AUTOSIZER

SED M SYSTEM EXERCISER

MD-11-DDQAC-A

EP-DDQAC-A-DL-A OCT 1976

FICHE 1 OF 1

digital

Made in U.S.A.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH A LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

THIS IS AN AUTOSIZER PROGRAM FOR THE SEMI1 SYSTEM.

THIS PROGRAM TYPES OUT THE KNOWN GUYS (DEVICE CONTROL REGISTER ADDRESS,
DEVICE IDENTIFICATION #,VECTOR ADDRESS) THAT ARE FOUND IN THE SYSTEM.
IF ANY OF THE KNOWN DEVICES THAT SHOULD BE IN THE SYSTEM
ARE NOT FOUND THEN ITS DEVICE CONTROL REGISTER ADDRESS IS
INCREMENTED BY 1 AND IS TYPED OUT ALONG WITH DEVICE IDENTIFICATION
AND VECTOR ADDRESS.
THEN IT TYPES OUT THE DEVICE CONTROL REGISTER ADDRESS OF ANY
UNKNOWN DEVICE FOUND IN THE SYSTEM.
THERE IS AN ENTRY OF '0' IN THE OUTPUT TABLE AT THE END OF KNOWN
DEVICES AND BEFORE THE BEGINING OF UNKNOWN DEVICES.
THE PROGRAM ALSO STORES AWAY THE OUTPUT TABLE INTO MEMORY
STARTING AT LOCATION 3000.

FOLLOWING ARE THE IDENTIFICATION #'S OF DEVICES:

DL11=1	CR11=2	KW11W=3
BME?3=4	DL11E=5	KW11L=6
DL11A=7	SWR=10	

```
                        .ENABLE  ABS,AMA
                        .MCALL .HEADER, .STYPE, .STYPOCT, .EQUAT, .SCATCH,.STRAP
                        .MCALL  SETUP,.SETUP, .SPOWER
000000                  $TN=0
000000                  $SWR=0
                        .TITLE  MAINDEC-11-DZSSA-B
                        ;*COPYRIGHT (C) APRIL 1975
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY GORA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-A1).
                        ;*

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR               ;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE               ;BASIC DEFINITION OF SCOPE CALL
177776                  PS=     177776                  ;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
```

```
        177774                  STKLMT= 177774                  ;STACK LIMIT REGISTER
        177772                  PIRQ=   177772                  ;PROGRAM INTERRUPT REQUEST REGISTER
        177570                  SWR=    177570                  ;SWITCH REGISTER
        177570                  DISPLAY=SWR

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

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

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
        100000                  BIT15=  100000
        040000                  BIT14=  40000
        020000                  BIT13=  20000
        010000                  BIT12=  10000
        004000                  BIT11=  4000
        002000                  BIT10=  2000
        001000                  BIT09=  1000
        000400                  BIT08=  400
        000200                  BIT07=  200
        000100                  BIT06=  100
```

```
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          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

                                        .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

                                        .SBTTL  STARTING ADDRESS(ES)
                        000200          .=200
000200  000137  001100                  JMP     @#.START                ;JUMP TO STARTING ADDRESS OF PROGRAM


                        001100          .=1100
                        000005  .START:  RESET
001102  012706  001100  1$:     MOV #STACK,SP
001106  012737  000340  177776          MOV #340,PS
001114  012706  001100                  MOV     #STACK,SP               ;SETUP THE STACK POINTER
001120  012737  007504  000034          MOV     #$TRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
001126  012737  000340  000036          MOV     #340,@#TRAPVEC+2        ;LEVEL 7
001134  012737  007344  000024          MOV     #$PWRDN,@#PWRVEC        ;POWER FAILURE VECTOR
001142  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;LEVEL 7

001150  012737  001214  000004          MOV #DE./SVC,@#4
```

```
                                ;
                                ;
001156  012700  001536          MOV #TABLE,R0
001162  012701  003000          MOV #LIST,R1
001166  005770  000000  BACK:   TST @(R0)
001172  012021                  MOV (R0)+,(R1)+
001174  012021                  MOV (R0)+,(R1)+
001176  012021                  MOV (R0)+, (R1)+
001200  005710                  TST (R0)
001202  001371                  BNE BACK
001204  012721  000000          MOV #0,(R1)+
001210  000137  001620          JMP RESUME
                                ;
                                ;
001214  012011          DEVSVC: MOV (R0)+,(R1)
001216  062721  000001          ADD #1,(R1)+
001222  012021                  MOV (R0)+,(R1)+
001224  012021                  MOV (R0)+,(R1)+
001226  012716  001166          MOV #BACK,(SP)
001232  000002                  RTI
                                ;
                                ;
001234  005015  052504  030461  DU11:   .ASCIZ <15><12>/DU11 /
001244  005015  046104  030461  DL11:   .ASCIZ <15><12>/DL11E /
001255      015  042012  030522  DR11:   .ASCIZ <15><12>/DR11C /
001266  005015  053513  030461  KW11:   .ASCIZ <15><12>/KW11W /
001277      015  041012  034115  BM873:  .ASCIZ <15><12>/BM873YA /
001312  005015  053513  030461  KW11L:  .ASCIZ <15><12>/KW11L /
001323      015  042012  030514  DL11A:  .ASCIZ <15><12>/DL11A /
001334  005015  044506  044516  FINISH:.ASCIZ<15><12>/FINISH /<15><12><12><12>
001352  005015  005012  047125  UNKNOWN:    .ASCIZ<15><12><12><12>/UNKNOWN DEVICES: /<15><12><12>
001403      015  000012          MCRLF: .ASCIZ<15><12>
001406  005015  052517  050124  HDR:  .ASCII <15><12>/OUTPUT TABLE STARTS: /<15><12><12><12>
001441      015  040412  042104          .ASCIZ <15><12>/ADDRESS   ID#  VECTOR /<15><12>
001474  005015  052501  047524  TITLE:  .ASCIZ<15><12>/AUTOSIZER----SEDM11 SYSTEM /<15><12><12>
                                ;
                                ;
                                ;
001536                          .EVEN
001536  160040          TABLE:    .WORD 160040
001540  000001                  .WORD 001
001542  000330                  .WORD 330
001544  167770                  .WORD 167770
001546  000002                  .WORD 002
001550  000300                  .WORD 300
001552  172400                  .WORD 172400
001554  000003                  .WORD 003
001556  000320                  .WORD 320
001560  173000                  .WORD 173000
001562  000004                  .WORD 004
001564  000000                  .WORD 000
001566  175610                  .WORD 175610
001570  000005                  .WORD 005
001572  000310                  .WORD 310
001574  177546                  .WORD 177546
001576  000006                  .WORD 006
```

```
001600  000100                          .WORD 100
001602  177560                          .WORD 177560
001604  000007                          .WORD 007
001606  000060                          .WORD 60
001610  177570                          .WORD 177570
001612  000010                          .WORD 008.
001614  000000                          .WORD 000
001616  000000                          0

001620  012737  002074  000004  RESUME:  MOV #NEWSVC,4
001626  012737  000340  000006          MOV #340,6
001634  012702  001536                  MOV #TABLE,R2
001640  012737  160000  002072          MOV #160000,TAG

001646  023712  002072          TEST:    CMP    TAG,(R2)
001652  001421                  BEQ     AD
001654  023727  002072  177540          CMP     TAG,#177540
001662  001415                  BEQ     AD
001664  005777  000202                  TST     @TAG

001670  013721  002072                  MOV TAG,(R1)+
001674  062737  000010  002072  REPEAT:  ADD #10,TAG
001702  023727  002072  177610          CMP TAG,#177610
001710  001415                          BEQ FIN
001712  000137  001646                  JMP     TEST
001716  023727  002072  173000  AD:      CMP     TAG,#173000
001724  001453                  BEQ     SKIP
001726  062702  000006                  ADD #6,R2
001732  062737  000010  002072          ADD #10,TAG
001740  000137  001646                  JMP TEST
001744  012711  000000          FIN:    MOV #0,(R1)
001750  012701  003000                  MOV #LIST,R1
001754  104400  001474                  TYPE, TITLE
001760  104400  001436                  TYPE, HDR
001764                          TYPOUT:
001764  012146                  MOV     (R1)+,-(SP)     ;SAVE (R1)+ FOR TYPEOUT
001766  104404                  TYPOS                   ;GO TYPE--OCTAL ASCII
001770     006                  .BYTE   6               ;TYPE 6 DIGITS
001771     000                  .BYTE   0               ;SUPPRESS LEADING ZEROS
001772  012146                  MOV     (R1)+,-(SP)     ;SAVE (R1)+ FOR TYPEOUT
001774  104404                  TYPOS                   ;GO TYPE--OCTAL ASCII
001776     006                  .BYTE   6               ;TYPE 6 DIGITS
001777     000                  .BYTE   0               ;SUPPRESS LEADING ZEROS
002000  012146                  MOV     (R1)+,-(SP)     ;SAVE (R1)+ FOR TYPEOUT
002002  104404                  TYPOS                   ;GO TYPE--OCTAL ASCII
002004     006                  .BYTE   6               ;TYPE 6 DIGITS
002005     000                  .BYTE   0               ;SUPPRESS LEADING ZEROS
002006  104400  001403          TYPE, MCRLF
002012  005711                  TST (R1)
002014  001363                  BNE TYPOUT
002016  012146                  MOV     (R1)+,-(SP)     ;SAVE (R1)+ FOR TYPEOUT
002020  104404                  TYPOS                   ;GO TYPE--OCTAL ASCII
002022     006                  .BYTE   6               ;TYPE 6 DIGITS
002023     000                  .BYTE   0               ;SUPPRESS LEADING ZEROS
```

```
003024  104400  001352                  TYPE , UNKNOWN
003030                          UNNON:
003030  012146                          MOV     (R1)+,-(SP)             ;SAVE (R1)+ FOR TYPEOUT
003032  104404                          TYPOS                           ;GO TYPE--OCTAL ASCII
003034     006                          .BYTE   6                       ;TYPE 6 DIGITS
003035     000                          .BYTE   0                       ;SUPPRESS LEADING ZEROS
003036  104400  001403                  TYPE,MCRLF
003042  005711                          TST (R1)
003044  001371                          BNE UNNON
003046  000000                          HALT
003050  000137  001100                  JMP     .START

003054  062737  001000  003072  SKIP:   ADD #1000,TAG
003062  062702  000006                  ADD     #6,R2
003066  000137  001646                  JMP TEST
003072  000000                  TAG:  .WORD 0


003074  012716  001674                  NEWSVC:  MOV #REPEAT,(SP)
003100  000002                                   RTI

        003000                  .=3000
003000  002000                  LIST: .BLKW 2000
                                ;;*****************************************************************

                                .SBTTL  TYPE ROUTINE

                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*
                                ;*2) USING A JSR INSTRUCTION
                                ;*      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
                                ;*      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
                                ;*      MESADDR                         ;FIRST ADRESS OF MESSAGE

007000  105737  007115          $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
007004  100002                          BPL     1$                      ;BR IF YES
007006  000000                          HALT                            ;HALT HERE IF NO TERMINAL
007010  000407                          BR      3$                      ;LEAVE
007012  010046                  1$:     MOV     R0,-(SP)                ;SAVE R0
007014  017600  000002                  MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
007020  112046                  2$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
007022  001005                          BNE     4$                      ;BR IF IT ISN'T THE TERMINATOR
007024  005726                          TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
007026  012600                          MOV     (SP)+,R0                ;RESTORE R0
007030  062716  000002          3$:     ADD     #2,(SP)                 ;ADJUST RETURN PC
```

```
007034  000002                          RTI                     ;RETURN
007036  004737  007070          4$:     JSR     PC,7$           ;GO TYPE THIS CHARACTER
007042  123726  007114          5$:     CMPB    $FILLC,(SP)+    ;IS IT TIME FOR FILLER CHARS.?
007045  001364                          BNE     2$              ;IF NO GO GET NEXT CHAR.
007050  013746  007112                  MOV     $NULL,-(SP)     ;GET # OF FILLER CHARS. NEEDED
                                                                ;AND THE NULL CHAR.
007054  105366  000001          6$:     DECB    1(SP)           ;DOES A NULL NEED TO BE TYPED?
007060  002770                          BLT     5$              ;BR IF NO--GO POP THE NULL OFF OF STACK
007062  004737  007070                  JSR     PC,7$           ;GO TYPE A NULL
007066  000772                          BR      6$              ;LOOP
007070  105777  000012          7$:     TSTB    @$TPS           ;WAIT UNTIL PRINTER IS READY
007074  100375                          BPL     7$
007076  116677  000002  000004          MOVB    2(SP),@$TPB     ;LOAD CHAR TO BE TYPED INTO DATA REG.
007104  000207                          RTS     PC
007106  177564                  $TPS:   177564                  ;TTY PRINTER STATUS REG. ADDRESS
007110  177566                  $TPB:   177566                  ;TTY PRINTER BUFFER REG. ADDRESS
007112  000                     $NULL:  .BYTE   0               ;CONTAINS NULL CHARACTER FOR FILLS
007113  002                     $FILLS: .BYTE   2               ;CONTAINS # OF FILLER CHARACTERS REQUIRED
007114  012                     $FILLC: .BYTE   12              ;INSERT FILL CHARS. AFTER A "LINE FEED"
007115  000                     $TPFLG: .BYTE   0               ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
                                ;;**************************************************************

                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;M=1 OR 0
                                ;*                                      ;1=TYPE LEADING ZEROS
                                ;*                                      ;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$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

007116  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;PICKUP THE MODE
007122  116637  000001  007341          MOVB    1(SP),$OFILL    ;LOAD ZERO FILL SWITCH
007130  112637  007343                  MOVB    (SP)+,$OMODE+1  ;NUMBER OF DIGITS TO TYPE
007134  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN ADDRESS
007140  000406                          BR      $TYPON
007142  112737  000001  007341  $TYPOC: MOVB    #1,$OFILL       ;SET THE ZERO FILL SWITCH
007150  112737  000006  007343          MOVB    #6,$OMODE+1     ;SET FOR SIX(6) DIGITS
007156  112737  000005  007340  $TYPON: MOVB    #5,$OCNT        ;SET THE ITERATION COUNT
007164  010346                          MOV     R3,-(SP)        ;SAVE R3
007166  010446                          MOV     R4,-(SP)        ;SAVE R4
007170  010546                          MOV     R5,-(SP)        ;SAVE R5
007172  113704  007343                  MOVB    $OMODE+1,R4     ;GET THE NUMBER OF DIGITS TO TYPE
```

```
 430  007176  005404                          NEG     R4
 431  007200  062704  000006                  ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
 432  007204  110437  007342                  MOVB    R4,$OMODE               ;SAVE IT FOR USE
 433  007210  113704  007341                  MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
 434  007214  016605  000012                  MOV     12(SP),R5               ;PICKUP THE INPUT NUMBER
 435  007220  005003                          CLR     R3                      ;CLEAR THE OUTPUT WORD
 436  007222  006105                  1$:     ROL     R5                      ;ROTATE MSB INTO "C"
 437  007224  000404                          BR      3$                      ;GO DO MSB
 438  007226  006105                  2$:     ROL     R5                      ;FORM THIS DIGIT
 439  007230  006105                          ROL     R5
 440  007232  006105                          ROL     R5
 441  007234  010503                          MOV     R5,R3
 442  007236  006103                  3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
 443  007240  105337  007342                  DECB    $OMODE                  ;TYPE THIS DIGIT?
 444  007244  100016                          BPL     7$                      ;BR IF NO
 445  007246  042703  177770                  BIC     #177770,R3              ;GET RID OF JUNK
 446  007252  001002                          BNE     4$                      ;TEST FOR 0
 447  007254  005704                          TST     R4                      ;SUPPRESS THIS 0?
 448  007256  001403                          BEQ     5$                      ;BR IF YES
 449  007260  005204                  4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
 450  007262  052703  000060                  BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
 451  007266  052703  000040          5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
 452  007272  110337  007336                  MOVB    R3,8$                   ;SAVE FOR TYPING
 453  007276  104400  007336                  TYPE    ,8$                     ;GO TYPE THIS DIGIT
 454  007302  105337  007343          7$:     DECB    $OCNT                   ;COUNT BY 1
 455  007306  003347                          BGT     2$                      ;BR IF MORE TO DO
 456  007310  002402                          BLT     6$                      ;BR IF DONE
 457  007312  005204                          INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
 458  007314  000744                          BR      2$                      ;GO DO THE LAST DIGIT
 459  007316  012605                  6$:     MOV     (SP)+,R5                ;RESTORE R5
 460  007320  012604                          MOV     (SP)+,R4                ;RESTORE R4
 461  007322  012603                          MOV     (SP)+,R3                ;RESTORE R3
 462  007324  016666  000002  000004          MOV     2(SP),4(SP)             ;SET THE STACK FOR RETURNING
 463  007332  012616                          MOV     (SP)+,(SP)
 464  007334  000002                          RTI                             ;RETURN
 465  007336  000                     8$:     .BYTE   0                       ;STORAGE FOR ASCII DIGIT
 466  007337  000                             .BYTE   0                       ;TERMINATOR FOR TYPE ROUTINE
 467  007340  000                     $OCNT:  .BYTE   0                       ;OCTAL DIGIT COUNTER
 468  007341  000                     $OFILL: .BYTE   0                       ;ZERO FILL SWITCH
 469  007342  000000                  $OMODE: 0                               ;NUMBER OF DIGITS TO TYPE
 470                                  ;;**************************************************************
 471
 472                                  .SBTTL  POWER DOWN AND UP ROUTINES
 473
 474                                  ;POWER DOWN ROUTINE
 475  007344  012737  007466  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC        ;SET FOR FAST UP
 476  007352  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;PRIO:7
 477  007360  010046                          MOV     R0,-(SP)                ;PUSH R0 ON STACK
 478  007362  010146                          MOV     R1,-(SP)                ;PUSH R1 ON STACK
 479  007364  010246                          MOV     R2,-(SP)                ;PUSH R2 ON STACK
 480  007366  010346                          MOV     R3,-(SP)                ;PUSH R3 ON STACK
 481  007370  010446                          MOV     R4,-(SP)                ;PUSH R4 ON STACK
 482  007372  010546                          MOV     R5,-(SP)                ;PUSH R5 ON STACK
 483  007374  010637  007472                  MOV     SP,$SAVP6               ;SAVE SP
 484  007400  012737  007412  000024          MOV     #$PWRUP,@#PWRVEC        ;SET UP VECTOR
 485  007406  000000                          HALT
```

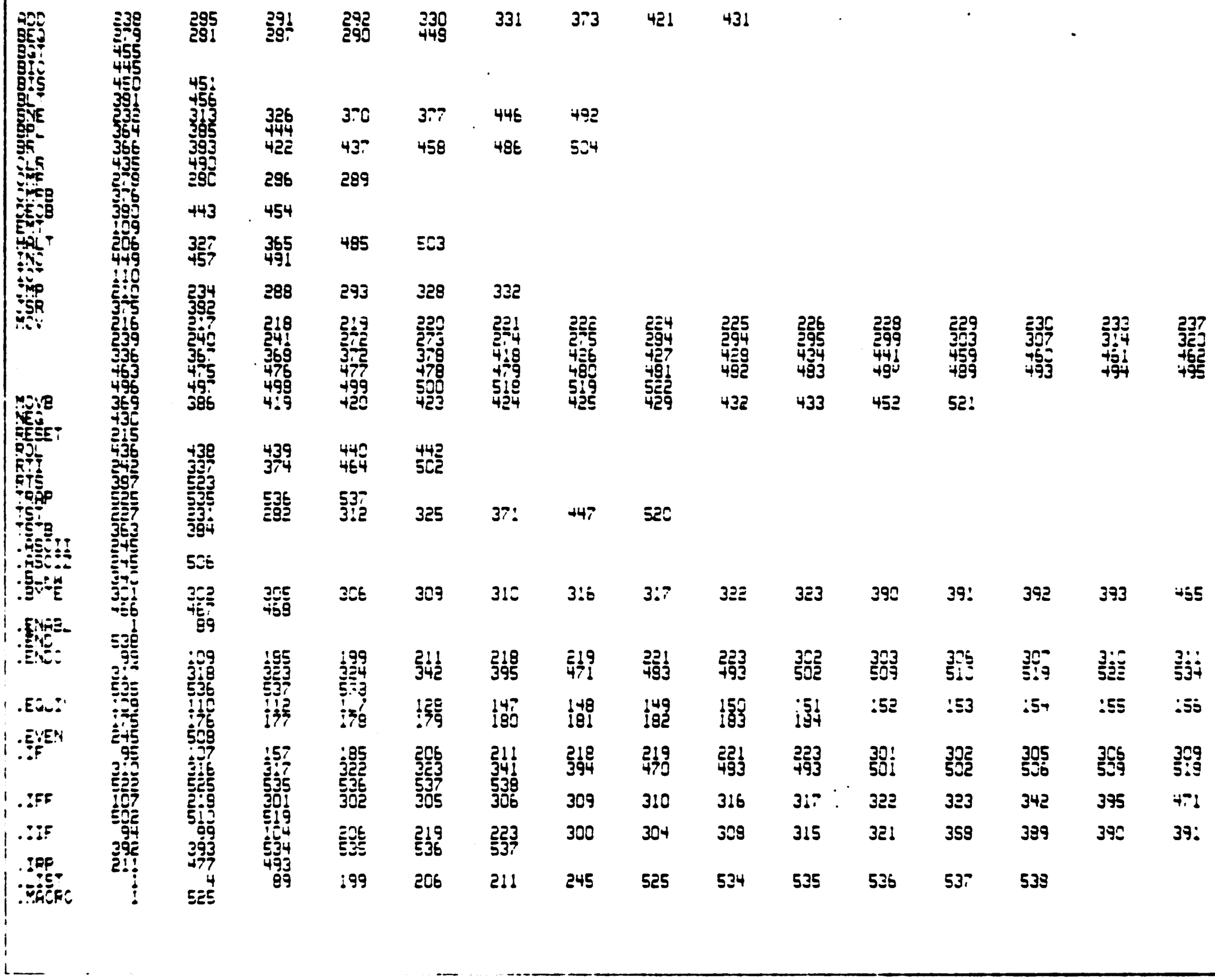
```
 486  007410  000776                          BR      .-2                     ;HANG UP
 487
 488                                  ;POWER UP ROUTINE
 489  007412  013706  007472          $PWRUP: MOV     $SAVR6,SP               ;GET SP
 490  007416  005037  007472                  CLR     $SAVR6                  ;WAIT LOOP FOR THE TTY
 491  007422  005237  007472          1$:     INC     $SAVR6                  ;WAIT FOR THE INC
 492  007426  001375                          BNE     1$                      ;OF  WORD
 493  007430  012605                          MOV     (SP)+,R5                ;POP STACK INTO R5
 494  007432  012604                          MOV     (SP)+,R4                ;POP STACK INTO R4
 495  007434  012503                          MOV     (SP)+,R3                ;POP STACK INTO R3
 496  007436  012602                          MOV     (SP)+,R2                ;POP STACK INTO R2
 497  007440  012601                          MOV     (SP)+,R1                ;POP STACK INTO R1
 498  007442  012600                          MOV     (SP)+,R0                ;POP STACK INTO R0
 499  007444  012737  007344  000024          MOV     #$PWRDN,@#PWRVEC        ;SET UP THE POWER DOWN VECTOR
 500  007452  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;PRIO:7
 501  007460  104400  007474                  TYPE    ,$POWER                 ;POWER FAIL MESSAGE
 502  007464  000002                          RTI
 503  007466  000000                  $ILLUP: HALT                            ;THE POWER UP SEQUENCE WAS STARTED
 504  007470  000776                          BR      .-2                     ; BEFORE THE POWER DOWN WAS COMPLETE
 505  007472  000000                  $SAVR6: 0                               ;PUT THE SP HERE
 506  007474  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
 507  007503  000122
 508                                          .EVEN
 509                                  ;;*****************************************************************
 510
 511                                  .SBTTL  TRAP DECODER
 512
 513                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 514                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 515                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 516                                  ;*GO TO THAT ROUTINE.
 517
 518  007504  010046                  $TRAP:  MOV     R0,-(SP)                ;SAVE R0
 519  007506  016600  000002                  MOV     2(SP),R0                ;GET TRAP ADDRESS
 520  007512  005740                          TST     -(R0)                   ;BACKUP BY 2
 521  007514  111000                          MOVB    (R0),R0                 ;GET RIGHT BYTE OF TRAP
 522  007516  016000  007524                  MOV     $TRPAD(R0),R0           ;INDEX TO TABLE
 523  007522  000200                          RTS     R0                      ;GO TO ROUTINE
 524
 525
 526                                  .SBTTL  TRAP TABLE
 527
 528                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 529                                  ;*BY THE "TRAP" INSTRUCTION.
 530
 531                                  ;       ROUTINE
 532                                  ;       -------
 533  007524                          $TRPAD:
 534  007524  007000                          $TYPE           ;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
 535  007526  007142                          $TYPOC          ;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING
 536  007530  007116                          $TYPOS          ;CALL=TYPOS     TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZE
 537  007532  007156                          $TYPON          ;CALL=TYPON     TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST C
 538          000001                          .END
```

```
AD      001716              279     281     289*
BACK    001166              227*    232     241
BIT0  = 000001              184*
BIT00 = 000001              174*    184
BIT01 = 000002              173*    183
BIT02 = 000004              172*    182
BIT03 = 000010              171*    181
BIT04 = 000020              170*    180
BIT05 = 000040              169*    179
BIT06 = 000100              168*    178
BIT07 = 000200              167*    177
BIT08 = 000400              166*    176
BIT09 = 001000              165*    175
BIT1  = 000002              183*
BIT10 = 002000              164*
BIT11 = 004000              163*
BIT12 = 010000              162*
BIT13 = 020000              161*
BIT14 = 040000              160*
BIT15 = 100000              159*
BIT2  = 000004              182*
BIT3  = 000010              181*
BIT4  = 000020              180*
BIT5  = 000040              179*
BIT6  = 000100              178*
BIT7  = 000200              177*
BIT8  = 000400              176*
BIT9  = 001000              175*
BM873   001277              245*
SPTVEC= 000014              191*
DEVSVC  001214              224     237*
DISPLA= 177570              116*
DL11    001244              245*
DL11A   001323              245*
DR11    001255              245*
DU11    001234              245*
EMTVEC= 000030              194*
ERRVEC= 000004              187*
FIN     001744              287     294*
FINISH  001334              245*
GNS   = ****** U            206     534     535     536     537
HDR     001406              245*    297
IOTVEC= 000020              192*
KW11    001266              245*
KW11L   001312              245*
LIST    003000              226     295     340*
MCRLF   001403              245*    311     324
NEWSVC  002074              272     336*
PC    =%000007              128*    375*    382*    387*
PIRQ  = 177772              114*
PIRQVE= 000240              198*
PS    = 177776              111*    112     217*
PSW   = 177776              112*
PWRVEC= 000024              193*    221*    222*    475*    476*    484*    499*    500*
REPEAT  001674              285*    336
RESUME  001620              234     272*
```

```
RESVEC= 000010           188*
R0     =%000000          119*  225*  227   228   229   230   231   237   239   240   367   368*  369
                         372*  477   498*  518   519*  520   521*  522*  523*
R1     =%000001          120*  226*  228*  229*  230*  233*  237*  238*  239*  240*  284*  294*  295*
                         299   303   307   312   314   320   325   478   497*
R2     =%000002          121*  274*  278   291*  331*  479   496*
R3     =%000003          122*  426   435*  441*  442*  445*  450*  451*  452   461*  480   495*
R4     =%000004          123*  427   429*  430*  431*  432   433*  447   449*  457*  460*  481   494*
R5     =%000005          124*  428   434*  436*  438*  439*  440*  441   459*  482   493*
R6     =%000006          125*  127
R7     =%000007          126*  128
SKIP     002054          290   330*
SP     =%000006          127*  216*  218*  241*  299*  303*  307*  314*  320*  336*  367*  368   369*
                         371   372   373*  376   378*  380*  386   418*  419   420   421*  426*  427*
                         428*  434   459   460   461   462*  463*  477*  478*  479*  480*  481*  482*
                         483   489*  493   494   495   496   497   498   518*  519
STACK  = 001100          108*  216   218
STKLMT= 177774           113*
SWR    = 177570          115*  116
SW0    = 000001          156*
SW00   = 000001          146*  156
SW01   = 000002          145*  155
SW02   = 000004          144*  154
SW03   = 000010          143*  153
SW04   = 000020          142*  152
SW05   = 000040          141*  151
SW06   = 000100          140*  150
SW07   = 000200          139*  149
SW08   = 000400          138*  148
SW09   = 001000          137*  147
SW1    = 000002          155*
SW10   = 002000          136*
SW11   = 004000          135*
SW12   = 010000          134*
SW13   = 020000          133*
SW14   = 040000          132*
SW15   = 100000          131*
SW2    = 000004          154*
SW3    = 000010          153*
SW4    = 000020          152*
SW5    = 000040          151*
SW6    = 000100          150*
SW7    = 000200          149*
SW8    = 000400          148*
SW9    = 001000          147*
TABLE    001536          225   245*  274
TAG      002072          275*  278   280   282   284   285*  286   289   292*  330*  333*
TBITVE= 000014           189*
TEST     001646          278*  288   293   332
TITLE    001474          245*  296
TKVEC  = 000060          196*
TPVEC  = 000064          197*
TRAPVE= 000034           195*  219*  220*
TRTVEC= 000014           190*
TYPE   = 104400          296   297   311   318   324   453   501   534*
TYPOC  = 104402          535*
```

```
TYPON = 104406            537*
TYPOS = 104404            300     304     308     315     321     536*
TYPOUT  001764            298*    313
UNKNOW  001352            245*    318
UNNON   002030            319*    326
$CMTAG= ****** U          218
$FILLC  007114            376     392*
$FILLS  007113            391*
$HD   = 000000            104
$ILLUP  007466            475     503*
$NULL   007112            378     390*
$OCNT   007340            425*    454*    467*
$OMODE  007342            420*    424*    429     432*    443*    469*
$POWER  007474            501     506*
$PWRDN  007344            221     475*    499
$PWRUP  007412            484     489*
$RDCHR= ****** U          538
$RDDEC= ****** U          538
$RDLIN= ****** U          538
$RDOCT= ****** U          538
$SAVRE= ****** U          538
$SAVR6  007472            483*    489     490*    491*    505*
$SETUP= 000014            211*    219     221     223
$STUP = 177777            211*
$SWR  = 000000             93*    104
$TN   = 000000             92*    104
$TPB    007110            386*    389*
$TPFLG  007115            363     393*
$TPS    007106            384     388*
$TRAP   007504            219     518*
$TRP  = 000010            525*    535*    536*    537*    538*
$TRPAD  007524            522     533*
$TYPBN= ****** U          538
$TYPDS= ****** U          538
$TYPE   007000            363*    525     534
$TYPOC  007142            423*    535
$TYPON  007156            422     425*    537
$TYPOS  007116            418*    536
$DFILL  007341            419*    423*    433     468*
.     = 007534            202*    206     208*    214*    245*    339*    340*    388     389     390     391     392     393
                          486     504
.START  001100            210     215*    328
```

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	470	509
[illegible]	[illegible]	[illegible]	299	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	506	507		
[illegible]	[illegible]	91#	[illegible]			
[illegible]	[illegible]	90#	199			
[illegible]	[illegible]	91#	470			
[illegible]	[illegible]	90#	509			
[illegible]	[illegible]	90#	341			
[illegible]	[illegible]	90#	394			



D02


```
.MCALL      90      91     199
.NLIST       1       4      99     199     206     211     245     525     534     535     536     537     538
.REM         5
.REPT      206
.SBTTL     105     200     207     343     396     472     511     526
.TITLE      94
.WORD      206     245     246     247     248     249     250     251     252     253     254     255     256     257     258
           259     260     261     262     263     264     265     266     267     268     333


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*DDQACA,DDQACA.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:SYSMAC.SML,DSKM:DDQACA.P11
RUN-TIME:  20 15 1 SECONDS
RUN-TIME RATIO: 58/38=1.5
CORE USED:  21K  (41 PAGES)
```

Spooler runtime 3 Seconds, 16 KCS, 59 disk reads, 3 disk writes, 16 pages