AD02

EXERCISER FOR 1024 CHANNEL

MD-11-DZADI-A

AD02

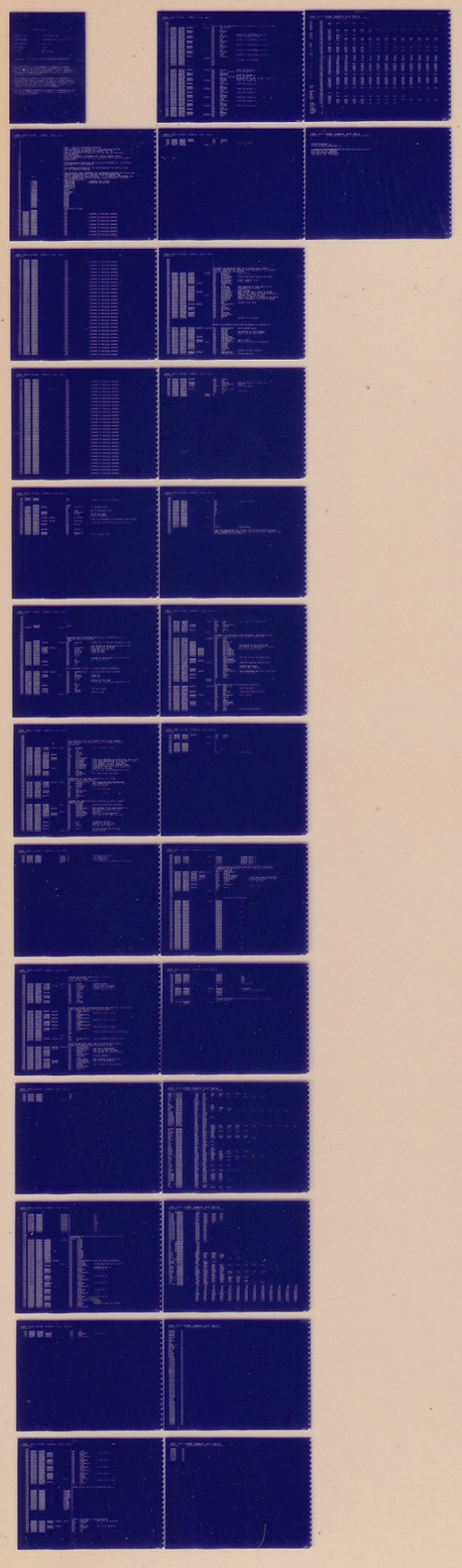
EP-DZADI-A-DL

MAY 1978

digital

FICHE 1 OF 1

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MD-11-DZADI-A-D
PRODUCT NAME:	EXERCISER FOR 1024 CH. ADØ2
DATE CREATED:	FEB. 1976
MAINTAINER:	IPGCP
AUTHOR:	RAY BALDWIN

```
                                        ;
                                        ;
                                        ;
                                        ;AD01 - 1024 MUX EXPANSION DIAGNOSTIC
                                        ;THIS DIAGNOSTIC IS INTENDED AS A MUX EXTENTION ROUTINE
                                        ;FOR THE AD01-UX DIAGNOSTIC AND ASSUMES THAT THE
                                        ;AD01-D DIAGNOSTIC HAS BEEN RUN AND THAT THE AID FRONT-END
                                        ;IS CALIBRATED.
                                        ;THE MUX REGISTER IS ACCESSED FROM VECTOR ADDRESS 176774
                                        ;WHICH CONSISTS OF DATA BITS 0-9 INCLUSIVE, BIT 0 BEING THE LEAST
                                        ;SIGNIFICANT
                                        ;
                                        ;THIS DIAGNOSTIC EXERCISES THE MULTIPLIER REGISTER AND MULTIPLEXER
                                        ;SWITCHES UP TO 1024 CHANNELS.
                                        ;
                                        ;THE CONTROLLER PORTION OF THE TEST EXERCISES THE DIGITAL LOGIC
                                        ;OF THE MULTIPLER REGISTER
                                        ;
                                        ;THE STABILITY TEST EXERCISES THE MULTIPLEXER CHANNELS BY TAKING 1000
                                        ;DECIMAL CONVENTIONS ON A CHANNEL SPECIFIED THROUGH THE
                                        ;SWITCH REGISTER BIT 0-9 INCLUSIVE.  IF AN ERROR IS ENCOUNTERED OVER
                                        ;THE ALLOWABLE COUNT SPREAD A DIGITAL VALUE HISTOGRAM IS PRINTED
                                        ;OUT ON THE TELETYPE.
          177566                        TYDB=177566             ;TELETYPE DATA BUFFER
          177564                        TYSR=177564             ;TELETYPE STATUS REG.
          176770                        ADCS=176770
          176772                        ADDR=176772
          176774                        ADMX=176774
          000000                        HLT=0
          177570                        SR=177570
          000240                        NOP=240
          000000                        R0=%0
          000001                        R1=%1
          000002                        R2=%2
          000003                        R3=%3
          000004                        R4=%4
          000005                        R5=%5
          000006                        SP=%6
          000007                        PC=%7
                                        ;LOAD TRAP CATCHER
          000000                        .=0
000000    000002                        .+2
000002    000000                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000004    000006                        .+2
000006    000000                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000010    000012                        .+2
000012    000000                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000014    000016                        .+2
000016    000000                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000020    000022                        .+2
000022    000000                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000024    000026                        .+2
000026    000000                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000030    000032                        .+2
000032    000000                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS
```

```
 57  000034  000036                    .+2
 58  000036  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 59  000040  000042                    .+2
 60  000042  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 61  000044  000046                    .+2
 62  000046  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 63  000050  000052                    .+2
 64  000052  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 65  000054  000056                    .+2
 66  000056  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 67  000060  000062                    .+2
 68  000062  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 69  000064  000066                    .+2
 70  000066  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 71  000070  000072                    .+2
 72  000072  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 73  000074  000076                    .+2
 74  000076  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 75  000100  000102                    .+2
 76  000102  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 77  000104  000106                    .+2
 78  000106  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 79  000110  000112                    .+2
 80  000112  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 81  000114  000116                    .+2
 82  000116  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 83  000120  000122                    .+2
 84  000122  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 85  000124  000126                    .+2
 86  000126  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 87  000130  000132                    .+2
 88  000132  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 89  000134  000136                    .+2
 90  000136  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 91  000140  000142                    .+2
 92  000142  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 93  000144  000146                    .+2
 94  000146  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 95  000150  000152                    .+2
 96  000152  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 97  000154  000156                    .+2
 98  000156  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
 99  000160  000162                    .+2
100  000162  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
101  000164  000166                    .+2
102  000166  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
103  000170  000172                    .+2
104  000172  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
105  000174  000176                    .+2
106  000176  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
107  000200  000202                    .+2
108  000202  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
109  000204  000206                    .+2
110  000206  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
111  000210  000212                    .+2
112  000212  000000                    HALT            ;TRAPPED TO PREVIOUS ADDRESS
```

```
   113  000214  000216                    .+2
   114  000216  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   115  000220  000222                    .+2
   116  000222  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   117  000224  000226                    .+2
   118  000226  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   119  000230  000232                    .+2
   120  000232  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   121  000234  000236                    .+2
   122  000236  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   123  000240  000242                    .+2
   124  000242  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   125  000244  000246                    .+2
   126  000246  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   127  000250  000252                    .+2
   128  000252  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   129  000254  000256                    .+2
   130  000256  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   131  000260  000262                    .+2
   132  000262  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   133  000264  000266                    .+2
   134  000266  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   135  000270  000272                    .+2
   136  000272  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   137  000274  000276                    .+2
   138  000276  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   139  000300  000302                    .+2
   140  000302  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   141  000304  000306                    .+2
   142  000306  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   143  000310  000312                    .+2
   144  000312  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   145  000314  000316                    .+2
   146  000316  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   147  000320  000322                    .+2
   148  000322  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   149  000324  000326                    .+2
   150  000326  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   151  000330  000332                    .+2
   152  000332  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   153  000334  000336                    .+2
   154  000336  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   155  000340  000342                    .+2
   156  000342  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   157  000344  000346                    .+2
   158  000346  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   159  000350  000352                    .+2
   160  000352  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   161  000354  000356                    .+2
   162  000356  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   163  000360  000362                    .+2
   164  000362  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   165  000364  000366                    .+2
   166  000366  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
   167  000370  000372                    .+2
   168  000372  000000                    HALT                ;TRAPPED TO PREVIOUS ADDRESS
```

```
  169  000374  000376                        .+2
  170  000376  000000                        HALT                    ;TRAPPED TO PHEVIOUS ADDRESS
  171                                         ;
  172                                         ;
  173          000200                         .=200
  174  000200  012706  000600                MOV     #STALK,%6       ;INITIALIZE STACK
  175  000204  000240                        NOP
  176  000206  000404                        BR      .+12            ;GO TO STABILITY TEST
  177  000210  012706  000600                MOV     #STACK,%6
  178  000214  000167  000460                JMP     CONTST          ;GO TO MUX TEST
  179  000220  000000                        HLT
  180  000222  016767  177342  002126        MOV     SR,CNTSPR       ;LOAD CNT SPREAD
  181  000230  000000                        HLT
  182  000232  016767  177332  001050        MOV     SR,SIGNK        ;LOAD 10 BITS=2000 11 BITS=4000 12 BITS 10000
  183  000240  000000                        HLT                     ;LOAD GAIN INTO BITS 11+12 CHAN INTO 0-9
  184  000242  005037  002534                CLR     @#PASS
  185  000246  000240                        NOP
  186  000250  000240                        NOP
  187  000252  000137  001030                JMP     @#STAB
  188  000256  000240                        NOP
  189  000260  012706  000600                MOV     #STACK,%6
  190  000264  000137  003420                JMP     @#DIS           ;GO TO DISPLAY TEST
```

```
                                                ;
                                                ;
                                                ;
                                                ;
            000600                              .=600
000600      000000              STACK:  0
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;EXERCISE EACH MULTIPLEXER REGISTER BIT INDEPENDENTLY BY A
                                                ;SEND AND RECEIVE PROCEDURE.
            000700                              .=700
000700      016700  176070          CONTST: MOV     ADMX,R0         ;CHECK THAT INITIALIZE CLEARED MUX. REG.
000704      001401                          BEQ     .+4
000706      000000                          HLT                     ;INT FAILED TO CLEAR MUX OR READ FAILED
000710      012767  177770  000110          MOV     #-10,COUNT      ;SET COUNT FOR 10 XFERS
000716      012700  000001                  MOV     #1,R0           ;LOAD MUX DATA REG (R0)
000722      010067  176046          A1:     MOV     R0,ADMX         ;LOAD MUX REG.
000726      016701  176042                  MOV     ADMX,R1         ;READ MUX REG.
000732      020100                          CMP     R1,R0
000734      001401                          BEQ     .+4
000736      000000                          HLT                     ;FAILED TO READ CORRECT MUX VALUE (R0)
000740      006100                          ROL     R0              ;BAD VALUE IN (R1)
000742      005267  000060                  INC     COUNT
000746      001402                          BEQ     .+6
000750      000167  177746                  JMP     A1
                                                ;
                                                ;RUN INCREMENT PATTERN ON A SEND, RECEIVE PROCEDURE
                                                ;
000754      012767  176000  000044          MOV     #-2000,COUNT    ;SET COUNT FOR 1024 TRANSFERS
000762      005000                          CLR     R0
000764      010067  176004          A2:     MOV     R0,ADMX         ;LOAD MUX
000770      016701  176000                  MOV     ADMX,R1         ;READ MUX
000774      020100                          CMP     R1,R0
000776      001401                          BEQ     .+4             ;BRANCH IF THE SAME
001000      000000                          HLT                     ;VALUE SENT IN R0,VALUE RECEIVED IN R1
001002      005200                          INC     R0
001004      005267  000016                  INC     COUNT
001010      001402                          BEQ     .+6
001012      000167  177746                  JMP     A2              ;TRY NEXT VALUE
001016      005037  176774                  CLR     @#ADMX
001022      000167  177652                  JMP     CONTST          ;RECYCLE TEST
001026      000000              COUNT:  0
```

```
                                        ;
                                        ;
                                        ;THE STABILITY TEST WILL RECEIVE ITS CHANNEL ADDRESS
                                        ;FROM THE SWITCH REGISTER BITS 0-9 AND AVERAGE 1000 DEC.
                                        ;CONVERTIONS
                                        ;
001030  016767  176534  175736  STAB:  MOV     SR,ADMX          ;FETCH CHANNEL ADDRESS
001036  105737  176770                 TSTB    @#ADCS
001042  100375                         BPL     .-4
001044  005737  176772                 TST     @#ADDR
001050  016767  176514  000234         MOV     SR,GAIN
001056  000367  000230                 SWAB    GAIN             ;
001062  042767  177747  000222         BIC     #177747,GAIN     ;
001070  016767  000216  175672         MOV     GAIN,ADCS        ;LOAD GAIN SELECTED IN SWITCH REG. BITS 11+12
001076  004237  001142                 JSR     %2,@#CONV        ;TAKE 1000 CONVERTIONS AT FIXED GAIN + CHAN.
001102  004237  001206                 JSR     %2,@#AVE         ;FIND AVERAGE VALUE OF 1024 CONV.
001106  004237  001476                 JSR     %2,@#SPRSET      ;FIND SPREAD +/-5 FROM AVERAGE VALUE
001112  004237  001632                 JSR     %2,@#CATEG       ;CATEGORIZE THE FIRST 1000 CONVERTIONS
001116  032767  100000  176444         BIT     #100000,SR       ;TEST BIT 15 OF S.R.
001124  001402                         BEQ     .+6              ;IF BIT 15=1 OMIT CHECKING FOR ERROR
001126  000137  001030                 JMP     @#STAB           ;
001132  004237  002262                 JSR     %2,@#CHECK       ;BIT 15=0,;CHECK FOR ERROR
001136  000137  001030                 JMP     @#STAB
                                        ;
                                        ;SUBROUTINE TO TAKE 1024 CONVERTIONS ON A FIXED
                                        ;CHANNEL WITH A FIXED GAIN.
001142  012737  176000  001026  CONV:  MOV     #-2000,@#COUNT   ;SET COUNT FOR 1024 CONVERTIONS
001150  012704  003620                 MOV     #TAB,%4          ;SET ADDRESS POINTER TO LOAD TABLE
001154  005237  176770                 INC     @#ADCS           ;KICK A/D START
001160  105737  176770                 TSTB    @#ADCS
001164  100375                         BPL     .-4              ;WAIT FOR A/D DONE
001166  013724  176772                 MOV     @#ADDR,(%4)+
001172  005237  001026                 INC     @#COUNT
001176  001402                         BEQ     .+6
001200  000137  001154                 JMP     @#CONV+12
001204  000202                         RTS     %2
                                        ;
                                        ;AVERAGE THE 1024 READINGS BY SUMMING ALL INTO A DOUBLE
                                        ;PRECISION WORD.
001206  005037  001306          AVE:   CLR     @#HIORD          ;INITIALIZE SUMMING LOCATIONS
001212  005037  001304                 CLR     @#LOORD          ;
001216  012737  176000  001300         MOV     #-2000,@#CNT1    ;SET COUNTER 1 FOR 1024 CONVERTIONS
001224  012704  003620                 MOV     #TAB,%4          ;INITIALIZE POINTER FOR DATA
001230  012467  000046          AVEA:  MOV     (%4)+,HOLD       ;GET DATA
001234  063767  001310  000040         ADD     @#SIGNK,HOLD     ;MAKE ALL VALUES POSITIVE
001242  066767  000034  000034         ADD     HOLD,LOORD       ;ADD SUMS TO LOW ORDER DIVIDEND
001250  005537  001306                 ADC     @#HIORD
001254  000240                         NOP
001256  000240                         NOP
001260  005267  000014                 INC     CNT1             ;INCREMENT COUNTER
001264  001402                         BEQ     .+6              ;BRANCH IF WE ARE DONE
001266  000137  001230                 JMP     @#AVEA           ;NO, GET NEXT DATA
001272  000240                         NOP
001274  000137  001314                 JMP     @#DIV            ;GO AND DIVIDE MESS BY 1024
001300  000000          CNT1:  0                                ;LOOP COUNTER
```

```
   293  001302  000000                  MOLD:   0                       ;ARITHMETIC STORAGE
   294  001304  000000                  LOORD:  0                       ;LOW ORDER DIV
   295  001306  000000                  HIORD:  0                       ;HIGH ORDER DIV
   296  001310  000000                  SIGNK:  0                       ;10 BITS=2000,11BITS=4000,12 BITS=10000
   297  001312  000000                  GAIN:   0
```

```
                                                ;DIVIDE THE SUMMED CONVERSIONS BY 1024 TO
                                                ;FIND AVERAGE VALUE
                                                ;AVERAGE VALUE REMAINS LOORD
                                                ;
001314  005037  001300          DIV:  CLR     @#CNT1          ;CLEAR COUNTER
001320  012767  177766  177752        MOV     #-12,CNT1       ;SET UP FOR 10 SHIFTS
001326  006237  001306          DIVA: ASR     @#HIORD         ;MOVES BIT 0 TO CARRY
001332  006037  001304                ROR     @#LOURD         ;MOVES CARRY TO BIT 15
001336  103405                      BCS     .+14
001340  005237  001300                INC     @#CNT1
001344  001411                      BEQ     .+24
001346  000137  001326                JMP     @#DIVA
001352  005237  001300                INC     @#CNT1
001356  001402                      BEQ     .+6
001360  000137  001326                JMP     @#DIVA
001364  005237  001304                INC     @#LOURD
                                                ;
                                                ;
                                                ;FIND VALUE WHICH WAS ADDED TO MAKE SIGN POSITIVE, AND SUBTRACT
                                                ;FROM AVERAGE VALUE IN LOW ORDER DIVIDEND
001370  032737  002000  001310        BIT     #2000,@#SIGNK
001376  001405                      BEQ     .+14            ;BRANCH IF NOT 10 BITS
001400  012737  176000  001462        MOV     #176000,@#VAL
001406  000137  001466                JMP     @#SET
001412  032737  004000  001310        BIT     #4000,@#SIGNK
001420  001405                      BEQ     .+14            ;BRANCH IF NOT 11 BITS
001422  012737  174000  001462        MOV     #174000,@#VAL
001430  000137  001466                JMP     @#SET
001434  032737  010000  001310        BIT     #10000,@#SIGNK
001442  001405                      BEQ     .+14            ;BRANCH IF NOT 12 BITS
001444  012737  170000  001462        MOV     #170000,@#VAL
001452  000137  001466                JMP     @#SET
001456  000000                      HLT                     ;WRONG JUSTIFICATION VALUE IN SIGNK
001460  000240                      NOP
001462  000000          VAL:  0
001464  000240                      NOP
001466  063737  001304  001462  SET:  ADD     @#LOORD,@#VAL   ;ADD LOW ORDER DIVIDEND TO VAL
001474  000202                      RTS     %2
                                                ;
                                                ;FIND VALUES WHICH ARE + AND - 5 COUNTS FROM AVERAGE
                                                ;VALUE AND STORE IN THEIR RESPECTIVE GATES.
001476  013737  001462  001600  SPRSET: MOV     @#VAL,@#CNTR
001504  012737  177773  001602        MOV     #-5,@#CNTR+2    ;SET FOR 5 INCREMENTS
001512  012704  001604                MOV     #POINTA,%4      ;SET R4 FOR DATA POINTER
001516  005237  001600                INC     @#CNTR          ;SET COUNTS (+) FROM AVERAGE
001522  013724  001600                MOV     @#CNTR,(%4)+
001526  005237  001602                INC     @#CNTR+2        ;UPDATE COUNTER
001532  001371                      BNE     .-14
001534  013724  001462                MOV     @#VAL,(%4)+     ;MOV AVERAGE VALUE TO GATE.
001540  012737  177773  001602        MOV     #-5,@#CNTR+2    ;SET FOR 5 DECREMENTS
001546  013737  001462  001600        MOV     @#VAL,@#CNTR
001554  005337  001600                DEC     @#CNTR          ;SET COUNTS (-) FROM AVERAGE
001560  013724  001600                MOV     @#CNTR,(%4)+
001564  005237  001602                INC     @#CNTR+2
001570  001371                      BNE     .-14
```

```
   354  001572  000202                          RTS     %2
   355  001574  000240                          NOP
   356  001576  000240                          NOP
   357  001600  000000                  CNTR:   0
   358  001602  000000                          0
```

```
001604  000000                  POINTA: 0                               ;+1
001606  000000                  POINTB: 0                               ;+2
001610  000000                  POINTC: 0                               ;+3
001612  000000                  POINTD: 0                               ;+4
001614  000000                  POINTE: 0                               ;+5
001616  000000                  POINTF: 0                               ;SAME
001620  000000                  POINTG: 0                               ;-1
001622  000000                  POINTH: 0                               ;-2
001624  000000                  POINTI: 0                               ;-3
001626  000000                  POINTJ: 0                               ;-4
001630  000000                  POINTK: 0                               ;-5
                                        ;
                                        ;
                                        ;CATEGORIZE THE FIRST 1000 CONVERSIONS FROM THE TABLE
                                        ;HEADED TAB.
001632  005067  000356          CATEG:  CLR     PLUS1
001636  005067  000354                  CLR     PLUS2
001642  005067  000352                  CLR     PLUS3
001646  005067  000350                  CLR     PLUS4
001652  005067  000346                  CLR     PLUS5
001656  005067  000330                  CLR     SAME
001662  005067  000322                  CLR     MINUS1
001666  005067  000314                  CLR     MINUS2
001672  005067  000306                  CLR     MINUS3
001676  005067  000300                  CLR     MINUS4
001702  005067  000272                  CLR     MINUS5
001706  005067  000314                  CLR     JUNK
001712  012737  176030  001602          MOV     #-1750,@#CNTR+2 ;SET FOR 1000 CONVERTIONS
001720  012704  003620                  MOV     #TAB,%4
001724  012405                  DILL:   MOV     (%4)+,%5        ;START HERE FOR EACH COMPARE
001726  020537  001604                  CMP     %5,@#POINTA
001732  001004                          BNE     .+12            ;BRANCH IF NOT +1
001734  005237  002214                  INC     @#PLUS1         ;VALUE WAS +1
001740  000137  002164                  JMP     @#CAL
001744  020537  001606                  CMP     %5,@#POINTB
001750  001004                          BNE     .+12
001752  005237  002216                  INC     @#PLUS2         ;VALUE WAS +2
001756  000137  002164                  JMP     @#CAL
001762  020537  001610                  CMP     %5,@#POINTC
001766  001004                          BNE     .+12
001770  005237  002220                  INC     @#PLUS3         ;VALUE WAS +3
001774  000137  002164                  JMP     @#CAL
002000  020537  001612                  CMP     %5,@#POINTD
002004  001004                          BNE     .+12
002006  005237  002222                  INC     @#PLUS4         ;VALUE WAS +4
002012  000137  002164                  JMP     @#CAL
002016  020537  001614                  CMP     %5,@#PUINTE
002022  001004                          BNE     .+12
002024  005237  002224                  INC     @#PLUS5         ;VALUE WAS +5
002030  000137  002164                  JMP     @#CAL
002034  020537  001616                  CMP     %5,@#POINTF
002040  001004                          BNE     .+12
002042  005237  002212                  INC     @#SAME          ;VALUE WAS SAME AS AVERAGE
002046  000137  002164                  JMP     @#CAL
002052  020537  001620                  CMP     %5,@#POINTG
```

```
   415  002056  001004                    BNE     .+12
   416  002060  005237  002210            INC     @#MINUS1        ;VALUE WAS -1
   417  002064  000137  002164            JMP     @#CAL
   418  002070  020537  001622            CMP     R5,@#POINTH
```

```
002074 001004                  BNE    .+12
002076 005237 002206           INC    @#MINUS2        ;VALUE WAS -2
002102 000137 002164           JMP    @#CAL
002106 020537 001624           CMP    %5,@#POINTI
002112 001004                  BNE    .+12
002114 005237 002204           INC    @#MINUS3        ;VALUE WAS -3
002120 000137 002164           JMP    @#CAL
002124 020537 001626           CMP    %5,@#POINTJ
002130 001004                  BNE    .+12
002132 005237 002202           INC    @#MINUS4        ;VALUE WAS -4
002136 000137 002164           JMP    @#CAL
002142 020537 001630           CMP    %5,@#POINTK
002146 001004                  BNE    .+12
002150 005237 002200           INC    @#MINUS5        ;VALUE WAS -5
002154 000137 002164           JMP    @#CAL
002160 005237 002226           INC    @#JUNK
002164 005237 001602   CAL:    INC    @#CNTR+2
002170 001402                  BEQ    .+6             ;BRANCH IF ALL DONE
002172 000137 001724           JMP    @#DILL
002176 000202                  RTS    %2
                               ;
                               ;STORE COUNTS HERE FOR CATEGORIZED RESULTS
                               ;
002200 000000          MINUS5: 0
002202 000000          MINUS4: 0
002204 000000          MINUS3: 0
002206 000000          MINUS2: 0
002210 000000          MINUS1: 0
002212 000000          SAME:   0
002214 000000          PLUS1:  0
002216 000000          PLUS2:  0
002220 000000          PLUS3:  0
002222 000000          PLUS4:  0
002224 000000          PLUS5:  0
002226 000000          JUNK:   0
                               ;
                               ;
                               ;
                               ;
                               ;
                               ;TYPE A LINE FEED & CARRIAGE RETURN
002230 012767 000015 175330  LFCR:  MOV    #15,TYDB        ;MOV C.R. TO TELETYPE
002236 105737 177564           TSTB   @#TYSR
002242 100375                  BPL    .-4
002244 012767 000012 175314    MOV    #12,TYDB        ;MOV LF TO TELETYPE
002252 105737 177564           TSTB   @#TYSR
002256 100375                  BPL    .-4
002260 000203                  RTS    %3
```

M1

```
                                                ;
                                                ;SUBROUTINE TO CHECK FOR ERROR FROM DESIRED ALLOWABLE COUNTSPREAD
002262  013704  002356          CHECK:  MOV     @#CNTSPR,%4
002266  001002                          BNE     .+6
002270  000137  002370                  JMP     @#CHECK1
002274  022704  000001                  CMP     #1,%4
002300  001002                          BNE     .+6             ;BRANCH IF COUNTSPREAD IS NOT 1
002302  000137  002414                  JMP     @#CHECK2        ;CHECK FOR ERROR AT CS=1
002306  022704  000002                  CMP     #2,%4
002312  001002                          BNE     .+6             ;BRANCH IF COUNTSPREAD IS NOT 2
002314  000137  002434                  JMP     @#CHECK3
002320  022704  000003                  CMP     #3,%4
002324  001002                          BNE     .+6             ;BRANCH IF COUNTSPREAD IS NOT 3
002326  000137  002454                  JMP     @#CHECK4
002332  022704  000004                  CMP     #4,%4
002336  001002                          BNE     .+6             ;BRANCH IF COUNTSPREAD IS NOT 4
002340  000137  002474                  JMP     @#CHECK5
002344  022704  000005                  CMP     #5,%4
002350  001003                          BNE     .+10            ;BRANCH IF COUNT SPREAD IS NOT 5
002352  000137  002514                  JMP     @#CHECK6
002356  000240          CNTSPR: NOP
002360  000000                  HLT                             ; ILLEGAL COUNT SPREAD
002362  000240                  NOP
002364  000240                  NOP
002366  000240                  NOP
                                                ;
                                                ;
002370  005737  002214          CHECK1: TST     @#PLUS1         ;TEST FOR VALUES +1
002374  001402                          BEQ     .+6     ;BRANCH IF NO VALUES AT +1
002376  000137  002414                  JMP     @#CHECK2
002402  005737  002210                  TST     @#MINUS1        ;TEST FOR VALUES -1
002406  001402                          BEQ     .+6     ;BRANCH IF NO VALUES AT -2
002410  000137  002536                  JMP     @#ERROR         ;CONVERT ALL VALUES TO DEC + TYPE
002414  005737  002216          CHECK2: TST     @#PLUS2         ;TEST FOR VALUES +2
002420  001003                          BNE     .+10
002422  005737  002206                  TST     @#MINUS2        ;TEST FOR VALUES -2
002426  001402                          BEQ     .+6
002430  000137  002536                  JMP     @#ERROR
002434  005737  002220          CHECK3: TST     @#PLUS3         ;TEST FOR VALUES +3
002440  001003                          BNE     .+10
002442  005737  002204                  TST     @#MINUS3        ;TEST FOR VALUES -3
002446  001402                          BEQ     .+6
002450  000137  002536                  JMP     @#ERROR
002454  005737  002222          CHECK4: TST     @#PLUS4         ;TEST FOR VALUES +4
002460  001003                          BNE     .+10
002462  005737  002202                  TST     @#MINUS4        ;TEST FOR VALUES -4
002466  001402                          BEQ     .+6
002470  000137  002536                  JMP     @#ERROR
002474  005737  002224          CHECK5: TST     @#PLUS5         ;TEST FOR VALUES +5
002500  001003                          BNE     .+10
002502  005737  002200                  TST     @#MINUS5
002506  001402                          BEQ     .+6
002510  000137  002536                  JMP     @#ERROR
002514  005737  002226          CHECK6: TST     @#JUNK
002520  001402                          BEQ     .+6
```

```
   524  002522  000137  002536                    JMP     @#ERHDR
   525  002526  005237  002534                    INC     @#PASS
   526  002532  000202                            RTS     R2              ;OK ALL IS GOOD
   527  002534  000000                    PASS:   0
```

```
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;AN ERROR WAS DETECTED FROM THE ALLOWABLE COUNT SPREAD
                                                ;PATTERN THEREFORE THE AVERAGE VALUE FOLLOWED BY THE HISTOGRAM
                                                ;WILL BE OUTPUT ON THE TELETYPE.
002536  004337  002230          ERROR1: JSR     %3,@#LFCR       ;L.F. + C.R.
002542  012704  002534                  MOV     #PASS,%4
002546  004337  003142                  JSR     %3,@#BINDEC     ;PRINT PASS COUNT SINCE LAST ERROR
002552  004337  002230                  JSR     %3,@#LFCR
002556  004337  002706                  JSR     %3,@#VALAVE     ;PRINT AVERAGE VALUE
002562  004337  002230                  JSR     %3,@#LFCR       ;L.F. + C.R.
002566  032737  040000  177570          BIT     #40000,@#SR
002574  001006                          BNE     .+16
002576  012705  003500                  MOV     #LINE,%5        ;SET LOCATION IN TYPE ROUTINE FOR
002602  004337  003116                  JSR     %3,@#TYPE       ;HISTOGRAM DATA OUTPUT
002606  004337  002230                  JSR     %3,@#LFCR       ;L.F. + C.R.
002612  012737  177765  001600          MOV     #-13,@#CNTR     ;SET COUNTER FOR M TRIPS TO BINDEC
002620  012704  002200                  MOV     #MINUS5,%4      ;REG. 4 WILL POINT AT VALUE TO BE CONV.
002624  004337  003142          LAP:    JSR     %3,@#BINDEC     ;CONVERT VALUE AND RAP OUT
002630  012705  003606                  MOV     #SPACE,%5       ;REG 4 WILL POINT AT SPACES TO BE OUTPUT
002634  004337  003116                  JSR     %3,@#TYPE       ;PRINT 3 SPACES TO EVEN OUTPUT ORDER
002640  005237  001600                  INC     @#CNTR
002644  001367                          BNE     LAP             ;BRANCH UNTIL DONE
002646  022737  001750  002226          CMP     #1750,@#JUNK
002654  001005                          BNE     .+16
002656  012705  003612                  MOV     #KJK,%5
002662  004337  003116                  JSR     %3,@#TYPE
002666  000202                          RTS     %2
002670  012704  002226                  MOV     #JUNK,%4
002674  004337  003142                  JSR     %3,@#BINDEC
002700  005037  002534                  CLR     @#PASS
002704  000202                          RTS     %2              ;RETURN TO TRY AGAIN
                                                ;
                                                ;
                                                ;
                                                ;PRINT THE AVERAGE VALUE WHICH IS STORED IN LOCATION VAL
                                                ;
002706  013737  001462  003052  VALAVE: MOV     @#VAL,@#STOR    ;GET AVERAGE VALUE
002714  005037  003054                  CLR     @#STORA
002720  012705  003100                  MOV     #PA,%5          ;R5 POINTS TO DATA STORAGE
002724  006337  003052                  ASL     @#STOR          ;SHIFT 1ST BIT INTO CARRY
002730  006137  003054                  ROL     @#STORA
002734  062737  000060  003054          ADD     #60,@#STORA
002742  013725  003054                  MOV     @#STORA,(%5)+
002746  012737  177773  001600          MOV     #-5,@#CNTR      ;DO 5 TIMES
002754  012737  177775  001602  LIP:    MOV     #-3,@#CNTR+2    ;SHIFT AND ROTATE 3 TIMES EACH
002762  005037  003054                  CLR     @#STORA
002766  006337  003052                  ASL     @#STOR
002772  006137  003054                  ROL     @#STORA
002776  005237  001602                  INC     @#CNTR+2
003002  001371                          BNE     .-14            ;BRANCH IF NOT 3 SHIFTS
003004  062737  000060  003054          ADD     #60,@#STORA
003012  013725  003054                  MOV     @#STORA,(%5)+   ;STORE ASCI DATA
```

```
   584  003016  005237  001600                  INC     @#CNTR
   585  003022  001354                          BNE     LIP
   586  003024  012705  003056                  MOV     #LOV,%5         ;R5 NOW POINTS AT OUTPUT DATA
   587  003030  012537  177566                  MOV     (%5)+,@#TYDB    ;OUTPUT
   588  003034  105737  177564                  TSTB    @#TYSR          ;WAIT FOR DONE
   589  003040  100375                          BPL     .-4
   590  003042  022715  177777                  CMP     #177777,(%5)
   591  003046  001370                          BNE     .-16            ;RAP NEXT
   592  003050  000203                          RTS     %3
   593  003052  000000              STOR:   0
   594  003054  000000              STORA:  0
```

```
                                                ;
003056  000101                        LOV:    101                     ;AVE. VAL XXXXXX
003060  000126                                126
003062  000105                                105
003064  000056                                56
003066  000126                                126
003070  000101                                101
003072  000114                                114
003074  000056                                56
003076  000040                                40
003100  000000                        PAT:    0
003102  000000                                0
003104  000000                                0
003106  000000                                0
003110  000000                                0
003112  000000                                0
003114  177777                                177777                  ;TERMINATOR
                                                ;
                                                ;THE TYPE SUBROUTINE WILL OUTPUT DATA ON THE ASR-33 TELETYPE
                                                ;UNTIL TERMINATED BY A NEG. BYTE. THE STARTING DATA LOCATION MUST
                                                ;BE PRELOADED INTO REGISTER 5
```

```
                                                ;
                                                ;
003116  105715                  TYPE:   TSTB    (%5)            ;IF A NEG, BYTE WE ARE DONE
003120  100407                          BMI     RET
003122  112537  177566                  MOVB    (%5)+,@#TYDB
003126  105737  177564                  TSTB    @#TYSR
003132  100375                          BPL     .-4
003134  000137  003116                  JMP     @#TYPE
003140  000203                  RET:    RTS     %3
                                        ;
                                        ;CONVERT A 16 BIT OCTAL WORD TO DECIMAL AND STORE IN ASCI.
                                        ;VALUE TO BE CONVERTED IS REFERENCED THROUGH REG. 4
003142  005724                  BINDEC: TST     (%4)+
003144  001004                          BNE     .+12
003146  012705  003606                  MOV     #SPACE,%5
003152  000137  003116                  JMP     @#TYPE
003156  012700  003406                  MOV     #NEG,%0         ;%0 POINTS TO NEG #S FOR SUB
003162  012701  003372                  MOV     #DEC,%1         ;%1 POINTS TO LOCS FOR DEC. VALUES
003166  012737  000060  003372          MOV     #60,@#DEC       ;LOAD ASCI BASE VALUES
003174  012737  000060  003374          MOV     #60,@#DEC+2
003202  012737  000060  003376          MOV     #60,@#DEC+4
003210  012737  000060  003400          MOV     #60,@#DEC+6
003216  012737  000060  003402          MOV     #60,@#DEC+10
003224  005744                          TST     -(%4)           ;SET REG 4 BACK TO CORRECT LOC
003226  011137  003302                  MOV     (%4),@#SAVE
003232  013737  003302  003300          MOV     @#SAVE,@#SUMS
003240  061037  003300          BIDE:   ADD     (%0),@#SUMS     ;SUBTRACT DECIMAL EQUIVALENTS
003244  100004                          BPL     .+12
003246  005720                          TST     (%0)+           ;UPDATE SUB VALUE
003250  005721                          TST     (%1)+           ;UPDATE DEC. LOCATOR
003252  000137  003232                  JMP     @#BIDE-6
003256  013737  003300  003302          MOV     @#SUMS,@#SAVE   ;SAVE REMAINDER FOR WHEN # GOES NEG.
003264  005211                          INC     (%1)            ;+1 TO ASCI VALUE
003266  005737  003300                  TST     @#SUMS
003272  001404                          BEQ     .+12
003274  000137  003240                  JMP     @#BIDE
003300  000000          SUMS:   0
003302  000000          SAVE:   0
                                        ;
                                        ;OUTPUT DEC+4, +6, +10 AND RETURN TO CONTROL
003304  012700  003376          DECOUT: MOV     #DEC+4,%0
003310  005710                          TST     (%0)            ;TEST FOR DONE
003312  100425                          BMI     DONE
003314  022710  000060                  CMP     #60,(%0)        ;CHECK FOR FIRST NON-ZERO
003320  001011                          BNE     DIG
003322  012737  000040  177566          MOV     #40,@#TYDB      ;RAP A SPACE
003330  105737  177564                  TSTB    @#TYSR
003334  100375                          BPL     .-4
003336  005720                          TST     (%0)+
003340  000137  003310                  JMP     @#DECOUT+4
003344  011037  177566          DIG:    MOV     (%0),@#TYDB
003350  105737  177564                  TSTB    @#TYSR
003354  100375                          BPL     .-4
003356  005720                          TST     (%0)+           ;CHECK FOR TERMINATOR
003360  100402                          BMI     DONE
```

```
   673  003362  000137  003344                    JMP     @#DIG
   674  003366  005724                    DONE:   TST     (%4)+
   675  003370  000203                            RTS     %3
   676                                             ;
   677                                             ;
   678  003372  000000                    DEC:    0
   679  003374  000000                            0
   680  003376  000000                            0
   681  003400  000000                            0
   682  003402  000000                            0
   683  003404  177777                            177777                  ;TERMINATOR
   684                                             ;
```

```
003406  154360                    NEG:    154360                  ;010000 OCTAL =
003410  176030                            176030                  ;001000 OCTAL =
003412  177634                            177634                  ;000100 OCTAL =
003414  177766                            177766                  ;000010 OCTAL =
003416  177777                            177777                  ;000001 OCTAL =
                                          ;
                                          ; LOAD THE SWITCH REGISTER BITS 0-9 INTO THE MULTIPLEXER
                                          ;REGISTER AND BITS 11&12 INTO THE GIN SELECT BITS OF THE
                                          ;STATUS REGISTER
003420  013737  177570  003476    DIS:    MOV     @#SR,@#BUFF
003426  000337  003476                    SWAB    @#BUFF
003432  042737  177747  003476            BIC     #177747,@#BUFF
003440  113737  003476  176770            MOVB    @#BUFF,@#ADCS          ; MOVE GAIN INTO STATUS REG.
003446  013737  177570  176774            MOV     @#SR,@#ADMX            ;LOAD CHANNEL-START CONVERT
003454  105737  176770                    TSTB    @#ADCS                 ; WANT FOR DONE
003460  100375                            BPL     .-4
003462  013700  176772                    MOV     @#ADDR,%0
003466  000005                            RESET                          ;DSPLAY DATA
003470  000137  003420                    JMP     @#DIS
003474  000240                            NOP
003476  000000                    BUFF:   0
                                          ;
                                          ;
                                          ;
                                                  ;HISTOGRAM STORED HERE
                                          ;
003500  032455                    LINE:   032455                  ;-5
003502  020040                            020040
003504  020040                            020040
003506  032055                            032055                  ;-4
003510  020040                            020040
003512  020040                            020040
003514  031455                            031455                  ;-3
003516  020040                            020040
003520  020040                            020040
003522  031055                            031055                  ;-2
003524  020040                            020040
003526  020040                            020040
003530  030455                            030455                  ;-1
003532  020040                            020040
003534  020040                            020040
003536  030040                            030040                  ;0
003540  020040                            020040
003542  020040                            020040
003544  030453                            030453                  ;+1
003546  020040                            020040
003550  020040                            020040
003552  031053                            031053                  ;+2
003554  020040                            020040
003556  020040                            020040
003560  031453                            031453                  ;+3
003562  020040                            020040
003564  020040                            020040
003566  032053                            032053                  ;+4
003570  020040                            020040
```

```
   741  003572  020040                          020040
   742  003574  032453                          032453                  ;+5
   743  003576  020040                          020040                  ;2SP
   744  003600  020040                          020040                  ;2SP
   745  003602  051117                          051117                  ;OR
   746  003604  177777                          177777                  ;TERMINATOR
   747                                          ;
   748                                          ;
   749  003606  020040                  SPACE:  020040                  ;2 SPACES
   750  003610  177440                          177440                  ;1 SPACE AND TERMINATOR
   751  003612  030061                  KJK:    030061
   752  003614  030060                          030060
   753  003616  177777                          177777
   754                                          ;
   755                                          ;ANALOG TO DIGITAL CONVERTIONS ARE STORED IN FOLLOWING
   756                                          ;1024 LOCATIONS
   757  003620  000000                  TAB:    0
   758          000001                                  .END
```

```
ADCS  = 176770            27#     245     251*    266*    267     698*    700
ADDR  = 176772            28#     247     269     702
ADMX  = 176774            29#     206     211*    212     225*    226     234*    244*    699*
AVE     001206          253     277#
AVEA    001230          281#    289
A1      000722          211#    219
A2      000764          225#    233
BIDE    003240          644#    648     653
BINDEC  003142          538     549     568     638#
BUFF    003476          695*    696*    697*    698     706#
CAL     002164          393     397     401     405     409     413     417     422     426     438     434     436#
CATEG   001632          255     375#
CHECK   002262          259     471#
CHECK1  002370          473     496#
CHECK2  002414          476     498     502#
CHECK3  002434          479     507#
CHECK4  002454          482     512#
CHECK5  002474          485     517#
CHECK6  002514          488     522#
CNTR    001600          340*    341*    343*    344     345*    346*    349*    350*    351     352*    357#    387*    436*
                        547*    552*    575*    576*    580*    584*
CNTSPR  002356          188*    471     489#
CNT1    001380          279*    287*    292#    303*    304*    308*    311*
CONTST  000700          178     206#    235
CONV    001142          252     264#    272
COUNT   001026          209*    217*    223*    231*    236#    264*    278*
DEC     003372          635     636*    637*    638*    639*    648*    658     678#
DECOUT  003304          650#    667
DIG     003344          662     668#    673
DILL    001724          389#    438
DIS     003420          198     695#    704
DIV     001314          291     303#
DIVA    001326          305#    310     313
DONE    003366          668     672     674#
ERROR   002536          501     506     511     516     521     524     536#
GAIN    001312          248*    249*    250*    251     297#
HIORD   001306          277*    284*    295#    305*
HLT   = 000000            30#     179     181     183     208     215     229     331     498
HOLD    001302          281*    282*    283     293#
JUNK    002226          386*    435*    454#    522     554     559
KJK     003612          556     751#
LAP     002624          549#    553
LFCR    002230          461#    536     539     541     546
LINE    003500          544     712#
LIP     002754          576#    585
LOORD   001304          278*    283*    294#    306*    314*    335
LOV     003056          586     597#
MINUS1  002210          381*    416*    447#    499
MINUS2  002206          382*    421*    446#    504
MINUS3  002204          383*    425*    445#    509
MINUS4  002202          384*    429*    444#    514
MINUS5  002200          385*    433*    443#    519     548
NEG     003406          634     686#
NOP   = 000240            32#
PA      003100          578     686#
PASS    002534          184*    525*    527#    537     561*
```

```
PC      =%000007       40#
PLUS1     002214      375*     392*     449#     496
PLUS2     002216      376*     396*     450#     502
PLUS3     002220      377*     400*     451#     507
PLUS4     002222      378*     404*     452#     512
PLUS5     002224      379*     408*     453#     517
POINTA    001604      342      360#     390
POINTB    001606      361#     394
POINTC    001610      362#     398
POINTD    001612      363#     402
POINTE    001614      364#     406
POINTF    001616      365#     410
POINTG    001620      366#     414
POINTH    001622      367#     418
POINTI    001624      368#     423
POINTJ    001626      369#     427
POINTK    001630      370#     431
RET       003140      621      626#
R0      =%000000       33#
R1      =%000001       34#
R2      =%000002       35#
R3      =%000003       36#
R4      =%000004       37#
R5      =%000005       38#
SAME      002212      380*     412*     448#
SAVE      003302      642*     643      649*     655#
SET       001466      322      326      330      335#
SIGNK     001310      182*     282      296#     319      323      327
SP      =%000006       39#
SPACE     003606      550      632      749#
SPRSET    001476      254      340#
SR      = 177570       31#     180      182      244      248      256      542      695      699
STAB      001030      187      244#     258      260
STACK     000600      174      177      189      197#
STOR      003052      568*     571*     578*     593#
STORA     003054      569*     572*     573*     574      577*     579*     582*     583      594#
SUMS      003300      643*     644*     649      651      654#
TAB       003620      265      280      308      757#
TYDB    = 177566       25#     461*     464*     587*     622*     663*     668*
TYPE      003116      545      551      557      620#     625      633
TYSR    = 177564       26#     462      465      588      623      664      669
VAL       001462      321*     325*     329*     333#     335*     340      347      349      568
VALAVE    002706      540      560#
.       = 003622       42#      43       45       47       49       51       53       55       57       59       61       63       65
                       67       69       71       73       75       77       79       81       83       85       87       89       91
                       93       95       97       99      101      103      105      107      109      111      113      115      117
                      119      121      123      125      127      129      131      133      135      137      139      141      143
                      145      147      149      151      153      155      157      159      161      163      165      167      169
                      173#     176      196#     205#     207      214      218      228      232      246      257      268      271
                      288      307      309      312      320      324      328      346      353      391      395      399      403
                      407      411      415      420      424      428      432      437      463      466      472      475      478
                      481      484      487      497      500      503      505      508      510      513      515      518      520
                      523      543      555      581      589      591      624      631      645      652      665      670      701
```

DZADI.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
COMMEN      1#
ENDCOM      1#
ESCAPE      1#
GETPRI      1#
GETSWR      1#
MULT        1#
NEWTST      1#
POP         1#
PUSH        1#
REPORT      1#
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#
STARS       1#
SWRSU       1#
TYPBIN      1#
TYPDEC      1#
TYPNAM      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
$$ESCA      1#
$$NEWT      1#
$$SKIP      1#
.EQUAT      1#
.HEADE      1#
.KT11       1#
.SETUP      1#
.SWRMI      1#
.SACT1      1#
.SAPTB      1#
.SAPTM      1#
.SAPTY      1#
.SASTA      1#
.SCATC      1#
.SCMTA      1#
.SD82D      1#
.SDB2O      1#
.SDIV       1#
.SEOP       1#
.SERRO      1#
.SERRT      1#
.SMULT      1#
.SPOWE      1#
.SRAND      1#
.SRDDE      1#
.SRDOC      1#
.SREAD      1#
.SR2AZ      1#
.SSAVE      1#
.SSB2D      1#
.SSB2O      1#
.SSCOP      1#
.SSIZE      1#
```

```
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$40CA      1#
.1170       1#
```

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADC      284
ADD      282   283   335   573   582   644
ASL      571   578
ASR      305
BCS      307
BEQ      207   214   218   228   232   257   271   288   329   312   320   324   328   437   497
         500   505   510   515   520   523   652
BIC      250   697
BIT      256   319   323   327   542
BMI      621   660   672
BNE      346   353   391   395   399   403   407   411   415   420   424   428   432   472   475
         478   481   484   487   503   508   513   518   543   553   555   581   585   591   631
         662
BPL      246   268   463   466   589   624   645   665   670   701
BR       176
CLR      184   224   234   277   278   303   375   376   377   378   379   380   381   382   383
         384   385   386   561   569   577
CMP      213   227   390   394   398   402   406   410   414   418   423   427   431   474   477
         480   483   486   554   590   661
DEC      350
HALT      44    46    48    50    52    54    56    58    60    62    64    66    68    70    72
          74    76    78    80    82    84    86    88    90    92    94    96    98   100   102
         104   106   108   110   112   114   116   118   120   122   124   126   128   130   132
         134   136   138   140   142   144   146   148   150   152   154   156   158   160   162
         164   166   168   170
INC      217   230   231   266   270   287   300   311   314   343   345   352   392   396   400
         404   408   412   416   421   425   429   433   435   436   525   552   560   564   650
JMP      170   187   190   219   233   235   250   260   272   289   291   310   313   322   326
         330   393   397   401   405   409   413   417   422   426   430   434   438   473   476
         479   482   485   488   498   501   506   511   516   521   524   625   633   640   653
         667   673   704
JSR      252   253   254   255   259   536   538   539   540   541   545   546   549   551   557
         560
MOV      174   177   180   182   189   206   209   210   211   212   223   225   226   244   248
         251   264   265   269   279   280   301   304   321   325   329   340   341   342   344
         347   348   349   351   387   388   389   461   464   471   537   544   547   548   550
         556   559   560   570   574   575   576   583   586   587   632   634   635   636   637
         638   639   640   642   643   649   650   663   660   695   699   702
MOVB     622   698
NOP      175   185   186   188   285   286   290   332   334   355   356   489   491   492   493
         705
RESET    703
ROL      216   572   579
ROR      306
RTS      273   336   354   439   467   526   550   562   592   626   675
SWAB     249   696
TST      247   496   499   502   504   507   509   512   514   517   519   522   638   641   646
         647   651   659   666   671   674
TSTB     245   267   462   465   588   620   623   664   669   700
.ENABL     1
.END     758
.LIST      1
.MACRO     1
.NLIST     1
.REPT     43
```

```
 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DZADI/SOL/CRF/PAGNUM=SYSMAC.SML[400,1066],DZADI[400,4571]
 RUN-TIME:  21 23 1 SECONDS
 RUN-TIME RATIO: 238/46=4.9
 CORE USED:  32K  (63 PAGES)
```