RK611

DISKLESS CONT. NO. 3

MD-11-DZR6C-A

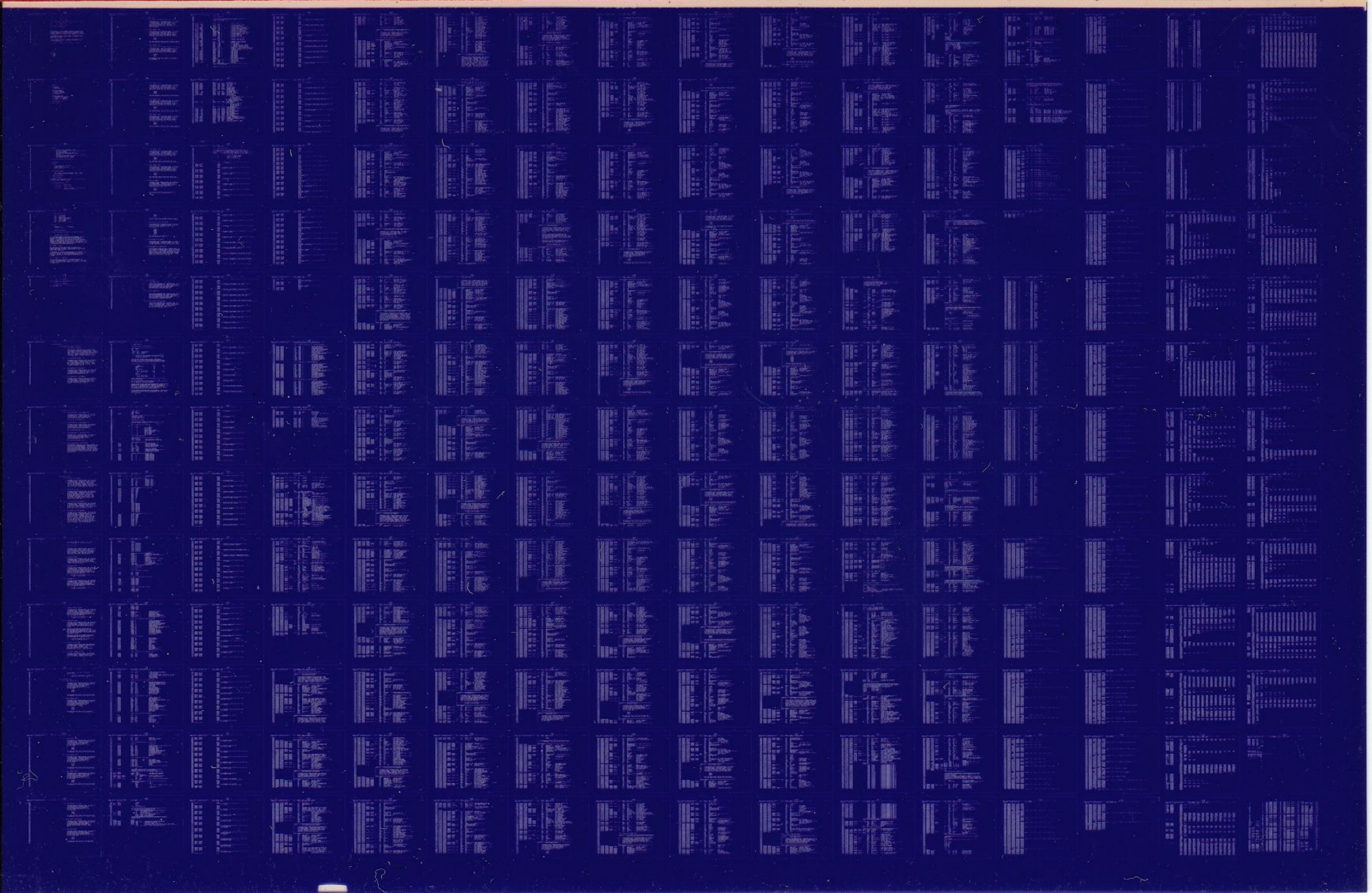
EP-DZR6C-A-DL-A

FICHE 1 OF 1

DEC 1976

digital

MADE IN USA



```
                        .REM    %

                                IDENTIFICATION

PRODUCT CODE:           MAINDEC-11-DZR6C-A-D

PRODUCT NAME:           RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 3

DATE:                   DECEMBER, 1976

MAINTAINER:             DIAGNOSTIC GROUP

AUTHOR:                 ROY SPITZER
```

TABLE OF CONTENTS

1.0 ABSTRACT

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 3

A. TESTS THE LOADING OF THE DRIVE BUS MESSAGE SHIFT REGISTER FOR CLASS B COMMANDS.
B. TESTS INDEX AND SECTOR PULSE DETECTION.
C. TESTS SILO AND NPR TRANSFERS FROM MEMORY IN 16 AND 18 BIT MODE.
D. TESTS NON-EXISTANT MEMORY AND UNIBUS PARITY ERROR DETECTION.
E. TESTS READ AND WRITE MFM LOOPBACK.
F. TESTS CLASS B INSTRUCTION ERRORS.

NO RK06 DRIVE IS REQUIRED FOR PROGRAM EXECUTION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 SYSTEM (16K CORE MEMORY)
CONSOLE TERMINAL
DECTAPE, PAPER TAPE READER, OR DECDISK
RK611 CONTROLLER

2.2 PRELIMINARY PROGRAMS

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 1 (DZR6A)
RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 2 (DZR6B)

3.0 OPERATING PROCEDURES

3.1 LOADING PROCEDURE

THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING ABSOLUTE LOADER OR FROM XXDP MEDIA SUPPORTED BY XXDP.

3.2 STARTING PROCEDURE

LOCATION 200 - START PROGRAM

LOCATION 204 - RESTART PROGRAM

LOCATION 214 - REQUEST BUS ADDRESS, VECTOR ADDRESS, AND PRIORITY MODIFICATION

3.3 OPTIONAL SWITCH SETTINGS

SW15 - HALT PROGRAM
SW14 - LOOP ON TEST
SW13 - INHIBIT ERROR TYPE OUT
SW12 - ABORT AFTER 20 ERRORS
SW11 - INHIBIT ITERATION COUNT
SW10 - BELL ON ERROR
SW9 - LOOP ON ERROR
SW8 - LOOP ON TEST IN SWITCHES 0-7

3.5 RUN TIME

FIRST PASS	30 SECONDS
SUBSEQUENT PASSES	8:40 MINUTES

4.0 OPERATING PROCEDURES

THE PROGRAM IS EXECUTED BY STARTING AT THE APPROPIATE ADDRESS.

4.1 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E., AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW ='

EACH TIME SWITCH SETTINGS ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROES ARE NOT REQUIRED. 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

4.2 CONTROL C (↑C) OPERATION

IF ↑C IS TYPED AT ANY TIME DURING THE PROGRAM EXECUTION THE PROGRAM IS HALTED IMMEDIATELY. IF A MONITOR IS PRESENT (XXDP, CHAIN, ACT, APT) THE PROGRAM RETURNS CONTROL TO THE MONITOR. IF NO MONITOR IS PRESENT, THE CPU IS HALTED. DEPRESSING THE CONTINUE KEY WILL DO A PROGRAM RESTART.

```
                                4.3     CONTROL S (↑S) OPERATION

                                IF ↑S IS TYPED AT ANY TIME THE PROGRAM WILL GO INTO A STALL LOOP
                                UNTIL A CONTROL Q (↑Q) IS TYPED.

                                4.4     CONTROL Q (↑Q) OPERATION

                                IF A ↑S HAS BEEN TYPED, TYPING THE ↑Q CANCELS THE STALL
                                INITIATED BY THE ↑S.
```

5.0 PROGRAM DESCRIPTION

**DRIVE MESSAGES FOR CLASS B INSTRUCTIONS

TEST 1 READ HEADER SEEK MESSAGE

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR. PUT THE RK611 CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ HEADER TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0 HEAD 0, DRIVE 0. CLOCK IN SEEK MESSAGE INTO SHIFT REGIST VERIFY THAT A SEEK IS LOADED WITH THE PROPER BITS IN MESSAGE SET. REPEAT FOR A READ HEADER WITH CDT SET IN 24 SECTOR FORMAT, CYLINDER 1777, HEAD 7, DRIVE 7.

TEST 2 WRITE HEADER SEEK MESSAGE

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0. CLOCK IN SEEK MESSAGE INTO SHIFT REGISTER. VERIFY THAT A SEEK IS LOADED WITH THE R.C BIT SET. REPEAT FOR A WRITE HEADER WITH CDT SET IN 24 SECTOR FORMAT, CYLINDER 1777, HEAD 7, DRIVE 7.

TEST 3 READ HEADER DRIVE CLEAR MESSAGE

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ HEADER WITH CDT SET IN 24 SECTOR FORMAT, CYLINDER 1777, HEAD 7, DRIVE 7. CLOCK SEEK MESSAGE AND MAKE SURE A DRIVE CLEAR IS GENERATED AND THE PROPER BITS ARE SET.

TEST 4 WRITE HEADER DRIVE CLEAR MESSAGE

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER WITH CDT SET IN 24 SECTOR FORMAT, CYLINDER 1777, HEAD 7, DRIVE 7. CLOCK SEEK MESSAGE AND LOAD GENERATED DRIVE CLEAR INTO SHIFT REGISTER. MAKE SURE THE DRIVE CLEAR IS GENERATED AND THE PROPER BITS ARE SET.

**INDEX AND SECTOR PULSE DETECT ON

TEST 5 SECTOR PULSE DETECT IN READ HEADER (PART 1)

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A READ HEADER TO AN RK06
IN 26 SECTOR MODE, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK ANY DRIVE CLEAR MESSAGES.
SIMULATE SECTOR PULSE, 255 ZEROES AND A ONE.

MAKE SURE READ GATE DOES SET.

TEST 6 SECTOR PULSE DETECT IN READ HEADER (PART 2)

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A READ HEADER TO AN RK06
IN 26 SECTOR MODE, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK ANY DRIVE CLEAR MESSAGES.
SIMULATE INDEX PULSE, 255 ZEROES AND A ONE.

MAKE SURE READ GATE DOES NOT SET.

TEST 7 SECTOR PULSE DETECT IN READ HEADER (PART 3)

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
DIAGNOSTIC MODE. ISSUE A READ HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CHECK MESSAGES.
SIMULATE 255 ZEROES AND A ONE.

MAKE SURE READ GATE DOES NOT SET.

TEST 10 INDEX PULSE DETECTION IN WRITE HEADER

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR. PUT
THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER
TO AN RK06, IN 26 SECTOR FORMAT, TO CYLINDER 0, HEAD 0,
DRIVE 0, WITH A ONE WORD TRANSFER. CLOCK THROUGH THE
SEEK AND THE DRIVE CLEAR MESSAGES. ISSUE 200 CONTROLLER
CLOCKS AND MAKE SURE WRITE GATE DOES NOT SET. SIMULATE
SECTOR PULSE AND 200 CONTROLLER CLOCKS MAKING SURE WRITE
GATE DOES NOT SET. SIMULATE INDEX PULSE AND MAKE SURE
WRITE GATE SETS.

**NPR READING OF MEMORY

TEST 11 NPR OUTPUT DATA TRANSFER

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 777, HEAD 7, DRIVE 7. SPECIFY A ONE WORD DATA TRANSFER. CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE INDEX PULSE. CLOCK IN FIRST WORD OF NPR TRANSFER. CHECK INPUT READY, OUTPUT READY, BUS ADDRESS, WORD COUNT, AND CONTENTS OF THE SILO. REPEAT FOR 8 DIFFERENT DATA PATTERNS.

TEST 12 PARTIAL SILO FILLING

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0. SPECIFY A ONE WORD DATA TRANSFER. CLOCK IN ALL SPECIFIED WORDS INTO THE SILO. CHECK WORD COUNT, BUS ADDRESS, INPUT READY, AND OUTPUT READY. MAKE SURE NO MORE THAN SPECIFIED DATA LENGTH IS CLOCKED INTO THE SILO. CHECK THE SILO FOR CORRECT DATA. REPEAT FOR WORD COUNTS 2-65.

TEST 13 SILO FILLING WITH NPR TRANSFERS

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0. SPECIFY A 66 WORD DATA TRANSFER, CLOCK IN ALL 66 WORDS INTO THE SILO. CHECK INPUT READY, OUTPUT READY, BUS ADDRESS, WORD COUNT, AND CONTENTS OF THE SILO.

TEST 14 SILO CAPICITY WITH NPR TRANSFERS

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0. HEAD 0, DRIVE 0. SPECIFY A 68 WORD DATA TRANSFER. CLOCK IN 66 WORDS INTO THE SILO. MAKE SURE THAT SILO WILL STOP FILLING AT 66 WORDS. TAKE ONE WORD FROM SILO AND CHECK IT. CLOCK IN NEXT WORD. MAKE SURE NO MORE THAN ONE WORD IS CLOCKED IN THE SILO. TAKE ONE WORD FROM SILO AND CHECK IT. CLOCK IN NEXT WORD. CLOCK IN NEXT WORD. MAKE SURE NO MORE THAN ONE WORD IS CLOCKED IN THE SILO. TAKE ONE WORD FROM SILO AND CHECK IT. ATTEMPT TO CLOCK IN NEXT WORD AND

```
313
314                     MAKE SURE NO WORDS ARE CLOCK INTO SILO.  UNLOAD THE
315                     SILO AND MAKE SURE ALL THE WORDS ARE CORRECT.
316
317
318             TEST 15 BUS ADDRESS INHIBIT
319
320                     CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
321                     IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN
322                     RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
323                     SPECIFY A 66 WORD DATA TRANSFER WITH BUS ADDRESS
324                     INHIBIT INCREMENT.  CHECK WORD COUNT, BUS ADDRESS,
325                     INPUT READY, OUTPUT READY, AND MAKE SURE ALL THE
326                     WORDS IN THE SILO ARE THE CORRECT SAME WORD.
327
328
329             TEST 16 NON-EXISTENT MEMORY
330
331                     CLEAR RK611 WITH A CONTROLLER CLEAR,  PUT CONTROLLER
332                     IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
333                     IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
334                     SPECIFY A ONE WORD DATA TO A NON-EXISTENT ADDRESS
335                     (760000) AND MAKE SURE THE NON-EXISTENT  MEMORY ERROR
336                     OCCURS IN THE RK611 CONTROLLER.
337
338
339             TEST 17 BUS ADDRESS BIT 16
340
341                     CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
342                     IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO  AN RK06
343                     IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
344                     SPECIFY A ONE WORD DATA TRANSFER FROM 200000.
345                     READ THE SILO AND MAKE SURE RIGHT CONTENTS ARE READ.
346                     REPEAT FOR A TWO WORD TRANSFER FROM ADDRESS 177776.
347                     CHECK BUS ADDRESS AND WORD COUNT.
348
349                     NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
350                            OF MEMORY IS ON THE SYSTEM.
351
352
353             TEST 20 BUS ADDRESS BIT 17
354
355                     CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
356                     IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
357                     IN 26 SECTOR FORMAT, CYLINDER 0, HEADER 0, DRIVE 0.
358                     SPECIFY A ONE WORD DATA TRANSFER FROM 400000.
359                     READ THE SILO AND MAKE SURE RIGHT CONTENTS ARE READ.
360                     REPEAT FOR A TWO WORD TRANSFER FROM ADDRESS 377776.
361                     CHECK BUS ADDRESS AND WORD COUNT.
362
363                     NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
364                            OF MEMORY IS ON THE SYSTEM.
365
366
367
```

```
                TEST 21 ADDRESSING GREATER THAN 96K

                        CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                        IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                        IN 26 SECTOR FORMAT, CYLINDER 0, HEADER 0, DRIVE 0.
                        SPECIFY A ONE WORD DATA TRANSFER FROM 600000.
                        READ THE SILO AND MAKE SURE RIGHT CONTENTS ARE READ.
                        REPEAT FOR A TWO WORD TRANSFER FROM ADDRESS 577776.
                        CHECK BUS ADDRESS AND WORD COUNT.

                        NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
                               OF MEMORY IS ON THE SYSTEM.


                TEST 22 UNIBUS PARITY ERROR

                        CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                        IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                        IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                        SPECIFY A ONE WORD DATA TRANSFER FROM A LOCATION
                        WITH BAD PARITY.  MAKE SURE A UNIBUS PARITY ERROR
                        OCCURS.

                        REPEAT FOR A ONE WORD DATA TRANSFER FROM THE
                        LOCATION PRIOR TO THE LOCATION WITH BAD PARITY.
                        MAKE SURE UNIBUS PARITY ERROR DOES NOT OCCUR.  REPEAT
                        FOR A ONE WORD DATA TRANSFER FROM THE LOCATION AFTER
                        THE LOCATION WITH BAD PARITY.  MAKE SURE UNIBUS PARITY
                        ERROR DOES NOT OCCUR.

                        REPEAT FOR A TWO WORD DATA TRANSFER STARTING WITH
                        THE ADDRESS PRIOR TO THE LOCATION WITH BAD PARITY.
                        MAKE SURE UNIBUS PARITY ERROR DOES OCCUR.

                        NOTE:   THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
                                EXISTS FOR SPECIFIED LOCATION.


                TEST 23 SILO FILL IN 18 BIT MODE

                        CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER
                        IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                        IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0. DRIVE 0,
                        SPECIFY 66  WORD DATA TRANSFER.  CLOCK
                        ALL 66 WORD INTO THE SILO.  CHECK THAT ALL 66
                        WORDS ARE CORRECT (16 LEAST SIGNIFICANT BITS).


                TEST 24 BIT 16 AND 17 READING WITH NPR

                        CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER
                        IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                        IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                        SPECIFY ONE WORD DATA TRANSFER FROM A LOCATION
```

```
                        WITH BAD PARITY.  MAKE SURE A UNIBUS PARITY
                        DOES NOT OCCUR.

                        NOTE:   THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
                                ENABLE EXISTS FOR SPECIFIED LOCATION.


                    **MFM READ LOOPBACK TESTS

                            TEST 25 READ LOOPBACK (PART 1)

                                    CLEAR RK611 WITH A CONTROLLER CLEAR, PUT CONTROLLER
                                    IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
                                    IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                    CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
                                    SIMULATE SECTOR PULSE, 255 ZEROES, A
                                    ONE, AND A HEADER CONSISTING OF THE THREE
                                    FOLLOWING WORDS:

                                            177777
                                            000000
                                            177777

                                    MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
                                    IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.


                            TEST 26 READ LOOPBACK (PART 2)

                                    CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                                    IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
                                    IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                    CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
                                    SIMULATE SECTOR PULSE, 255 ZEROES, A ONE,
                                    AND A HEADER CONSISTING OF THE THREE
                                    FOLLOWING WORDS:

                                            000000
                                            177777
                                            000000

                                    MAKE SURE THAT READY COMES UP AFTER THIRD WORD
                                    IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
```

```
                        TEST 27 READ LOOPBACK (PART 3)

                                CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                                IN DIAGNOSTIC MOVE.  ISSUE A READ HEADER TO AN RK06
                                IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES
                                SIMULATE SECTOR PULSE, 255 ZEROES, A
                                ONE, AND A HEADER CONSISTING OF THE THREE
                                FOLLOWING WORDS:

                                        125252
                                        052525
                                        125252

                                MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
                                IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.


                        TEST 30 READ LOOPBACK (PART 4)

                                CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                                IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
                                IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
                                SIMULATE SECTOR PULSE, 225 ZEROES, A
                                ONE, AND A HEADER CONSISTING OF THE THREE
                                FOLLOWING WORDS:

                                        044444
                                        022222
                                        111111

                                MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
                                IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.


                        TEST 31 READ LOOPBACK (PART 5)

                                CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                                IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
                                26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
                                SIMULATE SECTOR PULSE, 255 ZEROES, A
                                ONE, AND A HEADER CONSISTING OF THE THREE
                                FOLLOWING WORDS.

                                        052012
                                        100520
                                        052012

                                MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
                                IS TRANSRERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
```

```
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
```

TEST 32 READ HEADER IN 18 BIT MODE

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
SIMULATE SECTOR PULSE, 255 ZEROES, A
ONE, AND A HEADER CONSISTING OF THE THREE
FOLLOWING WORDS:

177777
000000
177777

MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
IS TRANSFERRED. CHECK THE SILO FOR CORRECT CONTENTS.

TEST 33 SYNCH DETECT IN READ HEADER

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A READ HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
SIMULATE SECTOR PULSE AND 350 ZEROES. MAKE
SURE READY REMAINS RESET AND THE SILO REMAINS
EMPTY.

TEST 34 ZERO SYNCH ON READ

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
SIMULATE SECTOR PULSE, 255 ZEROES SHIFTED BY A HALF
BIT TIME, A ONE, AND A HEADER CONSISTING OF THE
THREE FOLLOWING WORDS:

177777
000000
177777

MAKE SURE THAT READY COMES AFTER THE THIRD WORD
IS TRANSFERRED. CHECK THE SILO FOR CORRECT CONTENTS.

```
                        **MFM WRITE LOOPBACK TESTS

                TEST 35 WRITE ZEROS UNTIL SECTOR PULSE WITH WRITE HEADER

                        CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                        IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                        IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                        CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE
                        INDEX PULSE AND 500 DATA BITS.  MAKE SURE THAT
                        ZEROES ARE WRITTEN. SIMULATE SECTOR PULSE AND MAKE SURE
                        WRITE GATE RESETS.


                TEST 36 WRITE LOOPBACK (PART 1)

                        CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                        IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
                        IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                        CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
                        INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
                        CONSISTING OF THE FOLLOWING DATA, AND AN INDEX PULSE:

                                177777
                                000000
                                177777

                        MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX PUL
                        CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION


        TEST 37 WRITE LOOPBACK (PART 2)

                CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER

                IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
                INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
                CONSISTING OF THE FOLLOWING DATA, AND AN INDEX PULSE:

                        000000
                        177777
                        000000

                MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
                IS TRANSFERRED.  CHECK FOR CORRECT WRITE ENCODED DATA
                AND PRECOMPENSATION.
```

TEST 40 WRITE LOOPBACK (PART 3)

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. STIMULATE
INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
CONSISTING OF THE FOLLOWING DATA, AND AN INDEX PULSE:

125252
052525
125252

MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX PUL
CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION

TEST 41 WRITE LOOPBACK (PART 4)

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE
INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
CONSISTING OF THE FOLLOWING DATA, AND AND INDEX PULSE:

044444
022222
111111

MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX MOD
CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION

TEST 42 WRITE LOOPBACK (PART 5)

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE
INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
CONSISTING OF THE FOLLOWING DATA, AND INDEX PULSE:

052012
100520
052012

MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX PUL
CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION

```
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
```

TEST 43 WRITE LOOPBACK (PART 6)

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE
INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
CONSISTING OF THE FOLLOWING DATA, AND INDEX PULSE:

155555
066666
155555

MAKE SURE READY COMES UP AFTER SECOND INDEX PULSE.
CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION

TEST 44 WRITE LOOPBACK (PART 7)

CLEAR THE RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE
INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
CONSISTING OF THE FOLLOWING DATA, AND INDEX PULSE:

104210
104210
104210

MAKE SURE READY COMES UP AFTER SECOND INDEX PULSE.
CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION

TEST 45 WRITE TWO HEADERS

CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE
INDEX PULSE, SECTOR PULSE, THREE WORD HEADER
CONSISTING OF THE FOLLOWING DATA:

177777
000000
177777

FOLLOW THAT BY A SECTOR PULSE AND ONE THREE WORD
HEADER CONSISTING OF THE FOLLOWING DATA:

```
                                        000000
                                        177777
                                        000000

                SIMULATE AN INDEX PULSE AND MAKE SURE READY COMES UP.
                CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMENSATION.


TEST 46 DATA FIELD FILLING ON WRITE HEADER

                CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0, AND
                SPECIFY TWO 3 WORD HEADERS CONSISTING OF THE
                FOLLOWING DATA:

                                        125252
                                        052525
                                        125252
                                        052525
                                        125252
                                        052525

                MAKE SURE THE DATA SYNCH ANY OTHER BITS OF DATA AND
                ECC FIELD ARE WRITTEN CORRECTLY.


TEST 47 WRITE HEADER FOR 26 SECTORS

                CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0, SPECIF
                66 WORDS.  MAKE SURE ALL 26 SECTORS ARE WRITTEN CORRECTL


TEST 50 WRITE HEADER IN 24 SECTOR FORMAT

                CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.  PUT
                THE CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER
                OF SIX WORDS IN 24 SECTOR FORMAT TO AN RK06, CYLINDER 0,
                HEAD 0, DRIVE 0.  CLOCK THROUGH THE SEEK AND DRIVE CLEAR
                MESSAGES, SIMULATE INDEX PULSE, SECTOR PULSE, 3 HEADER
                WORDS, SYNCH AND DATA, ANOTHER SECTOR PULSE, 3 HEADER
                WORDS, AND AN INDEX PULSE.  CHECK DATA WRITTEN TO MAKE
                SURE ONLY LOW 16 BITS OF SILO ARE USED.
```

```
                                        **TYPE B INSTRUCTION ERRORS

                                                TEST 51 FORMAT ERROR (PART 1)

                                                        CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT TH
                                                        CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO A
                                                        RK06, IN 26 SECTOR FORMAT, CYLINDER 43, HEAD 0, DRIVE 0.
                                                        CLOCK IN MAINTENANCE MODE UNTIL PHASE ADDRESS 6.  TURN O
                                                        MAINTENANCE MODE.  MAKE SURE FORMAT ERROR,
                                                        DRIVE AVAILABLE AND CONTROLLER ERROR SET.


                                                TEST 52 FORMAT ERROR (PART 2)

                                                        CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT TH
                                                        CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO A
                                                        RK06, IN 24 SECTOR FORMAT, CYLINDER 3, HEAD 0, DRIVE 0.
                                                        CLOCK IN MAINTENANCE MODE UNTIL PHASE ADDRESS 6.  TURN O
                                                        MAINTENANCE MODE.  MAKE SURE FORMAT ERROR,
                                                        DRIVE AVAILABLE AND CONTROLLER ERROR SET.


                                                TEST 53 FAULT SETTING CONTROLLER ERROR

                                                        CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT TH
                                                        CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO
                                                        AN RK06, IN 26 SECTOR FORMAT, CYLINDER 3, HEAD 0, DRIVE
                                                        0.  CLOCK IN MAINTENANCE MODE UNTIL PHASE ADDRESS 6.
                                                        TURN OFF MAINTENANCE MODE.  MAKE SURE DRIVE
                                                        AVAILABLE AND CONTROLLER ERROR SET.
```

6.0 ERROR REPORTING

THE GENERAL FORMAT OF ERROR REPORTS IS:

```
OPERATION DESCRIPTION AND ERROR DESCRIPTION
TEST    ERROR
NUM     PC
XXXXXX  YYYYYY
  EXPECT  ACTUAL  OTHER PERTENANT
  REG     REG     INFORMATION
  ZZZZZZ  WWWWWW  AAAAAA
```

NOTE: MORE THAN ONE SET OF EXPECT/ACTUAL REGISTERS MAY BE PRINTED OUT. OTHER PERTENANT INFORMATION MAY CONSIST OF MORE THAN ONE WORD.

OTHER PERTINENT INFORMATION MAY CONTAIN A WORD LABELED "BIT COUNT". THIS COUNT IS REPORTED WHEN THE OPERATION BEING PERFORMED INVOLVES CLOCKING THE CONTROLLER THROUGH ONE OR MORE OF THE VARIOUS FIELDS THAT MAKE UP THE PHYSICAL SECTOR FORMAT.

THESE FIELDS (ALL VALUES GIVEN IN OCTAL) ARE:

FIELD	BITS	WORDS
HEADER PREAMBLE	400	20
HEADER	60	3
GAP	100	4
DATA PREAMBLE	400	20
DATA		
(22(10) SECTOR/TRACK)	10000	400
(20(10) SECTOR/TRACK)	11000	400
ECC	40	2
POSTAMBLE	20	1
GAP		
(22(10) SECTOR/TRACK)	160	7
(20(10) SECTOR/TRACK)	140	6

REFER TO THE RK06 UNIBUS DISK SUBSYSTEM SPECIFICATION FOR MORE DETAILED DESCRIPTION.

THE "BIT COUNT" REPORTED IS INITIALIZED AT THE START OF EACH FIELD AND IS INCREMENTED FOR EACH BIT PROCESSED.

WHEN THE OPERATION BEING PERFORMED INVOLVES WRITING, OTHER PERTINENT INFORMATION MAY CONTAIN WORDS LABELED PRESENT BIT, PRESENT BIT -1, PRESENT BIT -2, AND PRESENT BIT +1. THESE BITS ARE PRESENTED IN THIS WAY TO SHOW THE WRITE DATA STREAM IN THE SAME MANNER THAT THE RK611 LOOKS AT THE DATA STREAM TO COMPUTE PRECOMPENSATION ADVANCE AND PRECOMPENSATION DELAY IN THE WRITE OPERATION.

WHEN THE OPERATION BEING PERFORMED INVOLVES READING, OTHER PERTINENT INFORMATION MAY CONTAIN PREVIOUS BIT AND PRESENT BIT WORDS. THESE BITS RELATE TO RK611 LOGIC THAT DETERMINES WHEN THE BIT IS VALID FROM THE DECODER.

```
                                %
                        .NLIST  CND,MD,MC,TOC
                        .LIST   ME
                        .ENABL  ABS,AMA
167400          $SWR=   167400
000001          $TN=    1
                        .TITLE  RK611 DISKLESS CONTROLLER DIAGNOSTIC: P3 MD-11-DZR6CA
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ROY SPITZER
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                        ;*
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        12            ABORT PROGRAM AFTER 20 ERRORS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>
                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100          STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011          HT=     11                      ;;CODE FOR HORIZONTAL TAB
000012          LF=     12                      ;;CODE FOR LINE FEED
000015          CR=     15                      ;;CODE FOR CARRIAGE RETURN
000200          CRLF=   200                     ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776          PS=     177776                  ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774          STKLMT= 177774                  ;;STACK LIMIT REGISTER
177772          PIRQ=   177772                  ;;PROGRAM INTERRUPT REQUEST REGISTER
177570          DSWR=   177570                  ;;HARDWARE SWITCH REGISTER
177570          DDISP=  177570                  ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000          R0=     %0                      ;;GENERAL REGISTER
000001          R1=     %1                      ;;GENERAL REGISTER
000002          R2=     %2                      ;;GENERAL REGISTER
000003          R3=     %3                      ;;GENERAL REGISTER
000004          R4=     %4                      ;;GENERAL REGISTER
000005          R5=     %5                      ;;GENERAL REGISTER
000006          R6=     %6                      ;;GENERAL REGISTER
000007          R7=     %7                      ;;GENERAL REGISTER
```

```
                        SP=     %6              ;;STACK POINTER
                        PC=     %7              ;;PROGRAM COUNTER

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

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                        SW15=   100000
                        SW14=   40000
                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        SW06=   100
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        BIT15=  100000
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
                        BIT09=  1000
                        BIT08=  400
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
```

```
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                        ;*KT11 VECTOR ADDRESS

000250                  MMVEC=  250

                        ;*KT11 STATUS REGISTER ADDRESSES

177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516

                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

172300                  KIPDR0= 172300
172302                  KIPDR1= 172302
172304                  KIPDR2= 172304
172306                  KIPDR3= 172306
172310                  KIPDR4= 172310
172312                  KIPDR5= 172312
172314                  KIPDR6= 172314
172316                  KIPDR7= 172316

                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

172340                  KIPAR0= 172340
172342                  KIPAR1= 172342
172344                  KIPAR2= 172344
172346                  KIPAR3= 172346
```

```
1031    172350          KIPAR4= 172350
1032    172352          KIPAR5= 172352
1033    172354          KIPAR6= 172354
1034    172356          KIPAR7= 172356
1035
1036    000114          MEMVEC= 114             ;MEMORY PARITY VECTOR
1037    172100          MEMBAS= 172100          ;MEM PARITY OPTION
1038    000004          WR.PAR= 4               ;WRITE BAD PARITY
1039    000001          PAR.EN= 1               ;ENABLE PARITY ENABLE
1040    120210          AVECT1= 120210          ;DEFINE RK611 VECTOR ADDRESS
1041    000005          APRIOR= 5               ;DEFINE RK611 PRIORITY
1042    177440          ABASE=  177440          ;DEFINE BASE OF RK611 REGISTERS
1043
1044                    .SBTTL  RK611 CONTROLLER REGISTER DEFINITION
1045
1046    000000          RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
1047    000002          RKWC=   2               ;WORD COUNT REGISTER
1048    000004          RKBA=   4               ;BUS ADDRESS REGISTER
1049    000006          RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
1050    000010          RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
1051    000012          RKDS=   12              ;DRIVE STATUS REGISTER
1052    000014          RKER=   14              ;ERROR REGISTER
1053    000016          RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
1054    000020          RKDCYL= 20              ;DESIRED CYLINDER REGISTER
1055    000024          RKDB=   24              ;DATA BUFFER
1056    000026          RKMR1=  26              ;MAINTENANCE REGISTER 1
1057    000034          RKMR2=  34              ;MAINTENANCE REGISTER 2
1058    000036          RKMR3=  36              ;MAINTENANCE REGISTER 3
1059    000030          RKECPS= 30              ;ECC POSITION INFORMATION
1060    000032          RKECPT= 32              ;ECC PATTERN INFORMATION
1061    000022          RKSPAR= 22              ;SPARE REGISTER
1062
1063                    .SBTTL  DRIVE COMMANDS
1064
1065    000001          SELDRV= 01              ;SELECT DRIVE
1066    000003          PACK=   03              ;PACK ACKNOWLEDGE
1067    000005          CLEAR=  05              ;DRIVE CLEAR
1068    000007          UNLOAD= 07              ;UNLOAD
1069    000011          SRTSPL= 11              ;START SPINDLE
1070    000013          RECAL=  13              ;RECALIBRATE
1071    000015          OFFSET= 15              ;OFFSET
1072    000017          SEEK=   17              ;SEEK
1073    000021          RDDATA= 21              ;READ DATA
1074    000023          WRDATA= 23              ;WRITE DATA
1075    000025          RDHEAD= 25              ;READ HEADER
1076    000027          WRHEAD= 27              ;WRITE HEADER AND DATA
1077    000031          WRTCHK= 31              ;WRITE CHECK
1078    000300          INTR=   300             ;GENERATE INTERRUPT TO CPU
1079
1080                    .SBTTL  CONTROL AND STATUS REGISTER 1 BITS
1081
1082    000001          GO=     BIT0            ;GO BIT
1083    000100          IE=     BIT6            ;INTERRUPT ENABLE
1084    000200          RDY=    BIT7            ;CONTROLLER READY
1085    000400          BA16=   BIT8            ;BUS ADDRESS BIT 16
1086    001000          BA17=   BIT9            ;BUS ADDRESS BIT 17
```

```
1087            002000          CDT=    BIT10                   ;CONTROLLER DRIVE TYPE (0=RK06)
1088            004000          CTO=    BIT11                   ;CONTROLLER TIMED OUT WAITING FOR
1089                                                            ;  DRIVE RESPONSE
1090            010000          CFMT=   BIT12                   ;CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
1091            020000          SPAR=   BIT13                   ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
1092            040000          DI=     BIT14                   ;DRIVE INTERRUPT
1093            100000          CERR=   BIT15                   ;CONTROLLER ERROR
1094            100000          CCLR=   BIT15                   ;CONTROLLER CLEAR
1095
1096                            .SBTTL  CONTROL AND STATUS REGISTER 2 BITS
1097
1098            000007          DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
1099            000010          RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1100            000020          BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
1101            000040          SCLR=   BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
1102            000100          IR=     BIT6                    ;INPUT READY
1103            000200          OR=     BIT7                    ;OUTPUT READY
1104            000400          UFE=    BIT8                    ;UNIT FIELD ERROR
1105            001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1106            002000          PGE=    BIT10                   ;PROGRAMMING ERROR
1107            004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1108            010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
1109            020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1110            040000          WCE=    BIT14                   ;WRITE CHECK ERROR
1111            100000          DLT=    BIT15                   ;DATA LATE ERROR
1112
1113                            .SBTTL  ERROR REGISTER BIT DEFINITION
1114
1115          000001            ILF=    BIT0                    ;ILLEGAL FUNCTION CODE
1116          000002            SKI=    BIT1                    ;SEEK INCOMPLETE
1117          000004            NXF=    BIT2                    ;NON-EXECUTABLE DRIVE FUNCTION
1118          000010            DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
1119          000020            FMTE=   BIT4                    ;FORMAT ERROR
1120          000040            DTYE=   BIT5                    ;DRIVE TYPE ERROR
1121          000100            ECH=    BIT6                    ;ECC HARD
1122          000200            BSE=    BIT7                    ;BAD SECTOR ERROR
1123          000400            HVRC=   BIT8                    ;HEADER VRC ERRROR
1124          001000            COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
1125          002000            IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR
1126          004000            WLE=    BIT11                   ;WRITE LOCK ERROR
1127          010000            DTE=    BIT12                   ;DRIVE TIMING ERROR
1128          020000            OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
1129          040000            UNS=    BIT14                   ;DRIVE UNSAFE
1130          100000            DCK=    BIT15                   ;DATA CHECK
1131
1132                            .SBTTL  STATUS REGISTER BIT DEFINITION
1133
1134          000001            DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1135                                                            ;  THIS BIT IS RESET)
1136          000004            OFST=   BIT2                    ;DRIVE OFFSET
1137          000010            ACLO=   BIT3                    ;AC LOW
1138          000020            SPDLSS= BIT4                    ;SPEED LOSS
1139          000040            DROT=   BIT5                    ;DRIVE OFF TRACK
1140          000100            VV=     BIT6                    ;VOLUME VALID
1141          000200            DRDY=   BIT7                    ;DRIVE READY
1142          000400            DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
```

```
1143            004000                  WRL=    BIT11                   ;WRITE LOCK
1144            020000                  PIP=    BIT13                   ;POSITIONING IN PROGRESS
1145            040000                  DSC=    BIT14                   ;DRIVE STATUS CHANGE
1146            100000                  SVAL=   BIT15                   ;STATUS VALID
1147
1148                                    .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION
1149
1150            000017                  MESMSK= 17                      ;MESSAGE MASK
1151
1152            000020                  PAT=    BIT4                    ;FORCE EVEN PARITY ON DRIVE MESSAGE LINES
1153            000040                  DMD=    BIT5                    ;DIAGNOSTIC MODE
1154            000100                  MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
1155            000200                  MIND=   BIT7                    ;MAINTENANCE INDEX
1156            000400                  MCLK=   BIT8                    ;MAINTENANCE CLOCK
1157            001000                  MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
1158            002000                  MEWD=   BIT10                   ;MAINTENACNE ENCODED WRITE DATA
1159            004000                  PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
1160            010000                  PCD=    BIT12                   ;PRECOMPENSATION DELAY
1161            020000                  ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
1162            040000                  WRTGAT= BIT14                   ;WRITE GATE
1163            100000                  RDGATE= BIT15                   ;READ GATE
1164
1165                                    .SBTTL  TRANSMITTED MESSAGE A
1166
1167            000020                  S.SEEK= BIT4                    ;SEEK COMMAND
1168            000040                  S.RECL= BIT5                    ;RECALIBRATE COMMAND
1169            000100                  S.STSP= BIT6                    ;START SPINDLE COMMAND
1170            000200                  S.RTC=  BIT7                    ;DRIVE RETURN TO CENTERLINE COMMAND
1171            000400                  S.CLR=  BIT8                    ;CLEAR ERROR AND DSC
1172            001000                  S.FMT=  BIT9                    ;FORMAT
1173            002000                  S.UNLD= BIT10                   ;UNLOAD
1174            004000                  S.PACK= BIT11                   ;SET VOLUME VALID (PACK ACKNOWNLEDGE)
1175                                    .SBTTL  TRAP CATCHER
1176
1177            000000                          .=0
1178                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1179                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1180                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1181            000174                          .=174
1182  000174    000000        DISPREG:  .WORD   0                       ;;SOFTWARE DISPLAY REGISTER
1183  000176    000000        SWREG:    .WORD   0                       ;;SOFTWARE SWITCH REGISTER
1184                                    .SBTTL  STARTING ADDRESS(ES)
1185  000200    000137  003656                JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1186  000204    000137  003646                JMP     RESTRT          ;JUMP TO RESTART ROUTINE
1187            000214                        .=214
1188  000214    000137  003636                JMP     PARM            ;JUMP TO OPERATOR ASSIGNED PARMETERS
1189                                    .SBTTL  ACT11 HOOKS
1190
1191                                    ;;*****************************************************************
1192                                    ;HOOKS REQUIRED BY ACT11
1193            000220                        $SVPC=.                   ;SAVE PC
1194            000046                        .=46
1195  000046    044662                        $ENDAD                    ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1196            000052                        .=52
1197  000052    000000                        .WORD   0                 ;;2)SET LOC.52 TO ZERO
1198            000220                        .=$SVPC                   ;; RESTORE PC
```

```
                    000114                          .=MEMVEC
000114  046216                                      MEMERR
000116  000340                                      PR7
                    001000                          .=1000
                                            .SBTTL  APT PARAMETER BLOCK

                                            ;;*****************************************************************
                                            ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                            ;;*****************************************************************
                    001000                          .$X=.   ;;SAVE CURRENT LOCATION
                    000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                                      200     ;;FOR APT START UP
                    000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                                      $APTHDR ;;POINT TO APT HEADER BLOCK
                    001000                          .=.$X   ;;RESET LOCATION COUNTER
                                            ;;*****************************************************************
                                            ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                            ;INTERFACE SPEC.

001000                                      $APTHD:
001000  000000                              $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001214                              $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000000                              $TSTM:  .WORD           ;;RUN TIM OF LONGEST TEST
001006  000000                              $PASTM: .WORD           ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000000                              $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000032                                      .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102  000                     $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
001103  000                     $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
001104  000000                  $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                  $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                  $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                  $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
001114  000                     $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
001115  001                     $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                  $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                  $GDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                  $BDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                  $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
001126  000000                  $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
001130  000000                          .WORD   0       ;;RESERVED--NOT TO BE USED
001132  000000                          .WORD   0
001134  000                     $AUTOB: .BYTE   0       ;;AUTOMATIC MODE INDICATOR
001135  000                     $INTAG: .BYTE   0       ;;INTERRUPT MODE INDICATOR
001136  000000                          .WORD   0
001140  177570                  SWR:    .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
001142  177570                  DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
001144  177560                  $TKS:   177560          ;;TTY KBD STATUS
001146  177562                  $TKB:   177562          ;;TTY KBD BUFFER
001150  177564                  $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566                  $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000                     $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002                     $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012                     $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000                     $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000                  $TMP0:  .WORD   0       ;;USER DEFINED
001162  000000                  $TMP1:  .WORD   0       ;;USER DEFINED
001164  000000                  $TMP2:  .WORD   0       ;;USER DEFINED
001166  000000                  $TMP3:  .WORD   0       ;;USER DEFINED
001170  000000                  $TMP4:  .WORD   0       ;;USER DEFINED
001172  000000                  $TMP5:  .WORD   0       ;;USER DEFINED
001174  000000                  $TMP6:  .WORD   0       ;;USER DEFINED
001176  000000                  $TMP7:  .WORD   0       ;;USER DEFINED
001200  000000                  $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
001202  000000                  $ESCAPE:0               ;;ESCAPE ON ERROR ADDRESS
001204  177607  000377        $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001210  077                     $QUES:  .ASCII  /?/     ;;QUESTION MARK
001211  015                     $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001212  000012                  $LF:    .ASCIZ  <12>    ;;LINE FEED
                                ;;**********************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;**********************************************************************
                                .EVEN
```

```
1281  001214                    $MAIL:                  ;;APT MAILBOX
1282  001214  000000            $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
1283  001216  000000            $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
1284  001220  000000            $TESTN: .WORD   ATESTN  ;;TEST NUMBER
1285  001222  000000            $PASS:  .WORD   APASS   ;;PASS COUNT
1286  001224  000000            $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
1287  001226  000000            $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
1288  001230  000000            $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
1289  001232  000000            $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1290  001234                    $ETABLE:                ;;APT ENVIRONMENT TABLE
1291  001234      000           $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1292  001235      000           $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1293  001236  000000            $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1294  001240  000000            $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1295  001242  000000            $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1296                            ;*                      BITS 15-11=CPU TYPE
1297                            ;*                           11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1298                            ;*                           11/70=06,PDQ=07,Q=10
1299                            ;*                      BIT 10=REAL TIME CLOCK
1300                            ;*                      BIT  9=FLOATING POINT PROCESSOR
1301                            ;*                      BIT  8=MEMORY MANAGEMENT
1302  001244      000           $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1303  001245      000           $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1304                            ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1305                            ;*                           900 NSEC CORE=001
1306                            ;*                           300 NSEC BIPOLAR=002
1307                            ;*                           500 NSEC MOS=003
1308  001246  000000            $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1309                            ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1310  001250      000           $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1311  001251      000           $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1312  001252  000000            $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1313  001254      000           $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1314  001255      000           $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1315  001256  000000            $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1316  001260      000           $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1317  001261      000           $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1318  001262  000000            $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1319  001264  120210            $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1320  001266  000000            $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1321  001270  177440            $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1322  001272  000000            $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1323  001274  000000            $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
1324  001276  000000            $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
1325  001300                    $ETEND:
1326                            .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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

001300                          $ERRTB:
                                ;       ERROR 1:  ATTEMPTING TO CHECK SEEK MESSAGE FROM READ HEADER
                                ;                 CS1 INCORRECT.
001300  057676                          EM200
001302  064055                          EM3000
001304  053136                          DT001
001306  053600                          DF001
                                ;       ERROR 2:  ATTEMPTING TO CHECK SEEK MESSAGE FROM READ HEADER
                                ;                 MESSAGE A INCORRECT.
001310  057676                          EM200
001312  064120                          EM3001
001314  053136                          DT001
001316  053600                          DF001
                                ;       ERROR 3:  ATTEMPTING TO CHECK SEEK MESSAGE FROM READ HEADER
                                ;                 MESSAGE B INCORRECT.
001320  057676                          EM200
001322  064144                          EM3002
001324  053136                          DT001
001326  053600                          DF001
                                ;       ERROR 4:  ATTEMPTING TO CHECK SEEK MESSAGE FROM WRITE HEADER
                                ;                 CS1 INCORRECT.
001330  057760                          EM201
001332  064055                          EM3000
001334  053136                          DT001
001336  053600                          DF001
                                ;       ERROR 5:  ATTEMPTING TO CHECK SEEK MESSAGE FROM WRITE HEADER
                                ;                 MESSAGE A INCORRECT.
001340  057760                          EM201
001342  064120                          EM3001
001344  053136                          DT001
001346  053600                          DF001
                                ;       ERROR 6:  ATTEMPTING TO CHECK SEEK MESSAGE FROM WRITE HEADER
                                ;                 MESSAGE IS INCORRECT.
001350  057760                          EM201
001352  064144                          EM3002
001354  053136                          DT001
001356  053600                          DF001
                                ;       ERROR 7:  ATTEMPTING TO CHECK DRIVE CLEAR MESSAGE FROM READ HEADER
                                ;                 CS1 INCORRECT
001360  060043                          EM202
001362  064055                          EM3000
001364  053136                          DT001
```

```
1383  001366  053600                        DF001
1384                              ;         ERROR 10: ATTEMPTING TO CHECK DRIVE CLEAR MESSAGE FROM READ HEADER
1385                              ;                   MESSAGE A INCORRECT.
1386  001370  060043                        EM202
1387  001372  064120                        EM3001
1388  001374  053136                        DT001
1389  001376  053600                        DF001
1390                              ;         ERROR 11: ATTEMPTING TO CHECK DRIVE CLEAR MESSAGE FROM READ HEADER
1391                              ;                   MESSAGE B INCORRECT.
1392  001400  060043                        EM202
1393  001402  064144                        EM3002
1394  001404  053136                        DT001
1395  001406  053600                        DF001
1396                              ;         ERROR 12: ATTEMPTING TO CHECK DRIVE CLEAR MESSAGE FROM WRITE HEADER
1397                              ;                   CS1 INCORRECT.
1398  001410  060134                        EM203
1399  001412  064055                        EM3000
1400  001414  053136                        DT001
1401  001416  053600                        DF001
1402                              ;         ERROR 13: ATTEMPTING TO CHECK DRIVE CLEAR MESSAGE FROM WRITE HEADER
1403                              ;                   MESSAGE A INCORRECT.
1404  001420  060134                        EM203
1405  001422  064120                        EM3001
1406  001424  053136                        DT001
1407  001426  053600                        DF001
1408                              ;         ERROR 14: ATTEMPTING TO CHECK DRIVE CLEAR MESSAGE FROM WRITE HEADER
1409                              ;                   MESSAGE B INCORRECT.
1410  001430  060134                        EM203
1411  001432  064144                        EM3002
1412  001434  053136                        DT001
1413  001436  053600                        DF001
1414                              ;         ERROR 15: ATTEMPTING A READ HEADER TO CHECK SECTOR PULSE DETECT
1415                              ;                   CS1 INCORRECT AFTER SENDING DRIVE CLEAR.
1416  001440  060226                        EM204
1417  001442  064170                        EM3003
1418  001444  053156                        DT015
1419  001446  053624                        DF015
1420                              ;         ERROR 16: ATTEMPTING A READ HEADER TO CHECK SECTOR PULSE DETECT
1421                              ;                   CS1 INCORRECT AFTER DATE SIMULATION.
1422  001450  060226                        EM204
1423  001452  064240                        EM3004
1424  001454  053156                        DT015
1425  001456  053624                        DF015
1426                              ;         ERROR 17: ATTEMPTING A READ HEADER TO CHECK SECTOR PULSE DETECT
1427                              ;                   MAINT REG. 1 INCORRECT DURING DATA SIMULATION (SECTOR PULSE)
1428  001460  060226                        EM204
1429  001462  064312                        EM3005
1430  001464  053166                        DT017
1431  001466  053650                        DF017
1432                              ;         ERROR 20: ATTEMPTING A READ HEADER TO CHECK SECTOR PULSE DETECT
1433                              ;                   MAINT REG. 1 INCORRECT DURING DATA SIMULATION (INDEX PULSE)
1434  001470  060226                        EM204
1435  001472  064414                        EM3006
1436  001474  053166                        DT017
1437  001476  053650                        DF017
1438                              ;         ERROR 21: ATTEMPTING A READ HEADER TO CHECK SECTOR PULSE DETECT
```

```
1439                                    ;               MAINT REG. 1 INCORRECT DURING DATA SIMULATION (NO INDEX OR SECTOR)
1440  001500  060226                    ;       EM204
1441  001502  064515                    ;       EM3007
1442  001504  053166                    ;       DT017
1443  001506  053650                    ;       DF017
1444                                    ;       ERROR 22: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1445                                    ;               CS1 INCORRECT AFTER SENDING DRIVE CLEAR
1446  001510  060314                    ;       EM205
1447  001512  064170                    ;       EM3003
1448  001514  053204                    ;       DT022
1449  001516  053674                    ;       DF022
1450                                    ;       ERROR 23: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1451                                    ;               BUS ADD INCORRECT AFTER SENDING DRIVE CLEAR.
1452  001520  060314                    ;       EM205
1453  001522  064636                    ;       EM3008
1454  001524  053204                    ;       DT022
1455  001526  053674                    ;       DF022
1456                                    ;       ERROR 24: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1457                                    ;               WORD COUNT INCORRECT AFTER SENDING DRIVE CLEAR
1458  001530  060314                    ;       EM205
1459  001532  064716                    ;       EM3009
1460  001534  053204                    ;       DT022
1461  001536  053674                    ;       DF022
1462                                    ;       ERROR 25: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1463                                    ;               MAINT REG 1 INCORRECT DURING DATA SIMULATION (NO INDEX OR SECTOR)
1464  001540  060314                    ;       EM205
1465  001542  065214                    ;       EM3014
1466  001544  053224                    ;       DT025
1467  001546  053720                    ;       DF025
1468                                    ;       ERROR 26: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1469                                    ;               CS1 CHANGED DURING COMMAND EXECUTION
1470  001550  060314                    ;       EM205
1471  001552  064775                    ;       EM3010
1472  001554  053204                    ;       DT022
1473  001556  053674                    ;       DF022
1474                                    ;       ERROR 27: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1475                                    ;               BUS ADDRESS CHANGED BEFORE INDEX PULSE
1476  001560  060314                    ;       EM205
1477  001562  065042                    ;       EM3011
1478  001564  053204                    ;       DT022
1479  001566  053674                    ;       DF022
1480                                    ;       ERROR 30: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1481                                    ;               WORD COUNT CHANGED BEFORE INDEX PULSE
1482  001570  060314                    ;       EM205
1483  001572  065111                    ;       EM3012
1484  001574  053204                    ;       DT022
1485  001576  053674                    ;       DF022
1486                                    ;       ERROR 31: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1487                                    ;               CS1 CHANGED AFTER INDEX PULSE
1488  001600  060314                    ;       EM205
1489  001602  065157                    ;       EM3013
1490  001604  053236                    ;       DT031
1491  001606  053744                    ;       DF031
1492                                    ;       ERROR 32: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1493                                    ;               BUS ADDRESS CHANGED AFTER INDEX PULSE.
1494  001610  060314                    ;       EM205
```

```
1495  001612  065265                          EM3015
1496  001614  053236                          DT031
1497  001616  053744                          DF031
1498                              ;           ERROR 33: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1499                              ;                     WORD COUNT CHANGED AFTER INDEX PULSE
1500  001620  060314                          EM205
1501  001622  065333                          EM3016
1502  001624  053236                          DT031
1503  001626  053744                          DF031
1504                              ;           ERROR 34: ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT
1505                              ;                     MAINT REG 1 INCORRECT AFTER SECTOR PULSE
1506  001630  060314                          EM205
1507  001632  065713                          EM3025
1508  001634  053236                          DT031
1509  001636  053744                          DF031
1510                              ;           ERROR 35:  ATTEMPTING AN NPR READ OF ONE WORD
1511                              ;                     CS1 INCORRECT
1512  001640  060402                          EM206
1513  001642  064055                          EM3000
1514  001644  053262                          DT035
1515  001646  053770                          DF035
1516                              ;           ERROR 36:  ATTEMPTING AN NPR READ OF ONE WORD
1517                              ;                     CS2 INCORRECT
1518  001650  060402                          EM206
1519  001652  065400                          EM3018
1520  001654  053262                          DT035
1521  001656  053770                          DF035
1522                              ;           ERROR 37:  ATTEMPTING AN NPR READ OF ONE WORD
1523                              ;                     BUS ADDRESS INCORRECT
1524  001660  060402                          EM206
1525  001662  065444                          EM3019
1526  001664  053262                          DT035
1527  001666  053770                          DF035
1528                              ;           ERROR 40:  ATTEMPTING AN NPR READ OF ONE WORD
1529                              ;                     WORD COUNT REG INCORRECT
1530  001670  060402                          EM206
1531  001672  065472                          EM3020
1532  001674  053262                          DT035
1533  001676  053770                          DF035
1534                              ;           ERROR 41:  ATTEMPTING AN NPR READ OF ONE WORD
1535                              ;                     WORD READ INCORRECT
1536  001700  060402                          EM206
1537  001702  065523                          EM3021
1538  001704  053306                          DT041
1539  001706  054014                          DF041
1540                              ;           ERROR 42:  ATTEMPTING AN NPR READ OF ONE WORD
1541                              ;                     CS1 INCORRECT AFTER READING DATA BUFFER
1542  001710  060402                          EM206
1543  001712  065547                          EM3022
1544  001714  053316                          DT042
1545  001716  054040                          DF042
1546                              ;           ERROR 43:  ATTEMPTING AN NPR READ OF ONE WORD
1547                              ;                     CS2 INCORRECT AFTER READING DATA BUFFER
1548  001720  060402                          EM206
1549  001722  065617                          EM3023
1550  001724  053316                          DT042
```

```
1551  001726  054040                        DF042
1552                                ;       ERROR 44:  ATTEMPTING AN NPR READ OF ONE WORD
1553                                ;                  CS1 INCORRECT
1554  001730  060445                        EM207
1555  001732  064055                        EM3000
1556  001734  053262                        DT035
1557  001736  053770                        DF035
1558                                ;       ERROR 45:  ATTEMPTING AN NPR READ
1559                                ;                  CS2 INCORRECT
1560  001740  060445                        EM207
1561  001742  065400                        EM3018
1562  001744  053262                        DT035
1563  001746  053770                        DF035
1564                                ;       ERROR 46:  ATTEMPTING AN NPR READ
1565                                ;                  BUS ADDRESS INCORRECT
1566  001750  060445                        EM207
1567  001752  065444                        EM3019
1568  001754  053262                        DT035
1569  001756  053770                        DF035
1570                                ;       ERROR 47:  ATTEMPTING AN NPR READ
1571                                ;                  WORD COUNT INCORRECT
1572  001760  060445                        EM207
1573  001762  065472                        EM3020
1574  001764  053262                        DT035
1575  001766  053770                        DF035
1576                                ;       ERROR 50:  ATTEMPTING NPR READ CHECKING ZERO DETECT
1577                                ;                  CS1 INCORRECT
1578  001770  060474                        EM208
1579  001772  064055                        EM3000
1580  001774  053262                        DT035
1581  001776  053770                        DF035
1582                                ;       ERROR 51:  ATTEMPTING NPR READ CHECKING ZERO DETECT
1583                                ;                  CS2 INCORRECT
1584  002000  060474                        EM208
1585  002002  065400                        EM3018
1586  002004  053262                        DT035
1587  002006  053770                        DF035
1588                                ;       ERROR 52:  ATTEMPTING NPR READ CHECKING ZERO DETECT
1589                                ;                  BUS ADDRESS INCORRECT
1590  002010  060474                        EM208
1591  002012  065444                        EM3019
1592  002014  053262                        DT035
1593  002016  053770                        DF035
1594                                ;       ERROR 53:  ATTEMPTING NPR READ CHECKING ZERO DETECT
1595                                ;                  WORD COUNT INCORRECT
1596  002020  060474                        EM208
1597  002022  065472                        EM3020
1598  002024  053262                        DT035
1599  002026  053770                        DF035
1600                                ;       ERROR 54:  ATTEMPTING NPR READ
1601                                ;                  DATA BUFFER INCORRECT
1602  002030  060445                        EM207
1603  002032  065523                        EM3021
1604  002034  053332                        DT054
1605  002036  054064                        DF054
1606                                ;       ERROR 55:  ATTEMPTING NPR READ
```

```
1607                                    ;               CS1 INCORRECT AFTER READING DATA BUFFER
1608  002040  060445                    ;       EM207
1609  002042  065547                    ;       EM3022
1610  002044  053344                    ;       DT055
1611  002046  054110                    ;       DF055
1612                                    ;       ERROR 56:  ATTEMPTING NPR READ
1613                                    ;               CS2 INCORRECT AFTER READING DATA BUFFER
1614  002050  060445                    ;       EM207
1615  002052  065617                    ;       EM3023
1616  002054  053344                    ;       DT055
1617  002056  054110                    ;       DF055
1618                                    ;       ERROR 57:  ATTEMPTING NPR READ WITH BUS ADDRESS INCREMENT
1619                                    ;               INHIBIT CS1 INCORRECT
1620  002060  060545                    ;       EM209
1621  002062  064055                    ;       EM3000
1622  002064  053262                    ;       DT035
1623  002066  053770                    ;       DF035
1624                                    ;       ERROR 60:  ATTEMPTING NPR READ WITH BUS ADDRESS INCREMENT
1625                                    ;               INHIBIT CS2 INCORRECT
1626  002070  060545                    ;       EM209
1627  002072  065400                    ;       EM3018
1628  002074  053262                    ;       DT035
1629  002076  053770                    ;       DF035
1630                                    ;       ERROR 61:  ATTEMPTING NPR READ WITH BUS ADDRESS INCREMENT
1631                                    ;               INHIBIT BUS ADDRESS INCORRECT
1632  002100  060545                    ;       EM209
1633  002102  065444                    ;       EM3019
1634  002104  053262                    ;       DT035
1635  002106  053770                    ;       DF035
1636                                    ;       ERROR 62:  ATTEMPTING NPR READ WITH BUS ADDRESS INCREMENT
1637                                    ;               INHIBIT WORD COUNT INCORRECT
1638  002110  060545                    ;       EM209
1639  002112  065472                    ;       EM3020
1640  002114  053262                    ;       DT035
1641  002116  053770                    ;       DF035
1642                                    ;       ERROR 63:  ATTEMPTING NPR READ WITH IBA TO CHECK ZERO
1643                                    ;               DETECT-CS1 INCORRECT
1644  002120  060634                    ;       EM210
1645  002122  064055                    ;       EM3000
1646  002124  053262                    ;       DT035
1647  002126  053770                    ;       DF035
1648                                    ;       ERROR 64:  ATTEMPTING NPR READ WITH BAI TO CHECK ZERO
1649                                    ;               DETECT-CS2 INCORRECT
1650  002130  060634                    ;       EM210
1651  002132  065400                    ;       EM3018
1652  002134  053262                    ;       DT035
1653  002136  053770                    ;       DF035
1654                                    ;       ERROR 65:  ATTEMPTING NPR READ WITH BAI TO CHECK ZERO
1655                                    ;               DETECT-BUS ADDRESS INCORRECT
1656  002140  060634                    ;       EM210
1657  002142  065444                    ;       EM3019
1658  002144  053262                    ;       DT035
1659  002146  053770                    ;       DF035
1660                                    ;       ERROR 66:  ATTEMPTING NPR READ WITH BAI TO CHECK ZERO
1661                                    ;               DETECT-WORD COUNT INCORRECT
1662  002150  060634                    ;       EM210
```

```
1663  002152  065472                    EM3020
1664  002154  053262                    DT035
1665  002156  053770                    DF035
1666                          ;         ERROR 67:  ATTEMPTING NPR READ WITH BUS ADDRESS INHIBIT
1667                          ;                    INCREMENT-DATA BUFFER INCORRECT
1668  002160  060545                    EM209
1669  002162  065523                    EM3021
1670  002164  053332                    DT054
1671  002166  054064                    DF054
1672                          ;         ERROR 70:  ATTEMPTING NPR READ WITH BUS ADDRESS INHIBIT
1673                          ;                    INCREMENT-CS1 INCORRECT AFTER READING DATA BUFFER
1674  002170  060545                    EM209
1675  002172  065547                    EM3022
1676  002174  053344                    DT055
1677  002176  054110                    DF055
1678                          ;         ERROR 71:  ATTEMPTING NPR READ WITH ADDRESS BUFFER INHIBIT
1679                          ;                    INCREMENT-CS2 INCORRECT AFTER READING DATA BUFFER
1680  002200  060545                    EM209
1681  002202  065617                    EM3023
1682  002204  053344                    DT055
1683  002206  054110                    DF055
1684                          ;         ERROR 72:  ATTEMPTING TO FORCE NON-EXISTANT MEMORY
1685                          ;                    CS1 INCORRECT
1686  002210  060750                    EM211
1687  002212  064055                    EM3000
1688  002214  053362                    DT072
1689  002216  054134                    DF072
1690                          ;         ERROR 73:  ATTEMPTING TO FORCE NON-EXISTENT MEMORY
1691                          ;                    CS2 INCORRECT
1692  002220  060750                    EM211
1693  002222  065400                    EM3018
1694  002224  053362                    DT072
1695  002226  054134                    DF072
1696                          ;         ERROR 74:  ATTEMPTING TO FORCE NON-EXISTENT MEMORY
1697                          ;                    ERROR REG INCORRECT
1698  002230  060750                    EM211
1699  002232  065667                    EM3024
1700  002234  053362                    DT072
1701  002236  054134                    DF072
1702                          ;         ERROR 75:  ATTEMPTING TO FORCE NON-EXISTENT MEMORY
1703                          ;                    BUS ADDRESS INCORRECT
1704  002240  060750                    EM211
1705  002242  065444                    EM3019
1706  002244  053362                    DT072
1707  002246  054134                    DF072
1708                          ;         ERROR 76:  ATTEMPTING TO FORCE NON-EXISTENT MEMORY
1709                          ;                    WORD COUNT INCORRECT
1710  002250  060750                    EM211
1711  002252  065472                    EM3020
1712  002254  053362                    DT072
1713  002256  054134                    DF072
1714                          ;         ERROR 77:  ATTEMPTING TO CLEAR NON-EXISTENT MEMORY
1715                          ;                    CS1 INCORRECT
1716  002260  061020                    EM212
1717  002262  064055                    EM3000
1718  002264  053412                    DT077
```

```
1719  002266  054170                    DF077
1720                              ;       ERROR 100:  ATTEMPTING TO CLEAR NON-EXISTENT MEMORY
1721                              ;                   CS2 INCORRECT
1722  002270  061020                    EM212
1723  002272  065400                    EM3018
1724  002274  053412                    DT077
1725  002276  054170                    DF077
1726                              ;       ERROR 101: TESTING EXTENDED MEMORY ADDRESSING BITS
1727                              ;                   CS1 INCORRECT
1728  002300  061070                    EM213
1729  002302  064055                    EM3000
1730  002304  053262                    DT035
1731  002306  053770                    DF035
1732                              ;       ERROR 102: TESTING EXTENDED MEMORY ADDRESSING BITS
1733                              ;                   CS2 INCORRECT
1734  002310  061070                    EM213
1735  002312  065400                    EM3018
1736  002314  053262                    DT035
1737  002316  053770                    DF035
1738                              ;       ERROR 103: TESTING EXTENDED MEMORY ADDRESSING BITS
1739                              ;                   BUS ADDRESS INCORRECT
1740  002320  061070                    EM213
1741  002322  065444                    EM3019
1742  002324  053262                    DT035
1743  002326  053770                    DF035
1744                              ;       ERROR 104: TESTING EXTEND MEMORY ADDRESSING BITS
1745                              ;                   WORD COUNT INCORRECT
1746  002330  061070                    EM213
1747  002332  065472                    EM3020
1748  002334  053262                    DT035
1749  002336  053770                    DF035
1750                              ;       ERROR 105: TESTING EXTENDED MEMORY ADDRESSING BITS
1751                              ;                   DATA BUFFER INCORRECT
1752  002340  061070                    EM213
1753  002342  065523                    EM3021
1754  002344  053306                    DT041
1755  002346  054014                    DF041
1756                              ;       ERROR 106: ATTEMPTING TO FORCE UNIBUS PARITY ERROR
1757                              ;                   CS1 INCORRECT
1758  002350  061140                    EM214
1759  002352  064055                    EM3000
1760  002354  053362                    DT072
1761  002356  054134                    DF072
1762                              ;       ERROR 107: ATTEMPTING TO FORCE UNIBUS PARITY ERROR
1763                              ;                   CS2 INCORRECT
1764  002360  061140                    EM214
1765  002362  065400                    EM3018
1766  002364  053362                    DT072
1767  002366  054134                    DF072
1768                              ;       ERROR 110: ATTEMPTING TO FORCE UNIBUS PARITY ERROR
1769                              ;                   BUS ADDRESS INCORRECT
1770  002370  061140                    EM214
1771  002372  065444                    EM3019
1772  002374  053362                    DT072
1773  002376  054134                    DF072
1774                              ;       ERROR 111: ATEMPTING TO FORCE UNIBUS PARITY ERROR
```

```
1775                                    ;               WORD COUNT INCORRECT
1776  002400  061140                    EM214
1777  002402  065472                    EM3020
1778  002404  053362                    DT072
1779  002406  054134                    DF072
1780                                    ;       ERROR 112: ATTEMPTING NPR READ OF LOCATION PRIOR TO BAD PARITY
1781                                    ;               CS1 INCORRECT
1782  002410  061210                    EM215
1783  002412  064055                    EM3000
1784  002414  053262                    DT035
1785  002416  053770                    DF035
1786                                    ;       ERROR 113: ATTEMPTING NPR READ OF LOCATION PRIOR TO BAD PARITY
1787                                    ;               CS2 INCORRECT
1788  002420  061210                    EM215
1789  002422  065400                    EM3018
1790  002424  053262                    DT035
1791  002426  053770                    DF035
1792                                    ;       ERROR 114: ATTEMPTING NPR READ OR LOCATION PRIOR TO BAD PARITY
1793                                    ;               BUS ADDRESS INCORRECT
1794  002430  061210                    EM215
1795  002432  065444                    EM3019
1796  002434  053262                    DT035
1797  002436  053770                    DF035
1798                                    ;       ERROR 115: ATTEMPTING NPR READ  OF LOCATION PRIOR TO BAD PARITY
1799                                    ;               WORD COUNT INCORRECT
1800  002440  061210                    EM215
1801  002442  065400                    EM3018
1802  002444  053262                    DT035
1803  002446  053770                    DF035
1804                                    ;       ERROR 116: ATTEMPTING TO FORCE UNIBUS PARITY ERROR
1805                                    ;               ERROR REG INCORRECT
1806  002450  061140                    EM214
1807  002452  065667                    EM3024
1808  002454  053362                    DT072
1809  002456  054134                    DF072
1810                                    ;       ERROR 117: ATTEMPTING TO CLEAR UNIBUS PARITY ERROR
1811                                    ;               CS1 INCORRECT
1812  002460  061274                    EM216
1813  002462  064055                    EM3000
1814  002464  053412                    DT077
1815  002466  054170                    DF077
1816                                    ;       ERROR 120: ATTEMPTING TO CLEAR UNIBUS PARITY ERROR
1817                                    ;               CS2 INCORRECT
1818  002470  061274                    EM216
1819  002472  065400                    EM3018
1820  002474  053412                    DT077
1821  002476  054170                    DF077
1822                                    ;       ERROR 121: ATTEMPTING 18 BIT NPR READ
1823                                    ;               CS1 INCORRECT
1824  002500  061344                    EM217
1825  002502  064055                    EM3000
1826  002504  053262                    DT035
1827  002506  053770                    DF035
1828                                    ;       ERROR 122: ATTEMPTING 18 BIT NPR READ
1829                                    ;               CS2 INCORRECT
1830  002510  061344                    EM217
```

```
1831  002512  065400                    EM3018
1832  002514  053262                    DT035
1833  002516  053770                    DF035
1834                          ;         ERROR 123: ATTEMPTING 18 BIT NPR READ
1835                          ;           BUS ADDRESS INCORRECT
1836  002520  061344                    EM217
1837  002522  065444                    EM3019
1838  002524  053262                    DT035
1839  002526  053770                    DF035
1840                          ;         ERROR 124: ATTEMPTING 18 BIT NPR READ
1841                          ;           WORD COUNT INCORRECT
1842  002530  061344                    EM217
1843  002532  065472                    EM3020
1844  002534  053262                    DT035
1845  002536  053770                    DF035
1846                          ;         ERROR 125: ATTEMPTING 18 BIT NPR READ CHECKING ZERO DETECT
1847                          ;           CS1 INCORRECT
1848  002540  061377                    EM218
1849  002542  064055                    EM3000
1850  002544  053262                    DT035
1851  002546  053770                    DF035
1852                          ;         ERROR 126: ATTEMPTING 18 BIT NPR READ CHECKING ZERO DETECT
1853                          ;           CS2 INCORRECT
1854  002550  061377                    EM218
1855  002552  065400                    EM3018
1856  002554  053262                    DT035
1857  002556  053770                    DF035
1858                          ;         ERROR 127: ATTEMPTING 18 BIT NPR READ CHECKING ZERO DETECT
1859                          ;           BUS ADDRESS INCORRECT
1860  002560  061377                    EM218
1861  002562  065444                    EM3019
1862  002564  053262                    DT035
1863  002566  053770                    DF035
1864                          ;         ERROR 130: ATTEMPTING 18 BIT NPR READ CHECKING ZERO DETECT
1865                          ;           WORD COUNT INCORRECT
1866  002570  061377                    EM218
1867  002572  065472                    EM3020
1868  002574  053262                    DT035
1869  002576  053770                    DF035
1870                          ;         ERROR 131: ATTEMPTING 18 BIT NPR READ
1871                          ;           DATA BUFFER INCORRECT
1872  002600  061344                    EM217
1873  002602  065523                    EM3021
1874  002604  053316                    DT042
1875  002606  054040                    DF042
1876                          ;         ERROR 132: ATTEMPTING 18 BIT NPR READ
1877                          ;           CS1 INCORRECT AFTER READING DATA BUFFER
1878  002610  061344                    EM217
1879  002612  065547                    EM3022
1880  002614  053344                    DT055
1881  002616  054110                    DF055
1882                          ;         ERROR 133: ATTEMPTING 18 BIT NPR READ
1883                          ;           CS2 INCORRECT AFTER READING DATA BUFFER
1884  002620  061344                    EM217
1885  002622  065617                    EM3023
1886  002624  053344                    DT055
```

```
1887  002626  054110                    DF055
1888                            ;       ERROR 134: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1889                            ;           CS1 INCORRECT
1890  002630  061457                    EM219
1891  002632  064055                    EM3000
1892  002634  053262                    DT035
1893  002636  053770                    DF035
1894                            ;       ERROR 135: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1895                            ;           CS2 INCORRECT
1896  002640  061457                    EM219
1897  002642  065400                    EM3018
1898  002644  053262                    DT035
1899  002646  053770                    DF035
1900                            ;       ERROR 136: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1901                            ;           BUS ADDRESS INCORRECT
1902  002650  061457                    EM219
1903  002652  065444                    EM3019
1904  002654  053262                    DT035
1905  002656  053770                    DF035
1906                            ;       ERROR 137: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1907                            ;           WORD COUNT INCORRECT
1908  002660  061457                    EM219
1909  002662  065472                    EM3020
1910  002664  053262                    DT035
1911  002666  053770                    DF035
1912                            ;       ERROR 140: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1913                            ;           CHECKING ZERO DETECT
1914                            ;           CS1 INCORRECT
1915  002670  061537                    EM220
1916  002672  064055                    EM3000
1917  002674  053262                    DT035
1918  002676  053770                    DF035
1919                            ;       ERROR 141: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1920                            ;           CHECKING ZERO DETECT
1921                            ;           CS2 INCORRECT
1922  002700  061537                    EM220
1923  002702  065400                    EM3018
1924  002704  053262                    DT035
1925  002706  053770                    DF035
1926                            ;       ERROR 142: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1927                            ;           CHECKING ZERO DETECT
1928                            ;           BUS ADDRESS INCORRECT
1929  002710  061537                    EM220
1930  002712  065444                    EM3019
1931  002714  053262                    DT035
1932  002716  053770                    DF035
1933                            ;       ERROR 143: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1934                            ;           CHECKING ZERO DETECT
1935                            ;           WORD COUNT INCORRECT
1936  002720  061537                    EM220
1937  002722  065472                    EM3020
1938  002724  053262                    DT035
1939  002726  053770                    DF035
1940                            ;       ERROR 144: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1941                            ;           DATA BUFFER INCORRECT
1942  002730  061457                    EM219
```

```
1943  002732  065523                          EM3021
1944  002734  053306                          DT041
1945  002736  054014                          DF041
1946                                  ;       ERROR 145: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1947                                  ;                  CS1 INCORRECT AFTER READING DATA BUFFER
1948  002740  061457                          EM219
1949  002742  065547                          EM3022
1950  002744  053412                          DT077
1951  002746  054170                          DF077
1952                                  ;       ERROR 146: ATTEMPTING 18 BIT NPR READ WITH BIT 16(PA) SET
1953                                  ;                  CS2 INCORRECT AFTER READING DATA BUFFER
1954  002750  061457                          EM219
1955  002752  065617                          EM3023
1956  002754  053412                          DT077
1957  002756  054170                          DF077
1958                                  ;       ERROR 147:  UNEXPECTED MEMORY PARITY ENABLE TRAP
1959  002760  057631                          EM000
1960  002762  055011                          DH000C
1961  002764  053132                          DT000
1962  002766  053574                          DF000
1963                                  ;       ERROR 150: ATTEMPTING SIMULATION OF DATA IN READ HEADER
1964                                  ;                  CS1 INCORRECT
1965  002770  061616                          EM221
1966  002772  064055                          EM3000
1967  002774  053426                          DT150
1968  002776  054214                          DF150
1969                                  ;       ERROR 151: ATTEMPTING READ HEADER IN MAINTANENCE MODE
1970                                  ;                  CS1 INCORRECT AFTER COMPLETION OF COMMAND
1971  003000  061673                          EM222
1972  003002  065763                          EM3026
1973  003004  053316                          DT042
1974  003006  054040                          DF042
1975                                  ;       ERROR 152: ATTEMPTING READ HEADER IN MAINTANENCE MODE
1976                                  ;                  CS2 INCORRECT AFTER COMPLETION OF COMMAND
1977  003010  061673                          EM222
1978  003012  066032                          EM3027
1979  003014  053316                          DT042
1980  003016  054040                          DF042
1981                                  ;       ERROR 153: ATTEMPTING DATA BUFFER READ AFTER READ HEADER
1982                                  ;                  CS1 INCORRECT AFTER UNLOADING DATA BUFFER
1983  003020  061746                          EM223
1984  003022  065547                          EM3022
1985  003024  053344                          DT055
1986  003026  054110                          DF055
1987                                  ;       ERROR 154: ATTEMPTING DATA BUFFER READ AFTER READ HEADER
1988                                  ;                  CS2 INCORRECT AFTER UNLOADING DATA
1989  003030  061746                          EM223
1990  003032  065617                          EM3023
1991  003034  053344                          DT055
1992  003036  054110                          DF055
1993                                  ;       ERROR 155: ATTEMPTING DATA BUFFER READ AFTER READ HEADER
1994                                  ;                  DATA READ INCORRECT
1995  003040  061746                          EM223
1996  003042  065523                          EM3021
1997  003044  053332                          DT054
1998  003046  054064                          DF054
```

```
1999                                    ;       ERROR 156: ATTEMPTING SIMULATION OF DATA IN READ HEADER (20 BIT FORMAT)
2000                                    ;                  CS1 INCORRECT
2001   003050  062024                             EM224
2002   003052  064055                             EM3000
2003   003054  053426                             DT150
2004   003056  054214                             DF150
2005                                    ;       ERROR 157: ATTEMPTING READ HEADER (20 BIT FORMAT) IN MAINT MODE/
2006                                    ;                  CS1 INCORRECT AFTER COMMAND COMPLETION
2007   003060  062121                             EM225
2008   003062  065763                             EM3026
2009   003064  053316                             DT042
2010   003066  054040                             DF042
2011                                    ;       ERROR 160: ATTEMPTING READ HEADER (20 BIT FORMAT) IN MAINT MODE
2012                                    ;                  CS2 INCORRECT AFTER COMPLETION OF COMMAND
2013   003070  062121                             EM225
2014   003072  066032                             EM3027
2015   003074  053316                             DT042
2016   003076  054040                             DF042
2017                                    ;       ERROR 161: ATTEMPTING DATA BUFFER READ AFTER 20 BIT READ HEADER
2018                                    ;                  CS1 INCORRECT AFTER UNLOADING DATA BUFFER
2019   003100  062206                             EM226
2020   003102  065547                             EM3022
2021   003104  053344                             DT055
2022   003106  054110                             DF055
2023                                    ;       ERROR 162: ATTEMPTING DATA BUFFER READ AFTER 20 BIT READ HEADER
2024                                    ;                  CS2 INCORRECT AFTER UNLOADING DATA BUFFER
2025   003110  062206                             EM226
2026   003112  065617                             EM3023
2027   003114  053344                             DT055
2028   003116  054110                             DF055
2029                                    ;       ERROR 163: ATTEMPTING DATA BUFFER READ AFTER 20 BIT READ HEADER
2030                                    ;                  DATA BUFFER INCORRECT
2031   003120  062206                             EM226
2032   003122  065523                             EM3021
2033   003124  053332                             DT054
2034   003126  054064                             DF054
2035                                    ;       ERROR 164: ATTEMPTING TO CHECK SYNCH DETECT ON READ HEADER
2036                                    ;                  CS1 INCORRECT
2037   003130  062306                             EM227
2038   003132  064055                             EM3000
2039   003134  053442                             DT164
2040   003136  054240                             DF164
2041                                    ;       ERROR 165: ATTEMPTING TO CHECK SYNCH DETECT ON READ HEADER
2042                                    ;                  CS2 INCORRECT
2043   003140  062306                             EM227
2044   003142  065400                             EM3018
2045   003144  053442                             DT164
2046   003146  054240                             DF164
2047                                    ;       ERROR 166: ATTEMPTING TO CHECK SYNCH DETECT ON READ HEADER
2048                                    ;                  CS1 INCORRECT READING EMPTY SILO
2049   003150  062306                             EM227
2050   003152  066101                             EM3028
2051   003154  053442                             DT164
2052   003156  054240                             DF164
2053                                    ;       ERROR 167: ATTEMPTING TO CHECK SYNCH DETECT ON READ HEADER
2054                                    ;                  CS1 INCORRECT READING EMPTY SILO
```

```
2055  003160  062306                          EM227
2056  003162  066150                          EM3029
2057  003164  053442                          DT164
2058  003166  054240                          DF164
2059                                  ;       ERROR 170:  WRITE BIT ERRORS
2060  003170  000000                 EMW:    0
2061  003172  000000                         0
2062  003174  053460                         DT170
2063  003176  054264                         DF170
2064                                  ;       ERROR 171:  WRITE GATE NOT RESET WITH SECTOR PULSE
2065  003200  062522                         EM231
2066  003202  066217                         EM3030
2067  003204  053504                         DT171
2068  003206  054310                         DF171
2069                                  ;       ERROR 172:  WRITE GATE NOT SET WITH SECTOR PULSE RESET
2070  003210  062644                         EM232
2071  003212  066217                         EM3030
2072  003214  053504                         DT171
2073  003216  054310                         DF171
2074                                  ;       ERROR 173:  WRITE GATE NOT RESET WITH SECOND INDEX PULSE
2075  003220  063072                         EM235
2076  003222  066217                         EM3030
2077  003224  053504                         DT171
2078  003226  054310                         DF171
2079                                  ;       ERROR 174:  CS1 INCORRECT AT END OF WRITE HEADER
2080  003230  063202                         EM236
2081  003232  064055                         EM3000
2082  003234  053156                         DT015
2083  003236  053624                         DF015
2084                                  ;       ERROR 175: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 26)
2085                                  ;       CS1 INCORRECT
2086  003240  063552                         EM241
2087  003242  064055                         EM3000
2088  003244  053514                         DT175
2089  003246  054334                         DF175
2090                                  ;       ERROR 176: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 26)
2091                                  ;       CS2 INCORRECT
2092  003250  063552                         EM241
2093  003252  065400                         EM3018
2094  003254  053514                         DT175
2095  003256  054334                         DF175
2096                                  ;       ERROR 177: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 26)
2097                                  ;       DRIVE STATUS REG INCORRECT
2098  003260  063552                         EM241
2099  003262  066245                         EM3031
2100  003264  053514                         DT175
2101  003266  054334                         DF175
2102                                  ;       ERROR 200: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 26)
2103                                  ;       ERROR REG INCORRECT
2104  003270  063552                         EM241
2105  003272  066300                         EM3032
2106  003274  053514                         DT175
2107  003276  054334                         DF175
2108                                  ;       ERROR 201: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 24)
2109                                  ;       CS1 INCORRECT
2110  003300  063636                         EM242
```

```
2111  003302  064055                  EM3000
2112  003304  053514                  DT175
2113  003306  054334                  DF175
2114                          ;       ERROR 202: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 24)
2115                          ;       CS2 INCORRECT
2116  003310  063636                  EM242
2117  003312  065400                  EM3018
2118  003314  053514                  DT175
2119  003316  054334                  DF175
2120                          ;       ERROR 203: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 24)
2121                          ;       DRIVE STATUS REG INCORRECT
2122  003320  063636                  EM242
2123  003322  066245                  EM3031
2124  003324  053514                  DT175
2125  003326  054334                  DF175
2126                          ;       ERROR 204: ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 24)
2127                          ;       ERROR REGISTER INCORRECT
2128  003330  063636                  EM242
2129  003332  066300                  EM3032
2130  003334  053514                  DT175
2131  003336  054334                  DF175
2132                          ;       ERROR 205:  ATTEMPTING TO CONTROLLER ERROR WITH FAULT BIT IN DRIVE MESS
2133                          ;       CS1 INCORRECT
2134  003340  063722                  EM243
2135  003342  064055                  EM3000
2136  003344  053514                  DT175
2137  003346  054334                  DF175
2138                          ;       ERROR 206:  ATTEMPTING TO CONTROLLER ERROR WITH FAULT BIT IN DRIVE MESS
2139                          ;       CS2 INCORRECT
2140  003350  063722                  EM243
2141  003352  065400                  EM3018
2142  003354  053514                  DT175
2143  003356  054334                  DF175
2144                          ;       ERROR 207:  ATTEMPTING TO CONTROLLER ERROR WITH FAULT BIT IN DRIVE MESS
2145                          ;       DRIVE STATUS REG INCORRECT
2146  003360  063722                  EM243
2147  003362  066245                  EM3031
2148  003364  053514                  DT175
2149  003366  054334                  DF175
2150                          ;       ERROR 210:  ATTEMPTING TO CONTROLLER ERROR WITH FAULT BIT IN DRIVE MESS
2151                          ;       ERROR REG INCORRECT
2152  003370  063722                  EM243
2153  003372  066300                  EM3032
2154  003374  053514                  DT175
2155  003376  054334                  DF175
2156                          ;       ERROR 211:  ATTEMPTING TO CLEAR CONTROLLER ERROR
2157                          ;       CS1 INCORRECT
2158  003400  064023                  EM244
2159  003402  064055                  EM3000
2160  003404  053540                  DT211
2161  003406  054360                  DF211
2162                          ;       ERROR 212:  ATTEMPTING TO CLEAR CONTROLLER ERROR
2163                          ;       CS2 INCORRECT
2164  003410  064023                  EM244
2165  003412  065400                  EM3018
2166  003414  053540                  DT211
```

F04


```
2167  003416  054360                        DF211
2168                                ;       ERROR 213:  ATTEMPTING TO CLEAR CONTROLLER
2169                                ;       DRIVE STATUS REG INCORRECT
2170  003420  064023                        EM244
2171  003422  066245                        EM3031
2172  003424  053540                        DT211
2173  003426  054360                        DF211
2174                                ;       ERROR 214: ATTEMPTINT TO CLEAR CONTROLLER
2175                                ;       ERROR REG INCORRECT
2176  003430  064023                        EM244
2177  003432  066300                        EM3032
2178  003434  053540                        DT211
2179  003436  054360                        DF211
```

```
                                .SBTTL  TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER

003440  000000          T.CS1:  .WORD   0               ;CONTROL AND STATUS REGISTER 1
003442  000000          T.WC:   .WORD   0               ;WORD COUNT REGISTER
003444  000000          T.BA:   .WORD   0               ;BUS ADDRESS REGISTER
003446  000000          T.DA:   .WORD   0               ;DESIRED TRACK SECTOR REGISTER
003450  000000          T.CS2:  .WORD   0               ;CONTROL AND STATUS REGISTER 2
003452  000000          T.DS:   .WORD   0               ;DRIVE STATUS REGISTER
003454  000000          T.ER:   .WORD   0               ;ERROR REGISTER
003456  000000          T.ASOF: .WORD   0               ;ATTENTION SUMMARY AND OFFSET REGISTER
003460  000000          T.DCYL: .WORD   0               ;DESIRED CYLINDER REGISTER
003462  000000          T.DB:   .WORD   0               ;DATA BUFFER
003464  000000          T.MR1:  .WORD   0               ;MAINTENANCE REGISTER 1
003466  000000          T.MR2:  .WORD   0               ;MAINTENANCE REGISTER 2
003470  000000          T.MR3:  .WORD   0               ;MAINTENANCE REGISTER 3
003472  000000          T.ECPS: .WORD   0               ;ECC POSITION INFORMATION
003474  000000          T.ECPT: .WORD   0               ;ECC PATTERN INFORMATION
003476  000000          T.SPAR: .WORD   0               ;SPARE REGISTER

                                .SBTTL  EXPECTED RK611 CONTROLLER REGISTERS

003500  000000          E.CS1:  .WORD   0               ;CONTROL AND STATUS REGISTER 1
003502  000000          E.WC:   .WORD   0               ;WORD COUNT REGISTER
003504  000000          E.BA:   .WORD   0               ;BUS ADDRESS REGISTER
003506  000000          E.DA:   .WORD   0               ;DESIRED TRACK SECTOR REGISTER
003510  000000          E.CS2:  .WORD   0               ;CONTROL AND STATUS REGISTER 2
003512  000000          E.DS:   .WORD   0               ;DRIVE STATUS REGISTER
003514  000000          E.ER:   .WORD   0               ;ERROR REGISTER
003516  000000          E.ASOF: .WORD   0               ;ATTENTION SUMMARY AND OFFSET REGISTER
003520  000000          E.DCYL: .WORD   0               ;DESIRED CYLINDER REGISTER
003522  000000          E.DB:   .WORD   0               ;DATA BUFFER
003524  000000          E.MR1:  .WORD   0               ;MAINTENANCE REGISTER 1
003526  000000          E.MR2:  .WORD   0               ;MAINTENANCE REGISTER 2
003530  000000          E.MR3:  .WORD   0               ;MAINTENANCE REGISTER 3
003532  000000          E.ECPS: .WORD   0               ;ECC POSITION INFORMATION
003534  000000          E.ECPT: .WORD   0               ;ECC PATTERN INFORMATION
003536  000000          E.SPAR: .WORD   0               ;SPARE REGISTER

                                .SBTTL  PREVIOUS RK611 CONTROLLER REGISTERS

003540  000000          P.CS1:  .WORD   0               ;CONTROL AND STATUS REGISTER 1
003542  000000          P.WC:   .WORD   0               ;WORD COUNT REGISTER
003544  000000          P.BA:   .WORD   0               ;BUS ADDRESS REGISTER
003546  000000          P.DA:   .WORD   0               ;DESIRED TRACK SECTOR REGISTER
003550  000000          P.CS2:  .WORD   0               ;CONTROL AND STATUS REGISTER 2
003552  000000          P.DS:   .WORD   0               ;DRIVE STATUS REGISTER
003554  000000          P.ER:   .WORD   0               ;ERROR REGISTER
003556  000000          P.ASOF: .WORD   0               ;ATTENTION SUMMARY AND OFFSET REGISTER
003560  000000          P.DCYL: .WORD   0               ;DESIRED CYLINDER REGISTER
003562  000000          P.DB:   .WORD   0               ;DATA BUFFER
003564  000000          P.MR1:  .WORD   0               ;MAINTENANCE REGISTER 1
003566  000000          P.MR2:  .WORD   0               ;MAINTENANCE REGISTER 2
003570  000000          P.MR3:  .WORD   0               ;MAINTENANCE REGISTER 3
003572  000000          P.ECPS: .WORD   0               ;ECC POSITION INFORMATION
003574  000000          P.ECPT: .WORD   0               ;ECC PATTERN INFORMATION
003576  000000          P.SPAR: .WORD   0               ;SPARE REGISTER
```

```
2236                                    .SBTTL  PROGRAM DEFINED VARIABLES
2237
2238  003600  000210                    RKVEC:  .WORD   210             ;RK611 VECTOR
2239  003602  000240                    RKPRI:  .WORD   PR5             ;RK611 PRIORITY
2240  003604  000000                    TRAPPC: .WORD   0               ;PC FOR MEMORY CHECK ENABLE TRAP
2241  003606  000000                    SRTFLG: .WORD   0               ;START FLAG
2242                                                                    ;  0 = 200
2243                                                                    ;  1 = 214
2244                                                                    ; -1 = 204
2245  003610  000000                    ERRCNT: .WORD   0               ;ERROR COUNT FOR SWITCH 12 ABORT
2246  003612  000000                    P1.BIT: .WORD   0               ;NEXT BIT IN DATA SIMULATION
2247  003614  000000                    PR.BIT: .WORD   0               ;PRESENT BIT IN DATA SIMULATION
2248  003616  000000                    M1.BIT: .WORD   0               ;PREVIOUS BIT IN DATA SIMULATION
2249  003620  000000                    M2.BIT: .WORD   0               ;BIT BEFORE PREVIOUS BIT
2250  003622  000000                    BITCNT: .WORD   0               ;BIT POSITION
2251  003624  000000                    WRDCNT: .WORD   0               ;WORD COUNT FOR NPR TRANSFER
2252  003626  000000                    SECCNT: .WORD   0               ;SECTOR COUNT
2253  003630  000000                    MEMPAR: .WORD   0               ;MEMORY EMABLE ON FIRST 24K
2254  003632  000015                    WAITIM: .WORD   15              ;WAIT TIME FOR CONTROLLER READY
2255  003634  000000                    SAVSWR: .WORD   0               ;STORAGE FOR SWITCH REGISTER
```

```
                                        .SBTTL  PROGRAM SETUP
                                
003636  012737  000001  003606  PARM:   MOV     #1,SRTFLG       ;LOAD START FLAG FOR PARMETER START
003644  000406                          BR      START1
                                
003646  012737  177777  003606  RESTRT: MOV     #-1,SRTFLG      ;LOAD START FLAG FOR RESTART
003654  000402                          BR      START1
                                
003656  005037  003606          START:  CLR     SRTFLG          ;CLEAR START FLAG
003662  000005                  START1: RESET                   ;RESET THE WHOLE SYSTEM
003664  012706  001100                  MOV     #STACK,SP       ;INITIALIZE STACK POINTER
003670  004737  051224                  JSR     PC,$TKINT       ;INIT KEYBOARD
003674  012746  000340                  MOV     #PR7,-(SP)      ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
003700  012746  003706                  MOV     #1$,-(SP)       ;LOAD START OF PROGRAM
003704  000002                          RTI                     ;LOAD PSW
                                
003706                          1$:
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
003706  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
003712  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
003714  022706  001140                  CMP     #SWR,R6 ;;DONE?
003720  001374                          BNE     .-6             ;;LOOP BACK IF NO
003722  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
003726  012737  046566  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
003734  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
003742  012737  047520  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
003750  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
003756  012737  053042  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
003764  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
003772  012737  052706  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
004000  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
004006  013737  044526  044520          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
004014  005037  001200                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
004020  005037  001202                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
004024  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
004032  012737  004032  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
004040  012737  004040  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
004046  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
004052  012737  004106  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
004060  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
004066  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
004074  022777  177777  175036          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
004102  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
004104  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
004106  012716  004114          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
004112  000002                          RTI
004114  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
004122  012737  000174  001142          MOV     #DISPREG,DISPLAY
004130  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
                                
004134  005037  001222                  CLR     $PASS           ;;CLEAR PASS COUNT
```

```
2312  004140  132737  000200  001235          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2313  004146  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
2314  004150  012737  001236  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
2315  004156                          67$:
2316  004156  005037  003610                  CLR     ERRCNT          ;CLEAR ERROR COUNT FOR SWITCH 12 ABORT
2317                                  .SBTTL  TYPE PROGRAM NAME
2318                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2319  004162  005227  177777                  INC     #-1             ;;FIRST TIME?
2320  004166  001063                          BNE     68$             ;;BRANCH IF NO
2321  004170  022737  044662  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
2322  004176  001457                          BEQ     68$             ;;BRANCH IF YES
2323  004200  104401  004246                  TYPE    ,69$            ;;TYPE ASCIZ STRING
2324                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2325  004204  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2326  004210  001012                          BNE     70$             ;;BRANCH IF YES
2327  004212  123727  001234  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
2328  004220  001406                          BEQ     70$             ;;BRANCH IF YES
2329  004222  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2330  004230  001005                          BNE     71$             ;;BRANCH IF NO
2331  004232  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
2332  004234  000403                          BR      71$
2333  004236  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2334  004244                          71$:
2335  004244  000434                          BR      68$             ;;GET OVER THE ASCIZ
2336                                  ;;69$:  .ASCIZ  <CRLF>/RK611 DISKLESS DIAGNOSTIC: PART 3  MAINDEC-11-DZR6CA/<CRLF>
2337  004336                          68$:
2338  004336  005227  177777                  INC     #-1             ;TEST IF FIRST PASS
2339  004342  001002                          BNE     6$              ;NO - SKIP
2340  004344  104401  054526                  TYPE    ,OPR006         ;TYPE RUN TIME MESSAGE
2341  004350  022737  000001  003606  6$:     CMP     #1,SRTFLG       ;CHECK IF PARAMETER START
2342  004356  001122                          BNE     15$             ;NO, CONTINUE SETUP
2343  004360  104401  054414          5$:     TYPE    ,OPR001         ;TYPE "RK611 BUS ADDRESS (  ) ="
2344  004364  013746  001270                  MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
2345  004370  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2346  004372  104401  054446                  TYPE    ,OPR002
2347  004376  104412                          RDOCT                   ;GET VALUE
2348  004400  012637  001160                  MOV     (SP)+,$TMPO
2349  004404  001407                          BEQ     7$              ;CHECK IF <CR>
2350  004406  022737  160000  001160          CMP     #160000,$TMPO   ;CHECK IF IN I/O PAGE
2351  004414  101361                          BHI     5$
2352  004416  013737  001160  001270          MOV     $TMPO,$BASE     ;LOAD NEW BUS ADDRESS
2353  004424  104401  054454          7$:     TYPE    ,OPR003         ;TYPE "RK611 VECTOR ADDRESS (  ) ="
2354  004430  013746  001264                  MOV     $VECT1,-(SP)    ;TYPE OUT VECTOR ADDRESS
2355  004434  042716  160000                  BIC     #160000,(SP)
2356  004440  104402                          TYPOC
2357  004442  104401  054446                  TYPE    ,OPR002
2358  004446  104412                          RDOCT                   ;GET VALUE
2359  004450  012637  001160                  MOV     (SP)+,$TMPO
2360  004454  001412                          BEQ     10$             ;CHECK IF <CR>
2361  004456  022737  001000  001160          CMP     #1000,$TMPO     ;CHECK IF LEGAL
2362  004464  101757                          BLOS    7$
2363  004466  042737  017777  001264          BIC     #17777,$VECT1   ;LOAD NEW VECTOR ADDRESS
2364  004474  053737  001160  001264          BIS     $TMPO,$VECT1
2365  004502  104401  054504          10$:    TYPE    ,OPR004         ;TYPE "RK611 PRIORITY (  ) ="
2366  004506  005046                          CLR     -(SP)
2367  004510  113716  001265                  MOVB    $VECT1+1,(SP)
```

```
2368  004514  006216                        ASR     (SP)            ;SHIFT 5 BITS RIGHT
2369  004516  006216                        ASR     (SP)
2370  004520  006216                        ASR     (SP)
2371  004522  006216                        ASR     (SP)
2372  004524  006216                        ASR     (SP)
2373  004526  104402                        TYPOC
2374  004530  104401  054446                TYPE    ,OPR002
2375  004534  104412                        RDOCT                   ;GET VALUE
2376  004536  012637  001160                MOV     (SP)+,$TMP0
2377  004542  001430                        BEQ     15$             ;CHECK FOR DEFAULT
2378  004544  022737  000007  001160        CMP     #7,$TMP0        ;CHECK IF LEGAL
2379  004552  103753                        BLO     10$
2380  004554  022737  000004  001160        CMP     #4,$TMP0
2381  004562  101347                        BHI     10$
2382  004564  006337  001160                ASL     $TMP0           ;SHIFT 5 BITS LEFT
2383  004570  006337  001160                ASL     $TMP0
2384  004574  006337  001160                ASL     $TMP0
2385  004600  006337  001160                ASL     $TMP0
2386  004604  006337  001160                ASL     $TMP0
2387  004610  042737  160000  001264        BIC     #160000,$VECT1  ;STORE NEW PRIORITY
2388  004616  153737  001160  001265        BISB    $TMP0,$VECT1+1
2389  004624  013737  001264  003600  15$:   MOV     $VECT1,RKVEC    ;STORE RK611 VECTOR
2390  004632  042737  160000  003600        BIC     #160000,RKVEC
2391  004640  113737  001265  003602        MOVB    $VECT1+1,RKPRI  ;STORE PRIORITY
2392  004646  004737  046260                JSR     PC,$SIZE        ;SIZE MEMORY
2393  004652  013702  001270                MOV     $BASE,R2        ;SET RK611 BASE
2394  004656  005037  001202                CLR     $ESCAPE         ;CLEAR ESCAPE
2395
2396  004662  004737  045002        NEWPAS: JSR     PC,PARCHK       ;CHECK OF MEMORY CHECK ENABLE
2397  004666  012746  000000                MOV     #PR0,-(SP)      ;ALLOW ALL INTERRUPTS
2398  004672  012746  004700                MOV     #TST1,-(SP)
2399  004676  000002                        RTI
```

```
                                .SBTTL  **DRIVE MESSAGES FOR CLASS B INSTRUCTIONS

                                ;;*****************************************************************
                                ;*TEST 1         READ HEADER SEEK MESSAGE
                                ;*
                                ;*      CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
                                ;*      PUT THE RK611 CONTROLLER IN DIAGNOSTIC MODE.  ISSUE
                                ;*      A READ HEADER TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0.
                                ;*      HEAD 0, DRIVE 0. CLOCK IN SEEK MESSAGE INTO SHIFT REGISTER.
                                ;*      VERIFY THAT A SEEK IS LOADED WITH THE PROPER BITS IN
                                ;*      MESSAGE SET.  REPEAT FOR A READ HEADER WITH CDT SET
                                ;*      IN 24 SECTOR FORMAT, CYLINDER 1777, HEAD 7, DRIVE 7.
                                ;*
                                ;;*****************************************************************
004700  000004                  TST1:   SCOPE
004702  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
004710  013702  001270          MOV     $BASE,R2        ;LOAD RK611 BASE
004714  012762  100000  000000  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
004722  012762  000040  000026  MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
004730  012762  001777  000020  MOV     #1777,RKDCYL(R2)  ;LOAD CYLINDER ADDRESS REG.
004736  012762  003400  000006  MOV     #3400,RKDA(R2)  ;LOAD TRACK
004744  012762  000007  000010  MOV     #7,RKCS2(R2)    ;LOAD DRIVE NUM.
004752  012762  012025  000000  MOV     #CDT!CFMT!RDHEAD,RKCS1(R2)  ;ISSUE RDHEAD WITH CDT SET IN
                                                        ;  24 SECTOR FORMAT
004760  012700  000016          MOV     #3*4+2,R0       ;CLOCK IN DRIVE MESSAGE
004764  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
004772  012762  000040  000026          MOV     #DMD,RKMR1(R2)
005000  005300                  DEC     R0
005002  001370                  BNE     1$
005004  016237  000000  003440  MOV     RKCS1(R2),T.CS1 ;STORE COMMAND STATUS REG. 1
005012  016237  000034  003466  MOV     RKMR2(R2),T.MR2 ;STORE MAINT REG. 2 (MESS A)
005020  016237  000036  003470  MOV     RKMR3(R2),T.MR3 ;STORE MAINT REG. 3 (MESS B)
005026  012737  012025  003500  MOV     #CDT!CFMT!RDHEAD,E.CS1  ;LOAD EXPECTED CS1
005034  012737  071027  003526  MOV     #S.SEEK!S.FMT!70007,E.MR2  ;LOAD EXPECTED MR2
005042  012737  037760  003530  MOV     #37760,E.MR3    ;LOAD EXPECTED MR3
005050  023737  003500  003440  CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
005056  001405                  BEQ     2$              ;YES, CHECK MESSAGE A
005060  104001                  ERROR   1               ;CS1 INCORRECT
005062  012762  100000  000000  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
005070  000412                  BR      TST2            ;;GO TO NEXT TEST
005072  023737  003526  003466  2$:     CMP     E.MR2,T.MR2     ;CHECK MESS A CORRECT
005100  001401                  BEQ     3$              ;YES, CHECK MESSAGE B
005102  104002                  ERROR   2               ;MESS A INCORRECT
005104  023737  003530  003470  3$:     CMP     E.MR3,T.MR3     ;CHECK MESS B CORRECT
005112  001401                  BEQ     TST2            ;;YES, GO ON TO NEXT TEST
005114  104003                  ERROR   3               ;MESS B INCORRECT

                                ;;*****************************************************************
                                ;*TEST 2         WRITE HEADER SEEK MESSAGE
                                ;*
                                ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                                ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                                ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                ;*      CLOCK IN SEEK MESSAGE INTO SHIFT REGISTER.  VERIFY
                                ;*      THAT A SEEK IS LOADED WITH THE RTC BIT SET.  REPEAT
                                ;*      FOR A WRITE HEADER WITH CDT SET IN 24 SECTOR FORMAT,
```

```
2456                                    ;*      CYLINDER 1777, HEAD 7, DRIVE 7.
2457                                    ;*
2458                                    ;;*****************************************************************
2459  005116  000004                    TST2:   SCOPE
2460  005120  012737  000144  001200            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
2461  005126  013702  001270            .       MOV     $BASE,R2        ;LOAD RK611 BASE
2462  005132  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2463  005140  012762  000040  000026            MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
2464  005146  012762  001777  000020            MOV     #1777,RKDCYL(R2)  ;LOAD CYLINDER ADDRESS REG.
2465  005154  012762  003400  000006            MOV     #3400,RKDA(R2)  ;LOAD TRACK
2466  005162  012762  000007  000010            MOV     #7,RKCS2(R2)    ;LOAD DRIVE NUM.
2467  005170  012762  012027  000000            MOV     #CDT!CFMT!WRHEAD,RKCS1(R2)  ;ISSUE WRHEAD WITH CDT SET IN
2468                                                                    ;  24 SECTOR FORMAT
2469  005176  012700  000016                    MOV     #3*4+2,R0       ;CLOCK IN DRIVE MESSAGE
2470  005202  012762  000440  000026    1$:     MOV     #DMD!MCLK,RKMR1(R2)
2471  005210  012762  000040  000026            MOV     #DMD,RKMR1(R2)
2472  005216  005300                            DEC     R0
2473  005220  001370                            BNE     1$
2474  005222  016237  000000  003440            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND STATUS REG. 1
2475  005230  016237  000034  003466            MOV     RKMR2(R2),T.MR2 ;STORE MAINT REG. 2 (MESS A)
2476  005236  016237  000036  003470            MOV     RKMR3(R2),T.MR3 ;STORE MAINT REG. 3 (MESS B)
2477  005244  012737  012027  003500            MOV     #CDT!CFMT!WRHEAD,E.CS1  ;LOAD EXPECTED CS1
2478  005252  012737  071227  003526            MOV     #S.SEEK!S.RTC!S.FMT!70007,E.MR2  ;LOAD EXPECTED MR2
2479  005260  012737  037760  003530            MOV     #37760,E.MR3    ;LOAD EXPECTED MR3
2480  005266  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
2481  005274  001405                            BEQ     2$              ;YES, CHECK MESSAGE A
2482  005276  104004                            ERROR   4               ;CS1 INCORRECT
2483  005300  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2484  005306  000412                            BR      TST3            ;;GO TO NEXT TEST
2485  005310  023737  003526  003466    2$:     CMP     E.MR2,T.MR2     ;CHECK MESS A CORRECT
2486  005316  001401                            BEQ     3$              ;YES, CHECK MESSAGE B
2487  005320  104005                            ERROR   5               ;MESS A INCORRECT
2488  005322  023737  003530  003470    3$:     CMP     E.MR3,T.MR3     ;CHECK MESS B CORRECT
2489  005330  001401                            BEQ     TST3            ;;YES, GO ON TO NEXT TEST
2490  005332  104006                            ERROR   6               ;MESS B INCORRECT
2491
2492                                    ;;*****************************************************************
2493                                    ;*TEST 3        READ HEADER DRIVE CLEAR MESSAGE
2494                                    ;*
2495                                    ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
2496                                    ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER WITH CDT SET
2497                                    ;*      IN 24 SECTOR FORMAT, CYLINDER 1777, HEAD 7, DRIVE 7.
2498                                    ;*      CLOCK SEEK MESSAGE AND MAKE SURE A DRIVE CLEAR IS
2499                                    ;*      GENERATED AND THE PROPER BITS ARE SET.
2500                                    ;*
2501                                    ;;*****************************************************************
2502  005334  000004                    TST3:   SCOPE
2503  005336  012737  000144  001200            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
2504  005344  013702  001270            .       MOV     $BASE,R2        ;LOAD RK611 BASE
2505  005350  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2506  005356  012762  000040  000026            MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
2507  005364  012762  001777  000020            MOV     #1777,RKDCYL(R2)  ;LOAD CYLINDER ADDRESS REG.
2508  005372  012762  003400  000006            MOV     #3400,RKDA(R2)  ;LOAD TRACK
2509  005400  012762  000007  000010            MOV     #7,RKCS2(R2)    ;LOAD DRIVE NUMBER
2510  005406  012762  012025  000000            MOV     #CDT!CFMT!RDHEAD,RKCS1(R2)  ;ISSUE COMMAND WITH CDT SET IN
2511                                                                    ;  24 SECTOR FORMAT
```

```
2512  005414  012700  000156                  MOV     #27.*4+2,R0     ;LOAD COUNT TO LOAD DRIVE CLEAR
2513  005420  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
2514  005426  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2515  005434  005300                          DEC     R0
2516  005436  001370                          BNE     1$
2517  005440  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2518  005446  016237  000034  003466          MOV     RKMR2(R2),T.MR2 ;STORE MAINT. REG. 2 (MESS A)
2519  005454  016237  000036  003470          MOV     RKMR3(R2),T.MR3 ;STORE MAINT. REG. 3 (MESS B)
2520  005462  012737  012025  003500          MOV     #CDT!CFMT!RDHEAD,E.CS1  ;LOAD EXPECTED CS1
2521  005470  012737  071407  003526          MOV     #S.CLR!S.FMT!70007,E.MR2  ;LOAD EXPECTED MAINT REG. 2
2522  005476  005037  003530                  CLR     E.MR3           ;LOAD EXPECTED MAINT REG.
2523  005502  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG 1 CORRECT
2524  005510  001405                          BEQ     2$              ;YES, CHECK CS2
2525  005512  104007                          ERROR   7               ;CS1 INCORRECT
2526  005514  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2527  005522  000412                          BR      TST4            ;;GO TO NEXT TEST
2528  005524  023737  003526  003466  2$:     CMP     E.MR2,T.MR2     ;CHECK MESS A CORRECT
2529  005532  001401                          BEQ     3$              ;YES, CHECK MESS B
2530  005534  104010                          ERROR   10              ;MESS A INCORRECT
2531  005536  023737  003530  003470  3$:     CMP     E.MR3,T.MR3     ;CHECK MESS B CORRECT
2532  005544  001401                          BEQ     TST4            ;;YES, GO ON TO NEXT TEST
2533  005546  104011                          ERROR   11              ;MESS B INCORRECT
2534
2535                                  ;;*****************************************************************
2536                                  ;*TEST 4        WRITE HEADER DRIVE CLEAR MESSAGE
2537                                  ;*
2538                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR. PUT CONTROLLER
2539                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER WITH CDT SET
2540                                  ;*      IN 24 SECTOR FORMAT, CYLINDER 1777, HEAD 7, DRIVE 7.
2541                                  ;*      CLOCK SEEK MESSAGE AND LOAD GENERATED DRIVE CLEAR
2542                                  ;*      INTO SHIFT REGISTER.  MAKE SURE THE DRIVE CLEAR IS
2543                                  ;*      GENERATED AND THE PROPER BITS ARE SET.
2544                                  ;*
2545                                  ;*****************************************************************
2546  005550  000004          TST4:   SCOPE
2547  005552  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
2548  005560  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
2549  005564  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2550  005572  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
2551  005600  012762  001777  000020          MOV     #1777,RKDCYL(R2)  ;LOAD CYLINDER ADDRESS REG.
2552  005606  012762  003400  000006          MOV     #3400,RKDA(R2)  ;LOAD TRACK
2553  005614  012762  000007  000010          MOV     #7,RKCS2(R2)    ;LOAD DRIVE NUMBER
2554  005622  012762  012027  000000          MOV     #CDT!CFMT!WRHEAD,RKCS1(R2)  ;ISSUE COMMAND WITH CDT SET IN
2555                                                                  ;  24 SECTOR FORMAT
2556  005630  012700  000156                  MOV     #27.*4+2,R0     ;LOAD COUNT TO LOAD DRIVE CLEAR
2557  005634  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
2558  005642  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2559  005650  005300                          DEC     R0
2560  005652  001370                          BNE     1$
2561  005654  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2562  005662  016237  000034  003466          MOV     RKMR2(R2),T.MR2 ;STORE MAINT. REG. 2 (MESS A)
2563  005670  016237  000036  003470          MOV     RKMR3(R2),T.MR3 ;STORE MAINT. REG. 3 (MESS B)
2564  005676  012737  012027  003500          MOV     #CDT!CFMT!WRHEAD,E.CS1  ;LOAD EXPECTED CS1
2565  005704  012737  071407  003526          MOV     #S.CLR!S.FMT!70007,E.MR2  ;LOAD EXPECTED MAINT REG. 2
2566  005712  005037  003530                  CLR     E.MR3           ;LOAD EXPECTED MAINT REG.
2567  005716  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG 1 CORRECT
```

```
2568  005724  001405                          BEQ     2$              ;YES, CHECK CS2
2569  005726  104012                          ERROR   12              ;CS1 INCORRECT
2570  005730  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2571  005736  000412                          BR      TST5            ;;GO TO NEXT TEST
2572  005740  023737  003526  003466  2$:     CMP     E.MR2,T.MR2     ;CHECK MESS A CORRECT
2573  005746  001401                          BEQ     3$              ;YES, CHECK MESS B
2574  005750  104013                          ERROR   13              ;MESS A INCORRECT
2575  005752  023737  003530  003470  3$:     CMP     E.MR3,T.MR3     ;CHECK MESS B CORRECT
2576  005760  001401                          BEQ     TST5            ;;YES, GO ON TO NEXT TEST
2577  005762  104014                          ERROR   14              ;MESS B INCORRECT
2578
2579
2580                                  .SBTTL  **INDEX AND SECTOR PULSE DETECT ON
2581
2582                                  ;;*****************************************************************
2583                                  ;*TEST 5        SECTOR PULSE DETECT IN READ HEADER (PART 1)
2584                                  ;*
2585                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
2586                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
2587                                  ;*      IN 26 SECTOR MODE, CYLINDER 0, HEAD 0, DRIVE 0.
2588                                  ;*      CLOCK BOTH SEEK ANY DRIVE CLEAR MESSAGES.
2589                                  ;*      SIMULATE SECTOR PULSE, 255 ZEROES AND A ONE.
2590                                  ;*
2591                                  ;*      MAKE SURE READ GATE DOES SET.
2592                                  ;*
2593                                  ;;*****************************************************************
2594  005764  000004                  TST5:   SCOPE
2595  005766  012737  000012  001200          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
2596  005774  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
2597  006000  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2598  006006  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
2599  006014  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)  ;INITIALIZE ROM ADDRESS
2600  006022  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2601  006030  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
2602  006036  012700  000312                  MOV     #50.*4+2,R0     ;CLOCK UNTIL READY FOR SECTOR PULSE
2603  006042  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
2604  006050  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2605  006056  005300                          DEC     R0
2606  006060  001370                          BNE     1$
2607  006062  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2608  006070  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
2609  006076  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
2610  006104  001405                          BEQ     2$              ;YES, CLOCK ZEROES
2611  006106  104015                          ERROR   15              ;CS1 INCORRECT
2612  006110  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2613  006116  000553                          BR      TST6            ;;GO ON TO NEXT TEST
2614
2615  006120  012762  000140  000026  2$:     MOV     #DMD!MSP,RKMR1(R2)  ;SIMULATE SECTOR PULSE
2616  006126  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2617  006134  012737  022040  003524          MOV     #DMD!MEWD!ECCW,E.MR1  ;LOAD EXPECT MAINT REG. 1
2618  006142  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
2619  006146  005037  003614                  CLR     PR.BIT          ;INITIALIZE PRESENT AND PREVIOUS
2620  006152  005037  003616                  CLR     M1.BIT          ;  BITS TO GENERATE ZEROES
2621  006156  012700  000200                  MOV     #128.,R0        ;GENERATE 128 ZEROS UNTIL READ GATE
2622  006162  004737  046106          5$:     JSR     PC,RDBIT        ;READ A ZERO
2623  006166  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINT REG. 1
```

```
2624  006174  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINTENANCE REG. 1 CORRECT
2625  006202  001405                          BEQ     6$              ;YES, SIMULATE NEXT BIT
2626  006204  104017                          ERROR   17              ;MAINT REG. 1 INCORRECT
2627  006206  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2628  006214  000514                          BR      TST6            ;;GO ON TO NEXT TEST
2629
2630  006216  005237  003622          6$:     INC     BITCNT          ;INCREMENT BIT COUNT
2631  006222  005300                          DEC     R0              ;CHECK READY OF READ GATE
2632  006224  001356                          BNE     5$              ;NO, CONTINUE
2633  006226  012737  122040  003524          MOV     #DMD!MEWD!ECCW!RDGATE,E.MR1  ;LOAD EXPECTED MR1
2634  006234  012700  000177                  MOV     #127.,R0        ;GENERATE 127 ZEROS
2635  006240  004737  046106          10$:    JSR     PC,RDBIT        ;READ A ZERO
2636  006244  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINT REG. 1
2637  006252  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINT REG. 1 CORRECT
2638  006260  001405                          BEQ     11$             ;YES, SIMULATE NEXT BIT
2639  006262  104017                          ERROR   17              ;MAINT REG. 1 INCORRECT
2640  006264  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2641  006272  000465                          BR      TST6            ;;GO ON TO NEXT TEST
2642
2643  006274  005237  003622          11$:    INC     BITCNT          ;INCREMENT BIT COUNT
2644  006300  005300                          DEC     R0              ;CHECK IF ALL ZEROS ISSUED
2645  006302  001356                          BNE     10$             ;NO, CONTINUE
2646  006304  012737  000001  003614          MOV     #1,PR.BIT       ;LOAD ONE FOR READING 1
2647  006312  004737  046106                  JSR     PC,RDBIT        ;READ A ONE
2648  006316  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG.
2649  006324  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINT REG. 1
2650  006332  001405                          BEQ     12$             ;YES, CONTINUE
2651  006334  104017                          ERROR   17              ;MAINTENANCE REG. 1 INCORRECT
2652  006336  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2653  006344  000440                          BR      TST6            ;;GO ON TO NEXT TEST
2654  006346  005237  003622          12$:    INC     BITCNT          ;INCREMENT BIT COUNT
2655  006352  013737  003614  003616          MOV     PR.BIT,M1.BIT   ;LOAD ZERO FOR NEXT BIT
2656  006360  005037  003614                  CLR     PR.BIT
2657  006364  004737  046106                  JSR     PC,RDBIT        ;SIMULATE ZERO
2658  006370  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
2659  006376  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINT REG. 1 CORRECT
2660  006404  001405                          BEQ     13$             ;CHECK CS1 CORRECT
2661  006406  104017                          ERROR   17              ;MAINTENANCE REG. 1 INCORRECT
2662  006410  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2663  006416  000413                          BR      TST6            ;;GO TO NEXT TEST
2664
2665  006420  016237  000000  003440  13$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2666  006426  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
2667  006434  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
2668  006442  001401                          BEQ     TST6            ;;YES, GO TO NEXT TEST
2669  006444  104016                          ERROR   16              ;CS1 INCORRECT
2670
2671                          ;;*****************************************************************
2672                          ;*TEST 6        SECTOR PULSE DETECT IN READ HEADER (PART 2)
2673                          ;*
2674                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
2675                          ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
2676                          ;*      IN 26 SECTOR MODE, CYLINDER 0, HEAD 0, DRIVE 0.
2677                          ;*      CLOCK BOTH SEEK ANY DRIVE CLEAR MESSAGES.
2678                          ;*      SIMULATE INDEX PULSE, 255 ZEROES AND A ONE.
2679                          ;*
```

```
2680                                    ;*      MAKE SURE READ GATE DOES NOT SET.
2681                                    ;*
2682                                    ;;*****************************************************************
2683  006446  000004                    TST6:   SCOPE
2684  006450  012737  000012  001200            MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
2685  006456  013702  001270                    MOV     $BASE,R2        ;LOAD RK611 BASE
2686  006462  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2687  006470  012762  000040  000000            MOV     #DMD,RKCS1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
2688  006476  012762  000140  000026            MOV     #DMD!MSP,RKMR1(R2)  ;INITIALIZE ROM ADDRESS
2689  006504  012762  000040  000026            MOV     #DMD,RKMR1(R2)
2690  006512  012762  000025  000000            MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
2691  006520  012700  000312                    MOV     #50.*4+2,R0     ;CLOCK UNTIL READY FOR SECTOR PULSE
2692  006524  012762  000440  000026  1$:       MOV     #DMD!MCLK,RKMR1(R2)
2693  006532  012762  000040  000026            MOV     #DMD,RKMR1(R2)
2694  006540  005300                            DEC     R0
2695  006542  001370                            BNE     1$
2696  006544  016237  000000  003440            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2697  006552  012737  000025  003500            MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
2698  006560  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
2699  006566  001405                            BEQ     2$              ;YES, CLOCK IN ZEROS
2700  006570  104015                            ERROR   15
2701  006572  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2702  006600  000535                            BR      TST7            ;;GO ON TO NEXT TEST
2703  006602                            2$:
2704  006602  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE INDX PULSE
2705  006610  012700  000004                    MOV     #4,R0
2706  006614  012762  000640  000026  3$:       MOV     #DMD!MIND!MCLK,RKMR1(R2)
2707  006622  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)
2708  006630  005300                            DEC     R0
2709  006632  001370                            BNE     3$
2710  006634  012762  000040  000026            MOV     #DMD,RKMR1(R2)
2711  006642  005037  003622                    CLR     BITCNT          ;INITIALIZE BIT COUNT
2712  006646  012737  022040  003524            MOV     #DMD!MEWD!ECCW,E.MR1  ;LOAD EXPECTED MAINTENANCE REG. 1
2713  006654  005037  003614                    CLR     PR.BIT          ;INITIALIZE PRESENT AND PREVIOUS
2714  006660  005037  003616                    CLR     M1.BIT          ;   BITS TO GENERATE ZEROS
2715  006664  012700  000377                    MOV     #255.,R0        ;GENERATE 255 ZEROES
2716  006670  004737  046106            5$:     JSR     PC,RDBIT        ;READ A ZERO
2717  006674  016237  000026  003464            MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
2718  006702  023737  003524  003464            CMP     E.MR1,T.MR1     ;CHECK READ GATE NOT SET
2719  006710  001405                            BEQ     6$              ;READ GATE NOT SET SIMULATE NEXT BIT
2720  006712  104020                            ERROR   20              ;MAINT REG. 1 INCORRECT
2721  006714  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
2722  006722  000464                            BR      TST7            ;;GO ON TO NEXT TEST
2723
2724  006724  005237  003622            6$:     INC     BITCNT          ;INCREMENT BIT COUNT
2725  006730  005300                            DEC     R0              ;CHECK IF ALL ZEROES ISSUED
2726  006732  001356                            BNE     5$              ;NO, CONTINUE
2727  006734  012737  000001  003614            MOV     #1,PR.BIT       ;LOAD ONE FOR READING 1
2728  006742  004737  046106                    JSR     PC,RDBIT        ;READ A ONE
2729  006746  016237  000026  003464            MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
2730  006754  023737  003524  003464            CMP     E.MR1,T.MR1     ;CHECK READ GATE NOT SET
2731  006762  001405                            BEQ     7$              ;YES, CONTINUE
2732  006764  104020                            ERROR   20              ;MAINT REG. 1 INCORRECT
2733  006766  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
2734  006774  000437                            BR      TST7            ;;GO ON TO NEXT TEST
2735
```

```
2736  006776  005237  003622          7$:     INC     BITCNT          ;INCREMENT BIT COUNT
2737  007002  013737  003614  003616          MOV     PR.BIT,M1.BIT   ;LOAD ZERO FOR NEXT BIT
2738  007010  005037  003614                  CLR     PR.BIT
2739  007014  004737  046106                  JSR     PC,RDBIT        ;SIMULATE ZERO
2740  007020  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
2741  007026  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINTENANCE REG. 1 CORRECT
2742  007034  001404                          BEQ     9$              ;CHECK CS1 CORRECT
2743  007036  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2744  007044  000413                          BR      TST7            ;;GO TO NEXT TEST
2745
2746  007046  016237  000000  003440  9$:     MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2747  007054  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
2748  007062  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
2749  007070  001401                          BEQ     TST7            ;;YES, GO TO NEXT TEST
2750  007072  104016                          ERROR   16              ;CS1 INCORRECT
2751
2752                                  ;;*****************************************************************
2753                                  ;*TEST 7        SECTOR PULSE DETECT IN READ HEADER (PART 3)
2754                                  ;*
2755                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
2756                                  ;*      DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
2757                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
2758                                  ;*      CLOCK BOTH SEEK AND DRIVE CHECK MESSAGES.
2759                                  ;*      SIMULATE 255 ZEROES AND A ONE.
2760                                  ;*
2761                                  ;*      MAKE SURE READ GATE DOES NOT SET.
2762                                  ;*
2763                                  ;;*****************************************************************
2764  007074  000004                  TST7:   SCOPE
2765  007076  012737  000012  001200          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
2766  007104  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
2767  007110  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2768  007116  012762  000040  000000          MOV     #DMD,RKCS1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
2769  007124  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)  ;INITIALIZE ROM ADDRESS
2770  007132  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2771  007140  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
2772  007146  012700  000312                  MOV     #50.*4+2,R0     ;CLOCK UNTIL READY FOR SECTOR PULSE
2773  007152  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
2774  007160  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2775  007166  005300                          DEC     R0
2776  007170  001370                          BNE     1$
2777  007172  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2778  007200  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
2779  007206  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
2780  007214  001405                          BEQ     2$              ;YES, CLOCK IN ZEROS
2781  007216  104015                          ERROR   15
2782  007220  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2783  007226  000515                          BR      TST10           ;;GO ON TO NEXT TEST
2784  007230                          2$:
2785  007230  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
2786  007234  012737  022040  003524          MOV     #DMD!MEWD!ECCW,E.MR1  ;LOAD EXPECTED MAINTENANCE REG. 1
2787  007242  005037  003614                  CLR     PR.BIT          ;INITIALIZE PRESENT AND PREVIOUS
2788  007246  005037  003616                  CLR     M1.BIT          ;  BITS TO GENERATE ZEROS
2789  007252  012700  000377                  MOV     #255.,R0        ;GENERATE 255 ZEROES
2790  007256  004737  046106          5$:     JSR     PC,RDBIT        ;READ A ZERO
2791  007262  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
```

```
2792  007270  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK READ GATE NOT SET
2793  007276  001405                          BEQ     6$              ;READ GATE NOT SET SIMULATE NEXT BIT
2794  007300  104021                          ERROR   21              ;MAINT REG. 1 INCORRECT
2795  007302  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
2796  007310  000464                          BR      TST10           ;;GO ON TO NEXT TEST
2797
2798  007312  005237  003622          6$:     INC     BITCNT          ;INCREMENT BIT COUNT
2799  007316  005300                          DEC     R0              ;CHECK IF ALL ZEROES ISSUED
2800  007320  001356                          BNE     5$              ;NO, CONTINUE
2801  007322  012737  000001  003614          MOV     #1,PR.BIT       ;LOAD ONE FOR READING 1
2802  007330  004737  046106                  JSR     PC,RDBIT        ;READ A ONE
2803  007334  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
2804  007342  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK READ GATE NOT SET
2805  007350  001405                          BEQ     7$              ;YES, CONTINUE
2806  007352  104021                          ERROR   21              ;MAINT REG. 1 INCORRECT
2807  007354  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;ISSUE CONTROLLER CLEAR
2808  007362  000437                          BR      TST10           ;;GO ON TO NEXT TEST
2809
2810  007364  005237  003622          7$:     INC     BITCNT          ;INCREMENT BIT COUNT
2811  007370  013737  003614  003616          MOV     PR.BIT,M1.BIT   ;LOAD ZERO FOR NEXT BIT
2812  007376  005037  003614                  CLR     PR.BIT
2813  007402  004737  046106                  JSR     PC,RDBIT        ;SIMULATE ZERO
2814  007406  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG. 1
2815  007414  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINTENANCE REG. 1 CORRECT
2816  007422  001404                          BEQ     9$              ;CHECK CS1 CORRECT
2817  007424  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2818  007432  000413                          BR      TST10           ;;GO TO NEXT TEST
2819
2820  007434  016237  000000  003440  9$:     MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2821  007442  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
2822  007450  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
2823  007456  001401                          BEQ     TST10           ;;YES, GO TO NEXT TEST
2824  007460  104016                          ERROR   16              ;CS1 INCORRECT
2825
2826                                  ;;*****************************************************************
2827                                  ;*TEST 10       INDEX PULSE DETECTION IN WRITE HEADER
2828                                  ;*
2829                                  ;*      CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.  PUT
2830                                  ;*      THE CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER
2831                                  ;*      TO AN RK06, IN 26 SECTOR FORMAT, TO CYLINDER 0, HEAD 0,
2832                                  ;*      DRIVE 0, WITH A ONE WORD TRANSFER.  CLOCK THROUGH THE
2833                                  ;*      SEEK AND THE DRIVE CLEAR MESSAGES.  ISSUE 200 CONTROLLER
2834                                  ;*      CLOCKS AND MAKE SURE WRITE GATE DOES NOT SET.  SIMULATE
2835                                  ;*      SECTOR PULSE AND 200 CONTROLLER CLOCKS MAKING SURE WRITE
2836                                  ;*      GATE DOES NOT SET.  SIMULATE INDEX PULSE AND MAKE SURE
2837                                  ;*      WRITE GATE SETS.
2838                                  ;*
2839                                  ;;*****************************************************************
2840  007462  000004                  TST10:  SCOPE
2841  007464  012737  000012  001200          MOV     #10.,$TIMES     ;;DO 10. ITERATIONS
2842  007472  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
2843  007476  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2844  007504  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
2845  007512  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)  ;INITIAILZE ROM ADDRESS
2846  007520  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2847  007526  012762  067204  000004          MOV     #WRBUFF,RKBA(R2)  ;LOAD BUS ADDRESS
```

```
2848  007534  012762  177777  000002          MOV     #-1,RKWC(R2)     ;LOAD WORD COUNT
2849  007542  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)  ;ISSUE WRITE HEADER
2850  007550  012700  000312                  MOV     #50.*4+2,R0      ;CLOCK UNTIL READY FOR INDEX PULSE
2851  007554  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
2852  007562  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2853  007570  005300                          DEC     R0
2854  007572  001370                          BNE     1$
2855  007574  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2856  007602  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG.
2857  007610  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT REG.
2858  007616  012737  000027  003500          MOV     #WRHEAD,E.CS1   ;LOAD EXPECTED CS1
2859  007624  012737  067204  003504          MOV     #WRBUFF,E.BA    ;LOAD EXPECTED BUS ADDRESS
2860  007632  012737  177777  003502          MOV     #-1,E.WC        ;LOAD EXPECTED WORD COUNT
2861  007640  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
2862  007646  001403                          BEQ     2$              ;YES, CONTINUE
2863  007650  104022                          ERROR   22              ;CS1 INCORRECT
2864  007652  000137  010600                  JMP     60$             ;CLEAR CONTROLLER
2865
2866  007656  023737  003504  003444  2$:     CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
2867  007664  001403                          BEQ     3$              ;YES, CONTINUE
2868  007666  104023                          ERROR   23              ;BUS ADDRESS INCORRECT
2869  007670  000137  010600                  JMP     60$             ;CLEAR RK611
2870
2871  007674  023737  003502  003442  3$:     CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
2872  007702  001403                          BEQ     4$              ;YES, CONTINUE
2873  007704  104024                          ERROR   24              ;WORD COUNT INCORRECT
2874  007706  000137  010600                  JMP     60$             ;CLEAR RK611
2875
2876  007712  012737  022040  003524  4$:     MOV     #ECCW!MEWD!DMD,E.MR1  ;LOAD EXPECTED MAINT REG. 1
2877  007720  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
2878  007724  012700  000310                  MOV     #200.,R0        ;ISSUE 200 MAINT BITS
2879  007730  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2)
2880  007736  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2881  007744  012762  000440  000026          MOV     #DMD!MCLK,RKMR1(R2)
2882  007752  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2883  007760  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINT REG. 1
2884  007766  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINTENANCE REG. 1 CORRECT
2885  007774  001403                          BEQ     6$              ;YES, CONTINUE
2886  007776  104025                          ERROR   25              ;MAINT REG 1 INCORRECT
2887  010000  000137  010600                  JMP     60$             ;CLEAR RK611
2888
2889  010004  005300                  6$:     DEC     R0              ;CHECK IF READY FOR SECTOR PULSE
2890  010006  001350                          BNE     5$              ;NO, GET NEXT BIT
2891  010010  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2892  010016  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG
2893  010024  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT REG
2894  010032  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
2895  010040  001403                          BEQ     10$             ;YES, CONTINUE
2896  010042  104026                          ERROR   26              ;CS1 INCORRECT
2897  010044  000137  010600                  JMP     60$             ;CLEAR RK611
2898
2899  010050  023737  003504  003444  10$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
2900  010056  001403                          BEQ     11$             ;YES, CONTINUE
2901  010060  104027                          ERROR   27              ;BUS ADDRESS INCORRECT
2902  010062  000137  010600                  JMP     60$             ;CLEAR RK611
2903
```

```
2904  010066  023737  003502  003442  11$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT REG. CORRECT
2905  010074  001403                          BEQ     12$             ;YES, CONTINUE
2906  010076  104030                          ERROR   30              ;WORD COUNT INCORRECT
2907  010100  000137  010600                  JMP     60$             ;CLEAR RK611
2908
2909  010104  012700  000004          12$:    MOV     #4,R0           ;SIMULATE SECTOR PULSE
2910  010110  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)
2911  010116  012762  000540  000026  13$:    MOV     #DMD!MSP!MCLK,RKMR1(R2)
2912  010124  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)
2913  010132  005300                          DEC     R0
2914  010134  001370                          BNE     13$
2915  010136  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2916  010144  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2917  010152  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS AND REG.
2918  010160  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT REG.
2919  010166  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 INCORRECT
2920  010174  001402                          BEQ     15$             ;YES, CONTINUE
2921  010176  104026                          ERROR   26              ;CS1 INCORRECT
2922  010200  000577                          BR      60$             ;CLEAR RK611
2923
2924  010202  023737  003504  003444  15$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
2925  010210  001402                          BEQ     16$             ;YES, CONTINUE
2926  010212  104027                          ERROR   27              ;BUS ADDRESS INCORRECT
2927  010214  000571                          BR      60$             ;CLEAR RK611
2928
2929  010216  023737  003502  003442  16$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
2930  010224  001402                          BEQ     20$             ;YES, CONTINUE
2931  010226  104030                          ERROR   30              ;WORD COUNT INCORRECT
2932  010230  000563                          BR      60$             ;CLEAR RK611
2933
2934  010232  005037  003622          20$:    CLR     BITCNT          ;INITIALIZE BIT COUNT
2935  010236  012700  000310                  MOV     #200.,R0        ;ISSUE 200 MAINT BITS
2936  010242  012762  000440  000026  21$:    MOV     #DMD!MCLK,RKMR1(R2)
2937  010250  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2938  010256  012762  000440  000026          MOV     #DMD!MCLK,RKMR1(R2)
2939  010264  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2940  010272  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINT REG. 1
2941  010300  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MAINT REG. 1 CORRECT
2942  010306  001402                          BEQ     22$             ;YES, CONTINUE
2943  010310  104025                          ERROR   25              ;MAINT REG. 1 INCORRECT
2944  010312  000532                          BR      60$             ;CLEAR RK611
2945
2946  010314  005300          22$:    DEC     R0              ;CHECK IF READY FOR INDEX PULSE
2947  010316  001351                          BNE     21$             ;NO, GET NEXT BIT
2948  010320  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
2949  010326  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
2950  010334  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
2951  010342  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
2952  010350  001402                          BEQ     23$             ;YES, CONTINUE
2953  010352  104026                          ERROR   26              ;CS1 INCORRECT
2954  010354  000511                          BR      60$             ;CLEAR RK611
2955
2956  010356  023737  003504  003444  23$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
2957  010364  001402                          BEQ     24$             ;YES, CONTINUE
2958  010366  104027                          ERROR   27              ;BUS ADDRESS INCORRECT
2959  010370  000503                          BR      60$             ;CLEAR RK611
```

```
2960
2961  010372  023737  003502  003442  24$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
2962  010400  001402                          BEQ     25$             ;YES, CONTINUE
2963  010402  104030                          ERROR   30              ;WORD COUNT INCORRECT
2964  010404  000475                          BR      60$             ;CLEAR RK611
2965
2966  010406  012762  000240  000026  25$:    MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE PULSE
2967  010414  012700  000004                  MOV     #4,R0
2968  010420  012762  000640  000026  26$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
2969  010426  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
2970  010434  005300                          DEC     R0
2971  010436  001370                          BNE     26$
2972  010440  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2973  010446  012700  000002                  MOV     #2,R0           ;SIMULATE TWO CLOCK PULSES FOR WRITE
2974                                                                  ;  GATE TO COME UP
2975  010452  012762  000440  000026  27$:    MOV     #DMD!MCLK,RKMR1(R2)
2976  010460  012762  000040  000026          MOV     #DMD,RKMR1(R2)
2977  010466  005300                          DEC     R0
2978  010470  001370                          BNE     27$
2979  010472  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINTENANCE REG.
2980  010500  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
2981  010506  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
2982  010514  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
2983  010522  012737  062040  003524          MOV     #WRTGAT!MEWD!ECCW!DMD,E.MR1  ;LOAD EXPECTED MAINT REG. 1
2984  010530  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
2985  010536  001401                          BEQ     28$             ;YES, CONTINUE
2986  010540  104031                          ERROR   31              ;CS1 INCORRECT
2987  010542  023737  003504  003444  28$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
2988  010550  001401                          BEQ     29$             ;YES, CONTINUE
2989  010552  104032                          ERROR   32              ;BUS ADDRESS INCORRECT
2990  010554  023737  003502  003442  29$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
2991  010562  001401                          BEQ     30$             ;YES, CONTINUE
2992  010564  104033                          ERROR   33              ;WORD COUNT INCORRECT
2993  010566  023737  003524  003464  30$:    CMP     E.MR1,T.MR1     ;CHECK MAINT REG 1 CORRECT
2994  010574  001401                          BEQ     60$             ;YES, CLEAR RK611
2995  010576  104034                          ERROR   34              ;MAINT REG. 1 INCORRECT
2996  010600  012762  100000  000000  60$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
2997
2998                                  .SBTTL  **NPR READING OF MEMORY
2999
3000                                  ;;*****************************************************************
3001                                  ;*TEST 11       NPR OUTPUT DATA TRANSFER
3002                                  ;*
3003                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
3004                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
3005                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 777, HEAD 7, DRIVE 7.
3006                                  ;*      SPECIFY A ONE WORD DATA TRANSFER.  CLOCK BOTH SEEK
3007                                  ;*      AND DRIVE CLEAR MESSAGES.  SIMULATE INDEX PULSE.
3008                                  ;*      CLOCK IN FIRST WORD OF NPR TRANSFER.  CHECK INPUT READY,
3009                                  ;*      OUTPUT READY, BUS  ADDRESS, WORD COUNT, AND CONTENTS OF
3010                                  ;*      THE SILO.  REPEAT FOR 8 DIFFERENT DATA PATTERNS.
3011                                  ;*
3012                                  ;;*****************************************************************
3013  010606  000004                  TST11:  SCOPE
3014  010610  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3015  010616  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
```

```
3016  010622  012703  066664                MOV   #PATTERN,R3       ;LOAD PATTERN ADDRESS
3017  010626  012704  000010                MOV   #8.,R4            ;LOAD PATTERN COUNT
3018  010632  012737  067206  003504        MOV   #WRBUFF+2,E.BA    ;LOAD EXPECTED BUS ADDRESS
3019  010640  012737  000027  003500        MOV   #WRHEAD,E.CS1     ;LOAD EXPECTED CS1
3020  010646  005037  003502                CLR   E.WC              ;LOAD EXPECTED WORD COUNT
3021  010652  012737  010660  001110        MOV   #1$,$LPERR        ;LOAD LOOP ON ERROR LOCATION FOR
3022                                                                ;  SUBTEST LOOP
3023
3024  010660                          1$:
3025  010660  012762  100000  000000        MOV   #CCLR,RKCS1(R2)   ;CLEAR RK611
3026  010666  011337  067204                MOV   (R3),WRBUFF       ;LOAD BUFFER FOR WRITE HEADER
3027  010672  012762  000040  000026        MOV   #DMD,RKMR1(R2)    ;PUT RK611 IN MAINT MODE
3028  010700  012762  000777  000020        MOV   #777,RKDCYL(R2)   ;LOAD CYLINDER ADDRESS
3029  010706  012762  003400  000006        MOV   #3400,RKDA(R2)    ;LOAD DISK ADDRESS
3030  010714  012762  067204  000004        MOV   #WRBUFF,RKBA(R2)  ;LOAD BUS ADDRESS
3031  010722  012762  000007  000010        MOV   #7,RKCS2(R2)      ;LOAD OTHER NUMBER
3032  010730  012762  177777  000002        MOV   #-1,RKWC(R2)      ;LOAD WORD COUNT
3033  010736  012762  000027  000000        MOV   #WRHEAD,RKCS1(R2) ;ISSUE WRITE HEADER
3034  010744  012700  000312                MOV   #50.*4+2,R0       ;ISSUE CLOCKS UNTIL READY FOR
3035                                                                ;  INDEX PULSE
3036  010750  012762  000440  000026  2$:   MOV   #DMD!MCLK,RKMR1(R2)
3037  010756  012762  000040  000026        MOV   #DMD,RKMR1(R2)
3038  010764  005300                        DEC   R0
3039  010766  001370                        BNE   2$
3040  010770  012700  000004                MOV   #4,R0             ;SIMULATE INDEX PULSE
3041  010774  012762  000240  000026        MOV   #DMD!MIND,RKMR1(R2)
3042  011002  012762  000640  000026  3$:   MOV   #DMD!MIND!MCLK,RKMR1(R2)
3043  011010  012762  000240  000026        MOV   #DMD!MIND,RKMR1(R2)
3044  011016  005300                        DEC   R0
3045  011020  001370                        BNE   3$
3046  011022  012762  000040  000026        MOV   #DMD,RKMR1(R2)
3047  011030  012700  000045                MOV   #37.,R0           ;SIMULATE 1 NPR TRANSFER
3048  011034  012762  000440  000026  4$:   MOV   #DMD!MCLK,RKMR1(R2)
3049  011042  012762  000040  000026        MOV   #DMD,RKMR1(R2)
3050  011050  005300                        DEC   R0
3051  011052  001370                        BNE   4$
3052  011054  016237  000000  003440        MOV   RKCS1(R2),T.CS1   ;STORE COMMAND AND STATUS REG. 1
3053  011062  016237  000004  003444        MOV   RKBA(R2),T.BA     ;STORE BUS ADDRESS REG
3054  011070  016237  000002  003442        MOV   RKWC(R2),T.WC     ;STORE WORD COUNT
3055  011076  012700  000024                MOV   #20.,R0           ;WAIT FOR OUTPUT READY
3056  011102  005300                  5$:   DEC   R0
3057  011104  001376                        BNE   5$
3058  011106  016237  000010  003450        MOV   RKCS2(R2),T.CS2   ;STORE COMMAND AND STATUS REG. 2
3059  011114  012737  000307  003510        MOV   #OR!IR!7,E.CS2    ;LOAD EXPECTED CS2
3060  011122  023737  003500  003440        CMP   E.CS1,T.CS1       ;CHECK CS1 CORRECT
3061  011130  001401                        BEQ   6$                ;YES, CHECK CS2
3062  011132  104035                        ERROR 35                ;CS1 INCORRECT
3063  011134  023737  003510  003450  6$:   CMP   E.CS2,T.CS2       ;CHECK CS2 CORRECT
3064  011142  001401                        BEQ   7$                ;YES, CONTINUE
3065  011144  104036                        ERROR 36                ;CS2 INCORRECT
3066  011146  023737  003504  003444  7$:   CMP   E.BA,T.BA         ;CHECK IF BUS ADDRESS INCREMENT OCCURRED
3067  011154  001401                        BEQ   8$                ;YES, CONTINUE
3068  011156  104037                        ERROR 37                ;BUS ADDRESS INCORRECT
3069  011160  023737  003502  003442  8$:   CMP   E.WC,T.WC         ;CHECK WORD COUNT REG CORRECT
3070  011166  001401                        BEQ   9$                ;YES, CONTINUE
3071  011170  104040                        ERROR 40                ;WORD COUNT INCORRECT
```

```
3072  011172  016237  000024  003462  9$:     MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
3073  011200  011337  003522                  MOV     (R3),E.DB       ;LOAD EXPECTED DATA BUFFER
3074  011204  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3075  011212  001401                          BEQ     15$             ;YES,CONTINUE
3076  011214  104041                          ERROR   41              ;DATA BUFFER INCORRECT
3077  011216  016237  000000  003440  15$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3078  011224  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3079  011232  012737  000107  003510          MOV     #IR!7,E.CS2     ;LOAD EXPECTED CS2
3080  011240  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
3081  011246  001401                          BEQ     17$             ;YES, CONTINUE
3082  011250  104042                          ERROR   42              ;CS1 INCORRECT
3083  011252  023737  003510  003450  17$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND AND STATUS REG. 2 CORRECT
3084  011260  001401                          BEQ     20$             ;YES, CONTINUE
3085  011262  104043                          ERROR   43              ;CS2 INCORRECT
3086  011264  104415                  20$:    SCOP1                   ;CHECK IF LOOP ON ERROR
3087  011266  005723                          TST     (R3)+           ;GENERATE ADDRESS OF NEXT CONFIG
3088  011270  005304                          DEC     R4              ;CHECK IF ALL 8 CONFIGS TRIED
3089  011272  001000                          BNE     TST12           ;;NO, TRY NEXT DATA PATTERN
3090
3091                          ;;*****************************************************************
3092                          ;*TEST 12       PARTIAL SILO FILLING
3093                          ;*
3094                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
3095                          ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
3096                          ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
3097                          ;*      SPECIFY A ONE WORD DATA TRANSFER.  CLOCK IN ALL
3098                          ;*      SPECIFIED WORDS INTO THE SILO.  CHECK WORD COUNT,
3099                          ;*      BUS ADDRESS, INPUT READY, AND OUTPUT READY.  MAKE
3100                          ;*      SURE NO MORE THAN SPECIFIED DATA LENGTH IS CLOCKED
3101                          ;*      INTO THE SILO.  CHECK THE SILO FOR CORRECT DATA.
3102                          ;*      REPEAT FOR WORD COUNTS 2-65.
3103                          ;*
3104                          ;*****************************************************************
3105  011274  000004  TST12:  SCOPE
3106  011276  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3107  011304  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
3108  011310  012737  000001  001160          MOV     #1,$TMPO        ;LOAD NUMBER OF WORDS FOR DATA TRANSFER
3109  011316  012704  000065                  MOV     #65,R4          ;LOAD ITERATION COUNT
3110  011322  012737  000027  003500          MOV     #WRHEAD,E.CS1   ;LOAD EXPECTED CS1
3111  011330  012737  011336  001110          MOV     #1$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
3112                                                                  ;  SUBTEST LOOP
3113
3114  011336                  1$:
3115  011336  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3116  011344  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINT MODE
3117  011352  012762  066762  000004          MOV     #NPRBUF,RKBA(R2)  ;LOAD BUS ADDRESS
3118  011360  012737  066762  003504          MOV     #NPRBUF,E.BA
3119  011366  013737  001160  003502          MOV     $TMPO,E.WC      ;LOAD WORD COUNT
3120  011374  005437  003502                  NEG     E.WC
3121  011400  013762  003502  000002          MOV     E.WC,RKWC(R2)
3122  011406  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)  ;ISSUE WRITE HEADER
3123  011414  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE CLOCKS UNTIL READY FOR
3124                                                                  ;  INDEX PULSE
3125  011420  012762  000440  000026  2$:     MOV     #DMD!MCLK,RKMR1(R2)
3126  011426  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3127  011434  005300                          DEC     R0
```

```
3128  011436  001370                        BNE     2$
3129  011440  012700  000004                MOV     #4,R0           ;SIMULATE INDEX PULSE
3130  011444  012762  000240  000026        MOV     #DMD!MIND,RKMR1(R2)
3131  011452  012762  000640  000026  3$:   MOV     #DMD!MIND!MCLK,RKMR1(R2)
3132  011460  012762  000240  000026        MOV     #DMD!MIND,RKMR1(R2)
3133  011466  005300                        DEC     R0
3134  011470  001370                        BNE     3$
3135  011472  012762  000040  000026        MOV     #DMD,RKMR1(R2)
3136  011500  012737  000300  003510        MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
3137  011506  012700  000045          4$:   MOV     #37.,R0         ;SIMULATE 1 NPR TRANSFER
3138  011512  012762  000440  000026  5$:   MOV     #DMD!MCLK,RKMR1(R2)
3139  011520  012762  000040  000026        MOV     #DMD,RKMR1(R2)
3140  011526  005300                        DEC     R0
3141  011530  001370                        BNE     5$
3142  011532  016237  000000  003440        MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3143  011540  016237  000004  003444        MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
3144  011546  016237  000002  003442        MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3145  011554  022737  066762  003504        CMP     #NPRBUF,E.BA    ;CHECK IF FIRST WORD
3146  011562  001004                        BNE     7$              ;NO, STORE CS2
3147  011564  012700  000024                MOV     #20.,R0         ;WAIT FOR OUTPUT READY
3148  011570  005300                  6$:   DEC     R0
3149  011572  001376                        BNE     6$
3150  011574  016237  000010  003450  7$:   MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3151  011602  062737  000002  003504        ADD     #2,E.BA         ;INCREMENT WORD COUNT AND BUS ADD
3152  011610  005237  003502                INC     E.WC
3153  011614  023737  003500  003440        CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3154  011622  001401                        BEQ     8$              ;YES, CHECK CS2
3155  011624  104044                        ERROR   44              ;CS1 INCORRECT
3156  011626  023737  003510  003450  8$:   CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3157  011634  001401                        BEQ     9$              ;YES, CHECK BUSS ADDRESS REG.
3158  011636  104045                        ERROR   45              ;CS2 INCORRECT
3159  011640  023737  003504  003444  9$:   CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
3160  011646  001401                        BEQ     10$             ;YES, CHECK WORD COUNT
3161  011650  104046                        ERROR   46              ;BUS ADDRESS INCORRECT
3162  011652  023737  003502  003442  10$:  CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3163  011660  001401                        BEQ     11$             ;YES, CHECK IF ALL WORDS TRANSFERRED
3164  011662  104047                        ERROR   47              ;WORD COUNT INCORRECT
3165  011664  005737  003502          11$:  TST     E.WC            ;CHECK IF FINISHED
3166  011670  001306                        BNE     4$              ;NO, TRANSFER NEXT WORD
3167  011672  012700  000112                MOV     #2*37.,R0       ;ISSUE ENOUGH CLOCKS FOR
3168  011676  012762  000440  000026  15$:  MOV     #DMD!MCLK,RKMR1(R2)  ;  2 NPR TRANSFERS
3169  011704  012762  000040  000026        MOV     #DMD,RKMR1(R2)
3170  011712  005300                        DEC     R0
3171  011714  001370                        BNE     15$
3172  011716  016237  000000  003440        MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3173  011724  016237  000010  003450        MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3174  011732  016237  000004  003444        MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG.
3175  011740  016237  000002  003442        MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3176  011746  023737  003500  003440        CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3177  011754  001401                        BEQ     16$             ;YES, CHECK CS2
3178  011756  104050                        ERROR   50              ;CS1 INCORRECT
3179  011760  023737  003510  003450  16$:  CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3180  011766  001401                        BEQ     17$             ;YES, CHECK BUS ADDRESS
3181  011770  104051                        ERROR   51              ;CS2 INCORRECT
3182  011772  023737  003504  003444  17$:  CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
3183  012000  001401                        BEQ     18$             ;YES, CHECK WORD COUNT
```

```
3184  012002  104052                        ERROR   52              ;BUS ADDRESS INCORRECT
3185  012004  023737  003502  003442  18$:  CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3186  012012  001401                        BEQ     19$             ;YES, CONTINUE
3187  012014  104053                        ERROR   53              ;WORD COUNT INCORRECT
3188  012016  013701  001160          19$:  MOV     $TMPO,R1        ;LOAD NUMBER OF WORDS LOADED
3189  012022  012703  066762                MOV     #NPRBUFF,R3     ;LOAD START ADDRESS
3190  012026  005037  003624                CLR     WRDCNT          ;INITIALIZE WORD COUNT FOR PRINTOUT
3191  012032  016237  000024  003462  25$:  MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
3192  012040  012337  003522                MOV     (R3)+,E.DB      ;LOAD EXPECTED DATA BUFFER
3193  012044  023737  003522  003462        CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3194  012052  001401                        BEQ     26$             ;YES, CONTINUE
3195  012054  104054                        ERROR   54              ;DATA BUFFER INCORRECT
3196  012056  016237  000000  003440  26$:  MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3197  012064  016237  000010  003450        MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3198  012072  022701  000001                CMP     #1,R1           ;CHECK IF LAST WORD IN SILO
3199  012076  001003                        BNE     27$             ;NO, CONTINUE
3200  012100  012737  000100  003510        MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
3201  012106  023737  003500  003440  27$:  CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3202  012114  001401                        BEQ     28$             ;YES, CONTINUE
3203  012116  104055                        ERROR   55              ;CS1 INCORRECT
3204  012120  023737  003510  003450  28$:  CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3205  012126  001401                        BEQ     29$             ;YES, CONTINUE
3206  012130  104056                        ERROR   56              ;CS2 INCORRECT
3207  012132  005237  003624          29$:  INC     WRDCNT          ;INCREMENT WORD COUNT
3208  012136  005301                        DEC     R1              ;DECREMENT WORDS READ
3209  012140  001334                        BNE     25$             ;CHECK IF ALL WORDS READ
3210  012142  104415                        SCOP1                   ;CHECK IF LOOP ON ERROR
3211  012144  005237  001160                INC     $TMPO           ;INCREMENT WORDS READ FROM MEM
3212  012150  005304                        DEC     R4              ;CHECK IF FINISHED
3213  012152  001402                        BEQ     TST13           ;;YES, GO ON TO NEXT TEST
3214  012154  000137  011336                JMP     1$              ;TRY NEXT WORD COUNT
3215
3216                          ;;*****************************************************************
3217                          ;*TEST 13       SILO FILLING WITH NPR TRANSFERS
3218                          ;*
3219                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
3220                          ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
3221                          ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
3222                          ;*      SPECIFY A 66 WORD DATA TRANSFER, CLOCK IN ALL 66 WORDS
3223                          ;*      INTO THE SILO.  CHECK INPUT READY, OUTPUT READY, BUS
3224                          ;*      ADDRESS, WORD COUNT, AND CONTENTS OF THE SILO.
3225                          ;*
3226                          ;;*****************************************************************
3227  012160  000004          TST13:  SCOPE
3228  012162  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3229  012170  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
3230  012174  012737  000027  003500          MOV     #WRHEAD,E.CS1   ;LOAD EXPECTED CS1
3231  012202  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3232  012210  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINT MODE
3233  012216  012737  066764  003504          MOV     #NPRBUF+2,E.BA  ;LOAD BUS ADDRESS
3234  012224  012762  066762  000004          MOV     #NPRBUF,RKBA(R2)
3235  012232  012737  177677  003502          MOV     #-65.,E.WC      ;LOAD WORD COUNT
3236  012240  012762  177676  000002          MOV     #-66.,RKWC(R2)
3237  012246  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)  ;ISSUE WRITE HEADER
3238  012254  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE CLOCKS UNTIL READY FOR
3239                                                                  ;  INDEX PULSE
```

```
3240  012260  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
3241  012266  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3242  012274  005300                          DEC     R0
3243  012276  001370                          BNE     1$
3244  012300  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
3245  012304  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3246  012312  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
3247  012320  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3248  012326  005300                          DEC     R0
3249  012330  001370                          BNE     2$
3250  012332  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3251  012340  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
3252  012346  012700  000045          4$:     MOV     #37.,R0         ;SIMULATE 1 NPR TRANSFER
3253  012352  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2)
3254  012360  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3255  012366  005300                          DEC     R0
3256  012370  001370                          BNE     5$
3257  012372  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3258  012400  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
3259  012406  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3260  012414  022737  066764  003504          CMP     #NPRBUF+2,E.BA  ;CHECK IF FIRST WORD
3261  012422  001004                          BNE     7$              ;NO, STORE CS2
3262  012424  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
3263  012430  005300                  6$:     DEC     R0
3264  012432  001376                          BNE     6$
3265  012434  016237  000010  003450  7$:     MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3266  012442  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3267  012450  001401                          BEQ     8$              ;YES, CHECK CS2
3268  012452  104044                          ERROR   44              ;CS1 INCORRECT
3269  012454  023737  003510  003450  8$:     CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3270  012462  001401                          BEQ     9$              ;YES, CHECK BUS ADDRESS
3271  012464  104045                          ERROR   45              ;CS2 INCORRECT
3272  012466  023737  003504  003444  9$:     CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
3273  012474  001401                          BEQ     10$             ;YES, CHECK WORD COUNT
3274  012476  104046                          ERROR   46              ;BUS ADDRESS INCORRECT
3275  012500  023737  003502  003442  10$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3276  012506  001401                          BEQ     11$             ;YES, CHECK IF ALL WORDS TRANSFERRED
3277  012510  104047                          ERROR   47              ;WORD COUNT INCORRECT
3278  012512  062737  000002  003504  11$:    ADD     #2,E.BA         ;INCREMENT WORD COUNT AND BUS ADDRESS
3279  012520  005237  003502                  INC     E.WC            ;
3280  012524  100710                          BMI     4$              ;CHECK IF FINISHED (NO, BRANCH)
3281  012526  001004                          BNE     12$             ;CHECK IF LAST WORD
3282  012530  012737  000200  003510          MOV     #OR,E.CS2       ;LOAD EXPECTED CS2
3283  012536  000703                          BR      4$              ;PROCESS LAST WORD
3284
3285  012540  005037  003502          12$:    CLR     E.WC            ;ADJUST EXPECTED WORD COUNT
3286  012544  162737  000002  003504          SUB     #2,E.BA         ;  AND BUS ADDRESS
3287  012552  012700  000112                  MOV     #2*37.,R0       ;ISSUE ENOUGH CLOCKS FOR
3288  012556  012762  000440  000026  15$:    MOV     #DMD!MCLK,RKMR1(R2)  ;  2 NPR TRANSFERS
3289  012564  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3290  012572  005300                          DEC     R0
3291  012574  001370                          BNE     15$
3292  012576  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3293  012604  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3294  012612  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG
3295  012620  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
```

```
3296  012626  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3297  012634  001401                          BEQ     16$             ;YES, CHECK CS2
3298  012636  104050                          ERROR   50              ;CS1 INCORRECT
3299  012640  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3300  012646  001401                          BEQ     17$             ;YES, CHECK BUS ADDRESS
3301  012650  104051                          ERROR   51              ;CS2 INCORRECT
3302  012652  023737  003504  003444  17$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
3303  012660  001401                          BEQ     18$             ;YES, CHECK WORD COUNT
3304  012662  104052                          ERROR   52              ;BUS ADDRESS INCORRECT
3305  012664  023737  003502  003442  18$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3306  012672  001401                          BEQ     19$             ;YES, CHECK DATA
3307  012674  104053                          ERROR   53              ;WORD COUNT INCORRECT
3308  012676  012701  000102          19$:    MOV     #66.,R1         ;LOAD NUMBER OF WORDS LOADED
3309  012702  012703  066762                  MOV     #NPRBUF,R3      ;LOAD START ADDRESS
3310  012706  005037  003624                  CLR     WRDCNT          ;INITIALIZE WORD COUNT FOR  PRINT OUT
3311  012712  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
3312  012720  016237  000024  003462  25$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
3313  012726  012337  003522                  MOV     (R3)+,E.DB      ;LOAD EXPECTED DATA BUFFER
3314  012732  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3315  012740  001401                          BEQ     26$             ;YES, CONTINUE
3316  012742  104054                          ERROR   54              ;DATA BUFFER INCORRECT
3317  012744  016237  000000  003440  26$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
3318  012752  005737  003624                  TST     WRDCNT          ;CHECK IF FIRST WORD
3319  012756  001015                          BNE     28$             ;NO, GET CS2
3320  012760  012700  000024                  MOV     #20.,R0         ;WAIT FOR INPUT READY TO SET
3321  012764  005300                  27$:    DEC     R0
3322  012766  001376                          BNE     27$
3323  012770  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3324  012776  022701  000001                  CMP     #1,R1           ;CHECK IF LAST WORD IN SILO
3325  013002  001003                          BNE     28$             ;NO, CONTINUE
3326  013004  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
3327  013012  023737  003500  003440  28$:    CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
3328  013020  001401                          BEQ     29$             ;YES, CHECK CS2
3329  013022  104055                          ERROR   55              ;CS1 INCORRECT
3330  013024  023737  003510  003450  29$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND AND STATUS REG. 2 CORRECT
3331  013032  001401                          BEQ     30$             ;YES, GET NEXT WORD
3332  013034  104056                          ERROR   56              ;CS2 INCORRECT
3333  013036  005237  003624          30$:    INC     WRDCNT          ;INCREMENT WORD COUNT
3334  013042  005301                          DEC     R1              ;DECREMENT WORDS READ
3335  013044  001325                          BNE     25$             ;CHECK IF ALL WORDS READ
3336
3337                          ;;*****************************************************************
3338                          ;*TEST 14       SILO CAPICITY WITH NPR TRANSFERS
3339                          ;*
3340                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
3341                          ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
3342                          ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
3343                          ;*      SPECIFY A 68 WORD DATA TRANSFER.  CLOCK IN 66
3344                          ;*      WORDS INTO THE SILO.  MAKE SURE THAT SILO WILL STOP
3345                          ;*      FILLING AT 66 WORDS.  TAKE ONE WORD FROM SILO AND
3346                          ;*      CHECK IT.  CLOCK IN NEXT WORD.  MAKE SURE NO MORE
3347                          ;*      THAN  ONE WORD IS CLOCKED IN THE SILO.  TAKE ONE
3348                          ;*      WORD FROM SILO AND CHECK IT.  CLOCK IN NEXT WORD.
3349                          ;*      CLOCK IN NEXT WORD.  MAKE SURE NO MORE THAN ONE WORD IS
3350                          ;*      CLOCKED IN THE SILO.  TAKE ONE WORD FROM SILO AND
3351                          ;*      CHECK IT. ATTEMPT TO CLOCK IN NEXT WORD AND
```

```
                                        ;*      MAKE SURE NO WORDS ARE CLOCK INTO SILO.  UNLOAD THE
                                        ;*      SILO AND MAKE SURE ALL THE WORDS ARE CORRECT.
                                        ;*
                                        ;;*****************************************************************
013046  000004                        TST14:  SCOPE
013050  012737  000144  001200              MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
013056  013702  001270                      MOV     $BASE,R2        ;LOAD RK611 BASE
013062  012737  000027  003500              MOV     #WRHEAD,E.CS1   ;LOAD EXPECTED CS1
013070  012762  100000  000000              MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
013076  012762  000040  000026              MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINT MODE
013104  012737  066764  003504              MOV     #NPRBUFF+2,E.BA ;LOAD BUS ADDRESS
013112  012762  066762  000004              MOV     #NPRBUF,RKBA(R2)
013120  012737  177675  003502              MOV     #-67.,E.WC      ;LOAD WORD COUNT
013126  012762  177674  000002              MOV     #-68.,RKWC(R2)
013134  012762  000027  000000              MOV     #WRHEAD,RKCS1(R2)  ;ISSUE WRITE HEADER
013142  012700  000312                      MOV     #50.*4+2,R0     ;ISSUE CLOCKS UNTIL READY FOR
                                                                    ;  INDEX PULSE
013146  012762  000440  000026  1$:         MOV     #DMD!MCLK,RKMR1(R2)
013154  012762  000040  000026              MOV     #DMD,RKMR1(R2)
013162  005300                              DEC     R0
013164  001370                              BNE     1$
013166  012700  000004                      MOV     #4,R0           ;SIMULATE INDEX PULSE
013172  012762  000240  000026              MOV     #DMD!MIND,RKMR1(R2)
013200  012762  000640  000026  2$:         MOV     #DMD!MIND!MCLK,RKMR1(R2)
013206  012762  000240  000026              MOV     #DMD!MIND,RKMR1(R2)
013214  005300                              DEC     R0
013216  001370                              BNE     2$
013220  012737  000040  000026              MOV     #DMD,RKMR1
013226  012737  000300  003510              MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
013234  012700  000045              4$:     MOV     #37.,R0         ;SIMULATE 1 NPR TRANSFER
013240  012762  000440  000026  5$:         MOV     #DMD!MCLK,RKMR1(R2)
013246  012762  000040  000026              MOV     #DMD,RKMR1(R2)
013254  005300                              DEC     R0
013256  001370                              BNE     5$
013260  016237  000000  003440              MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND REG. 1
013266  016237  000004  003444              MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
013274  016237  000002  003442              MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
013302  022737  066764  003504              CMP     #NPRBUF+2,E.BA  ;CHECK IF FIRST WORD
013310  001004                              BNE     7$              ;NO, STORE CS2
013312  012700  000024                      MOV     #20.,R0         ;WAIT FOR OUTPUT READY
013316  005300                      6$:     DEC     R0
013320  001376                              BNE     6$
013322  016237  000010  003450  7$:         MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
013330  023737  003500  003440              CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
013336  001401                              BEQ     8$              ;YES, CHECK CS2
013340  104044                              ERROR   44              ;CS1 INCORRECT
013342  023737  003510  003450  8$:         CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
013350  001401                              BEQ     9$              ;YES, CHECK BUS ADDRESS
013352  104045                              ERROR   45              ;CS2 INCORRECT
013354  023737  003504  003444  9$:         CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
013362  001401                              BEQ     10$             ;YES, CHECK WORD COUNT
013364  104046                              ERROR   46              ;BUS ADDRESS INCORRECT
013366  023737  003502  003442  10$:        CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
013374  001401                              BEQ     11$             ;YES, CHECK IF 1ST 65 WORDS TRANSFERRED
013376  104047                              ERROR   47              ;WORD COUNT INCORRECT
013400  062737  000002  003504  11$:        ADD     #2,E.BA         ;INCREMENT BUS ADD AND WORD COUNT
```

```
3408  013406  005237  003502                    INC     E.WC
3409  013412  022737  177776  003502            CMP     #-2,E.WC        ;CHECK IF 65 WORDS IN SILO
3410  013420  101305                            BHI     4$              ;NO, GET NEXT WORD
3411  013422  001004                            BNE     12$             ;CHECK IF ALL 65 WORDS IN SILO
3412  013424  012737  000200  003510            MOV     #OR,E.CS2       ;LOAD EXPECTED CS2
3413  013432  000700                            BR      4$              ;PROCESS 66TH WORD
3414
3415  013434  005337  003502          12$:      DEC     E.WC            ;ADJUST WORD COUNT AND
3416  013440  162737  000002  003504            SUB     #2,E.BA         ;  BUS ADDRESS
3417  013446  012701  000003                    MOV     #3,R1
3418  013452  005037  003624                    CLR     WRDCNT
3419  013456  012703  066762                    MOV     #NPRBUF,R3      ;LOAD START ADDRESS
3420  013462  012700  000112          13$:      MOV     #2*37.,R0       ;ISSUE ENOUGH CLOCKS FOR
3421  013466  012762  000440  000026  15$:      MOV     #DMD!MCLK,RKMR1(R2)  ;  2 NPR TRANSFERS
3422  013474  012762  000040  000026            MOV     #DMD,RKMR1(R2)
3423  013502  005300                            DEC     R0
3424  013504  001370                            BNE     15$
3425  013506  016237  000000  003440            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3426  013514  016237  000010  003450            MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3427  013522  016237  000004  003444            MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG
3428  013530  016237  000002  003442            MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3429  013536  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1
3430  013544  001401                            BEQ     16$             ;YES, CHECK CS2
3431  013546  104050                            ERROR   50              ;CS1 INCORRECT
3432  013550  023737  003510  003450  16$:      CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3433  013556  001401                            BEQ     17$             ;YES,CHECK BUS ADDRESS
3434  013560  104051                            ERROR   51              ;CS2 INCORRECT
3435  013562  023737  003504  003444  17$:      CMP     E.BA,T.BA       ;CHECK BUS ADD CORRECT
3436  013570  001401                            BEQ     18$             ;YES, CHECK WORD COUNT
3437  013572  104052                            ERROR   52              ;BUS ADDRESS INCORRECT
3438  013574  023737  003502  003442  18$:      CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3439  013602  001401                            BEQ     19$             ;YES, READ 1 WORD FROM SILO
3440  013604  104053                            ERROR   53              ;WORD COUNT INCORRECT
3441  013606  012737  000300  003510  19$:      MOV     #IR!OR,E.CS2    ;LOAD EXPECT CS2
3442  013614  016237  000024  003462            MOV     RKDB(R2),T.DB   ;STORE DATA BUFFER
3443  013622  012337  003522                    MOV     (R3)+,E.DB      ;LOAD EXPECTED DATA BUFFER
3444  013626  023737  003522  003462            CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3445  013634  001401                            BEQ     25$             ;YES, CONTINUE
3446  013636  104054                            ERROR   54              ;DATA BUFFER INCORRECT
3447  013640  016237  000000  003440  25$:      MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3448  013646  012700  000024                    MOV     #20.,R0         ;WAIT FOR OUTPUT READY
3449  013652  005300                  26$:      DEC     R0
3450  013654  001376                            BNE     26$
3451  013656  016237  000010  003450            MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3452  013664  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK IF COMMAND STATUS REG. 1 CORRECT
3453  013672  001401                            BEQ     27$             ;YES, CHECK CS2
3454  013674  104055                            ERROR   55              ;CS1 INCORRECT
3455  013676  023737  003510  003450  27$:      CMP     E.CS2,T.CS2     ;CHECK IF COMMAND STATUS REG. 2 CORRECT
3456  013704  001401                            BEQ     30$             ;YES, DO NPR TRANSFER
3457  013706  104056                            ERROR   56              ;CS2 INCORRECT
3458  013710  012700  000045          30$:      MOV     #37.,R0         ;CLOCK IN ONE WORD (NPR TRANSFER)
3459  013714  012762  000440  000026  31$:      MOV     #DMD!MCLK,RKMR1(R2)
3460  013722  012762  000040  000026            MOV     #DMD,RKMR1(R2)
3461  013730  005300                            DEC     R0
3462  013732  001370                            BNE     31$
3463  013734  022701  000001                    CMP     #1,R1           ;CHECK IF 68TH WORD READ
```

```
3464  013740  001410                          BEQ     32$             ;YES, NO NPR WILL TAKE PLACE
3465  013742  062737  000002  003504          ADD     #2,E.BA         ;INCREMENT BUS ADD AND WORD COUNT
3466  013750  005237  003502                  INC     E.WC
3467  013754  012737  000200  003510          MOV     #OR,E.CS2       ;LOAD EXPECTED CS2
3468  013762  016237  000000  003440  32$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3469  013770  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3470  013776  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
3471  014004  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3472  014012  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3473  014020  001401                          BEQ     33$             ;YES, CHECK CS2
3474  014022  104044                          ERROR   44              ;CS1 INCORRECT
3475  014024  023737  003510  003450  33$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3476  014032  001401                          BEQ     34$             ;YES, CHECK BUS ADD
3477  014034  104045                          ERROR   45              ;CS2 INCORRECT
3478  014036  023737  003504  003444  34$:    CMP     E.BA,T.BA       ;CHECK BUS ADD CORRECT
3479  014044  001401                          BEQ     35$             ;YES, CHECK WORD COUNT
3480  014046  104046                          ERROR   46              ;BUS ADD INCORRECT
3481  014050  023737  003502  003442  35$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3482  014056  001401                          BEQ     36$             ;YES, CONTINUE
3483  014060  104047                          ERROR   47              ;WORD COUNT INCORRECT
3484  014062  005237  003624          36$:    INC     WRDCNT
3485  014066  005301                          DEC     R1              ;CHECK IF READ TO UNLOAD SILO
3486  014070  001402                          BEQ     39$             ;YES, READ DATA; BUFFER
3487  014072  000137  013462                  JMP     13$             ;NO, INPUT NEXT WORD
3488
3489  014076  162737  000002  003504  39$:    SUB     #2,E.BA         ;ADJUST WORD COUNT AND BUS ADDRESS
3490  014104  005037  003502                  CLR     E.WC
3491  014110  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
3492  014116  012701  000101                  MOV     #65.,R1         ;LOAD NUMBER OF WORDS LEFT
3493  014122  016237  000024  003462  40$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
3494  014130  012337  003522                  MOV     (R3)+,E.DB      ;LOAD EXPECTED DATA BUFFER
3495  014134  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3496  014142  001401                          BEQ     41$             ;YES, CHECK CS1
3497  014144  104054                          ERROR   54              ;DATA BUFFER INCORRECT
3498  014146  016237  000000  003440  41$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3499  014154  022737  000002  003624          CMP     #2,WRDCNT       ;CHECK IF FIRST WORDS
3500  014162  001004                          BNE     43$             ;NO, DO NOT WAIT FOR INPUT READY
3501  014164  012700  000024                  MOV     #20.,R0         ;WAIT FOR INPUT READY
3502  014170  005300                  42$:    DEC     R0
3503  014172  001376                          BNE     42$
3504  014174  016237  000010  003450  43$:    MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3505  014202  022701  000001                  CMP     #1,R1           ;CHECK IF LAST WORD
3506  014206  001003                          BNE     44$             ;NO, CONTINUE
3507  014210  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
3508  014216  023737  003500  003440  44$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
3509  014224  001401                          BEQ     45$             ;YES,CHECK CS2
3510  014226  104055                          ERROR   55              ;CS1 INCORRECT
3511  014230  023737  003510  003450  45$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND AND STATUS REG. 2 CORRECT
3512  014236  001401                          BEQ     46$             ;YES, READ NEXT WORD ON SILO
3513  014240  104056                          ERROR   56              ;CS2 INCORRECT
3514  014242  005237  003624          46$:    INC     WRDCNT          ;INCREMENT WORD COUNT
3515  014246  005301                          DEC     R1              ;CHECK IF ALL WORDS READ
3516  014250  001324                          BNE     40$             ;NO, READ NEXT WORD
3517
3518                                  ;;*****************************************************************
3519                                  ;*TEST 15        BUS ADDRESS INHIBIT
```

```
3520                                    ;*
3521                                    ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
3522                                    ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN
3523                                    ;*      RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
3524                                    ;*      SPECIFY A 66 WORD DATA TRANSFER WITH BUS ADDRESS
3525                                    ;*      INHIBIT INCREMENT.  CHECK WORD COUNT, BUS ADDRESS,
3526                                    ;*      INPUT READY, OUTPUT READY, AND MAKE SURE ALL THE
3527                                    ;*      WORDS IN THE SILO ARE THE CORRECT SAME WORD.
3528                                    ;*
3529                                    ;;*****************************************************************
3530  014252  000004                    T15:    SCOPE
3531  014254  012737  000144  001200            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3532  014262  013702  001270                    MOV     $BASE,R2        ;LOAD RK611 BASE
3533  014266  012737  000027  003500            MOV     #WRHEAD,E.CS1   ;LOAD EXPECTED CS1
3534  014274  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3535  014302  012762  000040  000026            MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINT MODE
3536  014310  012737  066762  003504            MOV     #NPRBUF,E.BA    ;LOAD BUS ADDRESS
3537  014316  012762  066762  000004            MOV     #NPRBUF,RKBA(R2)
3538  014324  012737  177677  003502            MOV     #-65.,E.WC      ;LOAD WORD COUNT
3539  014332  012762  177676  000002            MOV     #-66.,RKWC(R2)
3540  014340  012762  000020  000010            MOV     #BAI,RKCS2(R2)  ;SET BUS ADDRESS INCREMENT INHIBIT
3541  014346  012762  000027  000000            MOV     #WRHEAD,RKCS1(R2)  ;ISSUE WRITE HEADER
3542  014354  012700  000312                    MOV     #50.*4+2,R0     ;ISSUE CLOCKS UNTIL READY FOR
3543                                                                    ;  INDEX PULSE
3544  014360  012762  000440  000026  1$:       MOV     #DMD!MCLK,RKMR1(R2)
3545  014366  012762  000040  000026            MOV     #DMD,RKMR1(R2)
3546  014374  005300                            DEC     R0
3547  014376  001370                            BNE     1$
3548  014400  012700  000004                    MOV     #4,R0           ;SIMULATE INDEX PULSE
3549  014404  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)
3550  014412  012762  000640  000026  2$:       MOV     #DMD!MIND!MCLK,RKMR1(R2)
3551  014420  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)
3552  014426  005300                            DEC     R0
3553  014430  001370                            BNE     2$
3554  014432  012762  000040  000026            MOV     #DMD,RKMR1(R2)
3555  014440  012737  000320  003510            MOV     #IR!OR!BAI,E.CS2 ;LOAD EXPECTED CS2
3556  014446  012700  000045          4$:       MOV     #37.,R0         ;SIMULATE 1 NPR TRANSFER
3557  014452  012762  000440  000026  5$:       MOV     #DMD!MCLK,RKMR1(R2)
3558  014460  012762  000040  000026            MOV     #DMD,RKMR1(R2)
3559  014466  005300                            DEC     R0
3560  014470  001370                            BNE     5$
3561  014472  016237  000000  003440            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3562  014500  016237  000004  003444            MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
3563  014506  016237  000002  003442            MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3564  014514  022737  000101  003502            CMP     #65.,E.WC       ;CHECK IF FIRST WORD
3565  014522  001004                            BNE     7$              ;NO, STORE CS2
3566  014524  012700  000024                    MOV     #20.,R0         ;WAIT FOR OUTPUT READY
3567  014530  005300                    6$:     DEC     R0
3568  014532  001376                            BNE     6$
3569  014534  016237  000010  003450  7$:       MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3570  014542  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3571  014550  001401                            BEQ     8$              ;YES, CHECK CS2
3572  014552  104057                            ERROR   57              ;CS1 INCORRECT
3573  014554  023737  003510  003450  8$:       CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3574  014562  001401                            BEQ     9$              ;YES, CHECK BUS ADDRESS
3575  014564  104060                            ERROR   60              ;CS2 INCORRECT
```

```
3576  014566  023737  003504  003444  9$:     CMP     E.BA,T.BA       ;CHECK BUS ADDRESS
3577  014574  001401                          BEQ     10$             ;YES, CHECK WORD COUNT
3578  014576  104061                          ERROR   61              ;BUS ADDRESS INCORRECT
3579  014600  023737  003502  003442  10$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3580  014606  001401                          BEQ     11$             ;YES, CHECK IF ALL WORDS TRANSFERRED
3581  014610  104062                          ERROR   62              ;WORD COUNT INCORRECT
3582  014612  005237  003502          11$:    INC     E.WC            ;INCREMENT WORD COUNT
3583  014616  100713                          BMI     4$              ;CHECK IF FINISHED (NO, BRANCH)
3584  014620  001004                          BNE     12$             ;CHECK IF LAST WORD
3585  014622  012737  000220  003510          MOV     #OR!BAI,E.CS2   ;LOAD EXPECTED COMMAND STATUS REG. 2
3586  014630  000706                          BR      4$              ;PROCESS THE LAST WORD
3587
3588  014632  005037  003502          12$:    CLR     E.WC            ;ADJUST WORD COUNT
3589  014636  012700  000112                  MOV     #2*37.,R0       ;ISSUE ENOUGH CLOCKS FOR
3590  014642  012762  000440  000026  15$:    MOV     #DMD!MCLK,RKMR1(R2)  ;  2 NPR TRANSFERS
3591  014650  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3592  014656  005300                          DEC     R0
3593  014660  001370                          BNE     15$
3594  014662  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3595  014670  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3596  014676  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG.
3597  014704  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT REG.
3598  014712  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3599  014720  001401                          BEQ     16$             ;YES, CHECK CS2
3600  014722  104063                          ERROR   63              ;CS1 INCORRECT
3601  014724  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3602  014732  001401                          BEQ     17$             ;YES, CHECK BUS ADDRESS
3603  014734  104064                          ERROR   64              ;CS2 INCORRECT
3604  014736  023737  003504  003444  17$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
3605  014744  001401                          BEQ     18$             ;YES, CHECK WORD COUNT
3606  014746  104065                          ERROR   65              ;BUS ADDRESS INCORRECT
3607  014750  023737  003502  003442  18$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3608  014756  001401                          BEQ     19$             ;YES, CHECK DATA
3609  014760  104066                          ERROR   66              ;WORD COUNT INCORRECT
3610  014762  012701  000102          19$:    MOV     #66.,R1         ;LOAD NUMBERS OF WORDS LOADED
3611  014766  013737  066762  003522          MOV     NPRBUF,E.DB     ;LOAD EXPECTED DATA BUFFER
3612  014774  005037  003624                  CLR     WRDCNT          ;INITIALIZE WORD COUNT FOR PRINT OUT
3613  015000  016237  000024  003462  25$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
3614  015006  012737  000320  003510          MOV     #IR!OR!BAI,E.CS2 ;LOAD EXPECTED CS2
3615  015014  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3616  015022  001401                          BEQ     26$             ;YES, CONTINUE
3617  015024  104067                          ERROR   67              ;DATA BUFFER INCORRECT
3618  015026  016237  000000  003440  26$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3619  015034  005737  003624                  TST     WRDCNT          ;CHECK IF FIRST WORD
3620  015040  001015                          BNE     28$             ;NO, GET CS2
3621  015042  012700  000024                  MOV     #20.,R0         ;WAIT FOR INPUT READY TO SET
3622  015046  005300                  27$:    DEC     R0
3623  015050  001376                          BNE     27$
3624  015052  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3625  015060  022701  000001                  CMP     #1,R1           ;CHECK IF LAST WORD
3626  015064  001003                          BNE     28$             ;NO, CONTINUE
3627  015066  012737  000120  003510          MOV     #IR!BAI,E.CS2   ;LOAD EXPECTED CS2
3628  015074  023737  003500  003440  28$:    CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1 CORRECT
3629  015102  001401                          BEQ     29$             ;YES, CHECK CS2
3630  015104  104070                          ERROR   70              ;CS1 INCORRECT
3631  015106  023737  003510  003450  29$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG 2 CORRECT
```

```
3632  015114  001401                          BEQ     30$             ;YES, GET NEXT WORD
3633  015116  104071                          ERROR   71              ;CS2 INCORRECT
3634  015120  005237  003624          30$:    INC     WRDCNT          ;INCREMENT WORD COUNT
3635  015124  005301                          DEC     R1              ;DECREMENT WORDS READ
3636  015126  001324                          BNE     25$             ;CHECK IF ALL WORDS READ
3637
3638                                  ;;*****************************************************************
3639                                  ;*TEST 16       NON-EXISTENT MEMORY
3640                                  ;*
3641                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR,  PUT CONTROLLER
3642                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
3643                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
3644                                  ;*      SPECIFY A ONE WORD DATA TO A NON-EXISTENT ADDRESS
3645                                  ;*      (760000) AND MAKE SURE THE NON-EXISTENT  MEMORY ERROR
3646                                  ;*      OCCURS IN THE RK611 CONTROLLER.
3647                                  ;*
3648                                  ;;*****************************************************************
3649  015130  000004                  TST16:  SCOPE
3650  015132  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3651  015140  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
3652  015144  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3653  015152  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINT MODE
3654  015160  012737  160002  003504          MOV     #160002,E.BA    ;LOAD BUS ADDRESS
3655  015166  012762  160000  000004          MOV     #160000,RKBA(R2)
3656  015174  012737  177677  003502          MOV     #-65.,E.WC      ;LOAD WORD COUNT
3657  015202  012762  177676  000002          MOV     #-66.,RKWC(R2)
3658  015210  012737  101626  003500          MOV     #CERR!RDY!BA16!BA17!WRHEAD&<^C<GO>>,E.CS1
3659  015216  012762  001427  000000          MOV     #BA17!BA16!WRHEAD,RKCS1(R2)  ;ISSUE WRITE HEADER
3660  015224  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE CLOCKS UNTIL READY FOR
3661                                                                  ;  INDEX PULSE
3662  015230  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
3663  015236  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3664  015244  005300                          DEC     R0
3665  015246  001370                          BNE     1$
3666  015250  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
3667  015254  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3668  015262  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
3669  015270  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3670  015276  005300                          DEC     R0
3671  015300  001370                          BNE     2$
3672  015302  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3673  015310  012700  000045                  MOV     #37.,R0         ;SIMULATE 1 NPR TRANSFER
3674  015314  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2)
3675  015322  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3676  015330  005300                          DEC     R0
3677  015332  001370                          BNE     5$
3678  015334  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3679  015342  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3680  015350  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
3681  015356  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3682  015364  016237  000014  003454          MOV     RKER(R2),T.ER   ;STORE ERROR REG.
3683  015372  005037  003514                  CLR     E.ER            ;LOAD EXPECTED ERROR REG.
3684  015376  012737  004100  003510          MOV     #IR!NEM,E.CS2   ;LOAD EXPECTED CS2
3685  015404  032737  000200  003450          BIT     #OR,T.CS2
3686  015412  001403                          BEQ     7$
3687  015414  052737  000200  003510          BIS     #OR,E.CS2
```

```
3688  015422  023737  003500  003440  7$:     CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3689  015430  001401                          BEQ     8$              ;YES, CHECK CS2
3690  015432  104072                          ERROR   72              ;CS1 INCORRECT
3691  015434  023737  003510  003450  8$:     CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG. 2 CORRECT
3692  015442  001401                          BEQ     9$              ;YES, CHECK ERROR REG.
3693  015444  104073                          ERROR   73              ;CS2 INCORRECT
3694  015446  023737  003514  003454  9$:     CMP     E.ER,T.ER       ;CHECK ERROR REG CORRECT
3695  015454  001401                          BEQ     10$             ;CHECK BUS ADDRESS
3696  015456  104074                          ERROR   74              ;ERROR REG INCORRECT
3697  015460  023737  003504  003444  10$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
3698  015466  001401                          BEQ     11$             ;YES, CHECK WORD COUNT
3699  015470  104075                          ERROR   75              ;BUS ADDRESS INCORRECT
3700  015472  023737  003502  003442  11$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT REG.
3701  015500  001401                          BEQ     12$             ;YES, CLEAR RK611
3702  015502  104076                          ERROR   76              ;WORD COUNT INCORRECT
3703  015504  012762  100000  000000  12$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3704  015512  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3705  015520  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3706  015526  012737  000200  003500          MOV     #RDY,E.CS1      ;LOAD EXPECTED CS1
3707  015534  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
3708  015542  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG. 1
3709  015550  001401                          BEQ     15$             ;YES, CHECK CS2
3710  015552  104077                          ERROR   77              ;CS1 INCORRECT
3711  015554  023737  003510  003450  15$:    CMP     E.CS2,T.CS2     ;CHECK IF NEM CLEARED
3712  015562  001401                          BEQ     TST17           ;YES, GO ON TO NEXT TEST
3713  015564  104100                          ERROR   100             ;CS2 INCORRECT
3714
3715                                  ;;*****************************************************************
3716                                  ;*TEST 17       BUS ADDRESS BIT 16
3717                                  ;*
3718                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
3719                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO  AN RK06
3720                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
3721                                  ;*      SPECIFY A ONE WORD DATA TRANSFER FROM 200000.
3722                                  ;*      READ THE SILO AND MAKE SURE RIGHT CONTENTS ARE READ.
3723                                  ;*      REPEAT FOR A TWO WORD TRANSFER FROM ADDRESS 177776.
3724                                  ;*      CHECK BUS ADDRESS AND WORD COUNT.
3725                                  ;*
3726                                  ;*      NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
3727                                  ;*             OF MEMORY IS ON THE SYSTEM.
3728                                  ;*
3729                                  ;;*****************************************************************
3730  015566  000004                  TST17:  SCOPE
3731  015570  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3732  015576  005737  046316                  TST     $KT11           ;CHECK FOR MEMORY MANAGEMENT
3733  015602  100004                          BPL     1$              ;NO, BYPASS TEST
3734  015604  022737  002000  046564          CMP     #2000,$LSTBK    ;CHECK IF ENOUGH MEMORY
3735  015612  103417                          BLO     2$              ;YES, DO TEST
3736  015614                          1$:
3737  015614  012737  000001  001200          MOV     #1,$TIMES       ;FORCE INTERATION COUNT TO 1
3738  015622  005227  177777                  INC     #-1             ;ONLY DO ONCE
3739  015626  001007                          BNE     64$             ;NO, GO TO NEXT TEST
3740  015630  104401  054734                  TYPE    ,TSTBY1         ;TYPE TEST N BYPASSED
3741  015634  013746  001220                  MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
3742  015640  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3743  015642  104401  054744                  TYPE    ,TSTBY2
```

```
3744  015646  000137  016702          64$:    JMP     TST20           ;GO TO NEXT TEST
3745
3746  015652  013702  001270          2$:     MOV     $BASE,R2        ;LOAD K611 BASE
3747  015656  012737  002000  172354          MOV     #2000,KIPAR6    ;LOAD PAGE ADDRESS 6 FOR DATA
3748  015664  005237  177572                  INC     SR0             ;TURN ON MEMORY MANAGEMENT
3749  015670  013737  066762  140000          MOV     NPRBUF,140000   ;LOAD WORD IN MEMORY
3750  015676  005037  177572                  CLR     SR0             ;TURN OFF MEMORY MANAGEMENT
3751  015702  012737  015710  001110          MOV     #3$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
3752                                                                  ;  SUBTEST LOOP
3753
3754  015710                          3$:
3755  015710  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3756  015716  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
3757  015724  012762  177777  000002          MOV     #-1,RKWC(R2)
3758  015732  012737  000427  003500          MOV     #BA16!WRHEAD,E.CS1 ;LOAD COMMAND
3759  015740  012762  000427  000000          MOV     #BA16!WRHEAD,RKCS1(R2)
3760  015746  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
3761  015752  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2) ;  INDEX PULSE
3762  015760  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3763  015766  005300                          DEC     R0
3764  015770  001370                          BNE     5$
3765  015772  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
3766  015776  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3767  016004  012762  000640  000026  6$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
3768  016012  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3769  016020  005300                          DEC     R0
3770  016022  001370                          BNE     6$
3771  016024  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3772  016032  012700  000045                  MOV     #37.,R0         ;ISSUE 1 NPR TRANSFER
3773  016036  012762  000440  000026  7$:     MOV     #DMD!MCLK,RKMR1(R2)
3774  016044  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3775  016052  005300                          DEC     R0
3776  016054  001370                          BNE     7$
3777  016056  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3778  016064  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
3779  016072  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WOR
3780  016100  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
3781  016104  005300                  8$:     DEC     R0
3782  016106  001376                          BNE     8$
3783  016110  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3784  016116  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
3785  016124  012737  000002  003504          MOV     #2,E.BA         ;LOAD EXPECTED BUS ADDRESS
3786  016132  005037  003502                  CLR     E.WC            ;LOAD EXPECTED WORD COUNT
3787  016136  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3788  016144  001401                          BEQ     10$             ;YES, CHECK CS2
3789  016146  104101                          ERROR   101             ;CS1 INCORRECT
3790  016150  023737  003510  003450  10$:    CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
3791  016156  001401                          BEQ     11$             ;YES, CHECK BUS ADDRESS CORRECT
3792  016160  104102                          ERROR   102             ;CS2 INCORRECT
3793  016162  023737  003504  003444  11$:    CMP     E.BA,T.BA       ;CHECK IF BUS ADD INCORRECT
3794  016170  001401                          BEQ     12$             ;YES, CHECK WORD COUNT CORRECT
3795  016172  104103                          ERROR   103             ;BUS ADDRESS INCORRECT
3796  016174  023737  003502  003442  12$:    CMP     E.WC,T.WC       ;CHECK IF WORD COUNT CORRECT
3797  016202  001401                          BEQ     13$             ;YES, CHECK DATA BUFFER
3798  016204  104104                          ERROR   104             ;WORD COUNT INCORRECT
3799  016206  016237  000024  003462  13$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
```

```
3800  016214  013737  066762  003522          MOV     NPRBUF,E.DB     ;LOAD EXPECTED DATA BUFFER
3801  016222  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3802  016230  001401                          BEQ     15$             ;YES, CHECK IF LOOP ON ERROR
3803  016232  104105                          ERROR   105             ;DATA INCORRECT
3804  016234  104415                  15$:    SCOP1                   ;CHECK IF LOOP ON ERROR
3805  016236  012737  001777  172354          MOV     #2000-1,KIPAR6  ;LOAD PAGE ADDRESS 6 FOR DATA
3806  016244  005237  177572                  INC     SR0             ;TURN ON MEMORY MANAGEMENT
3807  016250  013737  066762  140076          MOV     NPRBUF,140076   ;LOAD WORDS IN MEMORY
3808  016256  013737  066764  140100          MOV     NPRBUF+2,140100
3809  016264  005037  177572                  CLR     SR0             ;TURN OFF MEMORY MANAGEMENT
3810  016270  012737  016276  001110          MOV     #20$,$LPERR     ;LOAD LOOP ON ERROR LOCATION FOR
3811                                                                  ;  SUBTEST LOOP
3812
3813  016276                          20$:
3814  016276  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3815  016304  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
3816  016312  012737  177777  003502          MOV     #-1,E.WC        ;LOAD WORD COUNT REG.
3817  016320  012762  177776  000002          MOV     #-2,RKWC(R2)
3818  016326  005037  003504                  CLR     E.BA            ;LOAD BUS ADDRESS
3819  016332  012762  177776  000004          MOV     #177776,RKBA(R2)
3820  016340  012737  000427  003500          MOV     #BA16!WRHEAD,E.CS1 ;LOAD COMMAND
3821  016346  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
3822  016354  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
3823  016360  012762  000440  000026  21$:    MOV     #DMD!MCLK,RKMR1(R2) ; INDEX PULSE
3824  016366  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3825  016374  005300                          DEC     R0
3826  016376  001370                          BNE     21$
3827  016400  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
3828  016404  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3829  016412  012762  000640  000026  22$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
3830  016420  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3831  016426  005300                          DEC     R0
3832  016430  001370                          BNE     22$
3833  016432  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3834  016440  012701  000002                  MOV     #2,R1           ;ISSUE 2 NPR TRANSFERS
3835  016444  012700  000045          23$:    MOV     #37.,R0
3836  016450  012762  000440  000026  24$:    MOV     #DMD!MCLK,RKMR1(R2)
3837  016456  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3838  016464  005300                          DEC     R0
3839  016466  001370                          BNE     24$
3840  016470  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3841  016476  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG.
3842  016504  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
3843  016512  005737  003502                  TST     E.WC            ;CHECK IF FIRST WORD
3844  016516  001404                          BEQ     26$             ;NO, GET CS2
3845  016520  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
3846  016524  005300                  25$:    DEC     R0
3847  016526  001376                          BNE     25$
3848  016530  016237  000010  003450  26$:    MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
3849  016536  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
3850  016544  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG 1 CORRECT
3851  016552  001401                          BEQ     28$             ;YES, CHECK CS2
3852  016554  104101                          ERROR   101             ;CS1 INCORRECT
3853  016556  023737  003510  003450  28$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG 2 CORRECT
3854  016564  001401                          BEQ     29$             ;YES, CHECK BUS ADDRESS
3855  016566  104102                          ERROR   102             ;CS2 INCORRECT
```

```
3856  016570  023737  003504  003444  29$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
3857  016576  001401                          BEQ     30$             ;YES, CHECK WORD COUNT
3858  016600  104103                          ERROR   103             ;BUS ADDRESS INCORRECT
3859  016602  023737  003502  003442  30$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
3860  016610  001401                          BEQ     31$             ;YES, GET TEXT WORD
3861  016612  104104                          ERROR   104             ;WORD COUNT INCORRECT
3862  016614  062737  000002  003504  31$:    ADD     #2,E.BA         ;INCREMENT BUS ADDRESS AND
3863  016622  005237  003502                  INC     E.WC            ;  WORD COUNT
3864  016626  005301                          DEC     R1              ;CHECK IF FINISHED
3865  016630  001305                          BNE     23$             ;NO, GET SECOND WORD
3866  016632  012700  000002                  MOV     #2,R0           ;LOAD COUNT AND ADDRESS WORD
3867  016636  012703  066762                  MOV     #NPRBUF,R3      ;  DATA COMPARE
3868  016642  016237  000024  003462  35$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
3869  016650  012337  003522                  MOV     (R3)+,E.DB      ;GET EXPECTED DATA
3870  016654  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF CORRECT
3871  016662  001401                          BEQ     36$             ;YES, CHECK IF FINISHED
3872  016664  104105                          ERROR   105             ;DATA BUFFER INCORRECT
3873  016666  005300                  36$:    DEC     R0              ;CHECK IF FINISHED
3874  016670  001364                          BNE     35$             ;NO READ SECOND WORD
3875  016672  104415                          SCOP1                   ;CHECK IF LOOP ON ERROR
3876  016674  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3877
3878                                  ;;*****************************************************************
3879                                  ;*TEST 20       BUS ADDRESS BIT 17
3880                                  ;*
3881                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
3882                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
3883                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEADER 0, DRIVE 0.
3884                                  ;*      SPECIFY A ONE WORD DATA TRANSFER FROM 400000.
3885                                  ;*      READ THE SILO AND MAKE SURE RIGHT CONTENTS ARE READ.
3886                                  ;*      REPEAT FOR A TWO WORD TRANSFER FROM ADDRESS 377776.
3887                                  ;*      CHECK BUS ADDRESS AND WORD COUNT.
3888                                  ;*
3889                                  ;*      NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
3890                                  ;*             OF MEMORY IS ON THE SYSTEM.
3891                                  ;*
3892                                  ;*****************************************************************
3893  016702  000004          TST20:  SCOPE
3894  016704  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
3895  016712  005737  046316                  TST     $KT11           ;CHECK FOR MEMORY MANAGEMENT
3896  016716  100004                          BPL     1$              ;NO, BYPASS TEST
3897  016720  022737  004000  046564          CMP     #4000,$LSTBK    ;CHECK IF ENOUGH MEMORY
3898  016726  103417                          BLO     2$              ;YES, DO TEST
3899  016730                          1$:
3900  016730  012737  000001  001200          MOV     #1,$TIMES       ;FORCE INTERATION COUNT TO 1
3901  016736  005227  177777                  INC     #-1             ;ONLY DO ONCE
3902  016742  001007                          BNE     64$             ;NO, GO TO NEXT TEST
3903  016744  104401  054734                  TYPE    ,TSTBY1         ;TYPE TEST N BYPASSED
3904  016750  013746  001220                  MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
3905  016754  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3906  016756  104401  054744                  TYPE    ,TSTBY2
3907  016762  000137  020016          64$:    JMP     TST21           ;GO TO NEXT TEST
3908
3909  016766  013702  001270          2$:     MOV     $BASE,R2        ;LOAD K611 BASE
3910  016772  012737  004000  172354          MOV     #4000,KIPAR6    ;LOAD PAGE ADDRESS 6 FOR DATA
3911  017000  005237  177572                  INC     SR0             ;TURN ON MEMORY MANAGEMENT
```

```
3912  017004  013737  066762  140000          MOV     NPRBUF,140000   ;LOAD WORD IN MEMORY
3913  017012  005037  177572                  CLR     SR0             ;TURN OFF MEMORY MANAGEMENT
3914  017016  012737  017024  001110          MOV     #3$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
3915                                                                  ;  SUBTEST LOOP
3916
3917  017024                          3$:
3918  017024  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3919  017032  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
3920  017040  012762  177777  000002          MOV     #-1,RKWC(R2)
3921  017046  012737  001027  003500          MOV     #BA17!WRHEAD,E.CS1 ;LOAD COMMAND
3922  017054  012762  001027  000000          MOV     #BA17!WRHEAD,RKCS1(R2)
3923  017062  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
3924  017066  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2) ;  INDEX PULSE
3925  017074  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3926  017102  005300                          DEC     R0
3927  017104  001370                          BNE     5$
3928  017106  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
3929  017112  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3930  017120  012762  000640  000026  6$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
3931  017126  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3932  017134  005300                          DEC     R0
3933  017136  001370                          BNE     6$
3934  017140  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3935  017146  012700  000045                  MOV     #37.,R0         ;ISSUE 1 NPR TRANSFER
3936  017152  012762  000440  000026  7$:     MOV     #DMD!MCLK,RKMR1(R2)
3937  017160  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3938  017166  005300                          DEC     R0
3939  017170  001370                          BNE     7$
3940  017172  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
3941  017200  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
3942  017206  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WOR
3943  017214  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
3944  017220  005300                  8$:     DEC     R0
3945  017222  001376                          BNE     8$
3946  017224  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
3947  017232  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
3948  017240  012737  000002  003504          MOV     #2,E.BA         ;LOAD EXPECTED BUS ADDRESS
3949  017246  005037  003502                  CLR     E.WC            ;LOAD EXPECTED WORD COUNT
3950  017252  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
3951  017260  001401                          BEQ     10$             ;YES, CHECK CS2
3952  017262  104101                          ERROR   101             ;CS1 INCORRECT
3953  017264  023737  003510  003450  10$:    CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
3954  017272  001401                          BEQ     11$             ;YES, CHECK BUS ADDRESS CORRECT
3955  017274  104102                          ERROR   102             ;CS2 INCORRECT
3956  017276  023737  003504  003444  11$:    CMP     E.BA,T.BA       ;CHECK IF BUS ADD INCORRECT
3957  017304  001401                          BEQ     12$             ;YES, CHECK WORD COUNT CORRECT
3958  017306  104103                          ERROR   103             ;BUS ADDRESS INCORRECT
3959  017310  023737  003502  003442  12$:    CMP     E.WC,T.WC       ;CHECK IF WORD COUNT CORRECT
3960  017316  001401                          BEQ     13$             ;YES, CHECK DATA BUFFER
3961  017320  104104                          ERROR   104             ;WORD COUNT INCORRECT
3962  017322  016237  000024  003462  13$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
3963  017330  013737  066762  003522          MOV     NPRBUF,E.DB     ;LOAD EXPECTED DATA BUFFER
3964  017336  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
3965  017344  001401                          BEQ     15$             ;YES, CHECK IF LOOP ON ERROR
3966  017346  104105                          ERROR   105             ;DATA INCORRECT
3967  017350  104415                  15$:    SCOP1                   ;CHECK IF LOOP ON ERROR
```

```
3968  017352  012737  003777  172354          MOV     #4000-1,KIPAR6  ;LOAD PAGE ADDRESS 6 FOR DATA
3969  017360  005237  177572                  INC     SR0             ;TURN ON MEMORY MANAGEMENT
3970  017364  013737  066762  140076          MOV     NPRBUF,140076   ;LOAD WORDS IN MEMORY
3971  017372  013737  066764  140100          MOV     NPRBUF+2,140100
3972  017400  005037  177572                  CLR     SR0             ;TURN OFF MEMORY MANAGEMENT
3973  017404  012737  017412  001110          MOV     #20$,$LPERR     ;LOAD LOOP ON ERROR LOCATION FOR
3974                                                                  ;  SUBTEST LOOP
3975
3976  017412                          20$:
3977  017412  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
3978  017420  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
3979  017426  012737  177777  003502          MOV     #-1,E.WC        ;LOAD WORD COUNT REG.
3980  017434  012762  177776  000002          MOV     #-2,RKWC(R2)
3981  017442  005037  003504                  CLR     E.BA            ;LOAD BUS ADDRESS
3982  017446  012762  177776  000004          MOV     #177776,RKBA(R2)
3983  017454  012737  001027  003500          MOV     #BA17!WRHEAD,E.CS1 ;LOAD COMMAND
3984  017462  012762  000427  000000          MOV     #BA16!WRHEAD,RKCS1(R2)
3985  017470  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
3986  017474  012762  000440  000026  21$:    MOV     #DMD!MCLK,RKMR1(R2) ; INDEX PULSE
3987  017502  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3988  017510  005300                          DEC     R0
3989  017512  001370                          BNE     21$
3990  017514  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
3991  017520  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3992  017526  012762  000640  000026  22$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
3993  017534  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
3994  017542  005300                          DEC     R0
3995  017544  001370                          BNE     22$
3996  017546  012762  000040  000026          MOV     #DMD,RKMR1(R2)
3997  017554  012701  000002                  MOV     #2,R1           ;ISSUE 2 NPR TRANSFERS
3998  017560  012700  000045          23$:    MOV     #37.,R0
3999  017564  012762  000440  000026  24$:    MOV     #DMD!MCLK,RKMR1(R2)
4000  017572  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4001  017600  005300                          DEC     R0
4002  017602  001370                          BNE     24$
4003  017604  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4004  017612  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG.
4005  017620  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
4006  017626  005737  003502                  TST     E.WC            ;CHECK IF FIRST WORD
4007  017632  001404                          BEQ     26$             ;NO, GET CS2
4008  017634  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
4009  017640  005300                  25$:    DEC     R0
4010  017642  001376                          BNE     25$
4011  017644  016237  000010  003450  26$:    MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4012  017652  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
4013  017660  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG 1 CORRECT
4014  017666  001401                          BEQ     28$             ;YES, CHECK CS2
4015  017670  104101                          ERROR   101             ;CS1 INCORRECT
4016  017672  023737  003510  003450  28$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG 2 CORRECT
4017  017700  001401                          BEQ     29$             ;YES, CHECK BUS ADDRESS
4018  017702  104102                          ERROR   102             ;CS2 INCORRECT
4019  017704  023737  003504  003444  29$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
4020  017712  001401                          BEQ     30$             ;YES, CHECK WORD COUNT
4021  017714  104103                          ERROR   103             ;BUS ADDRESS INCORRECT
4022  017716  023737  003502  003442  30$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
4023  017724  001401                          BEQ     31$             ;YES, GET TEXT WORD
```

```
 4024  017726  104104                        ERROR   104             ;WORD COUNT INCORRECT
 4025  017730  062737  000002  003504  31$:  ADD     #2,E.BA         ;INCREMENT BUS ADDRESS AND
 4026  017736  005237  003502                INC     E.WC            ;  WORD COUNT
 4027  017742  005301                        DEC     R1              ;CHECK IF FINISHED
 4028  017744  001305                        BNE     23$             ;NO, GET SECOND WORD
 4029  017746  012700  000002                MOV     #2,R0           ;LOAD COUNT AND ADDRESS WORD
 4030  017752  012703  066762                MOV     #NPRBUF,R3      ;  DATA COMPARE
 4031  017756  016237  000024  003462  35$:  MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
 4032  017764  012337  003522                MOV     (R3)+,E.DB      ;GET EXPECTED DATA
 4033  017770  023737  003522  003462        CMP     E.DB,T.DB       ;CHECK IF CORRECT
 4034  017776  001401                        BEQ     36$             ;YES, CHECK IF FINISHED
 4035  020000  104105                        ERROR   105             ;DATA BUFFER INCORRECT
 4036  020002  005300                36$:    DEC     R0              ;CHECK IF FINISHED
 4037  020004  001364                        BNE     35$             ;NO READ SECOND WORD
 4038  020006  104415                        SCOPE                   ;CHECK IF LOOP ON ERROR
 4039  020010  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 4040
 4041                          ;;*****************************************************************
 4042                          ;*TEST 21       ADDRESSING GREATER THAN 96K
 4043                          ;*
 4044                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
 4045                          ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
 4046                          ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEADER 0, DRIVE 0.
 4047                          ;*      SPECIFY A ONE WORD DATA TRANSFER FROM 600000.
 4048                          ;*      READ THE SILO AND MAKE SURE RIGHT CONTENTS ARE READ.
 4049                          ;*      REPEAT FOR A TWO WORD TRANSFER FROM ADDRESS 577776.
 4050                          ;*      CHECK BUS ADDRESS AND WORD COUNT.
 4051                          ;*
 4052                          ;*      NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
 4053                          ;*             OF MEMORY IS ON THE SYSTEM.
 4054                          ;*
 4055                          ;;*****************************************************************
 4056  020016  000004          TST21:  SCOPE
 4057  020020  012737  000144  001200        MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
 4058  020026  005737  046316                TST     $KT11           ;CHECK FOR MEMORY MANAGEMENT
 4059  020032  100004                        BPL     1$              ;NO, BYPASS TEST
 4060  020034  022737  006000  046564        CMP     #6000,$LSTBK    ;CHECK IF ENOUGH MEMORY
 4061  020042  103417                        BLO     2$              ;YES, DO TEST
 4062  020044                          1$:
 4063  020044  012737  000001  001200        MOV     #1,$TIMES       ;FORCE INTERATION COUNT TO 1
 4064  020052  005227  177777                INC     #-1             ;ONLY DO ONCE
 4065  020056  001007                        BNE     64$             ;NO, GO TO NEXT TEST
 4066  020060  104401  054734                TYPE    ,TSTBY1         ;TYPE TEST N BYPASSED
 4067  020064  013746  001220                MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
 4068  020070  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4069  020072  104401  054744                TYPE    ,TSTBY2
 4070  020076  000137  021132          64$:  JMP     @TST22          ;GO TO NEXT TEST
 4071
 4072  020102  013702  001270          2$:   MOV     $BASE,R2        ;LOAD K611 BASE
 4073  020106  012737  006000  172354        MOV     #6000,KIPAR6    ;LOAD PAGE ADDRESS 6 FOR DATA
 4074  020114  005237  177572                INC     SR0             ;TURN ON MEMORY MANAGEMENT
 4075  020120  013737  066762  140000        MOV     NPRBUF,140000   ;LOAD WORD IN MEMORY
 4076  020126  005037  177572                CLR     SR0             ;TURN OFF MEMORY MANAGEMENT
 4077  020132  012737  020140  001110        MOV     #3$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
 4078                                                                ;  SUBTEST LOOP
 4079
```

```
4080  020140                          3$:
4081  020140  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
4082  020146  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
4083  020154  012762  177777  000002          MOV     #-1,RKWC(R2)
4084  020162  012737  001427  003500          MOV     #BA16!BA17!WRHEAD,E.CS1 ;LOAD COMMAND
4085  020170  012762  001427  000000          MOV     #BA16!BA17!WRHEAD,RKCS1(R2)
4086  020176  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
4087  020202  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2) ;  INDEX PULSE
4088  020210  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4089  020216  005300                          DEC     R0
4090  020220  001370                          BNE     5$
4091  020222  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
4092  020226  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4093  020234  012762  000640  000026  6$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
4094  020242  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4095  020250  005300                          DEC     R0
4096  020252  001370                          BNE     6$
4097  020254  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4098  020262  012700  000045                  MOV     #37.,R0         ;ISSUE 1 NPR TRANSFER
4099  020266  012762  000440  000026  7$:     MOV     #DMD!MCLK,RKMR1(R2)
4100  020274  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4101  020302  005300                          DEC     R0
4102  020304  001370                          BNE     7$
4103  020306  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4104  020314  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
4105  020322  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WOR
4106  020330  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
4107  020334  005300                  8$:     DEC     R0
4108  020336  001370                          BNE     8$
4109  020340  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4110  020346  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
4111  020354  012737  000002  003504          MOV     #2,E.BA         ;LOAD EXPECTED BUS ADDRESS
4112  020362  005037  003502                  CLR     E.WC            ;LOAD EXPECTED WORD COUNT
4113  020366  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG. 1 CORRECT
4114  020374  001401                          BEQ     10$             ;YES, CHECK CS2
4115  020376  104101                          ERROR   101             ;CS1 INCORRECT
4116  020400  023737  003510  003450  10$:    CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
4117  020406  001401                          BEQ     11$             ;YES, CHECK BUS ADDRESS CORRECT
4118  020410  104102                          ERROR   102             ;CS2 INCORRECT
4119  020412  023737  003504  003444  11$:    CMP     E.BA,T.BA       ;CHECK IF BUS ADD INCORRECT
4120  020420  001401                          BEQ     12$             ;YES, CHECK WORD COUNT CORRECT
4121  020422  104103                          ERROR   103             ;BUS ADDRESS INCORRECT
4122  020424  023737  003502  003442  12$:    CMP     E.WC,T.WC       ;CHECK IF WORD COUNT CORRECT
4123  020432  001401                          BEQ     13$             ;YES, CHECK DATA BUFFER
4124  020434  104104                          ERROR   104             ;WORD COUNT INCORRECT
4125  020436  016237  000024  003462  13$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
4126  020444  013737  066762  003522          MOV     NPRBUF,E.DB     ;LOAD EXPECTED DATA BUFFER
4127  020452  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
4128  020460  001401                          BEQ     15$             ;YES, CHECK IF LOOP ON ERROR
4129  020462  104105                          ERROR   105             ;DATA INCORRECT
4130  020464  104415                  15$:    SCOP1                   ;CHECK IF LOOP ON ERROR
4131  020466  012737  005777  172354          MOV     #6000-1,KIPAR6  ;LOAD PAGE ADDRESS 6 FOR DATA
4132  020474  005237  177572                  INC     SR0             ;TURN ON MEMORY MANAGEMENT
4133  020500  013737  066762  140076          MOV     NPRBUF,140076   ;LOAD WORDS IN MEMORY
4134  020506  013737  066764  140100          MOV     NPRBUF+2,140100
4135  020514  005037  177572                  CLR     SR0             ;TURN OFF MEMORY MANAGEMENT
```

```
4136  020520  012737  020526  001110          MOV     #20$,$LPERR     ;LOAD LOOP ON ERROR LOCATION FOR
4137                                                                  ;  SUBTEST LOOP
4138
4139  020526                          20$:
4140  020526  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
4141  020534  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
4142  020542  012737  177777  003502          MOV     #-1,E.WC        ;LOAD WORD COUNT REG.
4143  020550  012762  177776  000002          MOV     #-2,RKWC(R2)
4144  020556  005037  003504                  CLR     E.BA            ;LOAD BUS ADDRESS
4145  020562  012762  177776  000004          MOV     #177776,RKBA(R2)
4146  020570  012737  001427  003500          MOV     #BA16!BA17!WRHEAD,E.CS1 ;LOAD COMMAND
4147  020576  012762  001027  000000          MOV     #BA17!WRHEAD,RKCS1(R2)
4148  020604  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
4149  020610  012762  000440  000026  21$:    MOV     #DMD!MCLK,RKMR1(R2) ; INDEX PULSE
4150  020616  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4151  020624  005300                          DEC     R0
4152  020626  001370                          BNE     21$
4153  020630  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
4154  020634  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4155  020642  012762  000640  000026  22$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
4156  020650  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4157  020656  005300                          DEC     R0
4158  020660  001370                          BNE     22$
4159  020662  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4160  020670  012701  000002                  MOV     #2,R1           ;ISSUE 2 NPR TRANSFERS
4161  020674  012700  000045          23$:    MOV     #37.,R0
4162  020700  012762  000440  000026  24$:    MOV     #DMD!MCLK,RKMR1(R2)
4163  020706  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4164  020714  005300                          DEC     R0
4165  020716  001370                          BNE     24$
4166  020720  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4167  020726  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG.
4168  020734  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
4169  020742  005737  003502                  TST     E.WC            ;CHECK IF FIRST WORD
4170  020746  001404                          BEQ     26$             ;NO, GET CS2
4171  020750  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
4172  020754  005300                  25$:    DEC     R0
4173  020756  001376                          BNE     25$
4174  020760  016237  000010  003450  26$:    MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4175  020766  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
4176  020774  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND STATUS REG 1 CORRECT
4177  021002  001401                          BEQ     28$             ;YES, CHECK CS2
4178  021004  104101                          ERROR   101             ;CS1 INCORRECT
4179  021006  023737  003510  003450  28$:    CMP     E.CS2,T.CS2     ;CHECK COMMAND STATUS REG 2 CORRECT
4180  021014  001401                          BEQ     29$             ;YES, CHECK BUS ADDRESS
4181  021016  104102                          ERROR   102             ;CS2 INCORRECT
4182  021020  023737  003504  003444  29$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
4183  021026  001401                          BEQ     30$             ;YES, CHECK WORD COUNT
4184  021030  104103                          ERROR   103             ;BUS ADDRESS INCORRECT
4185  021032  023737  003502  003442  30$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
4186  021040  001401                          BEQ     31$             ;YES, GET TEXT WORD
4187  021042  104104                          ERROR   104             ;WORD COUNT INCORRECT
4188  021044  062737  000002  003504  31$:    ADD     #2,E.BA         ;INCREMENT BUS ADDRESS AND
4189  021052  005237  003502                  INC     E.WC            ;  WORD COUNT
4190  021056  005301                          DEC     R1              ;CHECK IF FINISHED
4191  021060  001305                          BNE     23$             ;NO, GET SECOND WORD
```

```
                                                MOV     #2,R0           ;LOAD COUNT AND ADDRESS WORD
                                                MOV     #NPRBUF,R3      ;  DATA COMPARE
                                        35$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
                                                MOV     (R3)+,E.DB      ;GET EXPECTED DATA
                                                CMP     E.DB,T.DB       ;CHECK IF CORRECT
                                                BEQ     36$             ;YES, CHECK IF FINISHED
                                                ERROR   105             ;DATA BUFFER INCORRECT
                                        36$:    DEC     R0              ;CHECK IF FINISHED
                                                BNE     35$             ;NO READ SECOND WORD
                                                SCOP1                   ;CHECK IF LOOP ON ERROR
                                                MOV     #CCLR,RKCS1(R2) ;CLEAR RK611

                                ;;*****************************************************************
                                ;*TEST 22       UNIBUS PARITY ERROR
                                ;*
                                ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                                ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                                ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                ;*      SPECIFY A ONE WORD DATA TRANSFER FROM A LOCATION
                                ;*      WITH BAD PARITY.  MAKE SURE A UNIBUS PARITY ERROR
                                ;*      OCCURS.
                                ;*
                                ;*      REPEAT FOR A ONE WORD DATA TRANSFER FROM THE
                                ;*      LOCATION PRIOR TO THE LOCATION WITH BAD PARITY.
                                ;*      MAKE SURE UNIBUS PARITY ERROR DOES NOT OCCUR.  REPEAT
                                ;*      FOR A ONE WORD DATA TRANSFER FROM THE LOCATION AFTER
                                ;*      THE LOCATION WITH BAD PARITY.  MAKE SURE UNIBUS PARITY
                                ;*      ERROR DOES NOT OCCUR.
                                ;*
                                ;*      REPEAT FOR A TWO WORD DATA TRANSFER STARTING WITH
                                ;*      THE ADDRESS PRIOR TO THE LOCATION WITH BAD PARITY.
                                ;*      MAKE SURE UNIBUS PARITY ERROR DOES OCCUR.
                                ;*
                                ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
                                ;*              EXISTS FOR SPECIFIED LOCATION.
                                ;*
                                ;;*****************************************************************
 021132  000004                 T$T22:  SCOPE
 021134  012737  000144  001200         MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
 021142  005737  003630                 TST     MEMPAR          ;CHECK IF MEMORY PARITY AVAILABLE
 021146  001017                         BNE     1$              ;YES, DO TEST
 021150  012737  000001  001200         MOV     #1,$TIMES       ;FORCE INTERATION COUNT TO 1
 021156  005227  177777                 INC     #-1             ;ONLY DO ONCE
 021162  001007                         BNE     64$             ;NO, GO TO NEXT TEST
 021164  104401  054734                 TYPE    ,T$TBY1         ;TYPE TEST N BYPASSED
 021170  013746  001220                 MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
 021174  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 021176  104401  054744                 TYPE    ,T$TBY2
 021202  000137  022370         64$:    JMP     T$T23           ;GO TO NEXT TEST

 021206  013702  001270         1$:     MOV     $BASE,R2        ;LOAD RK611 BASE
 021212  004737  044702                 JSR     PC,WRTPAR       ;GENERATE BAD PARITY
 021216  012737  021224  001110         MOV     #2$,$LPERR      ;LOAD LOOP ON ERROR LOCATION FOR
                                                                ;  SUBTEST LOOP

 021224                         2$:
```

```
 4248  021224  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 4249  021232  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 4250  021240  012762  177777  000002          MOV     #-1,RKWC(R2)    ;LOAD WORD COUNT NEG
 4251  021246  005037  003502                  CLR     E.WC
 4252  021252  012762  067220  000004          MOV     #BADPAR,RKBA(R2);LOAD BUS ADDRESS
 4253  021260  012737  067222  003504          MOV     #BADPAR+2,E.BA
 4254  021266  012737  100226  003500          MOV     #CERR!RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
 4255  021274  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2) ;LOAD COMMAND
 4256  021302  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
 4257  021306  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2) ;  INDEX PULSE
 4258  021314  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 4259  021322  005300                          DEC     R0
 4260  021324  001370                          BNE     5$
 4261  021326  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
 4262  021332  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
 4263  021340  012762  000640  000026  6$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
 4264  021346  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
 4265  021354  005300                          DEC     R0
 4266  021356  001370                          BNE     6$
 4267  021360  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 4268  021366  012700  000045                  MOV     #37.,R0         ;ISSUE 1 NPR TRANSFER
 4269  021372  012762  000440  000026  7$:     MOV     #DMD!MCLK,RKMR1(R2)
 4270  021400  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 4271  021406  005300                          DEC     R0
 4272  021410  001370                          BNE     7$
 4273  021412  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG.1
 4274  021420  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
 4275  021426  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
 4276  021434  016237  000014  003454          MOV     RKER(R2),T.ER   ;STORE ERROR REG
 4277  021442  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
 4278  021446  005300                  8$:     DEC     R0
 4279  021450  001376                          BNE     8$
 4280  021452  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG.2
 4281  021460  012737  020300  003510          MOV     #IR!OR!UPE,E.CS2 ;LOAD EXPECTED CS2
 4282  021466  005037  003514                  CLR     E.ER            ;LOAD EXPECTED ERROR REG
 4283  021472  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
 4284  021500  001401                          BEQ     10$             ;YES CHECK CS2
 4285  021502  104106                          ERROR   106             ;CS1 INCORRECT
 4286  021504  023737  003510  003450  10$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
 4287  021512  001401                          BEQ     11$             ;YES, CHECK BUS ADDRESS
 4288  021514  104107                          ERROR   107             ;CS2 INCORRECT
 4289  021516  023737  003504  003444  11$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
 4290  021524  001401                          BEQ     12$             ;YES, CHECK WORD COUNT
 4291  021526  104110                          ERROR   110             ;BUS ADDRESS INCORRECT
 4292  021530  023737  003502  003442  12$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
 4293  021536  001401                          BEQ     13$             ;YES,CHECK ERROR REG.
 4294  021540  104111                          ERROR   111             ;WORD COUNT INCORRECT
 4295  021542  023737  003514  003454  13$:    CMP     E.ER,T.ER       ;CHECK ERROR REG CORRECT
 4296  021550  001401                          BEQ     14$             ;YES,CLEAR ERROR
 4297  021552  104116                          ERROR   116             ;ERROR REG CORRECT
 4298  021554  012762  100000  000000  14$:    MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 4299  021562  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND STATUS REG. 1
 4300  021570  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND STATUS REG. 2
 4301  021576  012737  000200  003500          MOV     #RDY,E.CS1      ;LOAD EXPECTED CS1
 4302  021604  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
 4303  021612  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
```

```
4304  021620  001401                          BEQ     15$             ;YES, CHECK CS2
4305  021622  104117                          ERROR   117             ;CS1 INCORRECT
4306  021624  023737  003510  003450  15$:    CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
4307  021632  001401                          BEQ     16$             ;YES, CONTINUE
4308  021634  104120                          ERROR   120             ;CS2 INCORRECT
4309  021636  104415                  16$:    SCOP1                   ;CHECK IF LOOP ON ERROR
4310  021640  012737  021646  001110          MOV     #20$,$LPERR     ;LOAD LOOP ON ERROR LOCATION FOR
4311                                                                  ;  SUBTEST LOOP
4312
4313  021646                          20$:
4314  021646  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
4315  021654  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611-IN-DIAGNOSTIC MODE
4316  021662  012762  177776  000002          MOV     #-2,RKWC(R2)    ;LOAD WORD COUNT REG
4317  021670  012737  177777  003502          MOV     #-1,E.WC
4318  021676  012762  067216  000004          MOV     #BADPAR-2,RKBA(R2) ;LOAD BUS ADDRESS REG
4319  021704  012737  067220  003504          MOV     #BADPAR,E.BA
4320  021712  012737  000027  003500          MOV     #WRHEAD,E.CS1   ;LOAD EXPECTED CS1
4321  021720  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2) ;LOAD COMMAND
4322  021726  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
4323  021732  012762  000440  000026  21$:    MOV     #DMD!MCLK,RKMR1(R2) ;  INDEX PULSE
4324  021740  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4325  021746  005300                          DEC     R0
4326  021750  001370                          BNE     21$
4327  021752  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
4328  021756  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4329  021764  012762  000640  000026  22$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
4330  021772  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4331  022000  005300                          DEC     R0
4332  022002  001370                          BNE     22$
4333  022004  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4334  022012  012700  000045                  MOV     #37.,R0         ;ISSUE 1 NPR TRANSFER
4335  022016  012762  000440  000026  23$:    MOV     #DMD!MCLK,RKMR1(R2)
4336  022024  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4337  022032  005300                          DEC     R0
4338  022034  001370                          BNE     23$
4339  022036  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4340  022044  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
4341  022052  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
4342  022060  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
4343  022064  005300                  24$:    DEC     R0
4344  022066  001376                          BNE     24$
4345  022070  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4346  022076  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
4347  022104  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4348  022112  001401                          BEQ     25$             ;YES,CHECK CS2
4349  022114  104112                          ERROR   112             ;CS1 INCORRECT
4350  022116  023737  003510  003450  25$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4351  022124  001401                          BEQ     26$             ;YES, CHECK BUS ADDRESS
4352  022126  104113                          ERROR   113             ;CS2 INCORRECT
4353  022130  023737  003504  003444  26$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
4354  022136  001401                          BEQ     27$             ;YES, CHECK WORD COUNT
4355  022140  104114                          ERROR   114             ;BUS ADDRESS INCORRECT
4356  022142  023737  003502  003442  27$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
4357  022150  001401                          BEQ     28$             ;YES, DO NEXT NPR TRANSFER
4358  022152  104115                          ERROR   115             ;WORD COUNT INCORRECT
4359  022154  012700  000045          28$:    MOV     #37.,R0         ;ISSUE 1 NPR TRANSFER
```

```
 4360  022160  012762  000440  000026  30$:    MOV     #DMD!MCLK,RKMR1(R2)
 4361  022166  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 4362  022174  005300                          DEC     R0
 4363  022176  001370                          BNE     30$
 4364  022200  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
 4365  022206  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
 4366  022214  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
 4367  022222  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
 4368  022230  016237  000014  003454          MOV     RKER(R2),T.ER   ;STORE ERROR REG
 4369  022236  012737  100226  003500          MOV     #CERR!RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
 4370  022244  012737  020300  003510          MOV     #OR!IR!UPE,E.CS2;LOAD EXPECTED CS2
 4371  022252  005237  003502                  INC     E.WC            ;LOAD EXPECTED WORD COUNT
 4372  022256  062737  000002  003504          ADD     #2,E.BA         ;LOAD EXPECTED BUS ADDRESS
 4373  022264  005037  003514                  CLR     E.ER            ;LOAD EXPECTED WORD COUNT
 4374  022270  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
 4375  022276  001401                          BEQ     31$             ;YES, CHECK CS2
 4376  022300  104106                          ERROR   106             ;CS1 INCORRECT
 4377  022302  023737  003510  003450  31$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
 4378  022310  001401                          BEQ     32$             ;YES, CHECK BUS ADDRESS
 4379  022312  104107                          ERROR   107             ;CS2 INCORRECT
 4380  022314  023737  003504  003444  32$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
 4381  022322  001401                          BEQ     33$             ;YES, CHECK WORD COUNT
 4382  022324  104110                          ERROR   110             ;BUS ADDRESS INCORRECT
 4383  022326  023737  003502  003442  33$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
 4384  022334  001401                          BEQ     34$             ;YES, CHECK ERROR REG
 4385  022336  104111                          ERROR   111             ;WORD COUNT INCORRECT
 4386  022340  023737  003514  003454  34$:    CMP     E.ER,T.ER       ;CHECK ERROR REG CORRECT
 4387  022346  001401                          BEQ     35$             ;YES, CONTINUE
 4388  022350  104116                          ERROR   116             ;WORD COUNT INCORRECT
 4389  022352  104415                  35$:    SCOP1                   ;CHECK IF WORD ON ERROR
 4390  022354  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 4391  022362  012737  000000  067220          MOV     #0,BADPAR       ;WRITE GOOD PARITY
 4392
 4393                                  ;;*****************************************************************
 4394                                  ;*TEST 23       SILO FILL IN 18 BIT MODE
 4395                                  ;*
 4396                                  ;*      CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER
 4397                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
 4398                                  ;*      IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0. DRIVE 0,
 4399                                  ;*      SPECIFY 66  WORD DATA TRANSFER.  CLOCK
 4400                                  ;*      ALL 66 WORD INTO THE SILO.   CHECK THAT ALL 66
 4401                                  ;*      WORDS ARE CORRECT (16 LEAST SIGNIFICANT BITS).
 4402                                  ;*
 4403                                  ;;*****************************************************************
 4404  022370  000004                  TST23:  SCOPE
 4405  022372  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
 4406  022400  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
 4407  022404  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 4408  022412  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINTENANCE MODE
 4409  022420  012762  177676  000002          MOV     #-66.,RKWC(R2)  ;LOAD WORD COUNT
 4410  022426  012737  177677  003502          MOV     #-65.,E.WC
 4411  022434  012762  066762  000004          MOV     #NPRBUF,RKBA(R2);LOAD BUS ADDRESS
 4412  022442  012737  066764  003504          MOV     #NPRBUF+2,E.BA
 4413  022450  012737  010027  003500          MOV     #WRHEAD!CFMT,E.CS1 ;LOAD EXPECTED CS1
 4414  022456  012762  010027  000000          MOV     #WRHEAD!CFMT,RKCS1(R2) ;ISSUE COMMAND
 4415  022464  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS DATA
```

```
4416                                                                    ;  INDEX PULSE
4417  022470  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2)
4418  022476  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4419  022504  005300                          DEC     R0
4420  022506  001370                          BNE     5$
4421  022510  012700  000004                  MOV     #4,R0                   ;SIMULATE INDEX PULSE
4422  022514  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4423  022522  012762  000640  000026  6$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
4424  022530  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4425  022536  005300                          DEC     R0
4426  022540  001370                          BNE     6$
4427  022542  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4428  022550  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
4429  022556  012701  000102                  MOV     #66.,R1         ;ISSUE 66 NPR TRANSFERS
4430  022562  012700  000050          7$:     MOV     #40.,R0
4431  022566  012762  000440  000026  8$:     MOV     #DMD!MCLK,RKMR1(R2)
4432  022574  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4433  022602  005300                          DEC     R0
4434  022604  001370                          BNE     8$
4435  022606  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4436  022614  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG. 2
4437  022622  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
4438  022630  022737  066764  003504          CMP     #NPRBUF+2,E.BA  ;CHECK IF FIRST WORD
4439  022636  001004                          BNE     10$             ;NO, CONTINUE
4440  022640  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
4441  022644  005300                  9$:     DEC     R0
4442  022646  001376                          BNE     9$
4443  022650  016237  000010  003450  10$:     MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4444  022656  005737  003502                  TST     E.WC            ;CHECK IF LAST WORD
4445  022662  001003                          BNE     11$             ;NO, CONTINUE
4446  022664  012737  000200  003510          MOV     #OR,E.CS2       ;LOAD EXPECTED CS2
4447  022672  023737  003500  003440  11$:     CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4448  022700  001401                          BEQ     12$             ;YES, CONTINUE
4449  022702  104121                          ERROR   121             ;CS1 INCORRECT
4450  022704  023737  003510  003450  12$:     CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4451  022712  001401                          BEQ     13$             ;YES, CONTINUE
4452  022714  104122                          ERROR   122             ;CS2 INCORRECT
4453  022716  023737  003504  003444  13$:     CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
4454  022724  001401                          BEQ     14$             ;YES, CONTINUE
4455  022726  104123                          ERROR   123             ;BUS ADDRESS INCORRECT
4456  022730  023737  003502  003442  14$:     CMP     E.WC,T.WC       ;CHECK WORD COUNT REG CORRECT
4457  022736  001401                          BEQ     15$             ;YES, CONTINUE
4458  022740  104124                          ERROR   124             ;WORD COUNT INCORRECT
4459  022742  062737  000002  003504  15$:     ADD     #2,E.BA         ;INCREMENT BUS ADDRESS AND
4460  022750  005237  003502                  INC     E.WC            ;  WORD COUNT
4461  022754  005301                          DEC     R1              ;CHECK IF SILO FULL
4462  022756  001301                          BNE     7$              ;NO, GET NEXT WORD
4463  022760  012700  000120                  MOV     #2*40.,R0       ;ISSUE CLOCKS FOR TWO NPR'S
4464  022764  012762  000440  000026  20$:     MOV     #DMD!MCLK,RKMR1(R2)
4465  022772  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4466  023000  005300                          DEC     R0
4467  023002  001370                          BNE     20$
4468  023004  162737  000002  003504          SUB     #2,E.BA         ;ADJUST BUS ADDRESS AND WORD COUNT
4469  023012  005037  003502                  CLR     E.WC
4470  023016  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4471  023024  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
```

```
4472  023032  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
4473  023040  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
4474  023046  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4475  023054  001401                          BEQ     21$             ;YES, CONTINUE
4476  023056  104125                          ERROR   125             ;CS1 INCORRECT
4477  023060  023737  003510  003450  21$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4478  023066  001401                          BEQ     22$             ;YES, CONTINUE
4479  023070  104126                          ERROR   126             ;CS2 INCORRECT
4480  023072  023737  003504  003444  22$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS INCORRECT
4481  023100  001401                          BEQ     23$             ;YES, CONTINUE
4482  023102  104127                          ERROR   127             ;BUS ADDRESS INCORRECT
4483  023104  023737  003502  003442  23$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
4484  023112  001401                          BEQ     24$             ;YES, CONTINUE
4485  023114  104130                          ERROR   130             ;WORD COUNT INCORRECT
4486  023116  012703  066762          24$:    MOV     #NPRBUF,R3      ;LOAD BUFFER ADDRESS FOR COMPARE
4487  023122  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
4488  023130  012701  000102                  MOV     #66.,R1         ;LOAD COUNT
4489  023134  005037  003624                  CLR     WRDCNT          ;INITIALIZE WORD COUNT
4490  023140  016237  000024  003462  25$:    MOV     RKDB(R2),T.DB   ;GET DATA BUFFER
4491  023146  012337  003522                  MOV     (R3)+,E.DB      ;GET EXPECTED DATA
4492  023152  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
4493  023160  001401                          BEQ     26$             ;YES, CONTINUE
4494  023162  104131                          ERROR   131             ;DATA READ INCORRECT
4495  023164  012700  000050          26$:    MOV     #40.,R0         ;SET STALL
4496  023170  005300                  27$:    DEC     R0              ;RUN STALL TO ZERO
4497  023172  001376                          BNE     27$
4498  023174  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4499  023202  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4500  023210  022701  000001                  CMP     #1,R1           ;CHECK IF LAST WORD
4501  023214  001003                          BNE     28$             ;NO, CONTINUE
4502  023216  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
4503  023224  023737  003500  003440  28$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4504  023232  001401                          BEQ     29$             ;YES, CONTINUE
4505  023234  104132                          ERROR   132             ;CS1 INCORRECT
4506  023236  023737  003510  003450  29$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4507  023244  001401                          BEQ     30$             ;YES, CONTINUE
4508  023246  104133                          ERROR   133             ;CS2 INCORRECT
4509  023250  005237  003624          30$:    INC     WRDCNT          ;INCREMENT WORD COUNT
4510  023254  005301                          DEC     R1              ;CHECK IF FINISHED
4511  023256  001330                          BNE     25$             ;NO, CONTINUE
4512
4513                                  ;;*****************************************************************
4514                                  ;*TEST 24       BIT 16 AND 17 READING WITH NPR
4515                                  ;*
4516                                  ;*      CLEAR RK611 WITH CONTROLLER CLEAR.  PUT CONTROLLER
4517                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
4518                                  ;*      IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
4519                                  ;*      SPECIFY ONE WORD DATA TRANSFER FROM A LOCATION
4520                                  ;*      WITH BAD PARITY.  MAKE SURE A UNIBUS PARITY
4521                                  ;*      DOES NOT OCCUR.
4522                                  ;*
4523                                  ;*      NOTE:   THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
4524                                  ;*              ENABLE EXISTS FOR SPECIFIED LOCATION.
4525                                  ;*
4526                                  ;;*****************************************************************
4527  023260  000004                  TST24:  SCOPE
```

```
4528  023262  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
4529  023270  005737  003630                  TST     MEMPAR          ;CHECK IF MEMORY PARITY AVAILABLE
4530  023274  001017                          BNE     1$              ;YES, DO TEST
4531  023276  012737  000001  001200          MOV     #1,$TIMES       ;FORCE INTERATION COUNT TO 1
4532  023304  005227  177777                  INC     #-1             ;ONLY DO ONCE
4533  023310  001007                          BNE     64$             ;NO, GO TO NEXT TEST
4534  023312  104401  054734                  TYPE    ,TSTBY1         ;TYPE TEST N BYPASSED
4535  023316  013746  001220                  MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
4536  023322  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4537  023324  104401  054744                  TYPE    ,TSTBY2
4538  023330  000137  024104          64$:    JMP     TST25           ;GO TO NEXT TEST
4539
4540  023334  013702  001270          1$:     MOV     $BASE,R2        ;LOAD RK611 BASE
4541  023340  004737  044702                  JSR     PC,WRTPAR       ;GENERATE BAD PARITY
4542  023344  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
4543  023352  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINTENANCE MODE
4544  023360  012762  177777  000002          MOV     #-1,RKWC(R2)    ;LOAD WORD COUNT
4545  023366  005037  003502                  CLR     E.WC
4546  023372  012762  067220  000004          MOV     #BADPAR,RKBA(R2) ;LOAD BUS ADDRESS
4547  023400  012737  067222  003504          MOV     #BADPAR+2,E.BA
4548  023406  012737  010027  003500          MOV     #WRHEAD!CFMT,E.CS1 ;LOAD EXPECTED CS1
4549  023414  012762  010027  000000          MOV     #WRHEAD!CFMT,RKCS1(R2) ;ISSUE COMMAND
4550  023422  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
4551                                                                  ;  INDEX PULSE
4552  023426  012762  000440  000026  5$:     MOV     #DMD!MCLK,RKMR1(R2)
4553  023434  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4554  023442  005300                          DEC     R0
4555  023444  001370                          BNE     5$
4556  023446  012700  000004                  MOV     #4,R0           ;SIMULATE INDEX PULSE
4557  023452  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4558  023460  012762  000640  000026  6$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
4559  023466  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
4560  023474  005300                          DEC     R0
4561  023476  001370                          BNE     6$
4562  023500  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4563  023506  012737  000300  003510          MOV     #IR!OR,E.CS2    ;LOAD EXPECTED CS2
4564  023514  012700  000050                  MOV     #40.,R0         ;ISSUE NPR TRANSFER
4565  023520  012762  000440  000026  8$:     MOV     #DMD!MCLK,RKMR1(R2)
4566  023526  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4567  023534  005300                          DEC     R0
4568  023536  001370                          BNE     8$
4569  023540  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4570  023546  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS
4571  023554  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
4572  023562  012700  000024                  MOV     #20.,R0         ;WAIT FOR OUTPUT READY
4573  023566  005300                  9$:     DEC     R0
4574  023570  001376                          BNE     9$
4575  023572  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4576  023600  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4577  023606  001401                          BEQ     12$             ;YES, CHECK CS2
4578  023610  104134                          ERROR   134             ;CS1 INCORRECT
4579  023612  023737  003510  003450  12$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4580  023620  001401                          BEQ     13$             ;YES, CHECK BUS ADDRESS
4581  023622  104135                          ERROR   135             ;CS2 INCORRECT
4582  023624  023737  003504  003444  13$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
4583  023632  001401                          BEQ     14$             ;YES, CHECK WORD COUNT
```

```
4584  023634  104136                          ERROR   136             ;BUS ADDRESS INCORRECT
4585  023636  023737  003502  003442  14$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT CORRECT
4586  023644  001401                          BEQ     15$             ;YES, CONTINUE
4587  023646  104137                          ERROR   137             ;WORD COUNT INCORRECT
4588  023650  012700  000120          15$:    MOV     #2*40.,R0       ;ISSUE CLOCKS FOR TWO NPR'S
4589  023654  012762  000440  000026  20$:    MOV     #DMD!MCLK,RKMR1(R2)
4590  023662  012762  000040  000026          MOV     #DMD,RKMR1(R2)
4591  023670  005300                          DEC     R0
4592  023672  001370                          BNE     20$
4593  023674  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND STATUS REG. 1
4594  023702  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4595  023710  016237  000004  003444          MOV     RKBA(R2),T.BA   ;STORE BUS ADDRESS REG.
4596  023716  016237  000002  003442          MOV     RKWC(R2),T.WC   ;STORE WORD COUNT
4597  023724  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4598  023732  001401                          BEQ     21$             ;YES, CHECK CS2
4599  023734  104140                          ERROR   140             ;CS1 INCORRECT
4600  023736  023737  003510  003450  21$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4601  023744  001401                          BEQ     22$             ;YES, CHECK BUS ADDRESS
4602  023746  104141                          ERROR   141             ;CS2 INCORRECT
4603  023750  023737  003504  003444  22$:    CMP     E.BA,T.BA       ;CHECK BUS ADDRESS CORRECT
4604  023756  001401                          BEQ     23$             ;YES, CHECK WORD COUNT
4605  023760  104142                          ERROR   142             ;BUS ADDRESS INCORRECT
4606  023762  023737  003502  003442  23$:    CMP     E.WC,T.WC       ;CHECK WORD COUNT REG. CORRECT
4607  023770  001401                          BEQ     24$             ;YES, CHECK DATA
4608  023772  104143                          ERROR   143             ;WORD COUNT INCORRECT
4609  023774  016237  000024  003462  24$:    MOV     RKDB(R2),T.DB   ;READ DATA BUFFER
4610  024002  012737  000157  003522          MOV     #157,E.DB       ;LOAD EXPECT DATA
4611  024010  023737  003522  003462          CMP     E.DB,T.DB       ;CHECK TO MAKE SURE CORRECT
4612  024016  001401                          BEQ     26$             ;YES, CONTINUE
4613  024020  104144                          ERROR   144             ;DATA INCORRECT
4614  024022  016237  000000  003440  26$:    MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
4615  024030  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
4616  024036  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
4617  024044  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4618  024052  001401                          BEQ     29$             ;YES, CHECK SC2
4619  024054  104145                          ERROR   145             ;CS1 INCORRECT
4620  024056  023737  003510  003450  29$:    CMP     E.CS2,T.CS2     ;CHECK CS1 CORRECT
4621  024064  001401                          BEQ     30$             ;YES, CONTINUE
4622  024066  104146                          ERROR   146             ;CS2 INCORRECT
4623  024070  012737  000000  067220  30$:    MOV     #0,BADPAR       ;WRITE GOOD PARITY
4624  024076  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
4625
4626                                  .SBTTL  **MFM READ LOOPBACK TESTS
4627
4628                                  ;;*****************************************************************
4629                                  ;*TEST 25       READ LOOPBACK (PART 1)
4630                                  ;*
4631                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR, PUT CONTROLLER
4632                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
4633                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
4634                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
4635                                  ;*      SIMULATE SECTOR PULSE, 255 ZEROES, A
4636                                  ;*      ONE, AND A HEADER CONSISTING OF THE THREE
4637                                  ;*      FOLLOWING WORDS:
4638                                  ;*
4639                                  ;*              177777
```

```
                                       ;*              000000
                                       ;*              177777
                                       ;*
                                       ;*      MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
                                       ;*      IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
                                       ;*
                                       ;;*************************************************************
024104  000004                         TST25:  SCOPE
024106  012737  000144  001200                 MOV     #100.,$TIMES            ;;DO 100. ITERATIONS
024114  013702  001270                         MOV     $BASE,R2                ;LOAD RK611 BASE
024120  012762  100000  000000                 MOV     #CCLR,RKCS1(R2)         ;CLEAR RK611
024126  012762  000040  000026                 MOV     #DMD,RKMR1(R2)          ;PUT RK611 IN DIAGNOSTIC MODE
024134  012762  000025  000000                 MOV     #RDHEAD,RKCS1(R2)       ;ISSUE READ HEADER
024142  012700  000312                         MOV     #50.*4+2,R0             ;ISSUE ENOUGH CLOCKS UNTIL READY
024146  012762  000440  000026         1$:     MOV     #DMD!MCLK,RKMR1(R2)     ;  FOR SECTOR PULSE
024154  012762  000040  000026                 MOV     #DMD,RKMR1(R2)
024162  005300                                 DEC     R0
024164  001370                                 BNE     1$
024166  012762  000140  000026                 MOV     #DMD!MSP,RKMR1(R2)      ;SIMULATE SECTOR PULSE
024174  012762  000040  000026                 MOV     #DMD,RKMR1(R2)
024202  005037  003614                         CLR     PR.BIT                  ;INITIALIZE PRESENT BIT AND
024206  005037  003616                         CLR     M1.BIT                  ;  PREVIOUS BIT
024212  012700  000341                         MOV     #225.,R0
024216  004737  046106                 2$:     JSR     PC,RDBIT                ;SIMULATE SYNCH
024222  005300                                 DEC     R0
024224  001374                                 BNE     2$
024226  012737  000001  003614                 MOV     #1,PR.BIT
024234  004737  046106                         JSR     PC,RDBIT
024240  012701  000003                         MOV     #3,R1                   ;LOAD NUMBER OF WORDS
024244  012703  066702                         MOV     #HEAD1,R3               ;LOAD ADDRESS OF DATA
024250  012737  000025  003500                 MOV     #RDHEAD,E.CS1           ;LOAD EXPECTED CS1
024256  012304                         5$:     MOV     (R3)+,R4                ;GET DATA
024260  012700  000020                         MOV     #16.,R0                 ;LOAD BIT COUNT
024264  013737  003614  003616         6$:     MOV     PR.BIT,M1.BIT           ;STORE PREVIOUS BIT
024272  006004                                 ROR     R4                      ;GET NEXT BIT
024274  103403                                 BCS     7$                      ;CHECK IF 1
024276  005037  003614                         CLR     PR.BIT                  ;NO, ZERO
024302  000403                                 BR      8$                      ;SIMULATE READ DATA

024304  012737  000001  003614         7$:     MOV     #1,PR.BIT               ;ONE
024312  004737  046106                 8$:     JSR     PC,RDBIT                ;SIMULATE READ DATA
024316  016237  000000  003440                 MOV     RKCS1(R2),T.CS1         ;READ COMMAND AND STATUS REG. 1
024324  023737  003500  003440                 CMP     E.CS1,T.CS1             ;CHECK IF CS1 CORRECT
024332  001417                                 BEQ     9$                      ;YES, SIMULATE NEXT BIT
024334  012737  000003  003624                 MOV     #3,WRDCNT               ;LOAD WORD COUNT
024342  160137  003624                         SUB     R1,WRDCNT
024346  012737  000020  003622                 MOV     #16.,BITCNT             ;LOAD BIT COUNT
024354  160037  003622                         SUB     R0,BITCNT
024360  104150                                 ERROR   150                     ;CS1 INCORRECT DURING HEADER
024362  012762  100000  000000                 MOV     #CCLR,RKCS1(R2)         ;CLEAR RK611
024370  000522                                 BR      TST26                   ;;GO ON TO NEXT TEST

024372  005300                         9$:     DEC     R0                      ;CHECK IF READY FOR NEXT WORD
024374  001333                                 BNE     6$                      ;NO, GET NEXT BIT
024376  005301                                 DEC     R1                      ;CHECK IF HEADER FINISHED
024400  001326                                 BNE     5$                      ;NO, GET NEXT WORD
```

```
4696  024402  012700  000004                MOV     #4,R0           ;LOAD COUNT FOR POSTAMBLE
4697  024406  013737  003614  003616  15$:  MOV     PR.BIT,M1.BIT   ;STORE LAST BIT
4698  024414  005037  003614                CLR     PR.BIT          ;LOAD NEXT BIT
4699  024420  004737  046106                JSR     PC,RDBIT        ;SIMULATE 1 BIT READ
4700  024424  005300                        DEC     R0              ;CHECK IF TIME FOR READY
4701  024426  001367                        BNE     15$             ;NO, CONTINUE WITH POSTAMBLE
4702  024430  016237  000000  003440        MOV     RKCS1(R2),T.CS1 ;GET CURRENT CS1
4703  024436  016237  000010  003450        MOV     RKCS2(R2),T.CS2 ;GET CURRENT CS2
4704  024444  012737  000224  003500        MOV     #RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4705  024452  012737  000300  003510        MOV     #OR!IR,E.CS2    ;LOAD EXPECTED CS2
4706  024460  023737  003500  003440        CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4707  024466  001401                        BEQ     16$             ;YES, CHECK CS2
4708  024470  104151                        ERROR   151             ;CS1 INCORRECT
4709  024472  023737  003510  003450  16$:  CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4710  024500  001401                        BEQ     17$             ;YES, CHECK DATA
4711  024502  104152                        ERROR   152             ;CS2 INCORRECT
4712  024504  005037  003624          17$:  CLR     WRDCNT          ;INITIALIZE WORD COUNT
4713  024510  012703  066702                MOV     #HEAD1,R3       ;GET ADDRESS OF DATA
4714  024514  012337  003522          20$:  MOV     (R3)+,E.DB      ;GET EXPECTED DATA
4715  024520  016237  000024  003462        MOV     RKDB(R2),T.DB   ;GET ACTUAL DATA
4716  024526  016237  000000  003440        MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4717  024534  016237  000010  003450        MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4718  024542  022737  000002  003624        CMP     #2,WRDCNT       ;CHECK IF LAST WORD IN DATA BUFFER
4719  024550  001003                        BNE     21$             ;NO, CHECK CS1
4720  024552  012737  000100  003510        MOV     #IR,E.CS2       ;STORE EXPECTED CS2
4721  024560  023737  003500  003440  21$:  CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4722  024566  001402                        BEQ     22$             ;YES, CHECK CS2
4723  024570  104153                        ERROR   153             ;CS1 INCORRECT
4724  024572  000421                        BR      TST26           ;;GO ON TO NEXT TEST
4725
4726  024574  023737  003510  003450  22$:  CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4727  024602  001402                        BEQ     23$             ;YES, CHECK DATA
4728  024604  104154                        ERROR   154             ;CS2 INCORRECT
4729  024606  000413                        BR      TST26           ;;GO ON TO NEXT TEST
4730
4731  024610  023737  003522  003462  23$:  CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
4732  024616  001401                        BEQ     24$             ;YES, GET NEXT HEADER WORD
4733  024620  104155                        ERROR   155             ;DATA INCORRECT
4734  024622  005237  003624          24$:  INC     WRDCNT          ;INCREMENT WORD COUNT
4735  024626  022737  000003  003624        CMP     #3,WRDCNT       ;CHECK IF ALL THREE WORDS CHECK
4736  024634  001327                        BNE     20$             ;NO, GET NEXT WORD
4737
4738                          ;;*****************************************************************
4739                          ;*TEST 26       READ LOOPBACK (PART 2)
4740                          ;*
4741                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
4742                          ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
4743                          ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
4744                          ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
4745                          ;*      SIMULATE SECTOR PULSE, 255 ZEROES, A ONE,
4746                          ;*      AND A HEADER CONSISTING OF THE THREE
4747                          ;*      FOLLOWING WORDS:
4748                          ;*
4749                          ;*              000000
4750                          ;*              177777
4751                          ;*              000000
```

```
                                        ;*
                                        ;*      MAKE SURE THAT READY COMES UP AFTER THIRD WORD
                                        ;*      IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
                                        ;*
                                        ;;*****************************************************************
024636  000004                  TST26:  SCOPE
024640  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
024646  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
024652  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
024660  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
024666  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
024674  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL READY
024700  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)  ;  FOR SECTOR PULSE
024706  012762  000040  000026          MOV     #DMD,RKMR1(R2)
024714  005300                          DEC     R0
024716  001370                          BNE     1$
024720  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)  ;SIMULATE SECTOR PULSE
024726  012762  000040  000026          MOV     #DMD,RKMR1(R2)
024734  005037  003614                  CLR     PR.BIT          ;INITIALIZE PRESENT BIT AND
024740  005037  003616                  CLR     M1.BIT          ;  PREVIOUS BIT
024744  012700  000341                  MOV     #225.,R0
024750  004737  046106          2$:     JSR     PC,RDBIT        ;SIMULATE SYNCH
024754  005300                          DEC     R0
024756  001374                          BNE     2$
024760  012737  000001  003614          MOV     #1,PR.BIT
024766  004737  046106                  JSR     PC,RDBIT
024772  012701  000003                  MOV     #3,R1           ;LOAD NUMBER OF WORDS
024776  012703  066710                  MOV     #HEAD2,R3       ;LOAD ADDRESS OF DATA
025002  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
025010  012304                  5$:     MOV     (R3)+,R4        ;GET DATA
025012  012700  000020                  MOV     #16.,R0         ;LOAD BIT COUNT
025016  013737  003614  003616  6$:     MOV     PR.BIT,M1.BIT   ;STORE PREVIOUS BIT
025024  006004                          ROR     R4              ;GET NEXT BIT
025026  103403                          BCS     7$              ;CHECK IF 1
025030  005037  003614                  CLR     PR.BIT          ;NO, ZERO
025034  000403                          BR      8$              ;SIMULATE READ DATA

025036  012737  000001  003614  7$:     MOV     #1,PR.BIT       ;ONE
025044  004737  046106          8$:     JSR     PC,RDBIT        ;SIMULATE READ DATA
025050  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;READ COMMAND AND STATUS REG. 1
025056  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
025064  001417                          BEQ     9$              ;YES, SIMULATE NEXT BIT
025066  012737  000003  003624          MOV     #3,WRDCNT       ;LOAD WORD COUNT
025074  160137  003624                  SUB     R1,WRDCNT
025100  012737  000020  003622          MOV     #16.,BITCNT     ;LOAD BIT COUNT
025106  160037  003622                  SUB     R0,BITCNT
025112  104150                          ERROR   150             ;CS1 INCORRECT DURING HEADER
025114  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
025122  000522                          BR      TST27           ;;GO ON TO NEXT TEST

025124  005300                  9$:     DEC     R0              ;CHECK IF READY FOR NEXT WORD
025126  001333                          BNE     6$              ;NO, GET NEXT BIT
025130  005301                          DEC     R1              ;CHECK IF HEADER FINISHED
025132  001326                          BNE     5$              ;NO, GET NEXT WORD
025134  012700  000004                  MOV     #4,R0           ;LOAD COUNT FOR POSTAMBLE
025140  013737  003614  003616  15$:    MOV     PR.BIT,M1.BIT   ;STORE LAST BIT
```

```
4808  025146  005037  003614                  CLR     PR.BIT          ;LOAD NEXT BIT
4809  025152  004737  046106                  JSR     PC,RDBIT        ;SIMULATE 1 BIT READ
4810  025156  005300                          DEC     R0              ;CHECK IF TIME FOR READY
4811  025160  001367                          BNE     15$             ;NO, CONTINUE WITH POSTAMBLE
4812  025162  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET CURRENT CS1
4813  025170  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;GET CURRENT CS2
4814  025176  012737  000224  003500          MOV     #RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4815  025204  012737  000300  003510          MOV     #OR!IR,E.CS2    ;LOAD EXPECTED CS2
4816  025212  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4817  025220  001401                          BEQ     16$             ;YES, CHECK CS2
4818  025222  104151                          ERROR   151             ;CS1 INCORRECT
4819  025224  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4820  025232  001401                          BEQ     17$             ;YES, CHECK DATA
4821  025234  104152                          ERROR   152             ;CS2 INCORRECT
4822  025236  005037  003624          17$:    CLR     WRDCNT          ;INITIALIZE WORD COUNT
4823  025242  012703  066710                  MOV     #HEAD2,R3       ;GET ADDRESS OF DATA
4824  025246  012337  003522          20$:    MOV     (R3)+,E.DB      ;GET EXPECTED DATA
4825  025252  016237  000024  003462          MOV     RKDB(R2),T.DB   ;GET ACTUAL DATA
4826  025260  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4827  025266  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4828  025274  022737  000002  003624          CMP     #2,WRDCNT       ;CHECK IF LAST WORD IN DATA BUFFER
4829  025302  001003                          BNE     21$             ;NO, CHECK CS1
4830  025304  012737  000100  003510          MOV     #IR,E.CS2       ;STORE EXPECTED CS2
4831  025312  023737  003500  003440  21$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4832  025320  001402                          BEQ     22$             ;YES, CHECK CS2
4833  025322  104153                          ERROR   153             ;CS1 INCORRECT
4834  025324  000421                          BR      TST27           ;;GO ON TO NEXT TEST
4835
4836  025326  023737  003510  003450  22$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4837  025334  001402                          BEQ     23$             ;YES, CHECK DATA
4838  025336  104154                          ERROR   154             ;CS2 INCORRECT
4839  025340  000413                          BR      TST27           ;;GO ON TO NEXT TEST
4840
4841  025342  023737  003522  003462  23$:    CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
4842  025350  001401                          BEQ     24$             ;YES, GET NEXT HEADER WORD
4843  025352  104155                          ERROR   155             ;DATA INCORRECT
4844  025354  005237  003624          24$:    INC     WRDCNT          ;INCREMENT WORD COUNT
4845  025360  022737  000003  003624          CMP     #3,WRDCNT       ;CHECK IF ALL THREE WORDS CHECK
4846  025366  001327                          BNE     20$             ;NO, GET NEXT WORD
4847
4848                                  ;;*****************************************************************
4849                                  ;*TEST 27       READ LOOPBACK (PART 3)
4850                                  ;*
4851                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
4852                                  ;*      IN DIAGNOSTIC MOVE.  ISSUE A READ HEADER TO AN RK06
4853                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
4854                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES
4855                                  ;*      SIMULATE SECTOR PULSE, 255 ZEROES, A
4856                                  ;*      ONE, AND A HEADER CONSISTING OF THE THREE
4857                                  ;*      FOLLOWING WORDS:
4858                                  ;*
4859                                  ;*              125252
4860                                  ;*              052525
4861                                  ;*              125252
4862                                  ;*
4863                                  ;*      MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
```

```
                                        ;*      IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
                                        ;*
                                        ;;*********************************************************************
025370  000004                  TST27:  SCOPE
025372  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
025400  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
025404  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
025412  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
025420  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
025426  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL READY
025432  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)  ;  FOR SECTOR PULSE
025440  012762  000040  000026          MOV     #DMD,RKMR1(R2)
025446  005300                          DEC     R0
025450  001370                          BNE     1$
025452  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)  ;SIMULATE SECTOR PULSE
025460  012762  000040  000026          MOV     #DMD,RKMR1(R2)
025466  005037  003614                  CLR     PR.BIT          ;INITIALIZE PRESENT BIT AND
025472  005037  003616                  CLR     M1.BIT          ;  PREVIOUS BIT
025476  012700  000341                  MOV     #225.,R0
025502  004737  046106          2$:     JSR     PC,RDBIT        ;SIMULATE SYNCH
025506  005300                          DEC     R0
025510  001374                          BNE     2$
025512  012737  000001  003614          MOV     #1,PR.BIT
025520  004737  046106                  JSR     PC,RDBIT
025524  012701  000003                  MOV     #3,R1           ;LOAD NUMBER OF WORDS
025530  012703  066716                  MOV     #HEAD3,R3       ;LOAD ADDRESS OF DATA
025534  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
025542  012304                  5$:     MOV     (R3)+,R4        ;GET DATA
025544  012700  000020                  MOV     #16.,R0         ;LOAD BIT COUNT
025550  013737  003614  003616  6$:     MOV     PR.BIT,M1.BIT   ;STORE PREVIOUS BIT
025556  006004                          ROR     R4              ;GET NEXT BIT
025560  103403                          BCS     7$              ;CHECK IF 1
025562  005037  003614                  CLR     PR.BIT          ;NO, ZERO
025566  000403                          BR      8$              ;SIMULATE READ DATA

025570  012737  000001  003614  7$:     MOV     #1,PR.BIT       ;ONE
025576  004737  046106          8$:     JSR     PC,RDBIT        ;SIMULATE READ DATA
025602  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;READ COMMAND AND STATUS REG. 1
025610  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
025616  001417                          BEQ     9$              ;YES, SIMULATE NEXT BIT
025620  012737  000003  003624          MOV     #3,WRDCNT       ;LOAD WORD COUNT
025626  160137  003624                  SUB     R1,WRDCNT
025632  012737  000020  003622          MOV     #16.,BITCNT     ;LOAD BIT COUNT
025640  160037  003622                  SUB     R0,BITCNT
025644  104150                          ERROR   150             ;CS1 INCORRECT DURING HEADER
025646  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
025654  000522                          BR      TST30           ;;GO ON TO NEXT TEST

025656  005300                  9$:     DEC     R0              ;CHECK IF READY FOR NEXT WORD
025660  001333                          BNE     6$              ;NO, GET NEXT BIT
025662  005301                          DEC     R1              ;CHECK IF HEADER FINISHED
025664  001326                          BNE     5$              ;NO, GET NEXT WORD
025666  012700  000004                  MOV     #4,R0           ;LOAD COUNT FOR POSTAMBLE
025672  013737  003614  003616  15$:    MOV     PR.BIT,M1.BIT   ;STORE LAST BIT
025700  005037  003614                  CLR     PR.BIT          ;LOAD NEXT BIT
025704  004737  046106                  JSR     PC,RDBIT        ;SIMULATE 1 BIT READ
```

```
4920  025710  005300                          DEC     R0              ;CHECK IF TIME FOR READY
4921  025712  001367                          BNE     15$             ;NO, CONTINUE WITH POSTAMBLE
4922  025714  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET CURRENT CS1
4923  025722  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;GET CURRENT CS2
4924  025730  012737  000224  003500          MOV     #RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
4925  025736  012737  000300  003510          MOV     #OR!IR,E.CS2    ;LOAD EXPECTED CS2
4926  025744  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4927  025752  001401                          BEQ     16$             ;YES, CHECK CS2
4928  025754  104151                          ERROR   151             ;CS1 INCORRECT
4929  025756  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4930  025764  001401                          BEQ     17$             ;YES, CHECK DATA
4931  025766  104152                          ERROR   152             ;CS2 INCORRECT
4932  025770  005037  003624          17$:    CLR     WRDCNT          ;INITIALIZE WORD COUNT
4933  025774  012703  066716                  MOV     #HEAD3,R3       ;GET ADDRESS OF DATA
4934  026000  012337  003522          20$:    MOV     (R3)+,E.DB      ;GET EXPECTED DATA
4935  026004  016237  000024  003462          MOV     RKDB(R2),T.DB   ;GET ACTUAL DATA
4936  026012  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
4937  026020  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
4938  026026  022737  000002  003624          CMP     #2,WRDCNT       ;CHECK IF LAST WORD IN DATA BUFFER
4939  026034  001003                          BNE     21$             ;NO, CHECK CS1
4940  026036  012737  000100  003510          MOV     #IR,E.CS2       ;STORE EXPECTED CS2
4941  026044  023737  003500  003440  21$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
4942  026052  001402                          BEQ     22$             ;YES, CHECK CS2
4943  026054  104153                          ERROR   153             ;CS1 INCORRECT
4944  026056  000421                          BR      TST30           ;;GO ON TO NEXT TEST
4945
4946  026060  023737  003510  003450  22$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
4947  026066  001402                          BEQ     23$             ;YES, CHECK DATA
4948  026070  104154                          ERROR   154             ;CS2 INCORRECT
4949  026072  000413                          BR      TST30           ;;GO ON TO NEXT TEST
4950
4951  026074  023737  003522  003462  23$:    CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
4952  026102  001401                          BEQ     24$             ;YES, GET NEXT HEADER WORD
4953  026104  104155                          ERROR   155             ;DATA INCORRECT
4954  026106  005237  003624          24$:    INC     WRDCNT          ;INCREMENT WORD COUNT
4955  026112  022737  000003  003624          CMP     #3,WRDCNT       ;CHECK IF ALL THREE WORDS CHECK
4956  026120  001327                          BNE     20$             ;NO, GET NEXT WORD
4957
4958                          ;;*****************************************************************
4959                          ;*TEST 30       READ LOOPBACK (PART 4)
4960                          ;*
4961                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
4962                          ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
4963                          ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
4964                          ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
4965                          ;*      SIMULATE SECTOR PULSE, 225 ZEROES, A
4966                          ;*      ONE, AND A HEADER CONSISTING OF THE THREE
4967                          ;*      FOLLOWING WORDS:
4968                          ;*
4969                          ;*              044444
4970                          ;*              022222
4971                          ;*              111111
4972                          ;*
4973                          ;*      MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
4974                          ;*      IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
4975                          ;*
```

```
4976                                    ;;*****************************************************************
4977  026122  000004                    TST30:  SCOPE
4978  026124  012737  000144  001200            MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
4979  026132  013702  001270                    MOV     $BASE,R2        ;LOAD RK611 BASE
4980  026136  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
4981  026144  012762  000040  000026            MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
4982  026152  012762  000025  000000            MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
4983  026160  012700  000312                    MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL READY
4984  026164  012762  000440  000026  1$:       MOV     #DMD!MCLK,RKMR1(R2)  ;  FOR SECTOR PULSE
4985  026172  012762  000040  000026            MOV     #DMD,RKMR1(R2)
4986  026200  005300                            DEC     R0
4987  026202  001370                            BNE     1$
4988  026204  012762  000140  000026            MOV     #DMD!MSP,RKMR1(R2)  ;SIMULATE SECTOR PULSE
4989  026212  012762  000040  000026            MOV     #DMD,RKMR1(R2)
4990  026220  005037  003614                    CLR     PR.BIT          ;INITIALIZE PRESENT BIT AND
4991  026224  005037  003616                    CLR     M1.BIT          ;  PREVIOUS BIT
4992  026230  012700  000341                    MOV     #225.,R0
4993  026234  004737  046106          2$:       JSR     PC,RDBIT        ;SIMULATE SYNCH
4994  026240  005300                            DEC     R0
4995  026242  001374                            BNE     2$
4996  026244  012737  000001  003614            MOV     #1,PR.BIT
4997  026252  004737  046106                    JSR     PC,RDBIT
4998  026256  012701  000003                    MOV     #3,R1           ;LOAD NUMBER OF WORDS
4999  026262  012703  066732                    MOV     #HEAD4,R3       ;LOAD ADDRESS OF DATA
5000  026266  012737  000025  003500            MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
5001  026274  012304                  5$:       MOV     (R3)+,R4        ;GET DATA
5002  026276  012700  000020                    MOV     #16.,R0         ;LOAD BIT COUNT
5003  026302  013737  003614  003616  6$:       MOV     PR.BIT,M1.BIT   ;STORE PREVIOUS BIT
5004  026310  006004                            ROR     R4              ;GET NEXT BIT
5005  026312  103403                            BCS     7$              ;CHECK IF 1
5006  026314  005037  003614                    CLR     PR.BIT          ;NO, ZERO
5007  026320  000403                            BR      8$              ;SIMULATE READ DATA
5008
5009  026322  012737  000001  003614  7$:       MOV     #1,PR.BIT       ;ONE
5010  026330  004737  046106          8$:       JSR     PC,RDBIT        ;SIMULATE READ DATA
5011  026334  016237  000000  003440            MOV     RKCS1(R2),T.CS1 ;READ COMMAND AND STATUS REG. 1
5012  026342  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
5013  026350  001417                            BEQ     9$              ;YES, SIMULATE NEXT BIT
5014  026352  012737  000003  003624            MOV     #3,WRDCNT       ;LOAD WORD COUNT
5015  026360  160137  003624                    SUB     R1,WRDCNT
5016  026364  012737  000020  003622            MOV     #16.,BITCNT     ;LOAD BIT COUNT
5017  026372  160037  003622                    SUB     R0,BITCNT
5018  026376  104150                            ERROR   150             ;CS1 INCORRECT DURING HEADER
5019  026400  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5020  026406  000522                            BR      TST31           ;;GO ON TO NEXT TEST
5021
5022  026410  005300                  9$:       DEC     R0              ;CHECK IF READY FOR NEXT WORD
5023  026412  001333                            BNE     6$              ;NO, GET NEXT BIT
5024  026414  005301                            DEC     R1              ;CHECK IF HEADER FINISHED
5025  026416  001326                            BNE     5$              ;NO, GET NEXT WORD
5026  026420  012700  000004                    MOV     #4,R0           ;LOAD COUNT FOR POSTAMBLE
5027  026424  013737  003614  003616  15$:      MOV     PR.BIT,M1.BIT   ;STORE LAST BIT
5028  026432  005037  003614                    CLR     PR.BIT          ;LOAD NEXT BIT
5029  026436  004737  046106                    JSR     PC,RDBIT        ;SIMULATE 1 BIT READ
5030  026442  005300                            DEC     R0              ;CHECK IF TIME FOR READY
5031  026444  001367                            BNE     15$             ;NO, CONTINUE WITH POSTAMBLE
```

```
5032  026446  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET CURRENT CS1
5033  026454  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;GET CURRENT CS2
5034  026462  012737  000224  003500          MOV     #RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5035  026470  012737  000300  003510          MOV     #OR!IR,E.CS2    ;LOAD EXPECTED CS2
5036  026476  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
5037  026504  001401                          BEQ     16$             ;YES, CHECK CS2
5038  026506  104151                          ERROR   151             ;CS1 INCORRECT
5039  026510  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5040  026516  001401                          BEQ     17$             ;YES, CHECK DATA
5041  026520  104152                          ERROR   152             ;CS2 INCORRECT
5042  026522  005037  003624          17$:    CLR     WRDCNT          ;INITIALIZE WORD COUNT
5043  026526  012703  066732                  MOV     #HEAD4,R3       ;GET ADDRESS OF DATA
5044  026532  012337  003522          20$:    MOV     (R3)+,E.DB      ;GET EXPECTED DATA
5045  026536  016237  000024  003462          MOV     RKDB(R2),T.DB   ;GET ACTUAL DATA
5046  026544  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
5047  026552  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
5048  026560  022737  000002  003624          CMP     #2,WRDCNT       ;CHECK IF LAST WORD IN DATA BUFFER
5049  026566  001003                          BNE     21$             ;NO, CHECK CS1
5050  026570  012737  000100  003510          MOV     #IR,E.CS2       ;STORE EXPECTED CS2
5051  026576  023737  003500  003440  21$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
5052  026604  001402                          BEQ     22$             ;YES, CHECK CS2
5053  026606  104153                          ERROR   153             ;CS1 INCORRECT
5054  026610  000421                          BR      TST31           ;;GO ON TO NEXT TEST
5055
5056  026612  023737  003510  003450  22$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5057  026620  001402                          BEQ     23$             ;YES, CHECK DATA
5058  026622  104154                          ERROR   154             ;CS2 INCORRECT
5059  026624  000413                          BR      TST31           ;;GO ON TO NEXT TEST
5060
5061  026626  023737  003522  003462  23$:    CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
5062  026634  001401                          BEQ     24$             ;YES, GET NEXT HEADER WORD
5063  026636  104155                          ERROR   155             ;DATA INCORRECT
5064  026640  005237  003624          24$:    INC     WRDCNT          ;INCREMENT WORD COUNT
5065  026644  022737  000003  003624          CMP     #3,WRDCNT       ;CHECK IF ALL THREE WORDS CHECK
5066  026652  001327                          BNE     20$             ;NO, GET NEXT WORD
5067
5068                                  ;;*****************************************************************
5069                                  ;*TEST 31       READ LOOPBACK (PART 5)
5070                                  ;*
5071                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
5072                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
5073                                  ;*      26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
5074                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
5075                                  ;*      SIMULATE SECTOR PULSE, 255 ZEROES, A
5076                                  ;*      ONE, AND A HEADER CONSISTING OF THE THREE
5077                                  ;*      FOLLOWING WORDS.
5078                                  ;*
5079                                  ;*              052012
5080                                  ;*              100520
5081                                  ;*              052012
5082                                  ;*
5083                                  ;*      MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
5084                                  ;*      IS TRANSRERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
5085                                  ;*
5086                                  ;*****************************************************************
5087  026654  000004                  TST31:  SCOPE
```

```
5088  026656  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
5089  026664  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
5090  026670  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5091  026676  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
5092  026704  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
5093  026712  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL READY
5094  026716  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)  ;  FOR SECTOR PULSE
5095  026724  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5096  026732  005300                          DEC     R0
5097  026734  001370                          BNE     1$
5098  026736  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)  ;SIMULATE SECTOR PULSE
5099  026744  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5100  026752  005037  003614                  CLR     PR.BIT          ;INITIALIZE PRESENT BIT AND
5101  026756  005037  003616                  CLR     M1.BIT          ;  PREVIOUS BIT
5102  026762  012700  000341                  MOV     #225.,R0
5103  026766  004737  046106          2$:     JSR     PC,RDBIT        ;SIMULATE SYNCH
5104  026772  005300                          DEC     R0
5105  026774  001374                          BNE     2$
5106  026776  012737  000001  003614          MOV     #1,PR.BIT
5107  027004  004737  046106                  JSR     PC,RDBIT
5108  027010  012701  000003                  MOV     #3,R1           ;LOAD NUMBER OF WORDS
5109  027014  012703  066740                  MOV     #HEAD5,R3       ;LOAD ADDRESS OF DATA
5110  027020  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
5111  027026  012304          5$:     MOV     (R3)+,R4        ;GET DATA
5112  027030  012700  000020                  MOV     #16.,R0         ;LOAD BIT COUNT
5113  027034  013737  003614  003616  6$:     MOV     PR.BIT,M1.BIT   ;STORE PREVIOUS BIT
5114  027042  006004                          ROR     R4              ;GET NEXT BIT
5115  027044  103403                          BCS     7$              ;CHECK IF 1
5116  027046  005037  003614                  CLR     PR.BIT          ;NO, ZERO
5117  027052  000403                          BR      8$              ;SIMULATE READ DATA
5118
5119  027054  012737  000001  003614  7$:     MOV     #1,PR.BIT       ;ONE
5120  027062  004737  046106          8$:     JSR     PC,RDBIT        ;SIMULATE READ DATA
5121  027066  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;READ COMMAND AND STATUS REG. 1
5122  027074  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
5123  027102  001417                          BEQ     9$              ;YES, SIMULATE NEXT BIT
5124  027104  012737  000003  003624          MOV     #3,WRDCNT       ;LOAD WORD COUNT
5125  027112  160137  003624                  SUB     R1,WRDCNT
5126  027116  012737  000020  003622          MOV     #16.,BITCNT     ;LOAD BIT COUNT
5127  027124  160037  003622                  SUB     R0,BITCNT
5128  027130  104150                          ERROR   150             ;CS1 INCORRECT DURING HEADER
5129  027132  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5130  027140  000522                          BR      TST32           ;;GO ON TO NEXT TEST
5131
5132  027142  005300          9$:     DEC     R0              ;CHECK IF READY FOR NEXT WORD
5133  027144  001333                          BNE     6$              ;NO, GET NEXT BIT
5134  027146  005301                          DEC     R1              ;CHECK IF HEADER FINISHED
5135  027150  001326                          BNE     5$              ;NO, GET NEXT WORD
5136  027152  012700  000004                  MOV     #4,R0           ;LOAD COUNT FOR POSTAMBLE
5137  027156  013737  003614  003616  15$:     MOV     PR.BIT,M1.BIT   ;STORE LAST BIT
5138  027164  005037  003614                  CLR     PR.BIT          ;LOAD NEXT BIT
5139  027170  004737  046106                  JSR     PC,RDBIT        ;SIMULATE 1 BIT READ
5140  027174  005300                          DEC     R0              ;CHECK IF TIME FOR READY
5141  027176  001367                          BNE     15$             ;NO, CONTINUE WITH POSTAMBLE
5142  027200  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET CURRENT CS1
5143  027206  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;GET CURRENT CS2
```

```
5144  027214  012737  000224  003500          MOV     #RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5145  027222  012737  000300  003510          MOV     #OR!IR,E.CS2    ;LOAD EXPECTED CS2
5146  027230  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
5147  027236  001401                          BEQ     16$             ;YES, CHECK CS2
5148  027240  104151                          ERROR   151             ;CS1 INCORRECT
5149  027242  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5150  027250  001401                          BEQ     17$             ;YES, CHECK DATA
5151  027252  104152                          ERROR   152             ;CS2 INCORRECT
5152  027254  005037  003624          17$:    CLR     WRDCNT          ;INITIALIZE WORD COUNT
5153  027260  012703  066740                  MOV     #HEAD5,R3       ;GET ADDRESS OF DATA
5154  027264  012337  003522          20$:    MOV     (R3)+,E.DB      ;GET EXPECTED DATA
5155  027270  016237  000024  003462          MOV     RKDB(R2),T.DB   ;GET ACTUAL DATA
5156  027276  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
5157  027304  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
5158  027312  022737  000002  003624          CMP     #2,WRDCNT       ;CHECK IF LAST WORD IN DATA BUFFER
5159  027320  001003                          BNE     21$             ;NO, CHECK CS1
5160  027322  012737  000100  003510          MOV     #IR,E.CS2       ;STORE EXPECTED CS2
5161  027330  023737  003500  003440  21$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
5162  027336  001402                          BEQ     22$             ;YES, CHECK CS2
5163  027340  104153                          ERROR   153             ;CS1 INCORRECT
5164  027342  000421                          BR      TST32           ;;GO ON TO NEXT TEST
5165
5166  027344  023737  003510  003450  22$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5167  027352  001402                          BEQ     23$             ;YES, CHECK DATA
5168  027354  104154                          ERROR   154             ;CS2 INCORRECT
5169  027356  000413                          BR      TST32           ;;GO ON TO NEXT TEST
5170
5171  027360  023737  003522  003462  23$:    CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
5172  027366  001401                          BEQ     24$             ;YES, GET NEXT HEADER WORD
5173  027370  104155                          ERROR   155             ;DATA INCORRECT
5174  027372  005237  003624          24$:    INC     WRDCNT          ;INCREMENT WORD COUNT
5175  027376  022737  000003  003624          CMP     #3,WRDCNT       ;CHECK IF ALL THREE WORDS CHECK
5176  027404  001327                          BNE     20$             ;NO, GET NEXT WORD
5177
5178                                  ;;*****************************************************************
5179                                  ;*TEST 32       READ HEADER IN 18 BIT MODE
5180                                  ;*
5181                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
5182                                  ;*      IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
5183                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
5184                                  ;*      SIMULATE SECTOR PULSE, 255 ZEROES, A
5185                                  ;*      ONE, AND A HEADER CONSISTING OF THE THREE
5186                                  ;*      FOLLOWING WORDS:
5187                                  ;*
5188                                  ;*              177777
5189                                  ;*              000000
5190                                  ;*              177777
5191                                  ;*
5192                                  ;*      MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
5193                                  ;*      IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
5194                                  ;*
5195                                  ;;*****************************************************************
5196  027406  000004                  TST32:  SCOPE
5197  027410  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
5198  027416  013702  001270                  MOV     $BASE,R2        ;LOAD RK611
5199  027422  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
```

```
5200  027430  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
5201  027436  012762  010025  000000          MOV     #CFMT!RDHEAD,RKCS1(R2) ;ISSUE READ HEADER (24 SECTOR FORMAT)
5202  027444  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL READY
5203  027450  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2) ;  FOR SECTOR PULSE
5204  027456  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5205  027464  005300                          DEC     R0
5206  027466  001370                          BNE     1$
5207  027470  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
5208  027476  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5209  027504  005037  003614                  CLR     PR.BIT          ;INITIALIZE PRESENT BIT AND
5210  027510  005037  003616                  CLR     M1.BIT          ;  PREVIOUS BIT
5211  027514  012700  000377                  MOV     #255.,R0
5212  027520  004737  046106          2$:     JSR     PC,RDBIT        ;SIMULATE SYNCH
5213  027524  005300                          DEC     R0
5214  027526  001374                          BNE     2$
5215  027530  012737  000001  003614          MOV     #1,PR.BIT
5216  027536  004737  046106                  JSR     PC,RDBIT
5217  027542  012701  000003                  MOV     #3,R1           ;LOAD NUMBER OF WORDS
5218  027546  012703  066702                  MOV     #HEAD1,R3       ;LOAD ADDRESS OF DATA
5219  027552  012737  010025  003500          MOV     #CFMT!RDHEAD,E.CS1 ;LOAD EXPECTED CS1
5220  027560  012304          5$:     MOV     (R3)+,R4        ;GET DATA
5221  027562  012700  000020                  MOV     #16.,R0         ;LOAD BIT COUNT
5222  027566  013737  003614  003616  6$:     MOV     PR.BIT,M1.BIT   ;STORE PRESENT BIT
5223  027574  006004                          ROR     R4              ;GET NEXT BIT
5224  027576  103403                          BCS     7$              ;CHECK IF 1
5225  027600  005037  003614                  CLR     PR.BIT          ;NO, ZERO
5226  027604  000403                          BR      8$              ;SIMULATE READ DATA
5227
5228  027606  012737  000001  003614  7$:     MOV     #1,PR.BIT       ;ONE
5229  027614  004737  046106          8$:     JSR     PC,RDBIT        ;SIMULATE READ DATA
5230  027620  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;READ COMMAND AND STATUS REG. 1
5231  027626  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
5232  027634  001417                          BEQ     9$              ;YES, SIMULATE NEXT BIT
5233  027636  012737  000003  003624          MOV     #3,WRDCNT       ;LOAD WORD COUNT
5234  027644  160137  003624                  SUB     R1,WRDCNT
5235  027650  012737  000020  003622          MOV     #16.,BITCNT     ;LOAD BIT COUNT
5236  027656  160037  003622                  SUB     R0,BITCNT
5237  027662  104156                          ERROR   156             ;CS1 INCORRECT DURING HEADER
5238  027664  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5239  027672  000522                          BR      TST33           ;;GO ON TO NEXT TEST
5240
5241  027674  005300          9$:     DEC     R0              ;CHECK IF READY FOR NEXT WORD
5242  027676  001333                          BNE     6$              ;NO, GET NEXT BIT
5243  027700  005301                          DEC     R1              ;CHECK IF HEADER FINISHED
5244  027702  001326                          BNE     5$              ;NO, GET NEXT WORD
5245  027704  012700  000004                  MOV     #4,R0           ;LOAD COUNT FOR POSTAMBLE
5246  027710  013737  003614  003616  15$:    MOV     PR.BIT,M1.BIT   ;STORE LAST BIT
5247  027716  005037  003614                  CLR     PR.BIT          ;LOAD NEXT BIT
5248  027722  004737  046106                  JSR     PC,RDBIT        ;READ BIT
5249  027726  005300                          DEC     R0              ;CHECK IF TIME FOR READY
5250  027730  001367                          BNE     15$             ;NO, CONTINUE WITH POSTAMBLE
5251  027732  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET CURRENT CS1
5252  027740  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;GET CURRENT CS2
5253  027746  012737  010224  003500          MOV     #CFMT!RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5254  027754  012737  000300  003510          MOV     #OR!IR,E.CS2    ;LOAD EXPECTED CS2
5255  027762  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
```

```
5256  027770  001401                          BEQ     16$             ;YES, CHECK CS2
5257  027772  104157                          ERROR   157             ;CS1 INCORRECT
5258  027774  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5259  030002  001401                          BEQ     17$             ;YES, CHECK DATA
5260  030004  104160                          ERROR   160             ;CS2 INCORRECT
5261  030006  005037  003624          17$:    CLR     WRDCNT          ;INITIALIZE WORD COUNT
5262  030012  012703  066702                  MOV     #HEAD1,R3       ;GET ADDRESS OF DATA
5263  030016  012337  003522          20$:    MOV     (R3)+,E.DB      ;GET EXPECTED DATA
5264  030022  016237  000024  003462          MOV     RKDB(R2),T.DB   ;GET ACTUAL DATA
5265  030030  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
5266  030036  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
5267  030044  022737  000002  003624          CMP     #2,WRDCNT       ;CHECK IF LAST WORD IN DATA BUFFER
5268  030052  001003                          BNE     21$             ;NO, CHECK CS1
5269  030054  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
5270  030062  023737  003500  003440  21$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
5271  030070  001402                          BEQ     22$             ;YES, CHECK CS2
5272  030072  104161                          ERROR   161             ;CS1 INCORRECT
5273  030074  000421                          BR      TST33           ;;GO ON TO NEXT TEST
5274
5275  030076  023737  003510  003450  22$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5276  030104  001402                          BEQ     23$             ;YES, CHECK DATA BUFFER
5277  030106  104162                          ERROR   162             ;CS2 INCORRECT
5278  030110  000413                          BR      TST33           ;;GO ON TO NEXT TEST
5279
5280  030112  023737  003522  003462  23$:    CMP     E.DB,T.DB       ;CHECK DATA BUFFER CORRECT
5281  030120  001401                          BEQ     24$             ;YES, GET NEXT WORD
5282  030122  104163                          ERROR   163             ;DATA BUFFER INCORRECT
5283  030124  005237  003624          24$:    INC     WRDCNT          ;INCREMENT WORD COUNT
5284  030130  022737  000003  003624          CMP     #3,WRDCNT       ;CHECK IF FINISHED
5285  030136  001327                          BNE     20$             ;NO READ NEXT WORD
5286
5287                                  ;;*****************************************************************
5288                                  ;*TEST 33       SYNCH DETECT IN READ HEADER
5289                                  ;*
5290                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
5291                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN RK06
5292                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
5293                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.
5294                                  ;*      SIMULATE SECTOR PULSE AND 350 ZEROES.  MAKE
5295                                  ;*      SURE READY REMAINS RESET AND THE SILO REMAINS
5296                                  ;*      EMPTY.
5297                                  ;*
5298                                  ;;*****************************************************************
5299  030140  000004                  TST33:  SCOPE
5300  030142  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
5301  030150  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
5302  030154  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5303  030162  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINT MODE
5304  030170  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2) ;ISSUE READ HEAD
5305  030176  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL READY
5306  030202  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2) ;  FOR SECTOR PULSE
5307  030210  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5308  030216  005300                          DEC     R0
5309  030220  001370                          BNE     1$
5310  030222  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE TO SECTOR PULSE
5311  030230  012762  000040  000026          MOV     #DMD,RKMR1(R2)
```

```
5312  030236  005037  003614                  CLR     PR.BIT          ;INITIALIZE PRESENT AND
5313  030242  005037  003616                  CLR     M1.BIT          ;  PREVIOUS BIT
5314  030246  012737  000025  003500          MOV     #RDHEAD,E.CS1   ;LOAD EXPECTED CS1
5315  030254  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
5316  030262  005037  003622                  CLR     BITCNT          ;SIMULATE 350 ZEROES
5317  030266  004737  046106          2$:     JSR     PC,RDBIT
5318  030272  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE CS1
5319  030300  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE CS2
5320  030306  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
5321  030314  001402                          BEQ     3$              ;YES, CHECK CS2
5322  030316  104164                          ERROR   164                     ;CS1 INCORRECT
5323  030320  000447                          BR      TST34           ;;GO ON TO NEXT TEST
5324
5325  030322  023737  003510  003450  3$:     CMP     E.CS2,T.CS2     ;CHECK IF CS2 CORRECT
5326  030330  001402                          BEQ     4$              ;YES, CHECK IF SILO EMPTY
5327  030332  104165                          ERROR   165             ;CS2 INCORRECT
5328  030334  000441                          BR      TST34           ;;GO ON TO NEXT TEST
5329  030336  005237  003622          4$:     INC     BITCNT          ;INCREMENT BIT COUNT
5330  030342  022737  000536  003622          CMP     #350.,BITCNT    ;CHECK IF FINISHED
5331  030350  001346                          BNE     2$              ;NO, SIMULATE NEXT ZERO
5332  030352  005762  000024                  TST     RKDB(R2)        ;READ DATA BUFFER
5333  030356  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE CS1 AND CS2
5334  030364  016237  000010  003450          MOV     RKCS2(R2),T.CS2
5335  030372  012737  100224  003500          MOV     #CERR!RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECT CS1
5336  030400  012737  100100  003510          MOV     #DCK!IR,E.CS2   ;LOAD EXPECTED CS2
5337  030406  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK FOR CONTROLLER ERROR
5338  030414  001401                          BEQ     5$              ;YES, CHECK FOR DATA LATE
5339  030416  104166                          ERROR   166             ;CS1 INCORRECT
5340  030420  023737  003510  003450  5$:     CMP     E.CS2,T.CS2     ;CHECK FOR DATA LATE
5341  030426  001401                          BEQ     6$              ;YES, CHECK RK611
5342  030430  104167                          ERROR   167             ;CS2 INCORRECT
5343  030432  012762  100000  000000  6$:     MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5344
5345                                  ;;*****************************************************************
5346                                  ;*TEST 34       ZERO SYNCH ON READ
5347                                  ;*
5348                                  ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
5349                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
5350                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES,
5351                                  ;*      SIMULATE SECTOR PULSE, 255 ZEROES SHIFTED BY A HALF
5352                                  ;*      BIT TIME, A ONE, AND A HEADER CONSISTING OF THE
5353                                  ;*      THREE FOLLOWING WORDS:
5354                                  ;*
5355                                  ;*              !77777
5356                                  ;*              000000
5357                                  ;*              177777
5358                                  ;*
5359                                  ;*      MAKE SURE THAT READY COMES AFTER THE THIRD WORD
5360                                  ;*      IS TRANSFERRED.  CHECK THE SILO FOR CORRECT CONTENTS.
5361                                  ;*
5362                                  ;;*****************************************************************
5363  030440  000004          TST34:  SCOPE
5364  030442  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
5365  030450  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
5366  030454  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5367  030462  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
```

```
5368  030470  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2)  ;ISSUE READ HEADER
5369  030476  012700  000312                  MOV     #50.*4+2,R0        ;ISSUE ENOUGH CLOCKS UNTIL READY
5370  030502  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)  ;  FOR SECTOR PULSE
5371  030510  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5372  030516  005300                          DEC     R0
5373  030520  001370                          BNE     1$
5374  030522  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2)  ;SIMULATE SECTOR PULSE
5375  030530  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5376  030536  012762  000440  000026          MOV     #DMD!MCLK,RKMR1(R2)  ;SHIFT DATA ONE HALF BIT TIME
5377  030544  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5378  030552  005037  003614                  CLR     PR.BIT             ;INITIALIZE PRESENT BIT AND
5379  030556  005037  003616                  CLR     M1.BIT             ;  PREVIOUS BIT
5380  030562  012700  000341                  MOV     #225.,R0
5381  030566  004737  046106          2$:     JSR     PC,RDBIT           ;SIMULATE SYNCH
5382  030572  005300                          DEC     R0
5383  030574  001374                          BNE     2$
5384  030576  012737  000001  003614          MOV     #1,PR.BIT
5385  030604  004737  046106                  JSR     PC,RDBIT
5386  030610  012701  000003                  MOV     #3,R1              ;LOAD NUMBER OF WORDS
5387  030614  012703  066702                  MOV     #HEAD1,R3          ;LOAD ADDRESS OF DATA
5388  030620  012737  000025  003500          MOV     #RDHEAD,E.CS1      ;LOAD EXPECTED CS1
5389  030626  012304          5$:     MOV     (R3)+,R4           ;GET DATA
5390  030630  012700  000020                  MOV     #16.,R0            ;LOAD BIT COUNT
5391  030634  013737  003614  003616  6$:     MOV     PR.BIT,M1.BIT      ;STORE PREVIOUS BIT
5392  030642  006004                          ROR     R4                 ;GET NEXT BIT
5393  030644  103403                          BCS     7$                 ;CHECK IF 1
5394  030646  005037  003614                  CLR     PR.BIT             ;NO, ZERO
5395  030652  000403                          BR      8$                 ;SIMULATE READ DATA
5396
5397  030654  012737  000001  003614  7$:     MOV     #1,PR.BIT          ;ONE
5398  030662  004737  046106          8$:     JSR     PC,RDBIT           ;SIMULATE READ DATA
5399  030666  016237  000000  003440          MOV     RKCS1(R2),T.CS1    ;READ COMMAND AND STATUS REG. 1
5400  030674  023737  003500  003440          CMP     E.CS1,T.CS1        ;CHECK IF CS1 CORRECT
5401  030702  001417                          BEQ     9$                 ;YES, SIMULATE NEXT BIT
5402  030704  012737  000003  003624          MOV     #3,WRDCNT          ;LOAD WORD COUNT
5403  030712  160137  003624                  SUB     R1,WRDCNT
5404  030716  012737  000020  003622          MOV     #16.,BITCNT        ;LOAD BIT COUNT
5405  030724  160037  003622                  SUB     R0,BITCNT
5406  030730  104150                          ERROR   150                ;CS1 INCORRECT DURING HEADER
5407  030732  012762  100000  000000          MOV     #CCLR,RKCS1(R2)    ;CLEAR RK611
5408  030740  000522                          BR      TST35              ;;GO ON TO NEXT TEST
5409
5410  030742  005300          9$:     DEC     R0                 ;CHECK IF READY FOR NEXT WORD
5411  030744  001333                          BNE     6$                 ;NO, GET NEXT BIT
5412  030746  005301                          DEC     R1                 ;CHECK IF HEADER FINISHED
5413  030750  001326                          BNE     5$                 ;NO, GET NEXT WORD
5414  030752  012700  000004                  MOV     #4,R0              ;LOAD COUNT FOR POSTAMBLE
5415  030756  013737  003614  003616  15$:     MOV     PR.BIT,M1.BIT      ;STORE LAST BIT
5416  030764  005037  003614                  CLR     PR.BIT             ;LOAD NEXT BIT
5417  030770  004737  046106                  JSR     PC,RDBIT           ;SIMULATE 1 BIT READ
5418  030774  005300                          DEC     R0                 ;CHECK IF TIME FOR READY
5419  030776  001367                          BNE     15$                ;NO, CONTINUE WITH POSTAMBLE
5420  031000  016237  000000  003440          MOV     RKCS1(R2),T.CS1    ;GET CURRENT CS1
5421  031006  016237  000010  003450          MOV     RKCS2(R2),T.CS2    ;GET CURRENT CS2
5422  031014  012737  000224  003500          MOV     #RDY!RDHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5423  031022  012737  000300  003510          MOV     #OR!IR,E.CS2       ;LOAD EXPECTED CS2
```

```
5424  031030  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
5425  031036  001401                          BEQ     16$             ;YES, CHECK CS2
5426  031040  104151                          ERROR   151             ;CS1 INCORRECT
5427  031042  023737  003510  003450  16$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5428  031050  001401                          BEQ     17$             ;YES, CHECK DATA
5429  031052  104152                          ERROR   152             ;CS2 INCORRECT
5430  031054  005037  003624          17$:    CLR     WRDCNT          ;INITIALIZE WORD COUNT
5431  031060  012703  066702                  MOV     #HEAD1,R3       ;GET ADDRESS OF DATA
5432  031064  012337  003522          20$:    MOV     (R3)+,E.DB      ;GET EXPECTED DATA
5433  031070  016237  000024  003462          MOV     RKDB(R2),T.DB   ;GET ACTUAL DATA
5434  031076  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG. 1
5435  031104  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
5436  031112  022737  000002  003624          CMP     #2,WRDCNT       ;CHECK IF LAST WORD IN DATA BUFFER
5437  031120  001003                          BNE     21$             ;NO, CHECK CS1
5438  031122  012737  000100  003510          MOV     #IR,E.CS2       ;STORE EXPECTED CS2
5439  031130  023737  003500  003440  21$:    CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
5440  031136  001402                          BEQ     22$             ;YES, CHECK CS2
5441  031140  104153                          ERROR   153             ;CS1 INCORRECT
5442  031142  000421                          BR      TST35           ;;GO ON TO NEXT TEST
5443
5444  031144  023737  003510  003450  22$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
5445  031152  001402                          BEQ     23$             ;YES, CHECK DATA
5446  031154  104154                          ERROR   154             ;CS2 INCORRECT
5447  031156  000413                          BR      TST35           ;;GO ON TO NEXT TEST
5448
5449  031160  023737  003522  003462  23$:    CMP     E.DB,T.DB       ;CHECK IF DATA CORRECT
5450  031166  001401                          BEQ     24$             ;YES, GET NEXT HEADER WORD
5451  031170  104155                          ERROR   155             ;DATA INCORRECT
5452  031172  005237  003624          24$:    INC     WRDCNT          ;INCREMENT WORD COUNT
5453  031176  022737  000003  003624          CMP     #3,WRDCNT       ;CHECK IF ALL THREE WORDS CHECK
5454  031204  001327                          BNE     20$             ;NO, GET NEXT WORD
5455
5456                                  .SBTTL  **MFM WRITE LOOPBACK TESTS
5457
5458                                  ;;*****************************************************************
5459                                  ;*TEST 35       WRITE ZEROS UNTIL SECTOR PULSE WITH WRITE HEADER
5460                                  ;*
5461                                  ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
5462                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
5463                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
5464                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE
5465                                  ;*      INDEX PULSE AND 500 DATA BITS.  MAKE SURE THAT
5466                                  ;*      ZEROES ARE WRITTEN. SIMULATE SECTOR PULSE AND MAKE SURE
5467                                  ;*      WRITE GATE RESETS.
5468                                  ;*
5469                                  ;;*****************************************************************
5470  031206  000004                  TST35:  SCOPE
5471  031210  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
5472  031216  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
5473  031222  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5474  031230  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
5475  031236  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;INITIALIZE ROM
5476  031244  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5477  031252  012762  067204  000004          MOV     #WRBUFF,RKBA(R2) ;ISSUE WRITE HEADER
5478  031260  012762  177777  000002          MOV     #-1,RKWC(R2)
5479  031266  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
```

```
005480  031274  012700  002364              MOV     #<256.+48.+64.+256.+10.>*2,R0 ;ISSUE ENOUGH CLOCKS
005481                                              ;  UNTIL READY FOR INDEX PULSE
005482  031300  012762  000440  000026  1$:  MOV     #DMD!MCLK,RKMR1(R2)
005483  031306  012762  000040  000026      MOV     #DMD,RKMR1(R2)
005484  031314  005300                      DEC     R0
005485  031316  001370                      BNE     1$
005486  031320  012700  000004              MOV     #4.,R0          ;SIMULATE INDEX PULSE
005487  031324  012762  000240  000026      MOV     #MIND!DMD,RKMR1(R2)
005488  031332  012762  000640  000026  2$:  MOV     #DMD!MIND!MCLK,RKMR1(R2)
005489  031340  012762  000240  000026      MOV     #DMD!MIND,RKMR1(R2)
005490  031346  005300                      DEC     R0
005491  031350  001370                      BNE     2$
005492  031352  012762  000040  000026      MOV     #DMD,RKMR1(R2)
005493  031360  012737  062040  003524      MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
005494                                              ;  MAINT. REG
005495  031366  012700  000002              MOV     #2,R0           ;WAIT FOR WRITE GATE
005496  031372  012762  000440  000026  3$:  MOV     #DMD!MCLK,RKMR1(R2)
005497  031400  012762  000040  000026      MOV     #DMD,RKMR1(R2)
005498  031406  005300                      DEC     R0
005499  031410  001370                      BNE     3$
005500  031412  005037  003626              CLR     SECCNT          ;CLEAR SECTOR COUNT
005501  031416  005037  003612              CLR     P1.BIT          ;INITIALIZE BIT GENERATION
005502  031422  005037  003614              CLR     PR.BIT
005503  031426  005037  003616              CLR     M1.BIT
005504  031432  005037  003620              CLR     M2.BIT
005505  031436  012700  000764              MOV     #500.,R0        ;LOAD COUNT FOR 500 BITS
005506  031442  012737  062414  003170      MOV     #EM230,EMW      ;LOAD ERROR MESSAGE
005507  031450  005037  003622              CLR     BITCNT          ;CLEAR BIT COUNT
005508  031454  004737  045136          5$:  JSR     PC,WRTBIT       ;WRITE ONE BIT
005509  031460  104170                      ERROR   170             ;ERROR IN WRITE
005510  031462  005237  003622              INC     BITCNT          ;INCREMENT NUMBER OF BITS WRITTEN
005511  031466  005300                      DEC     R0              ;CHECK IF FINISHED
005512  031470  001371                      BNE     5$              ;NO, CONTINUE
005513  031472  042737  040000  003524      BIC     #WRTGAT,E.MR1   ;GENERATE EXPECTED MR1
005514  031500  052737  000100  003524      BIS     #MSP,E.MR1
005515  031506  012762  000140  000026      MOV     #DMD!MSP,RKMR1(R2) ;RAISE SECTOR PULSE
005516  031514  016237  000026  003464      MOV     RKMR1(R2),T.MR1 ;STORE MAINT. REG. 1
005517  031522  023737  003524  003464      CMP     E.MR1,T.MR1     ;STORE MAINT REG 1
005518  031530  001401                      BEQ     10$             ;YES, LOWER SECTOR PULSE
005519  031532  104171                      ERROR   171             ;WRITE GATE DID NOT RESET
005520  031534  042737  000100  003524  10$: BIC     #MSP,E.MR1      ;GENERATE EXPECTED MR1
005521  031542  052737  040000  003524      BIS     #WRTGAT,E.MR1
005522  031550  012762  000040  000026      MOV     #DMD,RKMR1(R2)  ;RESET SECTOR PULSE
005523  031556  016237  000026  003464      MOV     RKMR1(R2),T.MR1 ;STORE MAINT REG 1
005524  031564  023737  003524  003464      CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT
005525  031572  001401                      BEQ     TST36           ;;YES, GO ON TO NEXT TEST
005526  031574  104172                      ERROR   172             ;WRITE GATE DID NOT SET
005527
005528                              ;;*****************************************************************
005529                              ;*TEST 36       WRITE LOOPBACK (PART 1)
005530                              ;*
005531                              ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
005532                              ;*      IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO AN RK06
005533                              ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
005534                              ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
005535                              ;*      INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
```

```
                                ;*      CONSISTING OF THE FOLLOWING DATA, AND AN INDEX PULSE:
                                ;*
                                ;*              177777
                                ;*              000000
                                ;*              177777
                                ;*
                                ;*      MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX PULSE.
                                ;*      CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION.
                                ;*
                                ;;*****************************************************************
031576  000004                  T$T36:  SCOPE
031600  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
031606  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
031612  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
031620  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
031626  012762  066702  000004          MOV     #HEAD1,RKBA(R2) ;ISSUE WRITE HEADER
031634  012762  177775  000002          MOV     #-3,RKWC(R2)
031642  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
031650  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
                                                                ;  READY FOR INDEX PULSE
031654  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
031662  012762  000040  000026          MOV     #DMD,RKMR1(R2)
031670  005300                          DEC     R0
031672  001370                          BNE     1$
031674  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
031700  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
031706  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
031714  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
031722  005300                          DEC     R0
031724  001370                          BNE     2$
031726  012762  000040  000026          MOV     #DMD,RKMR1(R2)
031734  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
031740  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
031746  012762  000040  000026          MOV     #DMD,RKMR1(R2)
031754  005300                          DEC     R0
031756  001370                          BNE     3$
031760  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
031766  012762  000040  000026          MOV     #DMD,RKMR1(R2)
031774  005037  003626                  CLR     SECCNT          ;INITIALIZE SECTOR COUNT
032000  012737  062766  003170          MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
032006  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
                                                                ;  MAINT REG 1
032014  005037  003612                  CLR     P1.BIT          ;INITIALIZE BIT GENERATION
032020  005037  003614                  CLR     PR.BIT
032024  005037  003616                  CLR     M1.BIT
032030  005037  003620                  CLR     M2.BIT
032034  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
032040  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
032044  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
032050  104170                          ERROR   170             ;DATA INCORRECT
032052  005237  003622                  INC     BITCNT
032056  005300                          DEC     R0              ;CHECK IF READY FOR DATA
032060  001371                          BNE     5$              ;NO, GENERATE NEXT BIT
032062  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
032070  004737  045136                  JSR     PC,WRTBIT
032074  104170                          ERROR   170             ;DATA INCORRECT
```

```
5592  032076  005037  003622                    CLR     BITCNT          ;INITIALIZE BIT COUNT
5593  032102  012737  063032  003170            MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
5594  032110  012703  066702                    MOV     #HEAD1,R3       ;LOAD ADDRESS OF DATA
5595  032114  012700  000003                    MOV     #3,R0           ;LOAD NUMBER WORDS IN HEADER
5596  032120  012304                  10$:      MOV     (R3)+,R4        ;GET NEXT WORD
5597  032122  012701  000020                    MOV     #16.,R1         ;LOAD BIT COUNT
5598  032126  013737  003616  003620  12$:      MOV     M1.BIT,M2.BIT   ;SHIFT BITS
5599  032134  013737  003614  003616            MOV     PR.BIT,M1.BIT
5600  032142  013737  003612  003614            MOV     P1.BIT,PR.BIT
5601  032150  006004                            ROR     R4              ;SHIFT IN NEXT BIT
5602  032152  103403                            BCS     14$             ;CHECK IF ONE
5603  032154  005037  003612                    CLR     P1.BIT          ;ZERO
5604  032160  000403                            BR      15$             ;CLOCK IN BIT
5605
5606  032162  012737  000001  003612  14$:      MOV     #1,P1.BIT       ;ONE
5607  032170  004737  045136            15$:    JSR     PC,WRTBIT       ;WRITE BIT
5608  032174  104170                            ERROR   170             ;BIT INCORRECT
5609  032176  005237  003622                    INC     BITCNT          ;INCREMENT BIT COUNT
5610  032202  005301                            DEC     R1              ;CHECK IF WORD FINISHED
5611  032204  001350                            BNE     12$             ;NO, CONTINUE WITH NEXT BIT
5612  032206  005300                            DEC     R0              ;CHECK IF HEADER COMPLETE
5613  032210  001343                            BNE     10$             ;NO, GET NEXT WORD
5614  032212  012701  000020                    MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
5615  032216  013737  003616  003620  18$:      MOV     M1.BIT,M2.BIT   ;SHIFT BITS
5616  032224  013737  003614  003616            MOV     PR.BIT,M1.BIT
5617  032232  013737  003612  003614            MOV     P1.BIT,PR.BIT
5618  032240  005037  003612                    CLR     P1.BIT
5619  032244  004737  045136                    JSR     PC,WRTBIT       ;WRITE ZERO
5620  032250  104170                            ERROR   170             ;BIT INCORRECT
5621  032252  005237  003622                    INC     BITCNT          ;INCREMENT BIT COUNT
5622  032256  005301                            DEC     R1              ;CHECK IF FINISHED
5623  032260  001356                            BNE     18$             ;NO, CLOCK NEXT BIT
5624  032262  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE INDEX
5625  032270  012700  000004                    MOV     #4,R0
5626  032274  012762  000640  000026  20$:      MOV     #DMD!MIND!MCLK,RKMR1(R2)
5627  032302  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)
5628  032310  005300                            DEC     R0
5629  032312  001370                            BNE     20$
5630  032314  012762  000040  000026            MOV     #DMD,RKMR1(R2)
5631  032322  016237  000026  003464            MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
5632  032330  012737  022040  003524            MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
5633  032336  023737  003524  003464            CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET)
5634  032344  001401                            BEQ     25$             ;YES, CHECK IF READY SET
5635  032346  104173                            ERROR   173             ;MAINT REG 1 INCORRECT
5636  032350  012700  000010            25$:    MOV     #8.,R0          ;FINISH COMMAND
5637  032354  012762  000440  000026  26$:      MOV     #DMD!MCLK,RKMR1(R2)
5638  032362  012762  000040  000026            MOV     #DMD,RKMR1(R2)
5639  032370  005300                            DEC     R0
5640  032372  001370                            BNE     26$
5641  032374  016237  000000  003440            MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
5642  032402  012737  000226  003500            MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5643  032410  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
5644  032416  001401                            BEQ     TST37           ;YES, GO ON TO NEXT TEST
5645  032420  104174                            ERROR   174             ;CS1 INCORRECT
5646
5647                          ;;*****************************************************************
```

```
                                ;*TEST 37       WRITE LOOPBACK (PART 2)
                                ;*
                                ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
                                ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
                                ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
                                ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
                                ;*      INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
                                ;*      CONSISTING OF THE FOLLOWING DATA, AND AN INDEX PULSE:
                                ;*
                                ;*              000000
                                ;*              177777
                                ;*              000000
                                ;*
                                ;*      MAKE SURE THAT READY COMES UP AFTER THE THIRD WORD
                                ;*      IS TRANSFERRED.  CHECK FOR CORRECT WRITE ENCODED DATA
                                ;*      AND PRECOMPENSATION.
                                ;*
                                ;;*****************************************************************
032422  000004                  TST37:  SCOPE
032424  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
032432  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
032436  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
032444  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
032452  012762  066710  000004          MOV     #HEAD2,RKBA(R2) ;ISSUE WRITE HEADER
032460  012762  177775  000002          MOV     #-3,RKWC(R2)
032466  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
032474  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
                                                                ;  READY FOR INDEX PULSE
032500  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
032506  012762  000040  000026          MOV     #DMD,RKMR1(R2)
032514  005300                          DEC     R0
032516  001370                          BNE     1$
032520  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
032524  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
032532  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
032540  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
032546  005300                          DEC     R0
032550  001370                          BNE     2$
032552  012762  000040  000026          MOV     #DMD,RKMR1(R2)
032560  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
032564  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
032572  012762  000040  000026          MOV     #DMD,RKMR1(R2)
032600  005300                          DEC     R0
032602  001370                          BNE     3$
032604  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
032612  012762  000040  000026          MOV     #DMD,RKMR1(R2)
032620  005037  003626                  CLR     SECCNT          ;INITIALIZE SECTOR COUNT
032624  012737  062766  003170          MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
032632  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
                                                                ;  MAINT REG 1
032640  005037  003612                  CLR     P1.BIT          ;INITIALIZE BIT GENERATION
032644  005037  003614                  CLR     PR.BIT
032650  005037  003616                  CLR     M1.BIT
032654  005037  003620                  CLR     M2.BIT
032660  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
032664  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
```

```
5704  032670  004737  045136          5$:   JSR     PC,WRTBIT       ;WRITE ONE BIT
5705  032674  104170                        ERROR   170             ;DATA INCORRECT
5706  032676  005237  003622                INC     BITCNT
5707  032702  005300                        DEC     R0              ;CHECK IF READY FOR DATA
5708  032704  001371                        BNE     5$              ;NO, GENERATE NEXT BIT
5709  032706  012737  000001  003612        MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
5710  032714  004737  045136                JSR     PC,WRTBIT
5711  032720  104170                        ERROR   170             ;DATA INCORRECT
5712  032722  005037  003622                CLR     BITCNT          ;INITIALIZE BIT COUNT
5713  032726  012737  063032  003170        MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
5714  032734  012703  066710                MOV     #HEAD2,R3       ;LOAD ADDRESS OF DATA
5715  032740  012700  000003                MOV     #3,R0           ;LOAD NUMBER WORDS IN HEADER
5716  032744  012304                  10$:  MOV     (R3)+,R4        ;GET NEXT WORD
5717  032746  012701  000020                MOV     #16.,R1         ;LOAD BIT COUNT
5718  032752  013737  003616  003620  12$:  MOV     M1.BIT,M2.BIT   ;SHIFT BITS
5719  032760  013737  003614  003616        MOV     PR.BIT,M1.BIT
5720  032766  013737  003612  003614        MOV     P1.BIT,PR.BIT
5721  032774  006004                        ROR     R4              ;SHIFT IN NEXT BIT
5722  032776  103403                        BCS     14$             ;CHECK IF ONE
5723  033000  005037  003612                CLR     P1.BIT          ;ZERO
5724  033004  000403                        BR      15$             ;CLOCK IN BIT
5725
5726  033006  012737  000001  003612  14$:  MOV     #1,P1.BIT       ;ONE
5727  033014  004737  045136          15$:  JSR     PC,WRTBIT       ;WRITE BIT
5728  033020  104170                        ERROR   170             ;BIT INCORRECT
5729  033022  005237  003622                INC     BITCNT          ;INCREMENT BIT COUNT
5730  033026  005301                        DEC     R1              ;CHECK IF WORD FINISHED
5731  033030  001350                        BNE     12$             ;NO, CONTINUE WITH NEXT BIT
5732  033032  005300                        DEC     R0              ;CHECK IF HEADER COMPLETE
5733  033034  001343                        BNE     10$             ;NO, GET NEXT WORD
5734  033036  012701  000020                MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
5735  033042  013737  003616  003620  18$:  MOV     M1.BIT,M2.BIT   ;SHIFT BITS
5736  033050  013737  003614  003616        MOV     PR.BIT,M1.BIT
5737  033056  013737  003612  003614        MOV     P1.BIT,PR.BIT
5738  033064  005037  003612                CLR     P1.BIT
5739  033070  004737  045136                JSR     PC,WRTBIT       ;WRITE ZERO
5740  033074  104170                        ERROR   170             ;BIT INCORRECT
5741  033076  005237  003622                INC     BITCNT          ;INCREMENT BIT COUNT
5742  033102  005301                        DEC     R1              ;CHECK IF FINISHED
5743  033104  001356                        BNE     18$             ;NO, CLOCK NEXT BIT
5744  033106  012762  000240  000026        MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE INDEX
5745  033114  012700  000004                MOV     #4,R0
5746  033120  012762  000640  000026  20$:  MOV     #DMD!MIND!MCLK,RKMR1(R2)
5747  033126  012762  000240  000026        MOV     #DMD!MIND,RKMR1(R2)
5748  033134  005300                        DEC     R0
5749  033136  001370                        BNE     20$
5750  033140  012762  000040  000026        MOV     #DMD,RKMR1(R2)
5751  033146  016237  000026  003464        MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
5752  033154  012737  022040  003524        MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
5753  033162  023737  003524  003464        CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET)
5754  033170  001401                        BEQ     25$             ;YES, CHECK IF READY SET
5755  033172  104173                        ERROR   173             ;MAINT REG 1 INCORRECT
5756  033174  012700  000010          25$:  MOV     #8.,R0          ;FINISH COMMAND
5757  033200  012762  000440  000026  26$:  MOV     #DMD!MCLK,RKMR1(R2)
5758  033206  012762  000040  000026        MOV     #DMD,RKMR1(R2)
5759  033214  005300                        DEC     R0
```

```
5760  033216  001370                        BNE     26$
5761  033220  016237  000000  003440        MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
5762  033226  012737  000226  003500        MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
5763  033234  023737  003500  003440        CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
5764  033242  001401                        BEQ     TST40           ;;YES, GO ON TO NEXT TEST
5765  033244  104174                        ERROR   174             ;CS1 INCORRECT
5766
5767                                ;;*****************************************************************
5768                                ;*TEST 40       WRITE LOOPBACK (PART 3)
5769                                ;*
5770                                ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
5771                                ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
5772                                ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
5773                                ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  STIMULATE
5774                                ;*      INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
5775                                ;*      CONSISTING OF THE FOLLOWING DATA, AND AN INDEX PULSE:
5776                                ;*
5777                                ;*              125252
5778                                ;*              052525
5779                                ;*              125252
5780                                ;*
5781                                ;*      MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX PULSE.
5782                                ;*      CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION.
5783                                ;*
5784                                ;*****************************************************************
5785  033246  000004                TST40:  SCOPE
5786  033250  012737  000144  001200        MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
5787  033256  013702  001270                MOV     $BASE,R2        ;LOAD RK611 BASE
5788  033262  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5789  033270  012762  000040  000026        MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
5790  033276  012762  066716  000004        MOV     #HEAD3,RKBA(R2) ;ISSUE WRITE HEADER
5791  033304  012762  177775  000002        MOV     #-3,RKWC(R2)
5792  033312  012762  000027  000000        MOV     #WRHEAD,RKCS1(R2)
5793  033320  012700  000312                MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
5794                                                                ;  READY FOR INDEX PULSE
5795  033324  012762  000440  000026  1$:   MOV     #DMD!MCLK,RKMR1(R2)
5796  033332  012762  000040  000026        MOV     #DMD,RKMR1(R2)
5797  033340  005300                        DEC     R0
5798  033342  001370                        BNE     1$
5799  033344  012700  000004                MOV     #4,R0           ;ISSUE INDEX PULSE
5800  033350  012762  000240  000026        MOV     #DMD!MIND,RKMR1(R2)
5801  033356  012762  000640  000026  2$:   MOV     #DMD!MIND!MCLK,RKMR1(R2)
5802  033364  012762  000240  000026        MOV     #DMD!MIND,RKMR1(R2)
5803  033372  005300                        DEC     R0
5804  033374  001370                        BNE     2$
5805  033376  012762  000040  000026        MOV     #DMD,RKMR1(R2)
5806  033404  012700  000010                MOV     #8.,R0          ;WAIT FOR WRITE GATE
5807  033410  012762  000440  000026  3$:   MOV     #DMD!MCLK,RKMR1(R2)
5808  033416  012762  000040  000026        MOV     #DMD,RKMR1(R2)
5809  033424  005300                        DEC     R0
5810  033426  001370                        BNE     3$
5811  033430  012762  000140  000026        MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
5812  033436  012762  000040  000026        MOV     #DMD,RKMR1(R2)
5813  033444  005037  003626                CLR     SECCNT          ;INITIALIZE SECTOR COUNT
5814  033450  012737  062766  003170        MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
5815  033456  012737  062040  003524        MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
```

```
5816                                                    ;  MAINT REG 1
5817  033464  005037  003612                  CLR     P1.BIT          ;INITIALIZE BIT GENERATION
5818  033470  005037  003614                  CLR     PR.BIT
5819  033474  005037  003616                  CLR     M1.BIT
5820  033500  005037  003620                  CLR     M2.BIT
5821  033504  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
5822  033510  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
5823  033514  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
5824  033520  104170                          ERROR   170             ;DATA INCORRECT
5825  033522  005237  003622                  INC     BITCNT
5826  033526  005300                          DEC     R0              ;CHECK IF READY FOR DATA
5827  033530  001371                          BNE     5$              ;NO, GENERATE NEXT BIT
5828  033532  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
5829  033540  004737  045136                  JSR     PC,WRTBIT
5830  033544  104170                          ERROR   170             ;DATA INCORRECT
5831  033546  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
5832  033552  012737  063032  003170          MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
5833  033560  012703  066716                  MOV     #HEAD3,R3       ;LOAD ADDRESS OF DATA
5834  033564  012700  000003                  MOV     #3,R0           ;LOAD NUMBER WORDS IN HEADER
5835  033570  012304                  10$:    MOV     (R3)+,R4        ;GET NEXT WORD
5836  033572  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT
5837  033576  013737  003616  003620  12$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
5838  033604  013737  003614  003616          MOV     PR.BIT,M1.BIT
5839  033612  013737  003612  003614          MOV     P1.BIT,PR.BIT
5840  033620  006004                          ROR     R4              ;SHIFT IN NEXT BIT
5841  033622  103403                          BCS     14$             ;CHECK IF ONE
5842  033624  005037  003612                  CLR     P1.BIT          ;ZERO
5843  033630  000403                          BR      15$             ;CLOCK IN BIT
5844
5845  033632  012737  000001  003612  14$:    MOV     #1,P1.BIT       ;ONE
5846  033640  004737  045136          15$:    JSR     PC,WRTBIT       ;WRITE BIT
5847  033644  104170                          ERROR   170             ;BIT INCORRECT
5848  033646  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
5849  033652  005301                          DEC     R1              ;CHECK IF WORD FINISHED
5850  033654  001350                          BNE     12$             ;NO, CONTINUE WITH NEXT BIT
5851  033656  005300                          DEC     R0              ;CHECK IF HEADER COMPLETE
5852  033660  001343                          BNE     10$             ;NO, GET NEXT WORD
5853  033662  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
5854  033666  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
5855  033674  013737  003614  003616          MOV     PR.BIT,M1.BIT
5856  033702  013737  003612  003614          MOV     P1.BIT,PR.BIT
5857  033710  005037  003612                  CLR     P1.BIT
5858  033714  004737  045136                  JSR     PC,WRTBIT       ;WRITE ZERO
5859  033720  104170                          ERROR   170             ;BIT INCORRECT
5860  033722  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
5861  033726  005301                          DEC     R1              ;CHECK IF FINISHED
5862  033730  001356                          BNE     18$             ;NO, CLOCK NEXT BIT
5863  033732  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE INDEX
5864  033740  012700  000004                  MOV     #4,R0
5865  033744  012762  000640  000026  20$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
5866  033752  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
5867  033760  005300                          DEC     R0
5868  033762  001370                          BNE     20$
5869  033764  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5870  033772  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
5871  034000  012737  022040  003524          MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
```

```
5872  034006  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET)
5873  034014  001401                          BEQ     25$             ;YES, CHECK IF READY SET
5874  034016  104173                          ERROR   173             ;MAINT REG 1 INCORRECT
5875  034020  012700  000010          25$:    MOV     #8.,R0          ;FINISH COMMAND
5876  034024  012762  000440  000026  26$:    MOV     #DMD!MCLK,RKMR1(R2)
5877  034032  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5878  034040  005300                          DEC     R0
5879  034042  001370                          BNE     26$
5880  034044  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
5881  034052  012737  000226  003500          MOV     #RDY!WRHEAD&^C<GO>,E.CS1 ;LOAD EXPECTED CS1
5882  034060  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
5883  034066  001401                          BEQ     TST41           ;;YES, GO ON TO NEXT TEST
5884  034070  104174                          ERROR   174             ;CS1 INCORRECT
5885
5886                                  ;;*****************************************************************
5887                                  ;*TEST 41       WRITE LOOPBACK (PART 4)
5888                                  ;*
5889                                  ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
5890                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
5891                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
5892                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
5893                                  ;*      INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
5894                                  ;*      CONSISTING OF THE FOLLOWING DATA, AND AND INDEX PULSE:
5895                                  ;*
5896                                  ;*              044444
5897                                  ;*              022222
5898                                  ;*              111111
5899                                  ;*
5900                                  ;*      MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX MODE.
5901                                  ;*      CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION.
5902                                  ;*
5903                                  ;;*****************************************************************
5904  034072  000004                  TST41:  SCOPE
5905  034074  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
5906  034102  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
5907  034106  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
5908  034114  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
5909  034122  012762  066732  000004          MOV     #HEAD4,RKBA(R2) ;ISSUE WRITE HEADER
5910  034130  012762  177775  000002          MOV     #-3,RKWC(R2)
5911  034136  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
5912  034144  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
5913                                                                  ;  READY FOR INDEX PULSE
5914  034150  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
5915  034156  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5916  034164  005300                          DEC     R0
5917  034166  001370                          BNE     1$
5918  034170  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
5919  034174  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
5920  034202  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
5921  034210  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
5922  034216  005300                          DEC     R0
5923  034220  001370                          BNE     2$
5924  034222  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5925  034230  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
5926  034234  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
5927  034242  012762  000040  000026          MOV     #DMD,RKMR1(R2)
```

```
5928  034250  005300                          DEC     R0
5929  034252  001370                          BNE     3$
5930  034254  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
5931  034262  012762  000040  000026          MOV     #DMD,RKMR1(R2)
5932  034270  005037  003626                  CLR     SECCNT             ;INITIALIZE SECTOR COUNT
5933  034274  012737  062766  003170          MOV     #EM233,EMW         ;LOAD ERROR MESSAGE
5934  034302  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
5935                                                                     ;  MAINT REG 1
5936  034310  005037  003612                  CLR     P1.BIT             ;INITIALIZE BIT GENERATION
5937  034314  005037  003614                  CLR     PR.BIT
5938  034320  005037  003616                  CLR     M1.BIT
5939  034324  005037  003620                  CLR     M2.BIT
5940  034330  012700  000400                  MOV     #256.,R0           ;SIMULATE SYNCH
5941  034334  005037  003622                  CLR     BITCNT             ;INITIALIZE BIT COUNT
5942  034340  004737  045136          5$:     JSR     PC,WRTBIT          ;WRITE ONE BIT
5943  034344  104170                          ERROR   170                ;DATA INCORRECT
5944  034346  005237  003622                  INC     BITCNT
5945  034352  005300                          DEC     R0                 ;CHECK IF READY FOR DATA
5946  034354  001371                          BNE     5$                 ;NO, GENERATE NEXT BIT
5947  034356  012737  000001  003612          MOV     #1,P1.BIT          ;PUT IN SYNCH BIT
5948  034364  004737  045136                  JSR     PC,WRTBIT
5949  034370  104170                          ERROR   170                ;DATA INCORRECT
5950  034372  005037  003622                  CLR     BITCNT             ;INITIALIZE BIT COUNT
5951  034376  012737  063032  003170          MOV     #EM234,EMW         ;LOAD ERROR MESSAGE
5952  034404  012703  066732                  MOV     #HEAD4,R3          ;LOAD ADDRESS OF DATA
5953  034410  012700  000003                  MOV     #3,R0              ;LOAD NUMBER WORDS IN HEADER
5954  034414  012304                  10$:    MOV     (R3)+,R4           ;GET NEXT WORD
5955  034416  012701  000020                  MOV     #16.,R1            ;LOAD BIT COUNT
5956  034422  013737  003616  003620  12$:    MOV     M1.BIT,M2.BIT      ;SHIFT BITS
5957  034430  013737  003614  003616          MOV     PR.BIT,M1.BIT
5958  034436  013737  003612  003614          MOV     P1.BIT,PR.BIT
5959  034444  006004                          ROR     R4                 ;SHIFT IN NEXT BIT
5960  034446  103403                          BCS     14$                ;CHECK IF ONE
5961  034450  005037  003612                  CLR     P1.BIT             ;ZERO
5962  034454  000403                          BR      15$                ;CLOCK IN BIT
5963
5964  034456  012737  000001  003612  14$:    MOV     #1,P1.BIT          ;ONE
5965  034464  004737  045136          15$:    JSR     PC,WRTBIT          ;WRITE BIT
5966  034470  104170                          ERROR   170                ;BIT INCORRECT
5967  034472  005237  003622                  INC     BITCNT             ;INCREMENT BIT COUNT
5968  034476  005301                          DEC     R1                 ;CHECK IF WORD FINISHED
5969  034500  001350                          BNE     12$                ;NO, CONTINUE WITH NEXT BIT
5970  034502  005300                          DEC     R0                 ;CHECK IF HEADER COMPLETE
5971  034504  001343                          BNE     10$                ;NO, GET NEXT WORD
5972  034506  012701  000020                  MOV     #16.,R1            ;LOAD BIT COUNT FOR NEXT WORD
5973  034512  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT      ;SHIFT BITS
5974  034520  013737  003614  003616          MOV     PR.BIT,M1.BIT
5975  034526  013737  003612  003614          MOV     P1.BIT,PR.BIT
5976  034534  005037  003612                  CLR     P1.BIT
5977  034540  004737  045136                  JSR     PC,WRTBIT          ;WRITE ZERO
5978  034544  104170                          ERROR   170                ;BIT INCORRECT
5979  034546  005237  003622                  INC     BITCNT             ;INCREMENT BIT COUNT
5980  034552  005301                          DEC     R1                 ;CHECK IF FINISHED
5981  034554  001356                          BNE     18$                ;NO, CLOCK NEXT BIT
5982  034556  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2) ;SIMULATE INDEX
5983  034564  012700  000004                  MOV     #4,R0
```

```
 5984  034570  012762  000640  000026  20$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
 5985  034576  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
 5986  034604  005300                          DEC     R0
 5987  034606  001370                          BNE     20$
 5988  034610  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 5989  034616  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
 5990  034624  012737  022040  003524          MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
 5991  034632  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET)
 5992  034640  001401                          BEQ     25$             ;YES, CHECK IF READY SET
 5993  034642  104173                          ERROR   173             ;MAINT REG 1 INCORRECT
 5994  034644  012700  000010          25$:    MOV     #8.,R0          ;FINISH COMMAND
 5995  034650  012762  000440  000026  26$:    MOV     #DMD!MCLK,RKMR1(R2)
 5996  034656  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 5997  034664  005300                          DEC     R0
 5998  034666  001370                          BNE     26$
 5999  034670  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
 6000  034676  012737  000226  003500          MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
 6001  034704  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
 6002  034712  001401                          BEQ     TST42           ;;YES, GO ON TO NEXT TEST
 6003  034714  104174                          ERROR   174             ;CS1 INCORRECT
 6004
 6005                                  ;;*****************************************************************
 6006                                  ;*TEST 42       WRITE LOOPBACK (PART 5)
 6007                                  ;*
 6008                                  ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
 6009                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
 6010                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
 6011                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
 6012                                  ;*      INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
 6013                                  ;*      CONSISTING OF THE FOLLOWING DATA, AND INDEX PULSE:
 6014                                  ;*
 6015                                  ;*              052012
 6016                                  ;*              100520
 6017                                  ;*              052012
 6018                                  ;*
 6019                                  ;*      MAKE SURE THAT READY COMES UP AFTER THE SECOND INDEX PULSE.
 6020                                  ;*      CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION.
 6021                                  ;*
 6022                                  ;*****************************************************************
 6023  034716  000004                  TST42:  SCOPE
 6024  034720  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
 6025  034726  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
 6026  034732  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 6027  034740  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
 6028  034746  012762  066740  000004          MOV     #HEAD5,RKBA(R2) ;ISSUE WRITE HEADER
 6029  034754  012762  177775  000002          MOV     #-3,RKWC(R2)
 6030  034762  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
 6031  034770  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
 6032                                                                  ;  READY FOR INDEX PULSE
 6033  034774  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
 6034  035002  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 6035  035010  005300                          DEC     R0
 6036  035012  001370                          BNE     1$
 6037  035014  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
 6038  035020  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
 6039  035026  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
```

```
6040  035034  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6041  035042  005300                          DEC     R0
6042  035044  001370                          BNE     2$
6043  035046  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6044  035054  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
6045  035060  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
6046  035066  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6047  035074  005300                          DEC     R0
6048  035076  001370                          BNE     3$
6049  035100  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
6050  035106  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6051  035114  005037  003626                  CLR     SECCNT          ;INITIALIZE SECTOR COUNT
6052  035120  012737  062766  003170          MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
6053  035126  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
6054                                                                  ;  MAINT REG 1
6055  035134  005037  003612                  CLR     P1.BIT          ;INITIALIZE BIT GENERATION
6056  035140  005037  003614                  CLR     PR.BIT
6057  035144  005037  003616                  CLR     M1.BIT
6058  035150  005037  003620                  CLR     M2.BIT
6059  035154  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
6060  035160  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6061  035164  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
6062  035170  104170                          ERROR   170             ;DATA INCORRECT
6063  035172  005237  003622                  INC     BITCNT
6064  035176  005300                          DEC     R0              ;CHECK IF READY FOR DATA
6065  035200  001371                          BNE     5$              ;NO, GENERATE NEXT BIT
6066  035202  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
6067  035210  004737  045136                  JSR     PC,WRTBIT
6068  035214  104170                          ERROR   170             ;DATA INCORRECT
6069  035216  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6070  035222  012737  063032  003170          MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
6071  035230  012703  066740                  MOV     #HEAD5,R3       ;LOAD ADDRESS OF DATA
6072  035234  012700  000003                  MOV     #3,R0           ;LOAD NUMBER WORDS IN HEADER
6073  035240  012304                  10$:     MOV     (R3)+,R4        ;GET NEXT WORD
6074  035242  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT
6075  035246  013737  003616  003620  12$:     MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6076  035254  013737  003614  003616          MOV     PR.BIT,M1.BIT
6077  035262  013737  003612  003614          MOV     P1.BIT,PR.BIT
6078  035270  006004                          ROR     R4              ;SHIFT IN NEXT BIT
6079  035272  103403                          BCS     14$             ;CHECK IF ONE
6080  035274  005037  003612                  CLR     P1.BIT          ;ZERO
6081  035300  000403                          BR      15$             ;CLOCK IN BIT
6082
6083  035302  012737  000001  003612  14$:     MOV     #1,P1.BIT       ;ONE
6084  035310  004737  045136          15$:     JSR     PC,WRTBIT       ;WRITE BIT
6085  035314  104170                          ERROR   170             ;BIT INCORRECT
6086  035316  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6087  035322  005301                          DEC     R1              ;CHECK IF WORD FINISHED
6088  035324  001350                          BNE     12$             ;NO, CONTINUE WITH NEXT BIT
6089  035326  005300                          DEC     R0              ;CHECK IF HEADER COMPLETE
6090  035330  001343                          BNE     10$             ;NO, GET NEXT WORD
6091  035332  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
6092  035336  013737  003616  003620  18$:     MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6093  035344  013737  003614  003616          MOV     PR.BIT,M1.BIT
6094  035352  013737  003612  003614          MOV     P1.BIT,PR.BIT
6095  035360  005037  003612                  CLR     P1.BIT
```

```
6096  035364  004737  045136                      JSR     PC,WRTBIT       ;WRITE ZERO
6097  035370  104170                              ERROR   170             ;BIT INCORRECT
6098  035372  005237  003622                      INC     BITCNT          ;INCREMENT BIT COUNT
6099  035376  005301                              DEC     R1              ;CHECK IF FINISHED
6100  035400  001356                              BNE     18$             ;NO, CLOCK NEXT BIT
6101  035402  012762  000240  000026              MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE INDEX
6102  035410  012700  000004                      MOV     #4,R0
6103  035414  012762  000640  000026  20$:       MOV     #DMD!MIND!MCLK,RKMR1(R2)
6104  035422  012762  000240  000026              MOV     #DMD!MIND,RKMR1(R2)
6105  035430  005300                              DEC     R0
6106  035432  001370                              BNE     20$
6107  035434  012762  000040  000026              MOV     #DMD,RKMR1(R2)
6108  035442  016237  000026  003464              MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
6109  035450  012737  022040  003524              MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
6110  035456  023737  003524  003464              CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET)
6111  035464  001401                              BEQ     25$             ;YES, CHECK IF READY SET
6112  035466  104173                              ERROR   173             ;MAINT REG 1 INCORRECT
6113  035470  012700  000010              25$:    MOV     #8.,R0          ;FINISH COMMAND
6114  035474  012762  000440  000026  26$:       MOV     #DMD!MCLK,RKMR1(R2)
6115  035502  012762  000040  000026              MOV     #DMD,RKMR1(R2)
6116  035510  005300                              DEC     R0
6117  035512  001370                              BNE     26$
6118  035514  016237  000000  003440              MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
6119  035522  012737  000226  003500              MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
6120  035530  023737  003500  003440              CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
6121  035536  001401                              BEQ     TST43           ;;YES, GO ON TO NEXT TEST
6122  035540  104174                              ERROR   174             ;CS1 INCORRECT
6123
6124                                      ;;*****************************************************************
6125                                      ;*TEST 43       WRITE LOOPBACK (PART 6)
6126                                      ;*
6127                                      ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
6128                                      ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
6129                                      ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
6130                                      ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
6131                                      ;*      INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
6132                                      ;*      CONSISTING OF THE FOLLOWING DATA, AND INDEX PULSE:
6133                                      ;*
6134                                      ;*              155555
6135                                      ;*              066666
6136                                      ;*              155555
6137                                      ;*
6138                                      ;*      MAKE SURE READY COMES UP AFTER SECOND INDEX PULSE.
6139                                      ;*      CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION.
6140                                      ;*
6141                                      ;*****************************************************************
6142  035542  000004                      TST43:  SCOPE
6143  035544  012737  000144  001200              MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
6144  035552  013702  001270                      MOV     $BASE,R2        ;LOAD RK611 BASE
6145  035556  012762  100000  000000              MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
6146  035564  012762  000040  000026              MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
6147  035572  012762  066746  000004              MOV     #HEAD6,RKBA(R2) ;ISSUE WRITE HEADER
6148  035600  012762  177775  000002              MOV     #-3,RKWC(R2)
6149  035606  012762  000027  000000              MOV     #WRHEAD,RKCS1(R2)
6150  035614  012700  000312                      MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
6151                                                                      ;  READY FOR INDEX PULSE
```

```
6152  035620  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
6153  035626  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6154  035634  005300                          DEC     R0
6155  035636  001370                          BNE     1$
6156  035640  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
6157  035644  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6158  035652  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
6159  035660  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6160  035666  005300                          DEC     R0
6161  035670  001370                          BNE     2$
6162  035672  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6163  035700  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
6164  035704  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
6165  035712  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6166  035720  005300                          DEC     R0
6167  035722  001370                          BNE     3$
6168  035724  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
6169  035732  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6170  035740  005037  003626                  CLR     SECCNT          ;INITIALIZE SECTOR COUNT
6171  035744  012737  062766  003170          MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
6172  035752  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
6173                                                                  ;  MAINT REG 1
6174  035760  005037  003612                  CLR     P1.BIT          ;INITIALIZE BIT GENERATION
6175  035764  005037  003614                  CLR     PR.BIT
6176  035770  005037  003616                  CLR     M1.BIT
6177  035774  005037  003620                  CLR     M2.BIT
6178  036000  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
6179  036004  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6180  036010  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
6181  036014  104170                          ERROR   170             ;DATA INCORRECT
6182  036016  005237  003622                  INC     BITCNT
6183  036022  005300                          DEC     R0              ;CHECK IF READY FOR DATA
6184  036024  001371                          BNE     5$              ;NO, GENERATE NEXT BIT
6185  036026  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
6186  036034  004737  045136                  JSR     PC,WRTBIT
6187  036040  104170                          ERROR   170             ;DATA INCORRECT
6188  036042  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6189  036046  012737  063032  003170          MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
6190  036054  012703  066746                  MOV     #HEAD6,R3       ;LOAD ADDRESS OF DATA
6191  036060  012700  000003                  MOV     #3,R0           ;LOAD NUMBER WORDS IN HEADER
6192  036064  012304                  10$:    MOV     (R3)+,R4        ;GET NEXT WORD
6193  036066  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT
6194  036072  013737  003616  003620  12$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6195  036100  013737  003614  003616          MOV     PR.BIT,M1.BIT
6196  036106  013737  003612  003614          MOV     P1.BIT,PR.BIT
6197  036114  006004                          ROR     R4              ;SHIFT IN NEXT BIT
6198  036116  103403                          BCS     14$             ;CHECK IF ONE
6199  036120  005037  003612                  CLR     P1.BIT          ;ZERO
6200  036124  000403                          BR      15$             ;CLOCK IN BIT
6201
6202  036126  012737  000001  003612  14$:    MOV     #1,P1.BIT       ;ONE
6203  036134  004737  045136          15$:    JSR     PC,WRTBIT       ;WRITE BIT
6204  036140  104170                          ERROR   170             ;BIT INCORRECT
6205  036142  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6206  036146  005301                          DEC     R1              ;CHECK IF WORD FINISHED
6207  036150  001350                          BNE     12$             ;NO, CONTINUE WITH NEXT BIT
```

```
6208  036152  005300                          DEC     R0              ;CHECK IF HEADER COMPLETE
6209  036154  001343                          BNE     10$             ;NO, GET NEXT WORD
6210  036156  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
6211  036162  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6212  036170  013737  003614  003616          MOV     PR.BIT,M1.BIT
6213  036176  013737  003612  003614          MOV     P1.BIT,PR.BIT
6214  036204  005037  003612                  CLR     P1.BIT
6215  036210  004737  045136                  JSR     PC,WRTBIT       ;WRITE ZERO
6216  036214  104170                          ERROR   170             ;BIT INCORRECT
6217  036216  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6218  036222  005301                          DEC     R1              ;CHECK IF FINISHED
6219  036224  001356                          BNE     18$             ;NO, CLOCK NEXT BIT
6220  036226  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE INDEX
6221  036234  012700  000004                  MOV     #4,R0
6222  036240  012762  000640  000026  20$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
6223  036246  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6224  036254  005300                          DEC     R0
6225  036256  001370                          BNE     20$
6226  036260  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6227  036266  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
6228  036274  012737  022040  003524          MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
6229  036302  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET)
6230  036310  001401                          BEQ     25$             ;YES, CHECK IF READY SET
6231  036312  104173                          ERROR   173             ;MAINT REG 1 INCORRECT
6232  036314  012700  000010          25$:    MOV     #8.,R0          ;FINISH COMMAND
6233  036320  012762  000440  000026  26$:    MOV     #DMD!MCLK,RKMR1(R2)
6234  036326  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6235  036334  005300                          DEC     R0
6236  036336  001370                          BNE     26$
6237  036340  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
6238  036346  012737  000226  003500          MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
6239  036354  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
6240  036362  001401                          BEQ     TST44           ;;YES, GO ON TO NEXT TEST
6241  036364  104174                          ERROR   174             ;CS1 INCORRECT
6242
6243                                  ;;*****************************************************************
6244                                  ;*TEST 44      WRITE LOOPBACK (PART 7)
6245                                  ;*
6246                                  ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
6247                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
6248                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
6249                                  ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
6250                                  ;*      INDEX PULSE, SECTOR PULSE, ONE THREE WORD HEADER
6251                                  ;*      CONSISTING OF THE FOLLOWING DATA, AND INDEX PULSE:
6252                                  ;*
6253                                  ;*              104210
6254                                  ;*              104210
6255                                  ;*              104210
6256                                  ;*
6257                                  ;*      MAKE SURE READY COMES UP AFTER SECOND INDEX PULSE.
6258                                  ;*      CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMPENSATION.
6259                                  ;*
6260                                  ;*****************************************************************
6261  036366  000004                  TST44:  SCOPE
6262  036370  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
6263  036376  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
```

```
6264  036402  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
6265  036410  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
6266  036416  012762  066754  000004          MOV     #HEAD7,RKBA(R2) ;ISSUE WRITE HEADER
6267  036424  012762  177775  000002          MOV     #-3,RKWC(R2)
6268  036432  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
6269  036440  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
6270                                                                  ;  READY FOR INDEX PULSE
6271  036444  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
6272  036452  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6273  036460  005300                          DEC     R0
6274  036462  001370                          BNE     1$
6275  036464  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
6276  036470  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6277  036476  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
6278  036504  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6279  036512  005300                          DEC     R0
6280  036514  001370                          BNE     2$
6281  036516  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6282  036524  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
6283  036530  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
6284  036536  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6285  036544  005300                          DEC     R0
6286  036546  001370                          BNE     3$
6287  036550  012762  000140  000026          MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
6288  036556  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6289  036564  005037  003626                  CLR     SECCNT          ;INITIALIZE SECTOR COUNT
6290  036570  012737  062766  003170          MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
6291  036576  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
6292                                                                  ;  MAINT REG 1
6293  036604  005037  003612                  CLR     P1.BIT          ;INITIALIZE BIT GENERATION
6294  036610  005037  003614                  CLR     PR.BIT
6295  036614  005037  003616                  CLR     M1.BIT
6296  036620  005037  003620                  CLR     M2.BIT
6297  036624  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
6298  036630  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6299  036634  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
6300  036640  104170                          ERROR   17O             ;DATA INCORRECT
6301  036642  005237  003622                  INC     BITCNT
6302  036646  005300                          DEC     R0              ;CHECK IF READY FOR DATA
6303  036650  001371                          BNE     5$              ;NO, GENERATE NEXT BIT
6304  036652  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
6305  036660  004737  045136                  JSR     PC,WRTBIT
6306  036664  104170                          ERROR   17O             ;DATA INCORRECT
6307  036666  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6308  036672  012737  063032  003170          MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
6309  036700  012703  066754                  MOV     #HEAD7,R3       ;LOAD ADDRESS OF DATA
6310  036704  012700  000003                  MOV     #3,R0           ;LOAD NUMBER WORDS IN HEADER
6311  036710  012304                  10$:    MOV     (R3)+,R4        ;GET NEXT WORD
6312  036712  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT
6313  036716  013737  003616  003620  12$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6314  036724  013737  003614  003616          MOV     PR.BIT,M1.BIT
6315  036732  013737  003612  003614          MOV     P1.BIT,PR.BIT
6316  036740  006004                          ROR     R4              ;SHIFT IN NEXT BIT
6317  036742  103403                          BCS     14$             ;CHECK IF ONE
6318  036744  005037  003612                  CLR     P1.BIT          ;ZERO
6319  036750  000403                          BR      15$             ;CLOCK IN BIT
```

```
6320
6321  036752  012737  000001  003612  14$:    MOV     #1,P1.BIT       ;ONE
6322  036760  004737  045136          15$:    JSR     PC,WRTBIT       ;WRITE BIT
6323  036764  104170                          ERROR   170             ;BIT INCORRECT
6324  036766  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6325  036772  005301                          DEC     R1              ;CHECK IF WORD FINISHED
6326  036774  001350                          BNE     12$             ;NO, CONTINUE WITH NEXT BIT
6327  036776  005300                          DEC     R0              ;CHECK IF HEADER COMPLETE
6328  037000  001343                          BNE     10$             ;NO, GET NEXT WORD
6329  037002  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
6330  037006  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6331  037014  013737  003614  003616          MOV     PR.BIT,M1.BIT
6332  037022  013737  003612  003614          MOV     P1.BIT,PR.BIT
6333  037030  005037  003612                  CLR     P1.BIT
6334  037034  004737  045136                  JSR     PC,WRTBIT       ;WRITE ZERO
6335  037040  104170                          ERROR   170             ;BIT INCORRECT
6336  037042  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6337  037046  005301                          DEC     R1              ;CHECK IF FINISHED
6338  037050  001356                          BNE     18$             ;NO, CLOCK NEXT BIT
6339  037052  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)  ;SIMULATE INDEX
6340  037060  012700  000004                  MOV     #4,R0
6341  037064  012762  000640  000026  20$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
6342  037072  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6343  037100  005300                          DEC     R0
6344  037102  001370                          BNE     20$
6345  037104  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6346  037112  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
6347  037120  012737  022040  003524          MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
6348  037126  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET)
6349  037134  001401                          BEQ     25$             ;YES, CHECK IF READY SET
6350  037136  104173                          ERROR   173             ;MAINT REG 1 INCORRECT
6351  037140  012700  000010          25$:    MOV     #8.,R0          ;FINISH COMMAND
6352  037144  012762  000440  000026  26$:    MOV     #DMD!MCLK,RKMR1(R2)
6353  037152  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6354  037160  005300                          DEC     R0
6355  037162  001370                          BNE     26$
6356  037164  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
6357  037172  012737  000226  003500          MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
6358  037200  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
6359  037206  001401                          BEQ     TST45           ;;YES, GO ON TO NEXT TEST
6360  037210  104174                          ERROR   174             ;CS1 INCORRECT
6361
6362                          ;;*****************************************************************
6363                          ;*TEST 45       WRITE TWO HEADERS
6364                          ;*
6365                          ;*      CLEAR RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
6366                          ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
6367                          ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0.
6368                          ;*      CLOCK BOTH SEEK AND DRIVE CLEAR MESSAGES.  SIMULATE
6369                          ;*      INDEX PULSE, SECTOR PULSE, THREE WORD HEADER
6370                          ;*      CONSISTING OF THE FOLLOWING DATA:
6371                          ;*
6372                          ;*              177777
6373                          ;*              000000
6374                          ;*              177777
6375                          ;*
```

```
                                        ;*      FOLLOW THAT BY A SECTOR PULSE AND ONE THREE WORD
                                        ;*      HEADER CONSISTING OF THE FOLLOWING DATA:
                                        ;*
                                        ;*              000000
                                        ;*              177777
                                        ;*              000000
                                        ;*
                                        ;*      SIMULATE AN INDEX PULSE AND MAKE SURE READY COMES UP.
                                        ;*      CHECK FOR CORRECT WRITE ENCODED DATA AND PRECOMENSATION.
                                        ;*
                                        ;;*****************************************************************
037212  000004                  T$T45:  SCOPE
037214  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
037222  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
037226  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
037234  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 TO DIAGNOSTIC MODE
037242  012762  066702  000004          MOV     #HEAD1,RKBA(R2) ;ISSUE WRITE HEADER
037250  012703  066702                  MOV     #HEAD1,R3
037254  012762  177772  000002          MOV     #-6,RKWC(R2)
037262  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
037270  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
                                                                ;  READY FOR INDEX PULSE
037274  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
037302  012762  000040  000026          MOV     #DMD,RKMR1(R2)
037310  005300                          DEC     R0
037312  001370                          BNE     1$
037314  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
037320  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
037326  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
037334  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
037342  005300                          DEC     R0
037344  001370                          BNE     2$
037346  005037  003626                  CLR     SECCNT          ;CLEAR SECTOR COUNT
037352  012762  000040  000026          MOV     #DMD,RKMR1(R2)
037360  012705  000002                  MOV     #2,R5           ;LOAD NUMBER OF HEADERS
037364  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
037370  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
037376  012762  000040  000026          MOV     #DMD,RKMR1(R2)
037404  005300                          DEC     R0
037406  001370                          BNE     3$
037410  012762  000140  000026  4$:     MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
037416  012762  000040  000026          MOV     #DMD,RKMR1(R2)
037424  012737  062766  003170          MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
037432  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
                                                                ;  MAINT REG 1
037440  005037  003612                  CLR     P1.BIT
037444  005037  003614                  CLR     PR.BIT
037450  005037  003616                  CLR     M1.BIT
037454  005037  003620                  CLR     M2.BIT
037460  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
037464  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
037470  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
037474  104170                          ERROR   170             ;DATA INCORRECT
037476  005237  003622                  INC     BITCNT
037502  005300                          DEC     R0              ;CHECK IF READY FOR DATA
037504  001371                          BNE     5$              ;NO,GENERATE NEXT BIT
```

```
6432  037506  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
6433  037514  004737  045136                  JSR     PC,WRTBIT
6434  037520  104170                          ERROR   170             ;DATA INCORRECT
6435  037522  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6436  037526  012737  063032  003170          MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
6437  037534  012700  000003                  MOV     #3,R0           ;LOAD NUMBER OF WORDS IN HEADER
6438  037540  012304                  10$:    MOV     (R3)+,R4        ;GET NEXT WORD
6439  037542  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT
6440  037546  013737  003616  003620  12$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6441  037554  013737  003614  003616          MOV     PR.BIT,M1.BIT
6442  037562  013737  003612  003614          MOV     P1.BIT,PR.BIT
6443  037570  006004                          ROR     R4              ;SHIFT IN NEXT BIT
6444  037572  103403                          BCS     14$             ;CHECK IF ONE
6445  037574  005037  003612                  CLR     P1.BIT          ;ZERO
6446  037600  000403                          BR      15$             ;CLOCK IN BIT
6447
6448  037602  012737  000001  003612  14$:    MOV     #1,P1.BIT       ;ONE
6449  037610  004737  045136          15$:    JSR     PC,WRTBIT       ;WRITE BIT
6450  037614  104170                          ERROR   170             ;BIT INCORRECT
6451  037616  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6452  037622  005301                          DEC     R1              ;CHECK IF WORD FINISHED
6453  037624  001350                          BNE     12$             ;NO, CONTINUE
6454  037626  005300                          DEC     R0              ;CHECK IF HEADER COMPLETE
6455  037630  001343                          BNE     10$             ;NO, GET NEXT WORD
6456  037632  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
6457  037636  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6458  037644  013737  003614  003616          MOV     PR.BIT,M1.BIT
6459  037652  013737  003612  003614          MOV     P1.BIT,PR.BIT
6460  037660  005037  003612                  CLR     P1.BIT
6461  037664  004737  045136                  JSR     PC,WRTBIT       ;WRITE ZERO
6462  037670  104170                          ERROR   170             ;BIT INCORRECT
6463  037672  005237  003622                  INC     BITCNT          ;INCREMENT
6464  037676  005301                          DEC     R1              ;CHECK IF READY FOR NEXT HEADER
6465  037700  001356                          BNE     18$             ;NO, CONTINUE
6466  037702  005237  003626                  INC     SECCNT          ;INCREMENT
6467  037706  005305                          DEC     R5              ;CHECK IF SECOND HEADER WRITTEN
6468  037710  001237                          BNE     4$              ;NO, DO SECOND HEADER
6469  037712  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2) ;SIMULATE INDEX PULSE
6470  037720  012700  000004                  MOV     #4,R0
6471  037724  012762  000640  000026  20$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
6472  037732  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6473  037740  005300                          DEC     R0
6474  037742  001370                          BNE     20$
6475  037744  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6476  037752  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
6477  037760  012737  022040  003524          MOV     #MEWD!ECCW!DMD,E.MR1  ;LOAD EXPECTED MR1
6478  037766  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET
6479  037774  001401                          BEQ     25$             ;YES, CHECK IF READY SET
6480  037776  104173                          ERROR   173             ;MAINT REG 1 INCORRECT
6481  040000  012700  000010          25$:    MOV     #8.,R0          ;FINISH COMMAND
6482  040004  012762  000440  000026  26$:    MOV     #DMD!MCLK,RKMR1(R2)
6483  040012  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6484  040020  005300                          DEC     R0
6485  040022  001370                          BNE     26$
6486  040024  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
6487  040032  012737  000226  003500          MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
```

```
6488  040040  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
6489  040046  001401                          BEQ     TST46           ;;YES, GO ON TO NEXT TEST
6490  040050  104174                          ERROR   174
6491
6492                                  ;;*****************************************************************
6493                                  ;*TEST 46       DATA FIELD FILLING ON WRITE HEADER
6494                                  ;*
6495                                  ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
6496                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
6497                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0, AND
6498                                  ;*      SPECIFY TWO 3 WORD HEADERS CONSISTING OF THE
6499                                  ;*      FOLLOWING DATA:
6500                                  ;*
6501                                  ;*              125252
6502                                  ;*              052525
6503                                  ;*              125252
6504                                  ;*              052525
6505                                  ;*              125252
6506                                  ;*              052525
6507                                  ;*
6508                                  ;*      MAKE SURE THE DATA SYNCH ANY OTHER BITS OF DATA AND
6509                                  ;*      ECC FIELD ARE WRITTEN CORRECTLY.
6510                                  ;*
6511                                  ;;*****************************************************************
6512  040052  000004                  TST46:  SCOPE
6513  040054  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
6514  040062  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
6515  040066  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
6516  040074  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
6517  040102  012762  066716  000004          MOV     #HEAD3,RKBA(R2) ;ISSUE READ HEADER
6518  040110  012703  066716                  MOV     #HEAD3,R3
6519  040114  012762  177772  000002          MOV     #-2*3,RKWC(R2)
6520  040122  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2)
6521  040130  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE CLOCKS UNTIL READY
6522                                                                  ;  FOR INDEX PULSE
6523  040134  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
6524  040142  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6525  040150  005300                          DEC     R0
6526  040152  001370                          BNE     1$
6527  040154  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
6528  040160  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6529  040166  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
6530  040174  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6531  040202  005300                          DEC     R0
6532  040204  001370                          BNE     2$
6533  040206  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6534  040214  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
6535  040220  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
6536  040226  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6537  040234  005300                          DEC     R0
6538  040236  001370                          BNE     3$
6539  040240  005037  003626                  CLR     SECCNT          ;CLEAR SECTOR COUNT
6540  040244  012705  000002                  MOV     #2,R5           ;LOAD NUMBER OF HEADERS
6541  040250  012762  000140  000026  4$:     MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
6542  040256  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6543  040264  012737  062766  003170          MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
```

```
6544  040272  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
6545                                                    ;  MAINT REG 1
6546  040300  005037  003612                  CLR     P1.BIT
6547  040304  005037  003614                  CLR     PR.BIT
6548  040310  005037  003616                  CLR     M1.BIT
6549  040314  005037  003620                  CLR     M2.BIT
6550  040320  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
6551  040324  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6552  040330  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
6553  040334  104170                  ERROR   170                     ;DATA INCORRECT
6554  040336  005237  003622                  INC     BITCNT
6555  040342  005300                  DEC     R0                      ;CHECK IF READY FOR DATA
6556  040344  001371                  BNE     5$                      ;NO, GENERATE NEXT BIT
6557  040346  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
6558  040354  004737  045136                  JSR     PC,WRTBIT
6559  040360  104170                  ERROR   170                     ;DATA INCORRECT
6560  040362  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6561  040366  012737  063032  003170          MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
6562  040374  012700  000003                  MOV     #3,R0           ;LOAD NUMBER OF WORDS IN HEADER
6563  040400  012304          10$:    MOV     (R3)+,R4        ;GET NEXT WORD
6564  040402  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT
6565  040406  013737  003616  003620  12$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6566  040414  013737  003614  003616          MOV     PR.BIT,M1.BIT
6567  040422  013737  003612  003614          MOV     P1.BIT,PR.BIT
6568  040430  006004                  ROR     R4                      ;SHIFT IN NEXT BIT
6569  040432  103403                  BCS     14$                     ;CHECK IF ONE
6570  040434  005037  003612                  CLR     P1.BIT          ;ZERO
6571  040440  000403                  BR      15$                     ;CLOCK IN BIT
6572
6573  040442  012737  000001  003612  14$:    MOV     #1,P1.BIT       ;ONE
6574  040450  004737  045136          15$:    JSR     PC,WRTBIT       ;WRITE BIT
6575  040454  104170                  ERROR   170                     ;BIT INCORRECT
6576  040456  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6577  040462  005301                  DEC     R1                      ;CHECK IF WORD FINISHED
6578  040464  001350                  BNE     12$                     ;NO, CONTINUE
6579  040466  005300                  DEC     R0                      ;CHECK IF HEADER COMPLETE
6580  040470  001343                  BNE     10$                     ;NO, GET NEXT WORD
6581  040472  012737  063264  003170          MOV     #EM237,EMW      ;LOAD ERROR MESSAGE
6582  040500  012701  000477                  MOV     #64.+255.,R1    ;LOAD COUNT
6583  040504  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6584  040512  013737  003614  003616          MOV     PR.BIT,M1.BIT
6585  040520  013737  003612  003614          MOV     P1.BIT,PR.BIT
6586  040526  005037  003612                  CLR     P1.BIT
6587  040532  004737  045136                  JSR     PC,WRTBIT       ;WRITE ZERO
6588  040536  104170                  ERROR   170                     ;BIT INCORRECT
6589  040540  005301                  DEC     R1                      ;CHECK IF READY FOR DATA
6590  040542  001360                  BNE     18$                     ;NO, CONTINUE
6591  040544  012737  000001  003612          MOV     #1,P1.BIT       ;SET SYNCH BIT
6592  040552  004737  045136                  JSR     PC,WRTBIT       ;WRITE SYNCH BIT
6593  040556  104170                  ERROR   170                     ;SYNCH BIT INCORRECT
6594  040560  012737  063332  003170          MOV     #EM238,EMW      ;LOAD ERROR MESSAGE
6595  040566  005037  003622                  CLR     BITCNT          ;CLEAR BIT COUNT
6596  040572  012701  007777                  MOV     #256.*16.-1,R1  ;LOAD COUNT
6597  040576  013737  003616  003620  20$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6598  040604  013737  003614  003616          MOV     PR.BIT,M1.BIT
6599  040612  013737  003612  003614          MOV     P1.BIT,PR.BIT
```

```
6600  040620  005037  003612                    CLR     P1.BIT
6601  040624  004737  045136                    JSR     PC,WRTBIT       ;WRITE ZEROS
6602  040630  104170                            ERROR   170             ;BIT INCORRECT
6603  040632  005237  003622                    INC     BITCNT          ;INCREMENT BIT COUNT
6604  040636  005301                            DEC     R1              ;CHECK IF READY FOR ECC
6605  040640  001356                            BNE     20$             ;NO, CONTINUE
6606  040642  012701  000040                    MOV     #32.,R1         ;LOAD COUNT
6607  040646  042737  020000  003524            BIC     #ECCW,E.MR1     ;RESET ECCW BIT
6608  040654  004737  045136          21$:      JSR     PC,WRTBIT       ;WRITE ECC FIELD
6609  040660  104170                            ERROR   170             ;BIT INCORRECT
6610  040662  005237  003622                    INC     BITCNT          ;INCREMENT COUNT
6611  040666  005301                            DEC     R1              ;CHECK IF READY FOR POST AMBLE
6612  040670  001371                            BNE     21$             ;NO CONTINUE
6613  040672  052737  020000  003524            BIS     #ECCW,E.MR1     ;SET ECCW
6614  040700  012701  000012                    MOV     #10.,R1         ;CHECK IF READY FOR NEXT HEADER
6615  040704  004737  045136          22$:      JSR     PC,WRTBIT       ;WRITE POSTAMBLE
6616  040710  104170                            ERROR   170             ;BIT INCORRECT
6617  040712  005237  003622                    INC     BITCNT          ;INCREMENT COUNT
6618  040716  005301                            DEC     R1              ;CHECK IF READY FOR NEXT HEADER
6619  040720  001371                            BNE     22$             ;NO, COMPLETE POSTAMBLE
6620  040722  005237  003626                    INC     SECCNT          ;INCREMENT COUNT
6621  040726  005305                            DEC     R5              ;CHECK IF ALL HEADERS RECEIVED
6622  040730  001402                            BEQ     23$             ;YES, SIMULATE INDEX
6623  040732  000137  040250                    JMP     4$              ;NO GET NEXT HEADER
6624
6625  040736  012762  000240  000026  23$:      MOV     #DMD!MIND,RKMR1(R2) ;SIMULATE INDEX PULSE
6626  040744  012700  000004                    MOV     #4,R0
6627  040750  012762  000640  000026  25$:      MOV     #DMD!MIND!MCLK,RKMR1(R2)
6628  040756  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)
6629  040764  005300                            DEC     R0
6630  040766  001370                            BNE     25$
6631  040770  012762  000040  000026            MOV     #DMD,RKMR1(R2)
6632  040776  016237  000026  003464            MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
6633  041004  012737  022040  003524            MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
6634  041012  023737  003524  003464            CMP     E.MR1,T.MR1     ;CHECK IF MR1 CORRECT
6635                                                                    ;  (WRITE GATE RESET)
6636  041020  001401                            BEQ     28$             ;YES, CHECK IF READY SET
6637  041022  104173                            ERROR   173             ;MAINTENANCE REG 1 INCORRECT
6638  041024  012700  000010          28$:      MOV     #8.,R0          ;FINISH COMMAND
6639  041030  012762  000440  000026  29$:      MOV     #DMD!MCLK,RKMR1(R2)
6640  041036  012762  000040  000026            MOV     #DMD,RKMR1(R2)
6641  041044  005300                            DEC     R0
6642  041046  001370                            BNE     29$
6643  041050  016237  000000  003440            MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
6644  041056  012737  000226  003500            MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
6645  041064  023737  003500  003440            CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
6646  041072  001401                            BEQ     TST47           ;;YES, GO TO NEXT TEST
6647  041074  104174                            ERROR   174             ;CS1 INCORRECT
6648
6649                                  ;;*****************************************************************
6650                                  ;*TEST 47       WRITE HEADER FOR 26 SECTORS
6651                                  ;*
6652                                  ;*      CLEAR THE RK611 WITH A CONTROLLER CLEAR.  PUT CONTROLLER
6653                                  ;*      IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO AN RK06
6654                                  ;*      IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, DRIVE 0, SPECIFYING
6655                                  ;*      66 WORDS.  MAKE SURE ALL 26 SECTORS ARE WRITTEN CORRECTLY.
```

```
6656                                    ;*
6657                                    ;;*****************************************************************
6658  041076  000004                    TST47:  SCOPE
6659  041100  012737  000010  001200            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
6660  041106  013702  001270                    MOV     $BASE,R2        ;LOAD RK611 BASE
6661  041112  012762  100000  000000            MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
6662  041120  012762  000040  000026            MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN DIAGNOSTIC MODE
6663  041126  012762  066762  000004            MOV     #NPRBUF,RKBA(R2)        ;ISSUE READ HEADER
6664  041134  012703  066762                    MOV     #NPRBUF,R3
6665  041140  012762  177676  000002            MOV     #-22.*3,RKWC(R2)
6666  041146  012762  000027  000000            MOV     #WRHEAD,RKCS1(R2)
6667  041154  012700  000312                    MOV     #50.*4+2,R0     ;ISSUE CLOCKS UNTIL READY
6668                                                                    ;  FOR INDEX PULSE
6669  041160  012762  000440  000026  1$:       MOV     #DMD!MCLK,RKMR1(R2)
6670  041166  012762  000040  000026            MOV     #DMD,RKMR1(R2)
6671  041174  005300                            DEC     R0
6672  041176  001370                            BNE     1$
6673  041200  012700  000004                    MOV     #4,R0           ;ISSUE INDEX PULSE
6674  041204  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)
6675  041212  012762  000640  000026  2$:       MOV     #DMD!MIND!MCLK,RKMR1(R2)
6676  041220  012762  000240  000026            MOV     #DMD!MIND,RKMR1(R2)
6677  041226  005300                            DEC     R0
6678  041230  001370                            BNE     2$
6679  041232  012762  000040  000026            MOV     #DMD,RKMR1(R2)
6680  041240  012700  000010                    MOV     #8.,R0          ;WAIT FOR WRITE GATE
6681  041244  012762  000440  000026  3$:       MOV     #DMD!MCLK,RKMR1(R2)
6682  041252  012762  000040  000026            MOV     #DMD,RKMR1(R2)
6683  041260  005300                            DEC     R0
6684  041262  001370                            BNE     3$
6685  041264  005037  003626                    CLR     SECCNT          ;CLEAR SECTOR COUNT
6686  041270  012705  000026                    MOV     #22.,R5         ;LOAD NUMBER OF HEADERS
6687  041274  012762  000140  000026  4$:       MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
6688  041302  012762  000040  000026            MOV     #DMD,RKMR1(R2)
6689  041310  012737  062766  003170            MOV     #EM233,EMW      ;LOAD ERROR MESSAGE
6690  041316  012737  062040  003524            MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
6691                                                                    ;  MAINT REG 1
6692  041324  005037  003612                    CLR     P1.BIT
6693  041330  005037  003614                    CLR     PR.BIT
6694  041334  005037  003616                    CLR     M1.BIT
6695  041340  005037  003620                    CLR     M2.BIT
6696  041344  012700  000400                    MOV     #256.,R0        ;SIMULATE SYNCH
6697  041350  005037  003622                    CLR     BITCNT          ;INITIALIZE BIT COUNT
6698  041354  004737  045136            5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
6699  041360  104170                            ERROR   170             ;DATA INCORRECT
6700  041362  005237  003622                    INC     BITCNT
6701  041366  005300                            DEC     R0              ;CHECK IF READY FOR DATA
6702  041370  001371                            BNE     5$              ;NO, GENERATE NEXT BIT
6703  041372  012737  000001  003612            MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
6704  041400  004737  045136                    JSR     PC,WRTBIT
6705  041404  104170                            ERROR   170             ;DATA INCORRECT
6706  041406  005037  003622                    CLR     BITCNT          ;INITIALIZE BIT COUNT
6707  041412  012737  063032  003170            MOV     #EM234,EMW      ;LOAD ERROR MESSAGE
6708  041420  012700  000003                    MOV     #3,R0           ;LOAD NUMBER OF WORDS IN HEADER
6709  041424  012304                    10$:    MOV     (R3)+,R4        ;GET NEXT WORD
6710  041426  012701  000020                    MOV     #16.,R1         ;LOAD BIT COUNT
6711  041432  013737  003616  003620  12$:     MOV     M1.BIT,M2.BIT   ;SHIFT BITS
```

```
 6712  041440  013737  003614  003616          MOV     PR.BIT,M1.BIT
 6713  041446  013737  003612  003614          MOV     P1.BIT,PR.BIT
 6714  041454  006004                          ROR     R4              ;SHIFT IN NEXT BIT
 6715  041456  103403                          BCS     14$             ;CHECK IF ONE
 6716  041460  005037  003612                  CLR     P1.BIT          ;ZERO
 6717  041464  000403                          BR      15$             ;CLOCK IN BIT
 6718
 6719  041466  012737  000001  003612  14$:    MOV     #1,P1.BIT       ;ONE
 6720  041474  004737  045136          15$:    JSR     PC,WRTBIT       ;WRITE BIT
 6721  041500  104170                          ERROR   170             ;BIT INCORRECT
 6722  041502  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
 6723  041506  005301                          DEC     R1              ;CHECK IF WORD FINISHED
 6724  041510  001350                          BNE     12$             ;NO, CONTINUE
 6725  041512  005300                          DEC     R0              ;CHECK IF HEADER COMPLETE
 6726  041514  001343                          BNE     10$             ;NO, GET NEXT WORD
 6727  041516  012737  063264  003170          MOV     #EM237,EMW      ;LOAD ERROR MESSAGE
 6728  041524  012701  000477                  MOV     #64.+255.,R1    ;LOAD COUNT
 6729  041530  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
 6730  041536  013737  003614  003616          MOV     PR.BIT,M1.BIT
 6731  041544  013737  003612  003614          MOV     P1.BIT,PR.BIT
 6732  041552  005037  003612                  CLR     P1.BIT
 6733  041556  004737  045136                  JSR     PC,WRTBIT       ;WRITE ZERO
 6734  041562  104170                          ERROR   170             ;BIT INCORRECT
 6735  041564  005301                          DEC     R1              ;CHECK IF READY FOR DATA
 6736  041566  001360                          BNE     18$             ;NO, CONTINUE
 6737  041570  012737  000001  003612          MOV     #1,P1.BIT       ;SET SYNCH BIT
 6738  041576  004737  045136                  JSR     PC,WRTBIT       ;WRITE SYNCH BIT
 6739  041602  104170                          ERROR   170             ;SYNCH BIT INCORRECT
 6740  041604  012737  063332  003170          MOV     #EM238,EMW      ;LOAD ERROR MESSAGE
 6741  041612  005037  003622                  CLR     BITCNT          ;CLEAR BIT COUNT
 6742  041616  012701  007777                  MOV     #256.*16.-1,R1  ;LOAD COUNT
 6743  041622  013737  003616  003620  20$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
 6744  041630  013737  003614  003616          MOV     PR.BIT,M1.BIT
 6745  041636  013737  003612  003614          MOV     P1.BIT,PR.BIT
 6746  041644  005037  003612                  CLR     P1.BIT
 6747  041650  004737  045136                  JSR     PC,WRTBIT       ;WRITE ZEROS
 6748  041654  104170                          ERROR   170             ;BIT INCORRECT
 6749  041656  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
 6750  041662  005301                          DEC     R1              ;CHECK IF READY FOR ECC
 6751  041664  001356                          BNE     20$             ;NO, CONTINUE
 6752  041666  012701  000040                  MOV     #32.,R1         ;LOAD COUNT
 6753  041672  042737  020000  003524          BIC     #ECCW,E.MR1     ;RESET ECCW BIT
 6754  041700  004737  045136          21$:    JSR     PC,WRTBIT       ;WRITE ECC FIELD
 6755  041704  104170                          ERROR   170             ;BIT INCORRECT
 6756  041706  005237  003622                  INC     BITCNT          ;INCREMENT COUNT
 6757  041712  005301                          DEC     R1              ;CHECK IF READY FOR POST AMBLE
 6758  041714  001371                          BNE     21$             ;NO CONTINUE
 6759  041716  052737  020000  003524          BIS     #ECCW,E.MR1     ;SET ECCW
 6760  041724  012701  000012                  MOV     #10.,R1         ;CHECK IF READY FOR NEXT HEADER
 6761  041730  004737  045136          22$:    JSR     PC,WRTBIT       ;WRITE POSTAMBLE
 6762  041734  104170                          ERROR   170             ;BIT INCORRECT
 6763  041736  005237  003622                  INC     BITCNT          ;INCREMENT COUNT
 6764  041742  005301                          DEC     R1              ;CHECK IF READY FOR NEXT HEADER
 6765  041744  001371                          BNE     22$             ;NO, COMPLETE POSTAMBLE
 6766  041746  005237  003626                  INC     SECCNT          ;INCREMENT COUNT
 6767  041752  005305                          DEC     R5              ;CHECK IF ALL HEADERS RECEIVED
```

```
 6768  041754  001402                          BEQ     23$             ;YES, SIMULATE INDEX
 6769  041756  000137  041274                  JMP     4$              ;NO GET NEXT HEADER
 6770
 6771  041762  012762  000240  000026  23$:    MOV     #DMD!MIND,RKMR1(R2) ;SIMULATE INDEX PULSE
 6772  041770  012700  000004                  MOV     #4,R0
 6773  041774  012762  000640  000026  25$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
 6774  042002  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
 6775  042010  005300                          DEC     R0
 6776  042012  001370                          BNE     25$
 6777  042014  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 6778  042022  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
 6779  042030  012737  022040  003524          MOV     #MEWD!ECCW!DMD,E.MR1 ;LOAD EXPECTED MR1
 6780  042036  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK IF MR1 CORRECT
 6781                                                                  ;  (WRITE GATE RESET)
 6782  042044  001401                          BEQ     28$             ;YES, CHECK IF READY SET
 6783  042046  104173                          ERROR   173             ;MAINTENANCE REG 1 INCORRECT
 6784  042050  012700  000010          28$:    MOV     #8.,R0          ;FINISH COMMAND
 6785  042054  012762  000440  000026  29$:    MOV     #DMD!MCLK,RKMR1(R2)
 6786  042062  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 6787  042070  005300                          DEC     R0
 6788  042072  001370                          BNE     29$
 6789  042074  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
 6790  042102  012737  000226  003500          MOV     #RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
 6791  042110  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
 6792  042116  001401                          BEQ     TST50           ;YES, GO TO NEXT TEST
 6793  042120  104174                          ERROR   174             ;CS1 INCORRECT
 6794
 6795                                  ;;*****************************************************************
 6796                                  ;*TEST 50       WRITE HEADER IN 24 SECTOR FORMAT
 6797                                  ;*
 6798                                  ;*      CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.  PUT
 6799                                  ;*      THE CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER
 6800                                  ;*      OF SIX WORDS IN 24 SECTOR FORMAT TO AN RK06, CYLINDER 0,
 6801                                  ;*      HEAD 0, DRIVE 0.  CLOCK THROUGH THE SEEK AND DRIVE CLEAR
 6802                                  ;*      MESSAGES, SIMULATE INDEX PULSE, SECTOR PULSE, 3 HEADER
 6803                                  ;*      WORDS, SYNCH AND DATA, ANOTHER SECTOR PULSE, 3 HEADER
 6804                                  ;*      WORDS, AND AN INDEX PULSE.  CHECK DATA WRITTEN TO MAKE
 6805                                  ;*      SURE ONLY LOW 16 BITS OF SILO ARE USED.
 6806                                  ;*
 6807                                  ;;*****************************************************************
 6808  042122  000004                  TST50:  SCOPE
 6809  042124  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
 6810  042132  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
 6811  042136  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
 6812  042144  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 TO DIAGNOSTIC MODE
 6813  042152  012762  066702  000004          MOV     #HEAD1,RKBA(R2) ;ISSUE WRITE HEADER
 6814  042160  012703  066702                  MOV     #HEAD1,R3
 6815  042164  012762  177772  000002          MOV     #-6,RKWC(R2)
 6816  042172  012762  010027  000000          MOV     #CFMT!WRHEAD,RKCS1(R2)
 6817  042200  012700  000312                  MOV     #50.*4+2,R0     ;ISSUE ENOUGH CLOCKS UNTIL
 6818                                                                  ;  READY FOR INDEX PULSE
 6819  042204  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
 6820  042212  012762  000040  000026          MOV     #DMD,RKMR1(R2)
 6821  042220  005300                          DEC     R0
 6822  042222  001370                          BNE     1$
 6823  042224  012700  000004                  MOV     #4,R0           ;ISSUE INDEX PULSE
```

```
6824  042230  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6825  042236  012762  000640  000026  2$:     MOV     #DMD!MIND!MCLK,RKMR1(R2)
6826  042244  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6827  042252  005300                          DEC     R0
6828  042254  001370                          BNE     2$
6829  042256  005037  003626                  CLR     SECCNT          ;CLEAR SECTOR COUNT
6830  042262  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6831  042270  012705  000002                  MOV     #2,R5           ;LOAD NUMBER OF HEADERS
6832  042274  012700  000010                  MOV     #8.,R0          ;WAIT FOR WRITE GATE
6833  042300  012762  000440  000026  3$:     MOV     #DMD!MCLK,RKMR1(R2)
6834  042306  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6835  042314  005300                          DEC     R0
6836  042316  001370                          BNE     3$
6837  042320  012762  000140  000026  4$:     MOV     #DMD!MSP,RKMR1(R2) ;SIMULATE SECTOR PULSE
6838  042326  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6839  042334  012737  063371  003170          MOV     #EM239,EMW      ;LOAD ERROR MESSAGE
6840  042342  012737  062040  003524          MOV     #DMD!MEWD!ECCW!WRTGAT,E.MR1 ;INITIALIZE EXPECTED
6841                                                                  ;  MAINT REG 1
6842  042350  005037  003612                  CLR     P1.BIT
6843  042354  005037  003614                  CLR     PR.BIT
6844  042360  005037  003616                  CLR     M1.BIT
6845  042364  005037  003620                  CLR     M2.BIT
6846  042370  012700  000400                  MOV     #256.,R0        ;SIMULATE SYNCH
6847  042374  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6848  042400  004737  045136          5$:     JSR     PC,WRTBIT       ;WRITE ONE BIT
6849  042404  104170                          ERROR   170             ;DATA INCORRECT
6850  042406  005237  003622                  INC     BITCNT
6851  042412  005300                          DEC     R0              ;CHECK IF READY FOR DATA
6852  042414  001371                          BNE     5$              ;NO,GENERATE NEXT BIT
6853  042416  012737  000001  003612          MOV     #1,P1.BIT       ;PUT IN SYNCH BIT
6854  042424  004737  045136                  JSR     PC,WRTBIT
6855  042430  104170                          ERROR   170             ;DATA INCORRECT
6856  042432  005037  003622                  CLR     BITCNT          ;INITIALIZE BIT COUNT
6857  042436  012737  063463  003170          MOV     #EM240,EMW      ;LOAD ERROR MESSAGE
6858  042444  012700  000003                  MOV     #3,R0           ;LOAD NUMBER OF WORDS IN HEADER
6859  042450  012304                  10$:    MOV     (R3)+,R4        ;GET NEXT WORD
6860  042452  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT
6861  042456  013737  003616  003620  12$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6862  042464  013737  003614  003616          MOV     PR.BIT,M1.BIT
6863  042472  013737  003612  003614          MOV     P1.BIT,PR.BIT
6864  042500  006004                          ROR     R4              ;SHIFT IN NEXT BIT
6865  042502  103403                          BCS     14$             ;CHECK IF ONE
6866  042504  005037  003612                  CLR     P1.BIT          ;ZERO
6867  042510  000403                          BR      15$             ;CLOCK IN BIT
6868
6869  042512  012737  000001  003612  14$:    MOV     #1,P1.BIT       ;ONE
6870  042520  004737  045136          15$:    JSR     PC,WRTBIT       ;WRITE BIT
6871  042524  104170                          ERROR   170             ;BIT INCORRECT
6872  042526  005237  003622                  INC     BITCNT          ;INCREMENT BIT COUNT
6873  042532  005301                          DEC     R1              ;CHECK IF WORD FINISHED
6874  042534  001350                          BNE     12$             ;NO, CONTINUE
6875  042536  005300                          DEC     R0              ;CHECK IF HEADER COMPLETE
6876  042540  001343                          BNE     10$             ;NO, GET NEXT WORD
6877  042542  012701  000020                  MOV     #16.,R1         ;LOAD BIT COUNT FOR NEXT WORD
6878  042546  013737  003616  003620  18$:    MOV     M1.BIT,M2.BIT   ;SHIFT BITS
6879  042554  013737  003614  003616          MOV     PR.BIT,M1.BIT
```

```
6880  042562  013737  003612  003614          MOV     P1.BIT,PR.BIT
6881  042570  005037  003612                  CLR     P1.BIT
6882  042574  004737  045136                  ISR     PC,WRTBIT       ;WRITE ZERO
6883  042600  104170                          ERROR   170             ;BIT INCORRECT
6884  042602  005237  003622                  INC     BITCNT          ;INCREMENT
6885  042606  005301                          DEC     R1              ;CHECK IF READY FOR NEXT HEADER
6886  042610  001356                          BNE     18$             ;NO, CONTINUE
6887  042612  005237  003626                  INC     SECCNT          ;INCREMENT
6888  042616  005305                          DEC     R5              ;CHECK IF SECOND HEADER WRITTEN
6889  042620  001237                          BNE     4$              ;NO, DO SECOND HEADER
6890  042622  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2) ;SIMULATE INDEX PULSE
6891  042630  012700  000004                  MOV     #4,R0
6892  042634  012762  000640  000026  20$:    MOV     #DMD!MIND!MCLK,RKMR1(R2)
6893  042642  012762  000240  000026          MOV     #DMD!MIND,RKMR1(R2)
6894  042650  005300                          DEC     R0
6895  042652  001370                          BNE     20$
6896  042654  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6897  042662  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;GET MAINT REG 1
6898  042670  012737  022040  003524          MOV     #MEWD!ECCW!DMD,E.MR1  ;LOAD EXPECTED MR1
6899  042676  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK MR1 CORRECT (WRITE GATE RESET
6900  042704  001401                          BEQ     25$             ;YES, CHECK IF READY SET
6901  042706  104173                          ERROR   173             ;MAINT REG 1 INCORRECT
6902  042710  012700  000010          25$:    MOV     #8.,R0          ;FINISH COMMAND
6903  042714  012762  000440  000026  26$:    MOV     #DMD!MCLK,RKMR1(R2)
6904  042722  012762  000040  000026          MOV     #DMD,RKMR1(R2)
6905  042730  005300                          DEC     R0
6906  042732  001370                          BNE     26$
6907  042734  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REG 1
6908  042742  012737  010226  003500          MOV     #RDY!CFMT!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
6909  042750  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK IF CS1 CORRECT
6910  042756  001401                          BEQ     TST51           ;;YES, GO ON TO NEXT TEST
6911  042760  104174                          ERROR   174
6912
6913                                  .SBTTL  **TYPE B INSTRUCTION ERRORS
6914
6915                                  ;;*****************************************************************
6916                                  ;*TEST 51       FORMAT ERROR (PART 1)
6917                                  ;*
6918                                  ;*      CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT THE
6919                                  ;*      CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN
6920                                  ;*      RK06, IN 26 SECTOR FORMAT, CYLINDER 43, HEAD 0, DRIVE 0.
6921                                  ;*      CLOCK IN MAINTENANCE MODE UNTIL PHASE ADDRESS 6.  TURN OFF
6922                                  ;*      MAINTENANCE MODE.  MAKE SURE FORMAT ERROR,
6923                                  ;*      DRIVE AVAILABLE AND CONTROLLER ERROR SET.
6924                                  ;*
6925                                  ;;*****************************************************************
6926  042762  000004                  TST51:  SCOPE
6927  042764  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
6928  042772  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
6929  042776  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR RK611 SUBSYSTEM
6930  043004  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINTENANCE MODE
6931  043012  012762  000043  000020          MOV     #43,RKDCYL(R2) ;LOAD CYLINDER ADDRESS REG
6932  043020  012762  000025  000000          MOV     #RDHEAD,RKCS1(R2) ;ISSUE READ HEADER
6933  043026  012700  000132                  MOV     #22.*4+2,R0     ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
6934  043032  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
6935  043040  012762  000040  000026          MOV     #DMD,RKMR1(R2)
```

```
6936  043046  005300                          DEC     R0
6937  043050  001370                          BNE     1$
6938  043052  005062  000026                  CLR     RKMR1(R2)       ;FINISH COMMAND IN NORMAL MODE
6939  043056  013700  003632                  MOV     WAITIM,R0       ;WAIT FOR READY
6940  043062  105762  000000          2$:     TSTB    RKCS1(R2)
6941  043066  100402                          BMI     3$
6942  043070  005300                          DEC     R0
6943  043072  001373                          BNE     2$
6944  043074  016237  000000  003440  3$:     MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
6945  043102  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
6946  043110  016237  000012  003452          MOV     RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
6947  043116  016237  000014  003454          MOV     RKER(R2),T.ER   ;STORE ERROR REG
6948  043124  012737  100224  003500          MOV     #CERR!RDY!RDHEAD&^C<GO>,E.CS1 ;LOAD EXPECTED CS1
6949  043132  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
6950  043140  012737  100001  003512          MOV     #SVAL!DRA,E.DS ;LOAD EXPECTED DRIVE STATUS REG
6951  043146  012737  000020  003514          MOV     #FMTE,E.ER      ;LOAD EXPECTED ERROR REG
6952  043154  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG 1 CORRECT
6953  043162  001401                          BEQ     4$              ;YES, CONTINUE
6954  043164  104175                          ERROR   175
6955  043166  023737  003510  003450  4$:     CMP     E.CS2,T.CS2     ;CHECK COMMAND AND STATU REG 2 CORRECT
6956  043174  001401                          BEQ     5$              ;YES, CONTINUE
6957  043176  104176                          ERROR   176
6958  043200  023737  003512  003452  5$:     CMP     E.DS,T.DS       ;CHECK IF DRIVE STRATUS REG CORRECT
6959  043206  001401                          BEQ     6$              ;YES, CONTINUE
6960  043210  104177                          ERROR   177
6961  043212  023737  003514  003454  6$:     CMP     E.ER,T.ER       ;CHECK IF ERR REG CORRECT
6962  043220  001401                          BEQ     7$              ;YES, CONTINUE
6963  043222  104200                          ERROR   200
6964  043224  013737  003440  003540  7$:     MOV     T.CS1,P.CS1     ;STORE PREVIOUS CONTENTS OF
6965  043232  013737  003450  003550          MOV     T.CS2,P.CS2     ;   COMMAND AND STATUS REG 1
6966  043240  013737  003452  003552          MOV     T.DS,P.DS       ;   COMMAND AND STATUS REG 2
6967  043246  013737  003454  003554          MOV     T.ER,P.ER       ;   DRIVE STATUS REG
6968                                          ;   AND ERROR REG
6969  043254  012762  100000  000000          MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
6970  043262  016237  000000  003440          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
6971  043270  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
6972  043276  016237  000012  003452          MOV     RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
6973  043304  016237  000014  003454          MOV     RKER(R2),T.ER   ;STORE ERROR REG
6974  043312  012737  000200  003500          MOV     #RDY,E.CS1      ;LOAD EXPECTED CS1
6975  043320  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
6976  043326  005037  003512                  CLR     E.DS            ;LOAD EXPECTED DRIVE STATUS REG
6977  043332  005037  003514                  CLR     E.ER            ;LOAD EXPECTED ERROR REG
6978  043336  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
6979  043344  001401                          BEQ     11$             ;YES, CONTINUE
6980  043346  104211                          ERROR   211             ;CS1 INCORRECT
6981  043350  023737  003510  003450  11$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
6982  043356  001401                          BEQ     12$             ;YES, CONTINUE
6983  043360  104212                          ERROR   212             ;CS2 INCORRECT
6984  043362  023737  003512  003452  12$:    CMP     E.DS,T.DS       ;CHECK DRIVE STATUS CORRECT
6985  043370  001401                          BEQ     13$             ;YES, CONTINUE
6986  043372  104213                          ERROR   213             ;DRIVE STATUS REG INCORRECT
6987  043374  023737  003514  003454  13$:    CMP     E.ER,T.ER       ;CHECK IF ERROR REG CORRECT
6988  043402  001401                          BEQ     14$             ;YES, GO ON TO NEXT TEST
6989  043404  104214                          ERROR   214             ;ERROR REG INCORRECT
6990  043406                          14$:
6991
```

```
                                ;;*****************************************************************
                                ;*TEST 52       FORMAT ERROR (PART 2)
                                ;*
                                ;*      CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT THE
                                ;*      CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ HEADER TO AN
                                ;*      RK06, IN 24 SECTOR FORMAT, CYLINDER 3, HEAD 0, DRIVE 0.
                                ;*      CLOCK IN MAINTENANCE MODE UNTIL PHASE ADDRESS 6.  TURN OFF
                                ;*      MAINTENANCE MODE.  MAKE SURE FORMAT ERROR,
                                ;*      DRIVE AVAILABLE AND CONTROLLER ERROR SET.
                                ;*
                                ;;*****************************************************************
043406  000004                TST52:  SCOPE
043410  012737  000144  001200        MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
043416  013702  001270                MOV     $BASE,R2        ;LOAD RK611 BASE
043422  012762  000040  000010        MOV     #SCLR,RKCS2(R2) ;CLEAR RK611 SUBSYSTEM
043430  012762  000040  000026        MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINTENANCE MODE
043436  012762  000003  000020        MOV     #3,RKDCYL(R2) ;LOAD CYLINDER ADDRESS REG
043444  012762  010025  000000        MOV     #RDHEAD!CFMT,RKCS1(R2) ;ISSUE READ HEADER
043452  012700  000132                MOV     #22.*4+2,R0     ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
043456  012762  000440  000026  1$:   MOV     #DMD!MCLK,RKMR1(R2)
043464  012762  000040  000026        MOV     #DMD,RKMR1(R2)
043472  005300                        DEC     R0
043474  001370                        BNE     1$
043476  005062  000026                CLR     RKMR1(R2)       ;FINISH COMMAND IN NORMAL MODE
043502  013700  003632                MOV     WAITIM,R0       ;WAIT FOR READY
043506  105762  000000        2$:     TSTB    RKCS1(R2)
043512  100402                        BMI     3$
043514  005300                        DEC     R0
043516  001373                        BNE     2$
043520  016237  000000  003440  3$:   MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
043526  016237  000010  003450        MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
043534  016237  000012  003452        MOV     RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
043542  016237  000014  003454        MOV     RKER(R2),T.ER   ;STORE ERROR REG
043550  012737  110224  003500        MOV     #CERR!RDY!RDHEAD!CFMT&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
043556  012737  000100  003510        MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
043564  012737  100001  003512        MOV     #SVAL!DRA,E.DS ;LOAD EXPECTED DRIVE STATUS REG
043572  012737  000030  003514        MOV     #FMTE!DRPAR,E.ER        ;LOAD EXPECTED ERROR REG
043600  023737  003500  003440        CMP     E.CS1,T.CS1     ;CHECK COMMAND AND STATUS REG 1 CORRECT
043606  001401                        BEQ     4$              ;YES, CONTINUE
043610  104201                        ERROR   201
043612  023737  003510  003450  4$:   CMP     E.CS2,T.CS2     ;CHECK COMMAND AND STATU REG 2 CORRECT
043620  001401                        BEQ     5$              ;YES, CONTINUE
043622  104202                        ERROR   202
043624  023737  003512  003452  5$:   CMP     E.DS,T.DS       ;CHECK IF DRIVE STRATUS REG CORRECT
043632  001401                        BEQ     6$              ;YES, CONTINUE
043634  104203                        ERROR   203
043636  023737  003514  003454  6$:   CMP     E.ER,T.ER       ;CHECK IF ERR REG CORRECT
043644  001401                        BEQ     7$              ;YES, CONTINUE
043646  104204                        ERROR   204
043650  013737  003440  003540  7$:   MOV     T.CS1,P.CS1     ;STORE PREVIOUS CONTENTS OF
043656  013737  003450  003550        MOV     T.CS2,P.CS2     ;  COMMAND AND STATUS REG 1
043664  013737  003452  003552        MOV     T.DS,P.DS       ;  COMMAND AND STATUS REG 2
043672  013737  003454  003554        MOV     T.ER,P.ER       ;  DRIVE STATUS REG
                                                              ;  AND ERROR REG
043700  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR RK611
043706  016237  000000  003440        MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
```

```
7048  043714  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
7049  043722  016237  000012  003452          MOV     RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
7050  043730  016237  000014  003454          MOV     RKER(R2),T.ER   ;STORE ERROR REG
7051  043736  012737  000200  003500          MOV     #RDY,E.CS1      ;LOAD EXPECTED CS1
7052  043744  012737  000100  003510          MOV     #IR,E.CS2       ;LOAD EXPECTED CS2
7053  043752  005037  003512                  CLR     E.DS            ;LOAD EXPECTED DRIVE STATUS REG
7054  043756  005037  003514                  CLR     E.ER            ;LOAD EXPECTED ERROR REG
7055  043762  023737  003500  003440          CMP     E.CS1,T.CS1     ;CHECK CS1 CORRECT
7056  043770  001401                          BEQ     11$             ;YES, CONTINUE
7057  043772  104211                          ERROR   211             ;CS1 INCORRECT
7058  043774  023737  003510  003450  11$:    CMP     E.CS2,T.CS2     ;CHECK CS2 CORRECT
7059  044002  001401                          BEQ     12$             ;YES, CONTINUE
7060  044004  104212                          ERROR   212             ;CS2 INCORRECT
7061  044006  023737  003512  003452  12$:    CMP     E.DS,T.DS       ;CHECK DRIVE STATUS CORRECT
7062  044014  001401                          BEQ     13$             ;YES, CONTINUE
7063  044016  104213                          ERROR   213             ;DRIVE STATUS REG INCORRECT
7064  044020  023737  003514  003454  13$:    CMP     E.ER,T.ER       ;CHECK IF ERROR REG CORRECT
7065  044026  001401                          BEQ     14$             ;YES, GO ON TO NEXT TEST
7066  044030  104214                          ERROR   214             ;ERROR REG INCORRECT
7067  044032                          14$:
7068
7069                                  ;;*****************************************************************
7070                                  ;*TEST 53       FAULT SETTING CONTROLLER ERROR
7071                                  ;*
7072                                  ;*      CLEAR THE RK06 SUBSYSTEM WITH A SUBSYSTEM CLEAR.  PUT THE
7073                                  ;*      CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE HEADER TO
7074                                  ;*      AN RK06, IN 26 SECTOR FORMAT, CYLINDER 3, HEAD 0, DRIVE
7075                                  ;*      0.  CLOCK IN MAINTENANCE MODE UNTIL PHASE ADDRESS 6.
7076                                  ;*      TURN OFF MAINTENANCE MODE.  MAKE SURE DRIVE
7077                                  ;*      AVAILABLE AND CONTROLLER ERROR SET.
7078                                  ;*
7079                                  ;;*****************************************************************
7080  044032  000004                  TST53:  SCOPE
7081  044034  012737  000144  001200          MOV     #100.,$TIMES    ;;DO 100. ITERATIONS
7082  044042  013702  001270                  MOV     $BASE,R2        ;LOAD RK611 BASE
7083  044046  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR RK611 SUBSYSTEM
7084  044054  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;PUT RK611 IN MAINTENANCE MODE
7085  044062  012762  000003  000020          MOV     #3,RKDCYL(R2) ;LOAD CYLINDER ADDRESS REG
7086  044070  012762  177775  000002          MOV     #-3,RKWC(R2)    ;LOAD WORD COUNT
7087  044076  012762  067204  000004          MOV     #WRBUFF,RKBA(R2) ;LOAD BUS ADDRESS
7088  044104  012762  000027  000000          MOV     #WRHEAD,RKCS1(R2) ;ISSUE WRITE HEADER
7089  044112  012700  000132                  MOV     #22.*4+2,R0     ;ISSUE CLOCKS UNTIL PHASE ADDRESS 6
7090  044116  012762  000440  000026  1$:     MOV     #DMD!MCLK,RKMR1(R2)
7091  044124  012762  000040  000026          MOV     #DMD,RKMR1(R2)
7092  044132  005300                          DEC     R0
7093  044134  001370                          BNE     1$
7094  044136  005062  000026                  CLR     RKMR1(R2)       ;FINISH COMMAND IN NORMAL MODE
7095  044142  013700  003632                  MOV     WAITIM,R0       ;WAIT FOR READY
7096  044146  105762  000000          2$:     TSTB    RKCS1(R2)
7097  044152  100402                          BMI     3$
7098  044154  005300                          DEC     R0
7099  044156  001373                          BNE     2$
7100  044160  016237  000000  003440  3$:     MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REG 1
7101  044166  016237  000010  003450          MOV     RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG 2
7102  044174  016237  000012  003452          MOV     RKDS(R2),T.DS   ;STORE DRIVE STATUS REG
7103  044202  016237  000014  003454          MOV     RKER(R2),T.ER   ;STORE ERROR REG
```

```
7104  044210  012737  100226  003500          MOV   #CERR!RDY!WRHEAD&<^C<GO>>,E.CS1 ;LOAD EXPECTED CS1
7105  044216  012737  000100  003510          MOV   #IR,E.CS2        ;LOAD EXPECTED CS2
7106  044224  012737  100001  003512          MOV   #SVAL!DRA,E.DS ;LOAD EXPECTED DRIVE STATUS REG
7107  044232  012737  000000  003514          MOV   #0,E.ER ;LOAD EXPECTED ERROR REG
7108  044240  023737  003500  003440          CMP   E.CS1,T.CS1      ;CHECK COMMAND AND STATUS REG 1 CORRECT
7109  044246  001401                          BEQ   4$               ;YES, CONTINUE
7110  044250  104205                          ERROR 205
7111  044252  023737  003510  003450  4$:     CMP   E.CS2,T.CS2      ;CHECK COMMAND AND STATU REG 2 CORRECT
7112  044260  001401                          BEQ   5$               ;YES, CONTINUE
7113  044262  104206                          ERROR 206
7114  044264  023737  003512  003452  5$:     CMP   E.DS,T.DS        ;CHECK IF DRIVE STRATUS REG CORRECT
7115  044272  001401                          BEQ   6$               ;YES, CONTINUE
7116  044274  104207                          ERROR 207
7117  044276  023737  003514  003454  6$:     CMP   E.ER,T.ER        ;CHECK IF ERR REG CORRECT
7118  044304  001401                          BEQ   7$               ;YES, CONTINUE
7119  044306  104210                          ERROR 210
7120  044310  013737  003440  003540  7$:     MOV   T.CS1,P.CS1      ;STORE PREVIOUS CONTENTS OF
7121  044316  013737  003450  003550          MOV   T.CS2,P.CS2      ;   COMMAND AND STATUS REG 1
7122  044324  013737  003452  003552          MOV   T.DS,P.DS        ;   COMMAND AND STATUS REG 2
7123  044332  013737  003454  003554          MOV   T.ER,P.ER        ;   DRIVE STATUS REG
7124                                          ;   AND ERROR REG
7125  044340  012762  100000  000000          MOV   #CCLR,RKCS1(R2)  ;CLEAR RK611
7126  044346  016237  000000  003440          MOV   RKCS1(R2),T.CS1  ;STORE COMMAND AND STATUS REG 1
7127  044354  016237  000010  003450          MOV   RKCS2(R2),T.CS2  ;STORE COMMAND AND STATUS REG 2
7128  044362  016237  000012  003452          MOV   RKDS(R2),T.DS    ;STORE DRIVE STATUS REG
7129  044370  016237  000014  003454          MOV   RKER(R2),T.ER    ;STORE ERROR REG
7130  044376  012737  000200  003500          MOV   #RDY,E.CS1       ;LOAD EXPECTED CS1
7131  044404  012737  000100  003510          MOV   #IR,E.CS2        ;LOAD EXPECTED CS2
7132  044412  005037  003512                  CLR   E.DS             ;LOAD EXPECTED DRIVE STATUS REG
7133  044416  005037  003514                  CLR   E.ER             ;LOAD EXPECTED ERROR REG
7134  044422  023737  003500  003440          CMP   E.CS1,T.CS1      ;CHECK CS1 CORRECT
7135  044430  001401                          BEQ   11$              ;YES, CONTINUE
7136  044432  104211                          ERROR 211              ;CS1 INCORRECT
7137  044434  023737  003510  003450  11$:    CMP   E.CS2,T.CS2      ;CHECK CS2 CORRECT
7138  044442  001401                          BEQ   12$              ;YES, CONTINUE
7139  044444  104212                          ERROR 212              ;CS2 INCORRECT
7140  044446  023737  003512  003452  12$:    CMP   E.DS,T.DS        ;CHECK DRIVE STATUS CORRECT
7141  044454  001401                          BEQ   13$              ;YES, CONTINUE
7142  044456  104213                          ERROR 213              ;DRIVE STATUS REG INCORRECT
7143  044460  023737  003514  003454  13$:    CMP   E.ER,T.ER        ;CHECK IF ERROR REG CORRECT
7144  044466  001401                          BEQ   14$              ;YES, GO ON TO NEXT TEST
7145  044470  104214                          ERROR 214              ;ERROR REG INCORRECT
7146  044472                          14$:
7147
```

```
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
                                ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO NEWPAS

044472                          $EOP:
044472  000004                          SCOPE
044474  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
044500  005037  001200                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
044504  005237  001222                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
044510  042737  100000  001222          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
044516  005327                          DEC     (PC)+           ;;LOOP?
044520  000001                  $EOPCT: .WORD   1
044522  003063                          BGT     $DOAGN          ;;YES
044524  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
044526  000001                  $ENDCT: .WORD   1
044530  044520                          $EOPCT
044532  104401  044540                  TYPE    ,65$            ;;TYPE ASCIZ STRING
044536  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12><15>/END PASS #/
044556                          64$:
044556  013746  001222                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER
044562  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
044564  104401  044572                  TYPE    ,67$            ;;TYPE ASCIZ STRING
044570  000421                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
044634                          66$:
044634  013746  001112                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                ;;TOTAL NUMBER OF ERRORS
044640  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
044642  104401  001211                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
044646  005037  001112                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
044652  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
044656  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
044660  000005                          RESET                   ;;CLEAR THE WORLD
044662  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
044664  000240                          NOP                     ;;SAVE ROOM
044666  000240                          NOP                     ;;FOR
044670  000240                          NOP                     ;;ACT11
044672                          $DOAGN:
044672  000137                          JMP     @(PC)+          ;;RETURN
044674  004662                  $RTNAD: .WORD   NEWPAS
044676    377     377     000   $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
        044702                          .EVEN

                                .SBTTL  GENERATE BAD PARITY IN MEMORY

044702  012703  172100          WRTPAR: MOV     #MEMBAS,R3      ;GET BASE OF VECTOR AREA
044706  013704  003630                  MOV     MEMPAR,R4       ;STORE FLAGS
044712  012705  000020                  MOV     #16.,R5         ;GET NUMBER OF REGISTERS
044716  006004                  5$:     ROR     R4              ;CHECK IF PARITY ENABLE ON THIS BANK
```

```
7204  044720  103002                        BCC     7$              ;NO, TRY NEXT BANK
7205  044722  012713  000004                MOV     #WR.PAR,(R3)    ;ALLOW SETTING OF BAD PARITY
7206  044726  062703  000002        7$:     ADD     #2,R3           ;CALCULATE NEXT ADDRESS
7207  044732  005305                        DEC     R5              ;CHECK IF FINISHED
7208  044734  001370                        BNE     5$              ;NO, GET NEXT PARITY MEMORY ADDRESS
7209  044736  012737  000157  067220        MOV     #157,BADPAR     ;WRITE BAD PARITY
7210  044744  012703  172100                MOV     #MEMBAS,R3      ;GET BASE VECTOR ADDRESS
7211  044750  013704  003630                MOV     MEMPAR,R4       ;LOAD FLAGS
7212  044754  012705  000020                MOV     #16.,R5         ;GET NUMBER OF REGISTERS
7213  044760  006004                15$:    ROR     R4              ;CHECK IF PARITY ENABLE ON THIS BANK
7214  044762  103002                        BCC     17$             ;NO, CHECK NEXT BANK
7215  044764  012713  000001                MOV     #PAR.EN,(R3)    ;ALLOW PARITY DETECTION
7216  044770  062703  000002        17$:    ADD     #2,R3           ;CALCULATE NEXT ADDRESS
7217  044774  005305                        DEC     R5              ;CHECK IF FINISHED
7218  044776  001370                        BNE     15$             ;NO, GET NEXT PARITY MEMORY ADDRESS
7219  045000  000207                        RTS     PC              ;RETURN
7220
7221                                .SBTTL  CHECK FOR MEMORY CHECK ENABLE
7222
7223  045002  012737  045066  000004  PARCHK: MOV     #20$,ERRVEC     ;SET VECTOR FOR MEMORY PARITY CHECK
7224  045010  012737  000340  000006        MOV     #PR7,ERRVEC+2
7225  045016  012737  000000  067220        MOV     #0,BADPAR       ;LOAD GOOD PARITY
7226  045024  005037  003630                CLR     MEMPAR          ;CLEAR FLAG
7227  045030  012703  172100                MOV     #MEMBAS,R3      ;LOAD REGISTER TO DETERMINE IF
7228                                                                ;  MEMORY CHECK ENABLE AVAILIBLE
7229  045034  012704  000001                MOV     #1,R4           ;INITIALIZE MASK
7230  045040  012713  000001        16$:    MOV     #PAR.EN,(R3)    ;ENABLE MEMORY CHECK
7231  045044  005713                        TST     (R3)
7232  045046  062703  000002                ADD     #2,R3
7233  045052  050437  003630                BIS     R4,MEMPAR       ;SET FLAG
7234  045056  000241                        CLC
7235  045060  006104                        ROL     R4              ;CHECK IF FINISHED
7236  045062  001366                        BNE     16$             ;NO, SET UP NEXT MEMORY PARITY MODULE
7237  045064  000406                        BR      22$             ;RESTORE TRAP VECTOR
7238
7239  045066  022626                20$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
7240  045070  062703  000002                ADD     #2,R3
7241  045074  000241                        CLC
7242  045076  006104                        ROL     R4              ;CHECK IF FINISHED
7243  045100  001357                        BNE     16$             ;NO, GET NEXT LOCATION
7244  045102  012737  000006  000004  22$:    MOV     #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
7245  045110  005037  000006                CLR     ERRVEC+2
7246  045114  005737  003630                TST     MEMPAR          ;CHECK IF MEMORY CHECK ENABLE AVAILIABLE
7247  045120  001005                        BNE     25$             ;YES, RETURN
7248  045122  012737  000116  000114        MOV     #MEMVEC+2,MEMVEC ;RESTORE TRAP CATCHER
7249  045130  005037  000116                CLR     MEMVEC+2
7250  045134  000207                25$:    RTS     PC              ;RETURN
7251
7252                                .SBTTL  SIMULATE ONE BIT OF WRITE DATA IN MAINTANENCE MODE
7253
7254  045136  052737  002400  003524  WRTBIT: BIS     #MCLK!MEWD,E.MR1 ;CREATE EXPECTED MAINT. REG. 1
7255  045144  012762  000440  000026        MOV     #DMD!MCLK,RKMR1(R2) ;PROVIED 1ST UPWARD TRANSITION
7256  045152  016237  000026  003464        MOV     RKMR1(R2),T.MR1 ;STORE MAINT. REG. 1
7257  045160  023737  003524  003464        CMP     E.MR1,T.MR1     ;CHECK IF MAINT REG 1 CORRECT
7258  045166  001416                        BEQ     3$              ;YES, PROVIDE DOWNWARD TRANSISTION
7259  045170  012737  045210  001202        MOV     #1$,$ESCAPE     ;LOAD ESCAPE FOR LOOP ON ERROR
```

```
7260  045176  012737  066324  003172          MOV     #EMW1,EMW+2     ;LOAD ERROR MESSAGE
7261  045204  011646                          MOV     (SP),-(SP)      ;SAVE RETURN
7262  045206  000207                          RTS     PC              ;MR1 INCORRECT ON UPWARD TRANSITION
7263
7264  045210  032777  001000  133722  1$:     BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7265  045216  001402                          BEQ     3$              ;NO, CONTINUE
7266  045220  000137  046072                  JMP     63$             ;YES, LOOP ON ERROR
7267
7268  045224  042737  014400  003524  3$:     BIC     #MCLK!PCA!PCD,E.MR1 ;INITIALIZE MAINTENANCE REG. 1
7269  045232  052737  042000  003524          BIS     #MEWD!WRTGAT,E.MR1
7270  045240  005737  003614                  TST     PR.BIT          ;CHECK IF ONE
7271  045244  001152                          BNE     20$             ;YES, SIMULATE ONE
7272  045246  005737  003616                  TST     M1.BIT          ;CHECK IF PREVIOUS ONE
7273  045252  001023                          BNE     10$             ;YES, NO TRANSISTION
7274  045254  042737  002000  003524          BIC     #MEWD,E.MR1     ;INDICATE TRANSISTION
7275  045262  005737  003612                  TST     P1.BIT          ;CHECK IF NEXT BIT = 1
7276  045266  001007                          BNE     5$              ;YES, CHECK FOR PRECOMP ADVANCE
7277  045270  005737  003620                  TST     M2.BIT          ;CHECK FOR PRECOMP. ADVANCE
7278  045274  001412                          BEQ     10$             ;NO, CLOCK IN ZERO
7279  045276  052737  010000  003524          BIS     #PCD,E.MR1      ;SET PRECOMP. DELAY
7280  045304  000406                          BR      10$             ;CLOCK IN ZERO
7281
7282  045306  005737  003620          5$:     TST     M2.BIT          ;CHECK FOR PRECOMP. ADVANCE
7283  045312  001003                          BNE     10$             ;CLOCK IN ZERO
7284  045314  052737  004000  003524          BIS     #PCA,E.MR1      ;SET PRECOMP. ADVANCE
7285  045322  012762  000040  000026  10$:     MOV     #DMD,RKMR1(R2)  ;CLOCK IN DATA BIT
7286  045330  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MR1
7287  045336  023737  003464  003524          CMP     T.MR1,E.MR1     ;CHECK IF MR1 CORRECT
7288  045344  001416                          BEQ     12$             ;YES, CONTINUE
7289  045346  012737  045366  001202          MOV     #11$,$ESCAPE    ;LOAD ESCAPE FOR LOOP ON ERROR
7290  045354  012737  066413  003172          MOV     #EMW2,EMW+2     ;LOAD ERROR MESSAGE
7291  045362  011646                          MOV     (SP),-(SP)      ;SAVE RETURN
7292  045364  000207                          RTS     PC              ;MR1 INCORRECT
7293
7294  045366  032777  001000  133544  11$:     BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7295  045374  001402                          BEQ     12$             ;NO, CONTINUE
7296  045376  000137  046072                  JMP     63$             ;YES, LOOP ON ERROR
7297
7298  045402  052737  002400  003524  12$:     BIS     #MCLK!MEWD,E.MR1 ;CREATE EXPECTED MAINT REG 1
7299  045410  012762  000440  000026          MOV     #DMD!MCLK,RKMR1(R2) ;PROVIDE 2ND UPWARD TRANSISITION
7300  045416  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINT REG. 1
7301  045424  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK IF MAINT REG. 1 CORRECT
7302  045432  001416                          BEQ     15$             ;YES, CONTINUE
7303  045434  012737  045454  001202          MOV     #13$,$ESCAPE    ;LOAD ESCAPE FOR LOOP ON ERROR
7304  045442  012737  066504  003172          MOV     #EMW3,EMW+2     ;LOAD ERROR MESSAGE
7305  045450  011646                          MOV     (SP),-(SP)      ;SAVE RETURN
7306  045452  000207                          RTS     PC              ;MR1 INCORRECT
7307
7308  045454  032777  001000  133456  13$:     BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7309  045462  001402                          BEQ     15$             ;NO, CONTINUE
7310  045464  000137  046072                  JMP     63$             ;YES, LOOP ON ERROR
7311
7312  045470  052737  002000  003524  15$:     BIS     #MEWD,E.MR1     ;RESET TRANSITION INDICATION
7313  045476  042737  000400  003524          BIC     #MCLK,E.MR1
7314  045504  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;SUPPLY LAST PART OF DATA
7315  045512  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MR1
```

```
7316  045520  023737  003464  003524          CMP     T.MR1,E.MR1     ;CHECK IF MR1 CORRECT
7317  045526  001414                          BEQ     18$             ;YES, RETURN
7318  045530  012737  045550  001202          MOV     #17$,$ESCAPE    ;LOAD ESCAPE FOR LOOP ON ERROR
7319  045536  012737  066573  003172          MOV     #EMW4,EMW+2     ;LOAD ERROR MESSAGE
7320  045544  011646                          MOV     (SP),-(SP)      ;SAVE RETURN
7321  045546  000207                          RTS     PC              ;MR1 INCORRECT
7322
7323  045550  032777  001000  133362  17$:    BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7324  045556  001145                          BNE     63$             ;YES, LOOP ON ERROR
7325  045560  005037  001202          18$:    CLR     $ESCAPE         ;CLEAR ESCAPE
7326  045564  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN
7327  045570  000207                          RTS     PC              ;RETURN
7328
7329  045572  005737  003612          20$:    TST     P1.BIT          ;CHECK IN NEXT BIT A ONE
7330  045576  001007                          BNE     30$             ;YES, CHECK IF PRECOMP DELAY
7331  045600  005737  003616                  TST     M1.BIT          ;CHECK FOR PRECOMP ADVANCE
7332  045604  001415                          BEQ     40$             ;NO, CLOCK IN DATA BIT
7333  045606  052737  004000  003524          BIS     #PCA,E.MR1      ;SET PRECOMP. ADVANCE
7334  045614  000411                          BR      40$             ;CHECK MR1
7335
7336  045616  042737  000400  003524  30$:    BIC     #MCLK,E.MR1     ;RESET MAINT CLOCK IN EXPECTED MR1
7337  045624  005737  003616                  TST     M1.BIT          ;CHECK FOR PRECOMP DELAY
7338  045630  001003                          BNE     40$             ;NO, CHECK MR1
7339  045632  052737  010000  003524          BIS     #PCD,E.MR1      ;SET SET PRECOMP DELAY
7340  045640  012762  000040  000026  40$:    MOV     #DMD,RKMR1(R2)  ;CLOCK IN DATA BIT
7341  045646  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MR1
7342  045654  023737  003464  003524          CMP     T.MR1,E.MR1     ;CHECK MR1 CORRECT
7343  045662  001414                          BEQ     42$             ;YES, CLOCK IN REST OF BIT
7344  045664  012737  045704  001202          MOV     #41$,$ESCAPE    ;LOAD ESCAPE FOR LOOP ON ERROR
7345  045672  012737  066413  003172          MOV     #EMW2,EMW+2     ;LOAD ERROR MESSAGE
7346  045700  011646                          MOV     (SP),-(SP)      ;SAVE RETURN
7347  045702  000207                          RTS     PC              ;MR1 INCORRECT
7348
7349  045704  032777  001000  133226  41$:    BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7350  045712  001067                          BNE     63$             ;YES, LOOP ON ERROR
7351  045714  052737  000400  003524  42$:    BIS     #MCLK,E.MR1     ;CHREATE EXPECTED MAINT. REG. 1
7352  045722  012762  000440  000026          MOV     #DMD!MCLK,RKMR1(R2) ;PROVIDE 2ND UPWARD TRANSITION
7353  045730  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MAINT REG 1
7354  045736  023737  003524  003464          CMP     E.MR1,T.MR1     ;CHECK IF MAINT REG 1 CORRECT
7355  045744  001414                          BEQ     45$             ;YES, CONTINUE
7356  045746  012737  045766  001202          MOV     #43$,$ESCAPE    ;LOAD ESCAPE
7357  045754  012737  066504  003172          MOV     #EMW3,EMW+2     ;LOAD ERROR MESSAGE
7358  045762  011646                          MOV     (SP),-(SP)      ;SAVE RETURN
7359  045764  000207                          RTS     PC              ;MR1 INCORRECT
7360
7361  045766  032777  001000  133144  43$:    BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7362  045774  001036                          BNE     63$             ;YES, LOOP ON ERROR
7363  045776  042737  002400  003524  45$:    BIC     #MEWD!MCLK,E.MR1 ;SET TRANSISTION
7364  046004  012762  000040  000026          MOV     #DMD,RKMR1(R2)  ;CLOCK TRANSISTION
7365  046012  016237  000026  003464          MOV     RKMR1(R2),T.MR1 ;STORE MR1
7366  046020  023737  003464  003524          CMP     T.MR1,E.MR1     ;CHECK IF MR1 CORRECT
7367  046026  001414                          BEQ     50$             ;YES, RETURN
7368  046030  012737  046050  001202          MOV     #47$,$ESCAPE    ;LOAD ESCAPE FOR LOOP ON ERROR
7369  046036  012737  066573  003172          MOV     #EMW4,EMW+2     ;LOAD ERROR MESSAGE
7370  046044  011646                          MOV     (SP),-(SP)      ;SAVE RETURN
7371  046046  000207                          RTS     PC              ;MR1 INCORRECT
```

```
7372
7373  046050  032777  001000  133062  47$:    BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7374  046056  001005                          BNE     63$             ;YES, LOOP ON ERROR
7375  046060  005037  001202          50$:    CLR     $ESCAPE         ;CLEAR ESCAPE
7376  046064  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN
7377  046070  000207                          RTS     PC              ;RETURN
7378
7379  046072  005037  001202          63$:    CLR     $ESCAPE         ;CLEAR ESCAPE
7380  046076  012706  001100                  MOV     #STACK,SP       ;FORCE STACK
7381  046102  000177  133002                  JMP     @$LPERR         ;LOOP ON ERROR
7382
7383                                  .SBTTL  SIMULATE ONE BIT OR READ DATA IN MAINTENANCE MODE
7384
7385  046106  005737  003614          RDBIT:  TST     PR.BIT          ;CHECK IF ONE
7386  046112  001024                          BNE     10$             ;YES, SIMULATE ONE
7387  046114  005737  003616                  TST     M1.BIT          ;CHECK IF PREVIOUS ONE
7388  046120  001404                          BEQ     4$              ;NO, INSERT TRANSITION
7389  046122  012762  000440  000026          MOV     #DMD!MCLK,RKMR1(R2)  ;YES, DO NOT INSERT TRANSITION
7390  046130  000403                          BR      5$              ;CLOCK IN ZERO
7391
7392  046132  012762  001440  000026  4$:     MOV     #DMD!MCLK!MERD,RKMR1(R2)  ;INSERT TRANSITION
7393  046140  012762  000040  000026  5$:     MOV     #DMD,RKMR1(R2)  ;CLOCK IN ZERO
7394  046146  012762  000440  000026          MOV     #DMD!MCLK,RKMR1(R2)
7395  046154  012762  000040  000026          MOV     #DMD,RKMR1(R2)
7396  046162  000207                          RTS     PC              ;RETURN
7397
7398  046164  012762  000440  000026  10$:    MOV     #DMD!MCLK,RKMR1(R2)  ;CLOCK IN ONE
7399  046172  012762  000040  000026          MOV     #DMD,RKMR1(R2)
7400  046200  012762  001440  000026          MOV     #DMD!MCLK!MERD,RKMR1(R2)
7401  046206  012762  000040  000026          MOV     #DMD,RKMR1(R2)
7402  046214  000207                          RTS     PC              ;RETURN
7403
7404                                  .SBTTL  MEMORY CHECK ENABLE TRAP
7405
7406  046216  012737  046232  001202  MEMERR: MOV     #10$,$ESCAPE    ;LOAD ESCAPE
7407  046224  011637  003604                  MOV     (SP),TRAPPC     ;STORE PC
7408  046230  104147                          ERROR   147             ;REPORT MEM PARITY ERROR
7409  046232  005037  001202          10$:    CLR     $ESCAPE         ;CLEAR ESCAPE
7410  046236  032777  001000  132674          BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7411  046244  001001                          BNE     15$             ;YES, FORCE STACK AND TRY AGAIN
7412  046246  000002                          RTI                     ;NO, RETURN
7413
7414  046250  012706  001100          15$:    MOV     #STACK,SP       ;INITIALIZE STACK
7415  046254  000177  132630                  JMP     @$LPERR         ;LOOP ON ERROR
7416
7417                                  .SBTTL  ROUTINE TO SIZE MEMORY
7418
7419                                  ;;*****************************************************************
7420                                  ;*CALL:
7421                                  ;*      JSR     PC,$SIZE
7422                                  ;*      RETURN
7423                                  ;*$LSTAD WILL CONTAIN:
7424                                  ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
7425                                  ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
7426                                  ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
7427                                  ;*$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

046260  010046                $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
046262  010146                        MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
046264  010246                        MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
046266  010346                        MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
046270  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
046274  013746  000006                MOV     @#ERRVEC+2,-(SP)
046300  010600                        MOV     SP,R0           ;;SAVE THE STACK POINTER
                                ;;SET THE ERRVEC PS TO THE PRESENT PS
046302  104400                        TRAP                    ;;PUSH OLD PSW AND PC ON STACK
046304  012637  000006                MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
046310  012701  003776                MOV     #3776,R1        ;;SETUP ADDRESS
046314  105727                        TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
046316  000200                $KT11:  .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
046320  100062                        BPL     $CORE           ;;BR IF NO
046322  012737  046460  000004        MOV     #$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
046330  005737  177572                TST     @#SR0           ;;KT11 ARE YOU THERE?
046334  052737  100000  046316        BIS     #100000,$KT11   ;;YES--SET KT11 KEY
046342  005046                        CLR     -(SP)           ;;INITIALIZE FOR "PAR" LOADING
046344  012702  172340                MOV     #KIPAR0,R2      ;;ADDRESS OF FIRST "PAR"
046350  012703  000010                MOV     #^D8,R3         ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
046354  012762  077406  177740  1$:   MOV     #77406,-40(R2)  ;;PDR = 4K, UP, READ/WRITE
046362  011622                        MOV     (SP),(R2)+      ;;LOAD "PAR"
046364  062716  000200                ADD     #200,(SP)       ;;UPDATE FOR NEXT "PAR"
046370  077307                        SOB     R3,1$           ;;LOOP UNTIL ALL EIGHT ARE LOADED
046372  012742  177600                MOV     #177600,-(R2)   ;;SETUP KIPAR7 FOR I/O
046376  005042                        CLR     -(R2)           ;;SETUP KIPAR6 FOR TESTING
046400  012737  046416  000004        MOV     #2$,@#ERRVEC    ;;CATCH TIMEOUT IF NO SR3
046406  012737  000020  172516        MOV     #20,@#SR3       ;;ENABLE 22 BIT MODE
046414  000401                        BR      3$              ;;THIS PDP-11 HAS A SR3 REGISTER
046416  022626                2$:     CMP     (SP)+,(SP)+     ;;CLEAN OFF THE STACK--NO SR3
046420  005237  177572        3$:     INC     @#SR0           ;;TURN ON MEMORY MANAGEMENT
046424  012737  046450  000004        MOV     #$KTOUT,@#ERRVEC ;;SET FOR TIME OUT
046432  005737  143776        4$:     TST     @#143776        ;;TRAP ON NON-EX-MEM
046436  062712  000040                ADD     #40,(R2)        ;;MAKE A 1K STEP
046442  023712  172356                CMP     @#KIPAR7,(R2)   ;;LAST ONE?
046446  101371                        BHI     4$              ;;NO--TRY IT
046450  011202                $KTOUT: MOV     (R2),R2         ;;GET LAST BANK+1
046452  005037  177572                CLR     @#SR0           ;;TURN OFF MEMORY MANAGEMENT
046456  000421                        BR      $SIZEX
046460  042737  100000  046316  $KTNEX: BIC   #100000,$KT11   ;;KT11 NON-EXISTENT
046466  012737  046516  000004  $CORE:  MOV   #$CROUT,@#ERRVEC ;;SET FOR TIMEOUT
046474  005002                        CLR     R2              ;;SET UP BANK
046476  062701  004000        1$:     ADD     #4000,R1        ;;INCREMENT BY 1K
046502  062702  000040                ADD     #40,R2          ;;1K STEP
046506  005711                        TST     (R1)            ;;TRAP ON TIME OUT
046510  022701  177776                CMP     #177776,R1      ;;LAST ONE
046514  001370                        BNE     1$              ;;NO--TRY AGAIN
046516  162701  004000        $CROUT: SUB     #4000,R1
046522  162702  000040        $SIZEX: SUB     #40,R2          ;;DROP BACK
046526  010006                        MOV     R0,SP           ;;RESTORE THE STACK
046530  012637  000006                MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
```

```
7484  046534  012637  000004                  MOV     (SP)+,@#ERRVEC
7485  046540  010137  046562                  MOV     R1,$LSTAD       ;;LAST ADDRESS
7486  046544  010237  046564                  MOV     R2,$LSTBK       ;;LAST BANK
7487  046550  012603                          MOV     (SP)+,R3        ;;RESTORE R3
7488  046552  012602                          MOV     (SP)+,R2        ;;RESTORE R2
7489  046554  012601                          MOV     (SP)+,R1        ;;RESTORE R1
7490  046556  012600                          MOV     (SP)+,R0        ;;RESTORE R0
7491  046560  000207                          RTS     PC
7492  046562  000000                  $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
7493  046564  000000                  $LSTBK: .WORD   0               ;;CONTAINS THE LAST BANK
7494                                  .SBTTL  SCOPE HANDLER ROUTINE
7495
7496                                  ;;**************************************************************
7497                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
7498                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
7499                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
7500                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7501                                  ;*SW14=1        LOOP ON TEST
7502                                  ;*SW11=1        INHIBIT ITERATIONS
7503                                  ;*SW09=1        LOOP ON ERROR
7504                                  ;*SW08=1        LOOP ON TEST IN SWR<7:0>
7505                                  ;*CALL
7506                                  ;*      SCOPE           ;;SCOPE=IOT
7507
7508  046566                          $SCOPE:
7509  046566  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
7510  046570  032777  040000  132342  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
7511  046576  001131                          BNE     $OVER           ;;YES IF SW14=1
7512                                  ;#####START OF CODE FOR THE XOR TESTER#####
7513  046600  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
7514                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
7515  046602  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
7516  046606  012737  046626  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
7517  046614  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
7518  046620  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
7519  046624  000500                          BR      $SVLAD          ;;GO TO THE NEXT TEST
7520  046626  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
7521  046630  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
7522  046634  000440                          BR      7$              ;;LOOP ON THE PRESENT TEST
7523  046636                          6$:;#####END OF CODE FOR THE XOR TESTER#####
7524  046636  032777  000400  132274          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
7525  046644  001421                          BEQ     2$              ;;BR IF NO
7526  046646  005046                          CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
7527  046650  117716  132264                  MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
7528  046654  001414                          BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
7529  046656  022716  000053                  CMP     #53,(SP)        ;;CHECK THE NUMBER IN THE SWR
7530  046662  002411                          BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
7531  046664  011637  001102                  MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
7532  046670  005316                          DEC     (SP)            ;;BACKUP BY ONE
7533  046672  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
7534  046674  062716  047100                  ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
7535  046700  013637  001106                  MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
7536  046704  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
7537  046706  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
7538  046710  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
7539  046714  001421                          BEQ     3$              ;;BR IF NO
```

```
7540  046716  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
7541  046724  101015                          BHI     3$              ;;BR IF NO
7542  046726  032777  001000  132204          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
7543  046734  001404                          BEQ     4$              ;;BR IF NO
7544  046736  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
7545  046744  000446                          BR      $OVER
7546  046746  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
7547  046752  005037  001200                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
7548  046756  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
7549  046760  032777  004000  132152  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
7550  046766  001011                          BNE     1$              ;;BR IF YES
7551  046770  005737  001222                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
7552  046774  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
7553  046776  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
7554  047002  023737  001200  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
7555  047010  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
7556  047012  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
7557  047020  013737  047076  001200          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
7558  047026  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
7559  047032  113737  001102  001220          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
7560  047040  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
7561  047044  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
7562  047050  005037  001202                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
7563  047054  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
7564  047062  013777  001102  132052  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
7565  047070  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
7566  047074  000002                          RTI                     ;;FIXES PS
7567  047076  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
7568  047100                          $SW08TBL:
7569  047100  004702                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
7570  047102  005120                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
7571  047104  005336                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
7572  047106  005552                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
7573  047110  005766                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
7574  047112  006450                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
7575  047114  007076                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
7576  047116  007464                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
7577  047120  010610                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
7578  047122  011276                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
7579  047124  012162                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
7580  047126  013050                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
7581  047130  014254                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
7582  047132  015132                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
7583  047134  015570                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
7584  047136  016704                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
7585  047140  020020                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
7586  047142  021134                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
7587  047144  022372                          .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
7588  047146  023262                          .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
7589  047150  024106                          .WORD   TST25+2         ;;STARTING ADDRESS OF TEST 25
7590  047152  024640                          .WORD   TST26+2         ;;STARTING ADDRESS OF TEST 26
7591  047154  025372                          .WORD   TST27+2         ;;STARTING ADDRESS OF TEST 27
7592  047156  026124                          .WORD   TST30+2         ;;STARTING ADDRESS OF TEST 30
7593  047160  026656                          .WORD   TST31+2         ;;STARTING ADDRESS OF TEST 31
7594  047162  027410                          .WORD   TST32+2         ;;STARTING ADDRESS OF TEST 32
7595  047164  030142                          .WORD   TST33+2         ;;STARTING ADDRESS OF TEST 33
```

```
7596  047166  030442                          .WORD   TST34+2         ;;STARTING ADDRESS OF TEST 34
7597  047170  031210                          .WORD   TST35+2         ;;STARTING ADDRESS OF TEST 35
7598  047172  031600                          .WORD   TST36+2         ;;STARTING ADDRESS OF TEST 36
7599  047174  032424                          .WORD   TST37+2         ;;STARTING ADDRESS OF TEST 37
7600  047176  033250                          .WORD   TST40+2         ;;STARTING ADDRESS OF TEST 40
7601  047200  034074                          .WORD   TST41+2         ;;STARTING ADDRESS OF TEST 41
7602  047202  034720                          .WORD   TST42+2         ;;STARTING ADDRESS OF TEST 42
7603  047204  035544                          .WORD   TST43+2         ;;STARTING ADDRESS OF TEST 43
7604  047206  036370                          .WORD   TST44+2         ;;STARTING ADDRESS OF TEST 44
7605  047210  037214                          .WORD   TST45+2         ;;STARTING ADDRESS OF TEST 45
7606  047212  040054                          .WORD   TST46+2         ;;STARTING ADDRESS OF TEST 46
7607  047214  041100                          .WORD   TST47+2         ;;STARTING ADDRESS OF TEST 47
7608  047216  042124                          .WORD   TST50+2         ;;STARTING ADDRESS OF TEST 50
7609  047220  042764                          .WORD   TST51+2         ;;STARTING ADDRESS OF TEST 51
7610  047222  043410                          .WORD   TST52+2         ;;STARTING ADDRESS OF TEST 52
7611  047224  044034                          .WORD   TST53+2         ;;STARTING ADDRESS OF TEST 53
7612                          ;;*****************************************************************
7613                          .SBTTL  LOOP ON INTERNAL ERROR
7614
7615  047226  032777  001000  131704  SCOP1$: BIT     #SW9,@SWR       ;CHECK IF LOOP ON ERROR
7616  047234  001405                          BEQ     5$              ;NO RETURN
7617  047236  105737  001103                  TSTB    $ERFLG          ;CHECK IF ERROR OCCURED
7618  047242  001402                          BEQ     5$              ;NO, RETURN
7619  047244  013716  001110                  MOV     $LPERR,(SP)     ;GO BACK TO BEGINNING OF LOOP
7620  047250  000002                  5$:     RTI                     ;RETURN
7621                          .SBTTL  APT COMMUNICATIONS ROUTINE
7622
7623                          ;;*****************************************************************
7624  047252  112737  000001  047516  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
7625  047260  112737  000001  047514  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
7626  047266  000403                  BR      $ATYC
7627  047270  112737  000001  047516  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
7628  047276                          $ATYC:
7629  047276  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
7630  047300  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
7631  047302  105737  047514                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
7632  047306  001450                          BEQ     5$              ;;IF NOT:  BR
7633  047310  122737  000001  001234          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
7634  047316  001031                          BNE     3$              ;;IF NOT:  BR
7635  047320  132737  000100  001235          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
7636  047326  001425                          BEQ     3$              ;;IF NOT:  BR
7637  047330  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
7638  047334  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
7639  047342  005737  001214          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
7640  047346  001375                          BNE     1$              ;;IF NOT:  WAIT
7641  047350  010037  001230                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
7642  047354  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
7643  047356  001376                          BNE     2$
7644  047360  163700  001230                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
7645  047364  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
7646  047366  010037  001232                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
7647  047372  012737  000004  001214          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
7648  047400  000413                          BR      5$
7649  047402  017637  000004  047426  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
7650  047410  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
7651  047416  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
```

```
7652  047422  004737  050260          JSR     PC,$TYPE        ;;CALL TYPE MACRO
7653  047426  000000          4$:     .WORD   0
7654  047430                  5$:
7655  047430  105737  047516  10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
7656  047434  001416          BEQ     12$             ;;IF NOT:  BR
7657  047436  005737  001234          TST     $ENV            ;;RUNNING UNDER APT?
7658  047442  001413          BEQ     12$             ;;IF NOT:  BR
7659  047444  005737  001214  11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
7660  047450  001375          BNE     11$             ;;IF NOT:  WAIT
7661  047452  017637  000004  001216  MOV     @4(SP),$FATAL   ;;GET ERROR #
7662  047460  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
7663  047466  005237  001214          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
7664  047472  105037  047516  12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
7665  047476  105037  047515          CLRB    $LFLG           ;;CLEAR LOG FLAG
7666  047502  105037  047514          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
7667  047506  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
7668  047510  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
7669  047512  000207                  RTS     PC              ;;RETURN
7670  047514     000  $MFLG:  .BYTE   0               ;;MESSG. FLAG
7671  047515     000  $LFLG:  .BYTE   0               ;;LOG FLAG
7672  047516     000  $FFLG:  .BYTE   0               ;;FATAL FLAG
7673          047520          .EVEN
7674          000200  APTSIZE=200
7675          000001  APTENV=001
7676          000100  APTSPOOL=100
7677          000040  APTCSUP=040
7678                  .SBTTL  ERROR HANDLER ROUTINE
7679
7680                  ;;**********************************************************************
7681                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
7682                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
7683                  ;*AND GO TO TYPERR ON ERROR
7684                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7685                  ;*SW15=1         HALT ON ERROR
7686                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
7687                  ;*SW10=1         BELL ON ERROR
7688                  ;*SW09=1         LOOP ON ERROR
7689                  ;*CALL
7690                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
7691
7692  047520          $ERROR:
7693  047520  104407          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
7694  047522  105237  001103  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
7695  047526  001775          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
7696  047530  013777  001102  131404  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
7697  047536  032777  002000  131374  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
7698  047544  001402          BEQ     1$              ;;NO - SKIP
7699  047546  104401  001204          TYPE    ,$BELL          ;;RING BELL
7700  047552  005237  001112  1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
7701  047556  011637  001116          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
7702  047562  162737  000002  001116  SUB     #2,$ERRPC
7703  047570  117737  131322  001114  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
7704  047576  032777  020000  131334  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
7705  047604  001004          BNE     20$             ;;SKIP TYPEOUTS
7706  047606  004737  047720          JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
7707  047612  104401  001211          TYPE    ,$CRLF
```

```
                                20$:
7708  047616                    
7709  047616  122737  000001  001234        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
7710  047624  001007                        BNE     2$              ;;NO,SKIP APT ERROR REPORT
7711  047626  113737  001114  047640        MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
7712  047634  004737  047270                JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
7713  047640     000                21$:    .BYTE   0
7714  047641     000                        .BYTE   0
7715  047642  000777                22$:    BR      22$             ;;APT ERROR LOOP
7716  047644  005777  131270        2$:     TST     @SWR            ;;HALT ON ERROR
7717  047650  100002                        BPL     3$              ;;SKIP IF CONTINUE
7718  047652  000000                        HALT                    ;;HALT ON ERROR!
7719  047654  104407                        CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
7720  047656  032777  001000  131254  3$:    BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
7721  047664  001402                        BEQ     4$              ;;BR IF NO
7722  047666  013716  001110                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
7723  047672  005737  001202        4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
7724  047676  001402                        BEQ     5$              ;;BR IF NONE
7725  047700  013716  001202                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
7726  047704                        5$:
7727  047704  022737  044662  000042        CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
7728  047712  001001                        BNE     6$              ;;BRANCH IF NO
7729  047714  000000                        HALT                    ;;YES
7730  047716                        6$:
7731  047716  000002                        RTI                     ;;RETURN
7732
7733                            ;;*****************************************************************
7734                            .SBTTL  TYPE ERROR ROUTINE
7735                            ;*ENTRY JSR PC,TYPERR
7736                            ;*RETURN RTS PC
7737                            ;*
7738                            ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
7739                            ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
7740                            ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
7741                            ;*THE ERROR.
7742                            ;;*****************************************************************
7743  047720  104413            TYPERR: SAVREG
7744  047722  113700  001114            MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
7745  047726  042700  177400            BIC     #177400,R0      ;CLEAR UNUSED BITS
7746  047732  005300                    DEC     R0              ;FORM INDEX FOR ERROR TABLE
7747  047734  006300                    ASL     R0
7748  047736  006300                    ASL     R0
7749  047740  006300                    ASL     R0
7750  047742  062700  001300    1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
7751  047746  012037  047762            MOV     (R0)+,2$        ;GET EM POINTER
7752  047752  001404                    BEQ     3$              ;BRANCH IF THERE ISN'T ONE
7753  047754  104401  001211            TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
7754  047760  104401                    TYPE                    ;TYPE ERROR MESSAGE (EM)
7755  047762  000000            2$:     .WORD   0               ;EM POINTER GOES HERE
7756  047764  012037  050000    3$:     MOV     (R0)+,4$        ;GET DH POINTER
7757  047770  001404                    BEQ     5$              ;BRANCH IF THERE ISN'T ONE
7758  047772  104401  001211            TYPE    ,$CRLF          ;TYPE CR-LF
7759  047776  104401                    TYPE                    ;TYPE DATA HEADER
7760  050000  000000            4$:     .WORD   0               ;DH POINTER GOES HERE
7761  050002  012001            5$:     MOV     (R0)+,R1        ;GET DT POINTER
7762  050004  001445                    BEQ     20$             ;BRANCH IF THERE ARE NONE
7763  050006  005004                    CLR     R4              ;RESET INDENT SWITCH
```

```
7764  050010  012000                        MOV     (R0)+,R0        ;GET DF POINTER
7765  050012  012002                        MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
7766  050014  104401  001211                TYPE    ,$CRLF
7767  050020  112003                10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
7768  050022  105720                        TSTB    (R0)+           ;BUMP PAST FORMAT WORD
7769  050024  005703                        TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
7770  050026  001416                        BEQ     14$             ;NO - SKIP DATA PRINT
7771  050030  005704                        TST     R4              ;CHECK IF INDENT WORDS
7772  050032  001004                        BNE     12$             ;YES, GO INDENT
7773  050034  013146                11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
7774  050036  104402                        TYPOC                   ;TYPE IT
7775  050040  005303                        DEC     R3              ;MORE DATA WORDS
7776  050042  001403                        BEQ     13$             ;NO-BRANCH
7777  050044  104401  054644          12$:  TYPE    ,$SPACE2        ;TYPE SEPARATORS
7778  050050  000771                        BR      11$             ;LOOP
7779  050052  104401  001211          13$:  TYPE    ,$CRLF          ;TYPE <CR><LF>
7780  050056  005710                        TST     (R0)            ;CHECK IF NEXT HEADER AVAILIBLE
7781  050060  001401                        BEQ     14$             ;NO, DO NOT CHANGE INDENT
7782  050062  005104                        COM     R4              ;CHANGE INDENT
7783  050064  005302                14$:    DEC     R2              ;MORE DH'S?
7784  050066  003414                        BLE     20$             ;NO-BRANCH
7785  050070  012037  050110          15$:  MOV     (R0)+,18$       ;GET NEXT DH POINTER
7786  050074  001751                        BEQ     10$             ;IF NO HEADER GET DATA
7787  050076  005704                        TST     R4              ;INDENT?
7788  050100  001402                        BEQ     17$             ;NO-BRANCH
7789  050102  104401  054644                TYPE    ,$SPACE2        ;INDENT
7790  050106  104401                17$:    TYPE                    ;TYPE DH
7791  050110  000000                18$:    .WORD   0               ;DH POINTER GOES HERE
7792  050112  104401  001211                TYPE    ,$CRLF
7793  050116  000740                        BR      10$             ;LOOP
7794  050120  104414                20$:    RESREG
7795  050122  005237  003610                INC     ERRCNT          ;INCREMENT ERROR COUNT
7796  050126  032777  010000  131004        BIT     #SW12,@SWR      ;CHECK IF ABORT AFTER 20 ERRORS
7797  050134  001421                        BEQ     25$             ;NO, RETURN
7798  050136  022737  000024  003610        CMP     #20.,ERRCNT     ;CHECK IF EROR THRESHOLD EXCEEDED
7799  050144  103015                        BHIS    25$             ;NO, RETURN
7800  050146  104401  054647                TYPE    ,ABORT          ;TYPE "PROGRAM HAS BEEN ABORTED BECAUSE
7801                                                                ;  ERROR THRESHOLD EXCEEDED"
7802  050152  005737  000042                TST     42              ;CHECK IF IN CHAIN MODE
7803  050156  001407                        BEQ     30$             ;NO, HALT
7804  050160  012737  000001  044520        MOV     #1,$EOPCT       ;FORCE END OF PASS COUNT TO ONE FOR ABORT
7805  050166  012706  001100                MOV     #STACK,SP       ;INITIALIZE STACK
7806  050172  000137  044472                JMP     $EOP            ;BRING IN NEXT PROGRAM IN CHAIN
7807  050176  000000                30$:    HALT
7808  050200  000207                25$:    RTS     PC
7809
7810                                  .SBTTL  CONTROLLED PROGRAM HALT
7811  050202  013702  001270          CTRHLT: MOV     $BASE,R2        ;SET '61 BASE
7812  050206  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
7813  050214  012706  001100                MOV     #STACK,SP       ;CLEAR STACK
7814  050220  104401  054600                TYPE    ,OPROD7         ;TYPE HALT MESSAGE
7815  050224  005737  000042                TST     42              ;TEST IF MONITOR PRESENT
7816  050230  001410                        BEQ     5$              ;NO - SKIP
7817  050232  105037  001103                CLRB    $ERFLG          ;CLEAR ERROR FLAG
7818  050236  005037  001202                CLR     $ESCAPE         ;CLEAR ESCAPE
7819  050242  005037  044520                CLR     $EOPCT          ;SET PASS COUNT TO 0
```

```
7820  050246  000137  044472                  JMP     $EOP            ;GO TO END OF PASS
7821
7822  050252  000000                  5$:     HALT                    ;HALT PROGRAM
7823  050254  000137  003662                  JMP     START1          ;DO RESTART IF CONTINUE
7824
7825                                  .SBTTL  TYPE ROUTINE
7826
7827                                  ;;*********************************************************************
7828                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
7829                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
7830                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
7831                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
7832                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
7833                                  ;*
7834                                  ;*CALL:
7835                                  ;*1) USING A TRAP INSTRUCTION
7836                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
7837                                  ;*OR
7838                                  ;*      TYPE
7839                                  ;*      MESADR
7840                                  ;*
7841
7842  050260  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
7843  050264  100002                  BPL     1$              ;;BR IF YES
7844  050266  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
7845  050270  000430                  BR      3$              ;;LEAVE
7846  050272  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
7847  050274  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
7848  050300  122737  000001  001234  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
7849  050306  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
7850  050310  132737  000100  001235  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
7851  050316  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
7852  050320  010037  050330          MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
7853  050324  004737  047260          JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
7854  050330  000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
7855  050332  132737  000040  001235  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
7856  050340  001003                  BNE     60$             ;;YES,SKIP TYPE OUT
7857  050342  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
7858  050344  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
7859  050346  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
7860  050350  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
7861  050352  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
7862  050356  000002                  RTI                     ;;RETURN
7863  050360  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
7864  050364  001430                  BEQ     8$
7865  050366  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
7866  050372  001006                  BNE     5$
7867  050374  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
7868  050376  104401                  TYPE                    ;;TYPE A CR AND LF
7869  050400  001211                  $CRLF
7870  050402  105037  050536          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
7871  050406  000755                  BR      2$              ;;GET NEXT CHARACTER
7872  050410  004737  050472  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
7873  050414  123726  001156  6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
7874  050420  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
7875  050422  013746  001154          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
```

```
7876
7877  050426  105366  000001          7$:     DECB    1(SP)           ;;AND THE NULL CHAR.
7878  050432  002770                          BLT     6$              ;;DOES A NULL NEED TO BE TYPED?
7879  050434  004737  050472                  JSR     PC,$TYPEC       ;;BR IF NO--GO POP THE NULL OFF OF STACK
7880  050440  105337  050536                  DECB    $CHARCNT        ;;GO TYPE A NULL
7881  050444  000770                          BR      7$              ;;DO NOT COUNT AS A COUNT
7882                                                                  ;;LOOP
7883                                  ;HORIZONTAL TAB PROCESSOR
7884
7885  050446  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
7886  050452  004737  050472          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
7887  050456  132737  000007  050536          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
7888  050464  001372                          BNE     9$              ;;TAB STOP
7889  050466  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
7890  050470  000724                          BR      2$              ;;GET NEXT CHARACTER
7891  050472  105777  130452          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
7892  050476  100375                          BPL     $TYPEC
7893  050500  116677  000002  130444          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
7894  050506  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
7895  050514  001003                          BNE     1$              ;;BRANCH IF NO
7896  050516  105037  050536                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
7897  050522  000406                          BR      $TYPEX          ;;EXIT
7898  050524  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
7899  050532  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
7900  050534  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
7901  050536  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
7902  050540  000207          $TYPEX: RTS     PC
7903
7904                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
7905
7906                                  ;;******************************************************************
7907                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
7908                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
7909                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
7910                                  ;*CALL:
7911                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
7912                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
7913                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
7914                                  ;*      .BYTE   M               ;;M=1 OR 0
7915                                  ;*                                      ;;1=TYPE LEADING ZEROS
7916                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
7917                                  ;*
7918                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
7919                                  ;*$TYPOS OR $TYPOC
7920                                  ;*CALL:
7921                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
7922                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
7923                                  ;*
7924                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
7925                                  ;*CALL:
7926                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
7927                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
7928
7929  050542  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
7930  050546  116637  000001  050765          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
7931  050554  112637  050767                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
```

```
7932  050560  062716  000002                    ADD     #2,(SP)          ;;ADJUST RETURN ADDRESS
7933  050564  000406                            BR      $TYPON
7934  050566  112737  000001  050765  $TYPOC:   MOVB    #1,$OFILL        ;;SET THE ZERO FILL SWITCH
7935  050574  112737  000006  050767            MOVB    #6,$OMODE+1      ;;SET FOR SIX(6) DIGITS
7936  050602  112737  000005  050764  $TYPON:   MOVB    #5,$OCNT         ;;SET THE ITERATION COUNT
7937  050610  010346                            MOV     R3,-(SP)         ;;SAVE R3
7938  050612  010446                            MOV     R4,-(SP)         ;;SAVE R4
7939  050614  010546                            MOV     R5,-(SP)         ;;SAVE R5
7940  050616  113704  050767                    MOVB    $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
7941  050622  005404                            NEG     R4
7942  050624  062704  000006                    ADD     #6,R4            ;;SUBTRACT IT FOR MAX. ALLOWED
7943  050630  110437  050766                    MOVB    R4,$OMODE        ;;SAVE IT FOR USE
7944  050634  113704  050765                    MOVB    $OFILL,R4        ;;GET THE ZERO FILL SWITCH
7945  050640  016605  000012                    MOV     12(SP),R5        ;;PICKUP THE INPUT NUMBER
7946  050644  005003                            CLR     R3               ;;CLEAR THE OUTPUT WORD
7947  050646  006105                  1$:       ROL     R5               ;;ROTATE MSB INTO "C"
7948  050650  000404                            BR      3$               ;;GO DO MSB
7949  050652  006105                  2$:       ROL     R5               ;;FORM THIS DIGIT
7950  050654  006105                            ROL     R5
7951  050656  006105                            ROL     R5
7952  050660  010503                            MOV     R5,R3
7953  050662  006103                  3$:       ROL     R3               ;;GET LSB OF THIS DIGIT
7954  050664  105337  050766                    DECB    $OMODE           ;;TYPE THIS DIGIT?
7955  050670  100016                            BPL     7$               ;;BR IF NO
7956  050672  042703  177770                    BIC     #177770,R3       ;;GET RID OF JUNK
7957  050676  001002                            BNE     4$               ;;TEST FOR 0
7958  050700  005704                            TST     R4               ;;SUPPRESS THIS 0?
7959  050702  001403                            BEQ     5$               ;;BR IF YES
7960  050704  005204                  4$:       INC     R4               ;;DON'T SUPPRESS ANYMORE 0'S
7961  050706  052703  000060                    BIS     #'0,R3           ;;MAKE THIS DIGIT ASCII
7962  050712  052703  000040            5$:     BIS     #' ,R3           ;;MAKE ASCII IF NOT ALREADY
7963  050716  110337  050762                    MOVB    R3,8$            ;;SAVE FOR TYPING
7964  050722  104401  050762                    TYPE    ,8$              ;;GO TYPE THIS DIGIT
7965  050726  105337  050764            7$:     DECB    $OCNT            ;;COUNT BY 1
7966  050732  003347                            BGT     2$               ;;BR IF MORE TO DO
7967  050734  002402                            BLT     6$               ;;BR IF DONE
7968  050736  005204                            INC     R4               ;;INSURE LAST DIGIT ISN'T A BLANK
7969  050740  000744                            BR      2$               ;;GO DO THE LAST DIGIT
7970  050742  012605                  6$:       MOV     (SP)+,R5         ;;RESTORE R5
7971  050744  012604                            MOV     (SP)+,R4         ;;RESTORE R4
7972  050746  012603                            MOV     (SP)+,R3         ;;RESTORE R3
7973  050750  016666  000002  000004            MOV     2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
7974  050756  012616                            MOV     (SP)+,(SP)
7975  050760  000002                            RTI                      ;;RETURN
7976  050762     000            8$:     .BYTE   0                        ;;STORAGE FOR ASCII DIGIT
7977  050763     000                    .BYTE   0                        ;;TERMINATOR FOR TYPE ROUTINE
7978  050764     000            $OCNT:  .BYTE   0                        ;;OCTAL DIGIT COUNTER
7979  050765     000            $OFILL: .BYTE   0                        ;;ZERO FILL SWITCH
7980  050766  000000            $OMODE: .WORD   0                        ;;NUMBER OF DIGITS TO TYPE
7981                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7982
7983                            ;;**************************************************************
7984                            ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
7985                            ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
7986                            ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
7987                            ;*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
                                        $TYPDS:
050770  010046                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
050772  010146                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
050774  010246                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
050776  010346                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
051000  010546                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
051002  012746  020200                        MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
051006  016605  000020                        MOV     20(SP),R5       ;;GET THE INPUT NUMBER
051012  100004                                BPL     1$              ;;BR IF INPUT IS POS.
051014  005405                                NEG     R5              ;;MAKE THE BINARY NUMBER POS.
051016  112766  000055  000001                MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
051024  005000                        1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
051026  012703  051204                        MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
051032  112723  000040                        MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
051036  005002                        2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
051040  016001  051174                        MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
051044  160105                        3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
051046  002402                                BLT     4$              ;;BR IF DONE
051050  005202                                INC     R2              ;;INCREASE THE BCD DIGIT BY 1
051052  000774                                BR      3$
051054  060105                        4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
051056  005702                                TST     R2              ;;CHECK IF BCD DIGIT=0
051060  001002                                BNE     5$              ;;FALL THROUGH IF 0
051062  105716                                TSTB    (SP)            ;;STILL DOING LEADING 0'S?
051064  100407                                BMI     7$              ;;BR IF YES
051066  106316                        5$:     ASLB    (SP)            ;;MSD?
051070  103003                                BCC     6$              ;;BR IF NO
051072  116663  000001  177777                MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
051100  052702  000060                6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
051104  052702  000040                7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
051110  110223                                MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
051112  005720                                TST     (R0)+           ;;JUST INCREMENTING
051114  020027  000010                        CMP     R0,#10          ;;CHECK THE TABLE INDEX
051120  002746                                BLT     2$              ;;GO DO THE NEXT DIGIT
051122  003002                                BGT     8$              ;;GO TO EXIT
051124  010502                                MOV     R5,R2           ;;GET THE LSD
051126  000764                                BR      6$              ;;GO CHANGE TO ASCII
051130  105726                        8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
051132  100003                                BPL     9$              ;;BR IF NO
051134  116663  177777  177776                MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
051142  105013                        9$:     CLRB    (R3)            ;;SET THE TERMINATOR
051144  012605                                MOV     (SP)+,R5        ;;POP STACK INTO R5
051146  012603                                MOV     (SP)+,R3        ;;POP STACK INTO R3
051150  012602                                MOV     (SP)+,R2        ;;POP STACK INTO R2
051152  012601                                MOV     (SP)+,R1        ;;POP STACK INTO R1
051154  012600                                MOV     (SP)+,R0        ;;POP STACK INTO R0
051156  104401  051204                        TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
051162  016666  000002  000004                MOV     2(SP),4(SP)     ;;ADJUST THE STACK
051170  012616                                MOV     (SP)+,(SP)
051172  000002                                RTI                     ;;RETURN TO USER
051174  023420                        $DTBL:  10000.
```

```
8044  051176  001750                        1000.
8045  051200  000144                        100.
8046  051202  000012                        10.
8047  051204  000004              $DBLK:    .BLKW   4
8048                              .SBTTL    TTY INPUT ROUTINE
8049
8050                              ;;*****************************************************************
8051                              .ENABL    LSB
8052  051214  000000              $TKCNT:   .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
8053  051216  000000              $TKQIN:   .WORD   0               ;;INPUT POINTER
8054  051220  000000              $TKQOUT:  .WORD   0               ;;OUTPUT POINTER
8055  051222  000001              $TKQSRT:  .BLKB   1               ;;TTY KEYBOARD QUEUE
8056          051223              $TKQEND=.
8057          051224              .EVEN
8058
8059                              ;*TK INITIALIZE ROUTINE
8060                              ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
8061                              ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
8062                              ;
8063                              ;*CALL:
8064                              ;*        JSR     PC,$TKINT
8065                              ;*        RETURN
8066                              ;
8067  051224  005037  051214     $TKINT:   CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
8068  051230  012737  051222  051216       MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
8069  051236  013737  051216  051220       MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
8070  051244  012737  051274  000060       MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
8071  051252  012737  000200  000062       MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
8072  051260  005777  127662              TST     @$TKB           ;;CLEAR DONE FLAG
8073  051264  012777  000100  127652       MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
8074  051272  000207                       RTS     PC              ;;RETURN TO CALLER
8075
8076                              ;*TK SERVICE ROUTINE
8077                              ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
8078                              ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
8079                              ;*IT IN THE QUEUE.
8080                              ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
8081                              ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT)
8082                              ;
8083  051274  117746  127646     $TKSRV:   MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
8084  051300  042716  177600              BIC     #↑C177,(SP)     ;;STRIP THE JUNK
8085  051304  021627  000003              CMP     (SP),#3         ;;IS IT A CONTROL C?
8086  051310  001007                      BNE     1$              ;;BRANCH IF NO
8087  051312  104401  052410              TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (↑C)
8088  051316  004737  051224              JSR     PC,$TKINT       ;;INIT THE KEYBOARD
8089  051322  005726                      TST     (SP)+           ;;CLEAN UP STACK
8090  051324  000137  050202              JMP     CTRHLT          ;;CONTROL C RESTART
8091  051330  021627  000007     1$:      CMP     (SP),#7         ;;IS IT A CONTROL G?
8092  051334  001004                      BNE     2$              ;;BRANCH IF NO
8093  051336  022737  000176  001140      CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
8094  051344  001500                      BEQ     6$              ;;GO TO SWR CHANGE
8095
8096  051346              2$:
8097  051346  022737  000001  051214      CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
8098  051354  001004                      BNE     3$              ;;BRANCH IF NO
8099  051356  104401  001204              TYPE    ,$BELL          ;;RING THE TTY BELL
```

```
8100  051362  005726                        TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
8101  051364  000451                        BR      5$              ;;EXIT
8102  051366  021627  000023          3$:   CMP     (SP),#23        ;;IS IT A CONTROL-S?
8103  051372  001021                        BNE     32$             ;;BRANCH IF NO
8104  051374  005077  127544                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
8105  051400  005726                        TST     (SP)+           ;;CLEAN CHAR OFF STACK
8106  051402  105777  127536          31$:  TSTB    @$TKS           ;;WAIT FOR A CHAR
8107  051406  100375                        BPL     31$             ;;LOOP UNTIL ITS THERE
8108  051410  117746  127532                MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
8109  051414  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
8110  051420  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
8111  051424  001366                        BNE     31$             ;;BRANCH IF NO
8112  051426  012777  000100  127510        MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
8113  051434  000002                        RTI                     ;;RETURN
8114  051436  005237  051214          32$:  INC     $TKCNT          ;;COUNT THIS CHARACTER
8115  051442  021627  000140                CMP     (SP),#140       ;;IS IT UPPER CASE?
8116  051446  002405                        BLT     4$              ;;BRANCH IF YES
8117  051450  021627  000175                CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
8118  051454  003002                        BGT     4$              ;;BRANCH IF YES
8119  051456  042716  000040                BIC     #40,(SP)        ;;MAKE IT UPPER CASE
8120  051462  112677  177530          4$:   MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
8121  051466  005237  051216                INC     $TKQIN          ;;UPDATE THE POINTER
8122  051472  023727  051216  051223        CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
8123  051500  001003                        BNE     5$              ;;BRANCH IF NO
8124  051502  012737  051222  051216        MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
8125  051510  000002                  5$:   RTI                     ;;RETURN
8126
8127                                        ;;**************************************************************
8128                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
8129                                        ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
8130                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
8131                                        ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
8132  051512  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
8133  051520  001124                        BNE     15$             ;;EXIT IF NOT
8134  051522  105777  127416                TSTB    @$TKS           ;;IS A CHAR WAITING?
8135  051526  100121                        BPL     15$             ;;IF NOT, EXIT
8136  051530  117746  127412                MOVB    @$TKB,-(SP)     ;;YES
8137  051534  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
8138  051540  021627  000007                CMP     (SP),#7         ;;IS IT A CONTROL-G?
8139  051544  001300                        BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
8140                                                                ;;AND EXIT
8141
8142                                        ;;**************************************************************
8143                                        ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
8144                                        ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
8145                                        ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
8146  051546  123727  001134  000001    6$:   CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
8147  051554  001674                        BEQ     2$              ;;BRANCH IF YES
8148  051556  005726                        TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
8149  051560  004737  051224                JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
8150  051564  005077  127354                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
8151  051570  112737  000001  001135        MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
8152
8153  051576  104401  052422                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
8154  051602  104401  052427          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
8155  051606  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
```

```
8156  051612  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
8157  051614  104401  052440                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
8158  051620  005046                  19$:     CLR     -(SP)           ;;CLEAR COUNTER
8159  051622  005046                          CLR     -(SP)           ;;THE NEW SWR
8160  051624  105777  127314          7$:     TSTB    @$TKS           ;;CHAR THERE?
8161  051630  100375                          BPL     7$              ;;IF NOT TRY AGAIN
8162
8163  051632  117746  127310                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
8164  051636  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
8165
8166  051642  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
8167  051646  001015                          BNE     9$              ;;BRANCH IF NOT
8168  051650  104401  052410                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
8169  051654  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
8170  051660  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
8171  051666  001003                          BNE     8$              ;;BRANCH IF NO
8172  051670  012777  000100  127246          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
8173  051676  000137  050202          8$:     JMP     CTRHLT          ;;CONTROL-C RESTART
8174
8175
8176  051702  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
8177  051706  001005                          BNE     10$             ;;BRANCH IF NOT
8178  051710  104401  052415                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
8179  051714  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
8180  051720  000737                          BR      19$             ;;LET'S TRY IT AGAIN
8181
8182
8183  051722  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
8184  051726  001022                          BNE     16$             ;;BRANCH IF NO
8185  051730  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
8186  051734  001403                          BEQ     11$             ;;BRANCH IF YES
8187  051736  016677  000002  127174          MOV     2(SP),@SWR      ;;SAVE NEW SWR
8188  051744  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
8189  051750  104401  001211          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
8190  051754  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
8191  051762  001003                          BNE     15$             ;;BRANCH IF NOT
8192  051764  012777  000100  127152          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
8193  051772  000002                  15$:    RTI                     ;;RETURN
8194  051774  004737  050472          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
8195  052000  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
8196  052004  002420                          BLT     18$             ;;BRANCH IF YES
8197  052006  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
8198  052012  003015                          BGT     18$             ;;BRANCH IF YES
8199  052014  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
8200  052020  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
8201  052024  001403                          BEQ     17$             ;;BRANCH IF YES
8202  052026  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
8203  052030  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
8204  052032  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
8205  052034  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
8206  052040  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
8207  052044  000667                          BR      7$              ;;GET THE NEXT ONE
8208  052046  104401  001210          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
8209  052052  000720                          BR      20$             ;;SIMULATE CONTROL-U
8210                                  .DSABL  LSB
8211
```

```
8212
8213                                    ;;**********************************************************************
8214                                    ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
8215                                    ;*CALL:
8216                                    ;*      RDCHR                       ;;GET A CHARACTER FROM THE QUEUE
8217                                    ;*      RETURN HERE                 ;;CHARACTER IS ON THE STACK
8218                                    ;*                                  ;;WITH PARITY BIT STRIPPED OFF
8219                                    ;
8220
8221  052054  011646                    $RDCHR: MOV     (SP),-(SP)          ;;PUSH DOWN THE PC AND
8222  052056  016666  000004  000002            MOV     4(SP),2(SP)         ;;THE PS
8223  052064  005066  000004                    CLR     4(SP)               ;;GET READY FOR A CHARACTER
8224  052070  005046                            CLR     -(SP)               ;;PUT NEW PS ON STACK
8225  052072  012746  052100                    MOV     #64$,-(SP)          ;;PUT NEW PC ON  STACK
8226  052076  000002                            RTI                         ;;POP NEW PC AND PS
8227  052100                            64$:
8228  052100  005737  051214            1$:     TST     $TKCNT              ;;WAIT ON A CHARACTER
8229  052104  001775                            BEQ     1$
8230  052106  005337  051214                    DEC     $TKCNT              ;;DECREMENT THE COUNTER
8231  052112  117766  177102  000004            MOVB    @$TKQOUT,4(SP)      ;;GET ONE CHARACTER
8232  052120  005237  051220                    INC     $TKQOUT             ;;UPDATE THE POINTER
8233  052124  023727  051220  051223            CMP     $TKQOUT,#$TKQEND    ;;DID IT GO OFF OF THE END?
8234  052132  001003                            BNE     2$                  ;;BRANCH IF NO
8235  052134  012737  051222  051220            MOV     #$TKQSRT,$TKQOUT    ;;RESET THE POINTER
8236  052142  000002                    2$:     RTI                         ;;RETURN
8237                                    ;;**********************************************************************
8238                                    ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
8239                                    ;*CALL:
8240                                    ;*      RDLIN                       ;;INPUT A STRING FROM THE TTY
8241                                    ;*      RETURN HERE                 ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
8242                                    ;*                                  ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
8243
8244  052144  010346                    $RDLIN: MOV     R3,-(SP)            ;;SAVE R3
8245  052146  005046                            CLR     -(SP)               ;;CLEAR THE RUBOUT KEY
8246  052150  012703  052400            1$:     MOV     #$TTYIN,R3          ;;GET ADDRESS
8247  052154  022703  052410            2$:     CMP     #$TTYIN+8.,R3       ;;BUFFER FULL?
8248  052160  101456                            BLOS    4$                  ;;BR IF YES
8249  052162  104410                            RDCHR                       ;;GO READ ONE CHARACTER FROM THE TTY
8250  052164  112613                            MOVB    (SP)+,(R3)          ;;GET CHARACTER
8251  052166  122713  000177            10$:    CMPB    #177,(R3)           ;;IS IT A RUBOUT
8252  052172  001022                            BNE     5$                  ;;BR IF NO
8253  052174  005716                            TST     (SP)                ;;IS THIS THE FIRST RUBOUT?
8254  052176  001007                            BNE     6$                  ;;BR IF NO
8255  052200  112737  000134  052376            MOVB    #'\,9$              ;;TYPE A BACK SLASH
8256  052206  104401  052376                    TYPE    ,9$
8257  052212  012716  177777                    MOV     #-1,(SP)            ;;SET THE RUBOUT KEY
8258  052216  005303                    6$:     DEC     R3                  ;;BACKUP BY ONE
8259  052220  020327  052400                    CMP     R3,#$TTYIN          ;;STACK EMPTY?
8260  052224  103434                            BLO     4$                  ;;BR IF YES
8261  052226  111337  052376                    MOVB    (R3),9$             ;;SETUP TO TYPEOUT THE DELETED CHAR.
8262  052232  104401  052376                    TYPE    ,9$                 ;;GO TYPE
8263  052236  000746                            BR      2$                  ;;GO READ ANOTHER CHAR.
8264  052240  005716                    5$:     TST     (SP)                ;;RUBOUT KEY SET?
8265  052242  001406                            BEQ     7$                  ;;BR IF NO
8266  052244  112737  000134  052376            MOVB    #'\,9$              ;;TYPE A BACK SLASH
8267  052252  104401  052376                    TYPE    ,9$
```

```
8268  052256  005016                          CLR     (SP)            ;;CLEAR THE RUBOUT KEY
8269  052260  122713  000025          7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
8270  052264  001003                          BNE     8$              ;;BR IF NO
8271  052266  104401  052415                  TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
8272  052272  000726                          BR      1$              ;;GO START OVER
8273  052274  122713  000022          8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
8274  052300  001011                          BNE     3$              ;;BRANCH IF NO
8275  052302  105013                          CLRB    (R3)            ;;CLEAR THE CHARACTER
8276  052304  104401  001211                  TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
8277  052310  104401  052400                  TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
8278  052314  000717                          BR      2$              ;;GO PICKUP ANOTHER CHACTER
8279  052316  104401  001210          4$:     TYPE    ,$QUES          ;;TYPE A '?'
8280  052322  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
8281  052324  111337  052376          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
8282  052330  104401  052376                  TYPE    ,9$
8283  052334  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
8284  052340  001305                          BNE     2$              ;;LOOP IF NOT RETURN
8285  052342  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
8286  052346  104401  001212                  TYPE    ,$LF            ;;TYPE A LINE FEED
8287  052352  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
8288  052354  012603                          MOV     (SP)+,R3        ;;RESTORE R3
8289  052356  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
8290  052360  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
8291  052366  012766  052400  000004          MOV     #$TTYIN,4(SP)
8292  052374  000002                          RTI                     ;;RETURN
8293  052376     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
8294  052377     000                          .BYTE   0               ;;TERMINATOR
8295  052400  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
8296  052410  041536  005015     000  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
8297  052415     136  006525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
8298  052422  043536  005015     000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
8299  052427     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
8300  052434  036440  000040
8301  052440  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
8302  052446  020075     000
8303          052452                  .EVEN
8304                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
8305
8306                                  ;;*****************************************************************
8307                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
8308                                  ;*CHANGE IT TO BINARY.
8309                                  ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
8310                                  ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
8311                                  ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
8312                                  ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
8313                                  ;*CALL:
8314                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
8315                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
8316                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
8317
8318  052452  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
8319  052454  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
8320  052462  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8321  052464  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8322  052466  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8323  052470  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
```

```
8324   052472  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
8325   052474  010037  052600          MOV     R0,5$           ;;AND SAVE IT
8326   052500  005001                  CLR     R1              ;;CLEAR DATA WORD
8327   052502  005002                  CLR     R2
8328   052504  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
8329   052506  001420                  BEQ     3$              ;;IF ZERO GET OUT
8330   052510  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
8331   052514  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
8332   052516  122716  000067          CMPB    #'7,(SP)
8333   052522  002423                  BLT     4$
8334   052524  006301                  ASL     R1              ;;*2
8335   052526  006102                  ROL     R2
8336   052530  006301                  ASL     R1              ;;*4
8337   052532  006102                  ROL     R2
8338   052534  006301                  ASL     R1              ;;*8
8339   052536  006102                  ROL     R2
8340   052540  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
8341   052544  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
8342   052546  000756                  BR      2$              ;;LOOP
8343   052550  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
8344   052552  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
8345   052556  010237  052610          MOV     R2,$HIOCT
8346   052562  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
8347   052564  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
8348   052566  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
8349   052570  000002                  RTI                     ;;RETURN
8350   052572  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
8351   052574  105010                  CLRB    (R0)            ;;SET A TERMINATOR
8352   052576  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
8353   052600  000000          5$:     .WORD   0
8354   052602  104401  001210          TYPE    ,$QUES          ;;"?" "CR" & "LF"
8355   052606  000730                  BR      1$              ;;TRY AGAIN
8356   052610  000000          $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
8357                                  .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
8358
8359                                  ;;******************************************************************
8360                                  ;*SAVE R0-R5
8361                                  ;*CALL:
8362                                  ;*      SAVREG
8363                                  ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
8364                                  ;*
8365                                  ;*TOP---(+16)
8366                                  ;* +2---(+18)
8367                                  ;* +4---R5
8368                                  ;* +6---R4
8369                                  ;* +8---R3
8370                                  ;*+10---R2
8371                                  ;*+12---R1
8372                                  ;*+14---R0
8373
8374   052612                         $SAVREG:
8375   052612  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
8376   052614  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
8377   052616  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
8378   052620  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
8379   052622  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
```

```
8380  052624  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
8381  052626  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
8382  052632  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
8383  052636  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
8384  052642  016646  000022                MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
8385  052646  000002                        RTI
8386
8387                                  ;*RESTORE R0-R5
8388                                  ;*CALL:
8389                                  ;*      RESREG
8390  052650                          $RESREG:
8391  052650  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
8392  052654  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
8393  052660  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
8394  052664  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
8395  052670  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
8396  052672  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
8397  052674  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
8398  052676  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
8399  052700  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
8400  052702  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
8401  052704  000002                        RTI
8402
8403                                  .SBTTL  POWER DOWN AND UP ROUTINE
8404
8405                                  ;;**************************************************************************
8406
8407                                  ;POWER DOWN ROUTINE
8408  052706  017737  126226  003634  $PWRDN: MOV     @SWR,SAVSWR     ;SAVE SWITCH REGISTER
8409  052714  012737  052734  000024          MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
8410  052722  012737  000340  000026          MOV     #PR7,PWRVEC+2
8411  052730  000000                        HALT
8412  052732  000776                        BR      .-2     ;HANG UP
8413
8414                                  ;;**************************************************************************
8415
8416                                  ;POWER UP ROUTINE
8417  052734  005037  053030          $PWRUP: CLR     $PWRCT          ;LOAD WAIT COUNT
8418  052740  012737  000144  053032          MOV     #100.,$PWRCT+2
8419  052746  005237  053030          1$:     INC     $PWRCT          ;WAIT FOR TELETYPE
8420  052752  001375                        BNE     1$
8421  052754  005337  053032                DEC     $PWRCT+2
8422  052760  001372                        BNE     1$
8423  052762  012737  052706  000024          MOV     #$PWRDN,PWRVEC  ;SET UP FOR POWER DOWN VECTOR
8424  052770  012737  000340  000026          MOV     #PR7,PWRVEC+2
8425  052776  012706  001100                MOV     #STACK,SP       ;FORCE STACK
8426  053002  104401  053034                TYPE    ,$POWER         ;TYPE POWER
8427  053006  004737  045002                JSR     PC,PARCHK       ;REINITIALIZE MEMORY CHECK ENABLE
8428  053012  013777  003634  126120        MOV     SAVSWR,@SWR     ;RESTORE SWITCH REGISTER
8429  053020  013702  001270                MOV     $BASE,R2        ;REINITIALISE R2 FOR '611 BASE
8430  053024  000177  126056                JMP     @$LPADR ;GO BACK TO LAST TEST
8431
8432  053030  000000  000000          $PWRCT: .WORD   0,0             ;TELETYPE TIME OUT
8433  053034  047520  042527  000122  $POWER: .ASCIZ  /POWER/
8434                                          .EVEN
8435                                  .SBTTL  TRAP DECODER
```

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

053042  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
053044  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
053050  005740                          TST     -(R0)           ;;BACKUP BY 2
053052  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
053054  006300                          ASL     R0              ;;POSITION FOR INDEXING
053056  016000  053076                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
053062  000200                          RTS     R0              ;;GO TO ROUTINE


                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

053064  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
053066  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
053074  000002                          RTI                     ;;RESTORE THE PSW


                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
053076  053064                  $TRPAD: .WORD   $TRAP2
053100  050260                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
053102  050566                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
053104  050542                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
053106  050602                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
053110  050770                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

053112  051602                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

053114  051512                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
053116  052054                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
053120  052144                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
053122  052452                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
053124  052612                          $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
053126  052650                          $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
053130  047226                          SCOP1$  ;;CALL=SCOP1    TRAP+15(104415) INTERNAL LOOP ON ERROR
```

```
 8481                                          .SBTTL  DATA TABLE FOR PRINT OUT
 8482
 8483  053132  001220  003604                  DTO00:  .WORD   $TESTN,TRAPPC
 8484  053136  001220  001116  003500          DTO01:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.MR2,T.MR2,E.MR3,T.MR3
 8485  053144  003440  003526  003466
 8486  053152  003530  003470
 8487  053156  001220  001116  003500          DTO15:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1
 8488  053164  003440
 8489  053166  001220  001116  003524          DTO17:  .WORD   $TESTN,$ERRPC,E.MR1,T.MR1,PR.BIT,M1.BIT,BITCNT
 8490  053174  003464  003614  003616
 8491  053202  003622
 8492  053204  001220  001116  003500          DTO22:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.BA,T.BA,E.WC,T.WC
 8493  053212  003440  003504  003444
 8494  053220  003502  003442
 8495  053224  001220  001116  003524          DTO25:  .WORD   $TESTN,$ERRPC,E.MR1,T.MR1,BITCNT
 8496  053232  003464  003622
 8497  053236  001220  001116  003500          DTO31:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.BA,T.BA,E.WC,T.WC,E.MR1,T.MR1
 8498  053244  003440  003504  003444
 8499  053252  003502  003442  003524
 8500  053260  003464
 8501  053262  001220  001116  003500          DTO35:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,E.BA,T.BA
 8502  053270  003440  003510  003450
 8503  053276  003504  003444
 8504  053302  003502  003442                          .WORD   E.WC,T.WC
 8505  053306  001220  001116  003522          DTO41:  .WORD   $TESTN,$ERRPC,E.DB,T.DB
 8506  053314  003462
 8507  053316  001220  001116  003500          DTO42:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2
 8508  053324  003440  003510  003450
 8509  053332  001220  001116  003522          DTO54:  .WORD   $TESTN,$ERRPC,E.DB,T.DB,WRDCNT
 8510  053340  003462  003624
 8511  053344  001220  001116  003500          DTO55:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,WRDCNT
 8512  053352  003440  003510  003450
 8513  053360  003624
 8514  053362  001220  001116  003500          DTO72:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,E.ER,T.ER
 8515  053370  003440  003510  003450
 8516  053376  003514  003454
 8517  053402  003504  003444  003502                  .WORD   E.BA,T.BA,E.WC,T.WC
 8518  053410  003442
 8519  053412  001220  001116  003500          DTO77:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2
 8520  053420  003440  003510  003450
 8521  053426  001220  001116  003500          DT150:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,WRDCNT,BITCNT
 8522  053434  003440  003624  003622
 8523  053442  001220  001116  003500          DT164:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS1,BITCNT
 8524  053450  003440  003510  003440
 8525  053456  003622
 8526  053460  001220  001116  003524          DT170:  .WORD   $TESTN,$ERRPC,E.MR1,T.MR1,P1.BIT,PR.BIT,M1.BIT,M2.BIT,BITCNT,SECCNT
 8527  053466  003464  003612  003614
 8528  053474  003616  003620  003622
 8529  053502  003626
 8530  053504  001220  001116  003524          DT171:  .WORD   $TESTN,$ERRPC,E.MR1,T.MR1
 8531  053512  003464
 8532  053514  001220  001116  003500          DT175:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,E.DS,T.DS,E.ER,T.ER
 8533  053522  003440  003510  003450
 8534  053530  003512  003452  003514
 8535  053536  003454
 8536  053540  001220  001116  003500          DT211:  .WORD   $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,E.DS,T.DS,E.ER,T.ER
```

```
 8537  053546  003440  003510  003450
 8538  053554  003512  003452  003514
 8539  053562  003454
 8540  053564  003540  003550  003552          .WORD   P.CS1,P.CS2,P.DS,P.ER
 8541  053572  003554
```

```
                                        .SBTTL  DATA FORMAT FOR PRINT OUT
                                
053574 000001                   DF000:  .WORD   1
053576     002  000                    .BYTE   2,0
053600 000005                   DF001:  .WORD   5                ;ERRORS 1-14
053602     000  000                    .BYTE   0,0
053604 054760                           .WORD   DH000A
053606     000  000                    .BYTE   0,0
053610 054776                           .WORD   DH000B
053612     002  000                    .BYTE   2,0
053614 055042                           .WORD   DH001A
053616     000  000                    .BYTE   0,0
053620 055121                           .WORD   DH001B
053622     006  000                    .BYTE   6,0
053624 000005                   DF015:  .WORD   5                ;ERRORS 15-16
053626     000  000                    .BYTE   0,0
053630 054760                           .WORD   DH000A
053632     000  000                    .BYTE   0,0
053634 054776                           .WORD   DH000B
053636     002  000                    .BYTE   2,0
053640 055200                           .WORD   DH015A
053642     000  000                    .BYTE   0,0
053644 055217                           .WORD   DH015B
053646     002  000                    .BYTE   2,0
053650 000005                   DF017:  .WORD   5                ;ERROR 17-21
053652     000  000                    .BYTE   0,0
053654 054760                           .WORD   DH000A
053656     000  000                    .BYTE   0,0
053660 054776                           .WORD   DH000B
053662     002  000                    .BYTE   2,0
053664 055235                           .WORD   DH017A
053666     000  000                    .BYTE   0,0
053670 055301                           .WORD   DH017B
053672     005  000                    .BYTE   5,0
053674 000005                   DF022:  .WORD   5                ;ERROR 22-24
053676     000  000                    .BYTE   0,0
053700 054760                           .WORD   DH000A
053702     000  000                    .BYTE   0,0
053704 054776                           .WORD   DH000B
053706     002  000                    .BYTE   2,0
053710 055347                           .WORD   DH022A
053712     000  000                    .BYTE   0,0
053714 055426                           .WORD   DH022B
053716     006  000                    .BYTE   6,0
053720 000005                   DF025:  .WORD   5                ;ERROR 25
053722     000  000                    .BYTE   0,0
053724 054760                           .WORD   DH000A
053726     000  000                    .BYTE   0,0
053730 054776                           .WORD   DH000B
053732     002  000                    .BYTE   2,0
053734 055503                           .WORD   DH025A
053736     000  000                    .BYTE   0,0
053740 055527                           .WORD   DH025B
053742     003  000                    .BYTE   3,0
053744 000005                   DF031:  .WORD   5                ;ERROR 31-34
053746     000  000                    .BYTE   0,0
```

```
8598   053750  054760                  .WORD   DH000A
8599   053752     000  000             .BYTE   0,0
8600   053754  054776                  .WORD   DH000B
8601   053756     002  000             .BYTE   2,0
8602   053760  055555                  .WORD   DH031A
8603   053762     000  000             .BYTE   0,0
8604   053764  055654                  .WORD   DH031B
8605   053766     010  000             .BYTE   8.,0
8606   053770  000005          DF035:  .WORD   5               ;ERRORS 35-40
8607   053772     000  000             .BYTE   0,0
8608   053774  054760                  .WORD   DH000A
8609   053776     000  000             .BYTE   0,0
8610   054000  054776                  .WORD   DH000B
8611   054002     002  000             .BYTE   2,0
8612   054004  055752                  .WORD   DH035A
8613   054006     000  000             .BYTE   0,0
8614   054010  056051                  .WORD   DH035B
8615   054012     010  000             .BYTE   8.,0
8616   054014  000005          DF041:  .WORD   5               ;ERROR 41
8617   054016     000  000             .BYTE   0,0
8618   054020  054760                  .WORD   DH000A
8619   054022     000  000             .BYTE   0,0
8620   054024  054776                  .WORD   DH000B
8621   054026     002  000             .BYTE   2,0
8622   054030  056146                  .WORD   DH041A
8623   054032     000  000             .BYTE   0,0
8624   054034  056163                  .WORD   DH041B
8625   054036     002  000             .BYTE   2,0
8626   054040  000005          DF042:  .WORD   5               ;ERRORS 42-43
8627   054042     000  000             .BYTE   0,0
8628   054044  054760                  .WORD   DH000A
8629   054046     000  000             .BYTE   0,0
8630   054050  054776                  .WORD   DH000B
8631   054052     002  000             .BYTE   2,0
8632   054054  056200                  .WORD   DH042A
8633   054056     000  000             .BYTE   0,0
8634   054060  056237                  .WORD   DH042B
8635   054062     004  000             .BYTE   4,0
8636   054064  000005          DF054:  .WORD   5               ;ERROR 54
8637   054066     000  000             .BYTE   0,0
8638   054070  054760                  .WORD   DH000A
8639   054072     000  000             .BYTE   0,0
8640   054074  054776                  .WORD   DH000B
8641   054076     002  000             .BYTE   2,0
8642   054100  056275                  .WORD   DH054A
8643   054102     000  000             .BYTE   0,0
8644   054104  056322                  .WORD   DH054B
8645   054106     003  000             .BYTE   3,0
8646   054110  000005          DF055:  .WORD   5               ;ERROR 55
8647   054112     000  000             .BYTE   0,0
8648   054114  054760                  .WORD   DH000A
8649   054116     000  000             .BYTE   0,0
8650   054120  054776                  .WORD   DH000B
8651   054122     002  000             .BYTE   2,0
8652   054124  056350                  .WORD   DH055A
8653   054126     000  000             .BYTE   0,0
```

```
8654  054130  056415                    .WORD   DH055B
8655  054132     005  000               .BYTE   5,0
8656  054134  000007          DF072:    .WORD   7
8657  054136     000  000               .BYTE   0,0
8658  054140  054760                    .WORD   DH000A
8659  054142     000  000               .BYTE   0,0
8660  054144  054776                    .WORD   DH000B
8661  054146     002  000               .BYTE   2,0
8662  054150  056463                    .WORD   DH072A
8663  054152     000  000               .BYTE   0,0
8664  054154  056522                    .WORD   DH072B
8665  054156     004  000               .BYTE   4,0
8666  054160  056560                    .WORD   DH072C
8667  054162     000  000               .BYTE   0,0
8668  054164  056637                    .WORD   DH072D
8669  054166     006  000               .BYTE   6,0
8670  054170  000007          DF077:    .WORD   7
8671  054172     000  000               .BYTE   0,0
8672  054174  054760                    .WORD   DH000A
8673  054176     000  000               .BYTE   0,0
8674  054200  054776                    .WORD   DH000B
8675  054202     002  000               .BYTE   2,0
8676  054204  056463                    .WORD   DH072A
8677  054206     000  000               .BYTE   0,0
8678  054210  056522                    .WORD   DH072B
8679  054212     004  000               .BYTE   4,0
8680  054214  000005          DF150:    .WORD   5                       ;ERROR 150
8681  054216     000  000               .BYTE   0,0
8682  054220  054760                    .WORD   DH000A
8683  054222     000  000               .BYTE   0,0
8684  054224  054776                    .WORD   DH000B
8685  054226     002  000               .BYTE   2,0
8686  054230  056714                    .WORD   DH150A
8687  054232     000  000               .BYTE   0,0
8688  054234  056750                    .WORD   DH150B
8689  054236     004  000               .BYTE   4,0
8690  054240  000005          DF164:    .WORD   5                       ;ERROR 164
8691  054242     000  000               .BYTE   0,0
8692  054244  054760                    .WORD   DH000A
8693  054246     000  000               .BYTE   0,0
8694  054250  054776                    .WORD   DH000B
8695  054252     002  000               .BYTE   2,0
8696  054254  057006                    .WORD   DH164A
8697  054256     000  000               .BYTE   0,0
8698  054260  057052                    .WORD   DH164B
8699  054262     005  000               .BYTE   5,0
8700  054264  000005          DF170:    .WORD   5           ;ERROR 170
8701  054266     000  000               .BYTE   0,0
8702  054270  054760                    .WORD   DH000A
8703  054272     000  000               .BYTE   0,0
8704  054274  054776                    .WORD   DH000B
8705  054276     002  000               .BYTE   2,0
8706  054300  057120                    .WORD   DH170A
8707  054302     000  000               .BYTE   0,0
8708  054304  057217                    .WORD   DH170B
8709  054306     010  000               .BYTE   8.,0
```

```
8710  054310  000005          DF171:  .WORD   5               ;ERRORS 171-174
8711  054312     000  000             .BYTE   0,0
8712  054314  054760                  .WORD   DH000A
8713  054316     000  000             .BYTE   0,0
8714  054320  054776                  .WORD   DH000B
8715  054322     002  000             .BYTE   2,0
8716  054324  057315                  .WORD   DH171A
8717  054326     000  000             .BYTE   0,0
8718  054330  057334                  .WORD   DH171B
8719  054332     002  000             .BYTE   2,0
8720  054334  000005          DF175:  .WORD   5               ;ERRORS 175-210
8721  054336     000  000             .BYTE   0,0
8722  054340  054760                  .WORD   DH000A
8723  054342     000  000             .BYTE   0,0
8724  054344  054776                  .WORD   DH000B
8725  054346     002  000             .BYTE   2,0
8726  054350  057352                  .WORD   DH175A
8727  054352     000  000             .BYTE   0,0
8728  054354  057451                  .WORD   DH175B
8729  054356     010  000             .BYTE   8.,0
8730  054360  000007          DF211:  .WORD   7               ;ERRORS 211-214
8731  054362     000  000             .BYTE   0,0
8732  054364  054760                  .WORD   DH000A
8733  054366     000  000             .BYTE   0,0
8734  054370  054776                  .WORD   DH000B
8735  054372     002  000             .BYTE   2,0
8736  054374  057352                  .WORD   DH175A
8737  054376     000  000             .BYTE   0,0
8738  054400  057451                  .WORD   DH175B
8739  054402     010  000             .BYTE   8.,0
8740  054404  057546                  .WORD   DH211A
8741  054406     000  000             .BYTE   0,0
8742  054410  057574                  .WORD   DH211B
8743  054412     004  000             .BYTE   4,0
```

```
8744                                    .SBTTL  ASCII MESSAGES
8745
8746  054414  005015  045522  030466  OPR001: .ASCIZ  <15><12>/RK611 VECTOR ADDRESS ( /
8747  054422  020061  042526  052103
8748  054430  051117  040440  042104
8749  054436  042522  051523  024040
8750  054444  000040
8751  054446  024440  036440  000040  OPR002: .ASCIZ  / ) = /
8752  054454  045522  030466  020061  OPR003: .ASCIZ  /RK611 VECTOR ADDRESS ( /
8753  054462  042526  052103  051117
8754  054470  040440  042104  042522
8755  054476  051523  024040  000040
8756  054504  045522  030466  020061  OPR004: .ASCIZ  /RK611 PRIORITY ( /
8757  054512  051120  047511  044522
8758  054520  054524  024040  000040
8759  054526  005015  047062  020104  OPR006: .ASCIZ  <15><12>/2ND PASS RUN TIME IS APPROX 8 MINUTES/<15><12>
8760  054534  040520  051523  051040
8761  054542  047125  052040  046511
8762  054550  020105  051511  040440
8763  054556  050120  047522  020130
8764  054564  020070  044515  052516
8765  054572  042524  006523  000012
8766  054600  005015  025052  025052  OPR007: .ASCIZ  <15><12>/*****   PROGRAM HALTED   ******/<15><12>
8767  054606  020052  020040  051120
8768  054614  043517  040522  020115
8769  054622  040510  052114  042105
8770  054630  020040  025040  025052
8771  054636  025052  006452  000012
8772  054644  020040     000          SPACE2: .ASCIZ  /  /
8773  054647     015  050012  047522  ABORT:  .ASCIZ  <15><12>/PROGRAM ABORTED BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
8774  054654  051107  046501  040440
8775  054662  047502  052122  042105
8776  054670  041040  041505  052501
8777  054676  042523  042440  051122
8778  054704  051117  052040  051110
8779  054712  051505  047510  042114
8780  054720  042440  041530  042505
8781  054726  042504  006504  000012
8782  054734  005015  042524  052123  TSTBY1: .ASCIZ  <15><12>/TEST /
8783  054742  000040
8784  054744  041040  050131  051501  TSTBY2: .ASCIZ  / BYPASSED/<15><12>
8785  054752  042523  006504  000012
```

```
                                        .SBTTL  DATA HEADERS
                                        
054760  042524  052123  020040  DH000A: .ASCIZ  /TEST    ERROR/
054766  020040  051105  047522
054774  000122
054776  052516  020115  020040  DH000B: .ASCIZ  /NUM     PC/
055004  020040  041520     000
055011     124  051505  020124  DH000C: .ASCII  /TEST    TRAP/<15><12>
055016  020040  052040  040522
055024  006520     012
055027     116  046525  020040          .ASCIZ  /NUM     PC/
055034  020040  050040  000103
055042  054105  042520  052103  DH001A: .ASCIZ  /EXPECT  ACUTAL  EXPECT  ACTUAL  EXPECT  ACTUAL/
055050  020040  041501  052125
055056  046101  020040  054105
055064  042520  052103  020040
055072  041501  052524  046101
055100  020040  054105  042520
055106  052103  020040  041501
055114  052524  046101     000
055121     122  041513  030523  DH001B: .ASCIZ  /RKCS1   RKCS1   MESS A  MESS A  MESS B  MESS B/
055126  020040  051040  041513
055134  030523  020040  046440
055142  051505  020123  020101
055150  046440  051505  020123
055156  020101  046440  051505
055164  020123  020102  046440
055172  051505  020123  000102
055200  054105  042520  052103  DH015A: .ASCIZ  /EXPECT  ACTUAL/
055206  020040  041501  052524
055214  046101     000
055217     122  041513  030523  DH015B: .ASCIZ  /RKCS1   RKCS1/
055224  020040  051040  041513
055232  030523     000
055235     105  050130  041505  DH017A: .ASCIZ  /EXPECT  ACTUAL  PRESENT PREVOUS BIT/
055242  020124  040440  052103
055250  040525  020114  050040
055256  042522  042523  052116
055264  050040  042522  047526
055272  051525  041040  052111
055300     000
055301     122  046513  030522  DH017B: .ASCIZ  /RKMR1   RKMR1   BIT     BIT     COUNT/
055306  020040  051040  046513
055314  030522  020040  041040
055322  052111  020040  020040
055330  041040  052111  020040
055336  020040  041440  052517
055344  052116     000
055347     105  050130  041505  DH022A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  EXPECT  ACTUAL/
055354  020124  040440  052103
055362  040525  020114  042440
055370  050130  041505  020124
055376  040440  052103  040525
055404  020114  042440  050130
055412  041505  020124  040440
055420  052103  040525  000114
```

```
8842  055426  045522  051503  020061  DH022B: .ASCIZ  /RKCS1   RKCS1   RKBA    RKBA    RKWC    RKWC/
8843  055434  020040  045522  051503
8844  055442  020061  020040  045522
8845  055450  040502  020040  020040
8846  055456  045522  040502  020040
8847  055464  020040  045522  041527
8848  055472  020040  020040  045522
8849  055500  041527     000
8850  055503     105  050130  041505  DH025A: .ASCIZ  /EXPECT  ACTUAL  BIT/
8851  055510  020124  040440  052103
8852  055516  040525  020114  041040
8853  055524  052111     000
8854  055527     122  046513  030522  DH025B: .ASCIZ  /RKMR1   RKMR1   COUNT/
8855  055534  020040  051040  046513
8856  055542  030522  020040  041440
8857  055550  052517  052116     000
8858  055555     105  050130  041505  DH031A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  EXPECT  ACTUAL  EXPECT  ACTUAL/
8859  055562  020124  040440  052103
8860  055570  040525  020114  042440
8861  055576  050130  041505  020124
8862  055604  040440  052103  040525
8863  055612  020114  042440  050130
8864  055620  041505  020124  040440
8865  055626  052103  040525  020114
8866  055634  042440  050130  041505
8867  055642  020124  040440  052103
8868  055650  040525  000114
8869  055654  045522  051503  020061  DH031B: .ASCIZ  /RKCS1   RKCS1   RKBA    RKBA    RKWC    RKWC    RKMR1   RKMR1/
8870  055662  020040  045522  051503
8871  055670  020061  020040  045522
8872  055676  040502  020040  020040
8873  055704  045522  040502  020040
8874  055712  020040  045522  041527
8875  055720  020040  020040  045522
8876  055726  041527  020040  020040
8877  055734  045522  051115  020061
8878  055742  020040  045522  051115
8879  055750  000061
8880  055752  054105  042520  052103  DH035A: .ASCII  /EXPECT  ACTUAL  EXPECT  ACTUAL  EXPECT  ACTUAL/
8881  055760  020040  041501  052524
8882  055766  046101  020040  054105
8883  055774  042520  052103  020040
8884  056002  041501  052524  046101
8885  056010  020040  054105  042520
8886  056016  052103  020040  041501
8887  056024  052524  046101
8888  056030  020040  054105  042520          .ASCIZ  /  EXPECT  ACTUAL/
8889  056036  052103  020040  041501
8890  056044  052524  046101     000
8891  056051     122  041513  030523  DH035B: .ASCII  /RKCS1   RKCS1   RKCS2   RKCS2   RKBA    RKBA/
8892  056056  020040  051040  041513
8893  056064  030523  020040  051040
8894  056072  041513  031123  020040
8895  056100  051040  041513  031123
8896  056106  020040  051040  041113
8897  056114  020101  020040  051040
```

```
8898  056122  041113     101
8899  056125     040  020040  051040          .ASCIZ  /    RKWC    RKWC/
8900  056132  053513  020103  020040
8901  056140  051040  053513  000103
8902  056146  054105  042520  052103  DH041A: .ASCIZ  /EXPECT  WORD/
8903  056154  020040  047527  042122
8904  056162     000
8905  056163     127  051117  020104  DH041B: .ASCIZ  /WORD    READ/
8906  056170  020040  051040  040505
8907  056176  000104
8908  056200  054105  042520  052103  DH042A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL/
8909  056206  020040  041501  052524
8910  056214  046101  020040  054105
8911  056222  042520  052103  020040
8912  056230  041501  052524  046101
8913  056236     000
8914  056237     122  041513  030523  DH042B: .ASCIZ  /RKCS1   RKCS1   RKCS2   RKCS2/
8915  056244  020040  051040  041513
8916  056252  030523  020040  051040
8917  056260  041513  031123  020040
8918  056266  051040  041513  031123
8919  056274     000
8920  056275     105  050130  041505  DH054A: .ASCIZ  /EXPECT  WORD    WORD/
8921  056302  020124  053440  051117
8922  056310  020104  020040  053440
8923  056316  051117  000104
8924  056322  047527  042122  020040  DH054B: .ASCIZ  /WORD    READ    COUNT/
8925  056330  020040  042522  042101
8926  056336  020040  020040  047503
8927  056344  047125  000124
8928  056350  054105  042520  052103  DH055A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  WORD/
8929  056356  020040  041501  052524
8930  056364  046101  020040  054105
8931  056372  042520  052103  020040
8932  056400  041501  052524  046101
8933  056406  020040  047527  042122
8934  056414     000
8935  056415     122  041513  030523  DH055B: .ASCIZ  /RKCS1   RKCS1   RKCS2   RKCS2   COUNT/
8936  056422  020040  051040  041513
8937  056430  030523  020040  051040
8938  056436  041513  031123  020040
8939  056444  051040  041513  031123
8940  056452  020040  041440  052517
8941  056460  052116     000
8942  056463     105  050130  041505  DH072A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL/
8943  056470  020124  040440  052103
8944  056476  040525  020114  042440
8945  056504  050130  041505  020124
8946  056512  040440  052103  040525
8947  056520  000114
8948  056522  045522  051503  020061  DH072B: .ASCIZ  /RKCS1   RKCS1   RKCS2   RKCS2/
8949  056530  020040  045522  051503
8950  056536  020061  020040  045522
8951  056544  051503  020062  020040
8952  056552  045522  051503  000062
8953  056560  054105  042520  052103  DH072C: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  EXPECT  ACTUAL/
```

```
8954  056566  020040  041501  052524
8955  056574  046101  020040  054105
8956  056602  042520  052103  020040
8957  056610  041501  052524  046101
8958  056616  020040  054105  042520
8959  056624  052103  020040  041501
8960  056632  052524  046101     000
8961  056637     122  042513  020122  DH072D: .ASCIZ  /RKER    RKER    RKBA    RKBA    RKWC    RKWC/
8962  056644  020040  051040  042513
8963  056652  020122  020040  051040
8964  056660  041113  020101  020040
8965  056666  051040  041113  020101
8966  056674  020040  051040  053513
8967  056702  020103  020040  051040
8968  056710  053513  000103
8969  056714  054105  042520  052103  DH150A: .ASCIZ  /EXPECT  ACTUAL  WORD    BIT/
8970  056722  020040  041501  052524
8971  056730  046101  020040  047527
8972  056736  042122  020040  020040
8973  056744  044502  000124
8974  056750  045522  051503  020061  DH150B: .ASCIZ  /RKCS1   RKCS1   COUNT   COUNT/
8975  056756  020040  045522  051503
8976  056764  020061  020040  047503
8977  056772  047125  020124  020040
8978  057000  047503  047125  000124
8979  057006  054105  042520  052103  DH164A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  BIT/
8980  057014  020040  041501  052524
8981  057022  046101  020040  054105
8982  057030  042520  052103  020040
8983  057036  041501  052524  046101
8984  057044  020040  044502  000124
8985  057052  045522  051503  020061  DH164B: .ASCIZ  /RKCS1   RKCS1   RKCS2   RKCS2   COUNT/
8986  057060  020040  045522  051503
8987  057066  020061  020040  045522
8988  057074  051503  020062  020040
8989  057102  045522  051503  020062
8990  057110  020040  047503  047125
8991  057116  000124
8992  057120  054105  042520  052103  DH170A: .ASCIZ  /EXPECT  ACTUAL  PRESENT PRESENT PRESENT PRESENT BIT     SECTOR/
8993  057126  020040  041501  052524
8994  057134  046101  020040  051120
8995  057142  051505  047105  020124
8996  057150  051120  051505  047105
8997  057156  020124  051120  051505
8998  057164  047105  020124  051120
8999  057172  051505  047105  020124
9000  057200  044502  020124  020040
9001  057206  020040  042523  052103
9002  057214  051117     000
9003  057217     122  046513  030522  DH170B: .ASCIZ  /RKMR1   RKMR1   BIT+1   BIT     BIT-1   BIT-2   COUNT   COUNT/
9004  057224  020040  051040  046513
9005  057232  030522  020040  041040
9006  057240  052111  030453  020040
9007  057246  041040  052111  020040
9008  057254  020040  041040  052111
9009  057262  030455  020040  041040
```

```
9010  057270  052111  031055  020040
9011  057276  041440  052517  052116
9012  057304  020040  041440  052517
9013  057312  052116     000
9014  057315     105  050130  041505  DH171A: .ASCIZ  /EXPECT  ACTUAL/
9015  057322  020124  040440  052103
9016  057330  040525  000114
9017  057334  045522  051115  020061  DH171B: .ASCIZ  /RKMR1   RKMR1/
9018  057342  020040  045522  051115
9019  057350  000061
9020  057352  054105  042520  052103  DH175A: .ASCIZ  /EXPECT  ACTUAL  EXPECT  ACTUAL  EXPECT  ACTUAL  EXPECT  ACTUAL/
9021  057360  020040  041501  052524
9022  057366  046101  020040  054105
9023  057374  042520  052103  020040
9024  057402  041501  052524  046101
9025  057410  020040  054105  042520
9026  057416  052103  020040  041501
9027  057424  052524  046101  020040
9028  057432  054105  042520  052103
9029  057440  020040  041501  052524
9030  057446  046101     000
9031  057451     122  041513  030523  DH175B: .ASCIZ  /RKCS1   RKCS1   RKCS2   RKCS2   RKDS    RKDS    RKER    RKER/
9032  057456  020040  051040  041513
9033  057464  030523  020040  051040
9034  057472  041513  031123  020040
9035  057500  051040  041513  031123
9036  057506  020040  051040  042113
9037  057514  020123  020040  051040
9038  057522  042113  020123  020040
9039  057530  051040  042513  020122
9040  057536  020040  051040  042513
9041  057544  000122
9042  057546  051120  053105  047511  DH211A: .ASCIZ  /PREVIOUS REG CONTENTS/
9043  057554  051525  051040  043505
9044  057562  041440  047117  042524
9045  057570  052116  000123
9046  057574  045522  051503  020061  DH211B: .ASCIZ  /RKCS1   RKCS2   RKDS    RKER/
9047  057602  020040  045522  051503
9048  057610  020062  020040  045522
9049  057616  051504  020040  020040
9050  057624  045522  051105     000
```

```
9051                                    .SBTTL  ERROR MESSAGES
9052
9053  057631     125  042516  050130  EM000:  .ASCIZ  /UNEXPECTED MEMORY PARITY ENABLE TRAP/
9054  057636  041505  042524  020104
9055  057644  042515  047515  054522
9056  057652  050040  051101  052111
9057  057660  020131  047105  041101
9058  057666  042514  052040  040522
9059  057674  000120
9060  057676  052101  042524  050115  EM200:  .ASCIZ  /ATTEMPTING TO CHECK SEEK MESSAGE FROM READ HEADER/
9061  057704  044524  043516  052040
9062  057712  020117  044103  041505
9063  057720  020113  042523  045505
9064  057726  046440  051505  040523
9065  057734  042507  043040  047522
9066  057742  020115  042522  042101
9067  057750  044040  040505  042504
9068  057756  000122
9069  057760  052101  042524  050115  EM201:  .ASCIZ  /ATTEMPTING TO CHECK SEEK MESSAGE FROM WRITE HEADER/
9070  057766  044524  043516  052040
9071  057774  020117  044103  041505
9072  060002  020113  042523  045505
9073  060010  046440  051505  040523
9074  060016  042507  043040  047522
9075  060024  020115  051127  052111
9076  060032  020105  042510  042101
9077  060040  051105     000
9078  060043     101  052124  046505  EM202:  .ASCIZ  /ATTEMPTING TO CHECK CLEAR DRIVE MESSAGE FROM READ HEADER/
9079  060050  052120  047111  020107
9080  060056  047524  041440  042510
9081  060064  045503  041440  042514
9082  060072  051101  042040  044522
9083  060100  042526  046440  051505
9084  060106  040523  042507  043040
9085  060114  047522  020115  042522
9086  060122  042101  044040  040505
9087  060130  042504  000122
9088  060134  052101  042524  050115  EM203:  .ASCIZ  /ATTEMPTING TO CHECK CLEAR DRIVE MESSAGE FROM WRITE HEADER/
9089  060142  044524  043516  052040
9090  060150  020117  044103  041505
9091  060156  020113  046103  040505
9092  060164  020122  051104  053111
9093  060172  020105  042515  051523
9094  060200  043501  020105  051106
9095  060206  046517  053440  044522
9096  060214  042524  044040  040505
9097  060222  042504  000122
9098  060226  052101  042524  050115  EM204:  .ASCIZ  /ATTEMPTING A READ HEADER TO CHECK SECTOR PULSE DETECT/
9099  060234  044524  043516  040440
9100  060242  051040  040505  020104
9101  060250  042510  042101  051105
9102  060256  052040  020117  044103
9103  060264  041505  020113  042523
9104  060272  052103  051117  050040
9105  060300  046125  042523  042040
9106  060306  052105  041505  000124
```

```
9107  060314  052101  042524  050115  EM205:  .ASCIZ  /ATTEMPTING A WRITE HEADER TO CHECK INDEX PULSE DETECT/
9108  060322  044524  043516  040440
9109  060330  053440  044522  042524
9110  060336  044040  040505  042504
9111  060344  020122  047524  041440
9112  060352  042510  045503  044440
9113  060360  042116  054105  050040
9114  060366  046125  042523  042040
9115  060374  052105  041505  000124
9116  060402  052101  042524  050115  EM206:  .ASCIZ  /ATTEMPTING AN NPR READ OF ONE WORD/
9117  060410  044524  043516  040440
9118  060416  020116  050116  020122
9119  060424  042522  042101  047440
9120  060432  020106  047117  020105
9121  060440  047527  042122     000
9122  060445     101  052124  046505  EM207:  .ASCIZ  /ATTEMPTING AN NPR READ/
9123  060452  052120  047111  020107
9124  060460  047101  047040  051120
9125  060466  051040  040505  000104
9126  060474  052101  042524  050115  EM208:  .ASCIZ  /ATTEMPTING NPR READ CHECKING ZERO DETECT/
9127  060502  044524  043516  047040
9128  060510  051120  051040  040505
9129  060516  020104  044103  041505
9130  060524  044513  043516  055040
9131  060532  051105  020117  042504
9132  060540  042524  052103     000
9133  060545     101  052124  046505  EM209:  .ASCIZ  /ATTEMPTING NPR READ WITH BUS ADDRESS INCREMENT INHIBIT/
9134  060552  052120  047111  020107
9135  060560  050116  020122  042522
9136  060566  042101  053440  052111
9137  060574  020110  052502  020123
9138  060602  042101  051104  051505
9139  060610  020123  047111  051103
9140  060616  046505  047105  020124
9141  060624  047111  044510  044502
9142  060632  000124
9143  060634  052101  042524  050115  EM210:  .ASCII  /ATTEMPTING NPR READ WITH BUS ADDRESS INCREMENT INHIBIT/
9144  060642  044524  043516  047040
9145  060650  051120  051040  040505
9146  060656  020104  044527  044124
9147  060664  041040  051525  040440
9148  060672  042104  042522  051523
9149  060700  044440  041516  042522
9150  060706  042515  052116  044440
9151  060714  044116  041111  052111
9152  060722  041440  042510  045503          .ASCIZ  / CHECKING ZERO DETECT/
9153  060730  047111  020107  042532
9154  060736  047522  042040  052105
9155  060744  041505  000124
9156  060750  052101  042524  050115  EM211:  .ASCIZ  /ATTEMPTING TO FORCE NON-EXISTENT MEMORY/
9157  060756  044524  043516  052040
9158  060764  020117  047506  041522
9159  060772  020105  047516  026516
9160  061000  054105  051511  042524
9161  061006  052116  046440  046505
9162  061014  051117  000131
```

```
9163  061020  052101  042524  050115  EM212:  .ASCIZ  /ATTEMPTING TO CLEAR NON-EXISTENT MEMORY/
9164  061026  044524  043516  052040
9165  061034  020117  046103  040505
9166  061042  020122  047516  026516
9167  061050  054105  051511  042524
9168  061056  052116  046440  046505
9169  061064  051117  000131
9170  061070  042524  052123  047111  EM213:  .ASCIZ  /TESTING EXTENDED MEMORY ADDRESSING BITS/
9171  061076  020107  054105  042524
9172  061104  042116  042105  046440
9173  061112  046505  051117  020131
9174  061120  042101  051104  051505
9175  061126  044523  043516  041040
9176  061134  052111  000123
9177  061140  052101  042524  050115  EM214:  .ASCIZ  /ATTEMPTING TO FORCE UNIBUS PARITY ERROR/
9178  061146  044524  043516  052040
9179  061154  020117  047506  041522
9180  061162  020105  047125  041111
9181  061170  051525  050040  051101
9182  061176  052111  020131  051105
9183  061204  047522  000122
9184  061210  052101  042524  050115  EM215:  .ASCIZ  /ATTEMPTING NPR READ OF LOCATION PRIOR TO BAD PARITY/
9185  061216  044524  043516  047040
9186  061224  051120  051040  040505
9187  061232  020104  043117  046040
9188  061240  041517  052101  047511
9189  061246  020116  051120  047511
9190  061254  020122  047524  041040
9191  061262  042101  050040  051101
9192  061270  052111  000131
9193  061274  052101  042524  050115  EM216:  .ASCIZ  /ATTEMPTING TO CLEAR UNIBUS PARITY ERROR/
9194  061302  044524  043516  052040
9195  061310  020117  046103  040505
9196  061316  020122  047125  041111
9197  061324  051525  050040  051101
9198  061332  052111  020131  051105
9199  061340  047522  000122
9200  061344  052101  042524  050115  EM217:  .ASCIZ  /ATTEMPTING 18 BIT NPR READ/
9201  061352  044524  043516  030440
9202  061360  020070  044502  020124
9203  061366  050116  020122  042522
9204  061374  042101     000
9205  061377     101  052124  046505  EM218:  .ASCIZ  /ATTEMPTING 18 BIT NPR READ CHECKING ZERO DETECT/
9206  061404  052120  047111  020107
9207  061412  034061  041040  052111
9208  061420  047040  051120  051040
9209  061426  040505  020104  044103
9210  061434  041505  044513  043516
9211  061442  055040  051105  020117
9212  061450  042504  042524  052103
9213  061456     000
9214  061457     101  052124  046505  EM219:  .ASCIZ  /ATTEMPTING 18 BIT NPR READ WITH BIT 16 (PA) SET/
9215  061464  052120  047111  020107
9216  061472  034061  041040  052111
9217  061500  047040  051120  051040
9218  061506  040505  020104  044527
```

```
9219  061514  044124  041040  052111
9220  061522  030440  020066  050050
9221  061530  024501  051440  052105
9222  061536     000
9223  061537     101  052124  046505  EM220:  .ASCII  /ATTEMPTING 18 BIT NPR READ WITH BIT 16 (PA) SET/
9224  061544  052120  047111  020107
9225  061552  034061  041040  052111
9226  061560  047040  051120  051040
9227  061566  040505  020104  044527
9228  061574  044124  041040  052111
9229  061602  030440  020066  050050
9230  061610  024501  051440  052105
9231  061616  052101  042524  050115  EM221:  .ASCIZ  /ATTEMPTING SIMULATION OF DATA IN READ HEADER/
9232  061624  044524  043516  051440
9233  061632  046511  046125  052101
9234  061640  047511  020116  043117
9235  061646  042040  052101  020101
9236  061654  047111  051040  040505
9237  061662  020104  042510  042101
9238  061670  051105     000
9239  061673     101  052124  046505  EM222:  .ASCIZ  /ATTEMPTING READ HEADER IN MAINTANENCE MODE/
9240  061700  052120  047111  020107
9241  061706  042522  042101  044040
9242  061714  040505  042504  020122
9243  061722  047111  046440  044501
9244  061730  052116  047101  047105
9245  061736  042503  046440  042117
9246  061744  000105
9247  061746  052101  042524  050115  EM223:  .ASCIZ  /ATTEMPTING DATA BUFFER READ AFTER READ HEADER/
9248  061754  044524  043516  042040
9249  061762  052101  020101  052502
9250  061770  043106  051105  051040
9251  061776  040505  020104  043101
9252  062004  042524  020122  042522
9253  062012  042101  044040  040505
9254  062020  042504  000122
9255  062024  052101  042524  050115  EM224:  .ASCIZ  /ATTEMPTING SIMULATION OF DATA IN READ HEADER (18 BIT FORMAT)/
9256  062032  044524  043516  051440
9257  062040  046511  046125  052101
9258  062046  047511  020116  043117
9259  062054  042040  052101  020101
9260  062062  047111  051040  040505
9261  062070  020104  042510  042101
9262  062076  051105  024040  034061
9263  062104  041040  052111  043040
9264  062112  051117  040515  024524
9265  062120     000
9266  062121     101  052124  046505  EM225:  .ASCIZ  /ATTEMPTING READ HEADER (18 BIT FORMAT) IN MAINT MODE/
9267  062126  052120  047111  020107
9268  062134  042522  042101  044040
9269  062142  040505  042504  020122
9270  062150  030450  020070  044502
9271  062156  020124  047506  046522
9272  062164  052101  020051  047111
9273  062172  046440  044501  052116
9274  062200  046440  042117  000105
```

```
 9275  062206  052101  042524  050115  EM226:  .ASCII  /ATTEMPTING DATA BUFFER READ AFTER/<12><15>
 9276  062214  044524  043516  042040
 9277  062222  052101  020101  052502
 9278  062230  043106  051105  051040
 9279  062236  040505  020104  043101
 9280  062244  042524  005122     015
 9281  062251     122  040505  020104          .ASCIZ  /READ HEADER IN 18 BIT FORMAT/
 9282  062256  042510  042101  051105
 9283  062264  044440  020116  034061
 9284  062272  041040  052111  043040
 9285  062300  051117  040515  000124
 9286  062306  052101  042524  050115  EM227:  .ASCII  /ATTEMPTING TO CHECK SYNCH DETECT ON READ HEADER/<15><12>
 9287  062314  044524  043516  052040
 9288  062322  020117  044103  041505
 9289  062330  020113  054523  041516
 9290  062336  020110  042504  042524
 9291  062344  052103  047440  020116
 9292  062352  042522  042101  044040
 9293  062360  040505  042504  006522
 9294  062366     012
 9295  062367     103  042510  045503          .ASCIZ  /CHECKING ZERO DETECT/
 9296  062374  047111  020107  042532
 9297  062402  047522  042040  052105
 9298  062410  041505  000124
 9299  062414  052101  042524  050115  EM230:  .ASCII  /ATTEMPTING TO CHECK WRITING OF ZEROES BETWEEN INDEX/<15><12>
 9300  062422  044524  043516  052040
 9301  062430  020117  044103  041505
 9302  062436  020113  051127  052111
 9303  062444  047111  020107  043117
 9304  062452  055040  051105  042517
 9305  062460  020123  042502  053524   —
 9306  062466  042505  020116  047111
 9307  062474  042504  006530     012
 9308  062501     101  042116  051440          .ASCIZ  /AND SECTOR PULSE/
 9309  062506  041505  047524  020122
 9310  062514  052520  051514  000105
 9311  062522  052101  042524  050115  EM231:  .ASCII  /ATTEMPTING TO RESET WRITE GATE BY SETTING/<15><12>
 9312  062530  044524  043516  052040
 9313  062536  020117  042522  042523
 9314  062544  020124  051127  052111
 9315  062552  020105  040507  042524
 9316  062560  041040  020131  042523
 9317  062566  052124  047111  006507
 9318  062574     012
 9319  062575     123  041505  047524          .ASCIZ  /SECTOR PULSE IN A WRITE HEADER COMMAND/
 9320  062602  020122  052520  051514
 9321  062610  020105  047111  040440
 9322  062616  053440  044522  042524
 9323  062624  044040  040505  042504
 9324  062632  020122  047503  046515
 9325  062640  047101  000104
 9326  062644  052101  042524  050115  EM232:  .ASCII  /ATTEMPTING TO SET WRITE GATE BY RESETTING/<15><12>
 9327  062652  044524  043516  052040
 9328  062660  020117  042523  020124
 9329  062666  051127  052111  020105
 9330  062674  040507  042524  041040
```

```
 9331  062702  020131  042522  042523
 9332  062710  052124  047111  006507
 9333  062716     012
 9334  062717     123  041505  047524          .ASCIZ  /SECTOR PULSE IN A WRITE HEADER COMMAND/
 9335  062724  020122  052520  051514
 9336  062732  020105  047111  040440
 9337  062740  053440  044522  042524
 9338  062746  044040  040505  042504
 9339  062754  020122  047503  046515
 9340  062762  047101  000104
 9341  062766  052101  042524  050115  EM233:  .ASCIZ  /ATTEMPTING TO WRITE SYNCH OF HEADER/
 9342  062774  044524  043516  052040
 9343  063002  020117  051127  052111
 9344  063010  020105  054523  041516
 9345  063016  020110  043117  044040
 9346  063024  040505  042504  000122
 9347  063032  052101  042524  050115  EM234:  .ASCIZ  /ATTEMPTING TO WRITE HEADER DATA/
 9348  063040  044524  043516  052040
 9349  063046  020117  051127  052111
 9350  063054  020105  042510  042101
 9351  063062  051105  042040  052101
 9352  063070  000101
 9353  063072  052101  042524  050115  EM235:  .ASCII  /ATTEMPTING TO RESET WRITE GATE WITH SECOND/<15><12>
 9354  063100  044524  043516  052040
 9355  063106  020117  042522  042523
 9356  063114  020124  051127  052111
 9357  063122  020105  040507  042524
 9358  063130  053440  052111  020110
 9359  063136  042523  047503  042116
 9360  063144  005015
 9361  063146  047111  042504  020130          .ASCIZ  /INDEX PULSE OF WRITE HEADER/
 9362  063154  052520  051514  020105
 9363  063162  043117  053440  044522
 9364  063170  042524  044040  040505
 9365  063176  042504  000122
 9366  063202  052101  042524  050115  EM236:  .ASCIZ  /ATTEMPTING TO COMPLETE WRITE HEADER IN MAINT MODE/
 9367  063210  044524  043516  052040
 9368  063216  020117  047503  050115
 9369  063224  042514  042524  053440
 9370  063232  044522  042524  044040
 9371  063240  040505  042504  020122
 9372  063246  047111  046440  044501
 9373  063254  052116  046440  042117
 9374  063262  000105
 9375  063264  052101  042524  050115  EM237:  .ASCIZ  /ATTEMPTING TO WRITE GAP OR DATA SYNCH/
 9376  063272  044524  043516  052040
 9377  063300  020117  051127  052111
 9378  063306  020105  040507  020120
 9379  063314  051117  042040  052101
 9380  063322  020101  054523  041516
 9381  063330  000110
 9382  063332  052101  042524  050115  EM238:  .ASCIZ /ATTEMPTING TO WRITE DATA FIELD/
 9383  063340  044524  043516  052040
 9384  063346  020117  051127  052111
 9385  063354  020105  040504  040524
 9386  063362  043040  042511  042114
```

```
 9387  063370      000
 9388  063371      101  052124  046505  EM239:  .ASCIZ  /ATTEMPING TO WRITE SYNCH OF HEADER USING 24 SECTOR FORMAT/
 9389  063376   044520  043516  052040
 9390  063404   020117  051127  052111
 9391  063412   020105  054523  041516
 9392  063420   020110  043117  044040
 9393  063426   040505  042504  020122
 9394  063434   051525  047111  020107
 9395  063442   032062  051440  041505
 9396  063450   047524  020122  047506
 9397  063456   046522  052101      000
 9398  063463      101  052124  046505  EM240:  .ASCIZ  /ATTEMPTING TO WRITE HEADER DATA USING 24 SECTOR FORMAT/
 9399  063470   052120  047111  020107
 9400  063476   047524  053440  044522
 9401  063504   042524  044040  040505
 9402  063512   042504  020122  040504
 9403  063520   040524  052440  044523
 9404  063526   043516  031040  020064
 9405  063534   042523  052103  051117
 9406  063542   043040  051117  040515
 9407  063550   000124
 9408  063552   052101  042524  050115  EM241:  .ASCIZ  /ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 26 SECTOR)/
 9409  063560   044524  043516  052040
 9410  063566   020117  047506  041522
 9411  063574   020105  047506  046522
 9412  063602   052101  042440  051122
 9413  063610   051117  024040  043103
 9414  063616   052115  036440  031040
 9415  063624   020066  042523  052103
 9416  063632   051117  000051
 9417  063636   052101  042524  050115  EM242:  .ASCIZ  /ATTEMPTING TO FORCE FORMAT ERROR (CFMT = 24 SECTOR)/
 9418  063644   044524  043516  052040
 9419  063652   020117  047506  041522
 9420  063660   020105  047506  046522
 9421  063666   052101  042440  051122
 9422  063674   051117  024040  043103
 9423  063702   052115  036440  031040
 9424  063710   020064  042523  052103
 9425  063716   051117  000051
 9426  063722   052101  042524  050115  EM243:  .ASCIZ  /ATTEMPING TO FORCE CONTROLLER ERROR WITH FAULT BIT IN DRIVE MESS/
 9427  063730   047111  020107  047524
 9428  063736   043040  051117  042503
 9429  063744   041440  047117  051124
 9430  063752   046117  042514  020122
 9431  063760   051105  047522  020122
 9432  063766   044527  044124  043040
 9433  063774   052501  052114  041040
 9434  064002   052111  044440  020116
 9435  064010   051104  053111  020105
 9436  064016   042515  051523      000
 9437  064023      101  052124  046505  EM244:  .ASCIZ  /ATTEMPTING TO CLEAR ERROR/
 9438  064030   052120  047111  020107
 9439  064036   047524  041440  042514
 9440  064044   051101  042440  051122
 9441  064052   051117      000
 9442  064055      103  046517  040515  EM3000: .ASCIZ  /COMMAND AND STATUS REG 1 INCORRECT/
```

```
 9443  064062  042116  040440  042116
 9444  064070  051440  040524  052524
 9445  064076  020123  042522  020107
 9446  064104  020061  047111  047503
 9447  064112  051122  041505  000124
 9448  064120  042515  051523  043501  EM3001: .ASCIZ  /MESSAGE A INCORRECT/
 9449  064126  020105  020101  047111
 9450  064134  047503  051122  041505
 9451  064142  000124
 9452  064144  042515  051523  043501  EM3002: .ASCIZ  /MESSAGE B INCORRECT/
 9453  064152  020105  020102  047111
 9454  064160  047503  051122  041505
 9455  064166  000124
 9456  064170  051503  020061  047111  EM3003: .ASCIZ  /CS1 INCORRECT AFTER SENDING DRIVE CLEAR/
 9457  064176  047503  051122  041505
 9458  064204  020124  043101  042524
 9459  064212  020122  042523  042116
 9460  064220  047111  020107  051104
 9461  064226  053111  020105  046103
 9462  064234  040505  000122
 9463  064240  051503  020061  047111  EM3004: .ASCIZ  /CS1 INCORRECT AFTER AFTER DATA SIMULATION/
 9464  064246  047503  051122  041505
 9465  064254  020124  043101  042524
 9466  064262  020122  043101  042524
 9467  064270  020122  040504  040524
 9468  064276  051440  046511  046125
 9469  064304  052101  047511  000116
 9470  064312  040515  047111  020124  EM3005: .ASCII  /MAINT REG. 1 INCORRECT DURING DATA SIMULATION/
 9471  064320  042522  027107  030440
 9472  064326  044440  041516  051117
 9473  064334  042522  052103  042040
 9474  064342  051125  047111  020107
 9475  064350  040504  040524  051440
 9476  064356  046511  046125  052101
 9477  064364  047511     116
 9478  064367     015  040412  052106          .ASCIZ  <15><12>/AFTER SECTOR PULSE/
 9479  064374  051105  051440  041505
 9480  064402  047524  020122  052520
 9481  064410  051514  000105
 9482  064414  040515  047111  020124  EM3006: .ASCII  /MAINT REG. 1 INCORRECT DURING DATA SIMULATION/
 9483  064422  042522  027107  030440
 9484  064430  044440  041516  051117
 9485  064436  042522  052103  042040
 9486  064444  051125  047111  020107
 9487  064452  040504  040524  051440
 9488  064460  046511  046125  052101
 9489  064466  047511     116
 9490  064471     015  040412  052106          .ASCIZ  <15><12>/AFTER INDEX PULSE/
 9491  064476  051105  044440  042116
 9492  064504  054105  050040  046125
 9493  064512  042523     000
 9494  064515     115  044501  052116  EM3007: .ASCII  /MAINT REG. 1 INCORRECT DURING DATA SIMULATION/
 9495  064522  051040  043505  020056
 9496  064530  020061  047111  047503
 9497  064536  051122  041505  020124
 9498  064544  052504  044522  043516
```

```
 9499  064552  042040  052101  020101
 9500  064560  044523  052515  040514
 9501  064566  044524  047117
 9502  064572  005015  047516  051440          .ASCIZ  <15><12>/NO SECTOR OR INDEX PULSE SUPPLIED/
 9503  064600  041505  047524  020122
 9504  064606  051117  044440  042116
 9505  064614  054105  050040  046125
 9506  064622  042523  051440  050125
 9507  064630  046120  042511  000104
 9508  064636  052502  020123  042101  EM3008: .ASCIZ  /BUS ADDRESS INCORRECT AFTER SENDING DRIVE CLEAR/
 9509  064644  051104  051505  020123
 9510  064652  047111  047503  051122
 9511  064660  041505  020124  043101
 9512  064666  042524  020122  042523
 9513  064674  042116  047111  020107
 9514  064702  051104  053111  020105
 9515  064710  046103  040505  000122
 9516  064716  047527  042122  041440  EM3009: .ASCIZ  /WORD COUNT INCORRECT AFTER SENDING DRIVE CLEAR/
 9517  064724  052517  052116  044440
 9518  064732  041516  051117  042522
 9519  064740  052103  040440  052106
 9520  064746  051105  051440  047105
 9521  064754  044504  043516  042040
 9522  064762  044522  042526  041440
 9523  064770  042514  051101     000
 9524  064775     103  030523  041440  EM3010: .ASCIZ  /CS1 CHANGED DURING COMMAND EXECUTION/
 9525  065002  040510  043516  042105
 9526  065010  042040  051125  047111
 9527  065016  020107  047503  046515
 9528  065024  047101  020104  054105
 9529  065032  041505  052125  047511
 9530  065040  000116
 9531  065042  052502  020123  042101  EM3011: .ASCIZ  /BUS ADDRESS CHANGED BEFORE INDEX PULSE/
 9532  065050  051104  051505  020123
 9533  065056  044103  047101  042507
 9534  065064  020104  042502  047506
 9535  065072  042522  044440  042116
 9536  065100  054105  050040  046125
 9537  065106  042523     000
 9538  065111     127  051117  020104  EM3012: .ASCIZ  /WORD COUNT CHANGED BEFORE INDEX PULSE/
 9539  065116  047503  047125  020124
 9540  065124  044103  047101  042507
 9541  065132  020104  042502  047506
 9542  065140  042522  044440  042116
 9543  065146  054105  050040  046125
 9544  065154  042523     000
 9545  065157     103  030523  041440  EM3013: .ASCIZ  /CS1 CHANGE AFTER INDEX PULSE/
 9546  065164  040510  043516  020105
 9547  065172  043101  042524  020122
 9548  065200  047111  042504  020130
 9549  065206  052520  051514  000105
 9550  065214  040515  047111  020124  EM3014: .ASCIZ  /MAINT REG 1 INCORRECT BEFORE INDEX PULSE/
 9551  065222  042522  020107  020061
 9552  065230  047111  047503  051122
 9553  065236  041505  020124  042502
 9554  065244  047506  042522  044440
```

```
9555  065252  042116  054105  050040
9556  065260  046125  042523     000
9557  065265     102  051525  040440  EM3015: .ASCIZ  /BUS ADDRESS CHANGED AFTER INDEX PULSE/
9558  065272  042104  042522  051523
9559  065300  041440  040510  043516
9560  065306  042105  040440  052106
9561  065314  051105  044440  042116
9562  065322  054105  050040  046125
9563  065330  042523     000
9564  065333     127  051117  020104  EM3016: .ASCIZ  /WORD COUNT CHANGED AFTER INDEX PULSE/
9565  065340  047503  047125  020124
9566  065346  044103  047101  042507
9567  065354  020104  043101  042524
9568  065362  020122  047111  042504
9569  065370  020130  052520  051514
9570  065376  000105
9571  065400  047503  046515  047101  EM3018: .ASCIZ  /COMMAND AND STATUS REG. 2 INCORRECT/
9572  065406  020104  047101  020104
9573  065414  052123  052101  051525
9574  065422  051040  043505  020056
9575  065430  020062  047111  047503
9576  065436  051122  041505  000124
9577  065444  052502  020123  042101  EM3019: .ASCIZ  /BUS ADD REG INCORRECT/
9578  065452  020104  042522  020107
9579  065460  047111  047503  051122
9580  065466  041505  000124
9581  065472  047527  042122  041440  EM3020: .ASCIZ  /WORD COUNT REG INCORRECT/
9582  065500  052517  052116  051040
9583  065506  043505  044440  041516
9584  065514  051117  042522  052103
9585  065522     000
9586  065523     104  052101  020101  EM3021: .ASCIZ  /DATA READ INCORRECT/
9587  065530  042522  042101  044440
9588  065536  041516  051117  042522
9589  065544  052103     000
9590  065547     103  030523  044440  EM3022: .ASCIZ  /CS1 INCORRECT AFTER READING DATA BUFFER/
9591  065554  041516  051117  042522
9592  065562  052103  040440  052106
9593  065570  051105  051040  040505
9594  065576  044504  043516  042040
9595  066604  052101  020101  052502
9596  065612  043106  051105     000
9597  065617     103  031123  044440  EM3023: .ASCIZ  /CS2 INCORRECT AFTER READING DATA BUFFER/
9598  065624  041516  051117  042522
9599  065632  052103  040440  052106
9600  065640  051105  051040  040505
9601  065646  044504  043516  042040
9602  065654  052101  020101  052502
9603  065662  043106  051105     000
9604  065667     105  051122  051117  EM3024: .ASCIZ  /ERROR REG INCORRECT/
9605  065674  051040  043505  044440
9606  065702  041516  051117  042522
9607  065710  052103     000
9608  065713     115  044501  052116  EM3025: .ASCIZ  /MAINT REG 1 INCORRECT AFTER INDEX PULSE/
9609  065720  051040  043505  030440
9610  065726  044440  041516  051117
```

```
 9611  065734  042522  052103  040440
 9612  065742  052106  051105  044440
 9613  065750  042116  054105  050040
 9614  065756  046125  042523     000
 9615  065763     103  030523  044440  EM3026: .ASCIZ  /CS1 INCORRECT AFTER COMMAND COMPLETION/
 9616  065770  041516  051117  042522
 9617  065776  052103  040440  052106
 9618  066004  051105  041440  046517
 9619  066012  040515  042116  041440
 9620  066020  046517  046120  052105
 9621  066026  047511  000116
 9622  066032  051503  020062  047111  EM3027: .ASCIZ  /CS2 INCORRECT AFTER COMMAND COMPLETION/
 9623  066040  047503  051122  041505
 9624  066046  020124  043101  042524
 9625  066054  020122  047503  046515
 9626  066062  047101  020104  047503
 9627  066070  050115  042514  044524
 9628  066076  047117     000
 9629  066101     103  030523  044440  EM3028: .ASCIZ  /CS1 INCORRECT AFTER READING EMPTY SILO/
 9630  066106  041516  051117  042522
 9631  066114  052103  040440  052106
 9632  066122  051105  051040  040505
 9633  066130  044504  043516  042440
 9634  066136  050115  054524  051440
 9635  066144  046111  000117
 9636  066150  051503  020062  047111  EM3029: .ASCIZ  /CS2 INCORRECT AFTER READING EMPTY SILO/
 9637  066156  047503  051122  041505
 9638  066164  020124  043101  042524
 9639  066172  020122  042522  042101
 9640  066200  047111  020107  046505
 9641  066206  052120  020131  044523
 9642  066214  047514     000
 9643  066217     115  044501  052116  EM3030: .ASCIZ  /MAINT REG 1 INCORRECT/
 9644  066224  051040  043505  030440
 9645  066232  044440  041516  051117
 9646  066240  042522  052103     000
 9647  066245     104  044522  042526  EM3031: .ASCIZ  /DRIVE STATUS REG INCORRECT/
 9648  066252  051440  040524  052524
 9649  066260  020123  042522  020107
 9650  066266  047111  047503  051122
 9651  066274  041505  000124
 9652  066300  051105  047522  020122  EM3032: .ASCIZ  /ERROR REG INCORRECT/
 9653  066306  042522  020107  047111
 9654  066314  047503  051122  041505
 9655  066322  000124
 9656  066324  051115  020061  047111  EMW1:   .ASCIZ  /MR1 INCORRECT ON 1ST UPWARD TRANSISTION OF MAINT CLOCK/
 9657  066332  047503  051122  041505
 9658  066340  020124  047117  030440
 9659  066346  052123  052440  053520
 9660  066354  051101  020104  051124
 9661  066362  047101  044523  052123
 9662  066370  047511  020116  043117
 9663  066376  046440  044501  052116
 9664  066404  041440  047514  045503
 9665  066412     000
 9666  066413     115  030522  044440  EMW2:   .ASCIZ  /MR1 INCORRECT ON 1ST DOWNWARD TRANSISTION OF MAINT CLOCK/
```

```
 9667  066420  041516  051117  042522
 9668  066426  052103  047440  020116
 9669  066434  051461  020124  047504
 9670  066442  047127  040527  042122
 9671  066450  052040  040522  051516
 9672  066456  051511  044524  047117
 9673  066464  047440  020106  040515
 9674  066472  047111  020124  046103
 9675  066500  041517  000113
 9676  066504  051115  020061  047111  EMW3:   .ASCIZ  /MR1 INCORRECT ON 2ND UPWARD TRANSISTION OF MAINT CLOCK/
 9677  066512  047503  051122  041505
 9678  066520  020124  047117  031040
 9679  066526  042116  052440  053520
 9680  066534  051101  020104  051124
 9681  066542  047101  044523  052123
 9682  066550  047511  020116  043117
 9683  066556  046440  044501  052116
 9684  066564  041440  047514  045503
 9685  066572     000
 9686  066573     115  030522  044440  EMW4:   .ASCIZ  /MR1 INCORRECT ON 2ND DOWNWARD TRANSISTION OF MAINT CLOCK/
 9687  066600  041516  051117  042522
 9688  066606  052103  047440  020116
 9689  066614  047062  020104  047504
 9690  066622  047127  040527  042122
 9691  066630  052040  040522  051516
 9692  066636  051511  044524  047117
 9693  066644  047440  020106  040515
 9694  066652  047111  020124  046103
 9695  066660  041517  000113
```

```
                        .SBTTL  DATA PATTERNS

                        .EVEN
PATTERN:  .WORD   000000
          .WORD   177777
          .WORD   125252
          .WORD   052515
          .WORD   101706
          .WORD   060422
          .WORD   130715
HEAD1:    .WORD   177777
          .WORD   000000
          .WORD   177777
HEAD2:    .WORD   000000
          .WORD   177777
          .WORD   000000
HEAD3:    .WORD   125252
          .WORD   052525
          .WORD   125252
          .WORD   052525
          .WORD   125252
          .WORD   052525
HEAD4:    .WORD   044444
          .WORD   022222
          .WORD   111111
HEAD5:    .WORD   052012
          .WORD   100520
          .WORD   052012
HEAD6:    .WORD   155555
          .WORD   066666
          .WORD   155555
HEAD7:    .WORD   104210
          .WORD   104210
          .WORD   104210
NPRBUF:   .BYTE   100.100
          .BYTE   101.101
          .BYTE   102.102
          .BYTE   103.103
          .BYTE   104.104
          .BYTE   105.105
          .BYTE   106.106
          .BYTE   107.107
          .BYTE   110.110
          .BYTE   111.111
          .BYTE   112.112
          .BYTE   113.113
          .BYTE   114.114
          .BYTE   115.115
          .BYTE   116.116
          .BYTE   117.117
          .BYTE   120.120
          .BYTE   121.121
          .BYTE   122.122
          .BYTE   123.123
          .BYTE   124.124
          .BYTE   125.125
```

```
 9752  067036      126     126                .BYTE   126,126
 9753  067040      127     127                .BYTE   127,127
 9754  067042      130     130                .BYTE   130,130
 9755  067044      131     131                .BYTE   131,131
 9756  067046      132     132                .BYTE   132,132
 9757  067050      133     133                .BYTE   133,133
 9758  067052      134     134                .BYTE   134,134
 9759  067054      135     135                .BYTE   135,135
 9760  067056      136     136                .BYTE   136,136
 9761  067060      137     137                .BYTE   137,137
 9762  067062      140     140                .BYTE   140,140
 9763  067064      141     141                .BYTE   141,141
 9764  067066      142     142                .BYTE   142,142
 9765  067070      143     143                .BYTE   143,143
 9766  067072      144     144                .BYTE   144,144
 9767  067074      145     145                .BYTE   145,145
 9768  067076      146     146                .BYTE   146,146
 9769  067100      147     147                .BYTE   147,147
 9770  067102      150     150                .BYTE   150,150
 9771  067104      151     151                .BYTE   151,151
 9772  067106      152     152                .BYTE   152,152
 9773  067110      153     153                .BYTE   153,153
 9774  067112      154     154                .BYTE   154,154
 9775  067114      155     155                .BYTE   155,155
 9776  067116      156     156                .BYTE   156,156
 9777  067120      157     157                .BYTE   157,157
 9778  067122      160     160                .BYTE   160,160
 9779  067124      161     161                .BYTE   161,161
 9780  067126      162     162                .BYTE   162,162
 9781  067130      163     163                .BYTE   163,163
 9782  067132      164     164                .BYTE   164,164
 9783  067134      165     165                .BYTE   165,165
 9784  067136      166     166                .BYTE   166,166
 9785  067140      167     167                .BYTE   167,167
 9786  067142      170     170                .BYTE   170,170
 9787  067144      171     171                .BYTE   171,171
 9788  067146      172     172                .BYTE   172,172
 9789  067150      173     173                .BYTE   173,173
 9790  067152      174     174                .BYTE   174,174
 9791  067154      175     175                .BYTE   175,175
 9792  067156      176     176                .BYTE   176,176
 9793  067160      177     177                .BYTE   177,177
 9794  067162      200     200                .BYTE   200,200
 9795  067164      201     201                .BYTE   201,201
 9796  067166      202     202                .BYTE   202,202
 9797  067170      203     203                .BYTE   203,203
 9798  067172      204     204                .BYTE   204,204
 9799  067174      205     205                .BYTE   205,205
 9800  067176      206     206                .BYTE   206,206
 9801  067200      207     207                .BYTE   207,207
 9802  067202      210     210                .BYTE   210,210
 9803  067204  000102              WRBUFF:  .BLKW   66.
 9804          067220              BADPAR=  WRBUFF+14
 9805          000001              .END
```

```
ABASE = 177440          1042*   1280    1321
ABORT   054647          7800    9773*
ACDW1 = 000000          1280    1323
ACDW2 = 000000          1280    1324
ACLO  = 000010          1137*
ACPUOP= 000000          1280    1295
ADDW0 = 000000          1280
ADDW1 = 000000          1280
ADDW10= 000000          1280
ADDW11= 000000          1280
ADDW12= 000000          1280
ADDW13= 000000          1280
ADDW14= 000000          1280
ADDW15= 000000          1280
ADDW2 = 000000          1280
ADDW3 = 000000          1280
ADDW4 = 000000          1280
ADDW5 = 000000          1280
ADDW6 = 000000          1280
ADDW7 = 000000          1280
ADDW8 = 000000          1280
ADDW9 = 000000          1280
ADEVCT= 000000          1280    1286
ADEVM = 000000          1280    1322
AENV  = 000000          1280    1291
AENVM = 000000          1280    1292
AFATAL= 000000          1280    1283
AMADR1= 000000          1280    1308
AMADR2= 000000          1280    1312
AMADR3= 000000          1280    1315
AMADR4= 000000          1280    1318
AMAMS1= 000000          1280    1302
AMAMS2= 000000          1280    1310
AMAMS3= 000000          1280    1313
AMAMS4= 000000          1280    1316
AMSGAD= 000000          1280    1288
AMSGLG= 000000          1280    1289
AMSGTY= 000000          1280    1282
AMTYP1= 000000          1280    1303
AMTYP2= 000000          1280    1311
AMTYP3= 000000          1280    1314
AMTYP4= 000000          1280    1317
APASS = 000000          1280    1285
APRIOR= 000005          1041*   1280
APTCSU= 000040          7677*   7855
APTENV= 000001          7633    7675*   7709    7848
APTSIZ= 000200          2312    7674*
APTSPO= 000100          7635    7676*   7850
ASWREG= 000000          1280    1293
ATESTN= 000000          1280    1284
AUNIT = 000000          1280    1287
AUSWR = 000000          1280    1294
AVECT1= 120210          1040*   1280    1319
AVECT2= 000000          1280    1320
BADPAR= 067220          4252    4253    4318    4319    4391*   4546    4547    4623*   7209*   7225*   9804*
BAI   = 000020          1100*   3540    3555    3585    3614    3627
```

```
BA16  = 000400          1085*   3658    3659    3758    3759    3820    3984    4084    4085    4146
BA17  = 001000          1086*   3658    3659    3921    3922    3983    4084    4085    4146    4147
BITCNT  003622          2250*   2618*   2630*   2643*   2654*   2711*   2724*   2736*   2785*   2798*   2810*   2877*   2934*
                        4686*   4687*   4796*   4797*   4906*   4907*   5016*   5017*   5126*   5127*   5235*   5236*   5316*
                        5329*   5330    5404*   5405*   5507*   5510*   5583*   5586*   5592*   5609*   5621*   5703*   5706*
                        5712*   5729*   5741*   5822*   5825*   5831*   5848*   5860*   5941*   5944*   5950*   5967*   5979*
                        6060*   6063*   6069*   6086*   6098*   6179*   6182*   6188*   6205*   6217*   6298*   6301*   6307*
                        6324*   6336*   6426*   6429*   6435*   6451*   6463*   6551*   6554*   6560*   6576*   6595*   6603*
                        6610*   6617*   6697*   6700*   6706*   6722*   6741*   6749*   6756*   6763*   6847*   6850*   6856*
                        6872*   6884*   8489    8495    8521    8523    8526
BIT0  = 000001           986*   1082    1115    1134
BIT00 = 000001           976*    986
BIT01 = 000002           975*    985
BIT02 = 000004           974*    984
BIT03 = 000010           973*    983
BIT04 = 000020           972*    982
BIT05 = 000040           971*    981
BIT06 = 000100           970*    980
BIT07 = 000200           969*    979
BIT08 = 000400           968*    978    7524
BIT09 = 001000           967*    977    7542    7720
BIT1  = 000002           985*   1116
BIT10 = 002000           966*   1087    1106    1125    1158    1173    7697
BIT11 = 004000           965*   1088    1107    1126    1143    1159    1174    7549
BIT12 = 010000           964*   1090    1108    1127    1160
BIT13 = 020000           963*   1091    1109    1128    1144    1161    7704
BIT14 = 040000           962*   1092    1110    1129    1145    1162    7510
BIT15 = 100000           961*   1093    1094    1111    1130    1146    1163
BIT2  = 000004           984*   1117    1136
BIT3  = 000010           983*   1099    1118    1137
BIT4  = 000020           982*   1100    1119    1138    1152    1167
BIT5  = 000040           981*   1101    1120    1139    1153    1168
BIT6  = 000100           980*   1083    1102    1121    1140    1154    1169
BIT7  = 000200           979*   1084    1103    1122    1141    1155    1170
BIT8  = 000400           978*   1085    1104    1123    1142    1156    1171
BIT9  = 001000           977*   1086    1105    1124    1157    1172
BPTVEC= 000014           993*
BSE   = 000200          1122*
CCLR  = 100000          1094*   2417    2438    2462    2483    2505    2526    2549    2570    2597    2612    2627    2640
                        2652    2662    2686    2701    2721    2733    2743    2767    2782    2795    2807    2817    2843
                        2996    3025    3115    3231    3360    3534    3652    3703    3755    3814    3876    3918    3977
                        4039    4081    4140    4202    4248    4298    4314    4390    4407    4542    4624    4650    4689
                        4760    4799    4870    4909    4980    5019    5090    5129    5199    5238    5302    5343    5366
                        5407    5473    5549    5669    5788    5907    6026    6145    6264    6390    6515    6661    6811
                        6969    7046    7125    7812
CDT   = 002000          1087*   2422    2432    2467    2477    2510    2520    2554    2564
CERR  = 100000          1093*   3658    4254    4369    5335    6948    7025    7104
CFMT  = 010000          1090*   2422    2432    2467    2477    2510    2520    2554    2564    4413    4414    4548    4549
                        5201    5219    5253    6816    6908    7009    7025
CKSWR = 104407          7509    7693    7719    8474*
CLEAR = 000005          1067*
COE   = 001000          1124*
CR    = 000015           901*   7894    7904
CRLF  = 000200           902*   2337    7865    7904
CTO   = 004000          1088*
CTRHLT  050202          7811*   8090    8173
```

DZR6CA.P11 05-OCT-76 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

```
DCK    = 100000          1130*  5336
DDISP  = 177570           908*  1253   2300
DDT    = 000400          1142*
DF000    053574          1962   8544*
DF001    053600          1347   1353   1359   1365   1371   1377   1383   1389   1395   1401   1407   1413   8546*
DF015    053624          1419   1425   2083   8556*
DF017    053650          1431   1437   1443   8566*
DF022    053674          1449   1455   1461   1473   1479   1485   8576*
DF025    053720          1467   8586*
DF031    053744          1491   1497   1503   1509   8596*
DF035    053770          1515   1521   1527   1533   1557   1563   1569   1575   1581   1587   1593   1599   1623
                         1629   1635   1641   1647   1653   1659   1665   1731   1737   1743   1749   1785   1791
                         1797   1803   1827   1833   1839   1845   1851   1857   1863   1869   1893   1899   1905
                         1911   1918   1925   1932   1939   8606*
DF041    054014          1539   1755   1945   8616*
DF042    054040          1545   1551   1875   1974   1980   2010   2016   8626*
DF054    054064          1605   1671   1998   2034   8636*
DF055    054110          1611   1617   1677   1683   1881   1887   1986   1992   2022   2028   8646*
DF072    054134          1689   1695   1701   1707   1713   1761   1767   1773   1779   1809   8656*
DF077    054170          1719   1725   1815   1821   1951   1957   8670*
DF150    054214          1968   2004   8680*
DF164    054240          2040   2046   2052   2058   8690*
DF170    054264          2063   8700*
DF171    054310          2068   2073   2078   8710*
DF175    054334          2089   2095   2101   2107   2113   2119   2125   2131   2137   2143   2149   2155   8720*
DF211    054360          2161   2167   2173   2179   8730*
DH000A   054760          8548   8558   8568   8578   8588   8598   8608   8618   8628   8638   8648   8658   8672
                         8682   8692   8702   8712   8722   8732   8788*
DH000B   054776          8550   8560   8570   8580   8590   8600   8610   8620   8630   8640   8650   8660   8674
                         8684   8694   8704   8714   8724   8734   8791*
DH000C   055011          1960   8793*
DH001A   055042          8552   8798*
DH001B   055121          8554   8806*
DH015A   055200          8562   8814*
DH015B   055217          8564   8817*
DH017A   055235          8572   8820*
DH017B   055301          8574   8827*
DH022A   055347          8582   8834*
DH022B   055426          8584   8842*
DH025A   055503          8592   8850*
DH025B   055527          8594   8854*
DH031A   055555          8602   8858*
DH031B   055654          8604   8869*
DH035A   055752          8612   8880*
DH035B   056051          8614   8891*
DH041A   056146          8622   8902*
DH041B   056163          8624   8905*
DH042A   056200          8632   8908*
DH042B   056237          8634   8914*
DH054A   056275          8642   8920*
DH054B   056322          8644   8924*
DH055A   056350          8652   8928*
DH055B   056415          8654   8935*
DH072A   056463          8662   8676   8942*
DH072B   056522          8664   8678   8948*
DH072C   056560          8666   8953*
```

```
DH072D  056637            8668    8961#
DH150A  056714            8686    8969#
DH150B  056750            8689    8974#
DH164A  057006            8696    8979#
DH164B  057052            8698    8985#
DH170A  057120            8706    8992#
DH170B  057217            8708    9003#
DH171A  057315            8716    9014#
DH171B  057334            8718    9017#
DH175A  057352            8726    8736    9020#
DH175B  057451            8728    8738    9031#
DH211A  057546            8740    9042#
DH211B  057574            8742    9046#
DI    = 040000            1092#
DISPLA  001142            1253#   2300*   2308*   7564*   7696*
DISPRE  000174            1182#   2308
DLT   = 100000            1111#
DMD   = 000040            1153#   2418    2425    2426    2463    2470    2471    2506    2513    2514    2550    2557    2558
                          2598    2599    2600    2603    2604    2615    2616    2617    2633    2687    2688    2689    2692
                          2693    2704    2706    2707    2710    2712    2768    2769    2770    2773    2774    2786    2844
                          2845    2846    2851    2852    2876    2879    2880    2881    2882    2910    2911    2912    2915
                          2936    2937    2938    2939    2966    2968    2969    2972    2975    2976    2983    3027    3036
                          3037    3041    3042    3043    3046    3048    3049    3116    3125    3126    3130    3131    3132
                          3135    3138    3139    3168    3169    3232    3240    3241    3245    3246    3247    3250    3253
                          3254    3288    3289    3361    3369    3370    3374    3375    3376    3379    3382    3383    3421
                          3422    3459    3460    3535    3544    3545    3549    3550    3551    3554    3557    3558    3590
                          3591    3653    3662    3663    3667    3668    3669    3672    3674    3675    3756    3761    3762
                          3766    3767    3768    3771    3773    3774    3815    3823    3824    3828    3829    3830    3833
                          3836    3837    3919    3924    3925    3929    3930    3931    3934    3936    3937    3978    3986
                          3987    3991    3992    3993    3996    3999    4000    4082    4087    4088    4092    4093    4094
                          4097    4099    4100    4141    4149    4150    4154    4155    4156    4159    4162    4163    4249
                          4257    4258    4262    4263    4264    4267    4269    4270    4315    4323    4324    4328    4329
                          4330    4333    4335    4336    4360    4361    4408    4417    4418    4422    4423    4424    4427
                          4431    4432    4464    4465    4543    4552    4553    4557    4558    4559    4562    4565    4566
                          4589    4590    4651    4654    4655    4658    4659    4761    4764    4765    4768    4769    4871
                          4874    4875    4878    4879    4981    4984    4985    4988    4989    5091    5094    5095    5098
                          5099    5200    5203    5204    5207    5208    5303    5306    5307    5310    5311    5367    5370
                          5371    5374    5375    5376    5377    5474    5475    5476    5482    5483    5487    5488    5489
                          5492    5493    5496    5497    5515    5522    5550    5556    5557    5561    5562    5563    5566
                          5568    5569    5572    5573    5576    5624    5626    5627    5630    5632    5637    5638    5670
                          5676    5677    5681    5682    5683    5686    5688    5689    5692    5693    5696    5744    5746
                          5747    5750    5752    5757    5758    5789    5795    5796    5800    5801    5802    5805    5807
                          5808    5811    5812    5815    5863    5865    5866    5869    5871    5876    5877    5908    5914
                          5915    5919    5920    5921    5924    5926    5927    5930    5931    5934    5982    5984    5985
                          5988    5990    5995    5996    6027    6033    6034    6038    6039    6040    6043    6045    6046
                          6049    6050    6053    6101    6103    6104    6107    6109    6114    6115    6146    6152    6153
                          6157    6158    6159    6162    6164    6165    6168    6169    6172    6220    6222    6223    6226
                          6228    6233    6234    6265    6271    6272    6276    6277    6278    6281    6283    6284    6287
                          6288    6291    6339    6341    6342    6345    6347    6352    6353    6391    6398    6399    6403
                          6404    6405    6409    6412    6413    6416    6417    6419    6469    6471    6472    6475    6477
                          6482    6483    6516    6523    6524    6528    6529    6530    6533    6535    6536    6541    6542
                          6544    6625    6627    6628    6631    6633    6639    6640    6662    6669    6670    6674    6675
                          6676    6679    6681    6682    6687    6688    6690    6771    6773    6774    6777    6779    6785
                          6786    6812    6819    6820    6824    6825    6826    6830    6833    6834    6837    6838    6840
                          6890    6892    6893    6896    6898    6903    6904    6930    6934    6935    7007    7011    7012
                          7084    7090    7091    7255    7285    7299    7314    7340    7352    7364    7389    7392    7393
```

```
                        7394    7395    7398    7399    7400    7401
DRA    = 000001         1134#   6950    7027    7106
DRDY   = 000200         1141#
DROT   = 000040         1139#
DRPAR  = 000010         1118#   7028
DRVMSK= 000007          1098#
DSC    = 040000         1145#
DSWR   = 177570          907#   1252    2299
DTE    = 010000         1127#
DTYE   = 000040         1120#
DT000    053132         1961    8483#
DT001    053136         1346    1352    1358    1364    1370    1376    1382    1388    1394    1400    1406    1412    8484#
DT015    053156         1418    1424    2082    8487#
DT017    053166         1430    1436    1442    8489#
DT022    053204         1448    1454    1460    1472    1478    1484    8492#
DT025    053224         1466    8495#
DT031    053236         1490    1496    1502    1508    8497#
DT035    053262         1514    1520    1526    1532    1556    1562    1568    1574    1580    1586    1592    1598    1622
                        1628    1634    1640    1646    1652    1658    1664    1730    1736    1742    1748    1784    1790
                        1796    1802    1826    1832    1838    1844    1850    1856    1862    1868    1892    1898    1904
                        1910    1917    1924    1931    1938    8501#
DT041    053306         1538    1754    1944    8505#
DT042    053316         1544    1550    1874    1973    1979    2009    2015    8507#
DT054    053332         1604    1670    1997    2033    8509#
DT055    053344         1610    1616    1676    1682    1880    1886    1985    1991    2021    2027    8511#
DT072    053362         1688    1694    1700    1706    1712    1760    1766    1772    1778    1808    8514#
DT077    053412         1718    1724    1814    1820    1950    1956    8519#
DT150    053426         1967    2003    8521#
DT164    053442         2039    2045    2051    2057    8523#
DT170    053460         2062    8526#
DT171    053504         2067    2072    2077    8530#
DT175    053514         2088    2094    2100    2106    2112    2118    2124    2130    2136    2142    2148    2154    8532#
DT211    053540         2160    2166    2172    2178    8536#
ECCW   = 020000         1161#   2617    2633    2712    2786    2876    2983    5493    5576    5632    5696    5752    5815
                        5871    5934    5990    6053    6109    6172    6228    6291    6347    6419    6477    6544    6607
                        6613    6633    6690    6753    6759    6779    6840    6898
ECH    = 000100         1121#
EMTVEC= 000030           996#   2283#   2284#
EMW      003170         2060#   5506#   5575#   5593#   5695#   5713#   5814#   5832#   5933#   5951#   6052#   6070#   6171#
                        6189#   6290#   6308#   6418#   6436#   6543#   6561#   6581#   6594#   6689#   6707#   6727#   6740#
                        6839#   6857#   7260#   7290#   7304#   7319#   7345#   7357#   7369#
EMW1     066324         7260    9656#
EMW2     066413         7290    7345    9666#
EMW3     066504         7304    7357    9676#
EMW4     066573         7319    7369    9686#
EM000    057631         1959    9053#
EM200    057676         1344    1350    1356    9060#
EM201    057760         1362    1368    1374    9069#
EM202    060043         1380    1386    1392    9078#
EM203    060134         1398    1404    1410    9088#
EM204    060226         1416    1422    1428    1434    1440    9098#
EM205    060314         1446    1452    1458    1464    1470    1476    1482    1488    1494    1500    1506    9107#
EM206    060402         1512    1518    1524    1530    1536    1542    1548    9116#
EM207    060445         1554    1560    1566    1572    1602    1608    1614    9122#
EM208    060474         1578    1584    1590    1596    9126#
EM209    060545         1620    1626    1632    1638    1668    1674    1680    9133#
```

```
EM210   060634          1644    1650    1656    1662    9143#
EM211   060750          1686    1692    1698    1704    1710    9156#
EM212   061020          1716    1722    9163#
EM213   061070          1728    1734    1740    1746    1752    9170#
EM214   061140          1758    1764    1770    1776    1806    9177#
EM215   061210          1782    1788    1794    1800    9184#
EM216   061274          1812    1818    9193#
EM217   061344          1824    1830    1836    1842    1872    1878    1884    9200#
EM218   061377          1848    1854    1860    1866    9205#
EM219   061457          1890    1896    1902    1908    1942    1948    1954    9214#
EM220   061537          1915    1922    1929    1936    9223#
EM221   061616          1965    9231#
EM222   061673          1971    1977    9239#
EM223   061746          1983    1989    1995    9247#
EM224   062024          2001    9255#
EM225   062121          2007    2013    9266#
EM226   062206          2019    2025    2031    9275#
EM227   062306          2037    2043    2049    2055    9286#
EM230   062414          5506    9299#
EM231   062522          2065    9311#
EM232   062644          2070    9326#
EM233   062766          5575    5695    5814    5933    6052    6171    6290    6418    6543    6689    9341#
EM234   063032          5593    5713    5832    5951    6070    6189    6308    6436    6561    6707    9347#
EM235   063072          2075    9353#
EM236   063202          2080    9366#
EM237   063264          6581    6727    9375#
EM238   063332          6594    6740    9382#
EM239   063371          6839    9388#
EM240   063463          6857    9398#
EM241   063552          2086    2092    2098    2104    9408#
EM242   063636          2110    2116    2122    2128    9417#
EM243   063722          2134    2140    2146    2152    9426#
EM244   064023          2158    2164    2170    2176    9437#
EM3000  064055          1345    1363    1381    1399    1513    1555    1579    1621    1645    1687    1717    1729    1759
                        1783    1813    1825    1849    1891    1916    1966    2002    2038    2081    2087    2111    2135
                        2159    9442#
EM3001  064120          1351    1369    1387    1405    9448#
EM3002  064144          1357    1375    1393    1411    9452#
EM3003  064170          1417    1447    9456#
EM3004  064240          1423    9463#
EM3005  064312          1429    9470#
EM3006  064414          1435    9482#
EM3007  064515          1441    9494#
EM3008  064636          1453    9508#
EM3009  064716          1459    9516#
EM3010  064775          1471    9524#
EM3011  065042          1477    9531#
EM3012  065111          1483    9538#
EM3013  065157          1489    9545#
EM3014  065214          1465    9550#
EM3015  065265          1495    9557#
EM3016  065333          1501    9564#
EM3018  065400          1519    1561    1585    1627    1651    1693    1723    1735    1765    1789    1801    1819    1831
                        1855    1897    1923    2044    2093    2117    2141    2165    9571#
EM3019  065444          1525    1567    1591    1633    1657    1705    1741    1771    1795    1837    1861    1903    1930
                        9577#
```

```
EM3020  065472  1531  1573  1597  1639  1663  1711  1747  1777  1843  1867  1909  1937  9581#
EM3021  065523  1537  1603  1669  1753  1873  1943  1996  2032  9586#
EM3022  065547  1543  1609  1675  1879  1949  1984  2020  9590#
EM3023  065617  1549  1615  1681  1885  1955  1990  2026  9597#
EM3024  065667  1699  1807  9604#
EM3025  065713  1507  9608#
EM3026  065763  1972  2008  9615#
EM3027  066032  1978  2014  9622#
EM3028  066101  2050  9629#
EM3029  066150  2056  9636#
EM3030  066217  2066  2071  2076  9643#
EM3031  066245  2099  2123  2147  2171  9647#
EM3032  066300  2105  2129  2153  2177  9652#
ERRCNT  003610  2245#  2316*  7795*  7798
ERRVEC=  000004  989#  2297  2298*  2309*  7223*  7224*  7244*  7245*  7437  7438  7442*  7447*  7459*
                7464*  7473*  7483*  7484*  7515  7516*  7518*  7521*
E.ASOF  003516  2208#
E.BA    003504  2203#  2859*  2866  2899  2924  2956  2987  3018*  3066  3118*  3145  3151*  3159
                3182  3233*  3260  3272  3278*  3286*  3302  3362*  3389  3401  3407*  3416*  3435
                3465*  3478  3489*  3536*  3576  3604  3654*  3697  3785*  3793  3818*  3856  3862*
                3948*  3956  3981*  4019  4025*  4111*  4119  4144*  4182  4188*  4253*  4289  4319*
                4353  4372*  4380  4412*  4438  4453  4459*  4468*  4480  4547*  4582  4603  8492
                8497  8501  8517
E.CS1   003500  2201#  2432*  2435  2477*  2480  2520*  2523  2564*  2567  2608*  2609  2666*  2667
                2697*  2698  2747*  2748  2778*  2779  2821*  2822  2858*  2861  2894  2919  2951
                2984  3019*  3060  3080  3110*  3153  3176  3201  3230*  3266  3296  3327  3359*
                3395  3429  3452  3472  3508  3533*  3570  3598  3628  3658*  3688  3706*  3708
                3758*  3787  3820*  3850  3921*  3950  3983*  4013  4084*  4113  4146*  4176  4254*
                4283  4301*  4303  4320*  4347  4369*  4374  4413*  4447  4474  4503  4548*  4576
                4597  4617  4670*  4682  4704*  4706  4721  4780*  4792  4814*  4816  4831  4890*
                4902  4924*  4926  4941  5000*  5012  5034*  5036  5051  5110*  5122  5144*  5146
                5161  5219*  5231  5253*  5255  5270  5314*  5320  5335*  5337  5388*  5400  5422*
                5424  5439  5642*  5643  5762*  5763  5881*  5882  6000*  6001  6119*  6120  6238*
                6239  6357*  6358  6487*  6488  6644*  6645  6790*  6791  6908*  6909  6948*  6952
                6974*  6978  7025*  7029  7051*  7055  7104*  7108  7130*  7134  8484  8487  8492
                8497  8501  8507  8511  8514  8519  8521  8523  8532  8536
E.CS2   003510  2205#  3059*  3063  3079*  3083  3136*  3156  3179  3200*  3204  3251*  3269  3282*
                3299  3311*  3326*  3330  3380*  3398  3412*  3432  3441*  3455  3467*  3475  3491*
                3507*  3511  3555*  3573  3585*  3601  3614*  3627*  3631  3684*  3687*  3691  3707*
                3711  3784*  3790  3849*  3853  3947*  3953  4012*  4016  4110*  4116  4175*  4179
                4281*  4286  4302*  4306  4346*  4350  4370*  4377  4428*  4446*  4450  4477  4487*
                4502*  4506  4563*  4579  4600  4616*  4620  4705*  4709  4720*  4726  4815*  4819
                4830*  4836  4925*  4929  4940*  4946  5035*  5039  5050*  5056  5145*  5149  5160*
                5166  5254*  5258  5269*  5275  5315*  5325  5336*  5340  5423*  5427  5438*  5444
                6949*  6955  6975*  6981  7026*  7032  7052*  7058  7105*  7111  7131*  7137  8501
                8507  8511  8514  8519  8523  8532  8536
E.DA    003506  2204#
E.DB    003522  2210#  3073*  3074  3192*  3193  3313*  3314  3443*  3444  3494*  3495  3611*  3615
                3800*  3801  3869*  3870  3963*  3964  4032*  4033  4126*  4127  4195*  4196  4491*
                4492  4610*  4611  4714*  4731  4824*  4841  4934*  4951  5044*  5061  5154*  5171
                5263*  5280  5432*  5449  8505  8509
E.DCYL  003520  2209#
E.DS    003512  2206#  6950*  6958  6976*  6984  7027*  7035  7053*  7061  7106*  7114  7132*  7140
                8532  8536
E.ECPS  003532  2214#
E.ECPT  003534  2215#
```

```
E.ER    003514  2207# 3683* 3694  4282* 4295  4373* 4386  6951* 6961  6977* 6987  7028* 7038
                7054* 7064  7107* 7117  7133* 7143  8514  8532  8536
E.MR1   003524  2211# 2617* 2624  2633* 2637  2649  2659  2712* 2718  2730  2741  2786* 2792
                2804  2815  2876* 2884  2941  2983* 2993  5493* 5513* 5514* 5517  5520* 5521*
                5524  5576* 5632* 5633  5696* 5752* 5753  5815* 5871* 5872  5934* 5990* 5991
                6053* 6109* 6110  6172* 6228* 6229  6291* 6347* 6348  6419* 6477* 6478  6544*
                6607* 6613* 6633* 6634  6690* 6753* 6759* 6779* 6780  6840* 6898* 6899  7254*
                7257  7268* 7269* 7274* 7279* 7284* 7287  7298* 7301  7312* 7313* 7316  7333*
                7336* 7339* 7342  7351* 7354  7363* 7366  8489  8495  8497  8526  8530
E.MR2   003526  2212# 2433* 2440  2478* 2485  2521* 2528  2565* 2572  8484
E.MR3   003530  2213# 2434* 2443  2479* 2488  2522* 2531  2566* 2575  8484
E.SPAR  003536  2216#
E.WC    003502  2202# 2860* 2871  2904  2929  2961  2990  3020* 3069  3119* 3120* 3121  3152*
                3162  3165  3185  3235* 3275  3279* 3285* 3305  3364* 3404  3408* 3409  3415*
                3438  3466* 3481  3490* 3538* 3564  3579  3582* 3588* 3607  3656* 3700  3786*
                3796  3816* 3843  3859  3863* 3949* 3959  3979* 4006  4022  4026* 4112* 4122
                4142* 4169  4185  4189* 4251* 4292  4317* 4356  4371* 4383  4410* 4444  4456
                4460* 4469* 4483  4545* 4585  4606  8492  8497  8504  8517
FMTE  = 000020  1119# 6951  7028
GNS   = ****** U  1181  2336  7171  7178  8466  8467  8468  8469  8470  8472  8474  8475  8476
                8477  8478  8479  8480
GO    = 000001  1082# 3658  4254  4369  4704  4814  4924  5034  5144  5253  5335  5422  5642
                5762  5881  6000  6119  6238  6357  6487  6644  6790  6908  6948  7025  7104
GTSWR = 104406  2331  8472#
HEAD1   066702  4669  4713  5218  5262  5387  5431  5551  5594  6392  6393  6813  6814  9706#
HEAD2   066710  4779  4823  5671  5714  9709#
HEAD3   066716  4889  4933  5790  5833  6517  6518  9712#
HEAD4   066732  4999  5043  5909  5952  9718#
HEAD5   066740  5109  5153  6028  6071  9721#
HEAD6   066746  6147  6190  9724#
HEAD7   066754  6266  6309  9727#
HT    = 000011   899# 7863  7904
HVRC  = 000400  1123#
IDAE  = 002000  1125#
IE    = 000100  1083#
ILF   = 000001  1115#
INTR  = 000300  1078#
IOTVEC= 000020   994# 2281* 2282*
IR    = 000100  1102# 3059  3079  3136  3200  3251  3311  3326  3380  3441  3491  3507  3555
                3614  3627  3684  3707  3784  3849  3947  4012  4110  4175  4281  4302  4346
                4370  4428  4487  4502  4563  4616  4705  4720  4815  4830  4925  4940  5035
                5050  5145  5160  5254  5269  5315  5336  5423  5438  6949  6975  7026  7052
                7105  7131
KIPAR0= 172340  1027# 7423  7451
KIPAR1= 172342  1028#
KIPAR2= 172344  1029#
KIPAR3= 172346  1030#
KIPAR4= 172350  1031#
KIPAR5= 172352  1032#
KIPAR6= 172354  1033# 3747* 3805* 3910* 3968* 4073* 4131*
KIPAR7= 172356  1034# 7467
KIPDR0= 172300  1016#
KIPDR1= 172302  1017#
KIPDR2= 172304  1018#
KIPDR3= 172306  1019#
KIPDR4= 172310  1020#
```

```
KIPDR5= 172312          1021#
KIPDR6= 172314          1022#
KIPDR7= 172316          1023#
LF    = 000012           900#  7898  7904
MCLK  = 000400          1156#  2425  2470  2513  2557  2603  2692  2706  2773  2851  2879  2881  2911
                        2936  2938  2968  2975  3036  3042  3048  3125  3131  3138  3168  3240  3246
                        3253  3288  3369  3375  3382  3421  3459  3544  3550  3557  3590  3662  3668
                        3674  3761  3767  3773  3823  3829  3836  3924  3930  3936  3986  3992  3999
                        4087  4093  4099  4149  4155  4162  4257  4263  4269  4323  4329  4335  4360
                        4417  4423  4431  4464  4552  4558  4565  4589  4654  4764  4874  4984  5094
                        5203  5306  5370  5376  5482  5488  5496  5556  5562  5568  5626  5637  5676
                        5682  5688  5746  5757  5795  5801  5807  5865  5876  5914  5920  5926  5984
                        5995  6033  6039  6045  6103  6114  6152  6158  6164  6222  6233  6271  6277
                        6283  6341  6352  6398  6404  6412  6471  6482  6523  6529  6535  6627  6639
                        6669  6675  6681  6773  6785  6819  6825  6833  6892  6903  6934  7011  7090
                        7254  7255  7268  7298  7299  7313  7336  7351  7352  7363  7389  7392  7394
                        7398  7400
MDS   = 001000          1105#
MEMBAS= 172100          1037#  7200  7210  7227
MEMERR  046216          1200   7406#
MEMPAR  003630          2253#  4231  4529  7201  7211  7226*  7233*  7246
MEMVEC= 000114          1036#  1199  7248*  7249*
MERD  = 001000          1157#  7392  7400
MESMSK= 000017          1150#
MEWD  = 002000          1158#  2617  2633  2712  2786  2876  2983  5493  5576  5632  5696  5752  5815
                        5871  5934  5990  6053  6109  6172  6228  6291  6347  6419  6477  6544  6633
                        6690  6779  6840  6898  7254  7269  7274  7298  7312  7363
MIND  = 000200          1155#  2704  2706  2707  2966  2968  2969  3041  3042  3043  3130  3131  3132
                        3245  3246  3247  3374  3375  3376  3549  3550  3551  3667  3668  3669  3766
                        3767  3768  3828  3829  3830  3929  3930  3931  3991  3992  3993  4092  4093
                        4094  4154  4155  4156  4262  4263  4264  4328  4329  4330  4422  4423  4424
                        4557  4558  4559  5487  5488  5489  5561  5562  5563  5624  5626  5627  5681
                        5682  5683  5744  5746  5747  5800  5801  5802  5863  5865  5866  5919  5920
                        5921  5982  5984  5985  6038  6039  6040  6101  6103  6104  6157  6158  6159
                        6220  6222  6223  6276  6277  6278  6339  6341  6342  6403  6404  6405  6469
                        6471  6472  6528  6529  6530  6625  6627  6628  6674  6675  6676  6771  6773
                        6774  6824  6825  6826  6890  6892  6893
MMVEC = 000250          1005#
MSP   = 000100          1154#  2599  2615  2688  2769  2845  2910  2911  2912  4658  4768  4878  4988
                        5098  5207  5310  5374  5475  5514  5515  5520  5572  5692  5811  5930  6049
                        6168  6287  6416  6541  6687  6837
M1.BIT  003616          2248#  2620*  2655*  2714*  2737*  2788*  2811*  4661*  4673*  4697*  4771*  4783*  4807*
                        4881*  4893*  4917*  4991*  5003*  5027*  5101*  5113*  5137*  5210*  5222*  5246*  5313*
                        5379*  5391*  5415*  5503*  5580*  5598  5599*  5615  5616*  5700*  5718  5719*  5735
                        5736*  5819*  5837  5838*  5854  5855*  5938*  5956  5957*  5973  5974*  6057*  6075
                        6076*  6092  6093*  6176*  6194  6195*  6211  6212*  6295*  6313  6314*  6330  6331*
                        6423*  6440  6441*  6457  6458*  6548*  6565  6566*  6583  6584*  6597  6598*  6694*
                        6711  6712*  6729  6730*  6743  6744*  6844*  6861  6862*  6878  6879*  7272  7331
                        7337  7387  8489  8526
M2.BIT  003620          2249#  5504*  5581*  5598*  5615*  5701*  5718*  5735*  5820*  5837*  5854*  5939*  5956*
                        5973*  6058*  6075*  6092*  6177*  6194*  6211*  6296*  6313*  6330*  6424*  6440*  6457*
                        6549*  6565*  6583*  6597*  6695*  6711*  6729*  6743*  6845*  6861*  6878*  7277  7282
                        8526
NED   = 010000          1108#
NEM   = 004000          1107#  3684
NEWPAS  004662          2396#  7194
```

```
NPRBUF  066762          3117    3118    3145    3189    3233    3234    3260    3309    3362    3363    3389    3419    3536
                        3537    3611    3749    3800    3807    3808    3867    3912    3963    3970    3971    4030    4075
                        4126    4133    4134    4193    4411    4412    4438    4486    6663    6664    9730#
NXF     = 000004        1117#
OFFSET= 000015          1071#
OFST    = 000004        1136#
OPI     = 020000        1128#
OPR001  054414          2343    8746#
OPR002  054446          2346    2357    2374    8751#
OPR003  054454          2353    8752#
OPR004  054504          2365    8756#
OPR006  054526          2340    8759#
OPR007  054600          7814    8766#
OR      = 000200        1103#   3059    3136    3251    3282    3311    3380    3412    3441    3467    3491    3555    3585
                        3614    3685    3687    3784    3849    3947    4012    4110    4175    4281    4346    4370    4428
                        4446    4487    4563    4705    4815    4925    5035    5145    5254    5423
PACK    = 000003        1066#
PARCHK  045002          2396    7223#   8427
PARM    003636          1188    2258#
PAR.EN= 000001          1039#   7215    7230
PAT     = 000020        1152#
PATTER  066664          3016    9699#
PCA     = 004000        1159#   7268    7284    7333
PCD     = 010000        1160#   7268    7279    7339
PGE     = 002000        1106#
PIP     = 020000        1144#
PIRQ    = 177772         906#
PIRQVE= 000240          1000#
PR.BIT  003614          2247#   2619#   2646#   2655    2656#   2713#   2727#   2737    2738#   2787#   2801#   2811    2812#
                        4660#   4666#   4673    4676#   4679#   4697    4698#   4770#   4776#   4783    4786#   4789#   4807
                        4808#   4880#   4886#   4893    4896#   4899#   4917    4918#   4990#   4996#   5003    5006#   5009#
                        5027    5028#   5100#   5106    5113    5116#   5119#   5137    5138#   5209#   5215#   5222    5225#
                        5228#   5246    5247#   5312#   5378#   5384#   5391    5394#   5397#   5415    5416#   5502#   5579#
                        5599    5600#   5616    5617#   5699#   5719    5720#   5736    5737#   5818#   5838    5839#   5855
                        5856#   5937#   5957    5958#   5974    5975#   6056#   6076    6077#   6093    6094#   6175#   6195
                        6196#   6212    6213#   6294#   6314    6315#   6331    6332#   6422#   6441    6442#   6458    6459#
                        6547#   6566    6567#   6584    6585#   6598    6599#   6693#   6712    6713#   6730    6731#   6744
                        6745#   6843#   6862    6863#   6879    6880#   7270    7385    8489    8526
PR0     = 000000         923#   2397
PR1     = 000040         924#
PR2     = 000100         925#
PR3     = 000140         926#
PR4     = 000200         927#
PR5     = 000240         928#   2239
PR6     = 000300         929#
PR7     = 000340         930#   1201    2268    7224    8410    8424
PS      = 177776         903#    904
PSW     = 177776         904#
PWRVEC= 000024           995#   2287#   2288#   8409#   8410#   8423#   8424#
P.ASOF  003556          2227#
P.BA    003544          2222#
P.CS1   003540          2220#   6964#   7041#   7120#   8540
P.CS2   003550          2224#   6965#   7042#   7121#   8540
P.DA    003546          2223#
P.DB    003562          2229#
P.DCYL  003560          2228#
```

DZR6CA.P11 05-OCT-76 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

```
P.DS    003552          2225#  6966*  7043*  7122*  8540
P.ECPS  003572          2233#
P.ECPT  003574          2234#
P.ER    003554          2226#  6967*  7044*  7123*  8540
P.MR1   003564          2230#
P.MR2   003566          2231#
P.MR3   003570          2232#
P.SPAR  003576          2235#
P.WC    003542          2221#
P1.BIT  003612          2246#  5501*  5578*  5589*  5600   5603*  5606*  5617   5618*  5698*  5709*  5720   5723*
                        5726*  5737   5738*  5817*  5828*  5839   5842*  5845*  5856   5857*  5936*  5947*  5958
                        5961*  5964*  5975   5976*  6055*  6066*  6077   6080*  6083*  6094   6095*  6174*  6185*
                        6196   6199*  6202*  6213   6214*  6293*  6304*  6315   6318*  6321*  6332   6333*  6421*
                        6432*  6442   6445*  6448*  6459   6460*  6546*  6557*  6567   6570*  6573*  6585   6586*
                        6591*  6599   6600*  6692*  6703*  6713   6716*  6719*  6731   6732*  6737*  6745   6746*
                        6842*  6853*  6863   6866*  6869*  6880   6881*  7275   7329   8526
RDBIT   046106          2622   2635   2647   2657   2716   2728   2739   2790   2802   2813   4663   4667   4680
                        4699   4773   4777   4790   4809   4883   4887   4900   4919   4993   4997   5010   5029
                        5103   5107   5120   5139   5212   5216   5229   5248   5317   5381   5385   5398   5417
                        7385#
RDCHR = 104410          8249   8475#
RDDATA= 000021          1073#
RDGATE= 100000          1163#  2633
RDHEAD= 000025          1075#  2422   2432   2510   2520   2601   2608   2666   2690   2697   2747   2771   2778
                        2821   4652   4670   4704   4762   4780   4814   4872   4890   4924   4982   5000   5034
                        5092   5110   5144   5201   5219   5253   5304   5314   5335   5368   5388   5422   6932
                        6948   7009   7025
RDLIN = 104411          8323   8476#
RDOCT = 104412          2347   2358   2375   8477#
RDY   = 000200          1084#  3658   3706   4254   4301   4369   4704   4814   4924   5034   5144   5253   5335
                        5422   5642   5762   5881   6000   6119   6238   6357   6487   6644   6790   6908   6948
                        6974   7025   7051   7104   7130
RECAL = 000013          1070#
RESREG= 104414          7794   8479#
RESTRT  003646          1186   2261#
RESVEC= 000010           990#
RKASOF= 000016          1053#
RKBA  = 000004          1048#  2847*  2856   2892   2917   2949   2981   3030*  3053   3117*  3143   3174   3234*
                        3258   3294   3363*  3387   3427   3470   3537*  3562   3596   3655*  3680   3778   3819*
                        3841   3941   3982*  4004   4104   4145*  4167   4252*  4274   4318*  4340   4366   4411*
                        4436   4472   4546*  4570   4595   5477*  5551*  5671*  5790*  5909*  6028*  6147*  6266*
                        6392*  6517*  6663*  6813*  7087*
RKCS1 = 000000          1046#  2417*  2422*  2429   2438*  2462*  2467*  2474   2483*  2505*  2510*  2517   2526*
                        2549*  2554*  2561   2570*  2597*  2601*  2607   2612*  2627*  2640*  2652*  2662*  2665
                        2686*  2687*  2690*  2696   2701*  2721*  2733*  2743*  2746   2767*  2768*  2771*  2777
                        2782*  2795*  2807*  2817*  2820   2843*  2849*  2855   2891   2916   2948   2980   2996*
                        3025*  3033*  3052   3077   3115*  3122*  3142   3172   3196   3231*  3237*  3257   3292
                        3317   3360*  3366*  3386   3425   3447   3468   3498   3534*  3541*  3561   3594   3618
                        3652*  3659*  3678   3703*  3704   3755*  3759*  3777   3814*  3821*  3840   3876*  3918*
                        3922*  3940   3977*  3984*  4003   4039*  4081*  4085*  4103   4140*  4147*  4166   4202*
                        4248*  4255*  4273   4298*  4299   4314*  4321*  4339   4364   4390*  4407*  4414*  4435
                        4470   4498   4542*  4549*  4569   4593   4614   4624*  4650*  4652*  4681   4689*  4702
                        4716   4760*  4762*  4791   4799*  4812   4826   4870*  4872*  4901   4909*  4922   4936
                        4980*  4982*  5011   5019*  5032   5046   5090*  5092*  5121   5129*  5142   5156   5199*
                        5201*  5230   5238*  5251   5265   5302*  5304*  5318   5333   5343*  5366*  5368*  5399
                        5407*  5420   5434   5473*  5479*  5549*  5553*  5641   5669*  5673*  5761   5788*  5792*
```

DZR6CA.P11 05-OCT-76 10:06 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    5880    5907*   5911*   5999    6026*   6030*   6118    6145*   6149*   6237    6264*   6268*   6356
                    6390*   6395*   6486    6515*   6520*   6643    6661*   6666*   6789    6811*   6816*   6907    6932*
                    6940    6944    6969*   6970    7009*   7017    7021    7046*   7047    7088*   7096    7100    7125*
                    7126    7812*
RKCS2 = 000010      1050#   2421*   2466*   2509*   2553*   3031*   3058    3078    3150    3173    3197    3265    3293
                    3323    3394    3426    3451    3469    3504    3540*   3569    3595    3624    3679    3705    3783
                    3842    3946    4011    4109    4174    4280    4300    4345    4365    4443    4471    4499    4575
                    4594    4615    4703    4717    4813    4827    4923    4937    5033    5047    5143    5157    5253
                    5266    5319    5334    5421    5435    6929*   6945    6971    7006*   7022    7048    7083*   7101
                    7127
RKDA  = 000006      1049#   2420*   2465*   2508*   2552*   3029*
RKDB  = 000024      1055#   3072    3191    3312    3442    3493    3613    3799    3868    3962    4031    4125    4194
                    4490    4609    4715    4825    4935    5045    5155    5264    5332    5433
RKDCYL= 000020      1054#   2419*   2464*   2507*   2551*   3028*   6931*   7008*   7085*
RKDS  = 000012      1051#   6946    6972    7023    7049    7102    7128
RKECPS= 000030      1059#
RKECPT= 000032      1060#
RKER  = 000014      1052#   3682    4276    4368    6947    6973    7024    7050    7103    7129
RKMR1 = 000026      1056#   2418*   2425*   2426*   2463*   2470*   2471*   2506*   2513*   2514*   2550*   2557*   2558*
                    2598*   2599*   2600*   2603*   2604*   2615*   2616*   2623    2636    2648    2658    2688*   2689*
                    2692*   2693*   2704*   2706*   2707*   2710*   2717    2729    2740    2769*   2770*   2773*   2774*
                    2791    2803    2814    2844*   2845*   2846*   2851*   2852*   2879*   2880*   2881*   2882*   2883
                    2910*   2911*   2912*   2915*   2936*   2937*   2938*   2939*   2940    2966*   2968*   2969*   2972*
                    2975*   2976*   2979    3027*   3036*   3037*   3041*   3042*   3043*   3046*   3048*   3049*   3116*
                    3125*   3126*   3130*   3131*   3132*   3135*   3138*   3139*   3168*   3169*   3232*   3240*   3241*
                    3245*   3246*   3247*   3250*   3253*   3254*   3288*   3289*   3361*   3369*   3370*   3374*   3375*
                    3376*   3379*   3382*   3383*   3421*   3422*   3459*   3460*   3535*   3544*   3545*   3549*   3550*
                    3551*   3554*   3557*   3558*   3590*   3591*   3653*   3662*   3663*   3667*   3668*   3669*   3672*
                    3674*   3675*   3756*   3761*   3762*   3766*   3767*   3768*   3771*   3773*   3774*   3815*   3823*
                    3824*   3828*   3829*   3830*   3833*   3836*   3837*   3919*   3924*   3925*   3929*   3930*   3931*
                    3934*   3936*   3937*   3978*   3986*   3987*   3991*   3992*   3993*   3996*   3999*   4000*   4082*
                    4087*   4088*   4092*   4093*   4094*   4097*   4099*   4100*   4141*   4149*   4150*   4154*   4155*
                    4156*   4159*   4162*   4163*   4249*   4257*   4258*   4262*   4263*   4264*   4267*   4269*   4270*
                    4315*   4323*   4324*   4328*   4329*   4330*   4333*   4335*   4336*   4360*   4361*   4408*   4417*
                    4418*   4422*   4423*   4424*   4427*   4431*   4432*   4464*   4465*   4543*   4552*   4553*   4557*
                    4558*   4559*   4562*   4565*   4566*   4589*   4590*   4651*   4654*   4655*   4658*   4659*   4761*
                    4764*   4765*   4768*   4769*   4871*   4874*   4875*   4878*   4879*   4981*   4984*   4985*   4988*
                    4989*   5091*   5094*   5095*   5098*   5099*   5200*   5203*   5204*   5207*   5208*   5303*   5306*
                    5307*   5310*   5311*   5367*   5370*   5371*   5374*   5375*   5376*   5377*   5474*   5475*   5476*
                    5482*   5483*   5487*   5488*   5489*   5492*   5496*   5497*   5515*   5516    5522*   5523    5550*
                    5556*   5557*   5561*   5562*   5563*   5566*   5568*   5569*   5572*   5573*   5624*   5626*   5627*
                    5630*   5631    5637*   5638*   5670*   5676*   5677*   5681*   5682*   5683*   5686*   5688*   5689*
                    5692*   5693*   5744*   5746*   5747*   5750*   5751    5757*   5758*   5789*   5795*   5796*   5800*
                    5801*   5802*   5805*   5807*   5808*   5811*   5812*   5863*   5865*   5866*   5869*   5870    5876*
                    5877*   5908*   5914*   5915*   5919*   5920*   5921*   5924*   5926*   5927*   5930*   5931*   5982*
                    5984*   5985*   5988*   5989    5995*   5996*   6027*   6033*   6034*   6038*   6039*   6040*   6043*
                    6045*   6046*   6049*   6050*   6101*   6103*   6104*   6107*   6108    6114*   6115*   6146*   6152*
                    6153*   6157*   6158*   6159*   6162*   6164*   6165*   6168*   6169*   6220*   6222*   6223*   6226*
                    6227    6233*   6234*   6265*   6271*   6272*   6276*   6277*   6278*   6281*   6283*   6284*   6287*
                    6288*   6339*   6341*   6342*   6345*   6346    6352*   6353*   6391*   6398*   6399*   6403*   6404*
                    6405*   6409*   6412*   6413*   6416*   6417*   6469*   6471*   6472*   6475*   6476    6482*   6483*
                    6516*   6523*   6524*   6528*   6529*   6530*   6533*   6535*   6536*   6541*   6542*   6625*   6627*
                    6628*   6631*   6632    6639*   6640*   6662*   6669*   6670*   6674*   6675*   6676*   6679*   6681*
                    6682*   6687*   6688*   6771*   6773*   6774*   6777*   6778    6785*   6786*   6812*   6819*   6820*
                    6824*   6825*   6826*   6830*   6833*   6834*   6837*   6838*   6890*   6892*   6893*   6896*   6897
                    6903*   6904*   6930*   6934*   6935*   6938*   7007*   7011*   7012*   7015*   7084*   7090*   7091*
```

```
                    7094*  7255*  7256   7285*  7286   7299*  7300   7314*  7315   7340*  7341   7352*  7353
                    7364*  7365   7389*  7392*  7393*  7394*  7395*  7398*  7399*  7400*  7401*
RKMR2 = 000034      1057#  2430   2475   2518   2562
RKMR3 = 000036      1058#  2431   2476   2519   2563
RKPRI   003602      2239#  2391*
RKSPAR= 000022      1061#
RKVEC   003600      2238#  2389*  2390*
RKWC  = 000002      1047#  2848*  2857   2893   2918   2950   2982   3032*  3054   3121*  3144   3175   3236*
                    3259   3295   3365*  3388   3428   3471   3539*  3563   3597   3657*  3681   3757*  3779
                    3817*  3842   3920*  3942   3980*  4005   4083*  4105   4143*  4168   4250*  4275   4316*
                    4341   4367   4409*  4437   4473   4544*  4571   4596   5478*  5552*  5672*  5791*  5910*
                    6029*  6148*  6267*  6394*  6519*  6665*  6815*  7086*
RLS   = 000010      1099#
SAVREG= 104413      7743   8478#
SAVSWR  003634      2255#  8408*  8428
SCLR  = 000040      1101#  6929   7006   7083
SCOP1 = 104415      3086   3210   3804   3875   3967   4038   4130   4201   4309   4389   8480#
SCOP1$  047226      7615#  8480
SECCNT  003626      2252#  5500*  5574*  5694*  5813*  5932*  6051*  6170*  6289*  6408*  6466*  6539*  6620*
                    6685*  6766*  6829*  6887*  8526
SEEK  = 000017      1072#
SELDRV= 000001      1065#
SKI   = 000002      1116#
SPACE2  054644      7777   7789   8772#
SPAR  = 020000      1091#
SPDLSS= 000020      1138#
SRTFLG  003606      2241#  2258*  2261*  2264*  2341
SRTSPL= 000011      1069#
SR0   = 177572      1009#  3748*  3750*  3806*  3809*  3911*  3913*  3969*  3972*  4074*  4076*  4132*  4135*
                    7448   7463*  7470*
SR1   = 177574      1010#
SR2   = 177576      1011#
SR3   = 172516      1012#  7460*
STACK = 001100       894#  2266   2279   7380   7414   7805   7813   8425
START   003656      1185   2264#
START1  003662      2259   2262   2265#  7823
STKLMT= 177774       905#
SVAL  = 100000      1146#  6950   7027   7106
SWR     001140      1252#  2277   2299*  2301   2307*  2314*  2329   7264   7294   7308   7323   7349   7361
                    7373   7410   7510   7524   7527   7542   7549   7615   7697   7704   7716   7720   7796
                    8093   8132   8187*  8408   8428*
SWREG   000176      1183#  2307   2329   8093   8132   8155
SW0   = 000001       958#
SW00  = 000001       948#   958
SW01  = 000002       947#   957
SW02  = 000004       946#   956
SW03  = 000010       945#   955
SW04  = 000020       944#   954
SW05  = 000040       943#   953
SW06  = 000100       942#   952
SW07  = 000200       941#   951
SW08  = 000400       940#   950
SW09  = 001000       939#   949
SW1   = 000002       957#
SW10  = 002000       938#
SW11  = 004000       937#
```

```
SW12  = 010000           936#   7796
SW13  = 020000           935#
SW14  = 040000           934#
SW15  = 100000           933#
SW2   = 000004           956#
SW3   = 000010           955#
SW4   = 000020           954#
SW5   = 000040           953#
SW6   = 000100           952#
SW7   = 000200           951#
SW8   = 000400           950#
SW9   = 001000           949#   7264   7294   7308   7323   7349   7361   7373   7410   7615
S.CLR = 000400          1171#   2521   2565
S.FMT = 001000          1172#   2433   2478   2521   2565
S.PACK= 004000          1174#
S.RECL= 000040          1168#
S.RTC = 000200          1170#   2478
S.SEEK= 000020          1167#   2433   2478
S.STSP= 000100          1169#
S.UNLD= 002000          1173#
TBITVE= 000014           991#
TKVEC = 000060           998#   8070*  8071*
TPVEC = 000064           999#
TRAPPC  003604          2240#   7407*  8483
TRAPVE= 000034           997#   2285*  2286*
TRTVEC= 000014           992#
TSTBY1  054734          3740    3903   4066   4236   4534   8782#
TSTBY2  054744          3743    3906   4069   4239   4537   8784#
TST1    004700          2398    2414#  7569
TST10   007462          2783    2796   2808   2818   2823   2840#  7576
TST11   010606          3013#   7577
TST12   011274          3089    3105#  7578
TST13   012160          3213    3227#  7579
TST14   013046          3356#   7580
TST15   014252          3530#   7581
TST16   015130          3649#   7582
TST17   015566          3712    3730#  7583
TST2    005116          2439    2444   2459#  7570
TST20   016702          3744    3893#  7584
TST21   020016          3907    4056#  7585
TST22   021132          4070    4229#  7586
TST23   022370          4240    4404#  7587
TST24   023260          4527#   7588
TST25   024104          4538    4647#  7589
TST26   024636          4690    4724   4729   4757#  7590
TST27   025370          4800    4834   4839   4867#  7591
TST3    005334          2484    2489   2502#  7571
TST30   026122          4910    4944   4949   4977#  7592
TST31   026654          5020    5054   5059   5087#  7593
TST32   027406          5130    5164   5169   5196#  7594
TST33   030140          5239    5273   5278   5299#  7595
TST34   030440          5323    5328   5363#  7596
TST35   031206          5408    5442   5447   5470#  7597
TST36   031576          5525    5546#  7598
TST37   032422          5644    5666#  7599
TST4    005550          2527    2532   2546#  7572
```

```
TST40   033246          5764    5785#   7600
TST41   034072          5883    5904#   7601
TST42   034716          6002    6023#   7602
TST43   035542          6121    6142#   7603
TST44   036366          6240    6261#   7604
TST45   037212          6359    6387#   7605
TST46   040052          6489    6512#   7606
TST47   041076          6646    6658#   7607
TST5    005764          2571    2576    2594#   7573
TST50   042122          6792    6808#   7608
TST51   042762          6910    6926#   7609
TST52   043406          7003#   7610
TST53   044032          7080#   7611
TST6    006446          2613    2628    2641    2653    2663    2668    2683#   7574
TST7    007074          2702    2722    2734    2744    2749    2764#   7575
TYPDS = 104405          7175    7182    8470#
TYPE  = 104401          2323    2340    2343    2346    2353    2357    2365    2374    3740    3743    3903    3906    4066
                        4069    4236    4239    4534    4537    7169    7176    7183    7699    7707    7753    7754    7758
                        7759    7766    7777    7779    7789    7790    7792    7800    7814    7868    7964    8039    8087
                        8099    8153    8154    8157    8168    8178    8189    8208    8256    8262    8267    8271    8276
                        8277    8279    8282    8286    8352    8354    8426    8466#
TYPERR  047720          7706    7743#
TYPOC = 104402          2345    2356    2373    3742    3905    4068    4238    4536    7774    8156    8467#
TYPON = 104404          8469#
TYPOS = 104403          8468#
T.ASOF  003456          2189#
T.BA    003444          2184#   2856*   2866    2892*   2899    2917*   2924    2949*   2956    2981*   2987    3053*   3066
                        3143*   3159    3174*   3182    3258*   3272    3294*   3302    3387*   3401    3427*   3435    3470*
                        3478    3562*   3576    3596*   3604    3680*   3697    3778*   3793    3841*   3856    3941*   3956
                        4004*   4019    4104*   4119    4167*   4182    4274*   4289    4340*   4353    4366*   4380    4436*
                        4453    4472*   4480    4570*   4582    4595*   4603    8492    8497    8501    8517
T.CS1   003440          2182#   2429*   2435    2474*   2480    2517*   2523    2561*   2567    2607*   2609    2665*   2667
                        2696*   2698    2746*   2748    2777*   2779    2820*   2822    2855*   2861    2891*   2894    2916*
                        2919    2948*   2951    2980*   2984    3052*   3060    3077*   3080    3142*   3153    3172*   3176
                        3196*   3201    3257*   3266    3292*   3296    3317*   3327    3386*   3395    3425*   3429    3447*
                        3452    3468*   3472    3498*   3508    3561*   3570    3594*   3598    3618*   3628    3678*   3688
                        3704*   3708    3777*   3787    3840*   3850    3940*   3950    4003*   4013    4103*   4113    4166*
                        4176    4273*   4283    4299*   4303    4339*   4347    4364*   4374    4435*   4447    4470*   4474
                        4498*   4503    4569*   4576    4593*   4597    4614*   4617    4681*   4682    4702*   4706    4716*
                        4721    4791*   4792    4812*   4816    4826*   4831    4901*   4902    4922*   4926    4936*   4941
                        5011*   5012    5032*   5036    5046*   5051    5121*   5122    5142*   5146    5156*   5161    5230*
                        5231    5251*   5255    5265*   5270    5318*   5320    5333*   5337    5399*   5400    5420*   5424
                        5434*   5439    5641*   5643    5761*   5763    5880*   5882    5999*   6001    6118*   6120    6237*
                        6239    6356*   6358    6486*   6488    6643*   6645    6789*   6791    6907*   6909    6944*   6952
                        6964    6970*   6978    7021*   7029    7041    7047*   7055    7100*   7108    7120    7126*   7134
                        8484    8487    8492    8497    8501    8507    8511    8514    8519    8521    8523    8532    8536
T.CS2   003450          2186#   3058*   3063    3078*   3083    3150*   3156    3173*   3179    3197*   3204    3265*   3269
                        3293*   3299    3323*   3330    3394*   3398    3426*   3432    3451*   3455    3469*   3475    3504*
                        3511    3569*   3573    3595*   3601    3624*   3631    3679*   3685    3691    3705*   3711    3783*
                        3790    3848*   3853    3946*   3953    4011*   4016    4109*   4116    4174*   4179    4280*   4286
                        4300*   4306    4345*   4350    4365*   4377    4443*   4450    4471*   4477    4499*   4506    4575*
                        4579    4594*   4600    4615*   4620    4703*   4709    4717*   4726    4813*   4819    4827*   4836
                        4923*   4929    4937*   4946    5033*   5039    5047*   5056    5143*   5149    5157*   5166    5252*
                        5258    5266*   5275    5319*   5325    5334*   5340    5421*   5427    5435*   5444    6945*   6955
                        6965    6971*   6981    7022*   7032    7042    7048*   7058    7101*   7111    7121    7127*   7137
                        8501    8507    8511    8514    8519    8532    8536
```

```
T.DA     003446  2:85#
T.DB     003462  2191#  3072*  3074   3191*  3193   3312*  3314   3442*  3444   3493*  3495   3613*  3615
                 3799*  3801   3868*  3870   3962*  3964   4031*  4033   4125*  4127   4194*  4196   4490*
                 4492   4609*  4611   4715*  4731   4825*  4841   4935*  4951   5045*  5061   5155*  5171
                 5264*  5280   5433*  5449   8505   8509
T.DCYL   003460  2190#
T.DS     003452  2187#  6946*  6958   6966   6972*  6984   7023*  7035   7043   7049*  7061   7102*  7114
                 7122   7128*  7140   8532   8536
T.ECPS   003472  2195#
T.ECPT   003474  2196#
T.ER     003454  2188#  3682*  3694   4276*  4295   4368*  4386   6947*  6961   6967   6973*  6987   7024*
                 7038   7044   7050*  7064   7103*  7117   7123   7129*  7143   8514   8532   8536
T.MR1    003464  2192#  2623*  2624   2636*  2637   2648*  2649   2658*  2659   2717*  2718   2729*  2730
                 2740*  2741   2791*  2792   2803*  2804   2814*  2815   2883*  2884   2940*  2941   2979*
                 2993   5516*  5517   5523*  5524   5631*  5633   5751*  5753   5870*  5872   5989*  5991
                 6108*  6110   6227*  6229   6346*  6348   6476*  6478   6632*  6634   6778*  6780   6897*
                 6899   7256*  7257   7286*  7287   7300*  7301   7315*  7316   7341*  7342   7353*  7354
                 7365*  7366   8489   8495   8497   8526   8530
T.MR2    003466  2193#  2430*  2440   2475*  2485   2518*  2528   2562*  2572   8484
T.MR3    003470  2194#  2431*  2443   2476*  2488   2519*  2531   2563*  2575   8484
T.SPAR   003476  2197#
T.WC     003442  2183#  2857*  2871   2893*  2904   2918*  2929   2950*  2961   2982*  2990   3054*  3069
                 3144*  3162   3175*  3185   3259*  3275   3295*  3305   3388*  3404   3428*  3438   3471*
                 3481   3563*  3579   3597*  3607   3681*  3700   3779*  3796   3842*  3859   3942*  3959
                 4005*  4022   4105*  4122   4168*  4185   4275*  4292   4341*  4356   4367*  4383   4437*
                 4456   4473*  4483   4571*  4585   4596*  4606   8492   8497   8504   8517
UFE    = 000400  1104#
UNLOAD= 000007  1068#
UNS    = 040000  1129#
UPE    = 020000  1109#  4281   4370
VV     = 000100  1140#
WAITIM   003632  2254#  6939   7016   7095
WCE    = 040000  1110#
WLE    = 004000  1126#
WRBUFF   067204  2847   2859   3018   3026*  3030   5477   7087   9803#  9804
WRDATA= 000023  1074#
WRDCNT   003624  2251#  3190*  3207*  3310*  3318   3333*  3418*  3484*  3499   3514*  3612*  3619   3634*
                 4489*  4509*  4684*  4685*  4712*  4718   4734*  4735   4794*  4795*  4822*  4828   4844*
                 4845   4904*  4905*  4932*  4938   4954*  4955   5014*  5015*  5042*  5048   5064*  5065
                 5124*  5125*  5152*  5158   5174*  5175   5233*  5234*  5261*  5267   5283*  5284   5402*
                 5403*  5430*  5436   5452*  5453   8509   8511   8521
WRHEAD= 000027  1076#  2467   2477   2554   2564   2849   2858   3019   3033   3110   3122   3230   3237
                 3359   3366   3533   3541   3658   3659   3758   3759   3820   3821   3921   3922   3983
                 3984   4084   4085   4146   4147   4254   4255   4320   4321   4369   4413   4414   4548
                 4549   5479   5553   5642   5673   5762   5792   5881   5911   6000   6030   6119   6149
                 6238   6268   6357   6395   6487   6520   6644   6666   6790   6816   6908   7088   7104
WRL    = 004000  1143#
WRTBIT   045136  5508   5584   5590   5607   5619   5704   5710   5727   5739   5823   5829   5846   5858
                 5942   5948   5965   5977   6061   6067   6084   6096   6180   6186   6203   6215   6299
                 6305   6322   6334   6427   6433   6449   6461   6552   6558   6574   6587   6592   6601
                 6608   6615   6698   6704   6720   6733   6738   6747   6754   6761   6848   6854   6870
                 6882   7254#
WRTCHK= 000031  1077#
WRTGAT= 040000  1162#  2983   5493   5513   5521   5576   5696   5815   5934   6053   6172   6291   6419
                 6544   6690   6840   7269
WRTPAR   044702  4243   4541   7200#
```

```
WR.PAR= 000004          1038#   7205
$APTHD  001000          1212    1218#
$ASTAT= ****** U        7655    7670
$ATYC   047276          7626    7628#
$ATY1   047252          7624#
$ATY3   047260          7625#   7853
$ATY4   047270          7627#   7712
$AUTOB  001134          1249#   2333*   8146    8303
$BASE   001270          1321#   2344    2352*   2393    2416    2461    2504    2548    2596    2685    2766    2842    3015
                        3107    3229    3358    3532    3651    3746    3909    4072    4242    4406    4540    4649    4759
                        4869    4979    5089    5198    5301    5365    5472    5548    5668    5787    5906    6025    6144
                        6263    6389    6514    6660    6810    6928    7005    7082    7811    8429
$BDADR  001122          1244#
$BDDAT  001126          1246#
$BELL   001204          1272#   7699    7732    8099    8296
$CDW1   001274          1323#
$CDW2   001276          1324#
$CHARC  050536          7870*   7880*   7887    7896*   7901#
$CKSWR  051512          8132#   8474
$CMTAG  001100          1232#   2274    2275    2283    2289    2290    2291
$CM3  = 000000          1262#
$CM4  = 000010          1262#   1263#   1264#   1265#   1266#   1267#   1268#   1269#   1270#
$CNTLC  052410          8087    8168    8296#
$CNTLG  052422          8153    8298#
$CNTLU  052415          8178    8271    8297#
$CORE   046466          7446    7473#
$CPUOP  001242          1295#
$CRLF   001211          1274#   7183    7707    7732    7753    7758    7766    7779    7792    7869    7904    8189    8276
                        8296    8357
$CROUT  046516          7473    7480#
$DBLK   051204          8005    8039    8047#
$DEVCT  001224          1286#
$DEVM   001272          1322#
$DOAGN  044672          7165    7186    7192#
$DTBL   051174          8008    8043#
$ENDAD  044662          1195    2321    7188#   7727
$ENDCT  044526          2289    7167#
$ENULL  044676          7195#
$ENV    001234          1291#   2327    7633    7657    7709    7848
$ENVM   001235          1292#   2312    7635    7850    7855
$EOP    044472          7157#   7806    7820
$EOPCT  044520          2289*   7164#   7168    7804*   7819*
$ERFLG  001103          1235#   7499    7538    7540    7546*   7568    7617    7694*   7732    7817*
$ERMAX  001115          1241#   2292*   7540    7563*   7568
$ERROR  047520          2283    7692#
$ERRPC  001116          1242#   7701*   7702*   7703    7732    8484    8487    8489    8492    8495    8497    8501    8505
                        8507    8509    8511    8514    8519    8521    8523    8526    8530    8532    8536
$ERRTB  001300          1341#   7750
$ERTTL  001112          1239#   7180    7184*   7700*   7732
$ESCAP  001202          1271#   2291*   2394*   7259*   7289*   7303*   7318*   7325*   7344*   7356*   7368*   7375*   7379*
                        7406*   7409*   7562*   7723    7725    7732    7818*
$ETABL  001234          1290#
$ETEND  001300          1224    1325#
$FATAL  001216          1283#   7661*
$FFLG   047516          7624*   7627*   7655    7664*   7672#
$FILLC  001156          1260#   7873    7904
```

```
$FILLS  001155          1259#   7904
$GDADR  001120          1243#
$GDDAT  001124          1245#
$GET42  044652          7185#
$GTSWR  051602          8154#   8472
$HD     =  000000        879
$HIBTS  001000          1219#
$HIOCT  052610          8345*   8356#
$ICNT   001104          1236#   7553*   7554    7556*   7567
$INTAG  001135          1250#   8151*   8170    8190    8303
$ITEMB  001114          1240#   7703*   7711    7732    7744
$KTNEX  046460          7447    7472#
$KTOUT  046450          7464    7469#
$KT11   046316          3732    3895    4058    7445#   7449*   7472*
$LF     001212          1275#   7732    7904    8286    8296    8357
$LFLG   047515          7665*   7671#
$LPADR  001106          1237#   2293*   7535*   7544*   7560*   7565    7567    8430
$LPERR  001110          1238#   2294*   3021*   3111*   3751*   3810*   3914*   3973*   4077*   4136*   4244*   4310*   7381
                        7415    7544    7561*   7567    7619    7722
$LSTAD  046562          7485*   7492#
$LSTBK  046564          3734    3897    4060    7486*   7493#
$MADR1  001246          1308#
$MADR2  001252          1312#
$MADR3  001256          1315#
$MADR4  001262          1318#
$MAIL   001214          1220    1224    1281#   2311    2327    7559    7709    7848
$MAMS1  001244          1302#
$MAMS2  001250          1310#
$MAMS3  001254          1313#
$MAMS4  001260          1316#
$MBADR  001002          1220#
$MFLG   047514          7625*   7631    7666*   7670#
$MNEW   052440          8157    8301#
$MSGAD  001230          1288#   7641*   7644
$MSGLG  001232          1289#   7646*
$MSGTY  001214          1282#   7639    7647*   7659    7663*
$MSWR   052427          8154    8299#
$MTYP1  001245          1303#
$MTYP2  001251          1311#
$MTYP3  001255          1314#
$MTYP4  001261          1317#
$MXCNT  047076          7557    7567#
$NULL   001154          1258#   7875    7904
$NWTST=  000001         2402#   2404    2447#   2449    2492#   2494    2535#   2537    2582#   2584    2671#   2673    2752#
                        2754    2826#   2828    3000#   3002    3091#   3093    3216#   3218    3337#   3339    3518#   3520
                        3638#   3640    3715#   3717    3878#   3880    4041#   4043    4204#   4206    4393#   4395    4513#
                        4515    4628#   4630    4738#   4740    4848#   4850    4958#   4960    5068#   5070    5178#   5180
                        5287#   5289    5345#   5347    5458#   5460    5528#   5530    5647#   5649    5767#   5769    5886#
                        5888    6005#   6007    6124#   6126    6243#   6245    6362#   6364    6492#   6494    6649#   6651
                        6795#   6797    6915#   6917    6992#   6994    7069#   7071
$OCNT   050764          7936*   7965*   7978#
$OMODE  050766          7931*   7935*   7940    7943*   7954*   7980#
$OVER   047062          7511    7536    7545    7555    7564#
$PASS   001222          1285#   2311*   7161*   7162*   7173    7195    7551    7568
$PASTM  001006          1222#
$POWER  053034          8426    8433#
```

```
$PWRCT  053030          8417*  8418*  8419*  8421*  8432#
$PWRDN  052706          2287   8408#  8423
$PWRUP  052734          8409   8417#
$QUES   001210          1273#  7732   7904   8209   8279   8296   8354   8357
$RDCHR  052054          8221#  8475
$RDDEC= ****** U        8478
$RDLIN  052144          8244#  8476
$RDOCT  052452          8318#  8477
$RDSZ = 000010          8237#
$RESRE  052650          8390#  8479
$RTNAD  044674          7194#
$R2A  = ****** U        8480
$SAVRE  052612          8374#  8478
$SCOPE  046566          2281   7508#
$SETUP= 000137          2258#  2280   2281   2283   2285   2287   2289   2290   2291   2293   2321   2324   7159
                        7509   7693   7719   7727   8091   8096   8097   8127   8303
$SIZE   046260          2392   7433#
$SIZEX  046522          7471   7481#
$STUP = 177777          2258#
$SVLAD  047026          7519   7558#
$SVPC = 000220          1193#  1198
$SWR  = 167400           867#   879    883    884    885    886    887    888    889    890   1270   1271   1272
                        2290   2291   2293   2294   2415   2460   2503   2547   2595   2684   2765   2841   3014
                        3106   3228   3357   3531   3650   3731   3894   4057   4230   4405   4528   4648   4758
                        4868   4978   5088   5197   5300   5364   5471   5547   5667   5786   5905   6024   6143
                        6262   6388   6513   6659   6809   6927   7004   7081   7154   7160   7187   7193   7195
                        7500   7501   7502   7503   7504   7510   7522   7524   7525   7538   7539   7540   7547
                        7548   7549   7561   7564   7567   7684   7685   7686   7687   7688   7697   7704   7716
                        7720   7732
$SWREG  001236          1293#  2314
$SWRMK= 000000           890    891   7504   7505   7528
$SWOBT  047100          7534   7568#
$TESTN  001220          1284#  3741   3904   4067   4237   4535   7559*  8483   8484   8487   8489   8492   8495
                        8497   8501   8505   8507   8509   8511   8514   8519   8521   8523   8526   8530   8532
                        8536
$TIMES  001200          1270#  2290*  2415*  2460*  2503*  2547*  2595*  2684*  2765*  2841*  3014*  3106*  3228*
                        3357*  3531*  3650*  3731*  3737*  3894*  3900*  4057*  4063*  4230*  4233*  4405*  4528*
                        4531*  4648*  4758*  4868*  4978*  5088*  5197*  5300*  5364*  5471*  5547*  5667*  5786*
                        5905*  6024*  6143*  6262*  6388*  6513*  6659*  6809*  6927*  7004*  7081*  7160*  7547*
                        7554   7557*  7567
$TKB    001146          1255#  8051   8072   8083   8108   8136   8163
$TKCNT  051214          8052#  8067*  8097   8114*  8228   8230*
$TKINT  051224          2267   8067#  8088   8149
$TKQEN= 051223          8056#  8122   8233
$TKQIN  051216          8053#  8068*  8069   8120*  8121*  8122   8124*
$TKQOU  051220          8054#  8069*  8231   8232*  8233   8235*
$TKQSR  051222          8055#  8068   8124   8235
$TKS    001144          1254#  8051   8073*  8104*  8106   8112*  8134   8150*  8160   8172*  8192*
$TKSRV  051274          8070   8083#
$TMP0   001160          1262#  2348*  2350   2352   2359*  2361   2364   2376*  2378   2380   2382*  2383*  2384*
                        2385*  2386*  2388   3108*  3119   3188   3211*
$TMP1   001162          1263#
$TMP2   001164          1264#
$TMP3   001166          1265#
$TMP4   001170          1266#
$TMP5   001172          1267#
```

```
$TMP6   001174          1268#
$TMP7   001176          1269#
$TN   = 000054           868#   879   2402  2415#  2439  2444  2447  2460#  2484  2489  2492  2503#  2527
                        2532  2535  2547#  2571  2576  2582  2595#  2613  2628  2641  2653  2663  2668
                        2671  2684#  2702  2722  2734  2744  2749  2752  2765#  2783  2796  2808  2818
                        2823  2826  2841#  3000  3014#  3089  3091  3106#  3213  3216  3228#  3337  3357#
                        3518  3531#  3638  3650#  3712  3715  3731#  3736  3878  3894#  3899  4041  4057#
                        4062  4204  4230#  4233  4393  4405#  4513  4528#  4531  4628  4648#  4690  4724
                        4729  4738  4758#  4800  4834  4839  4848  4868#  4910  4944  4949  4958  4978#
                        5020  5054  5059  5068  5088#  5130  5164  5169  5178  5197#  5239  5273  5278
                        5287  5300#  5323  5328  5345  5364#  5408  5442  5447  5458  5471#  5525  5528
                        5547#  5644  5647  5667#  5764  5767  5786#  5883  5886  5905#  6002  6005  6024#
                        6121  6124  6143#  6240  6243  6262#  6359  6362  6388#  6489  6492  6513#  6646
                        6649  6659#  6792  6795  6809#  6910  6915  6927#  6992  7004#  7069  7081#  7529
                        7569
$TPB    001152          1257#  7893*  7904
$TPFLG  001157          1261#  7842   7904
$TPS    001150          1256#  7891   7904
$TRAP   053042          2285   8443#
$TRAP2  053064          8454#  8465
$TRP  = 000016          8458#  8467#  8468#  8469#  8470#  8471#  8472  8473#  8474  8475#  8476#  8477#  8478#
                        8479#  8480#  8481#
$TRPAD  053076          8448   8465#
$TSTM   001004          1221#
$TSTNM  001102          1234#  7159*  7499  7531*  7558*  7559  7564  7568  7696  7732
$TTYIN  052400          8246   8247   8259  8277   8291   8295#
$TYPBN= ****** U        8471
$TYPDS  050770          7993#  8470
$TYPE   050260          7652   7842#  8458  8466
$TYPEC  050472          7872   7879   7886  7891#  7892   8194
$TYPEX  050540          7897   7899   7902#
$TYPOC  050566          7934#  8467
$TYPON  050602          7933   7936#  8469
$TYPOS  050542          7929#  8468
$UNIT   001226          1287#
$UNITM  001010          1223#
$USWR   001240          1294#
$VECT1  001264          1319#  2354   2363*  2364*  2367   2387*  2388*  2389  2391
$VECT2  001266          1320#
$XTSTR  046600          7513#
$$GET4= 000000          7187#
$$SWOB= 000054          7568#  7569   7570#  7571#  7572#  7573#  7574#  7575#  7576#  7577#  7578#  7579#  7580#
                        7581#  7582#  7583#  7584#  7585#  7586#  7587#  7588#  7589#  7590#  7591#  7592#  7593#
                        7594#  7595#  7596#  7597#  7598#  7599#  7600#  7601#  7602#  7603#  7604#  7605#  7606#
                        7607#  7608#  7609#  7610#  7611#  7612#
$OFILL  050765          7930*  7934*  7944  7979#
$OCAT=  ****** U        7510   7706
.     = 067410          1177#  1181#  1187#  1193   1194#  1196#  1198#  1199#  1202#  1208   1209#  1211#  1213#
                        1231#  1276   2278  2293   2294   2337#  7172#  7195   7196#  7567   7568   7673#  7732
                        7904   8047#  8051  8055#  8056   8057#  8295#  8296   8303#  8357   8412   9803#
.$ASTA= ****** U        7625   7628
.$X   = 001000          1208#  1213
```

```
BYPASS  1225#  3736  3899  4062  4233  4531
CLRMSG  1225#  2504  2548
CLRPSW  1225#
COMMEN  1001#
ENDCOM  1001#
ERROR    895#  2437  2442  2445  2482  2487  2490  2525  2530  2533  2569  2574  2577  2611  2626
         2639  2651  2661  2669  2700  2720  2732  2750  2781  2794  2806  2824  2863  2868  2873
         2886  2896  2901  2906  2921  2926  2931  2943  2953  2958  2963  2986  2989  2992  2995
         3062  3065  3068  3071  3076  3082  3085  3155  3158  3161  3164  3178  3181  3184  3187
         3195  3203  3206  3268  3271  3274  3277  3298  3301  3304  3307  3316  3329  3332  3397
         3400  3403  3406  3431  3434  3437  3440  3446  3454  3457  3474  3477  3480  3483  3497
         3510  3513  3572  3575  3578  3581  3600  3603  3606  3609  3617  3630  3633  3690  3693
         3696  3699  3702  3710  3713  3789  3792  3795  3798  3803  3852  3855  3858  3861  3872
         3952  3955  3958  3961  3966  4015  4018  4021  4024  4035  4115  4118  4121  4124  4129
         4178  4181  4184  4187  4198  4285  4288  4291  4294  4297  4305  4308  4349  4352  4355
         4358  4376  4379  4382  4385  4388  4449  4452  4455  4458  4476  4479  4482  4485  4494
         4505  4508  4578  4581  4584  4587  4599  4602  4605  4608  4613  4619  4622  4688  4708
         4711  4723  4728  4733  4798  4818  4821  4833  4838  4843  4908  4928  4931  4943  4948
         4953  5018  5038  5041  5053  5058  5063  5128  5148  5151  5163  5168  5173  5237  5257
         5260  5272  5277  5282  5322  5327  5339  5342  5406  5426  5429  5441  5446  5451  5509
         5519  5526  5585  5591  5608  5620  5635  5645  5705  5711  5728  5740  5755  5765  5824
         5830  5847  5859  5874  5884  5943  5949  5966  5978  5993  6003  6062  6068  6085  6097
         6112  6122  6181  6187  6204  6216  6231  6241  6300  6306  6323  6335  6350  6360  6428
         6434  6450  6462  6480  6490  6553  6559  6575  6588  6593  6602  6609  6616  6637  6647
         6699  6705  6721  6734  6739  6748  6755  6762  6783  6793  6849  6855  6871  6883  6901
         6911  6954  6957  6960  6963  6980  6983  6986  6989  7031  7034  7037  7040  7057  7060
         7063  7066  7110  7113  7116  7119  7136  7139  7142  7145  7408
ESCAPE  1001#
FORERR  1225#  6928  7005  7082
GETPRI  1001#  7441
GETSWR  1001#  2324#
LDLPER  1225#  3021  3111  3751  3810  3914  3973  4077  4136  4244  4310
MSG     2402#  2404  2447#  2449  2492#  2494  2535#  2537  2582#  2584  2671#  2673  2752#  2754  2826#
        2828   3000#  3002  3091#  3093  3216#  3218  3337#  3339  3518#  3520  3638#  3640  3715#  3717
        3878#  3880  4041#  4043  4204#  4206  4393#  4395  4513#  4515  4628#  4630  4738#  4740  4848#
        4850   4958#  4960  5068#  5070  5178#  5180  5287#  5289  5345#  5347  5458#  5460  5528#  5530
        5647#  5649  5767#  5769  5886#  5888  6005#  6007  6124#  6126  6243#  6245  6362#  6364  6492#
        6494   6649#  6651  6795#  6797  6915#  6917  6992#  6994  7069#  7071
MULT    1001#
NEWTST  1001#  2402  2447  2492  2535  2582  2671  2752  2826  3000  3091  3216  3337  3518  3638
        3715   3878  4041  4204  4393  4513  4628  4738  4848  4958  5068  5178  5287  5345  5458
        5528   5647  5767  5886  6005  6124  6243  6362  6492  6649  6795  6915  6992  7069
POP     1001#  7667  7668  8034  8346  8395
PUSH    1001#  7628  7630  7651  7993  8320  8375
RDLOOP  1225#  4649  4759  4869  4979  5089  5365
REPORT  1001#
SCOPE    896#  2414  2459  2502  2546  2594  2683  2764  2840  3013  3105  3227  3356  3530  3649
        3730   3893  4056  4229  4404  4527  4647  4757  4867  4977  5087  5196  5299  5363  5470
        5546   5666  5785  5904  6023  6142  6261  6387  6512  6658  6808  6926  7003  7080  7158
SEKMSG  1225#  2416  2461
SETPRI  1001#  8224
SETTRA  8458#  8467  8468  8469  8470  8472  8474  8475  8476  8477  8478  8479  8480
SETUP   1001#  2272
SKIP    1001#  2439  2444  2484  2489  2527  2532  2571  2576  2613  2628  2641  2653  2663  2668
        2702   2722  2734  2744  2749  2783  2796  2808  2818  2823  3089  3213  3712  4690  4724
        4729   4800  4834  4839  4910  4944  4949  5020  5054  5059  5130  5164  5169  5239  5273
```

```
                5278    5323    5328    5408    5442    5447    5525    5644    5764    5883    6002    6121    6240    6359    6489
                6646    6792    6910
SLASH           1001#
SPACE           1001#
SPREAD          1225#   2685    2766
STARS           1001#   1191    1205    1207    1214    1227    1276    1279    2402    2413    2447    2458    2492    2501    2535
                2545    2582    2593    2671    2682    2752    2763    2826    2839    3000    3012    3091    3104    3216    3226
                3337    3355    3518    3529    3638    3648    3715    3729    3878    3892    4041    4055    4204    4228    4393
                4403    4513    4526    4628    4646    4738    4756    4848    4866    4958    4976    5068    5086    5178    5195
                5287    5298    5345    5362    5458    5469    5528    5545    5647    5665    5767    5784    5886    5903    6005
                6022    6124    6141    6243    6260    6362    6386    6492    6511    6649    6657    6795    6807    6915    6925
                6992    7002    7069    7079    7150    7419    7496    7612    7623    7680    7733    7742    7827    7906    7983
                8050    8127    8142    8213    8237    8306    8359    8405    8414    8437
SWRSU           1001#   2295#
TRMTRP          8458#
TSTMEM          1225#   3732    3895    4058
TYPBIN          1001#
TYPDEC          1001#   7173    7180
TYPNAM          1001#   2317
TYPNUM          1001#
TYPOCS          1001#
TYPOCT          1001#   2344    3741    3904    4067    4237    4535    8155
TYPTXT          1001#   7169    7176
WRLOOP          1225#   5548    5668    5787    5906    6025    6144    6263
WRTRK           1225#   6514    6660
WRTRK1          1225#   6389    6810
$$CMRE          1225#
$$CMTM          1225#   1262    1263    1264    1265    1266    1267    1268    1269
$$ESCA          1001#
$$NEWT          1001#   2402    2447    2492    2535    2582    2671    2752    2826    3000    3091    3216    3337    3518    3638
                3715    3878    4041    4204    4393    4513    4628    4738    4848    4958    5068    5178    5287    5345    5458
                5528    5647    5767    5886    6005    6124    6243    6362    6492    6649    6795    6915    6992    7069
$$SET           8458#   8467    8468    8469    8470    8472    8474    8475    8476    8477    8478    8479    8480
$$SETM          2311#
$$SKIP          1001#   2439    2444    2484    2489    2527    2532    2571    2576    2613    2628    2641    2653    2663    2668
                2702    2722    2734    2744    2749    2783    2796    2808    2818    2823    3089    3213    3712    4690    4724
                4729    4800    4834    4839    4910    4944    4949    5020    5054    5059    5130    5164    5169    5239    5273
                5278    5323    5328    5408    5442    5447    5525    5644    5764    5883    6002    6121    6240    6359    6489
                6646    6792    6910
.EQUAT           867#    891
.HEADE           867#    869
.KT11            867#   1001
.SETUP           867#   2258
.SWRHI           867#    879
.SWRLO           867#    891#
.$ACTI           867#   1189
.$APTB          1277#
.$APTH           867#   1203
.$APTY           867#   7621
.$CATC           867#   1175
.$CMTA           867#   1225
.$EOP            867#   7148
.$ERRO           867#   7678
.$ERRT           867#
.$POWE           867#
.$RDOC           867#   8304
```

```
.$READ    867#   8048
.$SAVE    867#   8357
.$SCOP    867#   7494
.$SIZE    867#   7417
.$TRAP    867#   8435
.$TYPD    867#   7981
.$TYPE    867#   7825
.$TYPO    867#   7904


. ABS.  067410     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DSKZ:DZR6CA,DSKZ:DZR6CA.SEQ/CRF/SOL/DOC=DZR6CA
 RUN-TIME:  93  94  10 SECONDS
 RUN-TIME RATIO: 673/199=3.3
 CORE USED:  36K  (71 PAGES)

 DOCUMENT PAGES:  207
```