAT THE BATTERIES ARE OK. THIS DIAGNOSTIC PERFORMS NO OTHER TESTING.

- NOT APT COMPATIBLE

- NOT IN THE SCOPE OF THIS PROJECT TO MODIFY THIS PROGRAM TO MEET UNIQUE NEEDS OF MANUFACTURING.

3.0 REQUIREMENTS

3.1 RUN TIME ENVIRONMENT REQUIREMENTS

```
QBUS CPU
CONSOLE TERMINAL
AT LEAST 1 OF ANY OF THE FOLLOWING MEMORY MODULES:
      MCV11 CMOS RAM MEMORY MODULES WITH BATTERY BACKUP
      MSV11 MOS RAM MEMORY WITH BATTERY BACKUP
      MMV11 CORE MEMORY MODULES
XXDP+ LOADING MEDIA
```

3.2 DEVELOPMENT ENVIRONMENT REQUIREMENTS

FOR PROGRAM DEVELOPMENT DIAGNOSTIC ENGINEERING WILL BE ABLE TO USE OUR OWN EQUIPMENT, BUT FOR INTIAL DEBUG, IT IS ESSENTIAL THAT WE HAVE ACCESS TO A PROPERLY EQUIPPED RUN TIME SYSTEM AS DESCRIBED IN THE PREVIOUS SECTION.

THIS DIAGNOSTIC WILL BE VERIFIED ON A VARITY OF Q-BUS SYSTEMS.

4.0 FUNCTIONAL DESCRIPTION

THE PURPOSE OF THIS PROGRAM IS TO TEST THE DATA RETENTION OF NON-VOLATILE MODULES IN ANY QBUS SYSTEM. THIS WILL BE ACCOMPLISHED BY ASKING THE OPERATOR A SERIES OF QUESTIONS (MENU), GENERATING A RESTART HELP FILE, WRITE A BACKGROUND PATTERN (125252) THROUGHOUT MEMORY, DO A CHECKSUM OF THE PROGRAM AND THE TRAP VECTOR SPACE (10-376), TELLING THE OPERATOR TO POWERDOWN THE SYSTEM, POWER IT BACK UP, RESTART THE DIAGNOSTIC, VERIFY THE CHECKSUMS, CHECK ENTIRE MEMORY FOR THE BACK-GROUND PATTERN (125252), AND GIVE THE OPERATOR A MEMORY MAP OF ALL VO-LATILE AND NON VOLATILE MEMORY IN THE SYSTEM. THE OPERATOR MUST COM-PARE THIS MAP WITH THE ONE LEFT BY THE INSTALLER IN ORDER TO DETERMINE THAT PROPER DATA RETENTION EXISTS.

4.1 MENJ QUESTIONS

THE DIAGNOSTIC WILL PROMPT THE OPERATOR TO ANSWER THE FOLLOWING QUES-TIONS A <CR> AS AN ANSWER TO THE FOLLOWING QUESTIONS WILL CAUSE THE CONDITION STATED DIRECTLY TO THE LEFT OF THE QUESTION MARK TO BE IN EFFECT.

1. DO YOU WANT A HELP FILE (L) N?

   A "Y" ANSWER TO THIS WILL CAUSE GENERAL HELP FILE TO BE PRINTED BEFORE CONTINUING WITH THE QUESTIONS.

2. IS THERE NON-VOLATILE MEMORY AT ADDRESS 0 (L) Y?

   A "Y" OR <CR> ANSWER WILL INHIBIT THE ASKING OF QUESTION 3.

3. WHAT IS THE STARTING ADDRESS OF ONE OF THE NON-VOLATILE MEMO-RY MODULES (O) 0?

   THE PROGRAM WILL NOW MOVE ITSELF TO THAT AREA IN MEMORY. IF THE ADDRESS IS NOT A LEGAL 4K BOUNDARY ADDRESS THE DIAGNOSTIC WILL PRINT THE FOLLOWING ERROR MESSAGE AND REPEAT QUESTION 3.

   ```
   ILLEGAL ADDRESS - NOT A 4K BOUNDARY
   ```

   IF THE ADDRESS IS OUTSIDE THE MEMORY AREA OF THE SYSTEM THE FOLLOWING ERROR MESSAGE WILL BE PRINTED AND QUESTION 3 IS RE-PEATED.

   ```
   4K ADDRESS OR NUMBER OF 4K BANKS ARE OUT OF ADDRESS RANGE  OF
   THIS SYSTEM.
   ```

NOTE

* IF THE ADDRESS TYPED IN IS REALLY VOLATILE THEN WHEN THE PROGRAM IS RESTARTED THE SYSTEM WILL NOT OPERATE PROPERLY.

4. DO YOU WANT THE DIAGNOSTIC TO VALIDATE ALL MEMORY IN THIS SYSTEM (L) Y?

   A 'Y'' <CR> ANSWER WILL CAUSE NO MORE QUESTIONS TO BE PROMPT AND THE PROGRAM FLOW WILL GO TO THE SECTION ON POWERFAIL 4.2

5. WHAT IS THE STARTING ADDRESS OF THE NON-VOLATILE MEMORY UNDER TEST (O) 0?

   THIS ADDRESS MAY OR MAY NOT BE THE SAME AS THE ADDRESS TYPED IN QUESTION 3. IF IT IS THEN THE PROGRAM WILL RESIDE WITHIN THE QUESTIONED NON-VOLATILE MEMORY MODULE. IF NOT THEN THE PROGRAM WILL BE AT EITHER ADDRESS 0 IF QUESTION 2 WAS A 'Y'' OR AT THE ADDRESS QUESTION 3 ASKED FOR.

6. HOW MANY 4K BANKS OF MEMORY TO BE TESTED (D) 0?

   THE OPERATOR WILL INPUT THE NUMBER OF 4K BANKS TO BE TESTED FROM THE START ADDRESS. A '0'' OR <CR> ANSWER WILL MEAN TEST ALL THE BANKS FROM THE START ADDRESS TO END OF MEMORY.

4.2 POWER FAIL

BEFORE THIS DIAGNOSTIC INSTRUCTS THE OPERATOR TO POWER DOWN THE SYSTEM. THE DIAGNOSTIC WILL PRODUCE A RESTART HELP FILE ON THE CONSOLE TERMINAL. THIS HELP FILE IS NEEDED TO GUIDE THE OPERATOR IN A STEP BY STEP MENU THAT MUST BE PERFORMED IN ORDER TO RESTART THE DIAGNOSTIC AFTER POWER IS RETURNED. SECTION 4.5 GIVES AS PART OF THE EXAMPLES TYPICAL RESTART HELP FILES FOR VARIOUS TYPES OF SYSTEMS.

THE OPERATOR WILL THEN BE INSTRUCTED TO POWER DOWN THE SYSTEM FOR NO LESS THAN 2 MINUTES AND NO MORE THAN 100 HOURS. THE ACTUAL MESSAGE IS:

PLEASE POWER DOWN THIS SYSTEM. AFTER 2 MINUTES BUT NO LONGER THAN 100 HOURS, EXECUTE THE RESTART HELP FILE.

4.3 POWER UP

AFTER THE POWER WAS RETURNED TO THE SYSTEM THE OPERATOR MUST FOLLOW THE STEP BY STEP PROCEDURE ON HOW TO RESTART THE DIAGNOSTIC GIVEN BY THE PROGRAM BEFORE POWER DOWN. THE DIAGNOSTIC WILL START CHECKING ALL MEMORY FOR DATA RETENTION. ON THE CONSOLE TERMINAL A MEMORY MAP WILL BE PRINTED AT ALL AREAS WITHIN THE SYSTEM THAT CONTAIN VOLATILE AND NON-VOLATILE MEMORY.

AN EXAMPLE OF A MEMORY MAP GIVEN BY THIS DIAGNOSTIC:

```
                         MEMORY MAP
                         ----------

START ADR.   END ADR.   MEMORY TYPE           WORD ERROR  PARITY ERROR

0             37776     NON-VOLATILE MEMORY       0             0
40000        157776     VOLATILE MEMORY

THIS CONCLUDES THE NON-VOLATILE DATA RETENTION TEST
```

4.4 ERROR WITHIN MEMORY MAP

A) WORD ERROR - THE NUMBER OF WORD FAILURES FOUND WITHIN A GIVEN 4K CHUNK OF MEMORY IF THE DIAGNOSTIC FEELS THAT 4K IS NON-VOLATILE.

B) PARITY ERROR IS THE NUMBER OF PARITY ERRORS FOUND WITHIN A 4K PART OF MEMORY. THIS WILL ONLY PRINT OUT ERRORS FOUND IN NON-VOLATILE MEMORY. DUE TO THE EXCESSIVE RUNTIME OF THE DIAGNOSTIC WHEN SERVICING EVERY PARITY ERROR (WHICH INCLUDES ALL PARITY ERRORS WITHIN VOLATILE MEMORY) A DECISION WAS MADE TO LIMIT THE PARITY SERVICING CAPABILITY TO 10 PARITY ERRORS PER 4K OF MEMORY. IF THE MEMORY THE PARITY ERRORS WERE IN WERE NON-VOLATILE THEN A MESSAGE OF "10 OR MORE" WOULD BE PRINTED UNDER PARITY ERRORS.

C) PROGRAM CHECK SUM ERROR - BEFORE THE MEMORY MAP HEADER IS PRINTED THE PROGRAM DOES A CHECKSUM OF ITSELF TO MAKE SURE THAT THE MEMORY THE PROGRAM IS IN IS OK. IF NOT THEN THE FOLLOWING ERROR MESSAGE WILL NOTIFY THE OPERATOR.

CHECKSUM ERROR. THE MEMORY THE PROGRAM IS RESIDENT IN HAS VOLATILE LOCATIONS.

THE PROGRAM WILL CONTINUE BUT MAY NOT OPERATE PROPERLY.

D) TRAP VECTOR CHECKSUM. ANOTHER CHECKSUM WAS ALSO DONE WITH THE VECTOR PAGE (ADDRESS 0-376) OF BANK 0. THIS TIME THE ERROR WILL ONLY BE REPORTED IF BANK 0 WAS DETERMINED TO BE NON-VOLATILE.

ERROR MESSAGE:

```
TRAP LOCATIONS (ADDRESS 0-376) HAVE CHECK SUM ERROR IN
NON-VOLATILE MEMORY.
```

4.5 PROGRAM FLOW FOR AN UNMAPPED SYSTEM

4.5.1 -

THE FOLLOWING IS AN EXAMPLE OF AN UN-MAPPED SYSTEM WITH NON-VOLATILE MEMORY IN THE LOWEST PART OF MAIN MEMORY. AFTER LOADING THE DIAGNOSTIC, THE FOLLOWING SEQUENCY OF EVENTS WILL APPEAR ON THE CONSOLE TERMINAL.

```
CVMEMAO DATA RETENTION DIAGNOSTIC FOR NON-VOLATILE MEMORIES.

 THIS IS AN UN-MAPPED SYSTEM (NO MEMORY MANAGEMENT) WITH 28K OF
 MEMORY.

 DO YOU WANT A HELP FILE (L) N?  N

 IS THERE NON-VOLATILE MEMORY AT ADDRESS 0 (L) Y?  Y

 DO YOU WANT THE DIAGNOSTIC TO VALIDATE ALL MEMORY IN THIS SYSTEM
 (L) Y?  Y

              RESTART HELP FILE

 SET ENABLE/HALT FRONT PANEL SWITCH TO HALT POSITION, APPLY POWER,
 SET ENABLE/HALT FRONT PANEL SWTICH TO ENABLE POSITION, AND TYPE:

 @777707/506<CR>
 @P
              PLEASE STAND BY!
PLEASE POWER DOWN THIS SYSTEM.  AFTER 2 MINUTES BUT NO LONGER
 THAN 100 HOURS.  EXECUTE THE RESTART HELP FILE.
```

THE OPERATOR WILL NOW POWER DOWN, WAIT 2 MINUTES AND RETURN POWER TO THE SYSTEM. THE FOLLOWING IS A TYPICAL MEMORY MAP FOR AN UN-MAPPED SYSTEM.

```
                                MEMORY MAP
                                ----------

START ADR.    END ADR.    MEMORY TYPE          WORD ERROR  PARITY ERROR

0              37776      NON-VOLATILE MEMORY      0            0
40000         137776      VOLATILE MEMORY
140000        157776      NON-VOLATILE MEMORY      0            0

THIS CONCLUDES THE NON-VOLATILE DATA RETENTION TEST.
```

4.5.2 -

THE FOLLOWING IS AN EXAMPLE OF AN UN-MAP
PED SYSTEM WITH NON-VOLATILE
MEMORY SOMEWHERE OTHER THAN LOWEST PART OF MAIN MEMORY.

AFTER LOADING THE DIAGNOSTIC, THE FOLLOWING SEQUENCE OF EVENTS WILL APPEAR ON THE CONSOLE TERMINAL.

```
    CVMEMAO DATA RETENTION DIAGNOSTIC FOR NON-VOLATILE MEMORIES

     THIS IS AN UN-MAPPED SYSTEM (NO MEMORY MANAGEMENT)  WITH  24K  OF
     MEMORY.

     DO YOU WANT A HELP FILE (L) N?  N

     IS THERE NON VOLATILE MEMORY AT ADDRESS 0 (L) Y?  N

     WHAT IS THE STARTING ADDRESS OF ONE OF  THE  NON-VOLATILE  MEMORY
     MODULES L (0) 0?  20000

     DO YOU WANT THE DIAGNOSTIC TO VALIDATE ALL MEMORY IN THIS  SYSTEM
     (L) Y?  Y

                  RESTART HELP FILE

SET ENABLE/HALT FRONT PANEL SWITCH TO HALT POSTION, APPLY  POWER,
SET ENABLE/HALT FRONT PANEL SWITCH TO ENABLE POSITION, AND TYPE:

@$7/XXXXXX 20176<CR>
@P
                  PLEASE STAND BY!
PLEASE POWER DOWN THIS SYSTEM.  AFTER 2  MINUTES  BUT  NO  LONGER
 THAN 100 HR.  EXECUTE THE RESTART HELP FILE.
```

THE OPERATOR WILL NOW POWERDOWN, WAIT 2 MINUTES, AND RETURN POWER TO THE SYSTEM. NEXT THE OPERATOR MUST FOLLOW THE RESTART HELP FILE. AFTER COMPLETION OF THE HELP FILE THE DIAGNOSTIC WILL PRINT A TYPICAL MEMORY MAP FOR AN UN-MAPPED SYSTEM.

```
                              MEMORY MAP
                              ----------

START ADR.   END ADR.   MEMORY TYPE          WORD ERROR  PARITY ERROR
0            17776      VOLATILE MEMORY
20000        77776      NON-VOLATILE MEMORY       0             0
100000       177776     VOLATILE MEMORY

THIS CONCLUDES THE NON-VOLATILE DATA RETENTION TEST.
```

4.5.3 -

THE FOLLOWING IS AN EXAMPLE OF EXECUTING THE DIAGNOSTIC IN AN UN-MAPPED SYSTEM THAT CONTAIN 2 NON-VOLATILE ARRAYS. THE OPERATOR WANTS TO EXECUTE THE TEST ON THE NON-VOLATILE ARRAY THAT RESIDES IN THE LOWEST PART OF MAIN MEMORY AND HAVE THE PROGRAM RESIDE IN THE OTHER NON-VOLATILE ARRAY. EACH ARRAY HAS 4K OF MEMORY.

```
    CVMEMAO DATA RETENTION DIAGNOSTIC FOR NON-VOLATILE MEMORIES

     THIS IS AN UN-MAPPED SYSTEM (NO MEMORY MANAGEMENT)  WITH  28K  OF
     MEMORY.

     DO YOU WANT A HELP FILE (L) N?  N

     IS THERE NON-VOLATILE MEMORY AT ADDRESS 0 (L) Y?  N

     WHAT IS THE STARTING ADDRESS OF ONE OF  THE  NON-VOLATILE  MEMORY
     MODULES (O) 0?  20000

     DO YOU WANT THE DIAGNOSTIC TO VALIDATE ALL MEMORY IN THIS  SYSTEM
     (L) Y?  N

     WHAT IS THE STARTING ADDRESS OF  THE  NON-VALATILE  MEMORY  UNDER
     TEST (O) 0?  0

     HOW MANY 4K BANKS OF MEMORY ARE TO BE TESTED (D) 0?  1

                   RESTART HELP FILE

SET ENABLE/HALT FRONT PANEL SWITCH TO HALT POSITION, APPLY POWER,
SET ENABLE/HALT FRONT PANEL SWITCH TO ENABLE POSITION, AND TYPE:

@$7/XXXXXX 20204<CR>
@P
                   PLEASE STAND BY!
PLEASE POWER DOWN THIS SYSTEM.  AFTER 2  MINUTES  BUT  NO  LONGER
THAN 100 HR.  EXECUTE THE RESTART HELP FILE.
```

THE OPERATOR WILL NOW POWER DOWN, WAIT 2 MINUTES, AND RETURN POWER TO THE SYSTEM. NEXT THE OPERATOR MUST FOLLOW THE RESTART HELP FILE. AFTER COMPLETION OF THE HELP FILE THE DIAGNOSTIC WILL PRINT A MEMORY MAP FOR THAT PART OF MEMORY ONLY.

```
                              MEMORY MAP
                              ----------

START ADR.    END ADR.    MEMORY TYPE           WORD ERROR  PARITY ERROR
0             17776       NON-VOLATILE MEMORY     0             0

THIS CONCLUDES THE NON-VOLATILE DATA RETENTION TEST.
```

4.6 PROGRAM FLOW FOR A MAPPED SYSTEM

4.6.1 -

THE FOLLOWING IS AN EXAMPLE OF A MAPPED SYSTEM WITH NON-VOLATILE MEMORY IN THE LOWEST PART OF MAIN MEMORY.

AFTER LOADING THE DIAGNOSTIC, THE FOLLOWING SEQUENCE OF EVENTS WILL APPEAR ON THE CONSOLE TERMINAL.

```
CVMEMAO DATA RETENTION DIAGNOSTIC FOR NON-VOLATILE MEMORIES.

 THIS IS A MAPPED SYSTEM (MEMORY MANAGEMENT) WITH 124K OF MEMORY.

 DO YOU WANT A HELP FILE (L) N?  N

 IS THERE NON-VOLATILE MEMORY AT ADDRESS 0 (L) Y?  Y

 DO YOU WANT THE DIAGNOSTIC TO VALIDATE ALL MEMORY IN THIS  SYSTEM
 (L) Y?  Y

              RESTART HELP FILE

 SET ENABLE/HALT FRONT PANEL SWITCH TO HALT POSITION, APPLY POWER,
 TYPE:

 @772344/XXXXXX 0<CR>
 @772356/XXXXXX 177600<CR>
 @772304/XXXXXX 77406<CR>
 @772316/XXXXXX 77406<CR>
 @777572/XXXXXX 1<CR>

 SET ENABLE/HALT FRONT PANEL SWITCH TO ENABLE POSITION AND TYPE:

 @$7/XXXXXX 40510<CR>
 @P
              PLEASE STAND BY!
PLEASE POWER DOWN THIS SYSTEM.  AFTER 2  MINUTES  BUT  NO  LONGER
THAN 100 HOURS.  EXECUTE THE RESTART HELP FILE.
```

```
THE OPERATOR WILL NOW POWER DOWN, WAIT 2 MINUTES AND RETURN  POWER  TO
THE  SYSTEM.   THE FOLLOWING IS A TYPICAL MEMORY MAP FOR A MAPPED SYS-
TEM.

                            MEMORY MAP
                            ----------

START ADR.    END ADR.    MEMORY TYPE           WORD ERROR  PARITY ERROR
0              37776      NON-VOLATILE MEMORY       0             0
40000         757776      VOLATILE MEMORY

THIS CONCLUDES THE NON-VOLATILE DATA RETENTION TEST.
```

4.6.2 -

THE FOLLOWING IS AN EXAMPLE OF A MAPPED SYSTEM WITH NON-VOLATILE MEMORY SOMEWHERE OTHER THAN LOWEST PART OF MAIN MEMORY.

AFTER LOADING THE DIAGNOSTIC, THE FOLLOWING SEQUENCE OF EVENTS WILL APPEAR IN THE CONSOLE TERMINAL.

```
    CVMEAO DATA RETENTION DIAGNOSTIC FOR NON-VOLATILE MEMORIES

     THIS IS A MAPPED SYSTEM (MEMORY MANAGEMENT) WITH 2043K OF MEMORY.

     DO YOU WANT A HELP FILE (L) N?  N

     IS THERE NON-VOLATILE AT ADDRESS 0 (L) Y?  N

     WHAT IS THE STARTING ADDRESS OF THE NON-VOLATILE  MEMORY  MODULES
     (O) 0?  1000000

     DO YOU WANT THE DIAGNOSTIC TO VALIDATE ALL MEMORY IN THIS  SYSTEM
     (L) Y?  Y

                   RESTART HELP FILE

 SET ENABLE/HALT FRONT PANEL SWITCH TO HALT POSITION, APPLY POWER,
 TYPE:

 @772344/XXXXXX 10000<CR>
 @772356/XXXXXX 177600<CR>
 @772304/XXXXXX 77406<CR>
 @772316/XXXXXX 77406<CR>
 @777572/XXXXXX 1<CR>
 @772516/XXXXXX 20<CR>

 SET ENABLE/HALT FRONT PANEL SWITCH TO ENABLE POSITION AND TYPE:

 @$7/XXXXXX 41034<CR>
 @P
                   PLEASE STAND BY!
PLEASE POWER DOWN THIS SYSTEM.  AFTER 2 MINUTES BUT NO MORE  THAN  100
HR.  EXECUTE THE RESTART HELP FILE.
```

THE OPERATOR WILL NOW POWER DOWN, WAIT 2 MINUTES, AND RETURN POWER TO THE SYSTEM. NEXT THE OPERATOR MUST FOLLOW THE RESTART HELP FILE. AFTER COMPLETION OF THE HELP FILE THE DIAGNOSTIC WILL PRINT A TYPICAL MEMORY MAP FOR A MAPPED SYSTEM.

```
                    MEMORY MAP
                    ----------

START ADR.   END ADR.   MEMORY TYPE          WORD ERROR  PARITY ERROR
0             3777776   VOLATILE MEMORY           0
4000000       4077776   NON VOLATILE MEMORY       0            0
4100000       7777776   VOLATILE MEMORY           0
1000000       1017776   NON-VOLATILE MEMORY       0            0
1020000      17757776   VOLATILE MEMORY           0

THIS CONCLUDES THE NON-VOLATILE DATA RETENTION TEST.
```

4.7 HELP FILE

FUNCTIONAL DESCRIPTION

THE PURPOSE OF THIS PROGRAM IS TO TEST THE DATA RETENTION OF NON-VOLATILE MODULES IN ANY QBUS SYSTEM. THIS IS NOT A MEMORY DIAGNOSTIC. PLEASE RUN CZKMA OR VMSA DIAGNOSTICS BEFORE RUNNING THIS PROGRAM. THE FOLLOWING IS A BRIEF DISCRIPTION OF THE PROGRAM FLOW:

1. ASK THE OPERATOR A SERIES OF QUESTIONS (MENU)
2. RELOCATE PROGRAM TO NON-VOLATILE AREA (IF BANK 0 IS VOLATILE)
3. GENERATE AND PRINT A RESTART HELP FILE
4. WRITE A BACKGROUND PATTERN (125252) THROUGHOUT MEMORY
5. DO A CHECKSUM OF THE PROGRAM
6. DO A CHECKSUM OF THE TRAP VECTOR SPACE (ADDRESS 0-376)
7. TELL THE OPERATOR TO POWERDOWN THE SYSTEM

THE OPERATOR WILL NOW FOLLOW THE RESTART HELP FILE SOME TIME BETWEEN 2 MINUTES AND 100 HOURS. THE FOLLOWING IS THE FLOW OF THE PROGRAM AFTER POWER UP:

1. DO ANOTHER CHECKSUM OF THE PROGRAM AND CHECK IT AGAINST THE ONE DONE BEFORE POWER DOWN.
2. DO ANOTHER CHECKSUM OF THE TRAP VECTOR SPACE AND CHECK IT AGAINST THE ONE DONE BEFORE POWER DOWN. (NO TRAP CHECKSUM ERROR WILL BE REPORTED UNLESS BANK 0 WAS DETERMINED NON-VOLATILE)
3. CHECK ENTIRE MEMORY FOR THE BACKGROUND PATTERN (125252)
4. PRINT A MEMORY MAP OF ALL VOLATILE AND NON VOLATILE MEMORY IN THE SYSTEM

THE OPERATOR MUST COMPARE THIS MAP WITH THE ONE LEFT BY THE INSTALLER IN ORDER TO DETERMINE THAT PROPER DATA RETENTION EXISTS.

PREREQUISITE

1. CZKMA OR VMSA MUST SUCCESSFULLY COMPLETE
2. OPERATOR MUST KNOW STARTING ADDRESSES OF ALL VOLATILE AND NON-VOLATILE MEMORY IN THE SYSTEM
3. VIDEO TERMINALS MUST EITHER REMAIN POWERED WHEN PROCESSOR IS POWERED DOWN OR A COPY OF THE RESTART HELP FILE MUST BE MADE BEFORE POWERING DOWN THE SYSTEM
4. IF THE CPU DOES NOT HAVE AN ENABLE/HALT SWITCH, CHECK THAT THE CPU IS STRAPPED TO HALT WHEN POWERED UP. OTHERWISE THE DIAGNOSTIC WILL NOT OPERATE PROPERLY.

4.8 ERROR INFORMATION

IN THE EVENT THAT THE DIAGNOSTIC DID NOT CONTINUE WHEN RESTART HELP FILE WAS COMPLETED, THE FOLLOWING EXAMPLES ARE OPERATOR ERROR THAT COULD CAUSE THE DIAGNOSTIC NOT TO RESTART.

1. STARTING ADDRESS WAS VOLATILE MEMORY

2. ERROR WHEN SETTING UP RESTART REGISTERS WHEN USING CONSOLE ODT.

3. ENABLE/HALT FRONT PANEL SWITCH SET TO ENABLE WHEN POWER IS RETURNED.

4. CPU WITHOUT ENABLE/HALT FRONT PANEL WAS NOT STRAPPED TO HALT WHEN POWERED UP (W5-IN, W6-OUT)

IF NO OPERATOR ERROR WAS DETERMINED THEN THE DATA RETENTION OF THE NON-VOLATILE MEMORY THE PROGRAM WAS IN IS NOW BAD. CHECK THE BATTERIES.

4.9 SCHEDULE

THE PROGRAM IS EXPECTED TO TAKE 8 WEEKS FROM DESIGN THROUGH DEBUG AT A COST OF 11K.

THE COST AND TIME WILL FLUCTUATE IF THERE ARE ANY CHANGES TO THE FUNCTIONALITY OF THIS PROGRAM.

5.0 INTERFACE

THE DIAGNOSTIC WILL BE DESIGNED TO CONFORM WITH THE FOLLOWING INTERFACES.

1. SYSMAC

2. XXDP+

6.0 BIBLIOGRAPHY

THE FOLLOWING LISTS THE LITERATURE USED IN THE DESIGN OF THIS DIAGNOSTIC.

1. SYSMAC.MAN

2. SPMACJ.DOC

3. XXDPPLUS.DOC

7.0 GLOSSARY

&

```
     000000          .ENABLE ABS
                     .DSABLE GBL
                                                 .LIST   SEQ,BIN,CND


                                         .MACRO  .$SIZE  MMERR
                                         .SBTTL  ROUTINE TO SIZE MEMORY
                                         STARS
                                         ;*CALL:
                                         ;*      JSR     PC,$SIZE
                                         ;*      RETURN
                                         .IF DF KIPAR0
                                         ;*$LSTAD WILL CONTAIN:
                                         ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
                                         ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
                                         ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
                                         ;*$KT11 IS THE MEMORY MANAGEMENT KEY
                                         ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
                                         ;*      MUST BE SETUP BEFORE THE CALL
                                         ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
                                         ;*      DETERMINED BY ROUTINE
                                         .IFF
                                         ;*$LSTAD WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION
                                         .IFTF
                                         $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
                                                 MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
                                         .IFT
                                                 MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
                                                 MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
                                                 MOV     R4,-(SP)        ;;SAVE R4 ON THE STACK                  BK001
                                         .IFTF
                                                 MOV     @#114,-(SP)     ;;SAVE MEMORY ERROR VECTOR PS & PC
                                                 MOV     @#116,-(SP)
                                                 MOV     #116,@#114      ;;IGNORE PARITY ERRORS WHILE SIZING
                                                 MOV     #RTI,@#116
                                                 MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
                                                 MOV     @#ERRVEC+2,-(SP)
                                                 MOV     SP,R0           ;;SAVE THE STACK POINTER
                                         ;;SET THE ERRVEC PS TO THE PRESENT PS
                                                 GETPRI  @#ERRVEC+2
                                         .IFT
                                                 MOV     #3776,R1        ;;SETUP ADDRESS
                                                 TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
                                         $KT11:  .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
                                                 BPL     $CORE           ;;BR IF NO
                                                 MOV     #$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
                                                 TST     @#SR0           ;;KT11 ARE YOU THERE?
                                                 BIS     #100000,$KT11   ;;YES--SET KT11 KEY
                                                 MOV     #100$,@#ERRVEC  ;;SET FOR TIMEOUT                       BK001
                                                 TST     @#172516        ;;Q-BUS MAP ARE YOU THERE?              BK001
                                                 MOV     #200,$MAP       ;;TURN ON MAP INDICATOR                 BK001
                                                 MOV     #176200,$STOP   ;;END OF 2M OF MEMORY
                                                 BR      $MAPRG          ;;GO SET UP MAP REGISTERS               BK001
                                         100$:   MOV     #6200,@#$STOP   ;;COMPARISON VALUE FOR 18 BIT MAPPING   BK001
                                                 CMP     (SP)+,(SP)+     ;;CLEAN OFF STACK                       BK001
                                                 CLR     @#$MAP          ;;MAKE SURE MAP INDICATOR TURNED OFF    BK001
                                                 BR      $NOMAP          ;;                              BK001
```

```
$MAP:   .WORD   0                  ;;=200 IF MAP PRESENT                      BK001
$STOP:  .WORD   0                  ;;FILLED WITH APPROPRIATE COMPARISON VALUE       BK001
$MAPRG:
100$:
$NOMAP:
.IF NB  MMERR
        MOV     #MMERR,@#MMVEC     ;;SET IN CASE OF ERROR
        MOV     #340,@#MMVEC+2
.ENDC
        CLR     -(SP)              ;;INITIALIZE FOR 'PAR'' LOADING
        MOV     #KIPAR0,R2         ;;ADDRESS OF FIRST 'PAR''
        MOV     #^D8,R3            ;;LOAD EIGHT 'PAR.'S'' AND EIGHT 'PDR.'S''
1$:     MOV     #77406,-40(R2)     ;;PDR = 4K, UP, READ/WRITE
        MOV     (SP),(R2)+         ;;LOAD 'PAR''
        ADD     #200,(SP)          ;;UPDATE FOR NEXT 'PAR''
        SOB     R3,1$              ;;LOOP UNTIL ALL EIGHT ARE LOADED
        MOV     #177600,-(R2)      ;;SETUP KIPAR7 FOR I/O
        CLR     -(R2)              ;;SETUP KIPAR6 FOR TESTING
        MOV     #2$,@#ERRVEC       ;;CATCH TIMEOUT IF NO SR3
        MOV     #20,@#SR3          ;;ENABLE 22 BIT MODE AND UNIBUS MAP        BK001
        BR      3$                 ;;THIS PDP-11 HAS A SR3 REGISTER
2$:     CMP     (SP)+,(SP)+        ;;CLEAN OFF THE STACK--NO SR3
3$:     INC     @#SR0              ;;TURN ON MEMORY MANAGEMENT
        MOV     #$KTOUT,@#ERRVEC   ;;SET FOR TIME OUT
        TSTB    @#$MAP             ;;IS MAP THERE?                            BK001
        BPL     4$                 ;;NO-SKIP                                  BK001
        MOV     #$MMOUT,@#114      ;;SET UP MEMORY ERROR VECTOR               BK001
        MOV     @#ERRVEC+2,@#116   ;;LOCK OUT INTERRUPTS             BK001
4$:     TST     @#143776           ;;TRAP ON NON-EX-MEM
        ADD     #40,(R2)           ;;MAKE A 1K STEP
        CMP     @#$STOP,(R2)       ;;LAST ONE?
        BHI     4$                 ;;NO--TRY IT
$KTOUT: MOV     (R2),R2            ;;GET LAST BANK+1
        CLR     @#SR0              ;;TURN OFF MEMORY MANAGEMENT
        TSTB    @#$MAP             ;;IS MAP THERE?                            BK001
        BPL     $SIZEX             ;;NO-SKIP                                  BK001
        CLR     @#SR3              ;;TURN OFF MAP                             BK001
        BR      $SIZEX
$MMOUT: MOV     @#177744,R4        ;;SAVE MEMORY ERROR REGISTER               BK001
        MOV     R4,@#177744        ;;CLEAR BITS IN REGISTER                   BK001
        BIT     #1,R4              ;;MEMORY TIMEOUT?                          BK001
        BNE     $KTOUT             ;;YES-EXIT                                 BK001
        RTI                        ;;MUST BE PARITY ERROR-IGNORE IT           BK001
$KTNEX: BIC     #100000,$KT11      ;;KT11 NON-EXISTENT
$CORE:  MOV     #$CROUT,@#ERRVEC   ;;SET FOR TIMEOUT
        CLR     R2                 ;;SET UP BANK
1$:     ADD     #4000,R1           ;;INCREMENT BY 1K
        ADD     #40,R2             ;;1K STEP
        TST     (R1)               ;;TRAP ON TIME OUT
        CMP     #177776,R1         ;;LAST ONE
        BNE     1$                 ;;NO--TRY AGAIN
$CROUT: SUB     #4000,R1
$SIZEX: SUB     #40,R2             ;;DROP BACK
.IFF
        MOV     #2$,@#ERRVEC       ;;SET FOR TIMEOUT
        MOV     #20000,R1          ;;FIRST ADDRESS
1$:     TST     (R1)               ;;TEST THIS ADDRESS
```

```
                TST     (R1)+           ;;STEP TO NEXT ADDRESS
                BR      1$              ;;TRY ANOTHER
        2$:     SUB     #2,R1           ;;DROP BACK
        .IFTF
                MOV     R0,SP           ;;RESTORE THE STACK
                MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
                MOV     (SP)+,@#ERRVEC
                MOV     (SP)+,@#116     ;;RESTORE MEMORY ERROR VECTOR
                MOV     (SP)+,@#114
                MOV     R1,$LSTAD       ;;LAST ADDRESS
        .IFT
                MOV     R2,$LSTBK       ;;LAST BANK
                MOV     (SP)+,R4        ;;RESTORE R4                    BK001
                MOV     (SP)+,R3        ;;RESTORE R3
                MOV     (SP)+,R2        ;;RESTORE R2
        .IFTF
                MOV     (SP)+,R1        ;;RESTORE R1
                MOV     (SP)+,R0        ;;RESTORE R0
                RTS     PC
        $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
        .IFT
        $LSTBK: .WORD   0               ;;CONTAINS THE LAST BANK
        .ENDC
        .ENDM   .$SIZE
```

```
.MACRO  .$TYPE
.SBTTL  TYPE ROUTINE
STARS
;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
;*
;*CALL:
;*1) USING A TRAP INSTRUCTION
;*       TYPE    ,MESADR            ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;*OR
;*       TYPE
;*       MESADR
;*
$TYPE:   TSTB    $TPFLG1            ;;IS THERE A TERMINAL?
         BPL     1$                 ;;BR IF YES
         HALT                       ;;HALT HERE IF NO TERMINAL
         BR      3$                 ;;LEAVE
1$:      MOV     R0,-(SP)           ;;SAVE R0
         MOV     @2(SP),R0          ;;GET ADDRESS OF ASCIZ STRING
.IF DF $MAIL
         CMPB    #APTENV,$ENV       ;;RUNNING IN APT MODE
         BNE     62$                ;;NO,GO CHECK FOR APT CONSOLE
         BITB    #APTSPOOL,$ENVM    ;;SPOOL MESSAGE TO APT
         BEQ     62$                ;;NO,GO CHECK FOR CONSOLE
         MOV     R0,61$             ;;SETUP MESSAGE ADDRESS FOR APT
         JSR     PC,$ATY3           ;;SPOOL MESSAGE TO APT
61$:     .WORD   0                  ;;MESSAGE ADDRESS
62$:     BITB    #APTCSUP,$ENVM     ;;APT CONSOLE SUPPRESSED
         BNE     60$                ;;YES,SKIP TYPE OUT
.ENDC
2$:      MOVB    (R0)+,-(SP)        ;;PUSH CHARACTER TO BE TYPED ONTO STACK
         BNE     4$                 ;;BR IF IT ISN'T THE TERMINATOR
         TST     (SP)+              ;;IF TERMINATOR POP IT OFF THE STACK
60$:     MOV     (SP)+,R0           ;;RESTORE R0
3$:      ADD     #2,(SP)            ;;ADJUST RETURN PC
         RTI                        ;;RETURN
4$:      CMPB    #HT,(SP)           ;;BRANCH IF <HT>
         BEQ     8$
         CMPB    #CRLF,(SP)         ;;BRANCH IF NOT <CRLF>
         BNE     5$
         TST     (SP)+              ;;POP  <CR><LF> EQUIV
         MOV     PC,-(SP)
         ADD     #$CRLF1-.,(SP)
         MOV     (SP)+,20$
         TYPE                       ;;TYPE A CR AND LF
20$:     .WORD $CRLF1
         CLRB    $CHARCNT           ;;CLEAR CHARACTER COUNT
         BR      2$                 ;;GET NEXT CHARACTER
5$:      JSR     PC,$TYPEC          ;;GO TYPE THIS CHARACTER
6$:      CMPB    $FILLC,(SP)+       ;;IS IT TIME FOR FILLER CHARS.?
         BNE     2$                 ;;IF NO GO GET NEXT CHAR.
         MOV     $NULL,-(SP)        ;;GET # OF FILLER CHARS. NEEDED
                                    ;;AND THE NULL CHAR.
7$:      DECB    1(SP)              ;;DOES A NULL NEED TO BE TYPED?
```

```
                BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
                JSR     PC,$TYPEC       ;;GO TYPE A NULL
                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
                BR      7$              ;;LOOP
;HORIZONTAL TAB PROCESSOR
8$:             MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
9$:             JSR     PC,$TYPEC       ;;TYPE A SPACE
                BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
                BNE     9$              ;;TAB STOP
                TST     (SP)+           ;;POP SPACE OFF STACK
                BR      2$              ;;GET NEXT CHARACTER
$TYPEC:         TSTB    @$TKS1          ;;CHAR IN KYBD BUFFER?                  ;MJD001
                BPL     10$             ;;BR IF NOT                             ;MJD001
                MOV     @$TKB1,-(SP)    ;;GET CHAR                              ;MJD001
                BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS                 ;MJD001
                CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                         ;MJD001
                BNE     102$            ;;BR IF NOT                             ;MJD001
101$:           TSTB    @$TKS1          ;;WAIT FOR CHAR                         ;MJD001
                BPL     101$                                                    ;MJD001
                MOVB    @$TKB1,(SP)     ;;GET CHAR                              ;MJD001
                BIC     #177600,(SP)    ;;STRIP IT                              ;MJD001
                CMPB    #$XON,(SP)      ;;WAS IT XON?                           ;MJD001
                BNE     101$            ;;BR IF NOT                             ;MJD001
102$:           TST     (SP)+           ;;FIX STACK                             ;MJD001
10$:            TSTB    @$TPS1          ;;WAIT UNTIL PRINTER IS READY
                BPL     10$                                                     ;MJD001
                MOVB    2(SP),@$TPB1    ;;LOAD CHAR TO BE TYPED INTO DATA REG.
                CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
                BNE     1$              ;;BRANCH IF NO
                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
                BR      $TYPEX          ;;EXIT
1$:             CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
                BEQ     $TYPEX          ;;BRANCH IF YES
                INCB    (PC)+           ;;COUNT THE CHARACTER
$CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
$TYPEX: RTS     PC
.IIF NDF HT,HT= 11                              ;;CODE FOR HORIZONTAL TAB
.IIF NDF $TKS1,$TKS1:                                                           ;MJD001
.IIF EQ .-$TKS1,$TKS1:  .WORD   177560          ;;TTY KDB STATUS                ;MJD001
.IIF NDF $TKB1,$TKB1:                                                           ;MJD001
.IIF EQ .-$TKB1,$TKB1:  .WORD   177562          ;;TTY KBD BUFFER                ;MJD001
.IIF NDF $XON,$XON = 21                                                         ;MJD001
.IIF NDF $XOFF,$XOFF = 23                                                       ;MJD001
.IIF NDF LF,LF= 12                              ;;CODE FOR LINE FEED
.IIF NDF CR,CR= 15                              ;;CODE FOR CARRIAGE RETURN
.IIF NDF CRLF,CRLF=     200                             ;;CODE FOR CARRIAGE RETURN-LINE FEED
.IIF NDF $TPS1,$TPS1:
.IIF EQ .-$TPS1,$TPS1:  .WORD   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
.IIF NDF $TPB1,$TPB1:
.IIF EQ .-$TPB1,$TPB1:  .WORD   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
.IIF NDF $NULL,$NULL:
.IIF EQ .-$NULL,$NULL:  .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
.IIF NDF $FILLS,$FILLS:
.IIF EQ .-$FILLS,$FILLS:        .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS RE
.IIF NDF $FILLC,$FILLC:
.IIF EQ .-$FILLC,$FILLC:        .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE F
.IIF NDF $TPFLG1,$TPFLG1:
  1140      .IIF EQ .-$TPFLG1,$TPFLG1:      .BYTE   0               ;;''TERMINAL AVAILABLE'' FLAG (BIT<07>
  1141      .IIF NDF $QUES,$QUES:
  1142      .IIF EQ .-$QUES,$QUES:  .ASCII  ''?''           ;;QUESTION MARK
  1143      .IIF NDF $CRLF1,$CRLF1:
  1144      .IIF EQ .-$CRLF1,$CRLF1:        .ASCII  <15>            ;;CARRAIGE RETURN
  1145      .IIF NDF $LF1,$LF1:
  1146      .IIF EQ .-$LF1,$LF1:    .ASCIZ  <12>            ;;LINEFEED
  1147      .IIF NE 1&.,    .EVEN
  1148      .ENDM   .$TYPE
```

```
.MACRO  .$TYPOCT
.SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
STARS
;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
;*OCTAL (ASCII) NUMBER AND TYPE IT.
;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
;*CALL:
;*       MOV     NUM,-(SP)         ;;NUMBER TO BE TYPED
;*       TYPOS                     ;;CALL FOR TYPEOUT
;*       .BYTE   N                 ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
;*       .BYTE   M                 ;;M=1 OR 0
;*                                         ;;1=TYPE LEADING ZEROS
;*                                         ;;0=SUPPRESS LEADING ZEROS
;*
;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
;*$TYPOS OR $TYPOC
;*CALL:
;*       MOV     NUM,-(SP)         ;;NUMBER TO BE TYPED
;*       TYPON                     ;;CALL FOR TYPEOUT
;*
;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
;*CALL:
;*       MOV     NUM,-(SP)         ;;NUMBER TO BE TYPED
;*       TYPOC                     ;;CALL FOR TYPEOUT
$TYPOS:  MOV     @(SP),-(SP)       ;;PICKUP THE MODE
         MOVB    1(SP),$0FILL      ;;LOAD ZERO FILL SWITCH
         MOVB    (SP)+,$OMODE+1    ;;NUMBER OF DIGITS TO TYPE
         ADD     #2,(SP)           ;;ADJUST RETURN ADDRESS
         BR      $TYPON
$TYPOC:  MOVB    #1,$0FILL         ;;SET THE ZERO FILL SWITCH
         MOVB    #6,$OMODE+1       ;;SET FOR SIX(6) DIGITS
$TYPON:  MOVB    #5,$OCNT          ;;SET THE ITERATION COUNT
         MOV     R3,-(SP)          ;;SAVE R3
         MOV     R4,-(SP)          ;;SAVE R4
         MOV     R5,-(SP)          ;;SAVE R5
         MOVB    $OMODE+1,R4       ;;GET THE NUMBER OF DIGITS TO TYPE
         NEG     R4
         ADD     #6,R4             ;;SUBTRACT IT FOR MAX. ALLOWED
         MOVB    R4,$OMODE         ;;SAVE IT FOR USE
         MOVB    $0FILL,R4         ;;GET THE ZERO FILL SWITCH
         MOV     12(SP),R5         ;;PICKUP THE INPUT NUMBER
         CLR     R3                ;;CLEAR THE OUTPUT WORD
1$:      ROL     R5                ;;ROTATE MSB INTO "C"
         BR      3$                ;;GO DO MSB
2$:      ROL     R5                ;;FORM THIS DIGIT
         ROL     R5
         ROL     R5
         MOV     R5,R3
3$:      ROL     R3                ;;GET LSB OF THIS DIGIT
         DECB    $OMODE            ;;TYPE THIS DIGIT?
         BPL     7$                ;;BR IF NO
         BIC     #177770,R3        ;;GET RID OF JUNK
         BNE     4$                ;;TEST FOR 0
         TST     R4                ;;SUPPRESS THIS 0?
         BEQ     5$                ;;BR IF YES
4$:      INC     R4                ;;DON'T SUPPRESS ANYMORE 0'S
         BIS     #'0,R3            ;;MAKE THIS DIGIT ASCII
```

```
5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
        MOVB    R3,8$           ;;SAVE FOR TYPING
        MOV     PC,-(SP)        ;;FETCH ADDRESS OF NUMBER TO BE TYPED USING
                                ;;POSITION INDEPENDENT CODE
        ADD     #8$-.,(SP)      ;;STACK NOW CONTAINS ADDRESS OF NUMBER TO BE
                                ;;PRINTED
        MOV     (SP)+,10$       ;;SET UP ADDRESS FOR TYPE COMMAND
        TYPE                    ;;GO TYPE THIS DIGIT
10$:    .WORD   8$              ;;ADDRESS OF CHARACTER TO BE PRINTED
7$:     DECB    $OCNT           ;;COUNT BY 1
        BGT     2$              ;;BR IF MORE TO DO
        BLT     6$              ;;BR IF DONE
        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
        BR      2$              ;;GO DO THE LAST DIGIT
6$:     MOV     (SP)+,R5        ;;RESTORE R5
        MOV     (SP)+,R4        ;;RESTORE R4
        MOV     (SP)+,R3        ;;RESTORE R3
        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
        MOV     (SP)+,(SP)
        RTI                     ;;RETURN
8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
$OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
$OFILL: .BYTE   0               ;;ZERO FILL SWITCH
$OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
.ENDM   .$TYPOCT
```

```
.MACRO  .$TYPDEC
.SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
STARS
;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
;*REPLACED WITH SPACES.
;*CALL:
;*       MOV     NUM,-(SP)          ;;PUT THE BINARY NUMBER ON THE STACK
;*       TYPDS                      ;;GO TO THE ROUTINE
$TYPDS:  PUSH    <R0,R1,R2,R3,R4,R5>
         MOV     #20200,-(SP)       ;;SET BLANK SWITCH AND SIGN
         MOV     22(SP),R5          ;;GET THE INPUT NUMBER
         BPL     1$                 ;;BR IF INPUT IS POS.
         NEG     R5                 ;;MAKE THE BINARY NUMBER POS.
         MOVB    #'-,1(SP)          ;;MAKE THE ASCII NUMBER NEG.
1$:      CLR     R0                 ;;ZERO THE CONSTANTS INDEX
         MOV     PC,R3              ;;FETCH PC FOR ADDRESSING
         ADD     #$DBLK-.,R3        ;;SETUP THE OUTPUT POINTER - R3 POINTS TO $DBLK
                                    ;;TABLE
         MOVB    #' ,(R3)+          ;;SET THE FIRST CHARACTER TO A BLANK
2$:      CLR     R2                 ;;CLEAR THE BCD NUMBER
         MOV     PC,R4              ;;USE PC FOR LOCATING TABLE
         ADD     #$DTBL-.,R4        ;;R4 NOW POINTS TO TABLE
         ADD     R0,R4              ;;INDEX INTO TABLE
         MOV     (R4),R1            ;;GET THE CONSTANT
3$:      SUB     R1,R5              ;;FORM THIS BCD DIGIT
         BLT     4$                 ;;BR IF DONE
         INC     R2                 ;;INCREASE THE BCD DIGIT BY 1
         BR      3$
4$:      ADD     R1,R5              ;;ADD BACK THE CONSTANT
         TST     R2                 ;;CHECK IF BCD DIGIT=0
         BNE     5$                 ;;FALL THROUGH IF 0
         TSTB    (SP)               ;;STILL DOING LEADING 0'S?
         BMI     7$                 ;;BR IF YES
5$:      ASLB    (SP)               ;;MSD?
         BCC     6$                 ;;BR IF NO
         MOVB    1(SP),-1(R3)       ;;YES--SET THE SIGN
6$:      BIS     #'0,R2             ;;MAKE THE BCD DIGIT ASCII
7$:      BIS     #' ,R2             ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
         MOVB    R2,(R3)+           ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
         TST     (R0)+              ;;JUST INCREMENTING
         CMP     R0,#10             ;;CHECK THE TABLE INDEX
         BLT     2$                 ;;GO DO THE NEXT DIGIT
         BGT     8$                 ;;GO TO EXIT
         MOV     R5,R2              ;;GET THE LSD
         BR      6$                 ;;GO CHANGE TO ASCII
8$:      TSTB    (SP)+              ;;WAS THE LSD THE FIRST NON-ZERO?
         BPL     9$                 ;;BR IF NO
         MOVB    -1(SP),-2(R3)      ;;YES--SET THE SIGN FOR TYPING
9$:      CLRB    (R3)               ;;SET THE TERMINATOR
         POP     <R5,R4,R3,R2,R1,R0>
         MOV     PC,-(SP)           ;;FETCH ADDRESS OF NUMBER TO BE TYPED USING
                                    ;;POSITION INDEPENDENT CODE
         ADD     #$DBLK-.,(SP)      ;;STACK NOW CONTAINS ADDRESS OF NUMBER TO BE
                                    ;;PRINTED
```

```
                MOV     (SP)+,10$       ;;SET UP ADDRESS FOR TYPE COMMAND
                TYPE                    ;;NOW TYPE THE NUMBER
        10$:    .WORD   $DBLK
                MOV     2(SP),4(SP)     ;;ADJUST THE STACK
                MOV     (SP)+,(SP)
                RTI                     ;;RETURN TO USER
        $DTBL:  10000.
                1000.
                100.
                10.
        $DBLK:  .BLKW   4
        .ENDM   .$TYPDEC
```

```
.MACRO  .$TRAP  X,Y,Z
.SBTTL  TRAP DECODER
STARS
;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
;*GO TO THAT ROUTINE.
.IF B X
$TRAP:  MOV     R0,-(SP)        ;;SAVE R0
        MOV     R1,-(SP)        ;;SAVE R1
.IFF
$TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
        BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
        MOV     #1$,-(SP)       ;; T-BIT TRAPS
        RTI                     ;;SET THE NEW STATUS
1$:     MOV     R0,-(SP)        ;;SAVE R0
.ENDC
        MOV     4(SP),R0        ;;GET TRAP ADDRESS
        TST     -(R0)           ;;BACKUP BY 2
        MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
.IF NB Z
        BPL     $TRAP1          ;;NON-USER TRAP,BELOW 200
        BIC     #^C177,R0       ;;STRIP AWAY THE JUNK
        JMP     (PC)            ;;USER TRAP,ABOVE 177, GO TO
        .WORD   Z               ;; USER TRAP HANDLER-  Z
$TRAP1:
.ENDC
.IF NB Y
        CMP     #$TERM,R0       ;;CHECK FOR OUT OF BOUNDS
        BGT     .+6             ;;BR IF OK
        HALT                    ;;OUT OF BOUNDS
        BR      .-2             ;;HANGUP
.ENDC
        ASL     R0              ;;POSITION FOR INDEXING
        MOV     PC,R1           ;;FETCH THIS PROGRAM POINTER FOR PIC
        ADD     #$TRPAD-.,R1    ;;POINT TO TABLE
        ADD     R0,R1           ;;R1 NOW POINTS TO ROUTINE TRAP CALL WANTED
        MOV     (R1),R0         ;;R0 CONTAINS ADDRESS FOR RTS
        MOV     (SP)+,R1        ;;RESTORE R1
        RTS     R0              ;;GO TO ROUTINE
;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
$TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
        MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
        RTI                     ;;RESTORE THE PSW
.MACRO  SETTRAP A,B,MSG
        $$SET   A,B,\<TRAP+$TRP>,\$TRP,<MSG>
.NLIST
$TRP=$TRP+1
.LIST
.ENDM   SETTRAP
.MACRO  $$SET   A,B,C,D,COMNT
.IF EQ $TRP-1
.SBTTL  TRAP TABLE
;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;*BY THE "TRAP" INSTRUCTION.
;       ROUTINE
;       -------
```

```
$TRPAD: .WORD   $TRAP2
.ENDC
.IIF NDF GNS,.NLIST
        A=      C
.IIF NDF GNS,.LIST
        B       ;;CALL=A        TRAP+D(C)       COMNT
.ENDM   $$SET
.MACRO  TRMTRP
$TERM=.-$TRPAD
.ENDM   TRMTRP
.NLIST
$TRP=1
.LIST
.IF DF $TYPE
        SETTRAP TYPE,$TYPE,^/TTY TYPEOUT ROUTINE/
.ENDC
.IF DF $TYPOC
        SETTRAP TYPOC,$TYPOC,^/TYPE OCTAL NUMBER (WITH LEADING ZEROS)/
        SETTRAP TYPOS,$TYPOS,^/TYPE OCTAL NUMBER (NO LEADING ZEROS)/
        SETTRAP TYPON,$TYPON,^/TYPE OCTAL NUMBER (AS PER LAST CALL)/
.ENDC
.IF DF $TYPDS
        SETTRAP TYPDS,$TYPDS,^/TYPE DECIMAL NUMBER (WITH SIGN)/
.ENDC
.IF DF $TYPBN
        SETTRAP TYPBN,$TYPBN,^/TYPE BINARY (ASCII) NUMBER/
.ENDC
.IF DF $GTSWR
        SETTRAP GTSWR,$GTSWR,^/GET SOFT-SWR SETTING/
.ENDC
.IF DF $CKSWR
        SETTRAP CKSWR,$CKSWR,^/TEST FOR CHANGE IN SOFT-SWR/
.ENDC
.IF DF $RDCHR
        SETTRAP RDCHR,$RDCHR,^/TTY TYPEIN CHARACTER ROUTINE/
.ENDC
.IF DF $RDLIN
        SETTRAP RDLIN,$RDLIN,^/TTY TYPEIN STRING ROUTINE/
.ENDC
.IF DF $RDOCT
        SETTRAP RDOCT,$RDOCT,^/READ AN OCTAL NUMBER FROM TTY/
.ENDC
.IF DF $RDDEC
        SETTRAP RDDEC,$RDDEC,^/READ A DECIMAL NUMBER FROM TTY/
.ENDC
.IF DF $SAVREG
        SETTRAP SAVREG,$SAVREG,^/SAVE R0-R5 ROUTINE/
        SETTRAP RESREG,$RESREG,^/RESTORE R0-R5 ROUTINE/
.ENDC
.IF DF  $R2A
        SETTRAP R2AZ,$R2AZ
        SETTRAP R2AZ.,$R2AZ.
        SETTRAP R2AZQ,$R2AZQ
.ENDC
.ENDM   .$TRAP
```

```
     
000000                                   .HEADER <CVMEMAO DIAGNOSTIC FOR NON-VOLATILE MEMORIES>,<1981>,<KEN LANGLAIS>
000000                                   .EQUAT
000000                                   .SWRHI
000000                                   .$CATCH START1
000204                                   .KT11    ,X
     
000204                                   .SETUP   .$TRAP
                                         .NLIST   BEX
                                         .LIST    MEB
         000001                          $LSTIN   = 1
         000001                          $LSTTAG  = 1
         000400                          .        = 400          ;READ ROUTINES
     
000400                                   .$READ   26
000400   177560                .IIF EQ .-$TKS,$TKS:     .WORD    177560          ;;TTY KBD STATUS
000402   177562                .IIF EQ .-$TKB,$TKB:     .WORD    177562          ;;TTY KBD BUFFER
000404   011646                $RDCHR:   MOV      (SP),-(SP)     ;;PUSH DOWN THE PC
000406   016666 000004 000002            MOV      4(SP),2(SP)    ;;SAVE THE PS
000414   105777 177760         1$:       TSTB     @$TKS          ;;WAIT FOR
000420   100375                          BPL      1$             ;;A CHARACTER
000422   117766 177754 000004            MOVB     @$TKB,4(SP)    ;;READ THE TTY
000430   042766 177600 000004            BIC      #^C<177>,4(SP) ;;GET RID OF JUNK IF ANY
000436   026627 000004 000023            CMP      4(SP),#23      ;;IS IT A CONTROL-S?
000444   001013                          BNE      3$             ;;BRANCH IF NO
000446   105777 177726         2$:       TSTB     @$TKS          ;;WAIT FOR A CHARACTER
000452   100375                          BPL      2$             ;;LOOP UNTIL ITS THERE
000454   117746 177722                   MOVB     @$TKB,-(SP)    ;;GET CHARACTER
000460   042716 177600                   BIC      #^C177,(SP)    ;;MAKE IT 7-BIT ASCII
000464   022627 000021                   CMP      (SP)+,#21      ;;IS IT A CONTROL-Q?
000470   001366                          BNE      2$             ;;IF NOT DISCARD IT
000472   000750                          BR       1$             ;;YES, RESUME
000474   026627 000004 000021  3$:       CMP      4(SP),#$XON    ;;IS IT A RANDOM XON?                 ;RAN001
000502   001744                          BEQ      1$             ;;BRANCH IF YES                       ;RAN001
000504   026627 000004 000140            CMP      4(SP),#140     ;;IS IT UPPER CASE?
000512   002407                          BLT      4$             ;;BRANCH IF YES
000514   026627 000004 000175            CMP      4(SP),#175     ;;IS IT A SPECIAL CHAR?
000522   003003                          BGT      4$             ;;BRANCH IF YES
000524   042766 000040 000004            BIC      #40,4(SP)      ;;MAKE IT UPPER CASE
000532   000002                4$:       RTI                     ;;GO BACK TO USER
000534   010346                $RDLIN:   MOV      R3,-(SP)       ;;SAVE R3
000536   012703 000642         1$:       MOV      #$TTYIN,R3     ;;GET ADDRESS
000542   022703 000670         2$:       CMP      #$TTYIN+26,R3  ;;BUFFER FULL?
000546   101405                          BLOS     4$             ;;BR IF YES
000550   104406                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
000552   112613                          MOVB     (SP)+,(R3)     ;;GET CHARACTER
000554   122713 000177         10$:      CMPB     #177,(R3)      ;;IS IT A RUBOUT
000560   001003                          BNE      3$             ;;SKIP IF NOT
000562   104401 000670         4$:       TYPE     ,$QUES         ;;TYPE A '?'
000566   000763                          BR       1$             ;;CLEAR THE BUFFER AND LOOP
000570   111367 000044         3$:       MOVB     (R3),9$        ;;ECHO THE CHARACTER
000574   104401 000640                   TYPE     ,9$
000600   122723 000015                   CMPB     #15,(R3)+      ;;CHECK FOR RETURN
000604   001356                          BNE      2$             ;;LOOP IF NOT RETURN
000606   105063 177777                   CLRB     -1(R3)         ;;CLEAR RETURN (THE 15)
000612   104401 000672                   TYPE     ,$LF           ;;TYPE A LINE FEED
000616   012603                          MOV      (SP)+,R3       ;;RESTORE R3
```

```
        000620  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
        000622  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
        000630  012766  000642  000004          MOV     #$TTYIN,4(SP)
        000636  000002                          RTI                     ;;RETURN
        000640     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
        000641     000                          .BYTE   0               ;;TERMINATOR
        000670     077                  .IIF EQ .-$QUES,$QUES:  .ASCII  "?"     ;;QUESTION MARK
        000671     015                  .IIF EQ .-$CRLF,$CRLF:  .ASCII  <15>    ;CARRIAGE RETURN
        000672     012     000          .IIF EQ .-$LF,$LF:      .ASCIZ  <12>    ;;LINE FEED
        000674     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
        000701     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
        000706     015     012     123  $MSWR:  .ASCIZ  <15><12>/SWR = /
        000717     040     040     116  $MNEW:  .ASCIZ  /  NEW = /
 1432
 1433   000730                                  .$RDOCT BIG
        000730  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
        000732  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
        000740  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
        000742  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
        000744  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
        000746  104407                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
        000750  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
        000752  010067  000100                  MOV     R0,5$           ;;AND SAVE IT
        000756  005001                          CLR     R1              ;;CLEAR DATA WORD
        000760  005002                          CLR     R2
        000762  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
        000764  001420                          BEQ     3$              ;;IF ZERO GET OUT
        000766  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
        000772  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
        000774  122716  000067                  CMPB    #'7,(SP)
        001000  002423                          BLT     4$
        001002  006301                          ASL     R1              ;;*2
        001004  006102                          ROL     R2
        001006  006301                          ASL     R1              ;;*4
        001010  006102                          ROL     R2
        001012  006301                          ASL     R1              ;;*8
        001014  006102                          ROL     R2
        001016  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
        001022  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
        001024  000756                          BR      2$              ;;LOOP
        001026  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
        001030  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
        001034  010267  000026                  MOV     R2,$HIOCT
        001040  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
        001042  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
        001044  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
        001046  000002                          RTI                     ;;RETURN
        001050  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
        001052  105010                          CLRB    (R0)            ;;SET A TERMINATOR
        001054  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
        001056  000000                  5$:     .WORD   0
        001060  104401  000670                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
        001064  000730                          BR      1$              ;;TRY AGAIN
        001066  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
 1434
 1435   001070                                  .$RDDEC
        001070  011646                  $RDDEC: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR
```

```
      001072 016666 000004 000002           MOV     4(SP),2(SP)        ;;THE INPUT NUMBER
      001100 010046                         MOV     R0,-(SP)           ;;PUSH R0 ON STACK
      001102 010146                         MOV     R1,-(SP)           ;;PUSH R1 ON STACK
      001104 010246                         MOV     R2,-(SP)           ;;PUSH R2 ON STACK
      001106 104407                 1$:     RDLIN                      ;;READ AN ASCIZ LINE
      001110 012600                         MOV     (SP)+,R0           ;;ADDRESS OF 1ST CHAR.
      001112 010067 000120                  MOV     R0,6$              ;;SAVE INCASE OF BAD INPUT
      001116 005046                         CLR     -(SP)              ;;CLEAR DATA WORD
      001120 005002                         CLR     R2                 ;;SIGN SET POSITIVE
      001122 122710 000055                  CMPB    #'-,(R0)           ;;SEE IF A MINUS SIGN WAS TYPED
      001126 001001                         BNE     2$                 ;;BR IF NO MINUS SIGN
      001130 112002                         MOVB    (R0)+,R2           ;;SAVE FOR LATER USE
      001132 112001                 2$:     MOVB    (R0)+,R1           ;;PICKUP THIS CHARACTER
      001134 001424                         BEQ     3$                 ;;GET OUT IF ZERO
      001136 122701 000060                  CMPB    #'0,R1             ;;MAKE SURE THIS CHARACTER
      001142 003032                         BGT     5$                 ;;IS A DIGIT BETWEEN 0 & 9
      001144 122701 000071                  CMPB    #'9,R1
      001150 002427                         BLT     5$
      001152 032716 170000                  BIT     #^C7777,(SP)       ;;DON'T LET NUMBEP GET TO BIG
      001156 001024                         BNE     5$                 ;;BR IF NUMBER WOULD OVERFLOW
      001160 006316                         ASL     (SP)               ;;*2
      001162 011646                         MOV     (SP),-(SP)         ;;SAVE FOR LATER
      001164 006316                         ASL     (SP)               ;;*4
      001166 006316                         ASL     (SP)               ;;*8.
      001170 062616                         ADD     (SP)+,(SP)         ;;*10.
      001172 102416                         BVS     5$                 ;;OVERFLOW ISN'T ALLOWED
      001174 162701 000060                  SUB     #'0,R1             ;;STRIP AWAY THE ASCII JUNK
      001200 060116                         ADD     R1,(SP)            ;;ADD IN THIS DIGIT
      001202 102412                         BVS     5$                 ;;OVERFLOW ISN'T ALLOWED
      001204 000752                         BR      2$                 ;;LOOP
      001206 005702                 3$:     TST     R2                 ;;CHECK IF NUMBER IS NEG
      001210 001401                         BEQ     4$                 ;;BR IF NO
      001212 005416                         NEG     (SP)               ;;YES--NEGATE THE NUMBER
      001214 012666 000012          4$:     MOV     (SP)+,12(SP)       ;;SAVE THE RESULT
      001220 012602                         MOV     (SP)+,R2           ;;POP STACK INTO R2
      001222 012601                         MOV     (SP)+,R1           ;;POP STACK INTO R1
      001224 012600                         MOV     (SP)+,R0           ;;POP STACK INTO R0
      001226 000002                         RTI                        ;;RETURN
      001230 005726                 5$:     TST     (SP)+              ;;CLEAN PARTIAL NUMBER FROM STACK
      001232 105010                         CLRB    (R0)               ;;SET A TERMINATOR
      001234 104401                         TYPE                       ;;TYPE THE INPUT UP TO BAD CHAR.
      001236 000000                 6$:     .WORD   0                  ;;POINTER GOES HERE
      001240 104401 000670                  TYPE    ,$QUES             ;;"?" "CR" &"LF"
      001244 000720                         BR      1$                 ;;TRY AGAIN
1436
1437  001246                                .$SIZE
      001246 010046                 $SIZE:  MOV     R0,-(SP)           ;;SAVE R0 ON THE STACK
      001250 010146                         MOV     R1,-(SP)           ;;SAVE R1 ON THE STACK
      001252 010246                         MOV     R2,-(SP)           ;;SAVE R2 ON THE STACK
      001254 010346                         MOV     R3,-(SP)           ;;SAVE R3 ON THE STACK
      001256 010446                         MOV     R4,-(SP)           ;;SAVE R4 ON THE STACK              BK001
      001260 013746 000114                  MOV     @#114,-(SP)        ;;SAVE MEMORY ERROR VECTOR PS & PC
      001264 013746 000116                  MOV     @#116,-(SP)
      001270 012737 000116 000114           MOV     #116,@#114         ;;IGNORE PARITY ERRORS WHILE SIZING
      001276 012737 000002 000116           MOV     #RTI,@#116
      001304 013746 000004                  MOV     @#ERRVEC,-(SP)     ;;SAVE PRESENT ERROR VECTOR PS & PC
      001310 013746 000006                  MOV     @#ERRVEC+2,-(SP)
```

```
001314  010600                          MOV     SP,R0            ;;SAVE THE STACK POINTER
001316  104400                          TRAP                     ;;PUSH OLD PSW AND PC ON STACK
001320  012637  000006                  MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
001324  012701  003776                  MOV     #3776,R1         ;;SETUP ADDRESS
001330  105727                          TSTB    (PC)+            ;;USE MEMORY MANAGEMENT?
001332  000200                  $KT11:  .WORD   200              ;;SET TO USE MEMORY MANAGEMENT
001334  100135                          BPL     $CORE            ;;BR IF NO
001336  012737  001622  000004          MOV     #$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
001344  005737  177572                  TST     @#SR0            ;;KT11 ARE YOU THERE?
001350  052767  100000  177754          BIS     #100000,$KT11    ;;YES--SET KT11 KEY
001356  012737  001406  000004          MOV     #100$,@#ERRVEC   ;;SET FOR TIMEOUT                     BK001
001364  005737  172516                  TST     @#172516         ;;Q-BUS MAP ARE YOU THERE?            BK001
001370  012767  000200  000026          MOV     #200,$MAP        ;;TURN ON MAP INDICATOR               BK001
001376  012767  176200  000022          MOV     #176200,$STOP    ;;END OF 2M OF MEMORY
001404  000411                          BR      $MAPRG           ;;GO SET UP MAP REGISTERS             BK001
001406  012737  006200  001426  100$:   MOV     #6200,@#$STOP    ;;COMPARISON VALUE FOR 18 BIT MAPPING BK001
001414  022626                          CMP     (SP)+,(SP)+      ;;CLEAN OFF STACK                     BK001
001416  005037  001424                  CLR     @#$MAP           ;;MAKE SURE MAP INDICATOR TURNED OFF  BK001
001422  000402                          BR      $NOMAP           ;;                            BK001
001424  000000                  $MAP:   .WORD   0                ;;=200 IF MAP PRESENT                 BK001
001426  000000                  $STOP:  .WORD   0                ;;FILLED WITH APPROPRIATE COMPARISON VALUE        BK001
001430  005046                          CLR     -(SP)            ;;INITIALIZE FOR 'PAR'' LOADING
001432  012702  172340                  MOV     #KIPAR0,R2       ;;ADDRESS OF FIRST 'PAR''
001436  012703  000010                  MOV     #^D8,R3          ;;LOAD EIGHT 'PAR.'S'' AND EIGHT 'PDR.'S''
001442  012762  077406  177740  1$:     MOV     #77406,-40(R2)   ;;PDR = 4K, UP, READ/WRITE
001450  011622                          MOV     (SP),(R2)+       ;;LOAD 'PAR''
001452  062716  000200                  ADD     #200,(SP)        ;;UPDATE FOR NEXT 'PAR''
001456  077307                          SOB     R3,1$            ;;LOOP UNTIL ALL EIGHT ARE LOADED
001460  012742  177600                  MOV     #177600,-(R2)    ;;SETUP KIPAR7 FOR I/O
001464  005042                          CLR     -(R2)            ;;SETUP KIPAR6 FOR TESTING
001466  012737  001504  000004          MOV     #2$,@#ERRVEC     ;;CATCH TIMEOUT IF NO SR3
001474  012737  000020  172516          MOV     #20,@#SR3        ;;ENABLE 22 BIT MODE AND UNIBUS MAP   BK001
001502  000401                          BR      3$               ;;THIS PDP-11 HAS A SR3 REGISTER
001504  022626                  2$:     CMP     (SP)+,(SP)+      ;;CLEAN OFF THE STACK--NO SR3
001506  005237  177572          3$:     INC     @#SR0            ;;TURN ON MEMORY MANAGEMENT
001512  012737  001560  000004          MOV     #$KTOUT,@#ERRVEC ;;SET FOR TIME OUT
001520  105737  001424                  TSTB    @#$MAP           ;;IS MAP THERE?                       BK001
001524  100006                          BPL     4$               ;;NO-SKIP                             BK001
001526  012737  001602  000114          MOV     #$MMOUT,@#114    ;;SET UP MEMORY ERROR VECTOR          BK001
001534  013737  000006  000116          MOV     @#ERRVEC+2,@#116 ;;LOCK OUT INTERRUPTS         BK001
001542  005737  143776          4$:     TST     @#143776         ;;TRAP ON NON-EX-MEM
001546  062712  000040                  ADD     #40,(R2)         ;;MAKE A 1K STEP
001552  023712  001426                  CMP     @#$STOP,(R2)     ;;LAST ONE?
001556  101371                          BHI     4$               ;;NO--TRY IT
001560  011202                  $KTOUT: MOV     (R2),R2          ;;GET LAST BANK+1
001562  005037  177572                  CLR     @#SR0            ;;TURN OFF MEMORY MANAGEMENT
001566  105737  001424                  TSTB    @#$MAP           ;;IS MAP THERE?                       BK001
001572  100034                          BPL     $SIZEX           ;;NO-SKIP                             BK001
001574  005037  172516                  CLR     @#SR3            ;;TURN OFF MAP                        BK001
001600  000431                          BR      $SIZEX
001602  013704  177744          $MMOUT: MOV     @#177744,R4      ;;SAVE MEMORY ERROR REGISTER          BK001
001606  010437  177744                  MOV     R4,@#177744      ;;CLEAR BITS IN REGISTER              BK001
001612  032704  000001                  BIT     #1,R4            ;;MEMORY TIMEOUT?                     BK001
001616  001360                          BNE     $KTOUT           ;;YES-EXIT                            BK001
001620  000002                          RTI                      ;;MUST BE PARITY ERROR-IGNORE IT      BK001
001622  042767  100000  177502  $KTNEX: BIC     #100000,$KT11    ;;KT11 NON-EXISTENT
001630  012737  001660  000004  $CORE:  MOV     #$CROUT,@#ERRVEC ;;SET FOR TIMEOUT
```

```
        001636  005002                        CLR     R2              ;;SET UP BANK
        001640  062701  004000        1$:     ADD     #4000,R1        ;;INCREMENT BY 1K
        001644  062702  000040                ADD     #40,R2          ;;1K STEP
        001650  005711                        TST     (R1)            ;;TRAP ON TIME OUT
        001652  022701  177776                CMP     #177776,R1      ;;LAST ONE
        001656  001370                        BNE     1$              ;;NO--TRY AGAIN
        001660  162701  004000        $CROUT: SUB     #4000,R1
        001664  162702  000040        $SIZEX: SUB     #40,R2          ;;DROP BACK
        001670  010006                        MOV     R0,SP           ;;RESTORE THE STACK
        001672  012637  000006                MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
        001676  012637  000004                MOV     (SP)+,@#ERRVEC
        001702  012637  000116                MOV     (SP)+,@#116     ;;RESTORE MEMORY ERROR VECTOR
        001706  012637  000114                MOV     (SP)+,@#114
        001712  010167  000020                MOV     R1,$LSTAD       ;;LAST ADDRESS
        001716  010267  000016                MOV     R2,$LSTBK       ;;LAST BANK
        001722  012604                        MOV     (SP)+,R4        ;;RESTORE R4                    BK001
        001724  012603                        MOV     (SP)+,R3        ;;RESTORE R3
        001726  012602                        MOV     (SP)+,R2        ;;RESTORE R2
        001730  012601                        MOV     (SP)+,R1        ;;RESTORE R1
        001732  012600                        MOV     (SP)+,R0        ;;RESTORE R0
        001734  000207                        RTS     PC
        001736  000000                $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
        001740  000000                $LSTBK: .WORD   0               ;;CONTAINS THE LAST BANK
```

```
                                        ;THIS ROUTINE FETCHES A CHARACTER FROM THE TTY AND ECHOS IT

001742                                  ROUTINE FETCH
001742                                                                                  FETCH:

001742  104406                                  RDCHR                   ;FETCH ANSWER

001744                                          POP     @#ECHO          ;FETCH CHARACTER
001744  012637  010342                          MOV     (SP)+,@#ECHO    ;;POP STACK INTO @#ECHO

001750  104401  010342                          TYPE    ,ECHO           ;ECHO CHARACTER

001754  104401  016155                          TYPE    ,CRLF1          ;PRINT <CR><LF>

001760                                  ENDRTN
001760                                                                                  50000$:
001760                                                                                  50001$:
001760  000207                                                                          RTS     PC


```

```
                                         ;THIS ROUTINE READS THE ADDRESS TYPED, DETERMINES THAT IT IS A LEGAL 4K
                                         ;ADDRESS (I.E. DOES NOT RESIDE OUT SIDE OF THE SYSTEM ADDRESS SPACE) AND
                                         ;RETURNS WITH R4 NE #0 IF ADDRESS IS LEGAL


001762                                   ROUTINF FETADD
001762                                                                                  FETADD:

001762  104410                                  RDOCT                           ;READ OCTAL NUMBER

001764                                          POP     @#ECHO                  ;FETCH BITS 0 - 15
001764  012637  010342                          MOV     (SP)+,@#ECHO    ;;POP STACK INTO @#ECHO

001770                                          IF      #17777 SETIN @#ECHO THEN        ;CHECK FOR 4K
001770  032737  017777  010342                                                          BIT     #17777,@#ECHO
001776  001403                                                                          BEQ     50002$
                                                                                ;BOUNDARY
002000  104401  010152                                  TYPE    ,NOT4K          ;NOT LEGAL ADDRESS

002004                                          ELSE
002004  000432                                                                          BR      50003$
002006                                                                                  50002$:

002006                                                  LET     @#SAVLOW := @#ECHO      ;SAVE BITS 0 - 15 OF ADDRESS
002006  013737  010342  010344                                                          MOV     @#ECHO,@#SAVLOW

002014                                                  LET     @#SAVHI :- @#$HIOCT     ;SAVE BITS 16 - 22 OF ADDRESS
002014  013737  001066  010346                                                          MOV     @#$HIOCT,@#SAVHI

                                                        ;BUMP TO NEXT 4K IN ORDER TO DETERMINE IF THERE IS ENOUGH
                                                        ;MEMORY FOR THE PROGRAM AND TESTING

002022                                                  LET @#ECHO := @#ECHO + #20000   ;ADD 4K
002022  062737  020000  010342                                                          ADD     #20000,@#ECHO

002030                                                  IF #1 SETIN CONTROL THEN
002030  032767  000001  014122                                                          BIT     #1,CONTROL
002036  001402                                                                          BEQ     50004$

002040                                                          LET @#$HIOCT :- @#$HIOCT + CARRY        ;ADD OVERFLOW
002040  005537  001066                                                                  ADC     @#$HIOCT

002044                                                  ENDIF
002044                                                                                  50004$:

002044                                                  CALL ADDCHK
002044  004767  000024                                                                  JSR     PC,ADDCHK

002050                                                  IF #1 SETIN CONTROL THEN        ;MAPPED?
002050  032767  000001  014102                                                          BIT     #1,CONTROL
002056  001403                                                                          BEQ     50005$

002060                                                          LET R3 := R3 - #200     ;REMOVE 4K TEST INCREMENT
002060  162703  000200                                                                  SUB     #200,R3

002064                                                  ELSE
002064  000402                                                                          BR      50006$
```

```
        002066                                                                                   50005$:
   1494
   1495 002066                                                          LET R3 := R3 - #20000   ;REMOVE 4K TEST INCREMENT
        002066  162703  020000                                                                   SUB     #20000,R3
   1496
   1497 002072                                                  ENDIF
        002072                                                                                   50006$:
   1498
   1499 002072                                          ENDIF
        002072                                                                                   50003$:
   1500
   1501 002072                                  ENDRTN
        002072                                                                                   50000$:
        002072                                                                                   50001$:
        002072  000207                                                                           RTS     PC
```

```
                                    ;THIS ROUTINE CHECKS THAT THE 4K BOUNDARY IS WITHIN IN THE SYSTEM
                                    ;R3 WILL EITHER BE PAR COMPATIBLE ADDRESS FOR A MAPPED SYSTEM OR 4K ADDRESS
                                    ;FOR AN UNMAPPED SYSTEM

002074                              ROUTINE ADDCHK
002074                                                                        ADDCHK:

002074                                        IF #1 SETIN CONTROL THEN                 ;DO WE HAVE MEM. MAN
002074  032767  000001  014056                                                BIT     #1,CONTROL
002102  001431                                                                BEQ     50002$
                                                                         ;YES
002104                                                INCR R3 FROM #1 TO #3 BY #1    ;MAKE ADDRESS IN ECHO AND $HIOCT
002104  012703  000001                                                        MOV     #1,R3
002110  000401                                                                BR      50003$
002112                                                                        50004$:
002112  005203                                                                INC     R3
002114                                                                        50003$:
002114  020327  000003                                                        CMP     R3,#3
002120  003005                                                                BGT     50005$
                                                                         ;COMPATIBLE WITH PAR

002122                                                LET @#ECHO := @#ECHO ROTATE 1
002122  006137  010342                                                        ROL     @#ECHO

002126                                                LET @#$HIOCT := @#$HIOCT ROTATE 1
002126  006137  001066                                                        ROL     @#$HIOCT

002132                                                ENDINC
002132  000767                                                                BR      50004$
002134                                                                        50005$:

002134                                                INCR R3 FROM #1 TO #7 BY #1    ;NOW HIOCT WILL TAKE LOWER BITS
002134  012703  000001                                                        MOV     #1,R3
002140  000401                                                                BR      50006$
002142                                                                        50007$:
002142  005203                                                                INC     R3
002144                                                                        50006$:
002144  020327  000007                                                        CMP     R3,#7
002150  003003                                                                BGT     50010$
                                                                         ;AND PUT THEM INTO UPPER BITS

002152                                                        LET @#$HIOCT :- @#$HIOCT SHIFT 1
002152  006337  001066                                                        ASL     @#$HIOCT

002156                                                ENDINC
002156  000771                                                                BR      50007$
002160                                                                        50010$:

002160                                                LET R3 := @#$HIOCT
002160  013703  001066                                                        MOV     @#$HIOCT,R3

002164                                      ELSE                                 ;NO MEMORY MANAGEMENT
002164  000410                                                                BR      50011$
002166                                                                        50002$:

002166                                                LET R3 :- @#ECHO                 ;FETCH STARTING ADDRESS
002166  013703  010342                                                        MOV     @#ECHO,R3
```

```

002172                                          IF @#$HIOCT NE #0 THEN            ;ADDRESS TO HIGH
002172  005737  001066                                                                      TST     @#$HIOCT
002176  001403                                                                              BEQ     50012$

002200  104401  010224                                  TYPE     ,OUTMEM          ;TELL OPERATOR THAT ADDRESS
                                                                                  ;OUT OF MEMORY

002204                                                  INLINE <BR 1$>            ;EXIT FROM ROUTINE
002204  000407                                                                              BR 1$

002206                                          ENDIF
002206                                                                                      50012$:

002206                                  ENDIF
002206                                                                                      50011$:

002206                                  IF R3 HI ENDADD THEN                      ;OUTSIDE OF MEMORY
002206  020367  013762                                                                      CMP     R3,ENDADD
002212  101403                                                                              BLOS    50013$
                                                                                  ;IF R3 IS GREATER THAN R2
002214  104401  010224                          TYPE     ,OUTMEM          ;PRINT STARTING ADDRESS OUTSIDE
                                                                                  ;OF ADDRESS RANGE

002220                                  ELSE
002220  000401                                                                              BR      50014$
002222                                                                                      50013$:

002222                                          LET      R4 :- R4 + #1            ;INDICATE OK
002222  005204                                                                              INC     R4

002224                                  ENDIF
002224                                                                                      50014$:

002224                          1$: ENDRTN
002224                                                                                      50000$:
002224                                                                                      50001$:
002224  000207                                                                              RTS     P
```

```
002226   012   012   015  HELPF:  .ASCII  <12><12><15>'                      FUNCTIONAL DESCRIPTION'<12><12><15>
002305   040   124   110    .ASCII ' THE PURPOSE OF THIS PROGRAM IS TO TEST THE DATA RETENTION OF'<12><15>
002404   040   116   117    .ASCII ' NON-VOLATILE MODULES IN ANY QBUS SYSTEM. THIS IS NOT A MEMORY'<12><15>
002504   040   104   111    .ASCII ' DIAGNOSTIC. PLEASE RUN CZKMA OR VMSA DIAGNOSTICS BEFORE RUNNING THIS'<12><15>
002613   040   120   122    .ASCII ' PROGRAM. THE FOLLOWING IS A BRIEF DISCRIPTION OF THE PROGRAM FLOW:'<12><12><15>
002721   040   040   040    .ASCII '      1. ASK THE OPERATOR A SERIES OF QUESTIONS (MENU)'<12><15>
003010   040   040   040    .ASCII '      2. RELOCATE PROGRAM TO NON-VOLATILE AREA (IF BANK 0 IS VOLATILE)'<12><15>
003117   040   040   040    .ASCII '      3. GENERATE AND PRINT A RESTART HELP FILE'<12><15>
003177   040   040   040    .ASCII '      4. WRITE A BACKGROUND PATTERN (125252) THROUGHOUT MEMORY'<12><15>
003276   040   040   040    .ASCII '      5. DO A CHECKSUM OF THE PROGRAM'<12><15>
003344   040   040   040    .ASCII '      6. DO A CHECKSUM OF THE TRAP VECTOR SPACE (ADDRESS 0-376)'<12><15>
003444   040   040   040    .ASCII '      7. TELL THE OPERATOR TO POWERDOWN THE SYSTEM'<12><12><15>
003530   124   110   105    .ASCII 'THE OPERATOR WILL NOW FOLLOW THE RESTART HELP FILE SOME TIME BETWEEN'<12><15>
003636   062   040   115    .ASCII '2 MINUTES AND 100 HOURS. THE FOLLOWING IS THE FLOW OF THE PROGRAM'<12><15>
003741   101   106   124    .ASCII 'AFTER POWER UP:'<12><12><15>
003763   040   040   040    .ASCII '      1. DO ANOTHER CHECKSUM OF THE PROGRAM AND CHECK IT AGAINST THE'<12><15>
004070   040   040   040    .ASCII '         ONE DONE BEFORE POWER DOWN.'<12><15>
004135   040   040   040    .ASCII '      2. DO ANOTHER CHECKSUM OF THE TRAP VECTOR SPACE AND CHECK IT AGAINST'<12><15>
004250   040   040   040    .ASCII '         THE ONE DONE BEFORE POWER DOWN. (NO TRAP CHECKSUM ERROR WILL BE'<12><15>
004361   040   040   040    .ASCII '         REPORTED UNLESS BANK 0 WAS DETERMINED NON-VOLATILE)'<12><15>
004456   040   040   040    .ASCII '      3. CHECK ENTIRE MEMORY FOR THE BACKGROUND PATTERN (125252)'<12><15>
004557   040   040   040    .ASCII '      4. PRINT A MEMORY MAP OF ALL VOLATILE AND NON VOLATILE MEMORY'<12><15>
004663   040   040   040    .ASCII '         IN THE SYSTEM'<12><12><15>
004713   124   110   105    .ASCII 'THE OPERATOR MUST COMPARE THIS MAP WITH THE ONE LEFT BY THE'<12><15>
005010   111   116   123    .ASCII 'INSTALLER IN ORDER TO DETERMINE THAT PROPER DATA RETENTION EXISTS.'<12><12><15>
005115   040   040   040    .ASCII '                          PREREQUISITE'<12><12><15>
005157   040   040   040    .ASCII '      1. CZKMA OR VMSA MUST SUCCESSFULLY COMPLETE'<12><15>
005241   040   040   040    .ASCII '      2. OPERATOR MUST KNOW STARTING ADDRESSES OF ALL VOLATILE AND'<12><15>
005344   040   040   040    .ASCII '         NON-VOLATILE MEMORY IN THE SYSTEM'<12><15>
005417   040   040   040    .ASCII '      3. VIDEO TERMINALS MUST EITHER REMAIN POWERED WHEN PROCESSOR IS'<12><15>
005525   040   040   040    .ASCII '         POWERED DOWN OR A COPY OF THE RESTART HELP FILE MUST BE MADE'<12><15>
005633   040   040   040    .ASCII '         BEFORE POWERING DOWN THE SYSTEM'<12><15>
005704   040   040   040    .ASCII '      4. IF THE CPU DOES NOT HAVE AN ENABLE/HALT SWITCH, CHECK THAT THE'<12><15>
006014   040   040   040    .ASCII '         CPU IS STRAPPED TO HALT WHEN POWERED UP. OTHERWISE THE'<12><15>
006114   040   040   040    .ASCIZ '         DIAGNOSTIC MAY NOT OPERATE PROPERLY.'<12><12><15>
006174   012   015   104  HELP:    .ASCIZ  <12><15>'DO YOU WANT A HELP FILE (L) N? '
006236   012   015   101  ILLADD:  .ASCIZ  <12><15>'ADDRESS NOT AT 4K BOUNDARY!!!!!'<12><15>
006302   012   012   012  HELLO:   .ASCII  <12><12><12><15>'CVMEMAO DATA RETENTION DIAGNOSTIC FOR '
006354   116   117   116           .ASCIZ  'NON-VOLATILE MEMORIES'<12><15>
006404   012   015   111  BANKO:   .ASCIZ  <12><15>'IS THERE NON-VOLATILE MEMORY AT ADDRESS 0 (L) Y? '
006470   113   040   117  K:       .ASCIZ  'K OF MEMORY'<12><15>
006506   012   015   124  MAPPED:  .ASCIZ  <12><15>'THIS IS A MAPPED SYSTEM (MEMORY MANAGEMENT) WITH '
006572   012   015   124  UNMAP:   .ASCIZ  <12><15>'THIS IS AN UNMAPPED SYSTEM (NO MEMORY MANAGEMENT) WITH '
006664   012   015   127  WHERE:   .ASCII  <12><15>'WHAT IS THE STARTING ADDRESS OF ONE OF THE'
006740   040   116   117           .ASCIZ  ' NON-VOLATILE MEMORY'<12><15>'MODULES (O) 0? '
007006   012   015   127  WHERE1:  .ASCII  <12><15>'WHAT IS THE STARTING ADDRESS OF THE NON-VOLATILE '
007071   115   105   115           .ASCIZ  'MEMORY'<12><15>'UNDER TEST (O) 0? '
007124   012   015   104  WHOLE:   .ASCII  <12><15>'DO YOU WANT THE DIAGNOSTIC TO VALIDATE ALL MEMORY '
007210   111   116   040           .ASCIZ  'IN THIS SYSTEM (L) Y? '
007237   012   015   110  NOBLK:   .ASCIZ  <12><15>'HOW MANY 4K BANKS OF MEMORY ARE TO BE TESTED (D) 0? '
007326   012   012   012  RESHLP:  .ASCII  <12><12><12><12><12><15>
007334   040   040   040           .ASCII  '                    RESTART HELP FILE'<12><12><15>
007401   123   105   124           .ASCIZ  'SET ENABLE/FRONT PANEL SWITCH TO HALT POSITION, APPLY POWER'<12><15>
007477   074   103   122  CR1:     .ASCIZ  '<CR>'<12><15>
007506   124   131   120  PARO:    .ASCIZ  'TYPE:'<12><12><15>'@772344/XXXXXX '
007536   100   067   067  PAR7:    .ASCII  '@772356/XXXXXX 177600<CR>'<12><15>
007571   100   067   067           .ASCII  '@772304/XXXXXX  77406<CR>'<12><15>
```

```
 007624     100     067     067             .ASCII  '@772316/XXXXXX  77406<CR>'<12><15>
 007657     100     067     067             .ASCIZ  '@777572/XXXXXX      1<CR>'<12><15>
 007713     100     067     067     MMR3:   .ASCIZ  '@772516/XXXXXX     20<CR>'<12><15>
 007747     012     123     105     ENABLE: .ASCII  <12>'SET ENABLE/HALT FRONT PANEL SWITCH TO ENABLE '
 010025     120     117     123             .ASCII  'POSITION AND TYPE:'<12><12><15>
 010052     100     044     067             .ASCIZ  '@$7/XXXXXX '
 010066     074     103     122     RHELP:  .ASCII  '<CR>'<12><15>
 010074     100     120     012             .ASCII  '@P'<12><15>
 010100     040     040     040             .ASCIZ  '                         PLEASE STAND BY!'<12><15>
 010152     012     012     015     NOT4K:  .ASCIZ  <12><12><15>'ILLEGAL ADDRESS - NOT AT 4K BOUNDARY'<12><15>
 010224     012     012     015     OUTMEM: .ASCII  <12><12><15>'4K ADDRESS OR NUMBER OF 4K BANKS ARE OUT OF'
 010302     040     101     104             .ASCIZ  ' ADDRESS RANGE OF THIS SYSTEM'<12><15>
                                            .EVEN
 010342  000000                     ECHO:   .WORD 0
 010344  000000                     SAVLOW: .WORD 0
 010346  000000                     SAVHI:  .WORD 0

         006014                     DIF1 = ENDPR-Q4                 ;THE NUMBER TO BUMP THE POINTER TO BY PASS THE
                                                                    ;PROGRAM




                                    ;THIS ROUTINE BEGINS THE DIAGNOSTIC, IT WILL DETERMINE IF THIS MACHINE
                                    ;HAS MEMORY MANAGEMENT OR NOT, CALL THE SIZING ROUTINE, ASK THE MENU
                                    ;QUESTIONS, MOVE THE PROGRAM IF NEED BE, WRITE A BACKGROUND OF 125252 THROUGH
                                    ;DUT MEMORY, PRINT RESTART HELP FILE, IF APPLICABLE AND INSTRUCT
                                    ;OPERATOR TO POWER DOWN THE SYSTEM.

 010350                             ROUTINE START1
 010350                                                                         START1:

 010350                                     INLINE  <RESET>                 ;CLEAR THE WORLD
 010350  000005                                                                 RESET

 010352                                     LET     SP :- #SPINIT           ;FETCH STACK ADDRESS
 010352  012706  017644                                                         MOV     #SPINIT,SP

 010356                                     CALL    TRAPCT                  ;TRAP CATCHER
 010356  004767  005646                                                         JSR     PC,TRAPCT

 010362                                     LET     CONTROL :- #0           ;CLEAR OUT CONTROL WORD
 010362  005067  005572                                                         CLR     CONTROL

 010366                                     LET     PPOINT :  #0            ;PROGRAM POINTER.
 010366  005067  005576                                                         CLR     PPOINT

 010372  104401  006302                     TYPE    ,HELLO                  ;PRINT WHO WE ARE

 010376                                     CALL    $SIZE           ;THIS ROUTINE DETERMINES, MEMORY MANAGEMENT,
 010376  004767  170644                                                         JSR     PC,$SIZE
                                                            ; 16, 18, AND 22 BIT ADDRESS AND MEMORY SIZE

 010402                                     IF      $KT11 LT #0 THEN        ;SET MEMORY MANAGEMENT BIT
 010402  005767  170724                                                         TST     $KT11
 010406  002055                                                                 BGE     50002$

 010410                                             LET     CONTROL :- $MAP ;THIS SYSTEM HAS
```

```
     010410  016767  171010  005542                                                   MOV     $MAP,CONTROL

     010416                                      LET     CONTROL := CONTROL  SET.BY #1    ;INDICATE MEMORY
     010416  052767  000001  005534                                                   BIS     #1,CONTROL
                                                                                      ;MANAGEMENT

     010424                                      LET     ENDADD := @#KIPAR6 CLR.BY #177   ;ONLY WANT 4K BOUNDARY
     010424  013767  172354  005542                                                   MOV     @#KIPAR6,ENDADD
     010432  042767  000177  005534                                                   BIC     #177,ENDADD
                                                                      ;FETCH LAST PAR ADDRESS IN SYSTEM

     010440                                      LET R1 := #KIPAR0                  ;FETCH PAR ADDRESS
     010440  012701  172340                                                           MOV     #KIPAR0,R1

     010444                                      INCR R0 FROM #0 TO #6 BY #1        ;CLEAR OUT PARS
     010444  005000                                                                   CLR     R0
     010446  000401                                                                   BR      50003$
     010450                                                                           50004$:
     010450  005200                                                                   INC     R0
     010452                                                                           50003$:
     010452  020027  000006                                                           CMP     R0,#6
     010456  003002                                                                   BGT     50005$

     010460                                              LET (R1)+ := #0            ;4K
     010460  005021                                                                   CLR     (R1)+

     010462                                      ENDINC
     010462  000772                                                                   BR      50004$
     010464                                                                           50005$:

     010464                                      LET (R1) := #177600                ;SET UP I/O PAGE IN PAR7
     010464  012711  177600                                                           MOV     #177600,(R1)

     010470                                      LET R1 :- #KIPDR0                  ;FETCH PDR ADDRESS
     010470  012701  172300                                                           MOV     #KIPDR0,R1

     010474                                      INCR R0 FROM #0 TO #7 BY #1        ;SET UP FO 4K INTERVALS
     010474  005000                                                                   CLR     R0
     010476  000401                                                                   BR      50006$
     010500                                                                           50007$:
     010500  005200                                                                   INC     R0
     010502                                                                           50006$:
     010502  020027  000007                                                           CMP     R0,#7
     010506  003003                                                                   BGT     50010$

     010510                                              LET (R1)+ := #77406        ;4K
     010510  012721  077406                                                           MOV     #77406,(R1)+

     010514                                      ENDINC
     010514  000771                                                                   BR      50007$
     010516                                                                           50010$:

     010516                                      LET     @#SR0 := #1      ;SET MEMORY MANAGEMENT BIT
     010516  012737  000001  177572                                                   MOV     #1,@#SR0

     010524                                      IFB     $MAP LT #0 THEN            ;DO WE HAVE A 22 ADDRESS BIT
     010524  105767  170674                                                           TSTB    $MAP
```

```
 010530  002003                                                                 BGE     50011$
                                                                          ;MACHINE?

 010532                                               LET     @#SR3 := #20    ;SETUP FOR 22 BIT ADDRESSING
 010532 ·012737  000020  172516                                                 MOV     #20,@#SR3

 010540                                     ENDIF
 010540                                                                         50011$:

 010540                           ELSE
 010540  000406                                                                 BR      50012$
 010542                                                                         50002$:

 010542                                     LET     ENDADD := $LSTAD + #2    ;FETCH LAST VIRTUAL ADDRESS
 010542  016767  171170  005424                                                 MOV     $LSTAD,ENDADD
 010550  062767  000002  005416                                                 ADD     #2,ENDADD

 010556                           ENDIF
 010556                                                                         50012$:

 010556                           IF #1 SETIN CONTROL THEN          ;TELL OPERATOR WHAT KIND OF SYSTEM
 010556  032767  000001  005374                                                 BIT     #1,CONTROL
 010564  001403                                                                 BEQ     50013$

 010566  104401  006506                     TYPE    ,MAPPED          ;TYPE THIS IS A MAPPED SYSTEM
                                                                     ;(MEMORY MANAGEMENT) WITH

 010572                           ELSE
 010572  000402                                                                 BR      50014$
 010574                                                                         50013$:

 010574  104401  006572                     TYPE    ,UNMAP           ;TYPE THIS IS AN UN-MAPPED SYSTEM
                                                                     ;(NO MEMORY MANAGEMENT) WITH

 010600                           ENDIF
 010600                                                                         50014$:

 010600                           LET     $LSTBK := $LSTBK + #40   ;FETCH LAST BANK ADDRESS WITH OUT MEMORY
 010600  062767  000040  171132                                                 ADD     #40,$LSTBK

 010606                           LET     $LSTBK := $LSTBK SHIFT -1         ;CORRECT LSTBK FOR PRINTING
 010606  006267  171126                                                         ASR     $LSTBK

 010612                           LET     $LSTBK := $LSTBK CLR.BY #100000 ;CLEAR SIGN BIT
 010612  042767  100000  171120                                                 BIC     #100000,$LSTBK

 010620                           LET     $LSTBK := $LSTBK SHIFT -4         ;CONTINUE CORRECTING
 010620  006267  171114                                                         ASR     $LSTBK
 010624  006267  171110                                                         ASR     $LSTBK
 010630  006267  171104                                                         ASR     $LSTBK
 010634  006267  171100                                                         ASR     $LSTBK

 010640                           TYPDEC  $LSTBK                     ;TELL OPERATOR HOW MUCH MEMORY WE FOUND
 010640  016746  171074           MOV     $LSTBK,-(SP)    ;;SAVE $LSTBK FOR TYPEOUT
 010644  104405                   TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN

 010646  104401  006470           TYPE    ,K                         ;PRINT K OF MEMORY.
```

```

010652  104401  006174                  TYPE    .HELP                   ;PRINT DO YOUR WANT A HELP FILE

010656                                  CALL    @#FETCH                 ;FETCH ANSWER
010656  004737  001742                                                          JSR     PC,@#FETCH

010662                                  IF @#ECHO EQ #'Y THEN           ;PRINT HELP FILE IF YES
010662  023727  010342  000131                                                  CMP     @#ECHO,#'Y
010670  001002                                                                  BNE     50015$

010672  104401  002226                          TYPE    .HELPF                  ;PRINT HELP FILE

010676                                  ENDIF
010676                                                                          50015$:

010676  104401  006404                  TYPE    .BANKO          ;PRINT IS THERE NON-VOLATILE MEMORY AT ADDRESS 0

010702                                  CALL    @#FETCH                         ;FETCH ANSWER
010702  004737  001742                                                          JSR     PC,@#FETCH

010706                                  IF @#ECHO EQ #15 THEN                   ;LOOK FOR DEFAULT CHARACTER
010706  023727  010342  000015                                                  CMP     @#ECHO,#15
010714  001003                                                                  BNE     50016$

010716                                          LET @#ECHO := #'Y               ;CHANGE DEFAULT CHARACTER
010716  012737  000131  010342                                                  MOV     #'Y,@#ECHO

010724                                  ENDIF
010724                                                                          50016$:

010724                                  IF @#ECHO NE #'Y THEN                   ;IF NOT AT BANKO THEN WHERE
010724  023727  010342  000131                                                  CMP     @#ECHO,#'Y
010732  001440                                                                  BEQ     50017$

010734                                          LET R4 := #0                    ;CORRECT ADDRESS FLAG
010734  005004                                                                  CLR     R4

010736                                          LOOP
010736                                                                          50020$:

010736  104401  006664                                  TYPE    .WHERE  ;PRINT WHAT IS THE STARTING ADDRESS
                                                                        ;OF ONE OF THE NON-VOLATILE MEMORY
                                                                        ;MODULES.

010742                                                  CALL    @#FETADD        ;GO FETCH ADDRESS TO
010742  004737  001762                                                          JSR     PC,@#FETADD
                                                                                ;RELOCATE PROGRAM


010746                                          EXIF    R4 NE #0
010746  005704                                                                  TST     R4
010750  001001                                                                  BNE     50021$

010752                                          ENDLOOP
010752  000771                                                                  BR      50020$
010754                                                                          50021$:

010754                                  IF R3 NE #0 THEN        ;DO NOT HAVE TO RELOCALE IF ADDRESS IS ZERO
```

```
        010754  005703                                                                          TST     R3
        010756  001426                                                                          BEQ     50022$

        010760                                          IF #1 SETIN CONTROL THEN        ;WE HAVE MEMORY MANAGEMENT?
        010760  032767  000001  005172                                                          BIT     #1,CONTROL
        010766  001406                                                                          BEQ     50023$

        010770                                                  LET     @#KIPAR2 := R3  ;LOAD PAR2 WITH POINTER TO THE
        010770  010337  172344                                                                  MOV     R3,@#KIPAR2
                                                                                        ;4K CHUNK OF NON-VOLATILE MEMORY

        010774                                                  LET     R0 := #40400    ;POINT TO PROGRAM ADDRESS REG 2
        010774  012700  040400                                                                  MOV     #40400,R0
                                                                                        ;WE START AT ADDRESS 400 OF
                                                                                        ;4K BLOCK

        011000                                                  PUSH    R0
        011000  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK

        011002                                          ELSE    ;NO MEMORY MANAGEMENT
        011002  000404                                                                          BR      50024$
        011004                                                                                  50023$:

        011004                                                  LET     R0 := R3 + #400 ;WE START AT ADDRESS 400
        011004  010300                                                                          MOV     R3,R0
        011006  062700  000400                                                                  ADD     #400,R0
                                                                                        ;OF 4K BLOCK

        011012                                                  PUSH    R0
        011012  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK

        011014                                          ENDIF
        011014                                                                                  50024$:

        011014                                          LET     R1 := #Q1       ;FETCH STARTING ADDRESS TO BE MOVED
        011014  012701  011034                                                                  MOV     #Q1,R1

        011020                                          LOOP
        011020                                                                                  50025$:

        011020                                          EXIF    R1 EQ #ENDPR    ;LEAVE WHEN ENTIRE PROGRAM
        011020  020127  017646                                                                  CMP     R1,#ENDPR
        011024  001402                                                                          BEQ     50026$
                                                                                ;HAS BEEN MOVED.

        011026                                                  LET (R0)+ := (R1)+              ;MOVE PROGRAM
        011026  012120                                                                          MOV     (R1)+,(R0)+

        011030                                          ENDLOOP
        011030  000773                                                                          BR      50025$
        011032                                                                                  50026$:


        011032  000136                                  JMP     @(SP)+          ;GO TO WHERE PROGRAM HAS MOVED TO

        011034                                  ENDIF
        011034                                                                                  50022$:
```

```

011034                                  ENDIF
011034                                                                                 50017$:

011034                          Q1:     LET SP := PC + #SPINIT-.        ;FETCH NEW STACK
011034  010706                                                                 MOV     PC,SP
011036  062706  006606                                                         ADD     #SPINIT-.,SP

011042                                  CALL    TRAPCT                  ;RESET UP TRAP LOCATIONS
011042  004767  005162                                                         JSR     PC,TRAPCT

011046                                  LET R1 := @#34 + #$TRPAD-$TRAP  ;POINT TO NEW LOCATION OF TRAP TABLE
011046  013701  000034                                                         MOV     @#34,R1
011052  062701  000046                                                         ADD     #$TRPAD-$TRAP,R1

011056                                  LET R0 :- #$TRPAD               ;FETCH OLD TABLE ADDRESS TO USE IN
011056  012700  017474                                                         MOV     #$TRPAD,R0
                                                                        ;CALCULATING TABLE'S OFFSETS

011062                                  LET R3 := R0 - #$TRAP2          ;CALCULATE OFFSET BETWEEN TABLE AND
011062  010003                                                                 MOV     R0,R3
011064  162703  017462                                                         SUB     #$TRAP2,R3
                                                                        ;$TRAP2 ROUTINE

011070                                  INLINE <TST (R0)+>              ;BUMP OLD TABLE ADDRESS FOR NEXT CAL.
011070  005720                                                                 TST (R0)+

011072                                  LET R2 := R1 - R3               ;R2 NOW CONTAINS NEW ADDRESS
011072  010102                                                                 MOV     R1,R2
011074  160302                                                                 SUB     R3,R2

011076                                  LET (R1)+ := R2                 ;LOAD TABLE
011076  010221                                                                 MOV     R2,(R1)+

011100                                  LET R3 := R0 - #$TYPE           ;CALCULATE OFFSET BETWEEN TABLE AND
011100  010003                                                                 MOV     R0,R3
011102  162703  016370                                                         SUB     #$TYPE,R3
                                                                        ;$TYPE ROUTINE

011106                                  INLINE <TST (R0)+>              ;BUMP OLD TABLE ADDRESS FOR NEXT CAL.
011106  005720                                                                 TST (R0)+

011110                                  LET R2 := R1 - R3               ;R2 NOW CONTAINS NEW ADDRESS
011110  010102                                                                 MOV     R1,R2
011112  160302                                                                 SUB     R3,R2

011114                                  LET (R1)+ := R2                 ;LOAD TABLE
011114  010221                                                                 MOV     R2,(R1)+

011116                                  LET R3 := R0 - #$TYPOC          ;CALCULATE OFFSET BETWEEN TABLE AND
011116  010003                                                                 MOV     R0,R3
011120  162703  017212                                                         SUB     #$TYPOC,R3
                                                                        ;$TYPOC ROUTINE

011124                                  INLINE <TST (R0)+>              ;BUMP OLD TABLE ADDRESS FOR NEXT CAL.
011124  005720                                                                 TST (R0)+

011126                            LET R2 := R1 - R3            ;R2 NOW CONTAINS NEW ADDRESS
011126  010102                                                          MOV     R1,R2
011130  160302                                                          SUB     R3,R2

011132                            LET (R1)+ := R2              ;LOAD TABLE
011132  010221                                                          MOV     R2,(R1)+

011134                            LET R3 := R0 - #$TYPOS       ;CALCULATE OFFSET BETWEEN TABLE AND
011134  010003                                                          MOV     R0,R3
011136  162703  017166                                                  SUB     #$TYPOS,R3
                                                               ;$TYPOS ROUTINE

011142                            INLINE <TST (R0)+>           ;BUMP OLD TABLE ADDRESS FOR NEXT CAL.
011142  005720                                                          TST (R0)+

011144                            LET R2 := R1 - R3            ;R2 NOW CONTAINS NEW ADDRESS
011144  010102                                                          MOV     R1,R2
011146  160302                                                          SUB     R3,R2

011150                            LET (R1)+ := R2              ;LOAD TABLE
011150  010221                                                          MOV     R2,(R1)+

011152                            LET R3 := R0 - #$TYPON       ;CALCULATE OFFSET BETWEEN TABLE AND
011152  010003                                                          MOV     R0,R3
011154  162703  017226                                                  SUB     #$TYPON,R3
                                                               ;$TYPON ROUTINE

011160                            INLINE <TST (R0)+>           ;BUMP OLD TABLE ADDRESS FOR NEXT CAL.
011160  005720                                                          TST (R0)+

011162                            LET R2 := R1 - R3            ;R2 NOW CONTAINS NEW ADDRESS
011162  010102                                                          MOV     R1,R2
011164  160302                                                          SUB     R3,R2

011166                            LET (R1)+ := R2              ;LOAD TABLE
011166  010221                                                          MOV     R2,(R1)+

011170                            LET R3 := R0 - #$TYPDS       ;CALCULATE OFFSET BETWEEN TABLE AND
011170  010003                                                          MOV     R0,R3
011172  162703  016714                                                  SUB     #$TYPDS,R3
                                                               ;$TYPDS ROUTINE

011176                            INLINE <TST (R0)+>           ;BUMP OLD TABLE ADDRESS FOR NEXT CAL.
011176  005720                                                          TST (R0)+

011200                            LET R2 := R1 - R3            ;R2 NOW CONTAINS NEW ADDRESS
011200  010102                                                          MOV     R1,R2
011202  160302                                                          SUB     R3,R2

011204                            LET (R1)+ := R2              ;LOAD TABLE
011204  010221                                                          MOV     R2,(R1)+

011206  104401  007124            TYPE      ,WHOLE        ;PRINT DO YOU WANT THE DIAGNOSTIC TO VALIDATE
                                                          ;ALL MEMORY IN THIS SYSTEM.

011212                            CALL      @#FETCH            ;FETCH CHARACTER.
011212  004737  001742                                                  JSR     PC,@#FETCH
```

```

011216                                 IF @#ECHO EQ #15 THEN          ;LOOK FOR DEFAULT CHARACTER
011216  023727  010342  000015                                                            CMP     @#ECHO,#15
011224  001003                                                                            BNE     50027$

011226                                         LET     @#ECHO := #'Y          ;CHANGE DEFAULT CHARACTER
011226  012737  000131  010342                                                            MOV     #'Y,@#ECHO

011234                                 ENDIF
011234                                                                                    50027$:

011234                                 IF @#ECHO NE #'Y THEN          ;IF NOT ENTIRE SYSTEM FIND OUT WHERE
011234  023727  010342  000131                                                            CMP     @#ECHO,#'Y
011242  001461                                                                            BEQ     50030$

011244                                         LET     R4 := #0               ;ILLEGAL ADDRESS FLAG
011244  005004                                                                            CLR     R4

011246                                         LOOP
011246                                                                                    50031$:

011246  104401  007006                                 TYPE    ,WHERE1 ;PRINT: WHAT IS THE ADDRESS OF THE
                                                                       ;NON-VOLATILE MEMORY MEMORY UNDER TEST

011252                                                 CALL    @#FETADD              ;GO FETCH ADDRESS TO BE TESTED
011252  004737  001762                                                                    JSR     PC,@#FETADD

011256                                         EXIF    R4 NE #0
011256  005704                                                                            TST     R4
011260  001001                                                                            BNE     50032$

011262                                         ENDLOOP
011262  000771                                                                            BR      50031$
011264                                                                                    50032$:

011264                                         LET     R4 := #0               ;ILLEGAL ADDRESS FLAG
011264  005004                                                                            CLR     R4

011266                                         LET     PPOINT := R3           ;NEW PROGRAM POINTER
011266  010367  004676                                                                    MOV     R3,PPOINT

011272                                         LOOP
011272                                                                                    50033$:

011272  104401  007237                                 TYPE    ,NOBLK          ;PRINT WHAT IS THE NUMBER OF 4K
                                                                               ;BANKS TO BE TESTED

011276  104411                                         RDDEC                   ;FETCH DECIMAL NUMBER

011300                                                 POP     R0
011300  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0

011302                                                 LET     R3 := #0        ;CLEAR ADD REG.
011302  005003                                                                            CLR     R3

011304                                                 LET     R2 := #0        ;CLEAR ADD REG.
011304  005002                                                                            CLR     R2
```

```
011306                                  IF R0 NE #0 THEN        ;NO NEED TO CALCULATE END ADD.
011306  005700                                                          TST     R0
011310  001426                                                          BEQ     50035$
011312                                          INCR R1 FROM #1 TO R0 BY #1     ;MULTIPLY THE
011312  012701  000001                                                  MOV     #1,R1
011316  000401                                                          BR      50036$
011320                                                                  50037$:
011320  005201                                                          INC     R1
011322                                                                  50036$:
011322  020100                                                          CMP     R1,R0
011324  003004                                                          BGT     50040$
                                        ;NUMBER OF BANKS TO BE TESTED BY 4K
011326                                          LET R2 := R2 + #20000   ;ADD 4K
011326  062702  020000                                                  ADD     #20000,R2
011332                                          LET R3 := R3 + CARRY    ;ADD OVERFLOW TO
011332  005503                                                          ADC     R3
                                                                                ;UPPER WORD
011334                                  ENDINC
011334  000771                                                          BR      50037$
011336                                                                  50040$:
011336                                  LET R2 := R2 + @#SAVLOW ;ADD LOWER START ADDRESS
011336  063702  010344                                                  ADD     @#SAVLOW,R2
011342                                  LET R3 :- R3 + CARRY    ;CARRY OVER ANY OVERFLOW
011342  005503                                                          ADC     R3
011344                                  LET R3 := R3 + @#SAVHI  ;ADD UPPER START ADDRESS BITS
011344  063703  010346                                                  ADD     @#SAVHI,R3
011350                                  LET @#ECHO := R2        ;STORE AWAY LOWER
011350  010237  010342                                                  MOV     R2,@#ECHO
                                                                        ;ADDRESS
011354                                  LET @#$HIOCT := R3      ;STORE AWAY HIGHER
011354  010337  001066                                                  MOV     R3,@#$HIOCT
                                                                        ;ADDRESS
011360                                  CALL    @#ADDCHK
011360  004737  002074                                                  JSR     PC,@#ADDCHK
011364                          ELSE
011364  000401                                                          BR      50041$
011366                                                                  50035$:
011366                                  LET R4 :  R4 + #1       ;INDICATE OK
011366  005204                                                          INC     R4
011370                          ENDIF
011370                                                                  50041$:
011370                  EXIF    R4 NE #0
```

```
011370  005704                                                                          TST     R4
011372  001001                                                                          BNE     50034$

011374                                                  ENDLOOP
011374  000736                                                                          BR      50033$
011376                                                                                  50034$:

011376                                                  IF R0 NE #0 THEN                ;UPDATE END ADDRESS IF NECESSARY
011376  005700                                                                          TST     R0
011400  001402                                                                          BEQ     50042$

011402                                                          LET ENDADD := R3        ;STORE AWAY LAST ADDRESS
011402  010367  004566                                                                  MOV     R3,ENDADD

011406                                                  ENDIF
011406                                                                                  50042$:

011406                                          ENDIF
011406                                                                                  50030$:

011406  104401  007326                          TYPE    ,RESHLP                         ;PRINT RESTART HELP FILE HEADER

011412                                          IF #1 SETIN CONTROL THEN                ;ARE WE A MAPPED SYSTEM
011412  032767  000001  004540                                                          BIT     #1,CONTROL
011420  001424                                                                          BEQ     50043$
                                                                                        ;YES
011422  104401  007506                                  TYPE    ,PAR0                   ;PRINT @772340/

011426                                                  TYPOCS  @#KIPAR2        ;MEMORY MANAGEMENT LOCATION OF PROGRAM
011426  013746  172344                          MOV     @#KIPAR2,-(SP)  ;;SAVE @#KIPAR2 FOR TYPEOUT
011432  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
011434     006                                  .BYTE   6               ;;TYPE 6 DIGITS
011435     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS

011436  104401  007477                                  TYPE    ,CR1                    ;PRINT '<CR>'<12><15>

011442  104401  007536                                  TYPE    ,PAR7                   ;PRINT @772356/777600<CR>
                                                                                        ;       @772300/77406<CR>
                                                                                        ;       @772316/77406<CR>
                                                                                        ;       @777572/1<CR>

011446                                                  IF #200 SETIN CONTROL THEN      ;IS THIS A 22 BIT ADDRESSING
011446  032767  000200  004504                                                          BIT     #200,CONTROL
011454  001406                                                                          BEQ     50044$
                                                                                        ;MACHINE

011456                                                          IF ENDADD GT #7600 THEN ;DOES THIS MACHINE CONTAIN MORE
011456  026727  004512  007600                                                          CMP     ENDADD,#7600
011464  003402                                                                          BLE     50045$
                                                                                        ;THAN 124K?

011466  104401  007713                                                  TYPE    ,MMR3   ;YES PRINT @772516/20<CR>

011472                                                          ENDIF
011472                                                                                  50045$:

011472                                                  ENDIF
```

```
     011472                                                                     50044$:
1979
1980 011472                                    ENDIF
     011472                                                                     50043$:
1981
1982 011472  104401  007747                    TYPE    .ENABLE        ;PRINT SET SWITCH TO ENABLE AND TYPE
1983                                                                           ;@777707/
1984
1985                                           .ENABL  LSB
1986
1987 011476                                    LET     R1 := PC + #PWRUP-.            ;FETCH RESTART ADDRESS
     011476  010701                                                             MOV     PC,R1
     011500  062701  000372                                                     ADD     #PWRUP-.,R1
1988
1989 011504                                    IF #1 SETIN CONTROL THEN               ;ARE WE A MAPPED SYSTEM
     011504  032767  000001  004446                                             BIT     #1,CONTROL
     011512  001404                                                             BEQ     50046$
1990                                                                           ;YES
1991
1992 011514                                            LET     R1 := R1 CLR.BY #160000 ;CLEAR OUT PROGRAM ADDRESS
     011514  042701  160000                                                     BIC     #160000,R1
1993                                                                           ;REGISTER PRINTERS.
1994
1995 011520                                            LET     R1 := R1 SET.BY #40000  ;LOOK AT PAR2
     011520  052701  040000                                                     BIS     #40000,R1
1996
1997 011524                                    ENDIF
     011524                                                                     50046$:
1998
1999 011524                                    TYPOCS  R1                             ;PRINT RESTART ADDRESS
     011524  010146                            MOV     R1,-(SP)       ;;SAVE R1 FOR TYPEOUT
     011526  104403                            TYPOS                  ;;GO TYPE--OCTAL ASCII
     011530     006                            .BYTE   6              ;;TYPE 6 DIGITS
     011531     000                            .BYTE   0              ;;SUPPRESS LEADING ZEROS
2000
2001 011532  104401  010066                    TYPE    .RHELP                  ;PRINT REST OF HELP FILE
2002
2003                                           .DSABL  LSB
2004
2005                                           ;WRITE BACKGROUND THROUGH MEMORY
2006
2007 011536                                    IF #1 SETIN CONTROL THEN               ;ARE WE A MAPPED SYSTEM
     011536  032767  000001  004414                                             BIT     #1,CONTROL
     011544  001414                                                             BEQ     50047$
2008
2009 011546                                            LET @#KIPAR1 := PPOINT         ;FETCH TEST STARTING POINT
     011546  016737  004416  172342                                             MOV     PPOINT,@#KIPAR1
2010
2011 011554                                            IF PPOINT EQ #0 THEN           ;ARE WE IN BANK 0
     011554  005767  004410                                                     TST     PPOINT
     011560  001003                                                             BNE     50050$
2012
2013 011562                                                    LET     R0 := #20400   ;POINT TO ADDRESS 400, PROGRAM
     011562  012700  020400                                                     MOV     #20400,R0
2014                                                                           ;ADDRESS REGISTER 1
2015
2016 011566                                            ELSE
```

```
 011566  000402                                                                 BR      50051$
 011570                                                                         50050$:

 011570                                          LET     R0 := #20000     ;POINT TO ADDRESS 0, PROGRAM
 011570  012700  020000                                                         MOV     #20000,R0
                                                                          ;ADDRESS REGISTER 1

 011574                                  ENDIF
 011574                                                                         50051$:

 011574                          ELSE    ;UNMAPPED SYSTEM
 011574  000410                                                                 BR      50052$
 011576                                                                         50047$:

 011576                                  IF PPOINT EQ #0 THEN             ;ARE WE IN BANK 0
 011576  005767  004366                                                         TST     PPOINT
 011602  001003                                                                 BNE     50053$

 011604                                          LET     R0 := #400       ;START AFTER TRAP LOCATIONS
 011604  012700  000400                                                         MOV     #400,R0

 011610                                  ELSE
 011610  000402                                                                 BR      50054$
 011612                                                                         50053$:

 011612                                          LET     R0 :- PPOINT     ;START AT TEST BANK
 011612  016700  004352                                                         MOV     PPOINT,R0

 011616                                  ENDIF
 011616                                                                         50054$:

 011616                          ENDIF
 011616                                                                         50052$:

 011616                          LET     R4 :- #0                         ;CLEAR EXIT FROM LOOP FLAG
 011616  005004                                                                 CLR     R4

                                 .ENABL  LSB

 011620                          LET     R1 := PC + #ROMMD-.              ;FETCH ROMMD ROUTINE
 011620  010701                                                                 MOV     PC,R1
 011622  062701  004360                                                         ADD     #ROMMD-.,R1

 011626                          LET     @#4 := R1                        ;LOAD ADDRESS INTO 4 INCASE WE
 011626  010137  000004                                                         MOV     R1,@#4
                                                                          ;HIT A ROM MODULE

 011632                  Q4:     LOOP
 011632                                                                         50055$:

 011632                                  LET     (R0)+ :- #125252         ;WRITE DATA PATTERN
 011632  012720  125252                                                         MOV     #125252,(R0)+
                                                                          ;THROUGH OUT MEMORY

 011636                                  IF #1 SETIN CONTROL THEN         ;ARE WE A MAPPED SYSTEM
 011636  032767  000001  004314                                                 BIT     #1,CONTROL
 011644  001436                                                                 BEQ     50057$
```

```
     011646                                          IF @#KIPAR1 EQ @#KIPAR2 THEN      ;WE IN SAME BLOCK OF
     011646  023737  172342  172344                                                    CMP     @#KIPAR1,@#KIPAR2
     011654  001014                                                                    BNE     50060$
                                                                                       ;MEMORY

     011656                                               LET R1 := PC - #.-Q4         ;DISTANCE BETWEEN
     011656  010701                                                                    MOV     PC,R1
     011660  162701  000026                                                            SUB     #.-Q4,R1
                                                                          ;BEGINNING OF PROGRAM AND HERE

     011664                                               LET R1 :- R1 CLR.BY #160000       ;CLEAR OUT PAR
     011664  042701  160000                                                            BIC     #160000,R1
                                                                                               ;POINTER

     011670                                               LET R3 := R0 CLR.BY #160000       ;CLEAR OUT PAR
     011670  010003                                                                    MOV     R0,R3
     011672  042703  160000                                                            BIC     #160000,R3
                                                                                               ;POINTER

     011676                                               IF R3 EQ R1 THEN             ;HAVE WE REACHED
     011676  020301                                                                    CMP     R3,R1
     011700  001002                                                                    BNE     50061$
                                                                          ;BEGINNING OF PROGRAM

     011702                                                    LET R0 :- R0 + #DIF1    ;HAS BUMP
     011702  062700  006014                                                            ADD     #DIF1,R0
                                                                          ;POINTER TO END OF PROGRAM

     011706                                               ENDIF
     011706                                                                            50061$:
     011706                                          ENDIF
     011706                                                                            50060$:

     011706                                          IF R0 EQ #40000 THEN      ;HAVE WE COMPLETED 4K CHUNK
     011706  020027  040000                                                            CMP     R0,#40000
     011712  001005                                                                    BNE     50062$
                                                                          ;OF MEMORY

     011714                                               LET R0 := #20000             ;YES RETURN PAR1 POINTER
     011714  012700  020000                                                            MOV     #20000,R0

                                                          ;BUMP TO NEXT 4K
     011720                                               LET     @#KIPAR1 := @#KIPAR1 + #200
     011720  062737  000200  172342                                                    ADD     #200,@#KIPAR1

     011726                                          ENDIF
     011726                                                                            50062$:

     011726                                          IF @#KIPAR1 EQ ENDADD THEN        ;HAVE WE REACHED END OF
     011726  023767  172342  004240                                                    CMP     @#KIPAR1,ENDADD
     011734  001001                                                                    BNE     50063$
                                                                                       ;MEMORY

     011736                                               LET R4 := R4 + #1            ;YES SET FINISHED FLAG
     011736  005204                                                                    INC     R4
```

```
011740                                                  ENDIF
011740                                                                                  50063$:

011740                                          ELSE    ;UNMAPPED SYSTEM
011740  000413                                                                          BR      50064$
011742                                                                                  50057$:

011742                                                  LET R1 := PC - #.-Q4    ;FETCH BEGINNING ADDRESS OF
011742  010701                                                                          MOV     PC,R1
011744  162701  000112                                                                  SUB     #.-Q4,R1
                                                                                ;THIS PROGRAM

011750                                                  IF R0 EQ R1 THEN        ;HAVE WE REACHED BEGINNING OF
011750  020001                                                                          CMP     R0,R1
011752  001002                                                                          BNE     50065$
                                                                                ;THIS PROGRAM

011754                                                          LET R0 := R0 + #DIF1    ;YES BUMP POINTER TO
011754  062700  006014                                                                  ADD     #DIF1,R0
                                                                                        ;END OF PROGRAM

011760                                                  ENDIF
011760                                                                                  50065$:

011760                                                  IF R0 HIS ENDADD THEN   ;HAVE WE REACHED END OF MEMORY
011760  020067  004210                                                                  CMP     R0,ENDADD
011764  103401                                                                          BLO     50066$

011766                                                          LET     R4 := R4 + #1   ;YES SET FINISHED FLAG
011766  005204                                                                          INC     R4

011770                                                  ENDIF
011770                                                                                  50066$:

011770                                          ENDIF
011770                                                                                  50064$:

011770                                  EXIF    R4 NE #0                        ;LEAVE IF FINISHED FLAG IS SET
011770  005704                                                                          TST     R4
011772  001001                                                                          BNE     50056$

011774                                  ENDLOOP
011774  000716                                                                          BR      50055$
011776                                                                                  50056$:

                                        ;FETCH LOCATION OF POWER DOWN MESSAGE
011776                                  LET R1 := PC + #PWRDWN-.
011776  010701                                                                          MOV     PC,R1
012000  062701  003776                                                                  ADD     #PWRDWN-.,R1

                                        ;LOAD ADDRESS INTO TRAP INSTRUCTION
012004                                  LET 1$ := R1
012004  010167  000054                                                                  MOV     R1,1$

                                        ;CALCULATE CHECKSUM OF TRAP PAGE

012010                                  LET R1 :- #0                            ;ADDRESS OF TRAP LOCATION
```

```
012010  005001                                                                                  CLR     R1

012012                                  LET R3 := #0                            ;CHECKSUM WORK REGISTER
012012  005003                                                                                  CLR     R3

012014                                  LOOP
012014                                                                                          50067$:

012014                                          LET R3 := R3 + (R1)+            ;CALCULATE CHECKSUM
012014  062103                                                                                  ADD     (R1)+,R3

012016                                  EXIF R1 EQ #400                 ;EXIF WHEN REACHED END OF TRAP PAGE
012016  020127  000400                                                                          CMP     R1,#400
012022  001401                                                                                  BEQ     50070$

012024                                  ENDLOOP
012024  000773                                                                                  BR      50067$
012026                                                                                          50070$:

012026                                  LET CSTRPG := R3                        ;SAVE CHECKSUM
012026  010367  005466                                                                          MOV     R3,CSTRPG

                                        ;CALCULATE CHECKSUM OF THIS PROGRAM
012032                                  LET     R1 := PC - #.-04                ;R1 CONTAINS BEGINNING ADDRESS
012032  010701                                                                                  MOV     PC,R1
012034  162701  000202                                                                          SUB     #.-04,R1
                                                                                ;OF PROGRAM

012040                                  LET     R2 := PC + #CHKSUM-.            ;R2 CONTAINS END ADDRESS OF PROGRAM
012040  010702                                                                                  MOV     PC,R2
012042  062702  005460                                                                          ADD     #CHKSUM-.,R2

012046                                  LET     R3 := #0                        ;R3 CONTAINS CHECK SUM
012046  005003                                                                                  CLR     R3

012050                                  LOOP
012050                                                                                          50071$:

012050                                          LET     R3 := R3 + (R1)+        ;ADD CONTENTS OF PROGRAM
012050  062103                                                                                  ADD     (R1)+,R3

012052                                  EXIF    R1 EQ R2                ;LEAVE WHEN REACHED END OF PROGRAM
012052  020102                                                                                  CMP     R1,R2
012054  001401                                                                                  BEQ     50072$

012056                                  ENDLOOP
012056  000774                                                                                  BR      50071$
012060                                                                                          50072$:

012060                                  LET (R2) := R3                  ;LOAD CHECK SUM INTO CHECKSUM AREA
012060  010312                                                                                  MOV     R3,(R2)

                                        ;TELL OPERATOR TO POWER DOWN THE SYSTEM
012062  104401                          TYPE
012064  015776                  1$:     .WORD   PWRDWN

012066                                  INLINE  < BR . >                                        ;
```

```
        012066  000777                                                  BR .
 2161
 2162                                   .DSABL  LSB
 2163
 2164   012070                  ENDRTN
        012070                                                          50000$:
        012070                                                          50001$:
        012070  000207                                                  RTS     PC
```

```
                                                 .SBTTL  POWER UP ROUTINE

                                         ;THIS ROUTINE CHECKS THE ENTIRE SYSTEM FOR DATA RETENTION AND OUTPUTS A MEMORY
                                         ;MAP OF THE SYSTEM.

012072                                   ROUTINE PWRUP
012072                                                                                 PWRUP:

012072                                           LET SP := PC + #SPINIT-.            ;FETCH STACK BUFFER
012072  010706                                                                         MOV     PC,SP
012074  062706  005550                                                                 ADD     #SPINIT-.,SP

012100                                           IF #1 SETIN CONTROL THEN            ;IS THIS A MAPPED SYSTEM
012100  032767  000001  004052                                                         BIT     #1,CONTROL
012106  001462                                                                         BEQ     50002$
                                                                                     ;YES

012110                                                   LET @#KIPAR0 := #0          ;CLEAR OUT PAR0
012110  005037  172340                                                                 CLR     @#KIPAR0

012114                                                   LET @#KIPAR1 := #0          ;CLEAR OUT PAR1
012114  005037  172342                                                                 CLR     @#KIPAR1

012120                                                   LET R1 := #KIPAR3           ;FETCH MEMORY MANAGEMENT PAR
012120  012701  172346                                                                 MOV     #KIPAR3,R1
                                                                                     ;LOCATION (DO NOT TOUCH PAR2 FOR
                                                                                     ;IT CONTROLS OUR PC

012124                                                   INCR R0 FROM #1 TO #4 BY #1 ;CLEAR THOSES REGISTERS
012124  012700  000001                                                                 MOV     #1,R0
012130  000401                                                                         BR      50003$
012132                                                                                 50004$:
012132  005200                                                                         INC     R0
012134                                                                                 50003$:
012134  020027  000004                                                                 CMP     R0,#4
012140  003002                                                                         BGT     50005$

012142                                                           LET (R1)+ := #0     ;
012142  005021                                                                         CLR     (R1)+

012144                                                   ENDINC
012144  000772                                                                         BR      50004$
012146                                                                                 50005$:

012146                                                   LET (R1) := #177600         ;SET UP I/O PAGE
012146  012711  177600                                                                 MOV     #177600,(R1)

012152                                                   LET R1 := #KIPDR0           ;FETCH PDR LOCATION
012152  012701  172300                                                                 MOV     #KIPDR0,R1

012156                                                   INCR R0 FROM #0 TO #7 BY #1 ;SET THEM UP TO DO 4K INTERVALS
012156  005000                                                                         CLR     R0
012160  000401                                                                         BR      50006$
012162                                                                                 50007$:
012162  005200                                                                         INC     R0
012164                                                                                 50006$:
012164  020027  000007                                                                 CMP     R0,#7
```

```
     012170  003003                                                                                   BGT     50010$
2197
2198 012172                                                         LET (R1)+ := #77406     ;
     012172  012721  077406                                                                           MOV     #77406,(R1)+
2199
2200 012176                                                 ENDINC
     012176  000771                                                                                   BR      50007$
     012200                                                                                           50010$:
2201
2202 012200                                                 LET @#SR0 := #1                   ;SET MEMORY MANAGEMENT IF NOT
     012200  012737  000001  177572                                                                   MOV     #1,@#SR0
2203                                                                                          ;ALREADY DONE SO
2204
2205 012206                                                 IF #200 SETIN CONTROL THEN        ;IS THIS A 22 ADDRESS BIT SYSTEM
     012206  032767  000200  003744                                                                   BIT     #200,CONTROL
     012214  001403                                                                                   BEQ     50011$
2206
2207 012216                                                         LET @#SR3 := #20          ;YES, SET UP 22 BIT ADDRESSING
     012216  012737  000020  172516                                                                   MOV     #20,@#SR3
2208
2209 012224                                                 ENDIF
     012224                                                                                           50011$:
2210
2211 012224                                                 IF PPOINT EQ #0 THEN              ;ARE WE CHECKING ENTIRE SYSTEM?
     012224  005767  003740                                                                           TST     PPOINT
     012230  001003                                                                                   BNE     50012$
2212                                                                                          ;YES
2213
2214 012232                                                         LET R0 := #20400          ;DO NOT CHECK TRAP LOCATIONS
     012232  012700  020400                                                                           MOV     #20400,R0
2215
2216 012236                                                 ELSE    ;NO
     012236  000405                                                                                   BR      50013$
     012240                                                                                           50012$:
2217
2218 012240                                                         LET @#KIPAR1 := PPOINT    ;SET UP POINTER TO CHECK THAT
     012240  016737  003724  172342                                                                   MOV     PPOINT,@#KIPAR1
2219                                                                                          ;MODULE
2220
2221 012246                                                         LET R0 := #20000          ;POINT TO PAR1 ADDRESS ZERO
     012246  012700  020000                                                                           MOV     #20000,R0
2222
2223 012252                                                 ENDIF
     012252                                                                                           50013$:
2224
2225 012252                                         ELSE
     012252  000410                                                                                   BR      50014$
     012254                                                                                           50002$:
2226
2227 012254                                                 IF PPOINT EQ #0 THEN              ;ARE WE CHECKING ENTIRE SYSTEM?
     012254  005767  003710                                                                           TST     PPOINT
     012260  001003                                                                                   BNE     50015$
2228                                                                                          ;YES
2229
2230 012262                                                         LET R0 := #400            ;SKIP TRAP LOCATIONS
     012262  012700  000400                                                                           MOV     #400,R0
2231
```

```
 012266                                    ELSE
 012266  000402                                                                          BR      50016$
 012270                                                                                  50015$:

 012270                                              LET R0 := PPCINT          ;FETCH POINTER ADDRESS
 012270  016700  003674                                                                  MOV     PPOINT,R0

 012274                                    ENDIF
 012274                                                                                  50016$:

 012274                          ENDIF
 012274                                                                                  50014$:

                                 ;CALCULATE CHECKSUM OF TRAP PAGE

 012274                          LET R1 := #0                                  ;ADDRESS OF TRAP LOCATION
 012274  005001                                                                          CLR     R1

 012276                          LET R3 := #0                                  ;CHECKSUM WORK REGISTER
 012276  005003                                                                          CLR     R3

 012300                          LOOP
 012300                                                                                  50017$:

 012300                                    LET R3 := R3 + (R1)+                ;CALCULATE CHECKSUM
 012300  062103                                                                          ADD     (R1)+,R3

 012302                          EXIF R1 EQ #400                      ;EXIF WHEN REACHED END OF TRAP PAGE
 012302  020127  000400                                                                  CMP     R1,#400
 012306  001401                                                                          BEQ     50020$

 012310                          ENDLOOP
 012310  000773                                                                          BR      50017$
 012312                                                                                  50020$:

 012312                          IF R3 NE CSTRPG THEN                 ;DO CHECKSUMS AGREE?
 012312  020367  005202                                                                  CMP     R3,CSTRPG
 012316  001403                                                                          BEQ     50021$
                                                                      ;NO

 012320                                    PUSH #4000                 ;INDICATE CHECKSUM ERROR
 012320  012746  004000          MOV       #4000,-(SP)       ;;PUSH #4000 ON STACK

 012324                          ELSE
 012324  000402                                                                          BR      50022$
 012326                                                                                  50021$:

 012326                                    PUSH #0                    ;INDICATE NO ERROR
 012326  012746  000000          MOV       #0,-(SP)          ;;PUSH #0 ON STACK
                                                                                         ;CHECKSUM ERROR

 012332                          ENDIF
 012332                                                                                  50022$:

 012332                          CALL      TRAPCT                              ;LOAD UP TRAP VECTORS WITH
 012332  004767  003672                                                                  JSR     PC,TRAPCT
                                                                               ;TRAP CATCHERS
```

```

012336                                  LET     R1 := PC - #.-Q4                ;R1 CONTAINS BEGINNING ADDRESS
012336  010701                                                                          MOV     PC,R1
012340  162701  000506                                                                  SUB     #.-Q4,R1
                                                                                ;OF PROGRAM

012344                                  LET     R2 := PC + #CHKSUM-.            ;R2 CONTAINS END ADDRESS OF
012344  010702                                                                          MOV     PC,R2
012346  062702  005154                                                                  ADD     #CHKSUM-.,R2
                                                                                ;PROGRAM

012352                                  LET     R3 := #0                        ;R3 CONTAINS CHECK SUM
012352  005003                                                                          CLR     R3

012354                                  LOOP
012354                                                                                  50023$:

012354                                          LET     R3 := R3 + (R1)+        ;ADD CONTENTS OF PROGRAM
012354  062103                                                                          ADD     (R1)+,R3

012356                                  EXIF    R1 EQ R2                ;LEAVE WHEN REACHED END OF PROGRAM
012356  020102                                                                          CMP     R1,R2
012360  001401                                                                          BEQ     50024$

012362                                  ENDLOOP
012362  000774                                                                          BR      50023$
012364                                                                                  50024$:

012364                                  IF (R2) NE R3 THEN                      ;DO WE HAVE A CHECKSUM ERROR
012364  021203                                                                          CMP     (R2),R3
012366  001407                                                                          BEQ     50025$
                                                                                ;YES


                                                ;FETCH LOCATION OF ERROR MESSAGE
012370                                          LET R1 := PC + #ERR-.
012370  010701                                                                          MOV     PC,R1
012372  062701  002641                                                                  ADD     #ERR-.,R1


                                                ;LOAD ADDRESS INTO TRAP INSTRUCTION
012376                                          LET 2$ := R1
012376  010167  000002                                                                  MOV     R1,2$


                                                ;TELL OPERATOR CHECK SUM ERROR
012402  104401                                  TYPE
012404  015233                  2$:             .WORD   ERR

012406                                  ENDIF
012406                                                                                  50025$:

012406                                  LET CONTROL := CONTROL SET.BY (SP)+     ;INDICATE RESULT OF CHECK SUM
012406  052667  003546                                                                  BIS     (SP)+,CONTROL
                                                                                ;OF TRAP PAGE


012412                                  PUSH    PC
012412  010746                          MOV     PC,-(SP)        ;;PUSH PC ON STACK

012414                                  LET     (SP) := (SP) + #HEAD-.          ;CALCULATE HEAD LOACTION
```

```
 012414  062716  003112                                                          ADD     #HEAD-..(SP)
 012420                                    POP     30$                  ;TO PRINT IT
 012420  012667  000002                    MOV     (SP)+,30$       ;;POP STACK INTO 30$
 012424  104401                            TYPE
 012426  015526                   30$:     .WORD   HEAD                 ;PRINT MEMORY MAP HEADER
 012430                                    LET START := PPOINT          ;SAVE STARTING POINT
 012430  016767  003534  003540                                                  MOV     PPOINT,START
 012436                                    LET L4KST := PPOINT          ;WE ARE AT THIS 4K BANK
 012436  016767  003526  003534                                                  MOV     PPOINT,L4KST
 012444                                    LET R3 := #172100            ;FETCH CSR LOCATIONS
 012444  012703  172100                                                          MOV     #172100,R3
 012450                                    LET TRCATCH := #240          ;NOP TRAP CATCHER
 012450  012767  000240  003706                                                  MOV     #240,TRCATCH
 012456                                    INCR R2 FROM #0 TO #16. BY #1        ;ENABLE PARITY FOR ALL CSRS
 012456  005002                                                                  CLR     R2
 012460  000401                                                                  BR      50026$
 012462                                                                          50027$:
 012462  005202                                                                  INC     R2
 012464                                                                          50026$:
 012464  020227  000020                                                          CMP     R2,#16.
 012470  003003                                                                  BGT     50030$
 012472                                         LET (R3)+ := #1         ;
 012472  012723  000001                                                          MOV     #1,(R3)+
 012476                                    ENDINC
 012476  000771                                                                  BR      50027$
 012500                                                                          50030$:
 012500                                    LET TRCATCH := #0            ;RESUME HALT IF TRAP
 012500  005067  003660                                                          CLR     TRCATCH
 012504                                    LET R4 := #114          ;LOAD PARITY LOCATION INTO R4
 012504  012704  000114                                                          MOV     #114,R4
                                                                   ;SO THAT NEXT INSTRUCION WILL
                                                                   ;WORK ON ALL MACHINES
 012510                                    LET (R4) := PC + #PARTRP-.   ;LOAD PARITY TRAP LOCATION WITH
 012510  010714                                                                  MOV     PC,(R4)
 012512  062714  002140                                                          ADD     #PARTRP-.,(R4)
                                                                   ;ADDRESS OF PARITY TRAP ROUTINE
 012516                                    IF #125252 EQ (R0)+ THEN     ;IS THIS LOCATION VOLATILE
 012516  022720  125252                                                          CMP     #125252,(R0)+
 012522  001016                                                                  BNE     50031$
                                                ;IF PARITY ERROR THEN ERROR COUNT ALREADY DONE
 012524                                         IF #2 SETIN CONTROL THEN
 012524  032767  000002  003426                                                  BIT     #2,CONTROL
 012532  001404                                                                   BEQ     50032$

 012534                                                   LET CONTROL := CONTROL CLR.BY #2        ;CLEAR PARITY
 012534  042767  000002  003416                                                   BIC     #2,CONTROL
                                                                                          ;ERROR FLAG

 012542                                          ELSE
 012542  000405                                                                   BR      50033$
 012544                                                                           50032$:

                                                          ;INDICATE NON-VOLATILE
 012544                                                   LET CONTROL := CONTROL SET.BY #100000
 012544  052767  100000  003406                                                   BIS     #100000,CONTROL

 012552                                                   LET NONVOL := NONVOL + #1       ;UPDATE NONVOL COUNT
 012552  005267  003404                                                           INC     NONVOL

 012556                                          ENDIF
 012556                                                                           50033$:

 012556                                 ELSE
 012556  000412                                                                   BR      50034$
 012560                                                                           50031$:

                                                 ;IF PARITY ERROR THEN ERROR COUNT ALREADY DONE
 012560                                          IF #2 SETIN CONTROL THEN
 012560  032767  000002  003372                                                   BIT     #2,CONTROL
 012566  001404                                                                   BEQ     50035$

 012570                                                   LET CONTROL := CONTROL CLR.BY #2        ;CLEAR PARITY
 012570  042767  000002  003362                                                   BIC     #2,CONTROL
                                                                                          ;ERROR FLAG

 012576                                          ELSE
 012576  000402                                                                   BR      50036$
 012600                                                                           50035$:

 012600                                                   LET VOL := VOL + #1             ;UPDATE VOL COUNT
 012600  005267  003360                                                           INC     VOL

 012604                                          ENDIF
 012604                                                                           50036$:

 012604                                 ENDIF
 012604                                                                           50034$:

 012604                                 LET R4 :- #0                        ;FLAG FOR NON VOLATILE MEMORY HITS
 012604  005004                                                                   CLR     R4

                               ;        FAKED OUT LOOP BECAUSE LOOP IS OUT SIDE OF ADDRESS RANGE

 012606                        STLOOP:           IF #1 SETIN CONTROL THEN         ;ARE WE MAPPED
 012606  032767  000001  003344                                                   BIT     #1,CONTROL
 012614  001407                                                                   BEQ     50037$
                                                                           ;YES

 012616                                                   IF @#KIPAR1 EQ ENDADD THEN      ;IS IT TIME TO LEAVE?
```

```
012616  023767  172342  003350                                              CMP     @#KIPAR1,ENDADD
012624  001002                                                              BNE     50040$
                                                                            ;YES
012626                                         INLINE <JMP     EDLOOP> ;LEAVE LOOP
012626  000167  001356                                                      JMP     EDLOOP

012632                                         ENDIF
012632                                                                      50040$:

012632                                  ELSE
012632  000405                                                              BR      50041$
012634                                                                      50037$:

012634                                         IF RO EQ ENDADD THEN    ;IS IT TIME TO LEAVE?
012634  020067  003334                                                      CMP     RO,ENDADD
012640  001002                                                              BNE     50042$
                                                               ;YES

012642                                                INLINE <JMP     EDLOOP> ;LEAVE LOOP
012642  000167  001342                                                      JMP     EDLOOP

012646                                         ENDIF
012646                                                                      50042$:

012646                                  ENDIF
012646                                                                      50041$:

012646                                  IF #125252 EQ (RO)+ THEN       ;DOES THIS LOCATION CONTAIN
012646  022720  125252                                                      CMP     #125252,(RO)+
012652  001013                                                              BNE     50043$
                                                               ;CORRECT DATA

                                               ;IF PARITY ERROR THEN ERROR COUNT ALREADY DONE
012654                                         IF #2 SETIN CONTROL THEN
012654  032767  000002  003276                                              BIT     #2,CONTROL
012662  001404                                                              BEQ     50044$

                                                      ;CLEAR PARITY ERROR FLAG
012664                                                LET CONTROL := CONTROL CLR.BY #2
012664  042767  000002  003266                                              BIC     #2,CONTROL

012672                                         ELSE
012672  000402                                                              BR      50045$
012674                                                                      50044$:

012674                                                LET NONVOL := NONVOL + #1       ;YES, UPDATE
012674  005267  003262                                                      INC     NONVOL
                                                                            ;NON=VOLATILE COUNT

012700                                         ENDIF
012700                                                                      50045$:

012700                                  ELSE
012700  000412                                                              BR      50046$
012702                                                                      50043$:

                                               ;IF PARITY ERROR THEN ERROR COUNT ALREADY DONE
```

```
012702                                           IF #2 SETIN CONTROL THEN
012702 032767 000002 003250                                                  BIT    #2,CONTROL
012710 001404                                                                BEQ    50047$

                                                        ;CLEAR PARITY ERROR FLAG
012712                                                  LET CONTROL := CONTROL CLR.BY #2
012712 042767 000002 003240                                                  BIC    #2,CONTROL

012720                                           ELSE
012720 000402                                                                BR     50050$
012722                                                                       50047$:

012722                                                  LET VOL := VOL + #1  ;NO. UPDATE
012722 005267 003236                                                         INC    VOL
                                                                             ;VOLATILE COUNT

012726                                           ENDIF
012726                                                                       50050$:

012726                                    ENDIF
012726                                                                       50046$:

012726                                    IF #1 SETIN CONTROL THEN    ;ARE WE MAPPED
012726 032767 000001 003224                                                  BIT    #1,CONTROL
012734 001421                                                                BEQ    50051$

012736                                           IF @#KIPAR1 EQ @#KIPAR2 THEN  ;ARE PARS IN SAME BANK?
012736 023737 172342 172344                                                  CMP    @#KIPAR1,@#KIPAR2
012744 001014                                                                BNE    50052$
                                                                             ;YES


                                                        ;DISTANCE BETWEEN BEGINNING OF PROGRAM AND HERE
012746                                                  LET R1 := PC - #.-Q4
012746 010701                                                                MOV    PC,R1
012750 162701 001116                                                         SUB    #.-Q4,R1

                                                        ;CLEAR OUT PAR POINTER
012754                                                  LET R1 := R1 CLR.BY #160000
012754 042701 160000                                                         BIC    #160000,R1

                                                        ;CLEAR OUT PAR POINTER
012760                                                  LET R3 := R0 CLR.BY #160000
012760 010003                                                                MOV    R0,R3
012762 042703 160000                                                         BIC    #160000,R3

                                                        ;HAVE WE REACHED BEGINNING OF PROGRAM
012766                                                  IF R3 EQ R1 THEN
012766 020301                                                                CMP    R3,R1
012770 001002                                                                BNE    50053$

                                                               ;BUMP POINTER TO END OF PROGRAM
012772                                                         LET R0 := R0 + #DIF1
012772 062700 006014                                                         ADD    #DIF1,R0

012776                                                  ENDIF
012776                                                                       50053$:

```

```
012776                                         ENDIF
012776                                                                               50052$:

012776                                 ELSE
012776  000407                                                                       BR      50054$
013000                                                                               50051$:


                                               ;ADDRESS OF BEGINING OF PROGRAM
013000                                         LET R1 := PC - #.-Q4
013000  010701                                                                       MOV     PC,R1
013002  162701  001150                                                               SUB     #.-Q4,R1

013006                                         IF R0 EQ R1 THEN       ;POINTER REACHED PROGRAM?
013006  020001                                                                       CMP     R0,R1
013010  001002                                                                       BNE     50055$


                                                    ;BUMP POINTER TO END OF PROGRAM
013012                                              LET R0 := R0 + #DIF1
013012  062700  006014                                                               ADD     #DIF1,R0

013016                                         ENDIF
013016                                                                               50055$:

013016                                 ENDIF
013016                                                                               50054$:

013016                                 IF #17777 SETIN R0 THEN        ;ARE WE AT 4K BOUNDARY?
013016  032700  017777                                                               BIT     #17777,R0
013022  001402                                                                       BEQ     50056$
                                                                      ;NO

013024                                         INLINE <JMP STLOOP>    ;CONTINUE LOOP & BYPASS PRINT
013024  000167  177556                                                               JMP STLOOP
                                                                      ;ROUTINES

013030                                 ENDIF
013030                                                                               50056$:


                                       ;ENABLE PARITY
013030                                 LET R3 := #172100              ;FETCH CSR LOCATIONS
013030  012703  172100                                                               MOV     #172100,R3

013034                                 LET TRCATCH := #240            ;NOP TRAP CATCHER
013034  012767  000240  003322                                                       MOV     #240,TRCATCH

013042                                 INCR R2 FROM #0 TO #16. BY #1  ;ENABLE PARITY FOR ALL CSRS
013042  005002                                                                       CLR     R2
013044  000401                                                                       BR      50057$
013046                                                                               50060$:
013046  005202                                                                       INC     R2
013050                                                                               50057$:
013050  020227  000020                                                               CMP     R2,#16.
013054  003003                                                                       BGT     50061$

013056                                         LET (R3)+ := #1                       ;
013056  012723  000001                                                               MOV     #1,(R3)+

        013062                                           ENDINC
        013062  000771                                                                            BR      50060$
        013064                                                                                    50061$:

        013064                                           LET TRCATCH := #0               ;RESUME HALT IF TRAP
        013064  005067  003274                                                                    CLR     TRCATCH

        013070                                           LET R4 := #0            ;CLEAR PRINT FLAG
        013070  005004                                                                            CLR     R4

        013072                                           IF PARCNT NE #0 THEN    ;WE FOUND SOME PARITY ERRORS
        013072  005767  003074                                                                    TST     PARCNT
        013076  001405                                                                            BEQ     50062$

        013100                                                   IF VOL LOS NONVOL THEN   ;MAKE SURE IT IS
        013100  026767  003060  003054                                                            CMP     VOL,NONVOL
        013106  101001                                                                            BHI     50063$
                                                                                          ;STILL NON VOLATILE

        013110                                                           LET R4 := R4 + #1        ;HITS
        013110  005204                                                                            INC     R4

        013112                                                   ENDIF
        013112                                                                                    50063$:

        013112                                           ENDIF
        013112                                                                                    50062$:

        013112                                           IF VOL NE #0 THEN        ;HAVE FOUND ANY HITS?
        013112  005767  003046                                                                    TST     VOL
        013116  001405                                                                            BEQ     50064$
                                                                                  ;YES

        013120                                                   IF VOL LE NONVOL THEN    ;MIGHT BE VOL
        013120  026767  003040  003034                                                            CMP     VOL,NONVOL
        013126  003001                                                                            BGT     50065$
                                                                                          ;NO

        013130                                                           LET R4 := R4 + #1        ;HITS
        013130  005204                                                                            INC     R4

        013132                                                   ENDIF
        013132                                                                                    50065$:

        013132                                           ENDIF
        013132                                                                                    50064$:

        013132                                           IF R4 EQ #0 THEN         ;WE HAVE HITS IN NON VOLATILE
        013132  005704                                                                            TST     R4
        013134  001002                                                                            BNE     50066$
                                                                                  ;MEMORY

        013136                                                   INLINE <JMP      20$>    ;GO TO NORMAL ROUTINE (NO ERRORS)
        013136  000167  000450                                                                    JMP     20$

        013142                                           ENDIF
        013142                                                                                    50066$:
```

```

013142                                                      IF L4KST EQ START THEN  ;ONLY NEED TO PRINT ONE LINE
013142  026767  003032  003026                                                          CMP     L4KST,START
013150  001066                                                                          BNE     50067$

                                                                      ;TAKE CARE OF ANY PROBLEMS WITH START ADDRESS
                                                                      ;BEING VOLATILE WITHIN NON-VOLATILE MEMORY
013152                                                                LET CONTROL := CONTROL SET.BY #100000
013152  052767  100000  003000                                                          BIS     #100000,CONTROL

013160                                                                IF #1 SETIN CONTROL THEN        ;MAPPED
013160  032767  000001  002772                                                          BIT     #1,CONTROL
013166  001404                                                                          BEQ     50070$


                                                                              ;FETCH LAST BANK ADDRESS
013170                                                                        LET END := @#KIPAR1
013170  013767  172342  002770                                                          MOV     @#KIPAR1,END

013176                                                                ELSE
013176  000405                                                                          BR      50071$
013200                                                                                  50070$:


                                                                              ;FETCH LAST BANK ADDRESS
013200                                                                        LET END := R0 - #2        ;
013200  010067  002762                                                                  MOV     R0,END
013204  162767  000002  002754                                                          SUB     #2,END

013212                                                                ENDIF
013212                                                                                  50071$:

013212                                                                CALL    PRINT   ;PRINT ADDRESSES
013212  004767  001172                                                                  JSR     PC,PRINT

013216                                                                CALL    TAB                     ;PRINT TAB
013216  004767  001142                                                                  JSR     PC,TAB

013222                                                                PUSH    R1
013222  010146                          MOV     R1,-(SP)              ;;PUSH R1 ON STACK

013224                                                                LET R1 := VOL   ;FETCH VOL COUNT
013224  016701  002734                                                                  MOV     VOL,R1

013230                                                                TYPDEC  R1      ;PRINT ERROR COUNT
013230  010146                          MOV     R1,-(SP)              ;;SAVE R1 FOR TYPEOUT
013232  104405                          TYPDS                         ;;GO TYPE--DECIMAL ASCII WITH SIGN

013234                                                                CALL    TAB                     ;PRINT TAB
013234  004767  001124                                                                  JSR     PC,TAB

013240                                                                LET R1 := PARCNT        ;FETCH PARITY COUNT
013240  016701  002726                                                                  MOV     PARCNT,R1

013244                                                                TYPDEC  R1      ;PRINT PARITY ERROR CNT.
013244  010146                          MOV     R1,-(SP)              ;;SAVE R1 FOR TYPEOUT
013246  104405                          TYPDS                         ;;GO TYPE--DECIMAL ASCII WITH SIGN

013250                                                                IF PARCNT EQ #10. THEN  ;DO WE HAVE 10 OR MORE
```

```
       013250  026727  002716  000012                                                            CMP     PARCNT,#10.
       013256  001002                                                                            BNE     50072$
 2552                                                                                            ;PARITY ERRORS
 2553
 2554  013260  104401  015100                                                   TYPE      ,MORE  ;PRINT OR MORE
 2555
 2556  013264                                                           ENDIF
       013264                                                                                    50072$:
 2557
 2558  013264                                                           POP       R1       ;RESTORE R1
       013264  012601                          MOV       (SP)+,R1       ;;POP STACK INTO R1
 2559
 2560  013266                                                           CALL      CRLF2          ;PRINT <CR> <LF>
       013266  004767  001046                                                                    JSR     PC,CRLF2
 2561
 2562                                                                   ;UP DATE START BANK TO NEXT LEVEL TO BE TESTED
 2563  013272                                                           IF #1 SETIN CONTROL THEN         ;MAPPED
       013272  032767  000001  002660                                                            BIT     #1,CONTROL
       013300  001407                                                                            BEQ     50073$
 2564
 2565  013302                                                                   LET START := @#KIPAR1 + #200
       013302  013767  172342  002666                                                            MOV     @#KIPAR1,START
       013310  062767  000200  002660                                                            ADD     #200,START
 2566
 2567  013316                                                           ELSE
       013316  000402                                                                            BR      50074$
       013320                                                                                    50073$:
 2568
 2569  013320                                                                   LET START := R0
       013320  010067  002652                                                                    MOV     R0,START
 2570
 2571  013324                                                           ENDIF
       013324                                                                                    50074$:
 2572
 2573  013324                                                   ELSE
       013324  000532                                                                            BR      50075$
       013326                                                                                    50067$:
 2574
 2575  013326                                                           IF #1 SETIN CONTROL THEN         ;MAPPED
       013326  032767  000001  002624                                                            BIT     #1,CONTROL
       013334  001407                                                                            BEQ     50076$
 2576
 2577  013336                                                                   LET END := L4KST - #200 ;SUB 4K
       013336  016767  002636  002622                                                            MOV     L4KST,END
       013344  162767  000200  002614                                                            SUB     #200,END
 2578
 2579  013352                                                           ELSE
       013352  000406                                                                            BR      50077$
       013354                                                                                    50076$:
 2580
 2581  013354                                                                   LET END := L4KST - #2    ;LAST ADDRESS
       013354  016767  002620  002604                                                            MOV     L4KST,END
       013362  162767  000002  002576                                                            SUB     #2,END
 2582
 2583  013370                                                           ENDIF
       013370                                                                                    50077$:
 2584
```

```
013370                                                  CALL    PRINT   ;PRINT ADDRESSES
013370 004767 001014                                                                 JSR     PC,PRINT

013374                                                  IF CONTROL LT #0 THEN   ;ARE WE CHECKING NON-VOL
013374 005767 002560                                                                 TST     CONTROL
013400 002012                                                                        BGE     50100$

013402                                                         CALL    TAB           ;PRINT TAB
013402 004767 000756                                                                 JSR     PC,TAB

                                                               ;NO ERRORS IN THIS PART
013406                                                         TYPDEC  #0
013406 012746 000000              MOV    #0,-(SP)       ;;SAVE #0 FOR TYPEOUT
013412 104405                     TYPDS                 ;;GO TYPE--DECIMAL ASCII WITH SIGN

013414                                                         CALL    TAB           ;PRINT TAB
013414 004767 000744                                                                 JSR     PC,TAB

013420                                                         TYPDEC  #0      ;NO PARITY EITHER
013420 012746 000000              MOV    #0,-(SP)       ;;SAVE #0 FOR TYPEOUT
013424 104405                     TYPDS                 ;;GO TYPE--DECIMAL ASCII WITH SIGN

013426                                                  ENDIF
013426                                                                               50100$:

                                                        ;INDICATE THAT NOW THIS IS NON-VOLATILE
013426                                                  LET CONTROL := CONTROL SET.BY #100000
013426 052767 100000 002524                                                          BIS     #100000,CONTROL

013434                                                  CALL    CRLF2           ;PRINT <CR><LF>
013434 004767 000700                                                                 JSR     PC,CRLF2

013440                                                  IF #1 SETIN CONTROL THEN        ;MAPPED
013440 032767 000001 002512                                                          BIT     #1,CONTROL
013446 001404                                                                        BEQ     50101$


                                                               ;LAST 4K BANK
013450                                                         LET END := @#KIPAR1
013450 013767 172342 002510                                                          MOV     @#KIPAR1,END

013456                                                  ELSE
013456 000405                                                                        BR      50102$
013460                                                                               50101$:


                                                               ;LAST ADDRESS
013460                                                         LET END := R0 - #2    ;
013460 010067 002502                                                                 MOV     R0,END
013464 162767 000002 002474                                                          SUB     #2,END

013472                                                  ENDIF
013472                                                                               50102$:

013472                                                  LET START := L4KST
013472 016767 002502 002476                                                          MOV     L4KST,START

013500                                                  CALL PRINT      ;PRINT ADDRESSES
013500 004767 000704                                                                 JSR     PC,PRINT
```

```

013504                                                  CALL    TAB             ;PRINT TAB
013504  004767  000654                                                          JSR     PC,TAB

013510                                                  PUSH    R1      ;SAVE R1
013510  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK

013512                                                  LET R1 := VOL   ;FETCH VOL COUNT
013512  016701  002446                                                          MOV     VOL,R1

013516                                                  TYPDEC  R1      ;PRINT ANY ERRORS
013516  010146                  MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
013520  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN

013522                                                  CALL    TAB             ;PRINT TAB
013522  004767  000636                                                          JSR     PC,TAB

013526                                                  LET R1 := PARCNT        ;FETCH PARITY COUNT
013526  016701  002440                                                          MOV     PARCNT,R1

013532                                                  TYPDEC  R1      ;PRINT PARITY ERRORS
013532  010146                  MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
013534  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN

013536                                                  IF PARCNT EQ #10. THEN  ;DO WE HAVE 10 OR MORE
013536  026727  002430  000012                                                  CMP     PARCNT,#10.
013544  001002                                                                  BNE     50103$
                                                                                ;PARITY ERRORS

013546  104401  015100                                          TYPE    ,MORE   ;PRINT OR MORE

013552                                                  ENDIF
013552                                                                          50103$:

013552                                                  POP     R1              ;RESTORE R1
013552  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1

013554                                                  CALL    CRLF2           ;PRINT <CR><LF>
013554  004767  000560                                                          JSR     PC,CRLF2


                                                        ;UP DATE START BANK TO NEXT LEVEL TO BE TESTED
013560                                                  IF #1 SETIN CONTROL THEN
013560  032767  000001  002372                                                          #1,CONTROL
013566  001407                                                                  BEQ     50104$

013570                                                          LET START := @#KIPAR1 + #200
013570  013767  172342  002400                                                  MOV     @#KIPAR1,START
013576  062767  000200  002372                                                  ADD     #200,START

013604                                                  ELSE
013604  000402                                                                  BR      50105$
013606                                                                          50104$:

013606                                                          LET START := R0
013606  010067  002364                                                          MOV     R0,START

013612                                                  ENDIF
```

```
      013612                                                                  50105$:

      013612                                              ENDIF
      013612                                                                  50075$:

      013612                          20$:            IF R4 EQ #0 THEN        ;HAVE WE PRINTED ANYTHING UP TO NOW
      013612  005704                                                          TST     R4
      013614  001132                                                          BNE     50106$

      013616                                              IF START EQ L4KST THEN  ;MAKE SURE THAT CONTROL IS
      013616  026767  002354  002354                                          CMP     START,L4KST
      013624  001013                                                          BNE     50107$
                                                                      ;SET PROPERLY

      013626                                                    IF VOL GE NONVOL THEN
      013626  026767  002332  002326                                          CMP     VOL,NONVOL
      013634  002404                                                          BLT     50110$

      013636                                                          LET CONTROL := CONTROL CLR.BY #100000
      013636  042767  100000  002314                                          BIC     #100000,CONTROL

      013644 •                                                  ELSE
      013644  000403                                                          BR      50111$
      013646                                                                  50110$:

      013646                                                          LET CONTROL := CONTROL SET.BY #100000
      013646  052767  100000  002304                                          BIS     #100000,CONTROL

      013654                                                    ENDIF
      013654                                                                  50111$:

      013654                                              ENDIF
      013654                                                                  50107$:

      013654                                              IF VOL GE NONVOL THEN   ;NO, ARE WE IN NON VOLATILE
      013654  026767  002304  002300                                          CMP     VOL,NONVOL
      013662  002451                                                          BLT     50112$
                                                                      ;MEMORY?


                                                                ;  , WERE WE LOOKING FOR NON-VOLATILE
      013664                                                    IF CONTROL LT #0 THEN
      013664  005767  002270                                                  TST     CONTROL
      013670  002045                                                          BGE     50113$


                                                                      ;ARE WE A MAPPED SYSTEM?
      013672                                                          IF #1 SETIN CONTROL THEN
      013672  032767  000001  002260                                          BIT     #1,CONTROL
      013700  001407                                                          BEQ     50114$


                                                                      ;FETCH LAST 4K BANK
      013702                                                                LET END := L4KST - #200
      013702  016767  002272  002256                                          MOV     L4KST,END
      013710  162767  000200  002250                                          SUB     #200,END

      013716                                                          ELSE
      013716  000406                                                          BR      50115$
      013720                                                                  50114$:
```

```

                                                                                        ;FETCH LAST ADDRESS OF
                                                                                        ;OF NON-VOLATILE MEMORY
013720                                                                                  LET END := L4KST - #2
013720  016767  002254  002240                                                                  MOV     L4KST,END
013726  162767  000002  002232                                                                  SUB     #2,END

013734                                                                          ENDIF
013734                                                                                          50115$:

013734                                                                          CALL    PRINT                   ;PRINT ADDRESSES
013734  004767  000450                                                                          JSR     PC,PRINT

013740                                                                          CALL    TAB                     ;PRINT TAB
013740  004767  000420                                                                          JSR     PC,TAB

013744                                                                          TYPDEC  #0      ;NO ERRORS
013744  012746  000000                          MOV     #0,-(SP)                ;;SAVE #0 FOR TYPEOUT
013750  104405                                  TYPDS                           ;;GO TYPE--DECIMAL ASCII WITH SIGN

013752                                                                          CALL    TAB                     ;PRINT TAB
013752  004767  000406                                                                          JSR     PC,TAB

013756                                                                          TYPDEC  #0      ;NO PARITY ERRORS
013756  012746  000000                          MOV     #0,-(SP)                ;;SAVE #0 FOR TYPEOUT
013762  104405                                  TYPDS                           ;;GO TYPE--DECIMAL ASCII WITH SIGN

013764                                                                          CALL    CRLF2                   ;PRINT <CR><LF>
013764  004767  000350                                                                          JSR     PC,CRLF2

013770                                                                          LET START := L4KST
013770  016767  002204  002200                                                                  MOV     L4KST,START

                                                                                ;INDICATE NOW LOOKING FOR VOLATILE
013776                                                                          LET CONTROL := CONTROL CLR.BY #100000
013776  042767  100000  002154                                                                  BIC     #100000,CONTROL

014004                                                                  ENDIF
014004                                                                                          50113$:

014004                                                          ELSE
014004  000436                                                                                  BR      50116$
014006                                                                                          50112$:

                                                                        ;YES, WERE WE LOOKING FOR VOLATILE
014006                                                                  IF CONTROL GE #0 THEN
014006  005767  002146                                                                          TST     CONTROL
014012  002433                                                                                  BLT     50117$

                                                                                ;YES, ARE WE MAPPED
014014                                                                          IF #1 SETIN CONTROL THEN
014014  032767  000001  002136                                                                  BIT     #1,CONTROL
014022  001407                                                                                  BEQ     50120$

                                                                                        ;FETCH LAST 4K BANK BEFORE CHANGE
014024                                                                                  LET END := L4KST - #200
014024  016767  002150  002134                                                                  MOV     L4KST,END
```

```
     014032  162767  000200  002126                                                    SUB     #200,END

     014040                                                                ELSE
     014040  000406                                                                    BR      50121$
     014042                                                                            50120$:


                                                                                 ;FETCH LAST ADDRESS BEFORE CHANGE
     014042                                                                      LET END := L4KST - #2
     014042  016767  002132  002116                                                    MOV     L4KST,END
     014050  162767  000002  002110                                                    SUB     #2,END

     014056                                                                ENDIF
     014056                                                                            50121$:


                                                                           ;PRINT ADDRESSES
     014056                                                                CALL PRINT
     014056  004767  000326                                                            JSR     PC,PRINT

     014062                                                                CALL    CRLF2   ;PRINT <CR><LF>
     014062  004767  000252                                                            JSR     PC,CRLF2


                                                                           ;INDICATE NOW LOOKING FOR
                                                                           ;NON-VOLATILE MEMORY
     014066                                                                LET CONTROL := CONTROL SET.BY #100000
     014066  052767  100000  002064                                                    BIS     #100000,CONTROL


                                                                           ;BUMP POINTER
     014074                                                                LET START := L4KST
     014074  016767  002100  002074                                                    MOV     L4KST,START

     014102                                                      ENDIF
     014102                                                                            50117$:

     014102                                            ENDIF
     014102                                                                            50116$:

     014102                                  ENDIF
     014102                                                                            50106$:

     014102                                  IF #40 SETIN CONTROL THEN       ;DO WE PRINT TRAP PAGE CHECKSUM
     014102  032767  000040  002050                                                    BIT     #40,CONTROL
     014110  001410                                                                    BEQ     50122$
                                                                                 ;ERROR.  YES

     014112                                            LET R1 := PC + #TRAPER-.        ;FETCH MESSAGE
     014112  010701                                                                    MOV     PC,R1
     014114  062701  000775                                                            ADD     #TRAPER-.,R1

     014120  104401                                    TYPE
     014122  015111                  100$:             .WORD   TRAPER                  ;TELL OF ERROR

     014124                                            LET CONTROL := CONTROL CLR.BY #40       ;INDICATE
     014124  042767  000040  002026                                                    BIC     #40,CONTROL
                                                                                 ;MESSAGE ALREADY PRINTED
     014132                                  ENDIF
     014132                                                                            50122$:

```

```
 014132                                                 LET VOL := #0           ;CLEAR VOLATILE FLAG
 014132  005067  002026                                                                 CLR     VOL

 014136                                                 LET NONVOL := #0        ;CLEAR NON-VOLATILE FLAG
 014136  005067  002020                                                                 CLR     NONVOL

 014142                                                 LET PARCNT := #0        ;CLEAR PARITY COUNT
 014142  005067  002024                                                                 CLR     PARCNT

 014146                                                 IF #1 SETIN CONTROL THEN        ;ARE WE MAPPED?
 014146  032767  000001  002004                                                         BIT     #1,CONTROL
 014154  001411                                                                         BEQ     50123$
                                                                                ;YES

 014156                                                         LET @#KIPAR1 := @#KIPAR1 + #200 ;GOTO NEXT BANK
 014156  062737  000200  172342                                                         ADD     #200,@#KIPAR1

 014164                                                         LET R0 := #20000        ;RETURN TO ADDRESS 0 PAR1
 014164  012700  020000                                                                 MOV     #20000,R0

 014170                                                         LET L4KST := @#KIPAR1   ;FETCH NEXT BANK
 014170  013767  172342  002002                                                         MOV     @#KIPAR1,L4KST

 014176                                                 ELSE
 014176  000402                                                                         BR      50124$
 014200                                                                                 50123$:

 014200                                                         LET L4KST := R0         ;FETCH NEXT BANK
 014200  010067  001774                                                                 MOV     R0,L4KST

 014204                                                 ENDIF
 014204                                                                                 50124$:

 014204                                         INLINE <JMP     STLOOP>         ;CONTINUE LOOP
 014204  000167  176376                                                                 JMP     STLOOP

 014210                                 EDLOOP: IF R4 EQ #0 THEN        ;DO WE NEED TO PRINT LAST MEMORY UNDER TEST
 014210  005704                                                                         TST     R4
 014212  001041                                                                         BNE     50125$
                                                                ;YES

 014214                                                 IF #1 SETIN CONTROL THEN        ;ARE WE MAPPED?
 014214  032767  000001  001736                                                         BIT     #1,CONTROL
 014222  001407                                                                         BEQ     50126$

 014224                                                         LET END :- @#KIPAR1 - #200      ;FETCH LAST 4K BANK
 014224  013767  172342  001734                                                         MOV     @#KIPAR1,END
 014232  162767  000200  001726                                                         SUB     #200,END

 014240                                                 ELSE
 014240  000405                                                                         BR      50127$
 014242                                                                                 50126$:

 014242                                                         LET END := R0 - #2      ;FETCH LAST ADDRESS
 014242  010067  001720                                                                 MOV     R0,END
 014246  162767  000002  001712                                                         SUB     #2,END

```

```
2797 014254                                      ENDIF
     014254                                                                        50127$:
2798
2799 014254                                      CALL    PRINT                 ;PRINT ADDRESSES
     014254  004767  000130                                                        JSR     PC,PRINT
2800
2801 014260                                      IF CONTROL LT #0 THEN         ;WAS LAST CHUNK OF MEMORY
     014260  005767  001674                                                        TST     CONTROL
     014264  002012                                                                BGE     50130$
2802                                                                       ;NON-VOL - YES, PRINT NO ERRORS
2803
2804 014266                                              CALL    TAB           ;PRINT TAB
     014266  004767  000072                                                        JSR     PC,TAB
2805
2806 014272                                              TYPDEC  #0            ;PRINT ANY ERRORS
     014272  012746  000000                MOV     #0,-(SP)        ;;SAVE #0 FOR TYPEOUT
     014276  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2807
2808 014300                                              CALL    TAB           ;PRINT TAB
     014300  004767  000060                                                        JSR     PC,TAB
2809
2810 014304                                              TYPDEC  #0            ;PARITY ERRORS
     014304  012746  000000                MOV     #0,-(SP)        ;;SAVE #0 FOR TYPEOUT
     014310  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2811
2812 014312                                      ENDIF
     014312                                                                        50130$:
2813
2814 014312                                      CALL    CRLF2           ;PRINT <CR><LF>
     014312  004767  000022                                                        JSR     PC,CRLF2
2815
2816 014316                              ENDIF
     014316                                                                        50125$:
2817
2818                                     ;FETCH LOCATION OF FINISHED MESSAGE
2819 014316                              LET R1 := PC + #TAF-.
     014316  010701                                                                MOV     PC,R1
     014320  062701  001037                                                        ADD     #TAF-.,R1
2820
2821                                     ;LOAD ADDRESS INTO TRAP INSTRUCTION
2822 014324                              LET 18$ := R1
     014324  010167  000002                                                        MOV     R1,18$
2823
2824                                     ;TELL OPERATOR WE ARE FINISHED
2825 014330  104401                      TYPE
2826 014332  015357              18$:    .WORD   TAF
2827
2828 014334                              INLINE  <HALT>
     014334  000000                                                                HALT
2829
2830 014336                      ENDRTN
     014336                                                                        50000$:
     014336                                                                        50001$:
     014336  000207                                                                RTS     PC
```

```
 014340                       ROUTINE CRLF2
 014340                                                                               CRLF2:

                                      ;SAVE R1
 014340                               PUSH     R1
 014340  010146                       MOV      R1,-(SP)          ;;PUSH R1 ON STACK

                                      ;FETCH LOCATION OF <CR><LF> MESSAGE
 014342                               LET R1 := PC + #CRLF1-.
 014342  010701                                                                       MOV      PC,R1
 014344  062701  001611                                                               ADD      #CRLF1-.,R1

                                      ;LOAD ADDRESS INTO TRAP INSTRUCTION
 014350                               LET 5$ := R1
 014350  010167  000002                                                               MOV      R1,5$

                                      ;TELL OPERATOR <CR><LF>
 014354  104401                       TYPE
 014356  016155               5$:     .WORD    CRLF1

                                      ;RESTORE R1
 014360                               POP      R1
 014360  012601                       MOV      (SP)+,R1          ;;POP STACK INTO R1

 014362                       ENDRTN
 014362                                                                               50000$:
 014362                                                                               50001$:
 014362  000207                                                                       RTS      PC
```

```
   2852 014364                         ROUTINE TAB
        014364                                                                          TAB:
   2853
   2854                                        ;SAVE R1
   2855 014364                                 PUSH    R1
        014364  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   2856
   2857                                        ;FETCH LOCATION OF TAB MESSAGE
   2858 014366                                 LET R1 := PC + #TAB1-.
        014366  010701                                                                  MOV     PC,R1
        014370  062701  001060                                                          ADD     #TAB1-.,R1
   2859
   2860                                        ;LOAD ADDRESS INTO TRAP INSTRUCTION
   2861 014374                                 LET 6$ := R1
        014374  010167  000002                                                          MOV     R1,6$
   2862
   2863                                        ;TELL OPERATOR TAB
   2864 014400  104401                         TYPE
   2865 014402  015450                 6$:     .WORD   TAB1
   2866
   2867                                        ;RESTORE R1
   2868 014404                                 POP     R1
        014404  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
   2869
   2870 014406                         ENDRTN
        014406                                                                          50000$:
        014406                                                                          50001$:
        014406  000207                                                                  RTS     PC
```

```
                                        .SBTTL  PRINT ADDRESSES ROUTINE

                                ;THIS ROUTINE PRINTS THE ADDRESSES IN THE MEMORY MAP

014410                          ROUTINE PRINT
014410                                                                                  PRINT:

014410                                  IF #1 SETIN CONTROL THEN        ;ARE WE A MAPPED SYSTEM
014410  032767  000001  001542                                                          BIT     #1,CONTROL
014416  001416                                                                          BEQ     50002$

014420                                          LET R2 := START         ;YES, FETCH 4K BANK
014420  016702  001552                                                                  MOV     START,R2

014424                                          LET R1 := #0            ;ALWAYS AT 4K BOUNDARY
014424  005001                                                                          CLR     R1

014426                                          CALL P22BAD             ;PRINT START ADDRESS
014426  004767  000366                                                                  JSR     PC,P22BAD

014432                                          CALL    TAB             ;PRINT TAB
014432  004767  177726                                                                  JSR     PC,TAB

014436                                          LET R2 :- END           ;FETCH LAST 4K BANK
014436  016702  001524                                                                  MOV     END,R2

014442                                          LET R1 := #17776        ;LAST ADDRESS IN 4K BANK
014442  012701  017776                                                                  MOV     #17776,R1

014446                                          CALL P22BAD
014446  004767  000346                                                                  JSR     PC,P22BAD

014452                                  ELSE
014452  000442                                                                          BR      50003$
014454                                                                                  50002$:

014454                                          LET R1 := START         ;FETCH BEGINNING ADDRESS
014454  016701  001516                                                                  MOV     START,R1

014460                                          IF R1 GE #0 THEN
014460  005701                                                                          TST     R1
014462  002405                                                                          BLT     50004$

014464                                                  TYPOCS  #0,,4           ;TYPE UPPER BIT AS ZERO
014464  012746  000000                  MOV     #0,-(SP)        ;;SAVE #0 FOR TYPEOUT
014470  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
014472     004                          .BYTE   4               ;;TYPE 4 DIGIT(S)
014473     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS

014474                                          ELSE
014474  000404                                                                          BR      50005$
014476                                                                                  50004$:

014476                                                  TYPOCS  #1,,4           ;TYPE UPPER BIT AS ONE
014476  012746  000001                  MOV     #1,-(SP)        ;;SAVE #1 FOR TYPEOUT
014502  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
014504     004                          .BYTE   4               ;;TYPF 4 DIGIT(S)
```

```
 014505     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS

 014506                                  ENDIF
 014506                                                                      50005$:

 014506                                  TYPOCS  R1,,5,X         ;PRINT STARTING ADDRESS
 014506  010146                  MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
 014510  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
 014512     005                  .BYTE   5               ;;TYPE 5 DIGIT(S)
 014513     001                  .BYTE   1               ;;TYPE LEADING ZEROS

 014514                                  CALL    TAB             ;TAB
 014514  004767  177644                                                      JSR     PC,TAB

 014520                                  LET R1 := END           ;FETCH LAST ADDRESS
 014520  016701  001442                                                      MOV     END,R1

 014524                                  IF R1 GE #0 THEN
 014524  005701                                                              TST     R1
 014526  002405                                                              BLT     50006$

 014530                                          TYPOCS  #0,,4           ;TYPE UPPER BIT AS ZERO
 014530  012746  000000          MOV     #0,-(SP)        ;;SAVE #0 FOR TYPEOUT
 014534  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
 014536     004                  .BYTE   4               ;;TYPE 4 DIGIT(S)
 014537     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS

 014540                                  ELSE
 014540  000404                                                              BR      50007$
 014542                                                                      50006$:

 014542                                          TYPOCS  #1,,4           ;TYPE UPPER BIT AS ONE
 014542  012746  000001          MOV     #1,-(SP)        ;;SAVE #1 FOR TYPEOUT
 014546  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
 014550     004                  .BYTE   4               ;;TYPE 4 DIGIT(S)
 014551     000                  .BYTE   0               ;;SUPPRESS LEADING ZEROS

 014552                                  ENDIF
 014552                                                                      50007$:

 014552                                  TYPOCS  R1,,5,X         ;PRINT LAST ADDRESS
 014552  010146                  MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
 014554  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
 014556     005                  .BYTE   5               ;;TYPE 5 DIGIT(S)
 014557     001                  .BYTE   1               ;;TYPE LEADING ZEROS

 014560                          ENDIF
 014560                                                                      50003$:

 014560                          IF CONTROL LT #0 THEN           ;IS IT NON-VOLATILE
 014560  005767  001374                                                      TST     CONTROL
 014564  002022                                                              BGE     50010$


                                         ;FETCH LOCATION OF NON VOLATILE MESSAGE
 014566                                  LET R1 := PC + #NVMES-.
 014566  010701                                                              MOV     PC,R1
 014570  062701  000670                                                      ADD     #NVMES-.,R1
```

```

                                                ;LOAD ADDRESS INTO TRAP INSTRUCTION
014574                                          LET 5$ := R1
014574  010167  000002                                                                  MOV     R1,5$

                                                ;TELL OPERATOR NON VOLATILE
014600  104401                                  TYPE
014602  015460                          5$:     .WORD   NVMES

014604                                          IF START EQ #0 THEN             ;ARE WE IN PAGE 0
014604  005767  001366                                                                  TST     START
014610  001007                                                                          BNE     50011$
                                                                                ;YES

014612                                                  IF #4000 SETIN CONTROL THEN     ;WAS THERE TRAP PAGE
014612  032767  004000  001340                                                          BIT     #4000,CONTROL
014620  001403                                                                          BEQ     50012$
                                                                                        ;CHECKSUM ERROR
                                                                                        ;YES

                                                                ;INDICATE PRINT ERROR MESSAGE
014622                                                          LET CONTROL := CONTROL SET.BY #40       ;
014622  052767  000040  001330                                                          BIS     #40,CONTROL

014630                                                  ENDIF
014630                                                                                  50012$:

014630                                          ENDIF
014630                                                                                  50011$:

014630                                  ELSE
014630  000407                                                                          BR      50013$
014632                                                                                  50010$:

                                                ;FETCH LOCATION OF VOLATILE MESSAGE
014632                                          LET R1 := PC + #VMES-.
014632  010701                                                                          MOV     PC,R1
014634  062701  000647                                                                  ADD     #VMES-.,R1

                                                ;LOAD ADDRESS INTO TRAP INSTRUCTION
014640                                          LET 6$ := R1
014640  010167  000002                                                                  MOV     R1,6$

                                                ;TELL OPERATOR VOLATILE
014644  104401                                  TYPE
014646  015503                          6$:     .WORD   VMES

014650                                  ENDIF
014650                                                                                  50013$:

014650                          ENDRTN
014650                                                                                  50000$:
014650                                                                                  50001$:
014650  000207                                                                          RTS     PC
```

```
                                        .SBTTL  PARITY CATCHER

                                ;THIS ROUTINE CATCHES PARITY ERRORS AND STORES THEN IN PARCNT

 014652                         IROUTINE PARTRP
 014652                                                                              PARTRP:

 014652                                 LET PARCNT := PARCNT + #1       ;INDICATE PARITY ERROR
 014652  005267  001314                                                              INC     PARCNT

                                        ;DISABLE PARITY
 014656                                 LET TRCATCH := #240             ;IF CSR DOES NOT EXIST THEN JUST RTI
 014656  012767  000240  001500                                                      MOV     #240,TRCATCH

 014664                                 LET R2 := #172100               ;FIRST CSR ADDRESS
 014664  012702  172100                                                              MOV     #172100,R2

 014670                                 INCR R1 FROM #0 TO #16. BY #1   ;CLEAR ALL CSRS
 014670  005001                                                                      CLR     R1
 014672  000401                                                                      BR      50002$
 014674                                                                              50003$:
 014674  005201                                                                      INC     R1
 014676                                                                              50002$:
 014676  020127  000020                                                              CMP     R1,#16.
 014702  003002                                                                      BGT     50004$

 014704                                         LET (R2)+ := #0
 014704  005022                                                                      CLR     (R2)+

 014706                                 ENDINC
 014706  000772                                                                      BR      50003$
 014710                                                                              50004$:

 014710                                 LET TRCATCH := #0               ;RESUME HALT IF TRAP
 014710  005067  001450                                                              CLR     TRCATCH

 014714                                 IF #125252 EQ -(R0) THEN        ;RECHECK ADDRESS WITH PARITY OFF
 014714  022740  125252                                                              CMP     #125252,-(R0)
 014720  001003                                                                      BNE     50005$

 014722                                         LET NONVOL := NONVOL + #1       ;WE ARE NON-VOLATILE
 014722  005267  001234                                                              INC     NONVOL

 014726                                 ELSE
 014726  000402                                                                      BR      50006$
 014730                                                                              50005$:

 014730                                         LET VOL := VOL + #1             ;WE ARE VOLATILE
 014730  005267  001230                                                              INC     VOL

 014734                                 ENDIF
 014734                                                                              50006$:

 014734                                 LET CONTROL :- CONTROL SET.BY #2        ;INDICATE PARITY ERROR WAS
 014734  052767  000002  001216                                                      BIS     #2,CONTROL
                                                                                ;SERVICED

```

```
   3004 014742                                  LET R0 := R0 + #2                             ;RESTORE R0 TO WHAT IT WAS AFTER
        014742  062700  000002                                                                          ADD     #2,R0
   3005                                                                                         ;TRAP
   3006
   3007 014746                                  IF PARCNT LT #10. THEN  ;LEAVE PARITY DISABLE WHEN TO MANY ERRORS
        014746  026727  001220  000012                                                                  CMP     PARCNT,#10.
        014754  002020                                                                                  BGE     50007$
   3008
   3009                                                 ;ENABLE PARITY ERRORS THROUGH THE CSRS
   3010 014756                                          LET TRCATCH := #240     ;IF CSR DOES NOT EXIST THEN JUST RTI
        014756  012767  000240  001400                                                                  MOV     #240,TRCATCH
   3011
   3012 014764                                          LET R2 := #172100               ;FIRST CSR ADDRESS
        014764  012702  172100                                                                          MOV     #172100,R2
   3013
   3014 014770                                          INCR R1 FROM #0 TO #16. BY #1   ;CLEAR ALL CSRS
        014770  005001                                                                                  CLR     R1
        014772  000401                                                                                  BR      50010$
        014774                                                                                          50011$:
        014774  005201                                                                                  INC     R1
        014776                                                                                          50010$:
        014776  020127  000020                                                                          CMP     R1,#16.
        015002  003003                                                                                  BGT     50012$
   3015
   3016 015004                                                  LET (R2)+ := #1
        015004  012722  000001                                                                          MOV     #1,(R2)+
   3017
   3018 015010                                          ENDINC
        015010  000771                                                                                  BR      50011$
        015012                                                                                          50012$:
   3019
   3020 015012                                          LET TRCATCH := #0               ;RESUME HALT IF TRAP
        015012  005067  001346                                                                          CLR     TRCATCH
   3021
   3022 015016                                  ENDIF
        015016                                                                                          50007$:
   3023
   3024 015016                          ENDRTI
        015016                                                                                          50000$:
        015016                                                                                          50001$:
        015016  000002                                                                                  RTI
```

```
                                                 .SBTTL  PRINT CONTIGUOUS 22 BIT ADDRESS ROUTINE

                                                 ;THIS ROUTINE TAKES THE ADDRESS STORED IN R1 AND THE MEMORY ADDRESS IN
                                                 ;R2 AND CREATS A 22 BIT CONTIGUOUS ADDRESS WITH ADDRESS BITS 14 - 0
                                                 ;IN R1 AND BITS 21 - 15 IN R2

 015020                                  ROUTINE P22BAD
 015020                                                                                    P22BAD:

 015020                                          PUSH <R1,R2,R2>               ;USE STACK AS SCRATCH PAD AREA
 015020  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 015022  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 015024  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK

 015026                                          LET (SP) := (SP) SHIFT -2     ;PUT BITS 8 & 7 INTO 6 & 5
 015026  006216                                                                       ASR     (SP)
 015030  006216                                                                       ASR     (SP)

 015032                                          LET (SP) := SWAP (SP)         ;PUT BITS 6 & 5 INTO 14 & 13
 015032  000316                                                                       SWAB    (SP)

 015034                                          LET (SP) := (SP) CLR.BY #117777 ;CLEAR OUT UN-IMPORTANT BITS
 015034  042716  117777                                                               BIC     #117777,(SP)

 015040                                          LET R1 := R1 CLR.BY #160000   ;CLEAR OUT MEMORY MAN. POINTERS
 015040  042701  160000                                                               BIC     #160000,R1

 015044                                          LET R1 := R1 SET.BY (SP)+     ;TRANSFER BITS 8 & 7 IN R2 INTO BITS
 015044  052601                                                                       BIS     (SP)+,R1
                                                                               ;14 & 13 IN R1

 015046                                          LET R2 := R2 SHIFT -1   ;FETCH ADDRESS BITS & PUT BIT 8 INTO
 015046  006202                                                                       ASR     R2
                                                                               ;LOWER BYTE

 015050                                          LET R2 := SWAP R2       ;NOW ADDRESS BITS 21-15 ARE IN LSB
 015050  000302                                                                       SWAB    R2

 015052                                          LET R2 := R2 CLR.BY #177600   ;CLEAR OUT ALL OTHER BITS
 015052  042702  177600                                                               BIC     #177600,R2

 015056                                          TYPOCS  R2,,4           ;PRINT ADDRESS BITS 21 - 15
 015056  010246                                  MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
 015060  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
 015062     004                                  .BYTE   4               ;;TYPE 4 DIGIT(S)
 015063     000                                  .BYTE   0               ;;SUPPRESS LEADING ZEROS

 015064                                          TYPOCS  R1,,5,X         ;PRINT ADDRESS BITS 14 - 0
 015064  010146                                  MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
 015066  104403                                  TYPOS                   ;;GO TYPE--OCTAL ASCII
 015070     005                                  .BYTE   5               ;;TYPE 5 DIGIT(S)
 015071     001                                  .BYTE   1               ;;TYPE LEADING ZEROS

 015072                                          POP <R2,R1>
 015072  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
 015074  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1

```

```
   3060 015076                      ENDRTN
        015076                                                          50000$:
        015076                                                          50001$:
        015076  000207                                                  RTS     PC
```

```
3062                                          .SBTTL  DEFINITIONS
3063
3064 015100      040      117      122  MORE:    .ASCIZ  ' OR MORE'
3065 015111      124      122      101  TRAPER: .ASCII  'TRAP LOCATIONS (ADDRESS 0 - 376) HAVE CHECK SUM ERROR'
3066 015176      040      111      116           .ASCIZ  ' IN NON - VOLATILE MEMORY'<12><12><15>
3067 015233      012      012      015  ERR:     .ASCII  <12><12><15>'CHECK SUM ERROR. THE MEMORY THE PROGRAM IS'
3068 015310      040      122      105           .ASCIZ  ' RESIDENT'<12><15>'IN HAS VOLATILE LOCATIONS'<12><15>
3069 015357      012      012      015  TAF:     .ASCIZ  <12><12><15>'THIS CONCLUDES THE NON-VOLATILE DATA RETENTION TEST'<12><15>
3070 015450      040      040      040  TAB1:    .ASCIZ  '       '
3071 015460      040      040      040  NVMES:   .ASCIZ  '      NON VOLATILE'
3072 015503      040      040      040  VMES:    .ASCIZ  '      VOLATILE     '
3073 015526      012      012      015  HEAD:    .ASCII  <12><12><15>
3074 015531      040      040      040           .ASCII  '                                MEMORY MAP'<12><15>
3075 015604      040      040      040           .ASCII  '                                ----------'<15><12><12>
3076 015660      123      124      101           .ASCII  'START ADR.      END ADR.      MEMORY TYPE     WORD ERROR      '
3077 015756      120      101      122           .ASCIZ  'PARITY ERROR'<12><12><15>
3078 015776      120      114      105  PWRDWN:  .ASCII  'PLEASE POWER DOWN THIS SYSTEM. AFTER 2 MINUTES BUT'
3079 016060      040      116      117           .ASCII  ' NO LONGER'<12><15>'THAN 100 HOURS,'
3080 016113      040      105      130           .ASCIZ  ' EXECUTE THE RESTART HELP FILE.'<12><15>
3081 016155      012      015      000  CRLF1:   .ASCIZ  <12><15>
3082                                             .EVEN
3083 016160  000000                     CONTRO:  .WORD   0
3084 016162  000000                     NONVOL:  .WORD   0
3085 016164  000000                     VOL:     .WORD   0
3086 016166  000000                     END:     .WORD   0
3087 016170  000000                     PPOINT:  .WORD   0
3088 016172  000000                     PARCNT:  .WORD   0
3089 016174  000000                     ENDADD:  .WORD   0
3090 016176  000000                     START:   .WORD   0
3091 016200  000000                     L4KST:   .WORD   0
```

```
3093                                    ;THIS ROUTINE BUMPS THE WRITE POINTER 4K IF A WRITE TO A ROM MODULE CAUSES A
3094                                    ;TRAP
3095
3096 016202                             IROUTINE ROMMD
     016202                                                                           ROMMD:
3097
3098 016202                                       IF #1 SETIN CONTROL THEN             ;ARE WE IN A MAPPED SYSTEM
     016202  032767  000001  177750                                                           BIT     #1,CONTROL
     016210  001403                                                                           BEQ     50002$
3099                                                                                 ;YES
3100
3101 016212                                                LET     R0 := #40000      ;BUMP POINTER TO INDICATE
     016212  012700  040000                                                                   MOV     #40000,R0
3102                                                                                 ;WF HAVE REACHED THE END
3103                                                                                 ;OF THIS 4K CHUNK OF MEMORY
3104
3105 016216                                       ELSE
     016216  000403                                                                           BR      50003$
     016220                                                                                   50002$:
3106
3107 016220                                                INLINE  <TST -(R0)>       ;BUMP POINTER BACK TO WHAT IT
     016220  005740                                                                           TST -(R0)
3108                                                                                 ;WAS BEFORE TRAP
3109
3110 016222                                                LET     R0 := R0 + #40000 ;BUMP TO NEXT 4K BOUNDARY
     016222  062700  040000                                                                   ADD     #40000,R0
3111
3112 016226                                       ENDIF
     016226                                                                                   50003$:
3113
3114 016226                             ENDRTI
     016226                                                                                   50000$:
     016226                                                                                   50001$:
     016226  000002                                                                           RTI
```

```
                                     ;THIS ROUTINE SETS UP TRAP CATCHERS IN ALL TRAP LOCATIONS
                                     ;FROM 0 TO 400

 016230                              ROUTINE TRAPCT
 016230                                                                              TRAPCT:

 016230                                    PUSH    R0
 016230  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK

 016232                                    IF #1 SETIN CONTROL THEN            ;ARE WE IN MEMORY MANAGEMENT
 016232  032767  000001  177720                                                      BIT     #1,CONTROL
 016240  001407                                                                      BEQ     50002$
                                                                               ;SYSTEM

 016242                                          PUSH    @#KIPAR1
 016242  013746  172342                    MOV   @#KIPAR1,-(SP)        ;;PUSH @#KIPAR1 ON STACK

 016246                                          LET     @#KIPAR1 := #0        ;YES, PAR1 WILL POINT TO LOWEST
 016246  005037  172342                                                              CLR     @#KIPAR1
                                                                               ;4K

 016252                                          LET     R0 := #20000          ;R0 POINTS TO PAR1 AND
 016252  012700  020000                                                              MOV     #20000,R0
                                                                               ;ADDRESS 0

 016256                                    ELSE
 016256  000401                                                                      BR      50003$
 016260                                                                              50002$:

 016260                                          LET     R0 := #0              ;START WITH ADDRESS 0
 016260  005000                                                                      CLR     R0

 016262                                    ENDIF
 016262                                                                              50003$:

 016262                                    LET     R1 := PC + #TRCATCH-.       ;FIND TRAP CATCHER ROUTINE
 016262  010701                                                                      MOV     PC,R1
 016264  062701  000100                                                              ADD     #TRCATCH-.,R1

 016270                                    LOOP
 016270                                                                              50004$:

 016270                                    EXIF    R0 EQ #400 OR RC EQ #20400
 016270  020027  000400                                                              CMP     R0,#400
 016274  001423                                                                      BEQ     50005$
 016276  020027  020400                                                              CMP     R0,#20400
 016302  001420                                                                      BEQ     50005$

 016304                                          IF R0 EQ #34 OR R0 EQ #20034 THEN   ;LOAD TRAP ROUTINE
 016304  020027  000034                                                              CMP     R0,#34
 016310  001403                                                                      BEQ     50006$
 016312  020027  020034                                                              CMP     R0,#20034
 016316  001006                                                                      BNE     50007$
 016320                                                                              50006$:

 016320                                                LET (R0) := PC          ;LOAD TRAP INSTRUCTION LOCATION
 016320  010710                                                                      MOV     PC,(R0)
```

```
016322                                                  LET (R0)+ := (R0)+ + #$TRAP-.   ;''             ''
016322  062720  001104                                                                  ADD     #$TRAP-..(R0)+
016326                                                  LET (R0)+ := #340       ;LOAD PSW
016326  012720  000340                                                                  MOV     #340,(R0)+
016332                                          ELSE
016332  000403                                                                          BR      50010$
016334                                                                                  50007$:
016334                                                  LET (R0)+ := R1         ;LOAD TRAP CATCHER
016334  010120                                                                          MOV     R1,(R0)+
016336                                                  LET (R0)+ := #340       ;LOAD PSW
016336  012720  000340                                                                  MOV     #340,(R0)+
016342                                          ENDIF
016342                                                                                  50010$:
016342                                  ENDLOOP
016342  000752                                                                          BR      50004$
016344                                                                                  50005$:
016344                                  IF #1 SETIN CONTROL THEN                ;ARE WE IN A MAPPED SYSTEM
016344  032767  000001  177606                                                          BIT     #1,CONTROL
016352  001402                                                                          BEQ     50011$
016354                                          POP     @#KIPAR1
016354  012637  172342                  MOV     (SP)+,@#KIPAR1          ;;POP STACK INTO @#KIPAR1
016360                                  ENDIF
016360                                                                                  50011$:
016360                                  POP     R0
016360  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
016362                          ENDRTN
016362                                                                                  50000$:
016362                                                                                  50001$:
016362  000207                                                                          RTS     PC
                                ;DO AN RTI WHENEVER IT TRAPS.
016364                          IROUTINE TRCATCH
016364                                                                                  TRCATCH:
016364                                  INLINE  <HALT>  ;FOR DEBUG PURPOSE REPLACE WITH A HALT INSTRUCTION
016364  000000                                                                          HALT
016366                          ENDRTI
016366                                                                                  50000$:
016366                                                                                  50001$:
016366  000002                                                                          RTI
```

```
                                                .LIST   MEB

                                                .$TYPE
 016370
 016370 105767 000313               $TYPE:     TSTB    $TPFLG1             ;;IS THERE A TERMINAL?
 016374 100002                                 BPL     1$                  ;;BR IF YES
 016376 000000                                 HALT                        ;;HALT HERE IF NO TERMINAL
 016400 000407                                 BR      3$                  ;;LEAVE
 016402 010046                      1$:        MOV     R0,-(SP)            ;;SAVE R0
 016404 017600 000002                          MOV     @2(SP),R0           ;;GET ADDRESS OF ASCIZ STRING
 016410 112046                      2$:        MOVB    (R0)+,-(SP)         ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 016412 001005                                 BNE     4$                  ;;BR IF IT ISN'T THE TERMINATOR
 016414 005726                                 TST     (SP)+               ;;IF TERMINATOR POP IT OFF THE STACK
 016416 012600                      60$:       MOV     (SP)+,R0            ;;RESTORE R0
 016420 062716 000002               3$:        ADD     #2,(SP)             ;;ADJUST RETURN PC
 016424 000002                                 RTI                         ;;RETURN
 016426 122716 000011               4$:        CMPB    #HT,(SP)            ;;BRANCH IF <HT>
 016432 001435                                 BEQ     8$
 016434 122716 000200                          CMPB    #CRLF,(SP)          ;;BRANCH IF NOT <CRLF>
 016440 001013                                 BNE     5$
 016442 005726                                 TST     (SP)+               ;;POP  <CR><LF> EQUIV
 016444 010746                                 MOV     PC,-(SP)
 016446 062716 000242                          ADD     #$CRLF1-.,(SP)
 016452 012667 000002                          MOV     (SP)+,20$
 016456 104401                                 TYPE                        ;;TYPE A CR AND LF
 016460 016710                      20$:       .WORD $CRLF1
 016462 105067 000202                          CLRB    $CHARCNT            ;;CLEAR CHARACTER COUNT
 016466 000750                                 BR      2$                  ;;GET NEXT CHARACTER
 016470 004767 000056               5$:        JSR     PC,$TYPEC           ;;GO TYPE THIS CHARACTER
 016474 126726 000206               6$:        CMPB    $FILLC,(SP)+        ;;IS IT TIME FOR FILLER CHARS.?
 016500 001343                                 BNE     2$                  ;;IF NO GO GET NEXT CHAR.
 016502 016746 000176                          MOV     $NULL,-(SP)         ;;GET # OF FILLER CHARS. NEEDED
 016506 105366 000001               7$:        DECB    1(SP)               ;;DOES A NULL NEED TO BE TYPED?
 016512 002770                                 BLT     6$                  ;;BR IF NO--GO POP THE NULL OFF OF STACK
 016514 004767 000032                          JSR     PC,$TYPEC           ;;GO TYPE A NULL
 016520 105367 000144                          DECB    $CHARCNT            ;;DO NOT COUNT AS A COUNT
 016524 000770                                 BR      7$                  ;;LOOP
 016526 112716 000040               8$:        MOVB    #' ,(SP)            ;;REPLACE TAB WITH SPACE
 016532 004767 000014               9$:        JSR     PC,$TYPEC           ;;TYPE A SPACE
 016536 132767 000007 000124                   BITB    #7,$CHARCNT         ;;BRANCH IF NOT AT
 016544 001372                                 BNE     9$                  ;;TAB STOP
 016546 005726                                 TST     (SP)+               ;;POP SPACE OFF STACK
 016550 000717                                 BR      2$                  ;;GET NEXT CHARACTER
 016552 105777 000116               $TYPEC:    TSTB    @$TKS1              ;;CHAR IN KYBD BUFFER?                  ;MJD001
 016556 100022                                 BPL     10$                 ;;BR IF NOT                             ;MJD001
 016560 017746 000112                          MOV     @$TKB1,-(SP)        ;;GET CHAR                              ;MJD001
 016564 042716 177600                          BIC     #177600,(SP)        ;;STRIP EXTRANEOUS BITS                 ;MJD001
 016570 122716 000023                          CMPB    #$XOFF,(SP)         ;;WAS CHAR XOFF                         ;MJD001
 016574 001012                                 BNE     102$                ;;BR IF NOT                             ;MJD001
 016576 105777 000072               101$:      TSTB    @$TKS1              ;;WAIT FOR CHAR                         ;MJD001
 016602 100375                                 BPL     101$                                                        ;MJD001
 016604 117716 000066                          MOVB    @$TKB1,(SP)         ;;GET CHAR                              ;MJD001
 016610 042716 177600                          BIC     #177600,(SP)        ;;STRIP IT                              ;MJD001
 016614 122716 000021                          CMPB    #$XON,(SP)          ;;WAS IT XON?                           ;MJD001
 016620 001366                                 BNE     101$                ;;BR IF NOT                             ;MJD001
 016622 005726                      102$:      TST     (SP)+               ;;FIX STACK                             ;MJD001
 016624 105777 000050               10$:       TSTB    @$TPS1              ;;WAIT UNTIL PRINTER IS READY
 016630 100375                                 BPL     10$                                                         ;MJD001
```

```
     016632  116677  000002  000042          MOVB    2(SP),@$TPB1       ;;LOAD CHAR TO BE TYPED INTO DATA REG.
     016640  122766  000015  000002          CMPB    #CR,2(SP)          ;;IS CHARACTER A CARRIAGE RETURN?
     016646  001003                          BNE     1$                 ;;BRANCH IF NO
     016650  105067  000014                  CLRB    $CHARCNT           ;;YES--CLEAR CHARACTER COUNT
     016654  000406                          BR      $TYPEX             ;;EXIT
     016656  122766  000012  000002  1$:     CMPB    #LF,2(SP)          ;;IS CHARACTER A LINE FEED?
     016664  001402                          BEQ     $TYPEX             ;;BRANCH IF YES
     016666  105227                          INCB    (PC)+              ;;COUNT THE CHARACTER
     016670  000000                  $CHARCNT:.WORD  0                  ;;CHARACTER COUNT STORAGE
     016672  000207                  $TYPEX: RTS     PC
     016674  177560                  .IIF EQ .-$TKS1,$TKS1:  .WORD   177560          ;;TTY KDB STATUS          ;MJD001
     016676  177562                  .IIF EQ .-$TKB1,$TKB1:  .WORD   177562          ;;TTY KBD BUFFER          ;MJD001
     016700  177564                  .IIF EQ .-$TPS1,$TPS1:  .WORD   177564          ;;TTY PRINTER STATUS REG. ADDRESS
     016702  177566                  .IIF EQ .-$TPB1,$TPB1:  .WORD   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
     016704     000                  .IIF EQ .-$NULL,$NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
     016705     002                  .IIF EQ .-$FILLS,$FILLS:        .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS RE
     016706     012                  .IIF EQ .-$FILLC,$FILLC:        .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE F
     016707     000                  .IIF EQ .-$TPFLG1,$TPFLG1:      .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>
     016710     015                  .IIF EQ .-$CRLF1,$CRLF1:        .ASCII  <15>            ;;CARRAIGE RETURN
     016711     012     000          .IIF EQ .-$LF1,$LF1:    .ASCIZ  <12>            ;;LINEFEED

     016714                                  .$TYPDEC
     016714  010046                          MOV     R0,-(SP)           ;;PUSH R0 ON STACK
     016716  010146                          MOV     R1,-(SP)           ;;PUSH R1 ON STACK
     016720  010246                          MOV     R2,-(SP)           ;;PUSH R2 ON STACK
     016722  010346                          MOV     R3,-(SP)           ;;PUSH R3 ON STACK
     016724  010446                          MOV     R4,-(SP)           ;;PUSH R4 ON STACK
     016726  010546                          MOV     R5,-(SP)           ;;PUSH R5 ON STACK
     016730  012746  020200                  MOV     #20200,-(SP)       ;;SET BLANK SWITCH AND SIGN
     016734  016605  000022                  MOV     22(SP),R5          ;;GET THE INPUT NUMBER
     016740  100004                          BPL     1$                 ;;BR IF INPUT IS POS.
     016742  005405                          NEG     R5                 ;;MAKE THE BINARY NUMBER POS.
     016744  112766  000055  000001          MOVB    #'-,1(SP)          ;;MAKE THE ASCII NUMBER NEG.
     016752  005000                  1$:     CLR     R0                 ;;ZERO THE CONSTANTS INDEX
     016754  010703                          MOV     PC,R3              ;;FETCH PC FOR ADDRESSING
     016756  062703  000200                  ADD     #$DBLK-.,R3        ;;SETUP THE OUTPUT POINTER - R3 POINTS TO $DBLK
     016762  112723  000040                  MOVB    #' ,(R3)+          ;;SET THE FIRST CHARACTER TO A BLANK
     016766  005002                  2$:     CLR     R2                 ;;CLEAR THE BCD NUMBER
     016770  010704                          MOV     PC,R4              ;;USE PC FOR LOCATING TABLE
     016772  062704  000154                  ADD     #$DTBL-.,R4        ;;R4 NOW POINTS TO TABLE
     016776  060004                          ADD     R0,R4              ;;INDEX INTO TABLE
     017000  011401                          MOV     (R4),R1            ;;GET THE CONSTANT
     017002  160105                  3$:     SUB     R1,R5              ;;FORM THIS BCD DIGIT
     017004  002402                          BLT     4$                 ;;BR IF DONE
     017006  005202                          INC     R2                 ;;INCREASE THE BCD DIGIT BY 1
     017010  000774                          BR      3$
     017012  060105                  4$:     ADD     R1,R5              ;;ADD BACK THE CONSTANT
     017014  005702                          TST     R2                 ;;CHECK IF BCD DIGIT=0
     017016  001002                          BNE     5$                 ;;FALL THROUGH IF 0
     017020  105716                          TSTB    (SP)               ;;STILL DOING LEADING 0'S?
     017022  100407                          BMI     7$                 ;;BR IF YES
     017024  106316                  5$:     ASLB    (SP)               ;;MSD?
     017026  103003                          BCC     6$                 ;;BR IF NO
     017030  116663  000001  177777          MOVB    1(SP),-1(R3)       ;;YES--SET THE SIGN
     017036  052702  000060          6$:     BIS     #'0,R2             ;;MAKE THE BCD DIGIT ASCII
     017042  052702  000040          7$:     BIS     #' ,R2             ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
     017046  110223                          MOVB    R2,(R3)+           ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
```

```
        017050  005720                              TST     (R0)+               ;;JUST INCREMENTING
        017052  020027  000010                      CMP     R0,#10              ;;CHECK THE TABLE INDEX
        017056  002743                              BLT     2$                  ;;GO DO THE NEXT DIGIT
        017060  003002                              BGT     8$                  ;;GO TO EXIT
        017062  010502                              MOV     R5,R2               ;;GET THE LSD
        017064  000764                              BR      6$                  ;;GO CHANGE TO ASCII
        017066  105726                      8$:     TSTB    (SP)+               ;;WAS THE LSD THE FIRST NON-ZERO?
        017070  100003                              BPL     9$                  ;;BR IF NO
        017072  116663  177777  177776              MOVB    -1(SP),-2(R3)       ;;YES--SET THE SIGN FOR TYPING
        017100  105013                      9$:     CLRB    (R3)                ;;SET THE TERMINATOR
        017102  012605                              MOV     (SP)+,R5            ;;POP STACK INTO R5
        017104  012604                              MOV     (SP)+,R4            ;;POP STACK INTO R4
        017106  012603                              MOV     (SP)+,R3            ;;POP STACK INTO R3
        017110  012602                              MOV     (SP)+,R2            ;;POP STACK INTO R2
        017112  012601                              MOV     (SP)+,R1            ;;POP STACK INTO R1
        017114  012600                              MOV     (SP)+,R0            ;;POP STACK INTO R0
        017116  010746                              MOV     PC,-(SP)            ;;FETCH ADDRESS OF NUMBER TO BE TYPED USING
        017120  062716  000036                      ADD     #$DBLK-.,(SP)       ;;STACK NOW CONTAINS ADDRESS OF NUMBER TO BE
        017124  012667  000002                      MOV     (SP)+,10$           ;;SET UP ADDRESS FOR TYPE COMMAND
        017130  104401                              TYPE                        ;;NOW TYPE THE NUMBER
        017132  017156                      10$:    .WORD   $DBLK
        017134  016666  000002  000004              MOV     2(SP),4(SP)         ;;ADJUST THE STACK
        017142  012616                              MOV     (SP)+,(SP)
        017144  000002                              RTI                         ;;RETURN TO USER
        017146  023420                      $DTBL:  10000.
        017150  001750                              1000.
        017152  000144                              100.
        017154  000012                              10.
   3187
   3188 017166                                      .$TYPOCT
        017166  017646  000000              $TYPOS: MOV     @(SP),-(SP)         ;;PICKUP THE MODE
        017172  116667  000001  000223              MOVB    1(SP),$0FILL        ;;LOAD ZERO FILL SWITCH
        017200  112667  000221                      MOVB    (SP)+,$OMODE+1      ;;NUMBER OF DIGITS TO TYPE
        017204  062716  000002                      ADD     #2,(SP)             ;;ADJUST RETURN ADDRESS
        017210  000406                              BR      $TYPON
        017212  112767  000001  000203      $TYPOC: MOVB    #1,$0FILL           ;;SET THE ZERO FILL SWITCH
        017220  112767  000006  000177              MOVB    #6,$OMODE+1         ;;SET FOR SIX(6) DIGITS
        017226  112767  000005  000166      $TYPON: MOVB    #5,$OCNT            ;;SET THE ITERATION COUNT
        017234  010346                              MOV     R3,-(SP)            ;;SAVE R3
        017236  010446                              MOV     R4,-(SP)            ;;SAVE R4
        017240  010546                              MOV     R5,-(SP)            ;;SAVE R5
        017242  116704  000157                      MOVB    $OMODE+1,R4         ;;GET THE NUMBER OF DIGITS TO TYPE
        017246  005404                              NEG     R4
        017250  062704  000006                      ADD     #6,R4               ;;SUBTRACT IT FOR MAX. ALLOWED
        017254  110467  000144                      MOVB    R4,$OMODE           ;;SAVE IT FOR USE
        017260  116704  000137                      MOVB    $0FILL,R4           ;;GET THE ZERO FILL SWITCH
        017264  016605  000012                      MOV     12(SP),R5           ;;PICKUP THE INPUT NUMBER
        017270  005003                              CLR     R3                  ;;CLEAR THE OUTPUT WORD
        017272  006105                      1$:     ROL     R5                  ;;ROTATE MSB INTO "C"
        017274  000404                              BR      3$                  ;;GO DO MSB
        017276  006105                      2$:     ROL     R5                  ;;FORM THIS DIGIT
        017300  006105                              ROL     R5
        017302  006105                              ROL     R5
        017304  010503                              MOV     R5,R3
        017306  006103                      3$:     ROL     R3                  ;;GET LSB OF THIS DIGIT
        017310  105367  000110                      DECB    $OMODE              ;;TYPE THIS DIGIT?
        017314  100023                              BPL     7$                  ;;BR IF NO
```

```
     017316  042703  177770                    BIC     #177770,R3       ;;GET RID OF JUNK
     017322  001002                            BNE     4$               ;;TEST FOR 0
     017324  005704                            TST     R4               ;;SUPPRESS THIS 0?
     017326  001403                            BEQ     5$               ;;BR IF YES
     017330  005204                   4$:      INC     R4               ;;DON'T SUPPRESS ANYMORE 0'S
     017332  052703  000060                    BIS     #'0,R3           ;;MAKE THIS DIGIT ASCII
     017336  052703  000040           5$:      BIS     #' ,R3           ;;MAKE ASCII IF NOT ALREADY
     017342  110367  000052                    MOVB    R3,8$            ;;SAVE FOR TYPING
     017346  010746                            MOV     PC,-(SP)         ;;FETCH ADDRESS OF NUMBER TO BE TYPED USING
     017350  062716  000050                    ADD     #8$-.,(SP)       ;;STACK NOW CONTAINS ADDRESS OF NUMBER TO BE
     017354  012667  000002                    MOV     (SP)+,10$        ;;SET UP ADDRESS FOR TYPE COMMAND
     017360  104401                            TYPE                     ;;GO TYPE THIS DIGIT
     017362  017420                   10$:     .WORD   8$               ;;ADDRESS OF CHARACTER TO BE PRINTED
     017364  105367  000032           7$:      DECB    $OCNT            ;;COUNT BY 1
     017370  003342                            BGT     2$               ;;BR IF MORE TO DO
     017372  002402                            BLT     6$               ;;BR IF DONE
     017374  005204                            INC     R4               ;;INSURE LAST DIGIT ISN'T A BLANK
     017376  000737                            BR      2$               ;;GO DO THE LAST DIGIT
     017400  012605                   6$:      MOV     (SP)+,R5         ;;RESTORE R5
     017402  012604                            MOV     (SP)+,R4         ;;RESTORE R4
     017404  012603                            MOV     (SP)+,R3         ;;RESTORE R3
     017406  016666  000002  000004            MOV     2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
     017414  012616                            MOV     (SP)+,(SP)
     017416  000002                            RTI                      ;;RETURN
     017420     000                   8$:      .BYTE   0                ;;STORAGE FOR ASCII DIGIT
     017421     000                            .BYTE   0                ;;TERMINATOR FOR TYPE ROUTINE
     017422     000                   $OCNT:   .BYTE   0                ;;OCTAL DIGIT COUNTER
     017423     000                   $OFILL:  .BYTE   0                ;;ZERO FILL SWITCH
     017424  000000                   $OMODE:  .WORD   0                ;;NUMBER OF DIGITS TO TYPE

     017426                                    .$TRAP
     017426  010046                   $TRAP:   MOV     R0,-(SP)         ;;SAVE R0
     017430  010146                            MOV     R1,-(SP)         ;;SAVE R1
     017432  016600  000004                    MOV     4(SP),R0         ;;GET TRAP ADDRESS
     017436  005740                            TST     -(R0)            ;;BACKUP BY 2
     017440  111000                            MOVB    (R0),R0          ;;GET RIGHT BYTE OF TRAP
     017442  006300                            ASL     R0               ;;POSITION FOR INDEXING
     017444  010701                            MOV     PC,R1            ;;FETCH THIS PROGRAM POINTER FOR PIC
     017446  062701  000026                    ADD     #$TRPAD-.,R1     ;;POINT TO TABLE
     017452  060001                            ADD     R0,R1            ;;R1 NOW POINTS TO ROUTINE TRAP CALL WANTED
     017454  011100                            MOV     (R1),R0          ;;R0 CONTAINS ADDRESS FOR RTS
     017456  012601                            MOV     (SP)+,R1         ;;RESTORE R1
     017460  000200                            RTS     R0               ;;GO TO ROUTINE
     017462  011646                   $TRAP2:  MOV     (SP),-(SP)       ;;MOVE THE PC DOWN
     017464  016666  000004  000002            MOV     4(SP),2(SP)      ;;MOVE THE PSW DOWN
     017472  000002                            RTI                      ;;RESTORE THE PSW
     017474  017462                   $TRPAD:  .WORD   $TRAP2
     017476  016370                            $TYPE   ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
     017500  017212                            $TYPOC  ;;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
     017502  017166                            $TYPOS  ;;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
     017504  017226                            $TYPON  ;;CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
     017506  016714                            $TYPDS  ;;CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
     017510  000404                            $RDCHR  ;;CALL=RDCHR     TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
     017512  000534                            $RDLIN  ;;CALL=RDLIN     TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
     017514  000730                            $RDOCT  ;;CALL=RDOCT     TRAP+10(104410) READ AN OCTAL NUMBER FROM TTY
     017516  001070                            $RDDEC  ;;CALL=RDDEC     TRAP+11(104411) READ A DECIMAL NUMBER FROM TTY
```

```
   3192 017520  000000                  CSTRPG: .WORD   0
   3193 017522  000000                  CHKSUM: .WORD   0
   3194 017524                                  .BLKW   50
   3195 017644  000000                  SPINIT: .WORD   0
   3196
   3197 017646  000000                  ENDPR:  .WORD 0
   3198
   3199         000001                          .END
```

```
                                                                  I  8
CVMEMAO DIAGNOSTIC FOR NON-VOLA MACRO M1113  22-JUN-81 15:42  PAGE 43-5
SYMBOL TABLE                                                                                              SEQ 0099

ADDCHK  002074       HELLO   006302       Q4      011632       SW14  = 040000       $F$INC= 000210
ASSEMB= 000010       HELP    006174       RDCHR = 104406       SW15  = 100000       $F$LOO= 000200
BANKO   006404       HELPF   002226       RDDEC = 104411       SW2   = 000004       $F$NAM= 000160
BIT0  = 000001       HT    = 000011       RDLIN = 104407       SW3   = 000010       $F$NO = 000403
BIT00 = 000001       ILLADD  006236       RDOCT = 104410       SW4   = 000020       $F$OR = 000320
BIT01 = 000002       IOTVEC= 000020       RESHLP  007326       SW5   = 000040       $F$RTI= 000350
BIT02 = 000004       K       006470       RESVEC= 000010       SW6   = 000100       $F$RTN= 000300
BIT03 = 000010       KIPAR0= 172340       RHELP   010066       SW7   = 000200       $F$SEL= 000140
BIT04 = 000020       KIPAR1= 172342       ROMMD   016202       SW8   = 000400       $F$THE= 000330
BIT05 = 000040       KIPAR2= 172344       R6    =%000006       SW9   = 001000       $F$TRU= 000404
BIT06 = 000100       KIPAR3= 172346       R7    =%000007       SYMD  = 000000       $F$UNT= 000130
BIT07 = 000200       KIPAR4= 172350       SAVHI   010346       SYMS  = 000007       $F$WHI= 000120
BIT08 = 000400       KIPAR5= 172352       SAVLOW  010344       TAB     014364       $F$YES= 000402
BIT09 = 001000       KIPAR6= 172354       SCOPE = 000004       TAB1    015450       $HD   = 000003
BIT1  = 000002       KIPAR7= 172356       SIPAR0= 172240       TAF     015357       $HIOCT  001066
BIT10 = 002000       KIPDR0= 172300       SIPAR1= 172242       TBITVE= 000014       $IFLEV= 177777
BIT11 = 004000       KIPDR1= 172302       SIPAR2= 172244       TKVEC = 000060       $ISK0 = 000001
BIT12 = 010000       KIPDR2= 172304       SIPAR3= 172246       TPVEC = 000064       $ISK1 = 000001
BIT13 = 020000       KIPDR3= 172306       SIPAR4= 172250       TRAPCT  016230       $ISK2 = 000001
BIT14 = 040000       KIPDR4= 172310       SIPAR5= 172252       TRAPER  015111       $ISK3 = 000001
BIT15 = 100000       KIPDR5= 172312       SIPAR6= 172254       TRAPVE= 000034       $KTNEX  001622
BIT2  = 000004       KIPDR6= 172314       SIPAR7= 172256       TRCATC  016364       $KTOUT  001560
BIT3  = 000010       KIPDR7= 172316       SIPDR0= 172200       TRTVEC= 000014       $KT11   001332
BIT4  = 000020       LF    = 000012       SIPDR1= 172202       TYPDS = 104405       $LF     000672
BIT5  = 000040       L4KST   016200       SIPDR2= 172204       TYPE  = 104401       $LF1    016711
BIT6  = 000100       MAPPED  006506       SIPDR3= 172206       TYPOC = 104402       $LOCTA= 177777
BIT7  = 000200       MMR3    007713       SIPDR4= 172210       TYPON = 104404       $LSTAD  001736
BIT8  = 000400       MMVEC = 000250       SIPDR5= 172212       TYPOS = 104403       $LSTBK  001740
BIT9  = 001000       MORE    015100       SIPDR6= 172214       UNMAP   006572       $LSTIN= 000001
BPTVEC= 000014       NOBLK   007237       SIPDR7= 172216       VMES    015503       $LSTTA= 000001
BYTECO= 000403       NONVOL  016162       SPINIT  017644       VOL     016164       $MAP    001424
CHKSUM  017522       NOT4K   010152       SR0   = 177572       WHERE   006664       $MAPRG  001430
CONTRO  016160       NVMES   015460       SR1   = 177574       WHERE1  007006       $MMOUT  001602
CR    = 000015       OUTMEM  010224       SR2   = 177576       WHOLE   007124       $MNEW   000717
CRLF  = 000200       PARCNT  016172       SR3   = 172516       $BGNLE= 177777       $MSWR   000706
CRLF1   016155       PARTRP  014652       STACK = 001100       $BRJMP= 177777       $NESTL= 177777
CRLF2   014340       PAR0    007506       START   016176       $CHARC  016670       $NOMAP  001430
CR1     007477       PAR7    007536       START1  010350       $CNTLG  000701       $NSK0 = 000350
CSTRPG  017520       PIRQ  = 177772       STKLMT= 177774       $CNTLU  000674       $NSK1 = 000110
DDISP = 177570       PIRQVE= 000240       STLOOP  012606       $CORE   001630       $NSK2 = 000110
DIAGMC= 000000       PPOINT  016170       SWREG   000176       $CRLF   000671       $NSK3 = 000110
DIF1  = 006014       PRINT   014410       SW0   = 000001       $CRLF1  016710       $NSK4 = 000110
DISPRE  000174       PR0   = 000000       SW00  = 000001       $CROUT  001660       $NULL   016704
DSWR  = 177570       PR1   = 000040       SW01  = 000002       $DBLK   017156       $OCNT   017422
ECHO    010342       PR2   = 000100       SW02  = 000004       $DTBL   017146       $OMODE  017424
EDLOOP  014210       PR3   = 000140       SW03  = 000010       $ERFLG= 000400       $QUES   000670
EMTVEC= 000030       PR4   = 000200       SW04  = 000020       $FILLC  016706       $RDCHR  000404
ENABLE  007747       PR5   = 000240       SW05  = 000040       $FILLS  016705       $RDDEC  001070
END     016166       PR6   = 000300       SW06  = 000100       $F$AND= 000310       $RDLIN  000534
ENDADD  016174       PR7   = 000340       SW07  = 000200       $F$BAD= 000401       $RDOCT  000730
ENDPR   017646       PS    = 177776       SW08  = 000400       $F$BLA= 000170       $RDSZ = 000026
ERR     015233       PSW   = 177776       SW09  = 001000       $F$CAS= 000150       $SAVE = 000001
ERROR = 104000       PWRDWN  015776       SW1   = 000002       $F$DEC= 000220       $SAVLE= 177777
ERRVEC= 000004       PWRUP   012072       SW10  = 002000       $F$DO = 000340       $SELLE= 177777
FETADD  001762       PWRVEC= 000024       SW11  = 004000       $F$FAL= 000405       $SETUP= 000004
FETCH   001742       P22BAD  015020       SW12  = 010000       $F$GOO= 000400       $SIZE   001246
HEAD    015526       Q1      011034       SW13  = 020000       $F$IF = 000110       $SIZEX  001664
```

```
$SSKO = 050005       $TPB1    016702      $TSK5 = 050037      $$ARGC= 000000      $$RETU= 000000
$STOP   001426       $TPFLG   016707      $TTYIN  000642      $$BYTE= 000403      $$RTN1= 050000
$STUP = 177777       $TPS1    016700      $TYPDS  016714      $$CASE= 000000      $$RTN2= 050001
$SWR  = 160000       $TRAP    017426      $TYPE   016370      $$DST = 000000      $$SRC = 000000
$TAGLE= 177777       $TRAP2   017462      $TYPEC  016552      $$ELOC= 000402      $$TGSV= 000000
$TAGNU= 050002       $TRP   = 000012      $TYPEX  016672      $$ERFL= 000000      $$TGS1= 000000
$TEMP = 000350       $TRPAD   017474      $TYPOC  017212      $$FLAG= 000001      $$TGS2= 000000
$TKB    000402       $TSKO  = 050011      $TYPON  017226      $$FROM= 000000      $$TO  = 000000
$TKB1   016676       $TSK1  = 050005      $TYPOS  017166      $$INH = 000402      $$TOTL= 000000
$TKS    000400       $TSK2  = 050010      $U    = 000403      $$LOC = 016352      $$$TAG= 050000
$TKS1   016674       $TSK3  = 050121      $XOFF = 000023      $$LOCN= 000000      $OFILL  017423
$TN   = 000001       $TSK4  = 050061      $XON  = 000021      $$REG = 177777

. ABS.  017650     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  60528 WORDS  ( 237 PAGES)
DYNAMIC MEMORY:  21870 WORDS  ( 84 PAGES)
ELAPSED TIME:  00:22:02
```