ICS-11

ICS-11 FIELD TEST PROGRAM

MD-11-DZICA-D

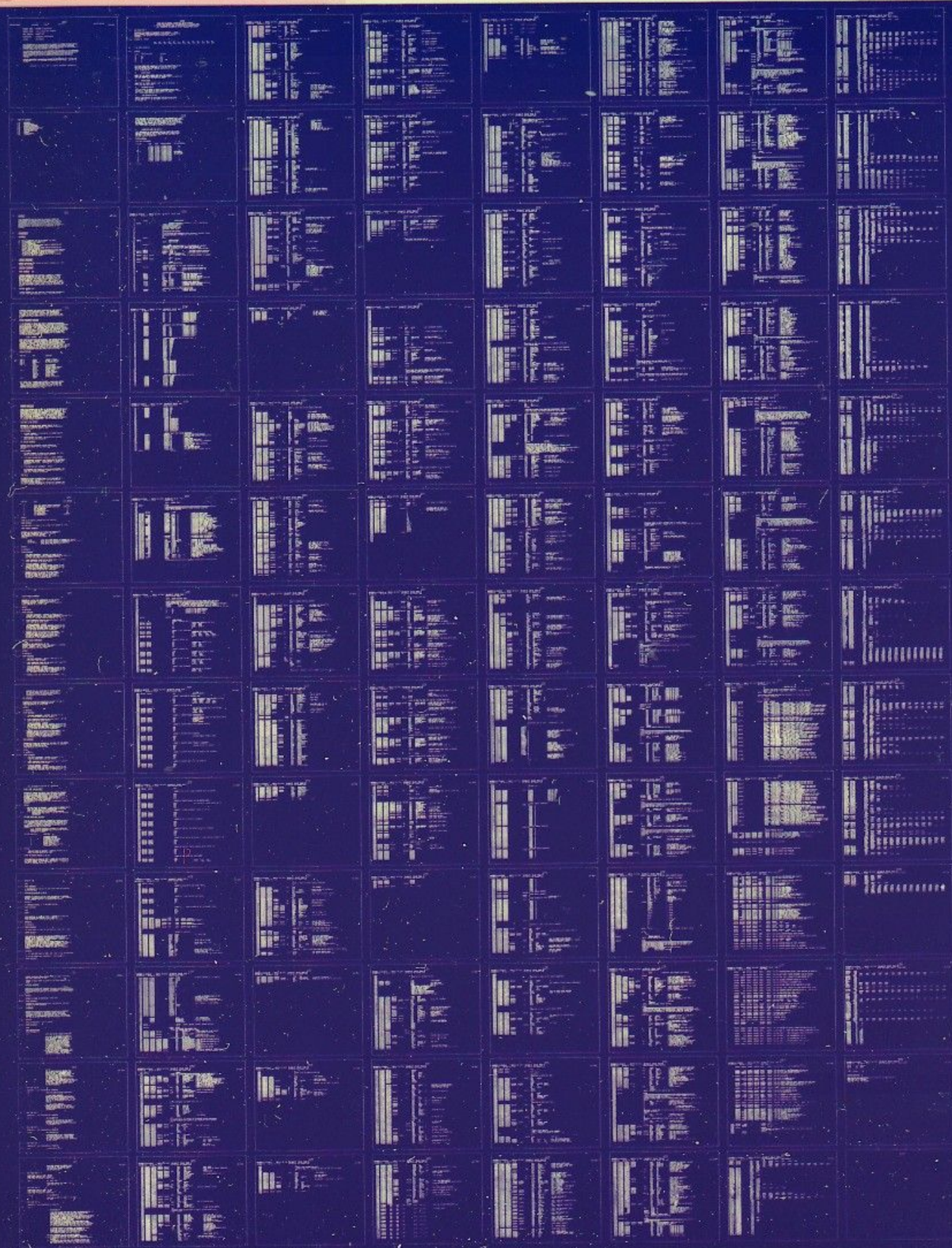
EP-DZICA-D-DL-D

FICHE 1 OF 1

OCT 1977

digital

MADE IN USA



PRODUCT CODE: MAINDEC-11-DZICA-D-D

PRODUCT NAME: ICS-11 FIELD TEST PROGRAM

DATE CREATED: AUGUST 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: DAN DEKNIS

0.0 TABLE OF CONTENTS

SEQ 0002

1.0 ABSTRACT

THIS PROGRAM ALLOWS THE USER TO CHECKOUT, DEBUG OR DEMONSTRATE THE INDUSTRIAL CONTROL SUBSYSTEM (ICS-11). THE PROGRAM IS DIVIDED INTO 8 DIFFERENT TESTS AIMED AT EXERCISING THE OPTIONS ON THE ICS-11 BUS. THE TESTS ARE SELECTED BY THE OPERATOR; ANY ADDITIONAL INFORMATION NEEDED TO RUN A PARTICULAR TEST IS REQUESTED BY THE PROGRAM.

2.0 REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11 COMPUTER WITH 8K OF MEMORY (OR MORE).
B. I/O TERMINAL (I.E. ASR33 TTY)
C. ICS-11 CONTROLLER
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.

3.0 LOADING PROCEDURE

NORMAL PROCEDURE FOR LOADING AN OBJECT PROGRAM INTO CORE SHOULD be FOLLOWED.

4.0 STARTING PROCEDURE

4.1 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 ICS 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).

4.2 RESTART ADDRESS 210

LOADING ADDRESS 210 AND STARTING, RETAINS ANY INFORMATION (IF ANY) THAT WAS PREVIOUSLY TYPED. ANY DEFAULT (OR

STANDARD) ANSWERS THAT WERE OVERRIDEN BY THE OPERATOR, WILL RETAIN THE OVERRIDEN VALUE. THE PROGRAM WILL TYPE OUT "6.3" TEST NO.? AND WAIT FOR THE OPERATOR TO SELECT A TEST TO RUN. ONCE A TEST IS SELECTED, ONE OR MORE ADDITIONAL QUESTIONS WILL BE ASKED (SEE SECTION 6.0).

5.0 SWITCH REGISTER SETTINGS

IF THE PROGRAM IS BEING RUN FROM A SWITCHLESS PROCESSOR (I.E. AN 11/34), THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

EACH TIME THE SETTINGS ARE ENTERED, THE ENTIRE REGISTER IMAGE MUST BE ENTERED. LEADING ZEROES ARE NOT REQUIRED. THE 'CONTROL U' FUNCTION MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY. ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE MADE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

5.1 SWITCH OPTIONS

TEST NO.	SWR	FUNCTION
TEST 0	SW13=1	INHIBIT TYPEOUT
TEST 1	SW00=1	RETURN TO MONITOR
TEST 2	SW04-SW13	DAC OUTPUT LEVEL
TEST 3	NONE	
TEST 4+5	SW15=1	HALT ON ERROR
	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.
	SW04-07	SELECT CHAN.
	SW00=1	RETURN TO MONITOR
TEST 7	NONE	

ALL TESTS "↑C" (CONTROL AND LETTER C) TYPED WILL RETURN YOU TO MONITOR. RUBOUT DELETES ENTIRE LINE THAT WAS TYPED. SWITCH REGISTER SETTINGS WILL USE EITHER THE HARDWARE SWITCH REGISTER (WHEN AVAILABLE) OR THE SOFTWARE SWITCH REGISTER (LOC. 176).

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

SINCE THIS PROGRAM HAS NO WAY OF DETERMINING THE PARTICULAR CONFIGURATION OF YOUR ICS-11, 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 ICS 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 PROPER 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" (CONTROL AND LETTER C DEPRESSED TOGETHER) OR BY SETTING SWR BIT 00=1.

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

3.4 INPUT MODULE ADDRESS?

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 BOX RANGE" - 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 MODULE ADDRESS?

SEQ 0007

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 BOX RANGE" - 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 COUNTER MODULE ADDRESSES?

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 BOX RANGE" - 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 ADDRESS?

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.

(4) "ADDRESS XXXXXX NOT WITHIN FILE BOX RANGE" - 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 AOD5 ADDRESS?

RESPOND BY TYPING THE ADDRESS OF THE AOD5, OR THE ADDRESS
OF THE AOD5 CONNECTED TO THE AOD7 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 TIME (IN MILLISE)?

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 PATTERN MODIFIER, PATTERN?

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 PATTERN MODIFIER, PATTERN ? 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 PATTE N

6.10.1 DEFAULT

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

6.11 INPUT OR OUTPUT MODULE (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 .YPED THAN THE QUESTION WILL BE RETYPED.

6.12 UNIPOLAR OR BIPOLAR (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.

6.12.2 ERROR MESSAGES

NONE

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 CHANNELS YOU WISH THE SAMPLES TO BE TAKEN ON. SC REPRESENTS THE STARTING CHANNEL AND EC REPRESENT 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! STARTING CHAN. GREATER THAN 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 AD05 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 AD05 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 AND 16 INPUT 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 RO WITH
USE OF THE RESET INSTRUCTION. 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. 5 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. 1 MIN.
FIRST PASS IS ALWAYS A SHORT PASS.

7.7 TEST 6 A/D CALIBRATION

THIS TEST ALLOWS THE USER TO CALIBRATE
THE ADDS.

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 OF
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 ERROR
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 ADDS ADDR OF THE
A/D SUBSYSTEM IS REQUIRED ALONG WITH WHAT CHANNELS
YOU WISH TO SAMPLE, THE GAIN YOU WISH TO USE, FOLLOWED
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 EACH
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, IF

THERE ARE NO ERRORS, THERE WILL BE NO TYPEOUT
EXCEPT FOR "REPEAT" AT THE BEGINNING OF THE NEXT PASS.

(4) IF A TOLERANCE OF ZERO IS SPECIFIED, A FORCED TYPEOUT 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

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

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.

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.

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	-----

10. LISTING

```
                        164000                          $SWR=164000

                                                        .HEADER ↑/MAINDEC-11-DZICA-D/, 1977, ↑/DAN DEKNIS/
                                                .TITLE  MAINDEC-11-DZICA-D
                                                ;*COPYRIGHT (C) 1977
                                                ;*DIGITAL EQUIPMENT CORP.
                                                ;*MAYNARD, MASS. 01754
                                                ;*
                                                ;*PROGRAM BY DAN DEKNIS
                                                ;*
                                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                                ;*
                        000001                  $TN=1

                                                        .$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  001704                                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

                        000210                          .=210
000210  000137  003374                                  JMP     @#START1

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

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

                                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
                        000000                  R0=     %0              ;;GENERAL REGISTER
                        000001                  R1=     %1              ;;GENERAL REGISTER
```

```
                        R2=     %2              ;;GENERAL REGISTER
                        R3=     %3              ;;GENERAL REGISTER
                        R4=     %4              ;;GENERAL REGISTER
                        R5=     %5              ;;GENERAL REGISTER
                        R6=     %6              ;;GENERAL REGISTER
                        R7=     %7              ;;GENERAL REGISTER
                        SP=     %6              ;;STACK POINTER
                        PC=     %7              ;;PROGRAM COUNTER

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

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

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        BIT15=  100000
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
                        BIT09=  1000
                        BIT08=  400
```

```
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR.*
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
000214                                  .$CMTAG
```

```
                        .SBTTL  COMMON TAGS

000214                  STARS
                        ;;**********************************************************************
                        ;*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      000         $AUTOB: .BYTE   0       ;;AUTOMATIC MODE INDICATOR
001135      000         $INTAG: .BYTE   0       ;;INTERRUPT MODE INDICATOR
001136  000000                 .WORD   0
001140  177570         SWR:    .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
001142  177570         DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
001144  177560         $TKS:   177560          ;;TTY KBD STATUS
001146  177562         $TKB:   177562          ;;TTY KBD BUFFER
001150  177564         $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566         $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
001154      000         $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001155      002         $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156      012         $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157      000         $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000         $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
001162      077         $QUES:  .ASCII  /?/     ;;QUESTION MARK
001163      015         $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001164  000012         $LF:    .ASCIZ  <12>    ;;LINE FEED
001166                  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


001166                  $ERRTB:

                                ;ADDS ERRORS
                                ;ERROR 1 ADDRESS ABILITY
001166  022543                 EM1                     ;"COULD NOT SEND-RECIEVE DATA"
001170  023140                 DH1                     ;ERROR   ADDS
                                                        ;  PC    ADDR
001172  001500                 DT1                     ;$ERRC,  ADCSR,0
001174  001476                 DF0                     ;0

                                ;ERROR 2 "BIT EXERCISER"
001176  022601                 EM2                     ;"SEND-RECIEVE DATA ERROR"
001200  023173                 DH2                     ;ERROR   A/D    GOOD    BAD
                                                        ;  PC    ADDR    DATA    DATA
001202  001506                 DT2                     ;$ERRPC, ADCSR,  $GDDAT, $BDDAT,0
001204  001476                 DF0                     ;0

                                ;ERROR 3 CONVERT BIT

001206  022633                 EM3                     ;CONVERT BIT FAILED TO SET
001210  023140                 DH1                     ;ERROR   A/D
001212  001500                 DT1                     ;  PC    ADDR
001214  001476                 DF0                     ;$ERRPC, ADCSR,0

                                ;ERROR 4 CONVERT BIT

001216  022667                 EM4                     ;CONVERT BIT FAILED TO CLEAR
001220  023140                 DH1                     ;ERROR   A/D
001222  001500                 DT1                     ;  PC    ADDR
001224  001476                 DF0                     ;$ERRPC, ADCSR,0

                                ;ERROR 5 READ C/R BIT
001226  022725                 EM5                     ;CAN'T READ DATA REGISTER
001230  023140                 DH1                     ;ERROR   A/D
001232  001500                 DT1                     ;  PC    ADDR
001234  001476                 DF0                     ;$ERRPC, ADCSR,0

                                ;ERROR 6 ADDS INTERRUPT

001236  022760                 EM6                     ;ADDS FAILED TO INTERRUPT
001240  023140                 DH1                     ;ERROR   A/D
001242  001500                 DT1                     ;  PC    ADDR
```

```
   246  001244  001476                    DF0                     ;$ERRPC, ADCSR,0
   247
   248                                    ;ERROR 7 ADOS ADDR OR GENERIC CODE
   249
   250  001246  023013                    EM7                     ;ADDR, OR GENERIC CODE INCORRECT ON INTR.
   251  001250  023267                    DH3                     ;ERROR   MODULE    ICAR
   252  001252  001506                    DT2                     ;  PC    ADDR      S/B    WAS
   253  001254  001476                    DF0                     ;$ERRPC, ADCSR,  $GDDAT, $BDDAT
   254
   255                                    ;ERROR 10 RIF BIT ACTION
   256
   257  001256  023072                    EM10                    ;RIF DID NOT CLEAR INTR. FLAG ON ADOS
   258  001260  023140                    DH1                     ;ERROR   A/D
   259  001262  001500                    DT1                     ;  PC    ADDR
   260  001264  001476                    DF0                     ;$ERRPC, ADCSR
   261
   262                                    ;ERROR 11 COUNTER MODULE ADDRESSABILITY
   263
   264  001266  023417                    EM11                    ;COUNTER MODULE COULD NOT READ-WRITE BUFFER
   265  001270  023501                    DH4                     ;ERROR COUNTER
   266  001272  001520                    DT3                     ;PC ADDR
   267  001274  001476                    DF0                     ;$ERRPC $BDADR
   268
   269                                    ;ERROR 12 COUNTER MODULE "BIT EXERCISE ROUTINE
   270
   271  001276  022601                    EM2
   272  001300  023173                    DH2
   273  001302  001526                    DT4
   274  001304  001476                    DF0
   275
   276                                    ;ERROR 13 COUNTER MODULE COUNTING
   277
   278  001306  023542                    EM12                    ;COUNTER MODULE MISSED A COUNT
   279  001310  023173                    DH2
   280  001312  001526                    DT4
   281  001314  001476                    DF0
   282
   283                                    ;ERROR 14 COUNTER MODULE FAILED TO INTERRUPT
   284
   285  001316  023606                    EM13                    ;COUNTER FAILED TO INTERRUPT
   286  001320  023501                    DH4
   287  001322  001520                    DT3
   288  001324  001476                    DF0
   289
   290                                    ;ERROR 15 COUNTER MODULE ADDR OR GENERIC CODE INCORRECT
   291
   292  001326  023653                    EM14
   293  001330  023267                    DH3
   294  001332  001526                    DT4
   295  001334  001476                    DF0
   296
   297                                    ;ERROR 16 COUNTER DIDN'T HALT ON OVERFLOW
   298
   299  001336  023751                    EM15
   300  001340  023173                    DH2
   301  001342  001526                    DT4
```

```
   302  001344  001476                        DF0
   303
   304                                        ;ERROR 17 RIF DIDN'T CLEAR INTERRUPT FLAG ON COUNTER
   305
   306  001346  024033                        EM16
   307  001350  023501                        DH4
   308  001352  001520                        DT3
   309  001354  001476                        DF0
   310
   311                                        ;ERROR 20 COUNTER MODULE INITIALIZATION PART 1
   312
   313  001356  024110                        EM17                    ;INIT FAILED TO CLEAR COUNTER BIT(S)
   314  001360  023173                        DH2
   315  001362  001526                        DT4
   316  001364  001476                        DF0
   317
   318                                        ;ERROR 21 ILLEGAL INTERRUPT POSTED ON ICS BUS
   319
   320  001366  024164                        EM20
   321  001370  023501                        DH4
   322  001372  001520                        DT3
   323  001374  001476                        DF0
   324
   325                                        ;ERROR 22 ICS-11 FAILED TO INTERRUPT
   326
   327  001376  024232                        EM21
   328  001400  023501                        DH4
   329  001402  001520                        DT3
   330  001404  001476                        DF0
   331
   332                                        ;ERROR 23 ICAR NOT ZERO AFTER FORCED ICS INTERRUPT.
   333
   334  001406  024270                        EM22
   335  001410  023267                        DH3
   336  001412  001526                        DT4
   337  001414  001476                        DF0
   338
   339                                        ;ERROR 24 SYS INITIALIZE FAILED TO CLEAR COUNTER INTERRUPT FLAG
   340
   341  001416  024341                        EM23
   342  001420  023501                        DH4
   343  001422  001520                        DT3
   344  001424  001476                        DF0
   345
   346                                        ;ERROR 25 COUNTER STARTED COUNTING AFTER SYSTEM INIT.
   347
   348  001426  024421                        EM24
   349  001430  023173                        DH2
   350  001432  001526                        DT4
   351  001434  001476                        DF0
   352
   353                                        ;ERROR 26 A005 READ DUAL ADDR ERROR
   354
   355  001436  024502                        EM25
   356  001440  024621                        DH5                     ;ERROR   A/D    DUAL
   357  001442  001540                        DT5                     ;  PC     ADDR    ADDR
```

```
                                        DF0

                                        ;ERROR 27 A005 WRITE DUAL ADDR. ERROR

                                        EM26
                                        DH5
                                        DT5
                                        DF0


                                        ;ERROR 30   SEND -RECIEVE DATA ERROR

                                        EM27
                                        DH2
                                        DT4
                                        DF0


                                        ;ERROR 31     SAME AS ERROR 30 ONLY NO HEADER TYPEOUT

                                        DF0
                                        $CRLF
                                        DT4
                                        DF0


                                DF0:    0
                                DT1:    .WORD   $ERRPC, ADCSR,0
                                DT2:    .WORD   $ERRPC,ADCSR,$GDDAT,$BDDAT,0
                                DT3:    .WORD   $ERRPC,$BDADR,0
                                DT4:    .WORD   $ERRPC,$BDADR,$GDDAT,$BDDAT,0
                                DT5:    .WORD   $ERRPC,ADCSR,$BDADR,0


                                        ;*TABLE OF CONSTANTS

                                FR110:  20320                   ;TTY DELAY TIME
                                FR50:   6532
                                FR40:   5650
                                FR20:   2724
                                FR16:   2052                    ;DELAY FOR 16 MILLSEC
                                FR5:    1024
                                FR3:    500                     ;DELAY FOR 3 MILLI SEC.
                                FREQ:   0                       ;DELAY TIME.
                                FREQ1:  0
                                FREQ2:  0
                                FREQ3:  0
                                FR1:    152                     ;DELAY FOR 1 MILLI SEC.
                                FR1120: 220                     ;ON AN 11/20  FRI X 3 FOR 11/45.
                                PATRNM: 0                       ;PATTERN MODIFIER
                                PATRN:  0                       ;PATTERN TO BE SENT TO OUTPUT MODULE
                                PATRNC: 0
                                PATJOY: 0
```

```
   414  001612  171000                  ICSMOD: 171000                  ;STARTING ADDRESS OF ICS MODULES.
   415  001614  171776                  ICSR:   171776
   416  001616  171774                  ICAR:   171774
   417  001620  000234                  ICSVT:  234
   418  001622  000236                  ICSVT2: 236
   419  001624  000064                  TPVCT:  64
   420  001626  000066                  TPVCT2: 66
   421  001630  000000                  INCFLG: 0
   422  001632  000000                  TPBSY:0
   423  001634  000000                  HEADER: 0
   424  001636  000000                  TOADR:0
   425  001640  000000                  TODAT:0
   426  001642  000000                  TOGEN:  0
   427  001644  000000                  TPBSYP: 0
   428  001646  000001                  ST200:  1                       ;CLEARED ON PROGRAM START AT 200
   429  001650  000000                  CORSIZ: 0                       ;END ADDR OF CORE.
   430  001652  000000                  EXPERT: 0                       ;0=NOVICE MODE--1=EXPERT MODE
   431  001654  000000                  CONNT:  0                       ;0=NORMAL; 1=MODS CONNECTED FOR TST 0
   432  001656  000000                  LPAV:   0
   433  001660  000000                  LINEPR: 0                       ;0=TTY OUTPUT---1=LINE PRINTER
   434  001662  177514                  LPCSR:  177514
   435  001664  177516                  LPDBR:  177516
   436  001666  000000                  TPCSR:  0
   437  001670  000000                  TPDBR:  0
   438  001672  000000                  TMPFIL: 0
   439  001674  000000                  TMPVEC: 0
   440  001676  000000                  ICSHGH: 0
   441  001700  000000                  CTLLOC: 0
   442  001702  000000                  ICSLMT: 0
   443
   444          000002                          EXIT=   2               ; RTI INSTRUCTION
   445          000207                          RETURN=207              ;RTS PC INSTRUCTION
   446          004737                          GOSUB=4737              ;JSR PC, INSTRUCTION
   447          022626                          POPSP2=22626
   448
   449  001704                          .SETUP  <.$TRAP,.$SCOPE,.$ERROR,.$POWER,SOFTSWR>
   450
   451  001704                  START:  SETUP
   452                          .SBTTL  INITIALIZE THE COMMON TAGS
   453                          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
   454  001704  012706  001100          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
   455  001710  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
   456  001712  022706  001140          CMP     #SWR,R6 ;;DONE?
   457  001716  001374                  BNE     .-6             ;;LOOP BACK IF NO
   458  001720  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
   459                          ;;INITIALIZE A FEW VECTORS
   460  001724  012737  017522  000020  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
   461  001732  012737  000340  000022  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
   462  001740  012737  016670  000030  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
   463  001746  012737  000340  000032  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
   464  001754  012737  021056  000034  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
   465  001762  012737  000340  000036  MOV     #340,@#TRAPVEC+2;LEVEL 7
   466  001770  012737  017350  000024  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
   467  001776  012737  000340  000026  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
   468  002004  005037  001160          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
   469  002010  012737  002010  001106  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
```

```
                                        SWRSU
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002016  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002022  012737  002056  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002030  012737  177570  001140  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002036  012737  177570  001142  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002044  022777  177777  177066  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002052  001012                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
002054  000403                  BR      65$             ;;BRANCH IF NO TIMEOUT
002056  012716  002064  64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002062  000002                  RTI
002064  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002072  012737  000174  001142          MOV     #DISPREG,DISPLAY
002100  012637  000004  66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002104  012737  176543  016434  MOV     #176543,$HINUM
002112  012737  123456  016436  MOV     #123456,$LONUM
002120  005037  001654          CLR     CONNT

                                ;
                                ;FILL LOCATIONS "214-1000" WITH .+2, IOT
                                ;

002124  012700  000214  FILVEC: MOV     #214,R0
002130  012701  000216          MOV     #216,R1
002134  010120          98$:    MOV     R1,(R0)+
002136  012720  000004          MOV     #4,(R0)+
002142  022121                  CMP     (R1)+,(R1)+
002144  022700  001000          CMP     #1000,R0
002150  003371                  BGT     98$
002152  104401                  TYPE                    ;TYPE HEADER
002154  021176                  MHEAD
                                ;**
                                ;** SINCE ISC-11 IS CAPABLE OF DIFFERENT CONTROL REGISTERS
                                ;** ASK OPERATOR FOR FILE BOX AND INTERRUPT VECTOR ADDRESS
                                ;**

002156  012737  002164  001700  MOV     #10$,CTLLOC
002164  104401  026653  10$:    TYPE, MFILE             ;ASK FOR FILE BOX
002170  012737  000001  002206  MOV     #1,22$
002176  000401                  BR      .+4
002200  000771                  BR      10$
002202  104414                  INOCT
002204  001672                  TMPFIL
002206  000001  22$:    1
002210  022737  000013  001672  CMP     #13,TMPFIL
002216  003003                  BGE     11$
002220  104401  026740          TYPE, ILLEG             ;ILLEGAL FILE BOX... RE-ASK
002224  000757                  BR      10$
002226  012737  171000  001702  11$:    MOV     #171000,ICSLMT          ;INIT. MODULE ADDRESSES
002234  012737  171776  001614          MOV     #171776,ICSR            ;INIT. ICSR
002242  005237  001672          INC     TMPFIL
002246  005337  001672  19$:    DEC     TMPFIL                  ;DEC. FILE
002252  001407                  BEQ     18$                     ;BRANCH IF FOUND
```

```
                                    SUB     #10,ICSR                ;NEXT ICSR
                                    ADD     #40,ICSLMT              ;NEXT GROUP OF MODULE ADDRESSES
                                    BR      19$                     ;LOOP
002272  013737  001614  001616  18$:  MOV     ICSR,ICAR               ;CREATE ICAR
002300  162737  000002  001616        SUB     #2,ICAR
002306  013737  001702  001676        MOV     ICSLMT,ICSHGH           ;SET UPPER ADDRESS LIMIT
002314  062737  000040  001676        ADD     #40,ICSHGH
002322  012737  002336  000004        MOV     #12$,@#4
002330  017700  177260                MOV     @ICSR,R0
002334  000404                      BR      97$
002336  022626                12$:  POPSP2
002340  104401  027154              TYPE,   NONXST
002344  000707                      BR      10$
002346  012737  000006  000004  97$:  MOV     #6,@#4
002354  012737  002362  001700        MOV     #99$,CTLLOC
002362  104401  026707          99$:  TYPE, MVECT                     ;ASK FOR INTERRUPT VECTOR
002366  012737  000001  002404        MOV     #1,23$
002374  000401                      BR      .+4
002376  000771                      BR      99$
002400  104414                      INOCT
002402  001674                      TMPVEC
002404  000001                23$:  1
002406  022737  000776  001674        CMP     #776,TMPVEC
002414  002404                      BLT     16$
002416  022737  000234  001674        CMP     #234,TMPVEC
002424  003403                      BLE     17$
002426  104401  026740          16$:  TYPE, ILLEG                     ;NOT IN FLOATING AREA
002432  000753                      BR      99$
002434  013737  001674  001620  17$:  MOV     TMPVEC,ICSVT
002442  062737  000002  001674        ADD     #2,TMPVEC
002450  013737  001674  001622        MOV     TMPVEC,ICSVT2           ;STORE VECTOR ADDRESS
                                ;**
                                ;**VERIFY THAT VECTOR GIVEN IS ACTUAL VECTOR BEFORE GOING
                                ;**ANY FARTHER
                                ;
002456  013737  000020  002624        MOV     @#20,93$                ;SAVE LOC. 20
002464  012737  002540  000020        MOV     #95$,@#20               ;SET UP IOT TRAP VECTOR
002472  012777  002606  177120        MOV     #94$,@ICSVT             ;SET UP INTERRUPT ROUTINE
002500  012777  000340  177114        MOV     #340,@ICSVT2            ;PS PICKUP
002506  012737  000240  177776        MOV     #240,@#PS               ;SET PRIORITY TO 5
002514  012777  000404  177072        MOV     #404,@ICSR              ;SET MAINT. TO INTERRUPT
002522  000240                      NOP
002524  104401  027033              TYPE,   NOINT                   ;NO INTERRUPT
002530  013737  002624  000020        MOV     93$,@#20                ;RESTORE LOC 20
002536  000413                      BR      96$                     ;RE-ASK QUESTION
002540  011605                95$:  MOV     (SP),R5                 ;INVALID VECTOR ADDRESS PRINT
002542  024545                      CMP     -(R5),-(R5)             ;SUB 4 FROM R5
002544  022626                      POPSP2                          ;POP STACK
002546  000240                      NOP
002550  104401  027076              TYPE,   FILINT                  ;PRINT WHERE VECTOR
002554                              TYPOCT  R5
002554  010546                      MOV     R5,-(SP)            ;;SAVE R5 FOR TYPEOUT
002556  104402                      TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
002560  104401  027131              TYPE,   CKJMP
002564  022626                      POPSP2
```

```
                                965$:   CLR     @ICSR
                                        MOV     93$,@#20
                                        HALT
                                        JMP     99$
                                94$:    MOV     93$,@#20                ;INTERRUPT OK, RESTORE 20
                                        POPSP2                          ;POP STACK
                                        CLR     @ICSR
                                        BR      .+4
                                93$:    0
                                ;
                                ;FILL LOCATIONS "214-1000" WITH .+2, HALT
                                ;

                                        MOV     #START1,CTLLOC
                                FILHLT: MOV     #214,R0
                                        MOV     #216,R1
                                90$:    MOV     R1,(R0)+
                                        MOV     #0,(R0)+
                                        CMP     (R1)+,(R1)+
                                        CMP     #1000,R0
                                        BGT     90$
                                        MOV     #KEYSRV,@#TKVEC
                                        MOV     #340,@#TKVEC+2
                                        BIS     #100,@#TKS
                                        MOV     #99$,@#4
                                        INC     LPAV
                                        TST     @LPCSR
                                        TYPE
                                        $CRLF
                                        TYPE
                                        MLPAV
                                        BR      8$
                                99$:    CLR     LPAV
                                        POPSP2


                                8$:     MOV     #3$,@#4         ;DETERMINE CORE SIZE.
                                        CLR     R0
                                2$:     TST     (0)+
                                        BR      2$
                                3$:     TST     -(0)
                                        MOV     R0,CORSIZ
                                        BPL     4$
                                        MOV     #077776,CORSIZ
                                4$:     MOV     #6,@#4
                                        MOV     FR1120,FR1              ;SET FREQ FOR 11/20
                                        MOV     #RTI,@#12               ;FIND OUT IF 11/20 OR 11/45
                                        SEV                             ;IF 11/45 THEN DELAY ITERATIONS
                                        XOR     %1,%1                   ;MUST BE INCREASED.
                                        BVS     5$
                                        ADD     FR1120,FR1              ;FLOW FELL THUR TO HERE -
                                        ADD     FR1120,FR1              ;IT MUST BE AN 11/45 OR EQUIV.
                                        MOV     @#4,-(SP)               ;SAVE LOC 4
                                        MOV     @#6,-(SP)               ;SAVE LOC 6
                                        MOV     #99$,@#4                ;SETUP FOR TRAP
                                        TST     @#177760                ;CHECK IF 11/70
```

```
   638  003262  006337  001576            ASL     FR1             ;11/70, DOUBLE
   639  003266  000401                    BR      .+4             ;SKIP
   640  003270  022626                99$: POPSP2                  ;RESTORE STACK
   641  003272  012637  000006            MOV     (SP)+,@#6       ;RESTORE LOC 6
   642  003276  012637  000004            MOV     (SP)+,@#4       ;RESTORE LOC 4
   643  003302  005037  000012        5$: CLR     @#12            ;RESTORE LOC 12.
   644  003306  013737  001576  001564    MOV     FR1,FR3         ;NOW WE MUST SET THE REST OF
   645  003314  006137  001564            ROL     FR3             ;OF THE DELAY TIMES UP.
   646  003320  013737  001564  001562    MOV FR3,FR5
   647  003326  063737  001576  001564    ADD     FR1,FR3
   648  003334  063737  001564  001562    ADD FR3,FR5
   649  003342  013737  001562  001560    MOV     FR5,FR16
   650  003350  006137  001560            ROL     FR16
   651  003354  063737  001562  001560    ADD     FR5,FR16
   652  003362  013737  001560  001556    MOV     FR16,FR20
   653  003370  063737  001576  001560    ADD     FR1,FR16
   654  003376  063737  001562  001556    ADD     FR5,FR20
   655  003404  013737  001556  001552    MOV     FR20,FR50
   656  003412  006137  001552            ROL     FR50
   657  003416  063737  001562  001552    ADD     FR5,FR50
   658  003424  063737  001562  001552    ADD     FR5,FR50
   659  003432  013737  001552  001550    MOV     FR50,FR110
   660  003440  006137  001550            ROL     FR110
   661  003444  063737  001562  001550    ADD     FR5,FR110
   662  003452  063737  001562  001550    ADD     FR5,FR110
   663  003460  013737  001556  001554    MOV     FR20,FR40
   664  003466  006137  001554            ROL     FR40
   665  003472  005037  001652            CLR     EXPERT
   666
   667
   668  003276  012737  001564  001566        MOV     #FR3,   FREQ
   669  003304  005037  001604            CLR     PATRN
   670  003310  005737  001646            TST     ST200
   671  003314  001406                    BEQ     7$
   672  003316  013737  001150  001666    MOV     $TPS,TPCSR
   673  003324  013737  001152  001670    MOV     $TPB,TPDBR
   674  003332  012700  013176        7$: MOV     #INADR, R0      ;CLEAR ADDR AREA
   675  003336  005037  001646            CLR     ST200           ;INDICATE START AT LOC 200.
   676  003342  005020                1$: CLR     (0)+
   677  003344  020027  013344            CMP     R0,#OUTS
   678  003350  001374                    BNE     1$
   679  003352  005037  001574            CLR     FREQ3
   680  003356  005037  001572            CLR     FREQ2
   681  003362  005037  001570            CLR     FREQ1
   682  003366  013737  001564  001566    MOV     FR3,FREQ
   683
   684  003374  012706  001100        START1: MOV     #1100,SP
   685  003400  005077  176210            CLR     @ICSR
   686  003404  005037  001100            CLR     $PASS
   687  003410  005037  001660            CLR     LINEPR
   688  003414  013737  001566  013134    MOV     FREQ,RTEMP
   689  003422  005737  001646            TST     ST200           ;HAS PROG. BEEN EVER STARTED
   690  003426  001411                    BEQ     3$              ;AT ADDR. 200? IF SO CONTINUE.
   691  003430  012737  020636  000034    MOV     #$TYPE,@#34     ;SET UP TO PRINT.
   692  003436  005037  000036            CLR     @#36
   693  003442  104401                    TYPE
```

```
                                                MSTERR
003444  022371                                  JMP     START           ;PROGRAM MUST BE STARTED AT ADDR. 200.
003446  000137  001704                          JMP     START           ;PROGRAM MUST BE STARTED AT ADDR. 200.
003452  013737  001550  001566  3$:             MOV     FR110,  FREQ    ;SET DELAY FOR TTY SETTLE TIME.
003460  005037  001570                          CLR     FREQ1
003464  013737  001666  001150                  MOV     TPCSR,$TPS
003472  013737  001670  001152                  MOV     TPDBR,$TPB
003500  104422                                  DELAY                   ;CAUSE A DELAY.
003502  000005                                  RESET                   ;ISSUE SYSTEM INITIALIZE.
003504  104422                                  DELAY                   ;TTY SETTLE DOWN TIME.
003506  012737  000340  177776                  MOV     #340,PS
003514  022737  000176  001140                  CMP     #176,SWR        ;SOFTWARE SWR IN USE?
003522  001001                                  BNE     99$             ;IF NOT, THEN BRANCH
003524  104405                                  GTSWR                   ;GO GET NEW SWR VALUE VIA TTY
003526  104401                          99$:    TYPE                    ;TYPE "TEST NO ?"
003530  022124                                  MTN
003532  012737  003650  003554                  MOV     #TSTNO, 1$
003540  012737  000001  003556                  MOV     #1,     1$+2
003546  000401                                  BR      .+4
003550  000766                                  BR      99$
003552  104414                                  INOCT                   ;GET TEST NUMBER
003554  003650                          1$:     TSTNO
003556  000001                                  1
003560  005737  003650                          TST     TSTNO
003564  100404                                  BMI     4$
003566  023727  003650  000007                  CMP     TSTNO,  #7.     ;LEGAL NUMBER?
003574  003403                                  BLE     2$
003576  104401                          4$:     TYPE                    ;NO-TYPE "NO SUCH TEST"
003600  022147                                  MTNL
003602  000674                                  BR      START1
003604  013737  003650  013134  2$:             MOV     TSTNO,RTEMP
003612  006337  013134                          ASL     RTEMP
003616  062737  003752  013134                  ADD     #TSTLST, RTEMP
003624  017737  007304  013134                  MOV     @RTEMP,RTEMP
003632  012777  000100  175304                  MOV     #100,@$TKS      ;ENABLE TTY TO INTERRUPT.
003640  005037  177776                          CLR     PS
003644  000177  007264                          JMP     @RTEMP          ;GOTO TEST

003650  000000                          TSTNO:  0                       ;TEST NUMBER
003652  003744                          TSTLST: TST0                    ;INPUT AND OUTPUT MODULE EXERCISER
003654  005322                                  TST1                    ;INPUT OR OUTPUT MODULE SIMPLE EXER.
003656  005604                                  TST2                    ;DAC CALIBRATION TEST
003660  005676                                  TST3                    ;DAC INTERACTION TEST
003662  005746                                  TST4                    ;COUNTER MODULE TEST
003664  007146                                  TST5                    ;A/D LOGIC TEST
003666  010026                                  TST6                    ;A/D CALIBRATION TEST
003670  011000                                  TST7                    ;A/D REPEATIBILITY TEST


                                                ;*
                                                ;*KEYBOARD INTERRUPT HANDLER
                                                ;*
003672  017737  175250  003742  KEYSRV: MOV     @$TKB,2$
003700  042737  177600  003742                  BIC     #177600,2$
003706  122737  000003  003742                  CMPB    #3,2$
003714  001410                                  BEQ     1$
003716  122737  000007  003742                  CMPB    #7,2$                   ;<CONTROL G> TYPED?
```

```
MAINDEC-11-DZICA-D     MACY11 30(1046)  02-AUG-77  09:23  PAGE 15
DZICAD.P11    28-JUL-77 14:52          INITIALIZE THE COMMON TAGS                                    SEQ 0030

   750  003724  001003                          BNE     3$                      ;IF NOT, THEN EXIT!
   751  003726  104401                          TYPE                            ;ALLOW <CONTROL G>
   752  003730  020431                          $CNTLG                          ;TO BE TYPED-OUT
   753  003732  104405                          GTSWR                           ;GO GET SWR VALUE VIA TTY
   754  003734  000002                  3$:     EXIT
   755  003736  000137  003374          1$:     JMP     START1
   756  003742  000000                  2$:     0
   757
   758
```

```
                                        ;*
                                        ;*TEST 0 INPUT AND OUTPUT MODULE EXERCISER
                                        ;*
003744  104401                  TST0:   TYPE                    ;TYPE HEADER.
003746  025204                          MHTO
003750  104415                          INAR                    ;GET INPUT MODULE ADDRS.
003752  104416                          OUTAR                   ;GET OUTPUT MODULE ADDRS.
003754  104417                          PATAR                   ;GET PATTERN (OR USE DEFALT).
003756  104420                          DELAR                   ;GET DELAY TIME (OR USE DEFALT).
003760  005037  001100                  CLR     $PASS
003764  005037  001634                  CLR     HEADER
                                                                ;INITIAL SETUP.
003770  012777  004444  175622          MOV     #TOINT,@ICSVT   ;SET UP FOR ICS INTERRUPTS.
003776  012777  004370  175620          MOV     #TOPINT,@TPVCT  ;SET UP FOR TTY INTERRUPTS.
004004  012777  000340  175610          MOV     #340,@ICSVT2    ;SET CPU PRIORITY TO 6 ON ICS INTR.
004012  012777  000200  175606          MOV     #200,@TPVCT2    ;SET CPU PRIORITY TO 4 ON TTY INTR.
004020  005037  001632                  CLR     TPBSY           ;SET NO TTY OUTPUT IN PROGRESS
004024  012700  013344                  MOV     #OUTS,R0        ;CLEAR TEMP STORAGE AREA OF INPUT
004030  005020                  1$:     CLR     (0)+            ;MODULE DATA CHANGE.
004032  020027  013404                  CMP     R0,#OUTSE
004036  001374                          BNE     1$
004040  005737  001654                  TST     CONNT
004044  001002                          BNE     2$
004046  104401                          TYPE                    ;TYPE HEADER.
004050  021360                          MTOH
004052  005037  001636          2$:     CLR     TOADR           ;CLEAR TYPE OUT CONSTANTS.
004056  005037  001640                  CLR     TODAT
004062  005037  001642                  CLR     TOGEN
                                                                ;TAKE CARE OF OUTPUT ADDRS.
004066  104422                          DELAY
004070  012737  004070  001106  TOOUTR: MOV     #TOOUTR,$LPADR
004076  104421                          DELAY2
004100  012702  013176                  MOV     #INADR,R2
004104  052777  000004  175502          BIS     #4,@ICSR        ;ALLOW ICS TO INTR.
004112  005037  177776                  CLR     PS              ;LET CPU ALLOW INTRS.
004116  012704  013344                  MOV     #OUTS,R4
004122  012700  013236          TOOUT:  MOV     #OUTADR,R0      ;GET OUTPUT MODULE LIST.
004126  012001                  1$:     MOV     (0)+,R1         ;GET FIRST ADDR
004130  001405                          BEQ     TOIN            ;IF N ADDR. - EXIT.
004132  013711  001604                  MOV     PATRN,(1)       ;SEND PATTERN TO OUTPUT MODULE.
004136  020027  013276                  CMP     R0,#CNTADR
004142  001371                          BNE     1$
                                                                ;TAKE CARE OF INPUT ADDRS.
004144  005737  001654          TOIN:   TST     CONNT           ;MODULES CONNECTED?
004150  001407                          BEQ     2$              ;IF NOT NORMAL CONTINUE.
004152  012701  013344                  MOV     #OUTS,R1        ;IF SO COS AREA = CURRENT PATTERN.
004156  013721  001604          3$:     MOV     PATRN,(1)+
004162  020127  013404                  CMP     R1,#OUTSE
004166  001373                          BNE     3$
004170  013737  001604  001610  2$:     MOV     PATRN,PATJOY
004176  104422                          DELAY                   ;DELAY TIME
004200  000240                          NOP
004202  005737  001632                  TST     TPBSY
004206  001374                          BNE     .-6
```

```
                                        1$:     MOV     (2),R1          ;PICK UP FIRST ADDR.
                                                BNE     .+6
                                                JMP     TOLOP
                                                CMP     (1),(4)         ;DATA CHANGED?
                                                BNE     TOIN2           ;IF YES TAKE CARE OF IT
                                        TOIN1=.
                                                TST     (2)+
                                                TST     (4)+            ;UPDATE COD POINTER
                                                CMP     R2,#OUTADR      ;DONE ALL INPUT MODULES?
                                                BNE     1$
                                                JMP     TOLOP

                                        TOIN2:          TST     CONNT
                                                BEQ     1$
                                        2$:     TST     TPBSY
                                                BNE     2$
                                                MOV     R1,@BDADR
                                                MOV     PATJOY,@GDDAT
                                                MOV     (1),@BDDAT
                                                TST     HEADER
                                                BEQ     3$
                                                ERROR   31
                                                JMP     TOLOP

                                        3$:     ERROR   30
                                                INC     HEADER
                                                JMP     TOLOP

                                        1$:     TST     TPBSY
                                                BNE     1$
                                                MOV     #340,PS
                                                MOV     R1,TOADR        ;GET ADDR. OF INPUT MODULE
                                                MOV     (1),TODAT       ;GET CHANGE DATA
                                                CLR     TOGEN           ;NO GEN CODE (NO INTERRUPT)
                                                INC     TPBSY           ;SET TTY BUSY.
                                                MOV     (1),(4)         ;RECORD NEW DATA
                                                FOCTA                   ;FORM INFO INTO AN ASCIZ MESSAG.
                                                CLR     -(6)
                                                MOV     #TOIN1,-(6)

                                        TOPINT: TSTB    @STPS           ;PRINTER BUSY?
                                                BPL     TOPINT
                                                BIT     #020000,@SWR    ;INHIBIT TYPEOUT?
                                                BEQ     3$              ;NO CONTINUE.
                                                CLR     TPBSY           ;YES-STOP TYPEOUT
                                                EXIT
                                        3$:
                                                MOVB    (3)+,@STPB      ;SEND CHAR.
                                                BEQ     1$              ;IF END GO TO END
                                                BIS     #100,@STPS      ;MAKE SURE TTY CAN INTR.
                                                EXIT                    ;EXIT

                                        1$:     CLR     @STPS           ;TTY SETTLE DOWN TIME
                                                CLR     TPBSY           ;CLEAR BUSY

                                                EXIT
```

```
                                        ;*
                                        ;*ICS INTERRUPT SERVICE ROUTINE
                                        ;*
004444  017737  175146  004666  TOINT:  MOV     @ICAR,6$        ;GET ADDR. OF INTRING MOD.
004452  006137  004666                  ROL     6$              ;FORM REAL ADDR.
004456  042737  177001  004666          BIC     #177001,6$
004464  053737  001612  004666          BIS     ICSMOD,6$
004472  017737  000170  004670          MOV     @6$,7$          ;FETCH DATA.
004500  005737  001632          10$:    TST     TPBSY           ;TYPEOUT PENDING?
004504  001406                          BEQ     5$              ;IF NOT CONTINUE.
004506  042777  000004  175100          BIC     #4,     @ICSR   ;STOP ICS FROM INTR.
004514  005037  177776                  CLR     PS              ;ALLOW INTERRUPTS FROM TTY.
004520  000767                          BR      10$             ;LOOP.
004522  012737  000300  177776  5$:     MOV     #300,PS         ;LOCK OUT INTR.S
004530  052777  000005  175056          BIS     #5,@ICSR        ;RE-ENABLE INTRS.AND SET RIF.
004536  017737  175054  001642          MOV     @ICAR,TOGEN     ;GET GENERIC CODE
004544  013737  004666  001636          MOV     6$,TOADR
004552  005777  175060                  TST     @TOADR
004556  013737  004670  001640          MOV     7$,TODAT
004564  005737  001654                  TST     CONNT
004570  001401                          BEQ     9$
004572  000002                          EXIT

004574  000337  001642          9$:     SWAB    TOGEN           ;RIGHT JUSTIFY GEN CODE.
004600  042737  177760  001642          BIC     #177760,TOGEN
004606  010037  004672                  MOV     R0,SAV0 ;SAVE R0
004612  012700  013176                  MOV     #INADR,R0       ;FIND OFFSET OF INPUT MODULE
004616  022037  001636          2$:     CMP     (0)+,TOADR      ;THAT INTERRUPTED.
004622  001404                          BEQ     3$
004624  020027  013236                  CMP     R0,#OUTADR      ;IF NOT ENTERED INPUT MODULE THAT
004630  001407                          BEQ     4$              ;THAN ADDR WILL NOT BE IN TABLE
004632  000771                          BR      2$              ;IN THAT CASE DON'T WORRY ABOUT IT.
004634  162700  013200          3$:     SUB     #INADR+2,R0     ;SUB TO GET OFFSET.
004640  062700  013344                  ADD     #OUTS,R0        ;ADD STORAGE OF COS TO RECORD
004644  013710  001640                  MOV     TODAT,(0)       ;NEW DATA FOR THAT MODULE.
004650  013700  004672          4$:     MOV     SAV0,R0 ;RESTORE R0.
004654  104430                          FOCTA                   ;NO-FORM ASCIZ STRING.
004656  005237  001632                  INC     TPBSY           ;SET OUTPUT BUSY
004662  000642                          BR      TOPINT          ;START OUTPUT
004664                          1$:
004664  000002                          EXIT

004666  000000          6$:     0
004670  000000          7$:     0
004672  000000          SAV0:   0
004674  005737  001632  TOLOP:  TST     TPBSY
004700  001375                  BNE     TOLOP
004702  000004                  SCOPE
004704  104423                  CPATR
004706  000137  004070          JMP     TOOUTR

                                        ;*
                                        ;*ROUTINE TO CONVERT 3 OCTAL NUMBERS TO AN ASCIZ STRING
```

```
                                        ;* CALL= FOCTA
                                        ;
004712  012703  027206          ROCTA:  MOV     #OUTBF,R3       ;SET UP BUFFER
004716  013737  001636  005004          MOV     TOADR,PACK1
004724  004737  005010                  JSR     PC,PACK         ;PACK ADDR.
004730  013737  001640  005004          MOV     TODAT,PACK1
004736  004737  005010                  JSR     PC,PACK         ;PACK DATA
004742  013737  001642  005004          MOV     TOGEN,PACK1
004750  001402                          BEQ     1$
004752  004737  005010                  JSR     PC,PACK         ;PACK GEN CODE (IF ANY)
004756  112723  000001          1$:     MOVB    #1,(3)+         ;FILLER CHARACTERS.
004762  112723  000001                  MOVB    #1,(3)+
004766  112723  000001                  MOVB    #1,(3)+
004772  105023                          CLRB    (3)+            ;STRING TERMINATOR.
004774  105023                          CLRB    (3)+
004776  012703  027204                  MOV     #OUTBF1,R3      ;RESET POINTER
005002  000002                          EXIT
                                        ;
005004  000000                  PACK1:  0
005006  000000                  PACK2:  0
                                        ;
005010  012737  000260  005006  PACK:   MOV     #260,PACK2
005016  005737  005004                  TST     PACK1
005022  100002                          BPL     .+6
005024  005237  005006                  INC     PACK2
005030  113723  005006                  MOVB    PACK2,(3)+
005034  000337  005004                  SWAB    PACK1
005040  013737  005004  005006          MOV     PACK1,PACK2
005046  006037  005006                  ROR     PACK2
005052  006037  005006                  ROR     PACK2
005056  006037  005006                  ROR     PACK2
005062  006037  005006                  ROR     PACK2
005066  042737  177770  005006          BIC     #177770,PACK2
005074  052737  000260  005006          BIS     #260,PACK2
005102  113723  005006                  MOVB    PACK2,(3)+
005106  013737  005004  005006          MOV     PACK1,PACK2
005114  006037  005006                  ROR     PACK2
005120  042737  177770  005006          BIC     #177770,PACK2
005126  052737  000260  005006          BIS     #260,PACK2
005134  113723  005006                  MOVB    PACK2,(3)+
005140  000337  005004                  SWAB    PACK1
005144  013737  005004  005006          MOV     PACK1,PACK2
005152  006037  005006                  ROR     PACK2
005156  006037  005006                  ROR     PACK2
005162  006037  005006                  ROR     PACK2
005166  006037  005006                  ROR     PACK2
005172  006037  005006                  ROR     PACK2
005176  006037  005006                  ROR     PACK2
005202  042737  177770  005006          BIC     #177770,PACK2
005210  052737  000260  005006          BIS     #260,PACK2
005216  113723  005006                  MOVB    PACK2,(3)+
005222  013737  005004  005006          MOV     PACK1,PACK2
005230  006037  005006                  ROR     PACK2
005234  006037  005006                  ROR     PACK2
005240  006037  005006                  ROR     PACK2
005244  042737  177770  005006          BIC     #177770,PACK2
```

```
   983  005252  052737  000260  005006          BIS     #260,PACK2
   984  005260  113723  005006                  MOVB    PACK2,(3)+
   985  005264  042737  177770  005004          BIC     #177770,PACK1
   986  005272  052737  000260  005004          BIS     #260,PACK1
   987  005300  113723  005004                  MOVB    PACK1,(3)+
   988  005304  112723  000240                  MOVB    #240,(3)+
   989  005310  112723  000240                  MOVB    #240,(3)+
   990  005314  000207                          RTS     PC
   991
   992  005316  000137  003374                  JMP     START1          ;EXIT
   993
```

```
                                        ;*
                                        ;*TEST 1 INPUT OR OUTPUT MODULE EXERCISER
                                        ;*

005322  104401                  TST1:  TYPE                    ;TYPE HEADER.
005324  025323                         MHT1
005326  005077  173612                 CLR     @STKS          ;DON'T ALLOW TTY INTERRUPTS
005332  005737  001652           5$:   TST     EXPERT         ;EXPERT MODE?
005336  001002                         BNE     .+6
005340  104401                         TYPE                   ;ASK "INPUT OR OUTPUT MODULE?"
005342  022166                         MI00
005344  104401                         TYPE
005346  022366                         MQ
005350  005037  015046                 CLR     CHAR
005354  105777  173564           1$:   TSTB    @STKS          ;WAIT FOR RESPONSE
005360  100375                         BPL     1$
005362  105777  173562           2$:   TSTB    @STPS          ;PRINTER BUSY?
005366  100375                         BPL     2$
005370  117777  173552  173554         MOVB    @STKB,@STPB    ;ECHO CHARACTER.
005376  117737  173544  013134         MOVB    @STKB,RTEMP
005404  142737  000240  013134         BICB    #240,RTEMP
005412  123727  013134  000015         CMPB    RTEMP,#15
005420  001412                         BEQ     4$
005422  123727  013134  000003         CMPB    RTEMP,#3       ;WAS ^C TYPED?
005430  001002                         BNE     .+6
005432  000137  003374                 JMP     START1         ;IF SO GOTO MONITR.
005436  113737  013134  015046         MOVB    RTEMP,CHAR
005444  000743                         BR      1$
005446  104401                  4$:    TYPE
005450  001163                         $CRLF
005452  122737  000117  015046         CMPB    #'O,    CHAR   ;DID HE TYPE "O"?
005460  001433                         BEQ     TST10          ;IF SO-GOTO OUTPUT ROUTINE
005462  122737  000111  015046         CMPB    #'I,    CHAR   ;DID HE TYPE "I"?
005470  001320                         BNE     5$             ;IF NOT RETYPE QUESTION

                                        ;*ROUTINE TO HANDLE INPUTTING FROM INPUT MODULE TO DISPLAY

005472  104415                         INAR                   ;GET INPUT MODULE ADDR.
005474  104401  024775                 TYPE,   MSWO
005500  104401  022004                 TYPE,   MWK

005504  012777  000100  173432         MOV     #100,@STKS     ;ALLOW TTY INTERRUPTS
005512  017700  005460           3$:   MOV     @INADR, R0     ;GET DATA FROM INPUT MODULE
005516  010037  001142                 MOV     R0, DISPLAY    ;PUT IN DISPLAY REGISTER IF 11/45.
005522  022737  000176  001140         CMP     #176,SWR       ;SOFTWARE SWR IN USE?
005530  001401                         BEQ     6$             ;BRANCH IF YES
005532  104427                         INITR                  ;RESET TO DISPLAY.
005534  032777  000001  173376   6$:   BIT     #1,     @SWR   ;IF BIT1 SET THEN GOTO MONITR.
005542  001763                         BEQ     3$             ;NOT SET-THEN LOOP.
005544  000137  003374                 JMP     START1

                                        ;*ROUTINE TO HANDLE OUTPUTTING FROM SWITCH REGISTER TO OUTPUT MODULE

005550  104416                  TST10: OUTAR                  ;GET OUTPUT MODULE ADDR.
005552  104420                         DELAR                  ;GET DELAY TIME (DEFAULT=3MS).
005554  005037  177776                 CLR     PS
```

```
1050  005560  104401  022004                    TYPE,   MWK
1051  005564  012777  000100  173352            MOV     #100,@#TKS      ;ALLOW TTY INTERRUPTS
1052  005572  017777  173342  005436  1$:       MOV     @SWR,@OUTADR    ;SEND DATA FRO THE SWR TO OUTPUT MOD.
1053  005600  104422                            DELAY
1054  005602  000773                            BR      1$
1055
```

```
1056                                            ;*
1057                                            ;*TEST 2 DAC CALIBRATION ROUTINE
1058                                            ;*
1059
1060  005604  104401                TST2:   TYPE                    ;TYPE HEADER.
1061  005606  025401                        MHT2
1062  005610  104424                        IDAC                    ;INPUT DAC ADDR.
1063  005612  005737  013336               TST     DACADR          ;ANY DACS PRESENT
1064  005616  001005                        BNE     4$
1065  005620  104401                        TYPE                    ;MESSAGE "NO DAC ADDR. IN BUFFER.
1066  005622  021436                        MNDA
1067  005624  000137  005604               JMP     TST2
1070  005630  000000                2$:     0
1071  005632  104401  022004        4$:     TYPE,   MWK
1072  005636  017737  173276  005630 1$:     MOV     @SWR,2$         ;GET VALUE OF SWR
1073  005644  042737  140017  005630        BIC     #140017, 2$
1074  005652  013777  005630  005456 3$:     MOV     2$,     @DACADR ;SEND VALUE.
1075  005660  062737  040000  005630        ADD     #40000, 2$      ;SET FOR NEXT CH.
1076  005666  103371                        BCC     3$              ;DONE ALL CHS?
1077  005670  000762                        BR      1$
1078
1079  005672  000137  003374               JMP     START1
1080
```

```
1079                                    ;*
1080                                    ;*TEST 3 DAC INTERACTION TEST
1081                                    ;*OUTPUTS RAMP TO 4 CHANNELS OF DAC THAT ARE "OUT OF PHASE"
1082                                    ;*
1083
1084  005676  104401            TST3:   TYPE                    ;TYPE HEADER.
1085  005700  025436                    MHT3
1086  005702  104424                    IDAC                    ;GET DAC ADDRESS
1087  005704  005737  013336            TST     DACADR          ;DAC PRESENT?
1088  005710  001004                    BNE     1$
1089  005712  104401                    TYPE                    ;NO-MESSAGE "NO DAC ADDR. IN BUFFER"
1090  005714  021436                    MNDA
1091  005716  000137  005676            JMP     TST3
1092  005722  005000            1$:     CLR     R0
1093  005724  104401  022004            TYPE,   MWK
1094  005730  010077  005402    2$:     MOV     R0,@DACADR      ;SEND WORD TO DAC
1095  005734  062700  000020            ADD     #20,    R0
1096  005740  000773                    BR      2$
1097
1098  005742  000137  003374            JMP     START1
1099
```

```
                                        ;*
                                        ;*TEST 4 COUNTER MODULE TEST
                                        ;*TEST 1 TO 16 MODULES
                                        ;*
005746  104401                  TST4:   TYPE                    ;TYPE HEADER.
005750  025473                          MHT4
005752  104425                          CNTAR                   ;GET COUNTER MODULE ADDRS.
005754  013737  001554  001566          MOV     FR40,FREQ
005762  104401  022004                  TYPE,   MWK
005766  012700  013276          TST4L1: MOV     #CNTADR,R0      ;GET LIST OF MODULES
005772  010037  005004                  MOV     R0,PACK1
005776  005710                          TST     (0)             ;ANY ADDRS. ENTERED?
006000  001004                          BNE     TST4L
006002  104401  023355                  TYPE    ,MNAE           ;"NO ADDRS ENTERED".
006006  000137  005746                  JMP     TST4
006012  012777  000100  173124  TST4L:  MOV     #100,@#TKS              ;ALLOW TTY INTERRUPTS
006020  013700  005004                  MOV     PACK1,R0
006024  020027  013336                  CMP     R0,#DACADR      ;DONE ALL COUNTERS?
006030  001005                          BNE     2$
006032  005237  001100          1$:     INC     $PASS
006036  104401                          TYPE
006040  021465                          MEND
006042  000751                          BR      TST4L1
006044  012037  001122          2$:     MOV     (0)+,$BDADR     ;GET ADDR. OF COUNTER MODULE.
006050  001770                          BEQ     1$              ;NOTE: ZERO IF NO ADDR. ENTERED.
006052  010037  005004                  MOV     R0,PACK1
                                        ;*
                                        ;*TEST THAT THE COUNTER MODULE UNDER TEST CAN BE ADDRESSED
                                        ;*
006056  012737  006056  001106  TST4A:  MOV     #TST4A,$LPADR   ;SET FOR SCOPE, ITERATIONS.
006064  012777  077777  173030          MOV     #77777,@$BDADR  ;SEND PATTERN.
006072  017737  173024  001126          MOV     @$BDADR,$BDDAT  ;GET IT BACK
006100  001001                          BNE     TST4AL          ;IF NON-ZERO - WE GOT SOMETHING BACK.

006102  104011                          ERROR   11              ;COULD NOT MAKE ANY DATA XFER
006104  000004                  TST4AL: SCOPE
                                        ;*
                                        ;*TEST THAT COUNTER MODULE BITS 00 THRU 15 CAN BE SET AND CLEARED
                                        ;*
006106  012737  000001  013134  TST4B:  MOV     #1,RTEMP        ;SET "GOOD DATA"
006114  012737  000010  001160          MOV     #10,$TIMES
006122  012737  006122  001106  1$:     MOV     #1$,$LPADR      ;SET FOR ITERATIONS
006130  013737  013134  001124          MOV     RTEMP,$GDDAT
006136  013777  001124  172756          MOV     $GDDAT,@$BDADR  ;SEND PATTERN TO COUNTER MODULE.
006144  017737  172752  001126          MOV     @$BDADR,$BDDAT  ;GET IT BACK.
006152  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DATA SENT=DATA RECEIVED?
006160  001402                          BEQ     2$

006162  104012                          ERROR   12              ;NO - REPORT ERROR
```

```
1156  006164  000412                          BR      3$              ;LOOP
1157
1158  006166  043777  001124  172726  2$:     BIC     $GDDAT,@$BDADR  ;TRY CLEARING BIT.
1159  006174  017737  172722  001126          MOV     @$BDADR,$BDDAT  ;DID IT CLEAR?
1160  006202  001403                          BEQ     3$              ;YES-LOOP.
1161  006204  005037  001124                  CLR     $GDDAT
1162  006210  104012                          ERROR   12
1163
1164  006212  000004                  3$:     SCOPE
1165  006214  006137  013134                  ROL     RTEMP           ;SET TO NEXT BIT.
1166  006220  103340                          BCC     1$              ;CONTINUE TESTING IF ALL BITS NOT DONE.
1167
1168                                          ;*
1169                                          ;*TEST THAT THE COUNTER MODULE CAN COUNT THRU
1170                                          ;*EACH STATE
1171                                          ;*
1172
1173  006222  012737  007076  013134  TST4C:  MOV     #CNTPAT,RTEMP   ;GET ADDR. OF PATTERNS
1174  006230  012737  000010  001160  1$:     MOV     #10,$TIMES
1175  006236  012737  006230  001106          MOV     #1$,$LPADR
1176
1177  006244  017777  004664  172650          MOV     @RTEMP,@$BDADR  ;SEND PATTERN.
1178  006252  005037  013136                  CLR     RTEMP1
1179  006256  017737  004652  001124          MOV     @RTEMP,$GDDAT
1180  006264  005237  001124                  INC     $GDDAT
1181  006270  005237  013136          4$:     INC     RTEMP1          ;WATCH TO SEE IF COUNTER COUNTS.
1182  006274  001413                          BEQ     5$              ;IF NO COUNT BY OVERFLOW-ERROR!
1183  006276  017737  172620  001126          MOV     @$BDADR,$BDDAT
1184  006304  027737  004624  001126          CMP     @RTEMP,$BDDAT
1185  006312  001766                          BEQ     4$
1186  006314  023737  001124  001126          CMP     $GDDAT,$BDDAT
1187  006322  001401                          BEQ     2$
1188
1189  006324  104013                  5$:     ERROR   13
1190  006326  000004                  2$:     SCOPE
1191  006330  062737  000002  013134          ADD     #2,RTEMP
1192  006336  023727  013134  007142          CMP     RTEMP,#CNTPAE
1193  006344  002731                          BLT     1$
1194
1195                                          ;*
1196                                          ;*TEST TO SEE IF COUNTER INITIALIZES PROPERLY--PART 1
1197                                          ;*
1198
1199  006346  012737  000010  001160  TST4D:  MOV     #10,$TIMES
1200  006354  012737  006354  001106  1$:     MOV     #1$,$LPADR
1201  006362  005077  172534                  CLR     @$BDADR
1202  006366  104427                          INITR
1203  006370  104422                          DELAY
1204  006372  005037  001124                  CLR     $GDDAT
1205  006376  017737  172520  001126          MOV     @$BDADR,$BDDAT
1206  006404  001401                          BEQ     2$
1207
1208  006406  104025                          ERROR   25
1209
1210  006410  000004                  2$:     SCOPE
1211
```

```
                                        ;*
                                        ;*TEST THAT THE COUNTER INITIALIZES PROPERLY PART 2
                                        ;*

006412  012737  000010  001160  TST4E:  MOV     #10,$TIMES      ;SET ITERATION COUNT
006420  012737  006432  001106          MOV     #1$,$LPADR      ;SET LOOP ADDRESS.
006426  005037  001124                  CLR     $GDDAT
006432  012777  177777  172462  1$:     MOV     #177777,@SBDADR ;SET THE COUNTER.
006440  000240                          NOP
006442  104427                          INITR                   ;SYSTEM INITIALIZE.
006444  017737  172452  001126          MOV     @SBDADR,SBDDAT  ;READ COUNTER.
006452  001401                          BEQ     2$

006454  104020                          ERROR   20

006456  000004                  2$:     SCOPE

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

```
                                        ;*
006460  105777  173130          TST4F:  TSTB    @ICSR           ;ANY INTERRUPTS POSTED?
006464  100001                  BPL     1$
                                
006466  104021                  ERROR   21              ;ILLEGAL INTERRUPT ON FCS BUS
                                
006470  000004          1$:     SCOPE
                                
006472  012777  006520  173120  MOV     #2$,@ICSVT      ;SET FOR INTERRUPT
006500  012777  000404  173106  MOV     #404,@ICSR      ;SET MAINT + INT ENABLE
006506  005037  177776          CLR     PS
006512  000240                  NOP
                                
006514  104022                  ERROR   22              ;FATAL ERROR - ICS DID NOT INTERRUPT
006516  000424                  BR      3$
                                
                                ;ICS INTERRUPTS TO HERE
                                
006520  017737  173072  001126  2$:     MOV     @ICAR,$BDDAT    ;READ ICAR.
006526  022626                  POPSP2                  ;READJUST STACK
006530  012777  000001  173056  MOV     #1,@ICSR
006536  005037  001124          CLR     $GDDAT
006542  005777  172354          TST     @$BDADR
006546  005037  177776          CLR     PS
006552  042737  000360  001126  BIC     #360,$BDDAT     ;IGNORE FILE BOX ADDR.
006560  005737  001126          TST     $BDDAT          ;NO OTHER ADDR. OR GEN BITS
006564  001401                  BEQ     3$              ;SHOULD SHOW UP
                                
006566  104023                  ERROR   23
                                
006570  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
                                ;*
006572  012777  006660  173020  TST4G:  MOV     #TST4GI,@ICSVT  ;SET UP FOR ICS INTERRUPT.
006600  012777  000340  173014  MOV     #340,@ICSVT2    ;PRIORITY 7 ON INTERRUPT.
006606  012737  006614  001106  MOV     #TST4GL,$LPADR  ;SET FOR ITERATIONS.
                                
006614  012737  000340  177776  TST4GL: MOV     #340,PS         ;DON'T ALLOW INTERRUPTS.
006622  012777  000004  172764  MOV     #4,@ICSR        ;ENABLE ICS TO INTERRUPT WHEN READY.
```

```
 1298  006630  012777  177777  172264          MOV     #177777,@SBDADR ;SET COUNTER TO ALL ONES.
 1299  006636  104422                          DELAY                   ;DELAY 40 MS.
 1300  006640  005037  177776                  CLR     PS              ;ALLOW INTERRUPTS.
 1301  006644  000240                          NOP
 1302  006646  005077  172742                  CLR     @ICSR
 1303  006652  104014                          ERROR   14              ;REPORT ERROR - COUNTER MODULE DIDN'T INTR.
 1304  006654  104427                          INITR
 1305  006656  000457                          BR      TST4GE          ;LOOP.
 1306
 1307                                          ;*RECEIVE ICS INTERRUPT HERE
 1308
 1309  006660  012706  001100          TST4GI: MOV     #1100,SP        ;RESET THE STACK POINTER.
 1310  006664  013737  172726  001126          MOV     @ICAR,$BDDAT    ;GET ADDR & GENERIC CODE.
 1311  006672  013737  001122  001124          MOV     $SBADR,$GDDAT   ;GET REAL ADDR.
 1312  006700  006037  001124                  ROR     $GDDAT          ;FORM ADDR. AS IT WOULD LOOK IN
 1313  006704  142737  177777  001125          BICB    #-1,$GDDAT+1    ;ICAR.
 1314  006712  005077  172676                  CLR     @ICSR
 1315  006716  005037  177776                  CLR     PS
 1316  006722  052737  003400  001124          BIS     #003400,$GDDAT  ;ADD GENERIC CODE.
 1317  006730  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CHECK ADDR. + GEN. CODE.
 1318  006736  001403                          BEQ     1$
 1319
 1320  006740  104015                          ERROR   15              ;ADDR OR GENERIC CODE INCORRECT.
 1321  006742  104427                          INITR
 1322  006744  000424                          BR      TST4GE          ;LOOP.
 1323
 1324  006746  104422          1$:             DELAY                   ;DELAY 40 MS.
 1325  006750  005037  001124                  CLR     $GDDAT
 1326  006754  017737  172142  001126          MOV     @$BDADR,$BDDAT
 1327  006762  001402                          BEQ     2$
 1328
 1329  006764  104016                          ERROR   16              ;COUNTER MODULE DIDN'T HALT ON OVERFLOW.
 1330  006766  000413                          BR      TST4GE          ;LOOP.
 1331
 1332  006770  052777  000001  172616  2$:     BIS     #1,@ICSR        ;SET RIF BIT IN ICS-11.
 1333  006776  005777  172120                  TST     @$BDADR         ;INITIATE RIF ON COUNTER MODULE.
 1334  007002  000240                          NOP                     ;
 1335  007004  105777  172604                  TSTB    @ICSR           ;ANY INTR. PENDING ON ICS BUS?
 1336  007010  100002                          BPL     TST4GE          ;NO-THEN LOOP.
 1337
 1338  007012  104017                          ERROR   17              ;RIF DIDN'T CLEAR INTERRUPT FLAG
 1339  007014  104427                          INITR
 1340  007016  012777  000001  172570  TST4GE: MOV     #1,@ICSR
 1341  007024  005777  172072                  TST     @$BDADR
 1342  007030  000004                          SCOPE
 1343
 1344                                          ;*
 1345                                          ;*TEST TO SEE IF RESET CLEARS INTERRUPT FLAG ON COUNTER
 1346                                          ;*
 1347
 1348  007032  012737  000010  001160  TST4H:  MOV     #10,$TIMES
 1349  007040  012777  177777  172054  1$:     MOV     #-1,@$BDADR     ;START COUNTER COUNTING
 1350  007046  012737  007040  001106          MOV     #1$,$LPADR
 1351  007054  104422                          DELAY                   ;DELAY FOR INTR.
 1352  007056  104427                          INITR                   ;SYSTEM INITIALIZE
 1353  007060  105777  172530                  TSTB    @ICSR           ;INTERRUPT POSTED?
```

```
1344  007064  100001                      BPL     2$
1345
1346  007066  104024                      ERROR   24              ;INIT FAILED TO CLEAR INTR. FLAG
1347  007070  000004              2$:     SCOPE
1348  007072  000137  006012              JMP     TST4L
1349
1350  007076  000000              CNTPAT: 0                       ;PATTERNS USED BY TST4C TO
1351  007100  000001                      1                       ;DETERMINE IF COUNTER MODULE
1352  007102  000003                      3                       ;CAN COUNT THRU EACH STATE.
1353  007104  000007                      7
1354  007106  000017                      17
1355  007110  000037                      37
1356  007112  000077                      77
1357  007114  000177                      177
1358  007116  000377                      377
1359  007120  000777                      777
1360  007122  001777                      1777
1361  007124  003777                      3777
1362  007126  007777                      7777
1363  007130  017777                      17777
1364  007132  037777                      37777
1365  007134  077777                      77777
1366  007136  177777                      177777
1367  007140  100000                      100000
1368  007142  000000              CNTPAE: 0
1369
1370
1371
1372
1373
1374
1375
```

```
                                        ;*
                                        ;*TEST 5 A/D LOGIC TEST
                                        ;*
007144  000000                  ADCSR:  0                       ;ADDR OF ADO5 UNDER TEST
007146  104401                  TST5:   TYPE                    ;TYPE HEADER.
007150  025534                          MHT5
007152  104426                          ADAR                    ;GET ADO5 ADDR.
007154  012746  007762          ADLOGL: MOV     #ADLOGE,-(6)
007160  104401  022004                  TYPE,   MWK
007164  013737  013340  007144  ADLOG:  MOV     ADADR,ADCSR     ;*GET ADO5'S ADDR.
007172  052777  000100  171744          BIS     #100,@$TKS
007200  005037  177776                  CLR     PS
007204  012737  007224  001106          MOV     #1$,$LPADR
007212  012703  007776                  MOV     #ADPATP,R3      ;*GET PATTERN POINTERS.
007216  012737  077770  001124          MOV     #077770,$GDDAT  ;*DATA TO BE SENT
007224  013777  001124  177712  1$:     MOV     $GDDAT,@ADCSR   ;*SEND DATA TO BE SENT
007232  017737  177706  001126          MOV     @ADCSR,$BDDAT   ;*GET DATA BACK.
007240  001001                          BNE     2$              ;*IF DATA PRESENT GO AHEAD

007242  104001                          ERROR   1               ;*ERROR "COULD NOT SEND /RECEIVE DATA"
                                                                ;*FROM ADO5
007244  000004                  2$:     SCOPE
                                        ;*BASIC "BIT BANG" OF ADO5

007246  011337  001124          ADLOG2: MOV     (3),$GDDAT      ;*GET PATTERN TO BE SENT.
007252  013777  001124  177664          MOV     $GDDAT,@ADCSR   ;*SEND PATTERN TO ADCSR.
007260  017737  177660  001126          MOV     @ADCSR,$BDDAT   ;*GET TO BACK.
007266  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;*DATA SENT=DATE RECIEVED?
007274  001401                          BEQ     2$              ;*IF SO-CONTINUE

007276  104002                          ERROR   2               ;*REPORT ERROR: "CSR READ/WRITE ERROR"

007300  000004                  2$:     SCOPE                   ;*LOOP

007302  062703  000002          1$:     ADD     #2,R3           ;*UPDATE PATTERN POINTERS
007306  020327  010020                  CMP     R3,#ADPATE      ;*DONE ALL PATTERNS?
007312  003755                          BLE     ADLOG2          ;*IF NOT-CONTINUE NEXT PATTERN.

                                        ;*WILL CONVERTING SET AND THEN CLEAR?

007314  012737  104000  001124  ADLOG3: MOV     #104000,$GDDAT  ;*SET CONVERT,READ CSR
007322  012737  007322  001106  1$:     MOV     #1$,$LPADR
007330  013777  001124  177606          MOV     $GDDAT,@ADCSR   ;*SEND TO ADCSR
007336  017737  177602  001126          MOV     @ADCSR,$BDDAT   ;*GET ADO5 CSR
007344  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;*DID CONVERT BIT SET?
007352  001402                          BEQ     ADL03A
007354  104003                          ERROR   3               ;*ERROR: "CONVERT BIT FAILED TO SET"
007356  000411                          BR      ADLOG4

007360  013737  001560  001566  ADL03A: MOV     FR16,FREQ       ;*SET TO DELAY 16 MILLISEC.
007366  104422                          DELAY
007370  005777  177550                  TST     @ADCSR          ;*DID CONVERT BIT CLEAR?
007374  100002                          BPL     1$+2
```

```
1432  007376  104004                        ERROR   4               ;*ERROR "CONVERT BIT FAILED TO CLEAR"
1433
1434                                                                  ;*LOOP.
1435                                                                  ;*NEXT TEST.
1436  007400  104427                1$:      INITR                   ;*ISSUE SYSTEM INITIALIZE
1437  007402  000004                ADLOG4:  SCOPE
1438
1439                                        ;*CAN WE READ THE DBR WITH CSR BIT//CLEAR (READ BIT).
1440
1441  007404  012777  052770  177532  ADLOG5:  MOV     #52770,@ADCSR   ;*LOAD CSR WITH ALL BUT READ CR BIT
1442  007412  022777  052770  177524           CMP     #52770,@ADCSR   ;*DID WE READ THE CR?
1443  007420  001001                           BNE     1$              ;*IF NOT NO ERROR.
1444
1445  007422  104005                           ERROR   5               ;*REPORT ERROR "CANNOT READ ADDS DATA
1446                                                                  ;*REGISTER WITH READ BIT CLEARED."
1447
1448  007424  000004                1$:      SCOPE
1449
1450                                        ;*WILL CONVERT DONE CAUSE INTERRUPT?
1451
1452  007426  012777  000340  172166  ADLOG6:  MOV     #340,@ICSVT2
1453  007434  012777  007506  172156           MOV     #ADLOG7,@ICSVT  ;*SET VECTOR FOR INTER.
1454  007442  012777  000004  172144           MOV     #4,@ICSR        ;*ALLOW ICS TO INT.
1455  007450  012777  104000  177466           MOV     #104000,@ADCSR  ;*SET ADDS TO CONVERT.
1456  007456  013737  001560  001566           MOV     FR16,FREQ       ;*ALLOW UP TO 16 MILLI SEC FOR NTO.
1457  007464  005037  177776                  CLR     PS
1458  007470  104422                          DELAY
1459  007472  000240                          NOP
1460  007474  104006                          ERROR   6               ;*REPORT ERROR "ADDS FAILED TO INTER. AT
1461  007476  005037  172112          ADL6L:  CLR     @ICSR
1462  007502  000005                          SCOPE
1463  007504  000445                          BR      ADLOG9
1464
1465                                        ;*CHECK INTR. ADDR. IN ICAR+GENERIC CODE
1466
1467  007506  022626                ADLOG7:  POPSP2                  ;*ADDS INTR TO HERE RESET SP.
1468  007510  017737  172102  001126           MOV     @ICAR,$BDDAT    ;*GET ADDR. AND GEN CODE
1469  007516  013737  007144  001124           MOV     ADCSR,$GDDAT    ;*FORM GOOD ADDR.
1470  007524  042737  177000  001124           BIC     #177000,$GDDAT
1471  007532  006237  001124                  ASR     $GDDAT
1472  007536  052737  003400  001124           BIS     #003400,$GDDAT  ;*ADD GENERIC CODE
1473  007544  023737  001126  001124           CMP     $BDDAT,$GDDAT   ;*IS ADDR + GENERIC CODE OK?
1474  007552  001402                           BEQ     ADLOG8
1475
1476  007554  104007                           ERROR   7               ;*REPORT ERROR "ADDS ADDR. OR GENERIC
1477
1478                                                                  ;*CODE INCORRECT.
1479  007556  000747                          BR      ADL6L
1480
1481                                        ;*CHECK TO SEE IF RIF CLEARS INTR FLAG ON ADDS
1482
1483  007560  052777  000001  172026  ADLOG8:  BIS     #1,@ICSR        ;*SET RIF BIT.
1484  007566  005777  177352                  TST     @ADCSR          ;*PUT IT TO WORK ON ADDS.
1485  007572  012777  007612  172020           MOV     #1$,@ICSVT      ;*SET INTERRUPT VECTORS.
1486  007600  005037  177776                  CLR     PS              ;*ALLOW INTERRUPTS
1487  007604  104422                          DELAY
```

```
1488  007606  000240                        NOP
1489  007610  000732                        BR      ADL6L           ;*EXIT A/D TESTS.
1490
1491  007612  022626                1$:     POPSP2                  ;*RESET SP.
1492  007614  104010                        ERROR   10              ;*REPORT ERROR" RIF DID NOT CLEAR INTR.
1493                                                                ;*FLAG ON ADO5".
1494  007616  000727                        BR      ADL6L
1495
1496                                        ;*
1497                                        ;*ADO5 DUAL ADDRESSING TEST
1498                                        ;*
1499
1500  007620  012737  007710  001106 ADLOG9: MOV     #2$,$LPADR
1501  007626  005037  001122                CLR     $BDADR          ;SET FIRST ADDR.
1502  007632  053737  001612  001122        BIS     ICSMOD,$BDADR
1503  007640  162737  000002  001122        SUB     #2,$BDADR
1504  007646  013737  001612  013134        MOV     ICSMOD,RTEMP    ;FIX STOP ADDR.
1505  007654  062737  000040  013134        ADD     #40,RTEMP
1506  007662  062737  000002  001122 1$:     ADD     #2,$BDADR       ;UPDATE ADDR.
1507  007670  023737  001122  013134        CMP     $BDADR,RTEMP    ;DONE ALL ADDRS?
1508  007676  001427                        BEQ     6$
1509  007700  023737  001122  007144        CMP     $BDADR,ADCSR    ;SAME ADDR AS ADO5 UNDER TEST?
1510  007706  001765                        BEQ     1$
1511  007710  012777  004100  177226 2$:     MOV     #4100,@ADCSR    ;SEND DATA TO A/D.
1512  007716  022777  004100  171176        CMP     #4100,@$BDADR   ;DUAL ADDR. READ ERROR?
1513  007724  001001                        BNE     3$
1514
1515  007726  104026                        ERROR   26              ;DUAL ADDR. READ ERROR
1516  007730  000004                3$:     SCOPE
1517
1518  007732  012777  004200  171162 4$:     MOV     #4200,@$BDADR   ;CHECK FOR WRITE ERROR
1519  007740  022777  004200  177176        CMP     #4200,@ADCSR    ;BAD WRITE?
1520  007746  001001                        BNE     5$
1521
1522  007750  104027                        ERROR   27              ;DUAL ADDR. WRITE ERROR
1523
1524  007752  000004                5$:     SCOPE
1525  007754  000742                        BR      1$
1526  007756  104427                6$:     INITR
1527  007760  000207                        RTS     PC              ;RETURN TO
1528                                                                ;ADLOGE
1529
1530                                        ;*END OF A/D TESTS
1531
1532  007762  005237  001100        ADLOGE: INC     $PASS
1533  007766  104401                1$:     TYPE                    ;*THE "END"
1534  007770  021465                        MEND
1535  007772  000137  007154                JMP     ADLOGL          ;*RETURN TO MONITOR.
1536
1537  007776  004000                ADPATP: 004000
1538  010000  004020                        004020
1539  010002  004040                        004040
1540  010004  004100                        004100
1541  010006  004200                        004200
1542  010010  004400                        004400
1543  010012  005000                        005000
```

K04


```
 1544  010014  006000                          006000
 1545  010016  004000                          004000
 1546  010020  004000                 ADPATE:  004000
 1547
 1548  010022  000137  003374                  JMP     START1
 1549
```

```
                                                ;*
                                                ;*TEST 6 A/D CALIBRATION TEST
                                                ;*
                                        TST6:   TYPE                    ;TYPE HEADER.
010026  104401                                  MHT6
010030  025570                                  ADAR                    ;GET ADDS ADDR.
010032  104426                                  QUBR                    ;UNI OR BI-POLAR?
010034  104431                                  ;T8CA AUDO1 CALIBRATION ROUTINE
                                                ;SWR0=1 RETURN TO MONITR
                                                ;SWR4-6 SELECT CHANNEL
                                                ;SWR7-10 SELECT MUX.
                                                ;SWR11=1 TYPE OUT CONVERSIONS
                                                ;SWR11=0 DISPLAY CONVERSIONS
                                                ;SWR12-14 SELECT GAIN (1 TO 1000)
010036  104401  024775                          TYPE,   MSW0
010042  104401  022004                          TYPE,   MWK
010046  052777  000100  171070        T8CAA:    BIS     #.00,@STKS      ;ALLOW TTY INTERRUPT
010054  013737  013340  007144                  MOV     ADADR,ADCSR
010062  001004                                  BNE     1$
010064  104401                                  TYPE
010066  024746                                  MNAD
010070  000137  003374                          JMP     START1
010074  032777  000001  171036        1$:       BIT     #1,@SWR         ;CHECK THE SWITCH REGISTER
010102  001402                                  BEQ     .+6             ;FOR RETURN TO MONITOR
010104  000137  003374                          JMP     START1
010110  004737  010740                          JSR     PC,CONVER       ;START CONVERSION
010114  017700  177024                          MOV     @ADCSR, %0      ;GET RESULTS
010120  004737  010726                          JSR     PC,REPET7       ;RIGHT JUSTIFY
010124  032777  004000  171006                  BIT     #4000,@SWR      ;DISPLAY OR TYPE DATA?
010132  001016                                  BNE     CALTP           ;TYPE IT
010134  023737  001600  001576                  CMP     FR1120,FR1
010142  001411                                  BEQ     2$
010144  022737  000176  001140                  CMP     #176,SWR        ;DISPLAY AVAILABLE?
010152  001405                                  BEQ     2$              ;IF NOT, THEN BRANCH
010154  000005                                  RESET
010156  000005                                  RESET
010160  000005                                  RESET
010162  000005                                  RESET
010164  000005                                  RESET
010166                                  2$:
010166  000727                                  BR      T8CAA
010170  104401                          CALTP:  TYPE
010172  001163                                  $CRLF
010174  004737  010240                          JSR     %7,CALIT        ;CONVERT DATA TO BINARY
010200  013737  001550  001566                  MOV     FR110,FREQ
010206  104422                                  DELAY                   ;LET TTY SETTLE DOWN.
010210  104401                                  TYPE
010212  025146                                  MCALT1
010214  017700  176724                          MOV     @ADCSR, %0      ;GET DATA
010220  004737  010726                          JSR     PC,REPET7
010224                                          TYPOCT  R0
010224  010046                                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
010226  104402                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
010230  104401                                  TYPE
```

```
1606  010232  025125                        MCALOT
1607  010234  104422                        DELAY
1608  010236  000703                        BR      TBCAA
1609  010240  005737  013342        CALIT:  TST     BIPOL
1610  010244  001406                        BEQ     3$
1611  010246  022700  007777                CMP     #7777,R0
1612  010252  001017                        BNE     2$
1613  010254  104401                        TYPE
1614  010256  025065                        MPOVFL
1615  010260  000207                        RETURN
1616  010262                        3$:
1617  010262  022700  004000                CMP     #4000,  %0      ;ROUTINE TO CONVERT BINARY
1618  010266  001003                        BNE     1$              ;READ FROM A/D BUFFER TO SIX
1619  010270  104401                        TYPE                    ;PLACE DECIMAL VOLTAGE
1620  010272  025036                        MMOVFL
1621  010274  000207                        RTS     PC
1622  010276  022700  003777        1$:     CMP     #3777,  %0      ;POSITIVE OR NEGATIVE
1623  010302  001003                        BNE     2$
1624  010304  104401                        TYPE
1625  010306  025065                        MPOVFL
1626  010310  000207                        RTS     PC
1627  010312  012703  010610        2$:     MOV     #A1,    %3      ;CLEAR STORAGE AREA
1628  010316  005023                        CLR     (3)+
1629  010320  020327  010624                CMP     %3,     #A1+14  ;DETERMINE IF
1630  010324  001374                        BNE     .-6
1631  010326  012703  010610                MOV     #A1,    %3      ;POSITIVE OR
1632  010332  012702  002000                MOV     #2000,  %2
1633  010336  005737  013342                TST     BIPOL
1634  010342  001401                        BEQ     .+4
1635  010344  006302                        ASL     %2
1636  010346  012701  010632                MOV     #HOLDIT,%1      ;NEGATIVE VOLTAGE
1637  010352  005737  013342                TST     BIPOL
1638  010356  001012                        BNE     CALIT1
1639  010360  032700  004000                BIT     #4000,R0
1640  010364  001405                        BEQ     CALITP
1641  010366  104401                        TYPE                    ;IF MINUS TYPE "-"
1642  010370  025172                        MMINUS
1643  010372  005100                        COM     %0
1644  010374  005200                        INC     %0
1645  010376  000402                        BR      CALIT1          ;IF POSITIVE
1646  010400  104401        CALITP: TYPE                            ;TYPE "+"
1647  010402  025174                        MPLUS
1648  010404  030200        CALIT1: BIT     %2,     %0
1649  010406  001044                        BNE     CALIT2          ;DO OCTAL TO
1650  010410  062701  000005                ADD     #5,     %1
1651  010414  006002        CALITE: ROR     %2                      ;DECIMAL CONVERSIONS
1652  010416  001372                        BNE     CALIT1
1653  010420  005737  010606                TST     HOLDTM          ;BY METHOD OF INCREMENTING
1654  010424  001403                        BEQ     .+10
1655  010426  112737  000001  010610        MOVB    #1,     A1
1656  010434  005037  010606                CLR     HOLDTM          ;DECIMAL COUNTER AND
1657  010440  012703  010610                MOV     #A1,    %3
1658  010444  052723  000260                BIS     #260,   (3)+
1659  010450  022703  010624                CMP     #A1+14, %3      ;DECREMENTING OCTAL DATA
1660  010454  001373                        BNE     .-10
1661  010456  012703  010611                MOV     #A1+1,  %3      ;UNTIL ZERO
```

```
1662  010462  112723  000001          MOVB    #1,     (3)+
1663  010466  105203                  INCB    %3
1664  010470  112723  000256          MOVB    #256,   (3)+    ;IF DECIMAL COUNTER = 12 (OCTAL)
1665  010474  105203                  INCB    %3
1666  010476  112723  000001          MOVB    #1,     (3)+
1667  010502  022703  010624          CMP     #A1+14, %3      ;CLEAR IT AND INCREMENT
1668  010506  001372                  BNE     .-12            ;NEXT COUNTER
1669  010510  105013                  CLRB    (3)
1670  010512  104401                  TYPE                    ;TYPE DECIMAL
1671  010514  010610                  A1
1672  010516  000207                  RTS     %7
1673  010520  005037  010606  CALIT2: CLR     HOLDTM
1674  010524  012703  010624          MOV     #A1+14, %3      ;FIRST USE TABLE HOLDIT
1675  010530  112137  010604  CALIT3: MOVB    (1)+,   SCAN            ;TO GET DECIMAL VALUE FOR DATA
1676  010534  063743  010604          ADD     SCAN,   -(3)    ;COMPUTED FROM ADDS
1677  010540  005737  010606          TST     HOLDTM
1678  010544  001404                  BEQ     CALIT4
1679  010546  005037  010606          CLR     HOLDTM          ;ADD THESE VALUES TOGETHER
1680  010552  062713  000001          ADD     #1,     (3)
1681  010556  122713  000012  CALIT4: CMPB    #12,    (3)     ;HOLDIT CONTAINS VALUES FOR DATA
1682  010562  101004                  BHI     CALIT5
1683  010564  162713  000012          SUB     #12,    (3)     ;BASED ON GAIN OF A/D
1684  010570  005237  010606          INC     HOLDTM
1685  010574  022703  010612  CALIT5: CMP     #A1+2,  %3      ;AT TIME DATA WAS TAKEN
1686  010600  001353                  BNE     CALIT3
1687  010602  000704                  BR      CALITE
1688  010604  000000          SCAN:   0
1689  010606  000000          HOLDTM: 000000
1690  010610  000000          A1:     000000
1691          010632          .=.+20
1692  010632    000     000     000 HOLDIT: .BYTE   0,0,0,0,5       ;TABLE OF VOLTAGE REPRESENTATION
1693  010635    000     005
1694
1695  010637    000     000     000         .BYTE   0,0,0,5,2       ;OF BINARY INPUT.
1696  010642    005     002
1697
1698  010644    000     000     005         .BYTE   0,0,5,2,1
1699  010647    002     001
1700
1701  010651    000     005     002         .BYTE   0,5,2,6,0
1702  010654    006     000
1703
1704  010656    005     002     001         .BYTE   5,2,1,3,0
1705  010661    003     000
1706
1707  010663    003     006     005         .BYTE   3,6,5,1,0
1708  010666    001     000
1709
1710  010670    001     010     007         .BYTE   1,8.,7,0,0
1711  010673    000     000
1712
1713  010675    001     011     003         .BYTE   1,9.,3,0,0
1714  010700    000     000
1715
1716  010702    005     011     001         .BYTE   5,9.,1,0,0
1717  010705    000     000
```

```
1718
1719  010707  010     011     000          .BYTE   8.,9.,0,0,0
1720  010712  000     000
1721
1722  010714  011     004     000          .BYTE   9.,4,0,0,0
1723  010717  000     000
1724
1725  010721  004     002     000          .BYTE   4,2,0,0,0
1726  010724  000     000
1727
1728                                 .EVEN
1729
1730  010726  006000                 REPET7: ROR     %0
1731  010730  006000                         ROR     %0              ;ROUTINE TO RIGHT
1732  010732  006000                         ROR     %0              ;JUSTIFY DATA BY
1733  010734  006000                         ROR     %0              ;ROTATING IT 4 PLACE TO
1734  010736  000207                         RTS     PC              ;THE RIGHT
1735  010740  017702  170174         CONVER: MOV     @SWR,R2
1736  010744  042702  000017                 BIC     #17,    %2      ;THISROUTINE SAMPLES SWR
1737  010750  052702  104000                 BIS     #104000,%2      ;AND SENDS CHANNEL, GAIN
1738  010754  010277  176164                 MOV     %2,     @ADCSR  ;CONVERT AND READ BITS TO ADUO1
1739  010760  005777  176160         CONVED: TST     @ADCSR
1740  010764  100775                         BMI     .-4             ;TO START CONVERSION
1741  010766  005077  176152                 CLR     @ADCSR          ;AND WAIT FOR CONVERSION COMPLETE
1742  010772  000207                         RTS     PC
1743
1744  010774  000137  003374                 JMP     START1
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
```

```
                                        ;*
                                        ;*TEST 7 ROUTINE TO MANUALLY TAKE SAMPLES, COMPARE RESULTS
                                        ;*PRINT ERROR IF ANY
                                        ;*CALL=JMP TST8 FROM MONITOR
                                        ;*
011000  104401              TST7:   TYPE                    ;TYPE HEADER
011002  025625                      MHT8
011004  104426                      ADAR
011006  104431                      QUBR
011010  013737  013340  007144      MOV     ADADR,FOCSR
011016  013737  001550  001566      MOV     FR110,FREQ
011024  104422                      DELAY                   ;ALLOW TTY TO SETTLE  BEFORE CONT.
011026  004737  007164              JSR     PC,ADLOG        ;BASIC LOGIS CHECK
011032  005737  001652          22$:TST     EXPERT
011036  001002                      BNE     .+6
011040  104401                      TYPE                    ;ASK FOR GAIN
011042  025763                      MGAIN
011044  012737  013100  011066      MOV     #GAIN,10$
011052  012737  000001  011070      MOV     #1,     10$+2
011060  000401                      BR      99$
011062  000763                      BR      22$
011064  104414              99$:    INOCT                   ;GET GAIN
011066  013100              10$:    GAIN
011070  000001                      1

011072  012700  013154              MOV     #GLIST,R0       ;GET LIST OF LEGAL GAINS.
011076  005710              20$:    TST     (0)             ;AT END OF LIST?
011100  001003                      BNE     21$             ;NO-CONTINUE.
011102  104401                      TYPE                    ;GAIN HE TYPED IS KNOWN.
011104  026333                      MNSG                    ;TELL HIM.
011106  000751                      BR      22$             ;REASK QUESTION.
011110  012001              21$:    MOV     (0)+,R1         ;GET GAIN FROM GAIN LIST.
011112  042701  170000              BIC     #170000,R1      ;MASK OUT REAL GAIN BITS.
011116  020137  013100              CMP     R1,GAIN         ;GAIN HE TYPED MATCH ONE IN GAIN LIST?
011122  001365                      BNE     20$             ;NO-CHECK NEXT IN LIST.
011124  014037  013100              MOV     -(0),GAIN       ;YES-REPLACE TYPED GAIN BY REAL GAIN.

011130  005737  001652          3$: TST     EXPERT
011134  001002                      BNE     .+6
011136  104401                      TYPE                    ;ASK FOR CHANS:
011140  026[ 0                     MCHAN
011142  0[ 037  013104              CLR     CHANS
011146  00[037  013106              CLR     CHANF
011152  005037  013110              CLR     CHANSR
011156  005037  013112              CLR     CHANFR
011162  013737  001650  013134      MOV     CORSIZ,RTEMP
011170  162737  027240  013134      SUB     #SENDAD,RTEMP
011176  005037  013136              CLR     RTEMP1
011202  005237  013136          15$:INC     RTEMP1

011206  162737  001000  013134      SUB     #1000,RTEMP
011214  100372                      BPL     15$
011216  005337  013136              DEC     RTEMP1
011222  013737  013136  013114      MOV     RTEMP1,CHANNO
011230  013746  013136              MOV     RTEMP1,-(SP)
```

```
1811  011234  004737  016440              GOSUB   ,$S20
1812  011240  062716  000010              ADD     #10,(SP)
1813  011244  005726                      TST     (SP)+
1814  011246  012737  013110  011270      MOV     #CHANSR,11$
1815  011254  012737  000002  011272      MOV     #2,11$+2
1816  011262  000401                      BR      .+4
1817  011264  000721                      BR      3$
1818  011266  104414                      INOCT                   ;GET CHANS
1819  011270  013110              11$:    CHANSR
1820  011272  000002                      2
1821  011274  005737  013112              TST     CHANFR
1822
1823  011300  001003                      BNE     .+10
1824  011302  013737  013110  013112      MOV     CHANSR,CHANFR
1825  011310  023737  013110  013112      CMP     CHANSR,CHANFR   ;CHAN. S=F?
1826  011316  003403                      BLE     4$
1827  011320  104401                      TYPE
1828  011322  026047                      MCHER1
1829  011324  000701                      BR      3$
1830  011326  023727  013112  000177  4$: CMP     CHANFR,#177     ;CHAN WITHIN LEGAL BOUNDS?
1831  011334  003403                      BLE     30$
1832  011336  104401                      TYPE                    ;NO-THEN TELL HIM
1833  011340  026253                      MCHANH
1834  011342  000672                      BR      3$
1835  011344  005737  001652          30$: TST    EXPERT
1836  011350  001002                      BNE     .+6
1837  011352  104401                      TYPE                    ;ASK FOR EXPECTED AVERAGE.
1838  011354  026302                      MAVEQ
1839  011356  012737  013124  011400      MOV     #AVEXP,12$
1840  011364  012737  000001  011402      MOV     #1,12$+2
1841  011372  000401                      BR      .+4
1842  011374  000763                      BR      30$
1843  011376  104414                      INOCT                   ;GET AVERAGE.
1844  011400  013124              12$:    AVEXP
1845  011402  000001                      .
1846  011404  032737  170000  013124      BIT     #170000,AVEXP           ;LEGAL AVERAGE?
1847  011412  001403                      BEQ     6$
1848  011414  104401                      TYPE                    ;NO TELL HIM(OR HER) ASK:
1849  011416  026353                      MNTL                    ;QUESTION AGAIN.
1850  011420  000751                      BR      30$
1851  011422  012737  013122  011502  6$: MOV     #TOLER,13$
1852  011430  012737  000001  011504      MOV     #1,13$+2
1853  011436  032737  004000  013124      BIT     #4000,AVEXP
1854  011444  001406                      BEQ     60$
1855  011446  005737  013342              TST     BIPOL
1856  011452  001003                      BNE     60$
1857  011454  052737  170000  013124      BIS     #170000,AVEXP
1858  011462  005737  001652          60$: TST    EXPERT
1859  011466  001002                      BNE     .+6
1860  011470  104401                      TYPE                    ;ASK FOR TOLERANCE.
1861  011472  026025                      MTOL
1862  011474  000401                      BR      .+4
1863  011476  000771                      BR      60$
1864  011500  104414                      INOCT
1865  011502  013122              13$:    TOLER
1866  011504  000001                      1
```

```
1867  011506  005737  001660                    TST     LINEPR
1868  011512  001411                            BEQ     14$
1869  011514  013737  001662  001150            MOV     LPCSR,$TPS
1870  011522  013737  001664  001152            MOV     LPDBR,$TPB
1871  011530  104401  022004                    TYPE,   MWK
1872  011534  000402                            BR      .+6
1873  011536  104401                    14$:    TYPE                    ;TYPE "REPEAT"
1874  011540  025114                            MREP
1875  011542  013737  013110  013106            MOV     CHANSR,CHANF
1876  011550  005337  013106                    DEC     CHANF
1877  011554  032777  000001  167356            BIT     #1,@SWR
1878  011562  001402                            BEQ     .+6
1879  011564  000137  003374                    JMP     START1
1880  011570  013737  013106  013104  5$:     MOV     CHANF,CHANS
1881  011576  005237  013104                    INC     CHANS
1882  011602  063737  013114  013106            ADD     CHANNO,CHANF
1883  011610  023737  013104  013112            CMP     CHANS,CHANFR
1884  011616  003347                            BGT     14$
1885  011620  023737  013106  013112            CMP     CHANF,CHANFR
1886  011626  003403                            BLE     .+10
1887  011630  013737  013112  013106            MOV     CHANFR,CHANF
1888  011636  004737  011650                    GOSUB   ,SAMPR          ;TAKE CONVERSIONS
1889  011642  004737  012152                    GOSUB   ,AVERR          ;TAKE AVERAGES
1890  011646  000750                            BR      5$
1891
1892                                            ;*
1893                                            ;*ROUTINE TO TAKE 256 SAMPLES OF AL CHANNELS SPECIFIED
1894                                            ;*CALL=GOSUB,SAMPR
1895
1896  011650  013737  001552  001566  SAMPR:   MOV     FRSD,FREQ       ;SUB PART TO SET UP DELAY BASEDON
1897  011656  013737  013106  013076            MOV     CHANF,CHAN
1898  011664  163737  013104  013076            SUB     CHANS,CHAN      ;NUMBER OF CHANNELLS WERE SAMPLING.
1899  011672  013737  013076  013120            MOV     CHAN,SAMOFF
1900  011700  001416                            BEQ     2$
1901  011702  006337  013120                    ASL     SAMOFF
1902  011706  163737  001562  001566  1$:     SUB     FRS,FREQ
1903  011714  005337  013076                    DEC     CHAN
1904  011720  001372                            BNE     1$
1905  011722  005737  001566                    TST     FREQ
1906  011726  100003                            BPL     2$
1907  011730  012737  000001  001566            MOV     #1,FREQ
1908  011736  012737  177400  013116  2$:     MOV     #-256.,SAMCNT   ;SET SAMPLE COUNT.
1909  011744  012701  027240                    MOV     #BUFFER,R1      ;SET FOR STORAGE.
1910  011750  013737  013104  013076  3$:     MOV     CHANS,CHAN      ;GET STARTING CHANNEL
1911  011756  104422                            DELAY
1912  011760  013737  013076  013074  4$:     MOV     CHAN,CHAN1      ;SET TO RIGHT JUSTIFY CHAN.
1913  011766  006337  013074                    ASL     CHAN1           ;FIX TO LOAD INTO AID WORD.
1914  011772  006337  013074                    ASL     CHAN1
1915  011776  006337  013074                    ASL     CHAN1
1916  012002  006337  013074                    ASL     CHAN1
1917  012006  004737  012044                    GOSUB   ,CONVT          ;TAKE CONVERSION.
1918  012012  010021                            MOV     R0,(1)+         ;STORE RESULT
1919  012014  005237  013076                    INC     CHAN            ;READY FOR NEXT CHAN.
1920  012020  023737  013076  013106            CMP     CHAN,CHANF      ;DONE ALL CHANNELLS
1921  012026  003754                            BLE     4$
1922  012030  005237  013116                    INC     SAMCNT          ;DONE 256 SAMPLES?
```

```
1923  012034  001345                          BNE     3$
1924  012036  012701  027240                  MOV     #BUFFER,R1
1925  012042  000207                          RETURN
1926
1927                                          ;*ROUTINE TO FORM A AID WORD FOR CONVERSIONS, START A CONVERSION,
1928                                          ;*WAIT FOR DONE, AND ENABLE READING OF THE AID DBR.
1929
1930                                          ;*CALL=GOSUB,CONVT,RESULT FOUND IN R0,RIGHT JUSTIFIED
1931  012044  013737  013100  013102  CONVT:  MOV     GAIN,ADWD               ;SET GAIN IN WORD
1932  012052  042737  007777  013102          BIC     #7777,ADWD
1933  012060  053737  013074  013102          BIS     CHAN1,ADWD              ;SET CHAN.
1934  012066  052737  104000  013102          BIS     #104000,ADWD            ;SET READ BIT, CONVERT BIT.
1935  012074  013777  013102  175042          MOV     ADWD,@ADCSR             ;START CONVERSION.
1936  012102  005777  175036          1$:     TST     @ADCSR          ;WAIT FOR CONVERSION COMPLETE.
1937  012106  100775                          BMI     1$
1938  012110  005077  175030                  CLR     @ADCSR          ;ENABLE READING OF DBR.
1939  012114  017700  175024                  MOV     @ADCSR,R0       ;READ AID RESULTS PUT IN R0
1940  012120  005737  013342                  TST     BIPOL
1941  012124  001405                          BEQ     2$
1942  012126  006000                          ROR     R0
1943  012130  006000                          ROR     R0
1944  012132  006000                          ROR     R0
1945  012134  006000                          ROR     R0
1946  012136  000207                          RETURN
1947  012140  006200          2$:     ASR     R0              ;RIGHT JUSTIFY, REMEMBERING SIGN.
1948  012142  006200                  ASR     R0
1949  012144  006200                  ASR     R0
1950  012146  006200                  ASR     R0
1951  012150  000207                  RETURN                  ;RETURN
1952                                          ;*
1953                                          ;*AVERAGING ROUTINE USED BE TEST 7
1954                                          ;*AT THIS POINT IN TIME ALL SAMPLES FOR ALL CHANNELS HAVE
1955                                          ;*BEEN TAKEN AND STORED IN "BUFFER" IN A
1956                                          ;*SEQUENTIAL INTERLEAVED BUFFER FORM" GIVEN BY THE FOLLOWING FORMULA:
1957                                          ;*"L=2*N+R1" WHERE L EQUALS THE LOCATION OF A SAMPLE,
1958                                          ;*N EQUALS THE NUMBER OF CHANNELS SAMPLES WERE TAKEN ON, AND R1
1959                                          ;*IS THE BUFFER POINTER SET TO "BUFFER" INITIALLY.
1960                                          ;*CALL=GOSUB,AVERR
1961
1962  012152  013737  013104  013076  AVERR:  MOV     CHANS,CHAN      ;SET TO FIRST CHAN.
1963  012160  005037  013074                  CLR     CHAN1           ;1ST CHAN OFFSET
1964  012164  012737  000400  013116  AVERRL: MOV     #256.,SAMCNT
1965  012172  013701  013074                  MOV     CHAN1,R1        ;PUT CHAN OFFSET IN R1.
1966  012176  062701  027240                  ADD     #BUFFER,R1      ;ADD BUFFER POINTER TO R1.
1967  012202  005037  013126                  CLR     AVTKN           ;SET INITIAL CONDITIONS FOR THIS CHAN.
1968  012206  005037  013134                  CLR     RTEMP
1969  012212  011137  013130                  MOV     (1),RLOW
1970  012216  011137  013132                  MOV     (1),RHIGH
1971  012222  023711  013132          2$:     CMP     RHIGH,(1)       ;FIND REAL HIGH VALUE.
1972  012226  003002                          BGT     3$
1973  012230  011137  013132                  MOV     (1),RHIGH
1974  012234  021137  013130          3$:     CMP     (1),RLOW        ;FIND REAL LOW VALUE.
1975  012240  003002                          BGT     4$
1976  012242  011137  013130                  MOV     (1),RLOW
1977  012246  012137  013136          4$:     MOV     (1)+,RTEMP1     ;GET CURRENT SAMPLE.
1978  012252  005737  013342                  TST     BIPOL
```

```
1979  012256  001003                          BNE     .+10
1980  012260  062737  004000  013136          ADD     #4000,RTEMP1    ;ADD CONSTANT TO SAMPLE.
1981  012266  062737  013136  013126          ADD     RTEMP1,AVTKN    ;"BOOT" ADD ALL SAMPLES.
1982  012274  005737  013134                  ADC     RTEMP
1983  012300  062701  013120                  ADD     SAMOFF,R1       ;UPDATE TO LOOK AT NEXT CHAN SAMPLE
1984  012304  005337  013116                  DEC     SAMCNT          ;DONE ALL SAMPLES PER THIS CHAN?
1985  012310  001344                          BNE     2$              ;NO-DO NEXT SAMPLE.
1986  012312  013737  013124  013136          MOV     AVEXP,RTEMP1    ;YES! SEE IF RHIGH OK:
1987  012320  063737  013122  013136          ADD     TOLER,RTEMP1
1988  012326  023737  013136  013132          CMP     RTEMP1,RHIGH
1989  012334  002427                          BLT     ERAV1           ;NO-THEN REPORT ERROR.
1990  012336  163737  013122  013136          SUB     TOLER,RTEMP1    ;YES-OK-CHECK LOWEST READING.
1991  012344  163737  013122  013136          SUB     TOLER,RTEMP1
1992  012352  023737  013130  013136          CMP     RLOW,RTEMP1
1993  012360  002415                          BLT     ERAV1           ;NO-REPORT ERROR.
1994  012362  005737  013122                  TST     TOLER           ;DOES OPERATOR WISH "FORCED" TYPEOUT?
1995  012366  001415                          BEQ     ERAV2           ;IF SO-DO IT
1996  012370  062737  000002  013074  AVERRN: ADD     #2,CHAN1        ;SET TO DO NEXT CHAN-BUT
1997  012376  005237  013076                  INC     CHAN            ;IF DONE ALL CHANS-EXIT-
1998  012402  023737  013076  013106          CMP     CHAN,CHANF      ;OTHERWIZE LOOP.
1999  012410  003665                          BLE     AVERRL
2000  012412  000207                          RETURN
2001
2002                                          ;*ERROR REPORTER
2003
2004  012414  104401                  ERAV1:  TYPE                    ;TYPE "REPEATIBILITY ERROR"
2005  012416  026447                          MREPER
2006  012420  000402                          BR      .+6
2007  012422  104401                  ERAV2:  TYPE                    ;TYPE "REPEATIBILITY FORCED TYPEOUT"
2008  012424  026411                          MREPFT
2009  012426  012702  000007          1$:     MOV     #7,R2           ;FIND AVERAGE.
2010  012432  006237  013134          10$:    ASR     RTEMP
2011  012436  006037  013126                  ROR     AVTKN           ;AVERAGE=TOTAL OF SAMPLES-
2012  012442  005302                          DEC     R2              ;DIVIDED BY 256.
2013  012444  001372                          BNE     10$
2014  012446  006237  013134                  ASR     RTEMP
2015  012452  006037  013126                  ROR     AVTKN
2016  012456  005537  013126                  ADC     AVTKN
2017  012462  005737  013342                  TST     BIPOL
2018  012466  001003                          BNE     .+10
2019
2020  012470  162737  004000  013126          SUB     #4000,AVTKN     ;SUBTRACT CONSTANT.
2021  012476  104401                          TYPE                    ;TYPE HEADER 1
2022  012500  026476                          MREPT1
2023  012502                                  TYPOCT  CHAN            ;TYPE CHAN.
2024  012502  013746  013076                  MOV     CHAN,-(SP)      ;;SAVE CHAN FOR TYPEOUT
2025  012506  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2026  012510  104401                          TYPE
2027  012512  026406                          M2SP
2028  012514  013702  013100                  MOV     GAIN,R2         ;TYPE GAIN.
2029  012520  042702  170000                  BIC     #170000,R2
2030  012524                                  TYPOCT  R2
2031  012524  010246                          MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
2032  012526  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2033  012530  104401                          TYPE
2034  012532  026406                          M2SP
```

```
2035  012534  013702  013130                MOV     RLOW,R2
2036  012540  042702  170000                BIC     #170000,R2
2037  012544                                TYPOCT  R2              ;TYPE LOWEST SAMPLE TAKEN.
2038  012544  010246                        MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
2039  012546  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2040  012550  104401                        TYPE
2041  012552  026406                        M2SP
2042
2043
2044
2045
2046
2047
2048
2049  012554  013702  013126                MOV     AVTKN,R2
2050  012560  042702  170000                BIC     #170000,R2
2051  012564                                TYPOCT  R2              ;TYPE AVERAGE OF SAMPLES TAKEN.
2052  012564  010246                        MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
2053  012566  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2054  012570  104401                        TYPE
2055  012572  026406                        M2SP
2056  012574  013702  013132                MOV     RHIGH,R2
2057  012600  042702  170000                BIC     #170000,R2
2058  012604                                TYPOCT  R2              ;TYPE HIGHEST SAMPLE TAKEN.
2059  012604  010246                        MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
2060  012606  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2061  012610  104401                        TYPE                    ;TYPE SECOND HEADER.
2062  012612  026550                        MREPT2
2063  012614  013737  013126  013134        MOV     AVTKN,RTEMP     ;GET AV.
2064  012622  162737  000005  013134        SUB     #5,RTEMP        ;MAKE -5 POINT TO GET LOW POINT.
2065  012630  012737  000400  013116        MOV     #256.,SAMCNT    ;SET # OF SAMPLES.
2066  012636  013701  013074                MOV     CHAN1,R1
2067  012642  062701  027240                ADD     #BUFFER,R1
2068  012646  005002                        CLR     R2
2069  012650  022137  013134        2$:     CMP     (1)+,RTEMP
2070  012654  002001                        BGE     3$
2071  012656  005202                        INC     R2
2072  012660  063701  013120        3$:     ADD     SAMOFF,R1
2073  012664  005337  013116                DEC     SAMCNT
2074  012670  001367                        BNE     2$
2075  012672  004737  013036                GOSUB   AVTYP           ;TYPE RLOW # OF SAMPLES.
2076  012676  012703  177765                MOV     #-11.,R3                ;SET TO DO 11 TIMES.
2077  012702  012737  000400  013116  4$:   MOV     #256.,SAMCNT    ;SET SAMPLE COUNT AT 256.
2078  012710  005002                        CLR     R2
2079  012712  013701  013074                MOV     CHAN1,R1        ;GET BUFFER POINTER.
2080  012716  062701  027240                ADD     #BUFFER,R1
2081  012722  022137  013134        5$:     CMP     (1)+,RTEMP      ;SAMPLE=COUNT POINTER?
2082  012726  001001                        BNE     6$
2083  012730  005202                        INC     R2
2084  012732  063701  013120        6$:     ADD     SAMOFF,R1
2085  012736  005337  013116                DEC     SAMCNT          ;CHECKED ALL SAMPLES?
2086  012742  001367                        BNE     5$
2087  012744  004737  013036                GOSUB   AVTYP           ;TYPE # OF SAMPLES AT THIS POINT.
2088  012750  005237  013134                INC     RTEMP           ;MOVE TO NEXT POINT.
2089  012754  005203                        INC     R3              ;DONE ALL POINTS?
2090  012756  001351                        BNE     4$              ;NO-THEN LOOP
```

```
2091  012760  005002                      CLR     R2              ;FIND ALL OVERSCALE POINTS.
2092  012762  012737  000400  013116      MOV     #256.,SAMCNT
2093  012770  013701  013074              MOV     CHAN1,R1
2094  012774  062701  027240              ADD     #BUFFER,R1
2095  013000  022137  013134      7$:     CMP     (1)+,RTEMP
2096  013004  003401                      BLE     8$
2097  013006  005202                      INC     R2
2098  013010  063701  013120      8$:     ADD     SAMOFF,R1
2099  013014  005337  013116              DEC     SAMCNT
2100  013020  001367                      BNE     7$
2101  013022  004737  013036              GOSUB   ,AVTYP
2102  013026  104401                      TYPE
2103  013030  001163                      $CRLF
2104  013032  000137  012370              JMP     AVERRN
2105  013036  010246              AVTYP:  MOV     R2,-(SP)        ;PUT # ON STACK
2106  013040  004737  016440              GOSUB   ,$$B2D          ;GOTO OCTAL-BCD ROUTINE.
2107  013044  062716  000002              ADD     #2,(SP)         ;GET RID OF 1ST 2 DIGITS
2108  013050  012637  013056              MOV     (SP)+,1$                ;TYPE STRING.
2109  013054  104401                      TYPE
2110  013056  000000              1$:     0
2111  013060  104401                      TYPE                    ;TYPE 2 SPACES.
2112  013062  026406                      M2SP
2113  013064  000207                      RETURN
2114
2115                                       ;*
2116                                       ;*POINTERS USED BY TEST 7  REPEATIBILITY TEST
2117                                       ;*
2118
2119  013066  000000              CHAN7:0
2120  013070  000000              NA07:0
2121  013072  000000              NA17:0
2122  013074  000000              CHAN1:  0                       ;LEFT JUSTIFIED CURRENT CHANNELL
2123  013076  000000              CHAN:   0                       ;CURRENT CHANNELL
2124  013100  000000              GAIN:   0                       ;CURRENT GAIN
2125  013102  000000              ADWD:   0                       ;WORD SENT TO A/D
2126  013104  000000              CHANS:  0                       ;STARTING CHANNEL
2127  013106  000000              CHANF:  0                       ;LAST CHANNEL
2128  013110  000000              CHANSR: 0
2129  013112  000000              CHANFR: 0
2130  013114  000000              CHANNO: 0
2131  013116  000000              SAMCNT: 0                       ;SAMPLE COUNT
2132  013120  000000              SAMOFF: 0                       ;SAMPLE OFFSET
2133  013122  000000              TOLER:  0                       ;TOLERANCE BEFORE ERROR IS REPORTED
2134  013124  000000              AVEXP:  0                       ;EXPECTED AVERAGE
2135  013126  000000              AVTKN:  0                       ;AVERAGE OF SAMPLES TAKEN
2136  013130  000000              RLOW:   0                       ;LOWEST SAMPLE TAKEN
2137  013132  000000              RHIGH:  0                       ;HIGHEST SAMPLE TAKEN
2138  013134  000000              RTEMP:  0
2139  013136  000000              RTEMP1: 0
2140  013140  000000              REPMAN: 0                       ;SECTION RESERVED FOR
2141  013142  000000                      0                       ;MANUAL REPEATIBILITY
2142  013144  000000                      0                       ;WHERE OPERATOR ENTERS
2143  013146  000000                      0                       ;IN INFORMATION
2144  013150     000      000             .BYTE   0,0
2145  013152  000000                      0
2146
```

```
                                        ;*
                                        ;* GAIN LIST
                                        ;*

013154  071000          GLIST:  071000          ;GAIN OF 1000.
013156  060200                  060200          ;GAIN OF 200.
013160  050100                  050100          ;GAIN OF 100.
013162  040050                  040050          ;GAIN OF 50.
013164  030020                  030020          ;GAIN OF 20.
013166  020010                  020010          ;GAIN OF 10.
013170  010002                  010002          ;GAIN OF 2.
013172  000001                  000001          ;GAIN OF 1.
013174  000000                  000000          ;ILLEGAL GAIN.

                                        ;*INPUT MODULE ADDRESSES
013176  000000          INADR:  0
013200  000000                  .WORD   0
013202  000000                  .WORD   0
013204  000000                  .WORD   0
013206  000000                  .WORD   0
013210  000000                  .WORD   0
013212  000000                  .WORD   0
013214  000000                  .WORD   0
013216  000000                  .WORD   0
013220  000000                  .WORD   0
013222  000000                  .WORD   0
013224  000000                  .WORD   0
013226  000000                  .WORD   0
013230  000000                  .WORD   0
013232  000000                  .WORD   0
013234  000000                  .WORD   0
                                        ;*OUTPUT MODULE ADDRESSES
013236  000000          OUTADR: 0
013240  000000                  .WORD   0
013242  000000                  .WORD   0
013244  000000                  .WORD   0
013246  000000                  .WORD   0
013250  000000                  .WORD   0
013252  000000                  .WORD   0
013254  000000                  .WORD   0
013256  000000                  .WORD   0
013260  000000                  .WORD   0
013262  000000                  .WORD   0
013264  000000                  .WORD   0
013266  000000                  .WORD   0
013270  000000                  .WORD   0
013272  000000                  .WORD   0
013274  000000                  .WORD   0
                                        ;*COUNTER MODULE ADDRESSES
013276  000000          CNTADR: 0
013300  000000                  .WORD   0
013302  000000                  .WORD   0
013304  000000                  .WORD   0
013306  000000                  .WORD   0
013310  000000                  .WORD   0
013312  000000                  .WORD   0
```

```
 2203  013314  000000                          .WORD   0
 2204  013316  000000                          .WORD   0
 2205  013320  000000                          .WORD   0
 2206  013322  000000                          .WORD   0
 2207  013324  000000                          .WORD   0
 2208  013326  000000                          .WORD   0
 2209  013330  000000                          .WORD   0
 2210  013332  000000                          .WORD   0
 2211  013334  000000                          .WORD   0
 2212                                          ;* DAC MODULE ADDRESS
 2213  013336  000000                  DACADR: 0
 2214                                          ;*ADDS MODULE ADDRESS
 2215  013340  000000                  ADADR:  0
 2216  013342  000000                  BIPOL:  0                       ;0=BIPOLAR-1=UNI
 2217                                          ;*TEMP STORAGE OF INPUT MODULE DATA
 2218  013344  000000                  OUTS:   0
 2219  013346  000000                          .WORD   0
 2220  013350  000000                          .WORD   0
 2221  013352  000000                          .WORD   0
 2222  013354  000000                          .WORD   0
 2223  013356  000000                          .WORD   0
 2224  013360  000000                          .WORD   0
 2225  013362  000000                          .WORD   0
 2226  013364  000000                          .WORD   0
 2227  013366  000000                          .WORD   0
 2228  013370  000000                          .WORD   0
 2229  013372  000000                          .WORD   0
 2230  013374  000000                          .WORD   0
 2231  013376  000000                          .WORD   0
 2232  013400  000000                          .WORD   0
 2233  013402  000000                          .WORD   0
 2234  013404  000000                  OUTSE:  0
 2235
 2236                                          ;*
 2237                                          ;*ROUTINE TO INPUT INPUT MODULE ADDRESS.
 2238                                          ;*
 2239
 2240  013406  005737  001652          RINA:   TST     EXPERT
 2241  013412  001002                          BNE     .+6
 2242  013414  104401                          TYPE                            ;ASK OPERATOR "INPUT MODULE ADDR(S)?
 2243  013416  021263                          MIPA
 2244  013420  012737  013176  013442          MOV     #INADR, RINA1
 2245  013426  012737  000020  013444          MOV     #16.,   RINA1+2
 2246  013434  000401                          BR      .+4
 2247  013436  000763                          BR      RINA
 2248  013440  104414                          INOCT                           ;CALL TO INPUT OCTAL ROUTINE
 2249  013442  013176                  RINA1:  INADR                           ;ADDR TO STORE INPUT ADDRESS AT (VARIES).
 2250  013444  000020                          16.                             ;NUMBER OF ADDRESS ALLOWED.
 2251  013446  005737  015046                  TST     CHAR
 2252  013452  001411                          BEQ     2$
 2253  013454  012700  013176                  MOV     #INADR, R0              ;NOW FORM REAL ICS ADDRESS.
 2254  013460  053710  001612          1$:     BIS     ICSMOD, (0)             ;NORMAL-ICS MOD=171000 UNLESS PATCHED
 2255  013464  004737  015510                  JSR     PC,CKADR
 2256  013470  020037  013442                  CMP     R0,RINA1                ;DONE ALL ADDRS?
 2257  013474  101771                          BLOS    1$                      ;IF NOT CONTINUE.
 2258  013476  000002                  2$:     EXIT                            ;IF SO-EXIT
```

```
2259
2260                                        ;*
2261                                        ;*ROUTINE TO INPUT OUTPUT MODULE ADDRESSESS
2262                                        ;*
2263
2264  013500  005737  001652        ROUTA:  TST     EXPERT
2265  013504  001002                       BNE     .+6
2266  013506  104401                       TYPE                    ;ASK OPERATOR "OUTPUT MODULE ADDRESSESS?
2267  013510  021321                       MOPA
2268  013512  012737  013236  013534       MOV     #OUTADR, ROUTA1
2269  013520  012737  000020  013536       MOV     #16.,   ROUTA1+2
2270  013526  000401                       BR      .+4
2271  013530  000763                       BR      ROUTA
2272  013532  104414                       INOCT                   ;CALL TO INPUT OCTAL ROUTINE
2273  013534  013236              ROUTA1: OUTADR                  ;ADDR TO STORE OUTPUT ADDRS.
2274  013536  000020                       16.                     ;
2275  013540  005737  015046               TST     CHAR
2276  013544  001411                       BEQ     2$
2277  013546  012700  013236               MOV     #OUTADR, R0     ;
2278  013552  053710  001612        1$:    BIS     ICSMOD, (0)
2279  013556  004737  015510               JSR     PC,CKAOR
2280  013562  020037  013534               CMP     R0,ROUTA1
2281  013566  101771                       BLOS    1$
2282  013570  000002              2$:     EXIT
2283
2284                                        ;*
2285                                        ;*ROUTINE TO INPUT COUNTER MODULE ADDRS
2286                                        ;*
2287
2288  013572  005737  001652        RCNTA:  TST     EXPERT
2289  013576  001002                       BNE     .+6
2290  013600  104401                       TYPE                    ;ASK OPERATOR FOR COUNTER MODULE
2291  013602  021500                       MCNT                    ;ADDRS.
2292  013604  012737  013276  013626       MOV     #CNTADR, RCNTA1
2293  013612  012737  000020  013630       MOV     #15.,   RCNTA1+2
2294  013620  000401                       BR      .+4
2295  013622  000763                       BR      RCNTA
2296  013624  104414                       INOCT                   ;GET THEM
2297  013626  013276              RCNTA1: CNTADR
2298  013630  000020                       16.
2299  013632  012700  013276               MOV     #CNTADR, R0
2300  013636  005737  015046               TST     CHAR
2301  013642  001407                       BEQ     2$
2302  013644  053710  001612        1$:    BIS     ICSMOD, (0)
2303  013650  004737  015510               JSR     PC,CKAOR
2304  013654  020037  013626               CMP     R0,RCNTA1
```

```
2305  013660  003771                        BLE     1$
2306
2307  013662  000002              2$:       EXIT
2308
2309                                        ;*
2310                                        ;*ROUTINE TO INPUT DAC ADDR
2311                                        ;*
2312  013664  005737  001652        RDACA:  TST     EXPERT
2313  013670  001002                        BNE     .+6
2314  013672  104401                        TYPE
2315  013674  022021                        MDAC
2316  013676  012737  013336  013720        MOV     #DACADR, RDAC1
2317  013704  012737  000001  013722        MOV     #1,      RDAC1+2
2318  013712  000401                        BR      .+4
2319  013714  000763                        BR      RDACA
2320  013716  104414                        INOCT
2321  013720  013336              RDAC1:    DACADR
2322  013722  000001                        1
2323  013724  005737  015046              TST     CHAR
2324  013730  001407                        BEQ     1$
2325  013732  053737  001612  013336        BIS     ICSMOD, DACADR
2326  013740  012700  013336                MOV     #DACADR,R0
2327  013744  004737  015510                JSR     PC,CKADR
2328  013750  000002              1$:       EXIT
2329
2330                                        ;*
2331                                        ;*ROUTINE TO INPUT ADDS ADDR
2332  013752  005737  001652        RADA:   TST     EXPERT
2333  013756  001002                        BNE     .+6
2334  013760  104401                        TYPE
2335  013762  022044                        MADU
2336  013764  012737  013340  014006        MOV     #ADADR, RADA1
2337  013772  012737  000001  014010        MOV     #1,     RADA1+2
2338  014000  000401                        BR      .+4
2339  014002  000763                        BR      RADA
2340  014004  104414                        INOCT
2341  014006  013340              RADA1:    ADADR
2342  014010  000001                        1
2343  014012  005737  015046              TST     CHAR
2344  014016  001407                        BEQ     1$
2345  014020  053737  001612  013340        BIS     ICSMOD, ADADR
2346  014026  012700  013340                MOV     #ADADR,R0
2347  014032  004737  015510                JSR     PC,CKADR
2348  014036  000002              1$:       EXIT
2349
2350                                        ;*
2351                                        ;*INPUT OCTAL ROUTINE
2352                                        ;*NMW TO CALL=WHERE TO STORE NUMBERS
2353                                        ;*NMW = NUMBER OF NUMBERS TO ACCEPT.
2354                                        ;*
2355
2356  014040  104401              INOCTR: TYPE                    ;TYPE A "?".
2357  014042  022366                      MQ
2358  014044  017601  000000              MOV     @(6),   R1      ;PICK UP STORAGE ADDRESS.
2359  014050  062716  000002              ADD     #2,     (6)     ;PICK UP # OF WORDS THAT-
2360  014054  017637  000000  015060      MOV     @(6),   NINC1   ;IS THE MAX # TO BE INPUTED.
```

```
2361  014062  162716  000002                    SUB     #2,     (6)     ;CLEAR COLON TYPED FLAG.
2362  014066  005037  001630                    CLR     INCFLG          ;CLEAR ANY CHAR. TYPED FLAG.
2363  014072  005037  013134                    CLR     RTEMP           ;POINT I6 TEMP STORAGE AREA.
2364  014076  012700  013344                    MOV     #OUTS,R0        ;CLEAR THE AREA.
2365  014102  005020                    20$:    CLR     (0)+
2366  014104  020027  013404                    CMP     R0,#OUTSE
2367  014110  003774                    BLE     20$
2368  014112  012700  013344                    MOV     #OUTS,R0        ;POINT TO AREA AGAIN.
2369  014116  005037  015056                    CLR     NINC            ;CLEAR CHAR. COUNT
2370  014122  005037  015050            1$:     CLR     NIN             ;CLEAR NUMBER TO BE FORMED
2371  014126  005037  015054                    CLR     CHARC           ;INPUTED CHAR. COUNT
2372  014132  105777  165006            2$:     TSTB    @$TKS           ;KEY TYPED?
2373  014136  100375                    BPL     2$                      ;NO - THEN WAIT.
2374  014140  117737  165002  015046            MOVB    @$TKB,  CHAR    ;YES - READ CHAR.
2375  014146  042737  177600  015046            BIC     #177600, CHAR   ;STRIP CHAR PARITY BIT - IF ANY.
2376  014154  023727  015046  000177            CMP     CHAR,#177       ;WAS IT A RUBOUT?
2377  014162  001002                    BNE     .+6                     ;NO - CONTINUE
2378  014164  000137  015006                    JMP     10$             ;YES - TYPE "?" - REINTIALIZE.
2379  014170  105777  164754                    TSTB    @$TPS           ;PRINTER BUSY?
2380  014174  100375                    BPL     .-4                     ;YES - THEN WAIT TILL NOT
2381  014176  013777  015046  164746            MOV     CHAR,   @$TPB   ;NO - ECHO CHAR.
2382  014204  023727  015046  000015            CMP     CHAR,   #15     ;CHAR. = (CR)?
2383  014212  001514                    BEQ     3$                      ;YES - THEN TERMINATE INPUT.
2384  014214  023727  015046  000003            CMP     CHAR,#3         ;WAS IT A ↑C?
2385  014222  001002                    BNE     .+6
2386  014224  000177  165450                    JMP     @CTLLOC         ;IF YES GOTO MONITR.
2387  014230  023727  015046  000014            CMP     CHAR,#14        ;CHAR = "↑L"?
2388  014236  001002                    BNE     .+6
2389  014240  000137  015112                    JMP     LPSET           ;YES - THEN SET LINEPRINTER MODE.
2390  014244  122737  000012  015046            CMPB    #12,CHAR        ;CHAR = "↑J"?
2391  014252  001002                    BNE     .+6
2392  014254  000137  015134                    JMP     CONNTR          ;YES - THEN SET JOINED MODE.
2393  014260  122737  000004  015046            CMPB    #4,CHAR         ;CHAR = "↑D"?
2394  014266  001002                    BNE     .+6
2395  014270  000137  015260                    JMP     RDELA3          ;YES - GET SECOUNDARY DELAY.
2396  014274  123727  015046  000005            CMPB    CHAR,#5         ;CHAR = "↑E"?
2397  014302  001002                    BNE     .+6
2398  014304  000137  015062                    JMP     EXSET           ;YES - SET EXPERT MODE.
2399  014310  123727  015046  000016            CMPB    CHAR,#16        ;CHAR = "↑N"?
2400  014316  001002                    BNE     .+6
2401  014320  000137  015076                    JMP     NOSET           ;YES - SET NOVICE MODE
2402  014324  023727  015046  000072            CMP     CHAR,   #':     ;CHAR = COLON?
2403  014332  001522                    BEQ     4$                      ;YES SET FOR "THROUGH" ENTRY.
2404  014334  023727  015046  000054            CMP     CHAR,   #',     ;CHAR = A COMMA?
2405  014342  001575                    BEQ     7$
2406  014344  023727  015046  000060            CMP     CHAR,#60        ;CHAR TYPED 7 ASCIZO?
2407  014352  002002                    BGE     .+6
2408  014354  000137  015016                    JMP     11$             ;NO - REPORT ERROR.
2409
2410  014360  023727  015046  000067            CMP     CHAR,#67        ;CHAR TYPED (ASCIZ 7?
2411  014366  003402                    BLE     .+6
2412  014370  000137  015016                    JMP     11$             ;NO - REPORT ERROR.
2413  014374  005237  013134                    INC     RTEMP           ;YES - INCREMENT CHAR. COUNT.
2414  014400  006137  015050                    ROL     NIN             ;LEFT JUSTIFY CURRENT NUMBER.
2415  014404  006137  015050                    ROL     NIN
2416  014410  006137  015050                    ROL     NIN
```

```
2417  014414  005237  015054                    INC     CHARC           ;INCREMENT CHAR COUNT.
2418  014420  042737  000007  015050            BIC     #7,     NIN     ;STRIP LOWER NUMBER.
2419  014426  042737  000260  015046            BIC     #260,   CHAR    ;STRIP CHAR. INPUTED.
2420  014434  053737  015046  015050            BIS     CHAR,   NIN     ;ADD TO CURRENT NUMBER.
2421  014442  000633                            BR      2$              ;LOOP.
2422  014444  005737  001630          3$:       TST     INCFLG          ;COME HERE ON <CR>;COLON FLAG SET?
2423  014450  001066                            BNE     5$              ;YES TAKE CARE OF IT.
2424  014452  023737  015056  015060            CMP     NINC,   NINC1   ;DID WE ASSEMBLE MORE NUMBERS THAN ALLOWED?
2425  014460  003146                            BGT     8$              ;IF SO - REPORT ERROR.
2426  014462  005737  013134                    TST     RTEMP           ;ANY CHAR. TYPED?
2427  014466  001005                            BNE     9$              ;YES - FUDGE RETURN.
2428  014470  012737  000017  015056            MOV     #15.,   NINC
2429  014476  005037  015046                    CLR     CHAR
2430  014502  013720  015050          9$:       MOV     NIN,    (0)+    ;STORE NUMBER.
2431  014506  013737  015060  013134            MOV     NINC1,RTEMP     ;FIX NUMBER INPUTED COUNT.
2432  014514  163737  015056  015060            SUB     NINC,   NINC1   ;FIX EXACT COUNT.
2433  014522  005337  015060                    DEC     NINC            ;= COUNT -1
2434  014526  012700  013344                    MOV     #OUTS,R0        ;SET TO READ BACK NUMBERS.
2435  014532  012021                  21$:      MOV     (0)+,(1)+       ;STORE IN CORRECT AREA.
2436  014534  005337  015056                    DEC     NINC            ;DONE ALL NUMBERS?
2437  014540  001002                            BNE     22$             ;NO - SKIP NEXT INSTR.
2438  014542  010176  000000                    MOV     R1,@(6)         ;FIX CALL ON ADDR. OF NUMBERS TO INPUT.
2439  014546  005337  013134          22$:      DEC     RTEMP           ;DONE ALL NUMBERS.
2440  014552  001367                            BNE     21$             ;NO - LOOP.
2441  014554  062716  000002                    ADD     #2,     (6)     ;YES - UPDATE SP.
2442  014560  013776  015060  000000            MOV     NINC1,@(6)      ;STORE IN CALL # OF NUMBERS TO INPUT.
2443  014566  062716  000002                    ADD     #2,(6)          ;FIX SP.
2444  014572  104401  001163                    TYPE,   $CRLF
2445  014576  000002                            EXIT                    ;EXIT.
2446
2447  014600  013737  015050  015052  4$:       MOV     NIN,    NIN1    ;ENTER HERE WHEN COLON TYPED. STORE CURRENT #.
2448  014606  005737  001630                    TST     INCFLG          ;CHECK COLON FLAG.
2449  014612  001105                            BNE     12$             ;IF SET UNKNOWN INPUT (2 COLONS!)
2450  014614  012737  000001  001630            MOV     #1,INCFLG       ;SET COLON FLAG.
2451  014622  000137  014122                    JMP     1$              ;EXIT TO LOOP
2452  014626  162737  000002  015052  5$:       SUB     #2,     NIN1    ;COME HERE ON COLON FLAG SET + ANOTHER # FORMED.
2453  014634  062737  000002  015052  6$:       ADD     #2,     NIN1    ;UPDATE FORMER NUMBER.
2454  014642  023737  015056  015060            CMP     NINC,   NINC1   ;NUMBERS OVERFLOW BUFFER AREA?
2455  014650  003052                            BGT     8$              ;YES - REPORT ERROR.
2456  014652  013720  015052                    MOV     NIN1,   (0)+    ;NO - STORE NEW NUMBER.
2457  014656  005237  015056                    INC     NINC            ;UPDATE NUMBER COUNT.
2458  014662  023737  015050  015052            CMP     NIN,    NIN1    ;UPDATED NUMBER = CURRENT NUMBER?
2459  014670  001361                            BNE     6$              ;NO - LOOP.
2460  014672  005037  001630                    CLR     INCFLG          ;YES - CLEAR COLON FLAG
2461  014676  005337  015056                    DEC     NINC
2462  014702  162700  000002                    SUB     #2,R0           ;FIX POINTER (STORAGE).
2463  014706  022737  000015  015046            CMP     #15.    CHAR    ;LAST CHAR TYPED A <CR>?
2464  014714  001653                            BEQ     3$              ;YES - EXIT.
2465  014716  022737  000054  015046            CMP     #',,CHAR        ;LAST CHAR TYPED A COMMA?
2466  014724  001404                            BEQ     7$              ;YES - TAKE CARE OF IT.
2467
2468  014726  062700  000002                    ADD     #2,     R0      ;FIX STORAGE POINTER.
2469  014732  000137  014122                    JMP     1$              ;LOOP.
2470  014736  005737  001630          7$:       TST     INCFLG          ;ENTER HERE ON COMMA TYPED. WAS COLON-
2471  014742  001331                            BNE     5$              ;PREVIOUSLY TYPED? IF SO TAKE CARE OF IT.
2472  014744  023737  015056  015060            CMP     NINC,   NINC1   ;BUFFER OVERFLOW?
```

```
2473  014752  003011                        BGT     8$              ;IF YES - REPORT ERROR.
2474  014754  005737  015054                TST     CHARC           ;WHY CHAR. TYPED?
2475  014760  001422                        BEQ     12$             ;IF NO - REPORT ERROR.
2476  014762  013720  015050                MOV     NIN,    (0)+    ;OTHERWISE STORE CURRENT NUMBER.
2477  014766  005237  015056                INC     NINC            ;INCR. NUMBER COUNT.
2478  014772  000137  014122                JMP     1$              ;LOOP.
2479
2480  014776  104401              8$:       TYPE                    ;"ADDR. BUFFER OVERFLOW ERROR
2481  015000  022070                        MABOV                   ;RETYPE"
2482  015002  000137  014040                JMP     INOCTR
2483
2484  015006  104401              10$:      TYPE                    ;TYPE A "?"
2485  015010  022360                        MQMARK
2486  015012  000137  014040                JMP     INOCTR
2487
2488  015016  104401              11$:      TYPE                    ;TYPE "LAST CHAR TYPED NOT AN OCTAL DIGIT"
2489  015020  026126                        MINNN
2490  015022  000137  014132                JMP     2$
2491  015026  104401              12$:      TYPE
2492  015030  026207                        MINKN
2493  015032  000137  014040                JMP     INOCTR
2494
2495                                        ;
2496                                        ;RETURN FOR ↑D, ↑J,↑L,↑N, ↑E
2497                                        ;
2498
2499  015036  000240              RASK:     NOP
2500  015040  162716  000004                SUB     #4,(6)
2501  015044  000002                        EXIT
2502
2503  015046  000000              CHAR:     0                       ;CHARACTER INPUTTED
2504  015050  000000              NIN:      0                       ;ASSEMBLED NUMBER
2505  015052  000000              NIN1:     0                       ;ASSEMBLED NUMBER, 1ST IF MULTIPLE
2506  015054  000000              CHARC:    0                       ;USED TO COUNT CHARS. IN NUMBER.
2507
2508  015056  000000              NINC:     0                       ;NUMBER OF ADDRESSED ASSEMBLED.
2509  015060  000000              NINC1:    0                       ;MAXIMUM NUMBER TO BE ASSEMBLED.
2510  015062  005237  001652      EXSET:    INC     EXPERT          ;ENTER HER ON "↑E".
2511  015066  104401                        TYPE                    ;SET EXPERT MODE.
2512  015070  021540                        MEXEN
2513  015072  000137  015036                JMP     RASK
2514
2515  015076  005037  001652      NOSET:    CLR     EXPERT          ;ENTER HERE ON "↑N".
2516  015102  104401                        TYPE                    ;SET NOIVE MODE
2517  015104  021570                        MNOEN
2518  015106  000137  015036                JMP     RASK
2519
2520
2521
2522  015112  005737  001656      LPSET:    TST     LPAV            ;ENTER HERE ON "↑L".
2523  015116  001404                        BEQ     1$              ;ANY LP AVAILABLE - IF NOT EXIT.
2524  015120  104401                        TYPE                    ;TYPE "MAKE LP READY"
2525  015122  021621                        MLEN
2526  015124  005237  001660                INC     LINEPR
2527  015130  000137  015036      1$:       JMP     RASK
2528
```

```
                                        ;*
                                        ;*ROUTINE TO CONNECT I/O MODULES (SOFT)
                                        ;*
015134  005137  001654          CONNTR: COM     CONNT
015140  001402                          BEQ     .+6
015142  104401                          TYPE
015144  021707                          MAC
015146  000137  015036                  JMP     RASK


                                        ;*
                                        ;*ROUTINE TO ASK FOR AND SET DELAY TIME.
                                        ;* 0 DELAY OR CR BY OPERATOR - USER DEFAULT OF 3 MILLISEC.

015152  005737  001652          RDELA:  TST     EXPERT
015156  001002                          BNE     .+6
015160  104401                          TYPE                    ;ASK OPERATOR -
015162  022234                          MDEL                    ;"DELAY TIME IN MS (DEFAULT=3MS)?"
015164  005037  001570                  CLR     FREQ1
015170  013737  001564  001566          MOV     FR3,    FREQ
015176  104401                          TYPE
015200  022366                          MQ
015202  104411                          RDDEC                   ;GET NUMBER.
015204  012637  013134                  MOV     (6)+,   RTEMP
015210  001417                          BEQ     2$              ;IF ZERRO, USE DEFAULT
015212  005037  001570                  CLR     FREQ1
015216  005037  001566                  CLR     FREQ
015222  063737  001576  001566  1$:     ADD     FR1,    FREQ    ;SET FREQUECY (OR DELAY).
015230  005537  001570                  ADC     FREQ1
015234  005737  001570                  TST     FREQ1
015240  100404                          BMI     3$
015242  005337  013134                  DEC     RTEMP
015246  001365                          BNE     1$
015250  000002                  2$:     EXIT

015252  104401                  3$:     TYPE                    ;TYPE" TIME TOO LONG".
015254  022274                          MTTL
015256  000735                          BR      RDELA


                                        ;*
                                        ;*ROUTINE TO FORM SECOUNDARY DELAY
                                        ;*
015260  104401  021751          RDELA3: TYPE,   MDEL1
015264  104401                  RDELA4: TYPE
015266  021754                          MDEL2
015270  104411                          RDDEC
015272  012637  013134                  MOV     (6)+,RTEMP
015276  001415                          BEQ     2$
015300  005037  001572                  CLR     FREQ2
015304  063737  001576  001566  1$:     ADD     FR1,FREQ
015312  005537  001574                  ADC     FREQ3
015316  005737  001574                  TST     FREQ3
015322  100405                          BMI     3$
015324  005337  013134                  DEC     RTEMP
015330  001365                          BNE     1$
015332  000137  015036          2$:     JMP     RASK
```

```
2585  015336  104401                  3$:     TYPE
2586  015340  022274                          MTTL
2587  015342  000750                          BR      RDELAY
2588
2589
2590                                          ;*
2591                                          ;*ROUTINE TO SET UNI OR BIPOLAR A/D
2592                                          ;*CALL = QUBR
2593                                          ;*
2594
2595  015344  005737  001652          RQUBR:  TST     EXPERT
2596  015350  001002                          BNE     .+6
2597  015352  104401                          TYPE                    ;UNI OR BI-POLAR?
2598  015354  024700                          MQUR8
2599  015356  104401                          TYPE
2600  015360  022366                          MQ
2601  015362  005037  015046                  CLR     CHAR
2602  015366  105777  163552          1$:     TSTB    @$TKS           ;WAIT FOR RESPONSE
2603  015372  100375                          BPL     1$
2604  015374  105777  163550          2$:     TSTB    @$TPS           ;ECHO.
2605  015400  100375                          BPL     2$
2606  015402  017737  163540  013134          MOV     @$TKB,RTEMP
2607  015410  013777  013134  163534          MOV     RTEMP,@$TPB
2608  015416  042737  000240  013134          BIC     #240,RTEMP
2609  015424  023727  013134  000003          CMP     RTEMP,#3        ;^C TYPED?
2610  015432  001002                          BNE     .+6
2611  015434  000137  003374                  JMP     START1
2612  015440  023727  013134  000015          CMP     RTEMP,#15       ;CR TYPED?
2613  015446  001404                          BEQ     3$
2614  015450  013737  013134  015046          MOV     RTEMP,CHAR
2615  015456  000743                          BR      1$
2616  015460  104401  001163          3$:     TYPE,   $CRLF
2617  015464  023727  015046  000125          CMP     CHAR,#'U
2618  015472  001003                          BNE     4$
2619  015474  005237  013342                  INC     BIPOL
2620  015500  000002                          EXIT
2621  015502  005037  013342          4$:     CLR     BIPOL
2622  015506  000002                          EXIT
2623
2624                                          ;*
2625                                          ;*ROUTINE TO CHECK THAT AN ADDRESS TYPED IS A LEGAL ICS-ADDR.
2626                                          ;*TYPE ERROR IF NOT - ENTER WITH ADDR IN (0)
2627                                          ;*CALL = JSR PC,CKADR
2628                                          ;*
2629
2630  015510  013737  000004  015610  CKADR:  MOV     @#4,3$          ;STORE LOCATION 4.
2631  015516  012737  015560  000004          MOV     #2$,@#4         ;SET TIME-OUT LOCATION FOR TRAP IF ANY.
2632  015524  005770  000000                  TST     @(0)            ;TEST MOD. ADDR. TYPED.
2633  015530  000240                          NOP
2634                                          ;**
2635                                          ;** MODULE ADDRESS DOES EXIST BUT MAY NOT BE ASSOCIATED WITH THIS
2636                                          ;** FILE BOX. THEREFORE, CHECK ADDRESS FOR WITHIN FILE BOX LIMITS
2637                                          ;**
2638
2639  015532  013737  015610  000004          MOV     3$,@#4          ;RESTORE TIME OUT VECTOR
2640  015540  023710  001702                  CMP     ICSLMT,(R0)
```

```
2641  015544  003022                        BGT     4$
2642  015546  023710  001676                CMP     ICSHGH,(R0)
2643  015552  003417                        BLE     4$
2644  015554  005720                1$:     TST     (0)+
2645  015556  000207                        RTS     PC              ;EXIT - SUBROUTINE.
2646  015560  104401  022517        2$:     TYPE,   MNRFN           ;TYPE ERROR MESSAGE.
2647  015564                                TYPOCT  (0)             ;TYPE ADDR.
2648  015564  011046                        MOV     (0),-(SP)       ;;SAVE (0) FOR TYPEOUT
2649  015566  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2650  015570  062706  000006                ADD     #6,R6
2651  015574  162716  000002                SUB     #2,(6)          ;FIX RETURN ADDRESS ON STACK
2652  015600  013737  015610  000004        MOV     3$,@#4          ;RESTORE VECTOR ADDRESS
2653  015606  000002                        EXIT                    ;SO THAT WE RETURN AND REASK
2654                                                                ;QUESTION O; MODULE ADDR.
2655  015610  000000                3$:     .WORD   0               ;TEMP STORAGE OF LOC. 4.
2656
2657
2658  015612  104401  026763        4$:     TYPE, NRANG1            ;NOT WITHIN FILE
2659  015616                                TYPOCT  (0)
2660  015616  011046                        MOV     (0),-(SP)       ;;SAVE (0) FOR TYPEOUT
2661  015620  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2662  015622  104401  026776                TYPE, NRANG2
2663  015626  005726                        TST     (R6)+           ;POP STACK PAST JSR RETURN
2664  015630  162716  000002                SUB     #2,(R6)         ;BACK POINTER TO RE-ASK QUESTION
2665  015634  000002                        EXIT                    ;RETURN
2666
2667
2668                                        ;*
2669                                        ;*ROUTINE USED OFR GENERATION A DELAY
2670                                        ;*CALL=DELAT
2671                                        ;*
2672
2673  015636  013737  001566  013134  RDELAY: MOV     FREQ,   RTEMP   ;GET DELAY TIME
2674  015644  001411                        BEQ     2$
2675  015646  013737  001570  015672        MOV     FREQ1,  RTEMP2
2676  015654  005337  013134          1$:   DEC     RTEMP           ;DELAY
2677  015660  001375                        BNE     1$
2678  015662  005337  015672                DEC     RTEMP2
2679  015666  100372                        BPL     1$
2680  015670  000002                2$:     EXIT                    ;RETURN
2681  015672  000000                RTEMP2: 0
2682
2683                                        ;*
2684                                        ;*ROUTINE TO GENERATE SECOUNDARY DELAY
2685                                        ;*CALL =DELAY2
2686                                        ;*
2687  015674  013737  001572  013134  RDELA2: MOV     FREQ2,RTEMP
2688  015702  001413                        BEQ     2$
2689  015704  013737  001574  015672        MOV     FREQ3,RTEMP2
2690  015712  005037  177776                CLR     PS
2691  015716  005337  013134          1$:   DEC     RTEMP
2692  015722  001375                        BNE     1$
2693  015724  005337  015672                DEC     RTEMP2
2694  015730  100372                        BPL     1$
2695  015732  000002                2$:     EXIT
2696
```

```
2697
2698                                        ;*
2699
2700
2701                                        ;* SYSTEM INITIALIZE ROUTINE
2702                                        ;*
2703
2704  015734                          RINIT:
2705  015734  013700  001104                MOV     $ICNT,R0
2706  015740  000005                        RESET
2707  015742  032777  000001  163170        BIT     #1,@SWR
2708  015750  001402                        BEQ     1$
2709  015752  000137  003374                JMP     START1
2710
2711  015756  052777  000100  163160  1$:   BIS     #100,   @STKS
2712  015764  000002                        EXIT
2713
2714
2715                                        ;*
2716                                        ;*ROUTINE USED TO MODIFY PATTERN SENT TO OUTPUT MODULE
2717                                        ;*FIRST LOCATION MAY BE CHANGED BY RPATA (PATTERN MODIFIER INPUT ROUTINE)
2718                                        ;*CALL=CPATR
2719                                        ;*
2720
2721  015766  023727  001602  000012  RCPAT:  CMP     PATRNM,#12
2722  015774  001442                        BEQ     RCPATR
2723  015776  000241                        CLC                             ;CLEAR C BIT
2724  016000  005237  001604          RCPATI: INC     @#PATRN                 ;MODIFY PATTERN
2725  016004  103412                        BCS     1$
2726  016006  022737  005137  016000        CMP     #5137,RCPATI            ;DOING A COMPLEMENT PATERN?
2727  016014  001406                        BEQ     1$                      ;IF SO DON'T ADD CARRY IF ANY!
2728  016016  063737  001606  001604        ADD     PATRNC,PATRN
2729  016024  005037  001606                CLR     PATRNC
2730  016030  000002                        EXIT
2731  016032  023727  016000  006037  1$:   CMP     RCPATI,#6037
2732  016040  001004                        BNE     2$
2733  016042  012737  100000  001606        MOV     #100000,PATRNC
2734  016050  000002                        EXIT
2735  016052  012737  000001  001606  2$:   MOV     #1,PATRNC
2736  016060  000002                        EXIT
2737                                        ;*MODIFIER LIST
2738  016062  005237                  RCPATL: 5237                    ;*INCREMENTING PATTERN          0
2739  016064  005337                          5337                    ;*DECREMENTING PATTERN          1
2740  016066  005737                          5737                    ;*NO CHANGE OF PATTERN          2
2741  016070  006137                          6137                    ;*ROTATE LEFT PATTERN           3
2742  016072  006037                          6037                    ;*ROTATE RIGHT PATTERN          4
2743  016074  006237                          6237                    ;*RANDOM NUMBER GENERATOR        5
2744  016076  006337                          6337                    ;*ARITH. SHIFT LEFT PATTERN     6
2745  016100  005137                          5137                    ;*COMPLEMENT PATTERN            7
2746
2747
2748                                        ;*
2749                                        ;*ALTERNATE CHANGE PATTERN ROUTINE IF RANDOME
2750                                        ;*NUMBER FOR PATTERN IS SELECTED.
2751                                        ;*
2752  016102                          RCPATR:
```

```
2753  016102  004737  016336                JSR     PC,$RAND        ;GENERATE A RANDOM NUMBER.
2754  016106  013737  016436  001604        MOV     $LONUM,PATRN    ;PUT NUMBER IN PATTERN.
2755  016114  000002                        EXIT
2756
2757
2758                                        ;*
2759                                        ;*PATTERN INPUTTER AND MODIFIER ROUTINE
2760                                        ;*FORM PATTERN MODIFIER, PATTERN.
2761                                        ;*CALL=PATAR
2762                                        ;*
2763  016116  005737  001652        RPATA:  TST     EXPERT
2764  016122  001002                        BNE     .+6
2765  016124  104401                        TYPE                    ;ASK OPERATOR-"PATTERN MODIFIER AND
2766
2767  016126  022316                        MPPM                    ;PATTERN?"
2768  016130  012737  001602  016162        MOV     #PATRNM, 1$     ;SET TO INPUT PATTERN + MODIFIER
2769  016136  012737  000002  016164        MOV     #2,     1$+2
2770  016144  005037  001602                CLR     PATRNM          ;SET DEFAULTS: 0 PATTERN
2771  016150  005037  001604                CLR     PATRN           ;PATTERN MOD. = INC.
2772  016154  000401                        BR      .+4
2773  016156  000757                        BR      RPATA
2774  016160  104414                        INOCT                   ;GET THEM.
2775  016162  001602                1$:     PATRNM
2776  016164  000002                        2
2777  016166  012737  016062  013134        MOV     #RCPATL,RTEMP   ;POINT TO BEGINNING OF MODIFIER LIST.
2778  016174  032737  177770  001602        BIT     #177770,PATRNM  ;LEGAL PATTERN MODIFIER?
2779  016202  001403                        BEQ     2$              ;IF YES 2$
2780  016204  104401  025730                TYPE,   MIVP            ;NO TYPE ERROR MESSAGE.
2781  016210  000742                        BR      RPATA           ;REASK QUESTION.
2782  016212                        2$:
2783  016212  006337  001602                ASL     PATRNM
2784  016216  042737  177761  001602        BIC     #177761, PATRNM ;FIX ADDITIVE.
2785  016224  063737  001602  013134        ADD     PATRNM, RTEMP   ;POINT TO MODIFIER.
2786  016232  017737  174676  016000        MOV     @RTEMP, RCPAT+12        ;CHANGE PATTERN MODIFIER ROUTINE.
2787  016240  000002                        EXIT                    ;RETURN.
2788  016242                                .$SAVE
2789                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
2790
2791  016242                                STARS
2792                                ;;*****************************************************************
2793                                ;*SAVE R0-R5
2794                                ;*CALL:
2795                                ;*      SAVREG
2796                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
2797                                ;*
2798                                ;*TOP---(+16)
2799                                ;* +2---(+18)
2800                                ;* +4---R5
2801                                ;* +6---R4
2802                                ;* +8---R3
2803                                ;*+10---R2
2804                                ;*+12---R1
2805                                ;*+14---R0
2806
2807  016242                        $SAVREG:
2808  016242                                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     22(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
                                $RESREG:
                                        MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
                                        MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
                                        MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
                                        MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
                                        POP     <R5,R4,R3,R2,R1,R0>
                                        MOV     (SP)+,R5        ;;POP STACK INTO R5
                                        MOV     (SP)+,R4        ;;POP STACK INTO R4
                                        MOV     (SP)+,R3        ;;POP STACK INTO R3
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        RTI
                                        .$RAND
                                .SBTTL  RANDOM NUMBER GENERATOR ROUTINE

                                STARS
                                ;;*****************************************************************
                                ;*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

                                $RAND:  PUSH    <R0,R1,R2>
                                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                        MOV     $LONUM,R0       ;;SET R0 WITH LOW
                                        MOV     $HINUM,R1       ;;SET R1 WITH HIGH
                                        MOV     #-7,R2          ;;SET SHIFT COUNT
                                1$:     ASL     R0              ;;SHIFT R0 LEFT AND
                                        ROL     R1              ;;ROTATE CARRY INTO R1 AND
                                        INC     R2              ;;CHECK FOR DONE
                                        BNE     1$              ;;CONTINUE SHIFT LOOP
                                        ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
                                        ADC     R1              ;;PROPOGATE CARRY
                                        ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
                                        ADD     #1057,R0        ;;ADD LOW CONSTANT
```

```
2865  016406  005501                        ADC     R1              ;;PROPOGATE CARRY
2866  016410  062701  047401                ADD     #47401,R1       ;;ADD HIGH CONSTANT
2867  016414  010037  016436                MOV     R0,$LONUM       ;;SAVE R0
2868  016420  010137  016434                MOV     R1,$HINUM       ;;SAVE R1
2869  016424                                POP     <R2,R1,R0>
2870  016424  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
2871  016426  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
2872  016430  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
2873  016432  000207                        RTS     PC              ;;RETURN
2874  016434  176543              $HINUM:   .WORD   176543
2875  016436  123456              $LONUM:   .WORD   123456
2876  016440                                .$B2D
2877                               .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
2878
2879  016440                       STARS
2880                               ;;*****************************************************************
2881                               ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
2882                               ;*UNSIGNED DECIMAL ASCIZ NUMBER.
2883                               ;*CALL
2884                               ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
2885                               ;*      JSR     PC,@#$B2D       ;;CALL
2886                               ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR.IS ON THE STACK
2887
2888
2889  016440  016637  000002  016470 $B2D:   MOV     2(SP),1$        ;;SAVE BINARY NUMBER
2890  016446  012746  016470              MOV     #1$,-(SP)       ;;SET POINTER
2891  016452  004737  016474              JSR     PC,@#$DB2D      ;;CALL DOUBLE LENGTH CONVERT
2892  016456  062716  000005              ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
2893  016462  012666  000002              MOV     (SP)+,2(SP)     ;;PICKUP POINTER
2894  016466  000207                       RTS     PC              ;;RETURN
2895  016470  000000  000000       1$:     .WORD   0,0
2896  016474                               .$DB2D
2897                               .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
2898
2899  016474                       STARS
2900                               ;;*****************************************************************
2901                               ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
2902                               ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
2903                               ;*POSITIVE.
2904                               ;*CALL
2905                               ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
2906                               ;*      JSR     PC,@#$DB2D
2907                               ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
2908                                                               ;;IS ON THE STACK
2909
2910
2911  016474  104412               $DB2D:  SAVREG                  ;;SAVE REGISTERS
2912  016476  016602  000002              MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
2913  016502  012700  016654              MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
2914  016506  010066  000002              MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
2915  016512  012201                       MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
2916  016514  012202                       MOV     (R2)+,R2
2917  016516  012737  000012  016572      MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
2918  016524  012704  016604              MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
2919  016530  012705  016606              MOV     #$TNPWR+2,R5
2920  016534  005003               1$:     CLR     R3              ;;CLEAR PARTIAL
```

```
2921  016536  161401                  2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
2922  016540  005602                          SBC     R2
2923  016542  161502                          SUB     (R5),R2
2924  016544  002402                          BLT     3$              ;;BR IF TEN POWER TO LARGE
2925  016546  005203                          INC     R3              ;;ADD 1 TO PARTIAL
2926  016550  000772                          BR      2$              ;;LOOP
2927  016552  062401                  3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
2928  016554  005502                          ADC     R2
2929  016556  062402                          ADD     (R4)+,R2
2930  016560  022525                          CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
2931  016562  052703  000060                  BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
2932  016566  110320                          MOVB    R3,(R0)+        ;;SAVE IT
2933  016570  005327                          DEC     (PC)+           ;;DONE?
2934  016572  000000                  4$:     .WORD   0
2935  016574  001357                          BNE     1$              ;;BR IF NO
2936  016576  105020                          CLRB    (R0)+           ;;TERMINATOR
2937  016600  104413                          RESREG                  ;;RESTORE REGISTERS
2938  016602  000207                          RTS     PC              ;;RETURN
2939  016604  145000                  $TNPWR: 145000                  ;;1.0E09
2940  016606  035632                          35632
2941  016610  160400                          160400                  ;;1.0E08
2942  016612  002765                          2765
2943  016614  113200                          113200                  ;;1.0E07
2944  016616  000230                          230
2945  016620  041100                          041100                  ;;1.0E06
2946  016622  000017                          17
2947  016624  103240                          103240                  ;;1.0E05
2948  016626  000001                          1
2949  016630  023420                          23420                   ;;1.0E04
2950  016632  000000                          0
2951  016634  001750                          1750                    ;;1.0E03
2952  016636  000000                          0
2953  016640  000144                          144                     ;;1.0E02
2954  016642  000000                          0
2955  016644  000012                          12                      ;;1.0E01
2956  016646  000000                          0
2957  016650  000001                          1                       ;;1.0E00
2958  016652  000000                          0
2959  016654  000014                  $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
2960
2961
2962
2963
2964
2965  016670                                  .$ERROR $ERRTYP
2966                                  .SBTTL  ERROR HANDLER ROUTINE
2967
2968  016670                          STARS
2969                                  ;;*****************************************************************
2970                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
2971                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
2972                                  ;*AND GO TO $ERRTYP ON ERROR
2973                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2974                                  ;*SW15=1         HALT ON ERROR
2975                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
2976                                  ;*CALL
```

```
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

016670                                  $ERROR:
016670  104406                                  CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
016672  105237  001103                  7$:     INCB    $ERFLG                  ;;SET THE ERROR FLAG
016676  001775                          2$      BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
016700  013777  001102  162234                  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER AND ERROR FLAG
016706  005237  001112                          INC     $ERTTL                  ;;INC THE ERROR COUNT
016712  011637  001116                          MOV     (SP),$ERRPC             ;;GET ADDRESS OF ERROR INSTRUCTION
016716  162737  000002  001116                  SUB     #2,$ERRPC
016724  117737  162166  001114                  MOVB    @$ERRPC,$ITEMB          ;;STRIP AND SAVE THE ERROR ITEM CODE
016732  032777  020000  162200                  BIT     #BIT13,@SWR             ;;SKIP TYPEOUT IF SET
016740  001004                                  BNE     20$                     ;;SKIP TYPEOUTS
016742  004737  016766                          JSR     PC,$ERRTYP              ;;GO TO USER ERROR ROUTINE
016746  104401  001163                          TYPE    ,$CRLF
016752                                  20$:
016752  005777  162162                  2$:     TST     @SWR                    ;;HALT ON ERROR
016756  100002                                  BPL     3$                      ;;SKIP IF CONTINUE
016760  000000                                  HALT                            ;;HALT ON ERROR!
016762  104406                                  CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
016764                                  3$:
016764  000002                                  RTI                             ;;RETURN
016766                                          .$ERRTYP
                                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

016766                                  STARS
                                        ;;**************************************************************
                                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                        ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                        ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

016766                                  $ERRTYP:
016766  104401  001163                          TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
016772  010046                                  MOV     R0,-(SP)                ;;SAVE R0
016774  005000                                  CLR     R0                      ;;PICKUP THE ITEM INDEX
016776  153700  001114                          BISB    @#$ITEMB,R0
017002  001004                                  BNE     1$                      ;;IF ITEM NUMBER IS ZERO, JUST
                                                                                ;;TYPE THE PC OF THE ERROR
017004                                          TYPOCT  $ERRPC,<ERROR ADDRESS>
017004  013746  001116                          MOV     $ERRPC,-(SP)            ;;SAVE $ERRPC FOR TYPEOUT
                                                                                ;;ERROR ADDRESS
017010  104402                                  TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
017012  000426                                  BR      6$                      ;;GET OUT
017014  005300                          1$:     DEC     R0                      ;;ADJUST THE INDEX SO THAT IT WILL
017016  006300                                  ASL     R0                      ;;      WORK FOR THE ERROR TABLE
017020  006300                                  ASL     R0
017022  006300                                  ASL     R0
017024  062700  001166                          ADD     #$ERRTB,R0              ;;FORM TABLE POINTER
017030  012037  017040                          MOV     (R0)+,2$                ;;PICKUP "ERROR MESSAGE" POINTER
017034  001404                                  BEQ     3$                      ;;SKIP TYPEOUT IF NO POINTER
017036  104401                                  TYPE                            ;;TYPE THE "ERROR MESSAGE"
017040  000000                          2$:     .WORD   0                       ;;"ERROR MESSAGE" POINTER GOES HERE
017042  104401  001163                          TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
017046  012037  017056                  3$:     MOV     (R0)+,4$                ;;PICKUP "DATA HEADER" POINTER
017052  001404                                  BEQ     5$                      ;;SKIP TYPEOUT IF 0
017054  104401                                  TYPE                            ;;TYPE THE "DATA HEADER"
```

```
                                4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
                                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
                                5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
                                        BNE     7$              ;;GO TYPE THE DATA
                                6$:     MOV     (SP)+,R0        ;;RESTORE R0
                                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
                                        RTS     PC              ;;RETURN
                                7$:     TYPOCT  @(R0)+          ;;TYPE AN OCTAL NUMBER
                                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
                                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
                                        BEQ     6$              ;;BR IF NO
                                        TYPE    ,8$             ;;TYPE TWO(2) SPACES
                                        BR      7$              ;;LOOP
017116  020040  000             8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
017122                                  .EVEN
                                        .$TYPOCT
017122                          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

017122                          STARS
                                ;;**************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

017122  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
017126  116637  000001  017345          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
017134  112637  017347                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
017140  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
017144  000406                          BR      $TYPON
017146  112737  000001  017345  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
017154  112737  000006  017347          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
017162  112737  000005  017344  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
017170  010346                          MOV     R3,-(SP)        ;;SAVE R3
017172  010446                          MOV     R4,-(SP)        ;;SAVE R4
017174  010546                          MOV     R5,-(SP)        ;;SAVE R5
017176  113704  017347                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
017202  005404                          NEG     R4
```

```
3089  017204  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3090  017210  110437  017346                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3091  017214  113704  017345                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3092  017220  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3093  017224  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
3094  017226  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3095  017230  000404                        BR      3$              ;;GO DO MSB
3096  017232  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
3097  017234  006105                        ROL     R5
3098  017236  006105                        ROL     R5
3099  017240  010503                        MOV     R5,R3
3100  017242  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3101  017244  105337  017346                DECB    $OMODE          ;;TYPE THIS DIGIT?
3102  017250  100016                        BPL     7$              ;;BR IF NO
3103  017252  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
3104  017256  001002                        BNE     4$              ;;TEST FOR 0
3105  017260  005704                        TST     R4              ;;SUPPRESS THIS 0?
3106  017262  001403                        BEQ     5$              ;;BR IF YES
3107  017264  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3108  017266  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3109  017272  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3110  017276  110337  017342                MOVB    R3,8$           ;;SAVE FOR TYPING
3111  017302  104401  017342                TYPE    ,8$             ;;GO TYPE THIS DIGIT
3112  017306  105337  017344        7$:     DECB    $OCNT           ;;COUNT BY 1
3113  017312  003347                        BGT     2$              ;;BR IF MORE TO DO
3114  017314  002402                        BLT     6$              ;;BR IF DONE
3115  017316  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3116  017320  000744                        BR      2$              ;;GO DO THE LAST DIGIT
3117  017322  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
3118  017324  012604                        MOV     (SP)+,R4        ;;RESTORE R4
3119  017326  012603                        MOV     (SP)+,R3        ;;RESTORE R3
3120  017330  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3121  017336  012616                        MOV     (SP)+,(SP)
3122  017340  000002                        RTI                     ;;RETURN
3123  017342     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3124  017343     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3125  017344     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3126  017345     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3127  017346  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3128  017350                                .SPOWER ,,PWRMSG,START
3129                                .SBTTL  POWER DOWN AND UP ROUTINES
3130
3131  017350                        STARS
3132                                ;;**********************************************************************
3133                                ;POWER DOWN ROUTINE
3134  017350  012737  017514  000024 $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
3135  017356  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
3136  017364                                PUSH    <R0,R1,R2,R3,R4,R5>
3137  017364  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3138  017366  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3139  017370  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3140  017372  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
3141  017374  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
3142  017376  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
3143  017400                                PUSH    @SWR
3144  017400  017746  161534                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
```

```
3145  017404  010637  017520                  MOV     SP,$SAVR6       ;;SAVE SP
3146  017410  012737  017422  000024          MOV     #$PWRUP,@#PWRVEC';;SET UP VECTOR
3147  017416  000000                          HALT
3148  017420  000776                          BR      .-2             ;;HANG UP
3149
3150  017422                          STARS
3151                                  ;;**************************************************************
3152                                  ;POWER UP ROUTINE
3153  017422  012737  017514  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
3154  017430  013706  017520                  MOV     $SAVR6,SP       ;;GET SP
3155  017434  005037  017520                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
3156  017440  005237  017520          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
3157  017444  001375                          BNE     1$              ;;OF  WORD
3158  017446                                  POP     @SWR
3159  017446  012677  161466                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
3160  017452                                  POP     <R5,R4,R3,R2,R1,R0>
3161  017452  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
3162  017454  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
3163  017456  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
3164  017460  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
3165  017462  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
3166  017464  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3167  017466  012737  017350  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
3168  017474  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
3169  017502  104401                          TYPE                    ;;REPORT THE POWER FAILURE
3170  017504  025176                  $PWRMG: .WORD   PWRMSG           ;;POWER FAIL MESSAGE POINTER
3171  017506  012716                          MOV     (PC)+,(SP)      ;;RESTART AT START
3172  017510  001704                  $PWRAD: .WORD   START           ;;RESTART ADDRESS
3173  017512  000002                          RTI
3174  017514  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
3175  017516  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
3176  017520  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
3177
3178  017522                                  .$SCOPE
3179                                  .SBTTL  SCOPE HANDLER ROUTINE
3180
3181  017522                          STARS
3182                                  ;;**************************************************************
3183                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3184                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3185                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3186                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3187                                  ;*SW14=1         LOOP ON TEST
3188                                  ;*SW11=1         INHIBIT ITERATIONS
3189                                  ;*CALL
3190                                  ;*      SCOPE           ;;SCOPE=IOT
3191
3192  017522                          $SCOPE:
3193  017522  104406                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
3194  017524  032777  040000  161406  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
3195  017532  001055                          BNE     $OVER           ;;YES IF SW14=1
3196                                  ;#####START OF CODE FOR THE XOR TESTER#####
3197  017534  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3198                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3199  017536  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3200  017542  012737  017562  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
```

```
3201  017550  005737  177060            TST     @#177060        ;;TIME OUT ON XOR?
3202  017554  012637  000004            MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3203  017560  000436                    BR      $SVLAD          ;;GO TO THE NEXT TEST
3204  017562  022626                5$: CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
3205  017564  012637  000004            MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3206  017570  000436                    BR      $OVER           ;;LOOP ON THE PRESENT TEST
3207  017572                        6$:;#####END OF CODE FOR THE XOR TESTER#####
3208  017572  105737  001103        2$: TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
3209  017576  001404                    BEQ     3$              ;;BR IF NO
3210  017600  105037  001103        4$: CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
3211  017604  005037  001160            CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3212  017610  032777  004000  161322 3$: BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
3213  017616  001011                    BNE     1$              ;;BR IF YES
3214  017620  005737  001100            TST     $PASS           ;;IF FIRST PASS OF PROGRAM
3215  017624  001406                    BEQ     1$              ;;     INHIBIT ITERATIONS
3216  017626  005237  001104            INC     $ICNT           ;;INCREMENT ITERATION COUNT
3217  017632  023737  001160  001104    CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
3218  017640  002012                    BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
3219  017642  012737  000001  001104 1$: MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
3220  017650  013737  017702  001160    MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
3221  017656  105237  001102        $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
3222  017662  011637  001106            MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
3223  017666  013777  001102  161246 $OVER: MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
3224  017674  013716  001106            MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
3225  017700  000002                    RTI                     ;;FIXES PS
3226  017702  003720                $MXCNT: 2000.               ;;MAX. NUMBER OF ITERATIONS
3227
3228  017704                            .$READ
3229                                .SBTTL  TTY INPUT ROUTINE
3230
3231  017704                            STARS
3232                                ;;**********************************************************************
3233                                .ENABL  LSB
3234
3235  017704                            STARS
3236                                ;;**********************************************************************
3237                                ;#SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
3238                                ;#ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
3239                                ;#SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
3240                                ;#WHEN OPERATING IN TTY FLAG MODE.
3241  017704  022737  000176  001140 $CKSWR: CMP     #$WREG,SWR      ;;IS THE SOFT-SWR SELECTED?
3242  017712  001074                    BNE     15$             ;;BRANCH IF NO
3243  017714  105777  161224            TSTB    @$TKS           ;;CHAR THERE?
3244  017720  100071                    BPL     15$             ;;IF NO, DON'T WAIT AROUND
3245  017722  117746  161220            MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
3246  017726  042716  177600            BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
3247  017732  022726  000007            CMP     #7,(SP)+        ;;IS IT A CONTROL G?
3248  017736  001062                    BNE     15$             ;;NO, RETURN TO USER
3249  017740  123727  001134  000001    CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
3250  017746  001456                    BEQ     15$             ;;BRANCH IF YES
3251
3252  017750  104401  020431            TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
3253  017754  104401  020436        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
3254  017760                            TYPOCT  $WREG           ;;OF THE SWR
3255  017760  013746  000176            MOV     $WREG,-(SP)     ;;SAVE $WREG FOR TYPEOUT
3256  017764  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
3257  017766  104401  020447            TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
3258  017772  005046            19$:    CLR     -(SP)           ;;CLEAR COUNTER
3259  017774  005046                    CLR     -(SP)           ;;THE NEW SWR
3260  017776  105777  161142      7$:    TSTB    @$TKS           ;;CHAR THERE?
3261  020002  100375                    BPL     7$              ;;IF NOT TRY AGAIN
3262
3263  020004  117746  161136            MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
3264  020010  042716  177600            BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
3265
3266
3267
3268  020014  021627  000025      9$:    CMP     (SP),#25        ;;IS IT A CONTROL-U?
3269  020020  001005                    BNE     10$             ;;BRANCH IF NOT
3270  020022  104401  020424            TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
3271  020026  062706  000006      20$:   ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
3272  020032  000757                    BR      19$             ;;LET'S TRY IT AGAIN
3273
3274
3275  020034  021627  000015      10$:   CMP     (SP),#15        ;;IS IT A <CR>?
3276  020040  001022                    BNE     16$             ;;BRANCH IF NO
3277  020042  005766  000004            TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
3278  020046  001403                    BEQ     11$             ;;BRANCH IF YES
3279  020050  016677  000002  161062    MOV     2(SP),@$WR      ;;SAVE NEW SWR
3280  020056  062706  000006      11$:   ADD     #6,SP           ;;CLEAR UP STACK
3281  020062  104401  001163      14$:   TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
3282  020066  123727  001135  000001    CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
3283  020074  001003                    BNE     15$             ;;BRANCH IF NOT
3284  020076  012777  000100  161040    MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
3285  020104  000002            15$:    RTI                     ;;RETURN
3286  020106  004737  001006      16$:   JSR     PC,$TYPEC       ;;ECHO CHAR
3287  020112  021627  000060            CMP     (SP),#60        ;;CHAR < 0?
3288  020116  002420                    BLT     18$             ;;BRANCH IF YES
3289  020120  021627  000067            CMP     (SP),#67        ;;CHAR > 7?
3290  020124  003015                    BGT     18$             ;;BRANCH IF YES
3291  020126  042726  000060            BIC     #60,(SP)+       ;;STRIP-OFF ASCII
3292  020132  005766  000002            TST     2(SP)           ;;IS THIS THE FIRST CHAR
3293  020136  001403                    BEQ     17$             ;;BRANCH IF YES
3294  020140  006316                    ASL     (SP)            ;;NO, SHIFT PRESENT
3295  020142  006316                    ASL     (SP)            ;;   CHAR OVER TO MAKE
3296  020144  006316                    ASL     (SP)            ;;   ROOM FOR NEW ONE.
3297  020146  005266  000002      17$:   INC     2(SP)           ;;KEEP COUNT OF CHAR
3298  020152  056616  177776            BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
3299  020156  000707                    BR      7$              ;;GET THE NEXT ONE
3300  020160  104401  001162      18$:   TYPE    ,$QUES          ;;TYPE ?<CR><LF>
3301  020164  000720                    BR      20$             ;;SIMULATE CONTROL-U
3302                            .DSABL  LSB
3303
3304
3305  020166                    STARS
3306                            ;;**************************************************************
3307                            ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
3308                            ;*CALL:
3309                            ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
3310                            ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
3311                            ;*                              ;;WITH PARITY BIT STRIPPED OFF
3312                            ;
```

```
3313
3314  020166  011646                    $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
3315  020170  016666  000004  000002            MOV     4(SP),2(SP)     ;;SAVE THE PS
3316  020176  105777  160742            1$:     TSTB    @$TKS           ;;WAIT FOR
3317  020202  100375                            BPL     1$              ;;A CHARACTER
3318  020204  117766  160736  000004            MOVB    @$TKB,4(SP)     ;;READ THE TTY
3319  020212  042766  177600  000004            BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
3320  020220  026627  000004  000023            CMP     4(SP),#23       ;;IS IT A CONTROL-S?
3321  020226  001013                            BNE     3$              ;;BRANCH IF NO
3322  020230  105777  160710            2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
3323  020234  100375                            BPL     2$              ;;LOOP UNTIL ITS THERE
3324  020236  117746  160704                    MOVB    @$TKB,-(SP)     ;;GET CHARACTER
3325  020242  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
3326  020246  022627  000021                    CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
3327  020252  001366                            BNE     2$              ;;IF NOT DISCARD IT
3328  020254  000750                            BR      1$              ;;YES, RESUME
3329  020256  026627  000004  000140  3$:      CMP     4(SP),#140      ;;IS IT UPPER CASE?
3330  020264  002407                            BLT     4$              ;;BRANCH IF YES
3331  020266  026627  000004  000175            CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
3332  020274  003003                            BGT     4$              ;;BRANCH IF YES
3333  020276  042766  000040  000004            BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
3334  020304  000002                    4$:     RTI                     ;;GO BACK TO USER
3335  020306                                    STARS
3336                                            ;;*****************************************************************
3337                                            ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
3338                                            ;*CALL:
3339                                            ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
3340                                            ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3341                                            ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
3342
3343  020306  010346                    $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
3344  020310  012703  020414            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
3345  020314  022703  020424            2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
3346  020320  101405                            BLOS    4$              ;;BR IF YES
3347  020322  104407                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
3348  020324  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
3349  020326  122713  000177            10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
3350  020332  001003                            BNE     3$              ;;SKIP IF NOT
3351  020334  104401  001162            4$:     TYPE    ,$QUES          ;;TYPE A '?'
3352  020340  000763                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
3353  020342  111337  020412            3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
3354  020346  104401  020412                    TYPE    ,9$
3355  020352  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
3356  020356  001356                            BNE     2$              ;;LOOP IF NOT RETURN
3357  020360  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
3358  020364  104401  001164                    TYPE    ,$LF            ;;TYPE A LINE FEED
3359  020370  012603                            MOV     (SP)+,R3        ;;RESTORE R3
3360  020372  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
3361  020374  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
3362  020402  012766  020414  000004            MOV     #$TTYIN,4(SP)
3363  020410  000002                            RTI                     ;;RETURN
3364  020412     000                    9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
3365  020413     000                            .BYTE   0               ;;TERMINATOR
3366  020414  000010                    $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
3367  020424  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
3368  020431     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
```

```
3369  020436  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
3370  020444  020075     000
3371  020447     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
3372  020454  036440  000040
3373  020460                          $RDDEC
3374                                  .SBTTL  READ A DECIMAL NUMBER FROM THE TTY
3375
3376  020460                          STARS
3377                                  ;;**********************************************************************
3378                                  ;*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
3379                                  ;*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
3380                                  ;*ARE READ A "?" FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
3381                                  ;*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
3382                                  ;*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
3383                                  ;*POSITIVE 32767 TO NEGATIVE 32768.
3384                                  ;*CALL:
3385                                  ;*      RDDEC                   ;;READ A DECIMAL NUMBER
3386                                  ;*      RETURN HERE             ;;NUMBER IS ON TOP OF THE STACK
3387                                  ;
3388
3389  020460  011646                  $RDDEC: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR
3390  020462  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE INPUT NUMBER
3391  020470                          PUSH    <R0,R1,R2>
3392  020470  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
3393  020472  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3394  020474  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3395  020476  104410                  1$:     RDLIN           ;;READ AN ASCIZ LINE
3396  020500  012600                  MOV     (SP)+,R0        ;;ADDRESS OF 1ST CHAR.
3397  020502  010037  020626          MOV     R0,6$           ;;SAVE INCASE OF BAD INPUT
3398  020506  005046                  CLR     -(SP)           ;;CLEAR DATA WORD
3399  020510  005002                  CLR     R2              ;;SIGN SET POSITIVE
3400  020512  122710  000055          CMPB    #'-,(R0)        ;;SEE IF A MINUS SIGN WAS TYPED
3401  020516  001001                  BNE     2$              ;;BR IF NO MINUS SIGN
3402  020520  112002                  MOVB    (R0)+,R2        ;;SAVE FOR LATER USE
3403  020522  112001                  2$:     MOVB    (R0)+,R1        ;;PICKUP THIS CHARACTER
3404  020524  001424                  BEQ     3$              ;;GET OUT IF ZERO
3405  020526  122701  000060          CMPB    #'0,R1          ;;MAKE SURE THIS CHARACTER
3406  020532  003032                  BGT     5$              ;;IS A DIGIT BETWEEN 0 & 9
3407  020534  122701  000071          CMPB    #'9,R1
3408  020540  002427                  BLT     5$
3409  020542  032716  170000          BIT     #^C7777,(SP)    ;;DON'T LET NUMBER GET TO BIG
3410  020546  001024                  BNE     5$              ;;BR IF NUMBER WOULD OVERFLOW
3411  020550  006316                  ASL     (SP)            ;;*2
3412  020552  011646                  MOV     (SP),-(SP)      ;;SAVE FOR LATER
3413  020554  006316                  ASL     (SP)            ;;*4
3414  020556  006316                  ASL     (SP)            ;;*8.
3415  020560  062616                  ADD     (SP)+,(SP)      ;;*10.
3416  020562  102416                  BVS     5$              ;;OVERFLOW ISN'T ALLOWED
3417  020564  162701  000060          SUB     #'0,R1          ;;STRIP AWAY THE ASCII JUNK
3418  020570  060116                  ADD     R1,(SP)         ;;ADD IN THIS DIGIT
3419  020572  102412                  BVS     5$              ;;OVERFLOW ISN'T ALLOWED
3420  020574  000752                  BR      2$              ;;LOOP
3421  020576  005702                  3$:     TST     R2              ;;CHECK IF NUMBER IS NEG
3422  020600  001401                  BEQ     4$              ;;BR IF NO
3423  020602  005416                  NEG     (SP)            ;;YES--NEGATE THE NUMBER
3424  020604  012666  000012          4$:     MOV     (SP)+,12(SP)    ;;SAVE THE RESULT
```

```
3425  020610                          POP     <R2,R1,R0>
3426  020610  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
3427  020612  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
3428  020614  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
3429  020616  000002                  RTI                     ;;RETURN
3430
3431  020620  005726          5$:     TST     (SP)+           ;;CLEAN PARTIAL NUMBER FROM STACK
3432  020622  105010                  CLRB    (R0)            ;;SET A TERMINATOR
3433  020624  104401                  TYPE                    ;;TYPE THE INPUT UP TO BAD CHAR.
3434  020626  000000          6$:     .WORD   0               ;;POINTER GOES HERE
3435  020630  104401  001162          TYPE    ,$QUES          ;;"?" "CR" &"LF"
3436  020634  000720                  BR      1$              ;;TRY AGAIN
3437
3438  020636                          .$TYPE
3439                          .SBTTL  TYPE ROUTINE
3440
3441  020636                          STARS
3442                          ;;**********************************************************************
3443                          ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
3444                          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3445                          ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3446                          ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3447                          ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3448                          ;*
3449                          ;*CALL:
3450                          ;*1) USING A TRAP INSTRUCTION
3451                          ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3452                          ;*OR
3453                          ;*      TYPE
3454                          ;*      MESADR
3455                          ;*
3456
3457  020636  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
3458  020642  100002                  BPL     1$              ;;BR IF YES
3459  020644  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
3460  020646  000407                  BR      3$              ;;LEAVE
3461  020650  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
3462  020652  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
3463  020656  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
3464  020660  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
3465  020662  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
3466  020664  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
3467  020666  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
3468  020672  000002                  RTI                     ;;RETURN
3469  020674  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
3470  020700  001430                  BEQ     8$
3471  020702  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
3472  020706  001006                  BNE     5$
3473  020710  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
3474  020712  104401                  TYPE                    ;;TYPE A CR AND LF
3475  020714  001163                  $CRLF
3476  020716  105037  021052          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
3477  020722  000755                  BR      2$              ;;GET NEXT CHARACTER
3478  020724  004737  021006  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
3479  020730  123726  001156  6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
3480  020734  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
```

```
3481  020736  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
3482                                                                    ;;AND THE NULL CHAR.
3483  020742  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
3484  020746  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
3485  020750  004737  021006                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
3486  020754  105337  021052                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
3487  020760  000770                          BR      7$              ;;LOOP
3488
3489                                          ;HORIZONTAL TAB PROCESSOR
3490
3491  020762  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3492  020766  004737  021006          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
3493  020772  132737  000007  021052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3494  021000  001372                          BNE     9$              ;;TAB STOP
3495  021002  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
3496  021004  000724                          BR      2$              ;;GET NEXT CHARACTER
3497  021006  105777  160136          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
3498  021012  100375                          BPL     $TYPEC
3499  021014  116677  000002  160130          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3500  021022  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
3501  021030  001003                          BNE     1$              ;;BRANCH IF NO
3502  021032  105037  021052                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3503  021036  000406                          BR      $TYPEX          ;;EXIT
3504  021040  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
3505  021046  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
3506  021050  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
3507  021052  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
3508  021054  000207          $TYPEX: RTS     PC
3509
3510
3511
3512
3513
3514  021056                              .$TRAP
3515                                      .SBTTL  TRAP DECODER
3516
3517  021056                              STARS
3518                                      ;;**********************************************************************
3519                                      ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3520                                      ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3521                                      ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3522                                      ;*GO TO THAT ROUTINE.
3523
3524  021056  010046          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
3525  021060  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
3526  021064  005740                  TST     -(R0)           ;;BACKUP BY 2
3527  021066  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
3528  021070  006300                  ASL     R0              ;;POSITION FOR INDEXING
3529  021072  016000  021112          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
3530  021076  000200                  RTS     R0              ;;GO TO ROUTINE
3531
3532
3533                                      ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
3534
3535  021100  011646          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
3536  021102  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
```

```
3537  021110  000002                          RTI                         ;;RESTORE THE PSW
3538
3539  021112                                  SETTRAP TYPE,$TYPE,↑/TTY TYPEOUT ROUTINE/
3540  021112                                  $$SET   TYPE,$TYPE,\<TRAP+$TRP>,\$TRP,<TTY TYPEOUT ROUTINE>
3541                                .SBTTL  TRAP TABLE
3542
3543                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3544                                ;*BY THE "TRAP" INSTRUCTION.
3545
3546                                ;       ROUTINE
3547                                ;       -------
3548  021112  021100                $TRPAD: .WORD   $TRAP2
3549  021114  020636                        $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
3550  021116                                SETTRAP TYPOC,$TYPOC,↑/TYPE OCTAL NUMBER (WITH LEADING ZEROS)/
3551  021116                                $$SET   TYPOC,$TYPOC,\<TRAP+$TRP>,\$TRP,<TYPE OCTAL NUMBER (WITH LEADING ZEROS)>
3552  021116  017146                        $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3553  021120                                SETTRAP TYPOS,$TYPOS,↑/TYPE OCTAL NUMBER (NO LEADING ZEROS)/
3554  021120                                $$SET   TYPOS,$TYPOS,\<TRAP+$TRP>,\$TRP,<TYPE OCTAL NUMBER (NO LEADING ZEROS)>
3555  021120  017122                        $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3556  021122                                SETTRAP TYPON,$TYPON,↑/TYPE OCTAL NUMBER (AS PER LAST CALL)/
3557  021122                                $$SET   TYPON,$TYPON,\<TRAP+$TRP>,\$TRP,<TYPE OCTAL NUMBER (AS PER LAST CALL)>
3558  021122  017162                        $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
3559
3560  021124                                SETTRAP GTSWR,$GTSWR,↑/GET SOFT-SWR SETTING/
3561  021124                                $$SET   GTSWR,$GTSWR,\<TRAP+$TRP>,\$TRP,<GET SOFT-SWR SETTING>
3562  021124  017754                        $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
3563
3564  021126                                SETTRAP CKSWR,$CKSWR,↑/TEST FOR CHANGE IN SOFT-SWR/
3565  021126                                $$SET   CKSWR,$CKSWR,\<TRAP+$TRP>,\$TRP,<TEST FOR CHANGE IN SOFT-SWR>
3566  021126  017704                        $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
3567  021130                                SETTRAP RDCHR,$RDCHR,↑/TTY TYPEIN CHARACTER ROUTINE/
3568  021130                                $$SET   RDCHR,$RDCHR,\<TRAP+$TRP>,\$TRP,<TTY TYPEIN CHARACTER ROUTINE>
3569  021130  020166                        $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
3570  021132                                SETTRAP RDLIN,$RDLIN,↑/TTY TYPEIN STRING ROUTINE/
3571  021132                                $$SET   RDLIN,$RDLIN,\<TRAP+$TRP>,\$TRP,<TTY TYPEIN STRING ROUTINE>
3572  021132  020306                        $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
3573  021134                                SETTRAP RDDEC,$RDDEC,↑/READ A DECIMAL NUMBER FROM TTY/
3574  021134                                $$SET   RDDEC,$RDDEC,\<TRAP+$TRP>,\$TRP,<READ A DECIMAL NUMBER FROM TTY>
3575  021134  020460                        $RDDEC  ;;CALL=RDDEC    TRAP+11(104411) READ A DECIMAL NUMBER FROM TTY
3576  021136                                SETTRAP SAVREG,$SAVREG,↑/SAVE R0-R5 ROUTINE/
3577  021136                                $$SET   SAVREG,$SAVREG,\<TRAP+$TRP>,\$TRP,<SAVE R0-R5 ROUTINE>
3578  021136  016242                        $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
3579  021140                                SETTRAP RESREG,$RESREG,↑/RESTORE R0-R5 ROUTINE/
3580  021140                                $$SET   RESREG,$RESREG,\<TRAP+$TRP>,\$TRP,<RESTORE R0-R5 ROUTINE>
3581  021140  016300                        $RESREG ;;CALL=RESREG   TRAP+13(104413) RESTORE R0-R5 ROUTINE
3582
3583  021142                                SETTRAP INOCT,INOCTR,<INPUT OCTAL ROUTINE>
3584  021142                                $$SET   INOCT,INOCTR,\<TRAP+$TRP>,\$TRP,<INPUT OCTAL ROUTINE>
3585  021142  014040                        INOCTR  ;;CALL=INOCT    TRAP+14(104414) INPUT OCTAL ROUTINE
3586  021144                                SETTRAP INAR,RINA,<INPUT INPUT ADDR. ROUTINE>
3587  021144                                $$SET   INAR,RINA,\<TRAP+$TRP>,\$TRP,<INPUT INPUT ADDR. ROUTINE>
3588  021144  013406                        RINA    ;;CALL=INAR     TRAP+15(104415) INPUT INPUT ADDR. ROUTINE
3589  021146                                SETTRAP OUTAR,ROUTA,<INPUT OUTPUT ADDR. ROUTINE>
3590  021146                                $$SET   OUTAR,ROUTA,\<TRAP+$TRP>,\$TRP,<INPUT OUTPUT ADDR. ROUTINE>
3591  021146  013500                        ROUTA   ;;CALL=OUTAR    TRAP+16(104416) INPUT OUTPUT ADDR. ROUTINE
3592  021150                                SETTRAP PATAR,RPATA,<INPUT PATTERN ROUTINE>
```

```
3593  021150                          $$SET   PATAR,RPATA,\<TRAP+$TRP>,\$TRP,<INPUT PATTERN ROUTINE>
3594  021150  016116                  RPATA   ;;CALL=PATAR    TRAP+17(104417) INPUT PATTERN ROUTINE
3595  021152                          SETTRAP DELAR,RDELA,<INPUT DELAY TIME ROUTINE>
3596  021152                          $$SET   DELAR,RDELA,\<TRAP+$TRP>,\$TRP,<INPUT DELAY TIME ROUTINE>
3597  021152  015152                  RDELA   ;;CALL=DELAR    TRAP+20(104420) INPUT DELAY TIME ROUTINE
3598  021154                          SETTRAP DELAY2,RDELA2,<SECOUNDARY DELAY ROUTINE>
3599  021154                          $$SET   DELAY2,RDELA2,\<TRAP+$TRP>,\$TRP,<SECOUNDARY DELAY ROUTINE>
3600  021154  015674                  RDELA2  ;;CALL=DELAY2   TRAP+21(104421) SECOUNDARY DELAY ROUTINE
3601  021156                          SETTRAP DELAY,RDELAY,<ROUTINE TO DELAY XX MILLISEC>
3602  021156                          $$SET   DELAY,RDELAY,\<TRAP+$TRP>,\$TRP,<ROUTINE TO DELAY XX MILLISEC>
3603  021156  015636                  RDELAY  ;;CALL=DELAY    TRAP+22(104422) ROUTINE TO DELAY XX MILLISEC
3604  021160                          SETTRAP CPATR,RCPAT,<ROUTINE TO CHANGE PATTERNS>
3605  021160                          $$SET   CPATR,RCPAT,\<TRAP+$TRP>,\$TRP,<ROUTINE TO CHANGE PATTERNS>
3606  021160  015766                  RCPAT   ;;CALL=CPATR    TRAP+23(104423) ROUTINE TO CHANGE PATTERNS
3607  021162                          SETTRAP IDAC,RDACA,<ROUTINE TO INPUT DAC ADDR>
3608  021162                          $$SET   IDAC,RDACA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO INPUT DAC ADDR>
3609  021162  013664                  RDACA   ;;CALL=IDAC     TRAP+24(104424) ROUTINE TO INPUT DAC ADDR
3610
3611  021164                          SETTRAP CNTAR,RCNTA,<ROUTINE TO INPUT COUNTER MODULE ADDR>
3612  021164                          $$SET   CNTAR,RCNTA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO INPUT COUNTER MODULE ADDR>
3613  021164  013572                  RCNTA   ;;CALL=CNTAR    TRAP+25(104425) ROUTINE TO INPUT COUNTER MODULE ADDR
3614  021166                          SETTRAP ADAR,RADA,<ROUTINE TO INPUT ADDS ADDR>
3615  021166                          $$SET   ADAR,RADA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO INPUT ADDS ADDR>
3616  021166  013752                  RADA    ;;CALL=ADAR     TRAP+26(104426) ROUTINE TO INPUT ADDS ADDR
3617  021170                          SETTRAP INITR,RINIT,<ROUTINE TO ISSUE SYSTEM INITIALIZE>
3618  021170                          $$SET   INITR,RINIT,\<TRAP+$TRP>,\$TRP,<ROUTINE TO ISSUE SYSTEM INITIALIZE>
3619  021170  015734                  RINIT   ;;CALL=INITR    TRAP+27(104427) ROUTINE TO ISSUE SYSTEM INITIALIZE
3620  021172                          SETTRAP FOCTA,ROCTA,<ROUTINE TO CONVERT OCTAL NUMBERS TO ASCIZ>
3621  021172                          $$SET   FOCTA,ROCTA,\<TRAP+$TRP>,\$TRP,<ROUTINE TO CONVERT OCTAL NUMBERS TO ASCI
3622  021172  004712                  ROCTA   ;;CALL=FOCTA    TRAP+30(104430) ROUTINE TO CONVERT OCTAL NUMBERS TO ASCI
3623  021174                          SETTRAP QUBR,RQUBR,<ROUTINE TO SET UNI OR BI-POLAR>
3624  021174                          $$SET   QUBR,RQUBR,\<TRAP+$TRP>,\$TRP,<ROUTINE TO SET UNI OR BI-POLAR>
3625  021174  015344                  RQUBR   ;;CALL=QUBR     TRAP+31(104431) ROUTINE TO SET UNI OR BI-POLAR
3626
3627
3628                                  ;*
3629                                  ;*ACSII MESSAGE SECTION
3630                                  ;*
3631  021176  005015  041511  026523  MHEAD:  .ASCIZ  <15><12>:ICS-11 FIELD TEST PROGRAM - MAINDEC-11-DZICA-D  :<15><12>
      021263     015  033012  032056  MIPA:   .ASCIZ  <15><12>/6.4 INPUT MODULE ADDRESSES /
      021321     015  033012  032456  MOPA:   .ASCIZ  <15><12>/6.5 OUTPUT MODULE ADDRESSES /
      021360  005015  040440  042104  MTOH:   .ASCIZ  <15><12>/ ADDR    DATA   GENERIC CODE (IF INTERRUPT)/
      021436  005015  047516  042040  MNDA:   .ASCIZ  <15><12>/NO DAC ADDR. ENTERED/


      021465     015  042412  042116  MEND:   .ASCIZ  <15><12>/END PASS/




      021500  005015  027066  020066  MCNT:   .ASCIZ  <15><12>/6.6 COUNTER MODULE ADDRESSES /
      021540  042536                  MEXEN:  .ASCII  /↑E/
      021542  005015  054105  042520          .ASCIZ  <15><12>/EXPERT MODE ENABLED/
      021570  047136                  MNOEN:  .ASCII  /↑N/
      021572  005015  054105  042520          .ASCIZ  <15><12>/EXPERT MODE DISABLED/
      021621     136     114          MLEN:   .ASCII  /↑L/
```

```
021623     015  034012  032056            .ASCII  <15><12>/8.4 LINE PRINTER MODE ENABLED/
021662  005015  040515  042513            .ASCIZ  <15><12>/MAKE PRINTER READY/
021707     136     112            MAC:    .ASCII  /↑J/
021711     015  044412  047457            .ASCIZ  <15><12>:I/O MODULES ASSUMED CONNECTED:
021751     136  000104            MDEL1:  .ASCIZ  /↑D/
021754  005015  042523  047503    MDEL2:  .ASCIZ  <15><12>/SECONDARY DELAY TIME?/
022004  005015  047527  045522    MWK:    .ASCIZ  <15><12>/WORKING.../

022021     015  033012  033456    MDAC:   .ASCIZ  <15><12>/6.7 DAC ADDRESS /
022044  005015  027066  020070    MADU:   .ASCIZ  <15><12>/6.8 ADDS ADDRESS /
022070  005015  047524  020117    MABOV:  .ASCIZ  <15><12>/TOO MANY NUMBERS-RETYPE- /
022124  005015  027066  020063    MTN:    .ASCIZ  <15><12>/6.3 TEST NUMBER./
022147     015  047012  020117    MTNL:   .ASCIZ  <15><12>/NO SUCH TEST/
022166  005015  027066  030461    MIOO:   .ASCIZ  <15><12>/6.11 INPUT OR OUTPUT MODULE(I OR O)/
022234  005015  027066  020071    MDEL:   .ASCIZ  <15><12>/6.9 DELAY TIME (IN MILLISEC) /
022274  005015  052516  041115    MTTL:   .ASCIZ  <15><12>/NUMBER TOO BIG!/
022316  005015  027066  030061    MPPM:   .ASCIZ  <15><12>/6.10 PATTERN MODIFIER, PATTERN /

022360  005015  037440  000040    MQMARK: .ASCIZ  <15><12>/ ? /
022366  020077     000            MQ:     .ASCIZ  /? /

022371     015  054412  052517    MSTERR: .ASCII  <15><12>/YOU MUST INITIALLY START PROGRAM AT ADDR. 200/
022450  005015  043101  042524            .ASCIZ  <15><12>/AFTER RESART AT ADDR. 210 IS ALLOWED/
022517     015  044412  053116    MNRFN:  .ASCIZ  <15><12>/INVALID ADDRESS: /

022543     015  041412  052517    EM1:    .ASCIZ  <15><12>/COULD NOT SEND-RECEIVE DATA/
022601     015  051412  047105    EM2:    .ASCIZ  <15><12>/SEND-RECIEVE DATA ERROR/
022633     015  041412  047117    EM3:    .ASCIZ  <15><12>/CONVERT BIT FAILED TO SET/
022667     015  041412  047117    EM4:    .ASCIZ  <15><12>/CONVERT BIT FAILED TO CLEAR/
022725     015  041412  047101    EM5:    .ASCIZ  <15><12>/CAN'T READ DATA REGISTER/
022760  005015  030101  032460    EM6:    .ASCIZ  <15><12>/ADDS FAILED TO INTERRUPT/
023013     015  040412  042104    EM7:    .ASCIZ  <15><12>/ADDR. OR GENERIC CODE INCORRECT ON INTERRUPT/
023072  005015  044522  020106    EM10:   .ASCIZ  <15><12>/RIF DID NOT CLEAR INTR FLAG ON ADDS/

023140  051105  047522  020122    DH1:    .ASCII  ?ERROR    A/D?
023154  005015  020040  041520            .ASCIZ  <15><12>/  PC    ADDR/

023173     105  051122  051117    DH2:    .ASCII  /ERROR   MODULE   GOOD    BAD/
023227     015  020012  050040            .ASCIZ  <15><12>/  PC    ADDR     DATA    DATA/

023267     105  051122  051117    DH3:    .ASCII  /ERROR   MODULE    ICAR/
023315     015  020012  050040            .ASCIZ  <15><12>/  PC    ADDR    EXP'D   REC'D/

023355     015  047012  020117    MNAE:   .ASCIZ  <15><12>/NO COUNTER MODULE ADDRS.ENTERED/

023417     015  041412  052517    EM11:   .ASCIZ  <15><12>/COULD NOT READ-WRITE INTO COUNTER MODULE BUFFER/

023501     015  020012  051105    DH4:    .ASCII  <15><12>/ ERROR  COUNTER/
023522  005015  020040  041520            .ASCIZ  <15><12>/  PC     ADDR/

023542  005015  047503  047125    EM12:   .ASCIZ  <15><12>/COUNTER MODULE-UP COUNT INCORRECT/

023606  005015  047503  047125    EM13:   .ASCIZ  <15><12>/COUNTER MODULE FAILED TO INTERRUPT/

023653     015  041412  052517    EM14:   .ASCIZ  <15><12>/COUNTER MODULE-ADDR. OR GENERIC CODE INCORRECT O: INTERRUPT/
```

```
023751     015  041412  052517  EM15:   .ASCIZ  <15><12>/COUNTER MODULE DIDN'T STOP COUNTING ON OVERFLOW/

024033     015  051012  043111  EM16:   .ASCIZ  <15><12>/RIF DIDN'T CLEAR INTERRUPT FLAG ON COUNTER/

024110  005015  054523  027123  EM17:   .ASCIZ  <15><12>/SYS. INIT. FAILED TO CLEAR COUNTER BIT(S)/

024164  005015  046111  042514  EM20:   .ASCIZ  <15><12>/ILLEGAL INTERRUPT POSTED ON ICS BUS/

024232  005015  041511  020123  EM21:   .ASCIZ  <15><12>/ICS FAILED TO INTERRUPT CPU/

024270  005015  046111  042514  EM22:   .ASCIZ  <15><12>/ILLEGAL ADDR. OR GEN CODE BITS IN ICAR/

024341     015  051412  051531  EM23:   .ASCIZ  <15><12>/SYS. INIT. FAILED TO CLEAR COUNTER INTR. FLAG/

024421     015  041412  052517  EM24:   .ASCIZ  <15><12>/COUNTER STARTED COUNTING AFTER SYS. INITIALIZE/

024502  005015  030101  032460  EM25:   .ASCIZ  <15><12>/ADDS READ DUAL ADDR. ERROR/

024537     015  040412  030060  EM26:   .ASCIZ  <15><12>/ADDS DUAL ADDR. ERROR/
024567     015  051412  047105  EM27:   .ASCIZ  <15><12>/SEND-RECIEVE DATA ERROR/

024621     015  042412  051122  DHS:    .ASCII  <15><12>?ERROR    A/D    DUAL?
024647     015  020012  050040          .ASCIZ  <15><12>/  PC      ADDR    ADDR/

024700  005015  027066  031061  MQUR8:  .ASCIZ  <15><12>/6.12 UNIPOLAR OR BIPOLAR ( U OR B) /

024746  005015  047516  040440  MNAD:   .ASCIZ  <15><12>?NO A/D ADDR. ENTERED?
024775     015  051412  030127  MSWD:   .ASCIZ  <15><12>/SWDD=1 MEANS RETURN TO MONITOR/

025036  030455  027060  030060  MMDVFL: .ASCIZ  /-10.0000 (OR OVERFLOW)/

025065     053  030061  030056  MPOVFL: .ASCIZ  /+10.0000 (OR OVERFLOW)/
025114  005015  042522  042520  MREP:   .ASCIZ  <15><12>/REPEAT/

025125     040  042522  042101  MCALOT: .ASCIZ  / READING (OCTAL)/

025146  053040  046117  051524  MCALT1: .ASCIZ  / VOLTS (DECIMAL) = /

025172  000055                  MMINUS: .ASCIZ  /-/
025174  000053                  MPLUS:  .ASCIZ  /+/

025176  005015  042522  052123  PWRMSG: .ASCIZ  <15><12>/RESTARTING AFTER A POWER FAILURE/<15><12><12>

025244  005001  042524  052123  MHT0:   .ASCIZ  <1><12>/TEST 0 - INPUT AND OUTPUT MODULE EXERCISER/<15><12>

025323     015  052012  051505  MHT1:   .ASCIZ  <15><12>/TEST 1 - INPUT OR OUTPUT MODULE EXERCISER/<15><12>

025401     015  052012  051505  MHT2:   .ASCIZ  <15><12>/TEST 2 - DAC CALIBRATION/<15><12>

025436  005015  042524  052123  MHT3:   .ASCIZ  <15><12>/TEST 3 - DAC INTERACTION/<15><12>

025473     015  052012  051505  MHT4:   .ASCIZ  <15><12>/TEST 4 - COUNTER MODULE TEST/<15><12>

025534  005015  042524  052123  MHT5:   .ASCIZ  <15><12>?TEST 5 - A/D LOGIC TEST?<15><12>
```

```
        025570  005015  042524  052123  MHT6:   .ASCIZ  <15><12>?TEST 6 - A/D CALIBRATION?<15><12>


        025625     015  052012  051505  MHT8:   .ASCIZ  <15><12>?TEST 7 - A/D REPEATABILITY?<15><12>
        025664  005015  027070  020064  MLPAV:  .ASCIZ  <15><12>/8.4 LINE PRINTER OPTION AVAILIBLE/
        025730  005015  047111  040526  MIVP:   .ASCIZ  <15><12>/INVALID PATTERN MODIFIER/

        025763     015  033012  030456  MGAIN:  .ASCIZ  <15><12>/6.13 GAIN /

        026000  005015  027066  032061  MCHAN:  .ASCIZ  <15><12>/6.14 CHANS (SC,EC)/
        026025     015  033012  030456  MTOL:   .ASCIZ  <15><12>/6.16 TOLERANCE /

        026047     015  042412  051122  MCHER1: .ASCIZ  <15><12>/ERROR! STARTING CHAN. GREATER THAN END CHAN./
        026126  005015  040514  052123  MINNN:  .ASCIZ  <15><12>/LAST CHARACTER NOT A OCTAL DIGIT--RETYPE IT.  /
        026207     015  037412  047111  MINKN:  .ASCIZ  <15><12>/?INPUT NOT UNDERSTOOD--RETYPE IT./

        026253     015  042412  051122  MCHANH: .ASCIZ  <15><12>/ERROR! NO SUCH CHAN./

        026302  005015  027066  032461  MAVEQ:  .ASCIZ  <15><12>/6.15 EXPECTED AVERAGE /
        026333     015  047012  020117  MNSG:   .ASCIZ  <15><12>/NO SUCH GAIN!/

        026353     015  047012  046525  MNTL:   .ASCIZ  <15><12>/NUMBER TO LARGE-MAX=7777/
        026406  020040     000          M2SP:   .ASCIZ  /  /
        026411     015  051012  050105  MREPFT: .ASCIZ  <15><12>/REPEATIBILITY FORCED TYPOUT/

        026447     015  051012  050105  MREPER: .ASCIZ  <15><12>/REPEATABILITY ERROR!/

        026476  005015  041440  040510  MREPT1: .ASCIZ  <15><12>/ CHAN    GAIN    LOW      AVER   HIGH/<15><12>

        026550  005015  046040  020117  MREPT2: .ASCII  <15><12>/ LO   -5   -4   -3   -2   -1   AV/
        026613     040  025440  020061          .ASCIZ  /  +1   +2   +3   +4   +5   HI/<15><12>

        026653     015  033012  030456  MFILE:  .ASCIZ  <15><12>/6.1 FILE BOX TO BE TESTED/
        026707     015  033012  031056  MVECT:  .ASCIZ  <15><12>/6.2 ICS VECTOR ADDRESS/
        026740  005015  046111  042514  ILLEG:  .ASCIZ  <15><12>/ILLEGAL NUMBER/<15><12>
        026763     015  040412  042104  NRANG1: .ASCIZ  <15><12>/ADDRESS /
        026776  047040  052117  053440  NRANG2: .ASCIZ  / NOT WITHIN FILE BOX RANGE/<15><12>
        027033     015  043012  046111  NOINT:  .ASCIZ  <15><12>/FILE BOX DID NOT INTERRUPT-FATAL/
        027076  005015  044506  042514  FILINT: .ASCIZ  <15><12>/FILE BOX INTERRUPTED AT /
        027131     040  026455  041440  CKJMP:  .ASCIZ  / -- CHECK JUMPERS /
        027154  005015  047516  026516  NONXST: .ASCIZ  <15><12>/NON-EXISTENT FILE BOX/
                                                .EVEN

3632    027204     015     012          OUTBF1: .BYTE   15,12
3633    027206  000000                  OUTBF:  0
3634            027240                          .=.+30
3635    027240                          BUFFER:
3636                                    ;*END ADDRESS IS FUNCTION OF HOW MANY CHANS. ARE BEING EXERCISED
3637                                    ;*AT ONE TIME IN TEST 7.
3638            027240                  $ENDAD=.
3639            000001                          .END
```

```
ADAOR   013340          1387    1569    1765    2215#   2336    2341    2345*   2346
ADAR  = 104426          1383    1556    1763    3616#
ADCSR   007144           385     386     391    1380#   1387*   1393*   1394    1403*   1404    1420*   1421    1429    1441*
                        1442    1455*   1469    1484    1509    1511*   1519    1569*   1578    1600    1738*   1739    1741*
                        1765*   1935*   1936    1938*   1939
ADLOG   007164          1387#   1768
ADLOGE  007762          1384    1532#
ADLOGL  007154          1384#   1535
ADLOG2  007246          1402#   1414
ADLOG3  007314          1418#
ADLOG4  007402          1425    1437#
ADLOG5  007404          1441#
ADLOG6  007426          1452#
ADLOG7  007506          1453    1467#
ADLOG8  007560          1474    1483#
ADLOG9  007620          1463    1500#
ADLO3A  007360          1423    1427#
ADL6L   007476          1461#   1479    1489    1494
ADPATE  010020          1413    1546#
ADPATP  007776          1391    1537#
ADWD    013102          1931*   1932*   1933*   1934*   1935    2125#
AVERR   012152          1889    1962#
AVERRL  012164          1964#   1999
AVERRN  012370          1996#   2104
AVEXP   013124          1839    1844    1846    1853    1857*   1986    2134#
AVTKN   013126          1967*   1981*   2011*   2015*   2016*   2020*   2049    2063    2135#
AVTYP   013033          2075    2087    2101    2105#
A1      010610          1627    1629    1631    1655*   1657    1659    1661    1667    1671    1674    1685    1690#
BIPOL   013342          1609    1633    1637    1855    1940    1978    2017    2216#   2619*   2621*
BIT0  = 000001           130#
BIT00 = 000001           120#     130
BIT01 = 000002           119#     129
BIT02 = 000004           118#     128
BIT03 = 000010           117#     127
BIT04 = 000020           116#     126
BIT05 = 000040           115#     125
BIT06 = 000100           114#     124
BIT07 = 000200           113#     123
BIT08 = 000400           112#     122
BIT09 = 001000           111#     121
BIT1  = 000002           129#
BIT10 = 002000           110#
BIT11 = 004000           109#    3212
BIT12 = 010000           108#
BIT13 = 020000           107#    2988
BIT14 = 040000           106#    3194
BIT15 = 100000           105#
BIT2  = 000004           128#
BIT3  = 000010           127#
BIT4  = 000020           126#
BIT5  = 000040           125#
BIT6  = 000100           124#
BIT7  = 000200           123#
BIT8  = 000400           122#
BIT9  = 001000           121#
BPTVEC= 000014           137#
```

```
BUFFER  027240          1909    1924    1966    2067    2080    2094    3635#
CALIT   010240          1595    1609#
CALITE  010414          1651#   1687
CALITP  010400          1640    1646#
CALIT1  010404          1638    1645    1648#   1652
CALIT2  010520          1649    1673#
CALIT3  010530          1675#   1686
CALIT4  010556          1678    1681#
CALIT5  010574          1682    1685#
CALTP   010170          1581    1593#
CHAN    013076          1897*   1898*   1899    1903*   1910*   1912    1919*   1920    1962*   1997*   1998    2024    2123#
CHANF   013106          1798*   1875*   1876*   1880    1882*   1885    1887*   1897    1920    1998    2127#
CHANFR  013112          1800*   1821    1824*   1825    1830    1883    1885    1887    2129#
CHANNO  013114          1809*   1882    2130#
CHANS   013104          1797*   1880*   1881*   1883    1898    1910    1962    2126#
CHANSR  013110          1799*   1814    1819    1824    1825    1875    2128#
CHAN1   013074          1912*   1913*   1914*   1915*   1916*   1933    1963*   1965    1996*   2066    2079    2093    2122#
CHAN7   013066          2119#
CHAR    015046          1007*   1020*   1024    1026    2251    2275    2300    2323    2343    2374*   2375*   2376    2381
                        2382    2384    2387    2390    2393    2396    2399    2402    2404    2406    2410    2419*   2420
                        2429*   2463    2465    2503#   2601*   2614*   2617
CHARC   015054          2371*   2417*   2474    2506#
CKADR   015510          2255    2279    2303    2327    2347    2630#
CKJMP   027131           580    3631#
CKSWR = 104436          2980    2996    3193    3566#
CNTADR  013276           801    1111    2196#   2292    2297    2299
CNTAR = 104425          1107    3613#
CNTPAE  007142          1192    1368#
CNTPAT  007076          1173    1350#
CONNT   001654           431#    489*    781     804     827     893    2532*
CONNTR  015134          2392    2532#
CONVED  010760          1739#
CONVER  010740          1577    1735#
CONVT   012044          1917    1931#
CORSIZ  001650           429#    623*    625*    1801
CPATR = 104423           922    3606#
CR    = 000015            45#   3500    3510
CRLF  = 000200            46#   3471    3510
CTLLOC  001700           441#    509*    540*    595*    2386
DACADR  013336          1063    1072*   1087    1094*   1120    2213#   2316    2321    2325*   2326
DDISP = 177570            52#    175     476
DELAR = 104420           768    1048    3597#
DELAY = 104422           700     702     790     811     1053    1203    1289    1314    1341    1428    1458    1487    1597
                        1607    1767    1911    3603#
DELAY2= 104421           792    3600#
DFO     001476           212     219     226     233     239     246     253     260     267     274     281     288     295
                         302     309     316     323     330     337     344     351     358     365     373     378     381
                         384#
DH1     023140           209     224     231     237     244     258    3631#
DH2     023173           216     272     279     300     314     349     371    3631#
DH3     023267           251     293     335    3631#
DH4     023501           265     286     307     321     328     342    3631#
DH5     024621           356     363    3631#
DISPLA  001142           175#    476*    484*    1037*   2983*   3223*
DISPRE  000174            26#    484
DSWR  = 177570            51#    174     475
```

```
DT1     001500          211     225     232     238     245     259     385#
DT2     001506          218     252     386#
DT3     001520          266     287     308     322     329     343     388#
DT4     001526          273     280     294     301     315     336     350     372     380     389#
DT5     001540          357     364     391#
EMTVEC= 000030          140#    462*    463*
EM1     022543          208     3631#
EM10    023072          257     3631#
EM11    023417          264     3631#
EM12    023542          278     3631#
EM13    023606          285     3631#
EM14    023653          292     3631#
EM15    023751          299     3631#
EM16    024033          306     3631#
EM17    024110          313     3631#
EM2     022601          215     271     3631#
EM20    024164          320     3631#
EM21    024232          327     3631#
EM22    024270          334     3631#
EM23    024341          341     3631#
EM24    024421          348     3631#
EM25    024502          355     3631#
EM26    024537          362     3631#
EM27    024567          370     3631#
EM3     022633          223     3631#
EM4     022667          230     3631#
EM5     022725          236     3631#
EM6     022760          243     3631#
EM7     023013          250     3631#
ERAV1   012414          1989    1993    2004#
ERAV2   012422          1995    2007#
ERRVEC= 000004          133#    473     474*    485*    3199    3200*   3202*   3205*
EXIT  = 000002          444#    754     860     865     870     895     914     943     2258    2282    2307    2328    2348
                        2445    2501    2562    2620    2622    2653    2665    2680    2695    2712    2730    2734    2736
                        2755    2787
EXPERT  001652          430#    665*    1001    1769    1793    1835    1858    2240    2264    2288    2312    2332    2510*
                        2515*   2543    2595    2763
EXSET   015062          2398    2510#
FILHLT  002634          596#
FILINT  027076          576     3631#
FILVEC  002124          495#
FOCTA = 104430          851     910     3622#
FREQ    001566          404#    668*    682*    688     696*    1108*   1427*   1456*   1596*   1766*   1896*   1902*   1905
                        1907*   2548*   2555    2556*   2578*   2673
FREQ1   001570          405#    681*    69      2547*   2554*   2557*   2558    2675
FREQ2   001572          406#    680*    257     2687
FREQ3   001574          407#    679*    2579*   2580    2689
FR1     001576          408#    627*    632*    633*    638*    644     647     653     1582    2556    2578
FR110   001550          397#    659*    660*    661*    662*    696     1596    1766
FR1120  001600          409#    627     632     633     1582
FR16    001560          401#    649*    650*    651*    652     653*    1427    1456
FR20    001556          400#    652*    654*    655     663
FR3     001564          403#    644*    645*    646     647*    648     668     682     2548
FR40    001554          399#    663*    664*    1108
FR5     001562          402#    646*    648*    649     651     654     657     658     661     662     1902
FR50    001552          398#    655*    656*    657*    658*    659     1896
```

```
GAIN    013100          1773    1778    1789    1791*   1931    2028    2124#
GLIST   013154          1781    2151#
GNS   = ******  U         25    3549    3552    3555    3558    3562    3566    3569    3572    3575    3578    3581    3585
                        3588    3591    3594    3597    3600    3603    3606    3609    3613    3616    3619    3622    3625
GOSUB = 004737           446#   1811    1888    1889    1917    2075    2087    2101    2106
GTSWR = 104405           706     753    3562#
HEADER  001634           423#    770*    834     840*
HOLDIT  010632          1636    1692#
HOLDTM  010606          1653    1656*   1673*   1677    1679*   1684*   1689#
HT    = 000011            43#   3469    3510
ICAR    001616           416#    529*    530*    877     889    1262    1300    1468
ICSHGH  001676           440#    531*    532*   2642
ICSLMT  001702           442#    521*    527*    531    2640
ICSMOD  001612           414#    880    1502    1504    2254    2278    2302    2325    2345
ICSR    001614           415#    522*    526*    529     534     567*    582*    588*    685*    794*    884*    888*   1245
                        1253*   1264*   1287*   1292*   1304*   1322*   1325    1330*   1343    1454*   1461*   1483*
ICSVT   001620           417#    554*    564*    772*   1252*   1282*   1453*   1485*
ICSVT2  001622           418#    556*    565*    774*   1283*   1452*
IDAC  = 104424          1062    1086    3609#
ILLEG   026740           519     552    3631#
INADR   013176           674     793     900     906    1036    2162#   2244    2249    2253
INAR  = 104415           765    1031    3588#
INCFLG  001630           421#   2362*   2422    2448    2450*   2460*   2470
INITR = 104427          1040    1202    1221    1294    1311    1329    1342    1436    1526    3619#
INOCT = 104414           514     545     713    1777    1818    1843    1864    2248    2272    2296    2320    2340    2774
                        3585#
INOCTR  014040          2356#   2482    2486    2493    3585
IOTVEC= 000020           138#    460*    461*
KEYSRV  003672           603     745#
LF    = 000012            44#   3504    3510
LINEPR  001660           433#    687*   1867    2526*
LPAV    001656           432#    607*    614*   2522
LPCSR   001662           434#    608    1869
LPDBR   001664           435#   1870
LPSET   015112          2389    2522#
MABOV   022070          2481    3631#
MAC     021707          2535    3631#
MADU    022044          2335    3631#
MAVEQ   026302          1838    3631#
MCALOT  025125          1606    3631#
MCALT1  025146          1599    3631#
MCHAN   026000          1796    3631#
MCHANH  026253          1833    3631#
MCHER1  026047          1828    3631#
MCNT    021500          2291    3631#
MDAC    022021          2315    3631#
MDEL    022234          2546    3631#
MDEL1   021751          2571    3631#
MDEL2   021754          2573    3631#
MEND    021465          1124    1534    3631#
MEXEN   021540          2512    3631#
MFILE   025653           510    3631#
MGAIN   025763          1772    3631#
MHEAD   021176           503    3631#
MHT0    025244           764    3631#
MHT1    025323           999    3631#
```

```
MHT2    025401          1061    3631#
MHT3    025436          1085    3631#
MHT4    025473          1106    3631#
MHT5    025534          1382    3631#
MHT6    025570          1555    3631#
MHT8    025625          1762    3631#
MINKN   026207          2492    3631#
MINNN   026126          2489    3631#
MIOO    022166          1004    3631#
MIPA    021263          2243    3631#
MIVP    025730          2780    3631#
MLEN    021621          2525    3631#
MLPAV   025664           612    3631#
MMINUS  025172          1642    3631#
MMOVFL  025036          1620    3631#
MNAD    024746          1572    3631#
MNAE    023355          1115    3631#
MNDA    021436          1066    1090    3631#
MNOEN   021570          2517    3631#
MNRFN   022517          2646    3631#
MNSG    026333          1785    3631#
MNTL    026353          1849    3631#
MOPA    021321          2267    3631#
MPLUS   025174          1647    3631#
MPOVFL  025065          1614    1625    3631#
MPPM    022316          2767    3631#
MQ      022366          1006    2357    2550    2600    3631#
MQMARK  022360          2485    3631#
MQUR8   024700          2598    3631#
MREP    025114          1874    3631#
MREPER  026447          2005    3631#
MREPFT  026411          2008    3631#
MREPT1  026476          2022    3631#
MREPT2  026550          2062    3631#
MSTERR  022371           694    3631#
MSWD    024175          1032    1566    3631#
MTN     022124           708    3631#
MTNL    022147           721    3631#
MTOL    026025          1861    3631#
MTTL    022274          2565    2586    3631#
MTOH    021360           784    3631#
MVECT   026707           541    3631#
MWK     022004          1033    1050    1069    1093    1109    1385    1567    1871    3631#
M2SP    026406          2027    2034    2041    2055    2112    3631#
NA07    013070          2120#
NA17    013072          2121#
NIN     015050          2370*   2414*   2415*   2416*   2418*   2420*   2430    2447    2458    2476    2504#
NINC    015056          2369*   2424    2428*   2432    2436*   2454    2457*   2461*   2472    2477*   2508#
NINC1   015060          2360*   2424    2431    2432*   2433*   2442    2454    2472    2509#
NIN1    015052          2447*   2452*   2453*   2456    2458    2505#
NOINT   027033           569    3631#
NONXST  027154           537    3631#
NOSET   015076          2401    2515#
NRANG1  026763          2658    3631#
NRANG2  026776          2662    3631#
OUTADR  013236           797     823     903    1052*   2179#   2268    2273    2277
```

```
OUTAR = 104416          766    1047    3591#
OUTBF   027206          929    3633#
OUTBF1  027204          942    3632#
OUTS    013344          677     777     796     806     907    2218#   2364    2368    2434
OUTSE   013404          779     808    2234#   2366
PACK    005010          931     933     936     948#
PACK1   005004         930*    932*    934*    945#     949     953*    954     962     967*    968     978     985*    986*
                        987    1112*   1119    1128*
PACK2   005006         946#    948*    951*    952     954*    955*    956*    957*    958*    959*    960*    961     962*
                       963*    964*    965*    966     968*    969*    970*    971*    972*    973*    974*    975*    976*
                        977    978*    979*    980*    981*    982*    983*    984
PATAR = 104417          767    3594#
PATJOY  001610         413#    810*    832
PATRN   001604         411#    669*    800     807     810    2724*   2728*   2754*   2771*
PATRNC  001606         412#   2728    2729*   2733*   2735*
PATRNM  001602         410#   2721    2768    2770*   2775    2778    2783*   2784*   2785
PIRQ  = 177772          50#
PIRQVE= 000240         144#
POPSP2= 022626         447#     536     574     581     587     615     640    1263    1467    1491
PR0   = 000000          67#
PR1   = 000040          68#
PR2   = 000100          69#
PR3   = 000140          70#
PR4   = 000200          71#
PR5   = 000240          72#
PR6   = 000300          73#
PR7   = 000340          74#
PS    = 177776          47#      48     566*    703*    728*    795*    845*    885*    887*   1049#   1254#   1267#   1286#
                       1290*   1305*   1389*   1457*   1486*   2690#
PSW   = 177776          48#
PWRMSG  025176         3170    3631#
PWRVEC= 000024          139#    466*    467*   3134*   3135*   3146*   3153*   3167*   3168*
QUBR  = 104431         1557    1764    3625#
RADA    013752         2332#   2339    3616
RADA1   014006         2336*   2337*   2341#
RASK    015036         2499#   2513    2518    2527    2536    2584
RCNTA   013572         2288#   2295    3613
RCNTA1  013626         2292*   2293*   2297#   2304
RCPAT   015766         2721#   2786*   3606
RCPATI  016000         2724#   2726    2731
RCPATL  016062         2738#   2777
RCPATR  016102         2722    2752#
RDACA   013664         2312#   2319    3609
RDAC1   013720         2316*   2317*   2321#
RDCHR = 104407         3347    3569#
RDDEC = 104411         2551    2574    3575#
RDELA   015152         2543#   2566    3597
RDELAY  015636         2673#   3603
RDELA2  015674         2697#   3600
RDELA3  015260         2355    2571#
RDELA4  015264         2572#   2587
RDLIN = 104410         3395    3572#
REPET7  010726         1579    1601    1730#
REPMAN  013140         2140#
RESREG= 104413         2937    3581#
RESVEC= 000010          134#
```

```
RETURN= 000207           445#
RHIGH   013132          1970*  1971   1973*  1988   2056   2137#
RINA    013406          2240#  2247   3588
RINA1   013442          2244*  2245*  2249#  2256
RINIT   015734          2704#  3619
RLOW    013130          1969*  1974   1976*  1992   2035   2136#
ROCTA   004712           929#  3622
ROUTA   013500          2264#  2271   3591
ROUTA1  013534          2268*  2269*  2273#  2280
RPATA   016116          2763#  2773   2781   3594
RQUBR   015344          2595#  3625
RTEMP   013134           688*   723*   724*   725*   726*   729   1013*  1014*  1015   1017   1020   1146*  1149
                        1165*  1173*  1177   1179   1184   1191*  1192   1504*  1505*  1507   1801*  1802*  1806*
                        1968*  1982*  2010*  2014*  2063*  2064*  2069   2081   2088*  2095   2138#  2363*  2413*
                        2426   2431*  2439*  2552*  2560*  2575*  2582*  2606*  2607   2608*  2609   2612   2614
                        2673*  2676*  2687*  2691*  2777*  2785*  2786
RTEMP1  013136          1178*  1181*  1803*  1804*  1808*  1809   1810   1977*  1980*  1981   1986*  1987*  1988
                        1990*  1991*  1992   2139#
RTEMP2  015672          2675*  2678*  2681#  2689*  2693*
SAMCNT  013116          1908*  1922*  1964*  1984*  2065*  2073*  2077*  2085*  2092*  2099*  2131#
SAMOFF  013120          1899*  1901*  1983   2072   2084   2098   2132#
SAMPR   011650          1888   1896#
SAVREG= 104412          2911   3578#
SAVO    004672           899*   909    918#
SCAN    010604          1675*  1676   1688#
STACK = 001100            38#   458
START   001704            29    451#   695   3172
START1  003374            32    595    684#   722    755    992   1019   1043   1077   1098   1548   1573   1576
                        1744   1879   2611   2709
STKLMT= 177774           49#
ST200   001646          428#   670    675*   689
SWR     001140          174#   456    475*   477    483*   704    857   1038   1041   1052   1070   1574   1580
                        1584   1735   1877   2707   2988   2993   3144  3159#  3194   3212   3241   3279#
SWREG   000176           27#   483   3241   3255
SW0   = 000001          102#
SW00  = 000001           92#   102
SW01  = 000002           91#   101
SW02  = 000004           90#   100
SW03  = 000010           89#    99
SW04  = 000020           88#    98
SW05  = 000040           87#    97
SW06  = 000100           86#    96
SW07  = 000200           85#    95
SW08  = 000400           84#    94
SW09  = 001000           83#    93
SW1   = 000002          101#
SW10  = 002000           82#
SW11  = 004000           81#
SW12  = 010000           80#
SW13  = 020000           79#
SW14  = 040000           78#
SW15  = 100000           77#
SW2   = 000004          100#
SW3   = 000010           99#
SW4   = 000020           98#
SW5   = 000040           97#
```

```
SW6    = 000100            96#
SW7    = 000200            95#
SW8    = 000400            94#
SW9    = 001000            93#
TBITVE= 000014            135#
TKVEC  = 000060            142#   603*   604*
THPFIL  001672            438#   515    517    523*   524*
THPVEC  001674            439#   546    548    550    554    555*   556
TOLER   013122           1851   1865   1987   1990   1991   1994   2133#
TPBSY   001632            422#   776*   813    829    843    849*   859*   868*   882    911*   919
TPBSYP  001644            427#
TPCSR   001666            436#   672*   698
TPOBR   001670            437#   673*   699
TPVCT   001624            419#   773*
TPVCT2  001626            420#   775*
TPVEC  = 000064            143#
TRAPVE= 000034            141#   464*   465*
TRTVEC= 000014            136#
TSTLST  003652            725    732#
TSTNO   003650            709    714    716    718    723    731#
TSTO    003744            732    763#
TST1    005322            733    998#
TST10   005550           1025   1047#
TST2    005604            734   1060#  1067
TST3    005676            735   1084#  1091
TST4    005746            736   1105#  1116
TST4A   006056           1134#
TST4AL  006104           1137   1140#
TST4B   006106           1146#
TST4C   006222           1173#
TST4D   006346           1199#
TST4E   006412           1216#
TST4F   006460           1245#
TST4G   006572           1282#
TST4GE  007016           1295   1312   1320   1326   1330#
TST4GI  006660           1282   1299#
TST4GL  006614           1284   1286#
TST4H   007032           1338#
TST4L   006012           1114   1118#  1348
TST4L1  005766           1111#  1125
TST5    007146            737   1381#
TST6    010026            738   1554#
TST7    011000            739   1761#
TYPE   = 104401            502    510    519    537    541    552    569    576    580    609    611    693    707
                           720    751    763    783    998   1003   1005   1022   1032   1033   1050   1060   1065
                          1069   1084   1089   1093   1105   1109   1115   1123   1381   1385   1533   1554   1566
                          1567   1571   1593   1598   1605   1613   1619   1624   1641   1646   1670   1761   1771
                          1784   1795   1827   1832   1837   1848   1860   1871   1873   2004   2007   2021   2026
                          2033   2040   2054   2061   2102   2109   2111   2242   2266   2290   2314   2334   2356
                          2444   2480   2484   2488   2491   2511   2516   2524   2534   2545   2549   2564   2571
                          2572   2585   2597   2599   2616   2646   2658   2662   2765   2780   2991   3009   3027
                          3029   3032   3034   3038   3045   3111   3169   3252   3253   3257   3270   3281   3300
                          3351   3354   3358   3433   3435   3474   3549#
TYPOC  = 104402            579   1604   2025   2032   2039   2053   2060   2649   2661   3018   3042   3256   3552#
TYPON  = 104404           3558#
TYPOS  = 104403           3555#
```

```
TOAOR   001636            424#   785*   846*   890*   891    901    930
TODAT   001640            425#   786*   847*   892*   908    932
TOGEN   001642            426#   787*   848*   889*   897*   898*   934
TOIN    004144            799    804#
TOINT   004444            772    877#
TOIN1 = 004224            820#   853
TOIN2   004242            819    827#
TOLOP   004674            817    825    837    841    919#   920
TOOUT   004122            797#
TOOUTR  004070            791#   923
TOPINT  004370            773    855#   856    912
TBCAA   010046           1568#  1592   1608
$AUTO8  001134            171#  3249   3373
$BOADR  001122            166#   388    389    391    831*  1126*  1135*  1136   1150*  1151   1158*  1159   1177*
                         1183   1201*  1205   1219*  1222   1266   1288*  1301   1316   1323   1331   1339*  1501*
                         1502*  1503*  1506*  1507   1509   1512   1518*
$BODAT  001126            168#   386    389    833*  1136*  1151*  1152   1159*  1183*  1184   1186   1205*  1222*
                         1262*  1268*  1269   1300*  1307   1316*  1394*  1404*  1405   1421*  1422   1468*  1473
$CHARC  021052           3476#  3486*  3493   3502*  3507#
$CKSUM  017704           3241#  3564   3566
$CMTAG  001100            154#   453    454    462    468    469
$CM3  = 000000            184#
$CNTLG  020431            752   3252   3368#
$CNTLU  020424           3270   3367#
$CRLF   001163            186#   379    610   1023   1594   2103   2444   2616   2991   2999   3009   3029   3034
                         3038   3281   3367   3437   3475   3510
$DB2D   016474           2891   2911#
$DECVL  016654           2913   2959#
$ENDAD= 027240           1802   3638#
$ERFLG  001103            157#  2981*  2999   3185   3208   3210*  3227
$ERMAX  001115            163#  3227
$ERROR  016670            462   2979#
$ERRPC  001116            164#   385    386    388    389    391   2985*  2986*  2987   2999   3016
$ERRTB  001166            204#  3024
$ERRTY  016766           2990   3008#
$ERTTL  001112            161#  2984*  2999
$FILLC  001156            182#  3479   3510
$FILLS  001155            181#  3510
$GDADR  001120            165#
$GODAT  001124            167#   386    389    832*  1149*  1150   1152   1158   1161*  1179*  1180*  1186   1204*
                         1218*  1265*  1301*  1302*  1303*  1306*  1307   1315*  1392*  1393   1402*  1403   1405
                         1418*  1420   1422   1469*  1470*  1471*  1472*  1473
$GTSWR  017754           3253#  3560   3562
$HO   = 000001             16     17
$HINUM  016434            487*  2855   2863   2868*  2874#
$ICNT   001104            158#  2705   3216*  3217   3219*  3226
$ILLUP  017514           3134   3153   3174#
$INTAG  001135            172#  3282   3373
$ITEMB  001114            162#  2987*  2999   3012
$LF     001164            187#  2999   3358   3367   3437   3510
$LONUM  016436            488*  2754   2854   2861   2867*  2875#
$LPADR  001106            159#   469*   791*  1134*  1148*  1175*  1200*  1217*  1284*  1340*  1390*  1419*  1500*
                         3222*  3224   3226
$LPERR  001110            160#
$MAIL = ****** U          487   2993   3222   3463
$MNEW   020447           3257   3371#
```

```
$MSWR   020436            3253    3369#
$MXCNT  017702            3220    3226#
$NULL   001154             180#   3481    3510
$OCNT   017344            3083*   3112*   3125#
$OMODE  017346            3078#   3082*   3087    3090*   3101*   3127#
$OVER   017666            3195    3206    3218    3223#
$PASS   001100             155#    686*    769*   1122*   1532*   3214    3227
$PWRDO  017510            3172#
$PWRUN  017350             466    3134#   3167
$PWRWK  017504            3170#
$PWRUP  017422            3146    3153#
$QUES   001162             185#   2999    3300    3351    3367    3435    3437    3510
$SEND   016336            2753    2850#
$SKDCHR 020166            3314#   3567    3569
$RODEC  020460            3389#   3573    3575
$RDLIN  020306            3343#   3570    3572
$RDOCT= ****** U          3573
$RDSZ = 000010            3335#
$RESRE  016300            2824#   3581
$R2A  = ****** U          3582
$SAVRE  016242            2807#   3576    3578
$SAVR6  017520            3145*   3154    3155*   3156*   3176#
$SB2D   016440            1811    2106    2889#
$SCOPE  017522             460    3192#
$SETUP= 000117             450#    459     460     462     464     466     468     469    2980    2996    2998    3193    3235
                          3373
$STUP = 177777             450#
$SVLAD  017656            3203    3221#
$SWR  = 164000               3#     16     184     185     468     469     470    2973    2974    2975    2976    2984    2988
                          2993    2997    2999    3173    3186    3187    3188    3189    3194    3206    3208    3209    3210
                          3211    3212    3223    3226
$SWRMK= 000000            3189
$TIMES  001160             184#    468*   1147*   1174*   1199*   1216*   1338*   3211*   3217    3220*   3226
$TKB    001146             177#    745    1012    1013    2374    2606    3233    3245    3263    3318    3324
$TKS    001144             176#    605*    727*   1000*   1008    1035*   1051*   1118*   1388*   1568*   2372    2602    2711*
                          3233    3243    3260    3284*   3316    3322
$TN   = 000001              16#
$TNPWR  016604            2918    2919    2939#
$TPB    001152             179#    673     699*    862*   1012*   1870*   2381*   2607*   3499*   3510
$TPFLG  001157             183#   3457    3510
$TPS    001150             178#    672     698*    855     864*    867*   1010    1869*   2379    2604    3497    3510
$TRAP   021056             464    3524#
$TRAP2  021100            3535#   3548
$TRP  = 000032            3539#   3540    3541    3550#   3551    3552    3553#   3554    3555    3556#   3557    3558    3559#
                          3561    3562    3563#   3565    3566    3567#   3568    3569    3570#   3571    3572    3573#   3574
                          3575    3576#   3577    3578    3579#   3580    3581    3582#   3584    3585    3586#   3587    3588
                          3589#   3590    3591    3592#   3593    3594    3595#   3596    3597    3598#   3599    3600    3601#
                          3602    3603    3604#   3605    3606    3607#   3608    3609    3610#   3612    3613    3614#   3615
                          3616    3617#   3618    3619    3620#   3621    3622    3623#   3624    3625    3626#
$TRPAD  021112            3529    3548#
$TSTNM  001102             156#   2983    2999    3185    3221*   3223    3227
$TTYIN  020414            3344    3345    3362    3366#
$TYPBN= ****** U          3559
$TYPDS= ****** U          3559
$TYPE   020636             691    3457#   3539    3549
$TYPEC  021006            3286    3478    3485    3492    3497#   3498
```

K08


```
$TYPEX  021054          3503    3505    3508#
$TYPOC  017146          3081#   3550    3552
$TYPON  017162          3080    3083#   3558
$TYPOS  017122          3076#   3555
$XTSTR  017534          3197#
$OFILL  017345          3077*   3081*   3091    3126#
$40CAT= ****** U        2990    3194
.     = 027240            21#     25#     31#    153#    188     457     469     512     543     589     639     711     814
                         816     820     950   1002    1018    1575    1630    1634    1654    1660    1668   1691#   1740
                        1770    1794    1816   1823    1836    1841    1859    1862    1872    1878    1886    1979    2006
                        2018    2241    2246   2265    2270    2289    2294    2313    2318    2333    2338    2377    2380
                        2385    2388    2391   2394    2397    2400    2407    2411    2533    2544    2596    2610    2764
                        2772   2959#    2999   3048#   3148    3175    3226    3227    3233   3366#    3367    3373    3437
                        3510   3634#    3638
```

```
COMMEN   145#
E.XOCOM  145#
ERROR     39#   836   839  1139  1155  1162  1189  1208  1225  1248  1257  1272  1293  1310  1319
         1328  1346  1397  1408  1424  1432  1445  1460  1476  1492  1515  1522
ESCAPE   145#
GETPRI   145#
GETSWR   145#
MULT     145#
NEWTST   145#
POP      145#  2829  2869  3158  3160  3425
PUSH     145#  2808  2850  3136  3143  3391
REPORT   145#
SCOPE     40#   921  1140  1164  1190  1210  1227  1250  1274  1332  1347  1399  1410  1437  1448
         1462  1516  1524
SETPRI   145#
SETTRA     3#  3539#  3550  3553  3556  3560  3564  3567  3570  3573  3576  3579  3583  3586  3589
         3592  3595  3598  3601  3604  3607  3611  3614  3617  3620  3623
SETUP    145#   451
SKIP     145#
SLASH    145#
SPACE    145#
STARS    145#   148   188  2791  2840  2879  2899  2968  3002  3052  3131  3150  3181  3231  3235
         3305  3335  3376  3441  3517
SWRSU    145#   470#
TRMTRP  3539#
TYPBIN   145#
TYPDEC   145#
TYPNAM   145#
TYPNUM   145#
TYPOCS   145#
TYPOCT   145#   577  1602  2023  2030  2037  2051  2058  2647  2659  3015  3040  3254
TYPTXT   145#
$$CMRE   146#
$$CMTM   146#
$$ESCA   145#
$$NEWT   145#
$$SET   3539#  3540  3551  3554  3557  3561  3565  3568  3571  3574  3577  3580  3584  3587  3590
         3593  3596  3599  3602  3605  3608  3612  3615  3618  3621  3624
$$SKIP   145#
.EQUAT     3#    34
.HEADE     3#     5
.SETUP     3#   449
.TRMTR     3#
.$CATC     3#    18
.$CMTA     3#   145
.$DB2D     3#  2896
.$ERRO     3#  2965
.$ERRT     3#  2999
.$POWE     3#  3128
.$RAND     3#  2837
.$RODE     3#  3373
.$READ     3#  3228
.$SAVE     3#  2788
.$SB2D     3#  2876
.$SCOP     3#  3178
.$TRAP     3#  3514
```

```
MAINDEC-11-DZICA-D      MACY11 30(1046)  02-AUG-77  09:23  PAGE 90
DZICAD.P11    28-JUL-77 14:52          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0103

.$TYPE      3#   3438
.$TYPO      3#   3049


. ABS.  027240     000


 ERRORS DETECTED:  0

 DSKZ:DZICAD.BIN,DSKZ:DZICAD.LST/CRF/SOL/NL:TOC=DSKM:DZICAD.P11
 RUN-TIME:  18  9  1 SECONDS
 RUN-TIME RATIO: 295/28=10.2
 CORE USED:  23K  (45 PAGES)
```

80N