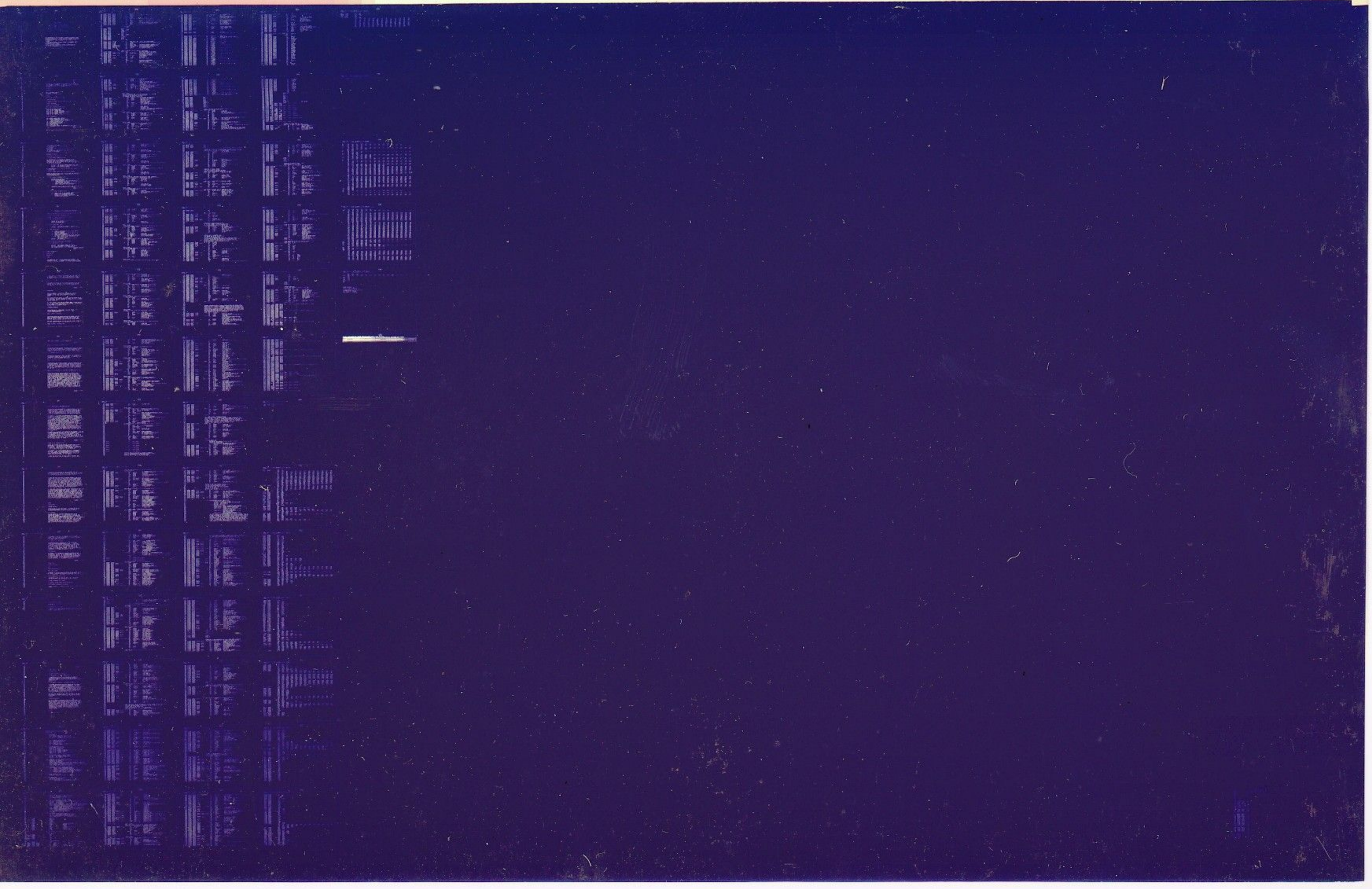
AD01-D

DIAGNOSTIC
MD-11-DZADB-B

EP-DZADB-B-DL-A

FICHE 1 OF 1

NOV 1976
digital
MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZADB-B

PRODUCT NAME: AD01-D DIAGNOSTIC TEST

DATE: MARCH 21,1976

MAINTAINER: DIAGNOSTIC GROUP

1. ABSTRACT

THIS TEST IS TO BE USED AS AN A/D DIAGNOSTIC FOR THE PDP-11 WITH THE ADDI-D CONVERTER. IT TESTS ALL LOGIC FUNCTIONS OF THE CONVERTER, WITH PROVISIONS FOR TESTING DRIFT, CALIBRATION, DIFFERENTIAL LINEARITY, AND REPEATABILITY AS WELL.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD COMPUTER
ADDI-D A/D CONVERTER
CALIBRATED DC VOLTAGE SOURCE

2.2 STORAGE

THE ROUTINE USES MEMORY 0 TO 16520

3. LOADING PROCEDURE

PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

BASIC SWITCH REGISTER SETTINGS ARE:

SW15=1 OR UP---HALT ON ERROR
SW14=1 OR UP---SCOPE LOOP
SW13=1 OR UP---INHIBIT PRINT OUT
SW11=1 OR UP---INHIBIT SUB-PROGRAM ITERATION
SW10=1 OR UP---INHIBIT AVERAGING
SW09=1 OR UP---INHIBIT + OR - 1 LSB TESTING
SW08=1 OR UP---TEST SIGN OPTION

4.2 STARTING ADDRESSES

200 = NORMAL (WORST CASE) TESTING
210 = RESTART ADDRESS FOR NORMAL TESTING
220 = DISPLAY CONVERSION LOOP (IF CPU IS AN 11/05
    USE SA 250 WITH SW10 SET INSTEAD)
230 = EXTERNAL CONVERSION LOOP
240 = FAST EXTERNAL CONVERSION LOOP
250 = CONVERSION PRINT LOOP
260 = SINGLE TEST LOOP
270 = MULTIPLEXER CHANNEL AND REPEATABILITY TEST (WAS-IS TEST)
300 = REPEATABILITY TEST (3 SIGMA SPECIFICATION TEST)

4.3 PROGRAM AND/OR OPERATOR ACTION

```
                        LOAD PROGRAM INTO MEMORY.
                        SET SWITCH REGISTER TO STARTING ADDRESS.
                        LOAD ADDRESS.
                        SET SWITCHES (SEE 4.1 AND 5.1) ALL DOWN FOR WORST CASE.
                        PRESS START.
                        THE PROGRAM WILL LOOP AND BELL WILL RING ONCE
                        PER PASS OF THE PROGRAM.
                        A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

                5.      OPERATING PROCEDURE

                5.1     OPERATIONAL SWITCH SETTINGS

                5.1.1   AT SA 200 (NORMAL INSTRUCTION TEST)...

                        ALL SWITCHES DOWN WILL EXECUTE ALL PORTIONS OF THE
                        BASIC TEST WHICH DO NOT REQUIRE OPERATOR INTERVENTION.
                        A PRINTOUT WILL OCCUR ON ERRORS, AND THE BELL WILL RING AT
                        THE COMPLETION OF EACH PASS.  AT THE START OF THE TEST, THE
                        PROGRAM HALTS THREE TIMES:

                                1ST HALT - LOAD INITIAL CHANNEL NUMBER (SW0 - SW5)
                                2ND HALT - LOAD NUMBER OF CHANNELS TO BE TESTED (SW0 - SW6)
                                3RD HALT - SET SR OPTIONS (SEE 4.1)

                        ALL SWITCHES DOWN AT ALL HALTS IS TAKEN AS WORST CASE TESTING
                        ON CHANNEL ZERO.

                5.1.2   AT SA 210 (NORMAL INSTRUCTION TEST RESTART ADDRESS)...

                        INSTEAD OF USING INITIAL HALTS, THIS SA USES THE INFORMATION
                        FROM THE PREVIOUS START AT 200.  SWITCH OPTIONS MAY BE SET AS
                        DESIRED BEFORE STARTING.

                5.1.3   AT SA 220 (DISPLAY CONVERSION LOOP)...

                                SW0-SW5 SET MUX CHANNEL ON WHICH YOU WISH TO CONVERT
                                SW6-SW7 SET DESIRED GAIN
                                SW14=1 CAUSES CONVERTER TO GO AT MAX RATE FOR 64
                                       CONVERSIONS AND THEN DISPLAY THE RESULTS, ONE AT A
                                       TIME, FOR ABOUT 40 MS EACH.
                                SW15=1 CAUSES A HALT AFTER EACH CONVERSION WITH THE
                                       RESULT DISPLAYED IN THE DATA LIGHTS

                        NOTE:   THIS ROUTINE WILL NOT RUN ON A PDP-11/05. USE SA 250 WITH
                                SW10 SET TO GET CONVERSION PRINTOUT INSTEAD.
```



```
                5.1.4   AT SA 230 (EXTERNAL CONVERSION LOOP)...

                                SW5=1 CAUSES IT NOT TO SET EXTERNAL ENABLE TO MAKE
                                      CERTAIN THAT NO CONVERSION OCCURS
                                SW6=1 CAUSES IT TO CONTINUOUSLY RUN INTERNAL
                                      CONVERSIONS WITH EXTERNAL ENABLE SET, AND TO
                                      PRINT OUT WHEN ERROR SETS
```

```
                                        SW7=1 PREVENTS IT FROM CLEARING DONE AFTER EACH
                                              CONVERSION, TO ALLOW CHECKING THE ERROR FUNCTION

                                THESE SWITCHES ARE MUTUALLY EXCLUSIVE AND ARE SCANNED IN ORDER.

                        5.1.5   AT SA 240 (FAST EXTERNAL CONVERSION LOOP)...

                                SWITCHES 13,14 AND 15 FUNCTION AS IN 4.1 ABOVE.

                        5.1.6   AT SA 250 (CONVERSION PRINT LOOP)...

                                        SW0-SW5 SET MUX CHANNEL ON WHICH YOU WISH TO CONVERT
                                        SW6-SW7 SET DESIRED GAIN
                                        SW10=1 INHIBITS AVERAGING

                        5.1.7   AT SA 260 (SINGLE TEST LOOP)...

                                        1ST HALT - LOAD STARTING ADDRESS OF DESIRED TEST
                                        2ND HALT - LOAD SR OPTIONS (BIT 11 MUST = 0, HOWEVER)

                        5.1.8   AT SA 270 (WAS-IS TEST)...

                                        SW5=1 CAUSES 8 COMPARISONS ON EACH CHANNEL BEFORE
                                              SWITCHING CHANNELS
                                        SW6=1 CAUSES A COMPARISON OF INITIAL TABLE AND NEW
                                              TABLE TO BE PRINTED OUT
                                        SW7=1 CAUSES WAS-IS COMPARISION OF INITIAL TABLE AND
                                              NEW TABLE, INSTEAD OF OLD TABLE AND NEW TABLE
                                        1ST HALT - LOAD INITIAL CHANNEL (BITS 05-00) AND
                                                   GAIN (BITS 07-06)
                                        2ND HALT - LOAD NUMBER OF CHANNELS TO BE TESTED
                                        3RD HALT - SET SR OPTIONS

                                THE BELL RINGS ONCE AFTER EVERY 4096 PASSES.

                        5.1.9   AT SA 300 (REPEATABILITY TEST)

                                        1ST HALT - LOAD INITIAL CHANNEL (SW 05-00)
                                        2ND HALT - LOAD NUMBER OF CHANNELS TO BE TESTED
                                        SW6=1 CAUSES PRINTOUT OF EACH DISTRIBUTION
                                        SW7=1 CAUSES PRINTOUT OF ONLY THOSE DISTRIBUTIONS WHICH FAIL
                                                                                MAINDEC-11-DZAD8-B
                                                                                PAGE 5

                        5.2     SUBROUTINE ABSTRACTS

                        5.2.1   BEGIN SA 200

                        5.2.2   RESTART 210

                        5.2.3   SCOPE
                                -----

                                THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE
                                INSTRUCTION SECTION.  IT RECORDS THE STARTING ADDRESS OF EACH
                                SUB-TEST AS IT IS BEING ENTERED.  IF A SCOPE LOOP IS REQUESTED,
```

IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP
IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL
BE 2048 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST
IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS.

5.2.4 HLT

THIS SUBROUTINE PRINTS OUT THE LOCATION COUNTER AT THE TIME OF
FAILURE, AND THE CONTENTS OF THE PROCESSOR STATUS REGISTER.
NOTE THAT THE LOCATION COUNTER WILL BE THE ADDRESS OF THE
HLT PLUS TWO.

5.2.5 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0
DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND
INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF
MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE
NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH A HALT (000000).
THUS AN ILLEGAL TRAP OR INTERRUPT WILL CAUSE A HALT AT THE TRAP
LOCATION PLUS TWO.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA, EXAMINE REGISTER
SIX. IT WILL CONTAIN THE CURRENT STACK ADDRESS. THE CONTENTS
OF THE CURRENT STACK ADDRESS IS THE VALUE OF THE LOCATION COUNTER
WHEN THE TRAP OR INTERRUPT OCCURRED.

5.2.6 CONVERSION AVERAGING

THIS IS DONE WITH 2 SUBROUTINES, CONAV AND CONVRT, WHICH
TOGETHER AVERAGE 128 CONVERSIONS. SW10=1 INHIBITS
CALLING THESE SUBROUTINES.

5.2.7 DSPLAY (DISPLAY CONVERSION)

THIS ROUTINE READS THE SWITCH REGISTER FOR CHANNEL AND GAIN,
DOES A CONVERSION CORRESPONDING TO THAT, AND DISPLAYS THE
RESULT IN THE DATA LIGHTS. THUS CALIBRATION CAN BE READILY
PERFORMED, AND ALL CHANNELS AND GAIN CAN BE CHECKED DIRECTLY.

NOTE THAT THIS ROUTINE WILL NOT RUN ON A PDP 11/05. SA 250 (WITH
SW10 SET) SHOULD BE USED TO GET CONVERSION PRINTOUT INSTEAD.

5.2.8 EXTEST (EXTERNAL CONVERSION TEST)

THIS ROUTINE COUNTS EXTERNAL CONVERSIONS AND PRINTS OUT AFTER

EVERY TENTH CONVERSION. IT ALSO ALLOWS CHECKING FOR VARIOUS
ERROR CASES INVOLVING EXTERNAL CONVERSIONS.

5.2.9 EXFAST (FAST EXTERNAL CONVERSION TEST)

THIS ROUTINE ENABLES EXTERNAL CONVERSIONS AND CHECKS THE ERROR
BIT AFTER EACH. AN ERROR WILL OCCUR WHEN THE EXTERNAL CONVERSION
SIGNAL IS OCCURRING FASTER THAN THE MAXIMUM USABLE RATE OF
CONVERSION, AND ALLOWS CHECKING THE MAXIMUM POSSIBLE THROUGHPUT.

5.2.10 FLY (CONVERSION PRINT LOOP)

THIS ROUTINE CONTINUOUSLY PRINTS THE RESULTS OF CONVERSIONS AT
THE GAIN AND CHANNEL SET IN THE SWITCHES. THE AVERAGE OF 128
CONVERSIONS IS PRINTED UNLESS SW10 IS SET, IN WHICH CASE THE
RESULTS OF SINGLE CONVERSIONS ARE PRINTED.

5.2.11 TESTX (SINGLE TEST LOOP)

THIS ROUTINE ALLOWS A SINGLE SUBTEST TO BE RUN CONTINUOUSLY FOR
SCOPE LOOP PURPOSES. WHILE A SCOPE LOOP SWITCH OPTION EXISTS,
IT REQUIRES THAT YOU ARE WITHIN THE TEST IN WHICH YOU WISH TO LOOP.
IN SOME CASES (SUCH AS WITH INTERMITTENT FAILURES) THAT'S NOT
EASY TO DO. THIS SUBROUTINE ALLOWS YOU TO LOAD THE ADDRESS OF
ANY TEST FROM TEST0 THRU TEST40 AT THE HALT AND THEN GO DIRECTLY
TO THAT TEST.

5.2.12 WAS-IS (MULTIPLEXER STATUS, SELECTION, AND REPEATABILITY TEST)

THIS ROUTINE STORES AN INITIAL CONVERSION VALUE FOR EACH
CHANNEL IN THE INITIAL TABLE (ITABLE). THIS INITIAL VALUE
IS EITHER THE AVERAGE OF 128 CONVERSIONS OR THE RESULT OF A
SINGLE CONVERSION, DEPENDING ON SW10. THEN A SINGLE CONVERSION IS
TAKEN ON EACH CHANNEL AND STORED IN OTABLE (OLD TABLE). FROM
THEN ON, SETS OF ONE CONVERSION FOR EACH CHANNEL ARE TAKEN AND
STORED IN NTABLE (NEW TABLE). OTABLE AND NTABLE ARE COMPARED
FOR + OR - 1 LSB AGREEMENT (UNLESS SW9 IS SET, IN WHICH CASE
THEY MUST AGREE EXACTLY), AND ERRORS ARE PRINTED OUT IN A
"CHANNEL XX WAS YYYYYY, IS ZZZZZZ" FORMAT. AFTER THE COMPARISON
OTABLE IS UPDATED WITH NTABLE AND THE PROCESS IS REPEATED.
IF SW5 IS SET, 8 COMPARISONS ARE MADE ON EACH CHANNEL BEFORE
GOING TO THE FOLLOWING CHANNEL. WHENEVER SW6 IS SET, A
COMPARISON OF ITABLE AND NTABLE IS PRINTED OUT, WHICH ALLOWS
DRIFT TO BE CHECKED. IF SW7 IS SET, NTABLE IS COMPARED WITH
ITABLE INSTEAD OF OTABLE. THE BELL IS RUNG AFTER EVERY 4096
PASSES.

5.2.13 SIGMA1 (REPEATABILITY SPECIFICATION TEST)

THIS ROUTINE TESTS REPEATABILITY TO SPECIFICATIONS ON THE CHANNELS DESIRED AT GAINS OF 4 AND 8. TEN THOUSAND CONVERSIONS ARE TAKEN AT EACH GAIN. THE SPECIFICATIONS HAVE BEEN INTERPRETED TO MEAN THAT 35 CONVERSIONS OUT OF THE TEN THOUSAND MAY BE OUTSIDE OF 2 STATES AT A GAIN OF 4, AND THAT 35 CONVERSIONS OUT OF TEN THOUSAND MAY BE OUTSIDE OF 3 STATES AT A GAIN OF 8.

AT A GAIN OF 4, THE FIRST 128 CONVERSIONS ARE AVERAGED. THEN TEN THOUSAND CONVERSIONS ARE MADE, AND EACH IS COMPARED TO THE AVERAGE. IF IT IS EQUAL TO THE AVERAGE,"EQUAL" IS INCREMENTED. IF IT IS ONE LESS THAN THE AVERAGE, "UNDER" IS INCREMENTED. IF IT IS ONE GREATER THAN THE AVERAGE, "OVER" IS INCREMENTED. IF IT IS FARTHER FROM THE AVERAGE THAN ONE COUNT, "OUT" IS INCREMENTED. AT THE END OF THE TEN THOUSAND CONVERSIONS, A CHECK IS MADE TO DETERMINE THE 2 ALLOWED STATES (EITHER EQUAL AND OVER OR EQUAL AND UNDER). THE TOTAL FOR THE OTHER STATE (EITHER OVER OR UNDER) IS THEN ADDED TO THE LOCATION "OUT". IF THIS TOTAL (OUT) IS NOW GREATER THAN 35 (DECIMAL), THE TEST FAILS. THE DISTRIBUTION PRINTED OUT WITH SW6 OR SW7 UP IS THIS FINAL SETUP - THE NUMBER EQUAL TO THE AVERAGE, THE NUMBER ONE LESS THAN THE AVERAGE, THE NUMBER ONE GREATER THAN THE AVERAGE, AND THE TOTAL OF THOSE DIFFERING BY MORE THAN ONE COUNT PLUS THOSE IN THE NON-ALLOWED STATE.

AT A GAIN OF 8, THE FIRST 128 CONVERSIONS ARE AVERAGED. THEN TEN THOUSAND CONVERSIONS ARE MADE, AND EACH IS COMPARED TO THE AVERAGE. IF IT IS WITHIN PLUS OR MINUS ONE OF THE AVERAGE, "EQUAL" IS INCREMENTED. IF NOT, "OUT" IS INCREMENTED. AT THE END OF TEN THOUSAND CONVERSIONS, IF MORE THAN 35 WERE OUTSIDE OF THE 3 ALLOWED STATES THE TEST FAILS. THE DISTRIBUTION PRINTED OUT WITH SW6 OR SW7 UP IS SIMPLY THE NUMBER THAT WERE WITHIN THE 3 STATES AND THE NUMBER OUTSIDE OF 3 STATES.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOADING AND STARTING AT 200 WITH ALL SWITCHES DOWN IS WORST CASE TESTING. IF AN ERROR IS DETECTED HERE, THERE WILL BE A PRINTOUT. WHEN AN ERROR IS DETECTED AND IT IS NECESSARY TO SCOPE ON IT, PLACE SW15 UP TO HALT ON ERROR, THEN SW14 UP TO LOOP ON ERROR, THEN SW13 UP TO DELETE PRINTOUTS.

5.3.2 TO GO DIRECTLY TO A SINGLE SUBTEST AND RUN IT CONTINUOUSLY, USE SA 260. INPUT THE STARTING ADDRESS OF THE DESIRED TEST (I.E. THE ADDRESS OF THE CORRESPONDING TESTXX TAG) AT THE INITIAL HALT AND CONTINUE. PUTTING SWITCH 11 UP RESTORES NORMAL PROGRAM FLOW THROUGH ALL THE SUBTESTS.

5.3.3 TO CHECK THE SIGN OPTION, SET SW8=1 DURING THE BASIC TEST (SA 200) AND INPUT POSITIVE OR NEGATIVE VOLTAGES AS DIRECTED.

5.3.4 TO CALIBRATE THE CONVERTER AND CHECK CHANNELS AND GAIN DIRECTLY, USE SA 220 (DISPLAY CONVERSION LOOP) WITH A CALIBRATED DC VOLTAGE SOURCE. IF THE CPU IS AN 11/05, USE SA 250 (WITH SW 10 SET) INSTEAD (CONVERSIONS WILL BE PRINTED ON THE TTY).

5.3.5 TO CHECK FOR NOISE AND DRIFT ON VARIOUS CHANNELS, USE SA 270 (WAS-IS TEST) WITH A DC VOLTAGE SOURCE TO EACH CHANNEL BEING TESTED. AT THE FIRST HALT, INPUT THE LOWEST CHANNEL NUMBER OF THOSE YOU WISH TO TEST IN SW05-SW00, AND THE GAIN SETTING AT WHICH YOU WISH TO TEST THEM IN SW07-SW06. AT THE SECOND HALT LOAD IN THE NUMBER OF CHANNELS YOU WISH TO TEST (ZERO WILL BE TAKEN AS ONE). AT THE THIRD HALT, SET IN THE SWITCH REGISTER OPTIONS AS DESIRED. DRIFT CAN BE TESTED BY RUNNING THE TEST AS LONG AS DESIRED (WITH SW13 SET TO INHIBIT PRINTOUTS IF DESIRED) AND THEN SETTING SW06 TO ZERO. THIS CAUSES A COMPARISON OF THE INITIAL VALUES AND CURRENT VALUES TO BE PRINTED OUT.

5.3.6 TO CHECK NOISE STATISTICALLY, USE SA 300 (REPEATABILITY TEST) WITH A DC VOLTAGE ON EACH CHANNEL BEING TESTED. AT THE FIRST HALT, SET THE INITIAL CHANNEL NUMBER TO BE TESTED IN SW05-SW00. AT THE SECOND HALT SET THE NUMBER OF CONSECUTIVE CHANNELS TO BE TESTED IN SW06-SW00. THE PASS-FAIL ASPECT OF THE TEST IS BASED ON THE SPECIFICATION FOR REPEATABILITY OF THE CONVERTER, AND REQUIRES THAT THE INPUT BE GROUNDED. HOWEVER, BY RUNNING WITH AN INPUT OTHER THAN GROUND, ENVIRONMENTAL NOISE CAN BE CHECKED. BY SETTING SW6 TO ONE, EACH DISTRIBUTION IS PRINTED OUT. THIS CAN BE USED AS AN INDICATION OF NOISE IRREGARDLESS OF THE TEST PASSING OR FAILING. AN EXPLANATION OF THE PRINTOUT IS GIVEN IN 5.2.13.

6. ERRORS

6.1 ERROR PRINTOUT

6.1.1 STANDARD PRINTOUT

PRINTOUTS ARE IN A TWO-WORD FORMAT. THE FIRST IS THE PC+2 OF THE DETECTED ERROR. THE SECOND IS THE CONTENTS OF THE PROCESSOR STATUS REGISTER WHEN THE ERROR WAS DETECTED.

6.1.2 GAIN LINEARITY

THE LAST SUBTEST IN THE BASIC INSTRUCTION TEST IS FOR DIFFERENTIAL LINEARITY. 1 1/4 LSB ACCURACY IS CHECKED. THE VALUE AT THE HIGHEST GAIN IS USED AS A BASIS FOR COMPARISON SINCE IT OFFERS THE HIGHEST RESOLUTION. USING 1 1/4 LSB TESTING, THE VALUE READ AT A GAIN OF 8 MUST BE WITHIN + OR - 1 1/4 LSB OF THE ACTUAL VOLTAGE INPUT. AT ANY OTHER GAIN, THE VALUE READ MUST ALSO BE WITHIN + OR - 1 1/4 LSB, AND OF COURSE 1 LSB HAS A DIFFERENT VOLTAGE EQUIVALENT AT EACH GAIN. A ROUTINE IS USED TO DETERMINE MAXIMUM AND MINIMUM ALLOWABLE VALUES AT GAINS OF 4,2, AND 1 GIVEN THE VALUE READ AT A GAIN OF 8.

IF AN ERROR IS DETECTED, THE MESSAGE "GAIN NOT LINEAR WITHIN + OR- 1 1/4 LSB" IS PRINTED OUT. THIS IS FOLLOWED BY A TABLE WHICH GIVES THE VALUES READ, AND THE MAXIMUM AND MINIMUM ALLOWABLE VALUES AT EACH GAIN.

6.1.3 CH XX WAS YYYYYY, IS ZZZZZZ

THIS ERROR MESSAGE IS PRINTED OUT BY THE MULTIPLEXER STATUS, SELECTION AND REPEATABILITY TEST. IT INDICATES A FAILURE OF 2 CONSECUTIVE CONVERSIONS ON A GIVEN CHANNEL (OR OF THE INITIAL VALUE AND THE CURRENT VALUE IF SW7 IS SET) TO BE WITHIN + OR - 1 LSB (OR TO BE EQUAL IF SW9 IS SET). XX IS THE CHANNEL NUMBER, YYYYYY IS THE RESULT OF THE PREVIOUS CONVERSION, AND ZZZZZZ IS THE RESULT OF THE CURRENT CONVERSION. ALL NUMBERS ARE PRINTED IN OCTAL.

6.2 ERROR RECOVERY

IN GENERAL, TEST FAILURES WILL PRINTOUT AN ERROR MESSAGE AND CONTINUE. IF THE "HALT ON ERROR" SWITCH IS SET, HITTING CONTINUE WILL RECOVER. IF THE PROGRAM HANGS UP IN A LOOP, THE ERROR IS LIKELY TO BE A SIGNAL WHICH WAS NEVER RECEIVED. IF A HALT OCCURS IN THE TRAP AND VECTOR AREA THE PROGRAM MUST BE RESTARTED. IF THE PROGRAM HALTS IN THE MAIN FLOW, CONSULT THE LISTING IF NO MESSAGE IS TYPED OUT.

7. RESTRICTIONS

7.1 STARTING PROCEDURE

NONE

7.2 OPERATIONAL RESTRICTIONS

7.2.1 BASIC TEST (SA200)

A CONSTANT DC VOLTAGE SHOULD BE APPLIED TO THE LOWEST CHANNEL TO BE TESTED. THIS VOLTAGE SHOULD BE NOT GREATER THAN 1.25 VOLTS IN MAGNITUDE. NO EXTERNAL TRIGGER SIGNAL SHOULD BE PRESENT.

7.2.2 DISPLAY CONVERSION TEST (SA 220)

THIS ROUTINE CANNOT BE USED ON A PDP 11/05. USE SA 250 (WITH SW10 SET) TO GET A CONVERSION PRINTOUT INSTEAD.

7.2.3 EXTERNAL CONVERSION TEST (SA230)

AN EXTERNAL TRIGGER SIGNAL SHOULD BE SUPPLIED.

7.2.4 FAST EXTERNAL CONVERSION LOOP (SA240)

A SIGNAL GENERATOR IS NEEDED TO PROVIDE A RAPID EXTERNAL TRIGGER SIGNAL.

7.2.5 MULTIPLEXER TEST (SA270)

A CONSTANT DC VOLTAGE SHOULD BE APPLIED TO EACH CHANNEL BEING TESTED.

7.2.6 REPEATABILITY TEST (SA 300)

A CONSTANT DC VOLTAGE SHOULD BE APPLIED TO EACH CHANNEL BEING TESTED. TO TEST THE CONVERTER TO SPECIFICATION, THE INPUT MUST BE THE CONVERTER GROUND.

8. MISCELLANEOUS

8.1 EXECUTION TIME

EACH PASS TAKES ABOUT 30 SECONDS.

8.2 TESTING SYSTEMS WITH MORE THAN ONE ADD1-D

BY SUBSTITUTING INTO THE LOCATIONS ADCS AND ADDBR THE ADDRESS OF THE CSR AND DBR OF A CONVERTER ASSIGNED A NON-STANDARD ADDRESS, AND SUBSTITUTING ITS INTERRUPT VECTOR ADDRESS INTO ADINT, AN ADD1-D MAY BE TESTED AT ANY ADDRESS ASSIGNED TO IT.

8.3 SPECIAL NOTES ON EXTERNALLY TRIGGERED CONVERSIONS

IF EXTERNAL CONVERSIONS HAVE BEEN ENABLED ONE CAN CHANGE THE CHANNEL NUMBER WITHOUT STARTING A CONVERSION. THIS LEADS TO THE PROBLEM OF A CONVERSION OCCURRING WHILE YOU ARE CHANGING THE CHANNEL NUMBER. THE RESULTS OF SUCH A CONVERSION WILL OF COURSE BE ERRONEOUS, BUT THE ERROR BIT WILL NOT BE SET. THUS, IT'S A GOOD PRACTICE TO CHANGE CHANNELS OR GAIN ONLY WHEN YOU KNOW NO SIGNAL IS COMING IN (SUCH AS RIGHT AFTER A CONVERSION HAS SET DONE) OR ELSE TO DISCARD THE FIRST CONVERSION AFTER YOU CHANGE CHANNEL AND GAIN.

NOTE ALSO THAT WHETHER OR NOT A DATO OR DATOB TO THE HIGH BYTE CAUSES A CONVERSION DEPENDS ON THE STATE OF EXTERNAL ENABLE BEFORE THE DATO INSTEAD OF THE STATE RESULTING FROM THE DATO.

9. PROGRAM DESCRIPTION

THIS SET OF TESTS IS DESIGNED TO TEST ALL OPERATIONS OF THE ADD1-D CONVERTER, WITH THE NECESSARY EXCEPTION THAT TIMING IN MOST CASES IS ONLY PARTIALLY TESTED. THE SEVERAL MODES OF OPERATING THE CONVERTER EXTEND THE USEFULNESS OF THE DEVICE AND THE COMPLEXITY OF THE DIAGNOSTIC. THE BASIC TEST IS DESIGNED TO TEST ALL OPERATIONS NOT REQUIRING OPERATOR INTERVENTION, SO EXTERNAL TRIGGERING TESTS ARE SEPARATE. THE REPERTOIRE OF SPECIAL TESTS IS DESIGNED TO GIVE THE OPERATOR MAXIMUM FLEXIBILITY IN CALIBRATION, CHECKOUT, AND FAILURE LOCATION.

10. LISTING

SEE FOLLOWING PAGES.

%

```
                        ;DIAGNOSTIC FOR THE ADD1-D A/D CONVERTER
                        ;COPYRIGHT 1970, 1971, 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;BY RICK FADDEN

                        ;STARTING ADDRESSES ARE:
                                ;200 = NORMAL (WORST CASE) TESTING
                                ;210 = RESTART ADDRESS FOR NORMAL TESTING
                                ;220 = DISPLAY CONVERSION LOOP (IF CPU IS A PDP 11/05,
                                ;               USE SA 250 (WITH SW10 SET) INSTEAD)

                                ;230 = EXTERNAL CONVERSION LOOP
                                ;240 = FAST EXTERNAL CONVERSION LOOP
                                ;250 = CONVERSION PRINT LOOP
                                ;260 = SINGLE TEST LOOP
                                ;270 = MULTIPLEXOR CHANNEL AND REPEATABILITY TEST (WAS-IS TEST)
                                ;300 = REPEATABILITY TEST (3 SIGMA)

                        ;BASIC SWITCH REGISTER SETTINGS ARE:
                                ;SW15=1 OR UP---HALT ON ERROR
                                ;SW14=1 OR UP---SCOPE LOOP
                                ;SW13=1 OR UP---INHIBIT PRINT OUT
                                ;SW11=1 OR UP---INHIBIT SUB-PROGRAM ITERATION
                                ;SW10=1 OR UP---INHIBIT AVERAGING
                                ;SW09=1 OR UP---INHIBIT + OR - 1 LSB TESTING
                                ;SW08=1 OR UP---TEST SIGN OPTION

                        ;NORMAL TESTING (SA200) HALTS 3 TIMES AT START OF PROGRAM
                                ;1ST HALT - LOAD INITIAL CHANNEL NUMBER
                                ;2ND HALT - LOAD NUMBER OF CHANNELS TO BE TESTED
                                ;3RD HALT - SET SR OPTIONS

                        ;DISPLAY CONVERSION LOOP HAS SEPARATE SR SETTINGS:
                                ;SW0-SW5 SET MUX CHANNEL ON WHICH YOU WISH TO CONVERT
                                ;SW6-SW7 SET DESIRED GAIN
                                ;SW14=1 CAUSES CONVERTER TO GO AT MAX RATE FOR 100(8) CONVERSIONS
                                ;AND THEN DISPLAY THE RESULTS, ONE AT TIME, FOR ABOUT 40 MS. EACH
                                ;SW15=1 CAUSES A HALT AFTER EACH CONVERSION, WITH THE RESULTS DISPLAYED
                                ;THESE SWITCHES ARE SCANNED DURING EACH CYCLE

                        ;EXTERNAL CONVERSION LOOP HAS 3 SPECIAL SR SETTINGS:
                                ;SW5=1 CAUSES IT TO NOT SET EXTERNAL ENABLE, TO MAKE
                                        ;CERTAIN THAT NO CONVERSIONS OCCUR
                                ;SW6=1 CAUSES IT TO CONTINUOUSLY RUN INTERNAL CONVERSIONS WITH
                                        ;EXTERNAL ENABLE SET, AND TO PRINTOUT WHEN ERROR SETS
                                ;SW7=1 PREVENTS IT FROM CLEARING DONE AFTER EACH CONVERSION, TO
                                        ;ALLOW CHECKING THE ERROR FUNCTION

                        ;FAST EXTERNAL CONVERSION LOOP
                                ;SWITCHES 13,14,AND 15 ARE USED AS IN THE BASIC SETTINGS

                        ;CONVERSION PRINT LOOP
                                ;SW0-SW5-SET MUX CHANNEL ON WHICH YOU WISH TO CONVERT
                                ;SW6-SW7-SET DESIRED GAIN
                                ;SW10=0 CAUSES AVERAGE OF 128 CONVERSIONS TO BE PRINTED
                                ;       =1 CAUSES INDIVIDUAL CONVERSIONS TO BE PRINTED
```

```
                                ;SINGLE INSTRUCTION TEST LOOP (SA 260) HALTS TWICE
                                        ;1ST HALT- LOAD STARTING ADDRESS OF THE TEST YOU WISH
                                        ;TO RUN (THE ADDRESS OF THE TESTXX TAG)
                                        ;2ND HALT- SET SR OPTIONS (BIT 11 MUST =0)

                                ;WAS-IS TEST HALTS 3 TIMES AT THE START:
                                        ;1ST HALT - LOAD INITIAL CHANNEL (BITS 05-00) AND GAIN (BITS 07-06)
                                        ;2ND HALT - LOAD NUMBER OF CHANNELS TO BE TESTED
                                        ;3RD HALT - SET SR OPTIONS
                                        ;SW5=1 CAUSES 8 COMPARISONS ON EACH CHANNEL BEFORE SWITCHING CHANNELS
                                        ;SW6=1 CAUSES A COMPARISON OF INITIAL TABLE AND NEW TABLE TO BE PRINTED OUT
                                        ;SW7=1 CAUSES WAS-IS COMPARISON OF INITIAL TABLE AND NEW TABLE,
                                          ;INSTEAD OF OLD TABLE AND NEW TABLE
                                        ;SW9=1 INHIBITS + OR - 1 LSB TESTING (VALUES MUST AGREE EXACTLY)
                                        ;SW10=1 INHIBITS AVERAGING WHEN CREATING THE INITIAL TABLE
                                        ;SW13=1 INHIBITS ERROR PRINTOUTS SO THAT DRIFT CAN BE CHECKED
                                          ;WITHOUT EXTRANEOUS PRINT OUTS
                                        ;BELL RINGS AFTER EVERY 4096 PASSES

                                ;REPEATABILITY TEST (3 SIGMA) HALTS TWICE
                                        ;1ST HALT - LOAD INITIAL CHANNEL (BITS 05-00)
                                        ;2ND HALT - LOAD NUMBER OF CHANNELS TO BE TESTED
                                        ;SW6=1 CAUSES PRINTOUT OF EACH DISTRIBUTION
                                        ;SW7=1 CAUSES PRINTOUT OF ONLY THOSE DISTRIBUTIONS WHICH FAIL

                                        SR=177570
                                        PSR=177776
                                        NOP=240
                                        HLT=EMT
                                        SCOPE=TRAP
                                        R0=%0                   ;SCRATCH AND DISPLAY
                                        ADCSR=%1                ;ADDRESS OF STATUS REGISTER
                                        R2=%2                   ;SCRATCH
                                        R3=%3                   ;SCRATCH
                                        INTVC=%4                ;INTERRUPT VECTOR ADDRESS
                                        R5=%5                   ;SCRATCH
                                        SP=%6                   ;STACK POINTER
                                        PC=%7                   ;PROGRAM COUNTER

                                ;LOAD TRAP CATCHER INTO 0 THRU 377

                                ;LOAD TRAP VECTORS
                                        .=30
                                        PRINT
                                        340
                                        .=34
                                        SCOPEC
                                        340

                                ;LOAD STARTING ADDRESS AREA
                                        .=200                   ;NORMAL (WORST CASE) STARTING ADDRESS
000200  012706  000600                  MOV     #STACK,%6
000204  000167  000472                  JMP     BEGIN
000210  012706  000600                  MOV     #STACK,%6       ;RESTART STARTING ADDRESS FOR
000214  000167  000512                  JMP     RESTRT          ;WORST CASE TESTING
000220  012706  000600                  MOV     #STACK,%6       ;DISPLAY CONVERSION LOOP
```

```
                                                JMP     DSPLAY
                                                MOV     #STACK,%6       ;EXTERNAL CONVERSION TEST LOOP
                                                JMP     EXTEST
                                                MOV     #STACK,%6
                                                JMP     EXFAST
                                                MOV     #STACK,%6       ;RUNNING CONVERSION PRINTOUT
                                                JMP     FLY
                                                MOV     #STACK,%6       ;ROUTINE TO LOOP THRU A GIVEN INSTRUCTION
                                                JMP     TESTX           ;TEST
                                                MOV     #STACK,%6       ;ROUTINE TO CONVERT ON EACH CHANNEL
                                                JMP     WASIS           ;AND COMPARE TO PREVIOUS CONVERSIONS
                                                MOV     #STACK,%6       ;ROUTINE TO TEST REPEATABILITY
                                                JMP     SIGMA1

                                        ;LOAD DATA AREA
                        000600                  .=600
000600  000000                          STACK:  0
000602  176770                          ADCS:   176770
000604  176772                          ADDBR:  176772
000606  000000                          ADCSRB: 0
000610  000130                          ADINT:  130
000612  000000                          COUNT:  0
000614  177564                          TCSR:   177564
000616  177566                          TDBR:   177566
000620  000000                          INTFLG: 0
000622  000000                          INITCH: 0
000624  000000                          NUMCH:  0
000626  000000                          RETRNX: 0
000630  000000                          PROC:   0

000632  016701  177744          SETUP:  MOV     ADCS,ADCSR      ;LOAD R1 WITH ADCSR ADDRESS
000636  010167  177744                  MOV     ADCSR,  ADCSRB  ;INITIALIZE ODD BYTE CSR ADDRESS POINTER
000642  005267  177740                  INC     ADCSRB
000646  016704  177736                  MOV     ADINT,INTVC     ;INITIALIZE INTERRUPT ADDRESS POINTER
000652  012767  004000  012762          MOV     #4000,ICOUNT    ;INITIALIZE MAXIMUM ITERATION COUNT
000660  012767  000756  012760          MOV     #TEST0+2,RETURN ;SETUP FIRST RETURN ADDRESS FOR SCOPE
000666  012767  000340  177102          MOV     #340,PSR        ;SET PROCESSOR PRIORITY TO 7
000674  005067  177720                  CLR     INTFLG          ;INITIALIZE INTERRUPT FLAG
000700  000207                          RTS     %7

                                        ;CHECK TO SEE THAT CSR WAS CORRECTLY INITIALIZED
000702  000000                  BEGIN:  HALT                    ;SET FIRST CHANNEL TO BE TESTED
000704  116767  176660  177711          MOVB    SR,INITCH+1     ;STORE FIRST CHANNEL TO BE TESTED
000712  042767  140377  177702          BIC     #140377,INITCH  ;MASK OFF ALL BUT CHANNEL
000720  000000                          HALT                    ;SET NUMBER OF CHANNELS TO BE TESTED
000722  016767  176642  177674          MOV     SR,NUMCH        ;STORE NUMBER OF CHANNELS
000730  000000                          HALT                    ;SET SR OPTIONS
000732  004767  177674          RESTRT: JSR     %7,SETUP        ;INITIALIZE
000736  105777  177652                  TSTB    @TCSR           ;WAIT FOR TTY READY
000742  100375                          BPL     .-4
000744  000005                          RESET                   ;REINITIALIZE PERIPHERALS
000746  005711                          TST     @ADCSR          ;CHECK FOR CORRECT INITIALIZATION OF CSR
000750  001401                          BEQ     .+4             ;BRANCH IF ALL ZERO
000752  104000                          HLT                     ;CSR WAS NOT INITIALIZED TO ZERO

000754  104400                  TEST0:  SCOPE
```

```
                                        ;SHOW THAT READ/WRITE BITS OF THE CSR CAN BE SET TO 0 OR 1 AND READ BACK
000756  005777  177622                        TST     @ADDBR          ;CLEAR DONE
000762  005011                                CLR     @ADCSR          ;SET TO 0
000764  032711  177577                        BIT     #177577,@ADCSR  ;CHECK ALL BITS BUT DONE
000770  001401                                BEQ     .+4             ;BRANCH IF ALL BITS READ BACK AS ZERO
000772  104000                                HLT                     ;NOT ALL BITS READ BACK AS ZERO
000774  005777  177604                        TST     @ADDBR          ;CLEAR DONE
001000  052767  000340  176770                BIS     #340,PSR        ;SET PROCESSOR PRIORITY AT 7 TO PREVENT INTERRUPTS
001006  012711  177777                        MOV     #177777,@ADCSR  ;LOAD ONES INTO ALL WRITEABLE BITS
001012  011102                                MOV     @ADCSR, R2      ;DONE MAY OR MAY NOT BE SET
001014  042702  000200                        BIC     #200,   R2      ;SO READ BACK CONTENTS OF THE REGISTER
001020  020227  137536                        CMP     R2,     #137536 ;BUT IGNORE DONE
001024  001401                                BEQ     .+4             ;BRANCH IF CORRECT
001026  104000                                HLT                     ;CSR DID NOT READ BACK CORRECTLY


001030  104400                      TEST1:    SCOPE
                                    ;TEST FOR DONE BEING SET WITHIN 40 MICROSECONDS (OR LONGER
                                    ;DEPENDING ON CPU AND MEMORY) AFTER A/D START
                                    ;TEST READING DBR RESETS DONE TO ZERO
                                    ;TEST INDIRECTLY FOR DONE CLEARING A/D START (VIA SECOND A/D START)
001032  105011                        CLRB    @ADCSR          ;INITIALIZE CSR LOWER BYTE
001034  005777  177544                TST     @ADDBR          ;READ DBR TO CLEAR DONE
001040  105711                        TSTB    @ADCSR          ;CHECK DONE BIT
001042  100001                        BPL     .+4             ;CONTINUE IF CLEARED
001044  104000                        HLT                     ;READING DBR DIDN'T CLEAR DONE
001046  105211                        INCB    @ADCSR          ;START CONVERSION
001050  012767  177740  177534        MOV     #-32.,COUNT     ;LOAD COUNTER FOR TIMING LOOP
001056  005267  177530                INC     COUNT           ;LOOP FOR AT LEAST 40 MICROSECONDS
001062  001375                        BNE     .-4
001064  105711                        TSTB    @ADCSR          ;CHECK DONE BIT
001066  100401                        BMI     .+4             ;CONTINUE IF SET
001070  104000                        HLT                     ;DONE WASN'T SET WITHIN 100 MICROSECONDS.
                                                                ;EITHER A/D DONE PREVIOUSLY FAILED TO CLEAR A/D START,
                                                                ;OR DONE ITSELF FAILED HERE


001072  104400                      TEST2:    SCOPE
                                    ;SHOW THAT A CLEAR INSTRUCTION DOESN'T CLEAR DONE
001074  005777  177504                TST     @ADDBR          ;CLEAR DONE
001100  012711  000001                MOV     #1,@ADCSR       ;INITIALIZE CONVERTER AND START CONVERSION
001104  105711                        TSTB    @ADCSR          ;WAIT FOR DONE
001106  100376                        BPL     .-2
001110  105011                        CLRB    @ADCSR          ;CLEAR LOWER BYTE
001112  105711                        TSTB    @ADCSR          ;CHECK DONE
001114  100401                        BMI     .+4             ;BRANCH IF SET
001116  104000                        HLT                     ;DONE WAS CLEARED


001120  104400                      TEST3:    SCOPE
                                    ;DATOB TO LOWER BYTE OF CSR SHOULDN'T START CONVERSION
001122  005777  177456                TST     @ADDBR          ;CLEAR DONE
001126  105011                        CLRB    @ADCSR          ;DATOB TO LOWER BYTE
001130  012767  177740  177454        MOV     #-32.,COUNT     ;WAIT
001136  005267  177450                INC     COUNT
001142  001375                        BNE     .-4
001144  105711                        TSTB    @ADCSR          ;CHECK DONE BIT
001146  100003                        BPL     .+10            ;BRANCH IF NOT SET
```

```
   775  001150  104000                          HLT                     ;DATOB TO LOWER BYTE CAUSED CONVERSION
   776  001152  005777  177426                  TST     @ADDBR          ;CLEAR DONE
   777  001156  105011                          CLRB    @ADCSR          ;REPEAT TEST, KNOWING BIT1=0
   778  001160  012767  177740  177424          MOV     #-32.,COUNT     ;WAIT AGAIN
   779  001166  005267  177420                  INC     COUNT
   780  001172  001375                          BNE     .-4
   781  001174  105711                          TSTB    @ADCSR          ;CHECK DONE BIT
   782  001176  100001                          BPL     .+4             ;CONTINUE IF CLEARED
   783  001200  104000                          HLT                     ;DATOB TO LOWER BYTE CAUSED CONVERSION
   784
   785  001202  104400                  TEST4:  SCOPE
   786                                   ;NEW MUX AND GAIN VIA DATO WITH BIT1=0 SHOULD START CONVERSION
   787  001204  005777  177374                  TST     @ADDBR          ;CLEAR DONE
   788  001210  105011                          CLRB    @ADCSR          ;CLEAR BIT1 IN CASE PREVIOUSLY SET
   789  001212  005011                          CLR     @ADCSR          ;START CONVERSION
   790  001214  012767  177740  177370          MOV     #-32.,COUNT     ;WAIT AWHILE
   791  001222  005267  177364                  INC     COUNT
   792  001226  001375                          BNE     .-4
   793  001230  105711                          TSTB    @ADCSR          ;CHECK DONE BIT
   794  001232  100401                          BMI     .+4             ;CONTINUE IF SET
   795  001234  104000                          HLT                     ;DONE WASN'T SET WITHIN ABOUT 100 MICROSECONDS
   796
   797  001236  104400                  TEST5:  SCOPE
   798                                   ;NEW MUX VIA DATOB (BIT1=0) SHOULD START CONVERSION
   799  001240  005777  177340                  TST     @ADDBR          ;CLEAR DONE
   800  001244  105011                          CLRB    @ADCSR          ;CLEAR BIT1
   801  001246  105077  177334                  CLRB    @ADCSRB         ;START CONVERSION VIA DATOB TO UPPER BYTE
   802  001252  012767  177740  177332          MOV     #-32.,COUNT     ;SET UP TIMING LOOP
   803  001260  005267  177326                  INC     COUNT           ;WAIT
   804  001264  001375                          BNE     .-4
   805  001266  105711                          TSTB    @ADCSR          ;CHECK DONE BIT
   806  001270  100401                          BMI     .+4             ;CONTINUE IF SET
   807  001272  104000                          HLT                     ;DONE WASN'T SET WITHIN ABOUT 100 MICROSECONDS
   808
   809  001274  104400                  TEST6:  SCOPE
   810                                   ;NEW MUX WITH BIT 1 PREVIOUSLY SET SHOULDN'T START CONVERSION
   811                                   ;NOTE THAT THE EFFECT OF BIT1 DEPENDS ON ITS STATE BEFORE THE DATO
   812  001276  152711  000002                  BISB    #2,@ADCSR       ;SET EXTERNAL ENABLE
   813  001302  005777  177276                  TST     @ADDBR          ;CLEAR DONE
   814  001306  005011                          CLR     @ADCSR          ;DATO TO CSR
   815  001310  012767  177740  177274          MOV     #-32.,COUNT     ;WAIT AWHILE
   816  001316  005267  177270                  INC     COUNT
   817  001322  001375                          BNE     .-4
   818  001324  105711                          TSTB    @ADCSR          ;CHECK DONE BIT
   819  001326  100001                          BPL     .+4             ;CONTINUE IF CLEARED
   820  001330  104000                          HLT                     ;CONVERSION OCCURRED
   821
   822  001332  104400                  TEST7:  SCOPE
   823                                   ;SETTING EXTERNAL ENABLE TO ONE VIA FULL WORD DATO SHOULD CAUSE CONVERSION
   824                                   ;IF BIT1 WAS PREVIOUSLY ZERO
   825  001334  005777  177244                  TST     @ADDBR          ;CLEAR DONE
   826  001340  105011                          CLRB    @ADCSR          ;CLEAR LOW BYTE OF STATUS REGISTER
   827  001342  012711  000002                  MOV     #2,@ADCSR       ;SET EXTERNAL ENABLE
   828  001346  012767  177740  177236          MOV     #-32.,COUNT     ;WAIT AWHILE
   829  001354  005267  177232                  INC     COUNT
   830  001360  001375                          BNE     .-4
```

```
                                                        ;CLEAR EXTERNAL ENABLE
                                                        ;CHECK DONE
                                                        ;BRANCH IF SET
                                                        ;CONVERSION DID NOT OCCUR

                        TEST8:  SCOPE
                        ;SETTING A/D START WITH BIT 1 PREVIOUSLY SET SHOULD START CONVERSION
                                                        ;CLEAR DONE
                                                        ;SET EXTERNAL ENABLE (BIT 1)
                                                        ;SET A/D START
                                                        ;WAIT
                                                        ;CHECK DONE
                                                        ;BRANCH IF SET
                                                        ;CONVERSION DID NOT OCCUR

                        TEST9:  SCOPE
                        ;TEST FOR NO EXTERNAL CONVERSION (DURING BASIC INSTRUCTION TEST NO EXTERNAL
                        ;SIGNAL SHOULD NORMALLY OCCUR).
                                                        ;INITIALIZE CSR LOW BYTE
                                                        ;CLEAR DONE
                                                        ;ENABLE EXTERNAL CONVERSIONS
                                                        ;INITIALIZE COUNTER
                                                        ;WAIT AWHILE
                                                        ;CHECK DONE
                                                        ;CONTINUE IF NO CONVERSION OCCURRED
                                                        ;SETUP MESSAGE ADDRESS
                                                        ;PRINTOUT ERROR MESSAGE

                        TEST10: SCOPE
                        ;TWO SUCCESSIVE DATO'S SHOULD SET THE ERROR FLAG
                                                        ;INITIALIZE CSR LOW BYTE
                                                        ;CLEAR DONE
                                                        ;FIRST DATO, CLEARS ERROR FLAG
                                                        ;SECOND DATO, SHOULD SET ERROR FLAG
                                                        ;CHECK ERROR BIT
                                                        ;CONTINUE IF ERROR WAS SET
                                                        ;ERROR FLAG WASN'T SET

                        TEST11: SCOPE
                        ;NEW MUX FOLLOWED IMMEDIATELY BY SETTING A/D START TO 1
                        ;SHOULD SET ERROR FLAG
                                                        ;CLEAR DONE
                                                        ;INITIALZE LOW BYTE OF CSR
                                                        ;DATO TO CSR
                                                        ;SET A/D START
                                                        ;CHECK ERROR
                                                        ;BRANCH IF SET
                                                        ;ERROR WAS NOT SET
                                                        ;WAIT FOR DONE

                        TEST12: SCOPE
                        ;BEGINNING CONVERSION AFTER SEEING DONE SET SHOULD CAUSE AN ERROR IF
```

```
                                        ;THE PREVIOUS DATA WASN'T READ FROM THE DBR
001544  005777  177034                          TST     @ADDBR          ;CLEAR DONE
001550  012711  000001                          MOV     #1,     @ADCSR  ;INITIALIZE CSR AND START CONVERSION
001554  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
001556  100376                                  BPL     .-2
001560  105211                                  INCB    @ADCSR          ;START SECOND CONVERSION
001562  005711                                  TST     @ADCSR          ;CHECK ERROR BIT
001564  100401                                  BMI     .+4             ;CONTINUE IF ERROR FLAG SET
001566  104000                                  HLT                     ;ERROR FLAG WASN'T SET

001570  104400                          TEST13: SCOPE
                                        ;CLEARING CSR WITHOUT READING PREVIOUS DATA SHOULD SET ERROR
001572  005777  177006                          TST     @ADDBR          ;CLEAR DONE
001576  012711  000001                          MOV     #1,     @ADCSR  ;INITIALIZE CSR AND CONVERT
001602  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
001604  100376                                  BPL     .-2
001606  005011                                  CLR     @ADCSR          ;CLEAR WITHOUT READING DATA
001610  005711                                  TST     @ADCSR          ;CHECK ERROR BIT
001612  100401                                  BMI     .+4             ;CONTINUE IF ERROR FLAG SET
001614  104000                                  HLT                     ;ERROR FLAG NOT SET
001616  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
001620  100376                                  BPL     .-2

001622  104400                          TEST14: SCOPE
                                        ;IF DATA IN BUFFER IS READ BEFORE NEXT CONVERSION
                                        ;NO ERROR SHOULD OCCUR
001624  005777  176754                          TST     @ADDBR          ;CLEAR DONE
001630  012711  000001                          MOV     #1,     @ADCSR  ;INITIALIZE CSR AND START CONVERSION
001634  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
001636  100376                                  BPL     .-2
001640  005777  176740                          TST     @ADDBR          ;READ DBR
001644  105211                                  INCB    @ADCSR          ;START SECOND CONVERSION
001646  005711                                  TST     @ADCSR          ;CHECK ERROR BIT
001650  100001                                  BPL     .+4             ;CONTINUE IF NO ERROR
001652  104000                                  HLT                     ;ERROR FLAG WAS SET
001654  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
001656  100376                                  BPL     .-2             ;TO ALLOW LOOPING ON TEST

001660  104400                          TEST15: SCOPE
                                        ;SETTING A/D START TO ONE TWICE CONSECUTIVELY SHOULD CAUSE
                                        ;AN ERROR
001662  005777  176716                          TST     @ADDBR          ;CLEAR DONE
001666  012711  000001                          MOV     #1,     @ADCSR  ;SET A/D START AND INITIALIZE CSR
001672  005211                                  INC     @ADCSR          ;SET A/D START AGAIN
001674  005711                                  TST     @ADCSR          ;CHECK ERROR BIT
001676  100401                                  BMI     .+4             ;CONTINUE IF ERROR SET
001700  104000                                  HLT                     ;ERROR FLAG WAS NOT SET
001702  105711                                  TSTB    @ADCSR          ;WAIT FOR DONE
001704  100376                                  BPL     .-2             ;TO ALLOW LOOPING ON TEST

001706  104400                          TEST16: SCOPE
                                        ;SETTING A/D START, FOLLOWED IMMEDIATELY BY A DATOB TO HIGH BYTE,
                                        ;SHOULD SET ERROR FLAG
001710  005777  176670                          TST     @ADDBR          ;CLEAR DONE
001714  105011                                  CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF CSR
001716  105211                                  INCB    @ADCSR          ;START CONVERSION
```

```
                                        CLR     @ADCSR          ;DATOB TO HIGH BYTE OF CSR
                                        TST     @ADCSR          ;WHICH ALSO CLEARS PREVIOUS ERROR FLAG
                                        BMI     .+4             ;CHECK ERROR FLAG AND BRANCH IF SET
                                        HLT                     ;ERROR FLAG WAS NOT SET
                                        TSTB    @ADCSR          ;WAIT FOR DONE TO
                                        BPL     .-2             ;ALLOW LOOPING ON TEST

                                TEST17: SCOPE
                                ;DATO WITH BIT 1=1 SHOULDN'T CAUSE ERROR EVEN THOUGH
                                ;PREVIOUS DATA WAS NOT READ
                                        TST     @ADDBR          ;CLEAR DONE
                                        CLRB    @ADCSR          ;INITIALIZE CSR LOW BYTE
                                        MOV     #2,     @ADCSR  ;CLEAR ERROR ,SET EXTERNAL ENABLE,
                                        TSTB    @ADCSR          ;AND INITIATE CONVERSION
                                        BPL     .-2             ;WAIT FOR DONE
                                        CLR     @ADCSR          ;SECOND DATO, WITHOUT READING DATA
                                        TST     @ADCSR          ;CHECK ERROR BIT
                                        BPL     .+4             ;BRANCH IF NOT SET
                                        HLT                     ;ERROR WAS SET

                                TEST18: SCOPE
                                ;DONE SHOULD CAUSE AN INTERRUPT IF INTERRUPT ENABLE IS SET
                                        TST     @ADDBR          ;CLEAR DONE
                                        CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF CSR
                                        BIS     #340,PSR        ;SET PROCESSOR PRIORITY TO LEVEL 7
                                        MOV     #TINTD,@INTVC   ;SETUP RETURN FROM INTERRUPT
                                        MOV     PSR,    2(INTVC)
                                        BIC     #340,PSR        ;SET PROCESSOR PRIORITY TO 0
                                        MOV     #1,@ADCSR       ;CLEAR ERROR AND START CONVERSION
                                        MOVB    #100,@ADCSR     ;SET INTERRUPT ENABLE
                                        TSTB    @ADCSR          ;CHECK DONE
                                        BPL     .-2
                                        HLT                     ;DONE DIDN'T CAUSE AN INTERRUPT
                                        MOV     2(INTVC),PSR    ;RESTORE PROCESSOR STATUS
                                        BR      .+4             ;SKIP NEXT INSTRUCTION
                                TINTD:  CMP     (6)+,(6)+       ;RESTORE SP
                                        CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
                                        MOV     #132,@INTVC     ;CHANGE INTERRUPT POINTER TO
                                        CLR     @#132           ;CAUSE HALT

                                TEST19: SCOPE
                                ;ERROR SHOULD CAUSE AN INTERRUPT IF INTERRUPT ENABLE IS SET
                                        MOV     #TINTE, @INTVC  ;SETUP INTERRUPT RETURN
                                        BIS     #340,PSR
                                        MOV     PSR,    2(INTVC)
                                        CLRB    @ADCSR          ;PREVENT INTERRUPT FROM OCCURRING
                                        BIC     #340,PSR        ;SET PROCESSOR TO LOWEST PRIORITY
                                        TST     @ADDBR          ;CLEAR DONE
                                        MOV     #100100, @ADCSR ;START CONVERSION, ENABLE INTERRUPTS, SET ERROR
                                        NOP                     ;CLOCK IN INTERRUPT
                                        BIS     #340,   PSR     ;RAISE PROCESSOR PRIORITY
                                        HLT                     ;NO INTERRUPT OCCURRED
                                        BR      .+4
                                TINTE:  CMP     (6)+,(6)+       ;RESTORE SP
                                        CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
                                        MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
```

```
  999  002152  005037  000132                CLR     @#132           ;TO CAUSE A HALT
 1000  002156  105711                        TSTB    @ADCSR          ;WAIT FOR DONE
 1001  002160  100376                        BPL     .-2
 1002
 1003  002162  104400                TEST20: SCOPE
 1004                                ;NO INTERRUPT SHOULD OCCUR IF PROCESSOR IS AT LEVEL 7 (BIT 2=0)
 1005  002164  052767  000340  175604        BIS     #340,   PSR     ;SET PROCESSOR PRIORITY TO LEVEL 7
 1006  002172  012714  002220                MOV     #TINT7A, @INTVC ;SETUP RETURN
 1007  002176  016764  175574  000002        MOV     PSR,    2(INTVC)
 1008  002204  105011                        CLRB    @ADCSR          ;MAKE SURE THAT CONVERSION WILL OCCUR
 1009  002206  012711  100100                MOV     #100100,@ADCSR  ;ENABLE INTERRUPTS, SET ERROR, AND START CONVERSION
 1010  002212  000240                        NOP                     ;CLOCK INTERRUPT
 1011  002214  105011                        CLRB    @ADCSR          ;DISABLE INTERRUPTS
 1012  002216  000402                        BR      .+6             ;CONTINUE IF NO INTERRUPT
 1013  002220  104000                TINT7A: HLT                     ;INTERRUPT OCCURRED WITH PROCESSOR AT LEVEL 7
 1014  002222  022626                        CMP     (6)+,(6)+       ;RESTORE STACK POINTER
 1015  002224  105711                        TSTB    @ADCSR          ;WAIT FOR DONE
 1016  002226  100376                        BPL     .-2
 1017  002230  012714  000132                MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
 1018  002234  005037  000132                CLR     @#132           ;TO CAUSE A HALT
 1019                                        .MACR   INT     STATUS,N,TESTNO
 1020                                ;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 'N' (BIT 2=0)
 1021                                TESTNO: SCOPE
 1022                                        BIS     #340,   PSR     ;SET PROCESSOR PRIORITY TO LEVEL 7
 1023                                        MOV     #TINT'N'A,@INTVC        ;SETUP RETURN
 1024                                        MOV     PSR,2(INTVC)
 1025                                        MOV     #1,     @ADCSR  ;INITIALIZE CSR
 1026                                        TSTB    @ADCSR          ;WAIT FOR DONE
 1027                                        BPL     .-2
 1028                                        MOV     #100,   @ADCSR  ;ENABLE INTERRUPTS
 1029                                        MOV     #STATUS,PSR     ;SET PROCESSOR PRIORITY TO LEVEL 'N'
 1030                                        MOV     2(INTVC),PSR    ;SET PROCESSOR BACK TO LEVEL 7
 1031                                        HLT                     ;NO INTERRUPT OCCURRED
 1032                                        BR      .+4
 1033                                TINT'N'A: CMP   (SP)+,(SP)+     ;RESTORE STACK POINTER
 1034                                        CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
 1035                                        MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN ADDRESS
 1036                                        CLR     @#132           ;TO CAUSE A HALT
 1037                                        .ENDM
 1038
 1039  002240                                INT     300,6,TEST21
 1040
 1041  002332                                INT     240,5,TEST22
 1042
 1043  002424                                INT     200,4,TEST23
 1044
 1045  002516                                INT     140,3,TEST24
 1046
 1047  002610                                INT     100,2,TEST25
 1048
 1049  002702                                INT     040,1,TEST26
 1050
 1051                                ;FIND THE LEVEL AT WHICH BIT 2=1 CAUSES AN INTERRUPT
 1052                                ;PRINT OUT A MESSAGE TELLING WHERE THE INTERRUPT OCCURRED
 1053                                ;UNLESS IT WAS AT LEVEL 5 (THE STANDARD LEVEL)
 1054                                ;MAKE SURE THAT IT ALWAYS OCCURS AT THE SAME LEVEL
```

```
                                        ;FIRST SEE THAT NO INTERRUPT OCCURS WITH PROCESSOR
                                        ;AT LEVEL 7 PRIORITY
003774  104400                  TEST27: SCOPE                   ;NO INTERRUPT SHOULD OCCUR WITH PROCESSOR AT LEVEL 7
003776  052767  000340  174772          BIS     #340,PSR        ;SET PROCESSOR TO LEVEL 7
004004  005777  175574                  TST     @ADDBR          ;CLEAR DONE
004010  105011                          CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF STATUS REGISTER
004012  005011                          CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
004014  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
004016  100376                          BPL     .-2
004020  012714  004042                  MOV     #TINT7B,@INTVC  ;SETUP RETURN
004024  016764  174746  000002          MOV     PSR,2(INTVC)    ;SETUP RETURN PROCESSOR STATUS
004032  112711  000104                  MOVB    #104,@ADCSR     ;SET INTERRUPT ENABLE AND BIT 2
004036  000240                          NOP                     ;CLOCK INTERRUPT
004040  000402                          BR      .+6             ;BRANCH IF NO INTERRUPT
004042  104000                  TINT7B: HLT                     ;INTERRUPT OCCURRED WITH PROCESSOR AT LEVEL 7
004044  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP
004046  105011                          CLRB    @ADCSR          ;DISABLE INTERRUPTS
004050  012714  000132                  MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
004054  005037  000132                  CLR     @#132           ;TO CAUSE A HALT

                                        ;MAKE SURE AN INTERRUPT CAN OCCUR WITH PROCESSOR PRIORITY AT LEVEL 0
004060  104400                  TEST28: SCOPE
004062  052767  000340  174706          BIS     #340,PSR        ;SET PROCESSOR TO LEVEL 7
004070  005777  175510                  TST     @ADDBR          ;CLEAR DONE
004074  105011                          CLRB    @ADCSR          ;CLEAR EXTERNAL ENABLE
004076  005011                          CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
004100  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
004102  100376                          BPL     .-2
004104  012714  004146                  MOV     #TINTX,@INTVC   ;SET UP RETURN
004110  016764  174662  000002          MOV     PSR,2(INTVC)    ;SETUP RETURN PROCESSOR STATUS
004116  042767  000340  174652          BIC     #340,PSR        ;SET PROCESSOR TO LEVEL 0
004124  112711  000104                  MOVB    #104,@ADCSR     ;SET INTERRUPT ENABLE AND BIT 2
004130  000240                          NOP                     ;CLOCK IN INTERRUPT
004132  105011                          CLRB    @ADCSR          ;DISABLE INTERRUPTS
004134  052767  000340  174634          BIS     #340,PSR        ;RESTORE PROCESSOR PRIORITY TO 7
004142  104000                          HLT                     ;NO INTERRUPT OCCURRED
004144  000402                          BR      .+6             ;SKIP NEXT 2 INSTRUCTIONS
004146  105011                  TINTX:  CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE AFTER INTERRUPT
004150  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
004152  012714  000132                  MOV     #132,   @INTVC  ;CHANGE INTERRUPT RETURN
004156  005037  000132                  CLR     @#132           ;TO CAUSE A HALT

                                        .MACR   INT2    TSTNO,PRP,STTS,INTLVL,NXTST
                                        ;TEST FOR AN INTERRUPT ON LEVEL 'INTLVL' WITH BIT2 SET
                                TSTNO:  SCOPE
                                        BIS     #340,PSR        ;SET PROCESSOR PRIORITY TO 7
                                        TST     @ADDBR          ;CLEAR DONE
                                        CLRB    @ADCSR          ;CLEAR EXTERNAL ENABLE
                                        CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
                                        TSTB    @ADCSR          ;WAIT FOR DONE
                                        BPL     .-2
                                        MOV     #TINT'PRP'B,@INTVC      ;SETUP RETURN
                                        MOV     PSR,2(INTVC)    ;SETUP RETURN PROCESSOR STATUS
                                        MOV     #STTS,PSR       ;SET PROCESSOR TO LEVEL 'PRP' PRIORITY
                                        MOVB    #104,@ADCSR     ;SET INTERRUPT ENABLE AND BIT 2
                                        NOP                     ;CLOCK INTERRUPT
```

```
                                        CLRB    @ADCSR          ;DISABLE INTERRUPTS
                                        MOV     2(INTVC),PSR    ;RESTORE PROCESSOR STATUS
                                        MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
                                        CLR     @#132           ;TO CAUSE A HALT
                                        BR      NXTST           ;GO TO NEXT TEST

                                TINT'PRP'B:     CLRB    @ADCSR          ;DISABLE FURTHER INTERRUPTS
                                        MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
                                        CLR     @#132           ;TO CAUSE A HALT
                                        CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
                                        TST     INTFLG          ;CHECK FOR PREVIOUS FLAG
                                        BMI     SET'INTLVL              ;BRANCH IF FLAG SET
                                        MOV     #10000'INTLVL,INTFLG    ;SET FLAG AND LEVEL
                                        MOV     #MSG2,%2        ;SETUP FOR PRINTOUT
                                        JSR     %7,TOUT         ;PRINT MESSAGE 'BIT2 SET CAUSES AN
                                        MOV     #INTLVL,%2              ;INTERRUPT AT LEVEL'
                                        JSR     %7,PRSHRT       ;PRINT LEVEL NUMBER
                                        JSR     %7,CRLF         ;CARRIAGE RETURN, LINEFEED
                                        BR      NXTST           ;GO TO NEXT TEST
                                SET'INTLVL:     CMP     INTFLG,#10000'INTLVL    ;CHECK PREVIOUS LEVEL
                                        BPL     NXTST           ;GO TO NEXT TEST IF INTERRUPT PREVIOUSLY AT AN EQUAL
                                                                ;OR HIGHER LEVEL
                                        HLT                     ;INTERRUPT OCCURRED PREVIOUSLY ONLY AT A LOWER LEVEL
                                        .ENDM

003162                                  INT2    TEST29,6,300,7,TEST30

003346                                  INT2    TEST30,5,240,6,TEST31

                                ;TEST FOR AN INTERRUPT ON LEVEL 5 WITH BIT2 SET
003532  104400                  TEST31: SCOPE
003534  052767  000340  174234          BIS     #340,PSR        ;SET PROCESSOR PRIORITY TO  7
003542  005777  175036                  TST     @ADDBR          ;CLEAR DONE
003546  105011                          CLRB    @ADCSR          ;CLEAR EXTERNAL ENABLE
003550  005011                          CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
003552  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
003554  100376                          BPL     .-2
003556  012714  003626                  MOV     #TINT4B,@INTVC  ;SETUP RETURN
003562  016764  174210  000002          MOV     PSR,2(INTVC)    ;SETUP RETURN PROCESSOR STATUS
003570  012767  000200  174200          MOV     #200,PSR        ;SET PROCESSOR TO LEVEL 4 PRIORITY
003576  112711  000104                  MOVB    #104,@ADCSR     ;SET INTERRUPT ENABLE AND BIT 2
003602  000240                          NOP                     ;CLOCK INTERRUPT
003604  105011                          CLRB    @ADCSR          ;DISABLE INTERRUPTS
003606  016467  000002  174162          MOV     2(INTVC),PSR    ;RESTORE PROCESSOR STATUS
003614  012714  000132                  MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
003620  005037  000132                  CLR     @#132           ;TO CAUSE A HALT
003624  000422                          BR      TEST32          ;GO TO NEXT TEST
003626  105011                  TINT4B: CLRB    @ADCSR          ;DISABLE FURTHER INTERRUPTS
003630  012714  000132                  MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
003634  005037  000132                  CLR     @#132           ;TO CAUSE A HALT
003640  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
003642  005767  174752                  TST     INTFLG          ;CHECK FOR PREVIOUS FLAG
003646  100404                          BMI     SET5            ;BRANCH IF FLAG SET
003650  012767  100005  174742          MOV     #100005,INTFLG  ;SET FLAG AND LEVEL
003656  000405                          BR      TEST32          ;GO TO NEXT TEST
003660  026727  174734  100005  SET5:   CMP     INTFLG,#100005  ;CHECK PREVIOUS LEVEL
```

```
1167  003666  100001                        BPL     TEST32          ;GO TO NEXT TEST IF INTERRUPT PREVIOUSLY  OCCURRED AT
1168                                                                  ;AN EQUAL OR HIGHER LEVEL
1169  003670  104000                        HLT                     ;INTERRUPT OCCURRED PREVIOUSLY ONLY AT A LOWER LEVEL
1170
1171  003672                                INT2    TEST32,3,140,4,TEST36
1172
1173                                ;NO INTERRUPT SHOULD OCCUR IF INTERRUPT ENABLE IS SET AND ERROR AND DONE ARE
1174                                ;BOTH CLEAR
1175  004056  104400                TEST36: SCOPE
1176  004060  005777  174520               TST     @ADDBR          ;CLEAR DONE TO PREVENT ERROR
1177  004064  105011                       CLRB    @ADCSR          ;CLEAR EXTERNAL ENABLE
1178  004066  005011                       CLR     @ADCSR          ;INITIALIZE STATUS REGISTER
1179  004070  105711                       TSTB    @ADCSR          ;WAIT FOR DONE
1180  004072  100376                       BPL     .-2
1181  004074  005777  174504               TST     @ADDBR          ;CLEAR DONE
1182  004100  012714  004144               MOV     #NINT,@INTVC    ;SETUP RETURN IN CASE
1183  004104  052767  000340  173664       BIS     #340,PSR
1184  004112  016764  173660  000002       MOV     PSR,2(INTVC)
1185  004120  042767  000340  173650       BIC     #340,PSR        ;SET PROCESSOR TO LOWEST PRIORITY
1186  004126  112711  000100               MOVB    #100,@ADCSR     ;ENABLE INTERRUPTS
1187  004132  000240                       NOP                     ;WINDOW TO ALLOW INTERRUPTS
1188  004134  052767  000340  173634       BIS     #340,PSR        ;SET PROCESSOR TO HIGHEST PRIORITY
1189  004142  000402                       BR      .+6             ;SKIP NEXT 2 INSTRUCTIONS IF NO INTERRUPT
1190  004144  104000                NINT:  HLT                     ;INTERRUPT OCCURRED
1191  004146  022626                       CMP     (SP)+,(SP)+     ;RESTORE STACK POINTER
1192  004150  012714  000132               MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
1193  004154  005037  000132               CLR     @#132           ;TO CAUSE A HALT
1194  004160  105011                       CLRB    @ADCSR          ;DISABLE INTERRUPTS
1195
1196                                ;CHECK FOR REPEATABILITY OF READING DBR AT EACH GAIN
1197  004162  104400                TEST37: SCOPE
1198  004164  005777  174414               TST     @ADDBR          ;CLEAR DONE
1199  004170  105011                       CLRB    @ADCSR          ;CLEAR LOW BYTE
1200  004172  016767  174424  001726       MOV     INITCH,SELECT   ;LOAD INITIAL CHANNEL
1201  004200  004767  000040               JSR     %7,REPTST       ;CHECK REPEATABILITY
1202  004204  052767  000010  001714       BIS     #10,SELECT      ;SET GAIN TO X2
1203  004212  004767  000026               JSR     %7,REPTST       ;CHECK REPEATABILITY
1204  004216  052767  000020  001702       BIS     #20,SELECT      ;SET GAIN TO X8
1205  004224  004767  000014               JSR     %7,REPTST       ;CHECK REPEATABILITY
1206  004230  042767  000010  001670       BIC     #10,SELECT      ;SET GAIN TO X4
1207  004236  004767  000002               JSR     %7,REPTST       ;CHECK REPEATABILITY
1208  004242  000421                       BR      TEST38          ;GO TO NEXT TEST
1209  004244  012767  000100  174340 REPTST: MOV    #100,COUNT      ;SET UP COUNTER
1210  004252  016711  001650               MOV     SELECT,@ADCSR   ;CONVERT AT DESIRED GAIN
1211  004256  105711                       TSTB    @ADCSR          ;WAIT FOR DONE
1212  004260  100376                       BPL     .-2
1213  004262  017700  174316               MOV     @ADDBR,R0       ;LOAD DATA INTO REGISTER 0
1214  004266  027700  174312        RELOOP: CMP    @ADDBR,R0       ;REREAD DATA AND COMPARE
1215  004272  001401                       BEQ     .+4             ;CONTINUE IF BOTH ARE THE SAME
1216  004274  104000                       HLT                     ;2 READINGS OF DBR WEREN'T THE SAME
1217  004276  005367  174310               DEC     COUNT           ;COUNT DOWN
1218  004302  001371                       BNE     RELOOP          ;LOOP
1219  004304  000207                       RTS     %7              ;RETURN
1220
1221                                ;CHECK THE SIGN OPTION IF SW8 IS SET (FOR POSITIVE VOLTAGE)
1222  004306  104400                TEST38: SCOPE
```

```
1223   004310   032767   000400   173252          BIT     #400,SR          ;CHECK SW8
1224   004316   001475                            BEQ     TEST40           ;SKIP TEST IF NOT SET
1225   004320   012702   014304                   MOV     #MSG8,R2         ;"INPUT A POSITIVE VOLTAGE TO THE
1226   004324   004767   007464                   JSR     %7,TOUT          ;INITIAL CHANNEL"
1227   004330   012702   014366                   MOV     #MSG9,R2         ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
1228   004334   004767   007454                   JSR     %7,TOUT
1229   004340   000000                            HALT                     ;WAIT FOR CORRECT VOLTAGE
1230   004342   005067   007274                   CLR     ICOUNT           ;RUN TEST ONLY ONCE
1231   004346   012767   000100   174236          MOV     #100,COUNT       ;INITIALIZE COUNTER
1232   004354   005777   174224                   TST     @ADDBR           ;CLEAR DONE
1233   004360   105011                            CLRB    @ADCSR           ;CLEAR LOW BYTE
1234   004362   016711   174234          LOOP38:  MOV     INITCH,@ADCSR    ;START CONVERSION
1235   004366   105711                            TSTB    @ADCSR           ;WAIT FOR DONE
1236   004370   100376                            BPL     .-2
1237   004372   032777   176000   174204          BIT     #176000,@ADDBR   ;CHECK DATA AND CLEAR DONE
1238   004400   001402                            BEQ     .+6              ;CONTINUE IF BITS 10-15 ALL ZERO
1239   004402   104000                            HLT                      ;HIGH BITS NOT ALL ZERO
1240   004404   000403                            BR      TEST39           ;GO TO NEXT TEST AFTER FAILURE
1241   004406   005367   174200                   DEC     COUNT            ;COUNT DOWN
1242   004412   001363                            BNE     LOOP38           ;LOOP
1243
1244                                    ;CHECK THE SIGN OPTION FOR NEGATIVE VOLTAGE, IF SW8 WAS SET
1245   004414   104400                  TEST39:  SCOPE
1246   004416   012702   014433                   MOV     #MSG10,R2        ;"INPUT A NEGATIVE VOLTAGE TO THE
1247   004422   004767   007366                   JSR     %7,TOUT          ;INITIAL CHANNEL"
1248   004426   012702   014366                   MOV     #MSG9,R2         ;"THEN HIT 'CONTINUE' ON THE CONSOLE"
1249   004432   004767   007356                   JSR     %7,TOUT
1250   004436   000000                            HALT                     ;WAIT FOR NEGATIVE VOLTAGE
1251   004440   012767   000100   174144          MOV     #100,COUNT       ;SETUP COUNTER
1252   004446   005777   174132                   TST     @ADDBR           ;CLEAR DONE
1253   004452   105011                            CLRB    @ADCSR           ;INITIALIZE LOW BYTE
1254   004454   016711   174142          LOOP39:  MOV     INITCH, @ADCSR   ;START CONVERSION
1255   004460   105711                            TSTB    @ADCSR           ;WAIT FOR DONE
1256   004462   100376                            BPL     .-2
1257   004464   017702   174114                   MOV     @ADDBR,R2        ;LOAD DATA
1258   004470   005102                            COM     R2               ;COMPLEMENT DATA
1259   004472   032702   176000                   BIT     #176000,R2       ;CHECK BITS 10-15
1260   004476   001402                            BEQ     .+6              ;CONTINUE IF ALL WERE SET
1261   004500   104000                            HLT                      ;BITS 10-15 NOT ALL SET TO ONE IN DBR
1262   004502   000403                            BR      TEST40
1263   004504   005367   174102                   DEC     COUNT            ;COUNT DOWN
1264   004510   001361                            BNE     LOOP39           ;LOOP
1265
1266                                    ;CHECK THE GAIN BY TAKING 128 SAMPLES ON INITIAL CHANNEL AT EACH GAIN SETTING
1267                                    ;THE TOTAL FOR EACH CHANNEL IS DIVIDED BY 128 AND COMPARED WITH THE
1268                                    ;AVERAGE VALUE AT THE HIGHEST GAIN
1269                                    ;VALUES ARE TESTED FOR + OR - 1 1/4 LSB ACCURACY
1270                                    ;NOTE THAT THIS TEST IS NOT SCOPE LOOPABLE IF ERRORS OCCUR
1271   004512   104400                  TEST40:  SCOPE
1272   004514   012767   000100   007120          MOV     #100,   ICOUNT   ;PARTIALLY RESTORE ICOUNT
1273   004522   005067   001400                   CLR     SELECT           ;SETUP GAIN TO X1
1274   004526   056767   174070   001372          BIS     INITCH,SELECT    ;SET CHANNEL NUMBER
1275   004534   032767   002000   173026          BIT     #2000,SR         ;CHECK BIT 10 OF SWITCH REGISTER
1276   004542   001450                            BEQ     AVG40            ;BRANCH TO TAKE AVERAGES IF NOT SET
1277   004544   005777   174034                   TST     @ADDBR           ;CLEAR DONE
1278   004550   105011                            CLRB    @ADCSR           ;INITIALIZE LOW BYTE
```

```
1279  004552  016711  001350              MOV     SELECT,@ADCSR   ;START CONVERSION ON DESIRED CHANNEL
1280  004556  105711                      TSTB    @ADCSR          ;WAIT FOR DONE
1281  004560  100376                      BPL     .-2
1282  004562  017767  174016  001170      MOV     @ADDBR,GAIN1    ;STORE VALUE
1283  004570  004767  001502              JSR     %7,DBRCK        ;CHECK FOR BITS 10-15 ALL EQUAL
1284  004574  052711  000010              BIS     #10,@ADCSR      ;SET GAIN TO X2 AND START CONVERSION
1285  004600  105711                      TSTB    @ADCSR          ;WAIT FOR DONE
1286  004602  100376                      BPL     .-2
1287  004604  017767  173774  001150      MOV     @ADDBR,GAIN2    ;STORE VALUE
1288  004612  004767  001460              JSR     %7,DBRCK        ;CHECK FOR BITS 10-15 ALL EQUAL
1289  004616  052711  000020              BIS     #20,@ADCSR      ;SET GAIN TO X8 AND START CONVERSION
1290  004622  105711                      TSTB    @ADCSR          ;WAIT FOR DONE
1291  004624  100376                      BPL     .-2
1292  004626  017767  173752  001132      MOV     @ADDBR,GAIN8    ;STORE VALUE
1293  004634  004767  001436              JSR     %7,DBRCK        ;CHECK FOR BITS 10-15 ALL EQUAL
1294  004640  042711  000010              BIC     #10,@ADCSR      ;SET GAIN TO X4 AND START CONVERSION
1295  004644  105711                      TSTB    @ADCSR          ;WAIT FOR DONE
1296  004646  100376                      BPL     .-2
1297  004650  017767  173730  001106      MOV     @ADDBR,GAIN4    ;STORE VALUE
1298  004656  004767  001414              JSR     %7,DBRCK        ;CHECK FOR BITS 10-15 ALL EQUAL
1299  004662  000450                      BR      CK40
1300
1301  004664  004767  001120      AVG40:  JSR     %7,CONVRT       ;CONVERT 128 TIMES AT GAIN OF 1
1302  004670  012767  005760  001072      MOV     #GAIN1,STORE    ;LOAD POINTER
1303  004676  004767  001226              JSR     %7,CONAV        ;AVERAGE THE TOTAL AND STORE IN GAIN1
1304  004702  052767  000010  001216      BIS     #10,SELECT      ;CHANGE GAIN TO X2
1305  004710  004767  001074              JSR     %7,CONVRT       ;CONVERT 128 TIMES AT GAIN OF 2
1306  004714  012767  005762  001046      MOV     #GAIN2,STORE    ;LOAD POINTER
1307  004722  004767  001202              JSR     %7,CONAV        ;AVERAGE THE TOTAL AND STORE IN GAIN2
1308  004726  042767  000010  001172      BIC     #10,SELECT      ;CHANGE GAIN TO X4
1309  004734  052767  000020  001164      BIS     #20,SELECT
1310  004742  004767  001042              JSR     %7,CONVRT       ;CONVERT 128 TIMES AT GAIN OF 4
1311  004746  012767  005764  001014      MOV     #GAIN4,STORE    ;LOAD POINTER
1312  004754  004767  001150              JSR     %7,CONAV        ;AVERAGE THE TOTAL AND STORE IN GAIN4
1313  004760  052767  000010  001140      BIS     #10,SELECT      ;CHANGE GAIN TO X8
1314  004766  004767  001015              JSR     %7,CONVRT       ;CONVERT 128 TIMES AT GAIN OF 8
1315  004772  012767  005766  000770      MOV     #GAIN8,STORE    ;LOAD POINTER
1316  005000  004767  001124              JSR     %7,CONAV        ;AVERAGE THE TOTAL AND STORE IN GAIN8
1317
1318  005004  016703  000756      CK40:   MOV     GAIN8,R3        ;LOAD AVERAGE AT GAIN OF 8 INTO REGISTER 3
1319  005010  062703  000003              ADD     #3,R3           ;CALCULATE HIGHEST ALLOWABLE VALUE AT GAIN OF 4
1320  005014  006203                      ASR     R3              ;(DIVIDE BY 2 AND TRUNCATE)
1321  005016  010367  000750              MOV     R3,HGAIN4       ;STORE MAX ALLOWABLE AT GAIN OF 4
1322  005022  016703  000740              MOV     GAIN8,R3        ;LOAD AVERAGE AT GAIN OF 8
1323  005026  162703  000003              SUB     #3,R3           ;CALCULATE LOWEST ALLOWABLE VALUE AT GAIN OF 4
1324  005032  006203                      ASR     R3              ;(DIVIDE BY 2)
1325  005034  005503                      ADC     R3              ;(ROUND UP)
1326  005036  010367  000732              MOV     R3,LGAIN4       ;STORE VALUE
1327  005042  016703  000720              MOV     GAIN8,R3        ;LOAD AVERAGE AT GAIN OF 8
1328  005046  062703  000006              ADD     #6,R3           ;CALCULATE HIGHEST ALLOWABLE VALUE AT GAIN OF 2
1329  005052  006203                      ASR     R3              ;(DIVIDE BY 4 AND TRUNCATE)
1330  005054  006203                      ASR     R3
1331  005056  010367  000714              MOV     R3,HGAIN2       ;STORE VALUE
1332  005062  016703  000700              MOV     GAIN8,R3        ;LOAD AVERAGE AT GAIN OF 8
1333  005066  162703  000006              SUB     #6,R3           ;CALCULATE MINIMUM ALLOWABLE VALUE AT GAIN OF 2
1334  005072  006203                      ASR     R3              ;(DIVIDE BY 2)
```

```
1335  005074  005503                          ADC     R3              ;(ROUND UP)
1336  005076  006203                          ASR     R3              ;(DIVIDE BY 2)
1337  005100  005503                          ADC     R3              ;(ROUND UP)
1338  005102  010367  000672                  MOV     R3,LGAIN2       ;STORE VALUE
1339  005106  016703  000654                  MOV     GAIN8,R3        ;LOAD VALUE AT GAIN OF 8
1340  005112  062703  000013                  ADD     #13,R3          ;CALCULATE MAXIMUM ALLOWABLE VALUE AT GAIN OF 1
1341  005116  006203                          ASR     R3              ;(DIVIDE BY 8 AND TRUNCATE)
1342  005120  006203                          ASR     R3
1343  005122  006203                          ASR     R3
1344  005124  010367  000652                  MOV     R3,HGAIN1       ;STORE VALUE
1345  005130  016703  000632                  MOV     GAIN8,R3        ;LOAD VALUE AT GAIN OF 8
1346  005134  162703  000013                  SUB     #13,R3          ;CALCULATE MINIMUM ALLOWABLE VALUE AT GAIN OF 1
1347  005140  006203                          ASR     R3              ;(DIVIDE BY 2)
1348  005142  005503                          ADC     R3              ;(ROUND UP)
1349  005144  006203                          ASR     R3              ;(DIVIDE BY 2)
1350  005146  005503                          ADC     R3              ;(ROUND UP)
1351  005150  006203                          ASR     R3              ;(DIVIDE BY 2)
1352  005152  005503                          ADC     R3              ;(ROUND UP)
1353  005154  010367  000624                  MOV     R3,LGAIN1       ;STORE VALUE
1354  005160  026767  000606  000576  CMP40:  CMP     HGAIN4,GAIN4    ;COMPARE MAX VALUE TO ACTUAL VALUE AT GAIN OF 4
1355  005166  100525                          BMI     GMSG            ;GAIN4 TOO HIGH
1356  005170  026767  000570  000576          CMP     GAIN4,LGAIN4    ;COMPARE MIN VALUE TO ACTUAL AT GAIN OF 4
1357  005176  100521                          BMI     GMSG            ;GAIN4 TOO LOW
1358  005200  026767  000572  000554          CMP     HGAIN2,GAIN2    ;COMPARE MAX VALUE TO ACTUAL AT GAIN OF 2
1359  005206  100515                          BMI     GMSG            ;GAIN2 TOO HIGH
1360  005210  026767  000546  000562          CMP     GAIN2,LGAIN2    ;COMPARE MIN VALUE TO ACTUAL AT GAIN OF 2
1361  005216  100511                          BMI     GMSG            ;GAIN2 TOO LOW
1362  005220  026767  000556  000532          CMP     HGAIN1,GAIN1    ;COMPARE MAX VALUE TO ACTUAL AT GAIN OF 1
1363  005226  100505                          BMI     GMSG            ;GAIN1 TOO HIGH
1364  005230  026767  000524  000546          CMP     GAIN1,LGAIN1    ;COMPARE MIN VALUE TO ACTUAL AT GAIN OF 1
1365  005236  100501                          BMI     GMSG            ;GAIN1 TOO LOW
1366
1367  005240  104400                  TEST41: SCOPE
1368                                  ;INIT PULSE SHOULD INITIALIZE THE INTERNAL "READ" FLIP-FLOP OFF
1369                                  ;IF THIS FAILS, THE VERY FIRST CONVERSION WILL SET ERROR
1370  005242  012767  000002  006372          MOV     #2,ICOUNT       ;RUN TEST TWICE ONLY
1371  005250  105777  173340                  TSTB    @TCSR           ;WAIT FOR TTY READY TO AVOID
1372  005254  100375                          BPL     .-4             ;CLOBBERING ANY OUTPUT
1373  005256  000005                          RESET                   ;SEND OUT INIT PULSE
1374  005260  005211                          INC     @ADCSR          ;START CONVERSION
1375  005262  005711                          TST     @ADCSR          ;CHECK ERROR
1376  005264  100001                          BPL     .+4             ;BRANCH IF NOT SET
1377  005266  104000                          HLT                     ;INTERNAL FLIP-FLOP INCORRECTLY INITIALIZED
1378  005270  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
1379  005272  100376                          BPL     .-2
1380
1381  005274  104400                  TEST42: SCOPE
1382                                  ;MAKE SURE THAT WITH BIT 0=0, INTERRUPTS DON'T OCCUR AT BOTH LEVELS AT  ONCE
1383  005276  012767  004000  006336          MOV     #4000,ICOUNT
1384  005304  105777  173274                  TSTB    @ADDBR          ;CLEAR DONE
1385  005310  105011                          CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF CSR
1386  005312  052767  000340  172456          BIS     #340,PSR        ;SET PROCESSOR TO LEVEL 7
1387  005320  012714  005366                  MOV     #TINT2,@INTVC   ;SETUP RETURN FROM INTERRUPT
1388  005324  016764  172446  000002          MOV     PSR,2(INTVC)
1389  005332  042767  000340  172436          BIC     #340,PSR        ;SET PROCESSOR PRIORITY TO LEVEL 0
1390  005340  012711  000001                  MOV     #1,@ADCSR       ;CLEAR ERROR AND START CONVERSION
```

```
1391   005344   112711   000100                    MOVB    #100,@ADCSR     ;SET INTERRUPT ENABLE
1392   005350   105711                             TSTB    @ADCSR          ;CHECK DONE
1393   005352   100376                             BPL     .-2             ;BRANCH TO RESTORE ORDER
1394   005354   104000                             HLT                     ;DONE DIDN'T CAUSE AN INTERRUPT
1395   005356   016467   000002   172412           MOV     2(INTVC),PSR    ;RESTORE PROCESSOR STATUS
1396   005364   000414                             BR      TINT3+4
1397   005366   022626                    TINT2:   CMP     (6)+,(6)+       ;RESTORE STACK POINTER
1398   005370   012714   005412                    MOV     #TINT3,@INTVC   ;SET UP RETURN FOR ERROR INTERRUPT
1399   005374   005067   172376                    CLR     PSR             ;SET PROCESSOR TO LEVEL 0
1400   005400   000240                             NOP                     ;ALLOW INTERRUPT
1401   005402   016467   000002   172366           MOV     2(INTVC),PSR    ;RESTORE PROCESSOR PRIORITY
1402   005410   000402                             BR      TINT3+4         ;BRANCH AROUND HALT
1403   005412   104000                    TINT3:   HLT
1404   005414   022626                             CMP     (6)+,(6)+       ;RESTORE STACK POINTER
1405   005416   105011                             CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
1406   005420   012714   000132                    MOV     #132,@INTVC     ;CHANGE INTERRUPT POINTER TO
1407   005424   005037   000132                    CLR     @#132           ;CAUSE A HALT
1408   005430   104400                             SCOPE
1409
1410   005432   004767   000672                    JSR     %7,BELL         ;RING BELL
1411   005436   000167   173274                    JMP     RESTRT+4        ;START INSTRUCTION TESTS OVER
1412
1413
1414   005442   032767   020000   172120  GMSG:    BIT     #20000,SR       ;CHECK SW13
1415   005450   001136                             BNE     GCNT            ;BRANCH IF SET TO INHIBIT PRINTOUT
1416   005452   012702   014516                    MOV     #MSG11,R2       ;"DIFFERENTIAL GAIN NOT LINEAR WITHIN
1417   005456   004767   006332                    JSR     %7,TOUT         ;+ OR -"
1418   005462   012702   014733                    MOV     #MSG20,R2       ;"1 1/4 LSB"
1419   005466   004767   006322                    JSR     %7,TOUT
1420   005472   012702   014616                    MOV     #MSG14,R2       ;OUTPUT HEADING
1421   005476   004767   006312                    JSR     %7,TOUT
1422   005502   012702   014665                    MOV     #MSG15,R2       ;"MEASURED"
1423   005506   004767   006302                    JSR     %7,TOUT
1424   005512   016702   000250                    MOV     GAIN8,R2        ;OUTPUT VALUE AT GAIN OF 8
1425   005516   004767   005650                    JSR     %7,PROCT
1426   005522   004767   000464                    JSR     %7,SPACE
1427   005526   016702   000232                    MOV     GAIN4,R2        ;OUTPUT VALUE AT GAIN OF 4
1428   005532   004767   005634                    JSR     %7,PROCT
1429   005536   004767   000450                    JSR     %7,SPACE
1430   005542   016702   000214                    MOV     GAIN2,R2        ;OUTPUT VALUE AT GAIN OF 2
1431   005546   004767   005620                    JSR     %7,PROCT
1432   005552   004767   000434                    JSR     %7,SPACE
1433   005556   016702   000176                    MOV     GAIN1,R2        ;OUTPUT VALUE AT GAIN OF 1
1434   005562   004767   005604                    JSR     %7,PROCT
1435   005566   004767   000420                    JSR     %7,SPACE
1436   005572   012702   014701                    MOV     #MSG16,R2       ;"MAX OK"
1437   005576   004767   006212                    JSR     %7,TOUT
1438   005602   016702   000160                    MOV     GAIN8,R2        ;OUTPUT VALUE AT GAIN OF 8
1439   005606   004767   005560                    JSR     %7,PROCT
1440   005612   004767   000374                    JSR     %7,SPACE
1441   005616   016702   000150                    MOV     HGAIN4,R2       ;OUTPUT MAX ALLOWABLE AT GAIN OF 4
1442   005622   004767   005544                    JSR     %7,PROCT
1443   005626   004767   000360                    JSR     %7,SPACE
1444   005632   016702   000140                    MOV     HGAIN2,R2       ;OUTPUT MAX ALLOWABLE AT GAIN OF 2
1445   005636   004767   005530                    JSR     %7,PROCT
1446   005642   004767   000344                    JSR     %7,SPACE
```

```
1447   005646  016702  000130                  MOV     HGAIN1,R2       ;OUTPUT MAX ALLOWABLE AT GAIN OF 1
1448   005652  004767  005514                  JSR     %7,PROCT
1449   005656  004767  000330                  JSR     %7,SPACE
1450   005662  012702  014716                  MOV     #MSG17,R2       ;"MIN OK"
1451   005666  004767  006122                  JSR     %7,TOUT
1452   005672  016702  000070                  MOV     GAIN8,R2        ;OUTPUT VALUE AT GAIN OF 8
1453   005676  004767  005470                  JSR     %7,PROCT
1454   005702  004767  000304                  JSR     %7,SPACE
1455   005706  016702  000062                  MOV     LGAIN4,R2       ;OUTPUT MIN ALLOWABLE AT GAIN OF 4
1456   005712  004767  005454                  JSR     %7,PROCT
1457   005716  004767  000270                  JSR     %7,SPACE
1458   005722  016702  000052                  MOV     LGAIN2,R2       ;OUTPUT MIN ALLOWABLE AT GAIN OF 2
1459   005726  004767  005440                  JSR     %7,PROCT
1460   005732  004767  000254                  JSR     %7,SPACE
1461   005736  016702  000042                  MOV     LGAIN1,R2       ;OUTPUT MIN ALLOWABLE AT GAIN OF 1
1462   005742  004767  005424                  JSR     %7,PROCT
1463   005746  012767  005242  005672  GCNT:   MOV     #TEST41+2,RETURN ;SETUP SCOPE LOOP RETURN POINTER
1464   005754  000167  177262                  JMP     TEST41+2        ;JUMP TO NEXT TEST
1465
1466   005760  000000          GAIN1:  0
1467   005762  000000          GAIN2:  0
1468   005764  000000          GAIN4:  0
1469   005766  000000          GAIN8:  0
1470   005770  000000          STORE:  0
1471   005772  000000          HGAIN4: 0
1472   005774  000000          LGAIN4: 0
1473   005776  000000          HGAIN2: 0
1474   006000  000000          LGAIN2: 0
1475   006002  000000          HGAIN1: 0
1476   006004  000000          LGAIN1: 0
1477   006006  000000          LSBFLG: 0
1478
1479                           ;CONVERT 128 TIMES AT GAIN AND CHANNEL STORED IN SELECT
1480                           ;STORE THE SUMS IN SUM1 THRU SUM4
1481   006010  005777  172570  CONVRT: TST     @ADDBR          ;CLEAR DONE
1482   006014  105011                  CLRB    @ADCSR          ;INITIALIZE LOW BYTE
1483   006016  005003                  CLR     R3              ;WILL HOLD POINTER FOR SUMMING CONVERSION RESULTS
1484   006020  005067  000072                  CLR     SUM1            ;CLEAR SUMMING LOCATIONS
1485   006024  005067  000070                  CLR     SUM2
1486   006030  005067  000066                  CLR     SUM3
1487   006034  005067  000064                  CLR     SUM4
1488   006040  012703  006116                  MOV     #SUM1,R3        ;LOAD POINTER FOR SUMMING FIRST 32 CONVERSIONS
1489   006044  012767  000040  172540  LOOP2:  MOV     #40,COUNT       ;LOAD COUNT TO ADD 32 CONVERSIONS
1490   006052  016711  000050          LOOP1:  MOV     SELECT,@ADCSR   ;START CONVERSION
1491   006056  105711                  TSTB    @ADCSR          ;WAIT FOR DONE
1492   006060  100376                  BPL     .-2
1493   006062  067713  172516                  ADD     @ADDBR,@R3      ;ADD THIS CONVERSION TO PREVIOUS SUM
1494   006066  004767  000204                  JSR     %7,DBRCK        ;CHECK FOR BITS 10-15 ALL EQUAL
1495   006072  005367  172514                  DEC     COUNT           ;COUNT DOWN
1496   006076  001365                  BNE     LOOP1           ;LOOP UNTIL 32 (DECIMAL) CONVERSIONS HAVE BEEN ADDED
1497   006100  020327  006124                  CMP     R3,#SUM4        ;CHECK FOR DONE
1498   006104  001001                  BNE     .+4             ;CONTINUE IF NOT DONE
1499   006106  000207                  RTS     %7              ;RETURN WHEN 128 CONVERSIONS HAVE BEEN SUMMED
1500   006110  062703  000002                  ADD     #2,R3           ;MOVE POINTER FOR SUMMING NEXT 32 CONVERSIONS
1501   006114  000753                  BR      LOOP2           ;BRANCH TO DO NEXT SET OF CONVERSIONS
1502   006116  000000          SUM1:   0
```

```
1503  006120  000000                  SUM2:   0
1504  006122  000000                  SUM3:   0
1505  006124  000000                  SUM4:   0
1506  006126  000000                  SELECT: 0                       ;CONTAINS GAIN AND CHANNEL TO BE TESTED
1507
1508                                  ;AVERAGE THE 4 SUMS OF 32 CONVERSIONS EACH CONTAINED IN SUM1 THRU SUM4
1509                                  ;STORE THE RESULT IN THE LOCATION WHOSE ADDRESS IS CONTAINED IN STORE
1510  006130  012705  006116          CONAV:  MOV     #SUM1,R5        ;SET UP POINTER
1511  006134  012503                          MOV     (R5)+,R3        ;LOAD SUM AND MOVE POINTER
1512  006136  005077  177626                  CLR     @STORE
1513  006142  006203                  B.LOOP: ASR     R3              ;DIVIDE BY 32
1514  006144  006203                          ASR     R3
1515  006146  006203                          ASR     R3
1516  006150  006203                          ASR     R3
1517  006152  006203                          ASR     R3
1518  006154  005503                          ADC     R3              ;ROUND OFF
1519  006156  060377  177606                  ADD     R3,@STORE       ;ADD TO PREVIOUS SUM OF AVERAGES
1520  006162  020527  006126                  CMP     R5,#SUM4+2      ;CHECK FOR FINISHING UP
1521  006166  001402                          BEQ     B.AVG           ;FINISH UP
1522  006170  012503                          MOV     (R5)+,R3        ;PICK UP NEXT SUM
1523  006172  000763                          BR      B.LOOP          ;CONTINUE
1524  006174  006277  177570          B.AVG:  ASR     @STORE          ;DIVIDE SUM OF THE FOUR AVERAGES
1525  006200  006277  177564                  ASR     @STORE          ;BY FOUR
1526  006204  005577  177560                  ADC     @STORE          ;ROUND OFF
1527  006210  000207                          RTS     %7              ;RETURN
1528
1529                                  ;SUBROUTINE TO ISSUE N SPACES
1530                                  ;N IS ONE PLUS VALUE CONTAINED IN SPACEX
1531                                  ;SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
1532                                  ;SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE
1533  006212  105777  172376          SPACE:  TSTB    @TCSR           ;WAIT FOR TTY READY
1534  006216  100375                          BPL     .-4
1535  006220  012777  000240  172370          MOV     #240,@TDBR      ;OUTPUT A SPACE
1536  006226  005367  000010                  DEC     SPACEX          ;DECREMENT COUNT
1537  006232  100367                          BPL     SPACE           ;LOOP IF NOT DONE
1538  006234  005067  000002                  CLR     SPACEX          ;RESET COUNT TO ZERO
1539  006240  000207                          RTS     %7              ;RETURN
1540  006242  000000                  SPACEX: 0
1541
1542                                  ;SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED
1543  006244  105777  172344          CRLF:   TSTB    @TCSR           ;WAIT FOR TTY READY
1544  006250  100375                          BPL     .-4
1545  006252  012777  000215  172336          MOV     #215,@TDBR      ;OUTPUT CARRIAGE RETURN
1546  006260  105777  172330                  TSTB    @TCSR           ;WAIT FOR TTY READY
1547  006264  100375                          BPL     .-4
1548  006266  012777  000212  172322          MOV     #212,@TDBR      ;OUTPUT LINEFEED
1549  006274  000207                          RTS     %7              ;RETURN
1550
1551                                  ;MAKE CERTAIN THAT BITS 10-15 OF ADDBR ARE ALL EQUAL
1552  006276  032777  176000  172300  DBRCK:  BIT     #176000,@ADDBR  ;TEST BITS 10-15 OF DBR
1553  006304  001001                          BNE     .+4             ;BRANCH IF ANY ARE SET
1554  006306  000207                          RTS     %7              ;RETURN IF ALL ZERO
1555  006310  017702  172270                  MOV     @ADDBR,R2       ;MOVE DATA TO REGISTER 2
1556  006314  005102                          COM     R2              ;COMPLEMENT
1557  006316  032702  176000                  BIT     #176000,R2      ;CHECK BITS 10-15
1558  006322  001401                          BEQ     .+4             ;BRANCH IF ALL ZERO
```

```
1559  006324  104000                          HLT                     ;BITS 10-15 NOT ALL EQUAL
1560  006326  000207                          RTS     %7
1561
1562                                  ;BELL ON PASS COMPLETE
1563  006330  105777  172260         BELL:   TSTB    @TCSR
1564  006334  100375                          BPL     .-4
1565  006336  012777  000207  172252        MOV     #207,@TDBR
1566  006344  105737  177564                TSTB    @#177564
1567  006350  100375                          BPL     .-4
1568  006352  005037  177566                CLR     @#177566
1569  006356  012767  000756  005262        MOV     #TEST0+2,RETURN
1570  006364  012767  004000  005250        MOV     #4000,  ICOUNT
1571  006372  000207                          RTS     %7
1572
1573                                  ;SUBROUTINE TO CHECK FOR END OF LINE
1574                                  ;R2 CONTAINS NUMBER OF CHARACTERS JUST OUTPUT
1575                                  ;R3 CONTAINS NUMBER TO BE OUTPUT BEFORE NEXT CALL
1576                                  ;L.CNT CONTAINS CURRENT TOTAL LENGTH OF LINE
1577  006374  060267  000024         FULLCK: ADD     R2,L.CNT        ;ADD CHARACTERS OUTPUT TO CURRENT SUM
1578  006400  066703  000020                 ADD     L.CNT,R3        ;ADD TOTAL TO PREDICTED
1579  006404  020327  000110                 CMP     R3,#110         ;CHECK FOR TOO LONG
1580  006410  100404                          BMI     .+12            ;BRANCH IF OK
1581  006412  004767  177626                JSR     %7,CRLF         ;IF TOO LONG, CARRIAGE RETURN
1582  006416  005067  000002                CLR     L.CNT           ;INITIALIZE COUNT
1583  006422  000207                          RTS     %7              ;RETURN
1584  006424  000000                 L.CNT:  0
1585
1586                                  ;DISPLAY LOOP OF CONVERTED VALUE
1587                                  ;MUX CHANNEL MAY BE SET IN BITS 00-05
1588                                  ;GAIN MAY BE SET IN BITS 06-07
1589                                  ;IF SW14=1, 64 CONVERSIONS ARE MADE AT FULL SPEED AND STORED
1590                                  ;THEN THE RESULTS ARE DISPLAYED ONE AT A TIME FOR ABOUT 40MS. EACH
1591                                  ;IF SW15=1, THE PROGRAM HALTS AFTER EACH CONVERSION WITH THE
1592                                  ;RESULTS DISPLAYED.  TRACE TRAPPING IS NOT USED IN THIS ROUTINE
1593                                  ;IF THE CPU IS AN 11/05, THE ROUTINE WILL HALT.
1594  006426  004767  172200         DSPLAY: JSR     %7,SETUP
1595  006432  012737  000002  000006         MOV     #RTI,@#6
1596  006440  012700  177777                MOV     #-1,R0
1597  006444  005037  177700                CLR     @#177700
1598  006450  005037  000006                CLR     @#6
1599  006454  005700                         TST     R0
1600  006456  001002                         BNE     DSPLY
1601  006460  000000                         HALT                    ;USE SA 250 WITH SW10 SET IF 11/05
1602  006462  000776                         BR      .-2
1603  006464  105011                 DSPLY:  CLRB    @ADCSR          ;INITIALIZE LOW BYTE
1604  006466  005777  172112                TST     @ADDBR          ;CLEAR DONE
1605  006472  032767  040000  171070        BIT     #40000,SR       ;CHECK SW14
1606  006500  001044                         BNE     FAST            ;BRANCH IF SET
1607  006502  004767  000032                JSR     %7,SET
1608  006506  016711  177414                MOV     SELECT,@ADCSR
1609  006512  105711                         TSTB    @ADCSR          ;WAIT FOR DONE
1610  006514  100376                         BPL     .-2
1611  006516  017700  172062                MOV     @ADDBR,R0       ;LOAD RESULT IN R0
1612  006522  000005                         RESET                   ;DISPLAY VALUE
1613  006524  000005                         RESET                   ;DISPLAY VALUE
1614  006526  005767  171036                TST     SR              ;CHECK SW15 FOR HALTING ON SINGLE CONVERSIONS
```

```
1615  006532  100001                          BPL     .+4             ;BRANCH IF NOT SET
1616  006534  000000                          HALT                    ;HALT AND DISPLAY VALUE
1617  006536  000752                          BR      DSPLY
1618  006540  116767  171024  177361  SET:    MOVB    SR,SELECT+1     ;SET UP CHANNEL
1619  006546  042767  140377  177352          BIC     #140377,SELECT
1620  006554  032767  000100  171006          BIT     #100,SR
1621  006562  001403                          BEQ     DSCNT1
1622  006564  052767  000010  177334          BIS     #10,SELECT
1623  006572  032767  000200  170770  DSCNT1: BIT     #200,SR
1624  006600  001403                          BEQ     DSCNT2
1625  006602  052767  000020  177316          BIS     #20,SELECT
1626  006610  000207                  DSCNT2: RTS     %7
1627  006612  012703  006674          FAST:   MOV     #FTABLE,R3      ;LOAD POINTER
1628  006616  004767  177716                  JSR     %7,SET
1629  006622  016711  177300          F.LOOP: MOV     SELECT,@ADCSR   ;START CONVERSION
1630  006626  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
1631  006630  100376                          BPL     .-2
1632  006632  017723  171746                  MOV     @ADDBR,(R3)+    ;STORE DATA
1633  006636  020327  007074                  CMP     R3,#FTABLE+200  ;CHECK FOR DONE
1634  006642  001367                          BNE     F.LOOP          ;IF NOT DONE, BRANCH
1635  006644  012703  006674          FSHOW:  MOV     #FTABLE,R3      ;LOAD POINTER
1636  006650  012300                          MOV     (R3)+,R0        ;LOAD R0
1637  006652  000005                          RESET                   ;DISPLAY DATA
1638  006654  000005                          RESET
1639  006656  000005                          RESET
1640  006660  000005                          RESET
1641  006662  020327  007074                  CMP     R3,#FTABLE+200  ;CHECK FOR END OF TABLE
1642  006666  001370                          BNE     FSHOW+4         ;CONTINUE
1643  006670  000167  177570                  JMP     DSPLY           ;RETURN TO BASIC LOOP
1644  006674  000000          FTABLE: 0
1645           007074                  .=.+176
1646
1647
1648                                  ;EXTERNAL CONVERSION TESTS TO CHECK CONVERSION SPEED AND CERTAIN ERROR
1649                                  ;CASES INVOLVING EXTERNAL CONVERSIONS.  ON EVERY TENTH CONVERSION EXTEST
1650                                  ;PRINTS OUT "10".  IF SW5 =1, EXTERNAL ENABLE IS NOT SET SO NO PRINTOUTS
1651                                  ;SHOULD OCCUR.  IF SW6 =1, INTERNAL CONVERSIONS ARE RUN CONTINOUSLY AND A
1652                                  ;PRINTOUT OCCURS WHEN ERROR IS SET.  SW7 PREVENTS DONE FROM BEING CLEARED,
1653                                  ;AND SHOULD THUS CAUSE AN ERROR PRINTOUT ON THE SECOND EXTERNAL CONVERSION.
1654                                  ;THESE SWITCHES ARE MUTUALLY EXCLUSIVE AND ARE SCANNED IN ORDER.
1655                                  ;TRACE TRAPPING IS NOT USED IN THIS ROUTINE
1656  007074  004767  171532  EXTEST: JSR     %7,SETUP
1657  007100  004767  177140          JSR     %7,CRLF         ;START ON A NEW LINE
1658  007104  005067  177314          CLR     L.CNT           ;INITIALIZE COUNTER FOR LENGTH OF LINE
1659  007110  005777  171470  EXTST1: TST     @ADDBR          ;CLEAR DONE
1660  007114  105011                  CLRB    @ADCSR          ;CLEAR LOW BYTE OF STATUS REGISTER
1661                                                          ;TO CLEAR EXTERNAL ENABLE (TO CLEAR
1662                                                          ;ERROR IF SET)
1663  007116  005011                  CLR     @ADCSR          ;START CONVERSION BY LOADING HIGH BYTE
1664  007120  105711                  TSTB    @ADCSR          ;WAIT FOR DONE - NOTE THAT AN EXTERNAL
1665  007122  100376                  BPL     .-2             ;CONVERSION MIGHT HAVE STARTED IT (SO ERROR
1666                                                          ;MAY STILL BE SET)
1667  007124  005777  171454          TST     @ADDBR          ;CLEAR DONE
1668  007130  005011                  CLR     @ADCSR          ;MAKE SURE ERROR IS CLEARED
1669  007132  105711                  TSTB    @ADCSR          ;(ALSO STARTS ANOTHER CONVERSION)
1670  007134  100376                  BPL     .-2
```

```
1671  007136  005777  171442                    TST     @ADDBR          ;CLEAR DONE
1672  007142  105711                            TSTB    @ADCSR          ;MAKE SURE THAT DONE CLEARED
1673  007144  100001                            BPL     .+4
1674  007146  104000                            HLT                     ;DONE WAS SET
1675  007150  032767  000040  170412  EXCK:     BIT     #40,SR          ;CHECK SW5
1676  007156  001034                            BNE     EXSW5           ;BRANCH IF SET
1677  007160  032767  000100  170402            BIT     #100,SR         ;CHECK SW6
1678  007166  001077                            BNE     EXINT           ;BRANCH IF SET
1679  007170  105767  170374                    TSTB    SR              ;CHECK SW7
1680  007174  100514                            BMI     EXSIT           ;BRANCH IF SET
1681  007176  012767  000012  171406            MOV     #12,COUNT       ;INITIALIZE COUNT
1682  007204  152711  000002                    BISB    #2,@ADCSR       ;SET EXTERNAL ENABLE
1683  007210  105711                  EXLOOP:   TSTB    @ADCSR          ;CHECK DONE
1684  007212  100005                            BPL     EXCNT1          ;BRANCH IF NOT SET
1685  007214  005367  171372                    DEC     COUNT           ;COUNT DOWN
1686  007220  001435                            BEQ     EXPRNT          ;BRANCH IF TENTH CONVERSION
1687  007222  005777  171356                    TST     @ADDBR          ;CLEAR DONE
1688  007226  011167  000214          EXCNT1:   MOV     @ADCSR,SAVCSR   ;SAVE CONTENTS OF CSR
1689  007232  005767  000210                    TST     SAVCSR          ;CHECK ERROR
1690  007236  100364                            BPL     EXLOOP          ;LOOP IF NOT SET
1691  007240  104000                            HLT                     ;ERROR WAS SET
1692  007242  005267  000156                    INC     ERFLG1          ;SET FLAG TO INDICATE ERROR PRINTOUT OCCURRED
1693  007246  000720                            BR      EXTST1          ;RESTART
1694  007250  005067  171336          EXSW5:    CLR     COUNT           ;INITIALIZE COUNTER
1695  007254  105711                  EXLP1:    TSTB    @ADCSR          ;CHECK DONE
1696  007256  100004                            BPL     EXCNT2          ;CONTINUE IF NOT SET
1697  007260  104000                            HLT                     ;DONE WAS SET
1698  007262  005267  000136                    INC     ERFLG1          ;SET FLAG TO INDICATE ERROR PRINTOUT OCCURRED
1699  007266  000710                            BR      EXTST1          ;RESTART AFTER FAILURE
1700  007270  005711                  EXCNT2:   TST     @ADCSR          ;CHECK ERROR
1701  007272  100004                            BPL     EXCNT3          ;CONTINUE IF NOT SET
1702  007274  104000                            HLT                     ;ERROR WAS SET
1703  007276  005267  000122                    INC     ERFLG1          ;SET FLAG TO INDICATE ERROR PRINTOUT OCCURRED
1704  007302  000702                            BR      EXTST1          ;RESTART AFTER FAILURE
1705  007304  105267  171302          EXCNT3:   INCB    COUNT           ;COUNT
1706  007310  001361                            BNE     EXLP1           ;LOOP IF NOT ZERO
1707  007312  000676                            BR      EXTST1          ;IF ZERO RETURN TO MAIN TO CHECK SWITCHES
1708  007314  005767  000104          EXPRNT:   TST     ERFLG1          ;CHECK FLAG FOR ERROR PRINTOUT
1709  007320  001406                            BEQ     EXCNT4          ;BRANCH IF NOT SET
1710  007322  004767  176716                    JSR     %7,CRLF         ;OUTPUT CARRIAGE RETURN, LINE FEED
1711  007326  005067  177072                    CLR     L.CNT           ;INITIALIZE COUNTER FOR NEW LINE
1712  007332  005067  000066                    CLR     ERFLG1          ;CLEAR FLAG
1713  007336  012702  000010          EXCNT4:   MOV     #10,R2          ;PRINTOUT "10"
1714  007342  004767  003774                    JSR     %7,PRSHRT
1715  007346  004767  176640                    JSR     %7,SPACE        ;OUTPUT SPACE
1716  007352  012702  000003                    MOV     #3,R2           ;LOAD NUMBER OF CHARACTERS OUTPUT IN R2
1717  007356  010203                            MOV     R2,R3           ;LOAD NUMBER BEFORE NEXT CHECK IN R3
1718  007360  004767  177010                    JSR     %7,FULLCK       ;CHECK FOR END OF LINE
1719  007364  000651                            BR      EXTST1          ;RETURN TO MAIN
1720  007366  012711  000003          EXINT:    MOV     #3,@ADCSR       ;START CONVERSION AND SET EXTERNAL ENABLE
1721  007372  032711  100200                    BIT     #100200,@ADCSR  ;CHECK DONE AND ERROR
1722  007376  001775                            BEQ     .-4             ;LOOP IF NOT SET
1723  007400  005777  171200                    TST     @ADDBR          ;CLEAR DONE IF SET
1724  007404  005711                            TST     @ADCSR          ;CHECK ERROR FLAG
1725  007406  100402                            BMI     .+6             ;BRANCH IF SET
1726  007410  005211                            INC     @ADCSR          ;START CONVERSION
```

```
1727  007412  000767                          BR      EXINT+4         ;CONTINUE
1728  007414  104000                          HLT                     ;ERROR WAS SET
1729  007416  005267  000002                  INC     ERFLG1          ;SET FLAG TO INDICATE ERROR PRINTOUT OCCURRED
1730  007422  000632                          BR      EXTST1          ;RESTART
1731  007424  000000              ERFLG1:     0                       ;USED TO DETERMINE NEED FOR STARTING A NEW LINE
1732  007426  152711  000002      EXSIT:      BISB    #2,@ADCSR       ;SET EXTERNAL ENABLE
1733  007432  005711                          TST     @ADCSR          ;WAIT FOR ERROR
1734  007434  100376                          BPL     .-2
1735  007436  104000                          HLT                     ;ERROR WAS SET
1736  007440  005267  177760                  INC     ERFLG1          ;SET FLAG TO INDICATE ERROR PRINTOUT OCCURRED
1737  007444  000621                          BR      EXTST1          ;RESTART
1738  007446  000000              SAVCSR:     0
1739
1740                                          ;FULL SPEED EXTERNAL CONVERSION ROUTINE
1741                                          ;THIS TEST SETS UP THE CONVERTER, ENABLES EXTERNAL CONVERSIONS, AND
1742                                          ;WAITS FOR DONE.  ON DONE, IT CLEARS DONE BY READING THE DBR, AND CHECKS
1743                                          ;ERROR.  IF ERROR IS NOT SET, IT LOOPS WAITING FOR DONE AS BEFORE.
1744                                          ;IF ERROR IS SET IT PRINTS AN ERROR ADDRESS.  THUS, IF A VARIABLE RATE EXTERNAL
1745                                          ;TRIGGER SIGNAL IS APPLIED, THE MAX THROUGHOUT OF THE CONVERTER CAN BE
1746                                          ;CHECKED.  TRACE TRAPPING IS NOT USED IN THIS ROUTINE.
1747  007450  004767  171156      EXFAST:     JSR     %7,     SETUP
1748  007454  005777  171124      EXFLOP:     TST     @ADDBR          ;CLEAR DONE
1749  007460  105011                          CLRB    @ADCSR          ;INITIALIZE LOW BYTE
1750  007462  005011                          CLR     @ADCSR          ;START CONVERSION
1751  007464  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
1752  007466  100376                          BPL     .-2
1753  007470  005777  171110                  TST     @ADDBR          ;CLEAR DONE
1754  007474  012711  000002                  MOV     #2,     @ADCSR  ;INITIALIZE EXTERNAL CONVERSION
1755  007500  105711              EXCONT:     TSTB    @ADCSR          ;WAIT FOR DONE
1756  007502  100376                          BPL     .-2
1757  007504  005711                          TST     @ADCSR          ;CHECK ERROR
1758  007506  100002                          BPL     .+6             ;BRANCH IF NOT SET
1759  007510  104000                          HLT                     ;ERROR WAS SET
1760  007512  000760                          BR      EXFLOP          ;RESTART
1761  007514  005777  171064                  TST     @ADDBR          ;CLEAR DONE
1762  007520  000767                          BR      EXCONT          ;LOOP
1763  007522  000000              EXCNT:      0
1764
1765                                          ;CONVERSION PRINT LOOP
1766                                                  ;SW0-SW5-SET MUX CHANNEL ON WHICH YOU WISH TO CONVERT
1767                                                  ;SW6-SW7-SET DESIRED GAIN
1768                                                  ;SW10=0 -128 CONVERSIONS ARE AVERAGED AND THE RESULT IS PRINTED OUT
1769                                                  ;       =1 -EACH INDIVIDUAL CONVERSION IS PRINTED OUT
1770  007524  004767  171102      FLY:        JSR     %7,SETUP        ;INITIALIZE REGISTERS
1771  007530  004767  176510                  JSR     %7,CRLF
1772  007534  005067  176664                  CLR     L.CNT           ;INITIALIZE COUNTER FOR LENGTH OF LINE
1773  007540  032767  002000  170022  FLY.3:  BIT     #2000,SR        ;CHECK BIT 10 OF SR
1774  007546  001031                          BNE     FLY.1           ;BRANCH IF SET TO INHIBIT AVERAGING
1775  007550  004767  176764                  JSR     %7,SET          ;SETUP GAIN AND CHANNEL
1776  007554  004767  176230                  JSR     %7,CONVRT       ;CONVERT 128 TIMES
1777  007560  012767  005760  176202          MOV     #GAIN1,STORE    ;SETUP STORAGE ADDRESS FOR RESULT
1778  007566  004767  176336                  JSR     %7,CONAV        ;AVERAGE RESULTS
1779  007572  016702  176162                  MOV     GAIN1,R2        ;LOAD RESULT INTO REGISTER 2
1780  007576  004767  003570          FLY.2:  JSR     %7,PROCT        ;PRINT RESULT IN OCTAL
1781  007602  105777  171006                  TSTB    @TCSR           ;WAIT FOR TTY READY
1782  007606  100375                          BPL     .-4
```

```
 1783  007610  012777  000240  171000          MOV     #240,@TDBR      ;OUTPUT A SPACE
 1784  007616  012702  000007                  MOV     #7,R2           ;LOAD NUMBER OF CHARACTERS OUTPUT IN R2
 1785  007622  010203                          MOV     R2,R3           ;LOAD NUMBER BEFORE NEXT CHECK IN R3
 1786  007624  004767  176544                  JSR     %7,FULLCK       ;CHECK FOR END OF LINE
 1787  007630  000743                          BR      FLY.3           ;LOOP
 1788                                  ;PRINT SINGLE CONVERSION RESULTS
 1789  007632  005777  170746          FLY.1:  TST     @ADDBR          ;CLEAR DONE
 1790  007636  105011                          CLRB    @ADCSR          ;INITIALIZE LOW BYTE OF CSR
 1791  007640  004767  176674                  JSR     %7,SET          ;SET UP GAIN AND CHANNEL
 1792  007644  016711  176256                  MOV     SELECT,@ADCSR   ;START CONVERSION
 1793  007650  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
 1794  007652  100376                          BPL     .-2
 1795  007654  017702  170724                  MOV     @ADDBR,R2       ;LOAD RESULT INTO R2
 1796  007660  000746                          BR      FLY.2           ;PRINT RESULT AND START OVER
 1797
 1798                                  ;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
 1799                                  ;LOAD THE STARTING ADDRESS OF THE TEST
 1800                                  ;YOU WISH TO RUN (THE ADDRESS OF THE TESTXX
 1801                                  ;TAG) AT THE 1ST HALT.  SET SWITCH REGISTER
 1802                                  ;OPTIONS AT THE 2ND HALT.
 1803                                  ;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
 1804  007662  004767  170744          TESTX:  JSR     %7,SETUP
 1805  007666  000000                          HALT                    ;WAIT FOR STARTING ADDRESS
 1806  007670  016767  167674  170730          MOV     SR,RETRNX       ;LOAD STARTING ADDRESS IN RETRNX
 1807  007676  062767  000002  170722          ADD     #2,RETRNX       ;ADD 2 TO POINT TO INSTRUCTION AFTER SCOPE
 1808  007704  000000                          HALT                    ;SET SR OPTIONS
 1809  007706  005067  003732                  CLR     SCOPEF          ;KEEP COUNT AT ZERO
 1810  007712  012767  007724  003726          MOV     #XLOOP,RETURN   ;LOAD SCOPE LOOP RETURN POINTER
 1811  007720  000177  170702                  JMP     @RETRNX         ;JUMP TO TEST
 1812  007724  005067  003714          XLOOP:  CLR     SCOPEF          ;KEEP COUNT AT ZERO
 1813  007730  000177  170672                  JMP     @RETRNX         ;JUMP TO TEST
 1814
 1815                                  ;CHANNEL AND REPEATABILITY TEST (WAS-IS TEST)
 1816                                  ;THE TEST HALTS THREE TIMES AT THE START
 1817                                          ;1ST HALT - LOAD INITIAL CHANNEL (BITS 05-00) AND GAIN (BITS 07-06)
 1818                                          ;2ND HALT - LOAD NUMBER OF CHANNELS TO BE TESTED
 1819                                          ;3RD HALT - SET SR OPTIONS
 1820                                  ;SWITCH REGISTER OPTIONS ARE:
 1821                                          ;SW13=1 OR UP--- INHIBIT PRINTOUT OF ERROR MESSAGES
 1822                                          ;SW10=1 OR UP--- INHIBIT AVERAGING WHEN FORMING INITIAL TABLE
 1823                                          ;SW09=1 OR UP--- INHIBIT + OR - 1 LSB TESTING
 1824                                          ;SW07=1 OR UP--- COMPARE NEW TABLE TO INITIAL TABLE
 1825                                                          ;INSTEAD OF TO OLD TABLE
 1826                                          ;SW06=1 OR UP--- PRINT OUT A COMPARISON OF INITIAL
 1827                                                          ;TABLE AND NEW TABLE
 1828                                          ;SW05=1 OR UP--- TAKE 8 COMPARISONS ON EACH CHANNEL
 1829                                                          ;BEFORE SWITCHING CHANNELS
 1830                                  ;THE TEST STORES AN INITIAL CONVERSION VALUE FOR EACH CHANNEL, WHICH IS
 1831                                  ;EITHER THE AVERAGE OF 128 CONVERSIONS, OR A SINGLE CONVERSION, DEPENDING
 1832                                  ;ON SW10.  THIS TABLE IS CALLED ITABLE (INITIAL TABLE).  THEN A SINGLE
 1833                                  ;CONVERSION IS DONE FOR EACH CHANNEL AND STORED IN OTABLE (OLD TABLE).
 1834                                  ;FROM THEN ON, SETS OF ONE CONVERSION FOR EACH CHANNEL ARE TAKEN AND
 1835                                  ;STORED IN NTABLE (NEW TABLE).  OTABLE AND NTABLE ARE THEN COMPARED FOR
 1836                                  ;+ OR - 1 LSB AGREEMENT (UNLESS SW9 IS SET, IN WHICH CASE THEY MUST
 1837                                  ;AGREE EXACTLY).  ERRORS ARE PRINTED OUT IN A "CHANNEL XX WAS YYYYYY,
 1838                                  ;IS ZZZZZZ "FORMAT.  AFTER THE COMPARISON OTABLE IS UPDATED WITH NTABLE
```

```
1839                                                    ;AND THE PROCESS IS REPEATED.  WHENEVER SW6 IS SET, A COMPARISON
1840                                                    ;OF ITABLE AND NTABLE IS PRINTED OUT, WHICH ALLOWS DRIFT TO BE CHECKED.
1841                                                    ;THE BELL IS RUNG AFTER EVERY 4096 LOOPS THRU THE TEST.
1842
1843  007734  004767  170672          WASIS:  JSR     %7,SETUP        ;INITIALIZE REGISTERS
1844  007740  000000                          HALT                    ;WAIT FOR INITIAL CHANNEL AND GAIN
1845  007742  004767  176572                  JSR     %7,SET          ;STORE INITIAL CHANNEL AND GAIN
1846  007746  016767  176154  170646          MOV     SELECT,INITCH
1847  007754  000000                          HALT                    ;WAIT FOR NUMBER OF CHANNELS
1848  007756  016767  167606  170640          MOV     SR,NUMCH        ;STORE
1849  007764  026727  170634  000101          CMP     NUMCH,#65.      ;PREVENT TOO LARGE A CHANNEL COUNT
1850  007772  100403                          BMI     .+10
1851  007774  012767  000100  170622          MOV     #64.,NUMCH
1852  010002  000000                          HALT                    ;WAIT FOR SR OPTIONS
1853  010004  032767  002000  167556  WSTART: BIT     #2000,SR        ;CHECK SW10
1854  010012  001023                          BNE     WCONT1          ;BRANCH IF SET TO INHIBIT AVERAGING
1855  010014  016767  170604  001172          MOV     NUMCH,WCOUNT    ;LOAD COUNTER
1856  010022  012767  010414  175740          MOV     #ITABLE,STORE   ;LOAD TABLE POINTER
1857  010030  004767  175754          WLOOP1: JSR     %7,CONVRT       ;CONVERT 128 TIMES AT GIVEN CHANNEL AND GAIN
1858  010034  004767  176070                  JSR     %7,CONAV        ;AVERAGE RESULTS
1859  010040  062767  000002  175722          ADD     #2,STORE        ;MOVE TABLE POINTER
1860  010046  105267  176055                  INCB    SELECT+1        ;INCREMENT CHANNEL NUMBER
1861  010052  005367  001136                  DEC     WCOUNT          ;COUNT DOWN
1862  010056  003364                          BGT     WLOOP1          ;LOOP
1863  010060  000404                          BR      WCONT2          ;BRANCH IF ITABLE IS LOADED
1864  010062  012700  010414          WCONT1: MOV     #ITABLE,R0      ;LOAD ITABLE WITH SINGLE CONVERSION
1865  010066  004767  001130                  JSR     %7,LOADTB
1866  010072  012700  010614          WCONT2: MOV     #OTABLE,R0      ;LOAD OTABLE WITH SINGLE CONVERSIONS
1867  010076  004767  001120                  JSR     %7,LOADTB
1868  010102  012767  010000  001110  WAGON:  MOV     #10000,COUNTR   ;INITIALIZE COUNTER
1869  010110  012700  011014          WAGAIN: MOV     #NTABLE,R0      ;LOAD NTABLE WITH SINGLE CONVERSIONS
1870  010114  004767  001102                  JSR     %7,LOADTB
1871  010120  012700  011014                  MOV     #NTABLE,R0      ;LOAD POINTER
1872  010124  032767  000200  167436          BIT     #200,SR         ;CHECK SW7 FOR COMPARING NEW DATA TO
1873  010132  001003                          BNE     .+10            ;INITIAL DATA INSTEAD OF PREVIOUS DATA
1874  010134  012705  010614                  MOV     #OTABLE,R5      ;LOAD POINTER
1875  010140  000402                          BR      .+6
1876  010142  012705  010414                  MOV     #ITABLE,R5      ;LOAD POINTER
1877  010146  005067  001042                  CLR     WCOUNT          ;INITIALIZE COUNTER
1878  010152  032767  000040  167410          BIT     #40,SR          ;CHECK SW5
1879  010160  001452                          BEQ     WLOOP2          ;BRANCH IF NOT SET
1880  010162  005067  001030                  CLR     CONCNT          ;IF SET, RUN 8 COMPARISONS IN A ROW ON EACH CHANNEL
1881  010166  105011                          CLRB    @ADCSR          ;CLEAR GAIN BITS
1882  010170  152711  000002                  BISB    #2,@ADCSR       ;PREVENT CONVERSION ON LOADING INITIAL CHANNEL
1883  010174  105077  170406                  CLRB    @ADCSRB         ;CLEAR CHANNEL BITS
1884  010200  056711  170416                  BIS     INITCH,@ADCSR   ;INITIALIZE CONVERTER
1885  010204  142711  000107                  BICB    #107,@ADCSR     ;INITIALIZE LOW BYTE
1886  010210  011003                  WLOOP4: MOV     @R0,R3          ;MOVE NEW DATA TO R3
1887  010212  161503                          SUB     @R5,R3          ;SUBTRACT OLD DATA
1888  010214  004767  001172                  JSR     %7,WCOMP        ;COMPARE VALUES AND PRINTOUT IF WRONG
1889  010220  005267  000772                  INC     CONCNT          ;COUNT
1890  010224  011060  177600                  MOV     @R0,-200(R0)    ;MOVE DATA TO OTABLE
1891  010230  026727  000762  000010          CMP     CONCNT,#8.      ;CHECK FOR 8 CONVERSIONS DONE ON CURRENT CHANNEL
1892  010236  100415                          BMI     WCONV           ;IF NOT 8, CONTINUE ON SAME CHANNEL
1893  010240  005267  000750                  INC     WCOUNT          ;IF 8, GO TO NEXT CHANNEL
1894  010244  026767  000744  170352          CMP     WCOUNT,NUMCH    ;CHECK FOR ALL CHANNELS DONE
```

```
1895  010252  100032                          BPL     WCONT3          ;IF DONE, RETURN TO MAIN
1896  010254  005067  000736                   CLR     CONCNT          ;CLEAR CONVERSION COUNT
1897  010260  005720                          TST     (R0)+           ;MOVE POINTER TO NEW DATA
1898  010262  005725                          TST     (R5)+           ;MOVE POINTER TO OLD DATA
1899  010264  105277  170316                  INCB    @ADCSRB         ;START CONVERSION BY INCREMENTING CHANNEL NUMBER
1900  010270  000401                          BR      .+4             ;SKIP NEXT INSTRUCTION
1901  010272  105211                  WCONV:  INCB    @ADCSR          ;START CONVERTER
1902  010274  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
1903  010276  100376                          BPL     .-2
1904  010300  017710  170300                  MOV     @ADDBR,@R0      ;LOAD DATA INTO NEW TABLE
1905  010304  000741                          BR      WLOOP4          ;LOOP
1906  010306  011003                  WLOOP2: MOV     @R0,R3          ;LOAD DATA FROM NTABLE
1907  010310  161503                          SUB     @R5,R3          ;SUBTRACT OLD DATA
1908  010312  004767  001074                  JSR     %7,WCOMP        ;COMPARE VALUES AND PRINTOUT IF WRONG
1909  010316  005720                          TST     (R0)+           ;MOVE POINTER TO NEW DATA
1910  010320  005725                          TST     (R5)+           ;MOVE POINTER TO OLD DATA
1911  010322  005267  000666                  INC     WCOUNT          ;COUNT
1912  010326  026767  000662  170270          CMP     WCOUNT,NUMCH    ;CHECK FOR END OF COMPARISONS
1913  010334  100001                          BPL     WCONT3          ;BRANCH IF ALL CHANNELS HAVE BEEN COMPARED
1914  010336  000763                          BR      WLOOP2          ;COMPARE VALUES FOR NEXT CHANNEL
1915  010340  032767  000100  167222  WCONT3: BIT     #100,SR         ;CHECK BIT 6 FOR PRINTOUT OF ITABLE AND NTABLE
1916  010346  001402                          BEQ     .+6             ;BRANCH IF NOT SET
1917  010350  004767  001172                  JSR     %7,WPRINT       ;PRINT OUT COMPARISON OF ITABLE AND NTABLE
1918  010354  012700  010614                  MOV     #OTABLE,R0      ;LOAD POINTER TO MOVE DATA
1919  010360  016020  000200          WLOOP3: MOV     200(R0),(R0)+   ;MOVE NTABLE INTO OTABLE
1920  010364  020027  011014                  CMP     R0,#NTABLE      ;CHECK FOR WHOLE TABLE MOVED
1921  010370  001373                          BNE     WLOOP3          ;LOOP IF NOT DONE
1922  010372  005367  000622                  DEC     COUNTR          ;COUNT DOWN
1923  010376  001402                          BEQ     .+6             ;BRANCH IF 4096TH LOOP
1924  010400  000167  177504                  JMP     WAGAIN          ;CONTINUE WITH TEST IF DONE HERE
1925  010404  004767  175720                  JSR     %7,BELL         ;RING BELL
1926  010410  000167  177466                  JMP     WAGON           ;CONTINUE WAS - IS TEST
1927  010414  000000                  ITABLE: 0
1928          010614                          .=.+176
1929  010614  000000                  OTABLE: 0
1930          011014                          .=.+176
1931  011014  000000                  NTABLE: 0
1932          011214                          .=.+176
1933  011214  000000                  WCOUNT: 0
1934  011216  000000                  CONCNT: 0
1935  011220  000000                  COUNTR: 0
1936
1937                                  ;SUBROUTINE TO LOAD CONVERSION DATA INTO A TABLE USING INTERRUPTS
1938                                  ;THE FIRST ADDRESS OF THE TABLE IS STORED IN R0 BEFORE CALLING
1939  011222  026727  167376  000101  LOADTB: CMP     NUMCH,#65.      ;PREVENT AN ILLEGAL NUMBER OF CHANNELS
1940  011230  100403                          BMI     .+10            ;BRANCH IF LESS THAN 65
1941  011232  012767  000100  167364          MOV     #64.,NUMCH      ;SET NUMBER OF CHANNELS TO 64
1942  011240  016767  167360  000142          MOV     NUMCH,LCOUNT    ;INITIALIZE COUNTER
1943  011246  012714  011330                  MOV     #LTBSRV,@INTVC  ;SETUP INTERRUPT RETURN
1944  011252  052767  000340  166516          BIS     #340,PSR        ;SET PROCESSOR PRIORITY TO LEVEL 7
1945  011260  016764  166512  000002          MOV     PSR,2(INTVC)    ;STORE SAME STATUS FOR SUBROUTINE
1946  011266  005777  167312                  TST     @ADDBR          ;CLEAR DONE
1947  011272  116711  167324                  MOVB    INITCH,@ADCSR   ;SET GAIN
1948  011276  116767  167321  000102          MOVB    INITCH+1,CHOUT  ;SET UP INITIAL CHANNEL
1949  011304  116777  000076  167274          MOVB    CHOUT,@ADCSRB   ;START CONVERSION AND CLEAR ERROR
1950  011312  152711  000100                  BISB    #100,@ADCSR     ;ENABLE INTERRUPTS
```

```
 1951  011316  042767  000340  166452          BIC     #340,PSR        ;SET PROCESSOR PRIORITY TO ZERO
 1952  011324  000777                          BR      .               ;SIT
 1953  011326  000776                          BR      .-2
 1954  011330  005711                  LTBSRV: TST     @ADCSR          ;CHECK ERROR
 1955  011332  100001                          BPL     .+4             ;BRANCH IF NOT SET
 1956  011334  104000                          HLT                     ;ERROR BIT SET
 1957  011336  017720  167242                  MOV     @ADDBR,(R0)+    ;STORE DATA
 1958  011342  005367  000042                  DEC     LCOUNT          ;COUNT
 1959  011346  003406                          BLE     LTBDON          ;BRANCH IF DONE
 1960  011350  005267  000032                  INC     CHOUT           ;INCREMENT CHANNEL NUMBER
 1961  011354  116777  000026  167224          MOVB    CHOUT,@ADCSRB   ;START CONVERSION, CLEAR ERROR IF SET
 1962  011362  000002                          RTI                     ;RETURN FROM INTERRUPT
 1963  011364  012714  000132          LTBDON: MOV     #132,@INTVC     ;CHANGE INTERRUPT RETURN
 1964  011370  005037  000132                  CLR     @#132           ;TO CAUSE A HALT
 1965  011374  105011                          CLRB    @ADCSR          ;CLEAR INTERRUPT ENABLE
 1966  011376  005726                          TST     (SP)+           ;MOVE STACK POINTER
 1967  011400  012667  166372                  MOV     (SP)+,PSR       ;RESTORE PSR
 1968  011404  000207                          RTS     %7              ;RETURN
 1969  011406  000000                  CHOUT:  0
 1970  011410  000000                  LCOUNT: 0
 1971
 1972                                  ;SUBROUTINE TO CHECK COMPARISON OF NTABLE TO OTABLE DATA AND PRINTOUT
 1973                                  ;RESULT OF SUBTRACTION IS IN R3 WHEN CALLED
 1974  011412  032767  001000  166150  WCOMP:  BIT     #1000,SR        ;CHECK FOR INHIBITING + OR - 1 LSB
 1975  011420  001403                          BEQ     .+10            ;BRANCH IF NOT SET
 1976  011422  005703                          TST     R3              ;TEST R3 FOR ZERO IF SET
 1977  011424  001007                          BNE     WMSG            ;IF NOT ZERO, PRINTOUT ERROR MESSAGE
 1978  011426  000207                          RTS     %7              ;RETURN IF OK
 1979  011430  005203                          INC     R3              ;CHECK FOR <-1
 1980  011432  100404                          BMI     WMSG            ;BRANCH TO ERROR MESSAGE IF <-1
 1981  011434  162703  000003                  SUB     #3,R3           ;CHECK FOR >+1
 1982  011440  100001                          BPL     WMSG            ;BRANCH TO ERROR MESSAGE IF >+1
 1983  011442  000207                          RTS     %7              ;RETURN
 1984
 1985
 1986                                  ;PRINTOUT WAS-IS MESSAGE FOR ONE PAIR OF CONVERSIONS
 1987  011444  032767  020000  166116  WMSG:   BIT     #20000,SR       ;CHECK SW13
 1988  011452  001401                          BEQ     .+4             ;BRANCH IF NOT SET
 1989  011454  000207                          RTS     %7              ;INHIBIT PRINTOUT IF SET
 1990  011456  012702  014212                  MOV     #MSG4,R2        ;"CH"
 1991  011462  004767  002326                  JSR     %7,TOUT
 1992  011466  016702  167130                  MOV     INITCH,R2       ;CALCULATE CHANNEL NUMBER
 1993  011472  000302                          SWAB    R2
 1994  011474  042702  177700                  BIC     #177700,R2
 1995  011500  066702  177510                  ADD     WCOUNT,R2
 1996  011504  004767  001632                  JSR     %7,PRSHRT       ;PRINT CHANNEL NUMBER IN OCTAL
 1997  011510  012702  014217                  MOV     #MSG5,R2
 1998  011514  004767  002274                  JSR     %7,TOUT         ;"WAS "
 1999  011520  011502                          MOV     @R5,R2          ;LOAD OLD DATA
 2000  011522  004767  001644                  JSR     %7,PROCT        ;PRINT OUT OLD DATA
 2001  011526  012702  014226                  MOV     #MSG6,R2
 2002  011532  004767  002256                  JSR     %7,TOUT         ;" IS"
 2003  011536  011002                          MOV     @R0,R2          ;LOAD NEW DATA
 2004  011540  004767  001626                  JSR     %7,PROCT        ;PRINT OUT NEW DATA
 2005  011544  000207                          RTS     %7              ;RETURN
 2006
```

```
                                        ;PRINT OUT COMPARISON OF INITIAL TABLE AND NEW TABLE
011546  012702  014235          WPRINT: MOV     #MSG7,R2        ;PRINT HEADING
011552  004767  002236                  JSR     %7,TOUT
011556  012700  010414                  MOV     #ITABLE,R0      ;LOAD POINTER
011562  016705  167034                  MOV     INITCH,R5       ;CALCULATE CHANNEL NUMBER
011566  000305                          SWAB    R5
011570  042705  177700                  BIC     #177700,R5      ;R5 CONTAINS FIRST CHANNEL NUMBER
011574  016767  167024  177412          MOV     NUMCH,WCOUNT    ;LOAD COUNTER
011602  004767  174436          WPLOOP: JSR     %7,CRLF         ;OUTPUT CARRIAGE RETURN AND LINE FEED
011606  010502                          MOV     R5,R2           ;LOAD CHANNEL NUMBER
011610  005205                          INC     R5              ;INCREMENT CHANNEL NUMBER
011612  004767  001554                  JSR     %7,PROCT        ;PRINT OUT CHANNEL NUMBER
011616  012767  000006  174416          MOV     #6,SPACEX       ;SPACE OVER
011624  004767  174362                  JSR     %7,SPACE
011630  011002                          MOV     @R0,R2          ;PRINT OUT INITIAL VALUE
011632  004767  001534                  JSR     %7,PROCT
011636  012767  000006  174376          MOV     #6,SPACEX       ;SPACE OVER
011644  004767  174342                  JSR     %7,SPACE
011650  016002  000400                  MOV     400(R0),R2      ;LOAD NEW VALUE
011654  004767  001512                  JSR     %7,PROCT
011660  005720                          TST     (R0)+           ;MOVE DATA POINTER
011662  005367  177326                  DEC     WCOUNT          ;COUNT DOWN
011666  003345                          BGT     WPLOOP          ;LOOP IF NOT DONE
011670  000207                          RTS     %7              ;RETURN

                                        ;ROUTINE TO TEST REPEATABILITY TO SPECIFICATIONS
                                        ;SW6=1 CAUSES PRINTOUT OF DISTRIBUTION
                                        ;SW7=1 CAUSES DISTRIBUTION TO BE PRINTED ONLY WHEN A FAILURE OCCURS
011672  012706  000600          SIGMA1: MOV     #STACK,%6
011676  004767  166730                  JSR     %7,SETUP
011702  000000                          HALT                    ;SET FIRST CHANNEL TO BE TESTED
011704  116767  165660  166711          MOVB    SR,INITCH+1
011712  042767  140277  166702          BIC     #140277,INITCH  ;CLEAR ERROR AND GAIN BITS
011720  000000                          HALT                    ;SET NUMBER OF CHANNELS TO BE TESTED
011722  016767  165642  166674          MOV     SR,NUMCH        ;STORE NUMBER OF CHANNELS
011730  012767  013030  174032          MOV     #AV,STORE       ;SETUP POINTER FOR STORING AVERAGE VALUE
011736  022767  000100  166660          CMP     #64.,NUMCH
011744  100003                          BPL     SIGMA2
011746  012767  000100  166650          MOV     #64.,NUMCH
011754  005067  001052  SIGMA2: CLR     CHCNT           ;CLEAR CHANNEL COUNT
011760  016767  166636  174140          MOV     INITCH,SELECT   ;SETUP CHANNEL TO START ON
011766  042767  000030  174132  SIGMA3: BIC     #30,SELECT      ;SET INITIAL GAIN TO X4
011774  052767  000020  174124          BIS     #20,SELECT
012002  016767  001016  001016          MOV     SAMSIZ,SAMCNT   ;INITIALIZE SAMPLE SIZE COUNTER
012010  004767  173774                  JSR     %7,CONVRT       ;TAKE 128 CONVERSIONS ON DESIRED CHANNEL
012014  004767  174110                  JSR     %7,CONAV        ;AVERAGE RESULTS AND STORE IN AV
012020  005067  001012                  CLR     OVER
012024  005067  001010                  CLR     UNDER
012030  005067  001006                  CLR     OUT
012034  005067  001004                  CLR     EQUAL
012040  005777  166540  SILOOP: TST     @ADDBR          ;CLEAR DONE
012044  016711  174056                  MOV     SELECT,@ADCSR   ;START CONVERSION
012050  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
012052  100376                          BPL     .-2
012054  017767  166524  000752          MOV     @ADDBR,RESULT   ;STORE RESULT
012062  166767  000742  000744          SUB     AV,RESULT       ;SUBTRACT AVERAGE
```

```
2063  012070  026727  000740  000001          CMP     RESULT,#1       ;CHECK FOR RESULT ONE GREATER THAN AVERAGE
2064  012076  001003                          BNE     SICK1           ;NOPE-TRY AGAIN
2065  012100  005267  000732                  INC     OVER            ;YES, RECORD IT
2066  012104  000417                          BR      SICNT6          ;RESULT HAS BEEN CATEGORIZED
2067  012106  026727  000722  177777  SICK1:  CMP     RESULT,#-1      ;CHECK FOR RESULT ONE LESS THAN AVERAGE
2068  012114  001003                          BNE     SICK2           ;NOPE-GO ON
2069  012116  005267  000716                  INC     UNDER           ;YES, RECORD IT
2070  012122  000410                          BR      SICNT6          ;RESULT HAS BEEN CATEGORIZED
2071  012124  005767  000704          SICK2:  TST     RESULT
2072  012130  001403                          BEQ     SICNT4          ;RESULT EQUALS AVERAGE
2073  012132  005267  000704                  INC     OUT             ;RESULT OUTSIDE LIMITS-COUNT IT
2074  012136  000402                          BR      SICNT6
2075  012140  005267  000700          SICNT4: INC     EQUAL
2076  012144  005367  000656          SICNT6: DEC     SAMCNT          ;10,000 CONVERSIONS COMPLETED?
2077  012150  001333                          BNE     SILOOP          ;NO-CONTINUE
2078  012152  016702  000660                  MOV     OVER,R2         ;CHECK FOR 2ND ALLOWED STATE
2079  012156  166702  000656                  SUB     UNDER,R2
2080  012162  100004                          BPL     SICNT1          ;BRANCH IF AV+1 IS 2ND ALLOWED STATE
2081  012164  066767  000646  000650          ADD     OVER,OUT        ;OUT IS TOTAL OUTSIDE OF 2 STATES
2082  012172  000403                          BR      SICK3           ;BRANCH TO CHECK FOR FAILURE
2083  012174  066767  000640  000640  SICNT1: ADD     UNDER,OUT       ;AV+1 IS 2ND ALLOWED STATE
2084  012202  032767  000100  165360  SICK3:  BIT     #100,SR         ;PRINTOUT IF SET
2085  012210  001402                          BEQ     .+6
2086  012212  004767  000306                  JSR     %7,PRINT4
2087  012216  026767  000620  000622          CMP     OUT,ALLOW4      ;TOO MANY OUT OF RANGE?
2088  012224  100416                          BMI     SIGMA5          ;NO, GO TO GAIN OF 8 TEST
2089  012226  105767  165336                  TSTB    SR              ;YES-CHECK SW 7
2090  012232  100002                          BPL     SICNT7          ;NOT SET-SKIP NEXT
2091  012234  004767  000264                  JSR     %7,PRINT4       ;SET-PRINT DISTRIBUTION
2092  012240  104000          SICNT7: HLT                             ;FAILURE AT GAIN OF 4
2093  012242  012702  013214                  MOV     #MSGI,R2        ;"CH"
2094  012246  004767  001542                  JSR     %7,TOUT
2095  012252  116702  173651                  MOVB    SELECT+1,R2     ;PRINT CHANNEL #
2096  012256  004767  001060                  JSR     %7,PRSHRT
2097  012262  052767  000010  173636  SIGMA5: BIS     #10,SELECT      ;SET GAIN TO X8
2098  012270  005067  000546                  CLR     OUT
2099  012274  005067  000544                  CLR     EQUAL
2100  012300  016767  000520  000520          MOV     SAMSIZ,SAMCNT   ;LOAD COUNTER FOR SIZE OF SAMPLE
2101  012306  004767  173476                  JSR     %7,CONVRT       ;TAKE 128 CONVERSIONS
2102  012312  004767  173612                  JSR     %7,CONAV        ;AVERAGE THE RESULTS
2103  012316  005777  166262          SILOP1: TST     @ADDBR          ;CLEAR DONE
2104  012322  016711  173600                  MOV     SELECT,@ADCSR   ;START CONVERSION
2105  012326  105711                          TSTB    @ADCSR          ;WAIT FOR DONE
2106  012330  100376                          BPL     .-2
2107  012332  017767  166246  000474          MOV     @ADDBR,RESULT   ;LOAD RESULT
2108  012340  166767  000464  000466          SUB     AV,RESULT       ;CHECK FOR BEING WITHIN + OR - 1 LSB
2109  012346  001413                          BEQ     SICNT2
2110  012350  026727  000460  000001          CMP     RESULT,#1
2111  012356  001407                          BEQ     SICNT2
2112  012360  026727  000450  177777          CMP     RESULT,#-1
2113  012366  001403                          BEQ     SICNT2
2114  012370  005267  000446                  INC     OUT             ;IF IT IS NOT, INCREMENT COUNTER
2115  012374  000402                          BR      SICNT5
2116  012376  005267  000442          SICNT2: INC     EQUAL
2117  012402  005367  000420          SICNT5: DEC     SAMCNT
2118  012406  001343                          BNE     SILOP1          ;IF NOT DONE WITH SAMPLE, LOOP
```

```
2119  012410  032767  000100  165152          BIT     #100,SR         ;CHECK FOR PRINT SWITCH
2120  012416  001402                          BEQ     .+6
2121  012420  004767  000260                  JSR     %7,PRINT8
2122  012424  026767  000412  000416          CMP     OUT,ALLOW8      ;TOO MANY OUT OF RANGE?
2123  012432  100416                          BMI     SICNT3          ;NO, GO TEST NEXT CHANNEL
2124  012434  105767  165130                  TSTB    SR              ;YES-CHECK SW 7
2125  012440  100002                          BPL     SICNT8          ;NOT SET-SKIP NEXT
2126  012442  004767  000236                  JSR     %7,PRINT8       ;SET-PRINT DISTRIBUTION
2127  012446  104000                  SICNT8: HLT                     ;FAILURE AT GAIN OF 8
2128  012450  012702  013214                  MOV     #MSGI,R2        ;"CH"
2129  012454  004767  001334                  JSR     %7,TOUT
2130  012460  116702  173443                  MOVB    SELECT+1,R2     ;PRINT CHANNEL #
2131  012464  004767  000652                  JSR     %7,PRSHRT
2132  012470  005267  000336          SICNT3: INC     CHCNT           ;COUNT CHANNELS TESTED
2133  012474  026767  000332  166122          CMP     CHCNT,NUMCH     ;CHECK FOR ALL CHANNELS TESTED
2134  012502  100404                          BMI     SAMNXT          ;IF NOT ALL TESTED, GO TO NEXT CHANNEL
2135  012504  004767  173620                  JSR     %7,BELL         ;RING BELL IF ALL TESTED
2136  012510  000167  177240                  JMP     SIGMA2          ;RUN TEST AGAIN
2137  012514  105267  173407          SAMNXT: INCB    SELECT+1        ;CHANGE CHANNEL NUMBER
2138  012520  000167  177242                  JMP     SIGMA3          ;LOOP
2139
2140  012524  032767  020000  165036  PRINT4: BIT     #20000,SR
2141  012532  001401                          BEQ     .+4
2142  012534  000207                          RTS     %7
2143  012536  012702  013052                  MOV     #MSGA,%2
2144  012542  004767  001246                  JSR     %7,TOUT
2145  012546  116702  173355                  MOVB    SELECT+1,R2
2146  012552  042702  177740                  BIC     #177740,R2
2147  012556  004767  000560                  JSR     %7,PRSHRT
2148  012562  012702  013073                  MOV     #MSGB,R2
2149  012566  004767  001222                  JSR     %7,TOUT
2150  012572  016702  000232                  MOV     AV,R2
2151  012576  004767  000540                  JSR     %7,PRSHRT
2152  012602  012702  013102                  MOV     #MSGC,R2
2153  012606  004767  001202                  JSR     %7,TOUT
2154  012612  016702  000220                  MOV     OVER,R2
2155  012616  004767  000520                  JSR     %7,PRSHRT
2156  012622  012702  013113                  MOV     #MSGD,R2
2157  012626  004767  001162                  JSR     %7,TOUT
2158  012632  016702  000202                  MOV     UNDER,R2
2159  012636  004767  000500                  JSR     %7,PRSHRT
2160  012642  012702  013125                  MOV     #MSGE,R2
2161  012646  004767  001142                  JSR     %7,TOUT
2162  012652  016702  000164                  MOV     OUT,R2
2163  012656  004767  000460                  JSR     %7,PRSHRT
2164  012662  012702  013135                  MOV     #MSGF,R2
2165  012666  004767  001122                  JSR     %7,TOUT
2166  012672  016702  000146                  MOV     EQUAL,R2
2167  012676  004767  000440                  JSR     %7,PRSHRT
2168  012702  000207                          RTS     %7
2169  012704  032767  020000  164656  PRINT8: BIT     #20000,SR
2170  012712  001401                          BEQ     .+4
2171  012714  000207                          RTS     %7
2172  012716  012702  013147                  MOV     #MSGG,R2
2173  012722  004767  001066                  JSR     %7,TOUT
2174  012726  116702  173175                  MOVB    SELECT+1,R2
```

```
2175  012732  042702  177740                  BIC     #177740,R2
2176  012736  004767  000400                  JSR     %7,PRSHRT
2177  012742  012702  013073                  MOV     #MSGB,R2
2178  012746  004767  001042                  JSR     %7,TOUT
2179  012752  016702  000052                  MOV     AV,R2
2180  012756  004767  000360                  JSR     %7,PRSHRT
2181  012762  012702  013170                  MOV     #MSGH,R2
2182  012766  004767  001022                  JSR     %7,TOUT
2183  012772  016702  000046                  MOV     EQUAL,R2
2184  012776  004767  000340                  JSR     %7,PRSHRT
2185  013002  012702  013125                  MOV     #MSGE,R2
2186  013006  004767  001002                  JSR     %7,TOUT
2187  013012  016702  000024                  MOV     OUT,R2
2188  013016  004767  000320                  JSR     %7,PRSHRT
2189  013022  000207                          RTS     %7
2190  013024  023420                  SAMSIZ: 23420                   ;10,000 DECIMAL
2191  013026  000000                  SAMCNT: 0
2192  013030  000000                  AV:     0
2193  013032  000000                  CHCNT:  0
2194  013034  000000                  RESULT: 0
2195  013036  000000                  OVER:   0
2196  013040  000000                  UNDER:  0
2197  013042  000000                  OUT:    0
2198  013044  000000                  EQUAL:  0
2199  013046  000044                  ALLOW4: 44                      ;35+1 DECIMAL
2200  013050  000044                  ALLOW8: 44                      ;35+1 DECIMAL
2201  013052  040057  043500  044501  MSGA:   .ASCII  ;/@@GAIN OF 4,CH /;
2202  013060  020116  043117  032040
2203  013066  041454  020110     057
2204  013073     057  040500  036526  MSGB:   .ASCII  ;/@AV= /;
2205  013100  027440
2206  013102  040057  053117  051105  MSGC:   .ASCII  ;/@OVER= /;
2207  013110  020075     057
2208  013113     057  052500  042116  MSGD:   .ASCII  ;/@UNDER= /;
2209  013120  051105  020075     057
2210  013125     057  047500  052125  MSGE:   .ASCII  ;/@OUT= /;
2211  013132  020075     057
2212  013135     057  042500  052521  MSGF:   .ASCII  ;/@EQUAL= /;
2213  013142  046101  020075     057
2214  013147     057  040100  040507  MSGG:   .ASCII  ;/@@GAIN OF 8,CH /;
2215  013154  047111  047440  020106
2216  013162  026070  044103  027440
2217  013170  040057  044527  044124  MSGH:   .ASCII  ;/@WITHIN 3 STATES= /;
2218  013176  047111  031440  051440
2219  013204  040524  042524  036523
2220  013212  027440
2221  013214  020057  044103  027440  MSGI:   .ASCII  ;/ CH /;
2222
2223                                  ;ENTERED WITH SYSTEM TRAP CALL (HLT)
2224                                  ;PRINT OUT THE ERROR PC+2 AND STATUS REGISTER
2225  013222  036727  164342  020000  PRINT:  BIT     SR,     #20000  ;TEST FOR INHIBIT PRINT OUT
2226  013230  001401                          BEQ     .+4             ;BRANCH TO PRINT
2227  013232  000431                          BR      CK              ;INHIBIT,CHECK FOR HALT
2228  013234  012667  000076                  MOV     (6)+,   SAVPC   ;PC OF FAILING ROUTINE
2229  013240  012667  000074                  MOV     (6)+,   SAVPSR  ;PSR OF ERROR CONDITION
2230  013244  024646                          CMP     -(6),   -(6)    ;RESTORE STACK
```

```
2231  013246  004767  172772                  JSR     %7,     CRLF    ;OUTPUT CARRIAGE RETURN AND LINE FEED
2232  013252  010267  000052                  MOV     %2,     SAVR2   ;SAVE R2
2233  013256  016702  000054                  MOV     SAVPC,  %2      ;LOAD REGISTER WITH FAILING PC+2
2234  013262  004767  000104                  JSR     %7,     PROCT   ;PRINT FAILING PC+2
2235  013266  105777  165322                  TSTB    @TCSR           ;WAIT FOR TTY READY
2236  013272  100375                          BPL     .-4
2237  013274  012777  000240  165314          MOV     #240,   @TDBR   ;OUTPUT A SPACE
2238  013302  016702  000032                  MOV     SAVPSR, %2      ;LOAD PROCESSOR STATUS
2239  013306  004767  000060                  JSR     %7,     PROCT   ;PRINT PROCESSOR STATUS
2240  013312  016702  000012                  MOV     SAVR2,  %2      ;RESTORE REGISTER
2241  013316  005767  164246          CK:     TST     SR              ;CHECK SR FOR HALT SWITCH
2242  013322  100001                          BPL     .+4             ;BRANCH IF NOT SET
2243  013324  000000                          HALT                    ;HALT ON ERROR UP
2244  013326  000002                          RTI                     ;RETURN TO MAIN LINE
2245  013330  000000                  SAVR2:  0
2246  013332  000000                  SAVR3:  0
2247  013334  000000                  SAVR4:  0
2248  013336  000000                  SAVPC:  0
2249  013340  000000                  SAVPSR: 0
2250
2251                                  ;SUBROUTINE TO PRINT OUT OCTAL NUMBER
2252                                  ;PRSHRT DELETES LEADING ZEROS
2253                                  ;PROCT PRINTS OUT 6 OCTAL DIGITS
2254  013342  012767  000001  000206  PRSHRT: MOV     #1,     PRSFLG  ;SET FLAG TO INDICATE SHORT PRINTOUT
2255  013350  005702                          TST     R2              ;CHECK FOR ZERO
2256  013352  001011                          BNE     PROCT+4         ;BRANCH IF NOT ZERO
2257  013354  105777  165234                  TSTB    @TCSR           ;WAIT FOR TTY READY
2258  013360  100375                          BPL     .-4
2259  013362  012777  000260  165226          MOV     #260,@TDBR      ;OUTPUT A SINGLE ZERO
2260  013370  000207                          RTS     %7              ;RETURN
2261  013372  005067  000160          PROCT:  CLR     PRSFLG          ;CLEAR FLAG TO INDICATE FULL PRINTOUT
2262  013376  010367  177730                  MOV     R3,SAVR3        ;SAVE REGISTERS
2263  013402  010467  177726                  MOV     %4,SAVR4
2264  013406  005004                          CLR     %4              ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
2265  013410  005067  000144                  CLR     PRFLG           ;INITIALIZE CARRY FLAG FOR ROTATES
2266  013414  012703  000260                  MOV     #260,   %3      ;SETUP R3
2267  013420  005702                          TST     %2              ;CHECK BIT 15 OF NUMBER
2268  013422  100001                          BPL     .+4             ;BRANCH IF ZERO
2269  013424  005203                          INC     %3              ;INCREMENT R3 IF ONE
2270  013426  006102                          ROL     %2              ;ROTATE LEFT MOST OCTAL TO RIGHT END
2271  013430  006102                          ROL     %2
2272  013432  005567  000122                  ADC     PRFLG           ;STORE CARRY
2273  013436  005767  000114          P.CK:   TST     PRSFLG          ;CHECK FOR SHORT PRINTOUT
2274  013442  001403                          BEQ     P.WAIT          ;BRANCH IF NOT SET
2275  013444  020327  000260                  CMP     %3,     #260    ;CHECK FOR ZERO IF SET
2276  013450  001407                          BEQ     P.CONT          ;IF SET, GO TO NEXT CHARACTER
2277  013452  105777  165136          P.WAIT: TSTB    @TCSR           ;WAIT FOR TTY READY
2278  013456  100375                          BPL     P.WAIT
2279  013460  010377  165132                  MOV     %3,     @TDBR   ;OUTPUT NEXT CHARACTER
2280  013464  005067  000066                  CLR     PRSFLG          ;PRINT REST OF NUMBER AFTER A NON-ZERO DIGIT
2281  013470  005204          P.CONT: INC     %4              ;COUNT
2282  013472  020427  000006                  CMP     %4,     #6      ;CHECK FOR DONE
2283  013476  001005                          BNE     .+14            ;BRANCH IF NOT DONE
2284  013500  016703  177626                  MOV     SAVR3,R3        ;RESTORE REGISTERS
2285  013504  016704  177624                  MOV     SAVR4,%4
2286  013510  000207                          RTS     %7              ;RETURN
```

```
2287  013512  000241                          CLC                     ;CLEAR CARRY
2288  013514  005767  000040                  TST     PRFLG           ;CHECK FOR PREVIOUS CARRY
2289  013520  001403                          BEQ     .+10            ;BRANCH IF PREVIOUSLY ZERO
2290  013522  005067  000032                  CLR     PRFLG           ;INITIALIZE FLAG
2291  013526  000261                          SEC                     ;SET CARRY
2292  013530  006102                          ROL     %2              ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTER
2293  013532  006102                          ROL     %2
2294  013534  006102                          ROL     %2
2295  013536  005567  000016                  ADC     PRFLG           ;STORE CARRY
2296  013542  010203                          MOV     %2,     %3      ;LOAD DATA INTO R3
2297  013544  042703  177770                  BIC     #177770,%3      ;CLEAR ALL BUT LOWEST OCTAL DIGIT
2298  013550  052703  000260                  BIS     #260,   %3      ;SET TO ASCII EQUIVALENT
2299  013554  000730                          BR      P.CK            ;LOOP
2300  013556  000000                  PRSFLG: 0
2301  013560  000000                  PRFLG:  0
2302
2303                                  ;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 4000 TIMES
2304  013562  032767  040000  164000  SCOPEC: BIT     #40000, SR      ;TEST SR FOR SCOPE
2305  013570  001012                          BNE     SCOPEB          ;YES, SCOPE
2306  013572  032767  004000  163770          BIT     #4000,  SR      ;NO-TEST FOR ITERATION
2307  013600  001013                          BNE     SCOPEG          ;INHIBIT ITERATION
2308  013602  026767  000036  000032          CMP     SCOPEF, ICOUNT  ;COMPARE CURRENT COUNT TO MAX NUMBER
2309  013610  100007                          BPL     SCOPEG          ;EXIT-DONE
2310  013612  005267  000026                  INC     SCOPEF          ;INCREMENT COUNT
2311  013616  022606                  SCOPEB: CMP     (6)+,%6         ;REPOSITION STACK
2312  013620  012667  164152                  MOV     (6)+,PSR        ;RESTORE PREVIOUS PROCESSOR STATUS
2313  013624  000177  000016                  JMP     @RETURN         ;REPEAT TEST
2314  013630  005067  000010          SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
2315  013634  011667  000006                  MOV     @%6,    RETURN  ;SAVE SCOPE RETURN POINTER
2316  013640  000002                          RTI                     ;RETURN INLINE-NEXT TEST
2317  013642  004000                  ICOUNT: 4000                    ;ITERATION COUNT
2318  013644  000000                  SCOPEF: 0                       ;COUNT LOCATION FOR ITERATION LOOP
2319  013646  001032                  RETURN: TEST1+2                 ;ADDRESS OF LAST TEST
2320
2321                                  ;TRACE SUBROUTINE TO PRINT OUT LOCATION COUNTER OF EACH
2322                                  ;INSTRUCTION, AND THE CONTENTS OF R0 AND R5
2323  013650  032767  000001  163712  TRSUB:  BIT     #1,SR
2324  013656  001401                          BEQ     .+4
2325  013660  000002                          RTI
2326  013662  032767  000002  163700          BIT     #2,SR
2327  013670  001413                          BEQ     TRC1
2328  013672  005767  000106                  TST     TRCNT
2329  013676  001004                          BNE     TRC2
2330  013700  012767  000012  000076          MOV     #12,    TRCNT
2331  013706  000002                          RTI
2332  013710  005367  000070          TRC2:   DEC     TRCNT
2333  013714  001401                          BEQ     TRC1
2334  013716  000002                          RTI
2335  013720  004767  172320          TRC1:   JSR     %7,     CRLF
2336  013724  010267  000056                  MOV     R2,     TSAVR2
2337  013730  010367  000054                  MOV     R3,     TSAVR3
2338  013734  010467  000052                  MOV     %4,     TSAVR4
2339  013740  011602                          MOV     @SP,    R2
2340  013742  004767  177424                  JSR     %7,     PROCT
2341  013746  004767  172240                  JSR     %7,     SPACE
2342  013752  010002                          MOV     R0,     R2
```

```
2343  013754  000240                          NOP
2344  013756  000240                          NOP
2345  013760  105777  164630                  TSTB    @TCSR
2346  013764  100375                          BPL     .-4
2347  013766  016702  000014                  MOV     TSAVR2, R2
2348  013772  016703  000012                  MOV     TSAVR3, R3
2349  013776  016704  000010                  MOV     TSAVR4, %4
2350  014002  000002                          RTI
2351  014004  000000                TRCNT:    0
2352  014006  000000                TSAVR2:   0
2353  014010  000000                TSAVR3:   0
2354  014012  000000                TSAVR4:   0
2355
2356                                ;MESSAGE SUBROUTINE
2357                                ;TO USE, MOVE ADDRESS OF MESSAGE TO REGISTER 2
2358                                ;THEN JSR %7, TOUT
2359  014014  142777  000177  164572 TOUT:    BICB    #177,@TCSR      ;CLR INT FLAG
2360  014022  111267  000042                  MOVB    @%2,A.EOMK      ;MOVE IN EOM MARKER
2361  014026  005202                A.INC:    INC     %2              ;MOVE BYTE POINTER
2362  014030  121267  000034        A.TOUT:   CMPB    @%2,A.EOMK      ;COMPARE FOR EOM
2363  014034  001001                          BNE     .+4             ;BRANCH IF NOT END OF MESSAGE
2364  014036  000207                          RTS     %7              ;RETURN IF EOM
2365  014040  121227  000100                  CMPB    @%2,#'@         ;CHECK FOR CR,LF REQUEST
2366  014044  001003                          BNE     .+10            ;BRANCH IF NOT
2367  014046  004767  172172                  JSR     %7,CRLF         ;OUTPUT CARRIAGE RETURN, LINEFEED
2368  014052  000765                          BR      A.INC           ;NEXT CHARACTER
2369  014054  105777  164534                  TSTB    @TCSR           ;WAIT FOR TTY
2370  014060  100375                          BPL     .-4
2371  014062  112277  164530                  MOVB    (2)+,@TDBR      ;OUTPUT NEXT CHARACTER
2372  014066  000760                          BR      A.TOUT          ;CONTINUE
2373  014070     000                A.EOMK:   .BYTE   0
2374
2375  014071     057  040500  020116 MSG1:    .ASCII ;/@AN EXTERNAL TRIGGER SIGNAL OCCURRED@/;
2376  014076  054105  042524  047122
2377  014104  046101  052040  044522
2378  014112  043507  051105  051440
2379  014120  043511  040516  020114
2380  014126  041517  052503  051122
2381  014134  042105  027500
2382  014140  040057  044502  020124 MSG2:    .ASCII ;/@BIT 2 SET CAUSES AN INTERRUPT AT LEVEL /;
2383  014146  020062  042523  020124
2384  014154  040503  051525  051505
2385  014162  040440  020116  047111
2386  014170  042524  051122  050125
2387  014176  020124  052101  046040
2388  014204  053105  046105  027440
2389  014212  040057  044103     057 MSG4:    .ASCII ;/@CH/;
2390  014217     057  053440  051501 MSG5:    .ASCII ;/ WAS /;
2391  014224  027440
2392  014226  026057  044440  020123 MSG6:    .ASCII ;/, IS /;
2393  014234     057
2394  014235     057  040100  044103 MSG7:    .ASCII ;/@@CHANNEL   INITIAL VALUE   NEW VALUE/;
2395  014242  047101  042516  020114
2396  014250  020040  047111  052111
2397  014256  040511  020114  040526
2398  014264  052514  020105  020040
```

```
  2399  014272  042516  020127  040526
  2400  014300  052514  027505
  2401  014304  040057  047111  052520  MSG8:   .ASCII  ;/@INPUT A POSITIVE VOLTAGE TO THE INITIAL CHANNEL/;
  2402  014312  020124  020101  047520
  2403  014320  044523  044524  042526
  2404  014326  053040  046117  040524
  2405  014334  042507  052040  020117
  2406  014342  044124  020105  047111
  2407  014350  052111  040511  020114
  2408  014356  044103  047101  042516
  2409  014364  027514
  2410  014366  040057  044124  047105  MSG9:   .ASCII  ;/@THEN HIT 'CONTINUE' ON THE CONSOLE/;
  2411  014374  044040  052111  023440
  2412  014402  047503  052116  047111
  2413  014410  042525  020047  047117
  2414  014416  052040  042510  041440
  2415  014424  047117  047523  042514
  2416  014432     057
  2417  014433     057  044500  050116  MSG10:  .ASCII  ;/@INPUT A NEGATIVE VOLTAGLE TO THE INITIAL CHANNEL/;
  2418  014440  052125  040440  047040
  2419  014446  043505  052101  053111
  2420  014454  020105  047526  052114
  2421  014462  043501  042514  052040
  2422  014470  020117  044124  020105
  2423  014476  047111  052111  040511
  2424  014504  020114  044103  047101
  2425  014512  042516  027514
  2426  014516  040057  042100  043111  MSG11:  .ASCII  ;/@@DIFFERENTIAL GAIN NOT LINEAR WITHIN + OR -/;
  2427  014524  042506  042522  052116
  2428  014532  040511  020114  040507
  2429  014540  047111  047040  052117
  2430  014546  046040  047111  040505
  2431  014554  020122  044527  044124
  2432  014562  047111  025440  047440
  2433  014570  020122  027455
  2434  014574  030457  046040  041123  MSG12:  .ASCII  ;/1 LSB@/;
  2435  014602  027500
  2436  014604  030501  031057  046040  MSG13:  .ASCII  ;A1/2 LSB@A;
  2437  014612  041123  040500
  2438  014616  020057  020040  020040  MSG14:  .ASCII  ;/          GAIN8  GAIN4  GAIN2  GAIN1@/;
  2439  014624  020040  020040  043440
  2440  014632  044501  034116  020040
  2441  014640  040507  047111  020064
  2442  014646  043440  044501  031116
  2443  014654  020040  040507  047111
  2444  014662  040061     057
  2445  014665     057  042515  051501  MSG15:  .ASCII  ;/MEASURED  /;
  2446  014672  051125  042105  020040
  2447  014700     057
  2448  014701     057  046500  054101  MSG16:  .ASCII  ;/@MAX OK    /;
  2449  014706  047440  020113  020040
  2450  014714  027440
  2451  014716  040057  044515  020116  MSG17:  .ASCII  ;/@MIN OK    /;
  2452  014724  045517  020040  020040
  2453  014732     057
  2454  014733     101  020061  027461  MSG20:  .ASCII  ;A1 1/4 LSB@A;
```

```
  2455  014740  020064  051514  040102
  2456  014746     101
  2457          000001                          .END
```

```
ADCS    000602          680#   693
ADCSR =%000001          638#   693*   694    714    721*   722    727*   728    740*   742    745*   749    758*
                        759    761*   762    769*   773    777*   781    788*   789*   793    800*   805    812*
                        814*   818    826*   827*   831*   832    839*   840*   844    851*   853*   857    864*
                        866*   867*   868    875    876*   877*   878*   879    882    889*   890    892*   893
                        900*   901    903*   904    907    914*   915    918*   919    922    929*   930*   931
                        934    941*   942*   943*   944    947    954*   955*   956    958*   959    966*   971*
                        972*   973    979*   988*   991*   997*  1000   1008*  1009*  1011*  1015   1040*  1042*
                       1044*  1046*  1048*  1050*  1060*  1061*  1062   1066*  1071*  1079*  1080*  1081   1086*
                       1088*  1092*  1137*  1139*  1144*  1145*  1146   1151*  1153*  1158*  1172*  1177*  1178*
                       1179   1186*  1194*  1199*  1210*  1211   1233*  1234*  1235   1253*  1254*  1255   1278*
                       1279*  1280   1284*  1285   1289*  1290   1294*  1295   1374*  1375   1378   1385*  1390*
                       1391*  1392   1405*  1482*  1490*  1491   1603*  1608*  1609   1629*  1630   1660*  1663*
                       1664   1668*  1669   1672   1682*  1683   1688   1695   1700   1720*  1721   1724   1726*
                       1732*  1733   1749*  1750*  1751   1754*  1755   1757   1790*  1792*  1793   1881*  1882*
                       1884*  1885*  1901*  1902   1947*  1950*  1954   1965*  2058*  2059   2104*  2105
ADCSRB  000606          692#   694*   695*   801*  1883*  1899*  1949*  1961*
ADDBR   000604          681#   720    725    741    757    768    776    787    799    813    825    838    852
                        865    888    899    913    917    928    940    953    965    990   1059   1078   1137
                       1139   1143   1172   1176   1181   1198   1213   1214   1232   1237   1252   1257   1277
                       1282   1287   1292   1297   1384   1481   1493   1552   1555   1604   1611   1632   1659
                       1667   1671   1687   1723   1748   1753   1761   1789   1795   1904   1946   1957   2057
                       2061   2103   2107
ADINT   000610          683#   696
ALLOW4  013046         2087   2199#
ALLOW8  013050         2122   2200#
AV      013030         2042   2062   2108   2150   2179   2192#
AVG40   004664         1276   1301#
A.EOMK  014070         2360*  2362   2373#
A.INC   014026         2361#  2368
A.TOUT  014030         2362#  2372
BEGIN   000703          659    704#
BELL    000330         1410   1563#  1925   2135
B.AVG   006174         1521   1524#
B.LOOP  006142         1513#  1523
CHCNT   013032         2046*  2132*  2133   2193#
CHOUT   011406         1948*  1949   1960*  1961   1969#
CK      013316         2227   2241#
CK40    005004         1299   1318#
CMP40   005160         1354#
CONAV   006130         1303   1307   1312   1316   1510#  1778   1858   2052   2102
CONCNT  011216         1880*  1889*  1891   1896*  1934#
CONVRT  006010         1301   1305   1310   1314   1481#  1776   1857   2051   2101
COUNT   000612          684#   746*   747*   770*   771*   778*   779*   790*   791*   802*   803*   815*   816*
                        828*   829*   841*   842*   854*   855*  1209*  1217*  1231*  1241*  1251*  1263*  1489*
                       1495*  1681*  1685*  1694*  1705*
COUNTR  011220         1868*  1922*  1935#
CRLF    006244         1137   1139   1172   1543#  1581   1657   1710   1771   2015   2231   2335   2367
DBRCK   006276         1283   1288   1293   1298   1494   1552#
DSCNT1  006572         1621   1623#
DSCNT2  006610         1624   1626#
DSPLAY  006426          663   1594#
DSPLY   006464         1600   1603#  1617   1643
EQUAL   013044         2056*  2075*  2099*  2116*  2166   2183   2198#
ERFLG1  007424         1692*  1698*  1703*  1708   1712*  1729*  1731#  1736*
EXCK    007150         1675#
```

```
EXCNT   007522          1763#
EXCNT1  007226          1684    1688#
EXCNT2  007270          1696    1700#
EXCNT3  007304          1701    1705#
EXCNT4  007336          1709    1713#
EXCONT  007500          1755#   1762
EXFAST  007450           667    1747#
EXFLOP  007454          1748#   1760
EXINT   007366          1678    1720#   1727
EXLOOP  007210          1683#   1690
EXLP1   007254          1695#   1706
EXPRNT  007314          1686    1708#
EXSIT   007426          1680    1732#
EXSW5   007250          1676    1694#
EXTEST  007074           665    1656#
EXTST1  007110          1659#   1693    1699    1704    1707    1719    1730    1737
FAST    006612          1606    1627#
FLY     007524           669    1770#
FLY.1   007632          1774    1789#
FLY.2   007576          1780#   1796
FLY.3   007540          1773#   1787
FSHOW   006644          1635#   1642
FTABLE  006674          1627    1633    1635    1641    1644#
FULLCK  006374          1577#   1718    1786
F.LOOP  006622          1629#   1634
GAIN1   005760          1282*   1302    1362    1364    1433    1466#   1777    1779
GAIN2   005762          1287*   1306    1358    1360    1430    1467#
GAIN4   005764          1297*   1311    1354    1356    1427    1468#
GAINB   005766          1292*   1315    1318    1322    1327    1332    1339    1345    1424    1438    1452    1469#
GCNT    005746          1415    1463#
GMSG    005442          1355    1357    1359    1361    1363    1365    1414#
HGAIN1  006002          1344*   1362    1447    1475#
HGAIN2  005776          1331*   1358    1444    1473#
HGAIN4  005772          1321*   1354    1441    1471#
HLT   = 104000           635#    716     724     732     744     751     764     775     783     795     807     820     834
                         846     870     881     895     906     921     933     946     961     975     994    1013    1040
                        1042    1044    1046    1048    1050    1069    1090    1137    1139    1169    1172    1190    1216
                        1239    1261    1377    1394    1403    1559    1674    1691    1697    1702    1728    1735    1759
                        1956    2092    2127
ICOUNT  013642           697*   1230*   1272*   1370*   1383*   1570*   2308    2317#
INITCH  000622           688#    705*    706*   1200    1234    1254    1274    1846*   1884    1947    1948    1992    2011
                        2038*   2039*   2047
INTFLG  000620           687#    700*   1137*   1139*   1162    1164*   1166    1172*
INTVC =:000004           641#    696*    968*    969*    976     980*    985*    987*    998*   1006*   1007*   1017*   1040*
                        1042*   1044*   1046*   1048*   1050*   1064*   1065*   1072*   1083*   1084*   1094*   1137*   1139*
                        1148*   1149*   1154    1155*   1159*   1172*   1182*   1184*   1192*   1387*   1388*   1395    1398*
                        1401    1406*   1943*   1945*   1963*
ITABLE  010414          1856    1864    1876    1927#   2010
LCOUNT  011410          1942*   1958*   1970#
LGAIN1  006004          1353*   1364    1461    1476#
LGAIN2  006000          1338*   1360    1458    1474#
LGAIN4  005774          1326*   1356    1455    1472#
LOADTB  011222          1865    1867    1870    1939#
LOOP1   006052          1490#   1496
LOOP2   006044          1489#   1501
LOOP3B  004362          1234#   1242
```

```
LOOP39  004454            1254#   1264
LSBFLG  006006            1477#
LTBDON  011364            1959    1963#
LTBSRV  011330            1943    1954#
L.CNT   006424            1577*   1578    1582*   1584#   1658*   1711*   1772*
MSGA    013052            2143    2201#
MSGB    013073            2148    2177    2204#
MSGC    013102            2152    2206#
MSGD    013113            2156    2208#
MSGE    013125            2160    2185    2210#
MSGF    013135            2164    2212#
MSGG    013147            2172    2214#
MSGH    013170            2181    2217#
MSGI    013214            2093    2128    2221#
MSG1    014071             859    2375#
MSG10   014433            1246    2417#
MSG11   014516            1416    2426#
MSG12   014574            2434#
MSG13   014604            2436#
MSG14   014616            1420    2438#
MSG15   014665            1422    2445#
MSG16   014701            1436    2448#
MSG17   014716            1450    2451#
MSG2    014140            1137    1139    1172    2382#
MSG20   014733            1418    2454#
MSG4    014212            1990    2389#
MSG5    014217            1997    2390#
MSG6    014226            2001    2392#
MSG7    014235            2008    2394#
MSG8    014304            1225    2401#
MSG9    014366            1227    1248    2410#
NINT    004144            1182    1190#
NOP   = 000240             634#
NTABLE  011014            1869    1871    1920    1931#
NUMCH   000624             689#    708*   1848*   1849    1851*   1855    1894    1912    1939    1941*   1942    2014    2041*
                          2043    2045*   2133
OTABLE  010614            1866    1874    1918    1929#
OUT     013042            2055*   2073*   2081*   2083*   2087    2098*   2114*   2122    2162    2187    2197#
OVER    013036            2053*   2065*   2078    2081    2154    2195#
PC    =%000007             644#
PRFLG   013560            2265*   2272*   2288    2290*   2295*   2301#
PRINT   013222             650    2225#
PRINT4  012524            2086    2091    2140#
PRINTB  012704            2121    2126    2169#
PROC    000630             691#
PROCT   013372            1425    1428    1431    1434    1439    1442    1445    1448    1453    1456    1459    1462    1780
                          2000    2004    2018    2022    2026    2234    2239    2256    2261#   2340
PRSFLG  013556            2254*   2261*   2273    2280*   2300#
PRSHRT  013342            1137    1139    1172    1714    1996    2096    2131    2147    2151    2155    2159    2163    2167
                          2176    2180    2184    2188    2254#
PSR   = 177776             633#    699*    726*    967*    969     970*    976*    986*    987     989*    993*   1005*   1007
                          1040*   1042*   1044*   1046*   1048*   1050*   1058*   1065    1077*   1084    1085*   1089*   1137*
                          1139*   1142*   1149    1150*   1154*   1172*   1183*   1184    1185*   1188*   1386*   1388    1389*
                          1395*   1399*   1401*   1944*   1945    1951*   1967*   2312*
P.CK    013436            2273#   2299
P.CONT  013470            2276    2281#
```

```
P.WAIT  013452          2274    2277#   2278
RELOOP  004266          1214#   1218
REPTST  004244          1201    1203    1205    1207    1209#
RESTRT  000732           661     710#   1411
RESULT  013034          2061*   2062*   2063    2067    2071    2107*   2108*   2110    2112    2194#
RETRNX  000626           690#   1806*   1807*   1811    1813
RETURN  013646           698*   1463*   1569*   1810*   2313    2315*   2319#
R0      =%000000         637#   1213*   1214    1596*   1599    1611*   1636*   1864*   1866*   1869*   1871*   1886    1890*
                        1897    1904*   1906    1909    1918*   1919*   1920    1957*   2003    2010*   2021    2025    2027
                        2342
R2      =%000002         639#    728*    729*    730    1225*   1227*   1246*   1248*   1257*   1258*   1259    1416*   1418*
                        1420*   1422*   1424*   1427*   1430*   1433*   1436*   1438*   1441*   1444*   1447*   1450*   1452*
                        1455*   1458*   1461*   1555*   1556*   1557    1577    1713*   1716*   1717    1779*   1784*   1785
                        1795*   1990*   1992*   1993*   1994*   1995*   1997*   1999*   2001*   2003*   2008*   2016*   2021*
                        2025*   2078*   2079*   2093*   2095*   2128*   2130*   2145*   2146*   2148*   2150*   2152*   2154*
                        2156*   2158*   2160*   2162*   2164*   2166*   2172*   2174*   2175*   2177*   2179*   2181*   2183*
                        2185*   2187*   2255    2336    2339*   2342*   2347*
R3      =%000003         640#   1318*   1319*   1320*   1321    1322*   1323*   1324*   1325*   1326    1327*   1328*   1329*
                        1330*   1331    1332*   1333*   1334*   1335*   1336*   1337*   1338    1339*   1340*   1341*   1342*
                        1343*   1344    1345*   1346*   1347*   1348*   1349*   1350*   1351*   1352*   1353    1483*   1488*
                        1493*   1497    1500*   1511*   1513*   1514*   1515*   1516*   1517*   1518*   1519    1522*   1578*
                        1579    1627*   1632*   1633    1635*   1636    1641    1717*   1785*   1886*   1887*   1906*   1907*
                        1976    1979*   1981*   2262    2281*   2337    2348*
R5      =%000005         642#   1510*   1511    1520    1522    1874*   1876*   1887    1898    1907    1910    1999    2011*
                        2012*   2013*   2016    2017*
SAMCNT  013026          2050*   2076*   2100*   2117*   2191#
SAMNXT  012514          2134    2137#
SAMSIZ  013024          2050    2100    2190#
SAVCSR  007446          1688*   1689    1738#
SAVPC   013336          2228*   2233    2248#
SAVPSR  013340          2229*   2238    2249#
SAVR2   013330          2232*   2240    2245#
SAVR3   013332          2246#   2262*   2284
SAVR4   013334          2247#   2263*   2285
SCOPE = 104400           636#    718     735     755     766     785     797     809     822     836     848     862     872
                         885     897     910     925     937     950     963     983    1003    1040    1042    1044    1046
                        1048    1050    1057    1076    1137    1139    1141    1172    1175    1197    1222    1245    1271
                        1367    1381    1408
SCOPEB  013616          2305    2311#
SCOPEC  013562           653    2304#
SCOPEF  013644          1809*   1812*   2308    2310*   2314*   2318#
SCOPEG  013630          2307    2309    2314#
SELECT  006126          1200*   1202*   1204*   1206*   1210    1273*   1274*   1279    1304*   1308*   1309*   1313*   1490
                        1506#   1608    1618*   1619*   1622*   1625*   1629    1792    1846    1860*   2047*   2048*   2049*
                        2058    2095    2097*   2104    2130    2137*   2145    2174
SET     006540          1607    1618#   1628    1775    1791    1845
SETUP   000632           693#    710    1594    1656    1747    1770    1804    1843    2036
SET4    004044          1172#
SET5    003660          1163    1166#
SET6    003520          1139#
SET7    003334          1137#
SICK1   012106          2064    2067#
SICK2   012124          2068    2071#
SICK3   012202          2082    2084#
SICNT1  012174          2080    2083#
SICNT2  012376          2109    2111    2113    2116#
```

```
SICNT3  012470            2123    2132#
SICNT4  012140            2072    2075#
SICNT5  012402            2115    2117#
SICNT6  012144            2066    2070    2074    2076#
SICNT7  012240            2090    2092#
SICNT8  012446            2125    2127#
SIGMA1  011672             675    2035#
SIGMA2  011754            2044    2046#   2136
SIGMA3  011766            2048#   2138
SIGMA5  012262            2088    2097#
SILOOP  012040            2057#   2077
SILOP1  012316            2103#   2118
SP    =%000006             643#   1040    1042    1044    1046    1048    1050    1070    1093    1137    1139    1161    1172
                          1191    1966    1967    2339
SPACE   006212            1426    1429    1432    1435    1440    1443    1446    1449    1454    1457    1460    1533#   1537
                          1715    2020    2024    2341
SPACEX  006242            1536*   1538*   1540#   2019*   2023*
SR    = 177570             632#    705     708    1223    1275    1414    1605    1614    1618    1620    1623    1675    1677
                          1679    1773    1806    1848    1853    1872    1878    1915    1974    1987    2038    2041    2084
                          2089    2119    2124    2140    2169    2225    2241    2304    2306    2323    2326
STACK   000600             658     660     662     664     666     668     670     672     674     679#   2035
STORE   005770            1302*   1306*   1311*   1315*   1470#   1512*   1519*   1524*   1525*   1526*   1777*   1856*   1859*
                          2042*
SUM1    006116            1484*   1488    1502#   1510
SUM2    006120            1485*   1503#
SUM3    006122            1486*   1504#
SUM4    006124            1487*   1497    1505#   1520
TCSR    000614             685#    711    1371    1533    1543    1546    1563    1781    2235    2257    2277    2345    2359*
                          2369
TDBR    000616             686#   1535*   1545*   1548*   1565*   1783*   2237*   2259*   2279*   2371*
TESTX   007662             671    1804#
TEST0   000754             698     718#   1569
TEST1   001030             735#   2319
TEST10  001474             858     862#
TEST11  001516             872#
TEST12  001542             885#
TEST13  001570             897#
TEST14  001622             910#
TEST15  001660             925#
TEST16  001706             937#
TEST17  001734             950#
TEST18  001764             963#
TEST19  002064             983#
TEST2   001072             755#
TEST20  002162            1003#
TEST21  002240            1040#
TEST22  002332            1042#
TEST23  002424            1044#
TEST24  002516            1046#
TEST25  002610            1048#
TEST26  002702            1050#
TEST27  002774            1057#
TEST28  003060            1076#
TEST29  003162            1137#
TEST3   001120             766#
TEST30  003346            1137    1139#
```

```
TEST31  003532          1139    1141#
TEST32  003672          1157    1165    1167    1172#
TEST36  004056          1172    1175#
TEST37  004162          1197#
TEST38  004306          1208    1222#
TEST39  004414          1240    1245#
TEST4   001202           785#
TEST40  004512          1224    1262    1271#
TEST41  005240          1367#   1463    1464
TEST42  005274          1381#
TEST5   001236           797#
TEST6   001274           809#
TEST7   001332           822#
TEST8   001374           836#
TEST9   001432           848#
TINTD   002050           968     978#
TINTE   002142           985     996#
TINTX   003146          1083    1092#
TINT1A  002760          1050#
TINT2   005366          1387    1397#
TINT2A  002666          1048#
TINT3   005412          1396    1398    1402    1403#
TINT3A  002574          1046#
TINT3B  003766          1172#
TINT4A  002502          1044#
TINT4B  003626          1148    1158#
TINT5A  002410          1042#
TINT5B  003442          1139#
TINT6A  002316          1040#
TINT6B  003256          1137#
TINT7A  002220          1006    1013#
TINT7B  003042          1064    1069#
TOUT    014014           860    1137    1139    1172    1226    1228    1247    1249    1417    1419    1421    1423    1437
                        1451    1991    1998    2002    2009    2094    2129    2144    2149    2153    2157    2161    2165
                        2173    2178    2182    2186    2359#
TRCNT   014004          2328    2330*   2332*   2351#
TRC1    013720          2327    2333    2335#
TRC2    013710          2329    2332#
TRSUB   013650          2323#
TSAVR2  014006          2336*   2347    2352#
TSAVR3  014010          2337*   2348    2353#
TSAVR4  014012          2338*   2349    2354#
UNDER   013040          2054*   2069*   2079    2083    2158    2196#
WAGAIN  010110          1869#   1924
WAGON   010102          1868#   1926
WASIS   007734           673    1843#
WCOMP   011412          1888    1908    1974#
WCONT1  010062          1854    1864#
WCONT2  010072          1863    1866#
WCONT3  010340          1895    1913    1915#
WCONV   010272          1892    1901#
WCOUNT  011214          1855*   1861*   1877*   1893*   1894    1911*   1912    1933#   1995    2014*   2028*
WLOOP1  010030          1857#   1862
WLOOP2  010306          1879    1906#   1914
WLOOP3  010360          1919#   1921
WLOOP4  010210          1886#   1905
```

DZADBB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
WMSG    011444          1977    1980    1982    1987#
WPLOOP  011602          2015#   2029
WPRINT  011546          1917    2008#
WSTART  010004          1953#
XLOOP   007724          1810    1812#
.     = 014747           647#    649#    652#    657#    [illegible]3#    712     715     723     731     743     748     750     760
                         763     772     774     780     [illegible]2     792     794     804     806     817     819     830     833
                         843     846     856     869     880     883     891     894     902     905     908     916     920
                         923     932     935     945     948     957     960     974     977     995    1001    1012    1016
                        1040    1042    1044    1046    1048    1050    1063    1068    1082    1091    1137    1139    1147
                        1172    1180    1189    1212    1215    1236    1238    1256    1260    1281    1286    1291    1296
                        1372    1376    1379    1393    1492    1498    1534    1544    1547    1553    1558    1564    1567
                        1580    1602    1610    1615    1631    1645#   1665    1670    1673    1722    1725    1734    1752
                        1756    1758    1782    1794    1850    1873    1875    1900    1903    1916    1923    1928#   1930#
                        1932#   1940    1952    1953    1955    1975    1988    2060    2085    2106    2120    2141    2170
                        2226    2236    2242    2258    2268    2283    2289    2324    2346    2363    2366    2370
```

```
INT      1019#   1039    1041    1043    1045    1047    1049
INT2     1097#   1136    1138    1171
```

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADC 1325 1335 1337 1348 1350 1352 1518 1526 2272 2295
ADD 1319 1328 1340 1493 1500 1519 1577 1578 1807 1859 1995 2081 2083
ASR 1320 1324 1329 1330 1334 1336 1341 1342 1343 1347 1349 1351 1513 1514 1515
1516 1517 1524 1525
BEQ 715 723 731 1215 1224 1238 1260 1276 1521 1558 1621 1624 1686 1709 1722
1879 1916 1923 1975 1988 2072 2085 2109 2111 2113 2120 2141 2170 2226 2274
2276 2289 2324 2327 2333
BGT 1862 2029
BIC 706 729 970 989 1085 1185 1206 1294 1308 1389 1619 1951 1994 2013 2039
2048 2146 2175 2297
BICB 831 1885 2359
BIS 726 967 986 993 1005 1040 1042 1044 1046 1048 1050 1058 1077 1089 1137
1139 1142 1172 1183 1188 1202 1204 1274 1284 1289 1304 1309 1313 1386 1622
1625 1884 1944 2049 2097 2298
BISB 812 853 1682 1732 1882 1950
BIT 722 1223 1237 1259 1275 1414 1552 1557 1605 1620 1623 1675 1677 1721 1773
1853 1872 1878 1915 1974 1987 2084 2119 2140 2169 2225 2304 2306 2323 2326
BLE 1959
BMI 750 763 794 806 833 845 869 880 894 905 932 945 1137 1139 1163
1172 1355 1357 1359 1361 1363 1365 1580 1680 1725 1850 1892 1940 1980 2088
2123 2134
BNE 748 772 780 792 804 817 830 843 856 1218 1242 1264 1415 1496 1498
1553 1600 1606 1634 1642 1676 1678 1706 1774 1854 1873 1921 1977 2064 2068
2077 2118 2256 2283 2305 2307 2329 2363 2366
BPL 712 743 760 774 782 819 858 883 891 902 908 916 920 923 935
948 957 960 974 1001 1016 1040 1042 1044 1046 1048 1050 1063 1082 1137
1139 1147 1167 1172 1180 1212 1236 1256 1281 1286 1291 1296 1372 1376 1379
1393 1492 1534 1537 1544 1547 1564 1567 1610 1615 1631 1665 1670 1673 1684
1690 1696 1701 1734 1752 1756 1758 1782 1794 1895 1903 1913 1955 1982 2044
2060 2080 2090 2106 2125 2236 2242 2258 2268 2278 2309 2346 2370
BR 977 995 1012 1040 1042 1044 1046 1048 1050 1068 1091 1137 1139 1157 1165
1172 1189 1208 1240 1262 1299 1396 1402 1501 1523 1602 1617 1693 1699 1704
1707 1719 1727 1730 1737 1760 1762 1787 1796 1863 1875 1900 1905 1914 1952
1953 2066 2070 2074 2082 2115 2227 2299 2368 2372
CLC 2287
CLR 700 721 789 814 854 866 867 877 903 943 958 981 999 1018 1040
1042 1044 1046 1048 1050 1061 1073 1080 1095 1137 1139 1145 1156 1160 1172
1178 1193 1230 1273 1399 1407 1483 1484 1485 1486 1487 1512 1538 1568 1582
1597 1598 1658 1663 1668 1694 1711 1712 1750 1772 1809 1812 1877 1880 1896
1964 2046 2053 2054 2055 2056 2098 2099 2261 2264 2265 2280 2290 2314
CLRB 740 761 769 777 788 800 801 826 851 864 876 941 954 966 979
988 997 1008 1011 1040 1042 1044 1046 1048 1050 1060 1071 1079 1088 1092
1137 1139 1144 1153 1158 1172 1177 1194 1199 1233 1253 1278 1385 1405 1482
1603 1660 1749 1790 1881 1883 1965
CMP 730 978 996 1014 1040 1042 1044 1046 1048 1050 1070 1093 1137 1139 1161
1166 1172 1191 1214 1354 1356 1358 1360 1362 1364 1397 1404 1497 1520 1579
1633 1641 1849 1891 1894 1912 1920 1939 2043 2063 2067 2087 2110 2112 2122
2133 2230 2275 2282 2308 2311
CMPB 2362 2365
COM 1258 1556
DEC 1217 1241 1263 1495 1536 1685 1861 1922 1958 2028 2076 2117 2332
EMT 635
HALT 647 704 707 709 1229 1250 1601 1616 1805 1808 1844 1847 1852 2037 2040
2243
INC 695 747 771 779 791 803 816 829 840 842 878 930 1374 1692 1698
1703 1726 1729 1736 1889 1893 1911 1960 1979 2017 2065 2069 2073 2075 2114

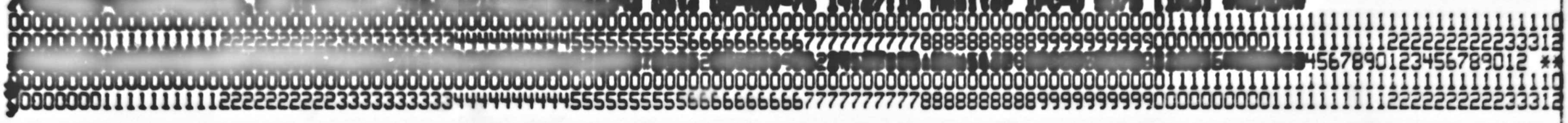
DZADBB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
        2116    2132    2269    2281    2310    2361
INCB     745     855     892     918     942    1705    1860    1899    1901    2137
JMP      659     661     663     665     667     669     671     673     675    1411    1464    1643    1811    1813    1924
        1926    2136    2138    2313
JSR      710     860    1137    1139    1172    1201    1203    1205    1207    1226    1228    1247    1249    1283    1288
        1293    1298    1301    1303    1305    1307    1310    1312    1314    1316    1410    1417    1419    1421    1423
        1425    1426    1428    1429    1431    1432    1434    1435    1437    1439    1440    1442    1443    1445    1446
        1448    1449    1451    1453    1454    1456    1457    1459    1460    1462    1494    1581    1594    1607    1628
        1656    1657    1710    1714    1715    1718    1747    1770    1771    1775    1776    1778    1780    1786    1791
        1804    1843    1845    1857    1858    1865    1867    1870    1888    1908    1917    1925    1991    1996    1998
        2000    2002    2004    2009    2015    2018    2020    2022    2024    2026    2036    2051    2052    2086    2091
        2094    2096    2101    2102    2121    2126    2129    2131    2135    2144    2147    2149    2151    2153    2155
        2157    2159    2161    2163    2165    2167    2173    2176    2178    2180    2182    2184    2186    2188    2231
        2234    2239    2335    2340    2341    2367
MOV      658     660     662     664     666     668     670     672     674     693     694     696     697     698     699
         708     727     728     746     758     770     778     790     802     815     827     828     841     859     889
         900     914     929     955     968     969     971     976     980     985     987     991     998    1006    1007
        1009    1017    1040    1042    1044    1046    1048    1050    1064    1065    1072    1083    1084    1094    1137
        1139    1148    1149    1150    1154    1155    1159    1164    1172    1182    1184    1192    1200    1209    1210
        1213    1225    1227    1231    1234    1246    1248    1251    1254    1257    1272    1279    1282    1287    1292
        1297    1302    1306    1311    1315    1318    1321    1322    1326    1327    1331    1332    1338    1339    1344
        1345    1353    1370    1383    1387    1388    1390    1395    1398    1401    1406    1416    1418    1420    1422
        1424    1427    1430    1433    1436    1438    1441    1444    1447    1450    1452    1455    1458    1461    1463
        1488    1489    1490    1510    1511    1522    1535    1545    1548    1555    1565    1569    1570    1595    1596
        1608    1611    1627    1629    1632    1635    1636    1681    1688    1713    1716    1717    1720    1754    1777
        1779    1783    1784    1785    1792    1795    1806    1810    1846    1848    1851    1855    1856    1864    1866
        1868    1869    1871    1874    1876    1886    1890    1904    1906    1918    1919    1941    1942    1943    1945
        1957    1963    1967    1990    1992    1997    1999    2001    2003    2008    2010    2011    2014    2016    2019
        2021    2023    2025    2035    2041    2042    2045    2047    2050    2058    2061    2078    2093    2100    2104
        2107    2128    2143    2148    2150    2152    2154    2156    2158    2160    2162    2164    2166    2172    2177
        2179    2181    2183    2185    2187    2228    2229    2232    2233    2237    2238    2240    2254    2259    2262
        2263    2266    2279    2284    2285    2296    2312    2315    2330    2336    2337    2338    2339    2342    2347
        2348    2349
MOVB     705     839     972    1066    1086    1137    1139    1151    1172    1186    1391    1618    1947    1948    1949
        1961    2038    2095    2130    2145    2174    2360    2371
NOP      992    1010    1067    1087    1137    1139    1152    1172    1187    1400    2343    2344
RESET    713    1373    1612    1613    1637    1638    1639    1640
ROL     2270    2271    2292    2293    2294
RTI     1595    1962    2244    2316    2325    2331    2334    2350
RTS      701    1219    1499    1527    1539    1549    1554    1560    1571    1583    1626    1968    1978    1983    1989
        2005    2030    2142    2168    2171    2189    2260    2286    2364
SEC     2291
SUB     1323    1333    1346    1887    1907    1981    2062    2079    2108
SWAB    1993    2012
TRAP     636
TST      714     720     725     741     757     768     776     787     799     813     825     838     852     865     868
         879     888     893     899     904     913     917     919     928     931     940     944     953     959     965
         990    1059    1078    1137    1139    1143    1162    1172    1176    1181    1198    1232    1252    1277    1375
        1481    1599    1604    1614    1659    1667    1671    1687    1689    1700    1708    1723    1724    1733    1748
        1753    1757    1761    1789    1897    1898    1909    1910    1946    1954    1966    1976    2027    2057    2071
        2103    2241    2255    2267    2273    2288    2328
TSTB     711     742     749     759     762     773     781     793     805     818     832     844     857     875     882
         890     901     907     915     922     934     947     956     973    1000    1015    1040    1042    1044    1046
        1048    1050    1062    1081    1137    1139    1146    1172    1179    1211    1235    1255    1280    1285    1290
        1295    1371    1378    1384    1392    1491    1533    1543    1546    1563    1566    1609    1630    1664    1669
        1672    1679    1683    1695    1751    1755    1781    1793    1902    2059    2089    2105    2124    2235    2257
```

```
.MAIN.  MACY11 27(732)  23-MAR-76  10:53  PAGE 60
DZADBB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

          2277    2345    2369
.ASCII    2201    2204    2206    2208    2210    2212    2214    2217    2221    2375    2382    2389    2390    2392    2394
          2401    2410    2417    2426    2434    2436    2438    2445    2448    2451    2454
.BYTE     2373
.END      2457
.LIST      647
.MACR     1019    1097
.NLIST     647
.REM         1
.REPT      647


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *DZADBB,DZADBB/SOL/CRF/EN:ABS=DZADBB
 RUN-TIME:  8 16 3 SECONDS
 RUN-TIME RATIO:  94/29=3.2
 CORE USED:  9K  (17 PAGES)
```

G05

Spooler runtime 9 Seconds, 40 KCS, 256 disk reads, 4 disk writes, 57 pages



```
   10                    ...B1                                    ...B5
   47                    ...C1                                    ...C5
  103                    ...D1              2048    2146          ...D5
  159                    ...E1              1804    1843          ...E5
  215                    ...F1  .REPT        647    1097          ...F5
  271                    ...G1   **END** USER DAVIES,TOM          ...G5
  327                    ...H1
  383                    ...I1
  439                    ...J1
  495                    ...K1
  506                    ...L1
  560                    ...M1
  616                    ...N1

  672  000270  012706    ...B2
  728  001012  011102    ...C2
  784  001200  104000    ...D2
  840  001406  005211    ...E2
  896  001566  104000    ...F2
  952                     ...G2
 1008  002204  105011    ...H2
 1064  003020  012714    ...I2
 1120                     ...J2
 1176  004060  005777    ...K2
 1232  004354  005777    ...L2
 1288  004612  004767    ...M2
 1344  005124  010367    ...N2

 1400  005400  000240    ...B3
 1456  005712  004767    ...C3
 1512  006136  005077    ...D3
 1568  006352  005037    ...E3
 1624  006600  001403    ...F3
 1680  007174  100514    ...G3
 1736  007440  005267    ...H3
 1792  007644  016711    ...I3
 1848  007756  016767    ...J3
 1904  010300  017710    ...K3
 1960  011350  005267    ...L3
 2016  011606  010502    ...M3
 2072  012130  001403    ...N3

 2128  012450  012702    ...B4
 2184  012776  004767    ...C4
 2240  013312  016702    ...D4
 2296  013542  010203    ...E4
 2352  014006  000000    ...F4
 2408  014356  044103    ...G4
                         ...H4
                         ...I4
EXLOOP 007210            ...J4
MSGE   013125            ...K4
                         ...L4
SIGMA5 012262            ...M4
TEST42 005274            ...N4
```