ICR-11

FIELD TEST PROGRAM

MD-11-DZIRB-A

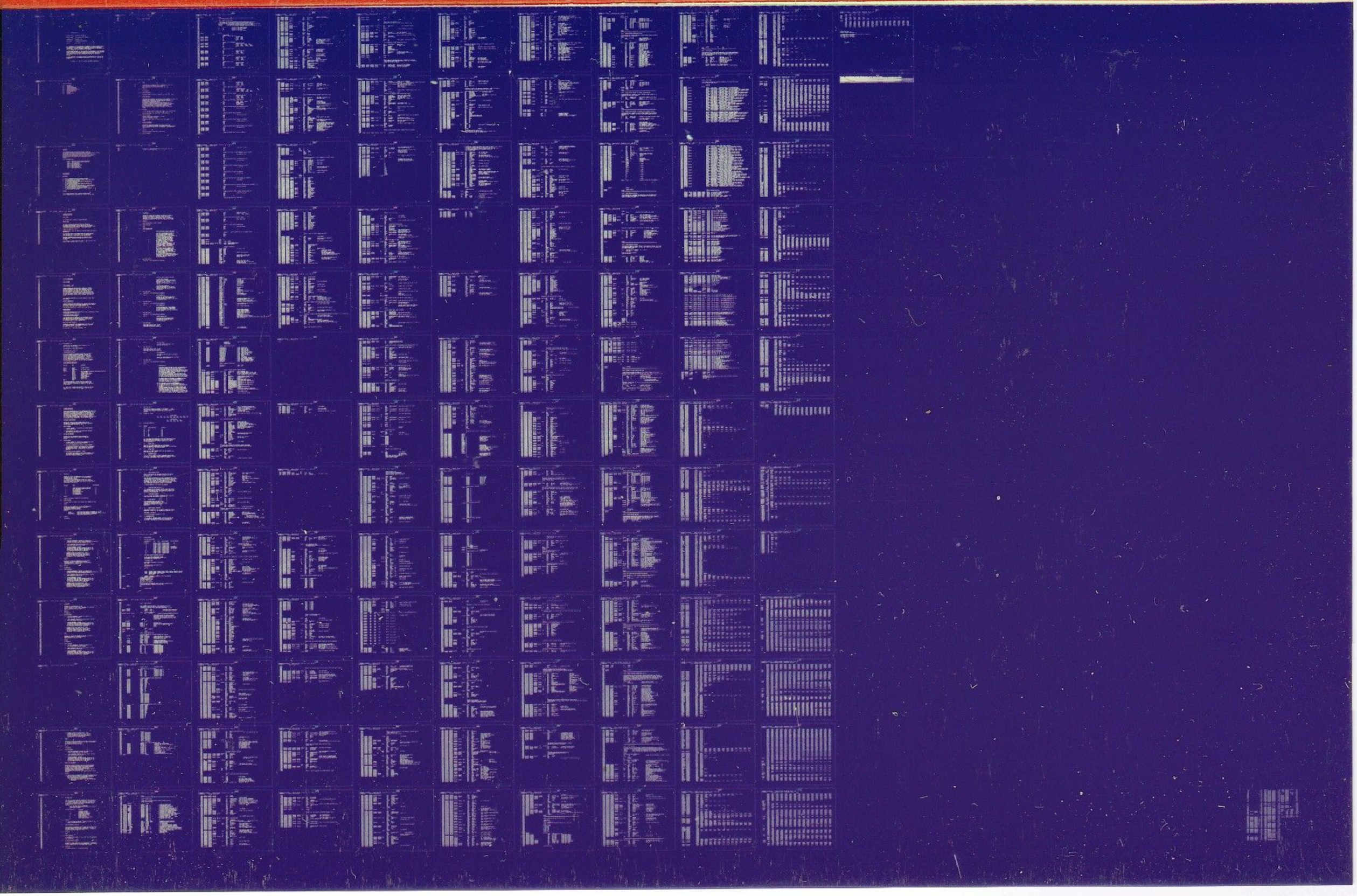
EP-DZIRB-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
                        .REM    %
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZIRB-A-D
PRODUCT NAME: ICR11 FIELD TEST PROGRAM
DATE CREATED: FEBRUARY 1976
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: DAN DEKNIS

```
                                0.0     TABLE OF CONTENTS

                                        1.0     ABSTRACT
                                        2.0     REQUIREMENTS
                                        3.0     LOADING PROCEDURE
                                        4.0     STARTING PROCEDURE
                                        5.0     SWITCH REGISTER SETTINGS
                                        6.0     PROGRAM QUESTIONS
                                        7.0     TEST DESCRIPTIONS
                                        8.0     SUMMARY
                                        9.0     MISCELLANEOUS
                                        10.0    LISTING
```

```
                                1.0     ABSTRACT
                                        ********

                                        THIS PROGRAM ALLOWS THE USER TO CHECKOUT, DEBUG OR
                                        DEMONSTRATE THE REMOTE INDUSTRIAL CONTROL SUBSYSTEM (ICR11) OPTIONS.
                                        THE PROGRAM IS DIVIDED INTO DIFFERENT TESTS AIMED AT
                                        EXERCISING THE OPTIONS WITHIN THE ICR11 FILE BOX.  THE TESTS
                                        ARE SELECTED BY THE OPERATOR; ANY ADDITIONAL INFORMATION
                                        NEEDED TO RUN A PARTICULAR TEST IS REQUESTED BY THE
                                        PROGRAM.

                                        TESTS PERFORMED ARE:

                                                TEST 0  INPUT AND OUTPUT MODULE EXERCISER
                                                TEST 1  INPUT OR OUTPUT MODULE EXERCISER
                                                TEST 2  DAC CALIBRATION
                                                TEST 3  DAC INTERACTION
                                                TEST 4  COUNTER MODULE TEST
                                                TEST 5  ADOS LOGIC TEST
                                                TEST 6  ADOS CALIBRATION
                                                TEST 7  ADOS REPEATABILITY



                                2.0     REQUIREMENTS
                                        ************

                                2.1     EQUIPMENT
                                        ---------

                                        A.  PDP-11 COMPUTER WITH 8K OF MEMORY (OR MORE).
                                        B.  LOCAL OR REMOTE I/O TERMINAL (ASR33, LA30, LA36, RT02)
                                        C.  ICR CONTROLLER AND FILE BOX
                                        D.  TO TEST INPUT MODULES; SOME FORM OF GENERATING AN
                                            INPUT (SUCH AS SWITCHES).
                                        E.  TO TEST OUTPUT MODULES; SOME FORM OF DETECTING AN
                                            OUTPUT (SUCH AS LIGHTS).
                                            NOTE: INPUT MODULES MAY BE CONNECTED TO OUTPUT MODULES
                                        F.  TO TEST D/A MODULES; A MEANS OF MEASURING D/A OUTPUT
                                            (SUCH AS FLUKE METER) AND AN OCILLISCOPE.
                                        G.  TO TEST A/D MODULES; A PRECISION VOLTAGE STANDARD
                                            (SUCH AS AN EDC) IS NEEDED.

                                2.2     STORAGE
                                        -------

                                        THIS PROGRAM OCCUPIES CORE LOCATIONS 000000-32000.
                                        DURING PROGRAM RUN ALL LOCATIONS UP TO 037474 MAY BE USED.
```

```
                                3.0     LOADING PROCEDURE
                                        ******************

                                3.1     LOCAL MODE
                                        ----------

                                        USE STANDARD PDP11 ABSOLUTE LOADER PROCEDURE

                                3.2     REMOTE MODE
                                        -----------

                                        THE ICR11 CONTROLLER DIAGNOSTIC HAS THE CAPABILITY
                                        OF LOADING THIS PROGRAM.  AFTER LOADING THE CONTROLLER
                                        DIAGNOSTIC(MD-11-DZIRA) PLACE DZIRB IN THE HIGH SPEED PAPER
                                        TAPE READER BEFORE GOING REMOTE.  WHEN DZIRA PRINTS:

                                        "RERUN OR LOAD FIELD TEST?"

                                        RESPOND WITH "L" AND THE TAPE WILL LOAD AND START THE TEST AT
                                        SECTION 6.3

                                        ON A MULTIPLE ICR11 FILE SYSTEM DZIRB WILL START TESTING
                                        THAT FILE BOX THE "L" CAME FROM, IF IT IS NESCESSARY TO
                                        TEST ANOTHER BOX A LOCAL START OF 200 MUST BE DONE

                                        THE PDP11 HARDWARE SWITCH REGISTER IS INHIBITED WHILE
                                        RUNNING REMOTE, THEREFORE SWITCHES MAY BE LEFT AS SET
                                        IN DZIRA

                                        THIS METHOD OF LOADING SHOULD BE USED TO LOAD THE FIELD TEST
                                        O N L Y  AND  R E M O T E L Y  O N L Y
                                        - - - -      - - - - - - - - - - - -
```

4.0 STARTING PROCEDURE

4.1 LOCAL METHOD

4.1.1 START ADDRESS 200

LOADING ADDRESS 200 AND STARTING, CLEARS ALL FORMERLY ENTERED INFORMATION (IF ANY) AND TYPES OUT "6.1 FILE BOX TO BE TESTED?" AND WAITS FOR A RESPONSE FROM THE OPERATOR (SEE SEC. 6.1) THEN IT WILL TYPE "6.2 ICR VECTOR ADDRESS?" (SEE SECTION 6.2). NEXT IT WILL TYPE OUT "6.3 TEST NO.?" AND WAITS FOR THE OPERATOR TO SELECT THE TEST HE WISHES TO RUN. WHEN A TEST IS SELECTED, ONE OR MORE ADDITIONAL QUESTIONS WILL BE ASKED (SEE SECTION 6.0).

THIS STARTING PROCEDURE WILL ALWAYS PRODUCE A LOCAL START AND CONTROL.

4.1.2 RESTART ADDRESS 210

LOADING ADDRESS 210 AND STARTING WILL RESTART THE PROGRAM AT SECTION 6.3 RETAING THE PREVIOUSLY ENTERED FILE BOX NUMBER AND VECTOR. START 200 MUST PRECEED RESTART 210.

THIS RESTART ADDRESS WILL ALWAYS FORCE LOCAL CONTROL.

4.2 REMOTE METHOD

LOADING THE DIAGNOSTIC AS IN SECTION 3.2 WILL AUTOMATICALLY START THE DIAGNOSTIC AT SECTION 6.3.

4.3 TRANSFERRING CONTROL REMOTELY

THIS PROGRAM HAS THE CAPABILITY OF TRANSFERRING CONTROL FROM ONE FILE BOX TO ANOTHER REMOTELY. THIS IS DONE BY TYPING ↑X (CONTROL AND X) IN RESPONSE TO "TEST NO" AT THE REMOTE END. ↑X WILL RESPOND WITH:

NEXT FILE TO BE TESTED? X

RESPOND BY TYPING AN OCTAL NUMBER FROM 0 TO 13 FOLLOWED BY A CARRIAGE RETURN. THIS NUMBER REPRESENTS THE FILE BOX WHICH YOU WANT TO TEST NEXT.

SEE SECTION 6.1.1 FOR ERROR MESSAGES

NEXT VECTOR ADDRESS? XXX

RESPOND WITH THE INTERRUPT VECTOR ADDRESS OF THE
NEXT FILE BOX TO BE TESTED.

SEE SECTION 6.2.1 FOR ERROR MESSAGES

THE PROGRAM WILL THEN TYPE:

"TRANSFERRING CONTROL"

AFTER THIS MESSAGE IS PRINTED, POWER DOWN THE FIRST
FILE BOX, REMOVE THE REMOTE TERMINAL. PROCEED TO
THE NEXT FILE BOX POWER IT DOWN, CONNECT THE REMOTE
TERMINAL AND POWER THE FILE BOX BACK UP. THIS WILL
NOW GIVE YOU CONTROL AT THIS FILE BOX. IF "TEST NO.?"
IS NOT DISPLAYED ON THE REMOTE TERMINAL (DUE TO
THE FACT THE REMOTE TERMINAL WAS OFF WHILE THE MESSAGE
WAS DISPLAYED) A CONTROL C (↑C) WILL GET YOU CONTROL.

5.0 SWITCH REGISTER SETTINGS

TEST NO.	SWR	FUNCTION
TEST 0	SW13=1	INHIBIT TYPEOUT
TEST 1	NONE	
TEST 2	SW04-SW13	DAC OUTPUT LEVEL
TEST 3	NONE	
TEST 4+5	SW15=1	HALT ON ERROR (INHIBITED WHEN RUNNING REMOTELY)
	SW14=1	LOOP ON TEST
	SW13=1	INHIBIT ERROR TYPEOUT
	SW11=1	INHIBIT ITERATIONS
TEST 6	SW14-12	SELECT GAIN
	SW11=1	TYPEOUT RESULTS
	SW11=0	DISPLAY RESULTS
	SW07-10	SELECT MUX.
	SW04-06	SELECT CHAN.
TEST 7	NONE	

WHEN RUNNING REMOTELY, TYPING "↑S" WILL DISPLAY THE
"SWITCH REGISTER" AND ALLOW THE USER TO ENTER A NEW
VALUE. THIS SWITCH REGISTER IS CLEARED AT THE BEGINNING
OF EACH TEST.

ALL TESTS "↑C" (CONTROL AND LETTER C) TYPED WILL RETURN
YOU TO MONITOR. RUBOUT DELETES ENTIRE LINE THAT WAS
TYPED.

```
                        6.0     PROGRAM QUESTIONS
                                ******************

                                SINCE THIS PROGRAM HAS NO WAY OF DETERMINING THE PARTICULAR
                                CONFIGURATION OF YOUR ICR11, VARIOUS QUESTIONS WILL BE
                                ASKED WHEN THE ANSWERS ARE REQUIRED BY THE PROGRAM.  SOME
                                QUESTIONS DO HAVE DEFAULT ANSWERS, IF IN DOUBT REFER TO
                                THE SECTION OF THE DOCUMENT THAT EXPLAINS THAT QUESTION.
                                NOTE: SECTIONAL HEADERS (I.E. "6.3" TEST NO.?) REFER
                                YOU TO SECTION OF THE DOCUMENT THAT DESCRIBES THAT QUESTION.
                                TO UTILIZE DEFAULT PARAMETERS, TYPE A CARRIAGE RETURN.

                        6.1     FILE BOX TO BE TESTED?
                                ----------------------

                                RESPOND BY TYPING AN OCTAL NUMBER FROM 0 TO 13
                                FOLLOWED BY A CARRIAGE RETURN. THIS NUMBER REPRESENTS
                                THE FILE BOX WHICH YOU WANT TO TEST

                        6.1.1   ERROR MESSAGE

                                (1) "ILLEGAL NUMBER" - "A NON-OCTAL OR NUMBER GREATER
                                    THAN 13 WAS TYPED

                                (2) "NON-EXISTENT FILE BOX" - NO SLAVE-SYNC RESPONSE
                                    CAME FROM FILE BOX REQUESTED

                        6.2     ICR VECTOR ADDRESS?
                                -------------------

                                RESPOND WITH THE INTERRUPT VECTOR ADDRESS OF
                                THE FILE BOX UNDER TEST FOLLOWED BY A CARRIAGE
                                RETURN.

                        6.2.1   ERROR MESSAGE

                                (1) "ILLEGAL NUMBER" IS TYPED IF THE ADDRESS YOU TYPED
                                    IS NOT WITHIN THE VECTOR RANGE "234-774"

                                (2) "FILE BOX INTERRUPTED AT XXXXXX -- CHECK JUMPERS"

                                    A MAINTENANCE INTERRUPT WAS FORCED AND THE INTERRUPT
                                    OCCURRED THRU XXXXXX.  CHECK JUMPERS TO INSURE PRO-
                                    PER VECTOR WAS GIVEN.  CONTINUE WILL  RE-ASK THE
                                    QUESTION

                                (3) "FILE BOX DID NOT INTERRUPT - FATAL"

                                    AN INTERRUPT WAS TRIED TO CHECK VECTOR ADDRESS, AND
                                    FILE BOX DID NOT INTERRUPT.  CANNOT CONTINUE FROM
                                    HERE UNTIL PROBLEM IS RESOLVED
```

6.3 TEST NO.?

RESPOND BY TYPING A NUMBER FROM 0 TO 7 FOLLOWED BY A
CARRIAGE RETURN. SEE SECTION 7 FOR MORE DETAILED
INFORMATION ABOUT TESTS.
ALL TESTS MAY BE TERMINATED BY TYPING "↑C", WHICH
WILL BRING YOU BACK TO SECTION 6.3 OR "↑P" WHICH WILL
PRINT THE PRESENT LINE ERROR COUNT,CLEAR THE ERROR COUNT,
AND RETURN TO SECTION 6.3

TEST NO	TEST	EXIT METHOD
0	INPUT AND OUTPUT MODULE EXERCISER	↑C
1	INPUT OR OUTPUT MODULE EXERCISER	↑C
2	DAC CALIBRATION	↑C
3	DAC ITERACTION	↑C
4	COUNTER MODULE TEST	↑C
5	A/D LOGIC TEST	↑C
6	A/D CALIBRATION	↑C
7	A/D REPEATABILITY	↑C

6.3.1 DEFAULT

THERE IS NO DEFAULT ANSWER FOR THIS QUESTION.

6.3.2 ERROR MESSAGE

"NO SUCH TEST" IS TYPED IF AN ILLEGAL TEST NUMBER IS TYPED.

6.4 INPUT?

RESPOND TO THIS QUESTION BY TYPING THE ADDRESS(ES)
OF THE INPUT MODULE(S) YOU WISH TO EXERCISE
FOLLOWED BY A CARRIAGE RETURN.
INPUT EXPECTED IS IN OCTAL FORM.
EXAMPLE:

20<CR>	INPUT DATA FROM MODULE IN ADDRESS 20, FILE 0
20,24<CR>	INPUT DATA FROM MODULES IN ADDRESS 20 AND 24, FILE 0
20:26,34<CR>	INPUT DATA FROM MODULES IN ADDRESSES 20,22,24,26, AND 34, FILE 0

6.4.1 DEFAULT

NO DEFAULTS.

```
                        6.4.2   ERROR MESSAGES

                                (1)  "TOO MANY ADDRESSES - RETYPE" IS TYPED IF YOU
                                     ENTERED MORE THAN 16 INPUT MODULE ADDRESSES (IN SAME FILE)
                                     FOR TEST 0 OR MORE THAN ONE ADDRESS FOR TEST 1.

                                (2)  "LAST CHARACTER TYPED NOT AN OCTAL DIGIT - RETYPE"
                                     INPUT MUST MUST BE AN OCTAL DIGIT.

                                (3)  "INVALID ADDRESS:  XXXXXX"
                                     THE PROGRAM ADDED THE BASE ADDRESS 171000 TO THE
                                     ADDRESS YOU TYPED AND TRIED TO ADDRESS IT BUT GOT
                                     NO SLAVE-SYNC RESPONSE FROM IT AND TRAPPED.  YOU
                                     WILL BE ASKED TO RE-TYPE ALL ADDRESSES.

                                (4) "ADDRESS XXXXXX NOT WITHIN FILE" - THE
                                    ADDRESS MAY WELL EXIST BUT IT IS NOT WITHIN
                                    THE MODULE RANGE FOR THE FILE UNDER TEST
                                    YOU WILL BE ASKED TO RE-ENTER ALL ADDRESSES
                                    (REFERENCE 9.1 FOR ADDRESS RANGES).

                        6.5     OUTPUT?
                                -------

                                RESPOND BY TYPING THE ADDRESS(ES) OF THE OUTPUT
                                MODULE(S) YOU WISH TO EXERCISE FOLLOWED BY
                                A CARRIAGE RETURN.  EXAMPLE: SEE 6.4 FOR EXAMPLES.
                                INPUT EXPECTED IS IN OCTAL FORM.

                        6.5.1   DEFAULT

                                NO DEFAULTS.

                        6.5.2   ERROR MESSAGES

                                (1)  "TOO MANY ADDRESSES - RETYPE" IS TYPED IF YOU
                                     ENTERED MORE THAN 16 INPUT MODULE ADDRESSES (IN SAME FILE)
                                     FOR TEST 0, OR MORE THAN ONE ADDRESS FOR TEST 1.

                                (2)  "LAST CHARACTER TYPED NOT AN OCTAL DIGIT - RETYPE"
                                     INPUT MUST BE AN OCTAL DIGIT.

                                (3)  "INVALID ADDRESS:  XXXXXX"
                                     THE PROGRAM ADDED THE BASE ADDRESS 171000 TO THE
                                     ADDRESS YOU TYPED AND TRIED TO ADDRESS IT BUT GOT
                                     NO SLAVE-SYNC RESPONSE FROM IT AND TRAPPED.  YOU
                                     WILL BE ASKED TO RE-TYPE ALL ADDRESSES.

                                (4) "ADDRESS XXXXXX NOT WITHIN FILE" - THE
                                    ADDRESS MAY WELL EXIST BUT IT IS NOT WITHIN
                                    THE MODULE RANGE FOR THE FILE UNDER TEST
                                    YOU WILL BE ASKED TO RE-ENTER ALL ADDRESSES
                                    (REFERENCE 9.1 FOR ADDRESS RANGES).
```

```
                        6.6     CNTR ADDR?
                                ----------

                                RESPOND BY THE ADDRESS(ES) OF THE
                                COUNTER MODULE(S) TO BE TESTED, FOLLOWED BY A CARRIAGE
                                RETURN.  EXAMPLE: SEE 6.4 FOR EXAMPLES.
                                INPUT EXPECTED IS IN OCTAL FORM.

                        6.6.1   DEFAULT

                                NO DEFAULTS.

                        6.6.2   ERROR MESSAGES

                                (1)  "TOO MANY ADDRESSES - RETYPE" IS TYPED IF YOU
                                     ENTER MORE THAN 16 ADDRS.

                                (2)  "LAST CHARACTER TYPED NOT AN OCTAL DIGIT - RETYPE"
                                     INPUT MUST BE AN OCTAL DIGIT.

                                (3)  "INVALID ADDRESS:  XXXXXX"
                                     THE PROGRAM ADDED THE BASE ADDRESS 171000 TO THE
                                     ADDRESS YOU TYPED AND TRIED TO ADDRESS IT BUT GOT
                                     NO SLAVE-SYNC RESPONSE FROM IT AND TRAPPED.  YOU
                                     WILL BE ASKED TO RE-TYPE ALL ADDRESSES.

                                (4) "ADDRESS XXXXXX NOT WITHIN FILE" - THE
                                    ADDRESS MAY WELL EXIST BUT IT IS NOT WITHIN
                                    THE MODULE RANGE FOR THE FILE UNDER TEST
                                    YOU WILL BE ASKED TO RE-ENTER ALL ADDRESSES
                                    (REFERENCE 9.1 FOR ADDRESS RANGES).

                        6.7     DAC?
                                ----

                                RESPOND BY TYPING THE ADDRESS OF THE DAC YOU WISH TO
                                EXERCISE FOLLOWED BY A CARRIAGE RETURN.
                                INPUT EXPECTED IS IN OCTAL FORM.

                        6.7.1   DEFAULT

                                NO DEFAULTS.

                        6.7.2   ERROR MESSAGES

                                (1)  "TOO MANY ADDRESSES - RETYPE" IS TYPED IF YOU
                                     TRY TO ENTER MORE THAN ONE ADDRESS.

                                (2)  "LAST CHARACTER TYPED NOT AN OCTAL DIGIT - RETYPE"
                                     INPUT MUST BE AN OCTAL DIGIT.

                                (3)  "INVALID ADDRESS:  XXXXXX"
                                     THE PROGRAM ADDED THE BASE ADDRESS 171000 TO THE
                                     ADDRESS YOU TYPED AND TRIED TO ADDRESS IT BUT GOT
                                     NO SLAVE-SYNC RESPONSE FROM IT AND TRAPPED.  YOU
                                     WILL BE ASKED TO RE-TYPE ALL ADDRESSES.
```

463

(4) "ADDRESS XXXXXX NOT WITHIN FILE" - THE
ADDRESS MAY WELL EXIST BUT IT IS NOT WITHIN
THE MODULE RANGE FOR THE FILE UNDER TEST
YOU WILL BE ASKED TO RE-ENTER ALL ADDRESSES
(REFERENCE 9.1 FOR ADDRESS RANGES).

6.8 AD05?

RESPOND BY TYPING THE ADDRESS OF THE AD05, OR THE ADDRESS
OF THE AD05 CONNECTED TO THE AD07 YOU WISH TO EXERCISE,
FOLLOWED BY A CARRIAGE RETURN.
INPUT EXPECTED IS IN OCTAL FORM.

6.8.1 DEFAULT

NO DEFAULTS.

6.8.2 ERROR MESSAGES

(1) "TOO MANY ADDRESSES - RETYPE" IS TYPED IF YOU
TRIED TO ENTER MORE THAN ONE ADDRESS.

(2) "LAST CHARACTER TYPED NOT AN OCTAL DIGIT - RETYPE"
INPUT MUST BE AN OCTAL DIGIT.

(3) "INVALID ADDRESS: XXXXXX"
THE PROGRAM ADDED THE BASE ADDRESS 171000 TO THE
ADDRESS YOU TYPED AND TRIED TO ADDRESS IT BUT GOT
NO SLAVE-SYNC RESPONSE FROM IT AND TRAPPED. YOU
WILL BE ASKED TO RE-TYPE ALL ADDRESSES.

6.9 DELAY (IN MILLISEC)?

RESPOND BY TYPING THE DELAY TIME YOU WISH THE PROGRAM
TO USE IN BETWEEN OUTPUTTING DATA TO OUTPUT MODULES,
FOLLOWED BY A CARRIAGE RETURN. FRACTIONAL TIMES NOT
ALLOWED. NOTE 1 SECOND EQUALS 1000 MILLISEC.
A SECONDARY DELAY TIME IS AVAILABLE FOR USE IN TEST 0.
SEE SECTION 8 FLAG MODE OPERATION.
INPUT EXPECTED IS IN DECIMAL FORM, MAXIMUM DELAY TIME=10
SECONDS.

6.9.1 DEFAULT

(1) FOR ANY START OR RESTART - 3 MILLISEC IS DEFAULT SINCE
MOST (BUT NOT ALL) OUTPUT MODULES HAVE A RESPONSE
TIME OF APPROXIMATELY 3 MILLISECONDS.
NOTE: DELAY TIMES ARE CALCULATED USING MACHINE
INSTRUCTION TIME LOOPS - THEY MAY VARY BETWEEN
PROCESSORS.

6.10 PAT MOD, PAT?

THIS IS A TWO PART QUESTION. RESPOND BY TYPING A SINGLE DIGIT REPRESENTING THE PATTERN MODIFIER, FOLLOWED BY A "," (COMMA), FOLLOWED BY THE STARTING PATTERN YOU WISH TO USE (0 TO 6 DIGITS) FOLLOWED BY A CARRIAGE RETURN. INPUTS EXPECTED ARE IN OCTAL FORM.
EXAMPLE:

6.10 PAT MOD, PAT ? 0,100000<CR>
WOULD GIVE AN INCREMENTING PATTERN,WITH STARTING PATTERN OF "100000".

PATTERN MODIFIER	FUNCTION
0	INCREMENT PATTERN
1	DECREMENT PATTERN
2	NO CHANGED OF PATTERN
3	ROTATE LEFT PATTERN
4	ROTATE RIGHT PATTERN
5	RANDOM PATTERN
6	ARITH. SHIFT LEFT PATTERN
7	COMPLEMENT PATTERN

6.10.1 DEFAULT

(1) FOR ANY START OR RESTART - STARTING PATTERN OF ALL ZEROS, PATTERN MODIFIER OF ZERO.

6.11 INPUT OR OUTPUT (I OR O)?

THIS QUESTION IS ASKED BY TEST 1 WHERE ONLY AN INPUT OR AN OUTPUT MODULE IS EXERCISED. RESPOND BY TYPING AN "I" FOR INPUT MODULE, OR "O" FOR OUTPUT MODULE. A CARRIAGE RETURN IS REQUIRED AFTER THE I OR O.

6.11.1 DEFAULTS

NONE

6.11.2 ERROR MESSAGES

IF NEITHER AN "I" NOR AN "O" WAS TYPED THAN THE QUESTION WILL BE RETYPED.

6.12 UNI OR BI-POLAR (U OR B)

RESPOND BY TYPING A "U" OR A "B" FOLLOWED BY A CARRIAGE RETURN. UNIPOLAR REFERS TO AN UNSIGNED A/D. THE STANDARD A/D IS BIPOLAR, THAT IS, ITS RESULTS ARE SIGNED.

6.12.1 DEFAULTS

ON CARRIAGE RETURN, IT IS ASSUMED BIPOLAR.

```
573
574                          6.12.2  ERROR MESSAGES
575
576                                  NONE
577
```

578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633

6.13 GAIN?

THIS QUESTION IS ASKED BY TEST 7. RESPOND BY TYPING THE
GAIN YOU WISH THE SAMPLES TO BE TAKEN AT.
LEGAL GAINS ARE: 1,2,10,20,50,100,200, AND 1000

6.13.1 DEFAULTS

NO DEFAULTS.

6.13.2 ERROR MESSAGES

"NO SUCH GAIN" IS TYPED IF THE GAIN YOU TYPED ISN'T LEGAL.

6.14 CHANS (SC,EC)?

THIS QUESTION IS ASKING YOU FOR THE OCTAL CHANNELS YOU WISH THE
SAMPLES TO BE TAKEN ON. SC REPRESENTS THE STARTING
CHANNEL AND EC REPRESENTS THE END CHANNEL. CHANNELS MUST
BE SAMPLED IN CONSECUTIVE ORDER.
CPU BASED ON CORE AVAILIBLE TO STORE SAMPLES AWAY IN.
RESPOND BY TYPING THE STARTING CHANNEL NUMBER, FOLLOWED BY
A COMMA, FOLLOWED BY THE END CHANNEL NUMBER, FOLLOWED
BY A CARRIAGE RETURN. TO SAMPLE ONLY ONE CHANNEL,
SIMPLY TYPE THAT CHANNEL NUMBER FOLLOWED BY A CARRIAGE
RETURN.

6.14.1 DEFAULTS

(1) AT LOAD AND START AT ADDR. 200, CHAN. 0 WILL BE SELECTED.

(2) AT ANY OTHER TIME, PREVIOUSLY TYPED CHANNEL(S).

6.14.2 ERROR MESSAGES

"ERROR! START CHAN > END CHAN." IS TYPED
WHEN THAT CONDITION IS TRUE.

"ERROR! NO SUCH CHAN" IS TYPED IF THE CHAN NUMBER IS TOO
LARGE.

6.15 EXPECTED AVERAGE?

RESPOND BY TYPING THE AVERAGE (IN OCTAL) WHICH YOU
EXPECT THE SAMPLES TO AVERAGE, FOLLOWED BY A CARRIAGE RETURN.
THIS QUESTION MAY NEED NOT TO BE ANSWERED - SEE WRITE FOR
TEST-7.

6.15.1 DEFAULT

DEFAULT OF 0000 OR PREVIOUSLY TYPED DATA.

6.15.2 ERROR MESSAGES

```
                                "NUMBER TOO LARGE-MAX=7777" SELF EXPLAINITORY THE A005
                                IS ONLY A 12 BIT CONVERTER.
```

```
                        6.16    TOLERANCE?
                                ----------

                                RESPOND BY TYPING THE TOLERANCE FOLLOWED BY A CARRIAGE
                                RETURN. A TOLERANCE OF ZERO WILL FORCE TYPEOUT OF
                                THE RESULTS OF TEST 7. TOLERANCE MAY BE ANY NUMBER,
                                HOWEVER, IT IS RECOMMENDED YOU READ TEST 7, AND EXAMINE
                                THE SPECIFICATIONS FOR THE ADDS TO DETERMINE THIS NUMBER.

                        6.16.1  DEFAULTS

                                ZERO OR PREVIOUSLY TYPED TOLERANCE.

                        6.16.2  ERROR MESSAGES

                                NONE

                        7.0     TEST DESCRIPTIONS
                                **** ************

                        7.1     TEST 0                  INPUT AND OUTPUT MODULE EXERCISER

                                                        THIS TEST IS DESIGNED TO EXERCISE
                                                        UP TO 16 OUTPUT MODULES OR 16
                                                        INPUT MODULES OR ANY COMBINATION
                                                        OF 16 I/O MODULES.  IT (1) OUTPUTS THE
                                                        PATTERN TO ALL OUTPUT MODULES,
                                                        (2) DELAYS THE SPECIFIED DELAY
                                                        TIME, (3) MODIFIES THE PATTERN,
                                                        (4) SAMPLES THE INPUT MODULES
                                                        FOR CHANGE OF DATA, IF A CHANGE
                                                        OF DATA HAS OCCURRED, IT STARTS
                                                        TYPEOUT OF THE CHANGE.
                                                        IF AN INPUT MODULE INTERRUPTS, ITS
                                                        CHANGE OF DATA IS TYPED ALONG WITH
                                                        ITS GENERIC CODE.  THE GENERIC
                                                        CODE FOR A STANDARD INPUT MODULE
                                                        IS "3".
                                                        (5) IT NEXT DELAYS A SECOND TIME
                                                        SPECIFIED BY THE SECONDARY DELAY
                                                        TIME (IF ANY).  SECONDARY DELAY TIME
                                                        SET BY "↑D" USED FOR EXERCISING
                                                        THE M6870 SINGLE SHOT OUTPUT MODULE.
                                                        (6) REPEAT STEPS 1-5
                                                        IF "↑J" WAS TYPED (SEE SECTION 8.5)
                                                        INPUT AND OUTPUT MODULES WILL BE ASSUMED
                                                        CONNECTED.  THE ONLY TYPEOUT WILL BE
                                                        IF THE DATA SENT OUT DOESN'T MATCH
                                                        THE DATA RECEIVED.

                7.1.1   RUN TIME TEST 0

                        INDEFINITE - RUN TERMINATED BY OPERATOR.
```

7.2 TEST 1 INPUT OR OUTPUT MODULE EXERCISER

(1) FOR INPUT MODULES:
READS THE INPUT MODULE CONTINUOUSLY
AND DISPLAYS ITS CONTENTS IN DISPLAY
REGISTER (11/45). IF YOU
HAVE AN 11/05 PROCESSOR YOU MUST USE
TEST 0.

(2) FOR OUTPUT MODULES:
TAKES THE CONTENTS OF THE SWITCH
REGISTER AND SENDS IT TO THE OUTPUT
MODULE, DELAYS SPECIFIED DELAY TIME
AND REPEATS.

7.2.1 RUN TIME TEST 1

INDEFINITE - RUN TERMINATED BY OPERATOR.

7.3 TEST 2 DAC CALIBRATION

OUTPUTS THE CONTENTS OF THE SWITCH
REGISTER TO ALL FOUR CHANNELS OF
THE DAC SPECIFIED IN ORDER TO MAINTAIN
A CALIBRATION LEVEL.

7.3.1 RUN TIME TEST 2

INDEFINITE - RUN TERMINATED BY OPERATOR.

7.4 TEST 3 DAC INTERACTION

OUTPUTS A RAMP TO ALL FOUR CHANNELS
TO THE SPECIFIED DAC. THESE RAMPS
ARE "OUT OF PHASE" WITH EACH OTHER
SO THAT INTERACTION AND DUAL ADDRESSING
CAN BE TESTED.

7.4.1 RUN TIME TEST 3

INDEFINITE - RUN TERMINATED BY OPERATOR.

7.5 TEST 4 COUNTER MODULE TEST

THIS TEST CHECKS OUT BASIC LOGIC
FUNCTIONS OF THE COUNTER MODULE.
THE USER MUST JUMPER TP3 AND
TP4 ON THE W7440.

7.5.1 RUN TIME TEST 4

SHORT PASS (SW11=1) APP. 15 SEC.
LONG PASS (SW11=0) APP. 2 MIN.
FIRST PASS IS ALWAYS A SHORT PASS.

```
                        7.6     TEST 5                  A/D LOGIC TEST

                                                        THIS TEST CHECKS OUT BASIC LOGIC
                                                        FUNCTIONS OF THE A/D.

                        7.6.1   RUN TIME TEST 5

                                SHORT PASS (SW11=1) APP. 15 SEC.
                                LONG PASS (SW11=0) APP. 2 MIN.
                                FIRST PASS IS ALWAYS A SHORT PASS.

                        7.7     TEST 6                  A/D CALIBRATION

                                                        THIS TEST ALLOWS THE USER TO CALIBRATE
                                                        THE ADOS.

                                                        SW11=1 WILL PRINT RESULTS LOCAL OR REMOTE
                                                        (SWR=4000) FOR EXAMINATIO.

                        7.7.1   RUN TIME TEST 6

                                INDEFINITE - RUN TERMINATED BY OPERATOR.

                        7.8     TEST 7

                                                        (1) REPEATIBILITY

                                                            THIS TEST ALLOWS THE USER TO TEST THE REPEATIBILITY
                                                            NUMBER OF CHANNELS AT ANY GAIN AND INPUT VOLTAGE.
                                                            THE TEST MAY OR MAY NOT PRINT OUT A TABLE
                                                            (SEE EXAMPLE OF PRINT-OUT) DEPEDING ON WHETHER AN ER
                                                            OCCURED OR IF FORCED TYPEOUT IS DESIRED. 256 SAMPLES
                                                            ARE TAKEN PER CHANNEL PER PASS. "REPEAT" IS TYPED
                                                            AT THE BEGINNING OF EACH PASS.
                                                            WHEN THIS TEST IS SELECTED, THE ADOS ADDR OF THE
                                                            A/D SUBSYSTEM IS REQUESTED ALONG WITH WHAT CHANNELS
                                                            YOU WISH TO SAMPLE, THE GAIN YOU WISH TO USE, FOLLOW
                                                            BY THE EXPECTED AVERAGE (OPTIONAL) AND THE TOLERANCE

                                                            IF FORCED TYPEOUT IS DESIRED, A TOLERANCE OF ZERO
                                                            SHOULD BE TYPED. IF THE EXPECTED AVERAGE IS
                                                            KNOWN RUN ONE PASS AT ANY EXPECTED AVERAGE AND
                                                            THE CURRENT AVERAGE WILL BE TYPED OUT.

                                                        THE TEST OPERATES IN THE FOLLOWING MANNER:
                                                        (1) IT TAKES 256 SAMPLES ON EACH CHANNEL SPECIFIED.
                                                        (2) IT COMPUTES HIGH,LOW, AND AVERAGE OF SAMPLES FOR EAC
                                                            CHANNEL
                                                        (3) IT COMPARES THE HIGH,LOW, AND AVERAGE AGAINST THE
                                                            EXPECTED AVERAGE YOU TYPED. IF THERE IS ANY ERROR
                                                            THE ERROR WILL BE TYPED FOR THAT PARTICULAR CHANNEL,
                                                            THERE ARE NO ERRORS, THERE WILL BE NO TYPEOUT
                                                            EXCEPT FOR "REPEAT" AT THE BEGINNING OF THE NEXT PAS
                                                        (4) IF A TOLERANCE OF ZERO IS SPECIFIED, A FORCED TYPEOU
                                                            WILL OCCUR OF THE RESULTS OF ALL CHANNELS SPECIFIED.
```

7.8.1 RUN TIME TEST 7

DEPENDENT ON NUMBER OF CHANNELS TO BE SAMPLED. 1 PASS FOR 1 CHANNEL TAKES APPROXIMATELY 2 MINUTES. IF MULTIPLE CHANNELS, TIME BETWEEN TYPEOUTS SHOULD NOT EXCEED 2 MINUTES.

7.8.2 EXAMPLE OF TYPEOUT

(LOCAL MODE)

LO	-5	-4	-3	-2	-1	AV	+1	+2	+3
0000	0000	0000	0000	0004	0006	0240	0006	0000	00

(REMOTE MODE)

LO	-1	AV	+1	HI
0000	0000	0000	0000	0000

8.0 FLAG MODE OPERATION

8.01 SUMMARY

FLAG	TESTS AFFECTED	SECTION
↑E	ALL	8.1
↑N	ALL	8.2
↑D	0	8.3
↑L	7	8.4
↑J	0	8.5
↑P	ALL	8.6
↑R	ALL	8.7
↑S	ALL	8.8

ALL FLAG MODES ARE ENABLED BY TYPING THE ASSOCIATED LETTER AND THE CONTROL KEY TOGETHER AT "6.3 TEST NO?". AFTER THE FLAG IS ACTED UPON, PROGRAM CONTROL WILL RETURN TO "6.3 TEST NO.?".

8.1 ↑E EXPERT MODE

WHEN "↑E" IS TYPED EXPERT MODE WILL BE ENABLED. WHEN THE PROGRAM IS OPERATING IN "EXPERT MODE" NO QUESTIONS WILL BE TYPED, ONLY THE QUESTION MARKS.

8.2 ↑N NOVICE MODE

WHEN "↑N" IS TYPED, "EXPERT MODE" WILL BE DISABLED. ALL QUESTIONS WILL BE TYPED.

8.3 ↑D SECONDARY DELAY

"↑D" IS TYPED IN ORDER TO ENTER A SECONDARY DELAY USED IN TEST 0. THIS DELAY IS ONLY NEEDED FOR TESTING A SINGLE SHOT MODULE WHEN RUNNING TEST0 WITH "↑J" OPTION.

```
                        8.4     ↑L      LINE PRINTER OPTION

                                WHEN A LINE PRINTER IS AVAILABLE FOR USE AS AN OUTPUT
                                DEVICE FOR TEST 7, YOU MAY SELECT OUTPUT TO GO TO IT
                                BY TYPING "↑L".

                                WHEN THE TEST IS STARTED AT LOC 200, THE PROGRAM WILL TYPE
                                "LINE PRINTER OPTION AVAILABLE" IF IT DETECTS A LINE PRINTER
                                ON THE SYSTEM.  IF "↑L" IS TYPED, AND NO LINE PRINTER IS
                                AVAILABLE, THE COMMAND WILL BE IGNORED; IF "↑L" IS HONORED,
                                THE PROGRAM WILL TELL YOU TO MAKE THE PRINTER READY.  TO DO
                                THIS, MAKE SURE ITS POWER IS ON AND IT IS SELECTED.

                                IF RUNNING REMOTELY THIS OPTION IS DISABLED.


                        8.5     ↑J      CONNECTED MODE FOR TEST 0

                                "↑J" INDICATES TO THE PROGRAM THAT ALL INPUT AND OUTPUT
                                MODULES EXERCISED BY TEST 0 ARE CONNECTED TO EACH OTHER
                                FOR TEST.  NO TYPE OUT WILL OCCUR IF THE DATA SENT OUT
                                MATCHES THE DATA RECEIVED.

                        8.6     ↑P      LINE ERROR SUMMARY

                                "↑P" WILL PRINT THE CURRENT LINE ERROR COUNT, ZERO THE
                                ERROR COUNT AND THEN RETURN TO "TEST NO.?"


                        8.7     ↑R      GOING REMOTE

                                "↑R" INDICATES THAT THE TEST IS NOW
                                GOING TO BE PERFORMED REMOTELY, LOCAL
                                KEYBOARD IS DISABLED EXCEPT FOR ↑C.  LOCAL
                                PRINTER WILL CONTINUE TO TYPE FOR
                                HARD COPY.


                        8.9     ↑S      REMOTE SWITCH REGISTER

                                WHEN RUNNING REMOTELY "↑S" IS USED TO CHANGE THE "SWITCH
                                REGISTER". TYPING "↑S" THE PROGRAM WILL RESPOND WITH:

                                SWR=123456  NEW SWR =

                                TYPE IN A NEW VALUE FOLLOWED BY A <CR>.

                        8.10    ↑X TRANSFER CONTROL

                                "↑X" IS USED TO TRANSFER CONTROL FROM ONE FILE BOX
                                TO ANOTHER WHEN RUNNING REMOTELY ON A MULTIPLE FILE
                                BOX SYSTEM.  SEE SECTION 4.3 FOR MORE INFORMATION.
```

```
                                9.0     MISCELLANEOUS

                                9.1     FILE BOX ASSIGNMENT

                                        FILE BOX        ICSR    ICAR    MODULE RANGE    VECTOR ADDRESS

                                           0            171776  171774  171000-171036       234
                                           1            171766  171764  171040-171076       -----
                                           2            171756  171754  171100-171136         ↑
                                           3            171746  171744  171140-171176         ↑
                                           4            171736  171734  171200-171236   ASSIGNABLE
                                           5            171726  171724  171240-171276   IN FLOATING
                                           6            171716  171714  171300-171336   VECTOR AREA
                                           7            171706  171704  171340-171376         ↑
                                          10            171676  171674  171400-171436         ↑
                                          11            171666  171664  171440-171476         ↑
                                          12            171656  171654  171500-171536         ↑
                                          13            171646  171644  171540-171576       -----

                                9.2     ICR POWER LOSS/OR POWERING DOWN ICR

                                        IF THE ICR EVER LOSES POWER EITHER
                                        THRU POWER LOSS OR OPERATOR INTERVENTION
                                        THE PROGRAM WILL RESPOND WITH:

                                        "ICR POWER LOSS"

                                        ONCE POWER IS RETURNED, THE PROGRAM
                                        WILL TYPE:

                                        "RESTARTING FROM ICR POWER LOSS"
                                        AND START AT 6.3

                               10.      LISTING
                                        %


                                        .MCALL  .HEADER,.SETUP,.$POWER,.$TYPE,.EQUAT,.$CMTAG,.$ERRTYP,.$TYPOCT
                                        .MCALL  .$CATCH,.$TRAP,.TRMTRP,SETTRAP,.$READ,.$RDDEC,.$ERROR,.$SCOPE
                                        .MCALL  .$SB2D,.$DB2D,.$SAVE,.$RAND
            164000                              $SWR=164000

000000                                  .HEADER ↑/MAINDEC-11-DZIRB-A/, 1975 ↑/DAN DEKNIS/
                                .TITLE  MAINDEC-11-DZIRB-A
                                ;*COPYRIGHT (C) 1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY DAN DEKNIS
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
                                ;*
            000001              $TN=1

000000                                  .$CATCH START
```

```
                                .SBTTL  TRAP CATCHER

000000                                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
000174                                  .=174
000174  000000                  DISPREG: .WORD  0              ;;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:   .WORD  0              ;;SOFTWARE SWITCH REGISTER

                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  001726                  JMP     @#START        ;;JUMP TO STARTING ADDRESS OF PROGRAM

                                                ;LOCATIONS 162-172 ARE USED IF PROGRAM
                                                ;WAS LOADED REMOTELY, THE CONTROLLER
                                                ;DIAGNOSTIC (MD-11-DZIRA) WILL LOAD THESE
                                                ;LOCATIONS WITH INFORMATION FROM ICR THAT
                                                ;LOADED DZIRB

000162                                  .=162

000162  000000                  ICRVEC: 0              ;VECTOR OF ICR
000164  000000                  REMFF:  0              ;INDICATOR OF HOW PROGRAM WAS LOADED
000166  000000                  ICARLD: 0              ;ICAR OF ICR OF CONTROLLER TEST
000170  000000                  ICSRLD: 0              ;ISCR "   "  "          "
000172  000000                  ICSMLD: 0              ;MODULE

000210                                  .=210
000210  000137  003610                  JMP     @#RSTART

000214                                  .EQUAT

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776         ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774         ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772         ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570         ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570         ;;HARDWARE DISPLAY REGISTER

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

```
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

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

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                          BIT15=  100000
040000                          BIT14=  40000
020000                          BIT13=  20000
010000                          BIT12=  10000
004000                          BIT11=  4000
002000                          BIT10=  2000
001000                          BIT09=  1000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
000020                          BIT04=  20
000010                          BIT03=  10
000004                          BIT02=  4
000002                          BIT01=  2
```

```
1080            000001                  BIT00=  1
1081                                    .EQUIV  BIT09,BIT9
1082                                    .EQUIV  BIT08,BIT8
1083                                    .EQUIV  BIT07,BIT7
1084                                    .EQUIV  BIT06,BIT6
1085                                    .EQUIV  BIT05,BIT5
1086                                    .EQUIV  BIT04,BIT4
1087                                    .EQUIV  BIT03,BIT3
1088                                    .EQUIV  BIT02,BIT2
1089                                    .EQUIV  BIT01,BIT1
1090                                    .EQUIV  BIT00,BIT0
1091
1092                                    ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1093             000004                 ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
1094             000010                 RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1095             000014                 TBITVEC=14              ;;"T" BIT
1096             000014                 TRTVEC= 14              ;;TRACE TRAP
1097             000014                 BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1098             000020                 IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1099             000024                 PWRVEC= 24              ;;POWER FAIL
1100             000030                 EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1101             000034                 TRAPVEC=34              ;;"TRAP" TRAP
1102             000060                 TKVEC=  60              ;;TTY KEYBOARD VECTOR
1103             000064                 TPVEC=  64              ;;TTY PRINTER VECTOR
1104             000240                 PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1105  000214                                    .$CMTAG
```

```
                                        STARS
                                        ;*****************************************************************

                                        .SBTTL  COMMON TAGS

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

001100                                          .=1100
001100                                  $CMTAG:                 ;;START OF COMMON TAGS
001100  000000                          $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
001102  000                             $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
001103  000                             $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
001104  000000                          $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                          $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                          $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                          $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
001114  000                             $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
001115  001                             $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                          $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                          $GDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                          $BDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                          $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
001126  000000                          $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
001130  000000                                  .WORD   0       ;;RESERVED--NOT TO BE USED
001132  000000                                  .WORD   0
001134  000000                                  .WORD   0
001136  177570                          SWR:    .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
001140  177570                          DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
001142  177560                          $TKS:   177560          ;;TTY KBD STATUS
001144  177562                          $TKB:   177562          ;;TTY KBD BUFFER
001146  177564                          $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
001150  177566                          $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
001152  000                             $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001153  002                             $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001154  012                             $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001155  000                             $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001156  000000                          $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
001160  077                             $QUES:  .ASCII  /?/     ;;QUESTION MARK
001161  015                             $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001162  000012                          $LF:    .ASCIZ  <12>    ;;LINE FEED
```

```
                                STARS
                                ;*****************************************************************

                                .SBTTL  ERROR POINTER TABLE

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

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


001164                          $ERRTB:

                                        ;ADDS ERRORS
                                        ;ERROR 1 ADDRESS ABILITY
001164  026561                          EM1                     ;NO DATA XFER
001166  023250                          DH1                     ;ERROR   ADDS
                                                                ;  PC    ADDR
001170  001476                          DT1                     ;$ERRC,  ADCSR,0
001172  001474                          DF0                     ;0

                                        ;ERROR 2 "BIT EXERCISER"
001174  026576                          EM2                     ;XFER DATA ERROR
001176  023254                          DH2                     ;ERROR   A/D    GOOD    BAD
                                                                ;  PC    ADDR    DATA    DATA
001200  001500                          DT2                     ;$ERRPC, ADCSR,  $GDDAT, $BDDAT;0
001202  001474                          DF0                     ;0

                                        ;ERROR 3 CONVERT BIT

001204  026616                          EM3                     ;CNVT BIT NOT SET
001206  023250                          DH1                     ;ERROR   A/D
001210  001476                          DT1                     ;  PC    ADDR
001212  001474                          DF0                     ;$ERRPC, ADCSR,0

                                        ;ERROR 4 CONVERT BIT

001214  026633                          EM4                     ;CNVT BIT NOT CLEAR
001216  023250                          DH1                     ;ERROR   A/D
001220  001476                          DT1                     ;  PC    ADDR
001222  001474                          DF0                     ;$ERRPC, ADCSR,0

                                        ;ERROR 5 READ C/R BIT
001224  026652                          EM5                     ;DATA REG ERROR
001226  023250                          DH1                     ;ERROR   A/D
001230  001476                          DT1                     ;  PC    ADDR
001232  001474                          DF0                     ;$ERRPC, ADCSR,0

                                        ;ERROR 6 ADDS INTERRUPT
```

```
1203  001234  026671                  EM6                     ;NO ADDS INTR
1204  001236  023250                  DH1                     ;ERROR   A/D
1205  001240  001476                  DT1                     ;  PC    ADDR
1206  001242  001474                  DF0                     ;$ERRPC, ADCSR,0
1207
1208                                  ;ERROR 7 ADDS ADDR OR GENERIC CODE
1209
1210  001244  026706                  EM7                     ;ICAR BAD ON INT
1211  001246  023264                  DH3                     ;ERROR   MODULE    ICAR
1212  001250  001500                  DT2                     ;  PC    ADDR      S/B    WAS
1213  001252  001474                  DF0                     ;$ERRPC, ADCSR,  $GDDAT, $BDDAT
1214
1215                                  ;ERROR 10 RIF BIT ACTION
1216
1217  001254  026726                  EM10                    ;RIF ERROR ON ADDS
1218  001256  023250                  DH1                     ;ERROR   A/D
1219  001260  001476                  DT1                     ;  PC    ADDR
1220  001262  001474                  DF0                     ;$ERRPC, ADCSR
1221
1222                                  ;ERROR 11 COUNTER MODULE ADDRESSABILITY
1223
1224  001264  027012                  EM11                    ;CNTR   XFER    ERROR
1225  001266  023274                  DH4                     ;ERROR COUNTER
1226  001270  001476                  DT3                     ;PC ADDR
1227  001272  001474                  DF0                     ;$ERRPC $BDADR
1228
1229                                  ;ERROR 12 COUNTER MODULE "BIT EXERCISE ROUTINE
1230
1231  001274  026576                  EM2                     ;XFER DATA ERROR
1232  001276  023254                  DH2
1233  001300  001500                  DT4
1234  001302  001474                  DF0
1235
1236                                  ;ERROR 13 COUNTER MODULE COUNTING
1237
1238  001304  027032                  EM12                    ;CNTR UP COUNT BAD
1239  001306  023254                  DH2
1240  001310  001500                  DT4
1241  001312  001474                  DF0
1242
1243                                  ;ERROR 14 COUNTER MODULE FAILED TO INTERRUPT
1244
1245  001314  027054                  EM13                    ;COUNTER FAILED TO INT
1246  001316  023274                  DH4
1247  001320  001476                  DT3
1248  001322  001474                  DF0
1249
1250                                  ;ERROR 15 COUNTER MODULE ADDR OR GENERIC CODE INCORRECT
1251
1252  001324  026706                  EM14                    ;ICAR BAD ON INT
1253  001326  023264                  DH3
1254  001330  001500                  DT4
1255  001332  001474                  DF0
1256
1257
1258
```

```
1259  001334  027077                  EM15                    ;CNTR COUNT ON OVFLW
1260  001336  023254                  DH2
1261  001340  001500                  DT4
1262  001342  001474                  DF0
1263
1264                                  ;ERROR 17 RIF DIDN'T CLEAR INTERRUPT FLAG ON COUNTER
1265
1266  001344  027122                  EM16                    ;RIF ERROR ON CNTR
1267  001346  023274                  DH4
1268  001350  001476                  DT3
1269  001352  001474                  DF0
1270
1271                                  ;ERROR 20 COUNTER MODULE INITIALIZATION PART 1
1272
1273  001354  027144                  EM17                    ;SYS INIT CNTR
1274  001356  023254                  DH2
1275  001360  001500                  DT4
1276  001362  001474                  DF0
1277
1278                                  ;ERROR 21 ILLEGAL INTERRUPT POSTED ON ICS BUS
1279
1280  001364  027164                  EM20                    ;ILL ICR INT
1281  001366  023274                  DH4
1282  001370  001476                  DT3
1283  001372  001474                  DF0
1284
1285                                  ;ERROR 22 ICR-11 FAILED TO INTERRUPT
1286
1287  001374  027201                  EM21                    ;NO ICR INTR
1288  001376  023274                  DH4
1289  001400  001476                  DT3
1290  001402  001474                  DF0
1291
1292                                  ;ERROR 23 ICAR NOT ZERO AFTER FORCED ICR INTERRUPT.
1293
1294  001404  027222                  EM22                    ;ICAR ERROR
1295  001406  023264                  DH3
1296  001410  001500                  DT4
1297  001412  001474                  DF0
1298
1299                                  ;ERROR 24 SYS INITIALIZE FAILED TO CLEAR COUNTER INTERRUPT FLAG
1300
1301  001414  027263                  EM23                    ;SYS INIT CNTR INT
1302  001416  023274                  DH4
1303  001420  001476                  DT3
1304  001422  001474                  DF0
1305
1306                                  ;ERROR 25 COUNTER STARTED COUNTING AFTER SYSTEM INIT.
1307
1308  001424  027307                  EM24                    ;CNTR CNT SYS INIT
1309  001426  023254                  DH2
1310  001430  001500                  DT4
1311  001432  001474                  DF0
1312
1313                                  ;ERROR 26 A005 READ DUAL ADDR ERROR
1314
```

```
1315  001434  027331                          EM25                    ;READ DUAL ERROR
1316  001436  023300                          DH5                     ;ERROR   A/D    DUAL
1317  001440  001506                          DT5                     ;  PC     ADDR    ADDR
1318  001442  001474                          DF0
1319
1320                                          ;ERROR 27 ADDS WRITE DUAL ADDR. ERROR
1321
1322  001444  027351                          EM26                    ;DUAL ADDR ERROR
1323  001446  023300                          DH5
1324  001450  001506                          DT5
1325  001452  001474                          DF0
1326
1327
1328                                          ;ERROR 30   SEND -RECIEVE DATA ERROR
1329
1330  001454  026576                          EM27                    ;XFER DATA ERROR
1331  001456  023254                          DH2
1332  001460  001500                          DT4
1333  001462  001474                          DF0
1334
1335
1336                                          ;ERROR 31     SAME AS ERROR 30 ONLY NO HEADER TYPEOUT
1337
1338  001464  001474                          DF0
1339  001466  001161                          $CRLF
1340  001470  001500                          DT4
1341  001472  001474                          DF0
1342
1343
1344  001474  000000              DF0:        0
1345  001476                      DT3:
1346  001476  001122              DT1:        .WORD   $BDADR
1347  001500                      DT4:
1348  001500  001122  001124  001126  DT2:    .WORD   $BDADR,$GDDAT,$BDDAT
1349  001506  001122  001124          DT5:    .WORD   $BDADR,$GDDAT
1350
1351
1352                                          ;*TABLE OF CONSTANTS
1353
1354  001512  020320              FR110:      20320                   ;TTY DELAY TIME
1355  001514  006532              FR50:       6532
1356  001516  005650              FR40:       5650
1357  001520  002724              FR20:       2724
1358  001522  002052              FR16:       2052                    ;DELAY FOR 16 MILLSEC
1359  001524  001024              FR5:        1024
1360  001526  024000              FR200:      24000
1361  001530  000000              FR32:       0                       ;DELAY FOR 3.2 MILLI SEC
1362  001532  000500              FR3:        500                     ;DELAY FOR 3 MILLI SEC.
1363  001534  000000              FREQ:       0                       ;DELAY TIME.
1364  001536  000000              FREQ1:      0
1365  001540  000000              FREQ2:      0
1366  001542  000000              FREQ3:      0
1367  001544  000152              FR1:        152                     ;DELAY FOR 1 MILLI SEC.
1368  001546  000220              FR1120:     220                     ;ON AN 11/20  FR1 X 3 FOR 11/45.
1369  001550  000000              PATRNM:     0                       ;PATTERN MODIFIER
1370  001552  000000              PATRN:      0                       ;PATTERN TO BE SENT TO OUTPUT MODULE
```

```
1371   001554   000000                PATRNC: 0
1372   001556   000000                PATJOY: 0
1373   001560   171000                ICSMOD: 171000          ;STARTING ADDRESS OF ICS MODULES.
1374   001562   171776                ICSR:   171776          ;ICR CSR
1375   001564   171774                ICAR:   171774          ;ICR ADDR REG
1376   001566   000234                ICSVT:  234             ;VECTOR ADDR
1377   001570   000236                ICSVT2: 236             ;VECTOR ADDR +2
1378   001572   000064                TPVCT:  64              ;TTY VECTOR
1379   001574   000066                TPVCT2: 66              ;TTY +2
1380   001576   000000                NOTYET: 0               ;REMOTE PRINT
1381   001600   000000                XTEMP:  0               ;
1382   001602   000000                ADBSY:  0               ;A/D BUSY FF
1383   001604   000000                DAFLG:  0               ;DA FF
1384   001606   000000                TTYTMP: 0               ;TIME O
1385   001610   000000                ERRLOP: 0               ;ERROR FLOP
1386   001612   000000                ICRVT:  0               ;SOFTWARE ICR INTERRUPT VECTOR
1387   001614   000000                INCFLG: 0               ;FLAG
1388   001616   000000                TPBSY:0                 ;TTY BUSY FF
1389   001620   000000                HEADER: 0               ;PRINT HEADER FLAG
1390   001622   000000                TOADR:0                 ;DATA FOR ERRORS
1391   001624   000000                TODAT:0
1392   001626   000000                TOGEN:  0
1393   001630   000000                TPBSYP: 0
1394   001632   000001                ST200:  1               ;CLEARED ON PROGRAM START AT 200
1395   001634   000000                CORSIZ: 0               ;END ADDR OF CORE.
1396   001636   000000                EXPERT: 0               ;0=NOVICE MODE--1=EXPERT MODE
1397   001640   000000                CONNT:  0               ;0=NORMAL; 1=MODS CONNECTED FOR TST 0
1398   001642   000000                LPAV:   0
1399   001644   000000                LINEPR: 0               ;0=TTY OUTPUT---1=LINE PRINTER
1400   001646   177514                LPCSR:  177514          ;LINE PRINTER REG
1401   001650   177516                LPDBR:  177516
1402   001652   000000                TPCSR:  0               ;TEST 7 PRINT MEDIA REG
1403   001654   000000                TPDBR:  0
1404   001656   000000                TMPFIL: 0               ;
1405   001660   000000                TMPVEC: 0               ;
1406   001662   000000                ICSHGH: 0               ;UPPER MODULE ADDRESS
1407   001664   000000                CTLLOC: 0               ;CNTRL C RETURN ADDRESS
1408   001666   000000                ICSLMT: 0               ;LOWER MODULE ADDRESS
1409   001670   000000                REMSWR: 0               ;SWR REG USED WHEN REMOTE
1410   001672   000000                SWRFF:  0               ;SWR FF (↑S)
1411   001674   000000                REMFF1: 0               ;REMOTE FF FOR PWR FAIL
1412   001676   000000                ERRCNT: 0               ;ERROR COUNT
1413   001700   000000                MODFF:  0               ;SET MODULE INTERRUPT PRESENT
1414   001702   000000                TMPSWR: 0               ;TEMP STORAGE OF SWR WHILE SWITCHING BETW/ LOCAL AND REM
1415   001704   000000                REMEND: 0               ;REMOTE TTY FLAG
1416   001706   000000                XTMPFL: 0
1417   001710   000000                XICSR:  0
1418   001712   000000                XICAR:  0
1419   001714   000000                XICSLT: 0
1420   001716   000000                XICSHG: 0
1421   001720   000000                XVEC:   0
1422   001722   000000                XICSVT: 0
1423   001724   000000                XICSV2: 0
1424
1425            000002                      EXIT=   2           ; RTI INSTRUCTION
1426            000207                      RETURN=207          ;RTS PC INSTRUCTION
```

```
                                GOSUB=4737              ;JSR PC, INSTRUCTION
                                POP2SP=22626
                                POPSP2=22626

                        .SBTTL  ICSR BIT EQUIVALENTS
                        ;
100000                  OUTBSY=BIT15                    ;ICSR - OUTPUT BUSY
040000                  MAINT3=BIT14                    ;ICSR - MAINT BIT 03
020000                  MAINT2=BIT13                    ;ICSR - MAINT BIT 02
010000                  ERRBIT=BIT12                    ;ICSR - ERROR INTERRUPT
004000                  MAINT1=BIT11                    ;ICSR - MAINT BIT 01
002000                  PWRFL=BIT10                     ;ICSR - POWER FAIL INTERRUPT
001000                  TBMTEN=BIT9                     ;ICSR - TBMT INTERRUPT ENABLE
000400                  MAINT0=BIT8                     ;ICSR - MAINT BIT 00
000200                  MODINT=BIT7                     ;ICSR - MODULE INTERRUPT
000100                  XRESET=BIT6                     ;ICSR - RESET
000040                  TTYEN=BIT5                      ;ICSR - TTY ENABLE
000020                  PWFEN=BIT4                      ;ICSR - POWER FAIL INTERRUPT ENA..
000010                  BMTEN=BIT3                      ;ICSR - BMT INTERRUPT ENABLE
000004                  MODEN=BIT2                      ;ICSR - MODULE INTERRUPT ENABLE
000002                  ERREN=BIT1                      ;ICSR - ERROR INTERRUPT ENABLE
000001                  XRIF=BIT0                       ;ICSR - RESET INTERRUPT FLAG
                        ;
                        .SBTTL  ICAR BIT EQUIVALENTS
                        ;
100000                  XTBMT=BIT15                     ;ICAR - TBMT
010000                  DA=BIT12                        ;ICAR -.DA
                        ;
                        ;


001726                          .SETUP  <.$TRAP,.$SCOPE,.$ERROR,.$POWER>

001726  012737  177570  001136  START:  MOV     #177570,SWR     ;SETUP HARDWARE  SWR
001734  005737  001632          TST     ST200           ;SEE IF STARTING AFTER INITIAL LOAD
001740  001002                  BNE     .+6             ;IF NOT, LEAVE REMOTE  FLOP  ALONE
001742  005037  000164          CLR     REMFF           ;FORCE  PROGRAM TO START IN LOCAL MODE
001746                          SETUP
001746  012706  001100          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001752  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001754  022706  001126          CMP     #$BDDAT,R6      ;;DONE?
001760  001374                  BNE     .-6             ;;LOOP BACK IF NO
001762  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
001766  012737  023670  000020  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001774  012737  000340  000022  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002002  012737  022566  000030  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002010  012737  000340  000032  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002016  012737  025272  000034  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002024  012737  000340  000036  MOV     #340,@#TRAPVEC+2;LEVEL 7
002032  012737  023534  000024  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002040  012737  000340  000026  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002046  005037  001156          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002052  012737  002052  001106  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002060                          SWRSU
002060  013746  000004          MOV     @#4,-(SP)       ;;SAVE ERROR VECTOR
002064  013746  000006          MOV     @#6,-(SP)
002070  012737  002104  000004  MOV     #64$,4          ;;SET UP TIME OUT VECTOR
```

```
1483  002076  005777  177034                    TST     @SWR            ;;TRY TO REFERENCE HARDWARE SWR
1484  002102  000407                            BR      65$             ;;BRANCH IF NO TIMEOUT TRAP OCCURS
1485  002104  012737  000176  001136  64$:      MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1486  002112  012737  000174  001140            MOV     #DISPREG,DISPLAY ;;POINT TO SOFTWARE DISPLAY REG
1487  002120  022626                            CMP     (SP)+,(SP)+     ;;RESTORE STACK
1488  002122  012637  000006          65$:      MOV     (SP)+,@#6       ;;RESTORE ERROR VECTOR
1489  002126  012637  000004                    MOV     (SP)+,@#4
1490  002132  005037  001576                    CLR     NOTYET          ;DONT ALLOW ↑R OR REMOTE TYPE TIL FILE KNOWN
1491  002136  012737  176543  022336            MOV     #176543,$HINUM
1492  002144  012737  123456  022340            MOV     #123456,$LONUM
1493  002152  005037  001640                    CLR     CONNT
1494
1495                                    ;
1496                                    ;FILL LOCATIONS "214-1000" WITH .+2, IOT
1497                                    ;
1498
1499  002156  013737  001136  001702            MOV     SWR,TMPSWR      ;STORE LOCAL SWR
1500  002164  005737  000164                    TST     REMFF           ;DID WE LOAD REMOTELY?
1501  002170  001433                            BEQ     FILVEC          ;NO, THEN GO ASK FOR ICR BOX
1502  002172  005237  001576                    INC     NOTYET          ;OK TO PRINT REMOTE  AND ACCEPT ↑R
1503  002176  012737  001670  001136            MOV     #REMSWR,SWR     ;SET REMOTE SWR
1504  002204  013737  000166  001564            MOV     ICARLD,ICAR     ;LOADED REMOTE GET INFORMATION
1505  002212  013737  000170  001562            MOV     ICSRLD,ICSR     ;FROM LOC 162-172 BEFORE
1506  002220  013737  000172  001666            MOV     ICSMLD,ICSLMT   ;GOING REMOTE
1507  002226  013700  000172                    MOV     ICSMLD,R0
1508  002232  062700  000040                    ADD     #40,R0
1509  002236  010037  001662                    MOV     R0,ICSHGH
1510  002242  013737  000162  001660            MOV     ICRVEC,TMPVEC
1511  002250  104400  025404                    TYPE,   MHEAD           ;EVERYTHING SET;TYPE HEADER
1512  002254  000137  002574                    JMP     SKPASK          ;DON'T ASK QUESTIONS
1513  002260  012700  000214          FILVEC:   MOV     #214,R0
1514  002264  012701  000216                    MOV     #216,R1
1515  002270  010120                  98$:      MOV     R1,(R0)+        ;SET UP .+2, IOT FOR
1516  002272  012720  000004                    MOV     #4,(R0)+        ;VECTOR CHECK
1517  002276  022121                            CMP     (R1)+,(R1)+
1518  002300  022700  001000                    CMP     #1000,R0
1519  002304  003371                            BGT     98$
1520  002306  104400                            TYPE                    ;TYPE HEADER
1521  002310  025404                            MHEAD
1522  002312  104400  025437                    TYPE,   MHEAD1
1523                                    ;**
1524                                    ;** SINCE ICR IS CAPABLE OF DIFFERENT CONTROL REGISTERS
1525                                    ;** ASK OPERATOR FOR FILE BOX AND INTERRUPT VECTOR ADDRESS
1526                                    ;**
1527
1528  002316  012737  002324  001664            MOV     #10$,CTLLOC
1529  002324  104400  031316          10$:      TYPE, MFILE             ;ASK FOR FILE BOX
1530  002330  012737  000001  002346            MOV     #1,22$
1531  002336  000401                            BR      .+4
1532  002340  000771                            BR      10$
1533  002342  104414                            INOCT
1534  002344  001656                            TMPFIL
1535  002346  000001          22$:      1
1536  002350  022737  000013  001656            CMP     #13,TMPFIL
1537  002356  002003                            BGE     11$
1538  002360  104400  031403                    TYPE, ILLEG             ;ILLEGAL FILE BOX... RE-ASK
```

```
                                        BR      10$
                    171000  001666  11$:    MOV     #171000,ICSLMT          ;INIT. MODULE ADDRESSES
                    171776  001562           MOV     #171776,ICSR            ;INIT. ICSR
                    001656                   INC     TMPFIL
                    001656          19$:     DEC     TMPFIL                  ;DEC. FILE
                                             BEQ     18$                     ;BRANCH IF FOUND
                    000010  001562           SUB     #10,ICSR                ;NEXT ICSR
                    000040  001666           ADD     #40,ICSLMT              ;NEXT GROUP OF MODULE ADDRESSES
                                             BR      19$                     ;LOOP
                    001562  001564  18$:     MOV     ICSR,ICAR               ;CREATE ICAR
                    000002  001564           SUB     #2,ICAR
                    001666  001662           MOV     ICSLMT,ICSHGH           ;SET UPPER ADDRESS LIMIT
                    000040  001662           ADD     #40,ICSHGH
                    002476  000004           MOV     #12$,@#4
                    177066                   MOV     @ICSR,R0
                                             BR      97$
                                    12$:     POPSP2
                    031765                   TYPE,   NONXST
                                             BR      10$
                    000006  000004  97$:     MOV     #6,@#4
                    002522  001664           MOV     #99$,CTLLOC
                    031352          99$:     TYPE, MVECT                     ;ASK FOR INTERRUPT VECTOR
                    000001  002544           MOV     #1,23$
                                             BR      .+4
                                             BR      99$
                                             INOCT
                                             TMPVEC
                                    23$:     1
                    000776  001660           CMP     #776,TMPVEC             ;MAKE SURE VECTOR IS LEGAL
                                             BLT     16$
                    000234  001660           CMP     #234,TMPVEC
                                             BLE     17$
                    031403          16$:     TYPE, ILLEG                     ;NOT IN FLOATING AREA
                                             BR      99$
                                    SKPASK=.
                    001660  001566  17$:     MOV     TMPVEC,ICSVT
                    000002  001660           ADD     #2,TMPVEC
                    001660  001570           MOV     TMPVEC,ICSVT2           ;STORE VECTOR ADDRESS
                                    ;**
                                    ;**VERIFY THAT VECTOR GIVEN IS ACTUAL VECTOR BEFORE GOING
                                    ;**ANY FARTHER
                                    ;

                    000020  003002           MOV     @#20,93$                ;SAVE LOC. 20
                    002712  000020           MOV     #95$,@#20               ;SET UP IOT TRAP VECTOR
                    002762  176726           MOV     #94$,@ICSVT             ;SET UP INTERRUPT ROUTINE
                    000340  176722           MOV     #340,@ICSVT2            ;PS PICKUP
                    000240  177776           MOV     #240,@#PS                       ;SET PRIORITY TO 5
                    176704                   TST     @ICAR                           ;CLEAR ANY LINE ERRORS
                    020004  176674           BIS     #MAINT2+MODEN,@ICSR             ;SET FOR MOD. INT, FORCE INTERRUPT
                                             CLR     R0
                                             INC     R0
                                             BNE     .-2
                                             NOP
                    031460                   TYPE,   NOINT                   ;NO INTERRUPT
                    003002  000020           MOV     93$,@#20                ;RESTORE LOC 20
```

```
1595  002710  000413                        BR      96$                     ;RE-ASK QUESTION
1596  002712  011605              95$:      MOV     (SP),R5                 ;INVALID VECTOR ADDRESS PRINT
1597  002714  024545                        CMP     -(R5),-(R5)             ;SUB 4 FROM R5
1598  002716  022626                        POPSP2                          ;POP STACK
1599  002720  000240                        NOP
1600  002722  104400  031523                TYPE,   FILINT                  ;PRINT WHERE VECTOR
1601  002726                                TYPOCT  R5
1602  002726  010546                        MOV     R5,-(SP)        ;;SAVE R5 FOR TYPEOUT
1603  002730  104401                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1604  002732  104400  031556                TYPE,   CKJMP                   ;INTERRUPTED FROM
1605  002736  022626                        POPSP2
1606  002740  052777  040000  176614  96$:  BIS     #40000,@ICSR            ;RESET ICR MICROCODE
1607  002746  013737  003002  000020        MOV     93$,@#20                ;HALT
1608  002754  000000                        HALT
1609  002756  000137  002522                JMP     99$
1610  002762  013737  003002  000020  94$:  MOV     93$,@#20                ;INTERRUPT OK, RESTORE 20
1611  002770  022626                        POPSP2                          ;POP STACK
1612  002772  052777  040000  176562        BIS     #40000,@ICSR
1613  003000  000401                        BR      .+4
1614  003002  000000              93$:      0
1615                                ;
1616                                ;FILL LOCATIONS "214-1000" WITH .+2, HALT, TO CATCH ILLEGAL VECTORING
1617                                ;
1618
1619  003004  012737  003652  001664        MOV     #START1,CTLLOC          ;SET CNTRL C RETURN
1620  003012  012700  000214        FILHLT: MOV     #214,R0
1621  003016  012701  000216                MOV     #216,R1
1622  003022  010120              90$:      MOV     R1,(R0)+                ;FILL .+2, HALT
1623  003024  012720  000000                MOV     #0,(R0)+
1624  003030  022121                        CMP     (R1)+,(R1)+
1625  003032  022700  001000                CMP     #1000,R0
1626  003036  003371                        BGT     90$
1627  003040  012737  004172  000060        MOV     #KEYSRV,@#60            ;SET TTY INTR.
1628  003046  052777  000100  176066        BIS     #100,@#TKS              ;ALLOW TTY INTR
1629  003054  012737  003124  000004        MOV     #9$,@#4                 ;SET UP FOR TIMEOUT ON LPT CHECK
1630  003062  012777  021500  176476        MOV     #ICRSRV,@ICSVT          ;SET UP ICR INTR VECTOR
1631  003070  005037  001612                CLR     ICRVT                   ;NO SERVICE NOW
1632  003074  052777  000026  176460        BIS     #MODEN+ERREN+PWFEN,@ICSR  ;ALLOW REMOTE INTERRUPTS
1633  003102  005237  001642                INC     LPAV
1634  003106  005777  176534                TST     @LPCSR                  ;IS THERE AN LPT?
1635  003112  104400                        TYPE                            ;YES,INDICATE LPT
1636  003114  001161                        $CRLF
1637  003116  104400                        TYPE
1638  003120  030310                        MLPAV
1639  003122  000403                        BR      8$
1640  003124  005037  001642        9$:     CLR     LPAV                    ;NO, LPT
1641  003130  022626                        POPSP2
1642
1643
1644  003132  012737  003146  000004  8$:   MOV     #3$,@#4                 ;DETERMINE CORE SIZE.
1645  003140  005000                        CLR     R0
1646  003142  005720                2$:     TST     (0)+
1647  003144  000776                        BR      2$
1648  003146  005740                3$:     TST     -(0)
1649  003150  010037  001634                MOV     R0,CORSIZ
1650  003154  100003                        BPL     4$
```

```
1651  003156  012737  077776  001634          MOV     #077776,CORSIZ
1652  003164  012737  000006  000004  4$:     MOV     #6,@#4
1653  003172  013737  001546  001544          MOV     FR1120,FR1          ;SET FREQ FOR 11/20
1654  003200  012737  000002  000012          MOV     #RTI,@#12           ;FIND OUT IF 11/20 OR 11/45
1655  003206  000262                          SEV                         ;IF 11/45 THEN DELAY ITERATIONS
1656  003210  074101                          XOR     %1,%1               ;MUST BE INCREASED.
1657  003212  102427                          BVS     5$
1658  003214  063737  001546  001544          ADD     FR1120,FR1          ;FLOW FELL THUR TO HERE -
1659  003222  063737  001546  001544          ADD     FR1120,FR1          ;IT MUST BE AN 11/45 OR EQUIV.
1660  003230  013746  000004                  MOV     @#4,-(SP)           ;SAVE LOC 4
1661  003234  013746  000006                  MOV     @#6,-(SP)           ;SAVE LOC 6
1662  003240  012737  003260  000004          MOV     #99$,@#4            ;SETUP  FOR TRAP
1663  003246  005737  177760                  TST     @#177760            ;ADDRESS 11/70  SIZE REG
1664  003252  006337  001544                  ASL     FR1                 ;TST OK, 11/70 PRESENT SHIFT CONST
1665  003256  000401                          BR      .+4
1666  003260  022626                  99$:    POP2SP                      ;RESET STACK
1667  003262  012637  000006                  MOV     (SP)+,@#6
1668  003266  012637  000004                  MOV     (SP)+,@#4
1669  003272  005037  000012          5$:     CLR     @#12                ;RESTORE LOC 12.
1670  003276  013737  001544  001532          MOV     FR1,FR3             ;NOW WE MUST SET THE REST OF
1671  003304  006137  001532                  ROL     FR3                 ;OF THE DELAY TIMES UP.
1672  003310  013737  001532  001524          MOV FR3,FR5
1673  003316  063737  001544  001532          ADD     FR1,FR3
1674  003324  063737  001532  001524          ADD FR3,FR5
1675  003332  013737  001524  001522          MOV     FR5,FR16
1676  003340  006137  001522                  ROL     FR16
1677  003344  063737  001524  001522          ADD     FR5,FR16
1678  003352  013737  001522  001520          MOV     FR16,FR20
1679  003360  063737  001544  001522          ADD     FR1,FR16
1680  003366  063737  001524  001520          ADD     FR5,FR20
1681  003374  013737  001520  001514          MOV     FR20,FR50
1682  003402  006137  001514                  ROL     FR50
1683  003406  063737  001524  001514          ADD     FR5,FR50
1684  003414  063737  001524  001514          ADD     FR5,FR50
1685  003422  013737  001514  001512          MOV     FR50,FR110
1686  003430  006137  001512                  ROL     FR110
1687  003434  063737  001524  001512          ADD     FR5,FR110
1688  003442  063737  001524  001512          ADD     FR5,FR110
1689  003450  013737  001520  001516          MOV     FR20,FR40
1690  003456  006137  001516                  ROL     FR40
1691  003462  013737  001544  001530          MOV     FR1,FR32            ;SETUP FOR 3.2 MILLISEC DELAY CONSTANT
1692  003470  006237  001530                  ASR     FR32                ;DIVIDE BY 2
1693  003474  006237  001530                  ASR     FR32                ;DIVIDE BY 2 AGAIN
1694  003500  063737  001532  001530          ADD     FR3,FR32
1695  003506  005037  001636                  CLR     EXPERT
1696
1697
1698  003512  005037  001552                  CLR     PATRN
1699  003516  005737  001632                  TST     ST200               ;STARTED 200?
1700  003522  001406                          BEQ     7$                  ;NO
1701  003524  013737  001146  001652          MOV     $TPS,TPCSR
1702  003532  013737  001150  001654          MOV     $TPB,TPDBR
1703  003540  012700  014670          7$:     MOV     #INADR, R0          ;CLEAR ADDR AREA
1704  003544  005037  001632                  CLR     ST200               ;INDICATE START AT LOC 200.
1705  003550  005020                  1$:     CLR     (0)+
1706  003552  020027  015036                  CMP     R0,#OUTS
```

```
1707  003556  001374                        BNE     1$
1708  003560  005037  001542                CLR     FREQ3
1709  003564  005037  001540                CLR     FREQ2
1710  003570  005037  001536                CLR     FREQ1
1711  003574  013737  001532  001534        MOV     FR3,FREQ
1712
1713  003602  005737  000164                TST     REMFF                   ;DID WE LOAD REMOTELY
1714  003606  001021                        BNE     START1                  ;YES, THEN DONT DO NEXT PART
1715  003610  005737  001632        RSTART: TST     ST200                   ;DID WE START AT 200
1716  003614  001411                        BEQ     3$                      ;YES, CONT
1717  003616  012737  024546  000034        MOV     #$TYPE,@#34             ;SETUP TO TYPE
1718  003624  005037  000036                CLR     @#36
1719  003630  104400  026406                TYPE,   MSTERR                  ;TYPE ERROR
1720  003634  000137  001726                JMP     START                   ;GO TO START  FOR LAZY USER
1721  003640  005037  000164        3$:     CLR     REMFF                   ;CLR REMOTE IND
1722  003644  013737  001702  001136        MOV     TMPSWR,SWR              ;SET BACK FOR LOCAL SWR
1723
1724  003652  012706  001100        START1: MOV     #1100,SP
1725  003656  005237  001576                INC     NOTYET
1726  003662  052777  040000  175672        BIS     #MAINT3,@ICSR           ;RESET SYSTEM
1727  003670  005037  001674                CLR     REMFF1
1728  003674  005037  001100                CLR     $PASS
1729  003700  005037  001644                CLR     LINEPR
1730  003704  005037  001704                CLR     REMEND
1731  003710  013737  001534  014626        MOV     FREQ,RTEMP
1732  003716  013737  001512  001534  3$:   MOV     FR110,  FREQ    ;SET DELAY FOR TTY SETTLE TIME.
1733  003724  005037  001536                CLR     FREQ1
1734  003730  013737  001652  001146        MOV     TPCSR,$TPS
1735  003736  013737  001654  001150        MOV     TPDBR,$TPB
1736  003744  104422                        DELAY                           ;CAUSE A DELAY.
1737  003746  000005                        RESET                           ;ISSUE SYSTEM INITIALIZE.
1738  003750  104422                        DELAY                           ;TTY SETTLE DOWN TIME.
1739  003752  012737  000340  177776        MOV     #340,PS
1740  003760  052777  040000  175574        BIS     #MAINT3,@ICSR
1741  003766  005777  175572                TST     @ICAR                   ;CLEAR ANY ERROR
1742  003772  104400                99$:    TYPE                            ;TYPE "TEST NO.?"
1743  003774  026172                        MTN
1744  003776  012737  004146  004020        MOV     #TSTNO, 1$
1745  004004  012737  000002  004022        MOV     #2,     1$+2
1746  004012  000401                        BR      .+4
1747  004014  000766                        BR      99$
1748  004016  104414                        INOCT                           ;GET TEST NUMBER
1749  004020  004146                1$:     TSTNO
1750  004022  000002                        2
1751  004024  005737  004146                TST     TSTNO                   ;MAKE SURE NUMBER IS POSITIV
1752  004030  100404                        BMI     4$
1753  004032  023727  004146  000007        CMP     TSTNO,  #7.             ;LEGAL NUMBER?
1754  004040  003403                        BLE     2$
1755  004042  104400                4$:     TYPE                            ;NO-TYPE "NO SUCH TEST"
1756  004044  026215                        MTNL
1757  004046  000701                        BR      START1
1758  004050  013737  004146  014626  2$:   MOV     TSTNO,RTEMP
1759  004056  006337  014626                ASL     RTEMP
1760  004062  062737  004152  014626        ADD     #TSTLST, RTEMP
1761  004070  017737  010532  014626        MOV     @RTEMP,RTEMP
1762  004076  012777  000100  175036        MOV     #100,@$TKS              ;ENABLE TTY TO INTERRUPT.
```

```
1763  004104  052777  000026  175450          BIS     #PWFEN+MODEN+ERREN,@ICSR
1764  004112  005037  001612                  CLR     ICRVT                   ;SET FOR NO SERVICE ROUTINE
1765  004116  005037  177776                  CLR     PS
1766  004122  005037  001604                  CLR     DAFLG
1767  004126  005037  001602                  CLR     ADBSY
1768  004132  005077  175000                  CLR     @SWR
1769  004136  005037  001700                  CLR     MODFF           ;CLEAR MOD INTR FLOP
1770  004142  000177  010460                  JMP     @RTEMP          ;GOTO TEST
1771
1772  004146  000000                  TSTNO:  0                       ;TEST NUMBER
1773  004150  000000                          0
1774  004152  004266                  TSTLST: TST0                    ;INPUT AND OUTPUT MODULE EXERCISER
1775  004154  005754                          TST1                    ;INPUT OR OUTPUT MODULE SIMPLE EXER.
1776  004156  006402                          TST2                    ;DAC CALIBRATION TEST
1777  004160  006472                          TST3                    ;DAC INTERACTION TEST
1778  004162  006654                          TST4                    ;COUNTER MODULE TEST
1779  004164  010200                          TST5                    ;A/D LOGIC TEST
1780  004166  011246                          TST6                    ;A/D CALIBRATION TEST
1781  004170  012274                          TST7                    ;A/D REPEATIBILITY TEST
1782
1783
1784                                          ;*
1785                                          ;*KEYBOARD INTERRUPT HANDLER
1786                                          ;*
1787  004172  017737  174746  004264  KEYSRV: MOV     @$TKB,2$
1788  004200  042737  177600  004264  3$:     BIC     #177600,2$
1789  004206  122737  000003  004264          CMPB    #3,2$
1790  004214  001421                          BEQ     1$
1791  004216  122737  000020  004264          CMPB    #20,2$                  ;CONTROL P, FOR ERROR COUNT
1792  004224  001014                          BNE     4$                      ;NO, THEN EXIT
1793
1794          004226                  TYPECT=.
1795  004226  104400  027235                  TYPE,   ERRORH                          ;TYPE ERROR
1796  004232  013700  001676                  MOV     ERRCNT,R0
1797  004236                                  TYPOCT  ERRCNT
1798  004236  013746  001676                  MOV     ERRCNT,-(SP)    ;;SAVE ERRCNT FOR TYPEOUT
1799  004242  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1800  004244  104400  001161                  TYPE,   $CRLF
1801  004250  005037  001676                  CLR     ERRCNT
1802  004254  000401                          BR      1$
1803  004256  000002                  4$:     EXIT
1804  004260  000137  003652          1$:     JMP     START1
1805  004264  000000                  2$:     0
1806
1807
1808                                          ;*
1809                                          ;*TEST 0 INPUT AND OUTPUT MODULE EXERCISER
1810                                          ;*
1811
1812  004266  104400                  TST0:   TYPE                    ;TYPE HEADER.
1813  004270  027671                          MHT0
1814  004272  104415                          INAR                    ;GET INPUT MODULE ADDRS.
1815  004274  104416                          OUTAR                   ;GET OUTPUT MODULE ADDRS.
1816  004276  104417                          PATAR                   ;GET PATTERN (OR USE DEFALT).
1817  004300  104420                          DELAR                   ;GET DELAY TIME (OR USE DEFALT).
1818  004302  012703  032017                  MOV     #OUTBF1-1,R3    ;SETUP TO TYPE OUT
```

```
1819  004306  005037  001100                  CLR     $PASS
1820  004312  005037  001620                  CLR     HEADER
1821                                                                ;INITIAL SETUP.
1822  004316  012737  005102  001612          MOV     #TOINT,ICRVT  ;SET UP FOR ICR INTERRUPTS.
1823  004324  012777  004732  175240          MOV     #TOPINT,@TPVCT ;SET UP FOR TTY INTERRUPTS.
1824  004332  012777  000000  175230          MOV     #0,@ICSVT2    ;SET CPU PRIORITY TO 0 ON ICR INTR.
1825  004340  012777  000340  175226          MOV     #340,@TPVCT2  ;SET CPU PRIORITY TO 4 ON TTY INTR.
1826  004346  005037  001616                  CLR     TPBSY         ;SET NO TTY OUTPUT IN PROGRESS
1827  004352  012700  015036                  MOV     #OUTS,R0      ;CLEAR TEMP STORAGE AREA OF INPUT
1828  004356  005020                  1$:     CLR     (0)+          ;MODULE DATA CHANGE.
1829  004360  020027  015076                  CMP     R0,#OUTSE
1830  004364  001374                          BNE     1$
1831  004366  005737  001640                  TST     CONNT
1832  004372  001002                          BNE     2$
1833  004374  104400                          TYPE                  ;TYPE HEADER.
1834  004376  025522                          MTOH
1835  004400  005037  001622          2$:     CLR     TOADR         ;CLEAR TYPE OUT CONSTANTS.
1836  004404  005037  001624                  CLR     TODAT
1837  004410  005037  001626                  CLR     TOGEN
1838  004414  052777  000001  175140          BIS     #XRIF,@ICSR
1839  004422  005777  175240                  TST     @ICSLMT
1840
1841                                                                ;TAKE CARE OF OUTPUT ADDRS.
1842  004426  104422                          DELAY
1843  004430  012737  004430  001106  TOOUTR: MOV     #TOOUTR,$LPADR
1844  004436  104421                          DELAY2
1845  004440  012702  014670                  MOV     #INADR,R2
1846  004444  052777  000026  175110          BIS     #MODEN+PWFEN+ERREN,@ICSR      ;ALLOW ICR TO INTR.
1847  004452  005037  177776                  CLR     PS            ;LET CPU ALLOW INTRS.
1848  004456  012704  015036                  MOV     #OUTS,R4
1849  004462  012700  014730          TOOUT:  MOV     #OUTADR,R0    ;GET OUTPUT MODULE LIST.
1850  004466  012001                  1$:     MOV     (0)+,R1       ;GET FIRST ADDR
1851  004470  001406                          BEQ     TOIN          ;IF N ADDR. - EXIT.
1852  004472  104432                          WTBSY                 ;WAIT FOR INACTIVE LINE
1853  004474  013711  001552                  MOV     PATRN,(1)     ;SEND PATTERN TO OUTPUT MODULE.
1854  004500  020027  014770                  CMP     R0,#CNTADR
1855  004504  001370                          BNE     1$
1856                                                                ;TAKE CARE OF INPUT ADDRS.
1857  004506  005737  001640          TOIN:   TST     CONNT         ;MODULES CONNECTED?
1858  004512  001407                          BEQ     2$            ;IF NOT NORMAL CONTINUE.
1859  004514  012701  015036                  MOV     #OUTS,R1      ;IF SO COS AREA = CURRENT PATTERN.
1860  004520  013721  001552          3$:     MOV     PATRN,(1)+
1861  004524  020127  015076                  CMP     R1,#OUTSE
1862  004530  001373                          BNE     3$
1863  004532  013737  001552  001556  2$:     MOV     PATRN,PATJOY
1864  004540  104412                          DELAYD                ;DELAY TIME
1865  004542  000240                          NOP
1866  004544  005737  001616                  TST     TPBSY
1867  004550  001374                          BNE     .-6
1868  004552  011201                  1$:     MOV     (2),R1        ;PICK UP FIRST ADDR.
1869  004554  001002                          BNE     .+6
1870  004556  000137  005332                  JMP     TOLOP
1871  004562  021114                          CMP     (1),(4)       ;DATA CHANGED?
1872  004564  001007                          BNE     TOIN2         ;IF YES TAKE CARE OF IT
1873          004566          TOIN1=.
1874  004566  005722                          TST     (2)+
```

```
1875  004570  005724                          TST     (4)+            ;UPDATE COD POINTER
1876  004572  020227  014730                  CMP     R2,#OUTADR      ;DONE ALL INPUT MODULES?
1877  004576  001365                          BNE     1$
1878  004600  000137  005332                  JMP     TOLOP
1879
1880  004604  005737  001640          TOIN2:  TST     CONNT
1881  004610  001425                          BEQ     1$
1882  004612  005737  001616          2$:     TST     TPBSY
1883  004616  001375                          BNE     2$
1884  004620  010137  001122                  MOV     R1,@BDADR
1885  004624  013737  001556  001124          MOV     PATJOY,@GDDAT
1886  004632  011137  001126                  MOV     (1),@BDDAT
1887  004636  005737  001620                  TST     HEADER
1888  004642  001403                          BEQ     3$
1889  004644  104031                          ERROR   31
1890  004646  000137  005332                  JMP     TOLOP
1891
1892  004652  104030                  3$:     ERROR   30
1893  004654  005237  001620                  INC     HEADER
1894  004660  000137  005332                  JMP     TOLOP
1895
1896  004664  005737  001616          1$:     TST     TPBSY
1897  004670  001375                          BNE     1$
1898  004672  012737  000340  177776          MOV     #340,PS
1899  004700  010137  001622                  MOV     R1,TOADR        ;GET ADDR. OF INPUT MODULE
1900  004704  011137  001624                  MOV     (1),TODAT       ;GET CHANGE DATA
1901  004710  005037  001626                  CLR     TOGEN           ;NO GEN CODE (NO INTERRUPT)
1902  004714  005237  001616                  INC     TPBSY           ;SET TTY BUSY.
1903  004720  011114                          MOV     (1),(4)         ;RECORD NEW DATA
1904  004722  104430                          FOCTA                   ;FORM INFO INTO AN ASCIZ MESSAG.
1905  004724  005046                          CLR     -(6)
1906  004726  012746  004566                  MOV     #TOIN1,-(6)
1907
1908          004732                  TOPINT=.
1909
1910  004732  032777  020000  174176          BIT     #020000,@SWR    ;INHIBIT TYPEOUT?
1911  004740  001406                          BEQ     5$              ;NO CONTINUE.
1912  004742  105777  174200                  TSTB    @TPS            ;PRINTER BUSY?
1913  004746  100371                          BPL     TOPINT
1914  004750  005037  001616                  CLR     TPBSY           ;YES-STOP TYPEOUT
1915  004754  000002                          EXIT
1916  004756  005737  000164          5$:     TST     REMFF           ;RUNNING REMOTE
1917  004762  001433                          BEQ     6$              ;NO, SKIP PRINTING
1918
1919  004764  032777  010000  174572          BIT     #DA,@ICAR       ;DA SET?
1920  004772  001401                          BEQ     .+4             ;NO CONT
1921  004774  104413                          CKRMTT                  ;YES, CHECK KEYBOARD
1922  004776  052777  001040  174556          BIS     #TBMTEN+TTYEN,@ICSR
1923  005004  032777  100000  174552  8$:     BIT     #XTBMT,@ICAR    ;PRINT TO REMOTE
1924  005012  001004                          BNE     9$
1925  005014  042777  001040  174540          BIC     #TTYEN+TBMTEN,@ICSR
1926  005022  000743                          BR      TOPINT
1927  005024  042777  001000  174530  9$:     BIC     #TBMTEN,@ICSR
1928  005032  111337  001606                  MOVB    (3),TTYTMP
1929  005036  013777  001606  174622          MOV     TTYTMP,@ICSLMT
1930  005044  042777  001040  174510  10$:    BIC     #TBMTEN+TTYEN,@ICSR
```

```
1931
1932  005052                                  6$:
1933  005052  112377  174072                  MOVB    (3)+,@STPB      ;SEND CHAR.
1934  005056  001404                          BEQ     4$              ;IF END GO TO END
1935  005060  052777  000100  174060  7$:     BIS     #100,@STPS      ;MAKE SURE TTY CAN INTR.
1936  005066  000002                          EXIT                    ;EXIT
1937
1938  005070  005077  174052          4$:     CLR     @STPS           ;TTY SETTLE DOWN TIME
1939  005074  005037  001616                  CLR     TPBSY           ;CLEAR BUSY
1940
1941  005100  000002                          EXIT
1942
1943
1944                                          ;*
1945                                          ;*ICR INTERRUPT SERVICE ROUTINE (RETURN FROM ICRVT)
1946                                          ;*
1947
1948  005102  017737  174456  005324  TOINT:  MOV     @ICAR,6$        ;GET ADDR. OF INTRING MOD.
1949  005110  006137  005324                  ROL     6$              ;FORM REAL ADDR.
1950  005114  042737  177001  005324          BIC     #177001,6$
1951  005122  053737  001560  005324          BIS     ICSMOD,6$
1952  005130  017737  000170  005326          MOV     @6$,7$          ;FETCH DATA.
1953  005136  005737  001616          10$:    TST     TPBSY           ;TYPEOUT PENDING?
1954  005142  001406                          BEQ     5$              ;IF NOT CONTINUE.
1955  005144  005077  174412                  CLR     @ICSR           ;STOP ICR FROM INTERRUPTING
1956  005150  104411                          ICRDL1
1957  005152  005037  177776                  CLR     PS              ;ALLOW INTERRUPTS FROM TTY.
1958  005156  000767                          BR      10$             ;LOOP.
1959  005160  012737  000300  177776  5$:     MOV     #300,PS         ;LOCK OUT INTR.S
1960  005166  052777  000025  174366          BIS     #MODEN+XRIF+PWFEN,@ICSR ;RE-ENABLE INTRS.AND SET RIF.
1961  005174  017737  174364  001626          MOV     @ICAR,TOGEN     ;GET GENERIC CODE
1962  005202  013737  005324  001622          MOV     6$,TOADR
1963  005210  005777  174406                  TST     @TOADR
1964  005214  013737  005326  001624          MOV     7$,TODAT
1965  005222  005737  001640                  TST     CONNT
1966  005226  001401                          BEQ     9$
1967  005230  000002                          EXIT
1968
1969  005232  000337  001626          9$:     SWAB    TOGEN           ;RIGHT JUSTIFY GEN CODE.
1970  005236  042737  177760  001626          BIC     #177760,TOGEN
1971  005244  010037  005330                  MOV     R0,SAV0 ;SAVE R0
1972  005250  012700  014670                  MOV     #INADR,R0       ;FIND OFFSET OF INPUT MODULE
1973  005254  022037  001622          2$:     CMP     (0)+,TOADR      ;THAT INTERRUPTED.
1974  005260  001404                          BEQ     3$
1975  005262  020027  014730                  CMP     R0,#OUTADR      ;IF NOT ENTERED INPUT MODULE THAT
1976  005266  001407                          BEQ     4$              ;THAN ADDR WILL NOT BE IN TABLE
1977  005270  000771                          BR      2$              ;IN THAT CASE DON'T WORRY ABOUT IT.
1978  005272  162700  014672          3$:     SUB     #INADR+2,R0     ;SUB TO GET OFFSET.
1979  005276  062700  015036                  ADD     #OUTS,R0        ;ADD STORAGE OF COS TO RECORD
1980  005302  013710  001624                  MOV     TODAT,(0)       ;NEW DATA FOR THAT MODULE.
1981  005306  013700  005330          4$:     MOV     SAV0,R0         ;RESTORE R0.
1982  005312  104430                          FOCTA                   ;NO-FORM ASCIZ STRING.
1983  005314  005237  001616                  INC     TPBSY           ;SET OUTPUT BUSY
1984  005320  000604                          BR      TOPINT          ;START OUTPUT
1985  005322                          1$:
1986  005322  000002                          EXIT
```

```
1987
1988  005324  000000                  6$:     0
1989  005326  000000                  7$:     0
1990  005330  000000                  SAVO:   0
1991  005332  005737  001616          TOLOP:  TST     TPBSY
1992  005336  001375                          BNE     TOLOP
1993  005340  000004                          SCOPE
1994  005342  104423                          CPATR
1995  005344  000137  004430                  JMP     TOOUTR
1996
1997                                          ;*
1998                                          ;*ROUTINE TO CONVERT 3 OCTAL NUMBERS TO AN ASCIZ STRING
1999                                          ;* CALL= FOCTA
2000
2001  005350  012703  032022          ROCTA:  MOV     #OUTBF,R3       ;SET UP BUFFER
2002  005354  013737  001622  005442          MOV     TOADR,PACK1
2003  005362  004737  005446                  JSR     PC,PACK         ;PACK ADDR.
2004  005366  013737  001624  005442          MOV     TODAT,PACK1
2005  005374  004737  005446                  JSR     PC,PACK         ;PACK DATA
2006  005400  013737  001626  005442          MOV     TOGEN,PACK1
2007  005406  001402                          BEQ     1$
2008  005410  004737  005446                  JSR     PC,PACK         ;PACK GEN CODE (IF ANY)
2009  005414  112723  000001          1$:     MOVB    #1,(3)+         ;FILLER CHARACTERS.
2010  005420  112723  000001                  MOVB    #1,(3)+
2011  005424  112723  000001                  MOVB    #1,(3)+
2012  005430  105023                          CLRB    (3)+            ;STRING TERMINATOR.
2013  005432  105023                          CLRB    (3)+
2014  005434  012703  032017                  MOV     #OUTBF1-1,R3    ;RESET POINTER
2015  005440  000002                          EXIT
2016
2017  005442  000000                  PACK1:  0
2018  005444  000000                  PACK2:  0
2019
2020  005446  012737  000260  005444  PACK:   MOV     #260,PACK2
2021  005454  005737  005442                  TST     PACK1
2022  005460  100002                          BPL     .+6
2023  005462  005237  005444                  INC     PACK2
2024  005466  113723  005444                  MOVB    PACK2,(3)+
2025  005472  000337  005442                  SWAB    PACK1
2026  005476  013737  005442  005444          MOV     PACK1,PACK2
2027  005504  006037  005444                  ROR     PACK2
2028  005510  006037  005444                  ROR     PACK2
2029  005514  006037  005444                  ROR     PACK2
2030  005520  006037  005444                  ROR     PACK2
2031  005524  042737  177770  005444          BIC     #177770,PACK2
2032  005532  052737  000260  005444          BIS     #260,PACK2
2033  005540  113723  005444                  MOVB    PACK2,(3)+
2034  005544  013737  005442  005444          MOV     PACK1,PACK2
2035  005552  006037  005444                  ROR     PACK2
2036  005556  042737  177770  005444          BIC     #177770,PACK2
2037  005564  052737  000260  005444          BIS     #260,PACK2
2038  005572  113723  005444                  MOVB    PACK2,(3)+
2039  005576  000337  005442                  SWAB    PACK1
2040  005602  013737  005442  005444          MOV     PACK1,PACK2
2041  005610  006037  005444                  ROR     PACK2
2042  005614  006037  005444                  ROR     PACK2
```

```
 2043  005620  006037  005444                        ROR     PACK2
 2044  005624  006037  005444                        ROR     PACK2
 2045  005630  006037  005444                        ROR     PACK2
 2046  005634  006037  005444                        ROR     PACK2
 2047  005640  042737  177770  005444                BIC     #177770,PACK2
 2048  005646  052737  000260  005444                BIS     #260,PACK2
 2049  005654  113723  005444                        MOVB    PACK2,(3)+
 2050  005660  013737  005442  005444                MOV     PACK1,PACK2
 2051  005666  006037  005444                        ROR     PACK2
 2052  005672  006037  005444                        ROR     PACK2
 2053  005676  006037  005444                        ROR     PACK2
 2054  005702  042737  177770  005444                BIC     #177770,PACK2
 2055  005710  052737  000260  005444                BIS     #260,PACK2
 2056  005716  113723  005444                        MOVB    PACK2,(3)+
 2057  005722  042737  177770  005442                BIC     #177770,PACK1
 2058  005730  052737  000260  005442                BIS     #260,PACK1
 2059  005736  113723  005442                        MOVB    PACK1,(3)+
 2060  005742  112723  000240                        MOVB    #240,(3)+
 2061  005746  112723  000240                        MOVB    #240,(3)+
 2062  005752  000207                                RTS     PC
 2063
 2064
 2065                                                ;*
 2066                                                ;*TEST 1 INPUT OR OUTPUT MODULE EXERCISER
 2067                                                ;*
 2068
 2069  005754  104400                    TST1:       TYPE                    ;TYPE HEADER.
 2070  005756  027733                                MHT1
 2071  005760  005077  173156                        CLR     @#TKS           ;DON'T ALLOW TTY INTERRUPTS
 2072  005764  005737  001636            5$:         TST     EXPERT          ;EXPERT MODE?
 2073  005770  001002                                BNE     .+6
 2074  005772  104400                                TYPE                    ;ASK "INPUT OR OUTPUT MODULE?"
 2075  005774  026234                                MIOO
 2076  005776  104400                                TYPE
 2077  006000  026403                                MQ
 2078  006002  052777  000041  173552                BIS     #XRIF+TTYEN,@#ICSR
 2079  006010  017737  173652  017140                MOV     @#ICSLMT,CHAR
 2080  006016  005077  173540                        CLR     @#ICSR
 2081  006022  005037  017140                        CLR     CHAR
 2082  006026  005737  000164            1$:         TST     REMFF           ;REMOTE RUNNING
 2083  006032  001451                                BEQ     6$              ;NO
 2084  006034  032777  002000  173520                BIT     #PWRFL,@#ICSR           ;POWER FAIL SENSED
 2085  006042  001402                                BEQ     .+6                     ;NO, CONT
 2086  006044  000137  025120                        JMP     RSTRT                   ;GO INDICATE POWER FAIL
 2087  006050  032777  000200  173504                BIT     #MODINT,@#ICSR          ;MODULE INTERRUPT POSTED
 2088  006056  001763                                BEQ     1$                      ;NO, THEN THERE IS NO DA TRY AGAIN
 2089  006060  005237  001700                        INC     MODFF                   ;INDICATE MOD INTR CASE NO DA
 2090  006064  032777  010000  173472                BIT     #DA,@#ICAR
 2091  006072  001013                                BNE     7$
 2092  006074  005737  001700                        TST     MODFF                   ;IS THER E MOD INHIBITING DA
 2093  006100  001752                                BEQ     1$                      ;NO GO TRY AGAIN
 2094  006102  005037  001700                        CLR     MODFF                   ;CLEAR MOD INTR TO ALLOW DA
 2095  006106  052777  000001  173446                BIS     #XRIF,@#ICSR
 2096  006114  005777  173546                        TST     @#ICSLMT
 2097  006120  000742                                BR      1$
 2098  006122  052777  000041  173432  7$:           BIS     #TTYEN+XRIF,@#ICSR
```

```
2099  006130  017737  173532  014626          MOV     @ICSLMT,RTEMP
2100  006136  104432                          WTBSY
2101  006140  013777  014626  173520          MOV     RTEMP,@ICSLMT
2102  006146  042777  000040  173406          BIC     #TTYEN,@ICSR
2103  006154  000406                          BR      2$
2104  006156  105777  172760          6$:     TSTB    @STKS           ;WAIT FOR RESPONSE
2105  006162  100321                          BPL     1$
2106  006164  117737  172754  014626          MOVB    @STKB,RTEMP
2107  006172  105777  172750          2$:     TSTB    @STPS           ;PRINTER BUSY?
2108  006176  100375                          BPL     2$
2109  006200  113777  014626  172742          MOVB    RTEMP,@STPB     ;ECHO CHARACTER.
2110  006206  142737  000240  014626          BICB    #240,RTEMP
2111  006214  123727  014626  000015          CMPB    RTEMP,#15
2112  006222  001412                          BEQ     4$
2113  006224  123727  014626  000003          CMPB    RTEMP,#3        ;WAS ↑C TYPED?
2114  006232  001002                          BNE     .+6
2115  006234  000137  003652                  JMP     START1          ;IF SO GOTO MONITR.
2116  006240  113737  014626  017140          MOVB    RTEMP,CHAR
2117  006246  000667                          BR      1$
2118  006250  104400                  4$:     TYPE
2119  006252  001161                          $CRLF
2120  006254  052777  000026  173300          BIS     #MODEN+PWFEN+ERREN,@ICSR
2121  006262  122737  000117  017140          CMPB    #'O,    CHAR    ;DID HE TYPE "O"?
2122  006270  001422                          BEQ     TST10           ;IF SO-GOTO OUTPUT ROUTINE
2123  006272  122737  000111  017140          CMPB    #'I,    CHAR    ;DID HE TYPE "I"?
2124  006300  001231                          BNE     5$              ;IF NOT RETYPE QUESTION
2125
2126                                          ;*ROUTINE TO HANDLE INPUTTING FROM INPUT MODULE TO DISPLAY
2127
2128  006302  104415                          INAR                    ;GET INPUT MODULE ADDR.
2129  006304  104400  027460                  TYPE,   MSWO
2130  006310  104400  026071                  TYPE,   MWK
2131
2132  006314  017700  006350          3$:     MOV     @INADR, R0      ;GET DATA FROM INPUT MODULE
2133  006320  005737  000164                  TST     REMFF
2134  006324  001003                          BNE     99$
2135  006326  010077  172606                  MOV     R0,@DISPLAY     ;PUT IN DISPLAY REGISTER IF 11/45.
2136  006332  104427                          INITR                   ;RESET SYSTEM
2137  006334  000767                  99$:    BR      3$
2138
2139                                          ;*ROUTINE TO HANDLE OUTPUTTING FROM SWITCH REGISTER (OR REMOTE TTY) TO OUTPUT MO
2140
2141  006336  104416                  TST10:  OUTAR                   ;GET OUTPUT MODULE ADDR.
2142  006340  104420                          DELAR                   ;GET DELAY TIME (DEFAULT=3MS).
2143  006342  005037  177776                  CLR     PS
2144  006346  104400  026071                  TYPE,   MWK
2145  006352  052777  000026  173202          BIS     #MODEN+PWFEN+ERREN,@ICSR
2146  006360  012777  000100  172554          MOV     #100,@STKS      ;ALLOW TTY INTERRUPTS
2147  006366  104432                  1$:     WTBSY                   ;WAIT FOR LINE TO GO INACTIVE
2148  006370  017777  172542  006332          MOV     @SWR,@OUTADR    ;SEND DATA FRO THE SWR TO OUTPUT MOD.
2149  006376  104422                          DELAY
2150  006400  000772                          BR      1$
2151
2152                                          ;*
2153                                          ;*TEST 2 DAC CALIBRATION ROUTINE
2154                                          ;*
```

2155

```
2156  006402  104400                TST2:   TYPE                    ;TYPE HEADER.
2157  006404  027775                        MHT2
2158  006406  104424                        IDAC                    ;INPUT DAC ADDR.
2159  006410  005737  015030              TST     DACADR            ;ANY DACS PRESENT
2160  006414  001005                      BNE     4$
2161  006416  104400                      TYPE                      ;MESSAGE "NO DAC ADDR. IN BUFFER.
2162  006420  025561                      MNDA
2163  006422  000137  006402              JMP     TST2
```

```
2164  006426  000000                    2$:     0
2165  006430  104400  026071            4$:     TYPE,   MWK
2166  006434  017737  172476  006426    1$:     MOV     @SWR,2$         ;GET VALUE OF SWR
2167  006442  042737  140017  006426            BIC     #140017, 2$
```

```
                                3$:     WTBSY                   ;WAIT FOR INACTIVE XMISSION LINE
006452  013777  006426  006350        MOV     $,@DACADR       ;SEND VALUE.
006460  062737  040000  006426        ADD     #40000,2$       ;SET FOR NEXT CH.
006466  103370                       BCC     3$              ;DONE ALL CHS?
006470  000761                       BR      1$
                                ;
                                ;TEST 3 - DAC INTERACTION TEST
                                ;       - STEP WAVEFORM IS DRAWN TO ALL CHANNELS IN A
                                ;       - STAGGERED FASHION
                                ;
006472  104400                  TST3:   TYPE                    ;TYPE HEADER.
006474  030032                          MHT3
006476  104424                          IDAC                    ;GET DAC ADDRESS
006500  005737  015030                  TST     DACADR          ;DAC PRESENT?
006504  001004                          BNE     1$
006506  104400                          TYPE                    ;NO-MESSAGE "NO DAC ADDR. IN BUFFER"
006510  025561                          MNDA
006512  000137  006472                  JMP     TST3
006516  104400  026071          1$:     TYPE,   MWK
006522  012700  000004                  MOV     #4,R0           ;SET CHANNEL COUNT
006526  005001                          CLR     R1              ;CLEAR CHAN SELECT
006530  012702  006602          2$:     MOV     #DACLST,R2      ;GET TABLE START
006534  012703  000026                  MOV     #22.,R3         ;TABLE START
006540  012204                  3$:     MOV     (2)+,R4         ;GET PATTERN
006542  006304                          ASL     R4
006544  006304                          ASL     R4
006546  006304                          ASL     R4
006550  006304                          ASL     R4
006552  050104                          BIS     R1,R4           ;SET CHAN SELECT
006554  005303                          DEC     R3              ;ALL LOADS DONE
006556  001404                          BEQ     4$              ;YES, 4$
006560  104432                          WTBSY                   ;WAIT FOR INACTIVE TRANSMISSION LINE
006562  010477  006242                  MOV     R4,@DACADR      ;LOAD DAC
006566  000764                          BR      3$
006570  062701  040000          4$:     ADD     #40000,R1       ;NEXT CHAN
006574  005300                          DEC     R0
006576  001354                          BNE     2$
006600  000750                          BR      1$+4
                                ;
                                ;TABLE OF DAC LOADS TO PRODUCE THE STEP
                                ;
006602  000000                  DACLST: 0000.   ;0.0000 VOLTS
006604  000063                          0051.   ;0.4976
006606  000146                          0102.   ;0.9751
006610  000231                          0153.   ;1.4927
006612  000314                          0204.   ;1.9902
006614  000377                          0255.   ;2.4870
006616  000462                          0306.   ;2.9854
006620  000545                          0357.   ;3.4829
006622  000630                          0408.   ;3.9805
006624  000713                          0459.   ;4.4780
006626  000776                          0510.   ;4.9756
```

```
2224  006630  001061                          0561.   ;5.4732
2225  006632  001144                          0612.   ;5.9707
2226  006634  001227                          0663.   ;6.4683
2227  006636  001312                          0714.   ;6.9658
2228  006640  001375                          0765.   ;7.4634
2229  006642  001460                          0816.   ;7.9610
2230  006644  001543                          0867.   ;8.4585
2231  006646  001626                          0918.   ;8.9561
2232  006650  001711                          0969.   ;9.4536
2233  006652  001774                          1020.   ;9.9512
2234
2235
2236
2237                                          ;*
2238                                          ;*TEST 4 COUNTER MODULE TEST
2239                                          ;*TEST 1 TO 16 MODULES
2240                                          ;*
2241
2242  006654  104400              TST4:       TYPE                    ;TYPE HEADER.
2243  006656  030067                          MHT4
2244  006660  104425                          CNTAR                   ;GET COUNTER MODULE ADDRS.
2245  006662  013737  001516  001534          MOV     FR40,FREQ
2246  006670  104400  026071                  TYPE,   MWK
2247
2248  006674  012700  014770      TST4L1:     MOV     #CNTADR,R0      ;GET LIST OF MODULES
2249  006700  010037  005442                  MOV     R0,PACK1
2250  006704  005710                          TST     (0)             ;ANY ADDRS. ENTERED?
2251  006706  001004                          BNE     TST4L
2252  006710  104400  026750                  TYPE    ,MNAE           ;"NO ADDRS ENTERED".
2253  006714  000137  006654                  JMP     TST4
2254
2255  006720  013700  005442      TST4L:      MOV     PACK1,R0
2256  006724  020027  015030                  CMP     R0,#DACADR      ;DONE ALL COUNTERS?
2257  006730  001007                          BNE     2$
2258  006732  005237  001100      1$:         INC     $PASS
2259  006736  104400  025610                  TYPE,   MEND
2260  006742  104400  026071                  TYPE,   MWK
2261  006746  000752                          BR      TST4L1
2262  006750  012037  001122      2$:         MOV     (0)+,$BDADR     ;GET ADDR. OF COUNTER MODULE.
2263  006754  001766                          BEQ     1$              ;NOTE: ZERO IF NO ADDR. ENTERED.
2264  006756  010037  005442                  MOV     R0,PACK1
2265
2266                                          ;*
2267                                          ;*TEST THAT THE COUNTER MODULE UNDER TEST CAN BE ADDRESSED
2268                                          ;*
2269
2270  006762  012737  006762  001106  TST4A:  MOV     #TST4A,$LPADR   ;SET FOR SCOPE, ITERATIONS.
2271  006770  104432                          WTBSY                   ;WAIT FOR TRANSMITTOR INACTIVE
2272  006772  012777  077777  172122          MOV     #77777,@$BDADR  ;SEND PATTERN.
2273  007000  104433                          ICRDLY                  ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2274  007002  017737  172114  001126          MOV     @$BDADR,$BDDAT  ;GET IT BACK
2275  007010  001001                          BNE     TST4AL          ;IF NON-ZERO - WE GOT SOMETHING BACK.
2276
2277  007012  104011                          ERROR   11              ;COULD NOT MAKE ANY DATA XFER
2278  007014  000004              TST4AL:     SCOPE
```

```
2279
2280                                            ;*
2281                                            ;*TEST THAT COUNTER MODULE BITS 00 THRU 15 CAN BE SET AND CLEARED
2282                                            ;*
2283
2284  007016  012737  000001  014626  TST4B:  MOV     #1,RTEMP          ;SET "GOOD DATA"
2285  007024  012737  000010  001156           MOV     #10,$TIMES
2286  007032  012737  007032  001106  1$:     MOV     #1$,$LPADR        ;SET FOR ITERATIONS
2287  007040  013737  014626  001124           MOV     RTEMP,$GDDAT
2288  007046  104432                          WTBSY                     ;WAIT FOR TRANSMITTOR INACTIVE
2289  007050  013777  001124  172044           MOV     $GDDAT,@$BDADR    ;SEND PATTERN TO COUNTER MODULE.
2290  007056  104433                          ICRDLY                    ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2291  007060  017737  172036  001126           MOV     @$BDADR,$BDDAT    ;GET IT BACK.
2292  007066  023737  001124  001126           CMP     $GDDAT,$BDDAT     ;DATA SENT=DATA RECEIVED?
2293  007074  001402                          BEQ     2$
2294
2295  007076  104012                          ERROR   12                ;NO - REPORT ERROR
2296  007100  000414                          BR      3$                ;LOOP
```

```
2297
2298  007102  104432                  2$:   WTBSY                   ;WAIT FOR TRANSMITTOR INACTIVE
2299  007104  043777  001124  172010        BIC     $GDDAT,@$BDADR  ;TRY CLEARING BIT.
2300  007112  104433                        ICRDLY                  ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2301  007114  017737  172002  001126        MOV     @$BDADR,$BDDAT  ;DID IT CLEAR?
2302  007122  001403                        BEQ     .+6             ;YES-LOOP.
2303  007124  005037  001124                CLR     $GDDAT
2304  007130  104012                        ERROR   12
2305
2306  007132  000004                  3$:   SCOPE
2307  007134  006137  014626                ROL     RTEMP           ;SET TO NEXT BIT.
2308  007140  103334                        BCC     1$              ;CONTINUE TESTING IF ALL BITS NOT DONE.
2309
2310                                        ;*
2311                                        ;*TEST THAT THE COUNTER MODULE CAN COUNT THRU
2312                                        ;*EACH STATE
2313                                        ;*
2314
2315  007142  012737  010132  014626  TST4C: MOV     #CNTPAT,RTEMP   ;GET ADDR. OF PATTERNS
2316  007150  012737  000010  001156  1$:   MOV     #10,$TIMES
2317  007156  012737  007150  001106        MOV     #1$,$LPADR
2318
2319  007164  104432                        WTBSY                   ;WAIT FOR TRANSMITTOR INACTIVE
2320  007166  017777  005434  171726        MOV     @RTEMP,@$BDADR  ;SEND PATTERN.
2321  007174  005037  014630                CLR     RTEMP1
2322  007200  017737  005422  001124        MOV     @RTEMP,$GDDAT
2323  007206  005237  001124                INC     $GDDAT
2324  007212  005237  014630          4$:   INC     RTEMP1          ;WATCH TO SEE IF COUNTER COUNTS.
2325  007216  001414                        BEQ     5$              ;IF NO COUNT BY OVERFLOW-ERROR!
2326  007220  104433                        ICRDLY                  ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2327  007222  017737  171674  001126        MOV     @$BDADR,$BDDAT
2328  007230  027737  005372  001126        CMP     @RTEMP,$BDDAT
2329  007236  001765                        BEQ     4$
2330  007240  023737  001124  001126        CMP     $GDDAT,$BDDAT
2331  007246  001401                        BEQ     2$
2332
2333  007250  104013                  5$:   ERROR   13
2334  007252  000004                  2$:   SCOPE
2335  007254  062737  000002  014626        ADD     #2,RTEMP
2336  007262  023727  014626  010176        CMP     RTEMP,#CNTPAE
2337  007270  002727                        BLT     1$
2338
2339                                        ;*
2340                                        ;*TEST TO SEE IF COUNTER INITIALIZES PROPERLY--PART 1
```

```
2341                                                    ;*
2342
2343  007272  012737  000010  001156  TST4D:  MOV     #10,$TIMES
2344  007300  012737  007300  001106  1$:     MOV     #1$,$LPADR
2345  007306  104432                          WTBSY                   ;WAIT FOR TRANSMITTOR INACTIVE
2346  007310  005077  171606                  CLR     @$BDADR
2347  007314  052777  000100  172240          BIS     #XRESET,@ICSR
2348  007322  104433                          ICRDLY
2349  007324  005037  001124                  CLR     $GDDAT
2350  007330  104433                          ICRDLY                  ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2351  007332  017737  171564  001126          MOV     @$BDADR,$BDDAT
2352  007340  001401                          BEQ     2$
2353
2354  007342  104025                          ERROR   25
2355
2356  007344  000004                  2$:     SCOPE
2357
2358                                                    ;*
2359                                                    ;*TEST THAT THE COUNTER INITIALIZES PROPERLY PART 2
2360                                                    ;*
2361
2362  007346  012737  000010  001156  TST4E:  MOV     #10,$TIMES      ;SET ITERATION COUNT
2363  007354  012737  007366  001106          MOV     #1$,$LPADR      ;SET LOOP ADDRESS.
2364  007362  005037  001124                  CLR     $GDDAT
2365  007366  104432                  1$:     WTBSY                   ;WAIT FOR TRANSMITTOR INACTIVE
2366  007370  012777  177777  171524          MOV     #177777,@$BDADR ;SET THE COUNTER.
2367  007376  104433                          ICRDLY
2368  007400  000240                          NOP
2369  007402  052777  000100  172152          BIS     #XRESET,@ICSR
2370  007410  104433                          ICRDLY                  ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2371  007412  052777  040000  172142          BIS     #MAINT3,@ICSR
2372  007420  000240                          NOP
2373  007422  000240                          NOP
2374  007424  017737  171472  001126          MOV     @$BDADR,$BDDAT  ;READ COUNTER.
2375  007432  001401                          BEQ     2$
```

```
                                        ERROR   20

                                2$:     SCOPE

                                        ;*
                                        ;*TEST THAT THE COUNTER HAS NO
                                        ;*INTERRUPTS POSTED ON THE BUS NOR ANY GENERIC CODE
                                        ;*

007440  032777  000200  172114  TST4F:  BIT     #200,@ICSR              ;ANY INTERRUPTS POSTED?
007446  001401                        BEQ     1$

007450  104021                        ERROR   21              ;ILLEGAL INTERRUPT ON ICR BUS

007452  000004                1$:     SCOPE

007454  012737  007520  001612        MOV     #2$,ICRVT       ;SET FOR INTERRUPT
007462  052777  020004  172072        BIS     #MAINT2+MODEN,@ICSR ;SET MAINT + INT ENABLE
007470  005037  177776                CLR     PS
007474  104433                        ICRDLY                  ;WAIT 3.2 MILLISEC FOR ROUND TRIP
007476  012737  000340  177776        MOV     #340,@#PSW
007504  052777  040000  172050        BIS     #MAINT3,@ICSR
007512  000240                        NOP

007514  104022                        ERROR   22              ;FATAL ERROR - ICR DID NOT INTERRUPT
007516  000432                        BR      3$

                                        ;ICR INTERRUPTS TO HERE (RETURN FROM ICRVT)

007520  017737  172040  001126  2$:     MOV     @ICAR,$BDDAT    ;READ ICAR.
007526  022626                        POPSP2                  ;READJUST STACK
007530  012777  000001  172024        MOV     #1,@ICSR
007536  005037  001124                CLR     $GDDAT
007542  005777  171354                TST     @$BDADR
007546  052777  040000  172006        BIS     #40000,@ICSR
007554  052777  000024  172000        BIS     #MODEN+PWFEN,@ICSR
007562  005037  177776                CLR     PS
007566  042737  000360  001126        BIC     #360,$BDDAT     ;IGNORE FILE BOX ADDR.
007574  005737  001126                TST     $BDDAT          ;NO OTHER ADDR. OR GEN BITS
007600  001401                        BEQ     3$              ;SHOULD SHOW UP

007602  104023                        ERROR   23

007604  000004                3$:     SCOPE

                                        ;*
                                        ;*TEST THAT THE COUNTER MODULE WILL INTERRUPT ON OVERFLOW.
                                        ;*ON INTERRUPT CHECK ADDR AND GENERIC CODE, HALT ON OVERFLOW.
                                        ;*CHECK RIF EFFECT ON MODULE
                                        ;*

007606  012737  007700  001612  TST4G:  MOV     #TST4GI,ICRVT   ;SET UP FOR ICR INTERRUPT.
007614  012777  000340  171746        MOV     #340,@ICSVT2    ;PRIORITY 7 ON INTERRUPT.
007622  012737  007630  001106        MOV     #TST4GL,$LPADR  ;SET FOR ITERATIONS.
```

```
2432  007630  012737  000340  177776  TST4GL: MOV     #340,PS             ;DON'T ALLOW INTERRUPTS.
2433  007636  052777  000026  171716          BIS     #MODEN+ERREN+PWFEN,@ICSR    ;ENABLE ICR TO INTERRUPT WHEN READY.
2434  007644  104432                          WTBSY                       ;WAIT FOR TRANSMITTOR INACTIVE
2435  007646  012777  177777  171246          MOV     #177777,@$BDADR     ;SET COUNTER TO ALL ONES.
2436  007654  104433                          ICRDLY                      ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2437  007656  104422                          DELAY                       ;DELAY 40 MS.
2438  007660  005037  177776                  CLR     PS                  ;ALLOW INTERRUPTS.
2439  007664  000240                          NOP
2440  007666  005077  171670                  CLR     @ICSR
2441  007672  104014                          ERROR   14                  ;REPORT ERROR - COUNTER MODULE DIDN'T INTR.
2442  007674  104427                          INITR
2443  007676  000462                          BR      TST4GE              ;LOOP.
2444
2445                                          ;*RECEIVE ICR INTERRUPT HERE (RETURN FROM ICRVT)
2446
2447  007700  012706  001100          TST4GI: MOV     #1100,SP            ;RESET THE STACK POINTER.
2448  007704  017737  171654  001126        MOV     @ICAR,$BDDAT        ;GET ADDR & GENERIC CODE.
2449  007712  013737  001122  001124        MOV     $BDADR,$GDDAT       ;GET REAL ADDR.
2450  007720  006037  001124                ROR     $GDDAT              ;FORM ADDR. AS IT WOULD LOOK IN
2451  007724  142737  177777  001125        BICB    #-1,$GDDAT+1        ;ICAR.
2452  007732  005077  171624                CLR     @ICSR
2453  007736  005037  001612                CLR     ICRVT
2454  007742  005037  177776                CLR     PS
2455  007746  052737  003400  001124        BIS     #003400,$GDDAT      ;ADD GENERIC CODE.
2456  007754  023737  001124  001126        CMP     $GDDAT,$BDDAT       ;CHECK ADDR. + GEN. CODE.
2457  007762  001403                        BEQ     1$
2458
2459  007764  104015                        ERROR   15                  ;ADDR OR GENERIC CODE INCORRECT.
2460  007766  104427                        INITR
2461  007770  000425                        BR      TST4GE              ;LOOP.
2462
2463  007772  104422                  1$:   DELAY                       ;DELAY 40 MS.
2464  007774  005037  001124                CLR     $GDDAT
2465  010000  017737  171116  001126        MOV     @$BDADR,$BDDAT
2466  010006  001402                        BEQ     2$
2467
2468  010010  104016                        ERROR   16                  ;COUNTER MODULE DIDN'T HALT ON OVERFLOW.
2469  010012  000414                        BR      TST4GE              ;LOOP.
2470
2471  010014  052777  000001  171540  2$:   BIS     #1,@ICSR            ;SET RIF BIT IN ICS-11.
2472  010022  005777  171074                TST     @$BDADR             ;INITIATE RIF ON COUNTER MODULE.
2473  010026  000240                        NOP                         ;
2474  010030  032777  000200  171524        BIT     #200,@ICSR          ;ANY INTR. PENDING ON ICS BUS?
2475  010036  001402                        BEQ     TST4GE              ;NO-THEN LOOP.
2476
2477  010040  104017                        ERROR   17                  ;RIF DIDN'T CLEAR INTERRUPT FLAG
2478  010042  104427                        INITR
2479  010044  052777  000001  171510  TST4GE: BIS     #1,@ICSR
2480  010052  005777  171044                TST     @$BDADR
2481  010056  000004                        SCOPE
2482
2483                                          ;*
2484                                          ;*TEST TO SEE IF RESET CLEARS INTERRUPT FLAG ON COUNTER
2485                                          ;*
2486
2487  010060  012737  000010  001156  TST4H:  MOV     #10,$TIMES
```

```
2488  010066  104432                  1$:   WTBSY                   ;WAIT FOR TRANSMITTOR INACTIVE
2489  010070  012777  177777  171024        MOV     #-1,@$BDADR     ;START COUNTER COUNTING
2490  010076  012737  010066  001106        MOV     #1$,$LPADR
2491  010104  104433                        ICRDLY                  ;WAIT 3.2 MILLISEC FOR ROUND TRIP
2492  010106  104422                        DELAY                   ;DELAY FOR INTR.
2493  010110  104427                        INITR                   ;SYSTEM INITIALIZE
2494  010112  032777  000200  171442        BIT     #200,@ICSR      ;INTERRUPT POSTED?
2495  010120  100001                        BPL     2$
2496
2497  010122  104024                        ERROR   24              ;INIT FAILED TO CLEAR INTR. FLAG
2498  010124  000004                  2$:   SCOPE
2499  010126  000137  006720              JMP     TST4L
2500
2501  010132  000000                  CNTPAT: 0                       ;PATTERNS USED BY TST4C TO
2502  010134  000001                          1                       ;DETERMINE IF COUNTER MODULE
2503  010136  000003                          3                       ;CAN COUNT THRU EACH STATE.
2504  010140  000007                          7
2505  010142  000017                          17
2506  010144  000037                          37
2507  010146  000077                          77
2508  010150  000177                          177
2509  010152  000377                          377
2510  010154  000777                          777
2511  010156  001777                          1777
2512  010160  003777                          3777
2513  010162  007777                          7777
2514  010164  017777                          17777
2515  010166  037777                          37777
2516  010170  077777                          77777
2517  010172  177777                          177777
2518  010174  100000                          100000
2519  010176  000000                  CNTPAE: 0
2520
2521
```

```
                                                ;*
                                                ;*TEST 5 A/D LOGIC TEST
                                                ;*
010200  104400                          TST5:   TYPE                    ;TYPE HEADER.
010202  030130                                  MHT5
010204  104426                                  ADAR                    ;GET ADO5 ADDR.
010206  012746  011202                  ADLOGL: MOV     #ADLOGE,-(6)
010212  104400  026071                          TYPE,   MWK
010216  005037  001612                          CLR     ICRVT           ;CLEAR INTERRUPT SERVICE POINTER
010222  000240                                  NOP
010224  000240                                  NOP
010226  000240                                  NOP
010230  052777  000026  171324                  BIS     #ERREN+PWFEN+MODEN,@ICSR

010236  013737  015032  001122  ADLOG:  MOV     ADADR,$BDADR    ;*GET ADO5'S ADDR.
010244  052777  000100  170670          BIS     #100,@$TKS
010252  005037  177776                  CLR     PS
010256  012737  010276  001106          MOV     #1$,$LPADR
010264  012703  011216                  MOV     #ADPATP,R3      ;*GET PATTERN POINTERS.
010270  012737  077770  001124          MOV     #077770,$GDDAT  ;*DATA TO BE SENT
010276  104432                  1$:     WTBSY                   ;WAIT FOR INACTIVE LINE
010300  013777  001124  170614          MOV     $GDDAT,@$BDADR  ;*SEND DATA TO BE SENT
010306  104433                          ICRDLY
010310  017737  170606  001126          MOV     @$BDADR,$BDDAT  ;*GET DATA BACK.
010316  001001                          BNE     2$              ;*IF DATA PRESENT GO AHEAD

010320  104001                          ERROR   1               ;*ERROR "COULD NOT SEND /RECEIVE DATA"
                                                                ;*FROM ADO5
010322  000004                  2$:     SCOPE
                                        ;*BASIC "BIT BANG" OF ADO5

010324  011337  001124          ADLOG2: MOV     (3),$GDDAT      ;*GET PATTERN TO BE SENT.
010330  104432                          WTBSY                   ;WAIT FOR INACTIVE
010332  013777  001124  170562          MOV     $GDDAT,@$BDADR  ;*SEND PATTERN TO ADCSR.
010340  104433                          ICRDLY                  ;WAIT
010342  017737  170554  001126          MOV     @$BDADR,$BDDAT  ;*GET TO BACK.
010350  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;*DATA SENT=DATE RECIEVED?
010356  001401                          BEQ     2$              ;*IF SO-CONTINUE

010360  104002                          ERROR   2               ;*REPORT ERROR: "CSR READ/WRITE ERROR"

010362  000004                  2$:     SCOPE                   ;*LOOP

010364  062703  000002          1$:     ADD     #2,R3           ;*UPDATE PATTERN POINTERS
010370  020327  011240                  CMP     R3,#ADPATE      ;*DONE ALL PATTERNS?
010374  003753                          BLE     ADLOG2          ;*IF NOT-CONTINUE NEXT PATTERN.

                                        ;*WILL CONVERTING SET AND THEN CLEAR?

010376  012737  104000  001124  ADLOG3: MOV     #104000,$GDDAT  ;*SET CONVERT,READ CSR
010404  012737  010404  001106  1$:     MOV     #1$,$LPADR
010412  005000                          CLR     R0
010414  104432                          WTBSY
010416  013777  001124  170476          MOV     $GDDAT,@$BDADR  ;*SEND TO ADCSR
010424  104432                          WTBSY
```

```
2578  010426  017737  170470  001126  2$:     MOV     @$BDADR,$BDDAT  ;*GET ADO5 CSR
2579  010434  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;*DID CONVERT BIT SET?
2580  010442  001404                          BEQ     ADL03A
2581  010444  005300                          DEC     R0              ;WAITED LONG ENOUGH
2582  010446  001367                          BNE     2$              ;NO, KEEP TRYING
2583
2584
2585  010450  104003                          ERROR   3               ;*ERROR: "CONVERT BIT FAILED TO SET"
2586  010452  000412                          BR      ADLOG4
2587
2588  010454  013737  001522  001534  ADL03A: MOV     FR16,FREQ       ;*SET TO DELAY 16 MILLISEC.
2589  010462  104422                          DELAY
2590  010464  104433                          ICRDLY
2591  010466  005777  170430                  TST     @$BDADR         ;*DID CONVERT BIT CLEAR?
2592  010472  100002                          BPL     1$+2
2593
2594  010474  104004                          ERROR   4               ;*ERROR "CONVERT BIT FAILED TO CLEAR"
2595
2596                                           ;*LOOP.
2597                                           ;*NEXT TEST.
2598  010476  104427          1$:     INITR   ;*ISSUE SYSTEM INITIALIZE
2599  010500  000004          ADLOG4: SCOPE
2600
2601                                  ;*CAN WE READ THE DBR WITH CSR BIT//CLEAR (READ BIT).
2602
2603  010502  104432          ADLOG5: WTBSY
2604  010504  012777  052770  170410          MOV     #52770,@$BDADR  ;*LOAD CSR WITH ALL BUT READ CR BIT
2605  010512  104433                          ICRDLY
2606  010514  022777  052770  170400          CMP     #52770,@$BDADR  ;*DID WE READ THE CR?
2607  010522  001001                          BNE     1$              ;*IF NOT NO ERROR.
2608
2609  010524  104005                          ERROR   5               ;*REPORT ERROR "CANNOT READ ADO5 DATA
2610                                           ;*REGISTER WITH READ BIT CLEARED."
2611
2612  010526  000004          1$:     SCOPE
2613
2614                                  ;*WILL CONVERT DONE CAUSE INTERRUPT?
2615
2616  010530  012777  000340  171032  ADLOG6: MOV     #340,@ICSVT2
2617  010536  012737  010660  001612          MOV     #ADLOG7,ICRVT   ;*SET VECTOR FOR INTER.
2618  010544  042777  000002  171010          BIC     #ERREN,@ICSR
2619  010552  052777  000025  171002          BIS     #MODEN+PWFEN+XRIF,@ICSR ;*ALLOW ICR TO INT.
2620  010560  104432                          WTBSY
2621  010562  012777  104000  170332          MOV     #104000,@$BDADR ;*SET ADO5 TO CONVERT.
2622  010570  013737  001522  001534          MOV     FR16,FREQ       ;*ALLOW UP TO 16 MILLI SEC FOR NTO.
2623  010576  005237  001602                  INC     ADBSY
2624  010602  005037  177776                  CLR     PS
2625  010606  104433                          ICRDLY
2626  010610  104422                          DELAY
2627  010612  000240                          NOP
2628  010614  104006                          ERROR   6               ;*REPORT ERROR "ADO5 FAILED TO INTER. AT
2629  010616  104432          ADL6L:  WTBSY
2630  010620  005077  170736                  CLR     @ICSR
2631  010624  042777  000026  170730          BIC     #ERREN+PWFEN+MODEN,@ICSR
2632  010632  052777  000026  170722          BIS     #ERREN+PWFEN+MODEN,@ICSR
2633  010640  000004                          SCOPE
```

```
2634  010642  042777  000026  170712          BIC     #ERREN+PWFEN+MODEN,@ICSR
2635  010650  052777  000026  170704          BIS     #ERREN+PWFEN+MODEN,@ICSR
2636  010656  000460                          BR      ADLOG9
2637
2638                                          ;*CHECK INTR. ADDR. IN ICAR+GENERIC CODE (RETURNED FROM ICRVT)
2639
2640  010660  022626                  ADLOG7: POPSP2                  ;*A005 INTR TO HERE RESET SP.
2641  010662  017737  170676  001126          MOV     @ICAR,$BDDAT    ;*GET ADDR. AND GEN CODE
2642  010670  005037  001602                  CLR     ADBSY
2643  010674  005737  001604                  TST     DAFLG
2644  010700  001406                          BEQ     1$
2645  010702  012746  000340                  MOV     #340,-(SP)
2646  010706  012746  010716                  MOV     #1$,-(SP)
2647  010712  000137  020202                  JMP     SWR0U1
2648  010716  013737  001122  001124  1$:     MOV     $BDADR,$GDDAT   ;*FORM GOOD ADDR.
2649  010724  042737  177000  001124          BIC     #177000,$GDDAT
2650  010732  006237  001124                  ASR     $GDDAT
2651  010736  052737  003400  001124          BIS     #003400,$GDDAT  ;*ADD GENERIC CODE
2652  010744  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;*IS ADDR + GENERIC CODE OK?
2653  010752  001402                          BEQ     ADLOG8
2654
2655  010754  104007                          ERROR   7               ;*REPORT ERROR "A005 ADDR. OR GENERIC
2656
2657                                          ;*CODE INCORRECT.
2658  010756  000717                          BR      ADL6L
2659
2660                                          ;*CHECK TO SEE IF RIF CLEARS INTR FLAG ON A005
2661
2662  010760  052777  000001  170574  ADLOG8: BIS     #1,@ICSR        ;*SET RIF BIT.
2663  010766  005777  170130                  TST     @$BDADR         ;*PUT IT TO WORK ON A005.
2664  010772  012737  011012  001566          MOV     #1$,ICSVT       ;*SET INTERRUPT VECTORS.
2665  011000  005037  177776                  CLR     PS              ;*ALLOW INTERRUPTS
2666  011004  104422                          DELAY
2667  011006  000240                          NOP
2668  011010  000702                          BR      ADL6L           ;*EXIT A/D TESTS.
2669
2670  011012  022626                  1$:     POPSP2                  ;*RESET SP.
2671  011014  104010                          ERROR   10              ;*REPORT ERROR" RIF DID NOT CLEAR INTR.
2672                                          ;*FLAG ON A005".
2673  011016  000677                          BR      ADL6L
2674
2675                                          ;*
2676                                          ;*A005 DUAL ADDRESSING TEST
2677                                          ;*
2678
2679  011020  012737  011120  001106  ADLOG9: MOV     #2$,$LPADR
2680  011026  005037  001124                  CLR     $GDDAT          ;SET FIRST ADDR.
2681  011032  005037  001612                  CLR     ICRVT
2682  011036  005037  177776                  CLR     PS
2683  011042  053737  001666  001124          BIS     ICSLMT,$GDDAT
2684  011050  162737  000002  001124          SUB     #2,$GDDAT
2685  011056  013737  001666  014626          MOV     ICSLMT,RTEMP    ;FIX STOP ADDR.
2686  011064  062737  000040  014626          ADD     #40,RTEMP
2687  011072  062737  000002  001124  1$:     ADD     #2,$GDDAT       ;UPDATE ADDR.
2688  011100  023737  001124  014626          CMP     $GDDAT,RTEMP    ;DONE ALL ADDRS?
2689  011106  001433                          BEQ     6$
```

```
2690  011110  023737  001124  001122          CMP     $GDDAT,$BDADR   ;SAME ADDR AS ADD5 UNDER TEST?
2691  011116  001765                          BEQ     1$
2692  011120  104432                  2$:     WTBSY
2693  011122  012777  004100  167772          MOV     #4100,@$BDADR   ;SEND DATA TO A/D.
2694  011130  104433                          ICRDLY
2695  011132  022777  004100  167764          CMP     #4100,@$GDDAT   ;DUAL ADDR. READ ERROR?
2696  011140  001001                          BNE     3$
2697
2698  011142  104026                          ERROR   26              ;DUAL ADDR. READ ERROR
2699  011144  000004                  3$:     SCOPE
2700
2701  011146  104432                  4$:     WTBSY
2702  011150  012777  004200  167746          MOV     #4200,@$GDDAT   ;CHECK FOR WRITE ERROR
2703  011156  104433                          ICRDLY
2704  011160  022777  004200  167734          CMP     #4200,@$BDADR   ;BAD WRITE?
2705  011166  001001                          BNE     5$
2706
2707  011170  104027                          ERROR   27              ;DUAL ADDR. WRITE ERROR
2708
2709  011172  000004                  5$:     SCOPE
2710  011174  000736                          BR      1$
2711  011176  104427                  6$:     INITR
2712  011200  000207                          RTS     PC              ;RETURN TO
2713                                                                  ;ADLOGE
2714
2715                                          ;*END OF A/D TESTS
2716
2717  011202  005237  001100          ADLOGE: INC     $PASS
2718  011206  104400                  1$:     TYPE                    ;*THE "END"
2719  011210  025610                          MEND
2720  011212  000137  010206                  JMP     ADLOGL          ;*RETURN TO MONITOR.
2721
2722  011216  004000                  ADPATP: 004000
2723  011220  004020                          004020
2724  011222  004040                          004040
2725  011224  004100                          004100
2726  011226  004200                          004200
2727  011230  004400                          004400
2728  011232  005000                          005000
2729  011234  006000                          006000
2730  011236  004000                          004000
2731  011240  004000                  ADPATE: 004000
2732
2733  011242  000137  003652                  JMP     START1
2734
2735                                          ;*
2736                                          ;*TEST 6 A/D CALIBRATION TEST
2737                                          ;*
2738
2739  011246  104400                  TST6:   TYPE                    ;TYPE HEADER.
2740  011250  030164                          MHT6
2741  011252  104426                          ADAR                    ;GET ADD5 ADDR.
2742  011254  104431                          QUBR                    ;UNI OR BI-POLAR?
2743                                          ;TBCA AUDO1 CALIBRATION ROUTINE
2744                                          ;SWR0=1 RETURN TO MONITR
2745                                          ;SWR4-6 SELECT CHANNEL
```

```
2746                                                    ;SWR7-10 SELECT MUX.
2747                                                    ;SWR11=1 TYPE OUT CONVERSIONS
2748                                                    ;SWR11=0 DISPLAY CONVERSIONS (11/45)
2749                                                    ;SWR12-14 SELECT GAIN (1 TO 1000)
2750
2751  011256  104400  027460                 TYPE,   MSWO
2752  011262  104400  026071                 TYPE,   MWK
2753  011266  052777  000100  167646  T8CAA: BIS     #100,@$TKS      ;ALLOW TTY INTERRUPT
2754  011274  013737  015032  001122        MOV     ADADR,$BDADR
2755  011302  001004                        BNE     1$
2756  011304  104400                        TYPE
2757  011306  027431                        MNAD
2758  011310  000137  003652                JMP     START1
2759  011314  004737  012130          1$:   JSR     PC,CONVER       ;START CONVERSION
2760  011320  104433                        ICRDLY
2761  011322  017700  167574                MOV     @$BDADR,        %0      ;GET RESULTS
2762  011326  004737  012116                JSR     PC,REPET7       ;RIGHT JUSTIFY
2763  011332  032777  004000  167576        BIT     #4000,@SWR      ;DISPLAY OR TYPE DATA?
2764  011340  001006                        BNE     CALTP           ;TYPE IT
2765  011342  005737  000164                TST     REMFF
2766  011346  001347                        BNE     T8CAA
2767  011350  010077  167564                MOV     R0,@DISPLAY
2768  011354                          2$:
2769  011354  000744                        BR      T8CAA
2770  011356  104400                  CALTP: TYPE
2771  011360  001161                        $CRLF
2772  011362  004737  011430                JSR     %7,CALIT        ;CONVERT DATA TO BINARY
2773  011366  013737  001512  001534        MOV     FR110,FREQ
2774  011374  104422                        DELAY                   ;LET TTY SETTLE DOWN.
2775  011376  104400                        TYPE
2776  011400  027604                        MCALT1
2777  011402  104433                        ICRDLY
2778  011404  017700  167512                MOV     @$BDADR,        %0      ;GET DATA
2779  011410  004737  012116                JSR     PC,REPET7
2780  011414                                TYPOCT  R0
2781  011414  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
2782  011416  104401                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2783  011420  104400                        TYPE
2784  011422  027573                        MCALOT
2785  011424  104422                        DELAY
2786  011426  000717                        BR      T8CAA
2787  011430  005737  015034          CALIT: TST     BIPOL
2788  011434  001406                        BEQ     3$
2789  011436  022700  007777                CMP     #7777,R0
2790  011442  001017                        BNE     2$
2791  011444  104400                        TYPE
2792  011446  027533                        MPOVFL
2793  011450  000207                        RETURN
2794  011452                          3$:
2795  011452  022700  004000                CMP     #4000,  %0      ;ROUTINE TO CONVERT BINARY
2796  011456  001003                        BNE     1$              ;READ FROM A/D BUFFER TO SIX
2797  011460  104400                        TYPE                    ;PLACE DECIMAL VOLTAGE
2798  011462  027514                        MMOVFL
2799  011464  000207                        RTS     PC
2800  011466  022700  003777          1$:   CMP     #3777,  %0      ;POSITIVE OR NEGATIVE
2801  011472  001003                        BNE     2$
```

```
                                            TYPE
                                            MPOVFL
                                            RTS     PC
012000                      2$:             MOV     #A1,    %3      ;CLEAR STORAGE AREA
                                            CLR     (3)+
012014                                      CMP     %3,     #A1+14  ;DETERMINE IF
                                            BNE     .-6
012000                                      MOV     #A1,    %3      ;POSITIVE OR
002000                                      MOV     #2000,  %2
015034                                      TST     BIPOL
                                            BEQ     .+4
012022                                      MOV     #HOLDIT,%1      ;NEGATIVE VOLTAGE
015034                                      TST     BIPOL
                                            BNE     CALIT1
004000                                      BIT     #4000,R0
                                            BEQ     CALITP
                                            TYPE                    ;IF MINUS TYPE "-"
                                            MMINUS
                                            COM     %0
                                            INC     %0
                                            BR      CALIT1          ;IF POSITIVE
                            CALITP:         TYPE                    ;TYPE "+"
                                            MPLUS
                            CALIT1:         BIT     %2,     %0
                                            BNE     CALIT2          ;DO OCTAL TO
000005                                      ADD     #5,     %1
                            CALITE:         ROR     %2              ;DECIMAL CONVERSIONS
                                            BNE     CALIT1
011776                                      TST     HOLDTM          ;BY METHOD OF INCREMENTING
                                            BEQ     .+10
000001  012000                             MOVB    #1,     A1
                                            CLR     HOLDTM          ;DECIMAL COUNTER AND
012000                                      MOV     #A1,    %3
000260                                      BIS     #260,   (3)+
012014                                      CMP     #A1+14, %3      ;DECREMENTING OCTAL DATA
                                            BNE     .-10
012001                                      MOV     #A1+1,  %3      ;UNTIL ZERO
000001                                      MOVB    #1,     (3)+
                                            INCB    %3
000256                                      MOVB    #256,   (3)+    ;IF DECIMAL COUNTER = 12 (OCTAL)
                                            INCB    %3
000001                                      MOVB    #1,     (3)+
012014                                      CMP     #A1+14, %3      ;CLEAR IT AND INCREMENT
                                            BNE     .-12            ;NEXT COUNTER
                                            CLRB    (3)
                                            TYPE                    ;TYPE DECIMAL
                                            A1
                                            RTS     %7
011776                      CALIT2:         CLR     HOLDTM
012014                                      MOV     #A1+14, %3      ;FIRST USE TABLE HOLDIT
011774                      CALIT3:         MOVB    (1)+,   SCAN            ;TO GET DECIMAL VALUE FOR DATA
011774                                      ADD     SCAN,   -(3)    ;COMPUTED FROM ADDS
011776                                      TST     HOLDTM
                                            BEQ     CALIT4
011776                                      CLR     HOLDTM          ;ADD THESE VALUES TOGETHER
```

```
2858  011742  062713  000001            ADD     #1,     (3)
2859  011746  122713  000012  CALIT4:   CMPB    #12,    (3)     ;HOLDIT CONTAINS VALUES FOR DATA
2860  011752  101004                    BHI     CALIT5
2861  011754  162713  000012            SUB     #12,    (3)     ;BASED ON GAIN OF A/D
2862  011760  005237  011776            INC     HOLDTM
2863  011764  022703  012002  CALIT5:   CMP     #A1+2,  %3      ;AT TIME DATA WAS TAKEN
2864  011770  001353                    BNE     CALIT3
2865  011772  000704                    BR      CALITE
2866  011774  000000  SCAN:   0
2867  011776  000000  HOLDTM: 000000
2868  012000  000000  A1:     000000
2869          012022  .=.+20
2870  012022    000     000     000   HOLDIT: .BYTE   0,0,0,0,5      ;TABLE OF VOLTAGE REPRESENTATION
2871  012025    000     005
2872
2873  012027    000     000     000           .BYTE   0,0,0,5,2      ;OF BINARY INPUT.
2874  012032    005     002
2875
2876  012034    000     000     005           .BYTE   0,0,5,2,1
2877  012037    002     001
2878
2879  012041    000     005     002           .BYTE   0,5,2,6,0
2880  012044    006     000
2881
2882  012046    005     002     001           .BYTE   5,2,1,3,0
2883  012051    003     000
2884
2885  012053    003     006     005           .BYTE   3,6,5,1,0
2886  012056    001     000
2887
2888  012060    001     010     007           .BYTE   1,8.,7,0,0
2889  012063    000     000
2890
2891  012065    001     011     003           .BYTE   1,9.,3,0,0
2892  012070    000     000
2893
2894  012072    005     011     001           .BYTE   5,9.,1,0,0
2895  012075    000     000
2896
2897  012077    010     011     000           .BYTE   8.,9.,0,0,0
2898  012102    000     000
2899
2900  012104    011     004     000           .BYTE   9.,4,0,0,0
2901  012107    000     000
2902
2903  012111    004     002     000           .BYTE   4,2,0,0,0
2904  012114    000     000
2905
2906                            .EVEN
2907
2908  012116  006000  REPET7: ROR     %0
2909  012120  006000          ROR     %0              ;ROUTINE TO RIGHT
2910  012122  006000          ROR     %0              ;JUSTIFY DATA BY
2911  012124  006000          ROR     %0              ;ROTATING IT 4 PLACE TO
2912  012126  000207          RTS     PC              ;THE RIGHT
2913  012130  017702  167002  CONVER: MOV     @SWR,R2
```

```
2914  012134  042702  000017              BIC     #17,    %2      ;THISROUTINE SAMPLES SWR
2915  012140  052702  104000              BIS     #104000,%2      ;AND SENDS CHANNEL, GAIN
2916  012144  104432                      WTBSY
2917  012146  052777  000027  167406      BIS     #PWFEN+ERREN+MODEN+XRIF,@ICSR
2918  012154  012737  012202  001612      MOV     #CONVED,ICRVT   ;SET UP INTERRUPT SERVICE
2919  012162  010277  166734              MOV     R2,@$BDADR
2920  012166  005237  001602              INC     ADBSY
2921  012172  005037  177776              CLR     PS
2922  012176  000240                      NOP
2923  012200  000776                      BR      .-2
2924
2925  012202  022626              CONVED: POP2SP
2926  012204  104432                      WTBSY
2927  012206  052777  000001  167346      BIS     #XRIF,@ICSR
2928  012214  005077  166702              CLR     @$BDADR
2929  012220  005037  001602              CLR     ADBSY
2930  012224  005737  001604              TST     DAFLG
2931  012230  001406                      BEQ     1$
2932  012232  012746  000340              MOV     #340,-(SP)
2933  012236  012746  012246              MOV     #1$,-(SP)
2934  012242  000137  020202              JMP     SWROU1
2935  012246  005037  001612      1$:     CLR     ICRVT
2936  012252  042777  000026  167302      BIC     #ERREN+PWFEN+MODEN,@ICSR
2937  012260  052777  000026  167274      BIS     #ERREN+PWFEN+MODEN,@ICSR
2938  012266  005037  177776              CLR     PS
2939  012272  000207                      RTS     PC
2940
```

```
2941                                            ;*
2942                                            ;*TEST 7 ROUTINE TO MANUALLY TAKE SAMPLES, COMPARE RESULTS
2943                                            ;*PRINT ERROR IF ANY
2944                                            ;*CALL=JMP TST7 FROM MONITOR
2945                                            ;*
2946
2947  012274  104400                TST7:   TYPE                    ;TYPE HEADER
2948  012276  030251                        MHT8
2949  012300  104426                        ADAR
2950  012302  104431                        QUBR
2951  012304  013737  015032  001122        MOV     ADADR,$BDADR
2952  012312  013737  001512  001534        MOV     FR110,FREQ
2953  012320  104422                        DELAY                   ;ALLOW TTY TO SETTLE  BEFORE CONT.
2954  012322  004737  010236                JSR     PC,ADLOG        ;BASIC LOGIS CHECK
2955  012326  005737  001636        22$:    TST     EXPERT
2956  012332  001002                        BNE     .+6
2957  012334  104400                        TYPE                    ;ASK FOR GAIN
2958  012336  030377                        MGAIN
2959  012340  012737  014572  012362        MOV     #GAIN,10$
2960  012346  012737  000001  012364        MOV     #1,     10$+2
2961  012354  000401                        BR      99$
2962  012356  000763                        BR      22$
2963  012360  104414                99$:    INOCT                   ;GET GAIN
2964  012362  014572                10$:    GAIN
2965  012364  000001                        1
2966
2967  012366  012700  014646                MOV     #GLIST,R0       ;GET LIST OF LEGAL GAINS.
2968  012372  005710                20$:    TST     (0)             ;AT END OF LIST?
2969  012374  001003                        BNE     21$             ;NO-CONTINUE.
2970  012376  104400                        TYPE                    ;GAIN HE TYPED IS KNOWN.
2971  012400  030775                        MNSG                    ;TELL HIM.
2972  012402  000751                        BR      22$             ;REASK QUESTION.
2973  012404  012001                21$:    MOV     (0)+,R1         ;GET GAIN FROM GAIN LIST.
2974  012406  042701  170000                BIC     #170000,R1      ;MASK OUT REAL GAIN BITS.
2975  012412  020137  014572                CMP     R1,GAIN         ;GAIN HE TYPED MATCH ONE IN GAIN LIST?
2976  012416  001365                        BNE     20$             ;NO-CHECK NEXT IN LIST.
2977  012420  014037  014572                MOV     -(0),GAIN       ;YES-REPLACE TYPED GAIN BY REAL GAIN.
2978
2979  012424  005737  001636        3$:     TST     EXPERT
2980  012430  001002                        BNE     .+6
2981  012432  104400                        TYPE                    ;ASK FOR CHANS:
2982  012434  030414                        MCHAN
2983  012436  005037  014576                CLR     CHANS
2984  012442  005037  014600                CLR     CHANF
2985  012446  005037  014602                CLR     CHANSR
2986  012452  005037  014604                CLR     CHANFR
2987  012456  013737  001634  014626        MOV     CORSIZ,RTEMP
2988  012464  162737  032054  014626        SUB     #$ENDAD,RTEMP
2989  012472  005037  014630                CLR     RTEMP1
2990  012476  005237  014630        15$:    INC     RTEMP1
```

```
2991  012502  162737  001000  014626          SUB     #1000,RTEMP
2992  012510  100372                          BPL     15$
2993  012512  005337  014630                  DEC     RTEMP1
2994  012516  013737  014630  014606          MOV     RTEMP1,CHANNO
2995  012524  013746  014630                  MOV     RTEMP1,-(SP)
2996  012530  004737  022342                  GOSUB   ,$SB2D
2997  012534  062716  000010                  ADD     #10,(SP)
2998  012540  005726                          TST     (SP)+
2999  012542  012737  014602  012564          MOV     #CHANSR,11$
3000  012550  012737  000002  012566          MOV     #2,11$+2
3001  012556  000401                          BR      .+4
3002  012560  000721                          BR      3$
3003  012562  104414                          INOCT                   ;GET CHANS
3004  012564  014602                  11$:    CHANSR
3005  012566  000002                          2
3006  012570  005737  014604                  TST     CHANFR
3007
3008  012574  001003                          BNE     .+10
3009  012576  013737  014602  014604          MOV     CHANSR,CHANFR
3010  012604  023737  014602  014604          CMP     CHANSR,CHANFR   ;CHAN. S=F?
3011  012612  003403                          BLE     4$
3012  012614  104400                          TYPE
3013  012616  030463                          MCHER1
3014  012620  000701                          BR      3$
3015  012622  023727  014604  000177  4$:     CMP     CHANFR,#177     ;CHAN WITHIN LEGAL BOUNDS?
3016  012630  003403                          BLE     30$
3017  012632  104400                          TYPE                    ;NO-THEN TELL HIM
3018  012634  030715                          MCHANH
3019  012636  000672                          BR      3$
3020  012640  005737  001636          30$:    TST     EXPERT
3021  012644  001002                          BNE     .+6
3022  012646  104400                          TYPE                    ;ASK FOR EXPECTED AVERAGE.
3023  012650  030744                          MAVEQ
3024  012652  012737  014616  012674          MOV     #AVEXP,12$
3025  012660  012737  000001  012676          MOV     #1,12$+2
3026  012666  000401                          BR      .+4
3027  012670  000763                          BR      30$
3028  012672  104414                          INOCT                   ;GET AVERAGE.
3029  012674  014616                  12$:    AVEXP
3030  012676  000001                          1
3031  012700  032737  170000  014616          BIT     #170000,AVEXP           ;LEGAL AVERAGE?
3032  012706  001403                          BEQ     6$
3033  012710  104400                          TYPE                    ;NO TELL HIM(OR HER) ASK:
3034  012712  031015                          MNTL                    ;QUESTION AGAIN.
3035  012714  000751                          BR      30$
3036  012716  012737  014614  012776  6$:     MOV     #TOLER,13$
3037  012724  012737  000001  013000          MOV     #1,13$+2
3038  012732  032737  004000  014616          BIT     #4000,AVEXP
3039  012740  001406                          BEQ     60$
3040  012742  005737  015034                  TST     BIPOL
3041  012746  001003                          BNE     60$
3042  012750  052737  170000  014616          BIS     #170000,AVEXP
3043  012756  005737  001636          60$:    TST     EXPERT
3044  012762  001002                          BNE     .+6
3045  012764  104400                          TYPE                    ;ASK FOR TOLERANCE.
3046  012766  030441                          MTOL
```

```
3047  012770  000401                        BR      .+4
3048  012772  000771                        BR      60$
3049  012774  104414                        INOCT
3050  012776  014614                13$:    TOLER
3051  013000  000001                        1
3052  013002  005737  001644                TST     LINEPR
3053  013006  001411                        BEQ     14$
3054  013010  013737  001646  001146        MOV     LPCSR,$TPS
3055  013016  013737  001650  001150        MOV     LPDBR,$TPB
3056  013024  104400  026071                TYPE,   MWK
3057  013030  000402                        BR      .+6
3058  013032  104400                14$:    TYPE                    ;TYPE "REPEAT"
3059  013034  027562                        MREP
3060  013036  013737  014602  014600        MOV     CHANSR,CHANF
3061  013044  005337  014600                DEC     CHANF
3062  013050  013737  014600  014576  5$:   MOV     CHANF,CHANS
3063  013056  005237  014576                INC     CHANS
3064  013062  063737  014606  014600        ADD     CHANNO,CHANF
3065  013070  023737  014576  014604        CMP     CHANS,CHANFR
3066  013076  003355                        BGT     14$
3067  013100  023737  014600  014604        CMP     CHANF,CHANFR
3068  013106  003403                        BLE     .+10
3069  013110  013737  014604  014600        MOV     CHANFR,CHANF
3070  013116  004737  013132                GOSUB   ,SAMPR           ;TAKE CONVERSIONS
3071  013122  004737  013546                GOSUB   ,AVERR          ;TAKE AVERAGES
3072  013126  000750                        BR      5$
3073
3074  013130  000000                TST8:   HALT
3075
3076                                        ;*
3077                                        ;*ROUTINE TO TAKE 256 SAMPLES OF ALL CHANNELS SPECIFIED
3078                                        ;*CALL=GOSUB,SAMPR
3079
3080  013132  013737  001514  001534  SAMPR: MOV     FRS0,FREQ       ;SUB PART TO SET UP DELAY BASEDON
3081  013140  013737  014600  014570        MOV     CHANF,CHAN
3082  013146  163737  014576  014570        SUB     CHANS,CHAN      ;NUMBER OF CHANNELLS WERE SAMPLING.
3083  013154  013737  014570  014612        MOV     CHAN,SAMOFF
3084  013162  001416                        BEQ     2$
3085  013164  006337  014612                ASL     SAMOFF
3086  013170  163737  001524  001534  1$:   SUB     FRS,FREQ
3087  013176  005337  014570                DEC     CHAN
3088  013202  001372                        BNE     1$
3089  013204  005737  001534                TST     FREQ
3090  013210  100003                        BPL     2$
3091  013212  012737  000001  001534        MOV     #1,FREQ
3092  013220  012737  177400  014610  2$:   MOV     #-256.,SAMCNT   ;SET SAMPLE COUNT.
3093  013226  012701  032054                MOV     #BUFFER,R1      ;SET FOR STORAGE.
3094  013232  013737  014576  014570  3$:   MOV     CHANS,CHAN      ;GET STARTING CHANNEL
3095  013240  104422                        DELAY
3096  013242  013737  014570  014566  4$:   MOV     CHAN,CHAN1      ;SET TO RIGHT JUSTIFY CHAN.
3097  013250  006337  014566                ASL     CHAN1           ;FIX TO LOAD INTO A/D WORD.
3098  013254  006337  014566                ASL     CHAN1
3099  013260  006337  014566                ASL     CHAN1
3100  013264  006337  014566                ASL     CHAN1
3101  013270  004737  013326                GOSUB   ,CONVT          ;TAKE CONVERSION.
3102  013274  010021                        MOV     R0,(1)+         ;STORE RESULT
```

```
3103  013276  005237  014570                    INC     CHAN            ;READY FOR NEXT CHAN.
3104  013302  023737  014570  014600            CMP     CHAN,CHANF      ;DONE ALL CHANNELLS
3105  013310  003754                            BLE     4$
3106  013312  005237  014610                    INC     SAMCNT          ;DONE 256 SAMPLES?
3107  013316  001345                            BNE     3$
3108  013320  012701  032054                    MOV     #BUFFER,R1
3109  013324  000207                            RETURN
3110
3111                                            ;*ROUTINE TO FORM A A/D WORD FOR CONVERSIONS, START A CONVERSION,
3112                                            ;*WAIT FOR DONE, AND ENABLE READING OF THE A/D DBR.
3113
3114                                            ;*CALL=GOSUB,CONVT,RESULT FOUND IN R0,RIGHT JUSTIFIED
3115  013326  013737  014572  014574  CONVT:    MOV     GAIN,ADWD               ;SET GAIN IN WORD
3116  013334  042737  007777  014574            BIC     #7777,ADWD
3117  013342  053737  014566  014574            BIS     CHAN1,ADWD              ;SET CHAN.
3118  013350  052737  104000  014574            BIS     #104000,ADWD            ;SET READ BIT, CONVERT BIT.
3119  013356  052777  000027  166176            BIS     #ERREN+MODEN+PWFEN+XRIF,@ICSR
3120  013364  012737  013416  001612            MOV     #1$,ICRVT
3121  013372  005237  001602                    INC     ADBSY
3122  013376  005037  177776                    CLR     PS
3123  013402  104432                            WTBSY
3124  013404  013777  014574  165510            MOV     ADWD,@$BDADR            ;START CONVERSION.
3125  013412  000240                            NOP
3126  013414  000776                            BR      .-2
3127  013416  104432                    1$:     WTBSY
3128  013420  052777  000001  166134            BIS     #XRIF,@ICSR
3129  013426  005077  165470                    CLR     @$BDADR         ;ENABLE READING OF DBR.
3130  013432  104433                            ICRDLY
3131  013434  017700  165462                    MOV     @$BDADR,R0      ;READ AID RESULTS PUT IN R0
3132  013440  022626                            POP2SP
3133  013442  005037  001602                    CLR     ADBSY
3134  013446  005737  001604                    TST     DAFLG
3135  013452  001406                            BEQ     3$
3136  013454  012746  000340                    MOV     #340,-(SP)
3137  013460  012746  013470                    MOV     #3$,-(SP)
3138  013464  000137  020202                    JMP     SWROU1
3139  013470  005037  001612            3$:     CLR     ICRVT
3140  013474  042777  000026  166060            BIC     #ERREN+PWFEN+MODEN,@ICSR
3141  013502  052777  000026  166052            BIS     #ERREN+PWFEN+MODEN,@ICSR
3142  013510  005037  177776                    CLR     PS
3143  013514  005737  015034                    TST     BIPOL
3144  013520  001405                            BEQ     2$
3145  013522  006000                            ROR     R0
3146  013524  006000                            ROR     R0
3147  013526  006000                            ROR     R0
3148  013530  006000                            ROR     R0
3149  013532  000207                            RETURN
3150  013534  006200                    2$:     ASR     R0              ;RIGHT JUSTIFY, REMEMBERING SIGN.
3151  013536  006200                            ASR     R0
3152  013540  006200                            ASR     R0
3153  013542  006200                            ASR     R0
3154  013544  000207                            RETURN                  ;RETURN
3155                                            ;*
3156                                            ;*AVERAGING ROUTINE USED BE TEST 7
3157                                            ;*AT THIS POINT IN TIME ALL SAMPLES FOR ALL CHANNELS HAVE
3158                                            ;*BEEN TAKEN AND STORED IN "BUFFER" IN A
```

```
3159                                                ;*SEQUENTIAL INTERLEAVED BUFFER FORM" GIVEN BY THE FOLLOWING FORMULA:
3160                                                ;*"L=2*N+R1" WHERE L EQUALS THE LOCATION OF A SAMPLE,
3161                                                ;*N EQUALS THE NUMBER OF CHANNELS SAMPLES WERE TAKEN ON, AND R1
3162                                                ;*IS THE BUFFER POINTER SET TO "BUFFER" INITIALLY.
3163                                                ;*CALL=GOSUB,AVERR
3164
3165  013546  013737  014576  014570  AVERR:  MOV     CHANS,CHAN      ;SET TO FIRST CHAN.
3166  013554  005037  014566          CLR     CHAN1           ;1ST CHAN OFFSET
3167  013560  012737  000400  014610  AVERRL: MOV     #256.,SAMCNT
3168  013566  013701  014566          MOV     CHAN1,R1        ;PUT CHAN OFFSET IN R1.
3169  013572  062701  032054          ADD     #BUFFER,R1      ;ADD BUFFER POINTER TO R1.
3170  013576  005037  014620          CLR     AVTKN           ;SET INITIAL CONDITIONS FOR THIS CHAN.
3171  013602  005037  014626          CLR     RTEMP
3172  013606  011137  014622          MOV     (1),RLOW
3173  013612  011137  014624          MOV     (1),RHIGH
3174  013616  023711  014624  2$:     CMP     RHIGH,(1)       ;FIND REAL HIGH VALUE.
3175  013622  003002                  BGT     3$
3176  013624  011137  014624          MOV     (1),RHIGH
3177  013630  021137  014622  3$:     CMP     (1),RLOW        ;FIND REAL LOW VALUE.
3178  013634  003002                  BGT     4$
3179  013636  011137  014622          MOV     (1),RLOW
3180  013642  012137  014630  4$:     MOV     (1)+,RTEMP1     ;GET CURRENT SAMPLE.
3181  013646  005737  015034          TST     BIPOL
3182  013652  001003                  BNE     .+10
3183  013654  062737  004000  014630  ADD     #4000,RTEMP1    ;ADD CONSTANT TO SAMPLE.
3184  013662  063737  014630  014620  ADD     RTEMP1,AVTKN    ;"BOOT" ADD ALL SAMPLES.
3185  013670  005537  014626          ADC     RTEMP
3186  013674  063701  014612          ADD     SAMOFF,R1       ;UPDATE TO LOOK AT NEXT CHAN SAMPLE
3187  013700  005337  014610          DEC     SAMCNT          ;DONE ALL SAMPLES PER THIS CHAN?
3188  013704  001344                  BNE     2$              ;NO-DO NEXT SAMPLE.
3189  013706  013737  014616  014630  MOV     AVEXP,RTEMP1    ;YES! SEE IF RHIGH OK:
3190  013714  063737  014614  014630  ADD     TOLER,RTEMP1
3191  013722  023737  014630  014624  CMP     RTEMP1,RHIGH
3192  013730  002427                  BLT     ERAV1           ;NO-THEN REPORT ERROR.
3193  013732  163737  014614  014630  SUB     TOLER,RTEMP1    ;YES-OK-CHECK LOWEST READING.
3194  013740  163737  014614  014630  SUB     TOLER,RTEMP1
3195  013746  023737  014622  014630  CMP     RLOW,RTEMP1
3196  013754  002415                  BLT     ERAV1           ;NO-REPORT ERROR.
3197  013756  005737  014614          TST     TOLER           ;DOES OPERATOR WISH "FORCED" TYPEOUT?
3198  013762  001415                  BEQ     ERAV2           ;IF SO-DO IT
3199  013764  062737  000002  014566  AVERRN: ADD     #2,CHAN1        ;SET TO DO NEXT CHAN-BUT
3200  013772  005237  014570          INC     CHAN            ;IF DONE ALL CHANS-EXIT-
3201  013776  023737  014570  014600  CMP     CHAN,CHANF      ;OTHERWIZE LOOP.
3202  014004  003665                  BLE     AVERRL
3203  014006  000207                  RETURN
3204
3205                                                ;*ERROR REPORTER
3206
3207  014010  104400          ERAV1:  TYPE                    ;TYPE "REPEATIBILITY ERROR"
3208  014012  031110                  MREPER
3209  014014  000402                  BR      .+6
3210  014016  104400          ERAV2:  TYPE                    ;TYPE "REPEATIBILITY FORCED TYPEOUT"
3211  014020  031052                  MREPFT
3212  014022  012702  000007  1$:     MOV     #7,R2           ;FIND AVERAGE.
3213  014026  006237  014626  10$:    ASR     RTEMP
3214  014032  006037  014620          ROR     AVTKN           ;AVERAGE=TOTAL OF SAMPLES-
```

```
 3215  014036  005302                        DEC     R2              ;DIVIDED BY 256.
 3216  014040  001372                        BNE     10$
 3217  014042  006237  014626                ASR     RTEMP
 3218  014046  006037  014620                ROR     AVTKN
 3219  014052  005537  014620                ADC     AVTKN
 3220  014056  005737  015034                TST     BIPOL
 3221  014062  001003                        BNE     .+10
```

```
3222  014064  162737  004000  014620          SUB     #4000,AVTKN     ;SUBTRACT CONSTANT.
3223  014072  004737  023050                  JSR     PC,ERRDLY
3224  014076  104400                          TYPE                    ;TYPE HEADER 1
3225  014100  031136                          MREPT1
3226  014102                                  TYPOCT  CHAN            ;TYPE CHAN.
3227  014102  013746  014570                  MOV     CHAN,-(SP)      ;;SAVE CHAN FOR TYPEOUT
3228  014106  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3229  014110  104400                          TYPE
3230  014112  031050                          M2SP
3231  014114  013702  014572                  MOV     GAIN,R2         ;TYPE GAIN.
3232  014120  042702  170000                  BIC     #170000,R2
3233  014124                                  TYPOCT  R2
3234  014124  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
3235  014126  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3236  014130  004737  023050                  JSR     PC,ERRDLY
3237  014134  104400  031154                  TYPE,   MREPT4
3238  014140  013702  014622                  MOV     RLOW,R2
3239  014144  042702  170000                  BIC     #170000,R2
3240  014150                                  TYPOCT  R2              ;TYPE LOWEST SAMPLE TAKEN.
3241  014150  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
3242  014152  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3243  014154  104400                          TYPE
3244  014156  031050                          M2SP
```

```
3245  014160  013702  014620                  MOV     AVTKN,R2
3246  014164  042702  170000                  BIC     #170000,R2
3247  014170                                  TYPOCT  R2              ;TYPE AVERAGE OF SAMPLES TAKEN.
3248  014170  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
3249  014172  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3250  014174  104400                          TYPE
3251  014176  031050                          M2SP
3252  014200  013702  014624                  MOV     RHIGH,R2
3253  014204  042702  170000                  BIC     #170000,R2
3254  014210                                  TYPOCT  R2              ;TYPE HIGHEST SAMPLE TAKEN.
3255  014210  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
3256  014212  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3257  014214  004737  023050                  JSR     PC,ERRDLY
3258  014220  005737  000164                  TST     REMFF           ;RUNNING REMOTE?
3259  014224  001403                          BEQ     .+10            ;NO, TYPE LONG HEADER
3260  014226  104400  031261                  TYPE,   MREPT3          ;YES, TYPE ABBREVIATED HEADER
3261  014232  000402                          BR      .+6
3262  014234  104400                          TYPE                    ;TYPE SECOND HEADER.
3263  014236  031173                          MREPT2
3264  014240  004737  023050                  JSR     PC,ERRDLY
3265  014244  104400  001161                  TYPE,   $CRLF
3266  014250  013737  014620  014626          MOV     AVTKN,RTEMP     ;GET AV.
3267  014256  005737  000164                  TST     REMFF           ;RUNNING REMOTE?
3268  014262  001404                          BEQ     98$             ;NO, THEN SET UP FOR LONG TYPEOUT
3269  014264  162737  000002  014626          SUB     #2,RTEMP        ;YES, MAKE -2 TO GET LOW POINT
3270  014272  000403                          BR      97$
3271  014274  162737  000005  014626  98$:    SUB     #5,RTEMP        ;MAKE -5 POINT TO GET LOW POINT.
3272  014302  012737  000400  014610  97$:    MOV     #256.,SAMCNT    ;SET # OF SAMPLES.
3273  014310  013701  014566                  MOV     CHAN1,R1
3274  014314  062701  032054                  ADD     #BUFFER,R1
3275  014320  005002                          CLR     R2
3276  014322  022137  014626          2$:     CMP     (1)+,RTEMP
3277  014326  002001                          BGE     3$
3278  014330  005202                          INC     R2
3279  014332  063701  014612          3$:     ADD     SAMOFF,R1
3280  014336  005337  014610                  DEC     SAMCNT
3281  014342  001367                          BNE     2$
3282  014344  004737  014530                  GOSUB   ,AVTYP          ;TYPE RLOW # OF SAMPLES.
3283  014350  005737  000164                  TST     REMFF           ;RUNNING REMOTE?
3284  014354  001403                          BEQ     99$             ;NO, SETUP FOR LONG TYPE
3285  014356  012703  177773                  MOV     #-5,R3          ;SETUP FOR ABBREV TYPEOUT
3286  014362  000402                          BR      4$
3287  014364  012703  177765          99$:    MOV     #-11.,R3                ;SET TO DO 11 TIMES.
3288  014370  012737  000400  014610  4$:     MOV     #256.,SAMCNT    ;SET SAMPLE COUNT AT 256.
3289  014376  005002                          CLR     R2
3290  014400  013701  014566                  MOV     CHAN1,R1        ;GET BUFFER POINTER.
3291  014404  062701  032054                  ADD     #BUFFER,R1
3292  014410  022137  014626          5$:     CMP     (1)+,RTEMP      ;SAMPLE=COUNT POINTER?
3293  014414  001001                          BNE     6$
3294  014416  005202                          INC     R2
3295  014420  063701  014612          6$:     ADD     SAMOFF,R1
3296  014424  005337  014610                  DEC     SAMCNT          ;CHECKED ALL SAMPLES?
3297  014430  001367                          BNE     5$
3298  014432  004737  014530                  GOSUB   ,AVTYP          ;TYPE # OF SAMPLES AT THIS POINT.
3299  014436  005237  014626                  INC     RTEMP           ;MOVE TO NEXT POINT.
3300  014442  005203                          INC     R3              ;DONE ALL POINTS?
```

```
3301  014444  001351                        BNE     4$              ;NO-THEN LOOP
3302  014446  005002                        CLR     R2              ;FIND ALL OVERSCALE POINTS.
3303  014450  012737  000400  014610        MOV     #256.,SAMCNT
3304  014456  013701  014566                MOV     CHAN1,R1
3305  014462  062701  032054                ADD     #BUFFER,R1
3306  014466  022137  014626        7$:     CMP     (1)+,RTEMP
3307  014472  003401                        BLE     8$
3308  014474  005202                        INC     R2
3309  014476  063701  014612        8$:     ADD     SAMOFF,R1
3310  014502  005337  014610                DEC     SAMCNT
3311  014506  001367                        BNE     7$
3312  014510  004737  014530                GOSUB   ,AVTYP
3313  014514  004737  023050                JSR     PC,ERRDLY
3314  014520  104400                        TYPE
3315  014522  001161                        $CRLF
3316  014524  000137  013764                JMP     AVERRN
3317  014530  010246        AVTYP:  MOV     R2,-(SP)        ;PUT # ON STACK
3318  014532  004737  022342                GOSUB   ,$$B2D          ;GOTO OCTAL-BCD ROUTINE.
3319  014536  062716  000007                ADD     #7,(SP)         ;GET RID OF 1ST.2 DIGITS
3320  014542  012637  014550                MOV     (SP)+,1$                ;TYPE STRING.
3321  014546  104400                        TYPE
3322  014550  000000        1$:     0
3323  014552  104400                        TYPE                    ;TYPE 2 SPACES.
3324  014554  031050                        M2SP
3325  014556  000207                        RETURN
3326
3327                                        ;*
3328                                        ;*POINTERS USED BY TEST 7  REPEATIBILITY TEST
3329                                        ;*
3330
3331  014560  000000        CHAN7:0
3332  014562  000000        NAD7:0
3333  014564  000000        NA17:0
3334  014566  000000        CHAN1:  0               ;LEFT JUSTIFIED CURRENT CHANNELL
3335  014570  000000        CHAN:   0               ;CURRENT CHANNELL
3336  014572  000000        GAIN:   0               ;CURRENT GAIN
3337  014574  000000        ADWD:   0               ;WORD SENT TO A/D
3338  014576  000000        CHANS:  0               ;STARTING CHANNEL
3339  014600  000000        CHANF:  0               ;LAST CHANNEL
3340  014602  000000        CHANSR: 0
3341  014604  000000        CHANFR: 0
3342  014606  000000        CHANNO: 0
3343  014610  000000        SAMCNT: 0               ;SAMPLE COUNT
3344  014612  000000        SAMOFF: 0               ;SAMPLE OFFSET
3345  014614  000000        TOLER:  0               ;TOLERANCE BEFORE ERROR IS REPORTED
3346  014616  000000        AVEXP:  0               ;EXPECTED AVERAGE
3347  014620  000000        AVTKN:  0               ;AVERAGE OF SAMPLES TAKEN
3348  014622  000000        RLOW:   0               ;LOWEST SAMPLE TAKEN
3349  014624  000000        RHIGH:  0               ;HIGHEST SAMPLE TAKEN
3350  014626  000000        RTEMP:  0
3351  014630  000000        RTEMP1: 0
3352  014632  000000        REPMAN: 0               ;SECTION RESERVED FOR
3353  014634  000000                0               ;MANUAL REPEATIBILITY
3354  014636  000000                0               ;WHERE OPERATOR ENTERS
3355  014640  000000                0               ;IN INFORMATION
3356  014642     000     000        .BYTE   0,0
```

```
3357  014644  000000                        0
3358
3359                                        ;*
3360                                        ;* GAIN LIST
3361                                        ;*
3362
3363  014646  071000              GLIST:    071000              ;GAIN OF 1000.
3364  014650  060200                        060200              ;GAIN OF 200.
3365  014652  050100                        050100              ;GAIN OF 100.
3366  014654  040050                        040050              ;GAIN OF 50.
3367  014656  030020                        030020              ;GAIN OF 20.
3368  014660  020010                        020010              ;GAIN OF 10.
3369  014662  010002                        010002              ;GAIN OF 2.
3370  014664  000001                        000001              ;GAIN OF 1.
3371  014666  000000                        000000              ;ILLEGAL GAIN.
3372
3373                                        ;*INPUT MODULE ADDRESSES
3374  014670  000000              INADR:    0
3375  014672  000000                        .WORD   0
3376  014674  000000                        .WORD   0
3377  014676  000000                        .WORD   0
3378  014700  000000                        .WORD   0
3379  014702  000000                        .WORD   0
3380  014704  000000                        .WORD   0
3381  014706  000000                        .WORD   0
3382  014710  000000                        .WORD   0
3383  014712  000000                        .WORD   0
3384  014714  000000                        .WORD   0
3385  014716  000000                        .WORD   0
3386  014720  000000                        .WORD   0
3387  014722  000000                        .WORD   0
3388  014724  000000                        .WORD   0
3389  014726  000000                        .WORD   0
3390                                        ;*OUTPUT MODULE ADDRESSES
3391  014730  000000              OUTADR:   0
3392  014732  000000                        .WORD   0
3393  014734  000000                        .WORD   0
3394  014736  000000                        .WORD   0
3395  014740  000000                        .WORD   0
3396  014742  000000                        .WORD   0
3397  014744  000000                        .WORD   0
3398  014746  000000                        .WORD   0
3399  014750  000000                        .WORD   0
3400  014752  000000                        .WORD   0
3401  014754  000000                        .WORD   0
3402  014756  000000                        .WORD   0
3403  014760  000000                        .WORD   0
3404  014762  000000                        .WORD   0
3405  014764  000000                        .WORD   0
3406  014766  000000                        .WORD   0
3407                                        ;*COUNTER MODULE ADDRESSES
3408  014770  000000              CNTADR:   0
3409  014772  000000                        .WORD   0
3410  014774  000000                        .WORD   0
3411  014776  000000                        .WORD   0
3412  015000  000000                        .WORD   0
```

```
3413  015002  000000                          .WORD   0
3414  015004  000000                          .WORD   0
3415  015006  000000                          .WORD   0
3416  015010  000000                          .WORD   0
3417  015012  000000                          .WORD   0
3418  015014  000000                          .WORD   0
3419  015016  000000                          .WORD   0
3420  015020  000000                          .WORD   0
3421  015022  000000                          .WORD   0
3422  015024  000000                          .WORD   0
3423  015026  000000                          .WORD   0
3424                                          ;* DAC[illegible]ULE ADDRESS
3425  015030  000000                  DACADR: 0
3426                                          ;*AD05 MODULE ADDRESS
3427  015032  000000                  ADADR:  0
3428  015034  000000                  BIPOL:  0                       ;0=BIPOLAR-1=UNI
3429                                          ;*TEMP STORAGE OF INPUT MODULE DATA
3430  015036  000000                  OUTS:   0
3431  015040  000000                          .WORD   0
3432  015042  000000                          .WORD   0
3433  015044  000000                          .WORD   0
3434  015046  000000                          .WORD   0
3435  015050  000000                          .WORD   0
3436  015052  000000                          .WORD   0
3437  015054  000000                          .WORD   0
3438  015056  000000                          .WORD   0
3439  015060  000000                          .WORD   0
3440  015062  000000                          .WORD   0
3441  015064  000000                          .WORD   0
3442  015066  000000                          .WORD   0
3443  015070  000000                          .WORD   0
3444  015072  000000                          .WORD   0
3445  015074  000000                          .WORD   0
3446  015076  000000                  OUTSE:  0
3447
3448                                          ;*
3449                                          ;*ROUTINE TO INPUT INPUT MODULE ADDRESS.
3450                                          ;*
3451
3452  015100  005737  001636          RINA:   TST     EXPERT
3453  015104  001002                          BNE     .+6
3454  015106  104400                          TYPE                    ;ASK OPERATOR "INPUT MODULE ADDR(S)?
3455  015110  025467                          MIPA
3456  015112  012737  014670  015134          MOV     #INADR, RINA1
3457  015120  012737  000020  015136          MOV     #16.,   RINA1+2
3458  015126  000401                          BR      .+4
3459  015130  000763                          BR      RINA
3460  015132  104414                          INOCT                   ;CALL TO INPUT OCTAL ROUTINE
3461  015134  014670                  RINA1:  INADR                   ;ADDR TO STORE INPUT ADDRESS AT (VARIES).
3462  015136  000020                          16.                     ;NUMBER OF ADDRESS ALLOWED.
3463  015140  005737  017140                  TST     CHAR
3464  015144  001411                          BEQ     2$
3465  015146  012700  014670                  MOV     #INADR, R0      ;NOW FORM REAL ICR ADDRESS.
3466  015152  053710  001560          1$:     BIS     ICSMOD, (0)     ;NORMAL-ICR MOD=171000 UNLESS PATCHED
3467  015156  004737  020766                  JSR     PC,CKADR
3468  015162  020037  015134                  CMP     R0,RINA1        ;DONE ALL ADDRS?
```

```
3469  015166  101771                    BLOS    1$              ;IF NOT CONTINUE.
3470  015170  000002            2$:     EXIT                    ;IF SO-EXIT
3471
3472                                    ;*
3473                                    ;*ROUTINE TO INPUT OUTPUT MODULE ADDRESSESS
3474                                    ;*
3475
3476  015172  005737  001636    ROUTA:  TST     EXPERT
3477  015176  001002                    BNE     .+6
3478  015200  104400                    TYPE                    ;ASK OPERATOR "OUTPUT MODULE ADDRESSESS?
3479  015202  025504                    MOPA
3480  015204  012737  014730  015226    MOV     #OUTADR, ROUTA1
3481  015212  012737  000020  015230    MOV     #16.,   ROUTA1+2
3482  015220  000401                    BR      .+4
3483  015222  000763                    BR      ROUTA
3484  015224  104414                    INOCT                   ;CALL TO INPUT OCTAL ROUTINE
3485  015226  014730    ROUTA1: OUTADR                          ;ADDR TO STORE OUTPUT ADDRS.
3486  015230  000020                    16.                     ;
3487  015232  005737  017140            TST     CHAR
3488  015236  001411                    BEQ     2$
3489  015240  012700  014730            MOV     #OUTADR, R0     ;
3490  015244  053710  001560    1$:     BIS     ICSMOD, (0)
3491  015250  004737  020766            JSR     PC,CKADR
3492  015254  020037  015226            CMP     R0,ROUTA1
3493  015260  101771                    BLOS    1$
3494  015262  000002            2$:     EXIT
3495
3496                                    ;*
3497                                    ;*ROUTINE TO INPUT COUNTER MODULE ADDRS
3498                                    ;*
3499
3500  015264  005737  001636    RCNTA:  TST     EXPERT
3501  015270  001002                    BNE     .+6
3502  015272  104400                    TYPE                    ;ASK OPERATOR FOR COUNTER MODULE
3503  015274  025624                    MCNT                    ;ADDRS.
3504  015276  012737  014770  015320    MOV     #CNTADR, RCNTA1
3505  015304  012737  000020  015322    MOV     #16.,   RCNTA1+2
3506  015312  000401                    BR      .+4
3507  015314  000763                    BR      RCNTA
3508  015316  104414                    INOCT                   ;GET THEM
3509  015320  014770    RCNTA1: CNTADR
3510  015322  000020                    16.
3511  015324  012700  014770            MOV     #CNTADR, R0
3512  015330  005737  017140            TST     CHAR
3513  015334  001407                    BEQ     2$
3514  015336  053710  001560    1$:     BIS     ICSMOD, (0)
3515  015342  004737  020766            JSR     PC,CKADR
3516  015346  020037  015320            CMP     R0,RCNTA1
3517  015352  003771                    BLE     1$
3518  015354  000002            2$:     EXIT
3519
3520                                    ;*
3521                                    ;*ROUTINE TO INPUT DAC ADDR
3522                                    ;*
3523  015356  005737  001636    RDACA:  TST     EXPERT
3524  015362  001002                    BNE     .+6
```

```
3525  015364  104400                          TYPE
3526  015366  026106                          MDAC
3527  015370  012737  015030  015412          MOV     #DACADR, RDAC1
3528  015376  012737  000001  015414          MOV     #1,      RDAC1+2
3529  015404  000401                          BR      .+4
3530  015406  000763                          BR      RDACA
3531  015410  104414                          INOCT
3532  015412  015030                  RDAC1:  DACADR
3533  015414  000001                          1
3534  015416  005737  017140                  TST     CHAR
3535  015422  001407                          BEQ     1$
3536  015424  053737  001560  015030          BIS     ICSMOD, DACADR
3537  015432  012700  015030                  MOV     #DACADR,R0
3538  015436  004737  020766                  JSR     PC,CKADR
3539  015442  000002                  1$:     EXIT
3540                                          ;*
3541                                          ;*ROUTINE TO INPUT ADDS ADDR
3542                                          ;*
3543  015444  005737  001636          RADA:   TST     EXPERT
3544  015450  001002                          BNE     .+6
3545  015452  104400                          TYPE
3546  015454  026122                          MADU
3547  015456  012737  015032  015500          MOV     #ADADR, RADA1
3548  015464  012737  000001  015502          MOV     #1,      RADA1+2
3549  015472  000401                          BR      .+4
3550  015474  000763                          BR      RADA
3551  015476  104414                          INOCT
3552  015500  015032                  RADA1:  ADADR
3553  015502  000001                          1
3554  015504  005737  017140                  TST     CHAR
3555  015510  001407                          BEQ     1$
3556  015512  053737  001560  015032          BIS     ICSMOD, ADADR
3557  015520  012700  015032                  MOV     #ADADR,R0
3558  015524  004737  020766                  JSR     PC,CKADR
3559  015530  000002                  1$:     EXIT
3560
3561                                          ;*
3562                                          ;*INPUT OCTAL ROUTINE
3563                                          ;*NMW TO CALL=WHERE TO STORE NUMBERS
3564                                          ;*NMW = NUMBER OF NUMBERS TO ACCEPT.
3565                                          ;*
3566
3567  015532  104400                  INOCTR: TYPE                    ;TYPE A "?".
3568  015534  026403                          MQ
3569  015536  005737  001576                  TST     NOTYET          ;DO WE KNOW IF FILE IS PRESENT
3570  015542  001407                          BEQ     73$             ;NO, DON'T TRY TO ADDRESS ICR
3571  015544  052777  000001  164010          BIS     #XRIF,@ICSR
3572  015552  017701  164110                  MOV     @ICSLMT,R1
3573  015556  005077  164000                  CLR     @ICSR
3574  015562  017601  000000          73$:    MOV     @(6),   R1      ;PICK UP STORAGE ADDRESS.
3575  015566  062716  000002                  ADD     #2,     (6)     ;PICK UP # OF WORDS THAT-
3576  015572  017637  000000  017152          MOV     @(6),   NINC1   ;IS THE MAX # TO BE INPUTED.
3577  015600  162716  000002                  SUB     #2,     (6)     ;CLEAR COLON TYPED FLAG.
3578  015604  005037  001614                  CLR     INCFLG          ;CLEAR ANY CHAR. TYPED FLAG.
3579  015610  005037  014626                  CLR     RTEMP           ;POINT T6 TEMP STORAGE AREA.
3580  015614  012700  015036                  MOV     #OUTS,R0        ;CLEAR THE AREA.
```

```
3581  015620  005020                    20$:    CLR     (0)+
3582  015622  020027  015076                    CMP     R0,#OUTSE
3583  015626  003774                            BLE     20$
3584  015630  012700  015036                    MOV     #OUTS,R0        ;POINT TO AREA AGAIN.
3585  015634  005037  017150                    CLR     NINC            ;CLEAR CHAR. COUNT
3586  015640  005037  017142            1$:     CLR     NIN             ;CLEAR NUMBER TO BE FORMED
3587  015644  005037  017146                    CLR     CHARC           ;INPUTED CHAR. COUNT
3588  015650  005737  000164            2$:     TST     REMFF           ;RUNNING REMOTE?
3589  015654  001477                            BEQ     99$             ;NO CONT
3590  015656  032777  002000  163676            BIT     #PWRFL,@ICSR    ;PWR FAIL?
3591  015664  001402                            BEQ     .+6             ;NO
3592  015666  000137  025120                    JMP     RSTRT           ;SERVICE POWER FAIL
3593  015672  032777  000200  163662            BIT     #MODINT,@ICSR   ;MOD INTR?
3594  015700  001763                            BEQ     2$              ;NO, KEEP CHECKING
3595  015702  005237  001700                    INC     MODFF           ;INDICATE MOD INTR
3596  015706  032777  010000  163650            BIT     #DA,@ICAR       ;DA SET?
3597  015714  001013                            BNE     66$             ;YES, SERVICE IT
3598  015716  005737  001700                    TST     MODFF           ;WAS MOD INTR POSTED?
3599  015722  001752                            BEQ     2$              ;NO, KEEP CHECKING
3600  015724  052777  000001  163630            BIS     #XRIF,@ICSR     ;YES, CLEAR IT OUT
3601  015732  005777  163730                    TST     @ICSLMT         ;
3602  015736  005037  001700                    CLR     MODFF           ;
3603  015742  000742                            BR      2$              ;KEEP CHECKING
3604  015744  052777  001041  163610    66$:    BIS     #TTYEN+TBMTEN+XRIF,@ICSR        ;READ IN CHAR
3605  015752  017737  163710  017140            MOV     @ICSLMT,CHAR                    ;FROM ICR REMOTE TTY
3606  015760  042737  177600  017140            BIC     #177600,CHAR
3607  015766  023727  017140  000177            CMP     CHAR,#177
3608  015774  001002                            BNE     .+6
3609  015776  000137  017100                    JMP     10$
3610  016002  005777  163554                    TST     @ICSR
3611  016006  100775                            BMI     .-4
3612  016010  032777  002000  163544    67$:    BIT     #PWRFL,@ICSR                    ;PWR FAIL?
3613  016016  001402                            BEQ     68$                             ;NO
3614  016020  000137  025120                    JMP     RSTRT                           ;SERVICE IT
3615  016024  032777  100000  163532    68$:    BIT     #XTBMT,@ICAR                    ;ECHO CHARACTER.
3616  016032  001766                            BEQ     67$
3617  016034  013777  017140  163624            MOV     CHAR,@ICSLMT
3618  016042  042777  001040  163512            BIC     #TTYEN+TBMTEN,@ICSR
3619  016050  000137  016112                    JMP     98$
3620  016054  105777  163062            99$:    TSTB    @$TKS           ;KEY TYPED?
3621  016060  100273                            BPL     2$              ;NO - THEN WAIT.
3622  016062  117737  163056  017140            MOVB    @$TKB,  CHAR    ;YES - READ CHAR.
3623  016070  042737  177600  017140            BIC     #177600,  CHAR  ;STRIP CHAR PARITY BIT - IF ANY.
3624  016076  023727  017140  000177            CMP     CHAR,#177       ;WAS IT A RUBOUT?
3625  016104  001002                            BNE     .+6             ;NO - CONTINUE
3626  016106  000137  017100                    JMP     10$             ;YES - TYPE "?" - REINTIALIZE.
3627  016112  105777  163030            98$:    TSTB    @$TPS           ;PRINTER BUSY?
3628  016116  100375                            BPL     .-4             ;YES - THEN WAIT TILL NOT
3629  016120  013777  017140  163022            MOV     CHAR,   @$TPB   ;NO - ECHO CHAR.
3630  016126  023727  017140  000015            CMP     CHAR,   #15     ;CHAR. = <CR>?
3631  016134  001570                            BEQ     3$              ;YES - THEN TERMINATE INPUT.
3632  016136  005737  001576                    TST     NOTYET          ;READY TO ACCECT CNTRL R
3633  016142  001421                            BEQ     97$             ;NO THEN 97$
3634  016144  023727  017140  000022            CMP     CHAR,#22                ;↑R - REMOTE
3635  016152  001015                            BNE     97$
3636  016154  005737  000164                    TST     REMFF
```

```
3637  016160  001012                          BNE     97$
3638  016162  012737  000001  000164          MOV     #1,REMFF
3639  016170  012737  001670  001136          MOV     #REMSWR,SWR
3640  016176  104400  031640                  TYPE,   REMOTE
3641  016202  000137  017130                  JMP     RASK
3642
3643  016206  023727  017140  000023  97$:    CMP     CHAR,#23        ;↑S - SWITCH REG.
3644  016214  001002                          BNE     .+6
3645  016216  000137  020166                  JMP     SWROUT
3646  016222  023727  017140  000030          CMP     CHAR,#30
3647  016230  001002                          BNE     .+6
3648  016232  000137  017226                  JMP     XFRCTL
3649  016236  023727  017140  000003          CMP     CHAR,#3         ;WAS IT A ↑C?
3650  016244  001002                          BNE     .+6
3651  016246  000177  163412                  JMP     @CTLLOC         ;IF YES GOTO MONITR.
3652  016252  005737  000164                  TST     REMFF           ;RUNNING REMOTE
3653  016256  001006                          BNE     88$             ;IF SO INHIBIT LINEPRINTER
3654  016260  023727  017140  000014          CMP     CHAR,#14        ;CHAR = "↑L"?
3655  016266  001002                          BNE     .+6
3656  016270  000137  017204                  JMP     LPSET           ;YES - THEN SET LINEPRINTER MODE.
3657  016274  122737  000012  017140  88$:    CMPB    #12,CHAR        ;CHAR = "↑J"?
3658  016302  001002                          BNE     .+6
3659  016304  000137  020226                  JMP     CONNTR          ;YES - THEN SET JOINED MODE.
3660  016310  122737  000004  017140          CMPB    #4,CHAR         ;CHAR = "↑D"?
3661  016316  001002                          BNE     .+6
3662  016320  000137  020352                  JMP     RDELA3          ;YES - GET SECOUNDARY DELAY.
3663  016324  123727  017140  000005          CMPB    CHAR,#5         ;CHAR = "↑E"?
3664  016332  001002                          BNE     .+6
3665  016334  000137  017154                  JMP     EXSET           ;YES - SET EXPERT MODE.
3666  016340  123727  017140  000016          CMPB    CHAR,#16        ;CHAR = "↑N"?
3667  016346  001002                          BNE     .+6
3668  016350  000137  017170                  JMP     NOSET           ;YES - SET NOVICE MODE
3669  016354  123727  017140  000020  69$:    CMPB    CHAR,#20        ;↑P--CONTROL P
3670  016362  001002                          BNE     .+6
3671  016364  000137  004226                  JMP     TYPECT          ;YES, GO TYPE ERROR COUNT
3672  016370  023727  017140  000072          CMP     CHAR,   #':     ;CHAR = COLON?
3673  016376  001535                          BEQ     4$              ;YES SET FOR "THROUGH" ENTRY.
3674  016400  023727  017140  000054          CMP     CHAR,   #',     ;CHAR = A COMMA?
3675  016406  001002                          BNE     .+6
3676  016410  000137  017030                  JMP     7$
3677  016414  023727  017140  000060          CMP     CHAR,#60        ;CHAR TYPED 7 ASCIZO?
3678  016422  002002                          BGE     .+6
3679  016424  000137  017110                  JMP     11$             ;NO - REPORT ERROR.
3680
3681  016430  023727  017140  000067          CMP     CHAR,#67        ;CHAR TYPED <ASCIZ 7?
3682  016436  003402                          BLE     .+6
3683  016440  000137  017110                  JMP     11$             ;NO - REPORT ERROR.
3684  016444  005237  014626                  INC     RTEMP           ;YES - INCREMENT CHAR. COUNT.
3685  016450  006137  017142                  ROL     NIN             ;LEFT JUSTIFY CURRENT NUMBER.
3686  016454  006137  017142                  ROL     NIN
3687  016460  006137  017142                  ROL     NIN
3688  016464  005237  017146                  INC     CHARC           ;INCREMENT CHAR COUNT.
3689  016470  042737  000007  017142          BIC     #7,     NIN     ;STRIP LOWER NUMBER.
3690  016476  042737  000260  017140          BIC     #260,   CHAR    ;STRIP CHAR. INPUTED.
3691  016504  053737  017140  017142          BIS     CHAR,   NIN     ;ADD TO CURRENT NUMBER.
3692  016512  000137  015650                  JMP     2$              ;LOOP.
```

```
3693  016516  005737  001614          3$:   TST     INCFLG          ;COME HERE ON <CR>;COLON FLAG SET?
3694  016522  001076                        BNE     5$              ;YES TAKE CARE OF IT.
3695  016524  023737  017150  017152        CMP     NINC,   NINC1   ;DID WE ASSEMBLE MORE NUMBERS THAN ALLOWED?
3696  016532  003156                        BGT     8$              ;IF SO - REPORT ERROR.
3697  016534  005737  014626                TST     RTEMP           ;ANY CHAR. TYPED?
3698  016540  001005                        BNE     9$              ;YES - FUDGE RETURN.
3699  016542  012737  000017  017150        MOV     #15.,   NINC
3700  016550  005037  017140                CLR     CHAR
3701  016554  013720  017142          9$:   MOV     NIN,    (0)+    ;STORE NUMBER.
3702  016560  013737  017152  014626        MOV     NINC1,RTEMP     ;FIX NUMBER INPUTED COUNT.
3703  016566  163737  017150  017152        SUB     NINC,   NINC1   ;FIX EXACT COUNT.
3704  016574  005337  017152                DEC     NINC1           ;= COUNT -1
3705  016600  012700  015036                MOV     #OUTS,R0        ;SET TO READ BACK NUMBERS.
3706  016604  012021                  21$:  MOV     (0)+,(1)+       ;STORE IN CORRECT AREA.
3707  016606  005337  017150                DEC     NINC            ;DONE ALL NUMBERS?
3708  016612  001002                        BNE     22$             ;NO - SKIP NEXT INSTR.
3709  016614  010176  000000                MOV     R1,@(6)         ;FIX CALL ON ADDR. OF NUMBERS TO INPUT.
3710  016620  005337  014626          22$:  DEC     RTEMP           ;DONE ALL NUMBERS.
3711  016624  001367                        BNE     21$             ;NO - LOOP.
3712  016626  062716  000002                ADD     #2,     (6)     ;YES - UPDATE SP.
3713  016632  013776  017152  000000        MOV     NINC1,@(6)      ;STORE ON CALL # OF NUMBERS TO INPUT.
3714  016640  062716  000002                ADD     #2,(6)          ;FIX SP.
3715  016644  104400  001161                TYPE,   $CRLF
3716  016650  005737  001576                TST     NOTYET
3717  016654  001405                        BEQ     43$
3718  016656  052777  000027  162676        BIS     #MODEN+PWFEN+ERREN+XRIF,@ICSR
3719  016664  005777  162776                TST     @ICSLMT
3720  016670  000002                  43$:  EXIT                    ;EXIT.
3721
3722  016672  013737  017142  017144  4$:   MOV     NIN,    NIN1    ;ENTER HERE WHEN COLON TYPED. STORE CURRENT #.
3723  016700  005737  001614                TST     INCFLG          ;CHECK COLON FLAG.
3724  016704  001105                        BNE     12$             ;IF SET UNKNOWN INPUT (2 COLONS!)
3725  016706  012737  000001  001614        MOV     #1,INCFLG       ;SET COLON FLAG.
3726  016714  000137  015640                JMP     1$              ;EXIT TO LOOP
3727  016720  162737  000002  017144  5$:   SUB     #2,     NIN1    ;COME HERE ON COLON FLAG SET + ANOTHER # FORMED.
3728  016726  062737  000002  017144  6$:   ADD     #2,     NIN1    ;UPDATE FORMER NUMBER.
3729  016734  023737  017150  017152        CMP     NINC,   NINC1   ;NUMBERS OVERFLOW BUFFER AREA?
3730  016742  003052                        BGT     8$              ;YES - REPORT ERROR.
3731  016744  013720  017144                MOV     NIN1,   (0)+    ;NO - STORE NEW NUMBER.
3732  016750  005237  017150                INC     NINC            ;UPDATE NUMBER COUNT.
3733  016754  023737  017142  017144        CMP     NIN,    NIN1    ;UPDATED NUMBER = CURRENT NUMBER?
3734  016762  001361                        BNE     6$              ;NO - LOOP.
3735  016764  005037  001614                CLR     INCFLG          ;YES - CLEAR COLON FLAG
3736  016770  005337  017150                DEC     NINC
3737  016774  162700  000002                SUB     #2,R0           ;FIX POINTER (STORAGE).
3738  017000  022737  000015  017140        CMP     #15,    CHAR    ;LAST CHAR TYPED A <CR>?
3739  017006  001643                        BEQ     3$              ;YES - EXIT.
3740  017010  022737  000054  017140        CMP     #',,CHAR        ;LAST CHAR TYPED A COMMA?
3741  017016  001404                        BEQ     7$              ;YES - TAKE CARE OF IT.
```

```
3742  017020  062700  000002              ADD     #2,     R0      ;FIX STORAGE POINTER.
3743  017024  000137  015640              JMP     1$              ;LOOP.
3744  017030  005737  001614      7$:     TST     INCFLG          ;ENTER HERE ON COMMA TYPED. WAS COLON-
3745  017034  001331                      BNE     5$              ;PREVIOUSLY TYPED? IF SO TAKE CARE OF IT.
3746  017036  023737  017150  017152      CMP     NINC,   NINC1   ;BUFFER OVERFLOW?
3747  017044  003011                      BGT     8$              ;IF YES - REPORT ERROR.
3748  017046  005737  017146              TST     CHARC           ;WHY CHAR. TYPED?
3749  017052  001422                      BEQ     12$             ;IF NO - REPORT ERROR.
3750  017054  013720  017142              MOV     NIN,    (0)+    ;OTHERWISE STORE CURRENT NUMBER.
3751  017060  005237  017150              INC     NINC            ;INCR. NUMBER COUNT.
3752  017064  000137  015640              JMP     1$              ;LOOP.
3753
3754  017070  104400              8$:     TYPE                    ;"ADDR. BUFFER OVERFLOW ERROR
3755  017072  026136                      MABOV                   ;RETYPE"
3756  017074  000137  015532              JMP     INOCTR
3757
3758  017100  104400              10$:    TYPE                    ;TYPE A "?"
3759  017102  026375                      MQMARK
3760  017104  000137  015532              JMP     INOCTR
3761
3762  017110  104400              11$:    TYPE                    ;TYPE "LAST CHAR TYPED NOT AN OCTAL DIGIT"
3763  017112  030620                      MINNN
3764  017114  000137  015532              JMP     INOCTR
3765  017120  104400              12$:    TYPE
3766  017122  030657                      MINKN
3767  017124  000137  015532              JMP     INOCTR
3768
3769                                      ;
3770                                      ;RETURN FOR ↑D, ↑J,↑L,↑N, ↑E, ↑S, ↑R
3771                                      ;
3772
3773  017130  000240              RASK:   NOP
3774  017132  162716  000004              SUB     #4,(6)
3775  017136  000002                      EXIT
3776
3777  017140  000000              CHAR:   0                       ;CHARACTER INPUTTED
3778  017142  000000              NIN:    0                       ;ASSEMBLED NUMBER
3779  017144  000000              NIN1:   0                       ;ASSEMBLED NUMBER, 1ST IF MULTIPLE
3780  017146  000000              CHARC:  0                       ;USED TO COUT CHARS. IN NUMBER.
```

```
3781  017150  000000                NINC:   0                       ;NUMBER OF ADDRESSED ASSEMBLED.
3782  017152  000000                NINC1:  0                       ;MUXIMUM NUMBER TO BE ASSEMBLED.
3783  017154  005237  001636        EXSET:  INC     EXPERT          ;ENTER HER ON "↑E".
3784  017160  104400                        TYPE                    ;SET EXPERT MODE.
3785  017162  025645                        MEXEN
3786  017164  000137  017130              JMP     RASK
3787
3788  017170  005037  001636        NOSET:  CLR     EXPERT          ;ENTER HERE ON "↑N".
3789  017174  104400                        TYPE                    ;SET NOIVE MODE
3790  017176  025675                        MNOEN
3791  017200  000137  017130              JMP     RASK
3792
3793
3794
3795  017204  005737  001642        LPSET:  TST     LPAV            ;ENTER HERE ON "↑L".
3796  017210  001404                        BEQ     1$              ;ANY LP AVAILABLE - IF NOT EXIT.
3797  017212  104400                        TYPE                    ;TYPE "MAKE LP READY"
3798  017214  025726                        MLEN
3799  017216  005237  001644              INC     LINEPR
3800  017222  000137  017130        1$:     JMP     RASK
3801
3802                                ;
3803                                ;ROUTINE TRANSFER CONTROL FROM ONE FILEBOX TO THE NEXT REMOTELY
3804                                ;
3805
3806  017226  104400  030523        XFRCTL: TYPE,   NFILE           ;ASK FOR NEXT FILE
3807  017232  012737  000001  017250      MOV     #1,22$
3808  017240  000401                      BR      .+4
3809  017242  000771                      BR      XFRCTL
3810  017244  104414                      INOCT
3811  017246  001706                      XTMPFL
3812  017250  000001              22$:    1
3813  017252  022737  000013  001706      CMP     #13,XTMPFL      ;CHECK BOX IS LEGAL
3814  017260  002003                      BGE     11$
3815  017262  104400  031403              TYPE,   ILLEG           ;ILLEGAL BOX
3816  017266  000757                      BR      XFRCTL
3817  017270  012737  171000  001714  11$:    MOV     #171000,XICSLT  ;SET UP MODULE RANGE.
3818  017276  012737  171776  001710          MOV     #171776,XICSR   ;SET UP CSR
3819  017304  005237  001706              INC     XTMPFL
3820  017310  005337  001706        19$:    DEC     XTMPFL          ;CALCULATE MOD RANGE & CSR
3821  017314  001407                      BEQ     18$             ;
3822  017316  162737  000010  001710      SUB     #10,XICSR
3823  017324  062737  000040  001714      ADD     #40,XICSLT
3824  017332  000766                      BR      19$
3825  017334  013737  001710  001712  18$:    MOV     XICSR,XICAR     ;CREATE ICAR
3826  017342  162737  000002  001712          SUB     #2,XICAR        ;
3827  017350  013737  001714  001716          MOV     XICSLT,XICSHG   ;CREATE HIGH LIMIT
3828  017356  062737  000040  001716          ADD     #40,XICSHG      ;
3829  017364  012737  017400  000004          MOV     #12$,@#4
3830  017372  017700  162312                  MOV     @XICSR,R0       ;SEE IF ICSR EXISTS
3831  017376  000404                          BR      97$             ;YES, OK
3832  017400  022626              12$:    POPSP2                  ;NO, TELL OPER.
3833  017402  104400  031765              TYPE,   NONXST
3834  017406  000707                      BR      XFRCTL          ;ASK AGAIN
3835  017410  012737  000006  000004  97$:    MOV     #6,@#4
3836  017416  012700  000214          99$:    MOV     #214,R0
```

```
3837  017422  012701  000216              MOV     #216,R1
3838  017426  010120                98$:  MOV     R1,(R0)+        ;SET UP .+2, IOT FOR
3839  017430  012720  000004              MOV     #4,(R0)+        ;VECTOR CHECK
3840  017434  022121                      CMP     (R1)+,(R1)+
3841  017436  022700  001000              CMP     #1000,R0
3842  017442  003371                      BGT     98$
3843  017444  104400  030543              TYPE,   NVECT           ;GET VECTOR
3844  017450  012737  000001  017466      MOV     #1,23$
3845  017456  000401                      BR      .+4
3846  017460  000756                      BR      99$
3847  017462  104414                      INOCT
3848  017464  001720                      XVEC
3849  017466  000001                23$:  1
3850  017470  022737  000776  001720      CMP     #776,XVEC       ;CHECK FOR LEGAL VECTOR
3851  017476  002404                      BLT     16$
3852  017500  022737  000234  001720      CMP     #234,XVEC
3853  017506  003403                      BLE     17$
3854  017510  104400  031403        16$:  TYPE,   ILLEG           ;NOT IN FLOATING AREA
3855  017514  000740                      BR      99$             ;RE-ASK
3856  017516  013737  001720  001722  17$:  MOV     XVEC,XICSVT     ;STORE VECTOR ADDRESS
3857  017524  062737  000002  001720      ADD     #2,XVEC
3858  017532  013737  001720  001724      MOV     XVEC,XICSV2
3859  017540  013737  000020  017724      MOV     @#20,93$
3860  017546  012737  017644  000020      MOV     #95$,@#20
3861  017554  012777  017704  162140      MOV     #94$,@XICSVT
3862  017562  012777  000340  162134      MOV     #340,@XICSV2
3863  017570  012737  000240  177776      MOV     #240,@#PS
3864  017576  052777  000001  162104      BIS     #XRIE,@XICSR
3865  017604  017700  162104              MOV     @XICSLT,R0
3866  017610  005777  162076              TST     @XICAR
3867  017614  052777  020004  162066      BIS     #MAINT2+MODEN,@XICSR
3868  017622  005000                      CLR     R0
3869  017624  005200                      INC     R0
3870  017626  001376                      BNE     .-2
3871  017630  104400  031460              TYPE,   NOINT
3872  017634  013737  017724  000020      MOV     93$,@#20
3873  017642  000410                      BR      96$
3874  017644  011605                95$:  MOV     (SP),R5
3875  017646  024545                      CMP     -(R5),-(R5)
3876  017650  022626                      POPSP2
3877  017652  104400  031523              TYPE,   FILINT
3878  017656                              TYPOCT  R5
3879  017656  010546                      MOV     R5,-(SP)        ;;SAVE R5 FOR TYPEOUT
3880  017660  104401                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3881  017662  022626                      POPSP2
3882  017664  052777  040000  162016  96$:  BIS     #MAINT3,@XICSR
3883  017672  013737  017724  000020      MOV     93$,@#20
3884  017700  000137  017416              JMP     99$
3885  017704  013737  017724  000020  94$:  MOV     93$,@#20
3886  017712  022626                      POPSP2
3887  017714  052777  040000  161766      BIS     #MAINT3,@XICSR
3888  017722  000401                      BR      .+4
3889  017724  000000                93$:  0
3890  017726  012700  000214              MOV     #214,R0
3891  017732  012701  000216              MOV     #216,R1
3892  017736  010120                90$:  MOV     R1,(R0)+                ;FILL .+2, HALT
```

```
3893  017740  012720  000000                  MOV     #0,(R0)+
3894  017744  022121                          CMP     (R1)+,(R1)+
3895  017746  022700  001000                  CMP     #1000,R0
3896  017752  003371                          BGT     90$
3897  017754  104400  030571                  TYPE,   XFERMS
3898  017760  052777  040000  161574          BIS     #MAINT3,@ICSR
3899  017766  012777  000000  161574          MOV     #0,@ICSVT2
3900  017774  013777  001570  161564          MOV     ICSVT2,@ICSVT
3901  020002  013737  001710  001562          MOV     XICSR,ICSR
3902  020010  013737  001714  001666          MOV     XICSLT,ICSLMT
3903  020016  013737  001716  001662          MOV     XICSHG,ICSHGH
3904  020024  013737  001712  001564          MOV     XICAR,ICAR
3905  020032  013737  001722  001566          MOV     XICSVT,ICSVT
3906  020040  012777  021500  161520          MOV     #ICRSRV,@ICSVT
3907  020046  005037  001612                  CLR     ICRVT
3908  020052  012737  003652  001664          MOV     #START1,CTLLOC
3909  020060  052777  000026  161474          BIS     #MODEN+ERREN+PWFEN,@ICSR
3910  020066  005037  177776                  CLR     @#PSW
3911  020072  000001                          WAIT
3912  020074  000776                          BR      .-2
3913  020076  000775                          BR      .-4
3914
3915                                  ;ROUTINE TO SET SWR FOR REMOTE USAGE
3916
3917  020100  012737  000001  020120  CONTSR: MOV     #1,1$                   ;SET UP
3918  020106  012737  001670  020116          MOV     #REMSWR,2$              ;
3919  020114  104414                          INOCT                           ;GET NEW SWR
3920  020116  000000                  2$:     0
3921  020120  000001                  1$:     1
3922  020122  005737  001672                  TST     SWRFF                   ;WHERE DID WR COME FROM
3923  020126  001415                          BEQ     3$                      ;INOCT, 3$
3924  020130  005037  001604                  CLR     DAFLG                   ;INTERRUPT RETURN THAT WAY
3925  020134  005037  001672                  CLR     SWRFF
3926  020140  104400  026071                  TYPE,   MWK
3927  020144  042777  000026  161410          BIC     #ERREN+PWFEN+MODEN,@ICSR        ;ALLOW NEW INTERRUPTS
3928  020152  052777  000026  161402          BIS     #ERREN+PWFEN+MODEN,@ICSR
3929  020160  000002                          RTI
3930  020162  000137  017130          3$:     JMP     RASK
3931
3932  020166  005737  001602          SWROUT: TST     ADBSY                   ;IS A/D WORKING?
3933  020172  001401                          BEQ     .+4                     ;NO,CONT
3934  020174  005237  001604                  INC     DAFLG                   ;SET FLAG
3935  020200  000002                          RTI                             ;YES, EXIT
3936  020202  104400                  SWROU1: TYPE                            ;ASK FOR NEW
3937  020204  027614                          MS
3938  020206  013700  001670                  MOV     REMSWR,R0
3939  020212                                  TYPOCT  R0
3940  020212  010046                          MOV     R0,-(SP)                ;;SAVE R0 FOR TYPEOUT
3941  020214  104401                          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3942  020216  104400                          TYPE
3943  020220  027627                          NEWSWR
3944  020222  000137  020100                  JMP     CONTSR
3945
3946                                          ;*
3947                                          ;#ROUTINE TO CONNECT I/O MODULES (SOFT)
3948                                          ;*
```

```
3949  020226  005137  001640          CONNTR: COM     CONNT
3950  020232  001402                          BEQ     .+6
3951  020234  104400                          TYPE
3952  020236  026004                          MAC
3953  020240  000137  017130                  JMP     RASK
3954
3955
3956                                          ;*
3957                                          ;*ROUTINE TO ASK FOR AND SET DELAY TIME
3958                                          ;* 0 DELAY OR CR BY OPERATOR - USER DEFAULT OF 3 MILLISEC.
3959
3960  020244  005737  001636          RDELA:  TST     EXPERT
3961  020250  001002                          BNE     .+6
3962  020252  104400                          TYPE                    ;ASK OPERATOR -
3963  020254  026273                          MDEL                    ;"DELAY TIME IN MS (DEFAULT=3MS)?"
3964  020256  005037  001536                  CLR     FREQ1
3965  020262  013737  001532  001534          MOV     FR3,    FREQ
3966  020270  104400                          TYPE
3967  020272  026403                          MQ
3968  020274  104406                          RDDEC                   ;GET NUMBER.
3969  020276  012637  014626                  MOV     (6)+,   RTEMP
3970  020302  001417                          BEQ     2$              ;IF ZERRO, USE DEFAULT
3971  020304  005037  001536                  CLR     FREQ1
3972  020310  005037  001534                  CLR     FREQ
3973  020314  063737  001544  001534  1$:     ADD     FR1,    FREQ    ;SET FREQUECY (OR DELAY).
3974  020322  005537  001536                  ADC     FREQ1
3975  020326  005737  001536                  TST     FREQ1
3976  020332  100404                          BMI     3$
3977  020334  005337  014626                  DEC     RTEMP
3978  020340  001365                          BNE     1$
3979  020342  000002                  2$:     EXIT
3980
3981  020344  104400                  3$:     TYPE                    ;TYPE" TIME TOO LONG".
3982  020346  026326                          MTTL
3983  020350  000735                          BR      RDELA
3984
3985                                          ;*
3986                                          ;*ROUTINE TO FORM SECOUNDARY DELAY
3987                                          ;*
3988  020352  104400  026036          RDELA3: TYPE,   MDEL1
3989  020356  104400                  RDELA4: TYPE
3990  020360  026041                          MDEL2
3991  020362  104406                          RDDEC
3992  020364  012637  014626                  MOV     (6)+,RTEMP
3993  020370  001415                          BEQ     2$
3994  020372  005037  001540                  CLR     FREQ2
3995  020376  063737  001544  001534  1$:     ADD     FR1,FREQ
3996  020404  005537  001542                  ADC     FREQ3
3997  020410  005737  001542                  TST     FREQ3
3998  020414  100405                          BMI     3$
3999  020416  005337  014626                  DEC     RTEMP
4000  020422  001365                          BNE     1$
4001  020424  000137  017130          2$:     JMP     RASK
4002  020430  104400                  3$:     TYPE
4003  020432  026326                          MTTL
4004  020434  000750                          BR      RDELA4
```

```
4005
4006
4007                                        ;*
4008                                        ;*ROUTINE TO SET UNI OR BIPOLAR A/D
4009                                        ;*CALL = QUBR
4010                                        ;*
4011
4012  020436  005737  001636        RQUBR:  TST     EXPERT
4013  020442  001002                        BNE     .+6
4014  020444  104400                        TYPE                    ;UNI OR BI-POLAR?
4015  020446  027372                        MQURB
4016  020450  104400                        TYPE
4017  020452  026403                        MQ
4018  020454  052777  000001  161100        BIS     #XRIF,@ICSR
4019  020462  017737  161200  017140        MOV     @ICSLMT,CHAR
4020  020470  005077  161066                CLR     @ICSR
4021  020474  005037  017140                CLR     CHAR
4022  020500  005737  000164        99$:    TST     REMFF
4023  020504  001454                        BEQ     1$
4024  020506  032777  002000  161046        BIT     #PWRFL,@ICSR
4025  020514  001402                        BEQ     .+6
4026  020516  000137  025120                JMP     RSTRT
4027  020522  032777  000200  161032        BIT     #MODINT,@ICSR
4028  020530  001763                        BEQ     99$
4029  020532  005237  001700                INC     MODFF
4030  020536  032777  010000  161020        BIT     #DA,@ICAR
4031  020544  001013                        BNE     98$
4032  020546  005737  001700                TST     MODFF
4033  020552  001752                        BEQ     99$
4034  020554  052777  000001  161000        BIS     #XRIF,@ICSR
4035  020562  005777  161100                TST     @ICSLMT
4036  020566  005037  001700                CLR     MODFF
4037  020572  000742                        BR      99$
4038  020574  052777  001041  160760  98$:   BIS     #TTYEN+TBMTEN+XRIF,@ICSR
4039  020602  017737  161060  014626        MOV     @ICSLMT,RTEMP
4040  020610  032777  100000  160746        BIT     #XTBMT,@ICAR
4041  020616  001774                        BEQ     .-6
4042  020620  013777  014626  161040        MOV     RTEMP,@ICSLMT
4043  020626  042777  001040  160726        BIC     #TTYEN+TBMTEN,@ICSR
4044  020634  000406                        BR      2$
4045  020636  105777  160300        1$:     TSTB    @STKS           ;WAIT FOR RESPONSE
4046  020642  100375                        BPL     1$
4047  020644  017737  160274  014626        MOV     @STKB,RTEMP
4048  020652  105777  160270        2$:     TSTB    @STPS           ;ECHO.
4049  020656  100375                        BPL     2$
4050  020660  013777  014626  160262        MOV     RTEMP,@STPB
4051  020666  042737  000240  014626        BIC     #240,RTEMP
4052  020674  023727  014626  000003        CMP     RTEMP,#3        ;^C TYPED?
4053  020702  001002                        BNE     .+6
4054  020704  000137  003652                JMP     START1
4055  020710  023727  014626  000015        CMP     RTEMP,#15       ;CR TYPED?
4056  020716  001404                        BEQ     3$
4057  020720  013737  014626  017140        MOV     RTEMP,CHAR
4058  020726  000664                        BR      99$
4059  020730  104400  001161        3$:     TYPE,   $CRLF
4060  020734  023727  017140  000125        CMP     CHAR,#'U
```

```
4061  020742  001003                        BNE     4$
4062  020744  005237  015034                INC     BIPOL
4063  020750  000402                        BR      .+6
4064  020752  005037  015034          4$:   CLR     BIPOL
4065  020756  052777  000026  160576        BIS     #MODEN+PWFEN+ERREN,@ICSR
4066  020764  000002                        EXIT
4067
4068                                  ;*
4069                                  ;*ROUTINE TO CHECK THAT AN ADDRESS TYPED IS A LEGAL ICR-ADDR.
4070                                  ;*TYPE ERROR IF NOT - ENTER WITH ADDR IN (0)
4071                                  ;*CALL = JSR PC,CKADR
4072                                  ;*
4073
4074  020766  013737  000004  021066  CKADR: MOV     @#4,3$          ;STORE LOCATION 4.
4075  020774  012737  021036  000004        MOV     #2$,@#4         ;SET TIME-OUT LOCATION FOR TRAP IF ANY.
4076  021002  005770  000000                TST     @(0)            ;TEST MOD. ADDR. TYPED.
4077  021006  000240                        NOP
4078                                  ;**
4079                                  ;** MODULE ADDRESS DOES EXIST BUT MAY NOT BE ASSOCIATED WITH THIS
4080                                  ;** FILE BOX. THEREFORE, CHECK ADDRESS FOR WITHIN FILE BOX LIMITS
4081                                  ;**
4082
4083  021010  013737  021066  000004        MOV     3$,@#4          ;RESTORE TIME OUT VECTOR
4084  021016  023710  001666                CMP     ICSLMT,(R0)
4085  021022  003022                        BGT     4$
4086  021024  023710  001662                CMP     ICSHGH,(R0)
4087  021030  003417                        BLE     4$
4088  021032  005720                1$:     TST     (0)+
4089  021034  000207                        RTS     PC              ;EXIT - SUBROUTINE.
4090  021036  104400  026535          2$:   TYPE,   MNRFN           ;TYPE ERROR MESSAGE.
4091  021042                                TYPOCT  (0)             ;TYPE ADDR.
4092  021042  011046                        MOV     (0),-(SP)       ;;SAVE (0) FOR TYPEOUT
4093  021044  104401                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4094  021046  062706  000006                ADD     #6,R6
4095  021052  162716  000002                SUB     #2,(6)          ;FIX RETURN ADDRESS ON STACK
4096  021056  013737  021066  000004        MOV     3$,@#4          ;RESTORE VECTOR ADDRESS
4097  021064  000002                        EXIT                    ;SO THAT WE RETURN AND REASK
4098                                                                ;QUESTION ON MODULE ADDR.
4099  021066  000000                3$:     .WORD   0               ;TEMP STORAGE OF LOC. 4.
4100
4101
4102  021070  104400  031426          4$:   TYPE, NRANG1            ;NOT WITHIN FILE
4103  021074                                TYPOCT  (0)
4104  021074  011046                        MOV     (0),-(SP)       ;;SAVE (0) FOR TYPEOUT
4105  021076  104401                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4106  021100  104400  031441                TYPE, NRANG2
4107  021104  005726                        TST     (R6)+           ;POP STACK PAST JSR RETURN
4108  021106  162716  000002                SUB     #2,(R6)         ;BACK POINTER TO RE-ASK QUESTION
4109  021112  000002                        EXIT                    ;RETURN
4110
```

```
                                        ;*
                                        ;*ROUTINE USED OFR GENERATION A DELAY
                                        ;*CALL=DELAT
                                        ;*

021114  013737  001534  021150  RDELAY: MOV     FREQ,   RTEMP3  ;GET DELAY TIME
021122  001411                          BEQ     2$
021124  013737  001536  021152          MOV     FREQ1,  RTEMP2
021132  005337  021150          1$:     DEC     RTEMP3          ;DELAY
021136  001375                          BNE     1$
021140  005337  021152                  DEC     RTEMP2
021144  100372                          BPL     1$
021146  000002                  2$:     EXIT                    ;RETURN
021150  000000                  RTEMP3: 0
021152  000000                  RTEMP2: 0
                                ;
                                ;DELAY ROUTINE WITH PSW SET AT 0
                                ;ROUTINE USED BY TEST 0
                                ;
021154  012737  000000  177776  RDELA0: MOV     #0,PS           ;LOWER PSW TO 0
021162  013737  001534  021150          MOV     FREQ,RTEMP3     ;GET DELAY TIME
021170  001411                          BEQ     2$
021172  013737  001536  021152          MOV     FREQ1,  RTEMP2
021200  005337  021150          1$:     DEC     RTEMP3          ;DELAY
021204  001375                          BNE     1$
021206  005337  021152                  DEC     RTEMP2
021212  100372                          BPL     1$
021214  000002                  2$:     EXIT                    ;RETURN

                                        ;*
                                        ;*ROUTINE TO GENERATE SECOUNDARY DELAY
                                        ;*CALL =DELAY2
                                        ;*
021216  013737  001540  014626  RDELA2: MOV     FREQ2,RTEMP
021224  001413                          BEQ     2$
021226  013737  001542  021152          MOV     FREQ3,RTEMP2
021234  005037  177776                  CLR     PS
021240  005337  014626          1$:     DEC     RTEMP
021244  001375                          BNE     1$
021246  005337  021152                  DEC     RTEMP2
021252  100372                          BPL     1$
021254  000002                  2$:     EXIT


                                        ;*
```

```
                                                ;* SYSTEM INITIALIZE ROUTINE
                                                ;*
                                        RINIT:
021256  013700  001104                          MOV     $ICNT,R0
021262  052777  040000  160272                  BIS     #MAINT3,@ICSR   ;RESET FUNCTION

021270  052777  000100  157644          1$:     BIS     #100,   @$TKS
021276  052777  000026  160256                  BIS     #MODEN+PWFEN+ERREN,@ICSR
021304  000002                                  EXIT

                                        ;::ROUTINE TO WAIT FOR TRANSMITTOR TO GO INACTIVE BEFORE TRANSMITTING
                                        ;:: TO THE ICR
                                        ;
021306  005777  160250                  BUSY:   TST     @ICSR   ;IS BUSY SET
021312  100006                                  BPL     1$
021314  032777  002000  160240                  BIT     #PWRFL,@ICSR
021322  001771                                  BEQ     BUSY
021324  000137  025120                          JMP     RSTRT
021330  000002                          1$:     EXIT

                                        ;ROUTINE TO CHECK REMOTE END FOR ↑P, ↑C, ↑S WHILE RUNNING

021332  052777  000041  160222        CHECK:  BIS     #TTYEN+XRIF,@ICSR
021340  017737  160322  001600                MOV     @ICSLMT,XTEMP
021346  042777  000040  160206                BIC     #TTYEN,@ICSR
021354  042737  000200  001600                BIC     #200,XTEMP
021362  022737  000003  001600                CMP     #3,XTEMP
021370  001002                                BNE     2$
021372  000137  003652                        JMP     START1
021376  022737  000023  001600        2$:     CMP     #23,XTEMP
021404  001004                                BNE     3$
021406  005237  001672                        INC     SWRFF
021412  000137  020166                        JMP     SWROUT
021416  022737  000020  001600        3$:     CMP     #20,XTEMP
021424  001002                                BNE     .+6
021426  000137  004226                        JMP     TYPECT
021432  000002                                RTI

                                        ;::ROUTINE TO WAIT 3.2 MSEC FOR ICR
                                        ;
021434  013746  001534                DLYICR: MOV     FREQ,-(6)       ;SAVE OLD FREQ DELAY ON STACK
021440  013737  001530  001534              MOV     FR32,FREQ       ;3.2 MSEC
021446  104422                              DELAY
021450  012637  001534                      MOV     (6)+,FREQ       ;GETBACK OLD FREQ
021454  000002                              EXIT
                                        ;
                                        ;ICR DELAY ROURINE WITH PSW AT ZERO
                                        ;
021456  013746  001534                DLYICX: MOV     FREQ,-(6)       ;SAVE OLD FREQ DELAY ON STACK
021462  013737  001530  001534              MOV     FR32,FREQ       ;3.2 MSEC
```

```
                                        DELAYD
021470  104412                          MOV     (6)+,FREQ       ;GETBACK OLD FREQ
021472  012637  001534                  EXIT
021476  000002
                                ;GENERAL INTERRUPT ROUTINE TO SERVICE ICR
                                ;INITIALLY ROUTINE WILL CHECK FOR DA (REMOTE TTY INPUT)
                                ;AND LINE ERRORS, THEN WILL RETURN TO SERVICE ROUTINE
                                ;STORED IN ICRVT

021500  032777  002000  160054  ICRSRV: BIT     #PWRFL,@ICSR                    ;PWR FAIL?
021506  001402                          BEQ     1$                              ;NO, CONT
021510  000137  025120                  JMP     RSTRT                           ;SERVICE POWER FAIL
021514  032777  010000  160040  1$:     BIT     #ERRBIT,@ICSR                   ;ERROR?
021522  001032                          BNE     3$                              ;YES, SERVICE AT 3$
021524  032777  010000  160032          BIT     #DA,@ICAR                       ;DA?
021532  001024                          BNE     2$                              ;SERVICE AT 2$
021534  005737  001612          4$:     TST     ICRVT                           ;TEST REQUESTING SERVICE?
021540  001014                          BNE     6$                              ;YES, GO TO 6$
021542  052777  000001  160012          BIS     #XRIF,@ICSR                     ;SET RIF BIT
021550  005777  160112                  TST     @ICSLMT                         ;ISSUE RIF
021554  042777  000026  160000          BIC     #MODEN+ERREN+PWFEN,@ICSR        ;SET INTR ENABLES
021562  052777  000026  157772          BIS     #MODEN+ERREN+PWFEN,@ICSR
021570  000002                          RTI                                     ;EXIT
021572  017737  157766  021640  6$:     MOV     @ICAR,5$                        ;CLEAR ERROR
021600  000177  160006                  JMP     @ICRVT                          ;GO TO TEST SERVICE ROUTINE
021604  000137  021332          2$:     JMP     CHECK                           ;CHECK DA
021610  005237  001676          3$:     INC     ERRCNT                          ;SERVICE ERROR
021614  017737  157744  021640          MOV     @ICAR,5$                        ;CLEAR ERROR
021622  042777  000026  157732          BIC     #ERREN+PWFEN+MODEN,@ICSR        ;RE ENABLE INTR
021630  052777  000026  157724          BIS     #ERREN+PWFEN+MODEN,@ICSR
021636  000002                          RTI                                     ;EXIT
021640  000000          5$:     0


                                ;*
                                ;*ROUTINE USED TO MODIFY PATTERN SENT TO OUTPUT MODULE
                                ;*FIRST LOCATION MAY BE CHANGED BY RPATA (PATTERN MODIFIER INPUT ROUTINE)
                                ;*CALL=CPATR
                                ;*

021642  023727  001550  000012  RCPAT:  CMP     PATRNM,#12
021650  001442                          BEQ     RCPATR
021652  000241                          CLC                             ;CLEAR C BIT
021654  005237  001552          RCPATI: INC     @#PATRN                 ;MODIFY PATTERN
021660  103412                          BCS     1$
021662  022737  005137  021654          CMP     #5137,RCPATI            ;DOING A COMPLEMENT PATERN?
021670  001406                          BEQ     1$                      ;IF SO DON'T ADD CARRY IF ANY!
021672  063737  001554  001552          ADD     PATRNC,PATRN
021700  005037  001554                  CLR     PATRNC
021704  000002                          EXIT
021706  023727  021654  006037  1$:     CMP     RCPATI,#6037
021714  001004                          BNE     2$
021716  012737  100000  001554          MOV     #100000,PATRNC
021724  000002                          EXIT
```

```
 4268  021726  012737  000001  001554  2$:     MOV     #1,PATRNC
 4269  021734  000002                          EXIT
 4270                                          ;*MODIFIER LIST
 4271  021736  005237                  RCPATL: 5237                    ;*INCREMENTING PATTERN          0
 4272  021740  005337                          5337                    ;*DECREMENTING PATTERN          1
 4273  021742  005737                          5737                    ;*NO CHANGE OF PATTERN          2
 4274  021744  006137                          6137                    ;*ROTATE LEFT PATTERN           3
 4275  021746  006037                          6037                    ;*ROTATE RIGHT PATTERN          4
 4276  021750  006237                          6237                    ;*RANDOM NUMBER GENERATOR         5
 4277  021752  006337                          6337                    ;*ARITH. SHIFT LEFT PATTERN     6
 4278  021754  005137                          5137                    ;*COMPLEMENT PATTERN            7
 4279
 4280
 4281                                          ;*
 4282                                          ;*ALTERNATE CHANGE PATTERN ROUTINE IF RANDOME
 4283                                          ;*NUMBER FOR PATTERN IS SELECTED.
 4284                                          ;*
 4285  021756                  RCPATR:
 4286  021756  004737  022212                  JSR     PC,$RAND        ;GENERATE A RANDOM NUMBER.
 4287  021762  013737  022340  001552          MOV     $LONUM,PATRN    ;PUT NUMBER IN PATTERN.
 4288  021770  000002                          EXIT
 4289
 4290
 4291                                          ;*
 4292                                          ;*PATTERN INPUTTER AND MODIFIER ROUTINE
 4293                                          ;*FORM PATTERN MODIFIER, PATTERN.
 4294                                          ;*CALL=PATAR
 4295                                          ;*
 4296  021772  005737  001636          RPATA:  TST     EXPERT
 4297  021776  001002                          BNE     .+6
 4298  022000  104400                          TYPE                    ;ASK OPERATOR-"PATTERN MODIFIER AND
```

```
                                        MPPM                    ;PATTERN?"
                                        MOV     #PATRNM, 1$     ;SET TO INPUT PATTERN + MODIFIER
                                        MOV     #2,      1$+2
                                        CLR     PATRNM          ;SET DEFAULTS: 0 PATTERN
                                        CLR     PATRN           ;PATTERN MOD. = INC.
                                        BR      .+4
                                        BR      RPATA
                                        INOCT                   ;GET THEM.
                                1$:     PATRNM
                                        2
                                        MOV     #RCPATL,RTEMP   ;POINT TO BEGINNING OF MODIFIER LIST.
                                        BIT     #177770,PATRNM  ;LEGAL PATTERN MODIFIER?
                                        BEQ     2$              ;IF YES 2$
                                        TYPE,   MIVP            ;NO TYPE ERROR MESSAGE.
                                        BR      RPATA           ;REASK QUESTION.
                                2$:
                                        ASL     PATRNM
                                        BIC     #177761, PATRNM ;FIX ADDITIVE.
                                        ADD     PATRNM, RTEMP   ;POINT TO MODIFIER.
                                        MOV     @RTEMP, RCPAT+12        ;CHANGE PATTERN MODIFIER ROUTINE.
                                        EXIT                    ;RETURN.
                                        .$SAVE
                                STARS
                                ;*****************************************************************

                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                ;*SAVE R0-R5
                                ;*CALL:
                                ;*      SAVREG
                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                ;*
                                ;*TOP---(+16)
                                ;* +2---(+18)
                                ;* +4---R5
                                ;* +6---R4
                                ;* +8---R3
                                ;*+10---R2
                                ;*+12---R1
                                ;*+14---R0

                                $SAVREG:
                                        PUSH    <R0,R1,R2,R3,R4,R5>
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
                                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                        MOV     24(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
                                        MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
                                        MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
                                        MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
                                        RTI

                                ;*RESTORE R0-R5
```

```
                                        ;*CALL:
                                        ;*      RESREG
022154                                  $RESREG:
022154  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
022160  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
022164  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
022170  012666  000022                          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
022174                                          POP     <R5,R4,R3,R2,R1,R0>
022174  012605                                  MOV     (SP)+,R5        ;;POP STACK INTO R5
022176  012604                                  MOV     (SP)+,R4        ;;POP STACK INTO R4
022200  012603                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
022202  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
022204  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
022206  012600                                  MOV     (SP)+,R0        ;;POP STACK INTO R0
022210  000002                                  RTI
022212                                          .$RAND
022212                                  STARS
                                        ;*******************************************************************

                                        .SBTTL  RANDOM NUMBER GENERATOR ROUTINE

                                        ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
                                        ;*WITH A RANGE OF 0 TO 2(+33)-1.
                                        ;*CALL:
                                        ;*      JSR     PC,$RAND        ;;CALL THE ROUTINE
                                        ;*      RETURN                  ;;RETURN HERE THE RANDOM
                                        ;*                              ;;NUMBER WILL BE IN
                                        ;*                              ;;$HINUM,$LONUM

022212                                  $RAND:  PUSH    <R0,R1,R2,R3>
022212  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
022214  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
022216  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
022220  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
022222  013700  022340                         MOV     $LONUM,R0       ;;SET R0 WITH LOW
022226  013701  022336                         MOV     $HINUM,R1       ;;SET R1 WITH HIGH
022232  012703  177771                         MOV     #-7,R3          ;;SET SHIFT COUNT
022236  005002                                 CLR     R2              ;;ZERO R2
022240  006300                         1$:     ASL     R0              ;;SHIFT R0 LEFT AND
022242  006101                                 ROL     R1              ;;ROTATE CARRY INTO R1 AND
022244  006102                                 ROL     R2              ;;ROTATE CARRY INTO R2
022246  005203                                 INC     R3              ;;CHECK FOR DONE
022250  001373                                 BNE     1$              ;;CONTINUE SHIFT LOOP
022252  063700  022340                         ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
022256  005501                                 ADC     R1              ;;PROPOGATE CARRY
022260  063701  022336                         ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
022264  005502                                 ADC     R2              ;;PROPOGATE CARRY
022266  062700  001057                         ADD     #1057,R0        ;;ADD LOW CONSTANT
022272  005501                                 ADC     R1              ;;PROPOGATE CARRY
022274  005502                                 ADC     R2              ;;PROPOGATE CARRY
022276  062701  047401                         ADD     #47401,R1       ;;ADD HIGH CONSTANT
022302  005502                                 ADC     R2              ;;PROPOGATE CARRY
022304  062702  000006                         ADD     #6,R2           ;;ADD HIGHEST CONSTART
022310  060200                                 ADD     R2,R0           ;;REPRIME R0 WITH HIGHEST DIGIT
022312  005501                                 ADC     R1              ;;PROPOGATE CARRY
022314  010037  022340                         MOV     R0,$LONUM       ;;SAVE R0
```

```
4411  022320  010137  022336                    MOV     R1,$HINUM       ;;SAVE R1
4412  022324                                    POP     <R3,R2,R1,R0>
4413  022324  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
4414  022326  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
4415  022330  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
4416  022332  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
4417  022334  000207                            RTS     PC              ;;RETURN
4418  022336  176543                $HINUM:     .WORD   176543
4419  022340  123456                $LONUM:     .WORD   123456
4420  022342                                    .$SB2D
4421  022342                        STARS
4422                                ;*****************************************************************
4423
4424                                .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
4425
4426                                ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
4427                                ;*UNSIGNED DECIMAL ASCIZ NUMBER.
4428                                ;*CALL
4429                                ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
4430                                ;*      JSR     PC,@#$SB2D      ;;CALL
4431                                ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
4432
4433
4434  022342  016637  000002  022366  $SB2D:   MOV     2(SP),1$        ;;SAVE BINARY NUMBER
4435  022350  012746  022366                    MOV     #1$,-(SP)       ;;SET POINTER
4436  022354  004737  022372                    JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
4437  022360  012666  000002                    MOV     (SP)+,2(SP)     ;;PICKUP POINTER
4438  022364  000207                            RTS     PC              ;;RETURN
4439  022366  000000  000000          1$:       .WORD   0,0
4440  022372                                    .$DB2D
4441  022372                        STARS
4442                                ;*****************************************************************
4443
4444                                .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
4445
4446                                ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
4447                                ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
4448                                ;*POSITIVE.
4449                                ;*CALL
4450                                ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
4451                                ;*      JSR     PC,@#$DB2D
4452                                ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
4453                                                                ;;IS ON THE STACK
4454
4455
4456  022372  104407                $DB2D:    SAVREG                  ;;SAVE REGISTERS
4457  022374  016602  000002                    MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
4458  022400  012700  022552                    MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
4459  022404  010066  000002                    MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
4460  022410  012201                            MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
4461  022412  012202                            MOV     (R2)+,R2
4462  022414  012737  000012  022470            MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
4463  022422  012704  022502                    MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
4464  022426  012705  022504                    MOV     #$TNPWR+2,R5
4465  022432  005003                1$:       CLR     R3              ;;CLEAR PARTIAL
4466  022434  161401                2$:       SUB     (R4),R1         ;;SUBTRACT TEN POWER
```

```
4467  022436  005602                        SBC     R2
4468  022440  161502                        SUB     (R5),R2
4469  022442  002402                        BLT     3$                ;;BR IF TEN POWER TO LARGE
4470  022444  005203                        INC     R3                ;;ADD 1 TO PARTIAL
4471  022446  000772                        BR      2$                ;;LOOP
4472  022450  062401              3$:       ADD     (R4)+,R1          ;;RESTORE SUBTRACTED VALUE
4473  022452  005502                        ADC     R2
4474  022454  062402                        ADD     (R4)+,R2
4475  022456  022525                        CMP     (R5)+,(R5)+       ;;MOVE TO NEXT TEN POWER
4476  022460  052703  000060                BIS     #'0,R3            ;;CHANGE PARTIAL TO ASCII
4477  022464  110320                        MOVB    R3,(R0)+          ;;SAVE IT
4478  022466  005327                        DEC     (PC)+             ;;DONE?
4479  022470  000000              4$:       .WORD   0
4480  022472  001357                        BNE     1$                ;;BR IF NO
4481  022474  105020                        CLRB    (R0)+             ;;TERMINATOR
4482  022476  104410                        RESREG                    ;;RESTORE REGISTERS
4483  022500  000207                        RTS     PC                ;;RETURN
4484  022502  145000              $TNPWR:   145000                    ;;1.0E09
4485  022504  035632                         35632
4486  022506  160400                        160400                    ;;1.0E08
4487  022510  002765                         2765
4488  022512  113200                        113200                    ;;1.0E07
4489  022514  000230                         230
4490  022516  041100                        041100                    ;;1.0E06
4491  022520  000017                         17
4492  022522  103240                        103240                    ;;1.0E05
4493  022524  000001                         1
4494  022526  023420                         23420                    ;;1.0E04
4495  022530  000000                         0
4496  022532  001750                         1750                     ;;1.0E03
4497  022534  000000                         0
4498  022536  000144                         144                      ;;1.0E02
4499  022540  000000                         0
4500  022542  000012                         12                       ;;1.0E01
4501  022544  000000                         0
4502  022546  000001                         1                        ;;1.0E00
4503  022550  000000                         0
4504  022552  000014              $DECVL:   .BLKB   12.               ;;RESERVE STORAGE FOR ASCIZ STRING
4505
4506
4507
4508
4509
4510
4511
4512          000000                        $SWR=0
4513
4514  022566                                .SERROR REMHLT
4515  022566                      STARS
4516                              ;*****************************************************************
4517
4518                              .SBTTL  ERROR HANDLER ROUTINE
4519
4520                              ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
4521                              ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4522                              ;*AND GO TO REMHLT ON ERROR
```

```
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
                                        $ERROR:
022566                                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
022566  105237  001103                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
022572  001775                                MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
022574  013777  001102  156336                INC     $ERTTL          ;;INC THE ERROR COUNT
022602  005237  001112                        MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
022606  011637  001116                        SUB     #2,$ERRPC
022612  162737  000002  001116                MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
022620  117737  156272  001114                JSR     PC,REMHLT       ;;GO TO USER ERROR ROUTINE
022626  004737  022634                        20$:
022632                                  2$:
022632                                        RTI                     ;;RETURN
022632  000002

                                        ;
                                        ;ROUTINE TO INHIBIT SW15 TO WORK AS HALT ON ERROR IF RUNNING REMOTELY
                                        ;

022634  032777  020000  156274        REMHLT: BIT     #SW13,@SWR              ;IS INHIBIT TYPEOUT SET
022642  001002                                BNE     1$                      ;YES, SKIP TYPEOUT OF ERROR
022644  004737  022670                        JSR     PC,$ERRTYP              ;GO TYPE ERROR
022650  005737  000164                1$:     TST     REMFF                   ;ARE WE RUNNING REMOTELY
022654  001004                                BNE     2$                      ;YES, THEN DON'T ALLOW HALT
022656  005777  156254                        TST     @SWR                    ;HALT ON ERROR SET
022662  100001                                BPL     2$                      ;NO, EXIT
022664  000000                                HALT                            ;HALT ON ERROR
022666  000207                        2$:     RTS     PC                      ;EXIT SUBROUTINE


022670                                  STARS
                                        ;******************************************************************

                                        .SBTTL  ERROR MESSAGE HANDLER

                                        ;THIS ROUTINE TAKES THE FORMAT "ERROR N" AND $ERRTB OF SYSMAC, BUT TYPES
                                        ;THE ERROR MESSAGE AND ERROR PC ON ONE LINE, THEN DH AND DT INFORMATION
                                        ;ON LINE BY LINE BASIS
                                        ;
                                        ;I.E    ADDS DID NOT INTERRUPT    PC - XXXXXX
                                        ;       A/D ADDR          YYYYYY
                                        ;
                                        ;USES DH FOR ASCII MESSAGES
                                        ;USES DT FOR DATA


022670  104400  001161        $ERRTYP:        TYPE    ,$CRLF  ;TYPE CRLF
022674  010046                        MOV     R0,-(SP)        ;SAVE R0
022676  010146                        MOV     R1,-(SP)        ;SAVE R1
022700  010246                        MOV     R2,-(SP)        ;SAVE R2
022702  005000                        CLR     R0              ;CLEAR R0
022704  153700  001114                BISB    @#$ITEMB,R0     ;GET ERROR NUMBER
022710  005300                        DEC     R0              ;ADJUST SO TABLE WILL WORK
```

```
4579  022712  006300                    ASL     R0
4580  022714  006300                    ASL     R0
4581  022716  006300                    ASL     R0
4582  022720  062700  001164            ADD     #$ERRTB,R0      ;ADD TABLE ADDRESS
4583  022724  012037  022734            MOV     (R0)+,2$        ;GET ERROR MESSAGE
4584  022730  001413                    BEQ     3$              ;SKIP IF ZERO
4585  022732  104400                    TYPE
4586  022734  000000              2$:   .WORD   0               ;ERROR MESSAGE
4587  022736  104400  027552            TYPE    ,PCPRT          ;PRINT PC
4588  022742                            TYPOCT  $ERRPC,<ERROR ADDRESS>
4589  022742  013746  001116            MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
4590                                                            ;;ERROR ADDRESS
4591  022746  104401                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4592  022750  004737  023050            JSR     PC,ERRDLY       ;DELAY
4593  022754  104400  001161      21$:  TYPE    $CRLF           ;CRLF
4594  022760  012001              3$:   MOV     (R0)+,R1        ;PICKUP DATA HEADER
4595  022762  001426                    BEQ     22$             ;IF ZERO EXIT
4596  022764  012002                    MOV     (R0)+,R2        ;PICKUP DATA
4597  022766  012137  023102            MOV     (R1)+,DATCNT    ;PICKUP LENGTHOF DATA
4598  022772  005737  023102      5$:   TST     DATCNT          ;IS COUNT ZERO
4599  022776  001420                    BEQ     22$             ;IF SO, EXIT
4600  023000  012137  023006            MOV     (R1)+,4$        ;GETDATA HEADER
4601  023004  104400                    TYPE
4602  023006  000000              4$:   .WORD   0
4603  023010  012237  023104            MOV     (R2)+,OCTMP     ;GET DATA
4604  023014                            TYPOCT  @OCTMP
4605  023014  017746  000064            MOV     @OCTMP,-(SP)    ;;SAVE @OCTMP FOR TYPEOUT
4606  023020  104401                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4607  023022  005337  023102            DEC     DATCNT
4608  023026  004737  023050            JSR     PC,ERRDLY
4609  023032  104400  001161      6$:   TYPE,   $CRLF
4610  023036  000755                    BR      5$
4611  023040  012602              22$:  MOV     (SP)+,R2
4612  023042  012601                    MOV     (SP)+,R1
4613  023044  012600                    MOV     (SP)+,R0
4614  023046  000207                    RTS     PC
4615
4616  023050  005737  000164      ERRDLY: TST   REMFF
4617  023054  001001                    BNE     .+4
4618  023056  000207                    RTS     PC
4619  023060  005037  023100            CLR     ERRTM1
4620  023064  104413              1$:   CKRMTT
4621  023066  005337  023100            DEC     ERRTM1
4622  023072  001374                    BNE     1$
4623  023074  000207                    RTS     PC
4624
4625  023076  000000      ERRTMP: 0
4626  023100  000000      ERRTM1: 0
4627  023102  000000      DATCNT: 0
4628  023104  000000      OCTMP:  0
4629  023106  027501  020104  042101  AADMS:  .ASCIZ  ?A/D ADDR  ?
4630  023114  051104  020040      000
4631  023121      115 042117  046125  MADMS:  .ASCIZ  /MODULE ADDR  /
4632  023126  020105  042101  051104
4633  023134  020040      000
4634  023137      107 047517  020104  GDDT:   .ASCIZ  /GOOD DATA  /
```

```
4635  023144  040504  040524  020040
4636  023152      000
4637  023153      102  042101  042040  BDDT:   .ASCIZ  /BAD DATA  /
4638  023160  052101  020101  000040
4639  023166  054105  023520  020104  EXIC:   .ASCIZ  /EXP'D ICAR  /
4640  023174  041511  051101  020040
4641  023202      000
4642  023203      122  041505  042047  REIC:   .ASCIZ  /REC'D ICAR  /
4643  023210  044440  040503  020122
4644  023216  000040
4645  023220  047103  051124  040440  CNAD:   .ASCIZ  /CNTR ADDR  /
4646  023226  042104  020122  000040
4647  023234  052504  046101  040440  DUAD:   .ASCIZ  /DUAL ADDR  /
4648  023242  042104  020122  000040
4649
4650                                  .EVEN
4651
4652  023250  000001  023106          DH1:    .WORD   1,AADMS
4653  023254  000003  023121  023137  DH2:    .WORD   3,MADMS,GDDT,BDDT
4654  023262  023153
4655  023264  000003  023121  023166  DH3:    .WORD   3,MADMS,EXIC,REIC
4656  023272  023203
4657  023274  000001  023220          DH4:    .WORD   1,CNAD
4658  023300  000002  023106  023234  DH5:    .WORD   2,AADMS,DUAD
4659
4660
4661
4662
4663  023306                  .STYPOCT
4664  023306          STARS
4665          ;*****************************************************************
4666
4667          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
4668
4669          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4670          ;*OCTAL (ASCII) NUMBER AND TYPE IT.
4671          ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4672          ;*CALL:
4673          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
4674          ;*      TYPOS                   ;;CALL FOR TYPEOUT
4675          ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4676          ;*      .BYTE   M               ;;M=1 OR 0
4677          ;*                                      ;;1=TYPE LEADING ZEROS
4678          ;*                                      ;;0=SUPPRESS LEADING ZEROS
4679          ;*
4680          ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
4681          ;*$TYPOS OR $TYPOC
4682          ;*CALL:
4683          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
4684          ;*      TYPON                   ;;CALL FOR TYPEOUT
4685          ;*
4686          ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
4687          ;*CALL:
4688          ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
4689          ;*      TYPOC                   ;;CALL FOR TYPEOUT
4690
```

```
 4691  023306  017646  000000            STYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 4692  023312  116637  000001  023531            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 4693  023320  112637  023533                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 4694  023324  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 4695  023330  000406                            BR      $TYPON
 4696  023332  112737  000001  023531  STYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 4697  023340  112737  000006  023533            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 4698  023346  112737  000005  023530  STYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 4699  023354  010346                            MOV     R3,-(SP)        ;;SAVE R3
 4700  023356  010446                            MOV     R4,-(SP)        ;;SAVE R4
 4701  023360  010546                            MOV     R5,-(SP)        ;;SAVE R5
 4702  023362  113704  023533                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 4703  023366  005404                            NEG     R4
 4704  023370  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 4705  023374  110437  023532                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 4706  023400  113704  023531                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 4707  023404  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 4708  023410  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
 4709  023412  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 4710  023414  000404                            BR      3$              ;;GO DO MSB
 4711  023416  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
 4712  023420  006105                            ROL     R5
 4713  023422  006105                            ROL     R5
 4714  023424  010503                            MOV     R5,R3
 4715  023426  006103                    3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 4716  023430  105337  023532                    DECB    $OMODE          ;;TYPE THIS DIGIT?
 4717  023434  100016                            BPL     7$              ;;BR IF NO
 4718  023436  042703  177770                    BIC     #177770,R3      ;;GET RID OF JUNK
 4719  023442  001002                            BNE     4$              ;;TEST FOR 0
 4720  023444  005704                            TST     R4              ;;SUPPRESS THIS 0?
 4721  023446  001403                            BEQ     5$              ;;BR IF YES
 4722  023450  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 4723  023452  052703  000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 4724  023456  052703  000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 4725  023462  110337  023526                    MOVB    R3,8$           ;;SAVE FOR TYPING
 4726  023466  104400  023526                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
 4727  023472  105337  023530            7$:     DECB    $OCNT           ;;COUNT BY 1
 4728  023476  003347                            BGT     2$              ;;BR IF MORE TO DO
 4729  023500  002402                            BLT     6$              ;;BR IF DONE
 4730  023502  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 4731  023504  000744                            BR      2$              ;;GO DO THE LAST DIGIT
 4732  023506  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
 4733  023510  012604                            MOV     (SP)+,R4        ;;RESTORE R4
 4734  023512  012603                            MOV     (SP)+,R3        ;;RESTORE R3
 4735  023514  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 4736  023522  012616                            MOV     (SP)+,(SP)
 4737  023524  000002                            RTI                     ;;RETURN
 4738  023526     000            8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 4739  023527     000                    .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 4740  023530     000            $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 4741  023531     000            $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 4742  023532  000000            $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 4743  023534                            .$POWER ,,PWRMSG,START
 4744  023534                    STARS
 4745                             ;*****************************************************************
 4746
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;POWER DOWN ROUTINE
023534  012737  023662  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
023542  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
023550                                  PUSH    <R0,R1,R2,R3,R4,R5>
023550  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
023552  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
023554  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
023556  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
023560  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
023562  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
023564  010637  023666                  MOV     SP,$SAVR6       ;;SAVE SP
023570  012737  023602  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
023576  000000                          HALT
023600  000776                          BR      .-2             ;;HANG UP

                                        ;POWER UP ROUTINE
023602  013706  023666          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
023606  005037  023666                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
023612  005237  023666          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
023616  001375                          BNE     1$              ;;OF  WORD
023620                                  POP     <R5,R4,R3,R2,R1,R0>
023620  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
023622  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
023624  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
023626  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
023630  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
023632  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
023634  012737  023534  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
023642  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
023650  104400                          TYPE                    ;;REPORT THE POWER FAILURE
023652  027642          $PWRMG: .WORD   PWRMSG                  ;;POWER FAIL MESSAGE POINTER
023654  012716                  MOV     (PC)+,(SP)              ;;RESTART AT START
023656  001726          $PWRAD: .WORD   START                   ;;RESTART ADDRESS
023660  000002                  RTI
023662  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
023664  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
023666  000000          $SAVR6: 0                               ;;PUT THE SP HERE


        164000                  $SWR=164000

023670                          .$SCOPE
023670                  STARS
                        ;*****************************************************************

                        .SBTTL  SCOPE HANDLER ROUTINE

                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                        ;*SW14=1         LOOP ON TEST
                        ;*SW11=1         INHIBIT ITERATIONS
                        ;*CALL
```

```
                                        ;*      SCOPE           ;;SCOPE=IOT
                                        $SCOPE:
023670                                  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
023670  032777  040000  155240                  BNE     $OVER           ;;YES IF SW14=1
023676  001055
                                        ;#####START OF CODE FOR THE XOR TESTER#####
023700  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
023702  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
023706  012737  023726  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
023714  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
023720  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023724  000436                                  BR      $SVLAD          ;;GO TO THE NEXT TEST
023726  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
023730  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023734  000436                                  BR      $OVER           ;;LOOP ON THE PRESENT TEST
023736                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
023736  105737  001103                  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
023742  001404                                  BEQ     3$              ;;BR IF NO
023744  105037  001103                  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
023750  005037  001156                          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
023754  032777  004000  155154          3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
023762  001011                                  BNE     1$              ;;BR IF YES
023764  005737  001100                          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
023770  001406                                  BEQ     1$              ;;      INHIBIT ITERATIONS
023772  005237  001104                          INC     $ICNT           ;;INCREMENT ITERATION COUNT
023776  023737  001156  001104                  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
024004  002012                                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
024006  012737  000001  001104          1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
024014  013737  024046  001156                  MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
024022  105237  001102                  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
024026  011637  001106                          MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
024032  013777  001102  155100          $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
024040  013716  001106                          MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
024044  000002                                  RTI                     ;;FIXES PS
024046  003720                          $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS


024050                                  STARS
                                        ;*********************************************************************

                                        .SBTTL  TTY INPUT ROUTINE(CONSOLE AND REMOTE)

                                        ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM EITHER TTY
                                        ;*CALL:
                                        ;*      RDCHR           ;INPUT SINGLE CHAR TO STACK
                                        ;*      RETURN HERE
                                        ;

024050  011646                          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN PC
024052  016666  000004  000002                  MOV     4(SP),2(SP)     ;;SAVE PSW
024060  052777  000001  155474                  BIS     #XRIF,@ICSR
024066  017766  155574  000004                  MOV     @ICSLMT,4(SP)
024074  005077  155462                          CLR     @ICSR
024100  005737  000164                  2$:     TST     REMFF           ;;RUNNING REMOTELY
024104  001445                                  BEQ     1$              ;;NO, THEN SERVICE LOCAL TTY
```

```
4859  024106  032777  002000  155446          BIT     #PWRFL,@ICSR
4860  024114  001402                          BEQ     .+6
4861  024116  000137  025120                  JMP     RSTRT
4862  024122  032777  000200  155432          BIT     #MODINT,@ICSR
4863  024130  001763                          BEQ     2$
4864  024132  005237  001700                  INC     MODFF
4865  024136  032777  010000  155420          BIT     #DA,@ICAR       ;;DATA AVAILABLE FROM REMOTE TTY
4866  024144  001013                          BNE     3$              ;;YES, CONTINUE
4867  024146  005737  001700                  TST     MODFF
4868  024152  001752                          BEQ     2$
4869  024154  052777  000001  155400          BIS     #XRIF,@ICSR
4870  024162  005777  155500                  TST     @ICSLMT
4871  024166  005037  001700                  CLR     MODFF
4872  024172  000742                          BR      2$
4873  024174  052777  000041  155360  3$:     BIS     #TTYEN+XRIF,@ICSR       ;;SET ENABLE AND RIF
4874  024202  017766  155460  000004          MOV     @ICSLMT,4(SP)   ;;READ REMOTE TTY
4875  024210  042777  000040  155344          BIC     #TTYEN,@ICSR    ;;CLEAR TTY ENABLE
4876  024216  000406                          BR      4$              ;;SKIP OVER LOCAL PART
4877  024220  105777  154716          1$:     TSTB    @$TKS           ;;READ LOCALLY
4878  024224  100325                          BPL     2$
4879  024226  117766  154712  000004          MOVB    @$TKB,4(SP)
4880  024234  042766  177600  000004  4$:     BIC     #^C(177),4(SP)  ;;GET RID OF JUNK IF ANY
4881  024242  052777  000024  155312          BIS     #MODEN+PWFEN,@ICSR
4882  024250  000002                          RTI                     ;;GO BACK TO USER
4883
4884  024252                          STARS
4885                                  ;*********************************************************************
4886
4887                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4888                                  ;*CALL:
4889                                  ;*      RDLIN                   ;;INPUT A STRING FROM EITHER TTY
4890                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON STACK
4891                                  ;*                              ;;TERMINATOR WILL BE A ZERO BYTE
4892
4893  024252  010346          $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
4894  024254  012703  024360  1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
4895  024260  022703  024370  2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL
4896  024264  101405                  BLOS    4$              ;;BR IF YES
4897  024266  104404                  RDCHR                   ;;GO FETCH ONE CHARACTER
4898  024270  112613                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
4899  024272  122713  000177          CMPB    #177,(R3)       ;;IS IT A RUBOUT
4900  024276  001003                  BNE     3$              ;;SKIP IF NOT
4901  024300  104400  001160  4$:     TYPE    ,$QUES          ;;TYPE A QUES MARK
4902  024304  000763                  BR      1$              ;;CLEAR BUFFER AND LOOP
4903  024306  111337  024356  3$:     MOVB    (R3),9$         ;;ECHO THE CHAR
4904  024312  104400  024356          TYPE    ,9$
4905  024316  122723  000015          CMPB    #15,(R3)+       ;;CHECK FOR RETURN
4906  024322  001356                  BNE     2$              ;;LOOP IF NOT RETURN
4907  024324  105063  177777          CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
4908  024330  104400  001162          TYPE,   $LF             ;;TYPE LF
4909  024334  012603                  MOV     (SP)+,R3        ;;RESTORE R3
4910  024336  011646                  MOV     (SP),-(SP)      ;;ADJUST STACK AND PUT ADDRESS OF THE
4911  024340  016666  000004  000002  MOV     4(SP),2(SP)     ;;FIRST ASCII CHAR ON IT
4912  024346  012766  024360  000004  MOV     #$TTYIN,4(SP)   ;;
4913  024354  000002                  RTI                     ;;RETURN
4914
```

```
 4915  024356  000                  9$:     .BYTE   0
 4916  024357  000                          .BYTE   0
 4917  024360  000010               $TTYIN: .BLKB   8.
 4918
 4919
 4920  024370                               .$RDDEC
 4921  024370                       STARS
 4922                               ;*****************************************************************
 4923
 4924                               .SBTTL  READ A DECIMAL NUMBER FROM THE TTY
 4925
 4926                               ;*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
 4927                               ;*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
 4928                               ;*ARE READ A "?" FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
 4929                               ;*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
 4930                               ;*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
 4931                               ;*POSITIVE 32767 TO NEGATIVE 32768.
 4932                               ;*CALL:
 4933                               ;*      RDDEC                   ;;READ A DECIMAL NUMBER
 4934                               ;*      RETURN HERE             ;;NUMBER IS ON TOP OF THE STACK
 4935                               ;
 4936
 4937  024370  011646               $RDDEC: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR
 4938  024372  016666  000004  000002       MOV     4(SP),2(SP)     ;;THE INPUT NUMBER
 4939  024400                               PUSH    <R0,R1,R2>
 4940  024400  010046                       MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 4941  024402  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 4942  024404  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 4943  024406  104405               1$:     RDLIN                   ;;READ AN ASCIZ LINE
 4944  024410  012600                       MOV     (SP)+,R0        ;;ADDRESS OF 1ST CHAR.
 4945  024412  010037  024536               MOV     R0,6$           ;;SAVE INCASE OF BAD INPUT
 4946  024416  005046                       CLR     -(SP)           ;;CLEAR DATA WORD
 4947  024420  005002                       CLR     R2              ;;SIGN SET POSITIVE
 4948  024422  122710  000055               CMPB    #'-,(R0)        ;;SEE IF A MINUS SIGN WAS TYPED
 4949  024426  001001                       BNE     2$              ;;BR IF NO MINUS SIGN
 4950  024430  112002                       MOVB    (R0)+,R2        ;;SAVE FOR LATER USE
 4951  024432  112001               2$:     MOVB    (R0)+,R1        ;;PICKUP THIS CHARACTER
 4952  024434  001424                       BEQ     3$              ;;GET OUT IF ZERO
 4953  024436  122701  000060               CMPB    #'0,R1          ;;MAKE SURE THIS CHARACTER
 4954  024442  003032                       BGT     5$              ;;IS A DIGIT BETWEEN 0 & 9
 4955  024444  122701  000071               CMPB    #'9,R1
 4956  024450  002427                       BLT     5$
 4957  024452  032716  170000               BIT     #^C7777,(SP)    ;;DON'T LET NUMBER GET TO BIG
 4958  024456  001024                       BNE     5$              ;;BR IF NUMBER WOULD OVERFLOW
 4959  024460  006316                       ASL     (SP)            ;;*2
 4960  024462  011646                       MOV     (SP),-(SP)      ;;SAVE FOR LATER
 4961  024464  006316                       ASL     (SP)            ;;*4
 4962  024466  006316                       ASL     (SP)            ;;*8.
 4963  024470  062616                       ADD     (SP)+,(SP)      ;;*10.
 4964  024472  102416                       BVS     5$              ;;OVERFLOW ISN'T ALLOWED
 4965  024474  162701  000060               SUB     #'0,R1          ;;STRIP AWAY THE ASCII JUNK
 4966  024500  060116                       ADD     R1,(SP)         ;;ADD IN THIS DIGIT
 4967  024502  102412                       BVS     5$              ;;OVERFLOW ISN'T ALLOWED
 4968  024504  000752                       BR      2$              ;;LOOP
 4969  024506  005702               3$:     TST     R2              ;;CHECK IF NUMBER IS NEG
 4970  024510  001401                       BEQ     4$              ;;BR IF NO
```

```
 4971  024512  005416                        NEG     (SP)                ;;YES--NEGATE THE NUMBER
 4972  024514  012666  000012          4$:   MOV     (SP)+,12(SP)        ;;SAVE THE RESULT
 4973  024520                              POP     <R2,R1,R0>
 4974  024520  012602                      MOV     (SP)+,R2            ;;POP STACK INTO R2
 4975  024522  012601                      MOV     (SP)+,R1            ;;POP STACK INTO R1
 4976  024524  012600                      MOV     (SP)+,R0            ;;POP STACK INTO R0
 4977  024526  000002                      RTI                         ;;RETURN
 4978
 4979  024530  005726                5$:   TST     (SP)+               ;;CLEAN PARTIAL NUMBER FROM STACK
 4980  024532  105010                      CLRB    (R0)                ;;SET A TERMINATOR
 4981  024534  104400                      TYPE                        ;;TYPE THE INPUT UP TO BAD CHAR.
 4982  024536  000000                6$:   .WORD   0                   ;;POINTER GOES HERE
 4983  024540  104400  001160              TYPE    $QUES               ;;"?" "CR" &"LF"
 4984  024544  000720                      BR      1$                  ;;TRY AGAIN
 4985
 4986                                  ;;**********************************************************************
 4987
 4988                                  .SBTTL  TYPE ROUTINE
 4989
 4990                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 4991                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 4992                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 4993                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 4994                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 4995                                  ;*
 4996                                  ;*CALL:
 4997                                  ;*1) USING A TRAP INSTRUCTION
 4998                                  ;*      TYPE    ,MESADR             ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 4999                                  ;*OR
 5000                                  ;*      TYPE
 5001                                  ;*      MESADR
 5002                                  ;*
 5003                                  ;*2) USING A JSR INSTRUCTION
 5004                                  ;*      MOV     PS,-(SP)            ;PUSH PROCESSOR STATUS WORD ON THE STACK
 5005                                  ;*      JSR     PC,$TYPE            ;CALL TYPE ROUTINE
 5006                                  ;*      MESSADR                     ;FIRST ADDRESS OF MESSAGE
 5007
 5008  024546  105737  001155          $TYPE:  TSTB    $TPFLG              ;IS THERE A TERMINAL?
 5009  024552  100002                      BPL     1$                  ;BR IF YES
 5010  024554  000000                      HALT                        ;HALT HERE IF NO TERMINAL
 5011  024556  000407                      BR      3$                  ;LEAVE
 5012  024560  010046                1$:   MOV     R0,-(SP)            ;SAVE R0
 5013  024562  017600  000002          9$:   MOV     @2(SP),R0           ;GET ADDRESS OF ASCIZ STRING
 5014  024566  112046                2$:   MOVB    (R0)+,-(SP)         ;PUSH CHARACTER TO BE TYPED ONTO STACK
 5015  024570  001005                      BNE     4$                  ;BR IF IT ISN'T THE TERMINATOR
 5016  024572  005726                      TST     (SP)+               ;IF TERMINATOR POP IT OFF THE STACK
 5017  024574  012600                      MOV     (SP)+,R0            ;RESTORE R0
 5018  024576  062716  000002          3$:   ADD     #2,(SP)             ;ADJUST RETURN PC
 5019  024602  000002                      RTI                         ;RETURN
 5020  024604  004737  024636          4$:   JSR     PC,7$               ;GO TYPE THIS CHARACTER
 5021  024610  123726  001154          5$:   CMPB    $FILLC,(SP)+        ;IS IT TIME FOR FILLER CHARS.?
 5022  024614  001364                      BNE     2$                  ;IF NO GO GET NEXT CHAR.
 5023  024616  013746  001152              MOV     $NULL,-(SP)         ;GET # OF FILLER CHARS. NEEDED
 5024                                                                  ;AND THE NULL CHAR.
 5025  024622  105366  000001          6$:   DECB    1(SP)               ;DOES A NULL NEED TO BE TYPED?
 5026  024626  002770                      BLT     5$                  ;BR IF NO--GO POP THE NULL OFF OF STACK
```

```
5027  024630  004737  024636              JSR     PC,7$                   ;GO TYPE A NULL
5028  024634  000772                      BR      6$                      ;LOOP
5029  024636  105777  154304      7$:     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
5030  024642  100375                      BPL     7$
5031  024644  116677  000002  154276      MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
5032  024652  005737  000164              TST     REMFF
5033  024656  001505                      BEQ     8$
5034  024660  032777  010000  154676 10$:  BIT     #DA,@ICAR
5035  024666  001401                      BEQ     .+4
5036  024670  104413                      CKRMTT
5037  024672  017746  154664              MOV     @ICSR,-(SP)             ;SAVE OLD ICSR
5038  024676  052777  001040  154656      BIS     #TBMTEN+TTYEN,@ICSR     ;SAVE OLD ICSR
5039  024704  016677  000004  154754      MOV     4(SP),@ICSLMT           ;SEND DATA TO ICR TTY
5040  024712  013737  001512  001606      MOV     FR110,TTYTMP
5041  024720  005037  001610              CLR     ERRLOP                  ;CLEAR ERROR COUNT
5042  024724  032777  010000  154630 13$:  BIT     #ERRBIT,@ICSR           ;ERROR BIT SET
5043  024732  001415                      BEQ     14$                     ;NO, THEN 14$
5044  024734  005237  001610              INC     ERRLOP                  ;LOG ERROR
5045  024740  005777  154620              TST     @ICAR                   ;CLEAR ERROR
5046  024744  000240                      NOP
5047  024746  000240                      NOP
5048  024750  022737  000012  001610      CMP     #10.,ERRLOP             ;TEN (10) CONSECUTIVE LINE ERRORS
5049  024756  001362                      BNE     13$                     ;NO, THEN CHECK LINE AGAIN
5050  024760  022626                      POP2SP                          ;ADJUST STACK
5051  024762  000137  025074              JMP     ERRLIN
5052  024766  005037  001610      14$:    CLR     ERRLOP                  ;CLEAR ERROR COUNT
5053  024772  032777  002000  154562      BIT     #PWRFL,@ICSR            ;IS PWR FAIL BIT SET
5054  025000  001403                      BEQ     15$                     ;NO, THEN 15$
5055  025002  022626                      POP2SP
5056  025004  000137  025120              JMP     RSTRT
5057  025010  032777  100000  154546 15$:  BIT     #XTBMT,@ICAR            ;TRANSMITTOR BUFFER EMPTY
5058  025016  001014                      BNE     12$
5059  025020  005337  001606              DEC     TTYTMP
5060  025024  001337                      BNE     13$
5061  025026  022626                      POP2SP
5062  025030  012600                      MOV     (SP)+,R0
5063  025032  022626                      POP2SP
5064  025034  005037  000164              CLR     REMFF
5065  025040  104400  031601              TYPE,   TBMTMS
5066  025044  000000                      HALT
5067  025046  000776                      BR      .-2
5068  025050  052777  000001  154504 12$:  BIS     #XRIF,@ICSR
5069  025056  005777  154604              TST     @ICSLMT
5070  025062  005777  154476              TST     @ICAR
5071  025066  012677  154470              MOV     (SP)+,@ICSR             ;SEND OLD CSR
5072  025072  000207              8$:     RTS     PC
5073
5074                                      ;;**************************************************************************
5075
5076  025074  032777  002000  154460 ERRLIN: BIT   #PWRFL,@ICSR           ;IS PWR FAIL BIT ALSO SET
5077  025102  001402                      BEQ     .+6                     ;NO, THEN IT MUST BE A BAD LINE
5078  025104  000137  025120              JMP     RSTRT                   ;RESTART
5079  025110  104400  031623              TYPE,   NOISY
5080  025114  000137  025146              JMP     RST4
5081
5082  025120  012706  001100      RSTRT:  MOV     #1100,SP
```

```
5083  025124  005737  000164            TST     REMFF
5084  025130  001402                    BEQ     .+6
5085  025132  005237  001674            INC     REMFF1
5086  025136  005037  000164            CLR     REMFF
5087  025142  104400  031661            TYPE,   PWRMES                  ;PRINT POWER FAIL SENSED
5088  025146  052777  040000  154406 RST4:  BIS   #MAINT3,@ICSR          ;ISSUE RESET
5089  025154  013700  001512            MOV     FR110,R0
5090  025160  005300                    DEC     R0
5091  025162  001376                    BNE     .-2
5092  025164  012700  000002        RST3:  MOV   #2.,R0                  ;SET UP DELAY
5093  025170  005001                    CLR     R1
5094  025172  032777  010000  154362 RST2:  BIT   #ERRBIT,@ICSR          ;ERROR BIT SET
5095  025200  001416                    BEQ     RST1                    ;NO, CONT
5096  025202  005777  154356            TST     @ICAR                   ;CLEAR ERROR
5097  025206  013702  001526            MOV     FR200,R2
5098  025212  005302                    DEC     R2
5099  025214  001376                    BNE     .-2
5100  025216  005301                    DEC     R1
5101  025220  001364                    BNE     RST2
5102  025222  005300                    DEC     R0
5103  025224  001362                    BNE     RST2
5104  025226  104400  031704            TYPE,   ICRDWN
5105  025232  000137  025164            JMP     RST3
5106  025236  052777  040000  154316 RST1:  BIS   #MAINT3,@ICSR
5107  025244  005737  001674            TST     REMFF1
5108  025250  001402                    BEQ     .+6
5109  025252  005237  000164            INC     REMFF
5110  025256  005037  001674            CLR     REMFF1
5111  025262  104400  031721        1$: TYPE,   RSTMES
5112  025266  000137  003652            JMP     START1
5113
5114
5115
5116
5117  025272                                .STRAP
5118  025272                        STARS
5119                                ;*****************************************************************
5120
5121                                .SBTTL  TRAP DECODER
5122
5123                                ;#THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
5124                                ;#AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5125                                ;#OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5126                                ;#GO TO THAT ROUTINE.
5127
5128  025272  010046            $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
5129  025274  016600  000002            MOV     2(SP),R0        ;;GET TRAP ADDRESS
5130  025300  005740                    TST     -(R0)           ;;BACKUP BY 2
5131  025302  111000                    MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
5132  025304  006300                    ASL     R0              ;;POSITION FOR INDEXING
5133  025306  016000  025314            MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
5134  025312  000200                    RTS     R0              ;;GO TO ROUTINE
5135
5136  025314                            SETTRAP TYPE,$TYPE,↑/TTY TYPEOUT ROUTINE/
5137  025314                            $$SET   TYPE,$TYPE,\<TRAP+$TRP>,\$TRP,<TTY TYPEOUT ROUTINE>
5138
```

```
                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
                                $TRPAD:
025314                                  $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
025314  024546                         SETTRAP TYPOC,$TYPOC,↑/TYPE OCTAL NUMBER (WITH LEADING ZEROS)/
025316                                  $$SET   TYPOC,$TYPOC,\<TRAP+$TRP>,\$TRP,<TYPE OCTAL NUMBER (WITH LEADING ZEROS)>
025316  023332                          $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
025320                                  SETTRAP TYPOS,$TYPOS,↑/TYPE OCTAL NUMBER (NO LEADING ZEROS)/
025320                                  $$SET   TYPOS,$TYPOS,\<TRAP+$TRP>,\$TRP,<TYPE OCTAL NUMBER (NO LEADING ZEROS)>
025320  023306                          $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
025322                                  SETTRAP TYPON,$TYPON,↑/TYPE OCTAL NUMBER (AS PER LAST CALL)/
025322                                  $$SET   TYPON,$TYPON,\<TRAP+$TRP>,\$TRP,<TYPE OCTAL NUMBER (AS PER LAST CALL)>
025322  023346                          $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
025324                                  SETTRAP RDCHR,$RDCHR,↑/TTY TYPEIN CHARACTER ROUTINE/
025324                                  $$SET   RDCHR,$RDCHR,\<TRAP+$TRP>,\$TRP,<TTY TYPEIN CHARACTER ROUTINE>
025324  024050                          $RDCHR  ;;CALL=RDCHR    TRAP+4(104404)  TTY TYPEIN CHARACTER ROUTINE
025326                                  SETTRAP RDLIN,$RDLIN,↑/TTY TYPEIN STRING ROUTINE/
025326                                  $$SET   RDLIN,$RDLIN,\<TRAP+$TRP>,\$TRP,<TTY TYPEIN STRING ROUTINE>
025326  024252                          $RDLIN  ;;CALL=RDLIN    TRAP+5(104405)  TTY TYPEIN STRING ROUTINE
025330                                  SETTRAP RDDEC,$RDDEC,↑/READ A DECIMAL NUMBER FROM TTY/
025330                                  $$SET   RDDEC,$RDDEC,\<TRAP+$TRP>,\$TRP,<READ A DECIMAL NUMBER FROM TTY>
025330  024370                          $RDDEC  ;;CALL=RDDEC    TRAP+6(104406)  READ A DECIMAL NUMBER FROM TTY
025332                                  SETTRAP $SAVREG,$SAVREG,↑/SAVE R0-R5 ROUTINE/
025332                                  $$SET   SAVREG,$SAVREG,\<TRAP+$TRP>,\$TRP,<SAVE R0-R5 ROUTINE>
025332  022116                          $SAVREG ;;CALL=SAVREG   TRAP+7(104407)  SAVE R0-R5 ROUTINE
025334                                  SETTRAP RESREG,$RESREG,↑/RESTORE R0-R5 ROUTINE/
025334                                  $$SET   RESREG,$RESREG,\<TRAP+$TRP>,\$TRP,<RESTORE R0-R5 ROUTINE>
025334  022154                          $RESREG ;;CALL=RESREG   TRAP+10(104410) RESTORE R0-R5 ROUTINE

025336                                  SETTRAP ICRDL1,DLYICX,<INTERRUPT ENABLED ICR DELAY ROUTINE>
025336                                  $$SET   ICRDL1,DLYICX,\<TRAP+$TRP>,\$TRP,<INTERRUPT ENABLED ICR DELAY ROUTINE>
025336  021456                          DLYICX  ;;CALL=ICRDL1   TRAP+11(104411) INTERRUPT ENABLED ICR DELAY ROUTINE
025340                                  SETTRAP DELAY0,RDELA0,<INTERRUPT ENABLED DELAY ROUTINE>
025340                                  $$SET   DELAY0,RDELA0,\<TRAP+$TRP>,\$TRP,<INTERRUPT ENABLED DELAY ROUTINE>
025340  021154                          RDELA0  ;;CALL=DELAY0   TRAP+12(104412) INTERRUPT ENABLED DELAY ROUTINE
025342                                  SETTRAP CKRMTT,CHECK,<REMOTE TTY CHECK NON INTERRUPT>
025342                                  $$SET   CKRMTT,CHECK,\<TRAP+$TRP>,\$TRP,<REMOTE TTY CHECK NON INTERRUPT>
025342  021332                          CHECK   ;;CALL=CKRMTT   TRAP+13(104413) REMOTE TTY CHECK NON INTERRUPT
025344                                  SETTRAP INOCT,INOCTR,<INPUT OCTAL ROUTINE>
025344                                  $$SET   INOCT,INOCTR,\<TRAP+$TRP>,\$TRP,<INPUT OCTAL ROUTINE>
025344  015532                          INOCTR  ;;CALL=INOCT    TRAP+14(104414) INPUT OCTAL ROUTINE
```

```
5185  025346
5186  025346
5187  025346  015100
5188  025350
5189  025350
5190  025350  015172
5191  025352
5192  025352
5193  025352  021772
5194  025354
5195  025354
5196  025354  020244
5197  025356
5198  025356
5199  025356  021216
5200  025360
5201  025360
5202  025360  021114
5203  025362
5204  025362
5205  025362  021642
5206  025364
5207  025364
5208  025364  015356
5209  025366
5210  025366
5211  025366  015264
5212  025370
5213  025370
5214  025370  015444
5215  025372
5216  025372
5217  025372  021256
5218  025374
5219  025374
5220  025374  005350
5221  025376
5222  025376
5223  025376  020436
5224  025400
5225  025400
5226  025400  021306
5227  025402
5228  025402
5229  025402  021434
5230
5231

SETTRAP INAR,RINA,<INPUT INPUT ADDR. ROUTINE>
$$SET   INAR,RINA,\<TRAP+$TRP>,\$TRP,<INPUT INPUT ADDR. ROUTINE>
RINA    ;;CALL=INAR     TRAP+15(104415) INPUT INPUT ADDR. ROUTINE
SETTRAP OUTAR,ROUTA,<INPUT OUTPUT ADDR. ROUTINE>
$$SET   OUTAR,ROUTA,\<TRAP+$TRP>,\$TRP,<INPUT OUTPUT ADDR. ROUTINE>
ROUTA   ;;CALL=OUTAR    TRAP+16(104416) INPUT OUTPUT ADDR. ROUTINE
SETTRAP PATAR,RPATA,<INPUT PATTERN ROUTINE>
$$SET   PATAR,RPATA,\<TRAP+$TRP>,\$TRP,<INPUT PATTERN ROUTINE>
RPATA   ;;CALL=PATAR    TRAP+17(104417) INPUT PATTERN ROUTINE
SETTRAP DELAR,RDELA,<INPUT DELAY TIME ROUTINE>
$$SET   DELAR,RDELA,\<TRAP+$TRP>,\$TRP,<INPUT DELAY TIME ROUTINE>
RDELA   ;;CALL=DELAR    TRAP+20(104420) INPUT DELAY TIME ROUTINE
SETTRAP DELAY2,RDELA2,<SECOUNDARY DELAY ROUTINE>
$$SET   DELAY2,RDELA2,\<TRAP+$TRP>,\$TRP,<SECOUNDARY DELAY ROUTINE>
RDELA2  ;;CALL=DELAY2   TRAP+21(104421) SECOUNDARY DELAY ROUTINE
SETTRAP DELAY,RDELAY,<ROUTINE TO DELAY XX MILLISEC>
$$SET   DELAY,RDELAY,\<TRAP+$TRP>,\$TRP,<ROUTINE TO DELAY XX MILLISEC>
RDELAY  ;;CALL=DELAY    TRAP+22(104422) ROUTINE TO DELAY XX MILLISEC
SETTRAP CPATR,RCPAT,<ROUTINE TO CHANGE PATTERNS>
$$SET   CPATR,RCPAT,\<TRAP+$TRP>,\$TRP,<ROUTINE TO CHANGE PATTERNS>
RCPAT   ;;CALL=CPATR    TRAP+23(104423) ROUTINE TO CHANGE PATTERNS
SETTRAP IDAC,RDACA,<ROUTINE TO INPUT DAC ADDR>
$$SET   IDAC,RDACA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO INPUT DAC ADDR>
RDACA   ;;CALL=IDAC     TRAP+24(104424) ROUTINE TO INPUT DAC ADDR
SETTRAP CNTAR,RCNTA,<ROUTINE TO INPUT COUNTER MODULE ADDR>
$$SET   CNTAR,RCNTA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO INPUT COUNTER MODULE ADDR>
RCNTA   ;;CALL=CNTAR    TRAP+25(104425) ROUTINE TO INPUT COUNTER MODULE ADDR
SETTRAP ADAR,RADA,<ROUTINE TO INPUT ADDS ADDR>
$$SET   ADAR,RADA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO INPUT ADDS ADDR>
RADA    ;;CALL=ADAR     TRAP+26(104426) ROUTINE TO INPUT ADDS ADDR
SETTRAP INITR,RINIT,<ROUTINE TO ISSUE SYSTEM INITIALIZE>
$$SET   INITR,RINIT,\<TRAP+$TRP>,\$TRP,<ROUTINE TO ISSUE SYSTEM INITIALIZE>
RINIT   ;;CALL=INITR    TRAP+27(104427) ROUTINE TO ISSUE SYSTEM INITIALIZE
SETTRAP FOCTA,ROCTA,<ROUTINE TO CONVERT OCTAL NUMBERS TO ASCIZ>
$$SET   FOCTA,ROCTA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO CONVERT OCTAL NUMBERS TO ASCI
ROCTA   ;;CALL=FOCTA    TRAP+30(104430) ROUTINE TO CONVERT OCTAL NUMBERS TO ASCI
SETTRAP QUBR,RQUBR,<ROUTINE TO SET UNI OR BI-POLAR>
$$SET   QUBR,RQUBR,\<TRAP+$TRP>,\$TRP,<ROUTINE TO SET UNI OR BI-POLAR>
RQUBR   ;;CALL=QUBR     TRAP+31(104431) ROUTINE TO SET UNI OR BI-POLAR
SETTRAP WTBSY,BUSY,<ICR WAIT FOR LINE INACTIVE>
$$SET   WTBSY,BUSY,\<TRAP+$TRP>,\$TRP,<ICR WAIT FOR LINE INACTIVE>
BUSY    ;;CALL=WTBSY    TRAP+32(104432) ICR WAIT FOR LINE INACTIVE
SETTRAP ICRDLY,DLYICR,<3.2 MSEC FOR ICR ROUND TRIP>
$$SET   ICRDLY,DLYICR,\<TRAP+$TRP>,\$TRP,<3.2 MSEC FOR ICR ROUND TRIP>
DLYICR  ;;CALL=ICRDLY   TRAP+33(104433) 3.2 MSEC FOR ICR ROUND TRIP

                        ;*
                        ;*ACSII MESSAGE SECTION
                        ;*
025404  005015  041511  030522  MHEAD:  .ASCIZ  <15><12>:ICR11 FIELD TEST PROGRAM:
025437     040  020055  040515  MHEAD1: .ASCIZ  ? - MAINDEC-11-DZIRB-A?<15><12>
025467     015  033012  032056  MIPA:   .ASCIZ  <15><12>/6.4 INPUT /
025504  005015  027066  020065  MOPA:   .ASCIZ  <15><12>/6.5 OUTPUT /
025522  005015  040440  042104  MTOH:   .ASCIZ  <15><12>/ ADDR    DATA    GENERIC CODE/
025561     015  047012  020117  MNDA:   .ASCIZ  <15><12>/NO DAC ADDR. ENTERED/
```

```
025610  006415  042412  042116  MEND:   .ASCIZ  <15><15><12>/END PASS/
025624  005015  027066  020066  MCNT:   .ASCIZ  <15><12>/6.6 CNTR ADDR /
025645     136     105          MEXEN:  .ASCII  /↑E/
025647     015  042412  050130          .ASCIZ  <15><12>/EXPERT MODE ENABLED/
025675     136     116          MNOEN:  .ASCII  /↑N/
025677     015  042412  050130          .ASCIZ  <15><12>/EXPERT MODE DISABLED/
025726  046136                  MLEN:   .ASCII  /↑L/
025730  005015  027070  020064          .ASCII  <15><12>/8.4 LINE PRINTER MODE/
025757     015  046412  045501          .ASCIZ  <15><12>/MAKE PRINTER READY/
026004  045136                  MAC:    .ASCII  /↑J/
026006  005015  027511  020117          .ASCIZ  <15><12>:I/O ASSUMED CONNECTED:
026036  042136     000          MDEL1:  .ASCIZ  /↑D/
026041     015  051412  041505  MDEL2:  .ASCIZ  <15><12>/SEC DELAY TIME(MSEC)?/
026071     015  053412  051117  MWK:    .ASCIZ  <15><12>/WORKING.../
026106  005015  027066  020067  MDAC:   .ASCIZ  <15><12>/6.7 DAC  /
026122  005015  027066  020070  MADU:   .ASCIZ  <15><12>/6.8 ADDS /
026136  005015  047524  020117  MABOV:  .ASCIZ  <15><12>/TOO MANY NUMBERS-RETYPE- /
026172  006415  033012  031456  MTN:    .ASCIZ  <15><15><12>/6.3 TEST NUMBER/
026215     015  047012  020117  MTNL:   .ASCIZ  <15><12>/NO SUCH TEST/
026234  005015  027066  030461  MIOO:   .ASCIZ  <15><12>/6.11 INPUT OR OUTPUT(I OR O)/
026273     015  033012  034456  MDEL:   .ASCIZ  <15><12>/6.9 DELAY (IN MILLISEC) /
026326  005015  052516  041115  MTTL:   .ASCIZ  <15><12>/NUMBER TOO BIG!/
026350  005015  027066  020061  MPPM:   .ASCIZ  <15><12>/6.10 PAT MOD, PAT /

026375     015  020012  020077  MQMARK: .ASCIZ  <15><12>/ ? /
026403     077  000040          MQ:     .ASCIZ  /? /

026406  005015  047531  020125  MSTERR: .ASCII  <15><12>/YOU MUST INITIALLY START PROGRAM AT ADDR. 200/
026465     015  040412  052106          .ASCIZ  <15><12>/AFTER RESTART AT ADDR. 210 IS ALLOWED/
026535     015  044412  053116  MNRFN:  .ASCIZ  <15><12>/INVALID ADDRESS: /

026561     116  020117  040504  EM1:    .ASCIZ  /NO DATA XFER/
026576                          EM27:
026576  043130  051105  042040  EM2:    .ASCIZ  /XFER DATA ERROR/
026616  047103  052126  047040  EM3:    .ASCIZ  /CNVT NOT SET/
026633     103  053116  020124  EM4:    .ASCIZ  /CNVT NOT CLEAR/
026652  040504  040524  051040  EM5:    .ASCIZ  /DATA REG ERROR/
026671     116  020117  030101  EM6:    .ASCIZ  /NO ADDS INTR/
026706                          EM14:
026706  041511  051101  041040  EM7:    .ASCIZ  /ICAR BAD ON INT/
026726  044522  020106  051105  EM10:   .ASCIZ  /RIF ERROR ON ADDS/
026750  005015  047516  041440  MNAE:   .ASCIZ  <15><12>/NO COUNTER MODULE ADDRS.ENTERED/
027012  047103  051124  054040  EM11:   .ASCIZ  /CNTR XFER ERROR/
027032  047103  051124  052455  EM12:   .ASCIZ  /CNTR-UP COUNT BAD/
027054  047103  051124  043040  EM13:   .ASCIZ  /CNTR FAILED TO INT/
027077     103  052116  020122  EM15:   .ASCIZ  /CNTR CONT ON OVFLW/
027122  044522  020106  051105  EM16:   .ASCIZ  /RIF ERROR ON CNTR/
027144  054523  027123  044440  EM17:   .ASCIZ  /SYS. INIT. CNTR/
027164  046111  020114  041511  EM20:   .ASCIZ  /ILL ICR INTR/

027201     116  020117  041511  EM21:   .ASCIZ  /NO ICR INTERRUPT/
027222  041511  051101  042440  EM22:   .ASCIZ  /ICAR ERROR/

027235     015  046012  047111  ERRORH: .ASCIZ  <15><12>/LINE ERROR COUNT = /
027263     123  051531  020056  EM23:   .ASCIZ  /SYS. INIT. CNTR INT/
027307     103  052116  020122  EM24:   .ASCIZ  /CNTR CNT SYS.INIT/
```

```
027331    122  040505  020104  EM25:    .ASCIZ  /READ DUAL ADDR./
027351    104  040525  020114  EM26:    .ASCIZ  /DUAL ADDR. ERROR/
027372 005015  027066  031061  MQURB:   .ASCIZ  <15><12>/6.12 UNI OR BI-POLAR(U OR B)/
027431    015  047012  020117  MNAD:    .ASCIZ  <15><12>?NO A/D ADDR. ENTERED?
027460 005015  041536  020040  MSWO:    .ASCIZ  <15><12>/↑C  -  RETURNS TO MONITOR/
027514 030455  027060  030060  MMOVFL:  .ASCIZ  /-10.00 (OVFLW)/
027533    053  030061  030056  MPOVFL:  .ASCIZ  /+10.00 (OVFLW)/
027552 020040  041520  036440  PCPRT:   .ASCIZ  /  PC = /
027562 005015  042522  042520  MREP:    .ASCIZ  <15><12>/REPEAT/

027573    040  042522  042101  MCALOT:  .ASCIZ  / READING/

027604 053040  000040          MCALT1:  .ASCIZ  / V /

027610 000055                   MMINUS:  .ASCIZ  /-/
027612 000053                   MPLUS:   .ASCIZ  /+/
027614 005015  051536           MS:      .ASCII  <15><12>/↑S/
027620 005015  053523  020122            .ASCIZ  <15><12>/SWR /
027627    040  047040  053505  NEWSWR:  .ASCIZ  /  NEW SWR /

027642 005015  050103  020125  PWRMSG:  .ASCIZ  <15><12>/CPU POWER FAILURE/<15><12><12>

027671    015  052012  051505  MHT0:    .ASCIZ  <15><12>?TEST 0 - I/O MODULE EXERCISER?<15><12>

027733    015  052012  051505  MHT1:    .ASCIZ  <15><12>?TEST 1 - I/O MODULE EXERCISER?<15><12>

027775    015  052012  051505  MHT2:    .ASCIZ  <15><12>/TEST 2 - DAC CALIBRATION/<15><12>

030032 005015  042524  052123  MHT3:    .ASCIZ  <15><12>/TEST 3 - DAC INTERACTION/<15><12>

030067    015  052012  051505  MHT4:    .ASCIZ  <15><12>/TEST 4 - COUNTER MODULE TEST/<15><12>

030130 005015  042524  052123  MHT5:    .ASCIZ  <15><12>?TEST 5 - A/D LOGIC TEST?<15><12>

030164 005015  042524  052123  MHT6:    .ASCIZ  <15><12>?TEST 6 - A/D CALIBRATION?<15><12>

030221    015  052012  051505  MHT10:   .ASCIZ  <15><12>?TEST 10 - LINE TEST?<15><12>

030251    015  052012  051505  MHT8:    .ASCIZ  <15><12>?TEST 7 - A/D REPEATABILITY?<15><12>
030310 005015  027070  020064  MLPAV:   .ASCIZ  <15><12>/8.4 PRINTER AVAILABLE/<15><12>
030342 005015  047111  040526  MIVP:    .ASCIZ  <15><12>/INVALID PATTERN MODIFIER/<15><12>

030377    015  033012  030456  MGAIN:   .ASCIZ  <15><12>/6.13 GAIN /

030414 005015  027066  032061  MCHAN:   .ASCIZ  <15><12>/6.14 CHANS (SC,EC)/
030441    015  033012  030456  MTOL:    .ASCIZ  <15><12>/6.16 TOLERANCE /

030463    015  042412  051122  MCHER1:  .ASCIZ  <15><12>/ERROR! START CHAN > END CHAN./
030523    015  047012  054105  NFILE:   .ASCIZ  <15><12>/NEXT FILE BOX/
030543    015  047012  054105  NVECT:   .ASCIZ  <15><12>/NEXT VECTOR ADDRESS/
030571    015  052012  040522  XFERMS:  .ASCIZ  <15><12>/TRANSFERRING CONTROL/
030620 005015  047516  020124  MINNN:   .ASCIZ  <15><12>/NOT OCTAL DIGIT--RETYPE IT/<15><12>
030657    015  047012  052117  MINKN:   .ASCIZ  <15><12>/NOT UNDERSTOOD--RETYPE IT/<15><12>
```

```
030715   015  042412  051122  MCHANH: .ASCIZ  <15><12>/ERROR! NO SUCH CHAN./

030744 005015  027066  032461  MAVEQ:  .ASCIZ  <15><12>/6.15 EXPECTED AVERAGE /
030775   015  047012  020117  MNSG:   .ASCIZ  <15><12>/NO SUCH GAIN!/

031015   015  047012  046525  MNTL:   .ASCIZ  <15><12>/NUMBER TO LARGE-MAX=7777/
031050 000040                 M2SP:   .ASCIZ  / /
031052 005015  042522  042520  MREPFT: .ASCIZ  <15><12>/REPEATIBILITY FORCED TYPOUT/

031110 005015  042522  042520  MREPER: .ASCIZ  <15><12>/REPEATABILITY ERROR/

031136 005015  044103  047101  MREPT1: .ASCIZ  <15><12>/CHAN,GAIN  /
031154 005015  047514  040454  MREPT4: .ASCIZ  <15><12>/LO,AVR,HGH  /
031173   015  020012  047514  MREPT2: .ASCII  <15><12>/ LO  -5  -4  -3  -2  -1  AV/
031230 025440  020061  025440          .ASCIZ  / +1  +2  +3  +4  +5  HI/
031261   015  046012  020117  MREPT3: .ASCIZ  <15><12>/LO  -2  -1  AV  +1  +2  HI/

031316 005015  027066  020061  MFILE:  .ASCIZ  <15><12>/6.1 FILE BOX TO BE TESTED/
031352 005015  027066  020062  MVECT:  .ASCIZ  <15><12>/6.2 ICR VECTOR ADDRESS/
031403   015  044412  046114  ILLEG:  .ASCIZ  <15><12>/ILLEGAL NUMBER/<15><12>
031426 005015  042101  051104  NRANG1: .ASCIZ  <15><12>/ADDRESS /
031441   040  047516  020124  NRANG2: .ASCIZ  / NOT IN FILE/<15><12>
031460 005015  044506  042514  NOINT:  .ASCIZ  <15><12>/FILE BOX DID NOT INTERRUPT-FATAL/
031523   015  043012  046111  FILINT: .ASCIZ  <15><12>/FILE BOX INTERRUPTED AT /
031556 026440  020055  044103  CKJMP:  .ASCIZ  / -- CHECK JUMPERS /
031601   015  051012  046505  TBMTMS: .ASCIZ  <15><12>/REMOTE TTY HUNG/
031623   015  047012  044517  NOISY:  .ASCIZ  <15><12>/NOISY LINE/
031640 005015  047507  047111  REMOTE: .ASCIZ  <15><12>/GOING REMOTE/<15><12>
031661   015  044412  051103  PWRMES: .ASCIZ  <15><12>/ICR POWER LOSS/<15><12>
031704 005015  041511  020122  ICRDWN: .ASCIZ  <15><12>/ICR DOWN/<15><12>
031721   015  005015  042522  RSTMES: .ASCIZ  <15><15><12>/RESTARTING FROM ICR POWER LOSS/<15><12>
031765   015  047012  047117  NONXST: .ASCIZ  <15><12>/NON-EXISTENT FILE BOX/
         032016                          .EVEN

5232  032016   000     015          OUTBFX: .BYTE   0,15
5233  032020   015     012          OUTBF1: .BYTE   15,12
5234  032022 000000                OUTBF:  0
5235         032054                       .=.+30
5236  032054                       BUFFER:
5237                               ;*END ADDRESS IS FUNCTION OF HOW MANY CHANS. ARE BEING EXERCISED
5238                               ;*AT ONE TIME IN TEST 8.
5239         032054                $ENDAD=.
5240         000001                       .END
```

```
AADMS   023106          4629*   4652    4658
ADADR   015032          2537    2754    2951    3427*   3547    3552    3556*   3557
ADAR  = 104426          2528    2741    2949    5214*
ADBSY   001602          1382*   1767*   2623*   2642*   2920*   2929*   3121*   3133*   3932
ADLOG   010236          2537*   2954
ADLOGE  011202          2529    2717*
ADLOGL  010206          2529*   2720
ADLOG2  010324          2554*   2568
ADLOG3  010376          2572*
ADLOG4  010500          2586    2599*
ADLOG5  010502          2603*
ADLOG6  010530          2616*
ADLOG7  010660          2617    2640*
ADLOG8  010760          2653    2662*
ADLOG9  011020          2636    2679*
ADLO3A  010454          2580    2588*
ADL6L   010616          2629*   2658    2668    2673
ADPATE  011240          2567    2731*
ADPATP  011216          2541    2722*
ADWD    014574          3115*   3116*   3117*   3118*   3124    3337*
AVERR   013546          3071    3165*
AVERRL  013560          3167*   3202
AVERRN  013764          3199*   3316
AVEXP   014616          3024    3029    3031    3038    3042*   3189    3346*
AVTKN   014620          3170*   3184*   3214*   3218*   3219*   3222*   3245    3266    3347*
AVTYP   014530          3282    3298    3312    3317*
A1      012000          2805    2807    2809    2833*   2835    2837    2839    2845    2849    2852    2863    2868*
BDDT    023153          4637*   4653
BIPOL   015034          2787    2811    2815    3040    3143    3181    3220    3428*   4062*   4064*
BIT0  = 000001          1090*   1448
BIT00 = 000001          1080*   1090
BIT01 = 000002          1079*   1089
BIT02 = 000004          1078*   1088
BIT03 = 000010          1077*   1087
BIT04 = 000020          1076*   1086
BIT05 = 000040          1075*   1085
BIT06 = 000100          1074*   1084
BIT07 = 000200          1073*   1083
BIT08 = 000400          1072*   1082
BIT09 = 001000          1071*   1081
BIT1  = 000002          1089*   1447
BIT10 = 002000          1070*   1438
BIT11 = 004000          1069*   1437    4824
BIT12 = 010000          1068*   1436    1453
BIT13 = 020000          1067*   1435
BIT14 = 040000          1066*   1434    4806
BIT15 = 100000          1065*   1433    1452
BIT2  = 000004          1088*   1446
BIT3  = 000010          1087*   1445
BIT4  = 000020          1086*   1444
BIT5  = 000040          1085*   1443
BIT6  = 000100          1084*   1442
BIT7  = 000200          1083*   1441
BIT8  = 000400          1082*   1440
BIT9  = 001000          1081*   1439
BMTEN = 000010          1445*
```

```
BPTVEC= 000014  1097#
BUFFER  032054  3093  3108  3169  3274  3291  3305  5236#
BUSY    021306  4171#  4174  5226
CALIT   011430  2772  2787#
CALITE  011604  2829#  2865
CALITP  011570  2818  2824#
CALIT1  011574  2816  2823  2826#  2830
CALIT2  011710  2827  2851#
CALIT3  011720  2853#  2864
CALIT4  011746  2856  2859#
CALIT5  011764  2860  2863#
CALTP   011356  2764  2770#
CHAN    014570  3081*  3082*  3083  3087*  3094*  3096  3103*  3104  3165*  3200*  3201  3227  3335#
CHANF   014600  2984*  3060*  3061*  3062  3064*  3067  3069*  3081  3104  3201  3339#
CHANFR  014604  2986*  3006  3009*  3010  3015  3065  3067  3069  3341#
CHANNO  014606  2994*  3064  3342#
CHANS   014576  2983*  3062*  3063*  3065  3082  3094  3165  3338#
CHANSR  014602  2985*  2999  3004  3009  3010  3060  3340#
CHAN1   014566  3096*  3097*  3098*  3099*  3100*  3117  3166*  3168  3199*  3273  3290  3304  3334#
CHAN7   014560  3331#
CHAR    017140  2079*  2081*  2116*  2121  2123  3463  3487  3512  3534  3554  3605*  3606*  3607
                3617  3622*  3623*  3624  3629  3630  3634  3643  3646  3649  3654  3657  3660
                3663  3666  3669  3672  3674  3677  3681  3690*  3691  3700*  3738  3740  3777#
                4019*  4021*  4057*  4060
CHARC   017146  3587*  3688*  3748  3780#
CHECK   021332  4181#  4237  5181
CKADR   020766  3467  3491  3515  3538  3558  4074#
CKJMP   031556  1604  5231#
CKRMTT= 104413  1921  4620  5036  5181#
CMAD    023220  4645#  4657
CNTADR  014770  1854  2248  3408#  3504  3509  3511
CNTAR = 104425  2244  5211#
CNTPAE  010176  2336  2519#
CNTPAT  010132  2315  2501#
CONNT   001640  1397#  1493*  1831  1857  1880  1965  3949*
CONNTR  020226  3659  3949#
CONTSR  020100  3917#  3944
CONVED  012202  2918  2925#
CONVER  012130  2759  2913#
CONVT   013326  3101  3115#
CORSIZ  001634  1395#  1649*  1651*  2987
CPATR = 104423  1994  5205#
CTLLOC  001664  1407#  1528*  1559*  1619*  3651  3908*
DA    = 010000  1453#  1919  2090  3596  4030  4226  4865  5034
DACADR  015030  2159  2169*  2184  2203*  2256  3425#  3527  3532  3536*  3537
DACLST  006602  2192  2213#
DAFLG   001604  1383#  1766*  2643  2930  3134  3924*  3934*
DATCNT  023102  4597*  4598  4607*  4627#
DDISP = 177570  1012#  1134
DELAR = 104420  1817  2142  5196#
DELAY = 104422  1736  1738  1842  2149  2437  2463  2492  2589  2626  2666  2774  2785  2953
                3095  4204  5202#
DELAY0= 104412  1864  4212  5178#
DELAY2= 104421  1844  5199#
DFO     001474  1172  1179  1186  1193  1199  1206  1213  1220  1227  1234  1241  1248  1255
                1262  1269  1276  1283  1290  1297  1304  1311  1318  1325  1333  1338  1341
```

```
                        1344#
DH1     023250          1169    1184    1191    1197    1204    1218    4652#
DH2     023254          1176    1232    1239    1260    1274    1309    1331    4653#
DH3     023264          1211    1253    1295    4655#
DH4     023274          1225    1246    1267    1281    1288    1302    4657#
DH5     023300          1316    1323    4658#
DISPLA  001140          1134#   1486*   2135*   2767*   4529*   4835*
DISPRE  000174           976#   1486
DLYICR  021434          4202#   5229
DLYICX  021456          4210#   5175
DSWR  = 177570          1011#   1133
DT1     001476          1171    1185    1192    1198    1205    1219    1346#
DT2     001500          1178    1212    1348#
DT3     001476          1226    1247    1268    1282    1289    1303    1345#
DT4     001500          1233    1240    1254    1261    1275    1296    1310    1332    1340    1347#
DT5     001506          1317    1324    1349#
DUAD    023234          4647#   4658
EMTVEC= 000030          1100#   1471*   1472*
EM1     026561          1168    5231#
EM10    026726          1217    5231#
EM11    027012          1224    5231#
EM12    027032          1238    5231#
EM13    027054          1245    5231#
EM14    026706          1252    5231#
EM15    027077          1259    5231#
EM16    027122          1266    5231#
EM17    027144          1273    5231#
EM2     026576          1175    1231    5231#
EM20    027164          1280    5231#
EM21    027201          1287    5231#
EM22    027222          1294    5231#
EM23    027263          1301    5231#
EM24    027307          1308    5231#
EM25    027331          1315    5231#
EM26    027351          1322    5231#
EM27    026576          1330    5231#
EM3     026616          1183    5231#
EM4     026633          1190    5231#
EM5     026652          1196    5231#
EM6     026671          1203    5231#
EM7     026706          1210    5231#
ERAV1   014010          3192    3196    3207#
ERAV2   014016          3198    3210#
ERRBIT= 010000          1436#   4224    5042    5094
ERRCNT  001676          1412#   1796    1798    1801*   4238*
ERRDLY  023050          3223    3236    3257    3264    3313    4592    4608    4616#
ERREN = 000002          1447#   1632    1763    1846    2120    2145    2433    2535    2618    2631    2632    2634    2635
                        2917    2936    2937    3119    3140    3141    3718    3909    3927    3928    4065    4164    4232
                        4233    4240    4241
ERRLIN  025074          5051    5076#
ERRLOP  001610          1385#   5041*   5044*   5048    5052*
ERRORH  027235          1795    5231#
ERRTMP  023076          4625#
ERRTM1  023100          4619*   4621*   4626#
ERRVEC= 000004          1093#   4811    4812*   4814*   4817*
EXIC    023166          4639#   4655
```

```
EXIT  = 000002          1425#  1803   1915   1936   1941   1967   1986   2015   3470   3494   3518   3539   3559
                        3720   3775   3979   4066   4097   4109   4123   4138   4152   4165   4176   4206   4214
                        4263   4267   4269   4288   4319
EXPERT  001636          1396#  1695*  2072   2955   2979   3020   3043   3452   3476   3500   3523   3543   3783*
                        3788*  3960   4012   4296
EXSET   017154          3665   3783#
FILHLT  003012          1620#
FILINT  031523          1600   3877   5231#
FILVEC  002260          1501   1513#
FOCTA = 104430          1904   1982   5220#
FREQ    001534          1363#  1711*  1731   1732*  2245*  2588*  2622*  2773*  2952*  3080*  3086*  3089   3091*
                        3965*  3972*  3973*  3995*  4116   4131   4202   4203*  4205*  4210   4211*  4213*
FREQ1   001536          1364#  1710*  1733*  3964*  3971*  3974*  3975   4118   4133
FREQ2   001540          1365#  1709*  3994*  4144
FREQ3   001542          1366#  1708*  3996*  3997   4146
FR1     001544          1367#  1653*  1658*  1659*  1664*  1670   1673   1679   1691   3973   3995
FR110   001512          1354#  1685*  1686*  1687*  1688*  1732   2773   2952   5040   5089
FR1120  001546          1368#  1653   1658   1659
FR16    001522          1358#  1675*  1676*  1677*  1678   1679*  2588   2622
FR20    001520          1357#  1678*  1680*  1681   1689
FR200   001526          1360#  5097
FR3     001532          1362#  1670*  1671*  1672   1673*  1674   1694   1711   3965
FR32    001530          1361#  1691*  1692*  1693*  1694*  4203   4211
FR40    001516          1356#  1689*  1690*  2245
FR5     001524          1359#  1672*  1674*  1675   1677   1680   1683   1684   1687   1688   3086
FR50    001514          1355#  1681*  1682*  1683*  1684*  1685   3080
GAIN    014572          2959   2964   2975   2977*  3115   3231   3336#
GDDT    023137          4534#  4653
GLIST   014646          2967   3363#
GNS   = ******  U        975   5147   5150   5153   5156   5159   5162   5165   5168   5171   5175   5178   5181
                        5184   5187   5190   5193   5196   5199   5202   5205   5208   5211   5214   5217   5220
                        5223   5226   5229
GOSUB = 004737          1427#  2996   3070   3071   3101   3282   3298   3312   3318
HEADER  001620          1389#  1820*  1887   1893*
HOLDIT  012022          2814   2870#
HOLDTM  011776          2831   2834*  2851*  2855   2857*  2862*  2867#
ICAR    001564          1375#  1504*  1548*  1549*  1587   1741   1919   1923   1948   1961   2090   2406   2448
                        2641   3596   3615   3904*  4030   4040   4226   4235   4239   4865   5034   5045   5057
                        5070   5096
ICARLD  000166           992#  1504
ICRDLY= 104433          2273   2290   2300   2326   2348   2350   2367   2370   2396   2436   2491   2545   2557
                        2590   2605   2625   2694   2703   2760   2777   3130   5229#
ICRDL1= 104411          1956   5175#
ICRDWN  031704          5104   5231#
ICRSRV  021500          1630   3906   4221#
ICRVEC  000162           990#  1510
ICRVT   001612          1386#  1631*  1764*  1822*  2393*  2428*  2453*  2531*  2617*  2681*  2918*  2935*  3120*
                        3139*  3907*  4228   4236
ICSHGH  001662          1406#  1509*  1550*  1551*  3903*  4086
ICSLMT  001666          1408#  1506*  1540*  1546*  1550   1839   1929*  2079   2096   2099   2101*  2683   2685
                        3572   3601   3605   3617*  3719   3902*  4019   4035   4039   4042*  4084   4182   4231
                        4855   4870   4874   5039*  5069
ICSMLD  000172           994#  1506   1507
ICSMOD  001560          1373#  1951   3466   3490   3514   3536   3556
ICSR    001562          1374#  1505*  1541*  1545*  1548   1553   1588*  1606*  1612*  1632*  1726*  1740*  1763*
                        1838*  1846*  1922*  1925*  1927*  1930*  1955*  1960*  2078*  2080*  2084   2087   2095*
```

		2098*	2102*	2120*	2145*	2347*	2369*	2371*	2386	2394*	2398*	2408*	2411*	2412*
		2433*	2440*	2452*	2471*	2474	2479*	2494	2535*	2618*	2619*	2630*	2631*	2632*
		2634*	2635*	2662*	2917*	2927*	2936*	2937*	3119*	3128*	3140*	3141*	3571*	3573*
		3590	3593	3600*	3604*	3610	3612	3618*	3718*	3898*	3901*	3909*	3927*	3928*
		4018*	4020*	4024	4027	4034*	4038*	4043*	4065*	4161*	4164*	4171	4173	4181*
		4183*	4221	4224	4230*	4232*	4233*	4240*	4241*	4854*	4856*	4859	4862	4869*
		4873*	4875*	4881*	5037	5038*	5042	5053	5068*	5071*	5076	5088*	5094	5106*
ICSRLD	000170	993#	1505											
ICSVT	001566	1376#	1574*	1584*	1630*	2664*	3900*	3905*	3906*					
ICSVT2	001570	1377#	1576*	1585*	1824*	2429*	2616*	3899*	3900					
IDAC =	104424	2158	2183	5208#										
ILLEG	031403	1538	1571	3815	3854	5231#								
INADR	014670	1703	1845	1972	1978	2132	3374#	3456	3461	3465				
INAR =	104415	1814	2128	5187#										
INCFLG	001614	1387#	3578*	3693	3723	3725*	3735*	3744						
INITR =	104427	2136	2442	2460	2478	2493	2598	2711	5217#					
INOCT =	104414	1533	1564	1748	2963	3003	3028	3049	3460	3484	3508	3531	3551	3810
		3847	3919	4306	5184#									
INOCTR	015532	3567#	3756	3760	3764	3767	5184							
IOTVEC=	000020	1098#	1469*	1470*										
KEYSRV	004172	1627	1787#											
LINEPR	001644	1399#	1729*	3052	3799*									
LPAV	001642	1398#	1633*	1640*	3795									
LPCSR	001646	1400#	1634	3054										
LPDBR	001650	1401#	3055											
LPSET	017204	3656	3795#											
MABOV	026136	3755	5231#											
MAC	026004	3952	5231#											
MADMS	023121	4631#	4653	4655										
MADU	026122	3546	5231#											
MAINT0=	000400	1440#												
MAINT1=	004000	1437#												
MAINT2=	020000	1435#	1588	2394	3867									
MAINT3=	040000	1434#	1726	1740	2371	2398	3882	3887	3898	4161	5088	5106		
MAVEQ	030744	3023	5231#											
MCALOT	027573	2784	5231#											
MCALT1	027604	2776	5231#											
MCHAN	030414	2982	5231#											
MCHANH	030715	3018	5231#											
MCHER1	030463	3013	5231#											
MCNT	025624	3503	5231#											
MDAC	026106	3526	5231#											
MDEL	026273	3963	5231#											
MDEL1	026036	3988	5231#											
MDEL2	026041	3990	5231#											
MEND	025610	2259	2719	5231#										
MEXEN	025645	3785	5231#											
MFILE	031316	1529	5231#											
MGAIN	030377	2958	5231#											
MHEAD	025404	1511	1521	5231#										
MHEAD1	025437	1522	5231#											
MHT0	027671	1813	5231#											
MHT1	027733	2070	5231#											
MHT10	030221	5231#												
MHT2	027775	2157	5231#											
MHT3	030032	2182	5231#											

```
MHT4    030067          2243    5231#
MHT5    030130          2527    5231#
MHT6    030164          2740    5231#
MHT8    030251          2948    5231#
MINKN   030657          3766    5231#
MINNN   030620          3763    5231#
MIOO    026234          2075    5231#
MIPA    025467          3455    5231#
MIVP    030342          4312    5231#
MLEN    025726          3798    5231#
MLPAV   030310          1638    5231#
MMINUS  027610          2820    5231#
MMOVFL  027514          2798    5231#
MNAD    027431          2757    5231#
MNAE    026750          2252    5231#
MNDA    025561          2162    2187    5231#
MNOEN   025675          3790    5231#
MNRFN   026535          4090    5231#
MNSG    030775          2971    5231#
MNTL    031015          3034    5231#
MODEN = 000004          1446#   1588    1632    1763    1846    1960    2120    2145    2394    2412    2433    2535    2619
                        2631    2632    2634    2635    2917    2936    2937    3119    3140    3141    3718    3867    3909
                        3927    3928    4065    4164    4232    4233    4240    4241    4881
MODFF   001700          1413#   1769*   2089*   2092    2094*   3595*   3598    3602*   4029*   4032    4036*   4864*   4867
                        4871*
MODINT= 000200          1441#   2087    3593    4027    4862
MOPA    025504          3479    5231#
MPLUS   027612          2825    5231#
MPOVFL  027533          2792    2803    5231#
MPPM    026350          4299    5231#
MQ      026403          2077    3568    3967    4017    5231#
MQMARK  026375          3759    5231#
MQURB   027372          4015    5231#
MREP    027562          3059    5231#
MREPER  031110          3208    5231#
MREPFT  031052          3211    5231#
MREPT1  031136          3225    5231#
MREPT2  031173          3263    5231#
MREPT3  031261          3260    5231#
MREPT4  031154          3237    5231#
MS      027614          3937    5231#
MSTERR  026406          1719    5231#
MSWO    027460          2129    2751    5231#
MTN     026172          1743    5231#
MTNL    026215          1756    5231#
MTOL    030441          3046    5231#
MTTL    026326          3982    4003    5231#
MTOH    025522          1834    5231#
MVECT   031352          1560    5231#
MWK     026071          2130    2144    2165    2189    2246    2260    2530    2752    3056    3926    5231#
M2SP    031050          3230    3244    3251    3324    5231#
NA07    014562          3332#
NA17    014564          3333#
NEWSWR  027627          3943    5231#
NFILE   030523          3806    5231#
NIN     017142          3586*   3685*   3686*   3687*   3689*   3691*   3701    3722    3733    3750    3778#
```

```
NINC    017150  3585* 3695  3699* 3703  3707* 3729  3732* 3736* 3746  3751* 3781#
NINC1   017152  3576* 3695  3702  3703* 3704* 3713  3729  3746  3782#
NIN1    017144  3722* 3727* 3728* 3731  3733  3779#
NOINT   031460  1593  3871  5231#
NOISY   031623  5079  5231#
NONXST  031765  1556  3833  5231#
NOSET   017170  3668  3788#
NOTYET  001576  1380# 1490* 1502* 1725* 3569  3632  3716
NRANG1  031426  4102  5231#
NRANG2  031441  4106  5231#
NVECT   030543  3843  5231#
OCTMP   023104  4603* 4605  4628#
OUTADR  014730  1849  1876  1975  2148* 3391# 3480  3485  3489
OUTAR =  104416  1815  2141  5190#
OUTBF   032022  2001  5234#
OUTBFX  032016  5232#
OUTBF1  032020  1818  2014  5233#
OUTBSY= 100000  1433#
OUTS    015036  1706  1827  1848  1859  1979  3430# 3580  3584  3705
OUTSE   015076  1829  1861  3446# 3582
PACK    005446  2003  2005  2008  2020#
PACK1   005442  2002* 2004* 2006* 2017# 2021  2025* 2026  2034  2039* 2040  2050  2057* 2058*
                2059  2249* 2255  2264*
PACK2   005444  2018# 2020* 2023* 2024  2026* 2027* 2028* 2029* 2030* 2031* 2032* 2033  2034*
                2035* 2036* 2037* 2038  2040* 2041* 2042* 2043* 2044* 2045* 2046* 2047* 2048*
                20[illegible] 2050* 2051* 2052* 2053* 2054* 2055* 2056
PATCR =  104417  [illegible]2# 5193#
PATDY   001556  [illegible]2# 1863* 1885
PATRN   001552  1[illegible]0# 1698* 1853  1860  1863  4257* 4261* 4287* 4303*
PATRNC  001554  1371# 4261  4262* 4266* 4268*
PATRNM  001550  1369# 4254  4300  4302* 4307  4310  4315* 4316* 4317
PC     =%000007 1024# 2003* 2005* 2008* 2062* 2712* 2759* 2762* 2779* 2799* 2804* 2912* 2939*
                2954* 3223* 3236* 3257* 3264* 3313* 3467* 3491* 3515* 3538* 3558* 4089* 4286*
                4417* 4436* 4438* 4478* 4483* 4534* 4546* 4552* 4592* 4608* 4614* 4618* 4623*
                4780  5020* 5027* 5072*
PCPRT   027552  4587  5231#
PIRQ   = 177772  1010#
PIRQVE= 000240  1104#
POPSP2= 022626  1429# 1555  1598  1605  1611  1641  2407  2640  2670  3832  3876  3881  3886
POP2SP= 022626  1428# 1666  2925  3132  5050  5055  5061  5063
PR0    = 000000 1027#
PR1    = 000040 1028#
PR2    = 000100 1029#
PR3    = 000140 1030#
PR4    = 000200 1031#
PR5    = 000240 1032#
PR6    = 000300 1033#
PR7    = 000340 1034#
PS     = 177776 1007# 1008  1586* 1739* 1765* 1847* 1898* 1957* 1959* 2143* 2395* 2413* 2432*
                2438* 2454* 2539* 2624* 2665* 2682* 2921* 2938* 3122* 3142* 3863* 4130* 4147*
PSW    = 177776 1008# 2397* 3910*
PWFEN  = 000020 1444# 1632  1763  1846  1960  2120  2145  2412  2433  2535  2619  2631  2632
                2634  2635  2917  2936  2937  3119  3140  3141  3718  3909  3927  3928  4065
                4164  4232  4233  4240  4241  4881
PWRFL  = 002000 1438# 2084  3590  3612  4024  4173  4221  4859  5053  5076
PWRMES  031661  5087  5231#
```

```
PWRMSG  027642          4779    5231#
PWRVEC= 000024          1099#   1475*   1476*   4750*   4751*   4760*   4776*   4777*
QUBR  = 104431          2742    2950    5223#
RADA    015444          3543#   3550    5214
RADA1   015500          3547*   3548*   3552#
RASK    017130          3641    3773#   3786    3791    3800    3930    3953    4001
RCNTA   015264          3500#   3507    5211
RCNTA1  015320          3504*   3505*   3509#   3516
RCPAT   021642          4254#   4318*   5205
RCPATI  021654          4257#   4259    4264
RCPATL  021736          4271#   4309
RCPATR  021756          4255    4285#
RDACA   015356          3523#   3530    5208
RDAC1   015412          3527*   3528*   3532#
RDCHR = 104404          4897    5159#
RDDEC = 104406          3968    3991    5165#
RDELA   020244          3960#   3983    5196
RDELAY  021114          4116#   5202
RDELA0  021154          4130#   5178
RDELA2  021216          4144#   5199
RDELA3  020352          3662    3988#
RDELA4  020356          3989#   4004
RDLIN = 104405          4943    5162#
REIC    023203          4642#   4655
REMEND  001704          1415#   1730*
REMFF   000164           991#   1462*   1500    1713    1721*   1916    2082    2133    2765    3258    3267    3283    3588
                        3636    3638*   3652    4022    4547    4616    4857    5032    5064*   5083    5086*   5109*
REMFF1  001674          1411#   1727*   5085*   5107    5110*
REMHLT  022634          4534    4544#
REMOTE  031640          3640    5231#
REMSWR  001670          1409#   1503    3639    3918    3938
REPET7  012116          2762    2779    2908#
REPMAN  014632          3352#
RESREG= 104410          4482    5171#
RESVEC= 000010          1094#
RETURN= 000207          1426#
RHIGH   014624          3173*   3174    3176*   3191    3252    3349#
RINA    015100          3452#   3459    5187
RINA1   015134          3456*   3457*   3461#   3468
RINIT   021256          4159#   5217
RLOW    014622          3172*   3177    3179*   3195    3238    3348#
ROCTA   005350          2001#   522C
ROUTA   015172          3476#   3483    5190
ROUTA1  015226          3480*   3481*   3485#   3492
RPATA   021772          4296#   4305    4313    5193
RQUBR   020436          4012#   5223
RSTART  003610           997    1715#
RSTMES  031721          5111    5231#
RSTRT   025120          2086    3592    3614    4026    4175    4223    4861    5056    5078    5082#
RST1    025236          5095    5106#
RST2    025172          5094#   5101    5103
RST3    025164          5092#   5105
RST4    025146          5080    5088#
RTEMP   014626          1731*   1758*   1759*   1760*   1761*   1770    2099*   2101    2106*   2109    2110*   2111    2113
                        2116    2284*   2287    2307*   2315*   2320    2322    2328    2335*   2336    2685*   2686*   2688
                        2987*   2988*   2991*   3171*   3185*   3213*   3217*   3266*   3269*   3271*   3276    3292    3299*
```

```
                    3306    3350#   3579*   3684*   3697    3702*   3710*   3969*   3977*   3992*   3999*   4039*   4042
                    4047*   4050    4051*   4052    4055    4057    4144*   4148*   4309*   4317*   4318
RTEMP1  014630      2321*   2324*   2989*   2990*   2993*   2994    2995    3180*   3183*   3184    3189*   3190*   3191
                    3193*   3194*   3195    3351#
RTEMP2  021152      4118*   4121*   4125#   4133*   4136*   4146*   4150*
RTEMP3  021150      4116*   4119*   4124#   4131*   4134*
R0      =%000000    1015#   1507*   1508*   1509    1513*   1515*   1516*   1518    1553*   1589*   1590*   1620*   1622*
                    1623*   1625    1645*   1649    1703*   1706    1796*   1827*   1829    1849*   1854    1971    1972*
                    1975    1978*   1979*   1981*   2132*   2135    2190*   2206*   2248*   2249    2255*   2256    2264
                    2574*   2581*   2767    2781    2789    2817    2967*   3102    3131*   3145*   3146*   3147*   3148*
                    3150*   3151*   3152*   3153*   3465*   3468    3489*   3492    3511*   3516    3537*   3557*   3580*
                    3582    3584*   3705*   3737*   3742*   3830*   3836*   3838*   3839*   3841    3865*   3868*   3869*
                    3890*   3892*   3893*   3895    3938*   3940    4084    4086    4160*   4342    4368*   4385    4389*
                    4393*   4398*   4402*   4408*   4410    4416*   4458*   4459    4477*   4481*   4573    4576*   4577*
                    4578*   4579*   4580*   4581*   4582*   4583    4594    4596    4613*   4753    4775*   4940    4944*
                    4945    4948    4950    4951    4976*   4980*   5012    5013*   5014    5017*   5062*   5089*   5090*
                    5092*   5102*   5128    5129*   5130    5131*   5132*   5133*   5134*
R1      =%000001    1016#   1514*   1515    1517    1621*   1622    1624    1850*   1859*   1861    1868*   1884    1899
                    2191*   2199    2205*   2973*   2974*   2975    3093*   3108*   3168*   3169*   3186*   3273*   3274*
                    3279*   3290*   3291*   3295*   3304*   3305*   3309*   3572*   3574*   3709    3837*   3838    3840
                    3891*   3892    3894    4343    4367*   4386    4390*   4394*   4399*   4400*   4403*   4405*   4409*
                    4411    4415*   4460*   4466*   4472*   4574    4594*   4597    4600    4612*   4754    4774*   4941
                    4951*   4953    4955    4965*   4966    4975*   5093*   5100*
R2      =%000002    1017#   1845*   1876    2192*   2913*   2919    3212*   3215*   3231*   3232*   3234    3238*   3239*
                    3241    3245*   3246*   3248    3252*   3253*   3255    3275*   3278*   3289*   3294*   3302*   3308*
                    3317    4344    4366*   4387    4392*   4395*   4401*   4404*   4406*   4407*   4408    4414*   4457*
                    4460    4461*   4467*   4468*   4473*   4474*   4575    4596*   4603    4611*   4755    4773*   4942
                    4947*   4950*   4969    4974*   5097*   5098*
R3      =%000003    1018#   1818*   2001*   2014*   2193*   2200*   2541*   2566*   2567    3285*   3287*   3300*   4345
                    4365*   4388    4391*   4396*   4413*   4465*   4470*   4476*   4477    4699    4708*   4714*   4715*
                    4718*   4723*   4724*   4725    4734*   4756    4772*   4893    4894*   4895    4898*   4899    4903
                    4905    4907*   4909*
R4      =%000004    1019#   1848*   2194*   2195*   2196*   2197*   2198*   2199*   2203    4346    4364*   4463*   4466
                    4472    4474    4700    4702*   4703*   4704*   4705    4706*   4720    4722*   4730*   4733*   4757
                    4771*
R5      =%000005    1020#   1596*   1597    1602    3874*   3875    3879    4347    4363*   4464*   4468    4475    4701
                    4707*   4709*   4711*   4712*   4713*   4714    4732*   4758    4770*
R6      =%000006    1021#   1023    1464*   1465*   1466    4094*   4107    4108*
R7      =%000007    1022#   1024
SAMCNT  014610      3092*   3106*   3167*   3187*   3272*   3280*   3288*   3296*   3303*   3310*   3343#
SAMOFF  014612      3083*   3085*   3186    3279    3295    3309    3344#
SAMPR   013132      3070    3080#
SAVREG= 104407      4456    5168#
SAVD    005330      1971*   1981    1990#
SCAN    011774      2853*   2854    2866#
SKPASK= 002574      1512    1573#
SP      =%000006    1023#   1468*   1480*   1481*   1487    1488    1489    1596    1602*   1660*   1661*   1667    1668
                    1724*   1798*   2447*   2645*   2646*   2781*   2932*   2933*   2995*   2997*   2998    3136*   3137*
                    3227*   3234*   3241*   3248*   3255*   3317*   3319*   3320    3874    3879*   3940*   4092*   4104*
                    4342*   4343*   4344*   4345*   4346*   4347*   4348*   4349*   4350*   4351*   4358*   4359*   4360*
                    4361*   4363    4364    4365    4366    4367    4368    4385*   4386*   4387*   4388*   4413    4414
                    4415    4416    4434    4435*   4437*   4457    4459*   4531    4573*   4574*   4575*   4589*   4605*
                    4611    4612    4613    4691*   4692    4693    4694*   4699*   4700*   4701*   4707    4732    4733
                    4734    4735*   4736*   4753*   4754*   4755*   4756*   4757*   4758*   4759    4765*   4770    4771
                    4772    4773    4774    4775    4780*   4811*   4814    4816    4817    4834    4836*   4852*   4853*
                    4855*   4874*   4879*   4880*   4893*   4898    4909    4910*   4911*   4912*   4937*   4938*   4940*
```

```
                    4941*  4942*  4944   4946*  4957   4959*  4960*  4961*  4962*  4963*  4966*  4971*  4972*
                    4974   4975   4976   4979   5012*  5013   5014*  5016   5017   5018*  5021   5023*  5025*
                    5031   5037*  5039   5062   5071   5082*  5128*  5129
STACK = 001100      1004#  1468
START   001726       980   1459#  1720   4781
START1  003652      1619   1714   1724#  1757   1804   2115   2733   2758   3908   4054   4187   5112
STKLMT= 177774      1009#
ST200   001632      1394#  1460   1699   1704*  1715
SWR     001136      1133#  1459*  1483   1485*  1499   1503*  1722*  1768*  1910   2148   2166   2763   2913
                    3639*  4544   4549   4806   4824
SWREG   000176       977#  1485
SWRFF   001672      1410#  3922   3925*  4190*
SWROUT  020166      3645   3932#  4191
SWROU1  020202      2647   2934   3138   3936#
SW0   = 000001      1062#
SW00  = 000001      1052#  1062
SW01  = 000002      1051#  1061
SW02  = 000004      1050#  1060
SW03  = 000010      1049#  1059
SW04  = 000020      1048#  1058
SW05  = 000040      1047#  1057
SW06  = 000100      1046#  1056
SW07  = 000200      1045#  1055
SW08  = 000400      1044#  1054
SW09  = 001000      1043#  1053
SW1   = 000002      1061#
SW10  = 002000      1042#
SW11  = 004000      1041#
SW12  = 010000      1040#
SW13  = 020000      1039#  4544
SW14  = 040000      1038#
SW15  = 100000      1037#
SW2   = 000004      1060#
SW3   = 000010      1059#
SW4   = 000020      1058#
SW5   = 000040      1057#
SW6   = 000100      1056#
SW7   = 000200      1055#
SW8   = 000400      1054#
SW9   = 001000      1053#
TBITVE= 000014      1095#
TBMTEN= 001000      1439#  1922   1925   1927   1930   3604   3618   4038   4043   5038
TBMTMS  031601      5065   5231#
TKVEC = 000060      1102#
TMPFIL  001656      1404#  1534   1536   1542*  1543*
TMPSWR  001702      1414#  1499*  1722
TMPVEC  001660      1405#  1510*  1565   1567   1569   1574   1575*  1576
TOLER   014614      3036   3050   3190   3193   3194   3197   3345#
TPBSY   001616      1388#  1826*  1866   1882   1896   1902*  1914*  1939*  1953   1983*  1991
TPBSYP  001630      1393#
TPCSR   001652      1402#  1701*  1734
TPDBR   001654      1403#  1702*  1735
TPVCT   001572      1378#  1823*
TPVCT2  001574      1379#  1825*
TPVEC = 000064      1103#
TRAPVE= 000034      1101#  1473*  1474*
```

```
TRTVEC= 000014  1096#
TSTLST  004152  1760  1774#
TSTNO   004146  1744  1749  1751  1753  1758  1772#
TSTO    004266  1774  1812#
TST1    005754  1775  2069#
TST10   006336  2122  2141#
TST2    006402  1776  2156#  2163
TST3    006472  1777  2181#  2188
TST4    006654  1778  2242#  2253
TST4A   006762  2270#
TST4AL  007014  2275  2278#
TST4B   007016  2284#
TST4C   007142  2315#
TST4D   007272  2343#
TST4E   007346  2362#
TST4F   007440  2386#
TST4G   007606  2428#
TST4GE  010044  2443  2461  2469  2475  2479#
TST4GI  007700  2428  2447#
TST4GL  007630  2430  2432#
TST4H   010060  2487#
TST4L   006720  2251  2255#  2499
TST4L1  006674  2248#  2261
TST5    010200  1779  2526#
TST6    011246  1780  2739#
TST7    012274  1781  2947#
TST8    013130  3074#
TTYEN = 000040  1443#  1922  1925  1930  2078  2098  2102  3604  3618  4038  4043  4181  4183
                4873  4875  5038
TTYTMP  001606  1384#  1928*  1929  5040*  5054*
TYPE  = 104400  1511  1520  1522  1529  1538  1556  1560  1571  1593  1600  1604  1635  1637
                1719  1742  1755  1795  1800  1812  1833  2069  2074  2076  2118  2129  2130
                2144  2156  2161  2165  2181  2186  2189  2242  2246  2252  2259  2260  2526
                2530  2718  2739  2751  2752  2756  2770  2775  2783  2791  2797  2802  2819
                2824  2848  2947  2957  2970  2981  3012  3017  3022  3033  3045  3056  3058
                3207  3210  3224  3229  3237  3243  3250  3260  3262  3265  3314  3321  3323
                3454  3478  3502  3525  3545  3567  3640  3715  3754  3758  3762  3765  3784
                3789  3797  3806  3815  3833  3843  3854  3871  3877  3897  3926  3936  3942
                3951  3962  3966  3981  3988  3989  4002  4014  4016  4059  4090  4102  4106
                4298  4312  4572  4585  4587  4593  4601  4609  4726  4778  4901  4904  4908
                4981  4983  5065  5079  5087  5104  5111  5147#
TYPECT= 004226  1794#  3671  4194
TYPOC = 104401  1603  1799  2782  3228  3235  3242  3249  3256  3880  3941  4093  4105  4591
                4606  5150#
TYPON = 104403  5156#
TYPOS = 104402  5153#
TOADR   001622  1390#  1835*  1899*  1962*  1963  1973  2002
TODAT   001624  1391#  1836*  1900*  1964*  1980  2004
TOGEN   001626  1392#  1837*  1901*  1961*  1969*  1970*  2006
TOIN    004506  1851  1857#
TOINT   005102  1822  1948#
TOIN1 = 004566  1873#  1906
TOIN2   004604  1872  1880#
TOLOP   005332  1870  1878  1890  1894  1991#  1992
TOOUT   004462  1849#
TOOUTR  004430  1843#  1995
```

F10


```
TOPINT= 004732  1823  1908#  1913  1926  1984
TBCAA   011266  2753#  2766  2769  2786
WTBSY = 104432  1852  2100  2147  2168  2202  2271  2288  2298  2319  2345  2365  2434  2488
                2543  2555  2575  2577  2603  2620  2629  2692  2701  2916  2926  3123  3127
                5226#
XFERMS  030571  3897  5231#
XFRCTL  017226  3648  3806#  3809  3816  3834
XICAR   001712  1418#  3825*  3826*  3866  3904
XICSHG  001716  1420#  3827*  3828*  3903
XICSLT  001714  1419#  3817*  3823*  3827  3865  3902
XICSR   001710  1417#  3818*  3822*  3825  3830  3864*  3867*  3882*  3887*  3901
XICSVT  001722  1422#  3856*  3861*  3905
XICSV2  001724  1423#  3858*  3862*
XRESET= 000100  1442#  2347  2369
XRIF  = 000001  1448#  1838  1960  2078  2095  2098  2619  2917  2927  3119  3128  3571  3600
                3604  3718  3864  4018  4034  4038  4181  4230  4854  4869  4873  5068
XTBMT = 100000  1452#  1923  3615  4040  5057
XTEMP   001600  1381#  4182*  4184*  4185  4188  4192
XTMPFL  001706  1416#  3811  3813  3819*  3820*
XVEC    001720  1421#  3848  3850  3852  3856  3857*  3858
$BDADR  001122  1127#  1346  1348  1349  1884*  2262*  2272*  2274  2289*  2291  2299*  2301  2320*
                2327  2346*  2351  2366*  2374  2410  2435*  2449  2465  2472  2480  2489*  2537*
                2544*  2546  2556*  2558  2576*  2578  2591  2604*  2606  2621*  2648  2663  2690
                2693*  2704  2754*  2761  2778  2919*  2928*  2951*  3124*  3129*  3131
$BDDAT  001126  1129#  1348  1466  1886*  2274*  2291*  2292  2301*  2327*  2328  2330  2351*  2374*
                2406*  2414*  2415  2448*  2456  2465*  2546*  2558*  2559  2578*  2579  2641*  2652
$CMTAG  001100  1115#  1464  1471  1477  1478
$CM3  = 000000  1143#
$CRLF   001161  1145#  1339  1636  1800  2119  2771  3265  3315  3715  4059  4538  4572  4593
                4609  4985
$DB2D   022372  4436  4456#
$DECVL  022552  4458  4504#
$ENDAD= 032054  2988  5239#
$ERFLG  001103  1118#  4527*  4538  4798  4820  4822*  4839
$ERMAX  001115  1124#  4839
$ERROR  022566  1471  4526#
$ERRPC  001116  1125#  4531*  4532*  4533  4538  4589
$ERRTB  001164  1164#  4582
$ERRTY  022670  4546  4572#
$ERTTL  001112  1122#  4530*  4538
$FILLC  001154  1141#  5021
$FILLS  001153  1140#
$GDADR  001120  1126#
$GDDAT  001124  1128#  1348  1349  1885*  2287*  2289  2292  2299  2303*  2322*  2323*  2330  2349*
                2364*  2409*  2449*  2450*  2451*  2455*  2456  2464*  2542*  2544  2554*  2556  2559
                2572*  2576  2579  2648*  2649*  2650*  2651*  2652  2680*  2683*  2684*  2687*  2688
                2690  2695  2702*
$HD   = 000001  965  966
$HINUM  022336  1491*  4390  4400  4411*  4418#
$ICNT   001104  1119#  4160  4828*  4829  4831*  4838
$ILLUP  023662  4750  4783#
$ITEMB  001114  1123#  4533*  4538  4577
$LF     001162  1146#  4538  4908  4985
$LONUM  022340  1492*  4287  4389  4398  4410*  4419#
$LPADR  001106  1120#  1478*  1843*  2270*  2286*  2317*  2344*  2363*  2430*  2490*  2540*  2573*  2679*
                4834*  4836  4838
```

```
$LPERR  001110          1121#
$MAIL = ****** U        1490    4536    4834
$MXCNT  024046          4832    4838#
$NULL   001152          1139#   5023
$OCNT   023530          4698#   4727*   4740#
$OMODE  023532          4693*   4697*   4702    4705*   4716*   4742#
$OVER   024032          4807    4818    4830    4835#
$PASS   001100          1116#   1728*   1819*   2258*   2717*   4826    4839
$PWRAD  023656          4781#
$PWRDN  023534          1475    4750#   4776
$PWRMG  023652          4779#
$PWRUP  023602          4760    4765#
$QUES   001160          1144#   4538    4901    4983    4985
$RAND   022212          4286    4384#
$RDCHR  024050          4852#   5157    5159
$RDDEC  024370          4937#   5163    5165
$RDLIN  024252          4893#   5160    5162
$RDOCT= ****** U        5163
$RESRE  022154          4357#   5171
$R2A  = ****** U        5172
$SAVRE  022116          4340#   5166    5168
$SAVR6  023666          4759*   4765    4766*   4767*   4785#
$SB2D   022342          2996    3318    4434#
$SCOPE  023670          1469    4805#
$SETUP= 000017          1458#   1469    1471    1473    1475    1477    1478
$STUP = 177777          1458#
$SVLAD  024022          4815    4833#
$SWR  = 164000           952#    965    1143    1144    1477    1478    1479    4512#   4523    4530    4534    4536    4538
                        4782    4788#   4799    4800    4801    4802    4806    4818    4820    4821    4822    4823    4824
                        4835    4838
$SWRMK= 000000          4802
$TIMES  001156          1143#   1477*   2285*   2316*   2343*   2362*   2487*   4823*   4829    4832*   4838
$TKB    001144          1136#   1787    2106    3622    4047    4879
$TKS    001142          1135#   1628*   1762*   2071*   2104    2146*   2538*   2753*   3620    4045    4163*   4877
$TN   = 000001           965#
$TNPWR  022502          4463    4464    4484#
$TPB    001150          1138#   1702    1735*   1933*   2109*   3055*   3629*   4050*   5031*
$TPFLG  001155          1142#   5008
$TPS    001146          1137#   1701    1734*   1912    1935*   1938*   2107    3054*   3627    4048    5029
$TRAP   025272          1473    5128#
$TRP  = 000034          5136#   5137    5138    5148#   5149    5150    5151#   5152    5153    5154#   5155    5156    5157#
                        5158    5159    5160#   5161    5162    5163#   5164    5165    5166#   5167    5168    5169#   5170
                        5171    5172#   5174    5175    5176#   5177    5178    5179#   5180    5181    5182#   5183    5184
                        5185#   5186    5187    5188#   5189    5190    5191#   5192    5193    5194#   5195    5196    5197#
                        5198    5199    5200#   5201    5202    5203#   5204    5205    5206#   5207    5208    5209#   5210
                        5211    5212#   5213    5214    5215#   5216    5217    5218#   5219    5220    5221#   5222    5223
                        5224#   5225    5226    5227#   5228    5229    5230#
$TRPAD  025314          5133    5146#
$TSTNM  001102          1117#   4529    4538    4798    4833*   4835    4839
$TTYIN  024360          4894    4895    4912    4917#
$TYPBN= ****** U        5157
$TYPDS= ****** U        5157
$TYPE   024546          1717    5008#   5136    5147
$TYPOC  023332          4696#   5148    5150
$TYPON  023346          4695    4698#   5156
$TYPOS  023306          4691#   5153
```

H10

DZIRBA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
$XTSTR  023700          4809#
$OFILL  023531          4692*   4696*   4706    4741#
$40CAT= ****** U        4534    4806
.       = 032054         971#    975#    988#    996#   1114#   1147    1461    1467    1478    1531    1562    1573    1591
                        1613    1665    1746    1794    1867    1869    1873    1908    1920    2022    2073    2085    2114
                        2808    2812    2832    2838    2846    2869#   2923    2956    2980    3001    3008    3021    3026
                        3044    3047    3057    3068    3126    3182    3209    3221    3259    3261    3453    3458    3477
                        3482    3501    3506    3524    3529    3544    3549    3591    3608    3611    3625    3628    3644
                        3647    3650    3655    3658    3661    3664    3667    3670    3675    3678    3682    3808    3845
                        3870    3888    3912    3913    3933    3950    3961    4013    4025    4041    4053    4063    4193
                        4297    4304    4504#   4538    4617    4762    4784    4838    4839    4860    4917#   4985    5035
                        5067    5077    5084    5091    5099    5108    5231#   5235#   5239
```

```
COMMEN      1#    1105#
ENDCOM      1#    1105#
ERROR    1005#    1889    1892    2277    2295    2304    2333    2354    2377    2389    2401    2418    2441    2459    2468
         2477     2497    2549    2562    2585    2594    2609    2628    2655    2671    2698    2707
ESCAPE      1#    1105#
GETPRI      1#
MULT        1#    1105#
NEWTST      1#    1105#
POP         1#    1105#   4362    4412    4769    4973
PUSH        1#    1105#   4341    4384    4752    4939
REPORT      1#
SCOPE    1006#    1993    2278    2306    2334    2356    2379    2391    2420    2481    2498    2551    2564    2599    2612
         2633     2699    2709
SETPRI      1#
SETTRA    950#   5136#   5148    5151    5154    5157    5160    5163    5166    5169    5173    5176    5179    5182    5185
         5188    5191    5194    5197    5200    5203    5206    5209    5212    5215    5218    5221    5224    5227
SETUP       1#    1105#   1463
SKIP        1#    1105#
SLASH       1#    1105#
SPACE    1105#
STARS       1#    1105#   1106    1147    4321    4371    4421    4441    4515    4556    4664    4744    4791    4841    4884
         4921     5118
SWRSU       1#    1479#
TRMTRP   5136#
TYPBIN      1#    1105#
TYPDEC      1#    1105#
TYPNAM      1#
TYPNUM      1#    1105#
TYPOCS      1#    1105#
TYPOCT      1#    1105#   1601    1797    2780    3226    3233    3240    3247    3254    3878    3939    4091    4103    4588
         4604
TYPTXT      1#    1105#
$SCMRE   1106#
$SCMTM   1106#
$SESCA      1#    1105#
$SNEWT      1#    1105#
$SSET    5136#    5137    5149    5152    5155    5158    5161    5164    5167    5170    5174    5177    5180    5183    5186
         5189     5192    5195    5198    5201    5204    5207    5210    5213    5216    5219    5222    5225    5229
$SSKIP      1#    1105#
.EQUAT      1#     949#     999
.HEADE      1#     949#     954
.KT11       1#
.SETUP      1#     949#    1457
.SWRHI      1#
.TRMTR    950#
.SACT1      1#
.SAPTB      1#
.SAPTH      1#
.SAPTY      1#
.SASTA      1#
.SCATC      1#     950#     967
.SCMTA      1#     949#    1105
.SDB2D      1#     951#    4440
.SDB2O      1#
.SDIV       1#
.SEOP       1#
```

```
.$ERRO      1#    950#   4514
.$ERRT      1#    949#
.$MULT      1#
.$POWE      1#    949#   4743
.$RAND      1#    951#   4370
.$RDDE      1#    950#   4920
.$RDOC      1#
.$READ      1#    950#
.$R2AZ      1#
.$SAVE      1#    951#   4320
.$SB2D      1#    951#   4420
.$SB2O      1#
.$SCOP      1#    950#   4790
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    950#   5117
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#    949#
.$TYPO      1#    949#   4663
.$40CA      1#
```

```
        2208  2261  2296  2402  2443  2461  2469  2586  2636  2658  2668  2673  2710  2769  2786
        2823  2865  2923  2961  2962  2972  3001  3002  3014  3019  3026  3027  3035  3047  3048
        3057  3072  3126  3209  3261  3270  3286  3458  3459  3482  3483  3506  3507  3529  3530
        3549  3550  3603  3808  3809  3816  3824  3831  3834  3845  3846  3855  3873  3888  3912
        3913  3983  4004  4037  4044  4058  4063  4304  4305  4313  4471  4610  4695  4710  4731
        4762  4784  4809  4815  4818  4872  4876  4902  4968  4984  5011  5028  5067
BVS     1657  4964  4967
CLC     4256
CLR     1462  1465  1477  1490  1493  1589  1631  1640  1645  1669  1695  1698  1704  1705  1708
        1709  1710  1718  1721  1727  1728  1729  1730  1733  1764  1765  1766  1767  1768  1769
        1801  1819  1820  1826  1828  1835  1836  1837  1847  1901  1905  1914  1938  1939  1955
        1957  2071  2080  2081  2094  2143  2191  2303  2321  2346  2349  2364  2395  2409  2413
        2438  2440  2452  2453  2454  2464  2531  2539  2574  2624  2630  2642  2665  2680  2681
        2682  2806  2834  2851  2857  2921  2928  2929  2935  2938  2983  2984  2985  2986  2989
        3122  3129  3133  3139  3142  3166  3170  3171  3275  3289  3302  3573  3578  3579  3581
        3585  3586  3587  3602  3700  3735  3788  3868  3907  3910  3924  3925  3964  3971  3972
        3994  4020  4021  4036  4064  4147  4262  4302  4303  4392  4465  4576  4619  4708  4766
        4823  4856  4871  4946  4947  5041  5052  5064  5086  5093  5110
CLRB    2012  2013  2847  4481  4822  4907  4980
CMP     1466  1487  1517  1518  1536  1567  1569  1597  1624  1625  1706  1753  1829  1854  1861
        1871  1876  1973  1975  2256  2292  2328  2330  2336  2456  2559  2567  2579  2606  2652
        2688  2690  2695  2704  2789  2795  2800  2807  2837  2845  2863  2975  3010  3015  3065
        3067  3104  3174  3177  3191  3195  3201  3276  3292  3306  3468  3492  3516  3582  3601
        3624  3630  3634  3643  3646  3649  3654  3672  3674  3677  3681  3695  3729  3733  3737
        3740  3746  3813  3840  3841  3850  3852  3875  3894  3895  4052  4055  4060  4084  4086
        4185  4188  4192  4254  4259  4264  4475  4816  4829  4895  5048
CMPB    1789  1791  2111  2113  2121  2123  2859  3657  3660  3663  3666  3669  4899  4905  4948
        4953  4955  5021
COM     2821  3949
DEC     1543  2200  2206  2581  2993  3061  3087  3187  3215  3280  3296  3310  3704  3707  3710
        3736  3820  3977  3999  4119  4121  4134  4136  4148  4150  4478  4578  4607  4621  5059
        5090  5098  5100  5102
DECB    4716  4727  5025
EMT     1005
HALT     975  1608  3074  4551  4761  4783  5010  5066
INC     1502  1542  1590  1633  1755  1893  1902  1983  2023  2089  2258  2323  2324  2623  2717
        2822  2862  2920  2990  3063  3103  3106  3121  3200  3278  3294  3299  3300  3308  3595
        3684  3688  3732  3751  3785  3799  3819  3869  3934  4029  4062  4190  4238  4257  4396
        4470  4530  4722  4730  4767  4828  4864  5044  5085  5109
INCB    2841  2843  4527  4833
IOT     1006
JMP      980   997  1512  1609  1720  1770  1804  1870  1878  1890  1894  1995  2086  2115  2163
        2188  2253  2499  2647  2720  2733  2758  2934  3138  3316  3592  3609  3614  3619  3626
        3641  3645  3648  3651  3656  3659  3662  3665  3668  3671  3676  3679  3683  3692  3726
        3743  3752  3756  3760  3764  3767  3786  3791  3800  3884  3930  3944  3953  4001  4026
        4054  4175  4187  4191  4194  4223  4236  4237  4861  5051  5056  5078  5080  5105  5112
JSR     2003  2005  2008  2759  2762  2772  2779  2954  3223  3236  3257  3264  3313  3467  3491
        3515  3538  3558  4286  4436  4534  4546  4592  4608  5020  5027
MOV     1459  1464  1468  1469  1470  1471  1472  1473  1474  1475  1476  1478  1480  1481  1482
        1485  1486  1488  1489  1491  1492  1499  1503  1504  1505  1506  1507  1509  1510  1513
        1514  1515  1516  1528  1530  1540  1541  1548  1550  1552  1553  1558  1559  1561  1574
        1576  1582  1583  1584  1585  1586  1594  1596  1602  1607  1610  1619  1620  1621  1622
        1623  1627  1629  1630  1644  1649  1651  1652  1653  1654  1660  1661  1662  1667  1668
        1670  1672  1675  1678  1681  1685  1689  1691  1701  1702  1703  1711  1717  1722  1724
        1731  1732  1734  1735  1739  1744  1745  1758  1761  1762  1787  1796  1798  1818  1822
        1823  1824  1825  1827  1843  1845  1848  1849  1850  1853  1859  1860  1863  1868  1884
```

M10

```
        2082  2092  2096  2133  2159  2184  2250  2410  2415  2472  2480  2591  2643  2663  2765
        2787  2811  2815  2831  2855  2930  2955  2968  2973  2998  3006  3020  3040  3043  3052
        3089  3134  3143  3181  3197  3220  3258  3267  3283  3452  3463  3476  3487  3500  3512
        3523  3534  3543  3554  3569  3588  3598  3601  3610  3632  3636  3652  3693  3697  3716
        3719  3723  3744  3748  3795  3866  3922  3932  3960  3975  3997  4012  4022  4032  4035
        4076  4088  4107  4171  4228  4231  4296  4547  4549  4598  4616  4720  4813  4826  4857
        4867  4870  4969  4979  5016  5032  5045  5069  5070  5083  5096  5107  5130
TSTB    1912  2104  2107  3620  3627  4045  4048  4820  4877  5008  5029
WAIT    3911
XOR     1656
.ASCII  1144  1145  5231
.ASCIZ  1146  4629  4631  4634  4637  4639  4642  4645  4647  5231
.BLKB   4504  4917
.BYTE   1117  1118  1123  1124  1139  1140  1141  1142  2870  2873  2876  2879  2882  2885  2888
        2891  2894  2897  2900  2903  3356  4738  4739  4740  4741  4915  4916  5232  5233
.ENABL     1   948
.END    5240
.ENDC    960   981  1005  1091  1105  1108  1115  1117  1143  1144  1149  1458  1468  1469  1471
        1473  1475  1477  1478  1479  1490  4323  4373  4423  4443  4517  4523  4527  4531  4534
        4535  4536  4537  4538  4558  4666  4746  4759  4769  4780  4782  4786  4793  4799  4802
        4806  4808  4819  4820  4822  4824  4828  4833  4834  4835  4838  4839  4843  4886  4923
        5120  5129  5132  5147  5148  5150  5153  5156  5157  5159  5160  5162  5163  5165  5166
        5168  5171  5172  5175  5178  5181  5184  5187  5190  5193  5196  5199  5202  5205  5208
        5211  5214  5217  5220  5223  5226  5229
.EQUIV  1005  1006  1008  1023  1024  1053  1054  1055  1056  1057  1058  1059  1060  1061  1062
        1081  1082  1083  1084  1085  1086  1087  1088  1089  1090
.EVEN   2906  4650  5231
.IF      956   978  1003  1063  1091  1107  1114  1116  1143  1144  1148  1149  1458  1464  1468
        1469  1471  1473  1475  1477  1478  1490  4322  4372  4422  4442  4516  4522  4527  4530
        4534  4535  4536  4537  4538  4557  4665  4745  4759  4769  4778  4780  4782  4786  4792
        4798  4802  4806  4818  4820  4821  4822  4824  4826  4834  4835  4837  4838  4839  4842
        4885  4922  5119  5128  5132  5136  5138  5148  5150  5153  5156  5157  5159  5160  5162
        5163  5165  5166  5168  5171  5172  5175  5178  5181  5184  5187  5190  5193  5196  5199
        5202  5205  5208  5211  5214  5217  5220  5223  5226  5229
.IFF     1003  1108  1114  1116  1143  1149  1468  4323  4373  4423  4443  4517  4522  4531  4536
        4538  4558  4666  4746  4778  4793  4818  4820  4822  4838  4839  4843  4886  4923  5120
        5129
.IFT    4535  4824
.IFTF   4534  4822
.IIF     955   960   965   966   975  1147  1469  1471  1477  1478  1479  1603  1799  2782  3228
        3235  3242  3249  3256  3880  3941  4093  4105  4523  4538  4590  4606  4799  4800  4801
        4802  4823  4835  4838  4839  4985  5147  5150  5153  5156  5159  5162  5165  5168  5171
        5175  5178  5181  5184  5187  5190  5193  5196  5199  5202  5205  5208  5211  5214  5217
        5220  5223  5226  5229
.IRP    1458  4342  4363  4385  4413  4753  4770  4940  4974
.LIST      1   948   975  1105  1143  1458  1479  4802  5136  5147  5148  5150  5151  5153  5154
        5156  5157  5159  5160  5162  5163  5165  5166  5168  5169  5171  5172  5175  5176  5178
        5179  5181  5182  5184  5185  5187  5188  5190  5191  5193  5194  5196  5197  5199  5200
        5202  5203  5205  5206  5208  5209  5211  5212  5214  5215  5217  5218  5220  5221  5223
        5224  5226  5227  5229  5230  5231
.MACRO     1  1106  5136
.MCALL   949   950   951  1105  1479
.NLIST     1   948   975  1105  1143  1458  1479  4802  5136  5147  5148  5150  5151  5153  5154
        5156  5157  5159  5160  5162  5163  5165  5166  5168  5169  5171  5172  5175  5176  5178
        5179  5181  5182  5184  5185  5187  5188  5190  5191  5193  5194  5196  5197  5199  5200
        5202  5203  5205  5206  5208  5209  5211  5212  5214  5215  5217  5218  5220  5221  5223
```

DZIRBA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
        5224    5226    5227    5229    5230    5231
.PAGE   1106    1147
.REM       1
.REPT    975    3375    3392    3409    3431
.SBTTL   969     979    1001    1109    1150    1431    1450    4324    4374    4424    4444    4518    4559    4667    4747
        4794    4844    4924    4988    5121    5139
.TITLE   955
.WORD    975     976     977    1116    1119    1120    1121    1122    1125    1126    1127    1128    1129    1130    1131
        1132    1133    1134    1346    1348    1349    3375    3376    3377    3378    3379    3380    3381    3382    3383
        3384    3385    3386    3387    3388    3389    3392    3393    3394    3395    3396    3397    3398    3399    3400
        3401    3402    3403    3404    3405    3406    3409    3410    3411    3412    3413    3414    3415    3416    3417
        3418    3419    3420    3421    3422    3423    3431    3432    3433    3434    3435    3436    3437    3438    3439
        3440    3441    3442    3443    3444    3445    4099    4418    4419    4439    4479    4586    4602    4652    4653
        4655    4657    4658    474z    4779    4781    4982
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*,DZIRBA.SEQ/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.B1,DZIRBA.P11
RUN-TIME:  35 51 8 SECONDS
RUN-TIME RATIO:  233/95=2.4
CORE USED:  29K  (57 PAGES)
```

Spooler running 10 Seconds, 75 KCS, 493 disk reads, 3 disk writes, 131 pages