NCV11

NCV-11 DIAG
CZNCCCO

AH-E772C-MC
FICHE 1 OF 1

FEB 1981

MADE IN USA

d13181

IDENTIFICATION

PRODUCT CODE:	AC-E771C-MC
DIAGNOSTIC CODE:	MAINDEC-11-CZNCC-C
PRODUCT NAME:	CZNCCC NCV11 DIAGNOSTIC TEST
DATE:	AUG. 4, 1980
MAINTAINER:	DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

0.0 HISTORY

VERSION "C" WAS CREATED TO:
#1 INSTALL A 3 LOCATION PATCH TO LOGIC TEST 134.
#2 CORRECT A BUG IN LOGIC TEST 151.
VERSION "B" WAS CREATED TO:
#1 INSTALL A 2 LOCATION PATCH TO THE LOGIC TEST.
#2 INSTALL A 3 BYTE PATCH TO THE M8036 MICRO-CODE.
#3 CHANGE THE OUTPUT REPORT OF DIFFERENTIAL LINEARITY.
#4 CORRECT A DEFECT IN ACCOUNTING ENORMOUS ERRORS IN TOTAL
ERROR COUNTER.
#5 CORRECT ERROR CODES FOR TWO TESTS.

1.0 ABSTRACT

THE NCV11 DIAGNOSTIC PROGRAM IS A SERIES OF TESTS DESIGNED
ALL LOGIC FUNCTIONS AND DATA PATHS ACCESSIBLE ON THE
M8026 AND M8036. THE M7952 DIAGNOSTIC SHOULD HAVE BEEN EXECUTED.
TOTAL PROGRAM CONTROL IS ACCOMPLISHED THRU THE CONSOLE TERMINAL
VIA KEYBOARD TEST SELECTION AND THE PROVISIONS OF SECTION 5.
THE PROGRAM CAN BE EXECUTED ON AN LSI-11 OR PDP-11 COMPUTER.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP11 FAMILY OR LSI-11 FAMILY COMPUTER
2. I/O TERMINAL (IE: LA36)
3. NCV11 OPTION INCLUDING KWV11 REAL TIME CLOCK (M7952)
4. A017 SELF-TEST CONNECTOR (70-12894) (REQUIRED FOR SECTION 9.2 + 9.3)
5. H3060 JOYSTICK (OPTIONAL)
6. HARDWARE TESTER AND PROM BLASTER (OPTIONAL)

2.2 STORAGE

THE PROGRAM REQUIRES 16K OF MEMORY.
IF MEMORY MANAGEMENT OR ADDITIONAL MEMORY ARE SENSED, THE
PROGRAM WILL USE THOSE HARDWARE OPTIONS.

2.3 PRELIMINARY PROGRAMS

THE KWV11 DIAGNOSTIC (MD-11-CVKWA) SHOULD HAVE BEEN RUN.

3.0 LOADING PROCEDURE

NORMAL PROCEDURE FOR LOADING A BINARY PROGRAM INTO MEMORY
SHOULD BE FOLLOWED.

4.0 STARTING PROCEDURE

1. MAKE SURE THE NCV11 DEVICE BUS ADDRESS, INTERRUPT VECTOR ON THE
   M8026 AGREE WITH THE DEFAULT VALUES DEFINED IN SECTION 7.1.
2. MAKE SURE THE NCV11 CLOCK BUS ADDRESS ON THE M7952 AGREE
   WITH THE DEFAULT VALUE DEFINED IN SECTION 7.1.
3. MAKE SURE THE NCV11 INTERRUPT PRIORITY LEVEL ON THE M8217
   AGREE WITH THE DEFAULT VALUE DEFINED IN SECTION 7.1.
4. INSURE THAT THE HALT SWITCH IS DISABLED (IF ANY).

LSI11: 5. TYPE THE STARTING ADDRESS OF 200 AND THE CHARACTER G.
6. THE PROGRAM WILL RESPOND BY TYPING THE PROGRAM TITLE.

LSI11: 7. THE PROGRAM WILL REQUEST SWITCH REGISTER VALUE.
8. THE PROGRAM WILL TYPE THE TEST SELECTION MENU FOR THE OPERATOR.
9. THE OPERATOR SELECT'S THE APPROPRIATE TEST BY TYPING THE TEST LETTER AND "CR"

IF THE PROGRAM IS EXECUTED ON A CPU WITH A SWITCH REGISTER AND THE
OPERATOR SETS ALL SWITCHES TO A 1, THE SOFTWARE S.R. WILL BE USED.
IF THE HARDWARE SWITCH REGISTER IS USED, THE SOFTWARE S.R. HAS NO EFFECT.

4.1 PROGRAM START

SEQ 0004

200 STARTING ADDRESS OF THE PROGRAM
204 RESTART ADDRESS OF THE PROGRAM
210 STARTING ADDRESS FOR THE FACTORY OPTION CHECK-OUT AREA.
(INFORMS THE PROGRAM THE TESTER AND BLASTER ARE CONNECTED)

5.0 SOFTWARE SWITCH REGISTER

5.1 LOGIC TEST OPTIONS

SWITCH	OCTAL	FUNCTION
SW15=1	100000	HALT ON ERROR
SW14=1	040000	LOOP ON TEST
SW13=1	020000	INHIBIT ERROR TYPEOUTS
SW11=1	004000	INHIBIT ITERATIONS
SW10=1	002000	BELL ON ERROR
SW09=1	001000	LOOP ON ERROR
SW08=1	0004XX	LOOP ON TEST IN SWR <7-0>

5.2 FIELD ADJUSTMENT LOOP

SWITCH	OCTAL	FUNCTION
SW15=1	100000	HALT ON ''INPUT VOLTAGE OUT OF RANGE'' ERROR
SW13=1	020000	INHIBIT ERROR OR CONVERSION TYPEOUT
SW09=1	001000	INHIBIT CONVERSIONS ON CAMERA #3 GAIN = 1
SW08=1	000400	INHIBIT CONVERSIONS ON CAMERA #2 GAIN = 1
SW07=1	000200	INHIBIT CONVERSIONS ON CAMERA #1 GAIN = 1
SW06=1	000100	INHIBIT CONVERSIONS ON CAMERA #0 GAIN = 1
SW04=1	000020	INHIBIT CONVERSIONS ON JOYSTICK
SW03=1	000010	INHIBIT CONVERSIONS ON CAMERA #3 GAIN - 2
SW02=1	000004	INHIBIT CONVERSIONS ON CAMERA #2 GAIN = 2
SW01=1	000002	INHIBIT CONVERSIONS ON CAMERA #1 GAIN = 2
SW00-1	000001	INHIBIT CONVERSIONS ON CAMERA #0 GAIN = 2

5.3 CONTROL

1. THE SOFTWARE SWITCH REGISTER 'SWREG' (LOC. 176) CAN BE CHANGED BY USING THE ODT FACILITIES.
2. THE SOFTWARE SWITCH REGISTER CAN BE CHANGED UNDER PROGRAM CONTROL BY TYPING THE 'CONTROL & G' KEYS. THIS KEYBOARD OPERATION WILL PRINT THE CURRENT CONTENTS AND ACCEPT NEW OCTAL SWITCH REGISTER DATA TERMINATED WITH A CARRIAGE RETURN.

LSI-11:

3. ONCE THE ODT MODE HAS BEEN ENTERED BECAUSE OF AN ERROR CONDITION WITH BIT15 SET (HALT ON ERROR), STEP #2 ABOVE IS OF NO VALUE, SO RESORT TO STEP #1 TO ALTER THE SOFTWARE SWITCH REGISTER IF DESIRED BEFORE TYPING 'P' (CONTINUE).
4. IF THE PROGRAM IS PERFORMING RESET INSTRUCTIONS, SEVERAL 'CONTROL & C' OR 'CONTROL & G' COMMANDS MAY BE NECESSARY TO BE ACKNOWLEDGE BY THE PROGRAM.
5. TESTING CAN BE ABORTED BY TYPING THE 'CONTROL & C' KEYS.

6.0 ERROR REPORTING

ALL ERRORS ARE ACCOMPANIED WITH AN ENGLISH LANGUAGE DESCRIPTIVE
COMMENT AS TO THE TYPE OF FAILURE. FURTHER QUALIFICATION OF
THE ERROR CAN BE OBTAINED IF NEEDED FROM THE COMMENT AT THE
ERROR PC OR FROM THE TEST ITSELF.

7.0 MISCELLANEOUS

7.1 NCV11 BUS ADDRESS MODIFICATION

MODIFY LOCATION '$BASE' (LOC. 1250) IF BASE NCV11 BUS ADDRESS IS NOT 772760.
MODIFY THE LOW NINE BITS OF LOCATION '$VECT1' (LOC. 1244) IF
BASE NCV11 INTERRUPT VECTOR IS NOT 370.
MODIFY THE HIGH THREE BITS OF LOCATION '$VECT1' (LOC. 1244) IF
THE INTERRUPT PRIORITY LEVEL IS NOT LEVEL 6 ON A UNIBUS CPU.
MODIFY LOCATION '$CDW1' (LOC. 1254) IF BASE NCV11 CLOCK BUS ADDRESS IS NOT 770420.

7.2 XXDP/APT NOTES

THIS DIAGNOSTIC IS CHAINABLE UNDER XXDP (REQUIRES 16K OR MORE).
IF RUNNING UNDER XXDP/ACT/APT, THE LOGIC AND DIFLIN TESTS WILL BE RUN.

7.3 POWER FAIL

A POWER FAILURE WILL CAUSE A RESTART MESSAGE ON POWER UP AT
WHICH TIME THE PROGRAM IS RESTARTED (ONLY ON SYSTEMS WITH
NON-VOLATILE MEMORY AND WITH APPROPRIATE HARDWARE).

7.4 MULTIPLE NCV11'S INTERFACE TESTING

THIS PROGRAM DOES NOT "AUTO-SIZE" THE NUMBER OF NCV11'S CONNECTED.
FOR EACH ADDITIONAL NCV11, THE OPERATOR MUST MODIFY LOCATIONS
$BASE, $VECT1 AND $CDW1 (REF. 7.1).

7.5 RESTRICTIONS

KWV11 REAL TIME CLOCK (M7952) MUST NOT BE CONNECTED TO M8026 "FAST-ON" TABS.
UNLESS OTHERWISE STATED, NO EXTERNAL CONNECTIONS TO M8036 OR A017.

7.6 ADDRESS MAKER TABLE

THE M8036 CONTAINS LOGIC TO CONVERT A/D DATA INTO A RELATIVE ADDRESS. BELOW IS A TABLE WHICH SHOWS THE ADDRESS MAKER OUTPUT WITH THE "TEST CONTROL" MAINT. BIT SET. SETTING THIS BIT ENABLES THE STATES OF THE "GAIN, ZB ENABLE AND RESOLUTION" BITS SIMULATE THE CONVERTED A/D DATA.

RESOLUTION			ADDRESS MAKER OUTPUT BITS															
RES0	RES1	WORD*16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	B	G	Z	G	Z	G	1	1	1	G	Z	G	Z	G	1	1	1
1	1	0	0	B	G	Z	G	Z	G	1	1	G	Z	G	Z	G	1	1	0
1	0	1	0	0	B	G	Z	G	Z	G	1	0	G	Z	G	Z	G	1	0
1	0	0	0	0	0	B	G	Z	G	Z	G	1	G	Z	G	Z	G	1	0
0	1	1	0	0	0	0	B	G	Z	G	Z	G	0	G	Z	G	Z	G	0
0	1	0	0	0	0	0	0	B	G	Z	G	Z	G	G	Z	G	Z	G	0
0	0	1	0	0	0	0	0	0	B	G	Z	G	Z	G	G	Z	G	Z	G

0 = ALWAYS A 0
1 = ALWAYS A 1
G = GAIN FLOP STATE
Z = ZB ENABLE FLOP STATE
B = THE "AND" OF ZB ENABLE WITH ZB MAINT. INPUT

8.0 EXECUTION TIME

LOGIC: EXECUTION TIME RANGES FROM ABOUT 5 SECONDS WITH NO ITERATIONS TO ABOUT 70 SECONDS WITH ITERATIONS ENABLED. AN END PASS MESSAGE INDICATES ALL SUB-TESTS HAVE COMPLETED. EACH ADDITIONAL 4K OF MEMORY WILL ADD ABOUT 10 SECONDS TO THE LOGIC TEST RUNTIME.

DIFFERENTIAL LINEARITY: EXECUTION TIME IS ABOUT 90 SECONDS.

9.0 PROGRAM TEST DESCRIPTIONS

THE PROGRAM CONSISTS OF FOUR MAIN SECTIONS PLUS THREE AUXILIARY AND
THREE FACTORY OPTION CHECK-OUT SECTIONS.
THE OPERATOR SELECTS EACH SECTION VIA THE CONSOLE KEYBOARD.
BELOW IS A BRIEF DESCRIPTION OF EACH SECTION AND THE KEYBOARD CHARACTER.

MAIN SECTIONS

9.1 L LOGIC TEST (M8026 + M8036)

<NO INTERVENTION> <SETUP ONLY IF SA=210>

THE TEST CONTAINS A SERIES OF INDEPENDENT SUB-TESTS DESIGNED
TO TEST LOGIC FUNCTIONS AND DATA PATHS OF THE NCV11 GAMMA CAMERA INTERFACE.
A COMPLETE LIST OF TESTS IS AVAILABLE IN THE TABLE OF CONTENTS
AT THE BEGINNING OF THE LISTING. THE COMMENT FIELD WITHIN
EACH SUB-TEST TITLE CAN BE BENEFICIAL TO UNDERSTANDING THE LOGIC TESTED.
ADDITIONAL LOGIC TESTS OF THE A017 WILL BE EXECUTED IF THE
PROGRAM WAS STARTED AT LOC. 210.

9.2 D (M) DIFFERENTIAL LINEARITY (A017)

<REQUIRES SELF-TEST CONNECTOR IN J1 ON THE A017> <SETUP INTERVENTION ONLY>
<A017 SWITCH SET TO "MAINT." POSITION>

A FINITE VARIATION OF INPUT VOLTAGE EQUALS A GIVEN CONVERTED
VALUE OR STATE. THE TEST IS TO VERIFY THE WIDTH OF EACH POSSIBLE
STATE. AN INPUT VOLTAGE IS RAMPED AND THE NCV11 SAMPLES THE RESULT.
THE NCV11 INCREMENTS THE RESPECTIVE MEMORY LOCATION CORRESPONDING
TO THE INPUT VALUE. THE PROGRAM THEN DETERMINES IF ANY STATE HAS
BEEN SKIPPED, IS EXCESSIVELY WIDE OR NARROW. WHEN "M" IS USED TO
SELECT THE SECTION, THE PROGRAM OUTPUT AN EXPANDED REPORT OF
THE 254. STATES. TIGHTER ERROR TOLERANCES ARE USED IF RUNNING
IN THE OPTION CHECKOUT AREA (SA 210).

9.3 F FINAL ACCEPTANCE

<REQUIRES SELF-TEST CONNECTOR IN J1 ON THE A017> <SETUP INTERVENTION ONLY>
<A017 SWITCH SET TO "MAINT." POSITION>

THE SUB-SECTION RUNS SECTION "L" AND "D" TO VERIFY PROPER OPERATION.

9.4 A ADJUSTMENT OF A017 AT FIELD SITE

<MANUAL INTERVENTION> <SETUP INTERVENTION>
<A017 SWITCH SET TO "OPER." POSITION>

COARSE ADJUSTMENT OT THE A017 CAN BE PERFORMED USING THIS SUB-SECTION.
THE OPERATOR MAY SELECT THE CAMERA AND GAIN TO BE SAMPLED USING THE
SWITCH REGISTER. A SERIES OF CONVERSIONS ARE TAKEN ON EACH CAMERA
AND THE OPERATOR IS INFORMED OF THE RESULTS.

SWITCH	OCTAL	FUNCTION
SW15=1	100000	HALT ON "INPUT VOLTAGE OUT OF RANGE" ERROR
SW13=1	020000	INHIBIT ERROR OR CONVERTED VALUE TYPE-OUT
SW09=1	001000	INHIBIT CONVERSIONS ON CAMERA #3 GAIN = 1
SW08=1	000400	INHIBIT CONVERSIONS ON CAMERA #2 GAIN = 1
SW07=1	000200	INHIBIT CONVERSIONS ON CAMERA #1 GAIN = 1
SW06=1	000100	INHIBIT CONVERSIONS ON CAMERA #0 GAIN = 1
SW04=1	000020	INHIBIT CONVERSIONS ON JOYSTICK
SW03=1	000010	INHIBIT CONVERSIONS ON CAMERA #3 GAIN = 2
SW02=1	000004	INHIBIT CONVERSIONS ON CAMERA #2 GAIN = 2
SW01=1	000002	INHIBIT CONVERSIONS ON CAMERA #1 GAIN = 2
SW00=1	000001	INHIBIT CONVERSIONS ON CAMERA #0 GAIN = 2

AUXILIARY SECTION

9.5 O OTHER TERMINAL ADDRESS

IN SOME CUSTOMER SITES, THE CONSOLE TERMINAL MAY NOT BE LOCATED
WITHIN EASY ACCESS TO THE NOV11. THIS SECTION ALLOWS THE OPERATOR
TO CHANGE THE PROGRAMS CONTROL TO ANOTHER TERMINAL.

9.6 H HELP THE OPERATOR

A LIST OF THE COMMANDS WILL BE TYPED TO REFRFSH THE OPERATOR'S MEMORY.

9.7 S SOFTWARE SWITCH REGISTER

ENABLES THE OPERATOR TO CHANGE THE SOFTWARE SWITCH REGISTER WHEN
NOT RUNNING ANOTHER SUB-SECTION.

FACTORY OPTION CHECK-OUT SECTION

9.8 I INITIAL ADJUSTMENT OF A017

<MANUAL INTERVENTION> <SETUP INTERVENTION> <HARDWARE TESTER>
<A017 SWITCH SET TO "OPER." POSITION>

A KNOWN GOOD DIGITAL TO ANALOG (D/A) SUPPLIES A REFERENCE INPUT
VOLTAGE. THE OPERATOR IS INFORMED WHICH ADJUSTMENT TO PERFORM.
THE PURPOSE OF THE TEST IS TO VERIFY THAT THE ADJUSTMENT TO THE PRE-AMP
SECTION CAN BE PREFORMED. THE TEST DOES NOT ATTEMPT TO CALIBRATE
THE PRE-AMP, BUT ONLY TO FUNCTIONALLY VERIFY THE ADJUSTMENTS.

9.9 B BLASTING THE LINEARITY CORRECTION PROM

<MANUAL INTERVENTION> <SETUP INTERVENTION> <HARDWARE TESTER>
<A017 SWITCH SET TO "MAINT." POSITION>

THE TEST IS DESIGNED TO CREATE THE LINEARITY CORRECTION PROM (LCP). THE TESTER HARDWARE CONTAINS RAM STORAGE TO EMULATE THE LCP. THE PROGRAM WILL RUN DIFFERENTIAL LINEARITY. THE PROGRAM WILL THEN USE THE RESULTS TO DETERMINE THE CORRECT VALUE TO BE PLACED IN THE LCP. THE PROGRAM WILL THEN INFORM THE PROM BLASTER TO GENERATE A CORRECTED LINEARITY PROM FOR THE A017 MODULE. UPON COMPLETION OF GENERATION AND VERIFICATION STAGE, THE OPERATOR IS INSTRUCTED TO REMOVE THE EMULATOR CABLE FROM THE A017 AND INSERT THE NEWLY CREATED LCP. THE PROGRAM WILL RE-EXECUTE DIFFERENTIAL LINEARITY.

9.10 C CONTROL PROGRAM PROM

<MANUAL INTERVENTION> <SETUP INTERVENTION> <HARDWARE TESTER>

THE ROUTINE WILL CREATE THE M8036 CONTROL PROM. THE OPERATOR MAY CREATE SEVERAL COPIES BY REPEATELY EXECUTING THIS SECTION. THE PROGRAM WILL INFORM THE OPERATOR THE STEPS TO PERFORM.

LOCATION	OCTAL	BINARY
00	215	10001101
01	244	10100100
02	116	01001110
03	144	01100100
04	215	10001101
05	244	10100100
06	116	01001110
07	144	01100100
10	215	10001101
11	244	10100100
12	116	01001110
13	144	01100100
14	215	10001101
15	244	10100100
16	116	01001110
17	144	01100100
20	106 (116)	01000110 (01001110)
21	144	01100100
22	314 (216)	11001100 (10001110)
23	304 (244)	11000100 (10100100)

NOTE: () INDICATES OLD CONTENTS OF M8036 ROM.

10.0 LISTING

```
                                ;DEVELOPED USING SYSMAC.C3
                                .LIST   ME
                                .NLIST  MD,MC,CND
                                        .ENABL  ABS,AMA
          167400                        $SWR=167400
          000001                        $TN=1
                                .TITLE  CZNCCC  NCV11   DIAGNOSTIC
                                ;*COPYRIGHT (C) 1980
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY R. SHOOP
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
                                .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
          000006                SP=     %6              ;;STACK POINTER
          000007                PC=     %7              ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
          000000                PR0=    0               ;;PRIORITY LEVEL 0
          000040                PR1=    40              ;;PRIORITY LEVEL 1
          000100                PR2=    100             ;;PRIORITY LEVEL 2
          000140                PR3=    140             ;;PRIORITY LEVEL 3
          000200                PR4=    200             ;;PRIORITY LEVEL 4
          000240                PR5=    240             ;;PRIORITY LEVEL 5
          000300                PR6=    300             ;;PRIORITY LEVEL 6
          000340                PR7     340             ;;PRIORITY LEVEL 7
```

```
(1)
(1)                                 ;*"SWITCH REGISTER" SWITCH DEFINITIONS
(1)             100000              SW15=   100000
(1)             040000              SW14=   40000
(1)             020000              SW13=   20000
(1)             010000              SW12=   10000
(1)             004000              SW11=   4000
(1)             002000              SW10=   2000
(1)             001000              SW09=   1000
(1)             000400              SW08=   400
(1)             000200              SW07=   200
(1)             000100              SW06=   100
(1)             000040              SW05=   40
(1)             000020              SW04=   20
(1)             000010              SW03=   10
(1)             000004              SW02=   4
(1)             000002              SW01=   2
(1)             000001              SW00=   1
(1)                                 .EQUIV  SW09,SW9
(1)                                 .EQUIV  SW08,SW8
(1)                                 .EQUIV  SW07,SW7
(1)                                 .EQUIV  SW06,SW6
(1)                                 .EQUIV  SW05,SW5
(1)                                 .EQUIV  SW04,SW4
(1)                                 .EQUIV  SW03,SW3
(1)                                 .EQUIV  SW02,SW2
(1)                                 .EQUIV  SW01,SW1
(1)                                 .EQUIV  SW00,SW0
(1)
(1)                                 ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)             100000              BIT15=  100000
(1)             040000              BIT14=  40000
(1)             020000              BIT13=  20000
(1)             010000              BIT12=  10000
(1)             004000              BIT11=  4000
(1)             002000              BIT10=  2000
(1)             001000              BIT09=  1000
(1)             000400              BIT08=  400
(1)             000200              BIT07=  200
(1)             000100              BIT06=  100
(1)             000040              BIT05=  40
(1)             000020              BIT04=  20
(1)             000010              BIT03=  10
(1)             000004              BIT02=  4
(1)             000002              BIT01=  2
(1)             000001              BIT00=  1
(1)                                 .EQUIV  BIT09,BIT9
(1)                                 .EQUIV  BIT08,BIT8
(1)                                 .EQUIV  BIT07,BIT7
(1)                                 .EQUIV  BIT06,BIT6
(1)                                 .EQUIV  BIT05,BIT5
(1)                                 .EQUIV  BIT04,BIT4
(1)                                 .EQUIV  BIT03,BIT3
(1)                                 .EQUIV  BIT02,BIT2
(1)                                 .EQUIV  BIT01,BIT1
(1)                                 .EQUIV  BIT00,BIT0
```

```
(1)
(1)                             ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)          000004             ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
(1)          000010             RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)          000014             TBITVEC=14              ;;"T" BIT
(1)          000014             TRTVEC= 14              ;;TRACE TRAP
(1)          000014             BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
(1)          000020             IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)          000024             PWRVEC= 24              ;;POWER FAIL
(1)          000030             EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
(1)          000034             TRAPVEC=34              ;;"TRAP" TRAP
(1)          000060             TKVEC=  60              ;;TTY KEYBOARD VECTOR
(1)          000064             TPVEC=  64              ;;TTY PRINTER VECTOR
(1)          000240             PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
 14                              .SBTTL  MEMORY MANAGEMENT DEFINITIONS
(1)
(1)                             ;*KT11 VECTOR ADDRESS
(1)
(1)          000250             MMVEC=  250
(1)
(1)                             ;*KT11 STATUS REGISTER ADDRESSES
(1)
(1)          177572             SR0=    177572
(1)          177574             SR1=    177574
(1)          177576             SR2=    177576
(1)          172516             SR3=    172516
(1)
(1)                             ;*USER "I" PAGE DESCRIPTOR REGISTERS
(1)
(1)          177600             UIPDR0= 177600
(1)          177602             UIPDR1= 177602
(1)          177604             UIPDR2= 177604
(1)          177606             UIPDR3= 177606
(1)          177610             UIPDR4= 177610
(1)          177612             UIPDR5= 177612
(1)          177614             UIPDR6= 177614
(1)          177616             UIPDR7= 177616
(1)
(1)                             ;*USER "D" PAGE DESCRIPTOR REGISTORS
(1)
(1)          177620             UDPDR0= 177620
(1)          177622             UDPDR1= 177622
(1)          177624             UDPDR2= 177624
(1)          177626             UDPDR3= 177626
(1)          177630             UDPDR4= 177630
(1)          177632             UDPDR5= 177632
(1)          177634             UDPDR6= 177634
(1)          177636             UDPDR7= 177636
(1)
(1)                             ;*USER "I" PAGE ADDRESS REGISTERS
(1)
(1)          177640             UIPAR0= 177640
(1)          177642             UIPAR1= 177642
(1)          177644             UIPAR2= 177644
(1)          177646             UIPAR3= 177646
(1)          177650             UIPAR4= 177650
```

```
(1)          177652                  UIPAR5= 177652
(1)          177654                  UIPAR6= 177654
(1)          177656                  UIPAR7= 177656
(1)
(1)                                  ;*USER "D" PAGE ADDRESS REGISTERS
(1)
(1)          177660                  UDPAR0= 177660
(1)          177662                  UDPAR1= 177662
(1)          177664                  UDPAR2= 177664
(1)          177666                  UDPAR3= 177666
(1)          177670                  UDPAR4= 177670
(1)          177672                  UDPAR5= 177672
(1)          177674                  UDPAR6= 177674
(1)          177676                  UDPAR7= 177676
(1)
(1)                                  ;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
(1)
(1)          172200                  SIPDR0= 172200
(1)          172202                  SIPDR1= 172202
(1)          172204                  SIPDR2= 172204
(1)          172206                  SIPDR3= 172206
(1)          172210                  SIPDR4= 172210
(1)          172212                  SIPDR5= 172212
(1)          172214                  SIPDR6= 172214
(1)          172216                  SIPDR7= 172216
(1)
(1)                                  ;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
(1)
(1)          172220                  SDPDR0= 172220
(1)          172222                  SDPDR1= 172222
(1)          172224                  SDPDR2= 172224
(1)          172226                  SDPDR3= 172226
(1)          172230                  SDPDR4= 172230
(1)          172232                  SDPDR5= 172232
(1)          172234                  SDPDR6= 172234
(1)          172236                  SDPDR7= 172236
(1)
(1)                                  ;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
(1)
(1)          172240                  SIPAR0= 172240
(1)          172242                  SIPAR1= 172242
(1)          172244                  SIPAR2= 172244
(1)          172246                  SIPAR3= 172246
(1)          172250                  SIPAR4= 172250
(1)          172252                  SIPAR5= 172252
(1)          172254                  SIPAR6= 172254
(1)          172256                  SIPAR7= 172256
(1)
(1)                                  ;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
(1)
(1)          172260                  SDPAR0= 172260
(1)          172262                  SDPAR1= 172262
(1)          172264                  SDPAR2= 172264
(1)          172266                  SDPAR3= 172266
(1)          172270                  SDPAR4= 172270
(1)          172272                  SDPAR5= 172272
```

```
(1)          172274                  SDPAR6= 172274
(1)          172276                  SDPAR7= 172276
(1)
(1)                                  ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
(1)
(1)          172300                  KIPDR0= 172300
(1)          172302                  KIPDR1= 172302
(1)          172304                  KIPDR2= 172304
(1)          172306                  KIPDR3= 172306
(1)          172310                  KIPDR4= 172310
(1)          172312                  KIPDR5= 172312
(1)          172314                  KIPDR6= 172314
(1)          172316                  KIPDR7= 172316
(1)
(1)                                  ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
(1)
(1)          172320                  KDPDR0= 172320
(1)          172322                  KDPDR1= 172322
(1)          172324                  KDPDR2= 172324
(1)          172326                  KDPDR3= 172326
(1)          172330                  KDPDR4= 172330
(1)          172332                  KDPDR5= 172332
(1)          172334                  KDPDR6= 172334
(1)          172336                  KDPDR7= 172336
(1)
(1)                                  ;*KERNEL "I" PAGE ADDRESS REGISTERS
(1)
(1)          172340                  KIPAR0= 172340
(1)          172342                  KIPAR1= 172342
(1)          172344                  KIPAR2= 172344
(1)          172346                  KIPAR3= 172346
(1)          172350                  KIPAR4= 172350
(1)          172352                  KIPAR5= 172352
(1)          172354                  KIPAR6= 172354
(1)          172356                  KIPAR7= 172356
(1)
(1)                                  ;*KERNEL "D" PAGE ADDRESS REGISTERS
(1)
(1)          172360                  KDPAR0= 172360
(1)          172362                  KDPAR1= 172362
(1)          172364                  KDPAR2= 172364
(1)          172366                  KDPAR3= 172366
(1)          172370                  KDPAR4= 172370
(1)          172372                  KDPAR5= 172372
(1)          172374                  KDPAR6= 172374
(1)          172376                  KDPAR7= 172376
(1)
 15
 16          172760                          ABASE=172760            ;NCV11 BASE ADDRESS
 17          140370                          AVECT1=140370           ;BR LEVEL 6 VECTOR TO 370
 18          170420                          ACDW1=170420            ;NCV11 CLOCK ADDRESS
```

```
                                        .SBTTL  OPERATIONAL SWITCH SETTINGS
(1)                                     ;*
(1)                                     ;*      SWITCH                  USE
(1)                                     ;*      ------          --------------------
(1)                                     ;*        15            HALT ON ERROR
(1)                                     ;*        14            LOOP ON TEST
(1)                                     ;*        13            INHIBIT ERROR TYPEOUTS
(1)                                     ;*        11            INHIBIT ITERATIONS
(1)                                     ;*        10            BELL ON ERROR
(1)                                     ;*         9            LOOP ON ERROR
(1)                                     ;*         8            LOOP ON TEST IN SWR<7:0>
                                        .SBTTL  TRAP CATCHER
(1)
(1)             000000                          .=0
(1)                                     ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1)                                     ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)                                     ;*LOCATION 0 CONTAIN 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)             000174                          .=174
(1)   000174    000000          DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
(1)   000176    000000          SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
(1)                                     .SBTTL  STARTING ADDRESS(ES)
(1)   000200    000137  002034          JMP     @#BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
 23   000204    000137  002014          JMP     RTRT            ;RESTART ADDRESS
 24   000210    000137  002022          JMP     TESTER          ;STARTING ADDRESS WHEN IN THE OPTION AREA
 25
 26             000100                          .=100
 27   000100    000104  000200  000002          104,200,2                       ;B EVENT SAFE GUARD
```

```
                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
                                        $SVPC=.                 ;SAVE PC
                                        .=46
                                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                                        .=52
                                        .WORD   0               ;;2)SET LOC.52 TO ZERO
                                        .=$SVPC                 ;; RESTORE PC
                                        .=1000
                                .SBTTL  APT PARAMETER BLOCK

                                ;;******************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;******************************************************************
                                        .$X=.   ;;SAVE CURRENT LOCATION
                                        .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
                                        200     ;;FOR APT START UP
                                        .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                                        $APTHDR ;;POINT TO APT HEADER BLOCK
                                        .=.$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  001174                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000030                  $TSTM:  .WORD   30      ;;RUN TIM OF LONGEST TEST
001006  000010                  $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000030                  $UNITM: .WORD   30      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000031                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
     32                                 .SBTTL  COMMON TAGS
    (1)
    (2)                                 ;;**********************************************************************
    (1)                                 ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
    (1)                                 ;*USED IN THE PROGRAM.
    (1)
    (1)           001100                        .=1100
    (1)  001100                         $CMTAG:                         ;;START OF COMMON TAGS
    (1)  001100  000000                         .WORD   0
    (1)  001102     000                 $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
    (1)  001103     000                 $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
    (1)  001104  000000                 $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
    (1)  001106  000000                 $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
    (1)  001110  000000                 $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
    (1)  001112  000000                 $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
    (1)  001114     000                 $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
    (1)  001115     001                 $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
    (1)  001116  000000                 $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
    (1)  001120  000000                 $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
    (1)  001122  000000                 $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
    (1)  001124  000000                 $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
    (1)  001126  000000                 $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
    (1)  001130  000000                         .WORD   0               ;;RESERVED--NOT TO BE USED
    (1)  001132  000000                         .WORD   0
    (1)  001134     000                 $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
    (1)  001135     000                 $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
    (1)  001136  000000                         .WORD   0
    (1)  001140  177570                 SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
    (1)  001142  177570                 DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
    (1)  001144  177560                 $TKS:   177560                  ;;TTY KBD STATUS
    (1)  001146  177562                 $TKB:   177562                  ;;TTY KBD BUFFER
    (1)  001150  177564                 $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
    (1)  001152  177566                 $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
    (1)  001154     000                 $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
    (1)  001155     002                 $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
    (1)  001156     012                 $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
    (1)  001157     000                 $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>-0-YES)
    (1)  001160  000000                 $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
    (1)  001162  000000                 $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
    (1)  001164  177607  000377         $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
    (1)  001170     077                 $QUES:  .ASCII  /?/             ;;QUESTION MARK
    (1)  001171     015                 $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
    (1)  001172  000012                 $LF:    .ASCIZ  <12>            ;;LINE FEED
    (2)                                 ;;**********************************************************************
    (2)                                 .SBTTL  APT MAILBOX-ETABLE
    (2)
    (3)                                 ;;**********************************************************************
    (2)                                 .EVEN
    (2)  001174                         $MAIL:                          ;;APT MAILBOX
    (2)  001174  000000                 $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
    (2)  001176  000000                 $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
    (2)  001200  000000                 $TESTN: .WORD   ATESTN          ;;TEST NUMBER
    (2)  001202  000000                 $PASS:  .WORD   APASS           ;;PASS COUNT
    (2)  001204  000000                 $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
    (2)  001206  000000                 $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
    (2)  001210  000000                 $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
```

```
(2)  001212  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
(2)  001214                      $ETABLE:                  ;;APT ENVIRONMENT TABLE
(2)  001214     000              $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
(2)  001215     000              $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
(2)  001216  000000              $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
(2)  001220  000000              $USWR:  .WORD   AUSWR   ;;USER SWITCHES
(2)  001222  000000              $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
(2)                              ;*                      BITS 15-11=CPU TYPE
(2)                              ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2)                              ;*                              11/70=06,PDQ=07,Q=10
(2)                              ;*                      BIT 10=REAL TIME CLOCK
(2)                              ;*                      BIT  9=FLOATING POINT PROCESSOR
(2)                              ;*                      BIT  8=MEMORY MANAGEMENT
(2)  001224     000              $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(2)  001225     000              $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
(2)                              ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
(2)                              ;*                              900 NSEC CORE=001
(2)                              ;*                              300 NSEC BIPOLAR=002
(2)                              ;*                              500 NSEC MOS=003
(2)  001226  000000              $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
(2)                              ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(2)  001230     000              $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(2)  001231     000              $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
(2)  001232  000000              $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(2)  001234     000              $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(2)  001235     000              $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
(2)  001236  000000              $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(2)  001240     000              $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(2)  001241     000              $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
(2)  001242  000000              $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
(2)  001244  140370              $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(2)  001246  000000              $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(2)  001250  172760              $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)  001252  000000              $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
(2)  001254  170420              $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
(2)  001256                      $ETEND:
(2)                              .MEXIT
```

```
    (1)                                 .SBTTL  ERROR POINTER TABLE
    (1)
    (1)                                 ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
    (1)                                 ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
    (1)                                 ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
    (1)                                 ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
    (1)                                 ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
    (1)
    (1)                                 ;*      EM              ;;POINTS TO THE ERROR MESSAGE
    (1)                                 ;*      DH              ;;POINTS TO THE DATA HEADER
    (1)                                 ;*      DT              ;;POINTS TO THE DATA
    (1)                                 ;*      DF              ;;POINTS TO THE DATA FORMAT
    (1)
    (1)
    (1)  001256                         $ERRTB:
     33  001256  030224  032504  032734         EM1,DH2,DT2,DF0         ;M8026  NCV11 TIMEOUT
         001264  033006
     34  001266  030264  032504  032734         EM2,DH2,DT2,DF0         ;M8026  COMMAND-STATUS REGISTER ERROR
         001274  033006
     35  001276  030330  032504  032734         EM3 DH2,DT2,DF0         ;M8026  SPECIAL FUNCTION REGISTER ERROR
         001304  033006
     36  001306  030376  032504  032734         EM4,DH2,DT2,DF0         ;M8026  WORD COUNT REGISTER ERROR
         001314  033006
     37  001316  030436  032504  032734         EM5,DH2,DT2,DF0         ;M8026  BUS ADDRESS REGISTER ERROR
         001324  033006
     38  001326  030477  032504  032734         EM6,DH2,DT2,DF0         ;M8026  OFFSET REGISTER ERROR
         001334  033006
     39  001336  030533  032504  032734         EM7,DH2,DT2,DF0         ;M8026  DUAL REGISTER SELECTION ERROR
         001344  033006
     40  001346  030577  032504  032734         EM10,DH2,DT2,DF0        ;M8026-M8036    LOW 16 BIT Z COUNT ERROR
         001354  033006
     41  001356  030644  032504  032734         EM11,DH2,DT2,DF0        ;M8026-M8036    HIGH 16 BIT Z COUNT ERROR
         001364  033006
     42  001366  030712  032504  032734         EM12,DH2,DT2,DF0        ;M8026  Z COUNT STATUS ERROR
         001374  033006
     43  001376  030745  032504  032734         EM13,DH2,DT2,DF0        ;M8026  Z COUNT INTERRUPT ERROR
         001404  033006
     44  001406  031003  032504  032734         EM14,DH2,DT2,DF0        ;M8036  JOYSTICK STATUS ERROR
         001414  033006
     45  001416  031037  032504  032734         EM15,DH2,DT2,DF0        ;M8036  JOYSTICK DATA ERROR
         001424  033006
     46  001426  031071  032504  032734         EM16,DH2,DT2,DF0        ;M8036  DATA INCREMENT ERROR
         001434  033006
     47  001436  031124  032504  032734         EM17,DH2,DT2,DF0        ;M8036  DATA DECREMENT ERROR
         001444  033006
     48  001446  031157  032504  032734         EM20,DH2,DT2,DF0        ;M8026-M8036    MATRIX MODE ADDRESS MAKER DATA ERROR
         001454  033006
     49  001456  031240  032504  032734         EM21,DH2,DT2,DF0        ;M8026  LIST MODF ADDRESS MAKER DATA ERROR
         001464  033006
     50  001466  031311  032504  032734         EM22,DH2,DT2,DF0        ;M8026  LIST MODE TRANSFER BUS ADDRESS ERROR
         001474  033006
     51  001476  031364  032504  032734         EM23,DH2,DT2,DF0        ;M8026  LIST MODE TRANSFER WORD COUNT ERROR
         001504  033006
     52  001506  031436  032504  032734         EM24,DH2,DT2,DF0        ;M8026  LIST MODE TRANSFER OFFSET ERROR
         001514  033006
     53  001516  031504  032504  032734         EM25,DH2,DT2,DF0        ;M8026  TIMEOUT STATUS ERROR
```

```
                001524  033006
     54         001526  031537  032504  032734          EM26,DH2,DT2,DF0        ;M8026  TIMEOUT INTERRUPT ERROR
                001534  033006
     55         001536  031575  032504  032734          EM27,DH2,DT2,DF0        ;M8026  SET ''EVENT'' OR ''TIME'' DATA ERROR
                001544  033006
     56         001546  031644  032504  032734          EM30,DH2,DT2,DF0        ;M8026  CELL INCREMENT DATA ERROR
                001554  033006
     57         001556  031704  032504  032734          EM31,DH2,DT2,DF0        ;M8026  CELL OVERFLOW STATUS ERROR
                001564  033006
     58         001566  031745  032504  032734          EM32,DH2,DT2,DF0        ;M8026  CELL OVERFLOW INTERRUPT ERROR
                001574  033006
     59         001576  032011  032530  032746          EM33,DH3,DT3,DF0        ;M8026  MATRIX MODE ADDRESS MUX ERROR
                001604  033006
     60         001606  032055  032504  032734          EM34,DH2,DT2,DF0        ;M8026  ''TESTX'' FUNCTION ERROR
                001614  033006
     61         001616  032112  032563  032762          EM35,DH4,DT4,DF0        ;M8026  DATA ERROR WHEN TRANSFERING TO EXTENDED MEMORY
                001624  033006
     62         001626  032177  032504  032734          EM36,DH2,DT2,DF0        ;M8026  LIST MODE TRANSFER STATUS ERROR
                001634  033006
     63         001636  032245  032530  032746          EM37,DH3,DT3,DF0        ;M8026  LIST MODE TRANSFER DATA ERROR
                001644  033006
     64         001646  032311  032504  032734          EM40,DH2,DT2,DF0        ;JUMPER-M8026-M7952     ''EVENT'' OR ''TIME'' MARK ERROR
                001654  033006
     65         001656  032371  032471  033000          EM41,DH1,DT5,DF0        ;M7952  CLOCK TIMEOUT
                001664  033006
     66         001666  032431  032471  032726          EM42,DH1,DT1,DF0        ;M8217  INTERRUPT LEVEL ERROR
                001674  033006
     67                                          ;DL11 BLASTER COMM. ADDRESSES (THE OPERATOR MUST CHANGE IF DIFFERENT)
     68         001676  176510                  DLICSR: 176510          ;INPUT STATUS REG.
     69         001700  176512                  DLIBD:  176512          ;INPUT DATA
     70         001702  176514                  DLOCSR: 176514          ;OUTPUT STATUS
     71         001704  176516                  DLODB:  176516          ;OUTPUT DATA
     72
     73                                          ;VSV01 ADDRESSES (THE OPERATOR MUST CHANGE IF DIFFERENT)
     74         001706  172600                  VTCHAR: 172600          ;CHAR. STATUS
     75         001710  172602                  VTYPOS: 172602
     76         001712  172603                  VTXPOS: 172603
     77         001714  172604                  VTCXY:  172604
     78         001716  172620                  VTCSR:  172620          ;MAP STATUS
     79         001720  172622                  VTMAP:  172622
     80         001722  172624                  VTPX:   172624
     81         001724  172630                  VTINT:  172630
     82
     83                                          ;TESTER ADDRESSES (THE OPERATOR MUST CHANGE IF DIFFERENT)
     84         001726  176416                  DACSR:  176416          ;KNOWN GOOD D/A STATUS
     85         001730  176420                  DAC0:   176420
     86         001732  176422                  DAC1:   176422
     87         001734  176424                  DAC2:   176424
     88         001736  176426                  DAC3:   176426
```

```
                                ;KWV11 PROGRAM GENERATED ADDRESSES (THE OPER. ONLY HAS TO CHANGE '$CDW1')

001740  170420                  KWCSR:  ACDW1           ;KWV11 STATUS REGISTER
001742  170421                  KWCSR1: ACDW1+1         ;KWV11 STATUS REG. HIGH BYTE
001744  170422                  KWPSR:  ACDW1+2         ;KWV11 PRESET REGISTER

                                ;NCV11 PROGRAM GENERATER ADDRESSES (THE OPER. ONLY HAS TO CHANGE '$BASE')

001746  172760                  CSR:    ABASE
001750  172762                  OFF:    ABASE+2
001752  172764                  WCR:    ABASE+4
001754  172766                  BAR:    ABASE+6
001756  172770                  SFR:    ABASE+10
001760  172772                  ADM:    ABASE+12
001762  172774                  JOY:    ABASE+14
001764  172776                  BAR1:   ABASE+16
001766  172761                  CSRHB:  ABASE+1
001770  172771                  SFRHB:  ABASE+11

                                ;NCV11 PROGRAM GENERATED VECTORS (THE OPER. ONLY HAS TO CHANGE LOW 9 BITS OF '$VECT1')

001772  140370                  VECTA0: AVECT1
001774  140372                  VECTA1: AVECT1+2
001776  140374                  VECTB0: AVECT1+4
002000  140376                  VECTB1: AVECT1+6

                                ;NCV11 PROGRAM GENERATED BR LEVEL (THE OPER. ONLY HAS TO CHANGE TOP 3 BITS OF '$VECT1')

002002  000300                  BRLEV:  300             ;BR LEVEL 6

002004  000000                  $TEMP:  0
002006  000001                  CPUDL0: 1
002010  000020                  CPUDL1: 20
002012  000004                  CPUDL2: 4

        040000                          CLRWCO=BIT14
        004000                          CLRALL=BIT11
        000400                          ENDDMA=BIT8
        000010                          TSTDMA=BIT3
        000004                          TSTCON=BIT2
        000002                          TESTZ=BIT1
        000001                          REDJOY=BIT0
        000000                          MORTST=0        ;DEFINE TO RESTORE TEST CODE
```

```
  140  002014  005237  050034          RTRT:   INC     AGAN            ;SET RESTART FLAG
  141  002020  000411                          BR      RTRTA
  142  002022  005237  050026          TESTER: INC     WFMODE          ;SET FLAG INDICATING OPTICN AREA MODE
  143  002026  005037  050034                  CLR     AGAN
  144  002032  000404                          BR      RTRTA
  145  002034  005037  050034          BEGIN:  CLR     AGAN
  146  002040  005037  050026                  CLR     WFMODE          ;CLEAR FLAG INDIC. OPTION TEST AREA MODE
  147  002044                          RTRTA:
  (1)                                 .SBTTL  INITIALIZE THE COMMON TAGS
  (1)                                 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
  (1)  002044  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
  (1)  002050  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
  (1)  002052  022706  001140                  CMP     #SWR,R6 ;;DONE?
  (1)  002056  001374                          BNE     .-6             ;;LOOP BACK IF NO
  (1)  002060  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
  (1)                                 ;;INITIALIZE A FEW VECTORS
  (1)  002064  012737  034112  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
  (1)  002072  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
  (1)  002100  012737  035106  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
  (1)  002106  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
  (1)  002114  012737  037272  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
  (1)  002122  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
  (1)  002130  012737  035710  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
  (1)  002136  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
  (1)  002144  013737  030136  030130          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
  (1)  002152  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
  (1)  002156  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
  (1)  002162  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
  (1)  002170  012737  002170  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
  (1)  002176  012737  002176  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
  (2)                                 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
  (2)                                 ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
  (2)  002204  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
  (2)  002210  012737  002244  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
  (2)  002216  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
  (2)  002224  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
  (2)  002232  022777  177777  176700          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
  (2)  002240  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
  (2)                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
  (2)  002242  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
  (2)  002244  012716  002252          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
  (2)  002250  000002                          RTI
  (2)  002252  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
  (2)  002260  012737  000174  001142          MOV     #DISPREG,DISPLAY
  (2)  002266  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
  (1)
  (2)  002272  005037  001202                  CLR     $PASS           ;;CLEAR PASS COUNT
  (2)  002276  132737  000200  001215          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
  (2)  002304  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
  (2)  002306  012737  001216  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
  (2)  002314                          67$:
```

```
149                                     .SBTTL  TYPE PROGRAM NAME
(1)                                     ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
(1)  002314  005227  177777             INC     #-1             ;;FIRST TIME?
(1)  002320  001044                     BNE     68$             ;;BRANCH IF NO
(1)  002322  022737  030170  000042     CMP     #$ENDAD,@#42    ;;ACT-11?
(1)  002330  001440                     BEQ     68$             ;;BRANCH IF YES
(1)  002332  104401  002400             TYPE    ,69$            ;;TYPE ASCIZ STRING
(2)                                     .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
(2)  002336  005737  000042             TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
(2)  002342  001012                     BNE     70$             ;;BRANCH IF YES
(2)  002344  123727  001214  000001     CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
(2)  002352  001406                     BEQ     70$             ;;BRANCH IF YES
(2)  002354  023727  001140  000176     CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
(2)  002362  001005                     BNE     71$             ;;BRANCH IF NO
(2)  002364  104406                     GTSWR                   ;;GET SOFT-SWR SETTINGS
(2)  002366  000403                     BR      71$
(2)  002370  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
(2)  002376                          71$:
(1)  002376  000415                     BR      68$             ;;GET OVER THE ASCIZ
(1)                                  ;;69$:  .ASCIZ  <CRLF>#CZNCCC   NCV11 DIAGNOSTIC#<CRLF>
(1)  002432                          68$:
150                                     .SBTTL  PRIME THE NCV11 ADDRESSES FROM THE DEFAULT VALUES
151  002432  112737  000103  052412     MOVB    #'C,RUNIT       ;LOAD "CONTROL" SECTION BUS TRAP FLAG
152  002440  012737  003212  000004     MOV     #BUSTRP,@#ERRVEC        ;LOAD BUS TRAP RETURN ADDRESS
153  002446  012700  001746             MOV     #CSR,R0         ;GET ADDRESS POINTER
154  002452  013701  001250             MOV     $BASE,R1        ;GET DEFAULT ADDRESS
155  002456  010120                  1$:    MOV     R1,(R0)+        ;LOAD AN VALUE
156  002460  062701  000002             ADD     #2,R1           ;BUMP THE VALUE
157  002464  022700  001766             CMP     #CSRHB,R0       ;TEST IF DONE
158  002470  001372                     BNE     1$              ;BR IF NOT
159  002472  013710  001250             MOV     $BASE,(R0)      ;LOAD ODD BYTE
160  002476  005220                     INC     (R0)+           ;ADDRESSES
161  002500  013710  001250             MOV     $BASE,(R0)
162  002504  062720  000011             ADD     #11,(R0)+       ;
163  002510  013701  001244             MOV     $VECT1,R1       ;GET DEFAULT VECTOR
164  002514  010102                     MOV     R1,R2           ;COPY R1
165  002516  000302                     SWAB    R2              ;EXCHANGE BYTES
166  002520  042702  177437             BIC     #177437,R2      ;STRIP OFF ALL BUT BR LEVEL BITS
167  002524  010237  002002             MOV     R2,BRLEV        ;SAVE FOR USE LATER
168  002530  042701  160000             BIC     #160000,R1      ;CLEAR OFF BR LEVEL
169  002534  010120                  2$:    MOV     R1,(R0)+        ;LOAD VECTOR
170  002536  005721                     TST     (R1)+           ;BUMP THE VALUE
171  002540  022700  002002             CMP     #VECTB1+2,R0    ;TEST IF DONE
172  002544  001373                     BNE     2$
173  002546  013737  001254  001740     MOV     $CDW1,KWCSR     ;GET CLOCK ADDRESS
174  002554  013737  001740  001742     MOV     KWCSR,KWCSR1    ;SET CLOCK HIGH BYTE ADDR.
175  002562  013737  001740  001744     MOV     KWCSR,KWPSR     ;SET CLOCK PRESET ADDRESS
176  002570  005237  001742             INC     KWCSR1
177  002574  062737  000002  001744     ADD     #2,KWPSR
```

```
                                                .SBTTL  DIAGNOSTIC IDENTIFICATION TYPEOUT
002602  005737  050034                        TST     AGAN            ;TEST IF RESTART
002606  001017                                BNE     RBEGO           ;BR IF YES
002610  105737  001134                        TSTB    $AUTOB          ;TEST IF ACT/APT
002614  001405                                BEQ     3$              ;BR IF NOT
002616  012737  177777  050024                MOV     #-1,RUNDIF      ;INDICATE TO RUN DIFLIN IN CHAIN MODE
002624  000137  003360                        JMP     LOGIC           ;YES; RUN LOGIC AND DIFLIN TESTS
002630  005737  050026          3$:           TST     WFMODE          ;TEST IF OPTION CHECKOUT MODE
002634  001402                                BEQ     4$              ;BR IF YES
002636  104401                                TYPE
002640  046174                                LISTO                   ;TELL OPERATOR ABOUT OPTION AREA TESTS
002642  104401                  4$:           TYPE
002644  046404                                LIST                    ;TELL OPERATOR ABOUT THE TEST
                                        .SBTTL  KEYBOARD COMMAND DECODER
002646  104401  047017          RBEGO:        TYPE,   LIST1           ;PRINT "DOT"
002652  104411                  1$:           RDLIN                   ;READ THE RESPONSE
002654  013637  052412                        MOV     @(SP)+,RUNIT    ;GET THE CHARACTER
002660  142737  000040  052412                BICB    #40,RUNIT       ;ENSURE UPPER CASE
002666  112737  000042  052413                MOVB    #'",RUNIT+1     ;FIX ASCII MESSAGE
002674  122737  000101  052412                CMPB    #'A,RUNIT       ;DETERMINE IF AN A
002702  001002                                BNE     2$              ;BR IF NOT
002704  000137  003300                        JMP     TFSITE          ;RUN SITE ADJUSTMENT LOOP
002710  005737  050026          2$:           TST     WFMODE          ;TEST IF ON THE TESTER
002714  001406                                BEQ     3$              ;BR IF NOT
002716  122737  000102  052412                CMPB    #'B,RUNIT       ;DETERMINE IF AN B
002724  001002                                BNE     3$              ;BR IF NOT
002726  000137  042274                        JMP     BLAST           ;RUN THE BLASTER SECTION
002732  122737  000104  052412  3$:           CMPB    #'D,RUNIT       ;DETERMINE IF AN D
002740  001004                                BNE     4$              ;BR IF NOT
002742  005037  050024                        CLR     RUNDIF          ;INDICATE DIFLIN IS NOT TO BE RUN
002746  000137  043550                        JMP     DIFLIN          ;AND RUN IT
002752  122737  000106  052412  4$:           CMPB    #'F,RUNIT       ;DETERMINE IF AN F
002760  001005                                BNE     5$              ;BR IF NOT
002762  012737  177777  050024                MOV     #-1,RUNDIF      ;INDICATE RUN DIFLIN AFTER THE LOGIC TEST
002770  000137  003360                        JMP     LOGIC           ;AND RUN LOGIC TEST
002774  005737  050026          5$:           TST     WFMODE          ;TEST IF OPTION CHECKOUT MODE
003000  001406                                BEQ     6$              ;BR IF NOT
003002  122737  000111  052412                CMPB    #'I,RUNIT       ;DETERMIN IF AN I
003010  001002                                BNE     6$              ;BR IF NOT
003012  000137  047024                        JMP     POTIME          ;RUN IN-HOUSE ADJUSTMENT LOOP
003016  122737  000114  052412  6$:           CMPB    #'L,RUNIT       ;DETERMINE IF AN L
003024  001004                                BNE     7$              ;BR IF NOT
003026  005037  050024                        CLR     RUNDIF          ;INDICATE NO DIFLIN
003032  000137  003360                        JMP     LOGIC
003036  122737  000110  052412  7$:           CMPB    #'H,RUNIT       ;DETERMINE IF AN H
003044  001002                                BNE     10$             ;BR IF NOT
003046  000137  002034                        JMP     BEGIN           ;START OVER
003052  122737  000037  052412  10$:          CMPB    #37,RUNIT       ;DETERMINE IF AN "?"
003060  001002                                BNE     11$             ;BR IF NOT
003062  000137  002034                        JMP     BEGIN           ;START OVER
003066  005737  050026          11$:          TST     WFMODE          ;TEST IF OPTION CHECKOUT MODE
003072  001406                                BEQ     12$             ;BR IF NOT
003074  122737  000124  052412                CMPB    #'T,RUNIT       ;DETERMINE IF AN T
003102  001002                                BNE     12$             ;BR IF NOT
003104  000137  042022                        JMP     BTALK           ;KEYBOARD LOOP WITH BLASTER
003110  005737  050026          12$:          TST     WFMODE          ;TEST IF OPTION CHECKOUT MODE
```

```
   235  003114  001406                          BEQ     13$             ;BR IF NOT
   236  003116  122737  000103  052412          CMPB    #'C,RUNIT       ;DETERMINE IF AN C
   237  003124  001002                          BNE     13$             ;BR IF NOT
   238  003126  000137  042756                  JMP     PBLAST          ;RUN M8036 PROM BLASTING CODE
   239  003132  122737  000117  052412  13$:    CMPB    #'O,RUNIT       ;DETERMINE IF AN O
   240  003140  001002                          BNE     14$             ;BR IF NOT
   241  003142  000137  003242                  JMP     XTTY            ;CHANGE THE TTY ADDRESS
   242  003146  122737  000123  052412  14$:    CMPB    #'S,RUNIT       ;DETERMINE IF AN S
   243  003154  001002                          BNE     15$             ;BR IF NOT
   244  003156  104406                          GTSWR                   ;GET SWITCH REGISTER VALUE
   245  003160  000632                          BR      RBEGO           ;RESTART
   246  003162  122737  000115  052412  15$:    CMPB    #'M,RUNIT       ;DETERMINE IF AN M
   247  003170  001005                          BNE     FATFNG          ;BR IF NOT
   248  003172  012737  000377  050024          MOV     #377,RUNDIF     ;INDICATE DIFLIN WITH EXPAND REPORT
   249  003200  000137  043550        .         JMP     DIFLIN          ;RUN DIF LIN
```

```
                                        ;THE OPERATOR SELECTED THE WRONG THING - THEY SOMETIME HAVE "FAT FINGERS"
003204  104401                  FATFNG: TYPE
003206  001170                          $QUES                   ;TYPE QUESTION MARK
003210  000616                          BR      RBEGO
                                        ;RETURN TO HERE UPON UNEXPECTED BUS TRAP
003212  104401  052342          BUSTRP: TYPE,   FATALO          ;REPORT FATAL TRAP TO THE OPERATOR
003216  011646                          MOV     (SP),-(SP)      ;GET TRAP ADDRESS
003220  104402                          TYPOC
003222  005777  175712          1$:     TST     @SWR            ;TEST IF HALT ON ERROR
003226  100001                          BPL     2$              ;BR IF SET
003230  000000                          HALT                    ;FATAL BUS TRAP DETECTED
003232  104401  052432          2$:     TYPE,   RUNITA
003236  000137  002014                  JMP     RTRT            ;RESTART
                                        ;ROUTINE TO INPUT NEW TTY ADDRESS FROM THE OPERATOR
003242  104401  052522          XTTY:   TYPE,   WARNO           ;TELL THE OPERATOR THE RULES
003246  104412                          RDOCT                   ;LISTEN TO THE OPERATOR
003250  012600                          MOV     (SP)+,R0        ;READ THEIR INPUT
003252  001754                          BEQ     FATFNG          ;BR IF JUST "CR"
003254  005710                          TST     (R0)            ;ADDRESS THE DEVICE AND CHECK BUS TRAP
                                        ;IF NO BUS TRAP - THEY MUST KNOW WHAT THEY ARE DOING .
003256  012701  001144                  MOV     #$TKS,R1        ;GET POINTER
003262  010021                  1$:     MOV     R0,(R1)+        ;LOAD THE VALUE
003264  005720                          TST     (R0)+           ;BUMP VALUE
003266  020127  001154                  CMP     R1,#$TKS+10     ;TEST IF DONE ALL ADDRESSES
003272  001373                          BNE     1$              ;BR IF NOT
003274  000137  002044                  JMP     RTRTA           ;RETYPE THE HEADER AGAIN ON THE NEW TERMINAL

                                ;OPERATOR CHOSE THE FIELD ADJUSTMENT LOOP
003300  104401  051166          TFSITE: TYPE,   PRIM6           ;TELL OPER. ABOUT A017
003304  104401  041344                  TYPE,   FIELDI          ;ASK THE OPERATOR FOR INPUT Z MODE
003310  005037  040360                  CLR     INOUTZ          ;DEFAULT TO USER SUPPLIED Z PULSES
003314  104411                          RDLIN                   ;READ OPER. INPUT
003316  013637  003352                  MOV     @(SP)+,10$      ;GET CHAR
003322  142737  000040  003352          BICB    #40,10$         ;ENSURE UPPER CASE
003330  122737  000131  003352          CMPB    #'Y,10$         ;TEST FOR 'Y' INPUT
003336  001003                          BNE     1$              ;BR IF NOT 'Y'
003340  012737  000002  040360          MOV     #2,INOUTZ       ;SET MAINT Z CONSTANT
003346  000137  037360          1$:     JMP     FSITE           ;NOW DO THE LOOP
003352  000000                  10$:    0

003354  000000                  ADNOKT: 0                       ;LAST ADDRESS WITHOUT M.M. OR XXDP
003356  000000                  NLSI11: 0                       ;NON-ZERO INDICATES LSI-11 PROCESSOR
```

```
297
298  003360  000005                    LOGIC:  RESET
299  003362  005737  001202                    TST     $PASS           ;TEST IF FIRST PASS
300  003366  001124                            BNE     TST1            ;;BR IF NOT
301  003370  005737  050026                    TST     WFMODE          ;CHECK IF ON TESTER
302  003374  001402                            BEQ     5$              ;BR IF NOT
303  003376  104401  052250                    TYPE,   PRIM4           ;TELL OPERATOR CABLE MUST BE CONNECTED
304  003402  005037  036124            5$:     CLR     $KT11           ;INDICATE NO KT11
305  003406  004737  036066                    JSR     PC,$SIZE        ;SIZE BASIC MEMORY SIZE
306  003412  162737  001000  036404            SUB     #1000,$LSTAD    ;PROTECT ABSLDR
307  003420  005737  000042                    TST     @#42            ;TEST IF XXDP CHAIN MODE ?
308  003424  001403                            BEQ     1$              ;BR IF NOT CHAIN MODE
309  003426  162737  006200  036404            SUB     #3200.,$LSTAD   ;LEAVE ROOM FOR XXDP MONITOR
310  003434  013737  036404  003354    1$:     MOV     $LSTAD,ADNOKT   ;SAVE BASIC MEMORY STACE
311  003442  012737  000200  036124            MOV     #BIT7,$KT11     ;WITH M.M.
312  003450  004737  036066                    JSR     PC,$SIZE        ;DETERMINE THE AMOUNT OF MEMORY
313  003454  005737  036124                    TST     $KT11           ;TEST IF KT11 IS HERE
314  003460  100025                            BPL     2$              ;BR IF NOT
315  003462  012737  000200  172342            MOV     #200,@#KIPAR1   ;
316  003470  012737  077406  172302            MOV     #77406,@#KIPDR1
317  003476  012737  000400  172344            MOV     #400,@#KIPAR2
318  003504  012737  077406  172304            MOV     #77406,@#KIPDR2
319  003512  012737  077406  172306            MOV     #77406,@#KIPDR3 ;4K, UP R/W FOR 60000
320  003520  012737  007600  172356            MOV     #7600,@#KIPAR7  ;I/O PAGE MAP
321  003526  012737  077406  172316            MOV     #77406,@#KIPDR7 ;4K, UP, R/W FOR I/O PAGE ACCESS
322  003534  105737  001134            2$:     TSTB    $AUTOB          ;TEST IF AUTO MODE
323  003540  001016                            BNE     4$              ;;BR IF AUTO MODE
324  003542  104401  052463                    TYPE,   MSGMEM          ;TELL OPERATOR THE AMOUNT OF MEMORY
325  003546  013700  036406                    MOV     $LSTBK,R0       ;GET LAST GOOD BANK
326  003552  006200                            ASR     R0
327  003554  006200                            ASR     R0
328  003556  006200                            ASR     R0
329  003560  006200                            ASR     R0
330  003562  006200                            ASR     R0
331  003564  005200                            INC     R0              ;CONVERT
332  003566  010046                            MOV     R0,-(SP)        ;TYPE OUT DEC. VALUE
333  003570  104405                            TYPDS
334  003572  104401  052503                    TYPE,   MSGK            ;AND THEN TYPE OUT THE REMAINDER OF MEMORY MESSAGE
335  003576  013737  000004  002004    4$:     MOV     @#ERRVEC,$TEMP  ;SAVE LOC 4 VALUE
336  003604  012737  003632  000004            MOV     #3$,@#ERRVEC    ;LOAD LOC 4
337  003612  005037  003356                    CLR     NLSI11          ;INDICATE NOT A LSI-11 CPU
338  003616  005037  177776                    CLR     PSW             ;SHOULD FAIL IF LSI-11
339  003622  013737  002004  000004            MOV     $TEMP,@#ERRVEC  ;RESTORE LOC 4 VALUE
340  003630  000403                            BR      TST1            ;,BR IF TEST
341  003632  005237  003356            3$:     INC     NLSI11                  ;SET AN LSI-11 FLAG
342  003636  000002                            RTI                             ;RETURN
```

```
                                ;;*****************************************************************
                                ;*TEST 1        VERIFY THE 8 NCV11 AND 2 NCV11 CLOCK BUS ADDRESSES RESPOND
                                ;;*****************************************************************
003640  000004                  TST1:   SCOPE
003642  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
003650  013737  000004  004016          MOV     @#ERRVEC,10$            ;SAVE BUS TRAP VECTOR
003656  012737  003754  000004          MOV     #1$,@#ERRVEC            ;LOAD BUS TRAP VECTOR
003664  005777  176056                  TST     @CSR                    ;ADDRESS
003670  005777  176054                  TST     @OFF                    ;
003674  005777  176052                  TST     @WCR                    ;THE
003700  005777  176050                  TST     @BAR                    ;
003704  005777  176046                  TST     @SFR                    ;NCV11
003710  005777  176044                  TST     @ADM                    ;
003714  005777  176044                  TST     @BAR1                   ;ADDRESSES
003720  012737  003762  000004          MOV     #2$,@#ERRVEC            ;LOAD NEW RETURN
003726  005777  176006                  TST     @KWCSR                  ;TEST CLOCK
003732  005777  176006                  TST     @KWPSR                  ;ADDRESSES
                                ;TEMP FIX THE DIAG TO NOT USE CLOCK JUMPERS
003736  012737  000001  004020          MOV     #1,DEADKW       ;INDICATE NCV11 CLOCK IS NOT THERE
003744  013737  004016  000004          MOV     10$,@#ERRVEC            ;RESTORE THE BUS TRAP VECTOR
003752  000423                          BR      TST2            ;;BR AND TEST THE NCV11
003754  022626                  1$:     CMP     (SP)+,(SP)+             ;CLEAN THE STACK
003756  104001                          ERROR   1                       ;BUS TRAP WHEN REFFRENCING THE NCV11
003760  000411                          BR      3$
003762  022626                  2$:     CMP     (SP)+,(SP)+             ;CLEAN THE STACK
003764  104041                          ERROR   41                      ;BUS TRAP WHEN REFERENCING THE NCV11 CLOCK
003766  012737  000001  004020          MOV     #1,DEADKW               ;INDICATE NCV11 CLOCK IS NOT THERE
003774  013737  004016  000004          MOV     10$,@#ERRVEC            ;RESTORE LOC 4
004002  000407                          BR      TST2            ;;
004004  013737  004016  000004  3$:     MOV     10$,@#ERRVEC            ;RESTORE BUS TRAP VECTOR
004012  000137  030102                  JMP     $EOP                    ;EXIT
004016  000000                  10$:    0
004020  000000                  DEADKW: 0                               ;NON-ZERO SAYS NO NCV11 CLOCK
```

```
                                        ;;*******************************************************************
                                        ;*TEST 2         FLOAT A 1 ACROSS 10 BITS OF THE COMMAND/STATUS REG.
                                        ;;*******************************************************************
004022  000004                  TST2:   SCOPE
004024  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004032  012737  004046  001110          MOV     #1$,$LPERR      ;LOAD LOOP ADDRESS
004040  012737  004000  002004          MOV     #BIT11,$TEMP            ;LOAD INITIAL REG. VALUE

004046  013777  002004  175672  1$:     MOV     $TEMP,@CSR              ;LOAD CSR REG.
004054  017737  175666  001126          MOV     @CSR,$BDDAT             ;READ CSR
004062  013737  002004  001124          MOV     $TEMP,$GDDAT            ;LOAD EXPECTED
004070  052737  000200  001124          BIS     #BIT7,$GDDAT            ;FUDGE THE "READY" BIT
004076  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE THE VALUES
004104  001401                        BEQ     2$              ;;BR IF SAME
004106  104002                        ERROR   2                       ;UNEXPECTED VALUE IN THE CSR REGISTER

004110  006237  002004          2$:     ASR     $TEMP                   ;TRY THE NEXT DATA BIT
004114  022737  000001  002004          CMP     #1,$TEMP                ;TEST IF NOW BIT 0
004122  001351                        BNE     1$                      ;BR IF NOT

                                        ;;*******************************************************************
                                        ;*TEST 3         VERIFY THAT "INIT" CLEARS THE CSR REGISTER
                                        ;;*******************************************************************
004124  000004                  TST3:   SCOPE
004126  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004134  012777  007776  175604          MOV     #7776,@CSR              ;LOAD CSR REG.
004142  000005                        RESET                           ;INITILIZE THE REGISTER
004144  012737  000200  001124          MOV     #BIT7,$GDDAT            ;LOAD EXPECTED VALUE
004152  017737  175570  001126          MOV     @CSR,$BDDAT             ;READ CSR REG.
004160  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE THE VALUES
004166  001401                        BEQ     TST4            ;;BR IF EQUAL
004170  104002                        ERROR   2                       ;"BUS INIT" FAILED TO CLEAR CSR REG.

                                        ;;*******************************************************************
                                        ;*TEST 4         VERIFY THAT "CLEAR ALL" CLEARS THE CSR REGISTER
                                        ;;*******************************************************************
004172  000004                  TST4:   SCOPE
004174  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004202  012777  007776  175536          MOV     #7776,@CSR              ;LOAD CSR REG.
004210  012777  004000  175540          MOV     #CLRALL,@SFR            ;GENERATE "CLR ALL L"
004216  012737  000200  001124          MOV     #BIT7,$GDDAT            ;LOAD EXPECTED VALUE
004224  017737  175516  001126          MOV     @CSR,$BDDAT             ;READ THE CSR REG.
004232  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
004240  001401                        BEQ     TST5            ;;BR IF SAME
004242  104002                        ERROR   2                       ;"CLR ALL L" FAILED TO CLEAR CSR REG.
```

```
                                        ;;*****************************************************************
                                        ;*TEST 5         VERIFY LOW BYTE OPERATION OF THE "CSR" REGISTER
                                        ;;*****************************************************************
004244  000004                  TST5:   SCOPE
004246  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004254  012777  003636  175464          MOV     #3636,@CSR              ;LOAD CSR REGISTER
004262  012737  003600  001124          MOV     #3600,$GDDAT            ;LOAD EXPECTED VALUE
004270  105077  175452                  CLRB    @CSR                    ;CLEAR LOW BYTE
004274  017737  175446  001126          MOV     @CSR,$BDDAT             ;READ STATUS REG.
004302  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
004310  001401                          BEQ     1$              ;;BR IF SAME
004312  104002                          ERROR   2                       ;CLEARING LOW BYTE OF THE CSR CHANGED THE HIGH B
004314  012777  003636  175424  1$:     MOV     #3636,@CSR              ;LOAD CSR REGISTER
004322  012737  000236  001124          MOV     #236,$GDDAT             ;LOAD EXPECTED VALUE
004330  105077  175432                  CLRB    @CSRHB                  ;CLEAR HIGH BYTE OF THE CSR
004334  017737  175406  001126          MOV     @CSR,$BDDAT             ;READ STATUS
004342  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
004350  001401                          BEQ     TST6            ;;BR IS SAME
004352  104002                          ERROR   2                       ;CLEARING HIGH BYTE OF CSR CHANGED THE LOW BYTE
                                        ;;*****************************************************************
                                        ;*TEST 6         FLOAT A 1 ACROSS 4 BITS OF THE SPECIAL FUNCTION REGISTER
                                        ;;*****************************************************************
004354  000004                  TST6:   SCOPE
004356  012737  004372  001110          MOV     #1$,$LPERR              ;LOAD LOOP ADDRESS
004364  012737  000020  002004          MOV     #BIT4,$TEMP             ;LOAD INITIAL REG. VALUE

004372  013777  002004  175356  1$:     MOV     $TEMP,@SFR              ;LOAD SFR REG.
004400  017737  175352  001126          MOV     @SFR,$BDDAT             ;READ SFR
004406  013737  002004  001124          MOV     $TEMP,$GDDAT            ;LOAD EXPECTED
004414  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE THE VALUES
004422  001401                          BEQ     2$              ;;BR IF SAME
004424  104003                          ERROR   3                       ;UNEXPECTED VALUE IN THE SFR REGISTER
004426  006237  002004          2$:     ASR     $TEMP                   ;TRY THE NEXT DATA BIT
004432  022737  000001  002004          CMP     #1,$TEMP                ;TEST IF NOW BIT 0
004440  001354                          BNE     1$                      ;BR IF NOT

                                        ;;*****************************************************************
                                        ;*TEST 7         VERIFY THAT CLEARING HIGH BYTE OF SFR DOES NOT CLEAR LOW BYTE
                                        ;;*****************************************************************
004442  000004                  TST7:   SCOPE
004444  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004452  012777  000014  175276          MOV     #14,@SFR                ;LOAD THE S.F. REGISTER
004460  012737  000014  001124          MOV     #14,$GDDAT              ;LOAD THE EXPECTED VALUE
004466  105077  175276                  CLRB    @SFRHB                  ;CLEAR HIGH BYTE OF S.F. REG.
004472  017737  175260  001126          MOV     @SFR,$BDDAT             ;READ THE REGISTER
004500  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE THE VALUES
004506  001401                          BEQ     TST10           ;;BR IF SAME
004510  104003                          ERROR   3                       ;CLEARING HIGH BYTE OF CSR REG. CHANGED THE LOW
```

```
                                        ;;*****************************************************************
                                        ;*TEST 10        VERIFY THAT "INIT" CLEARS THE SFR REGISTER
                                        ;;*****************************************************************
004512  000004                  TST10:  SCOPE
004514  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004522  012777  000016  175226          MOV     #16,@SFR                ;LOAD SFR REG.
004530  000005                          RESET                           ;INITILIZE THE REGISTER
004532  005037  001124                  CLR     $GDDAT                  ;LOAD EXPECTED
004536  017737  175214  001126          MOV     @SFR,$BDDAT             ;READ SFR REG.
004544  001401                          BEQ     TST11           ;;BR IF EQUAL
004546  104003                          ERROR   3                       ;"BUS INIT" FAILED TO CLEAR SFR REG.

                                        ;;*****************************************************************
                                        ;*TEST 11        VERIFY THAT "CLEAR ALL" CLEARS THE SFR REGISTER
                                        ;;*****************************************************************
004550  000004                  TST11:  SCOPE
004552  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004560  012777  000016  175170          MOV     #16,@SFR                ;LOAD SFR REG.
004566  052777  004000  175162          BIS     #CLRALL,@SFR            ;GENERATE "CLR ALL L"
004574  005037  001124                  CLR     $GDDAT                  ;LOAD EXPECTED VALUE
004600  017737  175152  001126          MOV     @SFR,$BDDAT             ;READ THE SFR REG.
004606  001401                          BEQ     TST12           ;;BR IF SAME
004610  104003                          ERROR   3                       ;"CLR ALL L" FAILED TO CLEAR SFR REG.
```

```
                                ;;*****************************************************************
                                ;*TEST 12        FLOAT A 1 ACROSS THE WORD COUNT REGISTER
                                ;;*****************************************************************
004612  000004                TST12:  SCOPE
004614  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004622  012737  000001  001124          MOV     #BIT0,$GDDAT            ;LOAD EXPECTED VALUE
004630  012737  004636  001110          MOV     #1$,$LPERR              ;LOAD LOOP ADDRESS ON ERROR

004636  012777  004000  175112  1$:     MOV     #CLRALL,@SFR            ;RESET THE DEVICE
004644  013777  001124  175100          MOV     $GDDAT,@WCR             ;LOAD WORD COUNT "A"
004652  017737  175074  001126          MOV     @WCR,$BDDAT             ;READ WORD COUNT
004660  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
004666  001401                        BEQ     2$              ;;BR IF SAME
004670  104004                        ERROR   4                       ;;WORD COUNT REG. IN ERROR

004672  006337  001124          2$:     ASL     $GDDAT                  ;CHANGE THE DATA
004676  001357                        BNE     1$                      ;BR IF MORE DATA TO LOAD
                                ;;*****************************************************************
                                ;*TEST 13        FLOAT A 1 ACROSS THE BUS ADDRESS REGISTER
                                ;;*****************************************************************
004700  000004                TST13:  SCOPE
004702  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004710  012737  000001  001124          MOV     #BIT0,$GDDAT            ;LOAD EXPECTED VALUE
004716  012737  004724  001110          MOV     #1$,$LPERR              ;LOAD LOOP ADDRESS ON ERROR

004724  012777  004000  175024  1$:     MOV     #CLRALL,@SFR            ;RESET THE DEVICE
004732  013777  001124  175014          MOV     $GDDAT,@BAR             ;LOAD BUS ADDRESS "A"
004740  017737  175010  001126          MOV     @BAR,$BDDAT             ;READ BUS ADDRESS
004746  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
004754  001401                        BEQ     2$              ;;BR IF SAME
004756  104005                        ERROR   5                       ;BUS ADDRESS REG. IN ERROR

004760  006337  001124          2$:     ASL     $GDDAT                  ;CHANGE THE DATA
004764  001357                        BNE     1$                      ;BR IF MORE DATA TO LOAD
                                ;;*****************************************************************
                                ;*TEST 14        FLOAT A 1 ACROSS THE OFFSET REGISTER
                                ;;*****************************************************************
004766  000004                TST14:  SCOPE
004770  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
004776  012737  000001  001124          MOV     #BIT0,$GDDAT            ;LOAD EXPECTED VALUE
005004  012737  005012  001110          MOV     #1$,$LPERR              ;LOAD LOOP ADDRESS ON ERROR

005012  012777  004000  174736  1$:     MOV     #CLRALL,@SFR            ;RESET THE DEVICE
005020  013777  001124  174722          MOV     $GDDAT,@OFF             ;LOAD OFFSET "A"
005026  017737  174716  001126          MOV     @OFF,$BDDAT             ;READ OFFSET
005034  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
005042  001401                        BEQ     2$              ;;BR IF SAME
005044  104006                        ERROR   6                       ;OFFSET REG. IN ERROR

005046  006337  001124          2$:     ASL     $GDDAT                  ;CHANGE THE DATA
005052  001357                        BNE     1$                      ;BR IF MORE DATA TO LOAD
```

```
                        ;;*****************************************************************
                        ;*TEST 15       VERIFY NO DUAL REGISTER SELECTION
                        ;;*****************************************************************
005054  000004          TST15:  SCOPE
005056  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
                        ;LOAD DIFFERENT NUMBERS INTO THE  WCR, BAR AND OFF REGISTERS
005064  012777  004000  174664          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
005072  012777  011111  174650          MOV     #11111,@OFF             ;LOAD OFFSET REGISTER
005100  012777  022222  174644          MOV     #22222,@WCR             ;LOAD W.C. REGISTER
005106  012777  033333  174640          MOV     #33333,@BAR             ;LOAD B.A. REGISTER
005114  012777  000036  174634          MOV     #36,@SFR                ;LOAD SPECIAL FUNCTION REGISTER
005122  012777  007636  174616          MOV     #7636,@CSR              ;LOAD COMMAND/STATUS REGISTER
                        ;NOW READ EACH REGISTER AND CHECK THE VALUE
005130  012737  011111  001124          MOV     #11111,$GDDAT           ;LOAD EXPECTED VALUE
005136  017737  174606  001126          MOV     @OFF,$BDDAT             ;READ A OFFSET REG.
005144  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
005152  001401                        BEQ     1$              ;;BR IF SAME
005154  104007                        ERROR   7               ;DUAL ADDRESS ERROR
005156  012737  022222  001124  1$:     MOV     #22222,$GDDAT           ;LOAD EXPECTED VALUE
005164  017737  174562  001126          MOV     @WCR,$BDDAT             ;READ W.C. REG.
005172  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
005200  001401                        BEQ     2$              ;;BR IF SAME
005202  104007                        ERROR   7               ;DUAL ADDRESS ERROR
005204  012737  033333  001124  2$:     MOV     #33333,$GDDAT           ;LOAD EXPECTED VLAUE
005212  017737  174536  001126          MOV     @BAR,$BDDAT             ;READ B.A. REG.
005220  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
005226  001401                        BEQ     3$              ;;BR IF SAME
005230  104007                        ERROR   7               ;DUAL ADDRESS ERROR
005232  012737  000036  001124  3$:     MOV     #36,$GDDAT              ;LOAD EXPECTED VALUE
005240  017737  174512  001126          MOV     @SFR,$BDDAT             ;READ SPECIAL FUNCTION REG
005246  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
005254  001401                        BEQ     4$              ;;BR IF SAME
005256  104007                        ERROR   7               ;DUAL ADDRESS ERROR
005260  012737  007636  001124  4$:     MOV     #7636,$GDDAT            ;LOAD EXPECTED VALUE
005266  017737  174454  001126          MOV     @CSR,$BDDAT             ;READ COMMAND/STATUS REGISTER
005274  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
005302  001401                        BEQ     5$              ;;BR IF SAME
005304  104007                        ERROR   7               ;DUAL ADDRESS ERROR
005306  012777  004000  174442  5$:     MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
```

```
                        ;;*****************************************************************
                        ;*TEST 16       VERIFY "CLR ALL" CLEARS THE EXTENDED OFFSET BITS
                        ;;*****************************************************************
005314  000004          TST16:  SCOPE
005316  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005324  005037  001124                  CLR     $GDDAT                  ;CLEAR EXPECTED VALUE
005330  012777  004000  174420          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
005336  012777  000003  174404          MOV     #3,@OFF                 ;LOAD EXTENDED OFFSET REGISTER
005344  012777  004000  174404          MOV     #CLRALL,@SFR            ;CLEAR THE EXTENDED OFFSET REG.'S
005352  017737  174372  001126          MOV     @OFF,$BDDAT             ;READ EXTENDED OFFSET REG.
005360  001401                          BEQ     TST17           ;;BR IF CLEARED
005362  104006                          ERROR   6                       ;CLEAR ALL FAILED TO CLEAR EXTENDED OFFSET REGIS
                        ;;*****************************************************************
                        ;*TEST 17       TEST THE "ACTIVE" FLOP CAN SET AND CLEAR
                        ;;*****************************************************************
005364  000004          TST17:  SCOPE
005366  012777  004000  174362          MOV     #CLRALL,@SFR                    ;CLEAR THE DEVICE
005374  052777  000010  174354          BIS     #TSTDMA,@SFR    ;SET THE "TEST DMA" TO PREVNT DATA TRANSFERS
005402  000240                          NOP
005404  000240                          NOP
005406  000240                          NOP
005410  052777  000022  174330          BIS     #BIT4!BIT1,@CSR         ;SET "MATRIX MODE" AND "BYTE" MODE
                        ;NOW SET THE "ACTIVE" FLOP AND VERIFY "INTERFACE IDLE" GOES LOW
005416  012737  000022  001124          MOV     #BIT4!BIT1,$GDDAT       ;LOAD EXPECTED VALUE
005424  052777  000001  174314          BIS     #BIT0,@CSR              ;SET "ACTIVE" TO A 1
005432  017737  174310  001126          MOV     @CSR,$BDDAT             ;READ STATUS
005440  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
005446  001404                          BEQ     1$              ;;BR IF EXPECTED
005450  052777  000400  174300          BIS     #ENDDMA,@SFR            ;STOP DMA IF POSSIBLE
005456  104002                          ERROR   2                       ;"ACTIVE" FLOP FAILED TO SET
                        ;POKE THE "END DMA" SIGNAL AND VERIFY "ACTIVE" CLEARS
005460  052777  000400  174270  1$:     BIS     #ENDDMA,@SFR            ;SEND "END DMA" SIGNAL
005466  012737  000222  001124          MOV     #BIT7!BIT4!BIT1,$GDDAT  ;SET "INTERFACE IDLE" INTO EXPECTED
005474  017737  174246  001126          MOV     @CSR,$BDDAT             ;READ STATUS
005502  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
005510  001401                          BEQ     2$              ;;BR IF SAME
005512  104002                          ERROR   2                       ;"END DMA" AGAIN FAILED TO CLEAR "ACTIVE" FLOP
                        ;POKE "ENABLE DMA" AND THEN ISSUE "CLR ALL" SIGNAL TO
                        ;       ENSURE THE "ACTIVE"  FLOP CLEARS
005514  012737  000200  001124  2$:     MOV     #BIT7,$GDDAT            ;LOAD EXPECTED
005522  052777  000001  174216          BIS     #BIT0,@CSR              ;POKE "ENABLE DMA"
005530  052777  004000  174220          BIS     #CLRALL,@SFR            ;GENERATE "CLR ALL"
005536  017737  174204  001126          MOV     @CSR,$BDDAT             ;READ STATUS
005544  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
005552  001401                          BEQ     TST20           ;;BR IF SAME
005554  104001                          ERROR   1                       ;"CLR ALL" FAILED TO CLEAR "ACTIVE" FLOPS
```

```
                                        ;;*****************************************************************
                                        ;*TEST 20        VERIFY Z INPUTS CAUSE THE LOW 8 BITS OF 32 BIT COUNTER TO CHANGEIN MATRI
                                        ;;*****************************************************************
005556  000004                        TST20:  SCOPE
005560  012737  000100  001160                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005566  012777  004000  174162                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
005574  012777  000010  174154                MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
005602  012777  000000  174142                MOV     #0,@WCR                 ;LOAD W.C. REG
005610  012777  000000  174136                MOV     #0,@BAR                 ;LOAD B.A.  REG
005616  012777  000022  174122                MOV     #BIT4!BIT1,@CSR         ;ENTER MATRIX MODE
005624  052777  000001  174114                BIS     #BIT0,@CSR              ;GO "ACTIVE"
005632  052777  000002  174116                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
005640  042777  000002  174110                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
005646  012737  000001  001124                MOV     #1,$GDDAT               ;LOAD EXPECTED
005654  017737  174074  001126                MOV     @BAR,$BDDAT             ;READ LOW 16 BITS
005662  001002                                BNE     1$              ;;BR IF NON-ZERO
005664  104010                                ERROR   10                      ;Z INPUT FAILED TO CAUSE THE LOW BYTE OF 32 BIT
005666  000420                                BR      TST21           ;;
005670  017737  174056  001126  1$:           MOV     @WCR,$BDDAT             ;READ HIGH 16 BITS
005676  001401                                BEQ     2$              ;;BR IF CLEARED
005700  104011                                ERROR   11                      ;HIGH 16 BIS CHANGED IN ERROR
005702  012737  000020  001124  2$:           MOV     #BIT4,$GDDAT            ;LOAD EXPECTED
005710  017737  174032  001126                MOV     @CSR,$BDDAT             ;READ STATUS
005716  032737  040000  001126                BIT     #BIT14,$BDDAT           ;TEST IF "CELL OVERFLOW"
005724  001401                                BEQ     TST21           ;;BR IF CORRECT
005726  104012                                ERROR   12                      ;"CELL OVERFLOW" FLOP SET IN ERROR
                                        ;;*****************************************************************
                                        ;*TEST 21        VERIFY Z INPUTS CAUSE THE LOW 16 BITS OF 32 BIT COUNTER TO CHANGE
                                        ;;*****************************************************************
005730  000004                        TST21:  SCOPE
005732  012737  000100  001160                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
005740  012777  004000  174010                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
005746  012777  000010  174002                MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
005754  012777  000000  173770                MOV     #0,@WCR                 ;LOAD W.C. REG
005762  012777  000376  173764                MOV     #376,@BAR               ;LOAD B.A.  REG
005770  012777  000022  173750                MOV     #BIT4!BIT1,@CSR         ;ENTER MATRIX MODE
005776  052777  000001  173742                BIS     #BIT0,@CSR              ;GO "ACTIVE"
006004  052777  000002  173744                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
006012  042777  000002  173736                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
006020  012737  000400  001124                MOV     #BIT8,$GDDAT            ;LOAD EXPECTED
006026  017737  173722  001126                MOV     @BAR,$BDDAT             ;READ LOW 16 BITS
006034  105737  001127                        TSTB    $BDDAT+1                ;TEST HIGH BYTE
006040  001002                                BNE     1$              ;BR IF NON-ZERO
006042  104010                                ERROR   10                      ;Z INPUT FAILED TO CAUSE THE HIGH BYTE OF THE LO
006044  000423                                BR      TST22           ;;
006046  012737  000000  001124  1$:           MOV     #0,$GDDAT               ;LOAD EXPECTED
006054  017737  173672  001126                MOV     @WCR,$BDDAT             ;READ HIGH 16 BITS
006062  001401                                BEQ     2$              ;;BR IF CLEARED
006064  104011                                ERROR   11                      ;HIGH 16 BITS CHANGED IN ERROR
006066  012737  000020  001124  2$:           MOV     #BIT4,$GDDAT            ;LOAD EXPECTED
006074  017737  173646  001126                MOV     @CSR,$BDDAT             ;READ STATUS
006102  032737  040000  001126                BIT     #BIT14,$BDDAT           ;TEST IF "CELL OVERFLOW"
006110  001401                                BEQ     TST22           ;;BR IF CORRECT
006112  104012                                ERROR   12                      ;"CELL OVEPFLOW" FLOP SET IN FRROR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 22       VERIFY Z INPUTS CAUSE THE LOW 24 BITS OF 32 BIT COUNTER TO CHANGE
                                        ;;*****************************************************************
006114  000004                  TST22:  SCOPE
006116  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006124  012777  004000  173624          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
006132  012777  000010  173616          MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
006140  012777  000000  173604          MOV     #0,@WCR                 ;LOAD W.C. REG
006146  012777  177776  173600          MOV     #-2,@BAR                ;LOAD B.A.  REG
006154  012777  000022  173564          MOV     #BIT4!BIT1,@CSR         ;ENTER MATRIX MODE
006162  052777  000001  173556          BIS     #BIT0,@CSR              ;GO "ACTIVE"
006170  052777  000002  173560          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
006176  042777  000002  173552          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
006204  012737  000001  001124          MOV     #1,$GDDAT               ;LOAD EXPECTED
006212  017737  173534  001126  1$:     MOV     @WCR,$BDDAT             ;READ HIGH 16 BITS
006220  001002                          BNE     2$                      ;BR IF SET
006222  104011                          ERROR   11                      ;LOW BYTE OF THE HIGH 16 BITS FAILED TO CHANGE
006224  000413                          BR      TST23           ;;
006226  012737  000020  001124  2$:     MOV     #BIT4,$GDDAT            ;LOAD EXPECTED
006234  017737  173506  001126          MOV     @CSR,$BDDAT             ;READ STATUS
006242  032737  040000  001126          BIT     #BIT14,$BDDAT           ;TEST IF "CELL OVERFLOW"
006250  001401                          BEQ     TST23           ;;BR IF CORRECT
006252  104012                          ERROR   12                      ;"CELL OVERFLOW" FLOP SET IN ERROR
                                        ;;*****************************************************************
                                        ;*TEST 23       VERIFY Z INPUTS CAUSE THE HIGH 8 BITS OF THE 32 BIT COUNTER TO CHANGE
                                        ;;*****************************************************************
006254  000004                  TST23:  SCOPE
006256  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006264  012777  004000  173464          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
006272  012777  000010  173456          MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
006300  012777  000377  173444          MOV     #377,@WCR               ;LOAD W.C. REG
006306  012777  177776  173440          MOV     #-2,@BAR                ;LOAD B.A.  REG
006314  012777  000022  173424          MOV     #BIT4!BIT1,@CSR         ;ENTER MATRIX MODE
006322  052777  000001  173416          BIS     #BIT0,@CSR              ;GO "ACTIVE"
006330  052777  000002  173420          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
006336  042777  000002  173412          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
006344  012737  000400  001124          MOV     #BIT8,$GDDAT            ;LOAD EXPECTED
006352  017737  173374  001126  1$:     MOV     @WCR,$BDDAT             ;READ HIGH 16 BITS
006360  105737  001127                  TSTB    $BDDAT+1                ;TEST HIGH BYTE
006364  001001                          BNE     TST24           ;;BR IF SET
006366  104011                          ERROR   11              ;HIGH 8 BITS OF THE 32 BIT COUNTER FAILED TO CHANGE
```

```
                                        ;;*********************************************************************
                                        ;*TEST 24        VERIFY Z INPUTS DO NOT CAUSE THE LOW 8 BITS OF 32 BIT COUNTER TO CHANGE
                                        ;;*********************************************************************
006370  000004                  TST24:  SCOPE
006372  012737  000100  001160          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
006400  012777  004000  173350          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
006406  012777  000000  173336          MOV     #0,@WCR                 ;LOAD W.C. REG
006414  012777  000000  173332          MOV     #0,@BAR                 ;LOAD B.A.  REG
006422  005077  173320                  CLR     @CSR                    ;ENSURE LIST MODE
006426  052777  000002  173322          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
006434  042777  000002  173314          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
006442  012737  000000  001124          MOV     #0,$GDDAT               ;LOAD EXPECTED
006450  017737  173300  001126          MOV     @BAR,$BDDAT             ;READ LOW 16 BITS
006456  001401                          BEQ     TST25           ;;BR IF ZERO
006460  104010                          ERROR   10              ;Z INPUT CAUSE THE 32 BIT COUNTER TO CHANGE IN LIST MODE

                                        ;;*********************************************************************
                                        ;*TEST 25        TEST THAT "WCA OVFL" SETS Z/WC OVERFLOW FLOP AND "CLR ALL" CLEARS IT
                                        ;;*********************************************************************
006462  000004                  TST25:  SCOPE
006464  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006472  012777  004000  173256          MOV     #CLRALL,@SFR            ;CLEAR DEVICE
006500  012777  000010  173250          MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
006506  012777  177777  173236          MOV     #-1,@WCR                ;LOAD Z COUNTER
006514  012777  177776  173232          MOV     #-2,@BAR                ;LOAD Z COUNTER
006522  012777  000022  173216          MOV     #BIT4!BIT1,@CSR         ;ENTER "MATRIX" MODE
006530  052777  000001  173210          BIS     #BIT0,@CSR              ;ENABLE NCV11
                                ;NOW ENABLE "TEST Z" INPUTS
006536  052777  000002  173212          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
006544  042777  000002  173204          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
006552  013700  002006                  MOV     CPUDL0,R0               ;LOAD GROSS TIMER
006556  005001                          CLR     R1
006560  012737  040020  001124          MOV     #BIT14!BIT4,$GDDAT      ;LOAD EXPECTED STATUS

006566  032777  040000  173152  1$:     BIT     #BIT14,@CSR             ;TEST FOR Z/WC OVERFLOW
006574  001011                          BNE     2$                      ;BR IF SET
006576  005301                          DEC     R1                      ;DELAY
006600  001372                          BNE     1$
006602  005300                          DEC     R0                      ;DELAY
006604  001370                          BNE     1$
006606  012777  000400  173142          MOV     #ENDDMA,@SFR            ;STOP DMA TRANSFERS
006614  104012                          ERROR   12              ;AFTER A GROSS TIME, THE Z/WC OVERFLOW FLOP FAILED TO SE
006616  000416                          BR      TST26           ;;

                                ;NOW GENERATE "CLR ALL" TO CLEAR Z/WC OVERFLOW BIT
006620  052777  004000  173130  2$:     BIS     #CLRALL,@SFR            ;CLEAR Z/WC FLOP
006626  012737  000200  001124          MOV     #BIT7,$GDDAT            ;LOAD EXPECTED
006634  017737  173106  001126          MOV     @CSR,$BDDAT             ;READ STATUS
006642  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
006650  001401                          BEQ     TST26           ;;BR IF SAME
006652  104012                          ERROR   12                      ;"CLR ALL" FAILED TO CLEAR "Z/WC" FLOP
```

```
                                        ;;*****************************************************************
                                        ;*TEST 26       TEST THAT "WCA OVFL" SETS Z/WC OVERFLOW FLOP AND "CLR WC OVFL" CLEARS IT
                                        ;;*****************************************************************
006554  000004                  TST26:  SCOPE
006656  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006664  012777  004000  173064          MOV     #CLRALL,@SFR            ;CLEAR DEVICE
006672  012777  000010  173056          MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
006700  012777  177777  173044          MOV     #-1,@WCR                ;LOAD Z COUNTER
006706  012777  177776  173040          MOV     #-2,@BAR                ;LOAD Z COUNTER
006714  012777  000022  173024          MOV     #BIT4!BIT1,@CSR         ;ENABLE "MATRIX" MODE
006722  052777  000001  173016          BIS     #BIT0,@CSR              ;SET NCV11 ACTIVE
                        ;NOW ENABLE "TEST Z" INPUTS
006730  052777  000002  173020          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
006736  042777  000002  173012          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES

006744  013700  002006                  MOV     CPUDL0,R0               ;LOAD GROSS TIMER
006750  005001                          CLR     R1
006752  012737  040020  001124          MOV     #BIT14!BIT4,$GDDAT      ;LOAD EXPECTED STATUS

006760  032777  040000  172760  1$:     BIT     #BIT14,@CSR             ;TEST FOR Z/WC OVERFLOW
006766  001014                          BNE     2$                      ;BR IF SET
006770  005301                          DEC     R1                      ;DELAY
006772  001372                          BNE     1$
006774  005300                          DEC     R0                      ;DELAY
006776  001370                          BNE     1$
007000  012777  000400  172750          MOV     #ENDDMA,@SFR            ;STOP DAM TRANSFERS
007006  042777  000002  172742          BIC     #TESTZ,@SFR             ;STOP "Z" INPUTS
007014  104012                          ERROR   12      ;AFTER A GROSS TIME, THE Z/WC OVERFLOW FLOP FAILED TO SET
007016  000416                          BR      TST27           ;;

                        ;NOW GENERATE "CLR WC OVFL" TO CLEAR Z/WC OVERFLOW BIT
007020  052777  040000  172730  2$:     BIS     #CLRWCO,@SFR            ;CLEAR Z/WC FLOP
007026  012737  000222  001124          MOV     #BIT7!BIT4!BIT1,$GDDAT ;LOAD EXPECTED
007034  017737  172706  001126          MOV     @CSR,$BDDAT             ;READ STATUS
007042  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007050  001401                          BEQ     TST27           ;;BR IF SAME
007052  104012                          ERROR   12                      ;"CLR WC OVFL" FAILED TO
                                                                ; CLEAR "Z/WC" FLOP (BIT 14)
```

```
                                ;;*****************************************************************
                                ;*TEST 27       TEST THAT "WCA OVFL" GENERATES AN INTERRUPT
                                ;;*****************************************************************
007054  000004                  TST27:  SCOPE
007056  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007064  012777  004000  172664          MOV     #CLRALL,@SFR            ;CLEAR DEVICE
007072  012777  000010  172656          MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
007100  012777  177777  172644          MOV     #-1,@WCR                ;LOAD Z COUNTER
007106  012777  177776  172640          MOV     #-2,@BAR                ;LOAD Z COUNTER
007114  012777  000022  172624          MOV     #BIT4!BIT1,@CSR         ;SET MATRIX MODE
007122  052777  000001  172616          BIS     #BIT0,@CSR              ;SET NCV11 ACTIVE
                        ;NOW ENABLE "TEST Z" INPUTS
007130  052777  000002  172620          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
007136  042777  000002  172612          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
007144  013700  002006                  MOV     CPUDL0,R0               ;LOAD GROSS TIMER
007150  005001                          CLR     R1
007152  012737  040020  001124          MOV     #BIT14!BIT4,$GDDAT      ;LOAD EXPECTED STATUS
                        
007160  032777  040000  172560  1$:     BIT     #BIT14,@CSR             ;TEST FOR Z/WC OVERFLOW
007166  001014                          BNE     2$                      ;BR IF SET
007170  005301                          DEC     R1                      ;DELAY
007172  001372                          BNE     1$
007174  005300                          DEC     R0                      ;DELAY
007176  001370                          BNE     1$
007200  012777  000400  172550          MOV     #ENDDMA,@SFR            ;STOP DMA TRANSFERS
007206  042777  000002  172542          BIC     #TESTZ,@SFR             ;STOP "Z" INPUTS
007214  104012                          ERROR   12      ;AFTER A GROSS TIME, THE Z/WC OVERFLOW FLOP FAILED TO SET
007216  000430                          BR      5$              ;;BR TO CLEAN UP
                        
                        ;NOW ENABLE THE WC/Z OVERFLOW INTERRUPT BIT AND WAIT FOR AN INTERRUPT
007220  012746  000000  2$:     MOV     #0,-(SP)
007224  012746  007232          MOV     #3$,-(SP)               ;LSI-11 HACK
007230  000002                  RTI
007232  012777  007276  172532  3$:     MOV     #4$,@VECTA0             ;LOAD INTR. VECTOR
007240  052777  000100  172500          BIS     #BIT6,@CSR              ;ENABLE INTERRUPT
007246  000240                          NOP
007250  000240                          NOP
007252  000240                          NOP
007254  000240                          NOP
007256  017737  172464  001126          MOV     @CSR,$BDDAT             ;READ STATUS
007264  012777  004000  172464          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
007272  104013                          ERROR   13                      ;WC/Z OVERFLOW FAILED TO GENERATE INTERRUPT
007274  000401                          BR      5$              ;;BR TO CLEAN UP
                        
007276  022626          4$:     CMP     (SP)+,(SP)+
007300  005077  172442  5$:     CLR     @CSR
007304  012777  004000  172444          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
007312  013777  001774  172452          MOV     VECTA1,@VECTA0
007320  005077  172450                  CLR     @VECTA1
```

```
                                ;;*****************************************************************
                                ;*TEST 30       VERIFY "WCA OVFL" CLEARS "ACTIVE"
                                ;;*****************************************************************
007324  000004                  TST30:  SCOPE
007326  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007334  012777  004000  172414          MOV     #CLRALL,@SFR            ;CLEAR DEVICE
007342  012777  000014  172406          MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA
007350  012777  177777  172374          MOV     #-1,@WCR                ;LOAD W.C. REG.
007356  012777  177776  172370          MOV     #-2,@BAR                ;LOAD BAR REG.
007364  005077  172360                  CLR     @OFF                    ;CLEAR OFFSET REG.
007370  012777  000022  172350          MOV     #BIT4!BIT1,@CSR         ;ENABLE MATRIX MODE
007376  052777  000001  172342          BIS     #BIT0,@CSR              ;SET "ACTIVE"
007404  052777  000002  172344          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
007412  042777  000002  172336          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
007420  017737  172322  001126          MOV     @CSR,$BDDAT             ;READ STATUS
007426  012737  040222  001124          MOV     #BIT14!BIT7!BIT4!BIT1,$GDDAT    ;LOAD EXPECTED
007434  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007442  001402                          BEQ     1$              ;;BR IF SAME
007444  104012                          ERROR   12                      ;"WCA OVFL" FAILED TO
                                                                ;CLEAR "ACTIVE" (BIT 7 SET)
007446  000407                          BR      TST31           ;;

007450  005037  001124          1$:     CLR     $GDDAT                  ;CLEAR THE EXPECTED
007454  017737  172270  001126          MOV     @OFF,$BDDAT             ;READ THE ACTUAL
007462  001401                          BEQ     TST31           ;;BR IF CLEARED
007464  104012                          ERROR   12                      ;"WCA INC X OFF" IN MATRIX MODE CHANGED
                                                                ;THE OFFSET REGISTER
```

```
                                        ;;*****************************************************************
                                        ;*TEST 31       VERIFY JOYSTICK DONE FLOP SETS
                                        ;;*****************************************************************
007466  000004                  TST31:  SCOPE
007470  012737  000040  001160          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
007476  012777  004000  172252          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
007504  052777  000001  172244          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICK DATA
007512  013700  002010                  MOV     CPUDL1,R0               ;LOAD DELAY COUNTER
007516  105777  172234          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICK READY
007522  100411                          BMI     2$                      ;BR IF SET
007524  005300                          DEC     R0                      ;DELAY
007526  001373                          BNE     1$                      ;BR IF NOT EXHAUSTED
007530  017737  172222  001126          MOV     @SFR,$BDDAT             ;READ STATUS
007536  012737  000200  001124          MOV     #BIT7,$GDDAT            ;LOAD EXPECTED
007544  104014                          ERROR   14                      ;"JOYSTICK READY" FAILED TO SET
                                ;NOW PERFORM A "TST" INSTRUCTION TO THE JOYSTICK REGISTER
                                ;       THE ACCESS OF THIS BUS ADDRESS SHOULD CLEAR JOY READY FLOP"
007546  005777  172210          2$:     TST     @JOY                    ;ADDRESS THE REGISTER
007552  017737  172200  001126          MOV     @SFR,$BDDAT             ;READ STATUS
007560  105737  001126                  TSTB    $BDDAT                  ;TEST IF THE BIT CLEARED
007564  100003                          BPL     TST32           ;;BR IF YES
007566  005037  001124                  CLR     $GDDAT                  ;LOAD EXPECTED
007572  104014                          ERROR   14                      ;ADDRESSING THE JOYSTICK ADDRESS FAILED TO CLEAR
                                        ;;*****************************************************************
                                        ;*TEST 32       VERIFY THAT "RESET" INSTRUCTION CLEARS THE JOY READY FLOP
                                        ;;*****************************************************************
007574  000004                  TST32:  SCOPE
007576  012737  000040  001160          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
007604  012777  004000  172144          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
007612  052777  000001  172136          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICK DATA
007620  013700  002010                  MOV     CPUDL1,R0               ;LOAD DELAY
007624  105777  172126          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICK READY
007630  100411                          BMI     2$                      ;BR IF SET
007632  005300                          DEC     R0                      ;DELAY
007634  001373                          BNE     1$                      ;BR IF NOT EXHAUSTED
007636  017737  172114  001126          MOV     @SFR,$BDDAT             ;READ STATUS
007644  012737  000200  001124          MOV     #BIT7,$GDDAT            ;LOAD EXPECTED VALUE
007652  104014                          ERROR   14                      ;JOY READY FAILED TO SET
                                ;NOW ISSUE A "CLR ALL" AND VERIFY THE "JOY READY" DOES NOT CLEAR
007654  052777  004000  172074  2$:     BIS     #CLRALL,@SFR            ;GENERATE "CLR ALL"
007662  017737  172070  001126          MOV     @SFR,$BDDAT             ;READ STATUS
007670  012737  000200  001124          MOV     #BIT7,$GDDAT    ;LOAD EXPECTED VALUE
007676  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007704  001401                          BEQ     3$              ;;BR IF SAME
007706  104014                          ERROR   14                      ;"CLR ALL" CLEARED THE "JOY READY" FLOP IN ERROR
                                ;NOW ISSUE A BUS "RESET" AND VERIFY THE JOY READY" CLEARS
007710  000005                  3$:     RESET                           ;BUS "INIT"
007712  017737  172040  001126          MOV     @SFR,$BDDAT             ;READ STATUS
007720  012737  000000  001124          MOV     #0,$GDDAT               ;LOAD EXPECTED
007726  017737  172024  001126          MOV     @SFR,$BDDAT             ;READ STATUS
007734  001401                          BEQ     TST33           ;;BR IF CLEARED
007736  104014                          ERROR   14                      ;BUS INIT FAILED TO CLEAR JOY READY
```

```
                                ;;*****************************************************************
                                ;*TEST 33       JOYSTICK DATA PATH  =  GAIN =0  ZB ENABLE =0  RES. = 000
                                ;;*****************************************************************
007740  000004                  TST33:  SCOPE
007742  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007750  012777  004000  172000          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
007756  012777  000014  171772          MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
007764  012777  000000  171754          MOV     #0,@CSR         ;LOAD CSR REGISTER
007772  052777  000001  171756          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
010000  105777  171752          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
010004  100375                          BPL     1$
010006  017737  171750  001126          MOV     @JOY,$BDDAT             ;READ THE REGISTER
010014  012737  000000  001124          MOV     #0,$GDDAT               ;LOAD EXPECTED
010022  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
010030  001401                          BEQ     TST34           ;;BR IF SAME
010032  104015                          ERROR   15                      ;JOYSTICK DATA PATH BIT SET

                                ;;*****************************************************************
                                ;*TEST 34       JOYSTICK DATA PATH  =  GAIN =1
                                ;;*****************************************************************
010034  000004                  TST34:  SCOPE
010036  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
010044  012777  004000  171704          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
010052  012777  000014  171676          MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
010060  012777  002000  171660          MOV     #2000,@CSR              ;LOAD CSR REGISTER
010066  052777  000001  171662          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
010074  105777  171656          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
010100  100375                          BPL     1$
010102  017737  171654  001126          MOV     @JOY,$BDDAT             ;READ THE REGISTER
010110  012737  124250  001124          MOV     #124250,$GDDAT          ;LOAD EXPECTED
010116  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
010124  001401                          BEQ     TST35           ;;BR IF SAME
010126  104015                          ERROR   15                      ;JOYSTICK DATA PATH ERROR

                                ;;*****************************************************************
                                ;*TEST 35       JOYSTICK DATA PATH  =  ZB ENABLE =1
                                ;;*****************************************************************
010130  000004                  TST35:  SCOPE
010132  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
010140  012777  004000  171610          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
010146  012777  000014  171602          MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
010154  012777  004000  171564          MOV     #4000,@CSR              ;LOAD CSR REGISTER
010162  052777  000001  171566          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
010170  105777  171562          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
010174  100375                          BPL     1$
010176  017737  171560  001126          MOV     @JOY,$BDDAT             ;READ THE REGISTER
010204  012737  050120  001124          MOV     #050120,$GDDAT          ;LOAD EXPECTED
010212  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
010220  001401                          BEQ     TST36           ;;BR IF SAME
010222  104015                          ERROR   15                      ;JOYSTICK DATA PATH ERROR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 36        JOYSTICK DATA PATH  RES. = 001
                                        ;;*****************************************************************
010224  000004                        TST36:  SCOPE
010226  012737  000010  001160                MOV     #10,$TIMES      ;;DO 10 ITERATIONS
010234  012777  004000  171514                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
010242  012777  000014  171506                MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
010250  012777  000023  171470                MOV     #23,@CSR                ;LOAD CSR REGISTER
010256  052777  000001  171472                BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
010264  105777  171466          1$:           TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
010270  100375                                BPL     1$
010272  017737  171464  001126                MOV     @JOY,$BDDAT             ;READ THE REGISTER
010300  012737  000401  001124                MOV     #401,$GDDAT             ;LOAD EXPECTED
010306  023737  001124  001126                CMP     $GDDAT,$BDDAT           ;COMPARE
010314  001401                                BEQ     TST37           ;;BR IF SAME
010316  104015                                ERROR   15                      ;JOYSTICK DATA PATH ERROR  RES. - 001

                                        ;;*****************************************************************
                                        ;*TEST 37        JOYSTICK DATA PATH  RES. = 010
                                        ;;*****************************************************************
010320  000004                        TST37:  SCOPE
010322  012737  000010  001160                MOV     #10,$TIMES      ;;DO 10 ITERATIONS
010330  012777  004000  171420                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
010336  012777  000014  171412                MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
010344  012777  000025  171374                MOV     #25,@CSR                ;LOAD CSR REGISTER
010352  052777  000001  171376                BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
010360  105777  171372          1$:           TSTB    @SFR                    ;WAIT FOR JOYSTICE REA Y
010364  100375                                BPL     1$
010366  017737  171370  001126                MOV     @JOY,$BDDAT             ;READ THE REGISTER
010374  012737  001002  001124                MOV     #1002,$GDDAT            ;LOAD EXPECTED
010402  023737  001124  001126                CMP     $GDDAT,$BDDAT           ;COMPARE
010410  001401                                BEQ     TST40           ;;BR IF SAME
010412  104015                                ERROR   15                      ;JOYSTICK DATA PATH ERROR  RES. = 010

                                        ;;*****************************************************************
                                        ;*TEST 40        JOYSTICK DATA PATH  RES. = 100
                                        ;;*****************************************************************
010414  000004                        TST40:  SCOPE
010416  012737  000010  001160                MOV     #10,$TIMES      ;;DO 10 ITERATIONS
010424  012777  004000  171324                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
010432  012777  000014  171316                MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
010440  012777  000031  171300                MOV     #31,@CSR                ;LOAD CSR REGISTER
010446  052777  000001  171302                BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
010454  105777  171276          1$:           TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
010460  100375                                BPL     1$
010462  017737  171274  001126                MOV     @JOY,$BDDAT             ;READ THE REGISTER
010470  012737  002004  001124                MOV     #2004,$GDDAT            ;LOAD EXPECTED
010476  023737  001124  001126                CMP     $GDDAT,$BDDAT           ;COMPARE
010504  001401                                BEQ     TST41           ;;BR IF SAME
010506  104015                                ERROR   15                      ;JOY STICK DATA PATH ERROR  RES.  100
```

```
907                                     ;;*****************************************************************
(3)                                     ;*TEST 41        VERIFY THE DATA INCREMENT FUNCTION
(3)                                     ;;*****************************************************************
(2)  010510  000004                     TST41:  SCOPE
(1)  010512  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
908  010520  012777  004000  171230             MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
(1)  010526  012777  000014  171222             MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
(1)  010534  012777  000421  171204             MOV     #421,@CSR               ;LOAD CSR REGISTER
(1)  010542  052777  000001  171206             BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
(1)  010550  105777  171202             1$.     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
(1)  010554  100375                             BPL     1$
(1)  010556  017737  171200  001126             MOV     @JOY,$BDDAT             ;READ THE REGISTER
(1)  010564  012737  000401  001124             MOV     #401,$GDDAT             ;LOAD EXPECTED
(1)  010572  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE
909  010600  001401                             BEQ     TST42           ;;BR IF SAME
910  010602  104016                             ERROR   16                      ;MAINT. CAM01 FAILED TO INCREMENT DATA REGISTER
911
912                                     ;;*****************************************************************
(3)                                     ;*TEST 42        VERIFY THE DATA INCREMENT CARRY BIT
(3)                                     ;;*****************************************************************
 2)  010604  000004                     TST42:  SCOPE
(1)  010606  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
913  010614  012777  004000  171134             MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
(1)  010622  012777  000014  171126             MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
(1)  010630  012777  002437  171110             MOV     #2437,@CSR              ;LOAD CSR REGISTER
(1)  010636  052777  000001  171112             BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
(1)  010644  105777  171106             1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
(1)  010650  100375                             BPL     1$
(1)  010652  017737  171104  001126             MOV     @JOY,$BDDAT             ;READ THE REGISTER
(1)  010660  012737  130260  001124             MOV     #130260,$GDDAT          ;LOAD EXPECTED
(1)  010666  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE
914  010674  001401                             BEQ     TST43           ;;BR IF SAME
915  010676  104016                             ERROR   16                      ;MAINT. CAM01 WITH RES.=7, G=1, ZB  0
916                                                                             ;FAILED TO CAUSE DATA INCREMENT CARRY PROPERLY
917
918                                     ;;*****************************************************************
(3)                                     ;*TEST 43        VERIFY THE DATA INCREMENT FUNCTION IS INHIBITED
(3)                                     ;;*****************************************************************
(2)  010700  000004                     TST43:  SCOPE
(1)  010702  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
919  010710  012777  004000  171040             MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
(1)  010716  012777  000014  171032             MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
(1)  010724  012777  006437  171014             MOV     #6437,@CSR              ;LOAD CSR REGISTER
(1)  010732  052777  000001  171016             BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
(1)  010740  105777  171012             1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
(1)  010744  100375                             BPL     1$
(1)  010746  017737  171010  001126             MOV     @JOY,$BDDAT             ;READ THE REGISTER
(1)  010754  012737  177777  001124             MOV     #177777,$GDDAT          ;LOAD EXPECTED
(1)  010762  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE
920  010770  001401                             BEQ     TST44           ;;BR IS SAME
921  010772  104016                             ERROR   16              ;FAILED TO INHIBIT DATA INCREMENT FUNCTION
922                                                                             ;IF THE BAD DATA WAS 0
```

```
                                        ;;*****************************************************************
                                        ;*TEST 44        VERIFY THE DATA DECREMENT FUNCTION
                                        ;;*****************************************************************
010774  000004                  TST44:  SCOPE
010776  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
011004  012777  004000  170744          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
011012  012777  000014  170736          MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
011020  012777  001023  170720          MOV     #1023,@CSR              ;LOAD CSR REGISTER
011026  052777  000001  170722          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
011034  105777  170716          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
011040  100375                          BPL     1$
011042  017737  170714  001126          MOV     @JOY,$BDDAT             ;READ THE REGISTER
011050  012737  000000  001124          MOV     #0,$GDDAT               ;LOAD EXPECTED
011056  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011064  001401                          BEQ     TST45           ;;BR IF SAME
011066  104017                          ERROR   17                      ;IF DATA WAS 401, THE DATA DECREMENT FUNCTION FAILED
                                                                        ;OTHERWISE DECREMENTED DATA ERROR

                                        ;;*****************************************************************
                                        ;*TEST 45        VERIFY THE DATA DECREMENT BORROW
                                        ;;*****************************************************************
011070  000004                  TST45:  SCOPE
011072  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
011100  012777  004000  170650          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
011106  012777  000014  170642          MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
011114  012777  007037  170624          MOV     #7037,@CSR              ;LOAD CSR REGISTER
011122  052777  000001  170626          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
011130  105777  170622          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
011134  100375                          BPL     1$
011136  017737  170620  001126          MOV     @JOY,$BDDAT             ;READ THE REGISTER
011144  012737  177376  001124          MOV     #177376,$GDDAT          ;LOAD EXPECTED
011152  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011160  001401                          BEQ     TST46           ;;BR IF SAME
011162  104017                          ERROR   17                      ;DECREMENTED DATA ERROR

                                        ;;*****************************************************************
                                        ;*TEST 46        VERIFY THE DATA DECREMENT FUNCTION IS INHIBITED
                                        ;;*****************************************************************
011164  000004                  TST46:  SCOPE
011166  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
011174  012777  004000  170554          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
011202  012777  000014  170546          MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROLLER FLOP
011210  012777  001021  170530          MOV     #1021,@CSR              ;LOAD CSR REGISTER
011216  052777  000001  170532          BIS     #REDJOY,@SFR            ;REQUEST JOYSTICE DATA
011224  105777  170526          1$:     TSTB    @SFR                    ;WAIT FOR JOYSTICE READY
011230  100375                          BPL     1$
011232  017737  170524  001126          MOV     @JOY,$BDDAT             ;READ THE REGISTER
011240  012737  000000  001124          MOV     #0,$GDDAT               ;LOAD EXPECTED
011246  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011254  001401                          BEQ     TST47           ;;BR IF SAME
011256  104017                          ERROR   17                      ;INHIBIT DECREMENT FUNCTION ERROR
                                                                        ;IF DATA WAS 177777
```

```
956                                     ;;*****************************************************************
(4)                                     ;*TEST 47       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 7 GAIN = 0 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  011260  000004                     TST47:  SCOPE
(2)  011262  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  011270  012777  004000  170460             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  011276  012777  000014  170452             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  011304  012777  000036  170434             MOV     #36,@CSR                ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  011312  052777  000002  170436             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  011320  052777  000001  170420             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  011326  042777  000002  170422             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  011334  017737  170420  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  011342  012737  003407  001124             MOV     #3407,$GDDAT            ;LOAD EXPECTED VALUE
(1)  011350  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  011356  001401                             BEQ     TST50           ;;BR IF SAME
(1)  011360  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 7  GAIN = 0  ZB ENABLE = 0
957                                     ;;*****************************************************************
(4)                                     ;*TEST 50       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 7 GAIN = 1 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  011362  000004                     TST50:  SCOPE
(2)  011364  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  011372  012777  004000  170356             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  011400  012777  000014  170350             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  011406  012777  002036  170332             MOV     #2036,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  011414  052777  000002  170334             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  011422  052777  000001  170316             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  011430  042777  000002  170320             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  011436  017737  170316  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  011444  012737  127657  001124             MOV     #127657,$GDDAT          ;LOAD EXPECTED VALUE
(1)  011452  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  011460  001401                             BEQ     TST51           ;;BR IF SAME
(1)  011462  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 7  GAIN = 1  ZB ENABLE = 0
958                                     ;;*****************************************************************
(4)                                     ;*TEST 51       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 7 GAIN = 0 ZB ENABLE = 1
(4)                                     ;;*****************************************************************
(3)  011464  000004                     TST51:  SCOPE
(2)  011466  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  011474  012777  004000  170254             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  011502  012777  000014  170246             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  011510  012777  004036  170230             MOV     #4036,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  011516  052777  000002  170232             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  011524  052777  000001  170214             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  011532  042777  000002  170216             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  011540  017737  170214  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  011546  012737  053527  001124             MOV     #053527,$GDDAT          ;LOAD EXPECTED VALUE
(1)  011554  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  011562  001401                             BEQ     TST52           ;;BR IF SAME
(1)  011564  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 7  GAIN = 0  ZB ENABLE = 1
```

```
960                                     ;;*****************************************************************
(4)                                     ;*TEST 52       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 6 GAIN = 0 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  011566  000004                     TST52:  SCOPE
(2)  011570  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  011576  012777  004000  170152             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  011604  012777  000014  170144             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  011612  012777  000034  170126             MOV     #34,@CSR                ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  011620  052777  000002  170130             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  011626  052777  000001  170112             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  011634  042777  000002  170114             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  011642  017737  170112  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  011650  012737  001406  001124             MOV     #1406,$GDDAT            ;LOAD EXPECTED VALUE
(1)  011656  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  011664  001401                             BEQ     TST53           ;;BR IF SAME
(1)  011666  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 6  GAIN = 0  ZB ENABLE = 0
961                                     ;;*****************************************************************
(4)                                     ;*TEST 53       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 6 GAIN = 1 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  011670  000004                     TST53:  SCOPE
(2)  011672  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  011700  012777  004000  170050             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  011706  012777  000014  170042             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  011714  012777  002034  170024             MOV     #2034,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  011722  052777  000002  170026             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  011730  052777  000001  170010             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  011736  042777  000002  170012             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  011744  017737  170010  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  011752  012737  053656  001124             MOV     #53656,$GDDAT           ;LOAD EXPECTED VALUE
(1)  011760  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  011766  001401                             BEQ     TST54           ;;BR IF SAME
(1)  011770  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 6  GAIN = 1  ZB ENABLE = 0
962                                     ;;*****************************************************************
(4)                                     ;*TEST 54       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 6 GAIN - 0 ZB ENABLE = 1
(4)                                     ;;*****************************************************************
(3)  011772  000004                     TST54:  SCOPE
(2)  011774  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  012002  012777  004000  167746             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  012010  012777  000014  167740             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  012016  012777  004034  167722             MOV     #4034,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  012024  052777  000002  167724             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  012032  052777  000001  167706             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  012040  042777  000002  167710             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  012046  017737  167706  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  012054  012737  125526  001124             MOV     #125526,$GDDAT          ;LOAD EXPECTED VALUE
(1)  012062  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  012070  001401                             BEQ     TST55           ;;BR IF SAME
(1)  012072  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 6  GAIN = 0  ZB ENABLE - 1
```

```
964                                     ;;*****************************************************************
(4)                                     ;*TEST 55       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 5 GAIN = 0 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  012074  000004                     TST55:  SCOPE
(2)  012076  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  012104  012777  004000  167644             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  012112  012777  000014  167636             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  012120  012777  000032  167620             MOV     #32,@CSR                ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  012126  052777  000002  167622             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  012134  052777  000001  167604             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  012142  042777  000002  167606             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  012150  017737  167604  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  012156  012737  000402  001124             MOV     #402,$GDDAT             ;LOAD EXPECTED VALUE
(1)  012164  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  012172  001401                             BEQ     TST56           ;;BR IF SAME
(1)  012174  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 5  GAIN = 0  ZB ENABLE = 0
965                                     ;;*****************************************************************
(4)                                     ;*TEST 56       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 5 GAIN = 1 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  012176  000004                     TST56:  SCOPE
(2)  012200  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  012206  012777  004000  167542             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  012214  012777  000014  167534             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  012222  012777  002032  167516             MOV     #2032,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  012230  052777  000002  167520             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  012236  052777  000001  167502             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  012244  042777  000002  167504             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  012252  017737  167502  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  012260  012737  025526  001124             MOV     #25526,$GDDAT           ;LOAD EXPECTED VALUE
(1)  012266  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  012274  001401                             BEQ     TST57           ;;BR IF SAME
(1)  012276  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 5  GAIN = 1  ZB ENABLE = 0
966                                     ;;*****************************************************************
(4)                                     ;*TEST 57       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 5 GAIN = 0 ZB ENABLE = 1
(4)                                     ;;*****************************************************************
(3)  012300  000004                     TST57:  SCOPE
(2)  012302  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  012310  012777  004000  167440             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  012316  012777  000014  167432             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  012324  012777  004032  167414             MOV     #4032,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  012332  052777  000002  167416             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  012340  052777  000001  167400             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  012346  042777  000002  167402             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  012354  017737  167400  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  012362  012737  052452  001124             MOV     #52452,$GDDAT           ;LOAD EXPECTED VALUE
(1)  012370  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  012376  001401                             BEQ     TST60           ;;BR IF SAME
(1)  012400  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 5  GAIN = 0  ZB ENABLE = 1
```

```
968                                     ;;*****************************************************************
(4)                                     ;*TEST 60        TEST ADDRESS MAKER  -  MATRIX MODE - RES = 4 GAIN = 0 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  012402  000004                     TST60:  SCOPE
(2)  012404  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  012412  012777  004000  167336             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  012420  012777  000014  167330             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  012426  012777  000030  167312             MOV     #30,@CSR                ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  012434  052777  000002  167314             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  012442  052777  000001  167276             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  012450  042777  000002  167300             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  012456  017737  167276  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  012464  012737  000202  001124             MOV     #202,$GDDAT             ;LOAD EXPECTED VALUE
(1)  012472  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  012500  001401                             BEQ     TST61           ;;BR IF SAME
(1)  012502  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 4  GAIN = 0  ZB ENABLE = 0
969                                     ;;*****************************************************************
(4)                                     ;*TEST 61        TEST ADDRESS MAKER  -  MATRIX MODE - RES = 4 GAIN = 1 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  012504  000004                     TST61:  SCOPE
(2)  012506  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  012514  012777  004000  167234             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  012522  012777  000014  167226             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  012530  012777  002030  167210             MOV     #2030,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  012536  052777  000002  167212             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  012544  052777  000001  167174             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  012552  042777  000002  167176             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  012560  017737  167174  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  012566  012737  012726  001124             MOV     #012726,$GDDAT          ;LOAD EXPECTED VALUE
(1)  012574  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  012602  001401                             BEQ     TST62           ;;BR IF SAME
(1)  012604  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 4  GAIN = 1  ZB ENABLE = 0
970                                     ;;*****************************************************************
(4)                                     ;*TEST 62        TEST ADDRESS MAKER  -  MATRIX MODE - RES = 4 GAIN = 0 ZB ENABLE = 1
(4)                                     ;;*****************************************************************
(3)  012606  000004                     TST62:  SCOPE
(2)  012610  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  012616  012777  004000  167132             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  012624  012777  000014  167124             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  012632  012777  004030  167106             MOV     #4030,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  012640  052777  000002  167110             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  012646  052777  000001  167072             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  012654  042777  000002  167074             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  012662  017737  167072  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  012670  012737  025252  001124             MOV     #025252,$GDDAT          ;LOAD EXPECTED VALUE
(1)  012676  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  012704  001401                             BEQ     TST63           ;;BR IF SAME
(1)  012706  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 4  GAIN = 0  ZB ENABLE = 1
```

```
 972                                    ;;*****************************************************************
 (4)                                    ;*TEST 63       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 3 GAIN = 0 ZB ENABLE = 0
 (4)                                    ;;*****************************************************************
 (3)  012710  000004                    TST63:  SCOPE
 (2)  012712  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
 (1)  012720  012777  004000  167030            MOV     #CLRALL,@SFR            ;CLEAR DEVICE
 (1)  012726  012777  000014  167022            MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
 (1)  012734  012777  000026  167004            MOV     #26,@CSR                ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
 (1)  012742  052777  000002  167006            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  012750  052777  000001  166770            BIS     #BIT0,@CSR              ;SET ENABLE NCV11
 (1)  012756  042777  000002  166772            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  012764  017737  166770  001126            MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
 (1)  012772  012737  000000  001124            MOV     #0,$GDDAT               ;LOAD EXPECTED VALUE
 (1)  013000  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
 (3)  013006  001401                            BEQ     TST64           ;;BR IF SAME
 (1)  013010  104020                            ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
 (1)                                    ;                       RESOLUTION = 3  GAIN = 0  ZB ENABLE = 0
 973                                    ;;*****************************************************************
 (4)                                    ;*TEST 64       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 3 GAIN = 1 ZB ENABLE = 0
 (4)                                    ;;*****************************************************************
 (3)  013012  000004                    TST64:  SCOPE
 (2)  013014  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
 (1)  013022  012777  004000  166726            MOV     #CLRALL,@SFR            ;CLEAR DEVICE
 (1)  013030  012777  000014  166720            MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
 (1)  013036  012777  002026  166702            MOV     #2026,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
 (1)  013044  052777  000002  166704            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  013052  052777  000001  166666            BIS     #BIT0,@CSR              ;SET ENABLE NCV11
 (1)  013060  042777  000002  166670            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  013066  017737  166666  001126            MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
 (1)  013074  012737  005252  001124            MOV     #5252,$GDDAT            ;LOAD EXPECTED VALUE
 (1)  013102  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
 (3)  013110  001401                            BEQ     TST65           ;;BR IF SAME
 (1)  013112  104020                            ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
 (1)                                    ;                       RESOLUTION = 3  GAIN = 1  ZB ENABLE = 0
 974                                    ;;*****************************************************************
 (4)                                    ;*TEST 65       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 3 GAIN = 0 ZB ENABLE = 1
 (4)                                    ;;*****************************************************************
 (3)  013114  000004                    TST65:  SCOPE
 (2)  013116  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
 (1)  013124  012777  004000  166624            MOV     #CLRALL,@SFR            ;CLEAR DEVICE
 (1)  013132  012777  000014  166616            MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
 (1)  013140  012777  004026  166600            MOV     #4026,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
 (1)  013146  052777  000002  166602            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  013154  052777  000001  166564            BIS     #BIT0,@CSR              ;SET ENABLE NCV11
 (1)  013162  042777  000002  166566            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  013170  017737  166564  001126            MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
 (1)  013176  012737  012424  001124            MOV     #12424,$GDDAT           ;LOAD EXPECTED VALUE
 (1)  013204  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
 (3)  013212  001401                            BEQ     TST66           ;;BR IF SAME
 (1)  013214  104020                            ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
 (1)                                    ;                       RESOLUTION = 3  GAIN = 0  ZB ENABLE = 1
```

```
976                                     ;;*****************************************************************
(4)                                     ;*TEST 66       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 2 GAIN = 0 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  013216  000004                     TST66:  SCOPE
(2)  013220  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  013226  012777  004000  166522             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  013234  012777  000014  166514             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  013242  012777  000024  166476             MOV     #24,@CSR                ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  013250  052777  000002  166500             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  013256  052777  000001  166462             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  013264  042777  000002  166464             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  013272  017737  166462  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  013300  012737  000000  001124             MOV     #0,$GDDAT               ;LOAD EXPECTED VALUE
(1)  013306  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  013314  001401                             BEQ     TST67           ;;BR IF SAME
(1)  013316  104020                              RROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 2  GAIN = 0  ZB ENABLE = 0
977                                     ;;*****************************************************************
(4)                                     ;*TEST 67       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 2 GAIN = 1 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  013320  000004                     TST67:  SCOPE
(2)  013322  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  013330  012777  004000  166420             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  013336  012777  000014  166412             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  013344  012777  002024  166374             MOV     #2024,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  013352  052777  000002  166376             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  013360  052777  000001  166360             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  013366  042777  000002  166362             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  013374  017737  166360  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  013402  012737  002552  001124             MOV     #2552,$GDDAT            ;LOAD EXPECTED VALUE
(1)  013410  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  013416  001401                             BEQ     TST70           ;;BR IF SAME
(1)  013420  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 2  GAIN = 1  ZB ENABLE = 0
978                                     ;;*****************************************************************
(4)                                     ;*TEST 70       TEST ADDRESS MAKER  -  MATRIX MODE - RES = 2 GAIN = 0 ZB ENABLE = 1
(4)                                     ;;*****************************************************************
(3)  013422  000004                     TST70:  SCOPE
(2)  013424  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  013432  012777  004000  166316             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  013440  012777  000014  166310             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  013446  012777  004024  166272             MOV     #4024,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  013454  052777  000002  166274             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  013462  052777  000001  166256             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  013470  042777  000002  166260             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  013476  017737  166256  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  013504  012737  005224  001124             MOV     #5224,$GDDAT            ;LOAD EXPECTED VALUE
(1)  013512  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  013520  001401                             BEQ     TST71           ;;BR IF SAME
(1)  013522  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1.                                     ;                       RESOLUTION = 2  GAIN = 0  ZB ENABLE =  1
```

```
980                                     ;;*****************************************************************
(4)                                     ;*TEST 71        TEST ADDRESS MAKER  -  MATRIX MODE - RES = 1 GAIN = 0 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  013524  000004                     TST71:  SCOPE
(2)  013526  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  013534  012777  004000  166214             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  013542  012777  000014  166206             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  013550  012777  000022  166170             MOV     #22,@CSR                ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  013556  052777  000002  166172             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  013564  052777  000001  166154             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  013572  042777  000002  166156             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  013600  017737  166154  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  013606  012737  000000  001124             MOV     #0,$GDDAT               ;LOAD EXPECTED VALUE
(1)  013614  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  013622  001401                             BEQ     TST72           ;;BR IF SAME
(1)  013624  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 1  GAIN = 0  ZB ENABLE = 0
981                                     ;;*****************************************************************
(4)                                     ;*TEST 72        TEST ADDRESS MAKER  -  MATRIX MODE - RES = 1 GAIN = 1 ZB ENABLE = 0
(4)                                     ;;*****************************************************************
(3)  013626  000004                     TST72:  SCOPE
(2)  013630  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  013636  012777  004000  166112             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  013644  012777  000014  166104             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  013652  012777  002022  166066             MOV     #2022,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VALUE
(1)  013660  052777  000002  166070             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  013666  052777  000001  166052             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  013674  042777  000002  166054             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  013702  017737  166052  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  013710  012737  001265  001124             MOV     #1265,$GDDAT            ;LOAD EXPECTED VALUE
(1)  013716  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  013724  001401                             BEQ     TST73           ;;BR IF SAME
(1)  013726  104020                             ERROR   20                      ;INCORRECT ADDRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 1  GAIN = 1  ZB ENABLE - 0
982                                     ;;*****************************************************************
(4)                                     ;*TEST 73        TEST ADDRESS MAKER  -  MATRIX MODE - RES = 1 GAIN - 0 ZB ENABLE = 1
(4)                                     ;;*****************************************************************
(3)  013730  000004                     TST73:  SCOPE
(2)  013732  012737  000010  001160             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
(1)  013740  012777  004000  166010             MOV     #CLRALL,@SFR            ;CLEAR DEVICE
(1)  013746  012777  000014  166002             MOV     #TSTDMA!TSTCON,@SFR     ;SET "TEST DMA AND CONTROL"
(1)  013754  012777  004022  165764             MOV     #4022,@CSR              ;LOAD RESOLUTION, GAIN, ZB ENABLE VA UE
(1)  013762  052777  000002  165766             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
(1)  013770  052777  000001  165750             BIS     #BIT0,@CSR              ;SET ENABLE NCV11
(1)  013776  042777  000002  165752             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
(1)  014004  017737  165750  001126             MOV     @ADM,$BDDAT             ;READ THE ADDRESS DATA MAKER
(1)  014012  012737  002512  001124             MOV     #2512,$GDDAT            ;LOAD EXPECTED VALUE
(1)  014020  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
(3)  014026  001401                             BEQ     TST74           ;;BR I  SAME
(1)  014030  104020                             ERROR   20                      ;INCORRECT A DRESS MAKER DATA
(1)                                     ;                       RESOLUTION = 1  GAIN = 0  ZB ENABLE   1
```

```
                                ;;*****************************************************************
                                ;*TEST 74        TEST ADDRESS MAKER  -  LIST MODE - ZB ENABLE = 0  GAIN = 0
                                ;;*****************************************************************
014032  000004                  TST74:  SCOPE
014034  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
014042  012777  004000  165706          MOV     #CLRALL,@SFR            ;CLEAR DEVICE
014050  012777  000014  165700          MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST "DMA AND CONTROL"
014056  012777  000000  165662          MOV     #0,@CSR                 ;ENSURE LIST MODE
014064  052777  000002  165664          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
014072  052777  000001  165646          BIS     #BIT0,@CSR              ;SET ENABLE NCV11
014100  042777  000002  165650          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
014106  017737  165646  001126          MOV     @ADM,$BDDAT             ;READ THE ADDRESS MAKER VALUE
014114  012737  003407  001124          MOV     #3407,$GDDAT            ;LOAD EXPECTED VALUE
014122  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
014130  001401                        BEQ     TST75           ;;BR IF SAME
014132  104021                        ERROR   21                      ;INCORRECT ADDRESS MAKER DATA
                                                                ;RESOLUTION 7 <DEFAULT WHEN ZB IS NOT ENABLED>

                                ;;*****************************************************************
                                ;*TEST 75        TEST ADDRESS MAKER  -  LIST MODE - ZB ENABLE = 1  GAIN = 0
                                ;;*****************************************************************
014134  000004                  TST75:  SCOPE
014136  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
014144  012777  004000  165604          MOV     #CLRALL,@SFR            ;CLEAR DEVICE
014152  012777  000014  165576          MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST "DMA AND CONTROL"
014160  012777  004000  165560          MOV     #4000,@CSR              ;ENSURE LIST MODE AND ZB ENABLED
014166  052777  000002  165562          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
014174  052777  000001  165544          BIS     #BIT0,@CSR              ;SET ENABLE NCV11
014202  042777  000002  165546          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
014210  017737  165544  001126          MOV     @ADM,$BDDAT             ;READ THE ADDRESS MAKER VALUE
014216  012737  052452  001124          MOV     #52452,$GDDAT           ;LOAD EXPECTED VALUE
014224  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED TO READ
014232  001401                        BEQ     TST76           ;;BR IF SAME
014234  104021                        ERROR   21                      ;INCORRECT ADDRESS MAKER DATA
                                                                ;RESOLUTION 5 <DEFAULT WHEN ZB IS ENABLED>


                                ;;*****************************************************************
                                ;*TEST 76        TEST ADDRESS MAKER  -  LIST MODE - ZB ENABLE = 0  GAIN = 1
                                ;;*****************************************************************
014236  000004                  TST76:  SCOPE
014240  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
014246  012777  004000  165502          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
014254  012777  000014  165474          MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST "DMA AND CONTROL"
014262  012777  002000  165456          MOV     #2000,@CSR              ;SET GAIN FLOP
014270  052777  000002  165460          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
014276  052777  000001  165442          BIS     #BIT0,@CSR              ;ENABLE THE NCV11
014304  042777  000002  165444          BIC     #TESTZ,@SFR             ;DISABLE THE "TEST Z" PULSES
014312  017737  165442  001126          MOV     @ADM,$BDDAT             ;READ THE ADDRESS MAKER VALUE
014320  012737  127657  001124          MOV     #127657,$GDDAT          ;LOAD THE EXPECTED
014326  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
014334  001401                        BEQ     TST77           ;;BR IF SAME
014336  104021                        ERROR   21                      ;INCORRECT ADDRESS MAKER DATA
                                                                ;RESOLUTION 7 -  GAIN FLOP SET
```

```
1029                                    ;;*****************************************************************
 (3)                                    ;*TEST 77       ENABLE A ONE WORD TRANSFER      SECTION LIST MODE
 (3)                                    ;;*****************************************************************
 (2)  014340  000004                    TST77:  SCOPE
 (1)  014342  012737  000040  001160            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1030  014350  012777  004000  165400            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1031  014356  012737  125252  060000            MOV     #125252,BUF0            ;PRIME TARGET BUFFER
1032  014364  012777  011110  165356            MOV     #11110,@OFF             ;LOAD THE OFFSET VALUE WITH A NUMBER
1033  014372  012777  177777  165352            MOV     #-1,@WCR                ;SET UP 1 WORD TRANSFER
1034  014400  012777  060000  165346            MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS FOR RESULT
1035  014406  012777  000014  165342            MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL AND DMA FLOPS
1036  014414  052777  000001  165324            BIS     #BIT0,@CSR              ;ENABLE DEVICE
1037  014422  052777  000002  165326            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  014430  042777  000002  165320            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1038  014436  000240                            NOP
1039  014440  000240                            NOP
1040  014442  000240                            NOP
1041  014444  052777  010000  165304            BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
1042  014452  000240                            NOP
1043  014454  000240                            NOP
1044  014456  000240                            NOP
1045  014460  017737  165262  001126            MOV     @CSR,$BDDAT             ;READ STATUS
1046  014466  012777  004000  165262            MOV     #CLRALL,@SFR            ;RESET THE DEVICE
1047  014474  012737  040200  001124            MOV     #40200,$GDDAT           ;LOAD EXPECTED
1048  014502  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;TEST STATUS
1049  014510  001402                            BEQ     4$              ;;BR IF EXPECTED
1050  014512  104036                            ERROR   36                      ;UNEXPECTED STATUS AFTER A 1 WORD TRANSFER
1051  014514  000453                            BR      TST100          ;;
1052  014516  017737  165232  001126  4$:       MOV     @BAR,$BDDAT             ;READ BUS ADDRESS
1053  014524  012737  060002  001124            MOV     #BUF0+2,$GDDAT          ;LOAD EXPECTED
1054  014532  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
1055  014540  001401                            BEQ     1$              ;;BR IF SAME
1056  014542  104022                            ERROR   22                      ;INCORRECT BUS ADDRESS VALUE
1057                                            ;AFTER A 1 WORD TRANSFER
1058
1059  014544  017737  165202  001126  1$:       MOV     @WCR,$BDDAT             ;READ W.C. REGISTER
1060  014552  012737  000000  001124            MOV     #0,$GDDAT               ;LOAD EXPECTED VALUE
1061  014560  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
1062  014566  001401                            BEQ     2$              ;;BR IF SAME
1063  014570  104023                            ERROR   23                      ;INCORRECT WORD COUNT REGISTER VALUE
1064                                            ;AFTER A 1 WORD TRANSFER
1065  014572  005037  001124          2$:       CLR     $GDDAT                  ;CLEAR THE EXPECTED VALUE
1066  014576  017737  165146  001126            MOV     @OFF,$BDDAT             ;READ THE OFFSET REGISTER
1067  014604  001401                            BEQ     3$              ;;BR IF CLEARED
1068  014606  104024                            ERROR   24                      ;OFFSET REG. FAILED TO CLEAR AFTER
1069                                            ;1 LIST MODE XFR.
1070  014610  013737  060000  001126  3$:       MOV     BUF0,$BDDAT             ;GET BUFFER DATA
1071  014616  012737  003407  001124            MOV     #3407,$GDDAT            ;LOAD EXPECTED
1072  014624  012737  060000  001122            MOV     #BUF0,$BDADR            ;LOAD BAD ADDRESS
1073  014632  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE DATA
1074  014640  001401                            BEQ     TST100          ;;BR IF SAME
1075  014642  104037                            ERROR   37                      ;STATUS WAS OK BUT DATA WAS INCORRECT
```

```
1077                                    ;;*****************************************************************
 (3)                                    ;*TEST 100      ENABLE A 512 WORD TRANSFER      SECTION LIST MODE
 (3)                                    ;;*****************************************************************
 (2)  014644  000004                    TST100: SCOPE
 (1)  014646  012737  000040  001160            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1078  014654  012777  004000  165074            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1079  014662  012700  060000                    MOV     #BUF0,R0                ;LOAD BUFFER POINTER
1080  014666  012720  125252            1$:     MOV     #125252,(R0)+           ;PRESET THE BUFFER WITH DATA
1081  014672  020027  062000                    CMP     R0,#BUF1                ;TEST IF DONE
1082  014676  001373                            BNE     1$                      ;BR IF NOT
1083  014700  012777  177000  165044            MOV     #-512.,@WCR             ;SET UP 512. WORD TRANSFER
1084  014706  012777  060000  165040            MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS FOR RESULT
1085  014714  012777  000014  165034            MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL AND DMA FLOPS
1086  014722  052777  000001  165016            BIS     #BIT0,@CSR              ;ENABLE DEVICE
1087  014730  012737  001000  002004            MOV     #512.,$TEMP             ;LOAD THE COUNTER
1088  014736                            2$:
 (1)  014736  052777  000002  165012            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  014744  042777  000002  165004            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1089  014752  052777  010000  164776            BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
1090  014760  005337  002004                    DEC     $TEMP                   ;FINISHED ALL WORDS?
1091  014764  001364                            BNE     2$                      ;BR UNTILL DONE
1092                                    ;THE TRANSFER IS NOW COMPLETE
1093  014766  017737  164754  001126            MOV     @CSR,$BDDAT             ;READ STATUS
1094  014774  012737  040200  001124            MOV     #40200,$GDDAT           ;LOAD EXPECTED STATUS
1095  015002  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE DATA
1096  015010  001402                            BEQ     3$              ;;BR IF EXPECTED STATUS
1097  015012  104036                            ERROR   36                      ;UNEXPECTED STATUS AFTER 512 WORD TRANSFER
1098  015014  000465                            BR      TST101          ;;
1099  015016  005037  001124            3$:     CLR     $GDDAT                  ;CLEAR EXPECTED
1100  015022  017737  164722  001126            MOV     @OFF,$BDDAT             ;READ OFFSET REG.
1101  015030  001401                            BEQ     4$              ;;BR IF CLEARED
1102  015032  104006                            ERROR   6               ;UNEXPECTED OFFSET REGISTER BIT SET
1103  015034  012777  004000  164714  4$:       MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1104  015042  017737  164706  001126            MOV     @BAR,$BDDAT             ;READ BUS ADDRESS
1105  015050  012737  062000  001124            MOV     #BUF1,$GDDAT            ;LOAD EXPECTED BAR VALUE
1106  015056  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
1107  015064  001401                            BEQ     5$              ;;BR IF SAME
1108  015066  104022                            ERROR   22      ;INCORRECT BUS ADDRESS VALUE AFTER ' ' WORD TRANSFER
1109
1110  015070  017737  164656  001126  5$:       MOV     @WCR,$BDDAT             ;READ W.C. REGI_
1111  015076  012737  000000  001124            MOV     #0,$GDDAT               ;LOAD EXPECTED W.C. VALUE
1112  015104  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
1113  015112  001401                            BEQ     6$              ;;BR IF SAME
1114  015114  104023                            ERROR   23      ;INCORRECT WORD COUNT REGISTER VALUE AFTER A 1 WORD TRANSFER
1115  015116  012737  003407  001124  6$:       MOV     #3407,$GDDAT            ;LOAD EXPECTED DATA
1116  015124  012737  060000  001122            MOV     #BUF0,$BDADR            ;LOAD STARTING ADDRESS
1117  015132  017737  163764  001126  7$:       MOV     @$BDADR,$BDDAT          ;READ DATA WORD
1118  015140  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE DATA
1119  015146  001401                            BEQ     10$             ;;BR IF EXPECTED
1120  015150  104037                            ERROR   37                      ;INCORRECT DATA IN LIST MODE XFER.
1121  015152  062737  000002  001122  10$:      ADD     #2,$BDADR               ;UPDATE POINTER
1122  015160  022737  062000  001122            CMP     #BUF0+1024.,$BDADR      ;TEST IF END OF BUFFER
1123  015166  001361                            BNE     7$              ;;BR IF NOT DONE
```

```
1125                                    ;;*****************************************************************
 (3)                                    ;*TEST 101      VERIFY "TIMEOUT" FLOP SETS AND "CLR ALL" CLEARS IT
 (3)                                    ;;*****************************************************************
 (2) 015170 000004                      TST101: SCOPE
 (1) 015172 012737 000040 001160                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1126 015200 012777 004000 164550                MOV     #CLRALL,@SFR            ;CLEAR DEVICE
1127 015206 012777 177777 164536                MOV     #-1,@WCR                ;LOAD W.C. REGISTER
1128 015214 012777 000003 164526                MOV     #3,@OFF                 ;LOAD EXTENDED ADDRESS BITS
1129 015222 012777 160000 164524                MOV     #160000,@BAR            ;LOAD BUS ADDRESS REGISTER TO A NON-EXIENT ADDRE
1130 015230 012777 000014 164520                MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROL AND DMA
1131 015236 012777 000016 164502                MOV     #16,@CSR                ;LOAD RESOLUTION TO VERIFY "SFR INIT" CLEARS
1132 015244 052777 000001 164474                BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1133 015252 052777 000002 164476                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 015260 042777 000002 164470                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1134 015266 000240                              NOP
1135 015270 000240                              NOP
1136 015272 000240                              NOP
1137 015274 052777 010000 164454                BIS     #BIT12,@SFR             ;ALLOW  1 DMA TRANSFER
1138 015302 000240                              NOP
1139 015304 000240                              NOP
1140 015306 000240                              NOP
1141 015310 012737 140200 001124                MOV     #BIT15!BIT14!BIT7,$GDDAT        ;LOAD EXPECTED
1142 015316 017737 164424 001126                MOV     @CSR,$BDDAT             ;READ STATUS REG.
1143 015324 100402                              BMI     1$                      ;BR IF "TIMEOUT" FLOP IS SET
1144 015326 104025                              ERROR   25                      ;"TIMEOUT" FLOP FAILED TO SET
1145 015330 000423                              BR      TST102          ;;
1146 015332 023737 001124 001126 1$:            CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
1147 015340 001401                              BEQ     2$              ;;BR IF SAME
1148 015342 104025                              ERROR   25                      ;"TIMEOUT" FLOP DID SET BUT FAILED TO GENERATE "
1149 015344 052777 004000 164404 2$:            BIS     #CLRALL,@SFR            ;GENERATE AN "CLR ALL" TO CLEAR TIMEOUT FLOP
1150 015352 012737 000200 001124                MOV     #BIT7,$GDDAT            ;LOAD EXPECTED
1151 015360 017737 164362 001126                MOV     @CSR,$BDDAT             ;READ STATUS
1152 015366 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE VALUE
1153 015374 001401                              BEQ     TST102          ;;BR IF SAME
1154 015376 104025                              ERROR   25              ;"CLR ALL" FAILED TO CLEAR TIMEOUT FLOP
1155                                    ;;*****************************************************************
 (3)                                    ;*TEST 102      VERIFY "TIMEOUT" FLOP SETS AND "CLR TIMEOUT" CLEARS IT
 (3)                                    ;;*****************************************************************
 (2) 015400 000004                      TST102: SCOPE
 (1) 015402 012737 000040 001160                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1156 015410 012777 004000 164340                MOV     #CLRALL,@SFR            ;CLEAR DEVICE
1157 015416 012777 177777 164326                MOV     #-1,@WCR                ;LOAD W.C. REGISTER
1158 015424 012777 000003 164316                MOV     #3,@OFF                 ;LOAD EXTENDED ADDRESS BITS
1159 015432 012777 160000 164314                MOV     #160000,@BAR            ;LOAD BUS ADDRESS REGISTER TO A NON-EXIENT ADDRE
1160 015440 012777 000014 164310                MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROL AND DMA
1161 015446 052777 000001 164272                BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1162 015454 052777 000002 164274                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 015462 042777 000002 164266                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1163 015470 000240                              NOP
1164 015472 000240                              NOP
1165 015474 000240                              NOP
1166 015476 052777 010000 164252                BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
1167 015504 000240                              NOP
1168 015506 000240                              NOP
1169 015510 000240                              NOP
1170 015512 012737 140200 001124                MOV     #BIT15!BIT14!BIT7,$GDDAT        ;LOAD EXPECTED
```

```
 1171  015520  017737  164222  001126          MOV     @CSR,$BDDAT             ;READ STATUS REG.
 1172  015526  100402                          BMI     1$                      ;BR IF "TIMEOUT" FLOP IS SET
 1173  015530  104025                          ERROR   25                      ;"TIMEOUT" FLOP FAILED TO SET
 1174  015532  000423                          BR      TST103          ;;
 1175  015534  023737  001124  001126  1$:     CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
 1176  015542  001401                          BEQ     2$                      ;;BR IF SAME
 1177  015544  104025                          ERROR   25                      ;"TIMEOUT" FLOP DID SET BUT FAILED TO GENERATE "
 1178  015546  052777  100000  164202  2$:     BIS     #BIT15,@SFR             ;GENERATE AN "CLR TIMEOUT" TO CLEAR TIMEOUT FLOP
 1179  015554  012737  040200  001124          MOV     #BIT14!BIT7,$GDDAT      ;LOAD EXPECTED
 1180  015562  017737  164160  001126          MOV     @CSR,$BDDAT             ;READ STATUS
 1181  015570  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUE
 1182  015576  001401                          BEQ     TST103          ;;BR IF SAME
 1183  015600  104025                          ERROR   25                      ;"CLR TIMEOUT" FAILED TO CLEAR TIMEOUT FLOP
 1184                                  ;;*****************************************************************
  (3)                                  ;*TEST 103      VERIFY "TIMEOUT" INTERRUPT
  (3)                                  ;;*****************************************************************
  (2)  015602  000004                  TST103: SCOPE
  (1)  015604  012737  000040  001160          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 1185  015612  012777  004000  164136          MOV     #CLRALL,@SFR            ;CLEAR DEVICE
 1186  015620  012777  177777  164124          MOV     #-1,@WCR                ;LOAD W.C. REGISTER
 1187  015626  012777  000003  164114          MOV     #3,@OFF                 ;LOAD EXTENDED ADDRESS BITS
 1188  015634  012777  160000  164112          MOV     #160000,@BAR            ;LOAD BUS ADDRESS REGISTER TO A NON-EXIENT ADDRE
 1189  015642  012777  000014  164106          MOV     #TSTCON!TSTDMA,@SFR     ;SET TEST CONTROL AND DMA
 1190  015650  012746  000000                  MOV     #0,-(SP)
 1191  015654  012746  015662                  MOV     #1$,-(SP)
 1192  015660  000002                          RTI
 1193  015662  012777  015754  164102  1$:     MOV     #2$,@VECTA0
 1194  015670  000240                          NOP
 1195  015672  000240                          NOP
 1196  015674  000240                          NOP
 1197  015676  000240                          NOP
 1198  015700  052777  000101  164040          BIS     #BIT6!BIT0,@CSR         ;ENABLE THE DEVICE
 1199  015706  052777  000002  164042          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
  (1)  015714  042777  000002  164034          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 1200  015722  000240                          NOP
 1201  015724  000240                          NOP
 1202  015726  000240                          NOP
 1203  015730  052777  010000  164020          BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSGFER
 1204  015736  000240                          NOP
 1205  015740  000240                          NOP
 1206  015742  000240                          NOP
 1207  015744  005077  163776                  CLR     @CSR                    ;CLEAR ENABLE
 1208  015750  104026                          ERROR   26                      ;"TIMEOUT" FAILED TO INTERRUPT
 1209  015752  000401                          BR      3$              ;;BR TO CLEAN UP
 1210
 1211  015754  022626                  2$:     CMP     (SP)+,(SP)+             ;CLEAN THE STACK
 1212  015756  005077  163764          3$:     CLR     @CSR
 1213  015762  012777  004000  163766          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 1214  015770  013777  001774  163774          MOV     VECTA1,@VECTA0          ;RESET VECTOR
 1215  015776  005077  163772                  CLR     @VECTA1
```

```
1217                                    ;;*****************************************************************
 (3)                                    ;*TEST 104      VERIFY INCREMENTING INTO EXTENDED ADDRESS BIT 16
 (3)                                    ;;*****************************************************************
 (2)  016002  000004                    TST104: SCOPE
 (1)  016004  012737  000040  001160            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1218  016012  012777  004000  163736            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1219  016020  023727  036406  002140            CMP     $LSTBK,#2140            ;TEST IF ENOUGH MEMORY
1220  016026  103450                            BLO     TST105          ;;BR IF NOT ENOUGH MEMORY
1221  016030  012777  177777  163714            MOV     #-1,@WCR                ;LOAD 1 WORD XFR
1222  016036  012777  177776  163710            MOV     #177776,@BAR            ;LOAD LAST ADDRESS
1223  016044  005077  163700                    CLR     @OFF                    ;ENSURE CLEARED EXTENDED ADDRESS BITS
1224  016050  012777  000014  163700            MOV     #TSTCON!TSTDMA,@SFR     ;ENABLE CONTROLER
1225  016056  012777  000016  163662            MOV     #16,@CSR                ;ENABLE THE MODE
1226  016064  052777  000001  163654            BIS     #BIT0,@CSR              ;ENABLE THE NCV11
1227  016072  052777  000002  163656            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  016100  042777  000002  163650            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1228  016106  052777  010000  163642            BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
1229  016114  000240                            NOP
1230  016116  000240                            NOP
1231  016120  000240                            NOP
1232  016122  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED VALUE
1233  016130  017737  163614  001126            MOV     @OFF,$BDDAT             ;READ ACUTAL VALUE
1234  016136  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
1235  016144  001401                            BEQ     TST105          ;;BR IF SAME
1236  016146  104006                            ERROR   6               ;EXTENDED ADDRESS BIT 16 FAILED TO SET
1237
1238                                    ;;*****************************************************************
 (3)                                    ;*TEST 105      VERIFY INCREMENTING INTO EXTENDED ADDRESS BIT 17
 (3)                                    ;;*****************************************************************
 (2)  016150  000004                    TST105: SCOPE
 (1)  016152  012737  000040  001160            MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1239  016160  012777  004000  163570            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1240  016166  023727  036406  006140            CMP     $LSTBK,#6140            ;TEST MEMORY SPACE >100K
1241  016174  103454                            BLO     TST106          ;;BR IF NOT ENOUGH MEMORY
1242  016176  012777  177777  163546            MOV     #-1,@WCR                ;LOAD 1 WORD XFR
1243  016204  012777  177776  163542            MOV     #177776,@BAR            ;LOAD LAST ADDRESS
1244  016212  012777  000001  163530            MOV     #1,@OFF                 ;LOAD EXTENDED ADDRESS BIT
1245  016220  012777  000014  163530            MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL
1246  016226  012777  000016  163512            MOV     #16,@CSR                ;ENABLE THE MODE
1247  016234  052777  000001  163504            BIS     #BIT0,@CSR              ;ENAVLE THE DEVICE
1248  016242  052777  000002  163506            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  016250  042777  000002  163500            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1249  016256  000240                            NOP
1250  016260  000240                            NOP
1251  016262  000240                            NOP
1252  016264  052777  010000  163464            BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
1253  016272  000240                            NOP
1254  016274  000240                            NOP
1255  016276  000240                            NOP
1256  016300  012737  000002  001124            MOV     #2,$GDDAT               ;LOAD EXPECTED
1257  016306  017737  163436  001126            MOV     @OFF,$BDDAT             ;READ ACTUAL
1258  016314  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
1259  016322  001401                            BEQ     TST106          ;;BR IF SAME
1260  016324  104006                            ERROR   6               ;EXTENDED ADDRESS BIT 17 FAILED TO SET
```

```
                                        ;;*****************************************************************
                                        ;*TEST 106       VERIFY "SET EVENT" DATA GENERATES A 177777 DATA WORD
                                        ;;*****************************************************************
                                TST106: SCOPE
016326  000004                          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
016330  012737  000040  001160          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
016336  012777  004000  163412          MOV     #-1,@WCR                ;SET UP 1 WORD TRANSFER
016344  012777  177777  163400          MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS FOR RESULT
016352  012777  060000  163374          MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL AND DMA FLOPS
016360  012777  000014  163370          MOV     #7070,BUF0              ;PRESET DATA
016366  012737  007070  060000          BIS     #BIT0,@CSR              ;ENABLE DEVICE
016374  052777  000001  163344          NOP
016402  000240                          NOP
016404  000240                          NOP
016406  000240                          BIS     #BIT10,@SFR             ;SET "EVENT" FLOP
016410  052777  002000  163340          BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
016416  052777  010000  163332          NOP
016424  000240                          NOP
016426  000240                          NOP
016430  000240                          MOV     BUF0,$BDDAT             ;READ DATA
016432  013737  060000  001126          MOV     #-1,$GDDAT              ;LOAD EXPECTED
016440  012737  177777  001124          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
016446  023737  001124  001126          BEQ     TST107          ;;BR IF SAME
016454  001401                          ERROR   27                      ;INCORRECT DATA VALUE FOR "EVENT" MARK
016456  104027                                                  ;AFTER A 1 WORD TRANSFER
                                        ;;*****************************************************************
                                        ;*TEST 107       VERIFY "SET TIME" DATA GENERATES A 000000 DATA WORD
                                        ;;*****************************************************************
                                TST107: SCOPE
016460  000004                          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
016462  012737  000040  001160          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
016470  012777  004000  163260          MOV     #-1,@WCR                ;SET UP 1 WORD TRANSFER
016476  012777  177777  163246          MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS FOR RESULT
016504  012777  060000  163242          MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL AND DMA FLOPS
016512  012777  000014  163236          MOV     #7070,BUF0              ;PRESET THE DATA
016520  012737  007070  060000          BIS     #BIT0,@CSR              ;ENABLE DEVICE
016526  052777  000001  163212          NOP
016534  000240                          NOP
016536  000240                          NOP
016540  000240                          BIS     #BIT9,@SFR              ;SET "TIME" FLOP
016542  052777  001000  163206          BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
016550  052777  010000  163200          NOP
016556  000240                          NOP
016560  000240                          NOP
016562  000240                          MOV     BUF0,$BDDAT             ;READ DATA
016564  013737  060000  001126          MOV     #0,$GDDAT               ;LOAD EXPECTED
016572  012737  000000  001124          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
016600  023737  001124  001126          BEQ     TST110          ;;BR IF SAME
016606  001401                          ERROR   27                      ;INCORRECT DATA VALUE FOR "TIME" MARK
016610  104027                                                  ;AFTER A 1 WORD TRANSFER
```

```
1305                                    ;;*****************************************************************
 (3)                                    ;*TEST 110      VERIFY "CLOCK ST1" GENERATES A EVENT (177777) DATA WORD
 (3)                                    ;;*****************************************************************
 (2) 016612 000004                      TST110: SCOPE
 (1) 016614 012737 000040 001160               MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1306 016622 012777 004000 163126               MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1307 016630 005737 004020                      TST     DEADKW                  ;TEST IF NCV11 CLOCK IS PRESENT
1308 016634 001044                             BNE     TST111          ;;BR IF NOT
1309 016636 012777 177777 163106               MOV     #-1,@WCR                ;SET UP 1 WORD TRANSFER
1310 016644 012777 060000 163102               MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS FOR RESULT
1311 016652 012777 000014 163076               MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL AND DMA FLOPS
1312 016660 012737 007070 060000               MOV     #7070,BUF0              ;PRESET THE DATA
1313 016666 052777 000001 163052               BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1314 016674 000240                             NOP
1315 016676 000240                             NOP
1316 016700 052777 000400 163032               BIS     #BIT8,@KWCSR            ;GENERATE CLOCK ST1 TO SET "EVENT" FLOP
1317 016706 052777 010000 163042               BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
1318 016714 000240                             NOP
1319 016716 000240                             NOP
1320 016720 013737 060000 001126               MOV     BUF0,$BDDAT             ;READ BUFFER DATA
1321 016726 012737 177777 001124               MOV     #-1,$GDDAT              ;LOAD EXPECTED DATA
1322 016734 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE
1323 016742 001401                             BEQ     TST111          ;;BR IF EXPECTED
1324 016744 104040                             ERROR   40                      ;CLOCK ST1 FAILED TO GENERATE EVENT FLAG
1325                                                                           ;CHECK THE M8026 TO M7952 JUMPERS
1326                                    ;;*****************************************************************
 (3)                                    ;*TEST 111      VERIFY "CLOCK OVERFLOW" GENERATES A TIME (000000) DATA WORD
 (3)                                    ;;*****************************************************************
 (2) 016746 000004                      TST111: SCOPE
 (1) 016750 012737 000040 001160               MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1327 016756 012777 004000 162772               MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1328 016764 005737 004020                      TST     DEADKW                  ;TEST IF NCV11 CLOCK IS PRESENT
1329 016770 001052                             BNE     TST112          ;;BR IF NOT
1330 016772 012777 177777 162752               MOV     #-1,@WCR                ;SET UP 1 WORD TRNSFER
1331 017000 012777 060000 162746               MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS FOR RESULT
1332 017006 012777 000014 162742               MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL AND DMA FLOPS
1333 017014 012737 007070 060000               MOV     #7070,BUF0              ;PRESET THE DATA
1334 017022 052777 000001 162716               BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1335 017030 005077 162704                      CLR     @KWCSR                  ;CLEAR STATUS
1336 017034 012777 177776 162702               MOV     #-2,@KWPSR              ;LOAD COUNTER PRESET
1337 017042 012777 000011 162670               MOV     #11,@KWCSR              ;ENABLE 1 MHZ. RATE AND CLOCK GO
1338 017050 105777 162664              1$:     TSTB    @KWCSR                  ;WAIT FOR CLOCK
1339 017054 100375                             BPL     1$
1340 017056 052777 010000 162672               BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
1341 017064 000240                             NOP
1342 017066 000240                             NOP
1343 017070 013737 060000 001126               MOV     BUF0,$BDDAT             ;READ DATA
1344 017076 012737 000000 001124               MOV     #0,$GDDAT               ;LOAD EXPECTED
1345 017104 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
1346 017112 001401                             BEQ     TST112          ;;BR IF SAME
1347 017114 104040                             ERROR   40                      ;CLOCK OVERFLOW FAILED TO GENERATE "TIME MARKS"
1348                                                                           ;CHECK THE M8026 TO M7952 JUMPERS
```

```
                        ;;*****************************************************************
                        ;*TEST 112      VERIFY "SET TIME" OVERRIDES "SET EVENT" DATA
                        ;;*****************************************************************
017116  000004          TST112: SCOPE
017120  012737  000040  001160          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
017126  012777  004000  162622          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
017134  012777  177777  162610          MOV     #-1,@WCR                ;SET UP 1 WORD TRANSFER
017142  012777  060000  162604          MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS FOR RESULT
017150  012777  000014  162600          MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE TEST CONTROL AND DMA FLOPS
017156  012737  007070  060000          MOV     #7070,BUF0              ;PRESET THE DATA
017164  052777  000001  162554          BIS     #BIT0,@CSR              ;ENABLE DEVICE
017172  000240                          NOP
017174  000240                          NOP
017176  000240                          NOP
017200  052777  003000  162550          BIS     #BIT10!BIT9,@SFR        ;SET "TIME AND EVENT" FLOPS
017206  052777  010000  162542          BIS     #BIT12,@SFR             ;ALLOW 1 DMA TRANSFER
017214  000240                          NOP
017216  000240                          NOP
017220  000240                          NOP
017222  013737  060000  001126          MOV     BUF0,$BDDAT             ;READ DATA
017230  012737  000000  001124          MOV     #0,$GDDAT               ;LOAD EXPECTED
017236  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
017244  001401                          BEQ     TST113          ;;BR IF SAME
017246  104027                          ERROR   27                      ;"TIME" MARK FAILED TO OVERIDE "EVENT" OR DATA M
                        ;*****************************************************************
                        ;*TEST 113      DO A ONE WORD MATRIX MODE TRANSFER      -CHECK FOR INCREMENT FUNCTION
                        ;*****************************************************************
017250  000004          TST113: SCOPE
017252  012737  000040  001160          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
017260  012777  004000  162470          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
017266  005037  060000                  CLR     BUF0                    ;CLEAR INITIAL INREMENT LOCATION
017272  012777  060000  162450          MOV     #BUF0,@OFF              ;LOAD INITIAL OFFSET REGISTER
017300  012777  000022  162440          MOV     #BIT4!BIT1,@CSR         ;ENABLE MATRIX MODE
017306  012777  000014  162442          MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
017314  052777  000001  162424          BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
017322  052777  000002  162426          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
017330  042777  000002  162420          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
017336  000240                          NOP
017340  000240                          NOP
017342  000240                          NOP
017344  052777  010000  162404          BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
017352  000240                          NOP
017354  000240                          NOP
017356  000240                          NOP
017360  013737  060000  001126          MOV     BUF0,$BDDAT             ;READ THE BUF0 LOCATION
017366  012737  000001  001124          MOV     #1,$GDDAT               ;LOAD E   CTED VALUE
017374  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
017402  001401                          BEQ     TST114          ;;BR IF SAME
017404  104030                          ERROR   30                      ;INCORRECT DATA IN MATRIX MODE
                                                        ;IF DATA WAS 0, THE ADDRESS ACCESSED WAS PROBLY WRONG
                                                        ;IF DATA WAS NON ZERO, THE "INCR" REGISTER IS STUCK TO A 1
```

```
1393                                    ;;*****************************************************************
 (3)                                    ;*TEST 114      VERIFY EACH BIT OF THE INCREMENT REG. DATA PATH
 (3)                                    ;;*****************************************************************
 (2) 017406  000004                     TST114: SCOPE
 (1) 017410  012737  000040  001160             MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1394                                    ;CHECK FOR A "CARRY" INTO EACH BIT
1395                                    ;       IE:     000002-000001
1396                                    ;               000004-000003
1397                                    ;               000010-000007
1398                                    ;               000020-000017
1399                                    ;               000040-000037
1400                                    ;               000100-000077
1401                                    ;               000200-000177
1402                                    ;               ETC
1403 017416  012737  017432  001110             MOV     #1$,$LPERR              ;LOAD RETURN ADDRESS IF ERROR
1404 017424  012737  000002  002004             MOV     #2,$TEMP                ;LOAD INITIAL VALUE
1405
1406 017432  012777  004000  162316  1$:        MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1407 017440  012777  060000  162302             MOV     #BUF0,@OFF              ;LOAD INITIAL OFFSET REGISTER
1408 017446  012777  000024  162272             MOV     #BIT4!BIT2,@CSR         ;ENABLE WORD MATRIX MODE
1409 017454  012777  000014  162274             MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
1410 017462  052777  000001  162256             BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1411 017470  013737  002004  060000             MOV     $TEMP,BUF0              ;LOAD PRESET VALUE
1412 017476  005337  060000                     DEC     BUF0                    ;TO EXPECTED -1
1413 017502  052777  000002  162246             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 017510  042777  000002  162240             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1414 017516  052777  010000  162232             BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
1415 017524  013737  002004  001124             MOV     $TEMP,$GDDAT            ;LOAD TYPEOUT EXPECTED
1416 017532  013737  060000  001126             MOV     BUF0,$BDDAT             ;READ THE BUF0 LOCATION
1417 017540  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
1418 017546  001401                             BEQ     2$              ;;BR IF SAME
1419 017550  104030                             ERROR   30                      ;INCORRECT DATA IN THE INCREMENT REGISTER
1420 017552  017737  162170  001126  2$:        MOV     @CSR,$BDDAT             ;READ STATUS
1421 017560  032737  020000  001126             BIT     #BIT13,$BDDAT           ;TEST FOR UNEXPECTED CELL OVERFLOW
1422 017566  001407                             BEQ     3$                      ;BR IF NOT
1423 017570  012737  000024  001124             MOV     #BIT4!BIT2,$GDDAT       ;LOAD THE EXPECTED TYPEOUT
1424 017576  104031                             ERROR   31                      ;UNEXPECTED "CELL OVERFLOW"  STATUS
1425 017600  052777  020000  162150             BIS     #BIT13,@SFR             ;TRY TO CLEAR THE FLAG
1426 017606  006337  002004          3$:        ASL     $TEMP                   ;CHANGE THE EXPCTED
1427 017612  001307                             BNE     1$                      ;BR IF MORE DATA BITS
```

```
                                        ;;*****************************************************************
                                        ;*TEST 115      CHECK FOR LOW BYTE "INC OVFL" TO SET CELL OVERFLOW AND "CLR CELL" TO CLE
                                        ;;*****************************************************************
                                        TST115: SCOPE
017614  000004                                  
017616  012737  000040  001160                  MOV     #40,$TIMES      ;;DO 40 ITERATIONS
017624  012777  004000  162124                  MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
017632  005077  162114                          CLR     @WCR                    ;CLEAR WC
017636  005077  162112                          CLR     @BAR                    ;CLEAR BAR
017642  012777  060000  162100                  MOV     #BUF0,@OFF              ;LOAD INITIAL OFFSET REGISTER
017650  012777  000022  162070                  MOV     #BIT4!BIT1,@CSR         ;ENABLE BYTE MATRIX MODE
017656  012777  000014  162072                  MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
017664  052777  000001  162054                  BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
017672  052777  000002  162056                  BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
017700  042777  000002  162050                  BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
017706  000240                                  NOP
017710  000240                                  NOP
017712  000240                                  NOP
017714  112737  000377  060000                  MOVB    #377,BUF0               ;SET LOW BYTE OF BUF0 LOC. TO BYTE -1
017722  112737  000200  060001                  MOVB    #200,BUF0+1             ;SET HIGH BYTE OF BUF0 TO KNOWN VALUE
017730  052777  010000  162020                  BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
017736  000240                                  NOP
017740  000240                                  NOP
017742  000240                                  NOP
017744  017737  161776  001126                  MOV     @CSR,$BDDAT             ;READ STATUS
017752  012737  020022  001124                  MOV     #BIT13!BIT4!BIT1,$GDDAT ;LOAD EXPECTED STATUS
017760  023737  001124  001126                  CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
017766  001401                                  BEQ     1$              ;;BR IF SAME
017770  104031                                  ERROR   31                      ;"CELL OVERFLOW" FLOP FAILED TO SET
                                                                        ;IN BYTE MODE FROM A LOW BYTE OVERFLOW
                                        
                                        
                                        ;NOW GENERATE "CLR CELL" TO CLEAR CELL OVERFLOW FLOP
017772  052777  020000  161756          1$:     BIS     #BIT13,@SFR             ;GENERATE "CLR CELL"
020000  017737  161742  001126                  MOV     @CSR,$BDDAT             ;READ STATUS
020006  012737  000022  001124                  MOV     #BIT4!BIT1,$GDDAT       ;LOAD EXPECTED
020014  023737  001124  001126                  CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
020022  001401                                  BEQ     2$              ;;BR IF SAME
020024  104031                                  ERROR   31                      ;"CLR CELL" FAILED TO CLEAR "CELL OVFL" FLOP
                                        
                                        ;NOW VERIFY THE BYTE DATA CELL
020026  013737  060000  001126          2$:     MOV     BUF0,$BDDAT             ;READ DATA
020034  012737  100377  001124                  MOV     #100377,$GDDAT          ;LOAD EXPECTED
020042  023737  001124  001126                  CMP     $GDDAT,$BDDAT           ;COMPARE
020050  001401                                  BEQ     TST116          ;;BR IF SAME
020052  104031                                  ERROR   31                      ;OVERFLOW FROM INC. REG. FAILED TO INHIBIT
                                                                        ;"INC CNT" FROM CHANGING THE DATA OR
                                                                        ;"CSR BYTE CELLS" FAILED TO INHIBIT "INC ENA HI"
```

```
                                        ;;*****************************************************************
                                        ;*TEST 116      CHECK FOR WORD "INC OVFL" TO SET CELL OVERFLOW AND "CLR ALL" TO CLEAR IT
                                        ;;*****************************************************************
020054  000004                  TST116: SCOPE
020056  012737  000040  001160          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
020064  012777  004000  161664          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
020072  005077  161656                  CLR     @BAR
020076  005077  161650                  CLR     @WCR
020102  012777  060000  161640          MOV     #BUF0,@OFF              ;LOAD INITIAL OFFSET REGISTER
020110  012777  000024  161630          MOV     #BIT4!BIT2,@CSR         ;ENABLE MATRIX MODE
020116  012777  000014  161632          MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
020124  052777  000001  161614          BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
020132  052777  000002  161616          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
020140  042777  000002  161610          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
020146  000240                          NOP
020150  000240                          NOP
020152  000240                          NOP
020154  012737  177777  060000          MOV     #-1,BUF0                ;SET BUF0 LOC. TO WORD -1
020162  052777  010000  161566          BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
020170  000240                          NOP
020172  000240                          NOP
020174  000240                          NOP
020176  017737  161544  001126          MOV     @CSR,$BDDAT             ;READ STATUS
020204  012737  020024  001124          MOV     #BIT13!BIT4!BIT2,$GDDAT ;LOAD EXPECTED STATUS
020212  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
020220  001401                          BEQ     1$              ;;BR IF SAME
020222  104031                          ERROR   31              ;"CELL OVERFLOW" FLOP FAILED TO SET
                                ;NOW GENERATE "CLR ALL" TO CLEAR CELL OVERFLOW FLOP
020224  052777  004000  161524  1$:     BIS     #CLRALL,@SFR            ;GENERATE "CLR ALL"
020232  017737  161510  001126          MOV     @CSR,$BDDAT             ;READ STATUS
020240  012737  000200  001124          MOV     #BIT7,$GDDAT            ;LOAD EXPECTED
020246  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
020254  001401                          BEQ     2$              ;;BR IF SAME
020256  104031                          ERROR   31              ;"CLR ALL" FAILED TO CLEAR "CELL OVFL" FLOP
                                ;NOW VERIFY THE WORD CELL DATA
020260  013737  060000  001126  2$:     MOV     BUF0,$BDDAT             ;READ DATA
020266  012737  177777  001124          MOV     #-1,$GDDAT              ;LOAD EXPECTED
020274  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
020302  001401                          BEQ     TST117          ;;BR IF SAME
020304  104031                          ERROR   31
```

```
1508                                    ;;*****************************************************************
 (3)                                    ;*TEST 117      CHECK FOR "CELL OVERFLOW" INTERRUPT
 (3)                                    ;;*****************************************************************
 (2) 020306 000004                      TST117: SCOPE
 (1) 020310 012737 000040 001160                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1509 020316 012777 004000 161432                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1510 020324 005077 161424                       CLR     @BAR
1511 020330 005077 161416                       CLR     @WCR
1512 020334 012777 060000 161406                MOV     #BUF0,@CSR              ;LOAD INITIAL OFFSET REGISTER
1513 020342 012777 000064 161376                MOV     #BIT5!BIT4!BIT2,@CSR    ;ENABLE INTR. AND MATRIX MODE
1514 020350 012777 000014 161400                MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
1515 020356 052777 000001 161362                BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1516 020364 052777 000002 161364                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 020372 042777 000002 161356                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1517 020400 000240                              NOP
1518 020402 000240                              NOP
1519 020404 000240                              NOP
1520 020406 012737 177777 060000                MOV     #-1,BUF0                ;SET BUF0 LOC. TO WORD -1
1521 020414 012746 000000                       MOV     #0,-(SP)
1522 020420 012746 020426                       MOV     #1$,-(SP)
1523 020424 000002                              RTI
1524 020426 012777 020454 161342 1$:            MOV     #2$,@VECTB0             ;LOAD RETURN VECTOR
1525 020434 052777 010000 161314                BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
1526 020442 000240                              NOP
1527 020444 000240                              NOP
1528 020446 000240                              NOP
1529 020450 104032                              ERROR   32                      ;"CELL OVERFLOW" FAILED TO CAUES AN INTERRUPT
1530 020452 000414                              BR      3$              ;;BR TO CLEAN UP
1531 020454 022626                      2$:     CMP     (SP)+,(SP)+
1532 020456 017737 161264 001126                MOV     @CSR,$BDDAT             ;READ STATUS
1533 020464 012737 020264 001124                MOV     #BIT13!BIT7!BIT5.BIT4'BIT2,$GDDAT       ;LOAD EXPECTED
1534 020472 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE VALUE
1535 020500 001401                              BEQ     3$              ;;BR IF "ACTIVE" CLEARED
1536 020502 104031                              ERROR   31                      ;"ACTIVE" FAILED TO CLEAR
1537 020504 012777 004000 161244 3$:            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1538 020512 013777 002000 161256                MOV     VECTB1,@VECTB0          ;RESET THE VECTORS
1539 020520 005077 161254                       CLR     @VECTB1
```

```
1541                                    ;;*****************************************************************
 (3)                                    ;*TEST 120      VERIFY CORRECT BR LEVEL THE NCV-11
 (3)                                    ;;*****************************************************************
 (2) 020524 000004                      TST120: SCOPE
 (1) 020526 012737 000040 001160                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1542 020534 013746 002002                       MOV     BRLEV,-(SP)     ;PUSH THE EXPECTED BR LEVEL ON STACK
1543 020540 012746 020546                       MOV     #1$,-(SP)       ;PUST THE RETURN ADDRESS ON STACK
1544 020544 000002                              RTI             ;FAKE AN INTERRUPT TO BE LSI-11 AND PDP-11 COMPATABLE
1545 020546 012777 004000 161202        1$:     MOV     #CLRALL,@SFR    ;CLEAR THE DEVICE
1546 020554 012777 020750 161214                MOV     #10$,@VECTB0    ;LOAD UNEXPECTED INTERRUPT RETURN
1547 020562 012777 000340 161210                MOV     #340,@VECTB1
1548 020570 005077 161160                       CLR     @BAR            ;INIT THE LOW Z COUNT
1549 020574 005077 161152                       CLR     @WCR            ;INIT THE HIGH Z COUNT
1550 020600 012777 060000 161142                MOV     #BUF0,@OFF      ;PRIME THE OFFSET REGISTER
1551 020606 012777 000064 161132                MOV     #64,@CSR        ;SELECT INTR. AND MATRIX MODE
1552 020614 012777 000014 161134                MOV     #TSTDMA!TSTCON,@SFR     ;MAINT. MODE
1553 020622 012737 177777 060000                MOV     #-1,BUF0        ;PRIME THE TARGET LOCATION
1554 020630 052777 000001 161110                BIS     #BIT0,@CSR      ;ENABLE THE NCV-11
1555 020636 052777 000002 161112                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 020644 042777 000002 161104                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1556 020652 000240                              NOP
1557 020654 000240                              NOP
1558 020656 000240                              NOP
1559 020660 052777 010000 161070                BIS     #BIT12,@SFR     ;ENABLE 1 TRANSFER
1560 020666 000240                              NOP
1561 020670 000240                              NOP
1562 020672 000240                              NOP
1563 020674 000240                              NOP
1564 020676 000240                              NOP
1565 020700 000240                              NOP
1566 020702 013700 002002                       MOV     BRLEV,R0        ;GET CURRENT BR LEVEL
1567 020706 162700 000040                       SUB     #40,R0          ;AND MAKE IT 1 LEVEL LOWER
1568 020712 005737 003356                       TST     NLSI11          ;TEST IF LSI-11 CPU
1569 020716 001401                              BEQ     2$              ;BR IF NOT
1570 020720 005000                              CLR     R0              ;LOAD BR LEVEL 0
1571 020722 012777 020756 161046        2$:     MOV     #20$,@VECTB0    ;RELOAD TO EXPECTED VECTOR
1572 020730 010046                              MOV     R0,-(SP)        ;PUSH ADJUSTED BR LEVEL
1573 020732 012746 020740                       MOV     #3$,-(SP)       ;PUSH RETURN ADDRESS
1574 020736 000002                              RTI                     ;LOWER CPU BR LEVEL
1575 020740 000240                      3$:     NOP
1576 020742 000240                              NOP
1577 020744 104042                              ERROR   42      ;NCV11 FAILED TO INTERRUPT - INCORRECT NCV11 BR LEVEL?
1578 020746 000404                              BR      21$             ;;BR TO CLEAN-UP
1579
1580                                    ;UNEXPECTED INTERRUPT WITH BR LEVEL INDICATED
1581 020750 022626                      10$:    CMP     (SP)+,(SP)+     ;CLEAN THE STACK
1582 020752 104042                              ERROR   42              ;NCV11 INTERRUPTED ON AN INCORRECT BR LEVEL
1583 020754 000401                              BR      21$             ;;BR TO CLEANUP
1584
1585                                    ;EXPECTED INTERRUPT DID OCCUR
1586 020756 022626                      20$:    CMP     (SP)+,(SP)+     ;CLEAN THE STACK
1587 020760 012777 004000 160770        21$:    MOV     #CLRALL,@SFR    ;CLEAN THE DEVICE
1588 020766 013777 002000 161002                MOV     VECTB1,@VECTB0  ;RESET THE VECTOR
1589 020774 005077 161000                       CLR     @VECTB1
```

```
1612                                    ;;*****************************************************************
 (4)                                    ;*TEST 121      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 13
 (4)                                    ;;*****************************************************************
 (3)  021000  000004                    TST121: SCOPE
 (2)  021002  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  021010  012777  004000  160740            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  021016  012777  060000  160724            MOV     #60000,@OFF             ;LOAD THE OFFSET REG.
 (1)  021024  012777  000014  160724            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  021032  012777  000024  160706            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  021040  052777  000001  160700            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  021046  052777  000002  160702            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  021054  042777  000002  160674            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  021062  005037  060000                    CLR     @#60000                 ;CLEAR THE TARGET LOCATION
 (1)  021066  052777  010000  160662            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  021074  012737  060000  001122            MOV     #60000,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  021102  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  021110  017737  160006  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  021116  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  021124  001401                            BEQ     TST122          ;;BR IF SAME
 (1)  021126  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 60000
1613                                    ;;*****************************************************************
 (4)                                    ;*TEST 122      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 12
 (4)                                    ;;*****************************************************************
 (3)  021130  000004                    TST122: SCOPE
 (2)  021132  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  021140  012777  004000  160610            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  021146  012777  070000  160574            MOV     #70000,@OFF             ;LOAD THE OFFSET REG.
 (1)  021154  012777  000014  160574            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  021162  012777  000024  160556            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  021170  052777  000001  160550            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  021176  052777  000002  160552            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  021204  042777  000002  160544            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  021212  005037  070000                    CLR     @#70000                 ;CLEAR THE TARGET LOCATION
 (1)  021216  052777  010000  160532            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  021224  012737  070000  001122            MOV     #70000,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  021232  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  021240  017737  157656  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  021246  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  021254  001401                            BEQ     TST123          ;;BR IF SAME
 (1)  021256  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 70000
```

```
1615                                    ;;*****************************************************************
 (4)                                    ;*TEST 123      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 11
 (4)                                    ;;*****************************************************************
 (3) 021260 000004                      TST123: SCOPE
 (2) 021262 012737 000100 001160                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1) 021270 012777 004000 160460                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1) 021276 012777 064000 160444                MOV     #64000,@OFF             ;LOAD THE OFFSET REG.
 (1) 021304 012777 000014 160444                MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1) 021312 012777 000024 160426                MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1) 021320 052777 000001 160420                BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2) 021326 052777 000002 160422                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2) 021334 042777 000002 160414                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1) 021342 005037 064000                       CLR     @#64000                 ;CLEAR THE TARGET LOCATION
 (1) 021346 052777 010000 160402                BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1) 021354 012737 064000 001122                MOV     #64000,$BDADR           ;LOAD EXPECTED ADDRESS
 (1) 021362 012737 000001 001124                MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1) 021370 017737 157526 001126                MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1) 021376 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE
 (3) 021404 001401                              BEQ     TST124          ;;BR IF SAME
 (1) 021406 104033                              ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 64000
1616                                    ;;*****************************************************************
 (4)                                    ;*TEST 124      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 10
 (4)                                    ;;*****************************************************************
 (3) 021410 000004                      TST124: SCOPE
 (2) 021412 012737 000100 001160                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1) 021420 012777 004000 160330                MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1) 021426 012777 062000 160314                MOV     #62000,@OFF             ;LOAD THE OFFSET REG.
 (1) 021434 012777 000014 160314                MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1) 021442 012777 000024 160276                MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1) 021450 052777 000001 160270                BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2) 021456 052777 000002 160272                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2) 021464 042777 000002 160264                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1) 021472 005037 062000                       CLR     @#62000                 ;CLEAR THE TARGET LOCATION
 (1) 021476 052777 010000 160252                BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1) 021504 012737 062000 001122                MOV     #62000,$BDADR           ;LOAD EXPECTED ADDRESS
 (1) 021512 012737 000001 001124                MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1) 021520 017737 157376 001126                MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1) 021526 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE
 (3) 021534 001401                              BEQ     TST125          ;;BR IF SAME
 (1) 021536 104033                              ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 62000
```

```
1618                                    ;;*****************************************************************
 (4)                                    ;*TEST 125      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 09
 (4)                                    ;;*****************************************************************
 (3)  021540  000004                    TST125: SCOPE
 (2)  021542  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  021550  012777  004000  160200            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  021556  012777  061000  160164            MOV     #61000,@OFF             ;LOAD THE OFFSET REG.
 (1)  021564  012777  000014  160164            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  021572  012777  000024  160146            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  021600  052777  000001  160140            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  021606  052777  000002  160142            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  021614  042777  000002  160134            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  021622  005037  061000                    CLR     @#61000                 ;CLEAR THE TARGET LOCATION
 (1)  021626  052777  010000  160122            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  021634  012737  061000  001122            MOV     #61000,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  021642  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  021650  017737  157246  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  021656  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  021664  001401                            BEQ     TST126          ;;BR IF SAME
 (1)  021666  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 61000
1619                                    ;;*****************************************************************
 (4)                                    ;*TEST 126      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 08
 (4)                                    ;;*****************************************************************
 (3)  021670  000004                    TST126: SCOPE
 (2)  021672  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  021700  012777  004000  160050            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  021706  012777  060400  160034            MOV     #60400,@OFF             ;LOAD THE OFFSET REG.
 (1)  021714  012777  000014  160034            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  021722  012777  000024  160016            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  021730  052777  000001  160010            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  021736  052777  000002  160012            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  021744  042777  000002  160004            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  021752  005037  060400                    CLR     @#60400                 ;CLEAR THE TARGET LOCATION
 (1)  021756  052777  010000  157772            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  021764  012737  060400  001122            MOV     #60400,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  021772  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  022000  017737  157116  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  022006  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  022014  001401                            BEQ     TST127          ;;BR IF SAME
 (1)  022016  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 60400
```

```
1621                                    ;;*****************************************************************
 (4)                                    ;*TEST 127      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 07
 (4)                                    ;;*****************************************************************
 (3)  022020  000004                    TST127: SCOPE
 (2)  022022  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  022030  012777  004000  157720            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  022036  012777  060200  157704            MOV     #60200,@OFF             ;LOAD THE OFFSET REG.
 (1)  022044  012777  000014  157704            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  022052  012777  000024  157666            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  022060  052777  000001  157660            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  022066  052777  000002  157662            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  022074  042777  000002  157654            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  022102  005037  060200                    CLR     @#60200                 ;CLEAR THE TARGET LOCATION
 (1)  022106  052777  010000  157642            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  022114  012737  060200  001122            MOV     #60200,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  022122  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  022130  017737  156766  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  022136  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  022144  001401                            BEQ     TST130          ;;BR IF SAME
 (1)  022146  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 60200
1622                                    ;;*****************************************************************
 (4)                                    ;*TEST 130      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 06
 (4)                                    ;;*****************************************************************
 (3)  022150  000004                    TST130: SCOPE
 (2)  022152  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  022160  012777  004000  157570            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  022166  012777  060100  157554            MOV     #60100,@OFF             ;LOAD THE OFFSET REG.
 (1)  022174  012777  000014  157554            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  022202  012777  000024  157536            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  022210  052777  000001  157530            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  022216  052777  000002  157532            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  022224  042777  000002  157524            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  022232  005037  060100                    CLR     @#60100                 ;CLEAR THE TARGET LOCATION
 (1)  022236  052777  010000  157512            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  022244  012737  060100  001122            MOV     #60100,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  022252  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  022260  017737  156636  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  022266  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  022274  001401                            BEQ     TST131          ;;BR IF SAME
 (1)  022276  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 60100
```

```
1624                                    ;;*****************************************************************
 (4)                                    ;*TEST 131      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 05
 (4)                                    ;;*****************************************************************
 (3)  022300  000004                    TST131: SCOPE
 (2)  022302  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  022310  012777  004000  157440            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  022316  012777  060040  157424            MOV     #60040,@OFF             ;LOAD THE OFFSET REG.
 (1)  022324  012777  000014  157424            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  022332  012777  000024  157406            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  022340  052777  000001  157400            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  022346  052777  000002  157402            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  022354  042777  000002  157374            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  022362  005037  060040                    CLR     @#60040                 ;CLEAR THE TARGET LOCATION
 (1)  022366  052777  010000  157362            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  022374  012737  060040  001122            MOV     #60040,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  022402  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  022410  017737  156506  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  022416  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  022424  001401                            BEQ     TST132          ;;BR IF SAME
 (1)  022426  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 60040
1625                                    ;;*****************************************************************
 (4)                                    ;*TEST 132      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 04
 (4)                                    ;;*****************************************************************
 (3)  022430  000004                    TST132: SCOPE
 (2)  022432  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1)  022440  012777  004000  157310            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1)  022446  012777  060020  157274            MOV     #60020,@OFF             ;LOAD THE OFFSET REG.
 (1)  022454  012777  000014  157274            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1)  022462  012777  000024  157256            MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1)  022470  052777  000001  157250            BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2)  022476  052777  000002  157252            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2)  022504  042777  000002  157244            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1)  022512  005037  060020                    CLR     @#60020                 ;CLEAR THE TARGET LOCATION
 (1)  022516  052777  010000  157232            BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1)  022524  012737  060020  001122            MOV     #60020,$BDADR           ;LOAD EXPECTED ADDRESS
 (1)  022532  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1)  022540  017737  156356  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1)  022546  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
 (3)  022554  001401                            BEQ     TST133          ;;BR IF SAME
 (1)  022556  104033                            ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                    ;THE WRONG ADDRESS - EXPECTED ADR. WAS 60020
```

```
1627                                    ;;*****************************************************************
 (4)                                    ;*TEST 133      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 03
 (4)                                    ;;*****************************************************************
 (3) 022560 000004                      TST133: SCOPE
 (2) 022562 012737 000100 001160               MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1) 022570 012777 004000 157160               MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1) 022576 012777 060010 157144               MOV     #60010,@OFF             ;LOAD THE OFFSET REG.
 (1) 022604 012777 000014 157144               MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1) 022612 012777 000024 157126               MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1) 022620 052777 000001 157120               BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2) 022626 052777 000002 157122               BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2) 022634 042777 000002 157114               BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1) 022642 005037 060010                      CLR     @#60010                 ;CLEAR THE TARGET LOCATION
 (1) 022646 052777 010000 157102               BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1) 022654 012737 060010 001122               MOV     #60010,$BDADR           ;LOAD EXPECTED ADDRESS
 (1) 022662 012737 000001 001124               MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1) 022670 017737 156226 001126               MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1) 022676 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE
 (3) 022704 001401                             BEQ     TST134          ;;BR IF SAME
 (1) 022706 104033                             ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                   ;THE WRONG ADDRESS - EXPECTED ADR. WAS 60010
1628                                    ;;*****************************************************************
 (4)                                    ;*TEST 134      VERIFY A 1 WORD MATRIX MODE XFR USING OFFSET  - 02
 (4)                                    ;;*****************************************************************
 (3) 022710 000004                      TST134: SCOPE
 (2) 022712 012737 000100 001160               MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (1) 022720 012777 004000 157030               MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
 (1) 022726 012777 060004 157014               MOV     #60004,@OFF             ;LOAD THE OFFSET REG.
 (1) 022734 012777 000014 157014               MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
 (1) 022742 012777 000024 156776               MOV     #BIT4!BIT2,@CSR         ;SELECT MATRIX WORD MODE
 (1) 022750 052777 000001 156770               BIS     #BIT0,@CSR              ;ENABLE NCV11
 (2) 022756 052777 000002 156772               BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (2) 022764 042777 000002 156764               BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 (1) 022772 005037 060004                      CLR     @#60004                 ;CLEAR THE TARGET LOCATION
 (1) 022776 052777 010000 156752               BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
 (1) 023004 012737 060004 001122               MOV     #60004,$BDADR           ;LOAD EXPECTED ADDRESS
 (1) 023012 012737 000001 001124               MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
 (1) 023020 017737 156076 001126               MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
 (1) 023026 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE
 (3) 023034 001401                             BEQ     TST135          ;;BR IF SAME
 (1) 023036 104033                             ERROR   33                      ;OFFSET INPUT TO MATRIX MUX. SELECTED
 (1)                                                                   ;THE WRONG ADDRESS - EXPECIED ADR. WAS 60004
```

```
1633                                    ;;*****************************************************************
 (3)                                    ;*TEST 135      VERIFY THE ADM INPUT TO THE MATRIX MUX. USING GAIN ENABLE
 (3)                                    ;;*****************************************************************
 (2)  023040  000004                    TST135: SCOPE
 (1)  023042  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1634                                    ;WITH GAIN SET IN TEST CONTROL MODE, THE SELF TEST DATA SHOULD BE BYTE 250
1635                                    ;WHEN PROCESSED THRU THE ADDRESS MAKER LOGIC THE NEW VALUE IS "2552"
1636                                    ;VERIFY THAT THE MATRIX MUX CAN ADD 62550 AND 2552 CORRECTLY
1637  023050  012777  004000  156700            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1638  023056  012777  062550  156664            MOV     #BUF0+2550,@OFF         ;LOAD THE OFFSET REG.
1639  023064  012777  000014  156664            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
1640  023072  012777  002024  156646            MOV     #2024,@CSR              ;ENABLE GAIN, WORD MATRIX MODE
1641  023100  052777  000001  156640            BIS     #BIT0,@CSR              ;ENABLE THE NCV11
1642  023106  052777  000002  156642            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  023114  042777  000002  156634            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1643  023122  005037  065322                    CLR     @#BUF0+5322             ;CLEAR THE TARGET ADDRESS
1644  023126  005037  062722                    CLR     @#BUF0+2722             ;CLEAR THE TARGET ADDRESS IF "TESTX" FAILS
1645  023132  052777  010000  156616            BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
1646  023140  000240                            NOP
1647  023142  000240                            NOP
1648  023144  000240                            NOP
1649  023146  012737  065322  001122            MOV     #BUF0+5322,$BDADR       ;LOAD THE EXPECTED ADDRESS
1650  023154  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD THE EXPECTED DATA
1651  023162  017737  155734  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
1652  023170  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
1653  023176  001413                            BEQ     TST136          ;;BR IF SAME
1654  023200  005737  062722                    TST     @#BUF0+2722             ;TEST OTHER ADDRESS
1655  023204  001002                            BNE     1$                      ;BR IF NON-ZERO
1656  023206  104033                            ERROR   33                      ;MATRIX MODE ADDER ERROR
1657                                            ;USING GAIN AND WORD MATRIX MODE, THE ADDRESSES SELECTED WAS INCORRECT
1658  023210  000406                            BR      TST136          ;;
1659  023212  013737  062722  001126  1$:       MOV     @#BUF0+2722,$BDDAT      ;LOAD INCORRECT DATA
1660  023220  005037  001124                    CLR     $GDDAT                  ;CLEAR THE EXPECTED DATA
1661  023224  104033                            ERROR   33                      ;MATRIX MODE ADDER ERROR DUE TO
1662                                                                    ;"TESTX" LOGIC SET IN ERROR
```

```
1664                                    ;;*****************************************************************
 (3)                                    ;*TEST 136      VERIFY THE ADM INPUT TO THE MATRIX MUX. ADDER  USING ZB ENABLE
 (3)                                    ;;*****************************************************************
 (2)  023226  000004                    TST136: SCOPE
 (1)  023230  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1665                                    ;WITH ZB ENABLE SET IN TEST CONTROL MODE, THE SELF TEST DATA SHOULD BE BYTE 224
1666                                    ;WHEN PROCESSED THRU THE ADDRESS MAKER LOGIC THE NEW VALUE IS 12450
1667                                    ;VERIFY THAT THE MATRIX MUX CAN ADD CORRECTLY
1668  023236  012777  004000  156512            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1669  023244  012777  065224  156476            MOV     #BUF0+5224,@OFF         ;LOAD THE OFFSET REGISTER
1670  023252  012777  000014  156476            MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
1671  023260  012777  004024  156460            MOV     #4024,@CSR              ;ENABLE ZB AND WORD MATRIX MODE
1672  023266  052777  000002  156462            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
1673  023274  052777  000001  156444            BIS     #BIT0,@CSR              ;ENABLE NCV11
1674  023302  042777  000001  156446            BIC     #BIT0,@SFR              ;DISABLE "TEST Z" PULSES
1675  023310  005037  072450                    CLR     @#BUF0+12450            ;CLEAR THE TARGET LOCATION
1676  023314  052777  010000  156434            BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
1677  023322  012737  072450  001122            MOV     #BUF0+12450,$BDADR      ;LOAD THE EXPECTED ADDRESS
1678  023330  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD THE EXPECTED DATA
1679  023336  017737  155560  001126            MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
1680  023344  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
1681  023352  001401                            BEQ     TST137          ;;BR IF SAME
1682  023354  104033                            ERROR   33              ;ADM INPUT TO THE MATRIX MUX SELECTED
1683                                                                    ;WRONG ADDRESS  -  EXPECTED ADDRESS WAS BUF0 + 2450
1684                                    ;;*****************************************************************
 (3)                                    ;*TEST 137      VERIFY LOW BYTE OPERATION OF THE "TESTX" FLOP
 (3)                                    ;;*****************************************************************
 (2)  023356  000004                    TST137: SCOPE
 (1)  023360  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1685  023366  012777  004000  156362            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1686  023374  012777  060000  156346            MOV     #BUF0,@OFF              ;LOAD OFFSET REG.
1687  023402  012777  000014  156346            MOV     #TSTDMA!TSICON,@SFR     ;LOAD TEST DMA AND CONTROL
1688  023410  012777  002024  156330            MOV     #2024,@CSR              ;SET "GAIN" AND WORD MATRIX MODE
1689  023416  052777  000001  156322            BIS     #BIT0,@CSR              ;SET NCV11 ACTIVE
1690  023424  052777  000002  156324            BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)  023432  042777  000002  156316            BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1691  023440  052777  000020  156310            BIS     #BIT4,@SFR              ;SET "TESTX" FLOP
1692  023446  005037  062552                    CLR     @#BUF0+2552             ;CLEAR THE TEST FAILED LOCATION
1693  023452  005037  060152                    CLR     @#BUF0+0152             ;CLEAR THE TESTX WORKED LOCATION
1694  023456  052777  010000  156272            BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
1695  023464  000240                            NOP
1696  023466  000240                            NOP
1697  023470  000240                            NOP
1698  023472  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
1699  023500  013737  060152  001126            MOV     @#BUF0+0152,$BDDAT      ;READ THE ACTUAL DATA
1700  023506  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE DATA
1701  023514  001413                            BEQ     TST140          ;;BR IF SAME
1702  023516  005737  062552                    TST     @#BUF0+2552             ;TEST TESTX FAILED LOCATION
1703  023522  001002                            BNE     1$              ;;BR IF YES
1704  023524  104034                            ERROR   34              ;TESTX FAILED TO INHIBIT BITS 8-15 OF THE "SUM ADDER"
1705  023526  000406                            BR      TST140          ;;
1706  023530  013737  062552  001126  1$:       MOV     @#BUF0+2552,$BDDAT      ;GET ACTUAL DATA
1707  023536  005037  001124                    CLR     $GDDAT                  ;CLEAR EXPECTED
1708  023542  104034                            ERROR   34                      ;TESTX FAILED TO INHIBIT BITS 8-15 OF THE "SUM A
```

```
1710                                    ;;*****************************************************************
 (3)                                    ;*TEST 140      VERIFY BIT 12 ADDER INPUT TO MATRIX MODE MUX
 (3)                                    ;;*****************************************************************
 (2) 023544 000004                      TST140: SCOPE
 (1) 023546 012737 000100 001160               MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1711 023554 012777 004000 156174               MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1712 023562 012777 050000 156160               MOV     #50000,@OFF             ;LOAD THE OFFSET REG.
1713 023570 012777 000014 156160               MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
1714 023576 012777 002030 156142               MOV     #2030,@CSR              ;SELECT GAIN MATRIX BYTE MODE
1715 023604 052777 000001 156134               BIS     #BIT0,@CSR              ;ENABLE NCV11
1716 023612 052777 000002 156136               BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 023620 042777 000002 156130               BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1717 023626 005037 062726                      CLR     @#BUF0+2726             ;CLEAR THE TARGET LOCATION
1718 023632 052777 010000 156116               BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
1719 023640 012737 062726 001122               MOV     #BUF0+2726,$BDADR       ;LOAD EXPECTED ADDRESS
1720 023646 012737 000001 001124               MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
1721 023654 017737 155242 001126               MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
1722 023662 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE
1723 023670 001401                             BEQ     TST141          ;;BR IF SAME
1724 023672 104033                             ERROR   33                      ;MATRIX ADDER FAILED TO SELECT CORRECT ADDRESS
1725                                                                   ;THE WRONG ADDRESS - EXPECTED ADR. WAS BUF0+2726
1726                                    ;;*****************************************************************
 (3)                                    ;*TEST 141      VERIFY BIT 13 ADDER INPUT TO MATRIX MODE MUX
 (3)                                    ;;*****************************************************************
 (2) 023674 000004                      TST141: SCOPE
 (1) 023676 012737 000100 001160               MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1727 023704 012777 004000 156044               MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1728 023712 012777 037774 156030               MOV     #37774,@OFF             ;LOAD THE OFFSET REG.
1729 023720 012777 000014 156030               MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
1730 023726 012777 002032 156012               MOV     #2032,@CSR              ;SELECT GAIN MATRIX BYTE MODE
1731 023734 052777 000001 156004               BIS     #BIT0,@CSR              ;ENABLE NCV11
1732 023742 052777 000002 156006               BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 023750 042777 000002 156000               BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1733 023756 005037 065522                      CLR     @#BUF0+5522             ;CLEAR THE TARGET LOCATION
1734 023762 052777 010000 155766               BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
1735 023770 012737 065522 001122               MOV     #BUF0+5522,$BDADR       ;LOAD EXPECTED ADDRESS
1736 023776 012737 000001 001124               MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
1737 024004 017737 155112 001126               MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
1738 024012 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE
1739 024020 001401                             BEQ     TST142          ;;BR IF SAME
1740 024022 104033                             ERROR   33                      ;MATRIX ADDER FAILED TO SELECT CORRECT ADDRESS
1741                                                                   ;THE WRONG ADDRESS - EXPECTED ADR. WAS BUF0+5522
```

```
1743                                    ;;*****************************************************************
 (3)                                    ;*TEST 142      VERIFY BIT 14 ADDER INPUT TO MATRIX MODE MUX
 (3)                                    ;;*****************************************************************
 (2) 024024  000004                     TST142: SCOPE
 (1) 024026  012737  000100  001160             MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1744                                    ;OFFSET =       50000
1745                                    ;ADM OUTPUT     =       53656
1746                                    ;TARGET         =       123656
1747 024034  012777  004000  155714             MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1748 024042  023727  036406  001340             CMP     $LSTBK,#1340            ;TEST IF ENOUGH MEMORY >20K
1749 024050  103445                             BLO     TST143          ;;BR IF NO ROOM
1750 024052  012777  050000  155670             MOV     #50000,@OFF             ;LOAD THE OFFSET REG.
1751 024060  012777  000014  155670             MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
1752 024066  012777  002034  155652             MOV     #2034,@CSR              ;SELECT GAIN MATRIX BYTE MODE
1753 024074  052777  000001  155644             BIS     #BIT0,@CSR              ;ENABLE NCV11
1754 024102  052777  000002  155646             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 024110  042777  000002  155640             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
1755 024116  005037  123656                     CLR     @#BUF0+43656            ;CLEAR THE TARGET LOCATION
1756 024122  052777  010000  155626             BIS     #BIT12,@SFR             ;ALLOW 1 WORD TRANSFER
1757 024130  012737  123656  001122             MOV     #BUF0+43656,$BDADR      ;LOAD EXPECTED ADDRESS
1758 024136  012737  000001  001124             MOV     #1,$GDDAT               ;LOAD EXPECTED DATA
1759 024144  017737  154752  001126             MOV     @$BDADR,$BDDAT          ;READ THE ACTUAL DATA
1760 024152  023737  001124  001126             CMP     $GDDAT,$BDDAT           ;COMPARE
1761 024160  001401                             BEQ     TST143          ;;BR IF SAME
1762 024162  104033                             ERROR   33                      ;MATRIX ADDER FAILED TO SELECT CORRECT ADDRESS
1763                                                                     ;THE WRONG ADDRESS - EXPECTED ADR. WAS BUF0+43656
```

```
                                ;;*****************************************************************
                                ;*TEST 143      CHECK FOR HIGH BYTE "INC OVFL" TO SET CELL OVERFLOW
                                ;;*****************************************************************
024164  000004                  TST143: SCOPE
024166  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
024174  012777  004000  155554          MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
024202  005077  155544                  CLR     @WCR                    ;CLEAR W.C.
024206  005077  155542                  CLR     @BAR                    ;CLEAR B.A.
024212  012777  060000  155530          MOV     #BUF0,@OFF              ;LOAD INITIAL OFFSET REGISTER
024220  012777  002022  155520          MOV     #BIT10!BIT4!BIT1,@CSR   ;ENABLE MATRIX MODE BYTE AND SET GAIN
024226  012777  000014  155522          MOV     #TSTDMA!TSTCON,@SFR     ;SET TEST DMA AND CONTROL
024234  052777  000001  155504          BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
024242  052777  000002  155506          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
024250  042777  000002  155500          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
024256  000240                          NOP
024260  000240                          NOP
024262  000240                          NOP
024264  112737  000100  061264          MOVB    #100,BUF0+1264          ;LOAD LOW BYTE TO A KNOWN VALUE
024272  112737  000377  061265          MOVB    #377,BUF0+1265          ;SET HIGH BYTE TO 377
024300  052777  010000  155450          BIS     #BIT12,@SFR             ;ALLOW 1 TRANSFER
024306  000240                          NOP
024310  000240                          NOP
024312  000240                          NOP
024314  012737  022022  001124          MOV     #BIT13'BIT10!BIT4:BIT1,$GDDAT   ;LOAD EXPECTED STATUS
024322  017737  155420  001126          MOV     @CSR,$BDDAT             ;READ THE ACTUAL STATUS
024330  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
024336  001401                          BEQ     1$              ;;BR IF SAME
024340  104030                          ERROR   30                      ;CELL OVERFLOW FAILED TO SET
                                                                ;WHEN INCREMENTING THE HIGH BYTE
024342  013737  061264  001126  1$:     MOV     BUF0+1264,$BDDAT        ;READ THE BUF0 LOCATION
024350  012737  177500  001124          MOV     #177500,$GDDAT          ;LOAD EXPECTED VALUE
024356  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE VALUES
024364  001401                          BEQ     TST144          ;;BR IF SAME
024366  104030                          ERROR   30              ;TARGET LOC. DATA WAS INCORRECT
```

```
                                        ;;*****************************************************************
                                        ;*TEST 144      VERIFY EACH BIT OF THE LOWER 16 BIT Z COUNTER   (TESTER ONLY)
                                        ;;*****************************************************************
024370  000004                  TST144: SCOPE
024372  012737  000400  001160          MOV     #400,$TIMES     ;;DO 400 ITERATIONS
024400  005737  050026                  TST     WFMODE                  ;TEST IF TESTER MODE
024404  001454                          BEQ     TST145          ;;BR IF NOT
024406  012737  000001  001124          MOV     #1,$GDDAT               ;LOAD EXPECTED VALUE
024414  012737  024422  001110          MOV     #1$,$LPERR              ;LOAD LOOP ADDRESS

024422  012777  004000  155326  1$:     MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
024430  012777  000010  155320          MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
024436  012777  000000  155306          MOV     #0,@WCR                 ;LOAD UPPER 16 BITS
024444  013777  001124  155302          MOV     $GDDAT,@BAR             ;
024452  005377  155276                  DEC     @BAR                    ;LOAD INITIAL COUNTER VALUE
024456  012777  000022  155262          MOV     #BIT4!BIT1,@CSR         ;ENTER MATRIX MODE
024464  052777  000001  155254          BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
024472  052777  000001  155226          BIS     #BIT0,@DACSR            ;GENERATE "7" PULSE
024500  105777  155222          2$:     TSTB    @DACSR                  ;WAIT FOR Z PULSE COMPLETION
024504  100375                          BPL     2$
024506  017737  155242  001126          MOV     @BAR,$BDDAT             ;READ EXPECTED REGISTER VALUE
024514  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
024522  001402                          BEQ     3$              ;;BR IF SAME
024524  104010                          ERROR   10              ;LOWER 16 BITS OF THE Z COUNTER IN ERROR
024526  000403                          BR      TST145          ;;
024530  006337  001124          3$:     ASL     $GDDAT                  ;CHANGE THE DATA BIT
024534  001332                          BNE     1$                      ;BR AND TRY NEXT BIT
                                        ;;*****************************************************************
                                        ;*TEST 145      VERIFY EACH BIT OF THE HIGHER 16 BIT Z COUNTER  (TESTER ONLY)
                                        ;;*****************************************************************
024536  000004                  TST145: SCOPE
024540  012737  000400  001160          MOV     #400,$TIMES     ;;DO 400 ITERATIONS
024546  005737  050026                  TST     WFMODE                  ;TEST IF TESTER MODE
024552  001454                          BEQ     TST146          ;;BR IF NOT
024554  012737  000001  001124          MOV     #1,$GDDAT               ;LOAD EXPECTED VALUE
024562  012737  024570  001110          MOV     #1$,$LPERR              ;LOAD LOOP ADDRESS

024570  012777  004000  155160  1$:     MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
024576  012777  000010  155152          MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
024604  013777  001124  155140          MOV     $GDDAT,@WCR             ;
024612  005377  155134                  DEC     @WCR                    ;LOAD INITIAL COUNTER VALUE
024616  012777  177777  155130          MOV     #-1,@BAR                ;LOAD LOWER 16 BITS
024624  012777  000022  155114          MOV     #BIT4!BIT1,@CSR         ;ENTER MATRIX MODE
024632  052777  000001  155106          BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
024640  052777  000001  155060          BIS     #BIT0,@DACSR            ;GENERATE "Z" PULSE
024646  105777  155054          2$:     TSTB    @DACSR                  ;WAIT FOR Z PULSE COMPLETION
024652  100375                          BPL     2$
024654  017737  155072  001126          MOV     @WCR,$BDDAT             ;READ EXPECTED REGISTER VALUE
024662  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
024670  001402                          BEQ     3$              ;;BR IF SAME
024672  104011                          ERROR   11              ;HIGHER 16 BITS OF THE Z COUNTER IN ERROR
024674  000403                          BR      TST146          ;;
024676  006337  001124          3$:     ASL     $GDDAT                  ;CHANGE THE DATA BIT
024702  001332                          BNE     1$                      ;BR AND TRY NEXT BIT
```

```
1842                                    ;;*****************************************************************
 (3)                                    ;*TEST 146      VERIFY THAT CAMERA 01 CHANNEL IS OPERATIONAL    (TESTER ONLY)
 (3)                                    ;;*****************************************************************
 (2) 024704 000004                      TST146: SCOPE
 (1) 024706 012737 000400 001160                MOV     #400,$TIMES     ;;DO 400 ITERATIONS
1843 024714 005737 050026                       TST     WFMODE                  ;TEST IF TESTER MODE
1844 024720 001443                              BEQ     TST147          ;;BR IF NOT
1845 024722 012737 000001 001124                MOV     #1,$GDDAT               ;LOAD EXPECTED VALUE
1846
1847 024730 012777 004000 155020 1$:            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1848 024736 012777 000010 155012                MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
1849 024744 012777 000000 155000                MOV     #0,@WCR                 ;LOAD UPPER 16 BITS
1850 024752 012777 000000 154774                MOV     #0,@BAR                 ;LOAD LOWER 16 BITS
1851 024760 012777 000422 154760                MOV     #BIT8!BIT4!BIT1,@CSR    ;ENTER MATRIX MODE ON CAMERA 01
1852 024766 052777 000001 154752                BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1853 024774 052777 000001 154724                BIS     #BIT0,@DACSR            ;GENERATE "Z" PULSE
1854 025002 105777 154720               2$:     TSTB    @DACSR                  ;WAIT FOR Z PULSE COMPLETION
1855 025006 100375                              BPL     2$
1856 025010 017737 154740 001126                MOV     @BAR,$BDDAT             ;READ EXPECTED REGISTER VALUE
1857 025016 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE
1858 025024 001401                              BEQ     TST147          ;;BR IF SAME
1859 025026 104010                              ERROR   10                      ;LOWER 16 BITS OF THE Z COUNTER
1860                                                                    ;IN ERROR WHEN USING CAMERA 01
1861
1862                                    ;;*****************************************************************
 (3)                                    ;*TEST 147      VERIFY THAT CAMERA 10 CHANNEL IS OPERATIONAL    (TESTER ONLY)
 (3)                                    ;;*****************************************************************
 (2) 025030 000004                      TST147: SCOPE
 (1) 025032 012737 000100 001160                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1863 025040 005737 050026                       TST     WFMODE                  ;TEST IF TESTER MODE
1864 025044 001443                              BEQ     TST150          ;;BR IF NOT
1865 025046 012737 000001 001124                MOV     #1,$GDDAT               ;LOAD EXPECTED VALUE
1866
1867 025054 012777 004000 154674 1$:            MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1868 025062 012777 000010 154666                MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
1869 025070 012777 000000 154654                MOV     #0,@WCR                 ;LOAD UPPER 16 BITS
1870 025076 012777 000000 154650                MOV     #0,@BAR                 ;LOAD LOWER 16 BITS
1871 025104 012777 001022 154634                MOV     #BIT9!BIT4!BIT1,@CSR    ;ENTER MATRIX MODE
1872 025112 052777 000001 154626                BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1873 025120 052777 000001 154600                BIS     #BIT0,@DACSR            ;GENERATE "Z" PULSE
1874 025126 105777 154574               2$:     TSTB    @DACSR                  ;WAIT FOR Z PULSE COMPLETION
1875 025132 100375                              BPL     2$
1876 025134 017737 154614 001126                MOV     @BAR,$BDDAT             ;READ EXPECTED REGISTER VALUE
1877 025142 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE
1878 025150 001401                              BEQ     TST150          ;;BR IF SAME
1879 025152 104010                              ERROR   10                      ;LOWER 16 BITS OF THE Z COUNTER
1880                                                                    ;IN ERROR WHEN USING CAMERA 10
```

```
1882                                    ;;*****************************************************************
 (3)                                    ;*TEST 150      VERIFY THAT CAMERA 11 CHANNEL IS OPERATIONAL    (TESTER ONLY)
 (3)                                    ;;*****************************************************************
 (2)  025154  000004                    TST150: SCOPE
 (1)  025156  012737  000100  001160            MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1883  025164  005737  050026                    TST     WFMODE                  ;TEST IF TESTER MODE
1884  025170  001443                            BEQ     TST151          ;;BR IF NOT
1885  025172  012737  000001  001124            MOV     #1,$GDDAT               ;LOAD EXPECTED VALUE
1886
1887  025200  012777  004000  154550  1$:       MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
1888  025206  012777  000010  154542            MOV     #TSTDMA,@SFR            ;SET TEST DMA FLOP
1889  025214  012777  000000  154530            MOV     #0,@WCR                 ;LOAD UPPER 16 BITS
1890  025222  012777  000000  154524            MOV     #0,@BAR                 ;LOAD LOWER 16 BITS
1891  025230  012777  001422  154510            MOV     #BIT9!BIT8!BIT4!BIT1,@CSR       ;ENTER MATRIX MODE
1892  025236  052777  000001  154502            BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
1893  025244  052777  000001  154454            BIS     #BIT0,@DACSR            ;GENERATE "Z" PULSE
1894  025252  105777  154450          2$:       TSTB    @DACSR                  ;WAIT FOR Z PULSE COMPLETION
1895  025256  100375                            BPL     2$
1896  025260  017737  154470  001126            MOV     @BAR,$BDDAT             ;READ EXPECTED REGISTER VALUE
1897  025266  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE
1898  025274  001401                            BEQ     TST151          ;;BR IF SAME
1899  025276  104010                            ERROR   10                      ;LOWER 16 BITS OF THE Z COUNTER
1900                                                                    ;IN ERROR WHEN USING CAMERA 11
```

```
1902                                    ;;*****************************************************************
 (3)                                    ;*TEST 151      DYNAMIC MATRIX MODE ADDRESS
 (3)                                    ;;*****************************************************************
 (2)  025300  000004                    TST151: SCOPE
 (1)  025302  012737  000002  001160            MOV     #2,$TIMES       ;;DO 2 ITERATIONS
1903                                    ;CELL INCREMENT OF A FLOATING LOCATION (60000,64000,62000,61000)
1904                                    ;FILL THE MEMORY BUFFER WITH A KNOWN PATTERN (125252)
1905                                    ;COLLECT DATA OF 0 AND INCREMENT A TARGET LOCATION
1906                                    ;VERIFY THAT NO OTHER ADDRESS IS CHANGED
1907                                    ;AFTER VERIFYING THE WHOLE BUFFER - UPDATE THE OFFSET REGISTER
1908                                    ;       AND INCREMENT ANOTHER LOCATION
1909  025310  012737  060000  050032            MOV     #BUF0,CURENT    ;PRIME THE CURRENT TARGET LOC.
1910  025316  012737  004000  025666            MOV     #BIT11,100$     ;LOAD FORCE BIT
1911  025324  012737  025332  001110            MOV     #2$,$LPERR      ;LOAD RETURN ADDRESS
1912                                    ;PRIME THE BUFFER WITH A 125252 PATTERN
1913  025332  012700  060000            2$:     MOV     #BUF0,R0        ;LOAD POINTER TO BUFFER
1914  025336  012701  125252                    MOV     #125252,R1      ;LOAD VALUE
1915  025342  013702  003354                    MOV     ADNOKT,R2       ;LOAD BUFFER END ADDRESS
1916  025346  010120                    3$:     MOV     R1,(R0)+        ;LOAD BUFFER WITH DATA
1917  025350  020002                            CMP     R0,R2           ;TEST FOR END
1918  025352  001375                            BNE     3$
1919  025354  012777  177770  022450            MOV     #-10,@CURENT    ;PRIME THE TARGET LOCATION
1920
1921  025362  012777  004000  154366            MOV     #CLRALL,@SFR    ;INIT THE DEVICE
1922  025370  013777  050032  154352            MOV     CURENT,@OFF     ;LOAD OFFSET TO TARGET
1923  025376  012777  177777  154346            MOV     #-1,@WCR        ;PRIME THE HIGH 16 BIT Z COUNTER
1924  025404  005077  154344                    CLR     @BAR            ;CLEAR LOW 16 BIT COUNTER
1925  025410  012777  000004  154340            MOV     #TSTCON,@SFR    ;ENABLE TEST CONNECTOR
1926  025416  012777  000024  154322            MOV     #24,@CSR        ;LOAD MATRIX WORD MODE
1927  025424  052777  000002  154324            BIS     #TESTZ,@SFR     ;SET TEST Z FLOP
1928  025432  052777  000001  154306            BIS     #BIT0,@CSR      ;ENABLE THE NCV11
1929
1930  025440  013700  002012                    MOV     CPUDL2,R0       ;PRIME THE DELAY
1931  025444  005001                            CLR     R1
1932  025446  032777  060000  154272    4$:     BIT     #BIT14!BIT13,@CSR       ;TEST FOR CELL OR Z OVERFLOW
1933  025454  001014                            BNE     5$              ;;BR IF EITHER IS SET
1934  025456  005301                            DEC     R1              ;DELAY
1935  025460  001372                            BNE     4$
1936  025462  005300                            DEC     R0              ;DELAY
1937  025464  001370                            BNE     4$
1938  025466  017737  154254  001126            MOV     @CSR,$BDDAT     ;READ BAD STATUS
1939  025474  012737  060224  001124            MOV     #BIT14.BIT13+224,$GDDAT ;LOAD EXPECTED STATUS
1940  025502  104002                            ERROR   2               ;DYNAMIC MATRIX MODE STATUS ERROR
1941  025504  000471                            BR      TST152          ;;
1942  025506  012700  060000            5$:     MOV     #BUF0,R0        ;GET BUFFER POINER
1943  025512  023700  050032            6$:     CMP     CURENT,R0       ;TEST IF TARGET ADDRESS
1944  025516  001415                            BEQ     7$              ;;BR IF YES
1945  025520  010037  001122                    MOV     R0,$BDADR       ;GET BAD ADDRESS FOR TYPE-OUT
1946  025524  011037  001126                    MOV     (R0),$BDDAT     ;GET BAD DATA
1947  025530  012737  125252  001124            MOV     #125252,$GDDAT  ;LOAD EXPECTED DATA
1948  025536  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE
1949  025544  001420                            BEQ     10$             ;;BR IF SAME
1950  025546  104033                            ERROR   33              ;CHANGED AN INCORRECT TARGET LOCATION
1951  025550  000447                            BR      TST152          ;;
1952  025552  013737  050032  001122    7$:     MOV     CURENT,$BDADR   ;LOAD EXPECTED ADDRESS
1953  025560  017737  022246  001126            MOV     @CURENT,$BDDAT  ;LOAD ACTUAL DATA
```

```
 1954  025566  012737  177777  001124          MOV     #-1,$GDDAT      ;LOAD EXPECTED DATA
 1955  025574  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE
 1956  025602  001401                          BEQ     10$             ;;BR IF SAME
 1957  025604  104030                          ERROR   30              ;CELL INCREMENT DATA ERROR
 1958
 1959  025606  005720                  10$:    TST     (R0)+           ;BUMP THE POINTER INTO THE BUFFER
 1960  025610  023700  003354                  CMP     ADNOXT,R0       ;TEST IF FINISHED THE BUFFER
 1961  025614  001336                          BNE     6$              ;BR AND RETEST THE REST OF THE BUFFER
 1962
 1963  025616  005737  001202                  TST     $PASS           ;TEST IF FIRST PASS
 1964  025622  001422                          BEQ     TST152          ;;BR IF YES
 1965  025624  032777  004000  153306          BIT     #SW11,@SWR      ;TEST INHIBIT INTER.
 1966  025632  001016                          BNE     TST152          ;;BR IF SET
 1967  025634  006237  025666                  ASR     100$            ;CHANGE THE FORCED ADDR. BIT
 1968  025640  022737  000002  025666          CMP     #2,100$         ;TEST IF FINISHED
 1969  025646  001410                          BEQ     TST152          ;;BR IF FINISHED
 1970  025650  012737  060000  050032          MOV     #BUF0,CURENT    ;MAKE UP NEW ADDRESS
 1971  025656  053737  025666  050032          BIS     100$,CURENT     ;
 1972  025664  000622                          BR      2$
 1973  025666  004000                  100$:   BIT11
 1974                                  ;;*******************************************************************
  (3)                                  ;*TEST 152       DYNAMIC LIST MODE ADDRESS
  (3)                                  ;;*******************************************************************
  (2)  025670  000004                  TST152: SCOPE
  (1)  025672  012737  000002  001160          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
 1975                                  ;FILL THE MEMORY BUFFER WITH A KNOWN PATTERN (125252)
 1976                                  ;DO A 1 WORD TRANSFER TO A TAGET LOCATION
 1977                                  ;VERIFY THAT NO OTHER LOCATION IN THE BUFFER IS CHANGED
 1978                                  ;AFTER VERIFYING BUMP THE BUS ADDRESS AND DO SUB-TEST AGAIN
 1979  025700  012737  060000  050032          MOV     #BUF0,CURENT    ;PRIME THE CURRENT TARGET LOC.
 1980  025706  012737  020000  026246          MOV     #BIT13,100$     ;LOAD FORCE BIT
 1981  025714  012737  025722  001110          MOV     #2$,$LPERR      ;LOAD RETURN ADDRESS
 1982                                  ;PRIME THE BUFFER WITH A 125252 PATTERN
 1983  025722  012700  060000          2$:     MOV     #BUF0,R0        ;LOAD POINTER TO BUFFER
 1984  025726  012701  125252                  MOV     #125252,R1      ;LOAD VALUE
 1985  025732  013702  003354                  MOV     ADNOXT,R2       ;LOAD BUFFER END ADDRESS
 1986  025736  010120                  3$:     MOV     R1,(R0)+        ;LOAD BUFFER WITH DATA
 1987  025740  020002                          CMP     R0,R2           ;TEST FOR END
 1988  025742  001375                          BNE     3$
 1989
 1990  025744  012777  004000  154004          MOV     #CLRALL,@SFR    ;INIT THE DEVICE
 1991  025752  012777  177777  153772          MOV     #-1,@WCR        ;PRIME THE WORD COUNT
 1992  025760  013777  050032  153766          MOV     CURENT,@BAR     ;LOAD TARGET LOCATION POINTER
 1993  025766  012777  000004  153762          MOV     #TSTCON,@SFR    ;ENABLE MAINT. MODE
 1994  025774  052777  000001  153744          BIS     #BIT0,@CSR      ;ENABLE THE NCV11
 1995  026002  052777  000002  153746          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
  (1)  026010  042777  000002  153740          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
 1996
 1997  026016  013700  002012                  MOV     CPUDL2,R0       ;PRIME THE DELAY
 1998  026022  005001                          CLR     R1
 1999  026024  032777  060000  153714  4$:     BIT     #BIT14!BIT13,@CSR       ;TEST FOR CELL OR Z OVERFLOW
 2000  026032  001014                          BNE     5$              ;;BR IF EITHER IS SET
 2001  026034  005301                          DEC     R1              ;DELAY
 2002  026036  001372                          BNE     4$
 2003  026040  005300                          DEC     R0              ;DELAY
 2004  026042  001370                          BNE     4$
```

```
2005  026044  017737  153676  001126          MOV     @CSR,$BDDAT     ;READ BAD STATUS
2006  026052  012737  060224  001124          MOV     #BIT14.BIT13+224,$GDDAT ;LOAD EXPECTED STATUS
2007  026060  104036                          ERROR   36              ;DYNAMIC LIST MODE STATUS ERROR
2008  026062  000472                          BR      TST153          ;;
2009  026064  012700  060000          5$:     MOV     #BUF0,R0        ;GET BUFFER POINER
2010  026070  023700  050032          6$:     CMP     CURENT,R0       ;TEST IF TARGET ADDRESS
2011  026074  001415                          BEQ     7$              ;;BR IF YES
2012  026076  010037  001122                  MOV     R0,$BDADR       ;GET BAD ADDRESS FOR TYPE-OUT
2013  026102  011037  001126                  MOV     (R0),$BDDAT     ;GET BAD DATA
2014  026106  012737  125252  001124          MOV     #125252,$GDDAT  ;LOAD EXPECTED DATA
2015  026114  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE
2016  026122  001421                          BEQ     10$             ;;BR IF SAME
2017  026124  104021                          ERROR   21              ;CHANGED AN INCORRECT TARGET LOCATION
2018  026126  000450                          BR      TST153          ;;
2019  026130  013737  050032  001122  7$:     MOV     CURENT,$BDADR   ;LOAD EXPECTED ADDRESS
2020  026136  017737  021670  001126          MOV     @CURENT,$BDDAT  ;GET ACTUAL DATA
2021  026144  012737  003407  001124          MOV     #3407,$GDDAT    ;LOAD EXPECTED DATA
2022  026152  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE
2023  026160  001402                          BEQ     10$             ;;BR IF SAME
2024  026162  104037                          ERROR   37              ;LIST MODE DATA ERROR
2025  026164  000431                          BR      TST153          ;;
2026  026166  005720                  10$:    TST     (R0)+           ;BUMP THE POINTER INTO THE BUFFER
2027  026170  023700  003354                  CMP     ADNOKT,R0       ;TEST IF FINISHED THE BUFFER
2028  026174  001335                          BNE     6$              ;BR AND RETEST THE REST OF THE BUFFER
2029
2030  026176  005737  001202                  TST     $PASS           ;TEST IF FIRST PASS
2031  026202  001422                          BEQ     TST153          ;;BR IF FIRST PASS
2032  026204  032777  004000  152726          BIT     #SW11,@SWR      ;TEST INHIBIT INTER.
2033  026212  001016                          BNE     TST153          ;;BR IF SET
2034  026214  006237  026246                  ASR     100$            ;CHANGE THE FORCE ADDRESS
2035  026220  022737  000002  026246          CMP     #2,100$         ;TEST IF END
2036  026226  001410                          BEQ     TST153          ;;BR IF FINISHED
2037  026230  012737  060000  050032          MOV     #BUF0,CURENT    ;MAKE NEW ADDRESS
2038  026236  053737  026246  050032          BIS     100$,CURENT     ;
2039  026244  000626                          BR      2$
2040  026246  020000                  100$:   BIT13
```

```
                                        ;;*****************************************************************
                                        ;*TEST 153      DYNAMIC LIST MODE TRANSFER - MAXIMUM BUFFER LENGTH IN LOWER 28K
                                        ;;*****************************************************************
                                        TST153: SCOPE
026250  000004                                 
026252  012737  000040  001160                  MOV     #40,$TIMES      ;;DO 40 ITERATIONS
026260  013700  003354                          MOV     ADNOKT,R0       ;GET LAST ADDRESS
026264  162700  060000                          SUB     #BUF0,R0        ;DETERMINE THE WORD COUNT VALUE
026270  005400                                  NEG     R0
026272  006200                                  ASR     R0
                                        ;PRIME THE BUFFER WITH A 125252 PATTERN
026274  012703  060000                  2$:     MOV     #BUF0,R3        ;LOAD POINTER TO BUFFER
026300  012701  125252                          MOV     #125252,R1      ;LOAD VALUE
026304  013702  003354                          MOV     ADNOKT,R2       ;LOAD BUFFER END ADDRESS
026310  010123                          3$:     MOV     R1,(R3)+        ;LOAD BUFFER WITH DATA
026312  020302                                  CMP     R3,R2           ;TEST FOR END
026314  001375                                  BNE     3$
                                        ;NOW COLLET THE LIST MODE DATA
026316  012777  004000  153432                  MOV     #CLRALL,@SFR    ;INIT THE DEVICE
026324  010077  153422                          MOV     R0,@WCR         ;PRIME THE WORD COUNT
026330  012777  060000  153416                  MOV     #BUF0,@BAR      ;LOAD TARGET LOCATION POINTER
026336  012777  000004  153412                  MOV     #TSTCON,@SFR    ;ENABLE MAINT. MODE
026344  052777  000001  153374                  BIS     #BIT0,@CSR      ;ENABLE THE NCV11
026352  052777  000002  153376                  BIS     #TESTZ,@SFR     ;ENABLE MAINT. Z
026360  013700  002012                          MOV     CPUDL2,R0       ;PRIME THE DELAY
026364  032777  060000  153354  4$:     BIT     #BIT14!BIT13,@CSR       ;TEST FOR CELL OR Z OVERFLOW
026372  001014                                  BNE     5$              ;;BR IF EITHER IS SET
026374  005301                                  DEC     R1              ;DELAY
026376  001372                                  BNE     4$
026400  005300                                  DEC     R0              ;DELAY
026402  001370                                  BNE     4$
026404  017737  153336  001126                  MOV     @CSR,$BDDAT     ;READ BAD STATUS
026412  012737  060200  001124                  MOV     #BIT14!BIT13+200,$GDDAT ;LOAD EXPECTED STATUS
026420  104036                                  ERROR   36      ;DYNAMIC LIST MODE STATUS ERROR - NO WORD COUNT OVERFLOW
026422  000423                                  BR      TST154          ;;
026424  012700  060000          5$:     MOV     #BUF0,R0        ;GET BUFFER POINER
026430  010037  001122          6$:     MOV     R0,$BDADR       ;GET BAD ADDRESS FOR TYPE-OUT
026434  011037  001126                  MOV     (R0),$BDDAT     ;GET BAD DATA
026440  012737  003407  001124                  MOV     #3407,$GDDAT    ;LOAD EXPECTED DATA
026446  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE
026454  001402                                  BEQ     10$             ;;BR IF SAME
026456  104037                                  ERROR   37              ;LIST MODE DATA ERROR
026460  000404                                  BR      TST154          ;;
026462  005720                  10$.    TST     (R0)+           ;BUMP THE POINTER INTO THE BUFFER
026464  023700  003354                          CMP     ADNOKT,R0       ;TEST IF FINISHED THE BUFFER
026470  001357                                  BNE     6$              ;BR AND RETEST THE REST OF THE BUFFER
```

```
                                        ;;*****************************************************************
                                        ;*TEST 154       ONE MATRIX DATA TRANSFER TO EACH 4K EXTENDED MEMORY
                                        ;;*****************************************************************
026472  000004                          TST154: SCOPE
026474  012737  000002  001160                  MOV     #2,$TIMES       ;;DO 2 ITERATIONS
026502  005737  036124                          TST     $KT11           ;TEST IF KT-11 INSTALLED
026506  100127                          BPL     TST155          ;;BR IF NONE
                                        ;DO A BYTE MATRIX MODE TRANSFER TO A 1 BYTE LOCATION IN EACH 4K EXTENDED
                                        ;       MEMORY BANK
026510  012737  000600  041744          MOV     #600,OUT        ;LOAD INITIAL BANK VALUE
026516  013737  041744  172346  1$:     MOV     OUT,@#KIPAR3    ;LOAD KT-PAR REG #3 <BUFFER IS AT LOC 60000>
026524  012700  060000                  MOV     #BUF0,R0        ;LOAD BUFFER POINTER
026530  012710  125252                  MOV     #125252,(R0)    ;PRIME BAD TARGET LOC. IF EXT ADD FAILS
026534  052737  001400  177572          BIS     #1400,@#SR0     ;ENABLE MAINT. MODE KT-11
026542  012710  000370                  MOV     #370,(R0)       ;LOAD TARGET LOCATION DATA VALUE
026546  042737  001400  177572          BIC     #1400,@#SR0     ;DISABLE MAINT. MODE KT-11
026554  013701  041744                  MOV     OUT,R1          ;GET BANK VALUE
026560  005002                          CLR     R2              ;CLEAR EXT. ADD. TEMPORARY
026562  006301                          ASL     R1              ;MOVE LEFT
 (1)    026564  006301                  ASL     R1              ;MOVE LEFT
 (1)    026566  006301                  ASL     R1              ;MOVE LEFT
 (1)    026570  006301                  ASL     R1              ;MOVE LEFT
026572  006301                          ASL     R1
026574  006102                          ROL     R2              ;SAVE EA BITS
026576  006301                          ASL     R1
026600  006102                          ROL     R2              ;SAVE EA BITS
026602  010237  041730                  MOV     R2,NARROW       ;SAVE EA BITS FOR TYPEOUT
026606  010137  001122                  MOV     R1,$BDADR       ;SAVE ADDRESS BITS FOR TYPEOUT
026612  060201                          ADD     R2,R1           ;MAKE COMPLETE ADDRESS

                                        ;NOW GET READY TO DO THE TRANSFER
026614  012777  004000  153134          MOV     #CLRALL,@SFR    ;INIT THE NCV11
026622  010177  153122                  MOV     R1,@OFF         ;LOAD COMBINED BUFFER ADDRESS
026626  012777  000022  153112          MOV     #BIT4!BIT1,@CSR ;ENABLE THE NCV11
026634  012777  000014  153114          MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE MAINT NCV11 MODE
026642  052777  000001  153076          BIS     #BIT0,@CSR      ;ENABLE THE NCV11
026650  052777  000002  153100          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)    026656  042777  000002  153072  BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
026664  012737  000371  001124          MOV     #371,$GDDAT     ;LOAD EXPECTED DATA
026672  052777  010000  153056          BIS     #BIT12,@SFR     ;ENABLE 1 BYTE TRANSFER
026700  052737  000001  177572          BIS     #BIT0,@#SR0     ;ENABLE KT-11
026706  011037  001126                  MOV     (R0),$BDDAT     ;GET ACTUAL DATA
026712  042737  000001  177572          BIC     #BIT0,@#SR0     ;DISABLE KT-11
026720  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE DATA
026726  001401                          BEQ     2$              ;;BR IF SAME
026730  104035                          ERROR   35              ;DATA TRANSFER ERROR TO EXTENDED MEMORY
026732  005737  001202          2$:     TST     $PASS           ;TEST PASS COUNTER
026736  001413                          BEQ     TST155          ;;BR IF FIRST PASS
026740  032777  004000  152172          BIT     #SW11,@SWR      ;TEST IF INHIBIT INTER.
026746  001007                          BNE     TST155          ;;BR IF SET
026750  062737  000200  041744          ADD     #200,OUT        ;UPDATE BANK VALUE
026756  023737  036406  041744          CMP     $LSTBK,OUT      ;TEST IF DONE
026764  101254                          BHI     1$              ;BR IF NOT
```

```
                                        ;;*****************************************************************
                                        ;*TEST 155      ONE LIST DATA TRANSFER TO EACH 4K EXTENDED MEMORY
                                        ;;*****************************************************************
026766  000004                          TST155: SCOPE
026770  012737  000002  001160                  MOV     #2,$TIMES       ;;DO 2 ITERATIONS
026776  005737  036124                          TST     $KT11           ;TEST IF KT-11 INSTALLED
027002  100130                          BPL     TST156          ;;BR IF NONE
                                        ;DO A 1 WORD MODE TRANSFER TO A 1 WORD BUFFER IN EACH 4K EXTENDED MEMORY BANK
027004  012737  000600  041744          MOV     #600,OUT        ;LOAD INITIAL BANK VALUE <60000>
027012  013737  041744  172346  1$:     MOV     OUT,@#KIPAR3    ;LOAD KT-PAR REG #3 <BUFFER IS AT LOC 60000>
027020  012700  060000                  MOV     #BUF0,R0        ;LOAD BUFFER POINTER
027024  012710  052525                  MOV     #52525,(R0)     ;LOAD BAD TARGFT LOC. IF EXT ADD FAILS
027030  052737  001400  177572          BIS     #1400,@#SR0     ;ENABLE MAINT. MODE KT-11
027036  012710  125252                  MOV     #125252,(R0)    ;PRESET THE TARGET LOCATION
027042  042737  001400  177572          BIC     #1400,@#SR0     ;DISABLE MAINT. MODE KT-11
027050  013701  041744                  MOV     OUT,R1          ;GET BANK VALUE
027054  005002                          CLR     R2              ;CLEAR EXT. ADD TEMPORARY
027056  006301                          ASL     R1
027060  006301                          ASL     R1
027062  006301                          ASL     R1
027064  006301                          ASL     R1
027066  006301                          ASL     R1
027070  006102                          ROL     R2              ;SAVE EXT. ADDRESS BITS
027072  006301                          ASL     R1
027074  006102                          ROL     R2              ;SAVE EXT. ADDRESS BITS
                                        ;NOW GET READY TO DO THE TRANSFER
027076  012777  004000  152652          MOV     #CLRALL,@SFR    ;INIT THE NCV11
027104  010277  152640                  MOV     R2,@OFF         ;LOAD THE EXTENDED ADDRESS BITS
027110  012777  177777  152634          MOV     #-1,@WCR        ;LOAD WORD COUNT
027116  010177  152632                  MOV     R1,@BAR         ;LOAD BUS ADDRESS
027122  012777  000014  152626          MOV     #TSTDMA!TSICON,@SFR     ;ENABLE MAINT NCV11 MODE
027130  052777  000001  152610          BIS     #BIT0,@CSR      ;ENABLE THE NCV11
027136  052777  000002  152612          BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
027144  042777  000002  152604          BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
027152  010137  001122                  MOV     R1,$BDADR       ;SAVE TARGET ADDRESS
027156  010237  041730                  MOV     R2,NARROW       ;SAVE EA BITS FOR ERROR TYPEOUT
027162  052777  010000  152566          BIS     #BIT12,@SFR     ;ENABLE 1 BYTE TRANSFER
027170  012737  003407  001124          MOV     #3407,$GDDAT    ;LOAD EXPECTED DATA
027176  052737  000001  177572          BIS     #BIT0,@#SR0     ;ENABLE KT-11
027204  011037  001126                  MOV     (R0),$BDDAT     ;GET ACTUAL DATA
027210  042737  000001  177572          BIC     #BIT0,@#SR0     ;DISABLE KT-11
027216  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE DATA
027224  001401                          BEQ     2$              ;;BR IF SAME
027226  104035                          ERROR   35              ;LIST MODE DATA TRANSFER ERROR TO EXTENDED MEMORY
027230  005737  001202          2$:     TST     $PASS           ;TEST PASS COUNTER
027234  001413                          BEQ     TST156          ;;BR IF FIRST PASS
027236  032777  004000  151674          BIT     #SW11,@SWR      ;TEST INHIBIT INTER.
027244  001007                          BNE     TST156          ;;BR IF SET
027246  062737  000200  041744          ADD     #200,OUT        ;UPDATE BANK VALUE
027254  023737  036406  041744          CMP     $LSTBK,OUT      ;TEST IF DONE
027262  101253                          BHI     1$              ;BR IF NOT
```

```
                                        ;;*****************************************************************
                                        ;*TEST 156      VERIFY BIT 15 MATRIX ADDER INPUT
                                        ;;*****************************************************************
                                        TST156: SCOPE
027266  012737  000100  001160                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                        ;ADM OUTPUT     =       127657
                                        ;OFFSET         =       140000
                                        ;TARGET         =       267657
027274  023727  036406  003140                CMP     $LSTBK,#3140    ;TEST IF >52K IS AVAILABLE
027302  103476                                BLO     TST157          ;;BR IF NO MORE MEMORY
027304  012737  002676  172346                MOV     #2676,@#KIPAR3  ;LOAD KT-PAR REG #3 <BUFFER IS AT LOC 60000>
027312  012700  060056                        MOV     #BUF0+56,R0     ;LOAD BUFFER POINTER
027316  012710  125252                        MOV     #125252,(R0)    ;LOAD BAD TARGET LOC. IF EXT ADD FAILS
027322  052737  001400  177572                BIS     #1400,@#SR0     ;ENABLE MAINT. MODE KT-11
027330  005010                                CLR     (R0)            ;LOAD TARGET LOCATION DATA VALUE
027332  042737  001400  177572                BIC     #1400,@#SR0     ;DISABLE MAINT. MODE KT-11

                                        ;NOW GET READY TO DO THE TRANSFER
027340  012777  004000  152410                MOV     #CLRALL,@SFR    ;INIT THE NCV11
027346  012777  140000  152374                MOV     #140000,@OFF    ;LOAD COMBINED BUFFER ADDRESS
027354  012777  002036  152364                MOV     #2036,@CSR      ;ENABLE THE NCV11
027362  012777  000014  152366                MOV     #TSTDMA!TSTCON,@SFR     ,ENABLE MAINT NCV11 MODE
027370  052777  000001  152350                BIS     #BIT0,@CSR      ;ENABLE THE NCV11
027376  052777  000002  152352                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1)    027404  042777  000002  152344        BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
027412  000240                                NOP
027414  000240                                NOP
027416  052777  010000  152332                BIS     #BIT12,@SFR     ;ENABLE 1 BYTE TRANSFER
027424  000240                                NOP
027426  000240                                NOP
027430  000240                                NOP
027432  012737  000001  041730                MOV     #1,NARROW       ;SAVE EA BITS FOR ERROR TYPEOUT
027440  012737  000400  001124                MOV     #400,$GDDAT     ;LOAD EXPECTED DATA
027446  052737  000001  177572                BIS     #BIT0,@#SR0     ;ENABLE KT-11
027454  011037  001126                        MOV     (R0),$BDDAT     ;SAVE ACTUAL DATA
027460  042737  000001  177572                BIC     #BIT0,@#SR0     ;DISABLE KT-11
027466  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE DATA
027474  001401                                BEQ     TST157          ;;BR IF SAME
027476  104020                                ERROR   20              ;BIT 15 INPUT TO MATRIX MODE ADDER FAILED
```

```
2214                                    ;;*****************************************************************
 (3)                                    ;*TEST 157      VERIFY HIGH BYTE OPERATION OF THE "TEST X"
 (3)                                    ;;*****************************************************************
 (2) 027500  000004                     TST157: SCOPE
 (1) 027502  012737  000040  001160             MOV     #40,$TIMES      ;;DO 40 ITERATIONS
2215                                    ;TEST X         =       1
2216                                    ;OFFSET         =100000
2217                                    ;ADM OUTPUT     =127657
2218                                    ;TARGET         =100257
2219  027510  012777  004000  152240             MOV     #CLRALL,@SFR    ;CLEAR THE DEVICE
2220  027516  023727  036406  001140             CMP     $LSTBK,#1140    ;TEST IF LEAST 20K
2221  027524  103445                           BLO     TST160          ;;BR IF NOT
2222  027526  012777  100000  152214             MOV     #100000,@OFF    ;SET BIT 15 OF OFFSET
2223  027534  012777  000034  152214             MOV     #TSTDMA!TSTCON!BIT4,@SFR        ;SET TEST DMA, CONTROL AND TESTX
2224  027542  012777  002036  152176             MOV     #2036,@CSR      ;ENABLE THE NCV11
2225  027550  052777  000001  152170             BIS     #BIT0,@CSR      ;ENABLE THE NCV11
2226  027556  052777  000002  152172             BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
 (1) 027564  042777  000002  152164             BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
2227  027572  005037  100256                     CLR     @#BUF0+20256    ;CLEAR THE TARGET LOC.
2228  027576  052777  010000  152152             BIS     #BIT12,@SFR     ;ALLOW 1 DMA TRANSFER
2229  027604  012737  100256  001122             MOV     #BUF0+20256,$BDADR      ;LOAD EXPECTED ADDRESSES VALUE
2230  027612  012737  000400  001124             MOV     #400,$GDDAT     ;LOAD EXPECTED VALUE READ
2231  027620  017737  151276  001126             MOV     @$BDADR,$BDDAT  ;READ ACTUAL DATA
2232  027626  023737  001124  001126             CMP     $GDDAT,$BDDAT   ;COMPARE
2233  027634  001401                           BEQ     TST160          ;;BR IF SAME
2234  027636  104034                           ERROR   34              ;MATRIX ADDER INPUT OF ADM15 AND TESTX L
2235                                                           ;FAILED TO BE INHIBITED
```

```
                                        ;;*****************************************************************
                                        ;*TEST 160      VERIFY BIT 16 INPUT TO THE MATRIX MODE ADDER
                                        ;;*****************************************************************
027640  000004                          TST160: SCOPE
027642  012737  000100  001160                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                        ;ADM OUTPUT     =       253527
                                        ;OFFSET         =       210000
                                        ;TARGET         =       463527
027650  023727  036406  005140                CMP     $LSTBK,#5140    ;TEST IF ENOUGH MEMORY >84K
027656  103500                                BLO     TST161          ;;BR IF NOT
027660  012737  004635  172346                MOV     #4635,@#KIPAR3  ;LOAD KT-PAR REG #3 <BUFFER IS AT LOC 60000>
027666  012700  060026                        MOV     #BUF0+26,R0     ;LOAD BUFFER POINTER
027672  012710  125252                        MOV     #125252,(R0)    ;LOAD BAD TARGET LOC.
027676  052737  001400  177572                BIS     #1400,@#SR0     ;ENABLE MAINT. MODE KT-11
027704  005010                                CLR     (R0)            ;PRESET THE TARGET LOCATION
027706  042737  001400  177572                BIC     #1400,@#SR0     ;DISABLE MAINT. MODE KT-11

                                        ;NOW GET READY TO DO THE TRANSFER
027714  012777  004000  152034                MOV     #CLRALL,@SFR    ;INIT THE NCV11
027722  012777  010001  152020                MOV     #10001,@OFF     ;LOAD THE EXTENDED ADDRESS BITS
027730  012777  000014  152020                MOV     #TSTDMA!TSTCON,@SFR     ;ENABLE MAINT NCV11 MODE
027736  012777  004036  152002                MOV     #4036,@CSR      ;LOAD MATRIX MODE AND ZB ENABLE
027744  052777  000001  151774                BIS     #BIT0,@CSR      ;ENABLE THE NCV11
027752  052777  000002  151776                BIS     #TESTZ,@SFR             ;ENABLE "TEST Z" PULSES
027760  042777  000002  151770                BIC     #TESTZ,@SFR             ;DISABLE "TEST Z" PULSES
027766  000240                                NOP
027770  000240                                NOP
027772  052777  010000  151756                BIS     #BIT12,@SFR     ;ENABLE 1 BYTE TRANSFER
030000  000240                                NOP
030002  000240                                NOP
030004  000240                                NOP
030006  012737  000002  041730                MOV     #2,NARROW       ;SAVE EA BITS FOR ERROR TYPEOUT
030014  012737  000400  001124                MOV     #400,$GDDAT     ;LOAD EXPECTED DATA
030022  052737  000001  177572                BIS     #BIT0,@#SR0     ;ENABLE KT-11
030030  111037  001126                        MOVB    (R0),$BDDAT     ;SAVE ACTUAL DATA
030034  042737  000001  177572                BIC     #BIT0,@#SR0     ;DISABLE KT-11
030042  105037  001126                        CLRB    $BDDAT          ;MASK OFF LOW BYTE
030046  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE DATA
030054  001401                                BEQ     TST161          ;;BR IF SAME
030056  104020                                ERROR   20              ;BIT 16 INPUT TO MATRIX MODE ADDER FAILED
```

```
                                        ;;*****************************************************************
                                        ;*TEST 161      DETERMINE IF DIFLIN IS TO BE RUN (F)
                                        ;;*****************************************************************
030060  000004                  TST161: SCOPE
030062  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
030070  005737  050024                  TST     RUNDIF          ;TEST IF DIFLIN IS TO BE RUN
030074  001402                          BEQ     $EOP            ;;BR IF NOT TO BE RUN
030076  000137  043550                  JMP     DIFLIN          ;JUMP AND RUN DIFLIN
                                .SBTTL  END OF PASS ROUTINE
                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO LOGIC
030102                          $EOP:
030102  000004                          SCOPE
030104  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
030110  005037  001160                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
030114  005237  001202                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
030120  042737  100000  001202          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
030126  005327                          DEC     (PC)+           ;;LOOP?
030130  000001                  $EOPCT: .WORD   1
030132  003022                          BGT     $DOAGN          ;;YES
030134  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
030136  000001                  $ENDCT: .WORD   1
030140  030130                          $EOPCT
030142  104401  030207                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
030146  013746  001202                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
030152  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
030154  104401  030204                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
030160  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
030164  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
030166  000005                          RESET                   ;;CLEAR THE WORLD
030170  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
030172  000240                          NOP                     ;;SAVE ROOM
030174  000240                          NOP                     ;;FOR
030176  000240                          NOP                     ;;ACT11
030200                          $DOAGN:
030200  000137                          JMP     @(PC)+          ;;RETURN
030202  003360                  $RTNAD: .WORD   LOGIC
030204     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
030207     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
030214  050040  051501  020123
030222  000043
```

```
2280                                                    .SBTTL  ERROR ASCII MESSAGES
2281
2282  030224  034115  031060  004466  EM1:    .ASCIZ  /M8026  NCV11 BUS ADDRESS TIMEOUT/
      030232  041516  030526  020061
      030240  052502  020123  042101
      030246  051104  051505  020123
      030254  044524  042515  052517
      030262  000124
2283  030264  034115  031060  004466  EM2:    .ASCIZ  /M8026  COMMAND-STATUS REGISTER ERROR/
      030272  047503  046515  047101
      030300  026504  052123  052101
      030306  051525  051040  043505
      030314  051511  042524  020122
      030322  051105  047522  000122
2284  030330  034115  031060  004466  EM3:    .ASCIZ  /M8026  SPECIAL FUNCTION REGISTER ERROR/
      030336  050123  041505  040511
      030344  020114  052506  041516
      030352  044524  047117  051040
      030360  043505  051511  042524
      030366  020122  051105  047522
      030374  000122
2285  030376  034115  031060  004466  EM4:    .ASCIZ  /M8026  WORD COUNT REGISTER ERROR/
      030404  047527  042122  041440
      030412  052517  052116  051040
      030420  043505  051511  042524
      030426  020122  051105  047522
      030434  000122
2286  030436  034115  031060  004466  EM5:    .ASCIZ  /M8026  BUS ADDRESS REGISTER ERROR/
      030444  052502  020123  042101
      030452  051104  051505  020123
      030460  042522  044507  052123
      030466  051105  042440  051122
      030474  051117     000
2287  030477     115  030070  033062  EM6:    .ASCIZ  /M8026  OFFSET REGISTER ERROR/
      030504  047411  043106  042523
      030512  020124  042522  044507
      030520  052123  051105  042440
      030526  051122  051117     000
2288  030533     115  030070  033062  EM7:    .ASCIZ  /M8026  DUAL REGISTER SELECTION ERROR/
      030540  042011  040525  020114
      030546  042522  044507  052123
      030554  051105  051440  046105
      030562  041505  044524  047117
      030570  042440  051122  051117
      030576     000
2289  030577     115  030070  033062  EM10:   .ASCIZ  /M8026-M8036    LOW 16 BIT Z COUNT ERROR/
      030604  046455  030070  033063
      030612  046011  053517  030440
      030620  020066  044502  020124
      030626  020132  047503  047125
      030634  020124  051105  047522
      030642  000122
2290  030644  034115  031060  026466  EM11:   .ASCIZ  /M8026-M8036    HIGH 16 BIT Z COUNT ERROR/
      030652  034115  031460  004466
      030660  044510  044107  030440
      030666  020066  044502  020124
```

```
                030674  020132  047503  047125
                030702  020124  051105  047522
                030710  000122
  2291          030712  034115  031060  004466  EM12:   .ASCIZ  /M8026  Z COUNT STATUS ERROR/
                030720  020132  047503  047125
                030726  020124  052123  052101
                030734  051525  042440  051122
                030742  051117     000
  2292          030745     115  030070  033062  EM13:   .ASCIZ  /M8026  Z COUNT INTERRUPT ERROR/
                030752  055011  041440  052517
                030760  052116  044440  052116
                030766  051105  052522  052120
                030774  042440  051122  051117
                031002     000
  2293          031003     115  030070  033063  EM14:   .ASCIZ  /M8036  JOYSTICK STATUS ERROR/
                031010  045011  054517  052123
                031016  041511  020113  052123
                031024  052101  051525  042440
                031032  051122  051117     000
  2294          031037     115  030070  033063  EM15:   .ASCIZ  /M8036  JOYSTICK DATA ERROR/
                031044  045011  054517  052123
                031052  041511  020113  040504
                031060  040524  042440  051122
                031066  051117     000
  2295          031071     115  030070  033063  FM16:   .ASCIZ  /M8036  DATA INCREMENT ERROR/
                031076  042011  052101  020101
                031104  047111  051103  046505
                031112  047105  020124  051105
                031120  047522  000122
  2296          031124  034115  031460  004466  EM17:   .ASCIZ  /M8036  DATA DECREMENT ERROR/
                031132  040504  040524  042040
                031140  041505  042522  042515
                031146  052116  042440  051122
                031154  051117     000
  2297          031157     115  030070  033062  EM20:   .ASCIZ  /M8026-M8036    MATRIX MODE ADDRESS MAKER DATA ERROR/
                031164  046455  030070  033063
                031172  046411  052101  044522
                031200  020130  047515  042504
                031206  040440  042104  042522
                031214  051523  046440  045501
                031222  051105  042040  052101
                031230  020101  051105  047522
                031236  000122
  2298          031240  034115  031060  004466  EM21:   .ASCIZ  /M8026  LIST MODE ADDRESS MAKER DATA ERROR/
                031246  044514  052123  046440
                031254  042117  020105  042101
                031262  051104  051505  020123
                031270  040515  042513  020122
                031276  040504  040524  042440
                031304  051122  051117     009
  2299          031311     115  030070  033062  EM22:   .ASCIZ  /M8026  LIST MODE TRANSFER BUS ADDRESS ERROR/
                031316  046011  051511  020124
                031324  047515  042504  052040
                031332  040522  051516  042506
                031340  020122  052502  020123
                031346  042101  051104  051505
```

```
        031354  020123  051105  047522
        031362  000122
  2300  031364  034115  031060  004466  EM23:   .ASCIZ  /M8026  LIST MODE TRANSFER WORD COUNT ERROR/
        031372  044514  052123  046440
        031400  042117  020105  051124
        031406  047101  043123  051105
        031414  053440  051117  020104
        031422  047503  047125  020124
        031430  051105  047522  000122
  2301  031436  034115  031060  004466  EM24:   .ASCIZ  /M8026  LIST MODE TRANSFER OFFSET ERROR/
        031444  044514  052123  046440
        031452  042117  020105  051124
        031460  047101  043123  051105
        031466  047440  043106  042523
        031474  020124  051105  047522
        031502  000122
  2302  031504  034115  031060  004466  EM25:   .ASCIZ  /M8026  TIMEOUT STATUS ERROR/
        031512  044524  042515  052517
        031520  020124  052123  052101
        031526  051525  042440  051122
        031534  051117     000
  2303  031537     115  030070  033062  EM26:   .ASCIZ  /M8026  TIMEOUT INTERRUPT ERROR/
        031544  052011  046511  047505
        031552  052125  044440  052116
        031560  051105  052522  052120
        031566  042440  051122  051117
        031574     000
  2304  031575     115  030070  033062  EM27:   .ASCIZ  /M8026  SET ''EVENT'' OR ''TIME'' DATA ERROR/
        031602  051411  052105  021040
        031610  053105  047105  021124
        031616  047440  020122  052042
        031624  046511  021105  042040
        031632  052101  020101  051105
        031640  047522  000122
  2305  031644  034115  031060  004466  EM30:   .ASCIZ  /M8026  CELL INCREMENT DATA ERROR/
        031652  042503  046114  044440
        031660  041516  042522  042515
        031666  052116  042040  052101
        031674  020101  051105  047522
        031702  000122
  2306  031704  034115  031060  004466  EM31:   .ASCIZ  /M8026  CELL OVERFLOW STATUS ERROR/
        031712  042503  046114  047440
        031720  042526  043122  047514
        031726  020127  052123  052101
        031734  051525  042440  051122
        031742  051117     000
  2307  031745     115  030070  033062  EM32:   .ASCIZ  /M8026  CELL OVERFLOW INTERRUPT ERROR/
        031752  041411  046105  020114
        031760  053117  051105  046106
        031766  053517  044440  052116
        031774  051105  052522  052120
        032002  042440  051122  051117
        032010     000
  2308  032011     115  030070  033062  EM33:   .ASCIZ  /M8026  MATRIX MODE ADDRESS MUX ERROR/
        032016  046411  052101  044522
        032024  020130  047515  042504
```

```
                032032  040440  042104  042522
                032040  051523  046440  054125
                032046  042440  051122  051117
                032054     000
 2309           032055     115  030070  033062  EM34:   .ASCIZ  /M8026  "TESTX" FUNCTION ERROR/
                032062  021011  042524  052123
                032070  021130  043040  047125
                032076  052103  047511  020116
                032104  051105  047522  000122
 2310           032112  034115  031060  004466  EM35:   .ASCIZ  /M8026  DATA ERROR WHEN TRANSFERING TO EXTENDED MEMORY/
                032120  040504  040524  042440
                032126  051122  051117  053440
                032134  042520  020116  051124
                032142  047101  043123  051105
                032150  047111  020107  047524
                032156  042440  052130  047105
                032164  042504  020104  042515
                032172  047515  054522     000
 2311           032177     115  030070  033062  EM36:   .ASCIZ  /M8026  LIST MODE TRANSFER STATUS ERROR/
                032204  046011  051511  020124
                032212  047515  042504  052040
                032220  040522  051516  042506
                032226  020122  052123  052101
                032234  051525  042440  051122
                032242  051117     000
 2312           032245     115  030070  033062  EM37:   .ASCIZ  /M8026  LIST MODE TRANSFER DATA ERROR/
                032252  046011  051511  020124
                032260  047515  042504  052040
                032266  040522  051516  042506
                032274  020122  040504  040524
                032302  042440  051122  051117
                032310     000
 2313           032311     112  046525  042520  EM40:   .ASCIZ  /JUMPER-M8026-M7952     "EVENT" OR "TIME" MARK ERROR/
                032316  026522  034115  031060
                032324  026466  033515  032471
                032332  004462  042442  042526
                032340  052116  020042  051117
                032346  021040  044524  042515
                032354  020042  040515  045522
                032362  042440  051122  051117
                032370     000
 2314           032371     115  034467  031065  EM41:   .ASCIZ  /M7952  CLOCK BUS ADDRESS TIMEOUT/
                032376  041411  047514  045503
                032404  041040  051525  040440
                032412  042104  042522  051523
                032420  052040  046511  047505
                032426  052125     000
 2315           032431     115  031070  033461  EM42:   .ASCIZ  /M8217  INCORRECT INTERRUPT LEVEL/
                032436  044411  041516  051117
                032444  042522  052103  044440
                032452  052116  051105  052522
                032460  052120  046040  053105
                032466  046105     000
 2316
 2317           032471     105  051122  041520  DH1:    .ASCIZ  /ERRPC  ADDR/
                032476  040411  042104  000122
```

```
 2318  032504  051105  050122  004503  DH2:    .ASCIZ  /ERRPC  ADDR    GOOD    BAD/
       032512  042101  051104  043411
       032520  047517  004504  040502
       032526  000104
 2319  032530  051105  050122  004503  DH3:    .ASCIZ  /ERRPC  ADDR    GOOD    BAD     BADADR/
       032536  042101  051104  043411
       032544  047517  004504  040502
       032552  004504  040502  040504
       032560  051104     000
 2320  032563     105  051122  041520  DH4:    .ASCIZ  /ERRPC  ADDR    GOOD    BAD     EA ADR- BADADR/
       032570  040411  042104  004522
       032576  047507  042117  041011
       032604  042101  042411  020101
       032612  042101  004522  040502
       032620  040504  051104     000
 2321  032625     015     012     007  OUTRNG: .BYTE   15,12,7
 2322  032630  047516  021040  021132          .ASCII  /NO "Z" PULSES OR /
       032636  050040  046125  042523
       032644  020123  051117     040
 2323  032651     111  050116  052125          .ASCII  /INPUT VOLTAGE OUT OF RANGE ON CHANNEL #/
       032656  053040  046117  040524
       032664  042507  047440  052125
       032672  047440  020106  040522
       032700  043516  020105  047117
       032706  041440  040510  047116
       032714  046105  021440
 2324  032720     060     015     012  OUTCHN: .BYTE   60,15,12,7,0
       032723     007     000
 2325          032726                       .EVEN
 2326  032726  001116  001250  000000  DT1:    .WORD   $ERRPC,$BASE,0
 2327  032734  001116  001250  001124  DT2:    .WORD   $ERRPC,$BASE,$GDDAT,$BDDAT,0
       032742  001126  000000
 2328  032746  001116  001250  001124  DT3:    .WORD   $ERRPC,$BASE,$GDDAT,$BDDAT,$BDADR,0
       032754  001126  001122  000000
 2329  032762  001116  001250  001124  DT4:    .WORD   $ERRPC,$BASE,$GDDAT,$BDDAT,NARROW,$BDADR,0
       032770  001126  041730  001122
       032776  000000
 2330  033000  001116  001254  000000  DT5:    .WORD   $ERRPC,$CDW1,0
 2331  033006  000000                  DF0:    .WORD   0
```

```
2336                                    .SBTTL  TTY INPUT ROUTINE
 (1)
 (2)                                    ;;*********************************************************************
 (1)                                    .ENABL  LSB
 (1)
 (2)                                    ;;*********************************************************************
 (1)                                    ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 (1)                                    ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
 (1)                                    ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
 (1)                                    ;*WHEN OPERATING IN TTY FLAG MODE.
 (1)  033010  022737  000176  001140    $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
 (1)  033016  001074                            BNE     15$             ;;BRANCH IF NO
 (1)  033020  105777  146120                    TSTB    @$TKS           ;;CHAR THERE?
 (1)  033024  100071                            BPL     15$             ;;IF NO, DON'T WAIT AROUND
 (1)  033026  117746  146114                    MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
 (1)  033032  042716  177600                    BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
 (1)  033036  022726  000007                    CMP     #7,(SP)+        ;;IS IT A CONTROL G?
 (1)  033042  001062                            BNE     15$             ;;NO, RETURN TO USER
 (1)  033044  123727  001134  000001            CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 (1)  033052  001456                            BEQ     15$             ;;BRANCH IF YES
 (1)
 (1)  033054  104401  033535                    TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
 (1)  033060  104401  033542            $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 (2)  033064  013746  000176                    MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 (2)  033070  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 (1)  033072  104401  033553                    TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 (1)  033076  005046                    19$:    CLR     -(SP)           ;;CLEAR COUNTER
 (1)  033100  005046                            CLR     -(SP)           ;;THE NEW SWR
 (1)  033102  105777  146036            7$:     TSTB    @$TKS           ;;CHAR THERE?
 (1)  033106  100375                            BPL     7$              ;;IF NOT TRY AGAIN
 (1)
 (1)  033110  117746  146032                    MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 (1)  033114  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 (1)
 (1)
 (1)
 (1)  033120  021627  000025            9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
 (1)  033124  001005                            BNE     10$             ;;BRANCH IF NOT
 (1)  033126  104401  033530                    TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
 (1)  033132  062706  000006            20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 (1)  033136  000757                            BR      19$             ;;LET'S TRY IT AGAIN
 (1)
 (1)
 (1)  033140  021627  000015            10$:    CMP     (SP),#15        ;;IS IT A <CR>?
 (1)  033144  001022                            BNE     16$             ;;BRANCH IF NO
 (1)  033146  005766  000004                    TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 (1)  033152  001403                            BEQ     11$             ;;BRANCH IF YES
 (1)  033154  016677  000002  145756            MOV     2(SP),@SWR      ;;SAVE NEW SWR
 (1)  033162  062706  000006            11$:    ADD     #6,SP           ;;CLEAR UP STACK
 (1)  033166  104401  001171            14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 (1)  033172  123727  001135  000001            CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 (1)  033200  001003                            BNE     15$             ;;BRANCH IF NOT
 (1)  033202  012777  000100  145734            MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 (1)  033210  000002                    15$:    RTI                     ;;RETURN
 (1)  033212  004737  034610            16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 (1)  033216  021627  000060                    CMP     (SP),#60        ;;CHAR < 0?
```

```
(1)  033222  002420                        BLT     18$             ;;BRANCH IF YES
(1)  033224  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
(1)  033230  003015                        BGT     18$             ;;BRANCH IF YES
(1)  033232  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
(1)  033236  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
(1)  033242  001403                        BEQ     17$             ;;BRANCH IF YES
(1)  033244  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
(1)  033246  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
(1)  033250  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
(1)  033252  005266  000002          17$:  INC     2(SP)           ;;KEEP COUNT OF CHAR
(1)  033256  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
(1)  033262  000707                        BR      7$              ;;GET THE NEXT ONE
(1)  033264  104401  001170          18$:  TYPE    ,$QUES          ;;TYPE ?<CR><LF>
(1)  033270  000720                        BR      20$             ;;SIMULATE CONTROL-U
(1)                                 .DSABL  LSB
(1)
(1)
(2)                                 ;;*****************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                 ;*                              ;;WITH PARITY BIT STRIPPED OFF
(1)                                 ;
(1)
(1)  033272  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  033274  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  033302  105777  145636          1$:   TSTB    @$TKS           ;;WAIT FOR
(1)  033306  100375                        BPL     1$              ;;A CHARACTER
(1)  033310  117766  145632  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
(1)  033316  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  033324  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
(1)  033332  001013                        BNE     3$              ;;BRANCH IF NO
(1)  033334  105777  145604          2$:   TSTB    @$TKS           ;;WAIT FOR A CHARACTER
(1)  033340  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
(1)  033342  117746  145600                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
(1)  033346  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
(1)  033352  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
(1)  033356  001366                        BNE     2$              ;;IF NOT DISCARD IT
(1)  033360  000750                        BR      1$              ;;YES, RESUME
(1)  033362  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
(1)  033370  002407                        BLT     4$              ;;BRANCH IF YES
(1)  033372  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
(1)  033400  003003                        BGT     4$              ;;BRANCH IF YES
(1)  033402  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
(1)  033410  000002                  4$:   RTI                     ;;GO BACK TO USER
(2)                                 ;;*****************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                 ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)
(1)  033412  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  033414  012703  033520          1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  033420  022703  033530          2$:   CMP     #$TTYIN+80.,R3  ;;BUFFER FULL?
```

```
(1)  033424  101405                        BLOS    4$              ;;BR IF YES
(1)  033426  104410                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  033430  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  033432  122713  000177          10$:  CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  033436  001003                        BNE     3$              ;;SKIP IF NOT
(1)  033440  104401  001170          4$:   TYPE    ,$QUES          ;;TYPE A '?'
(1)  033444  000763                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  033446  111337  033516          3$:   MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  033452  104401  033516                TYPE    ,9$
(1)  033456  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  033462  001356                        BNE     2$              ;;LOOP IF NOT RETURN
(1)  033464  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  033470  104401  001172                TYPE    ,$LF            ;;TYPE A LINE FEED
(1)  033474  012603                        MOV     (SP)+,R3        ;;RESTORE R3
(1)  033476  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  033500  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  033506  012766  033520  000004        MOV     #$TTYIN,4(SP)
(1)  033514  000002                        RTI                     ;;RETURN
(1)  033516     000                 9$:    .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  033517     000                        .BYTE   0               ;;TERMINATOR
(1)  033520  000010                $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
(1)  033530  052536  005015  000   $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
(1)  033535     136  006507  000012$CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
(1)  033542  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
(1)  033550  020075     000
(1)  033553     040  047040  053505 $MNEW:  .ASCIZ  /  NEW = /
(1)  033560  036440  000040
```

```
2338                                    .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 (1)                                    ;*CHANGE IT TO BINARY.
 (1)                                    ;*CALL:
 (1)                                    ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 (1)                                    ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 (1)                                    ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 (1)
 (1)  033564  011646                    $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 (1)  033566  016666  000004  000002            MOV     4(SP),2(SP)     ;;INPUT NUMBER
 (3)  033574  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 (3)  033576  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 (3)  033600  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 (1)  033602  104411                    1$:     RDLIN                   ;;READ AN ASCIZ LINE
 (1)  033604  012600                            MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 (1)  033606  005001                            CLR     R1              ;;CLEAR DATA WORD
 (1)  033610  005002                            CLR     R2
 (1)  033612  112046                    2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 (1)  033614  001412                            BEQ     3$              ;;IF ZERO GET OUT
 (1)  033616  006301                            ASL     R1              ;;*2
 (1)  033620  006102                            ROL     R2
 (1)  033622  006301                            ASL     R1              ;;*4
 (1)  033624  006102                            ROL     R2
 (1)  033626  006301                            ASL     R1              ;;*8
 (1)  033630  006102                            ROL     R2
 (1)  033632  042716  177770                    BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
 (1)  033636  062601                            ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
 (1)  033640  000764                            BR      2$              ;;LOOP
 (1)  033642  005726                    3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
 (1)  033644  010166  000012                    MOV     R1,12(SP)       ;;SAVE THE RESULT
 (1)  033650  010237  033664                    MOV     R2,$HIOCT
 (3)  033654  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
 (3)  033656  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 (3)  033660  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
 (1)  033662  000002                            RTI                     ;;RETURN
 (1)  033664  000000                    $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
2340                                    .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 (1)                                    ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 (1)                                    ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 (1)                                    ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 (1)                                    ;*REPLACED WITH SPACES.
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 (1)                                    ;*      TYPDS                   ;;GO TO THE ROUTINE
 (1)
 (1)  033666                            $TYPDS:
 (3)  033666  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 (3)  033670  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 (3)  033672  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 (3)  033674  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 (3)  033676  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 (1)  033700  012746  020200                    MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 (1)  033704  016605  000020                    MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 (1)  033710  100004                            BPL     1$              ;;BR IF INPUT IS POS.
 (1)  033712  005405                            NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 (1)  033714  112766  000055  000001            MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 (1)  033722  005000                    1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 (1)  033724  012703  034102                    MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 (1)  033730  112723  000040                    MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 (1)  033734  005002                    2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 (1)  033736  016001  034072                    MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 (1)  033742  160105                    3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 (1)  033744  002402                            BLT     4$              ;;BR IF DONE
 (1)  033746  005202                            INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 (1)  033750  000774                            BR      3$
 (1)  033752  060105                    4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 (1)  033754  005702                            TST     R2              ;;CHECK IF BCD DIGIT=0
 (1)  033756  001002                            BNE     5$              ;;FALL THROUGH IF 0
 (1)  033760  105716                            TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 (1)  033762  100407                            BMI     7$              ;;BR IF YES
 (1)  033764  106316                    5$:     ASLB    (SP)            ;;MSD?
 (1)  033766  103003                            BCC     6$              ;;BR IF NO
 (1)  033770  116663  000001  177777            MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 (1)  033776  052702  000060            6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 (1)  034002  052702  000040            7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 (1)  034006  110223                            MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 (1)  034010  005720                            TST     (R0)+           ;;JUST INCREMENTING
 (1)  034012  020027  000010                    CMP     R0,#10          ;;CHECK THE TABLE INDEX
 (1)  034016  002746                            BLT     2$              ;;GO DO THE NEXT DIGIT
 (1)  034020  003002                            BGT     8$              ;;GO TO EXIT
 (1)  034022  010502                            MOV     R5,R2           ;;GET THE LSD
 (1)  034024  000764                            BR      6$              ;;GO CHANGE TO ASCII
 (1)  034026  105726                    8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 (1)  034030  100003                            BPL     9$              ;;BR IF NO
 (1)  034032  116663  177777  177776            MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 (1)  034040  105013                    9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 (3)  034042  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
 (3)  034044  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
 (3)  034046  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
(3)  034050  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  034052  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  034054  104401  034102                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  034060  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  034066  012616                        MOV     (SP)+,(SP)
(1)  034070  000002                        RTI                     ;;RETURN TO USER
(1)  034072  023420                $DTBL:  10000.
(1)  034074  001750                        1000.
(1)  034076  000144                        100.
(1)  034100  000012                        10.
(1)  034102  000004                $DBLK:  .BLKW   4
2341                               .SBTTL  SCOPE HANDLER ROUTINE
(1)
(2)                                ;;**********************************************************************
(1)                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1)                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1)                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(1)                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                ;*SW14=1        LOOP ON TEST
(1)                                ;*SW11=1        INHIBIT ITERATIONS
(1)                                ;*SW09=1        LOOP ON ERROR
(1)                                ;*SW08=1        LOOP ON TEST IN SWR<7:0>
(1)                                ;*CALL
(1)                                ;*      SCOPE           ;;SCOPE=IOT
(1)
(1)  034112                        $SCOPE:
(1)  034112  104407                        CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
(2)  034114  004737  045700                JSR     PC,CTRLCG               ;TEST FOR CTRL C OR G
(1)  034120  032777  040000  145012  1$:   BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
(1)  034126  001114                        BNE     $OVER           ;;YES IF SW14=1
(1)                                ;#####START OF CODE FOR THE XOR TESTER#####
(1)  034130  000416                $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
(1)                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
(1)  034132  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1)  034136  012737  034156  000004        MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
(1)  034144  005737  177060                TST     @#177060        ;;TIME OUT ON XOR?
(1)  034150  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  034154  000463                        BR      $SVLAD          ;;GO TO THE NEXT TEST
(1)  034156  022626                5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
(1)  034160  012637  000004                MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  034164  000423                        BR      7$              ;;LOOP ON THE PRESENT TEST
(1)  034166                        6$:;#####END OF CODE FOR THE XOR TESTER#####
(1)  034166  032777  000400  144744        BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
(1)  034174  001404                        BEQ     2$              ;;BR IF NO
(1)  034176  127737  144736  001102        CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
(1)  034204  001465                        BEQ     $OVER           ;;BR IF YES
(1)  034206  105737  001103        2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
(1)  034212  001421                        BEQ     3$              ;;BR IF NO
(1)  034214  123737  001115  001103        CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
(1)  034222  101015                        BHI     3$              ;;BR IF NO
(1)  034224  032777  001000  144706        BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
(1)  034232  001404                        BEQ     4$              ;;BR IF NO
(1)  034234  013737  001110  001106  7$:   MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
(1)  034242  000446                        BR      $OVER
(1)  034244  105037  001103        4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
(1)  034250  005037  001160                CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
```

```
(1)  034254  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
(1)  034256  032777  004000  144654  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
(1)  034264  001011                          BNE     1$              ;;BR IF YES
(1)  034266  005737  001202                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
(1)  034272  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
(1)  034274  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
(1)  034300  023737  001160  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
(1)  034306  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
(1)  034310  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
(1)  034316  013737  034374  001160          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
(1)  034324  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
(1)  034330  113737  001102  001200          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
(1)  034336  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
(1)  034342  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  034346  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  034352  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  034360  013777  001102  144554  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1)  034366  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  034372  000002                          RTI                     ;;FIXES PS
(1)  034374  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
2342                                 .SBTTL  TYPE ROUTINE
(1)
(2)                                  ;;**************************************************************************
(1)                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                  ;*
(1)                                  ;*CALL:
(1)                                  ;*1) USING A TRAP INSTRUCTION
(1)                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                  ;*OR
(1)                                  ;*      TYPE
(1)                                  ;*      MESADR
(1)                                  ;*
(1)
(1)  034376  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(1)  034402  100002                          BPL     1$              ;;BR IF YES
(1)  034404  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
(1)  034406  000430                          BR      3$              ;;LEAVE
(1)  034410  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
(1)  034412  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
(1)  034416  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  034424  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
(1)  034426  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
(1)  034434  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
(1)  034436  010037  034446                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
(1)  034442  004737  035450                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
(1)  034446  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
(1)  034450  132737  000040  001215  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
(1)  034456  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
(1)  034460  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  034462  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
(1)  034464  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
(1)  034466  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
```

```
(1)  034470  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
(1)  034474  000002                          RTI                     ;;RETURN
(1)  034476  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
(1)  034502  001430                          BEQ     8$
(1)  034504  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
(1)  034510  001006                          BNE     5$
(1)  034512  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
(1)  034514  104401                          TYPE                    ;;TYPE A CR AND LF
(1)  034516  001171                          $CRLF
(1)  034520  105037  034654                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
(1)  034524  000755                          BR      2$              ;;GET NEXT CHARACTER
(1)  034526  004737  034610          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
(1)  034532  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
(1)  034536  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
(1)  034540  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
(1)                                                                  ;;AND THE NULL CHAR.
(1)  034544  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
(1)  034550  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  034552  004737  034610                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
(1)  034556  105337  034654                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1)  034562  000770                          BR      7$              ;;LOOP
(1)
(1)                                  ;HORIZONTAL TAB PROCESSOR
(1)
(1)  034564  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
(1)  034570  004737  034610          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
(1)  034574  132737  000007  034654          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1)  034602  001372                          BNE     9$              ;;TAB STOP
(1)  034604  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
(1)  034606  000724                          BR      2$              ;;GET NEXT CHARACTER
(1)  034610  105777  144334          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  034614  100375                          BPL     $TYPEC
(1)  034616  116677  000002  144326          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  034624  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
(1)  034632  001003                          BNE     1$              ;;BRANCH IF NO
(1)  034634  105037  034654                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
(1)  034640  000406                          BR      $TYPEX          ;;EXIT
(1)  034642  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
(1)  034650  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
(1)  034652  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
(1)  034654  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
(1)  034656  000207          $TYPEX: RTS     PC
(1)
```

```
2344                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 (1)                                    ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 (1)                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
 (1)                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 (1)                                    ;*      .BYTE   M               ;;M=1 OR 0
 (1)                                    ;*                                      ;;1=TYPE LEADING ZEROS
 (1)                                    ;*                                      ;;0=SUPPRESS LEADING ZEROS
 (1)                                    ;*
 (1)                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 (1)                                    ;*$TYPOS OR $TYPOC
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
 (1)                                    ;*
 (1)                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
 (1)
 (1)  034660  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 (1)  034664  116637  000001  035103            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 (1)  034672  112637  035105                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 (1)  034676  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 (1)  034702  000406                            BR      $TYPON
 (1)  034704  112737  000001  035103    $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 (1)  034712  112737  000006  035105            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 (1)  034720  112737  000005  035102    $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 (1)  034726  010346                            MOV     R3,-(SP)        ;;SAVE R3
 (1)  034730  010446                            MOV     R4,-(SP)        ;;SAVE R4
 (1)  034732  010546                            MOV     R5,-(SP)        ;;SAVE R5
 (1)  034734  113704  035105                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 (1)  034740  005404                            NEG     R4
 (1)  034742  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 (1)  034746  110437  035104                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 (1)  034752  113704  035103                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 (1)  034756  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 (1)  034762  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
 (1)  034764  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 (1)  034766  000404                            BR      3$              ;;GO DO MSB
 (1)  034770  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
 (1)  034772  006105                            ROL     R5
 (1)  034774  006105                            ROL     R5
 (1)  034776  010503                            MOV     R5,R3
 (1)  035000  006103                    3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 (1)  035002  105337  035104                    DECB    $OMODE          ;;TYPE THIS DIGIT?
 (1)  035006  100016                            BPL     7$              ;;BR IF NO
 (1)  035010  042703  177770                    BIC     #177770,R3      ;;GET RID OF JUNK
 (1)  035014  001002                            BNE     4$              ;;TEST FOR 0
 (1)  035016  005704                            TST     R4              ;;SUPPRESS THIS 0?
 (1)  035020  001403                            BEQ     5$              ;;BR IF YES
```

```
(1)  035022  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  035024  052703  000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)  035030  052703  000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  035034  110337  035100                    MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  035040  104401  035100                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)  035044  105337  035102            7$:     DECB    $OCNT           ;;COUNT BY 1
(1)  035050  003347                            BGT     2$              ;;BR IF MORE TO DO
(1)  035052  002402                            BLT     6$              ;;BR IF DONE
(1)  035054  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  035056  000744                            BR      2$              ;;GO DO THE LAST DIGIT
(1)  035060  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
(1)  035062  012604                            MOV     (SP)+,R4        ;;RESTORE R4
(1)  035064  012603                            MOV     (SP)+,R3        ;;RESTORE R3
(1)  035066  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  035074  012616                            MOV     (SP)+,(SP)
(1)  035076  000002                            RTI                     ;;RETURN
(1)  035100     000                    8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  035101     000                            .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  035102     000                    $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  035103     000                    $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
(1)  035104  000000                    $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
2345
2346                                   .SBTTL  ERROR HANDLER ROUTINE
(1)
(2)                                    ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
(1)                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1)                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1)                                    ;*AND GO TO $ERRTYP ON ERROR
(1)                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                    ;*SW15=1        HALT ON ERROR
(1)                                    ;*SW13=1        INHIBIT ERROR TYPEOUTS
(1)                                    ;*SW10=1        BELL ON ERROR
(1)                                    ;*SW09=1        LOOP ON ERROR
(1)                                    ;*CALL
(1)                                    ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
(1)
(1)  035106                            $ERROR:
(1)  035106  104407                            CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
(1)  035110  105237  001103            7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
(1)  035114  001775                            BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
(1)  035116  013777  001102  144016            MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1)  035124  032777  002000  144006            BIT     #BIT10,@SWR     ;;BELL ON ERROR?
(1)  035132  001402                            BEQ     1$              ;;NO - SKIP
(1)  035134  104401  001164                    TYPE    ,$BELL          ;;RING BELL
(1)  035140  005237  001112            1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
(1)  035144  011637  001116                    MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
(1)  035150  162737  000002  001116            SUB     #2,$ERRPC
(1)  035156  117737  143734  001114            MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
(1)  035164  032777  020000  143746            BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
(1)  035172  001004                            BNE     20$             ;;SKIP TYPEOUTS
(1)  035174  004737  035306                    JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
(1)  035200  104401  001171                    TYPE    ,$CRLF
(1)  035204                            20$:
(1)  035204  122737  000001  001214            CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  035212  001007                            BNE     2$              ;;NO,SKIP APT ERROR REPORT
(1)  035214  113737  001114  035226            MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
```

```
(1)  035222  004737  035460            JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
(1)  035226     000            21$:    .BYTE   0
(1)  035227     000                    .BYTE   0
(1)  035230  000777            22$:    BR      22$             ;;APT ERROR LOOP
(1)  035232  005777  143702    2$:     TST     @SWR            ;;HALT ON ERROR
(1)  035236  100002                    BPL     3$              ;;SKIP IF CONTINUE
(1)  035240  000000                    HALT                    ;;HALT ON ERROR!
(1)  035242  104407                    CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
(1)  035244  032777  001000  143666  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
(1)  035252  001402                    BEQ     4$              ;;BR IF NO
(1)  035254  013716  001110            MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
(1)  035260  005737  001162    4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
(1)  035264  001402                    BEQ     5$              ;;BR IF NONE
(1)  035266  013716  001162            MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
(1)  035272            5$:
(1)  035272  022737  030170  000042            CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
(1)  035300  001001                    BNE     6$              ;;BRANCH IF NO
(1)  035302  000000                    HALT                    ;;YES
(1)  035304            6$:
(1)  035304  000002                    RTI                     ;;RETURN
2347
2348                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
(1)
(2)                                 ;;**************************************************************
(1)                                 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
(1)                                 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
(1)                                 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
(1)
(1)  035306            $ERRTYP:
(1)  035306  104401  001171            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  035312  010046                    MOV     R0,-(SP)        ;;SAVE R0
(1)  035314  005000                    CLR     R0              ;;PICKUP THE ITEM INDEX
(1)  035316  153700  001114            BISB    @#$ITEMB,R0
(1)  035322  001004                    BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
(1)                                                            ;;TYPE THE PC OF THE ERROR
(2)  035324  013746  001116            MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
(2)                                                            ;;ERROR ADDRESS
(2)  035330  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  035332  000426                    BR      6$              ;;GET OUT
(1)  035334  005300            1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
(1)  035336  006300                    ASL     R0              ;;      WORK FOR THE ERROR TABLE
(1)  035340  006300                    ASL     R0
(1)  035342  006300                    ASL     R0
(1)  035344  062700  001256            ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
(1)  035350  012037  035360            MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
(1)  035354  001404                    BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
(1)  035356  104401                    TYPE                    ;;TYPE THE "ERROR MESSAGE"
(1)  035360  000000            2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
(1)  035362  104401  001171            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  035366  012037  035376    3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
(1)  035372  001404                    BEQ     5$              ;;SKIP TYPEOUT IF 0
(1)  035374  104401                    TYPE                    ;;TYPE THE "DATA HEADER"
(1)  035376  000000            4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
(1)  035400  104401  001171            TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  035404  011000            5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
(1)  035406  001004                    BNE     7$              ;;GO TYPE THE DATA
```

```
(1)  035410  012600                6$:     MOV     (SP)+,R0        ;;RESTORE R0
(1)  035412  104401  001171                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  035416  000207                        RTS     PC              ;;RETURN
(1)  035420                         7$:
(2)  035420  013046                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
(2)  035422  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  035424  005710                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
(1)  035426  001770                        BEQ     6$              ;;BR IF NO
(1)  035430  104401  035436                TYPE    ,8$             ;;TYPE TWO(2) SPACES
(1)  035434  000771                        BR      7$              ;;LOOP
(1)  035436  020040     000        8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
(1)          035442                        .EVEN
```

```
2350                                    .SBTTL  APT COMMUNICATIONS ROUTINE
 (1)
 (2)                                    ;;*****************************************************************
 (1)  035442  112737  000001  035706  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 (1)  035450  112737  000001  035704  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 (1)  035456  000403                          BR      $ATYC
 (1)  035460  112737  000001  035706  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 (1)  035466                          $ATYC:
 (3)  035466  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 (3)  035470  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 (1)  035472  105737  035704                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 (1)  035476  001450                          BEQ     5$              ;;IF NOT:  BR
 (1)  035500  122737  000001  001214          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 (1)  035506  001031                          BNE     3$              ;;IF NOT:  BR
 (1)  035510  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 (1)  035516  001425                          BEQ     3$              ;;IF NOT:  BR
 (1)  035520  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 (1)  035524  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 (1)  035532  005737  001174          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 (1)  035536  001375                          BNE     1$              ;;IF NOT:  WAIT
 (1)  035540  010037  001210                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 (1)  035544  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 (1)  035546  001376                          BNE     2$
 (1)  035550  163700  001210                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 (1)  035554  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 (1)  035556  010037  001212                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 (1)  035562  012737  000004  001174          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 (1)  035570  000413                          BR      5$
 (1)  035572  017637  000004  035616  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 (1)  035600  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 (3)  035606  013746  177776                  MOV     177776 -(SP)    ;;PUSH 177776 ON STACK
 (1)  035612  004737  034376                  JSR     PC,$J           ;;CALL TYPE MACRO
 (1)  035616  000000                  4$:     .WORD   0
 (1)  035620                          5$:
 (1)  035620  105737  035706          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 (1)  035624  001416                          BEQ     12$             ;;IF NOT:  BR
 (1)  035626  005737  001214                  TST     $ENV            ;;RUNNING UNDER APT?
 (1)  035632  001413                          BEQ     12$             ;;IF NOT:  BR
 (1)  035634  005737  001174          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 (1)  035640  001375                          BNE     11$             ;;IF NOT:  WAIT
 (1)  035642  017637  000004  001176          MOV     @4(SP),$FATAL   ;;GET ERROR #
 (1)  035650  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 (1)  035656  005237  001174                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 (1)  035662  105037  035706          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 (1)  035666  105037  035705                  CLRB    $LFLG           ;;CLEAR LOG FLAG
 (1)  035672  105037  035704                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 (3)  035676  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 (3)  035700  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 (1)  035702  000207                          RTS     PC              ;;RETURN
 (1)  035704     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
 (1)  035705     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
 (1)  035706     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
 (1)          035710                          .EVEN
 (1)          000200                  APTSIZE=200
 (1)          000001                  APTENV=001
 (1)          000100                  APTSPOOL=100
```

```
  (1)           000040                   APTCSUP=040
 2351                                    .SBTTL  POWER DOWN AND UP ROUTINES
  (1)
  (2)                                     ;;*****************************************************************
  (1)                                     ;POWER DOWN ROUTINE
  (1)  035710  012737  036050  000024     $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
  (1)  035716  012737  000340  000026             MOV     #340,@#PWRVEC+2 ;;PRIO:7
  (3)  035724  010046                             MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  035726  010146                             MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  035730  010246                             MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  035732  010346                             MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (3)  035734  010446                             MOV     R4,-(SP)        ;;PUSH R4 ON STACK
  (3)  035736  010546                             MOV     R5,-(SP)        ;;PUSH R5 ON STACK
  (3)  035740  017746  143174                     MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
  (1)  035744  010637  036054                     MOV     SP,$SAVR6       ;;SAVE SP
  (1)  035750  012737  035762  000024             MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
  (1)  035756  000000                             HALT
  (1)  035760  000776                             BR      .-2             ;;HANG UP
  (1)
  (2)                                     ;;*****************************************************************
  (1)                                     ;POWER UP ROUTINE
  (1)  035762  012737  036050  000024     $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
  (1)  035770  013706  036054                     MOV     $SAVR6,SP       ;;GET SP
  (1)  035774  005037  036054                     CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
  (1)  036000  005237  036054             1$:     INC     $SAVR6          ;;WAIT FOR THE INC
  (1)  036004  001375                             BNE     1$              ;;OF  WORD
  (3)  036006  012677  143126                     MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
  (3)  036012  012605                             MOV     (SP)+,R5        ;;POP STACK INTO R5
  (3)  036014  012604                             MOV     (SP)+,R4        ;;POP STACK INTO R4
  (3)  036016  012603                             MOV     (SP)+,R3        ;;POP STACK INTO R3
  (3)  036020  012602                             MOV     (SP)+,R2        ;;POP STACK INTO R2
  (3)  036022  012601                             MOV     (SP)+,R1        ;;POP STACK INTO R1
  (3)  036024  012600                             MOV     (SP)+,R0        ;;POP STACK INTO R0
  (1)  036026  012737  035710  000024             MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
  (1)  036034  012737  000340  000026             MOV     #340,@#PWRVEC+2 ;;PRIO:7
  (1)  036042  104401                             TYPE                    ;REPORT THE POWER FAILURE
  (1)  036044  036056                     $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
  (1)  036046  000002                             RTI
  (1)  036050  000000                     $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
  (1)  036052  000776                             BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
  (1)  036054  000000                     $SAVR6: 0                       ;;PUT THE SP HERE
  (1)  036056  005015  047520  042527     $POWER: .ASCIZ  <15><12>"POWER"
  (1)  036064  000122
  (1)                                             .EVEN
```

```
2353                                    .SBTTL  ROUTINE TO SIZE MEMORY
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*CALL:
 (1)                                    ;*      JSR     PC,$SIZE
 (1)                                    ;*      RETURN
 (1)                                    ;*$LSTAD WILL CONTAIN:
 (1)                                    ;*      WITH KT11 OPTION        -- LAST VIRTUAL ADDRESS OF THE LAST BANK
 (1)                                    ;*      WITHOUT KT11 OPTION     -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
 (1)                                    ;*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
 (1)                                    ;*$KT11 IS THE MEMORY MANAGEMENT KEY
 (1)                                    ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
 (1)                                    ;*      MUST BE SETUP BEFORE THE CALL
 (1)                                    ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
 (1)                                    ;*      DETERMINED BY ROUTINE
 (1)
 (1)  036066  010046                    $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
 (1)  036070  010146                            MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
 (1)  036072  010246                            MOV     R2,-(SP)        ;;SAVE R2 ON THE STACK
 (1)  036074  010346                            MOV     R3,-(SP)        ;;SAVE R3 ON THE STACK
 (1)  036076  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
 (1)  036102  013746  000006                    MOV     @#ERRVEC+2,-(SP)
 (1)  036106  010600                            MOV     SP,R0           ;;SAVE THE STACK POINTER
 (1)                                    ;;SET THE ERRVEC PS TO THE PRESENT PS
 (2)  036110  104400                            TRAP                    ;;PUSH OLD PSW AND PC ON STACK
 (2)  036112  012637  000006                    MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
 (1)  036116  012701  003776                    MOV     #3776,R1        ;;SETUP ADDRESS
 (1)  036122  105727                            TSTB    (PC)+           ;;USE MEMORY MANAGEMENT?
 (1)  036124  000200                    $KT11:  .WORD   200             ;;SET TO USE MEMORY MANAGEMENT
 (1)  036126  100070                            BPL     $CORE           ;;BR IF NO
 (1)  036130  012737  036302  000004            MOV     #$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
 (1)  036136  005737  177572                    TST     @#SR0           ;;KT11 ARE YOU THERE?
 (1)  036142  052737  100000  036124            BIS     #100000,$KT11   ;;YES--SET KT11 KEY
 (1)  036150  012737  036410  000250            MOV     #KTERR,@#MMVEC  ;;SET IN CASE OF ERROR
 (1)  036156  012737  000340  000252            MOV     #340,@#MMVEC+2
 (1)  036164  005046                            CLR     -(SP)           ;;INITIALIZE FOR "PAR" LOADING
 (1)  036166  012702  172340                    MOV     #KIPAR0,R2      ;;ADDRESS OF FIRST "PAR"
 (1)  036172  012703  000010                    MOV     #^D8,R3         ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
 (1)  036176  012762  077406  177740  1$:      MOV     #77406,-40(R2)  ;;PDR = 4K, UP, READ/WRITE
 (1)  036204  011622                            MOV     (SP),(R2)+      ;;LOAD "PAR"
 (1)  036206  062716  000200                    ADD     #200,(SP)       ;;UPDATE FOR NEXT "PAR"
 (1)  036212  077307                            SOB     R3,1$           ;;LOOP UNTIL ALL EIGHT ARE LOADED
 (1)  036214  012742  177600                    MOV     #177600,-(R2)   ;;SETUP KIPAR7 FOR I/O
 (1)  036220  005042                            CLR     -(R2)           ;;SETUP KIPAR6 FOR TESTING
 (1)  036222  012737  036240  000004            MOV     #2$,@#ERRVEC    ;;CATCH TIMEOUT IF NO SR3
 (1)  036230  012737  000020  172516            MOV     #20,@#SR3       ;;ENABLE 22 BIT MODE
 (1)  036236  000401                            BR      3$              ;;THIS PDP-11 HAS A SR3 REGISTER
 (1)  036240  022626                    2$:     CMP     (SP)+,(SP)+     ;;CLEAN OFF THE STACK--NO SR3
 (1)  036242  005237  177572            3$:     INC     @#SR0           ;;TURN ON MEMORY MANAGEMENT
 (1)  036246  012737  036272  000004            MOV     #$KTOUT,@#ERRVEC ;;SET FOR TIME OUT
 (1)  036254  005737  143776            4$:     TST     @#143776        ;;TRAP ON NON-EX-MEM
 (1)  036260  062712  000040                    ADD     #40,(R2)        ;;MAKE A 1K STEP
 (1)  036264  023712  172356                    CMP     @#KIPAR7,(R2)   ;;LAST ONE?
 (1)  036270  101371                            BHI     4$              ;;NO--TRY IT
 (1)  036272  011202                    $KTOUT: MOV     (R2),R2         ;;GET LAST BANK+1
 (1)  036274  005037  177572                    CLR     @#SR0           ;;TURN OFF MEMORY MANAGEMENT
```

```
   (1)  036300  000421                          BR      $SIZEX
   (1)  036302  042737  100000  036124  $KTNEX: BIC     #100000,$KT11   ;;KT11 NON-EXISTENT
   (1)  036310  012737  036340  000004  $CORE:  MOV     #$CROUT,@#ERRVEC ;;SET FOR TIMEOUT
   (1)  036316  005002                          CLR     R2              ;;SET UP BANK
   (1)  036320  062701  004000          1$:     ADD     #4000,R1        ;;INCREMENT BY 1K
   (1)  036324  062702  000040                  ADD     #40,R2          ;;1K STEP
   (1)  036330  005711                          TST     (R1)            ;;TRAP ON TIME OUT
   (1)  036332  022701  177776                  CMP     #177776,R1      ;;LAST ONE
   (1)  036336  001370                          BNE     1$              ;;NO--TRY AGAIN
   (1)  036340  162701  004000          $CROUT: SUB     #4000,R1
   (1)  036344  162702  000040          $SIZEX: SUB     #40,R2          ;;DROP BACK
   (1)  036350  010006                          MOV     R0,SP           ;;RESTORE THE STACK
   (1)  036352  012637  000006                  MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
   (1)  036356  012637  000004                  MOV     (SP)+,@#ERRVEC
   (1)  036362  010137  036404                  MOV     R1,$LSTAD       ;;LAST ADDRESS
   (1)  036366  010237  036406                  MOV     R2,$LSTBK       ;;LAST BANK
   (1)  036372  012603                          MOV     (SP)+,R3        ;;RESTORE R3
   (1)  036374  012602                          MOV     (SP)+,R2        ;;RESTORE R2
   (1)  036376  012601                          MOV     (SP)+,R1        ;;RESTORE R1
   (1)  036400  012600                          MOV     (SP)+,R0        ;;RESTORE R0
   (1)  036402  000207                          RTS     PC
   (1)  036404  000000          $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
   (1)  036406  000000          $LSTBK: .WORD   0               ;;CONTAINS THE LAST BANK
  2354  036410  000000          KTERR:  HALT                    ;KT11 FAILURE
  2355  036412  000776                  BR      KTERR
```

```
 2357                                   .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*SAVE R0-R5
  (1)                                   ;*CALL:
  (1)                                   ;*      SAVREG
  (1)                                   ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
  (1)                                   ;*
  (1)                                   ;*TOP---(+16)
  (1)                                   ;* +2---(+18)
  (1)                                   ;* +4---R5
  (1)                                   ;* +6---R4
  (1)                                   ;* +8---R3
  (1)                                   ;*+10---R2
  (1)                                   ;*+12---R1
  (1)                                   ;*+14---R0
  (1)
  (1)  036414                           $SAVREG:
  (3)  036414  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  036416  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  036420  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  036422  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (3)  036424  010446                           MOV     R4,-(SP)        ;;PUSH R4 ON STACK
  (3)  036426  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
  (1)  036430  016646  000022                   MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
  (1)  036434  016646  000022                   MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
  (1)  036440  016646  000022                   MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
  (1)  036444  016646  000022                   MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
  (1)  036450  000002                           RTI
  (1)
  (1)                                   ;*RESTORE R0-R5
  (1)                                   ;*CALL:
  (1)                                   ;*      RESREG
  (1)  036452                           $RESREG:
  (1)  036452  012666  000022                   MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
  (1)  036456  012666  000022                   MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
  (1)  036462  012666  000022                   MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
  (1)  036466  012666  000022                   MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
  (3)  036472  012605                           MOV     (SP)+,R5        ;;POP STACK INTO R5
  (3)  036474  012604                           MOV     (SP)+,R4        ;;POP STACK INTO R4
  (3)  036476  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
  (3)  036500  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
  (3)  036502  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
  (3)  036504  012600                           MOV     (SP)+,R0        ;;POP STACK INTO R0
  (1)  036506  000002                           RTI
```

```
 2359                                   .SBTTL  INTEGER MULTIPLY ROUTINE
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*CALL
  (1)                                   ;*      MOV     MULTIPLER,-(SP)
  (1)                                   ;*      MOV     MULTIPLICAND,-(SP)
  (1)                                   ;*      JSR     PC,@#$MULT
  (1)                                   ;*      RETURN  ;;PRODUCT IS ON THE STACK
  (1)                                   ;*
  (1)                                   ;*      STACK   PRODUCT
  (1)                                   ;*      -----   -------
  (1)                                   ;*      TOP     LSB'S
  (1)                                   ;*      +2      MSB'S
  (1)
  (1)  036510                           $MULT:
  (3)  036510  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  036512  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  036514  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (1)  036516  005046                           CLR     -(SP)           ;;CLEAR THE SIGN KEY
  (1)  036520  016601  000012                   MOV     12(SP),R1       ;;GET THE MULTIPLICAND
  (1)  036524  100002                           BPL     1$              ;;BR IF PLUS
  (1)  036526  005216                           INC     (SP)            ;;SET THE SIGN KEY
  (1)  036530  005401                           NEG     R1              ;;MAKE THE MULTIPLICAND POSTIVE
  (1)  036532  016602  000014           1$:     MOV     14(SP),R2       ;;GET THE MULTIPLIER
  (1)  036536  100002                           BPL     2$              ;;BR IF PLUS
  (1)  036540  005316                           DEC     (SP)            ;;UPDATE THE SIGN KEY
  (1)  036542  005402                           NEG     R2              ;;MAKE THE MULTIPLIER POSTIVE
  (1)  036544  012746  000021           2$:     MOV     #17.,-(SP)      ;;SET THE LOOP COUNT
  (1)  036550  005000                           CLR     R0              ;;SETUP FOR THE MULTIPLY LOOP
  (1)  036552  103001                   3$:     BCC     4$              ;;DON'T ADD IF MULTIPLICAND = 0
  (1)  036554  060200                           ADD     R2,R0
  (1)  036556  006000                   4$:     ROR     R0              ;;POSITION THE PARITIAL PRODUCT AND
  (1)  036560  006001                           ROR     R1              ;;THE MULTIPLICAND
  (1)  036562  005316                           DEC     (SP)            ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
  (1)  036564  001372                           BNE     3$              ;;BR IF NO
  (1)  036566  022616                           CMP     (SP)+,(SP)      ;;SHOULD PRODUCT BE NEGATIVE?
  (1)  036570  001403                           BEQ     5$              ;;GO TO EXIT IF NO
  (1)  036572  005400                           NEG     R0              ;;YES--SO MAKE IT SO
  (1)  036574  005401                           NEG     R1
  (1)  036576  005600                           SBC     R0
  (1)  036600  005726                   5$:     TST     (SP)+           ;;CLEAR SIGN INFO. OFF OF STACK
  (1)  036602  010066  000012                   MOV     R0,12(SP)       ;;PUT THE PRODUCT ON THE STACK (MSB'S)
  (1)  036606  010166  000010                   MOV     R1,10(SP)       ;;LSB'S
  (3)  036612  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
  (3)  036614  012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
  (3)  036616  012600                           MOV     (SP)+,R0        ;;POP STACK INTO R0
  (1)  036620  000207                           RTS     PC
```

```
 2361                                   .SBTTL  INTEGER DIVIDE ROUTINE
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*THIS ROUTINE WILL DIVIDE A 32-BIT TWO'S COMPLEMENT INTEGER
  (1)                                   ;*DIVIDEND BY A 16-BIT TWO'S COMPLEMENT INTEGER DIVISOR GIVING
  (1)                                   ;*A 16-BIT TWO'S COMPLEMENT INTEGER QUOTIENT AND A 16-BIT REMAINDER.
  (1)                                   ;*DIVISION WILL BE PERFORMED SO THAT THE REMAINDER IS OF THE
  (1)                                   ;*SAVE SIGN AS THE DIVIDEND.
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     LOW DIVIDEND,-(SP) ;;THE HIGH DIVIDEND MUST BE < 1/2
  (1)                                   ;*      MOV     HIGH DIVIDEND,-(SP);    AS LARGE AS THE DIVISOR
  (1)                                   ;*      MOV     DIVISOR,-(SP)
  (1)                                   ;*      JSR     PC,$DIV
  (1)                                   ;*      RETURN                  ;;QUOTIENT & REMAINDER ARE ON THE STACK
  (1)                                   ;*      "V"=0   IMPLIES NO ERROR
  (1)                                   ;*      "V"=1   IMPLIES ERROR OCCURRED
  (1)                                   ;*              "C"=0   DIVIDE OVERFLOW OCCURRED
  (1)                                   ;*              "C"=1   ATTEMPTED TO DIVIDE BY ZERO
  (1)                                   ;*
  (1)                                   ;*
  (1)                                   ;*      STACK   NO ERROR        OVERFLOW        DIVIDE BY ZERO
  (1)                                   ;*      -----   --------        --------        --------------
  (1)                                   ;*      TOP     REMAINDER       ALL ZEROS       ALL ONES
  (1)                                   ;*      +2      QUOTIENT        ALL ZEROS       ALL ONES
  (1)
  (1)  036622                           $DIV:
  (2)  036622  104400                           TRAP                    ;;PUSH OLD PSW AND PC ON STACK
  (1)  036624  042716  000017                   BIC     #17,(SP)        ;;STRIP AWAY CONDITION CODES
  (3)  036630  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  036632  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  036634  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  036636  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (1)  036640  005046                           CLR     -(SP)           ;;SAVE A PLACE FOR SIGNS
  (1)  036642  012746  000021                   MOV     #17.,-(SP)      ;;SETUP THE ITERATION COUNTER
  (1)  036646  016601  000024                   MOV     24(SP),R1       ;;PICKUP THE DIVIDEND
  (1)  036652  016600  000022                   MOV     22(SP),R0
  (1)  036656  100005                           BPL     1$              ;;CHECK THE SIGN
  (1)  036660  105366  000003                   DECB    3(SP)           ;;KEEP TRACK OF THE SIGN
  (1)  036664  005400                           NEG     R0              ;;AND NEGATE THE ORIGINAL
  (1)  036666  005401                           NEG     R1              ;;NUMBER
  (1)  036670  005600                           SBC     R0
  (1)  036672  016602  000020           1$:     MOV     20(SP),R2       ;;PICKUP THE DIVISOR
  (1)  036676  002407                           BLT     2$              ;;CHECK THE SIGN
  (1)  036700  003011                           BGT     3$              ;;DIVISOR OF 0 IS A NO-NO
  (1)  036702  052766  000003  000014           BIS     #3,14(SP)       ;;SET "V" & "C"
  (1)  036710  012700  177777                   MOV     #-1,R0          ;;SET REMAINDER TO ALL ONES
  (1)  036714  000424                           BR      7$              ;;EXIT
  (1)  036716  005266  000002           2$:     INC     2(SP)           ;;KEEP TRACK OF DIVISORS SIGN
  (1)  036722  000401                           BR      4$
  (1)  036724  005402                   3$:     NEG     R2              ;;NEGATE THE ORIGINAL NUMBER
  (1)  036726  000241                   4$:     CLC                     ;;CLEAR "C"
  (1)  036730  000405                           BR      6$              ;;START FORMING QUOTIENT
  (1)  036732  006100                   5$:     ROL     R0              ;;POSITION MSB'S
  (1)  036734  010003                           MOV     R0,R3           ;;COPY
  (1)  036736  060203                           ADD     R2,R3           ;;COMPARE DIVIDEND & DIVISOR
  (1)  036740  103001                           BCC     6$              ;;BR IF DIVIDEND > DIVISOR
```

```
(1)  036742  010300                        MOV     R3,R0           ;;REMAINDER AFTER THIS LOOP
(1)  036744  006101                6$:     ROL     R1              ;;QUOTIENT BIT ENTERS HERE
(1)  036746  005316                        DEC     (SP)            ;;DONE?
(1)  036750  001370                        BNE     5$              ;;BR IF NO
(1)  036752  005701                        TST     R1              ;;OVERFLOW?
(1)  036754  100005                        BPL     8$              ;;BR IF NO
(1)  036756  052766  000002 000014        BIS     #2,14(SP)       ;;SET "V" IN RETURN STATUS WORD
(1)  036764  005000                        CLR     R0              ;;SET REMAINDER TO ALL ZEROS
(1)  036766  010001                7$:     MOV     R0,R1           ;;COPY REMAINDER INTO QUOTIENT
(1)  036770  005726                8$:     TST     (SP)+           ;;CLEAR COUNTER FROM STACK
(1)  036772  005716                        TST     (SP)            ;;REMAINDER SIGN CORRECTION NEEDED?
(1)  036774  002004                        BGE     9$              ;;BR IF NO
(1)  036776  005400                        NEG     R0              ;;NEGATE REMAINDER
(1)  037000  105066  000001                CLRB    1(SP)           ;;CLEAR SIGN
(1)  037004  005316                        DEC     (SP)            ;;BUT DON'T FORGET QUOTIENT
(1)  037006  005726                9$:     TST     (SP)+           ;;QUOTIENT SIGN CORRECTION NEEDED?
(1)  037010  001401                        BEQ     10$             ;;BR IF NO
(1)  037012  005401                        NEG     R1              ;;NEGATE QUOTIENT
(1)  037014  010166  000020        10$:    MOV     R1,20(SP)       ;;RETURN QUOTIENT AND
(1)  037020  010066  000016                MOV     R0,16(SP)       ;;REMAINDER TO USER
(3)  037024  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  037026  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  037030  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  037032  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  037034  012666  000002                MOV     (SP)+,2(SP)     ;;SETUP TO RETURN CONDITION CODES
(1)  037040  000002                        RTI                     ;;RETURN
```

```
2363                                    .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
 (1)                                    ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
 (1)                                    ;*POSITIVE.
 (1)                                    ;*CALL
 (1)                                    ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 (1)                                    ;*      JSR     PC,@#$DB2D
 (1)                                    ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
 (1)                                                                    ;;IS ON THE STACK
 (1)
 (1)
 (1)  037042  104413                    $DB2D:  SAVREG                  ;;SAVE REGISTERS
 (1)  037044  016602  000002                    MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
 (1)  037050  012700  037222                    MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
 (1)  037054  010066  000002                    MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
 (1)  037060  012201                            MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
 (1)  037062  012202                            MOV     (R2)+,R2
 (1)  037064  012737  000012  037140            MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
 (1)  037072  012704  037152                    MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
 (1)  037076  012705  037154                    MOV     #$TNPWR+2,R5
 (1)  037102  005003                    1$:     CLR     R3              ;;CLEAR PARTIAL
 (1)  037104  161401                    2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
 (1)  037106  005602                            SBC     R2
 (1)  037110  161502                            SUB     (R5),R2
 (1)  037112  002402                            BLT     3$              ;;BR IF TEN POWER TO LARGE
 (1)  037114  005203                            INC     R3              ;;ADD 1 TO PARTIAL
 (1)  037116  000772                            BR      2$              ;;LOOP
 (1)  037120  062401                    3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
 (1)  037122  005502                            ADC     R2
 (1)  037124  062402                            ADD     (R4)+,R2
 (1)  037126  022525                            CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
 (1)  037130  052703  000060                    BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
 (1)  037134  110320                            MOVB    R3,(R0)+        ;;SAVE IT
 (1)  037136  005327                            DEC     (PC)+           ;;DONE?
 (1)  037140  000000                    4$:     .WORD   0
 (1)  037142  001357                            BNE     1$              ;;BR IF NO
 (1)  037144  105020                            CLRB    (R0)+           ;;TERMINATOR
 (1)  037146  104414                            RESREG                  ;;RESTORE REGISTERS
 (1)  037150  000207                            RTS     PC              ;;RETURN
 (1)  037152  145000                    $TNPWR: 145000                  ;;1.0E09
 (1)  037154  035632                            35632
 (1)  037156  160400                            160400                  ;;1.0E08
 (1)  037160  002765                            2765
 (1)  037162  113200                            113200                  ;;1.0E07
 (1)  037164  000230                            230
 (1)  037166  041100                            041100                  ;;1.0E06
 (1)  037170  000017                            17
 (1)  037172  103240                            103240                  ;;1.0E05
 (1)  037174  000001                            1
 (1)  037176  023420                            23420                   ;;1.0E04
 (1)  037200  000000                            0
 (1)  037202  001750                            1750                    ;;1.0E03
 (1)  037204  000000                            0
 (1)  037206  000144                            144                     ;;1.0E02
```

```
   (1)  037210  000000                          0
   (1)  037212  000012                          12                      ;;1.0E01
   (1)  037214  000000                          0
   (1)  037216  000001                          1                       ;;1.0E00
   (1)  037220  000000                          0
   (1)  037222  000014                $DECVL:   .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
  2364                                .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
   (1)
   (2)                                ;;******************************************************************
   (1)                                ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
   (1)                                ;*UNSIGNED DECIMAL ASCIZ NUMBER.
   (1)                                ;*CALL
   (1)                                ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
   (1)                                ;*      JSR     PC,@#$SB2D      ;;CALL
   (1)                                ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
   (1)
   (1)
   (1)  037236  016637  000002  037266 $SB2D:   MOV     2(SP),1$        ;;SAVE BINARY NUMBER
   (1)  037244  012746  037266                   MOV     #1$,-(SP)       ;;SET POINTER
   (1)  037250  004737  037042                   JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
   (1)  037254  062716  000005                   ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
   (1)  037260  012666  000002                   MOV     (SP)+,2(SP)     ;;PICKUP POINTER
   (1)  037264  000207                           RTS     PC              ;;RETURN
   (1)  037266  000000  000000          1$:      .WORD   0,0
```

```
2366                                    .SBTTL  TRAP DECODER
 (1)
 (2)                                    ;;**********************************************************************
 (1)                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 (1)                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 (1)                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 (1)                                    ;*GO TO THAT ROUTINE.
 (1)
 (1)  037272  010046                    $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 (1)  037274  016600  000002                    MOV     2(SP),R0        ;;GET TRAP ADDRESS
 (1)  037300  005740                            TST     -(R0)           ;;BACKUP BY 2
 (1)  037302  111000                            MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 (1)  037304  006300                            ASL     R0              ;;POSITION FOR INDEXING
 (1)  037306  016000  037326                    MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 (1)  037312  000200                            RTS     R0              ;;GO TO ROUTINE
 (1)
 (1)
 (1)                                    ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
 (1)
 (1)  037314  011646                    $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 (1)  037316  016666  000004  000002            MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 (1)  037324  000002                            RTI                     ;;RESTORE THE PSW
 (1)
 (3)                                    .SBTTL  TRAP TABLE
 (3)
 (3)                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 (3)                                    ;*BY THE "TRAP" INSTRUCTION.
 (3)
 (3)                                    ;       ROUTINE
 (3)                                    ;       -------
 (3)  037326  037314                    $TRPAD: .WORD   $TRAP2
 (3)  037330  034376                            $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 (3)  037332  034704                            $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 (3)  037334  034660                            $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 (3)  037336  034720                            $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 (3)  037340  033666                            $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 (1)
 (3)  037342  033060                            $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 (1)
 (3)  037344  033010                            $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 (3)  037346  033272                            $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 (3)  037350  033412                            $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 (3)  037352  033564                            $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 (3)  037354  036414                            $SAVREG ;;CALL=SAVREG   TRAP+13(104413) SAVE R0-R5 ROUTINE
 (3)  037356  036452                            $RESREG ;;CALL=RESREG   TRAP+14(104414) RESTORE R0-R5 ROUTINE
2370
```

```
2372                                                    .SBTTL  A TO D FIELD SITE AND ADJUSTMENT LOOP
2373  037360  004737  045700            FSITE:  JSR     PC,CTRLCG       ;TEST FOR CTRL C OR G
2374                                    ;CAMERA #0
2375  037364  032777  000100  141546          BIT     #SW06,@SWR      ;TEST IF CAM 0 G=1
2376  037372  001004                          BNE     1$
2377  037374  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2378  037400  000000                          0                       ;CAMERA 0
2379  037402  040444                          CAM0G1                  ;STORE RESULTS
2380  037404  032777  000001  141526  1$:     BIT     #SW00,@SWR      ;TEST IF CAM 0 G=2
2381  037412  001004                          BNE     2$
2382  037414  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2383  037420  002000                          BIT10                   ;GAIN = 2
2384  037422  040446                          CAM0G2
2385                                    ;CAMERA #1
2386  037424  032777  000200  141506  2$:     BIT     #SW7,@SWR       ;TEST IF CAM 1 G=1
2387  037432  001004                          BNE     3$
2388  037434  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2389  037440  000400                          BIT8                    ;CAMERA 1
2390  037442  040450                          CAM1G1                  ;STORE RESULTS
2391  037444  032777  000002  141466  3$:     BIT     #SW01,@SWR      ;TEST IF CAM 1 G=2
2392  037452  001004                          BNE     4$
2393  037454  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2394  037460  002400                          BIT10!BIT8              ;CAMERA 1 GAIN = 2
2395  037462  040452                          CAM1G2                  ;RESULTS
2396                                    ;CAMERA #2
2397  037464  032777  000400  141446  4$:     BIT     #SW8,@SWR       ;TEST IF CAM 2 G=1
2398  037472  001004                          BNE     5$
2399  037474  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2400  037500  001000                          BIT9                    ;CAMERA #2
2401  037502  040454                          CAM2G1                  ;RESULTS
2402  037504  032777  000004  141426  5$:     BIT     #SW2,@SWR       ;TEST IF CAM 2 G 2
2403  037512  001004                          BNE     6$
2404  037514  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2405  037520  003000                          BIT10!BIT9              ;GAIN = 2
2406  037522  040456                          CAM2G2                  ;RESULTS
2407                                    ;CAMERA #3
2408  037524  032777  001000  141406  6$:     BIT     #SW9,@SWR       ;TEST IF CAM 3 G-1
2409  037532  001004                          BNE     7$
2410  037534  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2411  037540  001400                          BIT9!BIT8               ;CAMERA #3
2412  037542  040460                          CAM3G1                  ;RESULTS
2413  037544  032777  000010  141366  7$:     BIT     #SW3,@SWR       ;TEST IF CAM 3 G 2
2414  037552  001004                          BNE     10$
2415  037554  004537  040106                  JSR     R5,CONVRT       ;CONVERT
2416  037560  003400                          BIT10!BIT9!BIT8         ;GAIN = 2
2417  037562  040462                          CAM3G2                  ;RESULTS
2418                                    ;JOYSTICK
2419  037564  032777  000020  141346  10$:    BIT     #SW4,@SWR       ;TEST IF JOYSTICK
2420  037572  001011                          BNE     CALRPT
2421  037574  052777  000001  142154          BIS     #BIT0,@SFR      ;ASK FOR JOYSTICK
2422  037602  105777  142150          11$:    TSTB    @SFR            ;WAIT FOR JOY DONE
2423  037606  100375                          BPL     11$
2424  037610  017737  142146  040464          MOV     @JOY,JOYG1      ;SAVE THE RESULT
```

```
                                        ;NOW TEST IS TYPEOUT IS ENABLED
                                        
037616  004537  040064          CALRPT: JSR     R5,CKTSWR       ;TEST BIT 6
037622  000100                          BIT6
037624  000405                          BR      1$              ;BR IF SET
037626  104401  040466                  TYPE,   CAM0TX          ;REPORT CAMERA #0
037632  004537  040362                  JSR     R5,CAMUNP       ;REPORT VALUES
037636  040444                          CAM0G1
037640  004537  040064          1$:     JSR     R5,CKTSWR       ;TEST BIT 0
037644  000001                          BIT0
037646  000405                          BR      2$
037650  104401  040505                  TYPE,   CAM0TW          ;REPORT G=2
037654  004537  040362                  JSR     R5,CAMUNP
037660  040446                          CAM0G2
                                        
037662  004537  040064          2$:     JSR     R5,CKTSWR       ;TEST BIT 7
037666  000200                          BIT7
037670  000405                          BR      3$              ;BR IF SET
037672  104401  040524                  TYPE,   CAM1TX          ;REPORT CAMERA #1
037676  004537  040362                  JSR     R5,CAMUNP       ;REPORT VALUES
037702  040450                          CAM1G1
037704  004537  040064          3$:     JSR     R5,CKTSWR       ;TEST BIT 1
037710  000002                          BIT1
037712  000405                          BR      4$
037714  104401  040543                  TYPE,   CAM1TW          ;REPORT CAM 1 G=2
037720  004537  040362                  JSR     R5,CAMUNP
037724  040452                          CAM1G2
                                        
037726  004537  040064          4$:     JSR     R5,CKTSWR       ;TEST BIT 8
037732  000400                          BIT8
037734  000405                          BR      5$              ;BR IF SET
037736  104401  040562                  TYPE,   CAM2TX          ;REPORT CAMERA #2
037742  004537  040362                  JSR     R5,CAMUNP       ;REPORT VALUES
037746  040454                          CAM2G1
037750  004537  040064          5$:     JSR     R5,CKTSWR       ;TEST BIT 2
037754  000004                          BIT2
037756  000405                          BR      6$
037760  104401  040601                  TYPE,   CAM2TW
037764  004537  040362                  JSR     R5,CAMUNP       ;REPORT CAMERA #2 G-2
037770  040456                          CAM2G2
                                        
037772  004537  040064          6$:     JSR     R5,CKTSWR       ;TEST BIT 9
037776  001000                          BIT9
040000  000405                          BR      7$              ;BR IF SET
040002  104401  040620                  TYPE,   CAM3TX          ;REPORT CAMERA #3
040006  004537  040362                  JSR     R5,CAMUNP       ;REPORT VLAUES
040012  040460                          CAM3G1
040014  004537  040064          7$:     JSR     R5,CKTSWR       ;TEST BIT 3
040020  000010                          BIT3
040022  000405                          BR      10$
040024  104401  040637                  TYPE,   CAM3TW          ;REPORT CAMERA #3 G 2
040030  004537  040362                  JSR     R5,CAMUNP
040034  040462                          CAM3G2
                                        
040036  004537  040064          10$:    JSR     R5,CKTSWR       ;TEST BIT 4
040042  000020                          BIT4
```

```
2482  040044  000405                    BR      11$
2483  040046  104401  040656            TYPE,   JOYTW
2484  040052  004537  040362            JSR     R5,CAMUNP       ;REPORT JOYSTICK
2485  040056  040464                    JOYG1
2486  040060  000137  037360    11$:    JMP     FSITE
2487
2488  040064  032577  141050    CKTSWR: BIT     (R5)+,@SWR              ;TEST IF INHIBIT THIS CAMERA
2489  040070  001005                    BNE     1$                      ;BR IF YES
2490  040072  032777  020000  141040    BIT     #SW13,@SWR              ;TEST INHIBIT TYPE-OUT
2491  040100  001001                    BNE     1$                      ;BR IF YES
2492  040102  005725                    TST     (R5)+                   ;BUMP EXIT POINTER
2493  040104  000205            1$:     RTS     R5                      ;EXIT
```

```
2495                                        ;SUBROUTINE TO TAKE 8 CONVERSIONS AND AVERAGE FOUR OF THEM
2496  040106  012537  040356                CONVRT: MOV     (R5)+,11$       ;SAVE CAMERA CHANNEL
2497  040112  012777  004000  141636        4$:     MOV     #CLRALL,@SFR            ;CLEAR THE DEVICE
2498  040120  013777  040356  141620              MOV     11$,@CSR                ;SELECT CAMERA AND GAIN
2499  040126  012777  060000  141620              MOV     #BUF0,@BAR              ;LOAD BUS ADDRESS
2500  040134  012777  177770  141610              MOV     #-10,@WCR               ;LOAD WORD COUNT
2501  040142  013777  040360  141606              MOV     INOUTZ,@SFR             ;SET Z INPUTS IF ENABLED
2502  040150  012700  000004                      MOV     #4.,R0                  ;LOAD COUNTER
2503  040154  005001                              CLR     R1                      ;FOR OUT OF RANGE VOLTAGE
2504  040156  052777  000001  141562              BIS     #BIT0,@CSR              ;ENABLE THE DEVICE
2505  040164  105777  141556              1$:     TSTB    @CSR                    ;WAIT FOR IDLE
2506  040170  100432                              BMI     3$
2507  040172  005301                              DEC     R1                      ;DELAY
2508  040174  001373                              BNE     1$
2509  040176  004737  045700                      JSR     PC,CTRLCG               ;TEST FOR CTRL G OR C
2510  040202  005300                              DEC     R0                      ;FINISHED OUT OF RANGE TIMER
2511  040204  001367                              BNE     1$
2512  040206  032777  020000  140724              BIT     #SW13,@SWR              ;TEST IF INHIBIT TYPEOUT
2513  040214  001012                              BNE     5$                      ;BR IF INHIBIT
2514  040216  113700  040357                      MOVB    11$+1,R0                ;GET CAMERA #
2515  040222  042700  177774                      BIC     #177774,R0              ;MASK OFF EXTRA BITS
2516  040226  062700  000060                      ADD     #60,R0                  ;MAKE INTO ASCII
2517  040232  110037  032720                      MOVB    R0,OUTCHN               ;INSERT INTO ASCII ERROR MESSAGE
2518  040236  104401  032625                      TYPE,   OUTRNG                  ;TELL OPERATOR
2519  040242  032777  100000  140670  5$:     BIT     #SW15,@SWR              ;TEST IF HALT ON NO CONVERSION ERROR
2520  040250  001401                              BEQ     6$                      ;BR IF CLEARED
2521  040252  000000                              HALT    ;NO CONVERT FLAG - INPUT VOLTAGE WAS OUT OF RANGE OR NO "Z" PULSES
2522  040254  000716                      6$:     BR      4$
2523  040256  012700  060000              3$:     MOV     #BUF0,R0                ;LOAD POINTER
2524  040262  005003                              CLR     R3                      ;CLEAR RESULT
2525  040264  005004                              CLR     R4
2526  040266  012737  000004  040354              MOV     #4,10$                  ;LOAD COUNTER
2527  040274  112002                      2$:     MOVB    (R0)+,R2                ;GET LOW BYTE
2528  040276  112001                              MOVB    (R0)+,R1                ;GET HIGH BYTE
2529  040300  042702  177400                      BIC     #177400,R2
2530  040304  042701  177400                      BIC     #177400,R1
2531  040310  060104                              ADD     R1,R4                   ;UPDATE RESULT
2532  040312  060203                              ADD     R2,R3                   ;UPDATE RESULT
2533  040314  005337  040354                      DEC     10$                     ;FINISHED ?
2534  040320  001365                              BNE     2$
2535  040322  006203                              ASR     R3
2536  040324  006203                              ASR     R3
2537  040326  006204                              ASR     R4
2538  040330  006204                              ASR     R4
2539  040332  010437  040354                      MOV     R4,10$                  ;SAVE IT
2540  040336  105037  040355                      CLRB    10$+1                   ;CLEAR HIGH BYTE
2541  040342  110337  040355                      MOVB    R3,10$+1                ;LOAD HIGH BYTE
2542  040346  013735  040354                      MOV     10$,@(R5)+              ;SAVE THE RESULT
2543  040352  000205                              RTS     R5
2544  040354  000000                      10$:    0
2545  040356  000000                      11$:    0
2546  040360  000002                      INOUTZ: TESTZ           ;0= USE CUSTOMER Z PULSES, 1= MAINT Z PULSES
2547
2548                                        ;SUBROUTINE TO TYPE THE CAMERA MESSAGE
2549  040362  013537  040440              CAMUNP: MOV     @(R5)+,10$              ;GET  RESULT
2550  040366  113737  040441  040442              MOVB    10$+1,11$
```

```
 2551  040374  105037  040441                  CLRB    10$+1
 2552  040400  105037  040443                  CLRB    11$+1
 2553  040404  013746  040440                  MOV     10$,-(SP)
 2554  040410  104403                          TYPOS
 2555  040412     003     001                  .BYTE   3,1                     ;TYPE 3 DIGIT NUMBER
 2556  040414  104401  040674                  TYPE,   G1B                     ;AND TRALING ASCII
 2557  040420  013746  040442                  MOV     11$,-(SP)
 2558  040424  104403                          TYPOS
 2559  040426     003     001                  .BYTE   3,1                     ;TYPE 3 DIGIT NUMBER
 2560  040430  000240                          NOP
 2561  040432  000240                          NOP
 2562  040434  000240                          NOP
 2563  040436  000205                          RTS     R5                      ;EXIT
 2564  040440  000000                  10$:    0
 2565  040442  000000                  11$:    0
 2566  040444  000000                  CAM0G1: 0
 2567  040446  000000                  CAM0G2: 0
 2568  040450  000000                  CAM1G1: 0
 2569  040452  000000                  CAM1G2: 0
 2570  040454  000000                  CAM2G1: 0
 2571  040456  000000                  CAM2G2: 0
 2572  040460  000000                  CAM3G1: 0
 2573  040462  000000                  CAM3G2: 0
 2574  040464  000000                  JOYG1:  0
 2575
 2576  040466     015     012          CAM0TX: .BYTE   15,12
 2577  040470  040503  030115  020060          .ASCIZ  /CAM00 G=1 X=/
       040476  036507  020061  036530
       040504     000
 2578  040505     015     012          CAM0TW: .BYTE   15,12
 2579  040507     103  046501  030060          .ASCIZ  /CAM00 G=2 X=/
       040514  043440  031075  054040
       040522  000075
 2580  040524     015     012          CAM1TX: .BYTE   15,12
 2581  040526  040503  030115  020061          .ASCIZ  /CAM01 G=1 X=/
       040534  036507  020061  036530
       040542     000
 2582  040543     015     012          CAM1TW: .BYTE   15,12
 2583  040545     103  046501  030460          .ASCIZ  /CAM01 G=2 X=/
       040552  043440  031075  054040
       040560  000075
 2584  040562     015     012          CAM2TX: .BYTE   15,12
 2585  040564  040503  030115  020062          .ASCIZ  /CAM02 G=1 X=/
       040572  036507  020061  036530
       040600     000
 2586  040601     015     012          CAM2TW: .BYTE   15,12
 2587  040603     103  046501  031060          .ASCIZ  /CAM02 G=2 X=/
       040610  043440  031075  054040
       040616  000075
 2588  040620     015     012          CAM3TX: .BYTE   15,12
 2589  040622  040503  030115  020063          .ASCIZ  /CAM03 G=1 X=/
       040630  036507  020061  036530
       040636     000
 2590  040637     015     012          CAM3TW: .BYTE   15,12
 2591  040641     103  046501  031460          .ASCIZ  /CAM03 G=2 X=/
       040646  043440  031075  054040
```

```
        040654  000075
2592    040656                          JOYTX:
2593    040656     015     012          JOYTW:  .BYTE   15,12
2594    040660  047512  051531  044524          .ASCIZ  /JOYSTICK X=/
        040666  045503  054040  000075
2595    040674  054440  000075          G1B:    .ASCIZ  / Y=/
2596    040700  005015  044504  043106  GAIN:   .ASCII  <15><12>/DIFFERENTIAL LINEARITY:/
        040706  051105  047105  044524
        040714  046101  046040  047111
        040722  040505  044522  054524
        040730     072
2597    040731     040  020040  040507  GAINX:  .ASCII  /   GAIN = /
        040736  047111  036440     040
2598    040743     061     040     040  GAIN1:  .BYTE   61,40,40,0
        040746     000
2599    040747     040  026455  000040  DASH:   .ASCIZ  / -- /
2600    040754  046040  041123  005015  LSBMSG: .ASCIZ  / LSB/<15><12>
        040762     000
2601    040763     040  045523  050111  SKPMSG: .ASCIZ  / SKIPPED STATE(S)/<15><12>
        040770  042520  020104  052123
        040776  052101  024105  024523
        041004  005015     000
2602    041007     040  025052  051105  ERMSG:  .ASCIZ  / **ERROR**/<15><12>
        041014  047522  025122  006452
        041022  000012
2603    041024  020040  020040  047440  OKMSG:  .ASCIZ  /     OK/<15><12>
        041032  006513  000012
2604    041036  047040  051101  047522  NARMSG: .ASCIZ  # NARROW STATE(S)#<15><12>
        041044  020127  052123  052101
        041052  024105  024523  005015
        041060     000
2605    041061     040  044527  042504  WIDMSG: .ASCIZ  # WIDE STAIE(S)#<15><12>
        041066  051440  040524  042524
        041074  051450  006451  000012
2606    041102  052123  052101  026505  STATE:  .ASCIZ  /STATE-- WIDTH/<15><12>
        041110  020055  044527  052104
        041116  006510  000012
2607    041122  046040  041123  046440  LINEA:  .ASCIZ  / LSB MAXIMUS AT /
        041130  054101  046511  051525
        041136  040440  020124     000
2608    041143     057     000          SLASH:  .ASCIZ  #/#
2609    041145     122  046105  052101  MSG21:  .ASCIZ  /RELATIVE ACCURACY:/<15><12>
        041152  053111  020105  041501
        041160  052503  040522  054503
        041166  006.72  000012
2610    041172  005015  047522  020115  COMP:   .ASCIZ  <15><12>/ROM BLASTING COMPLETED/
        041200  046102  051501  044524
        041206  0.3516  041440  046517
        041214  0..3120  052105  042105
        041222     000
2611    041223     015  052012  046511  TIMEO:  .ASCII  <15><12>/TIMEOUT FROM BLASTER   CHECK SWITCHES ON BLASTER/
        041230  047305  052125  043040
        041236  047522  020115  046102
        041244  051501  042524  020122
        041252  020040  044103  041505
        041260  020113  053523  052111
```

```
        041266  044103  051505  047440
        041274  020116  046102  051501
        041302  042524     122
 2612   041305     015  020012  051117          .ASCIZ  <15><12>/ OR REPLACE ROM IN BLASTER/<15><12>
        041312  051040  050105  040514
        041320  042503  051040  046517
        041326  044440  020116  046102
        041334  051501  042524  006522
        041342  000012
 2613   041344     015     012          FIELDI: .BYTE   15,12
 2614   041346  047111  042524  047122          .ASCIZ  /INTERNAL MAINT. Z PULSES <Y FOR YES> ? /
        041354  046101  046440  044501
        041362  052116  020056  020132
        041370  052520  051514  051505
        041376  036040  020131  047506
        041404  020122  042531  037123
        041412  037440  000040
 2615   041416  005015  053101  051105  AVRGO:  .ASCIZ  <15><12>/AVERAGE OF 128 STATES = /
        041424  043501  020105  043117
        041432  030440  034062  051440
        041440  040524  042524  020123
        041446  020075     000
 2616   041451     015     012     007  UNFIX:  .BYTE   15,12,7
 2617   041454  054105  042503  042105          .ASCIZ  /EXCEEDED CORRECTION COUNTER FOR /
        041462  042105  041440  051117
        041470  042522  052103  047511
        041476  020116  047503  047125
        041504  042524  020122  047506
        041512  020122     000
 2618   041515     015     012     007  ABORTB: .BYTE   15,12,7
 2619   041520  047522  020115  046102          .ASCII  /ROM BLASTING ABORTED/
        041526  051501  044524  043516
        041534  040440  047502  052122
        041542  042105
 2620   041544     015     012     007          .BYTE   15,12,7,0
        041547     000
 2621   041550  020040  026455  020040  SGNVAL: .ASCII  /  --  /
 2622   041556     053     040          SGNVL1: .BYTE   53,40
 2623   041560     060     060     056  PERTXT: .BYTE   60,60,56,60,45,15,12,0
        041563     060     045     015
        041566     012     000
 2624   041570     015     012     000  RTN:    .BYTE   15,12,0,0
        041573     000
 2625   041574  041040  046105  053517  BELMSG: .ASCIZ  / BELOW LIMIT/<15><12>
        041602  046040  046511  052111
        041610  005015     000
 2626   041613     040  047516  046522  NORMSG: .ASCIZ  / NORMAL STATES/<15><12>
        041620  046101  051440  040524
        041626  042524  006523  000012
 2627   041634  040440  047502  042526  ABOMSG: .ASCIZ  / ABOVE LIMIT/<15><12>
        041642  046040  046511  052111
        041650  005015     000
 2628   041653     040  040520  051523  PASMSG: .ASCIZ  / PASSED/<15><12>
        041660  042105  005015     000
 2629   041665     040  040506  046111  FAIMSG: .ASCIZ  / FAILED/<15><12>
        041672  042105  005015     000
```

```
 2630           041700                          .EVEN
 2631  041700   000000                LSBAVG: 0
 2632  041702   000000                LSBSVQ: 0
 2633  041704   000000                LSBSVR: 0
 2634  041706   000000                LSBSVW: 0
 2635  041710   000000                DIFEX1: 0
 2636  041712   042134                ROMPNT: ROMVAL
 2637  041714   027340                NOMIAL: 12000.          ;AVERAGE COUNT
 2638
 2639                                 ;THESE LOC.'S CLEARED BY DIFLIN
 2640  041716   000000                AVGVAL: 0
 2641  041720   000000                DIFERR: 0
 2642  041722   000000                BELOW:  0
 2643  041724   000000                ABOVE:  0
 2644  041726   000000                NORMAL: 0
 2645  041730   000000                NARROW: 0
 2646  041732   000000                NARA:   0
 2647  041734   000000                NARB:   0
 2648  041736   000000                WIDA:   0
 2649  041740   000000                WIDB:   0
 2650  041742   000000                DIF:    0
 2651  041744   000000                OUT:    0
 2652  041746   000000                FIRST:  0
```

```
                                        ;KEYBOARD CONVERSATION TO THE BLASTER   SELECT TEST "T"
                                        ;APPLY POWER TO BLASTER AND DEPRESS I/O AND EXECUTE BUTTONS
                                        ;ALL INDICATORS SHOULD LIGHT    THE BLASTER SHOULD SEND A ">" CHARACTER
                                        ;TYPE THE COMMANDS FOLLOWED BY A "CR", SOME WILL ECHO THE CR AND OTHERS
                                        ;WILL NOT       TYPE "ESC"/"ALT" KEYS TO ABORT CURRNET COMMAND
                                        ;TYPE   KF2/2           ;THIS SETS OCTAL COMMAND MODE
                                        ;TYPE   FM30            ;THIS SET OCTAL INPUT/OUTPUT
                                        ;TYPE   SP[0037/8]      ;THIS SIZES THE PROM -- 0037/8 IS REPORTED BY THE BLASTER
                                        ;TYPE   LD              ;THIS READS THE PROM INTO THE RAM
                                        ;TYPE   D00/37          ;THIS OUTPUTS THE RAM CONTENTS  TO THE TTY
                                        ;TYPE   ZP              ;THIS EXITS COMPUTER CONTROL TO THE BUTTON MODE

041750  105777  137722          LOOPC:  TSTB    @DLICSR         ;BLASTER INPUT ?
041754  100007                          BPL     LOOPD           ;NO
041756  000240                          NOP
041760  000240                          NOP
041762  000240                          NOP
041764  017720  137710                  MOV     @DLIBD,(R0)+    ;SAVE THE CHARACTER
041770  005237  042132                  INC     BLICNT          ;UPDATE COUNTER

041774  105777  137150          LOOPD:  TSTB    @$TPS           ;PRINTER READY ?
042000  100363                          BPL     LOOPC           ;BR IF NOT
042002  005737  042132                  TST     BLICNT          ;ANY CHARACTERS
042006  001413                          BEQ     LOOPE           ;BR IF NOT
042010  012177  137136                  MOV     (R1)+,@$TPB     ;PRINT THE CHAR.
042014  005337  042132                  DEC     BLICNT
042020  001353                          BNE     LOOPC           ;BR IF MORE DATA
042022  012700  060000          BTALK:  MOV     #BUF0,R0
042026  010001                          MOV     R0,R1
042030  005037  042132                  CLR     BLICNT
042034  000745                          BR      LOOPC
                                        ;COME HERE IS NO BLASTER DATA TO BE TYPED
042036  105777  137102          LOOPE:  TSTB    @$TKS           ;KEYBOARD INPUT
042042  100342                          BPL     LOOPC           ;BR IF NOT
042044  017702  137076                  MOV     @$TKB,R2                ;READ CHAR
042050  042702  177600                  BIC     #177600,R2      ;MASK THE DATA
042054  022702  000003                  CMP     #3,R2           ;TEST FOR CTRL C
042060  001413                          BEQ     2$              ;BR IF CTRL C
042062  022702  000177                  CMP     #177,R2         ;TEST FOR RUBOUT
042066  001412                          BEQ     LOOPF           ;BR IF ESC
042070  020227  000140                  CMP     R2,#140         ;TEST IF LOWER CASE
042074  103402                          BLO     1$              ;BR IF NOT
042076  042702  000040                  BIC     #40,R2          ;MAKE UPPER CASE
042102  010277  137576          1$:     MOV     R2,@DLODB       ;LOAD BLASTER DATA OUT
042106  000720                          BR      LOOPC
042110  000137  002646          2$:     JMP     RBEGO           ;JUMP TO RESTART
042114  012777  000033  137562 LOOPF:  MOV     #33,@DLODB      ;LOAD ESC <CANCLE>
042122  012777  000134  137022          MOV     #'\,@$TPB
042130  000707                          BR      LOOPC
042132  000000          BLICNT: 0
```

```
2706
2707                                    ;A017 CORRECTION PROM VALUES    <MEMORY RAM>
2708
2709  042134    200   200   200  ROMVAL: .BYTE   200,200,200,200,200 200,200,200 ;GAIN = 1 VALUE
      042137    200   200   200
      042142    200   200
2710  042144    200   200   200          .BYTE   200,200,200,200,200 200,200,200
      042147    200   200   200
      042152    200   200
2711  042154    200   200   200          .BYTE   200,200,200,200,200,200,200,200 ;GAIN = 2 VALUE
      042157    200   200   200
      042162    200   200
2712  042164    200   200   200          .BYTE   200,200,200,200,200,200,200,200
      042167    200   200   200
      042172    200   200
2713
2714                                    ;A017 CORRECTION PRAM VALUES    <RAM MEMORY>
2715                                    ;ROMVAL MUST BE IN COMPLEMENTED AND REVERSED ORDER <LSB=MSB>
2716
2717  042174    376   376   376  RAMVAL: .BYTE   376,376,376,376,376,376,376,376 ;GAIN = 1 VALUE
      042177    376   376   376
      042202    376   376
2718  042204    376   376   376          .BYTE   376,376,376,376,376,376,376,376
      042207    376   376   376
      042212    376   376
2719  042214    376   376   376          .BYTE   376,376,376,376,376,376,376,376 ;GAIN = 2 VALUE
      042217    376   376   376
      042222    376   376
2720  042224    376   376   376          .BYTE   376,376,376,376,376,376,376,376
      042227    376   376   376
      042232    376   376
2721
2722                                    ;M8036 PROGRAM PROM VALUES
2723
2724
2725  042234    215   244   116  PROROM: .BYTE   215,244,116,144,215,244,116,144 ;M8036 PROGRAM PROM
      042237    144   215   244
      042242    116   144
2726  042244    215   244   116          .BYTE   215,244,116,144,215,244,116,144
      042247    144   215   244
      042252    116   144
2727  042254    106   144   314          .BYTE   106,144,314,304,377,377,377,377
      042257    304   377   377
      042262    377   377
2728  042264    377   377   377          .BYTE   377,377,377,377,377,377,377,377
      042267    377   377   377
      042272    377   377
```

```
2730                                    ;BLASTER SECTION
2731  042274  104401  050764            BLAST:  TYPE,   PRIM0           ;TELL OPERATOR ABOUT THE CABLE
2732  042300  104401  051376                    TYPE,   PRIM5           ;AND SWITCHES
2733  042304  042777  000001  137364            BIC     #BIT0,@DLICSR   ;ENSURE INIT. ROM CONTROL LINE
2734  042312  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER TO INITILIZE
2735  042316  052562                            ESCP                    ;REALLY WE ARE WAITING FOR THE OPERATOR
2736  042320  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER THE KEYBOARD INPUT MODE
2737  042324  052564                            KF0
2738  042326  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER THE DATA FORMAT <ASCII-OCTAL>
2739  042332  052573                            FM0
2740                                    ;PRIME THE MEMORY RAM BUFFER
2741  042334  012700  042134                    MOV     #ROMVAL,R0      ;GET POINTER
2742  042340  112720  000200            1$:     MOVB    #200,(R0)+      ;LOAD THE BUFFER
2743  042344  022700  042174                    CMP     #ROMVAL+40,R0   ;TEST IF DONE
2744  042350  001373                            BNE     1$              ;BR IF BUFFER NOT PRIMED
2745  042352  012737  000036  046004            MOV     #36,PRIME1      ;LOAD GAIN AND RESOL. VALUES
2746  042360  004537  046010                    JSR     R5,LIMITS       ;LOAD DIFLIN ERROR LIMIT VALUES
2747  042364  046074                            G1LIM0                  ;G1 LIMIT - USERS
2748  042366  046114                            G1LIM1                  ;G1 LIMIT - OPTION CHECKOUT
2749  042370  112737  000061  040743            MOVB    #'1,GAIN1       ;LOAD GAIN TYPEOUT VALUE
2750  042376  012737  042135  041712            MOV     #ROMVAL+1,ROMPNT        ;LOAD RAM POINTER
2751  042404  004737  042450                    JSR     PC,4$           ;TEST THAT GAIN
2752  042410  012737  002036  046004            MOV     #2036,PRIME1    ;LOAD GAIN AND RESOL. VALUES
2753  042416  004537  046010                    JSR     R5,LIMITS       ;LOAD DIFLIN ERROR LIMIT VALUES
2754  042422  046134                            G2LIM0                  ;G2 LIMIT - USERS
2755  042424  046154                            G2LIM1                  ;G2 LIMIT - OPTION CHECKOUT
2756  042426  112737  000062  040743            MOVB    #'2,GAIN1       ;LOAD GAIN TYPEOUT VALUE
2757  042434  012737  042155  041712            MOV     #ROMVAL+21,ROMPNT       ;LOAD RAM POINTER
2758  042442  004737  042450                    JSR     PC,4$           ;TEST GAIN OF 2
2759  042446  000436                            BR      10$
2760  042450  012737  000003  042620  4$:       MOV     #3,100$         ;LOAD LOOP COUNTER
2761  042456  012737  000001  041710  5$:       MOV     #1,DIFEX1       ;LOAD DIFLIN EXIT FLAG
2762  042464  012737  042502  046006            MOV     #6$,DIFEX0      ;LOAD DIFLIN EXIT RETURN
2763  042472  004737  042622                    JSR     PC,ZAPRAM       ;ENSURE BLASTER RAM HAS BEEN LOADED WITH
2764                                            ;THE "RAMVAL" VALUE
2765  042476  000137  043722                    JMP     DIFLN0          ;RUN "DIFLIN"
2766  042502  004737  043362            6$:     JSR     PC,ADJFIX       ;ADJUST AND FIX THE CORRECTION WEIGHTS
2767  042506  005337  042620                    DEC     100$            ;HAS THIS SECTION BEEN EXECITED N TIMES
2768  042512  001361                            BNE     5$              ;BR IF NOT
2769  042514  005737  041720                    TST     DIFERR          ;TEST IF ERROR OCCURRED?
2770  042520  001410                            BEQ     8$              ;BR IF NOT
2771  042522  104401  041451            7$:     TYPE,   UNFIX           ;TELL THE OPERATOR
2772  042526  104401  040731                    TYPE,   GAINX           ;ABOUT IT
2773  042532  104401  041515                    TYPE,   ABORTB          ;TELL OPER. THE BLASTING IS ABORTED
2774  042536  000137  002646                    JMP     RBEG0           ;RESTART
2775  042542  000207            8$:     RTS     PC              ;EXIT
```

```
                                        ;THE RAM DATA IS NOW CORRECT - BLAST THE BLANK ROM
042544  004537  043064                  10$:    JSR     R5,RDBLST       ;TELL THE BLASTER TO COMP. THE DATA
042550  053013                                  CMO
042552  004537  043064                          JSR     R5,RDBLST       ;TELL THE BLASTER TO CHECK-SUM DATA
042556  053023                                  CCO
042560  004537  043064                          JSR     R5,RDBLST       ;TELL THE BLASTER TO BLAST THE ROM
042564  053026                                  PGO
042566  005237  043244                          INC     NOEXIT          ;SET FLAG TO EXIT WITHOUT RECPT OF ">" CHAR FROM BLASTER
042572  004537  043064                          JSR     R5,RDBLST       ;TELL THE BLASTER TO RETURN TO KEYPAD MODE
042576  053036                                  ZPO
                                        ;INFORM THE OPERATOR ABOUT THE BLASTER ROM
042600  104401  051767                          TYPE,   PRIM1   ;TELL OPERATOR TO REMOVE RAM CABLE AND INSTERT ROM INTO A017
042604  104411                                  RDLIN
042606  012600                                  MOV     (SP)+,R0        ;CLEAN STACK
042610  104401  041172                          TYPE,   COMP            ;COMPLETED
042614  000137  002646                          JMP     RBEGO
042620  000000                          100$:   0
                                        ;SUBROUTINE TO CONVERT THE MEMORY RAM VALUE INTO RAM MEMORY VALUE
042622  012703  042174                  ZAPRAM: MOV     #RAMVAL,R3      ;LOAD OUTPUT POINTER
042626  012700  042134                          MOV     #ROMVAL,R0      ;LOAD INPUT POINTER
042632  112001                          1$:     MOVB    (R0)+,R1        ;GET A BYTE
042634  105101                                  COMB    R1              ;INVERT VALUE
042636  005002                                  CLR     R2              ;CLEAR RESULT
042640  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042642  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042644  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042646  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042650  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042652  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042654  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042656  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042660  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042662  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042664  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042666  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042670  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042672  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042674  006001                                  ROR     R1              ;MOVE INTO CARRY BIT
042676  006102                                  ROL     R2              ;MOVE CARRY INTO BIT 0
042700  110223                                  MOVB    R2,(R3)+        ;SAVE OUTPUT BYTE
042702  020027  042174                          CMP     R0,#ROMVAL+40   ;TEST IF END
042706  001351                                  BNE     1$              ;BR IF NOT
                                        ;SUBROUTINE TO LOAD MEMORY RAM INTO RAM MEMORY
042710  004537  043246                          JSR     R5,CONROM       ;CONVERT "ROMVAL" TABLE INTO BLASTER FORMAT
042714  042174                                  RAMVAL
042716  052777  000001  136752                  BIS     #BIT0,@DLICSR   ;ENABLE RAM MEMORY TO BE LOADED
042724  004537  043064                          JSR     R5,RDBLST       ;TELL THE BLASTER THE NEW RAM DATA
042730  052601                                  DIO
042732  004537  043064                          JSR     R5,RDBLST       ;TELL THE BLASTER TO CHECKSUM THE DATA
042736  053023                                  CCO
042740  004537  043064                          JSR     R5,RDBLST       ;TELL THE BLASTER TO PROGRAM THE RAM-PAK
042744  053026                                  PGO
042746  042777  000001  136722                  BIC     #BIT0,@DLICSR   ;RE-ENTER EMMULATE MODE
042754  000207                                  RTS     PC              ;EXIT
```

```
2820                                    ;BLASTING THE PROGRAM PROM (M8036)
2821
2822  042756  104401  051540            PBLAST: TYPE,   PRIM3           ;INFORM THE OPERATOR
2823  042762  042777  000001  136706            BIC     #BIT0,@DLICSR   ;ENSURE ROM MODE
2824  042770  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER TO INITILIZE
2825  042774  052562                            ESCP
2826  042776  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER THE KEYBOARD INPUT MODE
2827  043002  052564                            KF0
2828  043004  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER THE DATA FORMAT <ASCII-OCTAL>
2829  043010  052573                            FM0
2830  043012  004537  043246                    JSR     R5,CONROM       ;CONVERT THE "PROROM" TABLE INTO BLASTER FORMAT
2831  043016  042234                            PROROM
2832  043020  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER THE PROM DATA
2833  043024  052601                            DI0
2834  043026  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLAS  ` TO CHECKSUM THE DATA
2835  043032  053023                            CC0
2836  043034  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER TO BLAST THE PROM
2837  043040  053026                            PG0
2838  043042  005237  043244                    INC     NOEXIT          ;SET FLAG TO EXIT WITHOUT RECPT OF ">" FROM BLASTER
2839  043046  004537  043064                    JSR     R5,RDBLST       ;TELL THE BLASTER TO RETURN TO KEYPAD MODE
2840  043052  053036                            ZP0
2841  043054  104401  041172                    TYPE,   COMP            ;INFORM THE OPERATOR IT'S DONE
2842  043060  000137  002646                    JMP     RBEGO
```

```
2844
2845                                    ;COME THERE TO OUTPUT A MESSAGE TO THE BLASTER
2846                                    ;EXIT ONLY WHEN THE BLASTER HAS ECHOED AN ">"
2847                                    ;REPORT AN ERROR UPON TIMEOUT UNLESS "NOEXIT" IS NON-ZERO
2848  043064  010046                  RDBLST: MOV     R0,-(SP)        ;SAVE R0
2849  043066  012500                          MOV     (R5)+,R0        ;GET ARG.
2850  043070  012737  000012  043242          MOV     #10.,13$        ;LOAD DELAY FOR ERROR COUNTER
2851  043076  111037  043234                  MOVB    (R0),10$        ;LOAD A CHAR
2852  043102  017737  136572  043236          MOV     @DLIBD,11$      ;FALSE READ TO CLEAR READY
2853  043110  105777  136566          1$:     TSTB    @DLOCSR         ;WAIT FOR OUTPUT READY
2854  043114  100375                          BPL     1$
2855  043116  105737  043234                  TSTB    10$             ;TEST IF FINISHED ?
2856  043122  001406                          BEQ     2$
2857  043124  112037  043234                  MOVB    (R0)+,10$       ;GET A CHAR.
2858  043130  001403                          BEQ     2$              ;BR IF TERM
2859  043132  113777  043234  136544          MOVB    10$,@DLODB      ;OUTPUT THE CHARACTER
2860  043140  105777  136532          2$:     TSTB    @DLICSR         ;WAIT FOR INPUT
2861  043144  100415                          BMI     4$              ;BR IF INPUT
2862  043146  005337  043240                  DEC     12$             ;DELAY
2863  043152  001372                          BNE     2$
2864  043154  005337  043242                  DEC     13$             ;DELAY
2865  043160  001367                          BNE     2$
2866  043162  005737  043244                  TST     NOEXIT          ;TEST IF EXIT IS ALLOWED WITHOUT ">"
2867  043166  001016                          BNE     5$              ;BR IF YES
2868  043170  104401  041223                  TYPE,   TIMEO
2869  043174  000137  002646                  JMP     RBEGO
2870  043200  017737  136474  043236  4$:     MOV     @DLIBD,11$      ;READ CHAR
2871  043206  042737  177600  043236          BIC     #177600,11$     ;MASK OFF BITS
2872  043214  022737  000076  043236          CMP     #'>,11$         ;TEST FOR >
2873  043222  001332                          BNE     1$              ;TRY NEXT CHAR.
2874  043224  012600                  5$:     MOV     (SP)+,R0        ;RESTORE R0
2875  043226  005037  043244                  CLR     NOEXIT
2876  043232  000205                          RTS     R5              ;EXIT
2877
2878  043234  000000                  10$:    0
2879  043236  000000                  11$:    0
2880  043240  000000                  12$:    0
2881  043242  000000                  13$:    0
2882  043244  000000                  NOEXIT: 0
```

```
2884
2885                                        ;CONVERT ROM DATA INTO PROM BLASTER FORMAT
2886  043246  012700  052614          CONROM: MOV     #DIDATA+3,R0    ;LOAD RESULT POINTER
2887  043252  012501                          MOV     (R5)+,R1        ;LOAD INITIAL VALUE POINTER
2888  043254  010137  043324                  MOV     R1,10$          ;COPY THE START
2889  043260  062737  000040  043324          ADD     #40,10$         ;MAKE THE END ADDRESS
2890  043266  111102                    1$:   MOVB    (R1),R2         ;GET A BYTE OF DATA
2891  043270  004737  043346                  JSR     PC,SHUF0        ;CONVERT IT TO ASCII
2892  043274  111102                          MOVB    (R1),R2         ;GET SAME BYTE AGAIN
2893  043276  004737  043340                  JSR     PC,SHUF3        ;CONVERT 2ND DIGIT
2894  043302  112102                          MOVB    (R1)+,R2        ;GET MSD
2895  043304  004737  043326                  JSR     PC,SHUF6        ;CONVERT IT
2896  043310  062700  000007                  ADD     #7,R0           ;UPDATE RESULT POINTER
2897  043314  020137  043324                  CMP     R1,10$          ;TEST IF DONE
2898  043320  001362                          BNE     1$
2899  043322  000205                          RTS     R5              ;EXIT
2900  043324  000000                    10$:  0
2901
2902                                        ;ROTATE THE DATA RIGHT AND MAKE ASCII CHARACTER
2903
2904  043326  042702  177477          SHUF6:  BIC     #177477,R2      ;MASK OFF BITS
2905  043332  006202                          ASR     R2
2906  043334  006202                          ASR     R2
2907  043336  006202                          ASR     R2
2908  043340  006202                    SHUF3:  ASR     R2
2909  043342  006202                          ASR     R2
2910  043344  006202                          ASR     R2
2911  043346  042702  177770          SHUF0:  BIC     #177770,R2      ;MASK OFF BITS
2912  043352  062702  000060                  ADD     #60,R2          ;MAKE ASCII
2913  043356  110240                          MOVB    R2,-(R0)        ;LOAD RESULT BYTE
2914  043360  000207                          RTS     PC              ;EXIT
2915
```

```
2917                                    ;SUBROUTINE TO ADJUST THE LSB VALUES
2918  043362  012700  062036            ADJFIX: MOV     #BUF1+36,R0     ;GET BUFFER POINTER
2919  043366  013702  041712                    MOV     ROMPNT,R2       ;GET ROM POINTER
2920  043372  010204                            MOV     R2,R4
2921  043374  062704  000017                    ADD     #17,R4          ;MAKE END VALUE
2922  043400  012001                    10$:    MOV     (R0)+,R1        ;GET LSB (17) M-1
2923  043402  061001                            ADD     (R0),R1         ;ADD LSB (00) M
2924  043404  163701  041716                    SUB     AVGVAL,R1       ;SUB 2 LSB AVG.
2925  043410  163701  041716                    SUB     AVGVAL,R1
2926  043414  010137  041706                    MOV     R1,LSBSVW       ;SAVE FOR LATER USAGE
2927  043420  010146                            MOV     R1,-(SP)        ;SAVE ON STACK
2928  043422  012746  000316                    MOV     #316,-(SP)      ;MULTI BY 316 (8)
2929  043426  004737  036510                    JSR     PC,$MULT
2930  043432  012637  043544                    MOV     (SP)+,100$      ;SAVE LSW
2931  043436  012637  043546                    MOV     (SP)+,101$      ;SAVE MSW
2932
2933  043442  013746  043544                    MOV     100$,-(SP)      ;
2934  043446  013746  043546                    MOV     101$,-(SP)
2935  043452  013746  041716                    MOV     AVGVAL,-(SP)
2936  043456  004737  036622                    JSR     PC,$DIV
2937  043462  102004                            BVC     1$              ;BR IF NO ERROR
2938  043464  000000                            HALT            ;MATH ERROR
2939  043466  000000                            HALT            ; VALUE OUT OF RANGE
2940  043470  000240                            NOP
2941  043472  000240                            NOP
2942  043474  012637  041704            1$:     MOV     (SP)+,LSBSVR    ;SAVE INTGR. REMAINDER
2943  043500  012637  041702                    MOV     (SP)+,LSBSVQ    ;SAVE QUOTIENT
2944
2945                                    ;NOW ADD THE QUOTIENT TO THE MEMORY RAM VALUE
2946
2947  043504  111203                            MOVB    (R2),R3         ;GET CURRENT MEMORY RAM VALUE
2948  043506  000240                            NOP
2949  043510  042703  177400                    BIC     #177400,R3      ;CLEAR OFF UPPER BITS
2950  043514  163703  041702                    SUB     LSBSVQ,R3       ;ADD THE QUOTIENT
2951  043520  110322                            MOVB    R3,(R2)+        ;RELOAD MEMORY RAM VALUE
2952
2953                                    ;NOW UPDATE 'M' VALUE POINTER
2954
2955  043522  062700  000036                    ADD     #36,R0          ;UPDATE POINTER
2956  043526  011001                            MOV     (R0),R1         ;GET VALUE
2957  043530  063701  041706                    ADD     LSBSVW,R1       ;CORRECT THE VALUE (#10)
2958  043534  010110                            MOV     R1,(R0)         ;RESTORE FOR NEXT USAGE
2959  043536  020204                            CMP     R2,R4           ;TEST WHEN FINISHED
2960  043540  001317                            BNE     10$             ;BR UNTIL DONE
2961
2962  043542  000207                            RTS     PC              ;EXIT
2963
2964  043544  000000                    100$:   0
2965  043546  000000                    101$:   0
```

```
2967
2968                                        ;;DIFFERENTIAL LINEARITY
2969  043550  005037  041710                DIFLIN: CLR     DIFEX1          ;CLEAR EXIT FLAG
2970  043554  112737  000061  040743                MOVB    #'1,GAIN1       ;LOAD GAIN MESSAGE VALUE
2971  043562  105737  001134                        TSTB    $AUTOB          ;TEST IF UNDER MONITOR
2972  043566  001007                                BNE     10$             ;BR IF  YES
2973  043570  104401  040700                        TYPE,   GAIN
2974  043574  005737  043550                        TST     DIFLIN          ;TEST BIT 15
2975  043600  100402                                BMI     10$             ;DONT TELL OPER. ABOUT A017
2976  043602  104401  050764                        TYPE,   PRIM0           ;TELL OPERATOR ABOUT A017
2977  043606  012737  000036  046004  10$:         MOV     #36,PRIME1      ;LOAD RESOLUTION AND GAIN
2978  043614  004537  046010                        JSR     R5,LIMITS       ;LOAD DIFLIN TOLERANCE
2979  043620  046074                                G1LIM0                  ,G1 USER LIMIT
2980  043622  046114                                G1LIM1                  ;G1 OPTION AREA
2981  043624  012737  043634  046006                MOV     #1$,DIFEX0      ;LOAD RETURN ADDRESS
2982  043632  000433                                BR      DIFLN0
2983  043634  112737  000062  040743  1$:          MOVB    #'2,GAIN1       ;LOAD GAIN MESSAGE VALUE
2984  043642  012737  002036  046004                MOV     #2036,PRIME1    ;LOAD RESOLUTION AND GAIN
2985  043650  004537  046010                        JSR     R5,LIMITS       ;LOAD DIFLIN TOLERANCE
2986  043654  046134                                G2LIM0                  ;G2 USER LIMIT
2987  043656  046154                                G2LIM1                  ;G2 OPTION AREA
2988  043660  012737  043702  046006                MOV     #2$,DIFEX0      ;LOAD RETURN ADDRESS
2989  043666  105737  001134                        TSTB    $AUTOB          ;RUNNING UNDER MONITOR
2990  043672  001002                                BNE     11$             ;BR IF YES
2991  043674  104401  040700                        TYPE,   GAIN
2992  043700  000410                        11$:    BR      DIFLN0
2993  043702  022737  177777  050024  2$:          CMP     #-1,RUNDIF      ;ENTER VIA "F" SELECTION ?
2994  043710  001002                                BNE     3$              ;BR IF NOT
2995  043712  000137  030102                        JMP     $EOP            ;YES REPORT END OF PASS
2996  043716  000137  002646            3$:         JMP     RBEG0
2997
2998
2999                                        ;DIFLIN ROUTINE
3000  043722  012700  041716                DIFLN0: MOV     #AVGVAL,R0      ;LOAD CLEARING POINTER
3001  043726  005020                        DIFLNX: CLR     (R0)+
3002  043730  022700  041750                        CMP     #FIRST+2,R0     ;FINISHED ?
3003  043734  001374                                BNE     DIFLNX          ;BR IF NOT DONE
3004  043736  012700  060000                        MOV     #BUF0,R0        ;LOAD BUFFER POINTER
3005  043742  005020                        1$:     CLR     (R0)+           ;CLEAR THE DATA BUFFER
3006  043744  022700  064000                        CMP     #BUF2,R0        ;TEST IF FINISHED
3007  043750  001374                                BNE     1$              ;BR IF NOT
3008                                        ;LOOK FOR THE DATA REGION FOR TO
3009  043752  004737  045632                        JSR     PC,STDAT0               ;FIND THE NON-ZERO TO ZERO TRANS. DATA
3010                                        ;DATA HAS NOW GONE INTO THE GOOD 0 DATA REGION
3011  043756  012777  000000  135760                MOV     #0,@KWPSR               ;INIT THE CLOCK PRESET
3012  043764  012777  000025  135746                MOV     #25,@KWCSR              ;ENABLE THE CLOCK
3013                                        ;NOW TIME "T1" LENGTH
3014  043772  004737  045476                2$.     JSR     PC,LISTDT               ;GET CURRENT DATA VALUE
3015  043776  022702  000000                        CMP     #0,R2                   ;TEST FOR EXIT OF 0 DATA REGION
3016  044002  001773                                BEQ     2$                      ;BR UNTILL DONE
3017  044004  105077  135732                        CLRB    @KWCSR1                 ;
3018  044010  152777  000002  135724                BISB    #BIT1,@KWCSR1           ;FIRE ST2
3019  044016  017737  135722  045670                MOV     @KWPSR,DIFT1            ;SAVE COUNTER VALUE
3020                                        ;NOW TIME "T2" LENGTH
3021  044024  004737  045476                3$:     JSR     PC,LISTDT               ;GET CURRENT DATA VALUE
3022  044030  022702  000377                        CMP     #377,R2                 ;TEST FOR ENTRY INTO MAX DATA REGION
```

```
3023  044034  001373                        BNE     3$                      ;BR IF NOT
3024  044036  105077  135700                CLRB    @KWCSR1
3025  044042  152777  000002  135672        BISB    #BIT1,@KWCSR1           ;FIRE ST2
3026  044050  017737  135670  045672        MOV     @KWPSR,DIFT2            ;SAVE COUNTER VALUE
3027                                ;NOW TIME "T3" LENGTH
3028  044056  004737  045476          4$:   JSR     PC,LISTDT               ;GET CURRENT DATA VALUE
3029  044062  022702  000377                CMP     #377,R2                 ;TEST FOR EXIT OF MAX DATA REGION
3030  044066  001773                        BEQ     4$                      ;BR IF NOT
3031  044070  105077  135646                CLRB    @KWCSR1
3032  044074  152777  000002  135640        BISB    #BIT1,@KWCSR1           ;FIRE ST2
3033  044102  017737  135636  045674        MOV     @KWPSR,DIFT3            ;SAVE COUNTER VALUE
3034  044110  005077  135624                CLR     @KWCSR                  ;STOP CLOCK
3035                                ;NOW DETERMINE THE AVG TIME (T3+T2)/2 AND SAVE IN "HT2T3"
3036  044114  013700  045672                MOV     DIFT2,R0                ;GET T2 VALUE
3037  044120  063700  045674                ADD     DIFT3,R0                ;ADD T3 VALUE
3038  044124  006000                        ROR     R0                      ;/2
3039  044126  010037  045676                MOV     R0,HT2T3                ;SAVE FOR LATER USE
3040                                ;NOW RETURN TO LIST MODE AND LOOK FOR THE NON-ZERO TO ZERO EDGE AGAIN
3041  044132  004737  045632          DIFLN1: JSR     PC,STDATO                       ;FIND THE EDGE
3042                                ;NOW START THE CLOCK USING THE AVG TIME
3043                                ;AND DO AN MATRIX MODE COLLECTION
3044  044136  013700  045676                MOV     HT2T3,R0                ;GET AVG CLOCK
3045  044142  005400                        NEG     R0                      ;FIX FOR CLOCK PRESET REG
3046  044144  010077  135574                MOV     R0,@KWPSR               ;LOAD CLOCK PRESET
3047  044150  012777  000020  135562        MOV     #20,@KWCSR              ;PRIME CLOCK
3048  044156  012777  004000  135572        MOV     #CLRALL,@SFR            ;INIT THE NCV11
3049  044164  012777  060000  135556        MOV     #BUF0,@OFF              ;LOAD OFFSET
3050  044172  005077  135554                CLR     @WCR                    ;CLEAR Z COUNTER
3051  044176  005077  135552                CLR     @BAR                    ;
3052  044202  013777  046004  135536        MOV     PRIME1,@CSR             ;SET MODE AND RES.
3053  044210  052777  000022  135540        BIS     #TESTZ!BIT4,@SFR        ;SET TESTZ AND TESTX
3054  044216  052777  000001  135522        BIS     #BIT0,@CSR              ;ENABLE NCV11
3055  044224  052777  000001  135506        BIS     #BIT0,@KWCSR            ;ENABLE CLOCK
3056  044232  105777  135502          1$:   TSTB    @KWCSR                  ;WAIT FOR CLOCK FLAG
3057  044236  100375                        BPL     1$
3058  044240  052777  000400  135510        BIS     #ENDDMA,@SFR            ;STOP DMA XFR
```

```
                                        ;THE MATRIX MODE XFR IF NOW COMPLETED
                                        ;NOW ADD THE CURRENT BYTE DATA COLLECTION INTO A TOTAL INCREMENT BUFFER
044246  005037  050016                        CLR     $TEMP1
044252  012700  060000                        MOV     #BUF0,R0
044256  012701  062000                        MOV     #BUF1,R1

044262  111037  050016                2$:     MOVB    (R0),$TEMP1             ;GET THE WORD
044266  105020                              CLRB    (R0)+                   ;CLEAR THE BYTE TABLE ENTRY
044270  063721  050016                        ADD     $TEMP1,(R1)+            ;UPDATE THE LIST
044274  103003                              BCC     3$                      ;BR IF NO OVERFLOW
044276  005741                              TST     -(R1)                   ;POSITCN POINTER
044300  012721  177777                        MOV     #-1,(R1)+               ;SET TO 177777
044304  022701  064000                3$:     CMP     #BUF2,R1                ;FINISHED?
044310  001364                              BNE     2$                      ;BR IF NOT
                                        ;GET THE AVERAGE OF THE 128 CENTER STATES
044312  012700  062100                        MOV     #BUF1+64.,R0    ;GET INITIAL POINTER
044316  005001                              CLR     R1
044320  005002                              CLR     R2
044322  062001                        4$:     ADD     (R0)+,R1        ;GET A VALUE
044324  103001                              BCC     5$              ;BR IF NO CARRY OVERFLOW
044326  005202                              INC     R2              ;UPDATE MSW
044330  022700  062300                5$:     CMP     #BUF1+192.,R0   ;FINISHED THE BUFFER
044334  001372                              BNE     4$              ;BR IF NOT DONE AVERAGE
044336  000241                              CLC
 (1)    044340  006002                      ROR     R2              ;ROTATE MSW
 (1)    044342  006001                      ROR     R1              ;INTO LSW
 (1)    044344  000241                      CLC
 (1)    044346  006002                      ROR     R2              ;ROTATE MSW
 (1)    044350  006001                      ROR     R1              ;INTO LSW
 (1)    044352  000241                      CLC
 (1)    044354  006002                      ROR     R2              ;ROTATE MSW
 (1)    044356  006001                      ROR     R1              ;INTO LSW
 (1)    044360  000241                      CLC
 (1)    044362  006002                      ROR     R2              ;ROTATE MSW
 (1)    044364  006001                      ROR     R1              ;INTO LSW
 (1)    044366  000241                      CLC
 (1)    044370  006002                      ROR     R2              ;ROTATE MSW
 (1)    044372  006001                      ROR     R1              ;INTO LSW
 (1)    044374  000241                      CLC
 (1)    044376  006002                      ROR     R2              ;ROTATE MSW
 (1)    044400  006001                      ROR     R1              ;INTO LSW
044402  010137  041716                        MOV     R1,AVGVAL       ;SAVE THE AVERAGE
044406  004737  045700                        JSR     PC,CTRLCG       ;TEST FOR ^C OR ^G
044412  023737  041716  041714              CMP     AVGVAL,NOMIAL   ;TEST IF AVG. IF >
044420  103644                              BLO     DIFLN1          ;BR IF NOT
```

```
                                        ;READ THE TOTAL INCREMENT BUFFER AND DETERMINE IF ANY VALUES OUT OF RANGE

044422  013737  041716  045116  READ1:  MOV     AVGVAL,101$     ;GET AVERAGE
044430  012700  000001                  MOV     #1,R0
044434  012701  062002                  MOV     #BUF1+2,R1
044440  012137  045114          1$:     MOV     (R1)+,100$      ;GET A WORD
044444  163737  045116  045114          SUB     101$,100$       ;REMOVE THE AVERAGE
044452  100006                          BPL     2$
044454  005137  045114                  COM     100$
044460  112737  000055  041556          MOVB    #'-,SGNVL1      ;INSERT "-" SIGN
044466  000403                          BR      3$
044470  112737  000053  041556  2$:     MOVB    #'+,SGNVL1      ;INSERT "+" SIGN
044476  013746  045114          3$:     MOV     100$,-(SP)
044502  012746  001750                  MOV     #1000.,-(SP)
044506  004737  036510                  JSR     PC,$MULT        ;MULTIPLY NUM-AVG BY 1000.
044512  012637  045132                  MOV     (SP)+,107$      ;GET RESULT
044516  012637  045134                  MOV     (SP)+,110$
044522  013746  045132                  MOV     107$,-(SP)      ;DIVIDE RESULT
044526  013746  045134                  MOV     110$,-(SP)
044532  013746  045116                  MOV     101$,-(SP)      ;DIVIDE AGAIN
044536  004737  036622                  JSR     PC,$DIV
044542  102003                          BVC     4$
044544  000000                          HALT    ;FATAL ERROR DURING CAL. OF ERROR TOLERANCES
044546  000240                          NOP
044550  000240                          NOP
044552  012637  045124          4$:     MOV     (SP)+,104$      ;GET REMAINDER
044556  012637  045126                  MOV     (SP)+,105$      ;GET QOUT
044562  013702  045116                  MOV     101$,R2         ;ROUND UP IF NEEDED
044566  006202                          ASR     R2
044570  023702  045124                  CMP     104$,R2         ;COMPARE RESULT
044574  002402                          BLT     5$
044576  005237  045126                  INC     105$            ;ROUND UP
044602  000240                  5$:     NOP
                                        ;DIFLIN ERROR CHECK

044604  023737  045126  046056  6$:     CMP     105$,NORTOL     ;TEST AGAINST NORMAL
044612  103446                          BLO     50$             ;BR IF OK
044614  023737  045126  046062          CMP     105$,NARTOL     ;TEST AGAINST WIDE/NARROW TOLERANCE
044622  103416                          BLO     20$             ;BR IF OK
044624  023737  045126  046066          CMP     105$,NURTOL     ;TEST AGAINST LARGE/SMALL TOLERANCE
044632  103424                          BLO     21$             ;BR IF OK
044634  122737  000053  041556          CMPB    #'+,SGNVL1      ;TEST IF +
044642  001403                          BEQ     7$
044644  005237  041722                  INC     BELOW           ;COUNT THE LOWER VALUE
044650  000435                          BR      51$
044652  005237  041724          7$:     INC     ABOVE           ;COUNT THE HIGHER VALUE
044656  000432                          BR      51$
044660  122737  000053  041556  20$:    CMPB    #'+,SGNVL1      ;TEST IF +
044666  001403                          BEQ     22$
044670  005237  041734                  INC     NARB            ;COUNT THE LOWER
044674  000423                          BR      51$
044676  005237  041740          22$:    INC     WIDB            ;COUNT THE HIGHER
044702  000420                          BR      51$
044704  122737  000053  041556  21$:    CMPB    #'+,SGNVL1      ;TEST IF +
044712  001403                          BEQ     23$
044714  005237  041732                  INC     NARA            ;COUNT THE LOWER
```

```
3150  044720  000411                        BR      51$
3151  044722  005237  041736        23$:    INC     WIDA            ;COUNT THE HIGHER
3152  044726  000406                        BR      51$
3153  044730  005237  041726        50$:    INC     NORMAL          ;COUNT THE NORMAL
3154  044734  122737  000115  052412        CMPB    #'M,RUNIT       ;TEST IF FORCE REPORT
3155  044742  001054                        BNE     77$
3156  044744  105737  001134        51$:    TSTB    $AUTOB          ;TEST IF MONITOR ?
3157  044750  001051                        BNE     77$             ;BR IF YES
3158                                ;REPORT THE STATE INFORMATION
3159  044752  005737  041746                TST     FIRST           ;FIRST TYPEOUT ?
3160  044756  001013                        BNE     10$             ;BR IF YES
3161  044760  005237  041746                INC     FIRST           ;SET FLAG
3162  044764  104401  041416                TYPE    ,AVRGO          ;TELL THE OPERATOR THE AVERG.
3163  044770  013746  041716                MOV     AVGVAL,-(SP)
3164  044774  104405                        TYPDS
3165  044776  104401  041570                TYPE,   RTN
3166  045002  104401  041102                TYPE,   STATE           ;ADD HEADER INFO
3167  045006  010046                10$:    MOV     R0,-(SP)        ;LOAD STATE #
3168  045010  104403                        TYPOS
3169  045012     003     001                .BYTE   3,1
3170  045014  122737  000115  052412        CMPB    #'M,RUNIT       ;TEST IF EXPAND OUTPUT SELECTED
3171  045022  001005                        BNE     11$
3172  045024  104401  040747                TYPE,   DASH            ;INSERT SPACING
3173  045030  013746  045114                MOV     100$,-(SP)      ;REPORT DIFFERENCE
3174  045034  104405                        TYPDS
3175  045036  013746  045126        11$:    MOV     105$,-(SP)
3176  045042  004737  037236                JSR     PC,$SB2D        ;CONVERT TO ASCII
3177  045046  012602                        MOV     (SP)+,R2        ;GET RESULT POINTER TO MESSAGE
3178  045050  062702  000002                ADD     #2,R2           ;ADJUST POINTER
3179  045054  112237  041560                MOVB    (R2)+,PERTXT
3180  045060  112237  041561                MOVB    (R2)+,PERTXT+1  ;LOAD THE PERCENT REPORT
3181  045064  112237  041563                MOVB    (R2)+,PERTXT+3
3182  045070  104401  041550                TYPE.   SGNVAL
3183
3184  045074  004737  045700        77$:    JSR     PC,CTRLCG       ;TEST IF CTRL C/G
3185  045100  005200                        INC     R0
3186  045102  022700  000377                CMP     #255.,R0        ;TEST IF DONE
3187  045106  001414                        BEQ     READ            ;BR IF DONE
3188  045110  000137  044440                JMP     1$              ;TRY NEXT VALUE
3189
3190  045114  000000              100$:     0
3191  045116  000000              101$:     0
3192  045120  000000              102$:     0
3193  045122  000000              103$:     0
3194  045124  000000              104$:     0
3195  045126  000000              105$:     0
3196  045130  000000              106$:     0
3197  045132  000000              107$:     0
3198  045134  000000              110$:     0
3199  045136  000000              111$:     0
```

```
3201
3202                                        ;NOW ACCOUNT FOR ALL THE ERRORS
3203  045140  005037  041720          READ:   CLR     DIFERR          ;CLEAR FLAG
3204  045144  013702  041724                  MOV     ABOVE,R2        ;GET HIGH VALUE
3205  045150  063702  041722                  ADD     BELOW,R2        ;ADD LOW
3206  045154  120237  046070                  CMPB    R2,OBSCNT       ;TEST IF EXCEEDS LIMIT <OBEASE ABOVE/BELOW>
3207  045160  101026                          BHI     1$              ;BR IF ERRORS
3208  045162  013702  041732                  MOV     NARA,R2         ;GET LOW VALUE
3209  045166  063702  041736                  ADD     WIDA,R2         ;ADD HIGH
3210  045172  120237  046064                  CMPB    R2,NURCNT       ;TEST IF EXCEEDS LIMIT <LARGE ABOVE/BELOW>
3211  045176  101017                          BHI     1$
3212  045200  013702  041734                  MOV     NARB,R2         ;GET LOW VALUE
3213  045204  063702  041740                  ADD     WIDB,R2         ;ADD HIGH
3214  045210  120237  046060                  CMPB    R2,NARCNT       ;TEST IF EXCEEDS LIMIT <SMALL ABOVE/BELOW>
3215  045214  101010                          BHI     1$
3216  045216  123737  046054  041726          CMPB    NORCNT,NORMAL   ;TEST IF MIN. NORMAL COUNT HAS BEEN HIT
3217  045224  101004                          BHI     1$
3218  045226  012737  041653  045354          MOV     #PASMSG,READ7   ;LOAD MESSAGE POINTER
3219  045234  000405                          BR      2$
3220  045236  012737  041665  045354  1$:     MOV     #FAIMSG,READ7   ;LOAD MESSAGE POINTER
3221  045244  005237  041720                  INC     DIFERR          ;SET ERROR FLAG
3222  045250  013746  041722          2$:     MOV     BELOW,-(SP)     ;GET NO. OF STATES BELOW LIMITS
3223  045254  104405                          TYPDS                   ;REPORT VALUE
3224  045256  104401  041574                  TYPE    ,BELMSG         ;TYPE MESSAGE
3225  045262  013702  041732                  MOV     NARA,R2         ;GET NARROW LOW
3226  045266  063702  041734                  ADD     NARB,R2         ;ADD NARROW HIGH
3227  045272  010246                          MOV     R2,-(SP)        ;GET NO. OF NARROW STATES
3228  045274  104405                          TYPDS
3229  045276  104401  041036                  TYPE    ,NARMSG         ;TYPE MESSAGE
3230  045302  013746  041726                  MOV     NORMAL,-(SP)    ;GET NORMAL COUNT
3231  045306  104405                          TYPDS
3232  045310  104401  041613                  TYPE,   NORMSG          ;REPORT NORMAL TEXT
3233  045314  013702  041736                  MOV     WIDA,R2         ;GET WIDE LOW
3234  045320  063702  041740                  ADD     WIDB,R2         ;ADD WIDE HIGH
3235  045324  010246                          MOV     R2,-(SP)
3236  045326  104405                          TYPDS
3237  045330  104401  041061                  TYPE    ,WIDMSG         ;TYPE MESSAGE
3238  045334  013746  041724                  MOV     ABOVE,-(SP)
3239  045340  104405                          TYPDS                   ;TYPE NO. OF STATES ABOVE LIMIT
3240  045342  104401  041634                  TYPE    ,ABOMSG         ;TYPE MESSAGE
3241
3242                                        ;REPORT PASS/FAIL MESSAGE
3243
3244  045346  104401  040700                  TYPE,   GAIN            ;RE-TYPE THE GAIN MESSAGE
3245  045352  104401                          TYPE
3246  045354  041653          READ7:  PASMSG                  ;REPORT PASS/FAIL
3247
```

```
3249
3250                                        ;TYPE OUT MEMORY RAM AND RAM MEMORY VALUES IF SELECTED
3251  045356  032777  004000  133554  READXX: BIT     #SW11,@SWR      ;TEST IF THIS  IS WANTED
3252  045364  001440                          BEQ     4$              ;BR IF NOT
3253  045366  005737  041710                  TST     DIFEX1          ;TEST IF BLASTING MODE
3254  045372  001435                          BEQ     4$              ;BR IF NOT
3255
3256  045374  012700  042134                  MOV     #ROMVAL,R0      ;LOAD INITIAL POINTER
3257  045400  004737  045430          1$:     JSR     PC,2$           ;TYPE OUT LINE
3258  045404  022700  042234                  CMP     #ROMVAL+100,R0  ;TEST IF ALL DONE
3259  045410  001373                          BNE     1$              ;BR IF NOT
3260  045412  104401  041570                  TYPE,   RTN
3261  045416  104401  041570                  TYPE,   RTN
3262  045422  104401  041570                  TYPE,   RTN
3263  045426  000417                          BR      4$
3264
3265  045430  012701  000020          2$:     MOV     #16.,R1         ;LOAD WIDE COUNTER
3266  045434  104401  041570                  TYPE,   RTN
3267  045440  112002                  3$:     MOVB    (R0)+,R2        ;GET A BYTE
3268  045442  042702  177400                  BIC     #177400,R2      ;MASK OFF HIGHER BITS
3269  045446  010246                          MOV     R2,-(SP)        ;LOAD VALUE
3270  045450  104403                          TYPOS
3271  045452     003     001                  .BYTE   3,1
3272  045454  104401  040752                  TYPE,   DASH+3
3273  045460  005301                          DEC     R1              ;FINISHED ?
3274  045462  001366                          BNE     3$
3275  045464  000207                          RTS     PC              ;EXIT
3276
3277
3278  045466  005037  041710          4$:     CLR     DIFEX1
3279  045472  000177  000310                  JMP     @DIFEX0         ;EXIT THIS CRAZYNESS
3280
```

```
                                        ;DIF LIN SUBROUTINE LIST MODE DATA COLLECTOR
                                        ;
045476  012777  004000  134252  LISTDT: MOV     #CLRALL,@SFR    ;CLEAR THE DEVICE
045504  005077  134240                  CLR     @OFF            ;PRIME OFFSET REG.
045510  012777  177770  134234          MOV     #-8.,@WCR       ;LOAD WC.
045516  012777  060000  134230          MOV     #BUF0,@BAR      ;LOAD BUS ADDR.
045524  012777  000000  134204          MOV     #0,@CSR         ;SET RES. AND MODE
045532  032737  002000  046004          BIT     #2000,PRIME1    ;TEST GAIN BIT
045540  001403                          BEQ     3$              ;BR IF CLEARED
045542  052777  002000  134176          BIS     #2000,@CSR      ;SELECT GAIN = 2
045550  052777  000002  134200  3$:     BIS     #TESTZ,@SFR     ;SET Z PULSES
045556  052777  000001  134162          BIS     #BIT0,@CSR      ;ENABLE THE DEVICE
                                        ;
045564  105777  134156          1$:     TSTB    @CSR            ;WAIT FOR IDLE
045570  100375                          BPL     1$
045572  012700  060000                  MOV     #BUF0,R0        ;LOAD POINTER TO NEW DATA
045576  005002                          CLR     R2
045600  012703  000010                  MOV     #8.,R3          ;LOAD COUNTER
045604  011001                  2$:     MOV     (R0),R1         ;GET DATA WORD
045606  005020                          CLR     (R0)+           ;CLR BUFFER WORD
045610  042701  177400                  BIC     #177400,R1      ;MAKE OFF BITS
045614  060102                          ADD     R1,R2           ;UPDATE TOTAL
045616  005303                          DEC     R3              ;FINISHED ?
045620  001371                          BNE     2$              ;BR IF NOT
045622  006202                          ASR     R2
045624  006202                          ASR     R2
045626  006202                          ASR     R2
045630  000207                          RTS     PC              ;EXIT
                                        ;SUBROUTINE TO FIND THE TRUE EDGE OF ZERO DATA
045632  012737  177777  045666  STDAT0: MOV     #-1,10$         ;SET ENTRY FLAG
045640  004737  045476          1$:     JSR     PC,LISTDT       ;GET DATA
045644  005702                          TST     R2              ;CHECK DATA
045646  001403                          BEQ     2$              ;BR IF ZERO
045650  005037  045666                  CLR     10$             ;CLEAR ALREADY NON-ZERO DATA FLAG
045654  000771                          BR      1$
                                        ;DATA WAS ZERO - CHECK IF WE STARTED IN 0 DATA REGION
045656  005737  045666          2$:     TST     10$             ;TEST FLAG
045662  001366                          BNE     1$              ;BR IF NOT A GOOD TIME TO EXIT
                                        ;DATA HAS NOW GONE TO A GOOD ZERO DATA REGION
                                        ;NOW EXIT
045664  000207                          RTS     PC              ;BYEBYE
045666  000000                  10$:    0
045670  000000                  DIFT1:  0
045672  000000                  DIFT2:  0
045674  000000                  DIFT3:  0
045676  000000                  HT2T3:  0
```

```
                                        ;SUBROUTINE TO TEST IF OPERATOR TYPED CTRL C OR G
045700  105777  133240          CTRLCG: TSTB    @$TKS           ;INPUT FLAG
045704  100033                          BPL     2$              ;BR IF NON
045706  017737  133234  045776          MOV     @$TKB,CTRCHA    ;GET CHARACTER
045714  042737  177600  045776          BIC     #177600,CTRCHA  ;MASK OFF BITS
045722  022737  000003  045776          CMP     #3,CTRCHA       ;TEST IF CTRL C
045730  001014                          BNE     1$              ;BR IF NOT ^C
045732  052777  000400  134016          BIS     #ENDDMA,@SFR    ;STOP NPR'S
045740  000240                          NOP
045742  000240                          NOP
045744  000240                          NOP
045746  052777  004000  134002          BIS     #CLRALL,@SFR    ;CLEAR THE DEVICE
045754  005726                          TST     (SP)+           ;CLEAN STACK
045756  000137  002014                  JMP     RTRT            ;RE-START
045762  022737  000007  045776  1$:     CMP     #7,CTRCHA       ;TEST IF CTRL G
045770  001001                          BNE     2$              ;BR IF NOT
045772  104406                          GTSWR                   ;GET SWR VALUE
045774  000207                  2$:     RTS     PC              ;EXIT
045776  000000                  CTRCHA: 0
046000  177777                  PRIME:  -1.
046002  177400                  PRIME0: 177400
046004  000036                  PRIME1: 36
046006  000000                  DIFEX0. 0

                                        ;SUBROUTINE TO DETERMINE ERROR LIMITS FOR DIFLIN
046010  012500                  LIMITS: MOV     (R5)+,R0        ;GET 1ST ARG.
046012  005737  050026                  TST     WFMODE          ;TEST IF ON TESTER
046016  001402                          BEQ     1$              ;BR IF NOT
046020  012500                          MOV     (R5)+,R0        ;GET TESTER LIMITS
046022  000401                          BR      2$
046024  005725                  1$:     TST     (R5)+           ;BUMP ADDRESS
046026  012701  046054          2$:     MOV     #NORCNT,R1      ;GET POINTER
046032  012021                          MOV     (R0)+,(R1)+     ;GET VALUE
 (1)    046034  012021                  MOV     (R0)+,(R1)+     ;GET VALUE
 (1)    046036  012021                  MOV     (R0)+,(R1)+     ;GET VALUE
 (1)    046040  012021                  MOV     (R0)+,(R1)+     ;GET VALUE
 (1)    046042  012021                  MOV     (R0)+,(R1)+     ;GET VALUE
 (1)    046044  012021                  MOV     (R0)+,(R1)+     ;GET VALUE
 (1)    046046  012021                  MOV     (R0)+,(R1)+     ;GET VALUE
 (1)    046050  012021                  MOV     (R0)+,(R1)+     ;GET VALUE
046052  000205                          RTS     R5              ;EXIT
```

```
3366                                    ;ACTUAL VALUES FOR DIF LIN ERROR TOLERANCE
3367
3368
3369  046054  000342                    NORCNT: .WORD   226.             ;"NORMAL" MIN COUNT VALUE
3370  046056  000025                    NORTOL: .WORD   21.              ;"NORMAL" TOLERANCE VALUE
3371  046060  000024                    NARCNT: .WORD   20.              ;"NARROW/WIDE" MAX. COUNT VALUE
3372  046062  000063                    NARTOL: .WORD   51.              ;"NARROW/WIDE" TOLERANCE VALUE
3373  046064  000010                    NURCNT: .WORD   8.               ;"LARGE/SMALL" MAX. COUNT VALUE
3374  046066  000145                    NURTOL: .WORD   101.             ;"LARGE/SMALL" TOLERANCE VALUE
3375  046070  000000                    OBSCNT: .WORD   0.               ;"OBESITY" MAX. COUNT VALUE
3376  046072  000377                    OBSTOL: .WORD   377              ;"OBESITY" TOLERANCE VALUE
3377
3378                                    ;DIFLIN ERROR TOLERANCE          GAIN = 1        USER
3379
3380  046074  000342  000025            G1LIM0: .WORD   226.,21.         ;226 MIN. COUNT, =<2.0%
3381  046100  000024  000063                    .WORD   20.,51.          ;20 MAX. COUNT, =<5.0%
3382  046104  000010  000145                    .WORD   8.,101.          ;8 MAX. COUNT, =<10.%
3383  046110  000000  000377                    .WORD   0.,377           ;0 MAX COUNT, >10%
3384
3385                                    ;DIFLIN ERROR TOLERANCE          GAIN = 1        OPTION AREA
3386
3387  046114  000342  000024            G1LIM1: .WORD   226.,20.         ;226 MIN COUNT, =<1.9%
3388  046120  000024  000061                    .WORD   20.,49.          ;20 MAX. COUNT, =<4.8%
3389  046124  000010  000140                    .WORD   8.,96.           ;8 MAX COUNT, =<9.5%
3390  046130  000000  000377                    .WORD   0,377            ;0 MAX COUNT, >9.5%
3391
3392                                    ;DIFLIN ERROR TOLERANCE          GAIN = 2        USER
3393
3394  046134  000330  000031            G2LIM0: .WORD   216.,25.         ;216. MIN COUNT, =<2.4%
3395  046140  000036  000075                    .WORD   30.,61.          ;30. MAX COUNT, =<6.0%
3396  046144  000010  000171                    .WORD   8.,121.          ;8. MAX COUNT, =<12.0%
3397  046150  000000  000377                    .WORD   0.,377           ;0 MAX COUNT, >12.0%
3398
3399                                    ;DIFLIN ERROR TOLERANCE          GAIN = 2        OPTION AREA
3400
3401  046154  000330  000025            G2LIM1: .WORD   216.,21.         ;216. MIN COUNT, =<2.0%
3402  046160  000036  000063                    .WORD   30.,51.          ;30. MAX COUNT, =<5.0%
3403  046164  000010  000145                    .WORD   8.,101.          ;8. MAX COUNT, -<10.0%
3404  046170  000000  000377                    .WORD   0.,377           ;0 MAX COUNT, >10.0%
3405
```

```
 3407  046174     015     012          LISTO:  .BYTE   15,12
 3408  046176  020111  020075  047111          .ASCII  /I = INITIAL A017 ADJUSTMENT USING THE ANALOG TESTER/
       046204  052111  040511  020114
       046212  030101  033461  040440
       046220  045104  051525  046524
       046226  047105  020124  051525
       046234  047111  020107  044124
       046242  020105  047101  046101
       046250  043517  052040  051505
       046256  042524     122
 3409  046261     015     012                  .BYTE   15,12
 3410  046263     102  036440  041040          .ASCII  /B = BLASTING OF THE LINEARITY CORRECTION PROM/
       046270  040514  052123  047111
       046276  020107  043117  052040
       046304  042510  046040  047111
       046312  040505  044522  054524
       046320  041440  051117  042522
       046326  052103  047511  020116
       046334  051120  046517
 3411  046340     015     012                  .BYTE   15,12
 3412  046342  020103  020075  047503          .ASCIZ  /C = CONTROL PROGRAM PROM BLASTING/
       046350  052116  047522  020114
       046356  051120  043517  040522
       046364  020115  051120  046517
       046372  041040  040514  052123
       046400  047111  000107
 3413  046404     015     012          LIST:   .BYTE   15,12
 3414  046406  020114  020075  047514          .ASCII  /L = LOGIC TEST OF M8026 AND M8036/
       046414  044507  020103  042524
       046422  052123  047440  020106
       046430  034115  031060  020066
       046436  047101  020104.  034115
       046444  031460     066
 3415  046447     015     012                  .BYTE   15,12
 3416  046451     104  036440  042040          .ASCII  /D - DIFFERENTIAL LINEARITY OF THE A017/
       046456  043111  042506  042522
       046464  052116  040511  020114
       046472  044514  042516  051101
       046500  052111  020131  043117
       046506  052040  042510  040440
       046514  030460     067
 3417  046517     015     012                  .BYTE   15,12
 3418  046521     106  036440  043040          .ASCII  /F = FINAL ACCEPTANCE (L AND D)/
       046526  047111  046101  040440
       046534  041503  050105  040524
       046542  041516  020105  046050
       046550  040440  042116  042040
       046556     051
 3419  046557     015     012                  .BYTE   15,12
 3420  046561     101  036440  040440          .ASCII  /A = ADJUSTMENT OF A017 AT FIELD SITE/
       046566  045104  051525  046524
       046574  047105  020124  043117
       046602  040440  030460  020067
       046610  052101  043040  042511
       046616  042114  051440  052111
       046624     105
```

```
3421  046625    015     012                       .BYTE   15,12
3422  046627    117  036440  047440               .ASCII  /O = OPERATE WITH TTY AT ADDRESS NNNNNN/
      046634 042520  040522  042524
      046642 053440  052111  020110
      046650 052124  020131  052101
      046656 040440  042104  042522
      046664 051523  047040  047116
      046672 047116     116
3423  046675    015     012                       .BYTE   15,12
3424  046677    123  036440  051440               .ASCII  /S = SOFTWARE SWITCH REGISTER CHANGE/
      046704 043117  053524  051101
      046712 020105  053523  052111
      046720 044103  051040  043505
      046726 051511  042524  020122
      046734 044103  047101  042507
3425  046742    015     012                       .BYTE   15,12
3426  046744 020110  020075  042510               .ASCIZ  /H = HELP THE OPERATOR AND RETYPE THIS LIST/
      046752 050114  052040  042510
      046760 047440  042520  040522
      046766 047524  020122  047101
      046774 020104  042522  054524
      047002 042520  052040  044510
      047010 020123  044514  052123
      047016    000
3427  047017    015     012          LIST1:   .BYTE   15,12
3428  047021    056     000          DOT:     .ASCIZ  /./
3429
3430        047024                              .EVEN
3431
```

```
3433
3434  047024  000005                  POTIME: RESET                   ;CLEAR THE WORLD
3435  047026  104401  052160                  TYPE,   PRIM2           ;TELL OPERATOR ABOUT CABLES
3436  047032  012777  006035  132646          MOV     #6035,@VTCHAR   ;HOME AND ERASE SCREEN
3437  047040  005077  132644                  CLR     @VTYPOS         ;CLEAR THE MAP P.C.
3438  047044  005037  045776                  CLR     CTRCHA          ;CLEAR TTY CHARACTER
3439
3440                                  ;ERASE THE SCREEN MAP
3441  047050  105777  132642          1$:     TSTB    @VTCSR          ;WAIT FOR READY
3442  047054  100375                  BPL     1$                      ;
3443  047056  052777  001000  132632          BIS     #BIT9,@VTCSR    ;CLEAR THE MAP
3444  047064  105777  132626          2$:     TSTB    @VTCSR          ;WAIT FOR READY AGAIN
3445  047070  100375                  BPL     2$
3446  047072  042777  001000  132616          BIC     #BIT9,@VTCSR    ;CLEAR THE ERASE BIT
3447                                  ;NOW LOAD THE REF. PATTERN INTO THE MAP
3448  047100  012701  010421                  MOV     #10421,R1       ;LOAD THE PIXEL VALUE
3449  047104  012700  003740                  MOV     #3740,R0        ;LOAD THE PIXEL ADDRESS
3450  047110  004737  047762          3$:     JSR     PC,DISPLY       ;LOAD THE DATA
3451  047114  005200                  INC     R0                      ;UPDATE THE ADDRESS
3452  047116  022700  004000                  CMP     #4000,R0        ;FINISHED ?
3453  047122  001372                  BNE     3$                      ;BR UNTIL DONE
3454
3455  047124  012700  000000                  MOV     #0,R0           ;LOAD PIXEL ADDRESS
3456  047130  004737  047762          4$:     JSR     PC,DISPLY       ;LOAD THE DATA
3457  047134  005200                  INC     R0                      ;UPDATE THE PIXEL ADDRESS
3458  047136  022700  000040                  CMP     #40,R0          ;TEST IF DONE
3459  047142  001372                  BNE     4$                      ;BR IF DONE THE VERTICAL
3460
3461  047144  012700  007740                  MOV     #7740,R0        ;LOAD PIXEL ADDRESS
3462  047150  004737  047762          5$:     JSR     PC,DISPLY       ;LOAD THE DATA
3463  047154  005200                  INC     R0                      ;UPDATE THE PIXEL ADDRESS
3464  047156  022700  010000                  CMP     #10000,R0       ;TEST IF DONE
3465  047162  001372                  BNE     5$                      ;BR IF DONE
3466
3467                                  ;POSITION THE MAP AND DISPLAY
3468
3469  047164  012777  000005  132524          MOV     #5,@V1CSR       ;LOAD ORGIN VALUE
3470  047172  012777  000060  132524          MOV     #60,@VTINT      ;LOAD LUT 0
3471  047200  012777  000463  132516          MOV     #463,@VTINT     ;LOAD LUT 1
3472  047206  052777  000400  132502          BIS     #BIT8,@V1CSR    ;ENABLE THE MAP
```

```
3474
3475  047214                        ;NOW SAMPLE THE CAMERA CHANNELS
3476                                SAMCAM:
3477                                ;CAMERA 0, X INPUT
3478  047214  004537  047340              JSR     R5,ADJCAM       ;
3479  047220  050044                      ADMSG0                  ;CAM 0 MESSAGE
3480  047222  001730                      DAC0                    ;DAC ADDRESS
3481  047224     000     000              .BYTE   0,0             ;CAMERA 0, X AXIS
3482                                ;DO CAMERA 1, X INPUT
3483  047226  004537  047340              JSR     R5,ADJCAM
3484  047232  050230                      ADMSG2                  ;CAM 1, X MESSAGE
3485  047234  001732                      DAC1                    ;DAC ADDRESS
3486  047236     001     000              .BYTE   1,0             ;CAMERA 1, X AXIS
3487                                ;DO CAMERA 2, X INPUT
3488  047240  004537  047340              JSR     R5,ADJCAM
3489  047244  050414                      ADMSG4                  ;CAM 2, X MESSAGE
3490  047246  001734                      DAC2                    ;DAC ADDRESS
3491  047250     002     000              .BYTE   2,0             ;CAMERA 2, X AXIS
3492                                ;DO CAMERA 3, X INPUT
3493  047252  004537  047340              JSR     R5,ADJCAM
3494  047256  050600                      ADMSG6                  ;CAM 3, X MESSAGE
3495  047260  001736                      DAC3                    ;DAC ADDRESS
3496  047262     003     000              .BYTE   3,0             ;CAMERA 3, X AXIS
3497                                ;NOW SAMPLE THE Y CAMERA CHANNELS
3498                                ;DO CAMERA 0 Y INPUT
3499  047264  004537  047340              JSR     R5,ADJCAM
3500  047270  050136                      ADMSG1                  ;CAM 0 Y MESSAGE
3501  047272  001732                      DAC1                    ;DAC ADDRESS
3502  047274     000     001              .BYTE   0,1             ;CAMERA 0, Y AXIS
3503                                ;DO CAMERA 1, Y INPUT
3504  047276  004537  047340              JSR     R5,ADJCAM
3505  047302  050322                      ADMSG3                  ;CAM 1, Y MESSAGE
3506  047304  001730                      DAC0                    ;DAC ADDRESS
3507  047306     001     001              .BYTE   1,1             ;CAMERA 1, Y AXIS
3508                                ;DO CAMERA 2, Y INPUT
3509  047310  004537  047340              JSR     R5,ADJCAM
3510  047314  050506                      ADMSG5                  ;CAM 2, Y MESSAGE
3511  047316  001736                      DAC3                    ;DAC ADDRESS
3512  047320     002     001              .BYTE   2,1             ;CAMERA 2, Y AXIS
3513                                ;DO CAMERA 3, Y INPUT
3514  047322  004537  047340              JSR     R5,ADJCAM
3515  047326  050672                      ADMSG7                  ;CAM 3, Y MESSAGE
3516  047330  001734                      DAC2                    ;DAC ADDRESS
3517  047332     003     001              .BYTE   3,1             ;CAMERA 3 Y AXIS
3518
3519  047334  000137  002646              JMP     RBEG0
```

```
                                        ;REPORT THE MESSAGE FIRST
                                
047340  012500                  ADJCAM: MOV     (R5)+,R0        ;GET ASCII POINTER
047342  005777  132340          3$:     TST     @VTCHAR         ;WAIT FOR READY
047346  100375                          BPL     3$
047350  012777  012000  132336          MOV     #12000,@VTCXY   ;POSITION THE CAHRACTERS
047356  005777  132324          1$:     TST     @VTCHAR         ;WAIT FOR CHAR. READY
047362  100375                          BPL     1$
047364  112077  132316                  MOVB    (R0)+,@VTCHAR   ;LOAD THE CHAR.
047370  105710                          TSTB    (R0)            ;TEST IF TERM.
047372  001371                          BNE     1$              ;BR IF NOT
                                
047374  013537  050022                  MOV     @(R5)+,DACADR   ;SAVE ADDRESS
047400  012537  050030                  MOV     (R5)+,CAMVAL    ;SAVE CAMERA DATA
                                
047404  012777  002315  000410  2$:     MOV     #2315,@DACADR   ;PRESET DAC TO - GAIN INPUT
047412  004737  047514                  JSR     PC,SAMDAT
047416  010137  050036                  MOV     R1,VAL0         ;SAVE RESULTS
047422  004737  047740                  JSR     PC,FIXLIN       ;POSITION THE CROSS HAIR LINE
047426  012777  004000  000366          MOV     #4000,@DACADR   ;PRESET DAC TO OFFSET
047434  004737  047514                  JSR     PC,SAMDAT       ;SAMPLE THE DATA
047440  010137  050040                  MOV     R1,VAL1         ;SAVE AVERAGE RESULTS
047444  004737  047740                  JSR     PC,FIXLIN       ;POSITION THE CROSS HAIR LINE
047450  012777  005463  000344          MOV     #5463,@DACADR   ;PRESET DAC TO + GAIN INPUT
047456  004737  047514                  JSR     PC,SAMDAT       ;SAMPLE THE DATA
047462  010137  050042                  MOV     R1,VAL2         ;SAVE AVERAGE RESULTS
047466  004737  047740                  JSR     PC,FIXLIN       ;POSITION THE CROSS HAIR LINE
047472  004737  045700                  JSR     PC,CTRLCG       ;TEST FOR CTRL C/G
047476  122737  000103  045776          CMPB    #'C,CTRCHA      ;CHECK IF CHARACTER WAS A \C\
047504  001337                          BNE     2$              ;BR IF NOT
047506  005037  045776                  CLR     CTRCHA          ;CLEAR CHAR
047512  000205                          RTS     R5              ;EXIT
                                
                                ;COLLECT 64 SAMPLES FROM THE SELECTED CAMERA
                                ;R1 CONTAINS THE AVERAGE
                                
047514  012700  060000          SAMDAT: MOV     #BUF0,R0
047520  012701  062000                  MOV     #BUF1,R1
047524  005020                  1$:     CLR     (R0)+           ;CLEAR THE BUFFER
047526  020001                          CMP     R0,R1           ;FINISHED ?
047530  001375                          BNE     1$              ;BR IF NOT
                                
047532  012777  177700  132212          MOV     #-100,@WCR      ;LOAD WORD COUNT
047540  012777  060000  132206          MOV     #BUF0,@BAR      ;LOAD ADDRESS
047546  113777  050030  132212          MOVB    CAMVAL,@CSRHB   ;SELECT THE CAMERA
047554  052777  000002  132174          BIS     #TESTZ,@SFR     ;SET TEST Z ENABLE
047562  052777  000001  132156          BIS     #BIT0,@CSR      ;ENABLE NCV11
                                
047570  105777  132152          2$:     TSTB    @CSR            ;WAIT UNTIL DONE
047574  100375                          BPL     2$
                                
047576  012777  004000  132152          MOV     #CLRALL,@SFR    ;CLEAR THE DEVICE
047604  005037  050016                  CLR     $TEMP1
047610  005037  050020                  CLR     $TEMP2
047614  005001                          CLR     R1
047616  012700  060000                  MOV     #BUF0,R0        ;LOAD BUFFER POINTER
```

```
3576  047622  105737  050031                TSTB    CAMVAL+1        ;TEST IF X OR Y DATA
3577  047626  001413                      BEQ     5$              ;BR IF X
3578  047630  012037  050016        3$:   MOV     (R0)+,$TEMP1    ;GET THE DATA
3579  047634  113737  050017  050020      MOVB    $TEMP1+1,$TEMP2 ;GET HIGH BYTE DATA
3580  047642  063701  050020              ADD     $TEMP2,R1       ;UPDATE AVERAGE
3581  047646  022700  060200              CMP     #BUF0+200,R0    ;FINISHED DATA ?
3582  047652  001366                      BNE     3$              ;BR IF NOT
3583  047654  000412                      BR      10$
3584  047656  012037  050016        5$:   MOV     (R0)+,$TEMP1    ;GET DATA
3585  047662  113737  050016  050020      MOVB    $TEMP1,$TEMP2
3586  047670  063701  050020              ADD     $TEMP2,R1       ;UPDATE THE AVERAGE
3587  047674  022700  060200              CMP     #BUF0+200,R0    ;FINISHED DATA ?
3588  047700  001366                      BNE     5$              ;BR IF NOT
3589  047702  000240                10$:  NOP
3590  047704  000240                      NOP
3594  047706  006201                      ASR     R1              ;AVERAGE THE DATA
  (1) 047710  000240                      NOP
  (1) 047712  006201                      ASR     R1              ;AVERAGE THE DATA
  (1) 047714  000240                      NOP
  (1) 047716  006201                      ASR     R1              ;AVERAGE THE DATA
  (1) 047720  000240                      NOP
  (1) 047722  006201                      ASR     R1              ;AVERAGE THE DATA
  (1) 047724  000240                      NOP
  (1) 047726  006201                      ASR     R1              ;AVERAGE THE DATA
  (1) 047730  000240                      NOP
  (1) 047732  006201                      ASR     R1              ;AVERAGE THE DATA
  (1) 047734  000240                      NOP
3595  047736  000207                      RTS     PC              ;EXIT
3596
3597                                ;SUBROUTINE TO LOAD X OR Y CROSSHAIRS
3598  047740  006201                FIXLIN: ASR     R1
3599  047742  000240                      NOP
3600  047744  000240                      NOP
3601  047746  000240                      NOP
3602  047750  062701  000100              ADD     #100,R1
3603  047754  110177  131732              MOVB    R1,@VTXPOS
3604  047760  000207                      RTS     PC
3605
3606                                ;SUBROUTINE TO LOAD PIXEL DATA
3607  047762  042777  000400  131726  DISPLY: BIC   #400,@VTCSR     ;DISABLE MAP
3608  047770  010077  131724              MOV     R0,@VTMAP       ;LOAD MAP PC
3609  047774  105777  131716        1$:   TSTB    @VTCSR          ;WAIT FOR MAP READY
3610  050000  100375                      BPL     1$
3611  050002  010177  131714              MOV     R1,@VTPX        ;LOAD PIXEL DATA
3612  050006  052777  000400  131702      BIS     #400,@VTCSR     ;ENABLE MAP
3613  050014  000207                      RTS     PC
3614
3615  050016  000000                $TEMP1: 0
3616  050020  000000                $TEMP2: 0
3617  050022  000000                DACADR: 0
3618  050024  000000                RUNDIF: 0
3619  050026  000000                WFMODE: 0               ;WF OPTION AREA FLAG
3620  050030  000000                CAMVAL: 0
3621  050032  000000                CURENT: 0
3622  050034  000000                AGAN:   0
3623  050036  000000                VAL0:   0
```

```
3624  050040  000000                    VAL1:   0
3625  050042  000000                    VAL2:   0
3626  050044  040503  042515  040522  ADMSG0: .ASCII  /CAMERA 00/
      050052  030040    060
3627  050055    015     012                    .BYTE   15,12
3628  050057    101  045104  051525           .ASCII  /ADJUST R09 FOR X OFFSET/
      050064  020124  030122  020071
      050072  047506  020122  020130
      050100  043117  051506  052105
3629  050106    015     012                    .BYTE   15,12
3630  050110  042101  052512  052123           .ASCIZ  /ADJUST R24 FOR X GAIN/
      050116  051040  032062  043040
      050124  051117  054040  043440
      050132  044501  000116
3631  050136  040503  042515  040522  ADMSG1:  ASCII  /CAMERA 00/
      050144  030040    060
3632  050147    015     012                    .BYTE   15,12
3633  050151    101  045104  051525           .ASCII  /ADJUST R13 FOR Y OFFSET/
      050156  020124  030522  020063
      050164  047506  020122  020131
      050172  043117  051506  052105
3634  050200    015     012                    .BYTE   15,12
3635  050202  042101  052512  052123           .ASCIZ  /ADJUST R20 FOR Y GAIN/
      050210  051040  030062  043040
      050216  051117  054440  043440
      050224  044501  000116
3636  050230  040503  042515  040522  ADMSG2: .ASCII  /CAMERA 01/
      050236  030040    061
3637  050241    015     012                    .BYTE   15,12
3638  050243    101  045104  051525           .ASCII  /ADJUST R10 FOR X OFFSET/
      050250  020124  030522  020060
      050256  047506  020122  020130
      050264  043117  051506  052105
3639  050272    015     012                    .BYTE   15,12
3640  050274  042101  052512  052123           .ASCIZ  /ADJUST R23 FOR X GAIN/
      050302  051040  031462  043040
      050310  051117  054040  043440
      050316  044501  000116
3641  050322  040503  042515  040522  ADMSG3: .ASCII  /CAMERA 01/
      050330  030040    061
3642  050333    015     012                    .BYTE   15,12
3643  050335    101  045104  051525           .ASCII  /ADJUST R14 FOR Y OFFSET/
      050342  020124  030522  020064
      050350  047506  020122  020131
      050356  043117  051506  052105
3644  050364    015     012                    .BYTE   15,12
3645  050366  042101  052512  052123           .ASCIZ  /ADJUST R19 FOR Y GAIN/
      050374  051040  034461  043040
      050402  051117  054440  043440
      050410  044501  000116
3646  050414  040503  042515  040522  ADMSG4: .ASCII  /CAMERA 02/
      050422  030040    062
3647  050425    015     012                    .BYTE   15,12
3648  050427    101  045104  051525           .ASCII  /ADJUST R11 FOR X OFFSET/
      050434  020124  030522  020061
      050442  047506  020122  020130
```

```
        050450  043117  051506  052105
3649    050456     015     012                  .BYTE   15,12
3650    050460  042101  052512  052123          .ASCIZ  /ADJUST R22 FOR X GAIN/
        050466  051040  031062  043040
        050474  051117  054040  043440
        050502  044501  000116
3651    050506  040503  042515  040522  ADMSG5: .ASCII  /CAMERA 02/
        050514  030040     062
3652    050517     015     012                  .BYTE   15,12
3653    050521     101  045104  051525          .ASCII  /ADJUST R15 FOR Y OFFSET/
        050526  020124  030522  020065
        050534  047506  020122  020131
        050542  043117  051506  052105
3654    050550     015     012                  .BYTE   15,12
3655    050552  042101  052512  052123          .ASCIZ  /ADJUST R18 FOR Y GAIN/
        050560  051040  034061  043040
        050566  051117  054440  043440
        050574  044501  000116
3656    050600  040503  042515  040522  ADMSG6: .ASCII  /CAMERA 03/
        050606  030040     063
3657    050611     015     012                  .BYTE   15,12
3658    050613     101  045104  051525          .ASCII  /ADJUST R12 FOR X OFFSET/
        050620  020124  030522  020062
        050626  047506  020122  020130
        050634  043117  051506  052105
3659    050642     015     012                  .BYTE   15,12
3660    050644  042101  052512  052123          .ASCIZ  /ADJUST R21 FOR X GAIN/
        050652  051040  030462  043040
        050660  051117  054040  043440
        050666  044501  000116
3661    050672  040503  042515  040522  ADMSG7: .ASCII  /CAMERA 03/
        050700  030040     063
3662    050703     015     012                  .BYTE   15,12
3663    050705     101  045104  051525          .ASCII  /ADJUST R16 FOR Y OFFSET/
        050712  020124  030522  020066
        050720  047506  020122  020131
        050726  043117  051506  052105
3664    050734     015     012                  .BYTE   15,12
3665    050736  042101  052512  052123          .ASCIZ  /ADJUST R17 FOR Y GAIN/
        050744  051040  033461  043040
        050752  051117  054440  043440
        050760  044501  000116
3666
3667    050764     015     012          PRIMO:  .BYTE   15,12
3668    050766  044124  020105  042523          .ASCII  /THE SELF-TEST CONNECTOR MUST BE INSTALLED ON A017/
        050774  043114  052055  051505
        051002  020124  047503  047116
        051010  041505  047524  020122
        051016  052515  052123  041040
        051024  020105  047111  052123
        051032  046101  042514  020104
        051040  047117  040440  030460
        051046     067
3669    051047     015     012                  .BYTE   15,12
3670    051051     124  042510  051440          .ASCII  \THE SWITCH ON A017 MUST BE IN 'MAINT.'' POSITION (TOWARD THE I/O CONNECT
        051056  044527  041524  020110
```

```
                051064  047117  040440  030460
                051072  020067  052515  052123
                051100  041040  020105  047111
                051105  021040  040515  047111
                051114  027124  020042  047520
                051122  044523  044524  047117
                051130  024040  047524  040527
                051136  042122  052040  042510
                051144  044440  047457  041440
                051152  047117  042516  052103
                051160  051117     051
 3671   051163     015     012     000          .BYTE   15,12,0
 3672   051166     015     012          PRIM6:  .BYTE   15,12
 3673   051170  044124  020105  042523          .ASCII  /THE SELF-TEST CONNECTOR MUST BE REMOVED FROM THE A017/
                051176  043114  052055  051505
                051204  020124  047503  047116
                051212  041505  047524  020122
                051220  052515  052123  041040
                051226  020105  042522  047515
                051234  042526  020104  051106
                051242  046517  052040  042510
                051250  040440  030460     067
 3674   051255     015     012                  .BYTE   15,12
 3675   051257     124  042510  051440          .ASCII  \THE SWITCH ON A017 MUST BE IN "OPER." POSITION (AWAY FROM THE I/O CONNE
                051264  044527  041524  020110
                051272  047117  040440  030460
                051300  020067  052515  052123
                051306  041040  020105  047111
                051314  021040  050117  051105
                051322  021056  050040  051517
                051330  052111  047511  020116
                051336  040450  040527  020131
                051344  051106  046517  052040
                051352  042510  044440  047457
                051360  041440  047117  042516
                051366  052103  051117     051
 3676   051373     015     012     000          .BYTE   15,12,0
 3677   051376  047111  042523  052122  PRIM5:  .ASCII  /INSERT RAM-PAK CABLE INTO THE SOCKET ON THE A017/
                051404  051040  046501  050055
                051412  045501  041440  041101
                051420  042514  044440  052116
                051426  020117  044124  020105
                051434  047523  045503  052105
                051442  047440  020116  044124
                051450  020105  030101  033461
 3678   051456     015     012                  .BYTE   15,12
 3679   051460  042523  020124  040522          .ASCII  /SET RAM-PAK SWITCHES TO "IN", "NORM" AND "OFF"/
                051466  026515  040520  020113
                051474  053523  052111  044103
                051502  051505  052040  020117
                051510  044442  021116  020054
                051516  047042  051117  021115
                051524  040440  042116  021040
                051532  043117  021106
 3680   051536     015     012                  .BYTE   15,12
 3681   051540  047111  042523  052122  PRIM3:  .ASCII  /INSERT BLANK ROM INTO ROM BLASTER SOCKET/
```

```
                051546  041040  040514  045516
                051554  051040  046517  044440
                051562  052116  020117  047522
                051570  020115  046102  051501
                051576  042524  020122  047523
                051604  045503  052105
 3682           051610     015     012                  .BYTE   15,12
 3683           051612  046120  040505  042523          .ASCII  \PLEASE DEPRESS THE ''I/O'' AND ''EXECUTE'' BUTTONS TOGETHER\
                051620  042040  050105  042522
                051626  051523  052040  042510
                051634  021040  027511  021117
                051642  040440  042116  021040
                051650  054105  041505  052125
                051656  021105  041040  052125
                051664  047524  051516  052040
                051672  043517  052105  042510
                051700     122
 3684           051701     015     012                  .BYTE   15,12
 3685           051703     124  042510  032040          .ASCII  \THE 4 PROCESS LEDS WILL LIGHT WHEN DONE CORRECTLY\
                051710  050040  047522  042503
                051716  051523  046040  042105
                051724  020123  044527  046114
                051732  046040  043511  052110
                051740  053440  042510  020116
                051746  047504  042516  041440
                051754  051117  042522  052103
                051762  054514
 3686           051764     015     012     000          .BYTE   15,12,0
 3687           051767     015     012                  PRIM1:  .BYTE   15,12
 3688           051771     120  042514  051501          .ASCII  /PLEASE REMOVE RAM-PAK CABLE/
                051776  020105  042522  047515
                052004  042526  051040  046501
                052012  050055  045501  041440
                052020  041101  042514
 3689           052024     015     012                  .BYTE   15,12
 3690           052026  040440  042116  044440          .ASCII  / AND INSERT THE BLASTED ROM INTO SOCKET ON A017 MODULE/
                052034  051516  051105  020124
                052042  044124  020105  046102
                052050  051501  042524  020104
                052056  047522  020115  047111
                052064  047524  051440  041517
                052072  042513  020124  047117
                052100  040440  030460  020067
                052106  047515  052504  042514
 3691           052114     015     012                  .BYTE   15,12
 3692           052116  042504  051120  051505          .ASCII  /DEPRESS THE ''CR'' KEY WHEN READY/
                052124  020123  044124  020105
                052132  041442  021122  045440
                052140  054505  053440  042510
                052146  020116  042522  042101
                052154     131
 3693           052155     015     012     000          .BYTE   15,12,0
 3694           052160     015     012                  PRIM2:  .BYTE   15,12
 3695           052162  050042  030512  020042          .ASCII  /''PJ1'' CABLE MUST BE CONNECTED TO THE A017 CONNECTOR/
                052170  040503  046102  020105
                052176  052515  052123  041040
```

```
        052204  020105  047503  047116
        052212  041505  042524  020104
        052220  047524  052040  042510
        052226  040440  030460  020067
        052234  047503  047116  041505
        052242  047524     122
 3696   052245     015     012     000          .BYTE   15,12,0
 3697   052250     015     012          PRIM4:  .BYTE   15,12
 3698   052252  050042  031512  020042          .ASCII  /"PJ3" CABLE MUST BE CONNECTED TO THE M8036 CONNECTOR/
        052260  040503  046102  020105
        052266  052515  052123  041040
        052274  020105  047503  047116
        052302  041505  042524  020104
        052310  047524  052040  042510
        052316  046440  030070  033063
        052324  041440  047117  042516
        052332  052103  051117
 3699   052336     015     012     000          .BYTE   15,12,0
 3700           052342                          .EVEN
 3701   052342     015     012     007  FATAL0: .BYTE   15,12,7 ;"RUNIT" MUST BE ON A WORD BOUNDRY
 3702   052345     106  052101  046101          .ASCII  /FATAL BUS TRAP WHEN PERFORMING TEST "/
        052352  041040  051525  052040
        052360  040522  020120  044127
        052366  047105  050040  051105
        052374  047506  046522  047111
        052402  020107  042524  052123
        052410  021040
 3703   052412  021103  040411  020124  RUNIT:  .ASCIZ  /C"     AT LOCATION /   ;"RUNIT" MUST BE A WORD BOUNDRY
        052420  047514  040503  044524
        052426  047117  000040
 3704   052432  005015  051120  043517  RUNITA: .ASCIZ  <15><12>/PROGRAM RESTARTING/    ;"RUNIT" MUST BE A WORD BOUNDRY
        052440  040522  020115  042522
        052446  052123  051101  044524
        052454  043516     000
 3705   052457     015     012     007          .BYTE   15,12,7,0
        052462     000
 3706   052463     015     012          MSGMEM: .BYTE   15,12
 3707   052465     122  047125  044516          .ASCIZ  /RUNNING WITH /
        052472  043516  053440  052111
        052500  020110     000
 3708   052503     040  020113  043117  MSGK:   .ASCIZ  / K OF MEMORY/<15><12>
        052510  046440  046505  051117
        052516  006531  000012
 3709   052522     015     012     007  WARN0:  .BYTE   15,12,7
 3710   052525     117  052103  046101          .ASCIZ  /OCTAL ADDRESS OF TERMINAL   /
        052532  040440  042104  042522
        052540  051523  047440  020106
        052546  042524  046522  047111
        052554  046101  036440  000040
 3711   052562     033     000          ESCP:   .BYTE   33,0
 3712   052564  043113  027462     062  KF0:    .ASCII  \KF2/2\
 3713   052571     015     000                  .BYTE   15,0
 3714   052573     106  031515     060  FM0:    .ASCII  \FM30\
 3715   052577     015     000                  .BYTE   15,0
 3716   052601     104  030111  031457  DIO:    .ASCII  \DIO/37\
        052606     067
```

```
 3717  052607      015     002                    .BYTE   15,2
 3718  052611                          DIDATA:
 3721  052611      060     060     060            .BYTE   60,60,60,40
  (1)  052614      040
  (1)  052615      060     060     060            .BYTE   60,60,60,40
  (1)  052620      040
  (1)  052621      060     060     060            .BYTE   60,60,60,40
  (1)  052624      040
  (1)  052625      060     060     060            .BYTE   60,60,60,40
  (1)  052630      040
  (1)  052631      060     060     060            .BYTE   60,60,60,40
  (1)  052634      040
  (1)  052635      060     060     060            .BYTE   60,60,60,40
  (1)  052640      040
  (1)  052641      060     060     060            .BYTE   60,60,60,40
  (1)  052644      040
  (1)  052645      060     060     060            .BYTE   60,60,60,40
  (1)  052650      040
  (1)  052651      060     060     060            .BYTE   60,60,60,40
  (1)  052654      040
  (1)  052655      060     060     060            .BYTE   60,60,60,40
  (1)  052660      040
  (1)  052661      060     060     060            .BYTE   60,60,60,40
  (1)  052664      040
  (1)  052665      060     060     060            .BYTE   60,60,60,40
  (1)  052670      040
  (1)  052671      060     060     060            .BYTE   60,60,60,40
  (1)  052674      040
  (1)  052675      060     060     060            .BYTE   60,60,60,40
  (1)  052700      040
  (1)  052701      060     060     060            .BYTE   60,60,60,40
  (1)  052704      040
  (1)  052705      060     060     060            .BYTE   60,60,60,40
  (1)  052710      040
  (1)  052711      060     060     060            .BYTE   60,60,60,40
  (1)  052714      040
  (1)  052715      060     060     060            .BYTE   60,60,60,40
  (1)  052720      040
  (1)  052721      060     060     060            .BYTE   60,60,60,40
  (1)  052724      040
  (1)  052725      060     060     060            .BYTE   60,60,60,40
  (1)  052730      040
  (1)  052731      060     060     060            .BYTE   60,60,60,40
  (1)  052734      040
  (1)  052735      060     060     060            .BYTE   60,60,60,40
  (1)  052740      040
  (1)  052741      060     060     060            .BYTE   60,60,60,40
  (1)  052744      040
  (1)  052745      060     060     060            .BYTE   60,60,60,40
  (1)  052750      040
  (1)  052751      060     060     060            .BYTE   60,60,60,40
  (1)  052754      040
  (1)  052755      060     060     060            .BYTE   60,60,60,40
  (1)  052760      040
  (1)  052761      060     060     060            .BYTE   60,60,60,40
  (1)  052764      040
```

```
   (1)  052765     060     060     060          .BYTE   60,60,60,40
   (1)  052770     040
   (1)  052771     060     060     060          .BYTE   60,60,60,40
   (1)  052774     040
   (1)  052775     060     060     060          .BYTE   60,60,60,40
   (1)  053000     040
   (1)  053001     060     060     060          .BYTE   60,60,60,40
   (1)  053004     040
   (1)  053005     060     060     060          .BYTE   60,60,60,40
   (1)  053010     040
3722    053011     003     000                  .BYTE   3,0
3723    053013     103  030115  031457  CM0:    .ASCII  \CM0/37\
        053020     067
3724    053021     015     000                  .BYTE   15,0
3725    053023     103  000103          CC0:    .ASCIZ  /CC/
3726    053026  043520  027460  033463  PG0:    .ASCII  \PG0/37\
3727    053034     015     000                  .BYTE   15,0
3728    053036  050132                  ZP0.    .ASCII  /ZP/
3729    053040     015     000                  .BYTE   15,0
3730                                            .EVEN
3731
3732                                    ;BUFFER AREA
3733
3734            060000                          .=60000
3735    060000  001000          BUF0:   .BLKW   512.
3736    062000  001000          BUF1:   .BLKW   512.
3737    064000  000240          BUF2:   NOP
3738
3739            000001                  .END
```

```
ABASE = 172760            16#    32    98    99   100   101   102   103   104   105   106   107
ABOMSG  041634          2627#  3240
ABORTB  041515          2618#  2773
ABOVE   041724          2643#  3139*  3204  3238
ACDW1 = 170420            18#    32    92    93    94
ACDW2 = 000000            32
ACPUOP= 000000            32
ADDW0 = 000000            32
ADDW1 = 000000            32
ADDW10= 000000            32
ADDW11= 000000            32
ADDW12= 000000            32
ADDW13= 000000            32
ADDW14= 000000            32
ADDW15= 000000            32
ADDW2 = 000000            32
ADDW3 = 000000            32
ADDW4 = 000000            32
ADDW5 = 000000            32
ADDW6 = 000000            32
ADDW7 = 000000            32
ADDW8 = 000000            32
ADDW9 = 000000            32
ADEVCT= 000000            32
ADEVM = 00000U            32
ADJCAM  047340          3477   3482   3487   3492   3498   3503   3508   3513   3522#
ADJFIX  043362          2766   2918#
ADM     001760           103#   358    956    957    958    960    961    962    964   965   966   968   969
                         970    972    973    974    976    977    978    980    981   982   991  1005  1019
ADMSG0  050044          3478   3626#
ADMSG1  050136          3499   3631#
ADMSG2  050230          3483   3636#
ADMSG3  050322          3504   3641#
ADMSG4  050414          3488   3646#
ADMSG5  050506          3509   3651#
ADMSG6  050600          3493   3656#
ADMSG7  050672          3514   3661#
ADNOKT  003354           294#   310*   1915   1960   1985   2027   2043   2050   2081
AENV  = 000000            32
AENVM = 000000            32
AFATAL= 000000            32
AGAN    050034           140*   143*   145*   180   3622#
AMADR1= 000000            32
AMADR2= 000000            32
AMADR3= 000000            32
AMADR4= 000000            32
AMAMS1= 000000            32
AMAMS2= 000000            32
AMAMS3= 000000            32
AMAM[illegible] 000000    32
AMSGAD= 000000            32
AMSGLG= 000000            32
AMSGTY= 000000            32
AMTYP1= 000000            32
AMTYP2= 000000            32
AMTYP3= 000000            32
```

```
AMTYP4= 000000            32
APASS = 000000            32
APRIOR= 000000            32
APTCSU= 000040          2342    2350#
APTENV= 000001          2342    2346    2350#
APTSIZ= 000200           147    2350#
APTSPO= 000100          2342    2350#
ASWREG= 000000            32
ATESTN= 000000            32
AUNIT = 000000            32
AUSWR = 000000            32
AVECT1= 140370           17#      32     111     112     113     114
AVECT2= 000000            32
AVGVAL  041716          2640#    2924    2925    2935    3000    3088*   3090    3096    3163
AVRGO   041416          2615#    3162
BAR     001754           101#     356     490*    491     517*    532     596*    601     617*    622     641*    659*    672*
                         676      684*    715*    749*    795*   1034*   1052    1084*   1104    1129*   1159*   1188*   1222*
                        1243*    1265*   1286*   1310*   1331*   1353*   1432*   1474*   1510*   1548*   1768*   1804*   1805*
                        1811     1828*   1850*   1856    1870*   1876    1890*   1896    1924*   1992*   2057*   2157*   2499*
                        3051*    3287*   3563*
BAR1    001764           105#     359
BEGIN   002034            22      145#    225     228
BELMSG  041574          2625#    3224
BELOW   041722          2642#    3137*   3205    3222
BIT0  = 000001            13#     131     473     486     499     566     582     598     619     643     661     686     717
                         751      798     956     957     958     960     961     962     964     965     966     968     969
                         970      972     973     974     976     977     978     980     981     982     989    1003    1017
                        1036     1086    1132    1161    1198    1226    1247    1268    1289    1313    1334    1356    1376
                        1410     1436    1479    1515    1554    1612    1643    1615    1616    1618    1619    1621    1622
                        1624     1625    1627    1628    1641    1673    1674    1689    1715    1731    1753    1772    1807
                        1808     1830    1831    1852    1853    1872    1873    1892    1893    1928    1994    2059    2114
                        2118     2120    2159    2165    2167    2197    2207    2209    2225    2255    2265    2267    2421
                        2435     2504    2733    2810    2817    2823    3054    3055    3293    3566
BIT00 = 000001            13#
BIT01 = 000002            13#
BIT02 = 000004            13#
BIT03 = 000010            13#
BIT04 = 000020            13#
BIT05 = 000040            13#
BIT06 = 000100            13#
BIT07 = 000200            13#
BIT08 = 000400            13#    2341
BIT09 = 001000            13#    2341    2346
BIT1  = 000002            13#     130     563     565     574     597     618     642     660     685     716     738     750
                         797      801    1374    1434    1448    1458    1770    1783    1806    1829    1851    1871    1891
                        2112     2448    3018    3025    3032
BIT10 = 002000            13#    1272    1360    1770    1783    2346    2383    2394    2405    2416
BIT11 = 004000            13#     126     383    1910    1973    2341
BIT12 = 010000            13#    1041    1089    1137    1166    1203    1228    1252    1273    1294    1317    1340    1361
                        1381     1414    1443    1485    1525    1559    1612    1613    1615    1616    1618    1619    1621
                        1622     1624    1625    1627    1628    1645    1676    1694    1718    1734    1756    1779    2117
                        2163     2201    2228    2259
BIT13 = 020000            13#    1421    1425    1448    1456    1490    1533    1783    1932    1939    1980    1999    2006
                        2040     2062    2069    2346
BIT14 = 040000            13#     125     610     633     652     691     693     723     725     756     758     801    1141
                        1170     1179    1932    1939    1999    2006    2062    2069    2341
```

```
BIT15 = 100000   13#  1141  1170  1178
BIT2  = 000004   13#   129  1408  1423  1477  1490  1513  1533  1612  1613  1615  1616  1618
                1619  1621  1622  1624  1625  1627  1628  2461
BIT3  = 000010   13#   128  2474
BIT4  = 000020   13#   433   563   565   574   597   608   618   631   642   650   660   685
                 691   716   723   738   750   756   797   801  1374  1408  1423  1434  1448
                1458  1477  1490  1513  1533  1612  1613  1615  1616  1618  1619  1621  1622
                1624  1625  1627  1628  1691  1770  1783  1806  1829  1851  1871  1891  2112
                2223  2481  3053
BIT5  = 000040   13#  1513  1533
BIT6  = 000100   13#   774  1198  2429
BIT7  = 000200   13#   311   388   400   409   574   581   705   738   801   826   845   850
                1141  1150  1170  1179  1497  1533  2442
BIT8  = 000400   13#   127   621   663  1316  1851  1891  2389  2394  2411  2416  2455  3472
BIT9  = 001000   13#  1293  1360  1871  1891  2400  2405  2411  2416  2468  3443  3446
BLAST   042274   205  2731#
BLICNT  042132  2673*  2677  2680*  2684*  2704#
BPTVEC= 000014   13#
BRLEV   002002   118#   167*  1542  1566
BTALK   042022   233  2682#
BUFO    060000  1031*  1034  1053  1070  1072  1079  1084  1116  1122  1265  1267*  1277  1286
                1288*  1298  1310  1312*  1320  1331  1333*  1343  1353  1355*  1365  1372*  1373
                1385  1407  1411*  1412*  1416  1433  1441*  1442*  1464  1476  1484*  1502  1512
                1520*  1550  1553*  1638  1643*  1644*  1649  1654  1659  1669  1675*  1677  1686
                1692*  1693*  1699  1702  1706  1717*  1719  1733*  1735  1755*  1757  1769  1777*
                1778*  1789  1909  1913  1942  1970  1979  1983  2009  2037  2044  2048  2057
                2072  2091  2138  2186  2227*  2229  2244  2499  2523  2682  3004  3049  3063
                3287  3297  3556  3563  3575  3581  3587  3735#
BUF1    062000  1081  1105  2918  3064  3075  3081  3098  3557  3736*
BUF2    064000  3006  3072  3737#
BUSTRP  003212   152   258#
CALRPT  037616  2420  2428#
CAMUNP  040362  2432  2438  2445  2451  2458  2464  2471  2477  2484  2549#
CAMVAL  050030  3533*  3564  3576  3620#
CAMOG1  040444  2379  2433  2566#
CAMOG2  040446  2384  2439  2567#
CAMOTW  040505  2437  2578#
CAMOTX  040466  2431  2576#
CAM1G1  040450  2390  2446  2568#
CAM1G2  040452  2395  2452  2569#
CAM1TW  040543  2450  2582#
CAM1TX  040524  2444  2580#
CAM2G1  040454  2401  2459  2570#
CAM2G2  040456  2406  2465  2571#
CAM2TW  040601  2463  2586#
CAM2TX  040562  2457  2584#
CAM3G1  040460  2412  2472  2572#
CAM3G2  040462  2417  2478  2573#
CAM3TW  040637  2476  2590#
CAM3TX  040620  2470  2588#
CCO     053023  2781  2814  2835  3725#
CKSWR = 104407  2341  2346  2366#
CKTSWR  040064  2428  2434  2441  2447  2454  2460  2467  2473  2480  2488#
CLRALL= 004000   126#   408   465   476   489   502   514   546   550   552   558   583   593
                 614   638   656   670   681   704   712   746   780   786   792   818   837
                 848   875   880   885   890   895   900   908   913   919   925   931   936
```

```
                        956   957   958   960   961   962   964   965   966   968   969   970   972
                        973   974   976   977   978   980   981   982   985   999  1013  1030  1046
                       1078  1103  1126  1149  1156  1185  1213  1218  1239  1263  1284  1306  1327
                       1351  1371  1406  1430  1473  1495  1509  1537  1545  1587  1612  1613  1615
                       1616  1618  1619  1621  1622  1624  1625  1627  1628  1637  1668  1685  1711
                       1727  1747  1766  1801  1824  1847  1867  1887  1921  1990  2055  2110  2154
                       2193  2219  2251  2497  3048  3284  3340  3571
CLRWCO= 040000          125#   737
CMO     053013         2779  3723#
COMP    041172         2610#  2791  2841
CONROM  043246         2808   2830  2886#
CONVRT  040106         2377   2382  2388  2393  2399  2404  2410  2415  2496#
CPUDL0  002006          121#   689   721   754
CPUDL1  002010          122#   820   839
CPUDL2  002012          123#  1930  1997  2061
CR      = 000015         13#  2342
CRLF    = 000200         13#   149  2342
CSR     001746           98#   153   353   385*  386   398*  401   407*  410   417*  419*  420   424*
                        427   519*  542   563*  566*  567   575   582*  584   597*  598*  609   618*
                        619*  632   642*  643*  651   660*  661*  673*  685*  686*  693   706   716*
                        717*  725   739   750*  751*  758   774*  779   785*  797*  798*  800   875*
                        880*  885*  890*  895*  900*  908*  913*  919*  925*  931*  936*  956*  957*
                        958*  960*  961*  962*  964*  965*  966*  968*  969*  970*  972*  973*  974*
                        976*  977*  978*  980*  981*  982*  987*  989*  1001*  1003*  1015*  1017*  1036*
                       1045  1086*  1093  1131*  1132*  1142  1151  1161*  1171  1180  1198*  1207*  1212*
                       1225*  1226*  1246*  1247*  1268*  1289*  1313*  1334*  1356*  1374*  1376*  1408*  1410*
                       1420  1434*  1436*  1447  1457  1477*  1479*  1489  1496  1513*  1515*  1532  1551*
                       1554*  1612*  1613*  1615*  1616*  1618*  1619*  1621*  1622*  1624*  1625*  1627*  1628*
                       1640*  1641*  1671*  1673*  1688*  1689*  1714*  1715*  1730*  1731*  1752*  1753*  1770*
                       1772*  1784  1806*  1807*  1829*  1830*  1851*  1852*  1871*  1872*  1891*  1892*  1926*
                       1928*  1932  1938  1994*  1999  2005  2059*  2062  2068  2112*  2114*  2159*  2195*
                       2197*  2224*  2225*  2254*  2255*  2498*  2504*  2505  3052*  3054*  3288*  3291*  3293*
                       3295  3566*  3568
CSRHB   001766          106#   157   426*  3564*
CTRCHA  045776         3332*  3333*  3334  3343  3347#  3438*  3548  3550*
CTRLCG  045700         2341   2373   2509  3089  3184  3330#  3547
CURENT  050032         1909*  1919*  1922  1943  1952  1953  1970*  1971*  1979*  1992  2010  2019  2020
                       2037*  2038*  3621#
DACADR  050022         3532*  3535*  3539*  3543*  3617#
DACSR   001726           84#  1808*  1809  1831*  1832  1853*  1854  1873*  1874  1893*  1894
DAC0    001730           85#  3479  3505
DAC1    001732           86#  3484  3500
DAC2    001734           87#  3489  3515
DAC3    001736           88#  3494  3510
DASH    040747         2599#  3172  3272
DDISP   = 177570         13#    32   147
DEADKW  004020          364*   372*  378#  1307  1328
DF0     033006           33    34    35    36    37    38    39    40    41    42    43    44    45
                         46    47    48    49    50    51    52    53    54    55    56    57    58
                         59    60    61    62    63    64    65    66  2331#
DH1     032471           65    66  2317#
DH2     032504           33    34    35    36    37    38    39    40    41    42    43    44    45
                         46    47    48    49    50    51    52    53    54    55    56    57    58
                         60    62    64  2318#
DH3     032530           59    63  2319#
DH4     032563           61  2320#
```

```
DIDATA  052611            2886    3718#
DIF     041742            2650#
DIFERR  041720            2641#   2769    3203*   3221*
DIFEX0  046006            2762*   2981*   2988*   3279    3351#
DIFEX1  041710            2635#   2761*   2969*   3253    3278*
DIFLIN  043550             209     249    2277    2969#   2974
DIFLNX  043726            3001#   3003
DIFLN0  043722            2765    2982    2992    3000#
DIFLN1  044132            3041#   3091
DIFT1   045670            3019*   3324#
DIFT2   045672            3026*   3036    3325#
DIFT3   045674            3033*   3037    3326#
DISPLA  001142              32#    147*   2341*   2346*
DISPLY  047762            3450    3456    3462    3607#
DISPRE  000174              22#    147
DIO     052601            2812    2833    3716#
DLIBD   001700              69#   2672    2852    2870
DLICSR  001676              68#   2667    2733*   2810*   2817*   2823*   2860
DLOCSR  001702              70#   2853
DLODB   001704              71#   2698*   2701*   2859*
DOT     047021            3428#
DSWR  = 177570              13#     32     147
DT1     032726              66    2326#
DT2     032734              33      34      35      36      37      38      39      40      41      42      43      44      45
                            46      47      48      49      50      51      52      53      54      55      56      57      58
                            60      62      64    2327#
DT3     032746              59      63    2328#
DT4     032762              61    2329#
DT5     033000              65    2330#
EMTVEC= 000030              13#    147*
EM1     030224              33    2282#
EM10    030577              40    2289#
EM11    030644              41    2290#
EM12    030712              42    2291#
EM13    030745              43    2292#
EM14    031003              44    2293#
EM15    031037              45    2294#
EM16    031071              46    2295#
EM17    031124              47    2296#
EM2     030264              34    2283#
EM20    031157              48    2297#
EM21    031240              49    2298#
EM22    031311              50    2299#
EM23    031364              51    2300#
EM24    031436              52    2301#
EM25    031504              53    2302#
EM26    031537              54    2303#
EM27    031575              55    2304#
EM3     030330              35    2284#
EM30    031644              56    2305#
EM31    031704              57    2306#
EM32    031745              58    2307#
EM33    032011              59    2308#
EM34    032055              60    2309#
EM35    032112              61    2310#
EM36    032177              62    2311#
```

```
EM37    032245              63    2312#
EM4     030376              36    2285#
EM40    032311              64    2313#
EM41    032371              65    2314#
EM42    032431              66    2315#
EM5     030436              37    2286#
EM6     030477              38    2287#
EM7     030533              39    2288#
ENDDMA= 000400             127#    570     573     699     731     764    3058    3336
ERMSG   041007            2602#
ERRVEC= 000004              13#    147*    152*    335    336*    339*    351    352*    360*    365*    373*    375*   2341*
                           2353*
ESCP    052562            2735    2825    3711#
FAIMSG  041665            2629#   3220
FATALO  052342             258    3701#
FATFNG  003204             247     252#    271
FIELDI  041344             283    2613#
FIRST   041746            2652#   3002    3159    3161*
FIXLIN  047740            3538    3542    3546    3598#
FMO     052573            2739    2829    3714#
FSITE   037360             291    2373#   2486
GAIN    040700            2596#   2973    2991    3244
GAINX   040731            2597#   2772
GAIN1   040743            2598#   2749*   2756*   2970*   2983*
GNS   = ****** U            22     149    2366
GTSWR = 104406              149     244    2366#   3345
G1B     040674            2556    2595#
G1LIMO  046074            2747    2979    3380#
G1LIM1  046114            2748    2980    3387#
G2LIMO  046134            2754    2986    3394#
G2LIM1  046154            2755    2987    3401#
HT    = 000011              13#   2342
HT21S   045676            3039*   3044    3327#
INOUTZ  040360             284*    290*   2501    2546#
IOTVEC= 000020              13#    147*
JOY     001762             104#    830     875     880     885     890     895     900     908     913     919     925     931
                           936    2424
JOYG1   040464            2424*   2485    2574#
JOYTW   040656            2483    2593#
JOYTX   040656            2592#
KDPAR0= 172360              14#
KDPAR1= 172362              14#
KDPAR2= 172364              14#
KDPAR3= 172366              14#
KDPAR4= 172370              14#
KDPAR5= 172372              14#
KDPAR6= 172374              14#
KDPAR7= 172376              14#
KDPDR0= 172320              14#
KDPDR1= 172322              14#
KDPDR2= 172324              14#
KDPDR3= 172326              14#
KDPDR4= 172330              14#
KDPDR5= 172332              14#
KDPDR6= 172334              14#
KDPDR7= 172336              14#
```

```
KFO     052564          2737    2827    3712#
KIPAR0= 172340            14#   2353
KIPAR1= 172342            14#    315*
KIPAR2= 172344            14#    317*
KIPAR3= 172346            14#   2090*   2137*   2185*   2243*
KIPAR4= 172350            14#
KIPAR5= 172352            14#
KIPAR6= 172354            14#
KIPAR7= 172356            14#    320*   2353
KIPDR0= 172300            14#
KIPDR1= 172302            14#    316*
KIPDR2= 172304            14#    318*
KIPDR3= 172306            14#    319*
KIPDR4= 172310            14#
KIPDR5= 172312            14#
KIPDR6= 172314            14#
KIPDR7= 172316            14#    321*
KTERR   036410          2353    2354#   2355
KWCSR   001740            92#    173*    174     175     361    1316*   1335*   1337*   1338    3012*   3034*   3047*   3055*
                        3056
KWCSR1  001742            93#    174*    176*   3017*   3018*   3024*   3025*   3031*   3032*
KWPSR   001744            94#    175*    177*    362    1336*   3011*   3019    3026    3033    3046*
LF    = 000012            13#   2342
LIMITS  046010          2746    2753    2978    2985    3354#
LINEA   041122          2607#
LIST    046404           191    3413#
LISTDT  045476          3014    3021    3028    3284#   3312
LISTO   046174           189    3407#
LIST1   047017           193    3427#
LOGIC   003360           185     213     222     298#   2278
LOOPC   041750          2667#   2676    2681    2685    2688    2699    2703
LOOPD   041774          2668    2675#
LOOPE   042036          2678    2687#
LOOPF   042114          2694    2701#
LSBAVG  041700          2631#
LSBMSG  040754          2600#
LSBSVQ  041702          2632#   2943*   2950
LSBSVR  041704          2633#   2942*
LSBSVW  041706          2634#   2926*   2957
MMVEC = 000250            14#   2353*
MORTST= 000000           132#    380    2332    2367
MSGK    052503           334    3708#
MSGMEM  052463           324    3706#
MSG21   041145          2609#
NARA    041732          2646#   3149*   3208    3225
NARB    041734          2647#   3143*   3212    3226
NARCNT  046060          3214    3371#
NARMSG  041036          2604#   3229
NARROW  041730          2105*   2162*   2205*   2263*   2329    2645#
NARTOL  046062          3131    3372#
NLSI11  003356           295#    337*    341*   1568
NOEXIT  043244          2784*   2838*   2866    2875*   2882#
NOMIAL  041714          2637#   3090
NORCNT  046054          3216    3360    3369#
NORMAL  041726          2644#   3153*   3216    3230
NORMSG  041613          2626#   3232
```

```
NORTOL  046056          3129    3370#
NURCNT  046064          3210    3373#
NURTOL  046066          3133    3374#
OBSCNT  046070          3206    3375#
OBSTOL  046072          3376#
OFF     001750            99#    354    503*   504    515*   522    551*   553    796*   809   1032*  1066   1100
                        1128*  1158*  1187*  1223*  1233   1244*  1257   1373*  1407*  1433*  1476*  1512*  1550*
                        1612*  1613*  1615*  1616*  1618*  1619*  1621*  1622*  1624*  1625*  1627*  1628*  1638*
                        1669*  1686*  1712*  1728*  1750*  1769*  1922*  2111*  2155*  2194*  2222*  2252*  3049*
                        3285*
OKMSG   041024          2603#
OUT     041744          2089*  2090    2096   2128*  2129   2136*  2137   2143   2175*  2176   2651#
OUTCHN  032720          2324#  2517*
OUTRNG  032625          2321#  2518
PASMSG  041653          2628#  3218    3246
PBLAST  042756           238   2822#
PERTXT  041560          2623#  3179*  3180*  3181*
PGO     053026          2783   2816    2837   3726#
PIRQ    = 177772          13#
PIRQVE= 000240            13#
POTIME  047024           218   3434#
PRIME   046000          3348#
PRIME0  046002          3349#
PRIME1  046004          2745*  2752*  2977*  2984*  3052   3289   3350#
PRIM0   050764          2731    2976   3667#
PRIM1   051767          2788    3687#
PRIM2   052160          3435    3694#
PRIM3   051540          2822    3681#
PRIM4   052250           303    3697#
PRIM5   051376          2732    3677#
PRIM6   051166           282    3672#
PROROM  042234          2725#   2831
PR0     = 000000          13#
PR1     = 000040          13#
PR2     = 000100          13#
PR3     = 000140          13#
PR4     = 000200          13#
PR5     = 000240          13#
PR6     = 000300          13#
PR7     = 000340          13#
PS      = 177776          13#
PSW     = 177776          13#    358*
PWRVEC= 000024            13#    147*   2351*
RAMVAL  042174          2717#  2795    2809
RBEG0   002646           181    193#    245    254   2700   2774   2792   2842   2869   2996   3518
RDBLST  043064          2734   2736    2738   2778   2780   2782   2785   2811   2813   2815   2824   2826   2828
                        2832   2834    2836   2839   2848#
RDCHR   = 104410        2336   2366#
RDLIN   = 104411         194    285    2338   2366#  2789
RDOCT   = 104412         269   2366#
READ    045140          3187   3203#
READXX  045356          3251#
READ1   044422          3096#
READ7   045354          3218*  3220*  3246#
REDJOY= 000001           131#   819     838    875    880    885    890    895    900    908    913    919    925
                         931    936
```

```
RESREG= 104414           2363    2366#
RESVEC= 000010             13#
ROMPNT  041712           2636#   2750*   2757*   2919
ROMVAL  042134           2636    2709#   2741    2743    2750    2757    2796    2805    3256    3258
RTN     041570           2624#   3165    3260    3261    3262    3266
RTRT    002014             23     140#    265    3342
RTRTA   002044            141     144     147#    279
RUNDIF  050024            184*    208*    212*    221*    248*   2275    2993    3618#
RUNIT   052412            151*    195*    196*    197*    198     203     206     210     216     219     223     226     231
                          236     239     242     246    3154    3170    3703#
RUNITA  052432            264    3704#
SAMCAM  047214           3475#
SAMDAT  047514           3536    3540    3544    3556#
SAVREG= 104413           2363    2366#
SDPAR0= 172260             14#
SDPAR1= 172262             14#
SDPAR2= 172264             14#
SDPAR3= 172266             14#
SDPAR4= 172270             14#
SDPAR5= 172272             14#
SDPAR6= 172274             14#
SDPAR7= 172276             14#
SDPDR0= 172220             14#
SDPDR1= 172222             14#
SDPDR2= 172224             14#
SDPDR3= 172226             14#
SDPDR4= 172230             14#
SDPDR5= 172232             14#
SDPDR6= 172234             14#
SDPDR7= 172236             14#
SFR     001756            102#    357     408*    435*    436     446*    449     456*    459     464*    465*    467     476*
                          489*    502*    514*    518*    537     546*    550*    552*    558*    559*    570*    573*    583*
                          593*    594*    599*    614*    615*    620*    638*    639*    644*    656*    657*    662*    670*
                          674*    681*    682*    688*    699*    704*    712*    713*    719*    731*    736*    737*    746*
                          747*    753*    764*    765*    780*    786*    792*    793*    799*    818*    819*    821     825
                          831     837*    838*    840     844     848*    849     856     858     875*    880*    885*    890*
                          895*    900*    908*    913*    919*    925*    931*    936*    956*    957*    958*    960*    961*
                          962*    964*    965*    966*    968*    969*    970*    972*    973*    974*    976*    977*    978*
                          980*    981*    982*    985*    986*    988*    990*    999*   1000*   1002*   1004*   1013*   1014*
                         1016*   1018*   1030*   1035*   1037*   1041*   1046*   1078*   1085*   1088*   1089*   1103*   1126*
                         1130*   1133*   1137*   1149*   1156*   1160*   1162*   1166*   1178*   1185*   1189*   1199*   1203*
                         1213*   1218*   1224*   1227*   1228*   1239*   1245*   1248*   1252*   1263*   1266*   1272*   1273*
                         1284*   1287*   1293*   1294*   1306*   1311*   1317*   1327*   1332*   1340*   1351*   1354*   1360*
                         1361*   1371*   1375*   1377*   1381*   1406*   1409*   1413*   1414*   1425*   1430*   1435*   1437*
                         1443*   1456*   1473*   1478*   1480*   1485*   1495*   1509*   1514*   1516*   1525*   1537*   1545*
                         1552*   1555*   1559*   1587*   1612*   1613*   1615*   1616*   1618*   1619*   1621*   1622*   1624*
                         1625*   1627*   1628*   1637*   1639*   1642*   1645*   1668*   1670*   1672*   1674*   1676*   1685*
                         1687*   1690*   1691*   1694*   1711*   1713*   1716*   1718*   1727*   1729*   1732*   1734*   1747*
                         1751*   1754*   1756*   1766*   1771*   1773*   1779*   1801*   1802*   1824*   1825*   1847*   1848*
                         1867*   1868*   1887*   1888*   1921*   1925*   1927*   1990*   1993*   1995*   2055*   2058*   2060*
                         2110*   2113*   2115*   2117*   2154*   2158*   2160*   2163*   2193*   2196*   2198*   2201*   2219*
                         2223*   2226*   2228*   2251*   2253*   2256*   2259*   2421*   2422    2497*   2501*   3048*   3053*
                         3058*   3284*   3292*   3336*   3340*   3565*   3571*
SFRHB   001770            107#    448*
SGNVAL  041550           2621#   3182
SGNVL1  041556           2622#   3103*   3105*   3135    3141    3147
```

```
SHUF0   043346          2891    2911#
SHUF3   043340          2893    2908#
SHUF6   043326          2895    2904#
SIPAR0= 172240            14#
SIPAR1= 172242            14#
SIPAR2= 172244            14#
SIPAR3= 172246            14#
SIPAR4= 172250            14#
SIPAR5= 172252            14#
SIPAR6= 172254            14#
SIPAR7= 172256            14#
SIPDR0= 172200            14#
SIPDR1= 172202            14#
SIPDR2= 172204            14#
SIPDR3= 172206            14#
SIPDR4= 172210            14#
SIPDR5= 172212            14#
SIPDR6= 172214            14#
SIPDR7= 172216            14#
SKPMSG  040763          2601#
SLASH   041143          2608#
SR0   = 177572            14#   2093*  2095*  2118*  2120*  2140*  2142*  2165*  2167*  2188*  2190*  2207*  2209*
                        2246*  2248*  2265*  2267*  2353*
SR1   = 177574            14#
SR2   = 177576            14#
SR3   = 172516            14#   2353*
STACK = 001100            13#    147
STATE   041102          2606#   3166
STDAT0  045632          3009    3041    3311#
STKLMT= 177774            13#
SWR     001140            32#    147*   149    261    1965   2032   2126   2173   2336*  2341   2346   2351*  2375
                        2380    2386   2391   2397   2402   2408   2413   2419   2488   2490   2512   2519   3251
SWREG   000176            22#    147    149    2336
SW0   = 000001            13#
SW00  = 000001            13#   2380
SW01  = 000002            13#   2391
SW02  = 000004            13#
SW03  = 000010            13#
SW04  = 000020            13#
SW05  = 000040            13#
SW06  = 000100            13#   2375
SW07  = 000200            13#
SW08  = 000400            13#
SW09  = 001000            13#
SW1   = 000002            13#
SW10  = 002000            13#
SW11  = 004000            13#   1965   2032   2126   2173   3251
SW12  = 010000            13#
SW13  = 020000            13#   2490   2512
SW14  = 040000            13#
SW15  = 100000            13#   2519
SW2   = 000004            13#   2402
SW3   = 000010            13#   2413
SW4   = 000020            13#   2419
SW5   = 000040            13#
SW6   = 000100            13#
```

```
SW7    = 000200              13#   2386
SW8    = 000400              13#   2397
SW9    = 001000              13#   2408
TBITVE= 000014               13#
TESTER  002022               24    142#
TESTZ  = 000002             130#   599    620    644    662    674    688    719    732    753    765    799    956
                            957    958    960    961    962    964    965    966    968    969    970    972    973
                            974    976    977    978    980    981    982    988    990   1002   1004   1016   1018
                           1037   1088   1133   1162   1199   1227   1248   1377   1413   1437   1480   1516   1555
                           1612   1613   1615   1616   1618   1619   1621   1622   1624   1625   1627   1628   1642
                           1672   1690   1716   1732   1754   1773   1927   1995   2060   2115   2160   2198   2226
                           2256   2546   3053   3292   3465
TFSITE  003300               200    282#
TIMEO   041223              2611#  2868
TKVEC  = 000060              13#
TPVEC  = 000064              13#
TRAPVE= 000034               13#   147*
TRTVEC= 000014               13#
TSTCON= 000004              129#   793    875    880    885    890    895    900    908    913    919    925    931
                            936    956    957    958    960    961    962    964    965    966    968    969    970
                            972    973    974    976    977    978    980    981    982    986   1000   1014   1035
                           1085   1130   1160   1189   1224   1245   1266   1287   1311   1332   1354   1375   1409
                           1435   1478   1514   1552   1612   1613   1615   1616   1618   1619   1621   1622   1624
                           1625   1627   1628   1639   1670   1687   1713   1729   1751   1771   1925   1993   2058
                           2113   2158   2196   2223   2253
TSTDMA= 000010              128#   559    594    615    639    657    682    713    747    793    875    880    885
                            890    895    900    908    913    919    925    931    936    956    957    958    960
                            961    962    964    965    966    968    969    970    972    973    974    976    977
                            978    980    981    982    986   1000   1014   1035   1085   1130   1160   1189   1224
                           1245   1266   1287   1311   1332   1354   1375   1409   1435   1478   1514   1552   1612
                           1613   1615   1616   1618   1619   1621   1622   1624   1625   1627   1628   1639   1670
                           1687   1713   1729   1751   1771   1802   1825   1848   1868   1888   2113   2158   2196
                           2223   2253
TST1    003640               300    340    350#
TST10   004512               451    455#
TST100  014644              1051   1074   1077#
TST101  015170              1098   1125#
TST102  015400              1145   1153   1155#
TST103  015602              1174   1182   1184#
TST104  016002              1217#
TST105  016150              1220   1235   1238#
TST106  016326              1241   1259   1262#
TST107  016460              1280   1283#
TST11   004550               460    463#
TST110  016612              1301   1305#
TST111  016746              1308   1323   1326#
TST112  017116              1329   1346   1350#
TST113  017250              1368   1370#
TST114  017406              1388   1393#
TST115  017614              1429#
TST116  020054              1467   1472#
TST117  020306              1505   1508#
TST12   004612               468    472#
TST120  020524              1541#
TST121  021000              1612#
TST122  021130              1612   1613#
```

```
TST123  021260          1613    1615#
TST124  021410          1615    1616#
TST125  021540          1616    1618#
TST126  021670          1618    1619#
TST127  022020          1619    1621#
TST13   004700           485#
TST130  022150          1621    1622#
TST131  022300          1622    1624#
TST132  022430          1624    1625#
TST133  022560          1625    1627#
TST134  022710          1627    1628#
TST135  023040          1628    1633#
TST136  023226          1653    1658    1664#
TST137  023356          1681    1684#
TST14   004766           498#
TST140  023544          1701    1705    1710#
TST141  023674          1723    1726#
TST142  024024          1739    1743#
TST143  024164          1749    1761    1765#
TST144  024370          1792    1795#
TST145  024536          1797    1815    1818#
TST146  024704          1820    1838    1842#
TST147  025030          1844    1858    1862#
TST15   005054           512#
TST150  025154          1864    1878    1882#
TST151  025300          1884    1898    1902#
TST152  025670          1941    1951    1964    1966    1969    1974#
TST153  026250          2008    2018    2025    2031    2033    2036    2042#
TST154  026472          2071    2079    2084#
TST155  026766          2086    2125    2127    2132#
TST156  027264          2134    2172    2174    2179#
TST157  027500          2184    2211    2214#
TST16   005314           548#
TST160  027640          2221    2233    2237#
TST161  030060          2242    2270    2274#
TST17   005364           554     557#
TST2    004022           366     374     381#
TST20   005556           586     592#
TST21   005730           604     611     613#
TST22   006114           626     634     637#
TST23   006254           649     653     655#
TST24   006370           666     669#
TST25   006462           677     680#
TST26   006654           701     708     711#
TST27   007054           734     741     745#
TST3    004124           397#
TST30   007324           791#
TST31   007466           806     810     817#
TST32   007574           833     836#
TST33   007740           859     874#
TST34   010034           876     879#
TST35   010130           881     884#
TST36   010224           886     889#
TST37   010320           891     894#
TST4    004172           403     406#
TST40   010414           896     899#
```

```
TST41   010510            901     907#
TST42   010604            909     912#
TST43   010700            914     918#
TST44   010774            920     924#
TST45   011070            926     930#
TST46   011164            932     935#
TST47   011260            937     956#
TST5    004244            412     416#
TST50   011362            956     957#
TST51   011464            957     958#
TST52   011566            958     960#
TST53   011670            960     961#
TST54   011772            961     962#
TST55   012074            962     964#
TST56   012176            964     965#
TST57   012300            965     966#
TST6    004354            429     431#
TST60   012402            966     968#
TST61   012504            968     969#
TST62   012606            969     970#
TST63   012710            970     972#
TST64   013012            972     973#
TST65   013114            973     974#
TST66   013216            974     976#
TST67   013320            976     977#
TST7    004442            445#
TST70   013422            977     978#
TST71   013524            978     980#
TST72   013626            980     981#
TST73   013730            981     982#
TST74   014032            982     984#
TST75   014134            994     998#
TST76   014236           1008    1012#
TST77   014340           1022    1029#
TYPDS = 104405            333    2278    2366#   3164    3174    3223    3228    3231    3236    3239
TYPE  = 104401            149     188     190     193     252     258     264     268     282     283     503     324     334
                         2278    2336    2340    2342    2344    2346    2348    2351    2366#   2431    2437    2444    2450
                         2457    2463    2470    2476    2483    2518    2556    2731    2732    2771    2772    2773    2788
                         2791    2822    2841    2868    2973    2976    2991    3162    3165    3166    3172    3182    3224
                         3229    3232    3237    3240    3244    3245    3260    3261    3262    3266    3272    3435
TYPOC = 104402            260    2336    2348    2366#
TYPON = 104404           2366#
TYPOS = 104403           2366#   2554    2558    3168    3270
UDPAR0= 177660             14#
UDPAR1= 177662             14#
UDPAR2= 177664             14#
UDPAR3= 177666             14#
UDPAR4= 177670             14#
UDPAR5= 177672             14#
UDPAR6= 177674             14#
UDPAR7= 177676             14#
UDPDR0= 177620             14#
UDPDR1= 177622             14#
UDPDR2= 177624             14#
UDPDR3= 177626             14#
UDPDR4= 177630             14#
```

```
UDPDR5= 177632            14#
UDPDR6= 177634            14#
UDPDR7= 177636            14#
UIPAR0= 177640            14#
UIPAR1= 177642            14#
UIPAR2= 177644            14#
UIPAR3= 177646            14#
UIPAR4= 177650            14#
UIPAR5= 177652            14#
UIPAR6= 177654            14#
UIPAR7= 177656            14#
UIPDR0= 177600            14#
UIPDR1= 177602            14#
UIPDR2= 177604            14#
UIPDR3= 177606            14#
UIPDR4= 177610            14#
UIPDR5= 177612            14#
UIPDR6= 177614            14#
UIPDR7= 177616            14#
UNFIX   041451          2616#   2771
VAL0    050036          3537*   3623#
VAL1    050040          3541*   3624#
VAL2    050042          3545*   3625#
VECTA0  001772           111#    773*    787*   1193*   1214*
VECTA1  001774           112#    787     788*   1214    1215*
VECTB0  001776           113#   1524*   1538*   1546*   1571*   1588*
VECTB1  002000           114#    171    1538    1539*   1547*   1588    1589*
VTCHAR  001706            74#   3436*   3523    3526    3528*
VTCSR   001716            78#   3441    3443*   3444    3446*   3469*   3472*   3607*   3609    3612*
VTCXY   001714            77#   3525*
VTINT   001724            81#   3470*   3471*
VTMAP   001720            79#   3608*
VTPX    001722            80#   3611*
VTXPOS  001712            76#   3603*
VTYPOS  001710            75#   3437*
WARNO   052522           268    3709#
WCR     001752           100#    355     477*    478     516*    527     595*    605     616*    628     640*    646     658*
                         664     671*    683*    714*    748*    794*   1033*   1059    1083*   1110    1127*   1157*   1186*
                        1221*   1242*   1264*   1285*   1309*   1330*   1352*   1431*   1475*   1511*   1549*   1767*   1803*
                        1826*   1827*   1834    1849*   1869*   1889*   1923*   1991*   2056*   2156*   2500*   3050*   3286*
                        3562*
WFMODE  050026           142*    146*    186     201     214     229     234     301    1796    1819    1843    1863    1883
                        3355    3619#
WIDA    041736          2648#   3151*   3209    3233
WIDB    041740          2649#   3145*   3213    3234
WIDMSG  041061          2605#   3237
XTTY    003242           241     268#
ZAPRAM  042622          2763    2795#
ZPO     053036          2786    2840    3728#
$APTHD  001000            31#
$ASTAT= ****** U        2350
$ATYC   035466          2350#
$ATY1   035442          2350#
$ATY3   035450          2342    2350#
$ATY4   035460          2346    2350#
$AUTOB  001134            32#    149*    182     322    2336    2971    2989    3156
```

```
$BASE   001250          32#   154   159   161  2326  2327  2328  2329
$BDADR  001122          32#  1072* 1116* 1117  1121* 1122  1612* 1613* 1615* 1616* 1618* 1619* 1621*
                      1622* 1624* 1625* 1627* 1628* 1649* 1651  1677* 1679  1719* 1721  1735* 1737
                      1757* 1759  1945* 1952* 2012* 2019* 2073* 2106* 2161* 2229* 2231  2328  2329
$BDDAT  001126          32#  386*  389   401*  402   410*  411   420*  421   427*  428   436*  438
                       449*  450   459*  467*  478*  479   491*  492   504*  505   522*  523   527*
                       528   532*  533   537*  538   542*  543   553*  567*  568   575*  576   584*
                       585   601*  605*  609*  610   622*  623   628*  632*  633   646*  651*  652
                       664*  665   676*  706*  707   739*  740   779*  800*  802   809*  825*  831*
                       832   844*  849*  851   856*  858*  875*  880*  885*  890*  895*  900*  908*
                       913*  919*  925*  931*  936*  956*  957*  958*  960*  961*  962*  964*  965*
                       966*  968*  969*  970*  972*  973*  974*  976*  977*  978*  980*  981*  982*
                       991*  993   1005* 1007  1019* 1021  1045* 1048  1052* 1054  1059* 1061  1066*
                      1070* 1073  1093* 1095  1100* 1104* 1106  1110* 1112  1117* 1118  1142* 1146
                      1151* 1152  1171* 1175  1180* 1181  1233* 1234  1257* 1258  1277* 1279  1298*
                      1300  1320* 1322  1343* 1345  1365* 1367  1385* 1387  1416* 1417  1420* 1421
                      1447* 1449  1457* 1459  1464* 1466  1489* 1491  1496* 1498  1502* 1504  1532*
                      1534  1612* 1613* 1615* 1616* 1618* 1619* 1621* 1622* 1624* 1625* 1627* 1628*
                      1651* 1652  1659* 1679* 1680  1699* 1700  1706* 1721* 1722  1737* 1738  1759*
                      1760  1784* 1785  1789* 1791  1811* 1812  1834* 1835  1856* 1857  1876* 1877
                      1896* 1897  1938* 1946* 1948  1953* 1955  2005* 2013* 2015  2020* 2022  2068*
                      2074* 2076  2119* 2121  2166* 2168  2208* 2210  2231* 2232  2266* 2268* 2269
                      2327  2328  2329
$BELL   001164          32#  2346
$CDW1   001254          32#   173  2330
$CHARC  034654        2342#*
$CKSWR  033010        2336#  2366
$CMTAG  001100          32#   147
$CM3  = 000000          32#
$CNTLG  033535        2336#
$CNTLU  033530        2336#
$CORE   036310        2353#
$CPUUP  001222          32#
$CRLF   001171          32#  2336  2342  2346  2348
$CROUT  036340        2353#
$DBLK   034102        2340#
$DB2D   037042        2363#  2364
$DECVL  037222        2363#
$DEVCT  001204          32#
$DEVM   001252          32#
$DIV    036622        2361#  2936  3114
$DOAGN  030200        2278#
$DTBL   034072        2340#
$ENDAD  030170          29    149  2278#  2346
$ENDCT  030136         147  2278#
$ENDMG  030207        2278#
$ENULL  030204        2278#
$ENV    001214          32#   149  2342  2346  2350
$ENVM   001215          32#   147  2342  2350
$EOP    030102         376  2276  2278#  2995
$EOPCT  030130        147*  2278#
$ERFLG  001103          32#  2341* 2346*
$ERMAX  001115          32#   147* 2341*
$ERROR  035106         147  2346#
$ERRPC  001116          32#  2326  2327  2328  2329  2330  2346*  2348
$ERRTB  001256          32#  2348
```

```
$ERRTY  035306          2346    2348#
$ERTTL  001112            32#   2346*
$ESCAP  001162            32#    147*   2341*   2346
$ETABL  001214            32#
$ETEND  001256            31      32#
$FATAL  001176            32#   2350*
$FFLG   035706          2350#*
$FILLC  001156            32#   2342
$FILLS  001155            32#   2342
$GDADR  001120            32#
$GDDAT  001124            32#    387*   388*   389    400*   402    409*   411    418*   421    425*   428    437*
                          438    447*   450    458*   466*   473*   477    479    483*   486*   490    492    496*
                          499*   503    505    509*   521*   523    526*   528    531*   533    536*   538    541*
                          543    549*   565*   568    574*   576    581*   585    600*   608*   621*   627*   631*
                          645*   650*   663*   675*   691*   705*   707    723*   738*   740    756*   801*   802
                          808*   826*   834*   845*   850*   851    857*   875*   880*   885*   890*   895*   900*
                          908*   913*   919*   925*   931*   936*   956*   957*   958*   960*   961*   962*   964*
                          965*   966*   968*   969*   970*   972*   973*   974*   976*   977*   978*   980*   981*
                          982*   992*   993    1006*  1007   1020*  1021   1047*  1048   1053*  1054   1060*  1061
                         1065*  1071*  1073   1094*  1095   1099*  1105*  1106   1111*  1112   1115*  1118   1141*
                         1146   1150*  1152   1170*  1175   1179*  1181   1232*  1234   1256*  1258   1278*  1279
                         1299*  1300   1321*  1322   1344*  1345   1366*  1367   1386*  1387   1415*  1417   1423*
                         1448*  1449   1458*  1459   1465*  1466   1490*  1491   1497*  1498   1503*  1504   1533*
                         1534   1612*  1613*  1615*  1616*  1618*  1619*  1621*  1622*  1624*  1625*  1627*  1628*
                         1650*  1652   1660*  1678*  1680   1698*  1700   1707*  1720*  1722   1736*  1738   1758*
                         1760   1783*  1785   1790*  1791   1798*  1804   1812   1816*  1821*  1826   1835   1839*
                         1845*  1857   1865*  1877   1885*  1897   1939*  1947*  1948   1954*  1955   2006*  2014*
                         2015   2021*  2022   2069*  2075*  2076   2116*  2121   2164*  2168   2206*  2210   2230*
                         2232   2264*  2269   2327   2328   2329
$GET42  030160          2278#
$GTSWR  033060          2336#   2366
$HD   = 000000            12
$HIBIS  001000            31#
$HIOCT  033664          2338#*
$ICNT   001104            32#   2341*
$ILLUP  036050          2351#
$INTAG  001135            32#   2336
$ITEMB  001114            32#   2346*   2348
$KTNEX  036302          2353#
$KTOUT  036272          2353#
$KT11   036124           304*    311*    313   2085   2133   2353#*
$LF     001172            32#   2336    2342   2346
$LFLG   035705          2350#*
$LPADR  001106            32#    147*   2341*
$LPERR  001110            32#    147*    382*   432*   474*   487*   500*  1403*  1799*  1822*  1911*  1981*  2341*
                         2346
$LSTAD  036404           306*    309*    310   2353#*
$LSTBK  036406           325    1219    1240   1748   2129   2176   2183   2220   2241   2353#*
$MADR1  001226            32#
$MADR2  001232            32#
$MADR3  001236            32#
$MADR4  001242            32#
$MAIL   001174            31      32#    147    149   2341   2342   2346
$MAMS1  001224            32#
$MAMS2  001230            32#
$MAMS3  001234            32#
```

```
$MAMS4  001240            32#
$MBADR  001002            31#
$MFLG   035704          2350#*
$MNEW   033553          2336#
$MSGAD  001210            32#   2350*
$MSGLG  001212            32#   2350*
$MSGTY  001174            32#   2350*
$MSWR   033542          2336#
$MTYP1  001225            32#
$MTYP2  001231            32#
$MTYP3  001235            32#
$MTYP4  001241            32#
$MULT   036510          2359#   2929    3108
$MXCNT  034374          2341#
$NULL   001154            32#   2342
$NWTST= 000001           350#    381#    397#    406#    416#    431#    445#    455#    463#    472#    485#    498#    512#
                         548#    557#    592#    613#    637#    655#    669#    680#    711#    745#    791#    817#    836#
                         874#    879#    884#    889#    894#    899#    907#    912#    918#    924#    930#    935#    956#
                         957#    958#    960#    961#    962#    964#    965#    966#    968#    969#    970#    972#    973#
                         974#    976#    977#    978#    980#    981#    982#    984#    998#   1012#   1029#   1077#   1125#
                        1155#   1184#   1217#   1238#   1262#   1283#   1305#   1326#   1350#   1370#   1393#   1429#   1472#
                        1508#   1541#   1612#   1613#   1615#   1616#   1618#   1619#   1621#   1622#   1624#   1625#   1627#
                        1628#   1633#   1664#   1684#   1710#   1726#   1743#   1765#   1795#   1818#   1842#   1862#   1882#
                        1902#   1974#   2042#   2084#   2132#   2179#   2214#   2237#   2274#
$OCNT   035102          2344#*
$OMODE  035104          2344#*
$OVER   034360          2341#
$PASS   001202            32#    147*    299    1963    2030    2124    2171    2278*   2341
$PASTM  001006            31#
$POWER  036056          2351#
$PWRDN  035710           147    2351#
$PWRMG  036044          2351#
$PWRUP  035762          2351#
$QUES   001170            32#    253    2336    2342    2346
$RDCHR  033272          2336#   2366
$RDDEC= ****** U        2366
$RDLIN  033412          2336#   2366
$RDOCT  033564          2338#   2366
$RDSZ = 000010          2336#
$RESRE  036452          2357#   2366
$RTNAD  030202          2278#
$R2A  = ****** U        2366
$SAVRE  036414          2357#   2366
$SAVR6  036054          2351#*
$SB2D   037236          2364#   3176
$SCOPE  034112           147    2341#
$SETUP= 000137           139#    147     149    2278    2336    2341    2346
$SIZE   036066           305     312    2353#
$SIZEX  036344          2353#
$STUP = 177777           139#
$SVLAD  034324          2341#
$SVPC = 000106            29#
$SWR  = 167400             5#     12      21      32     147     350     381     397     406     416     431     445     455
                         463     472     485     498     512     548     557     592     613     637     655     669     680
                         711     745     791     817     836     874     879     884     889     894     899     907     912
                         918     924     930     935     956     957     958     960     961     962     964     965     966
```

```
                        968    969    970    972    973    974    976    977    978    980    981    982    984
                        998   1012   1029   1077   1125   1155   1184   1217   1238   1262   1283   1305   1326
                       1350   1370   1393   1429   1472   1508   1541   1612   1613   1615   1616   1618   1619
                       1621   1622   1624   1625   1627   1628   1633   1664   1684   1710   1726   1743   1765
                       1795   1818   1842   1862   1882   1902   1974   2042   2084   2132   2179   2214   2237
                       2274   2278   2341   2346   2351
$SWREG  001216          32#    147
$SWRMK= 000000          21    2341
$TEMP   002004         120#    335*    339    383*    385    387*    393*    394    433*    435    437    441*    442
                      1087*   1090*  1404*   1411   1415   1426*
$TEMP1  050016        3062*   3066*  3068    3572*  3578*  3579    3584*   3585   3615#
$TEMP2  050020        3573*   3579*  3580    3585*  3586   3616#
$TESTN  001200          32#   2341*
$TIMES  001160          32#    147*    350*    381*    397*    406*    416*    445*    455*    463*    472*    485*    498*
                       512*    548*    592*    613*    637*    655*    669*    680*    711*    745*    791*    817*    836*
                       874*    879*    884*    889*    894*    899*    907*    912*    918*    924*    930*    935*    956*
                       957*    958*    960*    961*    962*    964*    965*    966*    968*    969*    970*    972*    973*
                       974*    976*    977*    978*    980*    981*    982*    984*    998*   1012*   1029*   1077*   1125*
                      1155*   1184*   1217*   1238*   1262*   1283*   1305*   1326*   1350*   1370*   1393*   1429*   1472*
                      1508*   1541*   1612*   1613*   1615*   1616*   1618*   1619*   1621*   1622*   1624*   1625*   1627*
                      1628*   1633*   1664*   1684*   1710*   1726*   1743*   1765*   1795*   1818*   1842*   1862*   1882*
                      1902*   1974*   2042*   2084*   2132*   2179*   2214*   2237*   2274*   2278*   2341*
$TKB    001146          32#   2336    2689    3332
$TKS    001144          32#    274     277   2336*   2687    3330
$TN   = 000162           6#     12     300     340    350#    366     374    381#    397#    403    406#    412    416#
                       429    431#    445#    451    455#    460    463#    468    472#    485#    498#    512#    548#
                       554    557#    586    592#    604    611    613#    626    634    637#    649    653    655#
                       666    669#    677    680#    701    708    711#    734    741    745#    791#    806    810
                       817#    833    836#    859    874#    876    879#    881    884#    886    889#    891    894#
                       896    899#    901    907#    909    912#    914    918#    920    924#    926    930#    932
                       935#    937    956#    957#    958#    960#    961#    962#    964#    965#    966#    968#    969#
                       970#    972#    973#    974#    976#    977#    978#    980#    981#    982#    984#    994    998#
                      1008   1012#   1022   1029#   1051   1074   1077#   1098   1125#   1145   1153   1155#   1174
                      1182   1184#   1217#   1220   1235   1238#   1241   1259   1262#   1280   1283#   1301   1305#
                      1308   1323   1326#   1329   1346   1350#   1368   1370#   1388   1393#   1429#   1467   1472#
                      1505   1508#   1541#   1612#   1613#   1615#   1616#   1618#   1619#   1621#   1622#   1624#   1625#
                      1627#   1628#   1633#   1653   1658   1664#   1681   1684#   1701   1705   1710#   1723   1726#
                      1739   1743#   1749   1761   1765#   1792   1795#   1797   1815   1818#   1820   1838   1842#
                      1844   1858   1862#   1864   1878   1882#   1884   1898   1902#   1941   1951   1964   1966
                      1969   1974#   2008   2018   2025   2031   2033   2036   2042#   2071   2079   2084#   2086
                      2125   2127   2132#   2134   2172   2174   2179#   2184   2211   2214#   2221   2233   2237#
                      2242   2270   2274#
$TNPWR  037152        2363#
$TPB    001152          32#   2342*   2679*   2702*
$TPFLG  001157          32#   2342
$TPS    001150          32#   2342    2675
$TRAP   037272         147    2366#
$TRAP2  037314        2366#
$TRP  = 000015        2366#
$TRPAD  037326        2366#
$TSTM   001004          31#
$TSTNM  001102          32#   2278*   2341*   2346
$TTYIN  033520        2336#
$TYPBN= ****** U      2366
$TYPDS  033666        2340#   2366
$TYPE   034376        2342#   2350    2366
```

```
$TYPEC  034610            2336    2342#
$TYPEX  034656            2342#
$TYPOC  034704            2344#   2366
$TYPON  034720            2344#   2366
$TYPOS  034660            2344#   2366
$UNIT   001206              32#
$UNITM  001010              31#
$USWR   001220              32#
$VECT1  001244              32#     163
$VECT2  001246              32#
$XTSTR  034130            2341#
$$GET4= 000000            2278#
$OFILL  035103            2344#*
$40CAT= ****** U          2341    2346
.       = 064002            22#     26#     29#     30#     31#     32#     147     2278    2325#   2336#   2340#   2341    2342
                          2346    2348#   2350#   2351    2363#   2630#   3430#   3700#   3734#   3735#   3736#
.$ASTA= ****** U          2350
.$X     = 001000            31#
```

```
COMMEN    13#
ENDCOM    13#
ERROR     13#   368   371   391   404   413   423   430   440   452   461   469   481   494   507
           525   530   535   540   545   555   571   578   587   603   607   612   625   630   635
           648   654   667   678   700   709   733   742   766   781   804   811   827   835   846
           853   860   877   882   887   892   897   902   910   915   921   927   933   938   956
           957   958   960   961   962   964   965   966   968   969   970   972   973   974   976
           977   978   980   981   982   995  1009  1023  1050  1056  1063  1068  1075  1097  1102
          1108  1114  1120  1144  1148  1154  1173  1177  1183  1208  1236  1260  1281  1302  1324
          1347  1369  1389  1419  1424  1451  1461  1468  1493  1500  1506  1529  1536  1577  1582
          1612  1613  1615  1616  1618  1619  1621  1622  1624  1625  1627  1628  1656  1661  1682
          1704  1708  1724  1740  1762  1787  1793  1814  1837  1859  1879  1899  1940  1950  1957
          2007  2017  2024  2070  2078  2123  2170  2212  2234  2271
ESCAPE    13#
GETPRI    13#  2353  2361
GETSWR    13#   149#
JOYMAC   862#   875   880   885   890   895   900   908   913   919   925   931   936
MULT      13#
NEWTST    13#   350   381   397   406   416   431   445   455   463   472   485   498   512   548
           557   592   613   637   655   669   680   711   745   791   817   836   874   879   884
           889   894   899   907   912   918   924   930   935   956   957   958   960   961   962
           964   965   966   968   969   970   972   973   974   976   977   978   980   981   982
           984   998  1012  1029  1077  1125  1155  1184  1217  1238  1262  1283  1305  1326  1350
          1370  1393  1429  1472  1508  1541  1612  1613  1615  1616  1618  1619  1621  1622  1624
          1625  1627  1628  1633  1664  1684  1710  1726  1743  1765  1795  1818  1842  1862  1882
          1902  1974  2042  2084  2132  2179  2214  2237  2274
OFFBIT   1591#  1612  1613  1615  1616  1618  1619  1621  1622  1624  1625  1627  1628
POP       13#  2338  2340  2350  2351  2357  2359  2361
PUSH      13#  2338  2340  2350  2351  2357  2359  2361
REPORT    13#
RESLM    941#   956   957   958   960   961   962   964   965   966   968   969   970   972   973
           974   976   977   978   980   981   982
SCOPE     13#   350   381   397   406   416   431   445   455   463   472   485   498   512   548
           557   592   613   637   655   669   680   711   745   791   817   836   874   879   884
           889   894   899   907   912   918   924   930   935   956   957   958   960   961   962
           964   965   966   968   969   970   972   973   974   976   977   978   980   981   982
           984   998  1012  1029  1077  1125  1155  1184  1217  1238  1262  1283  1305  1326  1350
          1370  1393  1429  1472  1508  1541  1612  1613  1615  1616  1618  1619  1621  1622  1624
          1625  1627  1628  1633  1664  1684  1710  1726  1743  1765  1795  1818  1842  1862  1882
          1902  1974  2042  2084  2132  2179  2214  2237  2274  2278
SETPRI    13#
SETTRA   2366#
SETUP     13#   147
SETZ     135#   599   620   644   662   674   688   719   753   799  1037  1088  1133  1162  1199
          1227  1248  1377  1413  1437  1480  1516  1555  1612  1613  1615  1616  1618  1619  1621
          1622  1624  1625  1627  1628  1642  1690  1716  1732  1754  1773  1995  2115  2160  2198
          2226  2256
SKIP      13#   300   323   340   366   374   390   403   412   422   429   439   451   460   468
           480   493   506   524   529   534   539   544   554   569   577   586   602   604   606
           611   626   629   634   649   653   666   677   701   708   734   741   767   782   803
           806   810   833   852   859   876   881   886   891   896   901   909   914   920   926
           932   937   956   957   958   960   961   962   964   965   966   968   969   970   972
           973   974   976   977   978   980   981   982   994  1008  1022  1049  1051  1055  1062
          1067  1074  1096  1098  1101  1107  1113  1119  1123  1145  1147  1153  1174  1176  1182
          1209  1220  1235  1241  1259  1280  1301  1308  1323  1329  1346  1368  1388  1418  1450
          1460  1467  1492  1499  1505  1530  1535  1578  1583  1612  1613  1615  1616  1618  1619
```

```
                1621    1622    1624    1625    1627    1628    1653    1658    1681    1701    1703    1705    1723    1739    1749
                1761    1786    1792    1797    1813    1815    1820    1836    1838    1844    1858    1864    1878    1884    1898
                1933    1941    1944    1949    1951    1956    1964    1966    1969    2000    2008    2011    2016    2018    2023
                2025    2031    2033    2036    2063    2071    2077    2079    2086    2122    2125    2127    2134    2169    2172
                2174    2184    2211    2221    2233    2242    2270    2276
SLASH             13#
SOFTSW             9#
SPACE             13#
STARS             13#      29      31      32     350     381     397     406     416     431     445     455     463     472     485
                 498     512     548     557     592     613     637     655     669     680     711     745     791     817     836
                 874     879     884     889     894     899     907     912     918     924     930     935     956     957     958
                 960     961     962     964     965     966     968     969     970     972     973     974     976     977     978
                 980     981     982     984     998    1012    1029    1077    1125    1155    1184    1217    1238    1262    1283
                1305    1326    1350    1370    1393    1429    1472    1508    1541    1612    1613    1615    1616    1618    1619
                1621    1622    1624    1625    1627    1628    1633    1664    1684    1710    1726    1743    1765    1795    1818
                1842    1862    1882    1902    1974    2042    2084    2132    2179    2214    2237    2274    2278    2336    2338
                2340    2341    2342    2344    2346    2348    2350    2351    2353    2357    2359    2361    2363    2364    2366
SWRSU             13#     147#
TRMTRP          2366#
TYPBIN            13#
TYPDEC            13#    2278
TYPNAM            13#     149
TYPNUM            13#
TYPOCS            13#
TYPOCT            13#    2336    2348
TYPTXT            13#
$$CMRE            32#
$$CMTM            32#
$$ESCA            13#
$$NEWT            13#     350     381     397     406     416     431     445     455     463     472     485     498     512     548
                 557     592     613     637     655     669     680     711     745     791     817     836     874     879     884
                 889     894     899     907     912     918     924     930     935     956     957     958     960     961     962
                 964     965     966     968     969     970     972     973     974     976     977     978     980     981     982
                 984     998    1012    1029    1077    1125    1155    1184    1217    1238    1262    1283    1305    1326    1350
                1370    1393    1429    1472    1508    1541    1612    1613    1615    1616    1618    1619    1621    1622    1624
                1625    1627    1628    1633    1664    1684    1710    1726    1743    1765    1795    1818    1842    1862    1882
                1902    1974    2042    2084    2132    2179    2214    2237    2274
$$SET           2366#
$$SETM           147#
$$SKIP            13#     300     340     366     374     403     412     429     451     460     468     554     586     604     611
                 626     634     649     653     666     677     701     708     734     741     806     810     833     859     876
                 881     886     891     896     901     909     914     920     926     932     937     956     957     959     960
                 961     962     964     965     966     968     969     970     972     973     974     976     977     978     980
                 981     982     994    1008    1022    1051    1074    1098    1145    1153    1174    1182    1220    1235    1241
                1259    1280    1301    1308    1323    1329    1346    1368    1388    1467    1505    1612    1613    1615    1616
                1618    1619    1621    1622    1624    1625    1627    1628    1653    1658    1681    1701    1705    1723    1739
                1749    1761    1792    1797    1815    1820    1838    1844    1858    1864    1878    1884    1898    1941    1951
                1964    1966    1969    2008    2018    2025    2031    2033    2036    2071    2079    2086    2125    2127    2134
                2172    2174    2184    2211    2221    2233    2242    2270
.EQUAT             7#     13
.HEADE             7#     12
.KT11             10#     14
.SETUP             8#    139
.SWRHI             9#     21
.SWRLO            21#
.$ACT1            11#     29
```

```
.$APTB     11#     32#
.$APTH     11#     31
.$APTY     11#   2350
.$CATC      7#     22
.$CKSW      9#
.$CMTA      7#     32
.$DB2D     10#   2363
.$DIV      10#   2361
.$EOP       7#   2273
.$ERRO      7#   2346
.$ERRT      9#   2348
.$MULT     10#   2359
.$PARM      8#
.$POWE      8#   2351
.$RDOC     10#   2338
.$READ      8#   2336
.$SAVE      8#     10#   2357
.$SB2D     10#   2364
.$SCOP      8#   2341
.$SIZE     10#   2353
.$SPAC      8#
.$SWDO      8#
.$TRAP      8#   2366
.$TYPB      9#
.$TYPD      9#   2340
.$TYPE      7#     8#   2342
.$TYPO      7#   2344


. ABS.  064002     000     CON   RW    ABS   LCL   I


 ERRORS DETECTED:  0

 CZNCCC,CZNCCC/CRF=CZNCCC
 RUN-TIME: 33 24 2 SECONDS
 RUN-TIME RATIO: 286/60=4.7
 CORE USED:  30K  (59 PAGES)
```