

PDP11

ENGINEERING EXERCISER
MD-11-DZNCB-C

EP-DZNCB-C-DL-A

COPYRIGHT © 1976

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZNCB-C-D
PRODUCT NAME: GAMMA 11 EXERCISER
DATE CREATED: MAY 1976
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: E. MOORE

COPYRIGHT (C) 1973, 1974 & 1976
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

C.0 TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	EQUIPMENT/DOCUMENTATION - OPTIONAL
2.3	PRELIMINARY PROGRAMS
2.4	STORAGE
3.0	LOADING PROCEDURE
4.0	STARTING PROCEDURE
5.0	OPERATOR OPTIONS
5.1	OPERATOR KEYBOARD OPTIONS
5.2	OPERATOR SWITCH REGISTER OPTION(S)
6.0	RECOMMENDED OPERATOR ACTION
6.1	CAMERA/ADC SETUP
6.2	JOYSTICK SETUP
6.2.1	H306/AR11
6.2.2	H306/NC11
7.0	MISCELLANEOUS
7.1	DEVICE BUS ADDRESS MODIFICATIONS
7.2	FREE RUN MODE (TYPE "F")
7.3	ERROR REPORTING
7.4	EXECUTION TIME
8.0	PROGRAM DESCRIPTION
8.1	DATA CONNECTION & DISPLAY
8.2	JOYSTICK MODES
8.2.1	H306/AR11
8.2.2	H306/NC11
9.0	LISTING

1.0 ABSTRACT

THIS IS A SELF CONTAINED PROGRAM DESIGNED TO CONVERT DATA FROM THE GAMMA CAMERA VIA THE NC11 INTERFACE AND DISPLAY IT ON A VTO1 OR VSVO1 DISPLAY. IT CAN BE USED FOR CAMERA/INTERFACE AND JOYSTICK SETUP OR FOR PRACTICE IN IMAGE MANIPULATION. IT IS MEANT AS A EXERCISER RATHER THAN A DIAGNOSTIC. TOTAL PROGRAM CONTROL IS ACCOMPLISHED THRU THE CONSOLE KEYBOARD.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP-11 COMPUTER
2. I/O TERMINAL (I.E.. ASR33 TT)
3. NC11 INTERFACE
4. GAMMA CAMERA (OR SUBSTITUTE)
5. VTO1 OR VSVO1 DISPLAY

2.2 EQUIPMENT DOCUMENTATION - OPTIONAL

1. H326 JOYSTICK
2. AR11 IF SELECTED FOR JOYSTICK MODE
3. NC11-A CAMERA FRONT-END INSTRUCTION MANUAL
4. DRAWING SET NC11-A-0

2.3 PRELIMINARY PROGRAMS

1. MAINDEC-11-DZNCB SHOULD HAVE PREVIOUSLY BEEN RUN

2.4 STORAGE

THIS PROGRAM USES ALL OF LOWER 12K OF MEMORY.

3.0 LOADING PROCEDURE

NORMAL PROCEDURE FOR LOADING A BINARY PROGRAM INTO MEMORY SHOULD BE FOLLOWED.

4.0 STARTING PROCEDURE

LOADING ADDRESS 200 AND STARTING WILL INITIALIZE THE SYSTEM AND
AWAIT A KEYBOARD COMMAND.

5.0 OPERATOR OPTIONS

5.1 OPERATOR KEYBOARD OPTIONS

TYPE "N"- COLLECT NEW DATA FROM CAMERA - CLEARS CORE MATRIX FIRST
TYPE "C"- COLLECT DATA FROM CAMERA - STARTS NC11, PREVIOUS CORE
MATRIX DATA NOT CLEARED
TYPE "S"- STOP NC11 DATA COLLECTION - ALSO TERMINATES FREE RUN
MODE
TYPE "Z"- ZOOM - SET NC11 TO GAIN 1
TYPE "R"- REGULAR - SET NC11 TO GAIN 2 (DEFAULT COND)
TYPE "V"- SELECT VTO1 DISPLAY
TYPE "W"- SELECT VSVO1 DISPLAY (DEFAULT COND USING 1ST BIT MAP)
TYPE "M"- SELECT VSVO1 DISPLAY AND USE 2ND BIT MAP
TYPE "E"- ERASE THE SELECTED DISPLAY
TYPE "A"- DISPLAY ISOTOPE A (DEFAULT COND)
TYPE "B"- DISPLAY ISOTOPE B (B GAMMA)
TYPE "D"- DISPLAY DATA - DISPLAY SELECTED ISOTOPE
TYPE "L"- GET LOWER THRESHOLD FROM KEYBOARD (DEFAULT=0) - ALL
MATRIX VALUES LESS THAN TYPED VALUE ARE NOT DISPLAYED
TYPE "U"- GET UPPER THRESHOLD FROM KEYBOARD (DEFAULT=177777) -
ALL MATRIX VALUES GREATER THAN TYPED VALUE ARE NOT
DISPLAYED
TYPE "F"- FREE RUN MODE - COLLECT AND DISPLAY NEW DATA
CONTINUOUSLY - TYPE "S" TO TERMINATE THIS MODE
TYPE "G"- INITIALIZE EVERYTHING - START FRESH
TYPE "J"- JOYSTICK CALIBRATION USING AR11 - USE "CNTRL C" TO
TERMINATE MODE
TYPE "K"- JOYSTICK CALIBRATION USING NC11 - USE "CNTRL C" TO
TERMINATE MODE - USE "CNTRL G" TO TURN OFF/ON BELL
(NO CONVERT INDICATION)
TYPE "T"- DISPLAY INTENSITY TEST PATTERN ON SELECTED DISPLAY -
THIS FEATURE IS FOR DISPLAY VERIFICATION ONLY

TYPE "CNTRL S C" - ABORT WHATEVER - BACK TO KEYBOARD MONITOR
TYPE "CNTRL S G" - TURN OFF/ON PRINTER BELL UPON A NO CONVERT -
NC11 JOYSTICK CAL ONLY

5.2 OPERATOR SWITCH REGISTER OPTION(S)

NONE

6.0 RECOMMENDED OPERATOR ACTION

6.1 CAMERA/ADC SETUP

1. PLACE RADIOACTIVE FLOOD SOURCE IN FRONT OF CAMERA DETECTOR OR USE A WEAK RADIOACTIVE SAMPLE ABOUT 3 FEET FROM A DETECTOR WITH THE COLLIMATOR REMOVED.
2. COLLECT DATA (C) AND DISPLAY (D)
3. ADJUST THE X AND Y GAIN CONTROLS ON THE MADO; UNTIL A ROUND CIRCLE* IS SEEN AND CENTERED WITHIN THE BOX ON SCREEN.
4. INCREASE THE CIRCLE* SIZE UNTIL THE EDGES JUST BEGIN TO TOUCH THE BOX.
5. COLLECT DATA FOR A 3 MINUTE PERIOD AND OBSERVE THE IMAGE. IF THERE IS AN ARTIFACT ON THE CENTER OF THE SCREEN THAT LOOKS LIKE A RIGHT ANGLE BRACKET, THE COINCIDENCE DELAY CIRCUIT SHOULD BE ADJUSTED.
6. IF THE MATRIX COUNT EQUALS 65,536, THE Z COUNT EQUALS 0 AND THE DISPLAY IS BLANK, THEN TURN OFF THE 'TEST CLOCK' SWITCH ON THE BACK OF THE INTERFACE AND THE SITUATION WILL BE REMEDIED.
7. IF THE MATRIX AND Z COUNTS EQUAL 0 THEN NO DATA WAS COLLECTED. THEREFORE, NO DISPLAY.

* THIS MAY BE A HEXAGON FOR SOME MAKES OF CAMERA.

6.2 JOYSTICK SETUP

6.2.1 M306/AR11

1. SELECT AR11 JOYSTICK CONFIGURATION BY TYPING "J".
2. ADJUST THE X & Y POTS ON THE M306 JOYSTICK SO THAT THE DOT (VTO1) OR CROSS HAIRS (VSV01) PROVIDE UNIFORM ACCESS WITHIN THE BOX DRAWN ON THE SCREEN. IDEALLY, A PHYSICAL CENTER OF THE JOYSTICK WILL PROVIDE A POINT APPROXIMATELY CENTERED IN THE BOX.
3. CHECK THAT WHEN THE JOYSTICK INTERRUPT BAR IS DEPRESSED THE DISPLAYED POINT DOES NOT FOLLOW THE JOYSTICK.
4. A "CNTRL C" WILL RETURN USER BACK TO THE KEYBOARD MONITOR.

1. SELECT NC11 JOYSTICK CONFIGURATION BY TYPING "K".
2. REPEAT STEP 2 IN 6.2.1. THE X & Y GAIN CONTROLS ON THE NAD01 ADC'S MAY REQUIRE ADJUSTMENT TO REALIZE THIS STEP*.
3. CHECK STEP 3 IN 6.2.1.
4. A "CNTRL C" WILL RETURN USER BACK TO THE KEYBOARD MONITOR.

7.0 MISCELLANEOUS

NC11	MODIFY LOCATION "NCADR" IF BASE BUS ADDRESS IS NOT 164000
VSV01	MODIFY LOCATION "VTVADR" IF BASE BUS ADDRESS (CHARACTER GENERATOR) IS NOT 172600 MODIFY LOCATION "VTMADR" IF BASE BUS ADDRESS (BIT MAP) IS NOT 172620
VT01	MODIFY LOCATION "VTADR" IF BASE BUS ADDRESS IS NOT 176756
AR11	MODIFY LOCATION "ARADR" IF BASE BUS ADDRESS IS NOT 170400

7.2 FREE RUN MODE (TYPE "F")

MODIFICATION OF LOCATION "TIME" WILL ALTER THE DATA COLLECTION TIME IN THE FREE RUN MODE. THE DEFAULT VALUE OF 10(8) PROVIDES ABOUT 1 SECOND.

7.3 ERROR REPORTING

1. INCORRECT KEYBOARD COMMANDS ARE RESPONDED WITH '??'
2. ANY SELECTED DEVICE (SEE SECTION 7.1) WHICH DOES NOT RESPOND ON THE UNIBUS WILL BE INDICATED.

7.4 EXECUTION TIME

THE EXECUTION TIME IS COMPLETELY DEPENDENT UPON THE USER FOR WHATEVER OPTION SELECTED.

8.0 PROGRAM DESCRIPTION

8.1 DATA COLLECTION & DISPLAY

THE USER MAY COLLECT DATA FROM THE GAMMA CAMERA SYSTEM VIA THE NC11 INTERFACE AND DISPLAY THIS DATA ON A VTO1 STORAGE SCOPE OR THE VSVO1 (BITMAP) DISPLAY. ALL FUNCTIONS ARE ENTERED THRU THE KEYBOARD AS DEFINED IN SECTION 5.1. WHEN THE DATA COLLECTION MODE IS SELECTED THE NC11 IS RECEIVING X & Y DATA FROM THE GAMMA CAMERA (LOOKING A RADIOACTIVE SOURCE). THIS INFORMATION IS TRANSFORMED VIA THE ADDRESS MAKER LOGIC WHICH FORMS A UNIQUE ADDRESS WITHIN A DEFINED MATRIX RELATIVE TO THE SCAN POSITION OF THE CAMERA. THE NC11 PERFORMS AN NPR INCREMENT TO MEMORY TO THIS UNIQUE ADDRESS. THE PROGRAM SELECTS THE ADDRESS MATRIX 64X64X16 (RESOLUTION 2) OFFSET TO ADDRESS 20000(8) FOR THE 'A' ISOTOPE, AND FOR THE 'B' ISOTOPE, (CAMERAS EQUIPPED WITH THE DUAL ISOTOPE ATTACHMENT) INFORMATION VIA B GAMMA LOGIC IS STORED STARTING AT ADDRESS 40000 (8). THE INTENSITY OF EACH CELL (MEMORY LOCATION) OF THE IMAGE IN MEMORY IS ADJUSTED WITH THE UPPER AND LOWER THRESHOLDS AND SCALED TO AN INTENSITY LEVEL (32 FOR VTO1 AND 16 FOR VSVO1). THE CELL WITH THE HIGHEST COUNT REPRESENTS THE HIGHEST INTENSITY LEVEL, THEREFORE APPEARS BRIGHTEST ON THE DISPLAY. AT THE COMPLETION OF THE IMAGE DISPLAY, THE FOLLOWING PARAMETERS ARE DISPLAYED:

LOWER THRESHOLD-	ALL VALUES GREATER THAN THIS VALUE ARE DISPLAYED
UPPER THRESHOLD-	ALL VALUES ABOVE THIS VALUE ARE OMITTED
Z COUNT-	32 BIT COUNTER OF A/D START PULSES
MATRIX COUNT-'	TOTAL COUNT OF THE CONTENTS OF EACH CELL IN THE 64X64 MATRIX
ZOOM-	DISPLAYED IF GAIN WAS SELECTED
B-GAMMA-	DISPLAYED IF 'B' ISOTOPE SELECTED

334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356

8.2 JOYSTICK MODES

8.2.1 H306/AR11

SELECTING THIS MODE WILL PROVIDE THE USER WITH A CHECK OF THE ABILITY OF THE H306 WHEN CONNECTED TO THE AR11 TO DISPLAY A X AND Y INDICATION ON THE SELECTED DISPLAY. THIS INDICATION SHOULD AGREE WITH THE PROCEDURE DEFINED IN SECTION 6.2.1. THE AR11 A/D CHANNELS HAVE THE FOLLOWING RELATIONSHIP: CHAN 0 = Y DATA, CHAN 1 = X DATA AND CHAN 2 IS THE INTERRUPT BAR INDICATOR (0=DEPRESSED COND.).

8.2.2 H306/NC11

SELECTING THE NC11 CONFIGURATION TAKES ADVANTAGE OF THE JOYSTICK MODE PROVIDED BY THE NC11. THIS MODE SELECTS THE EXTERNAL INPUTS AND THE NC11 PERFORMS LIKE A A/D WITH THE X & Y CONVERTED VALUES AVAILABLE IN X-Y HOLDING REGISTER. THE X AND Y DATA IS APPLIED TO THE SELECTED DISPLAY AND SHOULD CONFORM TO THE REQUIREMENTS DEFINED IN SECTION 6.2.2.

NOTE: WHEN USING THE VTO1 DISPLAY THE WRITE - THRU MODE IS USED WHICH DISPLAYS A SMALL CIRCLE OR DOT. THE VSV01 USES THE X & Y CROSS HAIRS AS THE DISPLAY INDICATOR.

9.0 LISTING

```

%
.TITLE MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER
;*COPYRIGHT (C) 1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY R.MOORE
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-CO), MAR 21, 1976.
;*
$TN=1
$SWR=160000      ::HALT ON ERROR. LOOP ON TEST, INHIBIT ERROR TYP0UT
.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
$STACK= 1100
.EQUIV EMT,ERROR      ::BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ::BASIC DEFINITION OF SCOPE CALL

```

000001
160000

001100

357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412

```

413      ;*MISCELLANEOUS DEFINITIONS
414      000011      HT=      11      ;;CODE FOR HORIZONTAL TAB
415      000012      LF=      12      ;;CODE FOR LINE FEED
416      000015      CR=      15      ;;CODE FOR CARRIAGE RETURN
417      000200      CRLF=    200      ;;CODE FOR CARRIAGE RETURN-LINE FEED
418      177776      PS=      177776   ;;PROCESSOR STATUS WORD
419      .EQUIV      PS,PSW
420      177774      STKLM=    177774   ;;STACK LIMIT REGISTER
421      177772      PIRQ=    177772   ;;PROGRAM INTERRUPT REQUEST REGISTER
422      177570      DSWR=    177570   ;;HARDWARE SWITCH REGISTER
423      177570      DDISP=    177570   ;;HARDWARE DISPLAY REGISTER
424
425      ;*GENERAL PURPOSE REGISTER DEFINITIONS
426      000000      R0=      %0      ;;GENERAL REGISTER
427      000001      R1=      %1      ;;GENERAL REGISTER
428      000002      R2=      %2      ;;GENERAL REGISTER
429      000003      R3=      %3      ;;GENERAL REGISTER
430      000004      R4=      %4      ;;GENERAL REGISTER
431      000005      R5=      %5      ;;GENERAL REGISTER
432      000006      R6=      %6      ;;GENERAL REGISTER
433      000007      R7=      %7      ;;GENERAL REGISTER
434      .EQUIV      R6,SP      ;;STACK POINTER
435      .EQUIV      R7,PC      ;;PROGRAM COUNTER
436
437      ;*PRIORITY LEVEL DEFINITIONS
438      000000      PR0=      0      ;;PRIORITY LEVEL 0
439      000040      PR1=      40     ;;PRIORITY LEVEL 1
440      000100      PR2=      100    ;;PRIORITY LEVEL 2
441      000140      PR3=      140    ;;PRIORITY LEVEL 3
442      000200      PR4=      200    ;;PRIORITY LEVEL 4
443      000240      PR5=      240    ;;PRIORITY LEVEL 5
444      000300      PR6=      300    ;;PRIORITY LEVEL 6
445      000340      PR7=      340    ;;PRIORITY LEVEL 7
446
447      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
448      100000      SW15=    100000
449      040000      SW14=    40000
450      020000      SW13=    20000
451      010000      SW12=    10000
452      004000      SW11=    4000
453      002000      SW10=    2000
454      001000      SW09=    1000
455      000400      SW08=    400
456      000200      SW07=    200
457      000100      SW06=    100
458      000040      SW05=    40
459      000020      SW04=    20
460      000010      SW03=    10
461      000004      SW02=    4
462      000002      SW01=    2
463      000001      SW00=    1
464      .EQUIV      SW09,SW9
465      .EQUIV      SW08,SW8
466      .EQUIV      SW07,SW7
467      .EQUIV      SW06,SW6
468      .EQUIV      SW05,SW5

```

```

469      .EQUIV SW04,SW4
470      .EQUIV SW03,SW3
471      .EQUIV SW02,SW2
472      .EQUIV SW01,SW1
473      .EQUIV SW00,SW0
474
475      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
476      BIT15= 100000
477      BIT14= 40000
478      BIT13= 20000
479      BIT12= 10000
480      BIT11= 4000
481      BIT10= 2000
482      BIT09= 1000
483      BIT08= 400
484      BIT07= 200
485      BIT06= 100
486      BIT05= 40
487      BIT04= 20
488      BIT03= 10
489      BIT02= 4
490      BIT01= 2
491      BIT00= 1
492      .EQUIV BIT09,BIT9
493      .EQUIV BIT08,BIT8
494      .EQUIV BIT07,BIT7
495      .EQUIV BIT06,BIT6
496      .EQUIV BIT05,BIT5
497      .EQUIV BIT04,BIT4
498      .EQUIV BIT03,BIT3
499      .EQUIV BIT02,BIT2
500      .EQUIV BIT01,BIT1
501      .EQUIV BIT00,BIT0
502
503      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
504      ERRVEC= 4      ;; TIME OUT AND OTHER ERRORS
505      RESVEC= 10     ;; RESERVED AND ILLEGAL INSTRUCTIONS
506      TBITVEC= 14    ;; "T" BIT
507      TRTVEC= 14     ;; TRACE TRAP
508      BPTVEC= 14     ;; BREAKPOINT TRAP (BPT)
509      IOTVEC= 20     ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
510      PWRVEC= 24     ;; POWER FAIL
511      EMTVEC= 30     ;; EMULATOR TRAP (EMT) **ERROR**
512      TRAPVEC= 34    ;; "TRAP" TRAP
513      TKVEC= 60      ;; TTY KEYBOARD VECTOR
514      TPVEC= 64      ;; TTY PRINTER VECTOR
515      PIRQVEC= 240   ;; PROGRAM INTERRUPT REQUEST VECTOR
516
517      ;SOME COMMON PROGRAM VALUES AND EQUATES
518
519      CLZ=20          ;CLEAR Z REG AT REG 14
520      CLALL=100       ;CLEAR NC11 AT REG 14
521      MATRIX=20000    ;STARTING ADRS OF MATRIX DATA
522
523      .SBTTL TRAP CATCHER

```

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 11
DZNCB.P11 TRAP CATCHER

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 11
DZNCB.P11 TRAP CATCHER

000000	000000	
000174	000174	
000176	000176	
000200	000137	001356

```

.=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
.=174
DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0          ;;SOFTWARE SWITCH REGISTER
.SBTTL   STARTING ADDRESS(ES)
        JMP      @*START  ;;JUMP TO STARTING ADDRESS OF PROGRAM

```

.SBTTL COMMON TAGS

```

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

```

535									
536									
537									
538									
539									
540									
541		001100							
542	001100								
543	001100	000000							
544	001102	000							
545	001103	000							
546	001104	000000							
547	001106	000000							
548	001110	000000							
549	001112	000000							
550	001114	000							
551	001115	001							
552	001116	000000							
553	001120	000000							
554	001122	000000							
555	001124	000000							
556	001126	000000							
557	001130	000000							
558	001132	000000							
559	001134	000							
560	001135	000							
561	001136	000000							
562	001140	177570							
563	001142	177570							
564	001144	177560							
565	001146	177562							
566	001150	177564							
567	001152	177566							
568	001154	000							
569	001155	002							
570	001156	012							
571	001157	000							
572	001160	077							
573	001161	015							
574	001162	000012							
575									

. = 1100

```

$CMTAG:      .WORD      0      ;; START OF COMMON TAGS
$PASS:       .WORD      0      ;; CONTAINS PASS COUNT
$STNM:       .BYTE      0      ;; CONTAINS THE TEST NUMBER
$ERFLG:      .BYTE      0      ;; CONTAINS ERROR FLAG
$ICNT:       .WORD      0      ;; CONTAINS SUBTEST ITERATION COUNT
$LPADR:      .WORD      0      ;; CONTAINS SCOPE LOOP ADDRESS
$LPERR:      .WORD      0      ;; CONTAINS SCOPE RETURN FOR ERRORS
$ERTTL:      .WORD      0      ;; CONTAINS TOTAL ERRORS DETECTED
$ITEMB:      .BYTE      0      ;; CONTAINS ITEM CONTROL BYTE
$ERMAX:      .BYTE      1      ;; CONTAINS MAX. ERRORS PER TEST
$ERRPC:      .WORD      0      ;; CONTAINS PC OF LAST ERROR INSTRUCTION
$GDADR:      .WORD      0      ;; CONTAINS ADDRESS OF 'GOOD' DATA
$BDADR:      .WORD      0      ;; CONTAINS ADDRESS OF 'BAD' DATA
$GDDAT:      .WORD      0      ;; CONTAINS 'GOOD' DATA
$BDDAT:      .WORD      0      ;; CONTAINS 'BAD' DATA
              .WORD      0      ;; RESERVED--NOT TO BE USED
              .WORD      0
$AUTOB:      .BYTE      0      ;; AUTOMATIC MODE INDICATOR
$INTAG:      .BYTE      0      ;; INTERRUPT MODE INDICATOR
              .WORD      0
$SWR:        .WORD      DSWR    ;; ADDRESS OF SWITCH REGISTER
$DISPLAY:    .WORD      DDISP   ;; ADDRESS OF DISPLAY REGISTER
$TKS:        177560            ;; TTY KBD STATUS
$TKB:        177562            ;; TTY KBD BUFFER
$TPS:        177564            ;; TTY PRINTER STATUS REG. ADDRESS
$TPB:        177566            ;; TTY PRINTER BUFFER REG. ADDRESS
$NULL:       .BYTE      0      ;; CONTAINS NULL CHARACTER FOR FILLS
$FILLS:      .BYTE      2      ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
$FILLC:      .BYTE     12      ;; INSERT FILL CHARS. AFTER A "LINE FEED"
$TPFLG:      .BYTE      0      ;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
$QUES:       .ASCII    /?/?    ;; QUESTION MARK
$CRLF:       .ASCII    <15>    ;; CARRIAGE RETURN
$LF:         .ASCII    <12>    ;; LINE FEED
*****

```

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:
 ;THIS PROGRAM DOES NOT USE THE ABOVE ERROR TABLE

;NC11 BUS ADRS ASSIGNMENTS - MODS ARE TO BE MADE HERE

64000 NCADR: 164000

;VSV01 BUS ADRS ASSIGNMENTS (CHAR GEN) - MODS ARE TO BE MADE HERE

172600 VTVADR: 172600

;VSV01 BUS ADRS ASSIGNMENTS (BIT MAP) - MODS ARE TO BE MADE HERE

172620 VTMADR: 172620

;VTO1 BUS ADRS ASSIGNMENTS - MODS ARE TO BE MADE HERE

176756 VTADR: 176756

;AR11 BUS ADRS ASSIGNMENTS - MODS ARE TO BE MADE HERE

170400 ARADR: 170400

[illegible]

;NC11 REGISTER ADDRESS POINTERS

```

NCCSR: 164000 :COMMAND/STATUS
NCOFF: 164002 :OFFSET
NCAREG: 164004 :ADDRESS
NCZHI: 164006 :HI Z COUNT
NCZLO: 164010 :LO Z COUNT
NCIREG: 164012 :I
NCSFUN: 164014 :SPECIAL FUNCTIONS

```

:VSVDI REGISTER ADDRESS POINTERS (CHARACTER GENERATOR)

```
VTVCRC: 172600      :CHAR/STATUS
VTVCHP: 172602      :CROSS HAIR POS
VTVPOS: 172604      :CHAR POS
```

:VSVDI REGISTER ADDRESS POINTERS (BIT MAP)

```
VTVCSR: :72620 :COMMAND/STATUS
VTVMAP: :72622 :MAP ADDR
VTVPA: :72624 :PIXEL ID
VTVFX1: :72626 :PIXEL BYTE
VTVINT: :72630 :INTENSITY LOOK UP
```

:VT01 REGISTER ADDRESS POINTERS

```

VTGER: 176756      : COMMAND STATUS
VTXDAC: 176760      : X DAC
VTYDAC: 176762      : Y DAC

```

;ARI: REGISTER ADDRESS POINTERS

```

ARCSR: 170403      :COMMAND STATUS 3 CHANNEL
ARBUF: 170402      :A/D BUFFER

```

COMMON PROGRAM TAGS AND STORAGE LOCATIONS

00000000	00000000	TEMPO:	0	:COMMON UTILITY LOC
00000000	00000000	TEMP1:	00	:COMMON UTILITY LOC
00000000	00000000	TEMP2:	00	:COMMON UTILITY LOC
00000000	00000000	DTYPE:	00	:0=VSVO1 DISPLAY, -1=VT01 DISP_LAY
00000000	00000000	INTENS:	20	:TOTAL # OF INTENSITY LEVELS, 16 OR 32
00000000	00000000	INTLUT:	0	:INITIAL INTENSITY SHADE
00000000	00640000	VTNSAV:	6400	:VSVO1 SET UP - FULL SCREEN MONO(BLK/WH), EMA DISPLAY
00000000	00000000	MAXADR:	00	:CURRENT CORE ADRS OF CELL BEING DISPLAYED
00000000	00000000	MAPADR:	00	:CURRENT ADRS OF BIT MAP LD
00000000	00000000	PIXCNT:	3	:CURRENT PIXEL BYTE COUNT
00000000	00000000	PIXASM:	0	:4 PIXELS ASSEMBLED HERE BEFORE BIT MAP LD
00000000	00000000	HORIZ:	0	:COUNTS 64 DISPLAYED POSITIONS EACH ROW (VT01)
00000000	00000000	VERT:	0	:COUNTS 64 ROWS FOR VT01 DISPLAY
00000000	00000000	XREF:	0	:X POS FOR VT01 DISPLAY
00000000	00000000	YREF:	0	:Y POS FOR VT01 DISPLAY
00000000	00000000	KBCBUF:	0	:CONTAINS ASCII CHAR THAT NEEDS CONVERSION FOR VT01 DISP
00000000	00000000	KBUFF:	0	:CONTAINS KEYBOARD CHAR TYPED
00000000	00000000	TTYOUT:	0	:OUTPUT CHAR TO PRINTER
00000000	00000000	THLG:	0	:LOW THRESHOLD VALUE APPLIED TO MATRIX CELLS
00000000	17777777	THHI:	177777	:HIGH THRESHOLD VALUE APPLIED TO MATRIX CELLS
00000000	00000000	COMSAV:	0	:SAVED NC11 CSR CONTENTS WHEN DISPLAYING
00000000	00000000	GAIN:	0	:GAIN 2=0, GAIN 1=40
00000000	01000000	TOTSIZ:	4096.	:TOTAL # OF CELLS IN MATRIX (ISTOPE A OR B)
00000000	00000000	CHARCT:	0	:COUNTS TOTAL # OF CELLS IN MATRIX (ISTOPE A OR B)
00000000	00000000	BASX:	0	:CONTAINS CURRENT X DAC POSITION (VT01)
00000000	00000000	BASY:	0	:CONTAINS CURRENT Y DAC POSITION (VT01)
00000000	00000006	INXY:	6	:CONTAINS CHAR SIZE OFFSET (VT01)
00000000	00000000	CPEARL:	0	:CONTAINS LARGEST CELL OF MATRIX WITHIN THRESHOLDS
00000000	00000000	ROWCNT:	0	:COUNTS 64 CELLS EACH ROW OF CORE MATRIX
00000000	02000000	TABLEX:	MATRIX	:CONTAINS STARTING ADRS OF MATRIX OF SELECTED ISOTOPE
00000000	00000000	FREERN:	0	:NON-ZERO SAYS COLLECT NEW DATA & DISPLAY IN FREE RUN MOD
00000000	00000010	TIME:	10	:HIGH ORDER COUNT DELAY FOR COLLECTING DATA-FREE RUN MOD
00000000	00000000	JTYPE:	0	:0=NC11 JOYSTICK SOURCE, -1=ARI1 JOYSTICK SOURCE
00000000	02000000	ARCHAN:	20000	:CONTAINS CURRENT ARI1 CHAN & UNIPOLAR BIT
00000000	00000000	MSELC:	0	:0 SAYS 1ST BIT MAP, -1 SAYS 2ND BIT MAP (VSVO1)
00000000	00000000	BELLEN:	0	:0 SAYS RING BELL ON NO CONVERT, -1 SAYS CON*

.SBTTL MAIN PROGRAM

```

001356 START:
.SBTTL INITIALIZE THE COMMON TAGS
::CLEAR THE COMMON TAGS (SCMTAG) AREA
MOV #SCMTAG,R6 ::FIRST LOCATION TO BE CLEARED
CLR (R6)+ ::CLEAR MEMORY LOCATION
CMP #SWR,R6 ::DONE?
BNE -6 ::LOOP BACK IF NO
MOV #STACK,SP ::SETUP THE STACK POINTER
::INITIALIZE A FEW VECTORS
MOV #STRAP,2*TRAPVEC ::TRAP VECTOR FOR TRAP CALLS
MOV #340,2*TRAPVEC+2;LEVEL 7
MOV #SPWRDN,2*PWRVEC ::POWER FAILURE VECTOR
MOV #340,2*PWRVEC+2;LEVEL 7
::SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
::EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
MOV 2*ERRVEC-(SP) ::SAVE ERROR VECTOR
MOV #645,2*ERRVEC ::SET UP ERROR VECTOR
MOV #DSWR,SWR ::SETUP FOR A HARDWARE SWICH REGISTER
MOV #DDISP,DISPLAY ::AND A HARDWARE DISPLAY REGISTER
CMP #-1,2*SWR ::TRY TO REFERENCE HARDWARE SWR
BNE 665 ::BRANCH IF NO TIMEOUT TRAP OCCURRED
::AND THE HARDWARE SWR IS NOT = -1
BR 655 ::BRANCH IF NO TIMEOUT
MOV #655,(SP) ::SET UP FOR TRAP RETURN
645: RTI
655: MOV #SWREG,SWR ::POINT TO SOFTWARE SWR
MOV #DISPREG,DISPLAY
665: MOV (SP)+,2*ERRVEC ::RESTORE ERROR VECTOR

TYPE MSG1 :GO IDENTIFY PROGRAM
JSR PC,SELCTA :DEFAULT TO VSVOI DISPLAY IF THERE
BCC 35 :BR IF DISPLAY VSVOI IS THERE
JSR PC,SELCTB :IF NOT GO LOOK FOR VTOI DISPLAY
BCC 35 :BR IF AT LEAST THERE IS A VTOI DISPLAY
TYPE MSG3 :GO TYPE 'BUS TIMEOUT ER - VSVOI DISPLAY'
TYPE MSG4 :GO TYPE 'BUS TIMEOUT ER - VTOI DISPLAY'
HALT :NO DISPLAY SEEN AT ASSIGNED BUS ADRES'S
731: MOV NCADR,R0 :GET NC11 BASE ADRES
732: MOV #NCCSR,R1 :GET POINTER ADRES
NCSET: MOV RC,(R1)+ :SET UP NC11 REG ADRES P*RES
ADD #2,R0 :BUMP REG ADRES
CMP #VTVCRG,R1 :ALL SET UP?
BNE NCSET :BR IF NOT
MOV #15,ERRVEC :SET UP TIMEOUT ADRES
TST 2*NCCSR :WILL TRAP TO LOC 4 IF NOT THERE
BR START1 :BR IF THERE
15: CMP (SP)+,(SP)+ :FIX STACK SINCE NO RTI
TYPE MSG5 :GO TYPE 'BUS TIMEOUT ER - NC11'
HALT :NC11 NOT SEEN AT ASSIGNED BUS ADRES
743: MOV #ERRVEC+2,2*ERRVEC :RESTORE ERR TRAP LOC TO PT TO 6
RESET :CLR WORLD
JSR PC,NCSTP1 :GO STOP NC11 & CLR CORE MATRIX
MOV #MATRIX,2*NCOFF :SET OFFSET(ADRES 20000)
CLR GAIN :REGULAR GAIN

```

E02

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 17
 DZNCB.P11 INITIALIZE THE COMMON TAGS

748	001644	012737	020000	001340
749	001652	005037	001312	
750	001656	012737	177777	001314
751	001664	012737	006122	000060
752	001672	012737	000340	000062
753	001700	012737	000100	177236
754	001706	104400	013543	

MOV	#MATRIX.TABLEX	: ISOTOPE A
CLR	TH_L	: CLEAR LOWER THRESHOLD
MOV	#177777,THHI	: CEILING THRESHOLD
MOV	#KBINT,TKVEC	: SET KEYBOARD INTR RETURN ADDR
MOV	#340,TKVEC+2	: SET UP NEW PRIORITY ON INTR
MOV	#100,ISTKS	: SET INTR ENABLE
TYPE	.MSG5	: GO ASK FOR KEYBOARD COMMANDS

:THIS IS A LIST OF KEYBOARD COMMANDS FOR THE NC11 DISPLAY/JOYSTICK

```

:D      DISPLAY DATA
:E      ERASE THE SCOPE
:L      GET LOWER THRESHOLD FROM KEYBOARD & DISPLAY DATA
:U      GET UPPER THRESHOLD FROM KEYBOARD & DISPLAY DATA
:N      COLLECT NEW DATA FROM CAMERA (CLEAR CORE MATRIX)
:C      COLLECT DATA FROM CAMERA
:S      STOP COLLECTION
:G      INITIALIZE EVERYTHING
:Z      ZOOM - SET GAIN TO 1
:R      REGULAR - SET GAIN TO 2
:A      DISPLAY ISOTOPE A
:B      DISPLAY ISOTOPE B
:V      SELECT VTO1 DISPLAY
:W      SELECT VSVO1 DISPLAY (DEFAULT USING 1ST BIT MAP)
:M      SELECT VSVO1 DISPLAY AND USE 2ND BIT MAP
:F      FREE RUN MODE - COLLECT & DISPLAY NEW DATA CONTINUALLY
:J      JOYSTICK CALIBRATION USING AR11
:K      JOYSTICK CALIBRATION USING NC11
:T      DISPLAY INTENSITY TEST IMAGE ON SELECTED DISPLAY
:CNTRL C ABORT WHATEVER - BACK TO KEYBOARD MONITOR
:CNTRL G TURN ON/OFF BELL (NO CONVERT INDICATION) - IN NC11
        :JOYSTICK CALIBRATION ONLY

```

:THIS CODE WILL DISPATCH PROGRAM TO THE PROPER
:ROUTINE VIA THE DISPATCH TABLE 'RTABLE'

```

LISEN: CLR FREERN      :KNOCK DOWN FREE RUN MODE IF SET
        MOV #STACK,SP  :RESET STACK PTR
        CLR KBUF        :INSURE NO KEYBOARD GARBAGE
        CLR PSW         :ALLOW KEYBOARD INTR
LISN:   MOV KBUF,RO     :LOOK FOR CHAR
        BEQ LISN        :WAIT FOR ONE
        CLR KBUF        :
        CMP RO,#101     :WAS IT A CHAR?
        BCS B00B00      :NOT A GOOD CHAR
        BIC #177740,RO  :ELIMINATE LOWER CASE
        CMP RO,#33      :IS IT AN ALPHA CHAR?
        BCC B00B00      :BR IF NOT
        ASL RO          :MAKE UP WORD OFFSET
        TST RTABLE-2(RO) :IS IT A LEGAL COMMAND?
        BEQ B00B00      :BR IF NOT
        JMP RTABLE-2(RO) :GO DO IT

```

:KEYBOARD DISPATCH TABLE

```

RTABLE: ROUTA      : 'A' DISPLAY ISOTOPE A
        ROUTB      : 'B' DISPLAY ISOTOPE B
        ROUTC      : 'C' COLLECT DATA
        ROUTD      : 'D' DISPLAY DATA
        ROUTE      : 'E' ERASE SCOPE
        ROUTF      : 'F' FREE RUN MODE(COLLECT NEW DATA & DISPLAY CONTINUOUSL
        ROUTG      : 'G' INITIALIZE EVERYTHING
        O          : 'H' B00B00
        O          : 'I' B00B00
        ROUTJ      : 'J' GO TO AR11 JOYSTICK CALIBRATION

```

```

755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782 001712 005037 001342
783 001716 012706 001100
784 001722 005037 001306
785 001726 005037 177776
786 001732 013700 001306
787 001736 001775
788 001740 005037 001306
789 001744 020027 000101
790 001750 103445
791 001752 042700 177740
792 001756 020027 000033
793 001762 103040
794 001764 006300
795 001766 005760 001776
796 001772 001434
797 001774 000170 001776
798
799
800
801 002000 002104
802 002002 002116
803 002004 002130
804 002006 002140
805 002010 002144
806 002012 002154
807 002014 002250
808 002016 000000
809 002020 000000
810 002022 002260

```

011	002024	002302	ROUTK	'K'	GO TO NC11 JOYSTICK CALIBRATION
012	002026	002312	ROUTL	'L'	GET LOWER THRESHOLD FROM KEYBOARD
013	002030	002342	ROUTM	'M'	SELECT VSVO1 DISPLAY AND USE 2ND BIT MAP
014	002032	002402	ROUTN	'N'	GET NEW DATA FROM CAMERA
015	002034	000000	O	'O'	BOOB00
015	002036	000000	O	'P'	BOOB00
017	002040	000000	O	'Q'	BOOB00
018	002042	002420	ROUTR	'R'	REGULAR GAIN
019	002044	002436	ROUTS	'S'	STOP COLLECTION
020	002046	002454	ROUTT	'T'	DISPLAY INTENSITY TEST IMAGE ON SELECTED DISPLAY
021	002050	002470	ROUTU	'U'	GET UPPER THRESHOLD FROM KEYBOARD
022	002052	002520	ROUTV	'V'	SELECT VTO1 DISPLAY
023	002054	002540	ROUTW	'W'	SELECT VSVO1 DISPLAY(DEFAULT COND USING 1ST BIT MAP
024	002056	000000	O	'X'	BOOB00
025	002060	000000	O	'Y'	BOOB00
026	002062	002576	ROUTZ	'Z'	ZOOM - SET GAIN TO 1

830	002064	012737	000077	001310	B00B00:	MOV	#77,TTYOUT	:SET UP '?'
831	002072	004737	006174			JSR	PC,TYP0	:TYPE IT
832	002076	004737	006154			JSR	PC,TYPCR	:DO A 'CR'
833	002102	000713				BR	LISN	:GO LOOK FOR GOOD CHAR
834								
835	002104	012737	020000	001340	ROUTA:	MOV	#MATRIX,TABLEX	:WILL DISPLAY ISOTOPE A
836	002112	000137	002616			JMP	CHANGE	:GO DO IT
837	002116	012737	040000	001340	ROUTC:	MOV	#MATRIX+20000,TABLEX	:WILL DISPLAY ISOTOPE B
839	002124	000137	002616			JMP	CHANGE	:GO DO IT
839	002130	004737	005440		ROUTC:	JSR	PC,NCSTRT	:GO START NC11
840	002134	000137	001732			JMP	LISN	:COLLECT DATA UNTIL KEYBRD COMMAND
841	002140	000137	002616		ROUTD:	JMP	CHANGE	:GO DISPLAY DATA
842	002144	004737	005536		ROUTE:	JSR	PC,ERASE	:GO ERASE SCOPE
843	002150	000137	001732			JMP	LISN	:GO WAIT ON NEXT COMMAND
844	002154	012737	177777	001342	ROUTF:	MOV	#-1,FREERN	:SELECT FREE RUN MODE
845	002162	004737	005506			JSR	PC,NCSTP1	:GO STOP NC11 & CLR CORE MATRIX AREA
846	002166	004737	005440			JSR	PC,NCSTRT	:GO START NC11
847	002172	005000				CLR	RD	:SET UP TIMER
849	002174	013701	001344			MOV	TIME,R1	:SET UP GROSS TIMER VALUE
849	002200	005300			1\$:	DEC	RD	:COUNT
850	002202	001376				BNE	1\$	:WAIT FOR ZERO
851	002204	042737	000040	001306		BIC	#BITS,KBUFF	:LOOK FOR POSSIBLE STOP KEY - RID LOWER CASE
852	002212	022737	000123	001306		CMP	#123,KBUFF	:STOP BEEN STRUCK?
853	002220	001007				BNE	2\$	:BR IF NOT
854	002222	042777	000003	176746		BIC	#3,&NCCSR	:STOP NC11
855	002230	005077	176742			CLR	&NCCSR	:ZERO NC CSR
856	002234	000137	001712			JMP	LISEN	:GO AWAIT NEXT COMMAND
857	002240	005301			2\$:	DEC	R1	:DO LOOP AGAIN?
858	002242	001356				BNE	1\$	:BR IF SO
859	002244	000137	002616			JMP	CHANGE	:NOW GO DISPLAY DATA JUST COLLECTED
860	002250	004737	005536		ROUTG:	JSR	PC,ERASE	:GO ERASE SELECTED DISPLAY
861	002254	000137	001616			JMP	START1	:INITIALIZE AND START FRESH
862	002260	004737	005356		ROUTJ:	JSR	PC,SELCTC	:GO SELECT AR11 FOR JOYSTICK CAL
863	002264	103402				BCS	1\$	:BR IF BUS ER FROM AR11
864	002266	000137	004146			JMP	TJOY	:GO TO IT
865	002272	104400	013577		1\$:	TYPE	,MSG7	:GO TYPE 'BUS TIMEOUT ER - AR11'
866	002276	000137	001732			JMP	LISEN	:GO BACK TO KEYBOARD LISEN

867	002302	005037	001346		ROUTK:	CLR	JTYPE	:JTYPE=0 SAYS NC11 FOR JOYSTICK CAL
868	002306	000137	004146			JMP	TJOY	:GO TO IT
869	002312	012737	000340	177776	ROUTL:	MOV	#PR7,PSW	:DON'T WANT ANY INTR'S NOW
870	002320	104400	013331			TYPE	,MSG2	:ASK FOR OCTAL DATA
871	002324	104410				RDOCT		:GO GET IT
872	002326	012637	001312			MOV	(SP)+,THLO	:SAVE IT
873	002332	005037	177776			CLR	PSW	:ENABLE INTR'S
874	002336	000137	002616			JMP	CHANGE	:GO DISPLAY
875	002342	012737	177777	001352	ROUTM:	MOV	#-1,MSELC	:SELECT 2ND BIT MAP
876	002350	004737	005112			JSR	PC,SELCTA	:GO SELECT VSVO1
877	002354	103407				BCS	2\$	:BR IF BUS TIMEOUT ER FROM VSVO1
878	002356	013700	001222			MOV	VTVCSR,RO	:GET 2ND BIT MAP ADRS
879	002362	042760	000400	177760		BIC	#BIT8,-20(RO)	:TURN OFF 1ST BIT MAP
880	002370	000137	001732		1\$:	JMP	LISN	:GO AWAIT NEXT COMMAND
881	002374	104400	013412		2\$:	TYPE	,MSG3	:GO TYPE 'BUS TIMEOUT ER - VSVO1 DISPLAY'
882	002400	000773				BR	1\$	:GO BACK TO KEYBOARD LISEN
883	002402	004737	005506		ROUTN:	JSR	PC,NCSTP1	:GO STOP NC11 & CLR CORE MATRIX
884	002406	013777	001316	176562		MOV	COMSAV,ANCCSR	:RESTART WITH PREVIOUS CSR CONTENTS
885	002414	000137	001732			JMP	LISN	:COLLECT DATA & AWAIT NEXT KEYBRD COMMAND
886	002420	005037	001320		ROUTR:	CLR	GAIN	:WANT REGULAR GAIN
887	002424	042777	000040	176544		BIC	#40,ANCCSR	:INSURE REGULAR GAIN
888	002432	000137	001732			JMP	LISN	:LOOK FOR NEXT COMMAND
889	002436	042777	000003	176532	ROUTS:	BIC	#3,ANCCSR	:STOP NC
890	002444	005077	176526			CLR	ANCCSR	:ZERO NC CSR
891	002450	000137	001732			JMP	LISN	:LOOK FOR NEXT COMMAND
892	002454	004737	005464		ROUTT:	JSR	PC,NCSTP	:GO STOP THE NC11 IF RUNNING
893	002460	004737	005756			JSR	PC,LDIMGE	:GO SET UP TEST CORE IMAGE
894	002464	000137	002616			JMP	CHANGE	:GO DISPLAY IT
895	002470	012737	000340	177776	ROUTU:	MOV	#PR7,PSW	:DON'T WANT ANY INTR'S
896	002476	104400	013331			TYPE	,MSG2	:ASK FOR OCTAL DATA
897	002502	104410				RDOCT		:GO GET IT
898	002504	012637	001314			MOV	(SP)+,THHI	:SAVE IT
899	002510	005037	177776			CLR	PSW	:ENABLE INTR'S
900	002514	000137	002616			JMP	CHANGE	:GO DISPLAY
901	002520	004737	005262		ROUTV:	JSR	PC,SELCTB	:GO SELECT VTO1
902	002524	103402				BCS	2\$	:BR IF TIMEOUT ER FROM VTO1
903	002526	000137	001732		1\$:	JMP	LISN	:GO AWAIT NEXT COMMAND
904	002532	104400	013453		2\$:	TYPE	,MSG4	:GO TYPE 'BUS TIMEOUT ER - VTO1 DISPLAY'
905	002536	000773				BR	1\$	:GO BACK TO KEYBOARD LISEN
906	002540	005037	001352		ROUTW:	CLR	MSELC	:SELECT 1ST BIT MAP
907	002544	004737	005112			JSR	PC,SELCTA	:GO SELECT VSVO1(DEFAULT COND)
908	002550	103407				BCS	2\$	:BR IF TIMOUT ER FROM VSVO1
909	002552	013700	001222			MOV	VTVCSR,RO	:GET 1ST BIT MAP ADRS
910	002556	042760	000400	000020		BIC	#BIT8,20(RO)	:TURN OFF 2ND BIT MAP
911	002564	000137	001732		1\$:	JMP	LISN	:GO AWAIT NEXT COMMAND
912	002570	104400	013412		2\$:	TYPE	,MSG3	:GO TYPE 'BUS TIMEOUT ER - VSVO1 DISPLAY'
913	002574	000773				BR	1\$	:GO BACK TO KEYBOARD LISEN
914	002576	012737	000040	001320	ROUTZ:	MOV	#40,GAIN	:ENABLE GAIN 1
915	002604	053777	001320	176364		BIS	GAIN,ANCCSR	:SET IT AT NC CSR
916	002612	000137	001732			JMP	LISN	:GO AWAIT NEXT COMMAND

```

917      :GO CLEAR SCREEN OF SELECTED DISPLAY
918
919 002616 004737 005536      CHANGE: JSR      PC,ERASE      ;GO ERASE SCOPE
920
921      :STOP NC11 AND GET SET TO DISPLAY
922
923 002622 004737 005464      DISDAT: JSR      PC,NCSTP      ;GO STOP NC11
924
925      :NOW DRAW A BOX AROUND POTENTIAL MATRIX DISPLAY
926      :AREA ON THE SELECTED DISPLAY
927 002626 005737 001254      BOX:  TST      DTYPE      ;WHAT DISPLAY?
928 002632 001067              BNE      BOX1      ;BR IF VTO1
929 002634 012700 001750      MOV      #1750,R0      ;SET UP BIT MAP ADRS - TOP LINE
930 002640 012701 073567      MOV      #73567,R1     ;SET UP PIXEL DATA IN R1 - INT 7
931 002644 004737 005634      JSR      PC,DISPY      ;GO LOAD BIT MAP
932 002650 005200      1$:      INC      R0      ;ADVANCE ADRS
933 002652 022700 001770      CMP      #1770,R0      ;TOP LINE DONE?
934 002656 001403              BEQ      2$      ;BR IF SO
935 002660 004737 005646      JSR      PC,DCONT      ;LOAD NEXT PIXEL WD
936 002664 000771              BR      1$      ;NEXT MAP LOAD
937 002666 012700 006010      2$:      MOV      #6010,R0      ;SET UP BIT MAP ADRS - BOT LINE
938 002672 004737 005634      JSR      PC,DISPY      ;GO LOAD BIT MAP
939 002676 005200      3$:      INC      R0      ;ADVANCE ADRS
940 002700 022700 006030      CMP      #6030,R0      ;BOT LINE DONE?
941 002704 001403              BEQ      4$      ;BR IF SO
942 002706 004737 005646      JSR      PC,DCONT      ;LOAD NEXT PIXEL WD
943 002712 000771              BR      3$      ;NEXT MAP LOAD
944 002714 012700 001730      4$:      MOV      #1730,R0      ;SET UP BIT MAP ADRS SIDE LINES
945 002720 062700 000017      5$:      ADD      #17,R0      ;OFFSET NEXT ROW
946 002724 012701 070000      MOV      #70000,R1     ;SET UP PIXEL 3 DATA IN R1 - INT 7
947 002730 022700 006047      CMP      #6047,R0      ;AT BOTTOM OF SCREEN?
948 002734 001411              BEQ      6$      ;GO START VSVO1 DISPLAY
949 002736 004737 005634      JSR      PC,DISPY      ;GO LOAD BIT MAP
950 002742 012701 000007      MOV      #7,R1      ;SET UP PIXEL 0 DATA IN R1 - INT 7
951 002746 062700 000021      ADD      #21,R0      ;OFFSET TO RIGHT LINE
952 002752 004737 005634      JSR      PC,DISPY      ;GO LOAD BIT MAP
953 002756 000760              BR      5$      ;DO NEXT ROW
954 002760 013777 001262 176234 6$:      MOV      VTVSAV,AVTVCSR ;START DISPLAY
955 002766 012737 002010 001266      MOV      #2010,MAPADR ;OFFSET BIT MAP ADRS
956 002774 012737 000003 001270      MOV      #3,PIXCNT ;SET UP PIXEL BYTE COUNT
957 003002 005037 001272      CLR      PIXASM ;CLR PIXEL ASSEMBLY WORD
958 003006 000137 003132      JMP      LARGES ;GO DISPLAY CELL DATA
959
960 003012 012700 003777      BOX1:  MOV      #3777,R0      ;SET UP X
961 003016 012701 003636      MOV      #3636,R1      ;SET UP Y
962 003022 004737 005676      1$:      JSR      PC,DISPY1 ;DISPLAY POINT
963 003026 005300      DEC      R0      ;MOVE X BY 1
964 003030 100374              BPL      1$      ;AGAIN TILL DONE
965 003032 005200      INC      R0      ;X=0,Y=3636
966 003034 004737 005676      2$:      JSR      PC,DISPY1 ;DISPLAY POINT
967 003040 005301      DEC      R1      ;MOVE Y BY 1
968 003042 022701 000635      CMP      #635,R1 ;AT BOTTOM LEFT?
969 003046 001372              BNE      2$      ;BR IF NOT
970 003050 005201      INC      R1      ;Y=636,X=0
971 003052 004737 005676      3$:      JSR      PC,DISPY1 ;DISPLAY POINT
972 003056 005200      INC      R0      ;MOV X BY 1

```

```

973 003060 022700 004000      CMP      #4000,R0      ;AT BOTTOM RIGHT?
974 003064 001372      BNE      3$      ;BR IF NOT
975 003066 005300      DEC      R0      ;X=3777,Y=636
976 003070 004737 005676      4$: JSR      PC,DISPY1    ;DISPLAY POINT
977 003074 005201      INC      R1      ;MOV Y BY 1
978 003076 022701 003636      CMP      #3636,R1    ;AT TOP RIGHT?
979 003102 001372      BNE      4$      ;BR IF NOT
980 003104 012737 000100 001274  MOV      #100,HORIZ    ;# OF HORIZONTAL POINTS PER ROW
981 003112 012737 000100 001276  MOV      #100,VERT    ;# OF ROWS
982 003120 005037 001300      CLR      XREF      ;START AT LEFT MARGIN
983 003124 012737 003606 001302  MOV      #1926.,YREF    ;AND TOP OF SCREEN
984
985 ;NOW LETS FIND THE LARGEST CELL
986 003132 013737 001322 001324  LARGES: MOV TOTSIZ,CHARCT ;NUMBER OF ELEMENTS TO CONSIDER
987 003140 017700 176174      MOV      @TABLEX,R0    ;THIS IS FIRST GUESS
988 003144 013701 001340      MOV      TABLEX,R1    ;R1 POINTS TO TABLE
989 003150 020021      LARGEL: CMP R0,(R1)+    ;COMPARE GUESS AGAINST NEW
990 003152 103005      BHS      GSG      ;GUESS STILL GOOD
991 003154 024137 001314      CMP      -(R1),THHI    ;GUESS SMALLER BUT CHECK NEW
992 003160 101001      BHI      OVTHHI    ;HI AGAINST UPPER THRESHOLD
993 003162 011100      MOV      (R1),R0    ;WITHIN BOUNDS. MAKE THIS NEW HI
994 003164 005721      OVTHHI: TST (R1)+    ;INCREASE REGISTER BY 2
995 003166 005337 001324      GSG:   DEC      CHARCT    ;COUNT THIS LAST COMPARISON
996 003172 001366      BNE      LARGEL
997
998 ;THE LARGEST CELL WITHIN THE UPPER THRESHOLD IS NOW IN R0
999 ;NOW ACCOUNT FOR LOWER THRESHOLD - BOW OUT IF TOO SMALL
1000 003174 163700 001312      SUB      THLO,R0    ;SUBTRACT LOWER THRESHOLD
1001 003200 101002      BHI      1$      ;BR IF CELL VALUE GREATER THAN LO THRESHOLD
1002 003202 000137 003500      JMP      DISDON    ;DON'T DISPLAY-ALL VALUES BELOW LO THRESHOLD
1003 003206 010037 001334      1$:  MOV      R0,CPERL    ;SAVE LARGEST CELL OF MATRIX
1004
1005 ;NOW PICK OUT EACH VALUE IN CORE MATRIX AND SCALE TO THE
1006 ;PROPER INTENSITY LEVEL. THEN DISPLAY IT ON THE SELECTED DISPLAY
1007
1008 003212 013737 001340 001264  DMATRIX: MOV      TABLEX,MXXADR    ;BEGIN AT TOP LEFT ROW
1009 003220 062737 017600 001264  ADD      #8064.,MXXADR    ;OFFSET TO BOTTOM OF CORE MATRIX
1010 003226 012737 000100 001336  MOV      #64.,ROWCNT    ;THERE ARE 64 CELLS PER ROW
1011 003234 017702 176024      DISLOP: MOV      @MXXADR,R2    ;GET A CELL VALUE FROM MATRIX
1012 003240 062737 000002 001264  ADD      #2,MXXADR    ;BUMP MATRIX ADRS
1013 003246 005337 001336      DEC      ROWCNT    ;COUNT CELL THIS ROW
1014 003252 001006      BNE      CKVALU    ;BR IF ROW NOT FINISHED
1015 003254 012737 000100 001336  MOV      #64.,ROWCNT    ;RESET NEXT ROW COUNT
1016 003262 162737 000400 001264  SUB      #256.,MXXADR    ;SET UP FOR NEXT ROW IN MATRIX
1017 003270 020237 001314      CKVALU: CMP      R2,THHI    ;CELL WITHIN HI THRESHOLD?
1018 003274 101003      BHI      OUTLIM    ;BR IF NOT
1019 003276 163702 001312      SUB      THLO,R2    ;SUB LOW THRESHOLD
1020 003302 101006      BHI      SCLCEL    ;BR IF CELL ABOVE LOW THRESHOLD
1021 003304 005737 001254      OUTLIM: TST      DTYPE    ;WHAT DISPLAY
1022 003310 001052      BNE      CHDN      ;BR IF VTOI
1023 003312 005004      CLR      R4      ;SET CELL TO LOWEST INTENSITY
1024 003314 000137 003504      JMP      MAPLD    ;LOAD BIT MAP
1025 003320 013701 001334      SCLCEL: MOV      CPERL,R1    ;NOW ESTABLISH WHAT INTENSITY LEVEL
1026 003324 012703 000006      MOV      #6,R3      ;SCALE TO 32 LEVELS
1027 003330 005737 001254      TST      DTYPE    ;IS IT A VSV01 DISPLAY?
1028 003334 001001      BNE      1$      ;BR IF NOT

```

K02

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 23
 DZNCB.P11 INITIALIZE THE COMMON TAGS

1029	003336	005303		DEC	R3	; SCALE TO 16 LEVELS ON VSV01 DISPLAY
1030	003340	005004		1\$: CLR	R4	; LEVEL BUILDS IN R4
1031	003342	006304		DVLOOP: ASL	R4	; MUL BY 2
1032	003344	020201		CMP	R2,R1	; COMPARE THIS CELL TO LARGEST (DIVIDED BY 2)
1033	003246	103404		BLO	NODEF	; BR IF SMALLER
1034	003350	005701		TST	R1	; CHECK CASE WHERE 0 DIVISOR
1035	003352	001401		BEQ	1\$	; BR IF 50
1036	003354	005204		INC	R4	; ACCOUNT FOR GOOD SUBTRACTION
1037	003356	160102		1\$: SUB	R1,R2	; NOW DO THE SUBTRACTION
1038	003360	000241		NODEF: CLC		
1039	003362	006001		ROR	R1	; MAKE DIVISOR SMALLER
1040	003364	005303		DEC	R3	; COUNT POSITION
1041	003366	001365		BNE	DVLOOP	; AGAIN IF NOT SCALED TO 16 OR 32 LEVELS YET
1042	003370	160102		SUB	R1,R2	; ACCOUNT FOR REMAINDER
1043	003372	101401		BLOS	GOON	; BR IF TOO SMALL
1044	003374	005204		INC	R4	; ROUND UP
1045	003376	005737	001254	GOON: TST	DTYPE	; WHAT DISPLAY?
1046	003402	001440		BEQ	MAPLD	; IF VSV01 GO LOAD MAP
1047	003404	006304		ASL	R4	; MAKE LEVEL A WORD POINTER
1048	003406	016405	011364	MOV	LEV TAB-2(R4),R5	; POINTER TO CORRECT LEVEL
1049	003412	006204		ASR	R4	; RESTORE ACTUAL LEVEL
1050	003414	001410		BEQ	CHDN	; IF 0 - DON'T DISPLAY
1051						
1052						
1053						; THIS CODE DISPLAYS EACH SCALED CELL ON THE VTO1 (ONE OF 32 LEVELS)
1054	003416	013700	001300	ML00P: MOV	XREF,R0	; GET READY TO DISPLAY
1055	003422	013701	001302	MOV	YREF,R1	; THESE ARE THE CORNER COORD.
1056	003426	004737	006212	JSR	PC,GET	; GET A POINT AND DISPLAY IT
1057	003432	005304		DEC	R4	; DO AS MANY POINTS AS THE LEVEL
1058	003434	001370		BNE	ML00P	
1059	003436	062737	000040 001300	CHDN: ADD	#32.,XREF	; SHIFT TO NEXT POSSIBLE DATUM
1060	003444	005337	001274	DEC	HORIZ	; DONE WITH LINE?
1061	003450	001271		BNE	DISLOP	; NO
1062	003452	012737	000100 001274	MOV	#100,HORIZ	; RESET LINE COUNTER
1063	003460	005037	001300	CLR	XREF	; DO CR
1064	003464	162737	000030 001302	SUB	#24.,YREF	; DO LF
1065	003472	005337	001276	DEC	VERT	; DONE PAGE?
1066	003476	001256		BNE	DISLOP	; NO. GO BACK FOR MORE
1067	003500	000137	003662	DISDON: JMP	PATWRT	; YES. GO DISPLAY PRAMETERS
1068						
1069						; THIS CODE DISPLAYS EACH SCALED CELL ON THE VSV01 (ONE OF 16 LEVELS)
1070						
1071	003504	013700	001266	MAPLD: MOV	MAPADR,R0	; SET UP BIT MAP ADRS
1072	003510	005704		TST	R4	; LOOK FOR NO INTENSITY
1073	003512	001401		BEQ	1\$	; BR IF NO INTENSITY
1074	003514	005304		DEC	R4	; OFFSET TO 0-17
1075	003516	000304		1\$: SWAB	R4	; PREPARE FOR PIXEL LOC
1076	003520	006304		ASL	R4	; MOVE TO TOP 4 BITS
1077	003522	006304		ASL	R4	
1078	003524	006304		ASL	R4	
1079	003526	006304		ASL	R4	
1080	003530	000241		CLC		
1081	003532	006037	001272	ROR	PIXASM	; NOW ASSEMBLE THIS PIXEL INTO PIXEL WORD
1082	003536	006037	001272	ROR	PIXASM	; NOW MAKE ROOM IN PIXEL WORD
1083	003542	006037	001272	ROR	PIXASM	
1084	003546	006037	001272	ROR	PIXASM	

DZNCB.P11 INITIALIZE THE COMMON TAGS

1085	003552	060437	001272		ADD	R4,PIXASM	;ADD THIS PIXEL TO OTHERS
1086	003556	005737	001270		TST	PIXCNT	;ALL 4 PIXELS DONE FOR THIS WORD?
1087	003562	001004			BNE	2\$	;BR IF NOT
1088	003564	013701	001272		MOV	PIXASM,R1	;LD PIXEL WORD INTO R1
1089	003570	004737	005634		JSR	PC,DISPY	;LOAD BIT MAP
1090	003574	005337	001270	2\$:	DEC	PIXCNT	;COUNT PIXEL
1091	003600	100215			SPL	DISLOP	;BR IF 4 PIXELS NOT ASSEMBLED YET
1092	003602	005037	001272		CLR	PIXASM	;CLR PIXEL ASSEMBLY WORD
1093	003606	012737	000003	001270	MOV	#3,PIXCNT	;RESET PIXEL COUNT
1094	003614	005237	001266		INC	MAPADR	;ADVANCE MAP ADRS
1095	003620	013700	001266		MOV	MAPADR,R0	;LOAD INTO R0
1096	003624	022700	005770		CMP	#5770,R0	;HAVE ALL 64 ROWS BEEN DONE?
1097	003630	001002			BNE	3\$	;BR IF NOT
1098	003632	000137	003662		JMP	PATWRT	;NOW GO DISPLAY PARAMETERS
1099	003636	042700	177740	3\$:	BIC	#177740,R0	;SAVE ROW POSITION BITS
1100	003642	022700	000030		CMP	#30,R0	;NOW LOOK FOR END OF ROW
1101	003646	001003			BNE	4\$	;BR IF NOT AT END
1102	003650	062737	000020	001266	ADD	#20,MAPADR	;ADVANCE MAP ADRS TO NEXT ROW
1103	003656	000137	003234	4\$:	JMP	DISLOP	;GO GET NEXT MATRIX DATUM
1104							
1105							;CODE TO WRITE PARAMETER DATA AT BOTTOM OF SCREEN
1106							
1107	003662	005737	001254		PATWRT: TST	DTYPE	;WHAT DISPLAY?
1108	003666	100404			BMI	1\$	;BR IF VTO1
1109	003670	012777	012000	175322	MOV	#12000,AVTVPOS	;VSV01 - SET CHAR POS,LINE 20, LEFT MARGIN
1110	003676	000414			BR	2\$	;START 1ST MSG
1111	003700	012737	000006	001332	1\$:	MOV #6,INCXY	;SET CHARACTER SIZE
1112	003706	012777	000014	175320	MOV	#14,AVTCSR	;SET MODE
1113	003714	012737	000000	001326	MOV	#0,BASX	;SET TO LEFT HAND MARGIN
1114	003722	012737	000600	001330	MOV	#394,BASY	;AND Y LEVEL
1115	003730	004737	006246	2\$:	JSR PC,VTWRIT		
1116	003734	012506			MESS1		; "CR. LOWER THRESHOLD"
1117	003736	013737	001312	006724	MOV THLO,DUMMY1		
1118	003744	004737	006700		JSR PC,AWRIT		
1119	003750	004737	006246		JSR PC,VTWRIT		
1120	003754	012512			MESS2		; "CR. UPPER THRESHOLD"
1121	003756	013737	001314	006724	MOV THHI,DUMMY1		
1122	003764	004737	006700		JSR PC,AWRIT		
1123	003770	004737	006246		JSR PC,VTWRIT		
1124	003774	012516			MESS3		; "Z COUNT ="
1125	003776	017737	175204	006724	MOV ANCZLO,DUMMY1		
1126	004004	017737	175174	006726	MOV ANCZHI,DUMMY2		
1127	004012	004737	006704		JSR PC,OTTY		
1128	004016	004737	006246		JSR PC,VTWRIT		
1129	004022	012522			MESS4		; "MATRIX COUNT="
1130	004024	013737	001322	001246	MOV TOTSIZ,TEMPO		
1131	004032	005002			CLR R2		
1132	004034	005003			CLR R3		
1133	004036	013701	001340		MOV TABLEX,R1		
1134	004042	062103			MXSML: ADD	(R1)+,R3	;NOW ADD UP ALL VALUES IN MATRIX
1135	004044	005502			ADC	R2	
1136	004046	005337	001246		DEC	TEMPO	
1137	004052	001373			BNE	MXSML	
1138	004054	010337	006724		MOV	R3,DUMMY1	
1139	004060	010237	006726		MOV	R2,DUMMY2	
1140	004064	004737	006704		JSR	PC,OTTY	

```

1141 004070 005737 001320      TST GAIN
1142 004074 001403      BEQ 1$
1143 004076 004737 006246      JSR PC,VTWRIT
1144 004102 012526      MESS5      ;"ZOOM"
1145 004104 023727 001340 040000 1$: CMP TABLEX,#MATRIX+20000 ;ISOTOPE B?
1146 004112 001003      BNE 2$      ;NO
1147 004114 004737 006246      JSR PC,VTWRIT      ;YES
1148 004120 012532      MESS6      ;"B-GAMMA"
1149 004122 005737 001342      2$: TST FREERN      ;IN FREE RUN MODE?
1150 004126 001402      BEQ 3$      ;BR IF NOT
1151 004130 000137 002154      JMP ROUTF      ;YES, GO GET NEW DATA
1152 004134 013777 001316 175034 3$: MOV COMSAV,ANCCSR ;RESUME THE NC11
1153 004142 000137 001732      JMP LISN      ;DONE WITH MESSAGES

```

```

;THIS CODE DRAWS A BUG(VT01) OR CROSS HAIRS(VSVO1) WITH THE X & Y DATA
;FROM THE NC11(TYPE 'K' OR DEFAULT) OR THE AR11(TYPE 'J') - THE BUG,
;ETC. WILL FOLLOW THE JOYSTICK WHEN THE INTERRUPT BAR IS NOT DEPRESSED -
;ALL PARTS WITHIN THE DISPLAYED BOX ON THE SELECTED DISPLAY SHOULD
;BE ACCESSIBLE IN A UNIFORM MANNER
;A CNTRL 'C' WILL GET USER BACK TO KEYBOARD MONITOR
↑JOY: JSR PC,ERASE ;START FRESH
      TST DTYPE ;WHAT DISPLAY?
      BNE JBOX1 ;BR IF VT01

```

```

;DRAW A BOX UP ON VSVO1 SCREEN FOR NC11 OR AR11 JOYSTICK CALIBRATION
JBOX: CLR R0 ;SET UP BIT MAP ADRS - TOP LINE
      MOV #73557,R1 ;SET UP PIXEL DATA IN R1 - INT 7
      JSR PC,DISPY ;GO LOAD BIT MAP
      1$: INC R0 ;ADVANCE BIT MAP ADRS
      CMP #40,R0 ;TOP LINE DONE?
      BEQ 2$ ;BR IF SO
      JSR PC,DCONT ;LOAD NEXT PIXEL
      BR 1$ ;NEXT MAP LOAD
      2$: MOV #7740,R0 ;SET UP BIT MAP ADRS - BOT LINE
      JSR PC,DISPY ;GO LOAD BIT MAP
      3$: INC R0 ;ADVANCE ADRS
      CMP #10000,R0 ;BOT LINE DONE?
      BEQ 4$ ;BR IF SO
      JSR PC,DCONT ;LOAD NEXT PIXEL WORD
      BR 3$ ;NEXT MAP LOAD
      4$: MOV #40,R0 ;SET UP BIT MAP ADRS SIDE LINES
      5$: MOV #7,R1 ;SET UP PIXEL 0 DATA IN R1 - INT 7
      JSR PC,DISPY ;GO LOAD BIT MAP
      MOV #70000,R1 ;SET UP PIXEL 3 DATA IN R1 - INT 7
      ADD #37,R0 ;OFFSET TO RIGHT SIDE LINE
      JSR PC,DISPY ;GO LOAD BIT MAP
      INC R0 ;GO TO NEXT ROW ON LEFT
      CMP #7740,R0 ;TO BOTTOM YET?
      BNE 5$ ;BR IF NOT
      MOV VTVSAV,AVTVCSR ;ENABLE BIT MAP
      BR DISBG ;CONTINUE TO ANALOG DATA

```

```

;DRAW BOX UP ON VT01 SCREEN - DIFF SIZE FOR AR11 JOYSTICK CALIBRATION
JBOX1: TST JTYPE ;AR11 ANALOG SOURCE?
      BMI JBOX2 ;BR IF SO

```

```

1197 004314 005000 CLR R0 ;X=0
1198 004316 005001 CLR R1 ;Y=0
1199 004320 004737 005676 1$: JSR PC,DISPY1 ;DISPLAY POINT
1200 004324 005200 INC R0 ;ADVANCE X
1201 004226 022700 003761 CMP #3761,R0 ;OUT TO RIGHT LIMIT?
1202 004332 001372 BNE 1$ ;BR IF NOT
1203 004334 005300 DEC R0 ;X=3760, Y=0
1204 004336 004737 005676 2$: JSR PC,DISPY1 ;DISPLAY POINT
1205 004342 005201 INC R1 ;ADVANCE Y
1206 004344 022701 003761 CMP #3761,R1 ;UP TO TOP YET?
1207 004350 001372 BNE 2$ ;BR IF NOT
1208 004352 005301 DEC R1 ;X=3760, Y=3760
1209 004354 004737 005676 3$: JSR PC,DISPY1 ;DISPLAY POINT
1210 004360 005300 DEC R0 ;ADVANCE X
1211 004362 100374 BPL 3$ ;BR IF NOT TO TOP LEFT
1212 004364 005200 INC R0 ;X=0, Y=3760
1213 004366 004737 005676 4$: JSR PC,DISPY1 ;DISPLAY POINT
1214 004372 005301 DEC R1 ;ADVANCE Y
1215 004374 100374 BPL 4$ ;BR IF NOT TO BOT LEFT
1216 004376 000431 BR DISBG ;CONTINUE TO ANALOG DATA
1217
1218 ;AR11 BOX IS LARGER
1219 004400 005000 JBOX2: CLR R0 ;X=0
1220 004402 005001 CLR R1 ;Y=0
1221 004404 004737 005676 1$: JSR PC,DISPY1 ;DISPLAY POINT
1222 004410 005200 INC R0 ;ADVANCE X
1223 004412 022700 003771 CMP #3771,R0 ;OUT TO RIGHT LIMIT?
1224 004416 001372 BNE 1$ ;BR IF NOT
1225 004420 005300 DEC R0 ;X=3770, Y=0
1226 004422 004737 005676 2$: JSR PC,DISPY1 ;DISPLAY POINT
1227 004426 005201 INC R1 ;ADVANCE Y
1228 004430 022701 003771 CMP #3771,R1 ;UP TO TOP LIMIT?
1229 004434 001372 BNE 2$ ;BR IF NOT
1230 004436 005301 DEC R1 ;X=3770, Y=3770
1231 004440 004737 005676 3$: JSR PC,DISPY1 ;DISPLAY POINT
1232 004444 005300 DEC R0 ;ADVANCE X
1233 004446 100374 BPL 3$ ;BR IF NOT TO TOP LEFT
1234 004450 005200 INC R0 ;X=0, Y=3770
1235 004452 004737 005676 4$: JSR PC,DISPY1 ;DISPLAY POINT
1236 004456 005301 DEC R1 ;ADVANCE Y
1237 004460 100374 BPL 4$ ;BR IF NOT TO BOT LEFT
1238
1239 ;NOW COLLECT ANALOG DATA FROM SELECTED SOURCE (NC11 OR AR11)
1240
1241 004462 005737 001346 DISBG: TST JTYPE ;WHAT SOURCE?
1242 004466 100520 BMI DISBG1 ;BR IF AR11
1243 004470 012777 000032 174500 MOV #32,ANCCSR ;EXT, JOYSTICK & ENA ADC
1244 004476 012777 000010 174506 1$: MOV #10,ANCSFUN ;CLR TIME OUT IF SET
1245 004504 012777 000004 174500 MOV #4,ANCSFUN ;START CONVERTERS
1246 004512 105777 174460 2$: TSTB ANCCSR ;DONE?
1247 004516 100025 BPL 3$ ;BR IF SO
1248 004520 032777 040000 174450 BIT #40000,ANCCSR ;CONVERSION FAIL TO FINISH?
1249 004526 001771 BEQ 2$ ;BR IF NOT
1250 004530 022737 000007 001306 CMP #7,KBUFF ;TURN ON/OFF BELL?
1251 004536 001004 BNE 10$ ;BR IF NOT
1252 004540 005037 001306 CLR KBUFF ;CLR OUT BELL CODE

```

Address	Hex	Label	Assembly	Comment
001354	001354	105:	COM BELLEN	:YES, DO IT
001354	001354		TST BELLEN	:SOUND BELL ON NO CONVERT
000207	001310		BNE 15	:BR IF BELL NOT DESIRED
006174			MOV #207,TTYOUT	:SET UP BELL CODE
			JSR PC,TYPE	:GO ZING BELL
174404	001246	35:	BR 15	:GO TRY AGAIN
100200	001246		MOV @NCAREG,TEMPO	:STORE CONVERSION VALUES
007210			BIC #100200,TEMPO	:MASK OFF BITS 7,15
000010	174372		JSR PC,XYAVE	:GO AVG LAST 32. X-Y JOYSTICK VALUES
174352			MOV #10,@NCSFUN	:CLEAR JOYSTICK DEPRESS FLAG
			TST @NCCSR	:LOOK FOR BUTTON DOWN
001254			BPL 45	:BR IF NOT
			TST DTTYPE	:DON'T MOVE BUG BUT REFRESH
			BNE 55	:REFRESH VTO1 ONLY
001246		45:	BR 15	:NO NEED ON VSV01
001254			JSR TEMPO	:MUL BY 2
			TST DTTYPE	:ALL DONE SCALING IF VSV01
			BNE 65	:BR IF SO
001246			COM TEMPO	:INVERT DATA FOR VTO1
001240	001246		BIC #401,TEMPO	:RID GARBAGE
001246			MOV TEMPO,RC	:SET X
001246			MOV TEMPO+1,R1	:SET Y
			R0	:VTO1 REQUIRES X & Y TO BE INFLATED BY 16
			R0	
			R0	
			R1	
			R1	
			R1	
000020	174320	53:	MOV #20,@VTCR	:SELECT WRITE THRU MODE
005676			JSR PC,DISPY1	:GO DISPLAY THE BUG - VTO1
			BR 15	:DO AGAIN
005726		63:	JSR PC,DISPY2	:GO DISPLAY CROSS HAIRS - VSV01
			BR 15	:DO ANOTHER CONVERSION
				:THIS CODE LOOKS AT THE AR11 FOR ANALOG DATA
004730		DISBG1:		
004730	012737	15:	MOV #20000,ARCHAN	:SELECT CHAN 0 & UNIPOLAR
004736	006024		JSR PC,ARCONV	:GO START A/D - CHAN 0
004742	001246		MOV R3,TEMPO	:SAVE Y
004746	001351		INCB ARCHAN+1	:SELECT CHAN 1
004752	006024		JSR PC,ARCONV	:GO START A/D - CHAN 1
004756	001250		MOV R3,TEMP1	:SAVE X
004762	001351		INCB ARCHAN+1	:SELECT CHAN 2
004766	006024		JSR PC,ARCONV	:GO START A/D - CHAN 2
004772	001252		MOV R3,TEMP2	:READ JOYSTICK BUTTON INDICATOR
004776	001246		SWAB TEMPO	:PUT Y IN HI BYTE
005002	001250	001246	MOVB TEMP1,TEMPO	:AND X IN LO BYTE
005010	007210		JSR PC,XYAVE	:GO AVG LAST 32. X-Y JOYSTICK VALUES
005014	001252		STB TEMP2	:LOOK FOR BAR DOWN (ZERO COND)
005020	001004		BNE 25	:BR IF NOT DEPRESSED
005022	005737	001254	TST DTTYPE	:DON'T MOVE POINT BUT REFRESH
005026	001023		BNE 35	:REFRESH VTO1
005030	000737		BR 15	:NO NEED ON VSV01
005032	005737	001254	TST DTTYPE	:EACH DISPLAY SCALES DIFFERENTLY
005036	001422	25:	BEG 45	:BR IF VSV01
005040	005137	001246	COM TEMPO	:INVERT FOR VTO1

21-SEP-76 15:32 PAGE 28
JANUARY 11 - JANUARY 11 EXERCISER MACY 11 27(732)
JANUARY 11 INITIALIZE THE COMMON TAGS

[illegible]

1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377

005112 013700 001166
005116 012701 001214
005122 010021
005124 062700 000002
005130 022701 001222
005134 001372
005136 013700 001170
005142 005737 001352
005146 001402
005150 062700 000020
005154 010021
005156 062700 000002
005162 022701 001234
005166 001372
005170 012737 005254 000004
005176 005777 174012
005202 005777 174014
005206 013700 001260
005212 010077 174014
005216 062700 000401
005222 032700 010000
005226 001771
005230 005037 001254
005234 012737 000020 001256
005242 000241
005244 012737 000004 000004
005252 000207
005254 022626
005256 000261
005260 000771

.SBTTL PROGRAM SUBROUTINES
:*****
:ROUTINE SELECTS VSV01 DISPLAY - SETS UP BUS ADRS AND INTENSIT. LEVEL
:AND INTENSITY LOOK-UP TABLE - THE CARRY BIT IS SET ON EXIT IF THE
:VSV01 IS NOT SEEN AT THE ASSIGNED BUS ADDRESS
:*****
SELCTA: MOV VTADR,RO ;GET VSV01 BASE ADRS
MOV #VTVCRG,R1 ;GET PTR ADRS
1\$: MOV RO,(R1)+ ;SET UP REG ADRS PTRS
ADD #2,RO ;BUMP REG ADRS
CMP #VTVCSR,R1 ;CHAR GEN REGS ALL SET UP?
BNE 1\$;BR IF NOT
MOV VTMADR,RO ;GET BASE ADRS OF BIT MAP
TST MSELCT ;USING SECOND MAP?
BEQ 2\$;BR IF NOT
ADD #20,RO ;POINT TO 2ND BIT MAP ADRS'S
2\$: MOV RO,(R1)+ ;CONTINUE TO BIT MAP ADRS'S
ADD #2,RO ;BUMP REG ADRS
CMP #VTCSR,R1 ;ALL SET UP?
BNE 2\$;BR IF NOT
MOV #5\$,#ERRVEC ;SET UP BUS TIMEOUT RETURN ADRS IF NO VSV01
TST #VTVCRG ;IS CHARACTER GENERATOR THERE?
TST #VTVCSR ;IS BIT MAP THERE?
3\$: MOV INTLUT,RO ;SET UP ADRS & DATA OF INTENSITY LOOK-UP TABLE
MOV RO,#VTVINT ;SET UP TABLE
ADD #401,RO ;ADVANCE ADRS & INTENSITY
BIT #10000,RO ;TABLE LOADED?
BEQ 3\$;BR IF NOT
CLR DTYPE ;DTYPE=0 SAYS VSV01
MOV #16,INTENS ;MAX 16. INTENSITIES
CLC ;ZERO CARRY SAYS VSV01 THERE
4\$: MOV #ERRVEC,#ERRVEC ;RESTORE ER TRAP LOC TO PT TO 6
RTS PC ;EXIT
5\$: CMP (SP)+,(SP)+ ;FIX STACK SINCE NO RTI
SEC ;CARRY ON EXIT SAYS NO VSV01
BR 4\$;GO EXIT

:*****
:ROUTINE SELECTS VTO1 DISPLAY - SETS UP BUS ADRS AND INTENSITY LEVEL -
:THE CARRY BIT IS SET ON EXIT IF THE VTO1 IS NOT SEEN AT THE
:ASSIGNED BUS ADDRESS
:*****

SELCTB: MOV VTADR,RO ;GET VTO1 BASE ADRS
MOV #VTCSR,R1 ;GET PTR ADRS
1\$: MOV RO,(R1)+ ;SET UP VTO1 REG ADRS PTRS
ADD #2,RO ;BUMP REG ADRS
CMP #ARCSR,R1 ;ALL SET UP?
BNE 1\$;BR IF NOT
MOV #3\$,#ERRVEC ;SET UP FOR VTO1 TIMEOUT
MOV #14,#VTCSR ;SELECT STORE MODE
MOV #32,INTENS ;MAX 32 INTENSITIES
MOV #-1,DTYPE ;DTYPE=-1 SAYS VTO1
CLC ;ZERO CARRY SAYS VTO1 THERE
2\$: MOV #ERRVEC+2,#ERRVEC ;RESTORE LOC 4 PTR TO PT TO 6
RTS PC ;EXIT

005262 013700 001172
005266 012701 001234
005272 010021
005274 062700 000002
005300 022701 001242
005304 001372
005306 012737 005350 000004
005314 012777 000014 173712
005322 012737 000040 001256
005330 012737 177777 001254
005336 000241
005340 012737 000006 000004
005346 000207

E03

 MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 30
 DZNCB.P11 PROGRAM SUBROUTINES

```

1379 005350 022526      3$:    CMP      (SP)+,(SP)+      ;FIX STACK SINCE NO RETURN
1379 005352 000261      SEC                                ;CARRY SET SAYS NO VTO1 DISPLAY
1380 005354 000771      BR        2$                      ;GO EXIT
1381
1382 ::*****
1383 ;ROUTINE SELECTS AND SETS UP BUS ADRS FOR AR11 JOYSTICK CALIBRATION
1384 ;THE CARRY SET ON EXIT IF THE AR11 IS NOT SEEN
1385 ::*****
1386 005356 013700 001174  SELECTC: MOV      ARADR,RO      ;GET AR11 BASE ADRS
1387 005362 012701 001242      MOV      #ARCSR,R1      ;GET PTR ADRS
1388 005366 010021      1$:    MOV      RO,(R1)+      ;SET UP REG ADRS PTRS
1389 005370 062700 000002      ADD      #2,RO      ;BUMP REG ADRS
1390 005374 022701 001246      CMP      #TEMPO,R1      ;ALL SET UP?
1391 005400 001372      BNE        1$      ;BR IF NOT
1392 005402 012737 005434 000004      MOV      #3$,2*ERRVEC      ;SET UP TIMEOUT RETURN ADRS IF AR11 NOT THERE
1393 005410 005077 173626      CLR      2ARCSR      ;SEE IF THERE
1394 005414 012737 177777 001346      MOV      #-1,JTYPE      ;JTYPE=-1 SAYS USE AR11 FOR JOYSTICK CAL
1395 005422 000241      CLC                                ;CARRY ZERO SAYS AR11 IS THERE
1396 005424 012737 000006 000004 2$:    MOV      #ERRVEC+2,2*ERRVEC      ;RESTORE LOC 4 TO PT TO 6
1397 005432 000207      RTS      PC      ;EXIT
1398 005434 000261      3$:    SEC                                ;CARRY SET SAYS THAT AR11 IS NOT ADRS ASSIGNED
1399 005436 000772      BR        2$                      ;GO EXIT
1400
1401 ::*****
1402 ;ROUTINE STARTS NC11 AT SELECTED GAIN
1403 ::*****
1404 005440 012777 001001 173530  NCSTRT: MOV      #1001,2NCCSR      ;SET UP 64*64 MATRIX AND ADC ENABLE
1405 005446 053777 001320 173522      BIS      GAIN,2NCCSR      ;SET UP GAIN
1406 005454 052777 000002 173514      BIS      #BIT1,2NCCSR      ;SET GO BIT
1407 005452 000207      RTS      PC      ;RETURN
1408
1409 ::*****
1410 ;ROUTINE STOPS NC11 AND SAVES NC11 STATUS
1411 ::*****
1412 005464 017737 173506 001316  NCSTP: MOV      2NCCSR,COMSAV      ;SAVE THE INTERFACE ACTION
1413 005472 042777 000003 173476      BIC      #3,2NCCSR      ;DISABLE NPP'S
1414 005500 000777 173472      CLR      2NCCSR      ;ZERO ALL STATUS
1415 005504 000207      RTS      PC      ;RETURN
1416
1417 ::*****
1418 ;ROUTINE STOPS NC11, SAVES STATUS AND CLEARS MATRIX CORE AREA
1419 ::*****
1420 005506 004737 005464      NCSTP1: JSR      PC,NCSTP      ;GO STOP NC11 AND SAVE STATUS
1421 005512 012777 000020 173472      MOV      #CLZ,2NCSFUN      ;ZERO Z REG COUNT
1422 005520 012700 020000      MOV      #MATRIX,RO      ;GET SET TO ZERO CORE MATRIX AREA
1423 005524 005020      1$:    CLR      (RO)+      ;ZERO LOC
1424 005526 020027 060000      CMP      RO,#MATRIX+40000      ;ALL DONE?
1425 005532 001374      BNE        1$      ;BR IF MORE
1426 005534 000207      RTS      PC      ;RETURN
1427
1428 ::*****
1429 ;ROUTINE WILL ERASE DISPLAY (VSV01 OR VTO1)
1430 ::*****
1431 005536 005737 001254      ERASE:  TST      DTYPE      ;WHAT DISPLAY?
1432 005542 100017      BPL        2$      ;BR IF VSV01
1433 005544 012777 000016 173462      MOV      #16,2VTOCSH      ;ERASE DISPLAY

```

```

1434 005552 005037 001246          CLR      TEMPO          ;SET UP VTO1 ERASE WAIT LOOPS
1435 005556 012737 000002 001250  MOV      #2,TEMP1      ;DO LOOP 2 TIMES
1436 005564 005337 001246          DEC      TEMPO          ;COUNT AWAY
1437 005570 001375          BNE      1$          ;TILL 0
1438 005572 005337 001250          DEC      TEMP1          ;2ND LOOP DONE?
1439 005576 001372          BNE      1$          ;BR IF NOT
1440 005600 000414          BR       4$          ;ALL DONE - RETURN
1441 005602 052777 001000 173412 2$: BIS      #1000,2VTVCSR      ;ERASE DISPLAY (CLR BIT MAP)
1442 005610 105777 173406 3$: TSTB     2VTVCSR      ;LOOK FOR READY
1443 005614 100375          BPL      3$          ;WAIT FOR IT
1444 005616 042777 001000 173376  BIC      #1000,2VTVCSR      ;TURN OFF ERASE DISPLAY
1445 005624 012777 002035 173362  MOV      #2035,2VTVCRG      ;CLR CHAR BUFFER & DISABLE CURSOR
1446 005632 000207 4$: RTS      PC          ;EXIT
1447
1448 ;*****
1449 ;ROUTINE WILL LOAD BIT MAP (VSVO1)
1450 ;RO CONTAINS BIT MAP ADRS AND R1 THE BIT MAP DATA
1451 ;*****
1452 005634 042777 000400 173360 DISPY: BIC      #400,2VTVCSR      ;STOP DISPLAY
1453 005642 010077 173356          MOV      RO,2VTVMAP      ;LOAD BIT MAP ADRS
1454 005646 042777 000400 173346 DCONT: BIC      #400,2VTVCSR      ;AGAIN IF ENTERING HERE
1455 005654 105777 173342 1$: TSTB     2VTVCSR      ;READY?
1456 005660 100375          BPL      1$          ;WAIT THEN
1457 005662 010177 173340          MOV      R1,2VTVPX      ;LOAD BIT MAP
1458 005666 052777 000400 173326  BIS      #400,2VTVCSR      ;RESUME DISPLAY
1459 005674 000207          RTS      PC          ;RETURN
1460
1461 ;*****
1462 ;ROUTINE WILL DISPLAY A POINT (VTO1)
1463 ;*****
1464 005676 105777 173332 DISPY1: TSTB     2VTVCSR      ;DISPLAY READY?
1465 005702 100375          BPL      DISPY1      ;WAIT FOR IT
1466 005704 042700 174000          BIC      #174000,RO      ;RID GARBAGE
1467 005710 042701 174000          BIC      #174000,R1      ;RID GARBAGE
1468 005714 010077 173316          MOV      RO,2VTXDAC      ;SET UP X DAC
1469 005720 010177 173314          MOV      R1,2VTYDAC      ;SET UP Y DAC
1470 005724 000207          RTS      PC          ;EXIT
1471
1472 ;*****
1473 ;ROUTINE WILL DISPLAY X & Y CROSS HAIRS ON VSVO1
1474 ;*****
1475 005726 005777 173262 DISPY2: TST      2VTVCRG      ;READY?
1476 005732 100375          BPL      DISPY2      ;WAIT IF NOT
1477 005734 105137 001246          COMB     TEMPO          ;X NEEDS TO BE INVERTED
1478 005740 013777 001246 173250  MOV      TEMPO,2VTVCHP      ;LOAD X & Y CROSS HAIRS
1479 005746 052777 016000 173240  BIS      #16000,2VTVCRG      ;ENABLE THE CROSS HAIRS
1480 005754 000207          RTS      PC          ;RETURN
1481
1482 ;*****
1483 ;ROUTINE WILL FILL CORE WITH AN IMAGE THAT WHEN
1484 ;DISPLAYED WILL CONTAIN ALL THE INTENSITY LEVELS -
1485 ;ROWS AT THE BOTTOM OF THE SCREEN WILL APPEAR BRIGHTEST -
1486 ;NOTE THAT THIS IS ONLY A DISPLAY TEST PATTERN FOR
1487 ;POSSIBLE DISPLAY ADJUSTMENTS BY THE USER
1488 ;*****
1489 005756 012700 000100 LDIMGE: MOV      #100,RO      ;COUNT 64 ROWS

```

G03

 MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MAY11 27(732) 21-SEP-76 15:32 PAGE 32
 DZNCB.P11 PROGRAM SUBROUTINES

```

1490 005762 013701 001340      MOV    TABLEX,R1      ;GET SELECTED ISOTOPE
1491 005766 012702 000100      MOV    #100,R2      ;COUNT 64 DATA POINTS PER ROW
1492 005772 012703 000100      MOV    #100,R3      ;100 WILL REPRESENT HIGHEST INTENSITY LEVEL(100-0)
1493 005776 010321      1$: MOV    R3,(R1)+      ;LOAD CORE IMAGE
1494 006000 005302      DEC    R2      ;DONE ROW?
1495 006002 001375      BNE    1$      ;BR IF NOT
1496 006004 005300      DEC    R0      ;DONE ROWS?
1497 006006 001405      BEQ    2$      ;BR IF SO
1498 006010 162703 000001      SUB    #1,R3      ;LOWER NEXT CELL VALUE
1499 006014 012702 000100      MOV    #100,R2      ;RESET ROW LENGTH COUNTER
1500 006020 000766      BR     1$      ;LOAD THIS ROW IMAGE
1501 006022 000207      2$: RTS    PC      ;RETURN FOR DISPLAY

1502
1503      ;*****
1504      ;ROUTINE WILL DO A CONVERSION AT CHAN # IN 'ARCHAN'
1505      ;EXIT WITH CONVERSION VALUE IN R3
1506      ;*****
1507 006024 005777 173214      ARCONV: TST    @ARBUF      ;CLR DONE FLAG
1508 006030 013777 001350      MOV    ARCHAN,@ARCSR      ;LD CHAN & UNIPOLAR BITS
1509 006036 052777 000001      BIS    #1,@ARCSR      ;START A/D
1510 006044 105777 173172      1$: TSTB   @ARCSR      ;LOOK FOR DONE
1511 006050 100375      BPL     1$      ;WAIT FOR IT
1512 006052 017703 173166      MOV    @ARBUF,R3      ;GET VALUE
1513 006056 162703 000500      SUB    #500,R3      ;OFFSET VALUE
1514 006062 032703 177400      BIT     #177400,R3      ;SEE IF OUT OF RANGE
1515 006066 100402      BMI     2$      ;BR IF TOO SMALL
1516 006070 001003      BNE     3$      ;BR IF TOO LARGE
1517 006072 000404      BR     4$      ;NORMAL EXIT
1518 006074 005003      2$: CLR    R3      ;LIMIT TO 0
1519 006076 000402      BR     4$      ;GO EXIT
1520 006100 012703 000377      3$: MOV    #377,R3      ;LIMIT TO 377
1521 006104 000207      4$: RTS    PC      ;EXIT

1522
1523      ;*****
1524      ;ROUTINE WILL DISPLAY A CHARACTER (VSVO1)
1525      ;*****
1526 006106 005777 173102      DCHAR: TST    @VTVCRG      ;READY?
1527 006112 100375      BPL     DCHAR      ;WAIT FOR IT
1528 006114 110377 173074      MOVB   R3,@VTVCRG      ;LOAD CHAR
1529 006120 000207      RTS     PC      ;EXIT

1530
1531      ;*****
1532      ;KEYBOARD INTERRUPT SERVICE ROUTINE
1533      ;*****
1534 006122 017737 173020 001306      KBINT: MOV    @STKB,KBUFF      ;READ KEY BOARD
1535 006130 042737 000200 001306      BIC     #200,KBUFF      ;RID PARITY
1536 006136 022737 000003 001306      CMP     #3,KBUFF      ;CNTRL 'C'?
1537 006144 001002      BNE     1$      ;BR IF NOT
1538 006146 000137 001712      JMP     LISEN      ;ABORT WHATEVER & LOOK FOR NEXT COMMAND
1539 006152 000002      1$: RTI

1540
1541      ;*****
1542      ;ROUTINE TYPES 'CR' AND 'LF' OR CHAR IN TTYOUT
1543      ;*****
1544 006154 012737 000015 001310      TYPCR: MOV    #15,TTYOUT      ;SET UP FOR A 'CR'
1545 006162 004737 006174      JSR    PC,TYPO

```

```

1546 006166 012737 000012 001310      MOV #12,TTYOUT      ;SET UP FOR LF
1547 006174 105777 172750      TSTB @STPB      ;WAIT FOR LAST CHARACTER
1548 006200 100375      BPL TYP0
1549 006202 013777 001310 172742      MOV TTYOUT,@STPB      ;SEND IT OUT
1550 006210 000207      RTS PC
1551
1552      ;*****
1553      ;GET IS A SUBROUTINE TO GET AN X,Y COORDINATE FROM THE
1554      ;CODED LEVEL MATRIX,DISPLAY IT AND EXIT.
1555      ;*****
1556 006212 111502      GET:  MOVB (R5),R2      ;MOVE A BYTE OF X AND Y
1557 006214 042702 177707      BIC #177707,R2      ;MASK ALL BUT X
1558 006220 006202      ASR R2      ;ABS VALUE IS 4*X
1559 006222 060200      ADD R2,R0      ;ADD TO CORNER VALUE
1560 006224 112502      MOVB (R5)+,R2      ;MOV SAME BYTE AND POKE POINTER
1561 006226 042702 177770      BIC #177770,R2      ;MASK ALL BUT Y
1562 006232 060201      ADD R2,R1      ;YABS Y INC IS 3
1563 006234 006302      ASL R2      ;SO DO Y=Y*3
1564 006236 060201      ADD R2,R1      ;ADD TO CORNER VALUE
1565 006240 004737 005676      JSR PC,DISPY1      ;GO DISPLAY POINT
1566 006244 000207      RTS PC      ;RETURN TO MAIN
1567
1568      ;*****
1569      ;THIS SUBROUTINE IS CALLED WITH THE ADDRESS OF A
1570      ;MESSAGE TABLE FOLLOWING THE CALL. THE MESSAGE
1571      ;TABLE CONTAINS A LIST OF POINTERS TO PHRASES.
1572      ;EACH PHRASE IS TERMINATED WITH A 200 BYTE. THE TABLE
1573      ;IS TERMINATED WITH A 0.
1574      ;*****
1575
1576 006246 017605 000000      VTWRIT: MOV @ (SP),R5      ;GET ADDRESS OF MESSAGE IN R5
1577 006252 062716 000002      ADD #2,(SP)      ;ADJUST RETURN
1578 006256 005715      VTPL:  TST (R5)      ;LOOK FOR ZERO AS TERMINATOR
1579 006260 001476      BEQ VTWDON      ;BR IF MSG DONE
1580 006262 012504      MOV (R5)+,R4      ;ADDRESS OF PHRASE TO R4
1581 006264 112403      VTWL:  MOVB (R4)+,R3      ;THIS IS LETTER TO BE DISPLAYED
1582 006266 100773      BMI VTPL      ;200 IS THE TERMINATOR
1583 006270 005737 001254      TST DTYPE      ;WHAT DISPLAY?
1584 006274 001003      BNE DVT01      ;BR IF VTO1
1585 006276 004737 006106      JSR PC,DCHAR      ;GO DISPLAY CHAR VSV01
1586 006302 000770      BR VTWL      ;GO LOOK AT NEXT CHAR
1587 006304 022703 000015      DVT01: CMP #15,R3      ;CR?
1588 006310 001004      BNE 1$      ;BR IF NOT
1589 006312 012737 000000 001326      MOV #0,BASX      ;RESET MARGIN
1590 006320 000761      BR VTWL      ;GO GET NEXT CHAR
1591 006322 022703 000012      1$:  CMP #12,R3      ;LF?
1592 006326 001004      BNE VTEXCP      ;BR IF NOT
1593 006330 162737 000060 001330      SUB #60,BASY      ;GO TO NEXT LINE
1594 006336 000752      BR VTWL      ;GO GET NEXT CHAR
1595 006340 110337 001304      VTEXCP: MOV R3,KBDDBUF      ;THERE ARE A FEW CHARACTERS
1596 006344 012737 000034 006460      MOV #34,TEMCHR      ;THAT REQUIRE CONVERSION
1597 006352 004737 006462      JSR PC,BRAN      ;BEFORE DISPLAY.
1598 006356 006373      VTEXCT-1
1599 006360 006430      CH74      ;CHARACTER IS <

```

```

1600 006362 006422          CH75          ;=
1601 006364 006416          CH72          ;:
1602 006366 006410          CH76          ;>
1603 006370 006402          CH77          ;?
1604 006372 000420          BR VTNOSP      ;CHARACTER IS OK AS IS
1605 006374      074      075      072  VTEXCT: .BYTE 74,75,72,76,77,0
1606 006377      076      077      000
1607
1608 006402 062737 000004 006460 CH77: ADD #4,TEMCHR      ;CONVERT OCTAL TO 42
1609 006410 062737 000007 006460 CH76: ADD #7,TEMCHR      ;CONVERT TO 46
1610 006416 005237 006460          CH72: INC TEMCHR        ;CONVERT TO 37
1611 006422 062737 000002 006460 CH75: ADD #2,TEMCHR      ;CONVERT TO 36
1612 006430 013703 006460          CH74: MOV TEMCHR,R3      ;R3 NOW CONTAINS PROPER VALUE
1613 006434 042703 177700          VTNOSP: BIC #177700,R3    ;CLEAR GARBAGE
1614 006440 006303          ASL R3          ;MULTIPLY 6 BIT OCTAL BY 4
1615 006442 006303          ASL R3          ;TO FIND RELATIVE POSITION IN TABLE
1616 006444 062703 012652          ADD #TABLE-4,R3          ;THIS IS ABSOLUTE POSITION
1617 006450 004737 006532          JSR PC,TIXDIS            ;DRAW CHARACTER
1618 006454 000703          BR VTWL          ;AND CONSIDER NEXT
1619 006456 000207          VTWDON: RTS PC
1620 006460 000000          TEMCHR: 0          ;SPECIAL CHARS ARE BUILT HERE
1621
1622 ;*****
1623 ;THIS ROUTINE COMPARES THE CHARACTER IN KDBUF (BYTE)
1624 ;AGAINST A LIST POINTED TO BY CONTENTS+1 OF CALL+2.
1625 ;IF A MATCH IS FOUND, CONTROL IS TRANSFERED TO ADDRESS
1626 ;CONTAINED IN CALL +2+2N WHERE N IS THE NUMBER OF
1627 ;THE ENTRY IN THE MATCH TABLE THAT MATCHED.
1628 ;IF NO MATCH IS FOUND, CONTROL IS TRANSFERED TO THE LOCATION FOLLOWING
1629 ;THE LAST TRANSFER ADDRESS.
1630 ;A ZERO IS THE MATCH TABLE TERMINATOR.
1631 ;THE N AND C BITS ARE CLEARED ON EXIT
1632 ;*****
1633 006462 017637 000000 006530 BRAN: MOV @ (SP),MACHERR      ;GET ADDRESS OF MATCH LIST
1634 006470 062716 000002          BRANL: ADD #2,(SP)        ;ADJUST RETURN POINTER
1635 006474 005237 006530          INC MACHERR              ;MOVE MATCH POINTER
1636 006500 105777 000024          TSTB @MACHERR            ;A ZERO INDICATES END OF LIST
1637 006504 001406          BEQ NOMACH
1638 006506 123777 001304 000014 CMPB KDBUF,@MACHERR
1639 006514 001365          BNE BRANL
1640 006516 017616 000000          MOV @ (SP),(SP)          ;NO MATCH. TRY AGAIN
1641 006522 000241          NOMACH: CLC                      ;PUT TRANSFER ADDRESS ON STACK
1642 006524 000250          CLN                          ;CLEAR CARRY BIT
1643 006526 000207          RTS PC                          ;AND N BIT
1644 006530 000000          MACHERR: 0                      ;RETURN APPROPRIATELY
1645
1646 ;*****
1647 ;THIS SUBROUTINE IS USED TO DISPLAY A MATRIX OF POINTS
1648 ;THE ROUTINE IS GENERALIZED AND MUST HAVE VARIOUS
1649 ;CONSTANTS SETUP BY THE CALLING PROGRAMS.
1650 ;R3 POINTS TO THE MATRIX CODE WORDS
1651 ;*****
1652
1653
1654 006532 013737 006660 006662 TIXDIS: MOV SHFTK,SHFTCT      ;SET SHIFT COUNT FOR DECODER
1655 006540 013737 006664 006666      MOV ROWK,ROW          ;SETUP THE NUMBER OF ROWS

```

```

1656 006546 013737 006670 006672      MOV COLK,COL      ;AND THE NUMBER OF COLUMNS
1657 006554 012302      MOV (R3)+,R2      ;GET THE WORD TO BE DECODED
1659 006556 013700 001326      MOV BASX,R0      ;R0 WILL CONTAIN THE X COORD.
1659 006562 013701 001330      MOV BASY,R1      ;R1 NOW CONTAINS THE Y COORDINATE
1660 006566 006002      TIXL: ROR R2      ;CHECK CODE WORD FOR BIT
1661 006570 103002      BCC TIXBNO      ;IF CARRY BIT CLEAR NO POINT TO DISPLAY
1662 006572 004737 005676      JSR PC,DISPY1      ;GO DISPLAY POINT
1663 006576 005337 006662      TIXBNO: DEC SHFTCT      ;COUNT INFO BITS IN WORD
1664 006602 001001      BNE TIXWOK      ;CODE WORD OK IF NON ZERO RESULT
1665 006604 011302      MOV (R3),R2      ;GET NEW CODE WORD
1666 006606 063700 001332      TIXWOK: ADD INCXY,R0      ;REFERENCE NEXT POINT
1667 006612 005337 006666      DEC ROW      ;DONT GO PAST END OF ROW
1668 006616 001363      BNE TIXL      ;NON ZERO OK
1669 006620 013737 006664 006666      MOV ROWK,ROW      ;RESET
1670 006626 063701 001332      ADD INCXY,R1      ;MOV Y UP A ROW
1671 006632 005337 006672      DEC COL      ;WHEN COL=0 WE ARE DONE
1672 006636 001403      BEQ TIXDON      ;RESET X TO LEFT HAND EDGE
1673 006640 013700 001326      MOV BASX,R0
1674 006644 000750      BR TIXL
1675 006646 063700 001332      TIXDON: ADD INCXY,R0      ;MAKE A SPACE BETWEEN LETTERS
1676 006652 010037 001326      MOV RC,BASX      ;AND RESET BASX TO NEW VALUE
1677 006656 000207      RTS PC
1678 006660 000017      SHFTK: 15.
1679 006662 000000      SHFTCT: 0
1680 006664 000005      ROWK: 5
1681 006666 000000      ROW: 0
1682 006670 000006      COLK: 6
1683 006672 000000      COL: 0
1684
1685 ;*****
1686 ;THIS ROUTINE CALLS 'ASCIZP WHICH CONVERTS A SINGLE OR DOUBLE
1687 ;PRECISION OCTAL NUMBER TO ASCII DECIMAL AND THEN WRITES IT TO SCREEN
1688 ;*****
1689 006674 007105      BLANK: ADUMMY
1690 006676 000000      0
1691 006700 005037 006726      AWRIT: CLR DUMMY2
1692 006704 004737 006732      OTTY: JSR PC,ASCZSP
1693 006710 006724      DUMMY1
1694 006712 007105      ADUMMY
1695 006714 004737 006246      JSR PC,VTWRIT
1696 006720 006674      BLANK
1697 006722 000207      RTS PC
1698
1699 ;*****
1700 ;THIS ROUTINE CONVERTS A DOUBLE PRECISION OCTAL NUMBER IN CORE
1701 ;TO AN ASCII DECIMAL NUMBER.
1702 ;THE CALL IS JSR PC,ASCIZ
1703 ;WITH THE ADDRESS OF THE LOW ORDER OCTAL WORD IN
1704 ;CALL+2
1705 ;AND THE ADDRESS OF THE PLACE THE ASCII DECIMAL IS TO BE STORED
1706 ;IN CALL+4
1707 ;THE HIGH ORDER OCTAL WORD MUST BE IN THE LOCATION FOLLOWING
1708 ;THE LOW ORDER WORD.
1709 ;THE SUBROUTINE SHOULD BE CALLED BY
1710 ;JSR PC,ASCZSP
1711 ;TO SUPPRESS LEADING ZEROES AND TO LEFT JUSTIFY THE

```

```

1712      :ANSWER. UNLESS THIS LAST CALL IS MADE THE OUTPUT
1713      :WILL ALWAYS BE 11 BYTES.
1714      :*****
1715      006724 000000      DUMMY1: 0
1716      006726 000000      DUMMY2: 0
1717      006730 000000      ZESUP: 0      ;ZERO SUPPRESS FLAG
1718      006732 005237 006730 ASCZSP: INC ZESUP      ;SET FLAG
1719      006736 017601 000000 ASCIZ: MOV 2(SP),R1      ;GET LOW ORDER ADDRESS
1720      006742 062716 000002      ADD #2,(SP)      ;ADJUST POINTER
1721      006746 012103      MOV (R1)+,R3      ;GET LOW ORDER
1722      006750 011102      MOV (R1),R2      ;GET HIGH ORDER
1723      006752 012737 000012 007070 MOV #10,DIVCNT      ;10 POSSIBLE DIGITS
1724      006760 012705 007072      MOV #ASCRES,R5      ;ADDRESS OF DIGITS
1725      006764 112725 000200      MOV #200,(R5)+      ;CHGEN TERMINATOR
1726      006770 012704 000012      ASCLP: MOV #10,R4      ;DIVISOR
1727      006774 010546      MOV R5,-(SP)      ;SAVE R5 DURING DIVISION
1728      006776 004737 007120      JSR PC,DIVIDE      ;DIVIDE BY 10
1729      007002 012605      MOV (SP)+,R5      ;RESTORE R5
1730      007004 062702 000060      ADD #60,R2      ;MAKE IT ASCII
1731      007010 110225      MOV R2,(R5)+      ;PUT IT IN THE ASCII STRING
1732      007012 010103      MOV R1,R3      ;PUT ANSWER IN DIVIDEND SLOT
1733      007014 010002      MOV R0,R2
1734      007016 005337 007070      DEC DIVCNT
1735      007022 001362      BNE ASCLP
1736      :NOW TRANSFER THE ASCII TO PROPER PLACE IN CORRECT ORDER
1737      007024 005737 006730      TST ZESUP      ;SUPPRESS ZEROES?
1738      007030 001410      BEQ MOVAL      ;NO MOVE ALL
1739      007032 005037 006730      CLR ZESUP      ;INIT THE SWITCH
1740      007036 124527 000060      SKZE: CMPB -(R5),#60      ;LOOK FOR ZEROES
1741      007042 001775      BEQ SKZE      ;AND PASS BY THEM
1742      007044 105725      TSTB (R5)+      ;BACK-UP
1743      007046 100001      BPL MOVAL      ;AT LEAST ONE ZERO
1744      007050 005205      INC R5      ;POINT TO FIRST NUMBER
1745      007052 017600 000000      MOVAL: MOV 2(SP),R0      ;ADDRESS OF RESULTS
1746      007056 062716 000002      ADD #2,(SP)      ;ADJUST RETURN
1747      007062 114520      MOVAL: MOVB -(R5),(R0)+      ;200 BYTE IS TERMINATOR
1748      007064 100376      BPL MOVAL
1749      007066 000207      RTS PC
1750      007070 000000      DIVCNT: 0
1751      007072 007105      ASCRES: .=.+11.
1752      007105 007120      ADUMMY: .=.+11.
1753      :EVEN
1754      :*****
1755      :THIS ROUTINE DIVIDES A POSITIVE DOUBLE PRECISION NUMBER
1756      :FOUND IN R2 (HI) AND R3 (LO) BY THE POSITIVE NUMBER IN
1757      :R4. THE ANSWER IS PLACED IN R0 (HI) AND R1 (LO).
1758      :THE ROUTINE LEFT JUSTIFIES DIVISOR KEEPING TRACK OF SHIFTS
1759      :SO THAT FIRST BIT IS IN BIT 14 OF R4. THE SHIFT COUNT IS THEN
1760      :ADDED TO 16. AND THE DIVISION IS STARTED BY COMPARING THE
1761      :DIVISOR (SHIFTED) TO THE HI ORDER OF THE DIVIDEND.
1762      :IF COMPARISON SHOWS DIVISOR SMALLER, A SUBTRACTION IS DONE
1763      :BETWEEN DIVSOR AND HI ORDER DIVIDEND AND THE ANSWER IS INCREMENTED.
1764      :REGARDLESS OF COMPARISON RESULT, BOTH THE ANSWER AND
1765      :THE DIVIDEND ARE MULTIPLIED BY 2.
1766      :THIS SEQUENCE IS PERFORMED THE NUMBER OF TIMES INDICATED BY
1767

```

```

1768      ;SHIFTCOUNT.
1769      ;R2R3 / R4=ROR1
1770      ;*****
1771      007120 005005      DIVIDE: CLR R5      ;INIT SHIFT COUNT
1772      007122 005000      CLR R0
1773      007124 005001      CLR R1      ;ZERO THE ANSWER
1774      007126 005205      DIVJUS: INC R5      ;THIS IS THE SHIFT COUNT
1775      007130 006304      ASL R4      ;DIVSOR X 2
1776      007132 100375      BPL DIVJUS      ;GO UNTIL BIT 15 SETS
1777      007134 006004      ROR R4      ;MOV IT BACK ONE
1778      007136 010537 007206      MOV R5,RMJUST      ;SAVE SIFT COUNT FOR REMAN. JUST
1779      007142 062705 000020      ADD #16.,R5      ;SIMULATE 16 BIT SHIFT
1780      007146 006301      DIVL00: ASL R1      ;ANSWER X 2
1781      007150 006100      ROL R0
1782      007152 020204      CMP R2,R4      ;COMPARE DIVISOR AND DIVIDEND
1783      007154 100402      BMI NOSUB      ;IF DIVSOR LARGER NO SUBTRCTION
1784      007156 160402      SUB R4,R2
1785      007160 005201      INC R1      ;NOTE SUBTRACTION IN ANSWER
1786      007162 006303      NOSUB: ASL R3      ;MAKE DIVIDEND LARGER
1787      007164 006102      ROL R2      ;BY 2
1788      007166 005305      DEC R5      ;DO ALL THIS SHIFCNT TIMES
1789      007170 001366      BNE DIVL00
1790      007172 000241      CLC
1791      007174 006002      RLJL: ROR R2      ;NOW RIGHT JUSTIFY THE REMAINDER
1792      007176 005337 007206      DEC RMJUST
1793      007202 001374      BNE RLJL
1794      007204 000207      RTS PC
1795      007206 000000      RMJUST: 0
1796
1797      ;*****
1798      ;THIS ROUTINE AVERAGES THE LAST 32. X-Y JOYSTICK VALUES
1799      ;*****
1800      007210 010046      XYAVE: MOV R0,-(SP)      ;SAVE R0
1801      007212 013700 001246      MOV TEMPO,R0      ;GET X & Y
1802      007216 013702 007436      MOV XYBUF,R2      ;GET CURRENT BUFFER POINTER
1803      007222 010022      MOV R0,(R2)+      ;SAVE NEW X-Y VALUE
1804      007224 020227 007436      CMP R2,#XYBUFE      ;END OF BUFFER?
1805      007230 103402      BLO 1$      ;NO
1806      007232 012702 007336      MOV #XYBUF,R2      ;YES, GO BACK TO BEGINING OF BUFFER
1807      007236 010237 007436      MOV R2,XYBUF      ;SAVE NEW BUFFER POINTER
1808      007242 012702 007336      MOV #XYBUF,R2      ;CALC AVE X
1809      007246 004737 007300      JSR PC,10$
1810      007252 110046      MOV R0,-(SP)      ;SAVE IT
1811      007254 012702 007337      MOV #XYBUF+1,R2      ;CALC AVE Y
1812      007260 004737 007300      JSR PC,10$      ;GO DO IT
1813      007264 000300      SWAB R0      ;GET X-Y IN R0
1814      007266 152600      BISB (SP)+,R0      ;GET SAVED X
1815      007270 010037 001246      MOV R0,TEMPO      ;PUT IN TEMPO
1816      007274 012600      MOV (SP)+,R0      ;RESTORE R0
1817      007276 000207      RTS PC      ;EXIT WITH AVE X-Y IN TEMPO
1818
1819      007300 005000      10$: CLR R0      ;ZERO SUM
1820      007302 005005      11$: CLR R5      ;DO A MOVB TO A REG (UNSIGNED)
1821      007304 152205      BISB (R2)+,R5
1822      007306 060500      ADD R5,R0      ;ADD IT IN
1823      007310 005202      INC R2      ;SKIP OTHER VALUE

```

```

1824 007312 020227 007436      CMP      R2,#XYBUFE      ;END OF BUFFER?
1825 007316 103771      BLO      11$          ;NO
1826 007320 006300      ASL      R0          ;DIVIDE BY 32.
1827 007322 006300      ASL      R0
1828 007324 006300      ASL      R0
1829 007326 000300      SWAB     R0
1830 007330 042700 177400      BIC      #177400,R0      ;CLR HI BYTE
1831 007334 000207      RTS      PC

```

```

1832
1833 007336      XYBUF:

```

```

1834 007336 000000      0
1835 007340 000000      0
1836 007342 000000      0
1837 007344 000000      0
1838 007346 000000      0
1839 007350 000000      0
1840 007352 000000      0
1841 007354 000000      0
1842 007356 000000      0
1843 007360 000000      0
1844 007362 000000      0
1845 007364 000000      0
1846 007366 000000      0
1847 007370 000000      0
1848 007372 000000      0
1849 007374 000000      0
1850 007376 000000      0
1851 007400 000000      0
1852 007402 000000      0
1853 007404 000000      0
1854 007406 000000      0
1855 007410 000000      0
1856 007412 000000      0
1857 007414 000000      0
1858 007416 000000      0
1859 007420 000000      0
1860 007422 000000      0
1861 007424 000000      0
1862 007426 000000      0
1863 007430 000000      0
1864 007432 000000      0
1865 007434 000000      0

```

```

1866 007436      XYBUFE=
1867 007436 007336      XYBUFP: XYBUF

```

```

1868
1869 ;:*****
1870 .SBTTL TTY INPUT ROUTINE

```

```

1871
1872 ;:*****
1873 .ENABL LSB

```

```

1874
1875 ;:*****
1876 ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
1877 ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
1878 ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
1879 ;*WHEN OPERATING IN TTY FLAG MODE.

```

1880	007440	022737	000176	001140	\$CKSWR:	CMP	#SWREG,SWR	:: IS THE SOFT-SWR SELECTED?
1881	007446	001074				BNE	15\$	:: BRANCH IF NO
1882	007450	105777	171470			TSTB	2\$TKS	:: CHAR THERE?
1883	007454	100071				BPL	15\$	:: IF NO, DON'T WAIT AROUND
1884	007456	117746	171464			MOVB	2\$TKB, -(SP)	:: SAVE THE CHAR
1885	007462	042716	177600			BIC	#1C177, (SP)	:: STRIP-OFF THE ASCII
1886	007466	022726	000007			CMP	#7, (SP)+	:: IS IT A CONTROL G?
1887	007472	001032				BNE	15\$	:: NO, RETURN TO USER
1888	007474	123737	001134	000001		CMPB	\$AUTOB, #1	:: ARE WE RUNNING IN AUTO-MODE?
1889	007502	001456				BEQ	15\$	:: BRANCH IF YES
1890								
1891	007504	104400	010312			TYPE	.\$CNTLG	:: ECHO THE CONTROL-G (↑G)
1892	007510	104400	010317		\$GTSWR:	TYPE	.\$MSWR	:: TYPE CURRENT CONTENTS
1893	007514	013746	000176			MOV	SWREG, -(SP)	:: SAVE SWREG FOR TYPEOUT
1894	007520	104401				TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
1895	007522	104400	010330			TYPE	.\$MNEW	:: PROMPT FOR NEW SWR
1896	007526	005046			19\$:	CLR	-(SP)	:: CLEAR COUNTER
1897	007530	005046				CLR	-(SP)	:: THE NEW SWR
1898	007532	105777	171406		7\$:	TSTB	2\$TKS	:: CHAR THERE?
1899	007536	100375				BPL	7\$	:: IF NOT TRY AGAIN
1900								
1901	007540	117746	171402			MOVB	2\$TKB, -(SP)	:: PICK UP CHAR
1902	007544	042716	177600			BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
1903								
1904								
1905								
1906	007550	021627	000025		9\$:	CMP	(SP), #25	:: IS IT A CONTROL-U?
1907	007554	001005				BNE	10\$	:: BRANCH IF NOT
1908	007556	104400	010305			TYPE	.\$CNTLU	:: YES, ECHO CONTROL-U (↑U)
1909	007562	062706	000006		20\$:	ADD	#6, SP	:: IGNORE PREVIOUS INPUT
1910	007566	000757				BR	19\$	:: LET'S TRY IT AGAIN
1911								
1912								
1913	007570	021627	000015		10\$:	CMP	(SP), #15	:: IS IT A <CR>?
1914	007574	001022				BNE	16\$	:: BRANCH IF NO
1915	007576	005766	000004			TST	4(SP)	:: YES, IS IT THE FIRST CHAR?
1916	007602	001403				BEQ	11\$	:: BRANCH IF YES
1917	007604	016677	000002	171326		MOV	2(SP), 2SWR	:: SAVE NEW SWR
1918	007612	062706	000006		11\$:	ADD	#6, SP	:: CLEAR UP STACK
1919	007616	104400	001161		14\$:	TYPE	.\$CRLF	:: ECHO <CR> AND <LF>
1920	007622	123727	001135	000001		CMPB	\$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS
1921	007630	001003				BNE	15\$	:: BRANCH IF NOT
1922	007632	012777	000100	171304		MOV	#100, 2\$TKS	:: RE-ENABLE TTY KBD INTERRUPTS
1923	007640	000002			15\$:	RTI		:: RETURN
1924	007642	004737	010512		16\$:	JSR	PC, \$TYPEC	:: ECHO CHAR
1925	007646	021627	000060			CMP	(SP), #60	:: CHAR < 0?
1926	007652	002420				BLT	18\$	:: BRANCH IF YES
1927	007654	021627	000067			CMP	(SP), #67	:: CHAR > 7?
1928	007660	003015				BGT	18\$	:: BRANCH IF YES
1929	007662	042726	000060			BIC	#60, (SP)+	:: STRIP-OFF ASCII
1930	007666	005766	000002			TST	2(SP)	:: IS THIS THE FIRST CHAR
1931	007672	001403				BEQ	17\$	:: BRANCH IF YES
1932	007674	006316				ASL	(SP)	:: NO, SHIFT PRESENT
1933	007676	006316				ASL	(SP)	:: CHAR OVER TO MAKE
1934	007700	006316				ASL	(SP)	:: ROOM FOR NEW ONE.
1935	007702	005266	000002		17\$:	INC	2(SP)	:: KEEP COUNT OF CHAR

CONC9.P11

TTY INPUT ROUTINE

```

193: 007706 056616 177776      BIS      -2(SP), (SP)      :: SET IN NEW CHAR
194: 007712 000707      BR       75              :: GET THE NEXT ONE
195: 007714 104400 001160      185:     TYPE      $0LES      :: TYPE ?(CR/LF)
196: 007720 000720      BR       205          :: SIMULATE CONTROL-U
      .DSABL  LSB

*****
* THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
* CALL:
*   RDCHR
*   RETURN HERE
*
      RDCHR: MOV      (SP), -(SP)      :: PUSH DOWN THE PC
      MOV      4(SP), 2(SP)          :: SAVE THE PS
15:  TSTB      $TKS                  :: WAIT FOR
      BPL      15                  :: A CHARACTER
      MOVB     $TKB, 4(SP)          :: READ THE TTY
      BIC      #177, 4(SP)          :: GET RID OF JUNK IF ANY
      CMP      4(SP), #23          :: IS IT A CONTROL-5?
      BNE      35                  :: BRANCH IF NO
      TSTB     $TKS                  :: WAIT FOR A CHARACTER
      BPL      25                  :: LOOP UNTIL ITS THERE
      MOVB     $TKB, -(SP)          :: GET CHARACTER
      BIC      #0177, (SP)          :: MAKE IT 7-BIT ASCII
      CMP      (SP)+, #21          :: IS IT A CONTROL-Q?
      BNE      25                  :: IF NOT DISCARD IT
      BR       15                  :: YES, RESUME
      CMP      4(SP), #140          :: IS IT UPPER CASE?
      BLT      45                  :: BRANCH IF YES
      CMP      4(SP), #175          :: IS IT A SPECIAL CHAR?
      BGT      45                  :: BRANCH IF YES
      BIC      #40, 4(SP)          :: MAKE IT UPPER CASE
45:  RTI                          :: GO BACK TO USER

*****
* THIS ROUTINE WILL INPUT A STRING FROM THE TTY
* CALL:
*   RDLIN
*   RETURN HERE
*
      RDLIN: MOV      R3, -(SP)      :: SAVE R3
      CLR      -(SP)              :: CLEAR THE RUBOUT KEY
15:  MOV      $TTYIN, R3          :: GET ADDRESS
25:  CMP      $TTYIN+7, R3        :: BUFFER FULL?
      BLOS     45                  :: BR IF YES
      RDCHR                    :: GO READ ONE CHARACTER FROM THE TTY
      MOVB     (SP)+, (R3)          :: GET CHARACTER
105: CMPB     #177, (R3)          :: IS IT A RUBOUT
      BNE      55                  :: BR IF NO
      TST      (SP)                :: IS THIS THE FIRST RUBOUT?
      BNE      65                  :: BR IF NO
      MOVB     #'\, 95             :: TYPE A BACK SLASH
      TYPE     , 95

```

```

1992 010110 012716 177777      MOV      R-1.(SP)      :: SET THE RUBOUT KEY
1993 010114 005303      DEC      R3              :: BACKUP BY ONE
1994 010116 020327      CMP      R3,STTYIN      :: STACK EMPTY?
1995 010122 003434      BLO      4$              :: BR IF YES
1996 010124 111337      MOVB     (R3),9$      :: SETUP TO TYPEOUT THE DELETED CHAR.
1997 010130 104400      TYPE     9$              :: GO TYPE
1998 010134 000746      BR       2$              :: GO READ ANOTHER CHAR.
1999 010136 005716      5$: TST      (SP)          :: RUBOUT KEY SET?
2000 010140 001406      SEQ      7$              :: BR IF NO
2001 010142 112737      MOVB     R-1,9$      :: TYPE A BACK SLASH
2002 010150 104400      TYPE     9$              ::
2003 010154 005016      CLR      (SP)          :: CLEAR THE RUBOUT KEY
2004 010156 122713      7$: CMPB     R25,(R3)      :: IS CHARACTER A CTRL U?
2005 010162 001003      BNE     8$              :: BR IF NO
2006 010164 104400      TYPE     $CNTLU      :: TYPE A CONTROL "U"
2007 010170 000726      BR       1$              :: GO START OVER
2008 010172 122713      9$: CMPB     R22,(R3)      :: IS CHARACTER A "IR"?
2009 010176 001011      BNE     3$              :: BRANCH IF NO
2010 010200 105013      CLRB     (R3)          :: CLEAR THE CHARACTER
2011 010202 104400      TYPE     $SCRLF      :: TYPE A "CR" & "LF"
2012 010206 104400      TYPE     $TTYIN      :: TYPE THE INPUT STRING
2013 010212 000717      BR       2$              :: GO PICKUP ANOTHER CHARACTER
2014 010214 104400      4$: TYPE     $QUES      :: TYPE A "?"
2015 010220 000712      BR       1$              :: CLEAR THE BUFFER AND LOOP
2016 010222 111337      3$: MOVB     (R3),9$      :: ECHO THE CHARACTER
2017 010226 104400      TYPE     9$              ::
2018 010232 122723      CMPB     R15,(R3)+    :: CHECK FOR RETURN
2019 010236 001305      BNE     2$              :: LOOP IF NOT RETURN
2020 010240 105063      CLRB     -1,(R3)      :: CLEAR RETURN (THE 15)
2021 010244 104400      TYPE     $LF          :: TYPE A LINE FEED
2022 010250 005726      TST      (SP)+        :: CLEAN RUBOUT KEY FROM THE STACK
2023 010252 012603      MOV      (SP)+,R3      :: RESTORE R3
2024 010254 011646      MOV      (SP)-,(SP)    :: ADJUST THE STACK AND PUT ADDRESS OF THE
2025 010256 016666      MOV      4(SP),2(SP)   :: FIRST ASCII CHARACTER ON IT
2026 010264 012766      MOV      $STTYIN,4(SP)
2027 010272 000002      RTI                    :: RETURN
2028 010274 000      9$: .BYTE     0              :: STORAGE FOR ASCII CHAR. TO TYPE
2029 010275 000      .BYTE     0              :: TERMINATOR
2030 010276 000007      STTYIN: .BLKB     7      :: RESERVE 7 BYTES FOR TTY INPUT
2031 010305 136 006525 000012 $CNTLU: .ASCIZ  /IU<15><12>  :: CONTROL "U"
2032 010312 043536 005015 000 $CNTLG: .ASCIZ  /IG<15><12>  :: CONTROL "G"
2033 010317 015 051412 051127 $MSWR: .ASCIZ  <15><12>/SWR = /
2034 010324 036440 000040      $MNEW: .ASCIZ  / NEW = /
2035 010330 020040 042516 020127
2036 010336 020075 000
2037 010342
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047

```

.EVEN

 .SBTTL TYPE ROUTINE

 *ROUTINE TO TYPE ASCII 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.
 *

```

2048      :CALL:
2049      :*1) USING A TRAF INSTRUCTION
2050      :*      TYPE      .MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2051      :*OR
2052      :*      TYPE
2053      :*      MESADR
2054      :*
2055
2056      010342 105737 001157      STYPE:  TSTB      STPFLG      ;; IS THERE A TERMINAL?
2057      010346 100002              BPL          1$          ;; BR IF YES
2058      010350 000000              HALT              ;; HALT HERE IF NO TERMINAL
2059      010352 000407              BR          3$          ;; LEAVE
2060      010354 010046      1$:      MOV      RO, -(SP)      ;; SAVE RO
2061      010356 017607 000002      MOV      22(SP), RO      ;; GET ADDRESS OF ASCIZ STRING
2062      010362 112046      2$:      MOV      (RO)+, -(SP)      ;; PUSH CHARACTER TO BE TYPED ONTO STACK
2063      010364 001005              BNE          4$          ;; BR IF IT ISN'T THE TERMINATOR
2064      010366 005726              TST      (SP)+          ;; IF TERMINATOR POP IT OFF THE STACK
2065      010370 012600      60$:      MOV      (SP)+, RO      ;; RESTORE RO
2066      010372 062716 000002      3$:      ADD      #2, (SP)      ;; ADJUST RETURN PC
2067      010376 000002              RTI              ;; RETURN
2068      010400 122716 000011      4$:      CMPB      #HT, (SP)      ;; BRANCH IF <HT>
2069      010404 001430              BEQ          8$          ;;
2070      010406 122716 000200              CMPB      #CRLF, (SP)      ;; BRANCH IF NOT <CRLF>
2071      010412 001006              BNE          5$          ;;
2072      010414 005726              TST      (SP)+          ;; POP <CR><LF> EQUIV
2073      010416 104400              TYPE              ;; TYPE A CR AND LF
2074      010420 001161              $CRLF
2075      010422 105037 010556              CLRB      $CHARCNT      ;; CLEAR CHARACTER COUNT
2076      010426 000755              BR          2$          ;; GET NEXT CHARACTER
2077      010430 004737 010512      5$:      JSR      PC, $TYPEC      ;; GO TYPE THIS CHARACTER
2078      010434 123726 001156      6$:      CMPB      $FILLC, (SP)+      ;; IS IT TIME FOR FILLER CHARS.?
2079      010440 001350              BNE          2$          ;; IF NO GO GET NEXT CHAR.
2080      010442 013746 001154              MOV      $NULL, -(SP)      ;; GET # OF FILLER CHARS. NEEDED
2081              AND THE NULL CHAR.
2082      010446 105366 000001      7$:      DECB      1(SP)      ;; DOES A NULL NEED TO BE TYPED?
2083      010452 002770              BLT          6$          ;; BR IF NO--GO POP THE JLL OFF OF STACK
2084      010454 004737 010512              JSR      PC, $TYPEC      ;; GO TYPE A NULL
2085      010460 105337 010556              DECB      $CHARCNT      ;; DO NOT COUNT AS A COUNT
2086      010464 000770              BR          7$          ;; LOOP
2087
2088      :HORIZONTAL TAB PROCESSOR
2089
2090      010466 112716 000040      8$:      MOV      #' , (SP)      ;; REPLACE TAB WITH SPACE
2091      010472 004737 010512      9$:      JSR      PC, $TYPEC      ;; TYPE A SPACE
2092      010476 132737 000007 010556      BITB      #7, $CHARCNT      ;; BRANCH IF NOT AT
2093      010504 001372              BNE          9$          ;; TAB STOP
2094      010506 005726              TST      (SP)+          ;; POP SPACE OFF STACK
2095      010510 000724              BR          2$          ;; GET NEXT CHARACTER
2096      010512 105777 170432      STYPEC:  TSTB      $STPS      ;; WAIT UNTIL PRINTER IS READY
2097      010516 100375              BPL          $TYPEC
2098      010520 116677 000002 170424              MOV      2(SP), $STPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.
2099      010526 122766 000015 000002              CMPB      #CR, 2(SP)      ;; IS CHARACTER A CARRIAGE RETURN?
2100      010534 001003              BNE          1$          ;; BRANCH IF NO
2101      010536 105037 010556              CLRB      $CHARCNT      ;; YES--CLEAR CHARACTER COUNT
2102      010542 000406              BR          $TYPEX      ;; EXIT
2103      010544 122766 000012 000002 1$:      CMPB      #LF, 2(SP)      ;; IS CHARACTER A LINE FEED?

```

```

2104 010552 001402          BEQ      STYPEX          ;;BRANCH IF YES
2105 010554 105227          INCB      (PC)+          ;;COUNT THE CHARACTER
2106 010556 000000          $CHARCNT: WORD 0          ;;CHARACTER COUNT STORAGE
2107 010560 000207          STYPEX: RTS      PC
2108
2109          ;;*****
2110          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
2111          ;;*****
2112          ;;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
2113          ;;OCTAL (ASCII) NUMBER AND TYPE IT.
2114          ;;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
2115          ;;*CALL:
2116          ;;      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
2117          ;;      TYPOS          ;;CALL FOR TYPEOUT
2118          ;;      .BYTE  N          ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
2119          ;;      .BYTE  M          ;;M=1 OR 0
2120          ;;          ;;1=TYPE LEADING ZEROS
2121          ;;          ;;0=SUPPRESS LEADING ZEROS
2122          ;;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
2123          ;;*STYPOS OR STYPOC
2124          ;;*CALL:
2125          ;;      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
2126          ;;      TYPON          ;;CALL FOR TYPEOUT
2127          ;;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
2128          ;;*CALL:
2129          ;;      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
2130          ;;      TYPCC          ;;CALL FOR TYPEOUT
2131
2132          010562 017646 000000          STYPOS: MOV      2(SP),-(SP)          ;;PICKUP THE MODE
2133          010566 116637 000001 011005  MOV      1(SP), $OFILL          ;;LOAD ZERO FILL SWITCH
2134          010574 112637 011007          MOV      (SP)+, $OMODE+1          ;;NUMBER OF DIGITS TO TYPE
2135          010600 062716 000002          ADD      #2, (SP)          ;;ADJUST RETURN ADDRESS
2136          010604 000406          BR      STYPON
2137          010606 112737 000001 011005  STYPOC: MOV      #1, $OFILL          ;;SET THE ZERO FILL SWITCH
2138          010614 112737 000006 011007  MOV      #6, $OMODE+1          ;;SET FOR SIX(6) DIGITS
2139          010622 112737 000005 011004  STYPON: MOV      #5, $OCNT          ;;SET THE ITERATION COUNT
2140          010630 010346          MOV      R3, -(SP)          ;;SAVE R3
2141          010632 010446          MOV      R4, -(SP)          ;;SAVE R4
2142          010634 010546          MOV      R5, -(SP)          ;;SAVE R5
2143          010636 113704 011007          MOV      $OMODE+1, R4          ;;GET THE NUMBER OF DIGITS TO TYPE
2144          010642 005404          NEG      R4
2145          010644 062704 000006          ADD      #6, R4          ;;SUBTRACT IT FOR MAX. ALLOWED
2146          010650 110437 011006          MOV      R4, $OMODE          ;;SAVE IT FOR USE
2147          010654 113704 011005          MOV      $OFILL, R4          ;;GET THE ZERO FILL SWITCH
2148          010660 016605 000012          MOV      12(SP), R5          ;;PICKUP THE INPUT NUMBER
2149          010664 005003          CLR      R3          ;;CLEAR THE OUTPUT WORD
2150          010666 006105          1$: ROL      R5          ;;ROTATE MSB INTO "C"
2151          010670 000404          BR      3$          ;;GO DO MSB
2152          010672 006105          2$: ROL      R5          ;;FORM THIS DIGIT
2153          010674 006105          ROL      R5
2154          010676 006105          ROL      R5
2155          010700 010503          MOV      R5, R3
2156          010702 006103          3$: ROL      R3          ;;GET LSB OF THIS DIGIT

```

F04

MAINDEC-11-DZNCB-C GAMMA 1: EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 44
 DZNCB.P11 BINARY TO OCTAL (ASCII) AND TYPE

2160	010704	105337	011006	DECB	\$OMODE	::TYPE THIS DIGIT?
2161	010710	100016		BPL	7\$	::BR IF NO
2162	010712	042703	177770	BIC	#177770,R3	::GET RID OF JUNK
2163	010716	001002		BNE	4\$	::TEST FOR 0
2164	010720	005704		TST	R4	::SUPPRESS THIS 0?
2165	010722	001403		BEQ	5\$	::BR IF YES
2166	010724	005204		4\$: INC	R4	::DON'T SUPPRESS ANYMORE 0'S
2167	010726	052703	000060	BIS	#'0,R3	::MAKE THIS DIGIT ASCII
2168	010732	052703	000040	5\$: BIS	#',R3	::MAKE ASCII IF NOT ALREADY
2169	010736	110337	011002	MOVB	R3,8\$	::SAVE FOR TYPING
2170	010742	104400	011002	TYPE	8\$	::GO TYPE THIS DIGIT
2171	010746	105337	011004	7\$: DECB	\$OCNT	::COUNT BY 1
2172	010752	003347		SGT	2\$	::BR IF MORE TO DO
2173	010754	002402		BLT	6\$	::BR IF DONE
2174	010756	005204		INC	R4	::INSURE LAST DIGIT ISN'T A BLANK
2175	010760	000744		BR	2\$	::GO DO THE LAST DIGIT
2176	010762	012605		5\$: MOV	(SP)+,R5	::RESTORE R5
2177	010764	012604		MOV	(SP)+,R4	::RESTORE R4
2178	010766	012603		MOV	(SP)+,R3	::RESTORE R3
2179	010770	016666	000002 000004	MOV	2(SP),4(SP)	::SET THE STACK FOR RETURNING
2180	010776	012616		MOV	(SP)+,(SP)	
2181	011000	000002		RTI		::RETURN
2182	011002	000		8\$: .BYTE	0	::STORAGE FOR ASCII DIGIT
2183	011003	000		.BYTE	0	::TERMINATOR FOR TYPE ROUTINE
2184	011004	000		\$OCNT: .BYTE	0	::OCTAL DIGIT COUNTER
2185	011005	000		\$OFILL: .BYTE	0	::ZERO FILL SWITCH
2186	011006	000000		\$OMODE: .WORD	0	::NUMBER OF DIGITS TO TYPE
2187				:*****		
2188				.SBTTL READ AN OCTAL NUMBER FROM THE TTY		
2189				:*****		
2190				:THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND		
2191				:CHANGE IT TO BINARY.		
2192				:THE INPUT CHARACTERS WILL BE CHECKED TO INSURE THEY ARE LEGAL		
2193				:OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED		
2194				:FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST		
2195				:THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.		
2196				:CALL:		
2197				:*		
2198				RDOCT		::READ AN OCTAL NUMBER
2199				* RETURN HERE		::LOW ORDER BITS ARE ON TOP OF THE STACK
2200				* *		::HIGH ORDER BITS ARE IN \$HI OCT
2201						
2202	011010	011646		\$RDOCT: MOV	(SP)-,(SP)	::PROVIDE SPACE FOR THE
2203	011012	016666	000004 000002	MOV	4(SP),2(SP)	::INPUT NUMBER
2204	011020	010046		MOV	R0,-(SP)	::PUSH R0 ON STACK
2205	011022	010146		MOV	R1,-(SP)	::PUSH R1 ON STACK
2206	011024	010246		MOV	R2,-(SP)	::PUSH R2 ON STACK
2207	011026	104407		1\$: RDLIN		::READ AN ASCII LINE
2208	011030	012600		MOV	(SP)+,R0	::GET ADDRESS OF 1ST CHARACTER
2209	011032	010037	011136	MOV	R0,5\$	::AND SAVE IT
2210	011036	005001		CLR	R1	::CLEAR DATA WORD
2211	011040	005002		CLR	R2	
2212	011042	112046		2\$: MOVB	(R0)+,-(SP)	::PICKUP THIS CHARACTER
2213	011044	001420		BEQ	3\$	::IF ZERO GET OUT
2214	011046	122716	000060	CMPB	#'0,(SP)	::MAKE SURE THIS CHARACTER
2215	011052	003026		BGT	4\$	::IS AN OCTAL DIGIT

G04

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 45
 DZNCB.P11 READ AN OCTAL NUMBER FROM THE TTY

2216	011054	122716	000067	CMPB	#7,(SP)	
2217	011060	002423		BLT	4\$	
2218	011062	006301		ASL	R1	::*2
2219	011064	006102		ROL	R2	
2220	011066	006301		ASL	R1	::*4
2221	011070	006102		ROL	R2	
2222	011072	006301		ASL	R1	::*8
2223	011074	006102		ROL	R2	
2224	011076	042716	177770	BIC	#1C7,(SP)	::STRIP THE ASCII JUNK
2225	011102	062601		ADD	(SP)+,R1	::ADD IN THIS DIGIT
2226	011104	000756		BR	2\$	::LOOP
2227	011106	005726		3\$: TST	(SP)+	::CLEAN TERMINATOR FROM STACK
2228	011110	010166	000012	MOV	R1,12(SP)	::SAVE THE RESULT
2229	011114	010237	011146	MOV	R2,\$HI OCT	
2230	011120	012602		MOV	(SP)+,R2	::POP STACK INTO R2
2231	011122	012601		MOV	(SP)+,R1	::POP STACK INTO R1
2232	011124	012600		MOV	(SP)+,R0	::POP STACK INTO R0
2233	011126	000002		RTI		::RETURN
2234	011130	005726		4\$: TST	(SP)+	::CLEAN PARTIAL FROM STACK
2235	011132	105010		CLRB	(R0)	::SET A TERMINATOR
2236	011134	104400		TYPE		::TYPE UP THRU THE BAD CHAR.
2237	011136	000000		5\$: .WORD	0	
2238	011140	104400	001160	TYPE	\$QUES	::?" "CR" & "LF"
2239	011144	000730		BR	1\$	::TRY AGAIN
2240	011146	000000		\$HI OCT: .WORD	0	::HIGH ORDER BITS GO HERE
2241				;*****		
2242				.SBTTL TRAP DECODER		
2243				;*****		
2244				;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION		
2245				;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS		
2246				;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL		
2247				;*GO TO THAT ROUTINE.		
2248						
2249						
2250	011150	010046		\$TRAP: MOV	R0,-(SP)	::SAVE R0
2251	011152	016600	000002	MOV	2(SP),R0	::GET TRAP ADDRESS
2252	011156	005740		TST	-(R0)	::BACKUP BY 2
2253	011160	111000		MOVB	(R0),R0	::GET RIGHT BYTE OF TRAP
2254	011162	006300		ASL	R0	::POSITION FOR INDEXING
2255	011164	016000	011172	MOV	\$TRPAD(R0),R0	::INDEX TO TABLE
2256	011170	000200		RTS	R0	::GO TO ROUTINE
2257						
2258				.SBTTL TRAP TABLE		
2259						
2260				;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED		
2261				;*BY THE "TRAP" INSTRUCTION.		
2262						
2263				ROUTINE		
2264				-----		
2265	011172			\$TRPAD:		
2266	011172	010342		\$TYPE	::CALL=TYPE	TRAP+0(104400) TTY TYPEOUT ROUTINE
2267	011174	010606		\$TYPOC	::CALL=TYPOC	TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
2268	011176	010562		\$TYPOS	::CALL=TYPOS	TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
2269	011200	010622		\$TYPON	::CALL=TYPON	TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
2270						
2271	011202	007510		\$GTSWR	::CALL=GTSWR	TRAP+4(104404) GET SOFT-SWR SETTING

```

2272
2273 011204 007440          $CKSWR  ::CALL=CKSWR   TRAP+5(104405)  TEST FOR CHANGE IN SOFT-SWR
2274 011206 007722          $RDCHR  ::CALL=RDCHR   TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
2275 011210 010042          $RDLIN  ::CALL=RDLIN   TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
2276 011212 011010          $RDOCT  ::CALL=RDOCT   TRAP+10(104410) READ AN OCTAL NUMBER FROM TTY
2277
2278          ::*****
2279          $SBTTL  POWER DOWN AND UP ROUTINES
2280
2281          ::*****
2282          $POWER DOWN ROUTINE
2283 011214 012737 011360 000024 $PWRDN: MOV    $SILLUP, @PWRVEC  ;; SET FOR FAST UP
2284 011222 012737 000340 000026      MOV    #340, @PWRVEC+2  ;; PRI0:7
2285 011230 010046          MOV    R0, -(SP)  ;; PUSH R0 ON STACK
2286 011232 010146          MOV    R1, -(SP)  ;; PUSH R1 ON STACK
2287 011234 010246          MOV    R2, -(SP)  ;; PUSH R2 ON STACK
2288 011236 010346          MOV    R3, -(SP)  ;; PUSH R3 ON STACK
2289 011240 010446          MOV    R4, -(SP)  ;; PUSH R4 ON STACK
2290 011242 010546          MOV    R5, -(SP)  ;; PUSH R5 ON STACK
2291 011244 017746 167670      MOV    @SWR, -(SP)  ;; PUSH @SWR ON STACK
2292 011250 010637 011364      MOV    SP, $SAVR6  ;; SAVE SP
2293 011254 012737 011266 000024      MOV    $PWRUP, @PWRVEC  ;; SET UP VECTOR
2294 011262 000000          HALT
2295 011264 000776          BR      .-2          ;; HANG UP
2296
2297          ::*****
2298          $POWER UP ROUTINE
2299 011266 012737 011360 000024 $PWRUP: MOV    $SILLUP, @PWRVEC  ;; SET FOR FAST DOWN
2300 011274 013706 011364      MOV    $SAVR6, SP  ;; GET SP
2301 011300 005037 011364      CLR    $SAVR6  ;; WAIT LOOP FOR THE TTY
2302 011304 005237 011364      1$: INC    $SAVR6  ;; WAIT FOR THE INC
2303 011310 001375          BNE    1$  ;; OF WORD
2304 011312 012677 167622      MOV    (SP)+, @SWR  ;; POP STACK INTO @SWR
2305 011316 012505          MOV    (SP)+, R5  ;; POP STACK INTO R5
2306 011320 012604          MOV    (SP)+, R4  ;; POP STACK INTO R4
2307 011322 012603          MOV    (SP)+, R3  ;; POP STACK INTO R3
2308 011324 012602          MOV    (SP)+, R2  ;; POP STACK INTO R2
2309 011326 012601          MOV    (SP)+, R1  ;; POP STACK INTO R1
2310 011330 012600          MOV    (SP)+, R0  ;; POP STACK INTO R0
2311 011332 012737 011214 000024      MOV    $PWRDN, @PWRVEC  ;; SET UP THE POWER DOWN VECTOR
2312 011340 012737 000340 000026      MOV    #340, @PWRVEC+2  ;; PRI0:7
2313 011346 104400          TYPE  PWRMSG  ;; REPORT THE POWER FAILURE
2314 011350 013222          $PWRMG: .WORD PWRMSG  ;; POWER FAIL MESSAGE POINTER
2315 011352 012716          MOV    (PC)+, (SP)  ;; RESTART AT START1
2316 011354 001616          $PWRAD: .WORD START1  ;; RESTART ADDRESS
2317 011356 000002          RTI
2318 011360 000000          $SILLUP: HALT
2319 011362 000776          BR      .-2          ;; THE POWER UP SEQUENCE WAS STARTED
2319 011364 000000          $SAVR6: 0          ;; BEFORE THE POWER DOWN WAS COMPLETE
2319          ;; PUT THE SP HERE

```

2320 .SBTTL VTO1 INTENSITY POINTERS & DATA
2321 :TABLE OF POINTERS TO X AND Y INTENSITY PATTERNS FOR VTO1 DISPLAY
2322

2323	011366	011466	LEVTAB: L1
2324	011370	011467	L2
2325	011372	011471	L3
2326	011374	011474	L4
2327	011376	011500	L5
2328	011400	011505	L6
2329	011402	011513	L7
2330	011404	011522	L8
2331	011406	011532	L9
2332	011410	011543	L10
2333	011412	011555	L11
2334	011414	011570	L12
2335	011416	011604	L13
2336	011420	011621	L14
2337	011422	011637	L15
2338	011424	011656	L16
2339	011426	011676	L17
2340	011430	011717	L18
2341	011432	011741	L19
2342	011434	011764	L20
2343	011436	012010	L21
2344	011440	012035	L22
2345	011442	012063	L23
2346	011444	012112	L24
2347	011446	012142	L25
2348	011450	012173	L26
2349	011452	012225	L27
2350	011454	012260	L28
2351	011456	012314	L29
2352	011460	012351	L30
2353	011462	012407	L31
2354	011464	012446	L32

2355 :THIS IS THE CODED MATRIX TABLE FOR INTENSITY DISPLAY
2356 :THE ENTRIES ARE PACKED 2X,Y COORDINATES TO A WORD
2357 :THE EVEN BYTE CONTAINS THE FIRST PAIR, 3 BITS PER
2358 :COORDINATE.
2359

2360					
2361	011466	033		L1:	.BYTE 33
2362	011467	016	062	L2:	.BYTE 16,62
2363	011471	026	032	L3:	.BYTE 26,32,64
2364	011474	021	015	L4:	.BYTE 21,15,56,62
2365	011477	062			
2366	011500	005	020	L5:	.BYTE 05,20,43,57,71
2367	011503	057	071		
2368	011505	024	000	L6:	.BYTE 24,00,32,47,65,51
2369	011510	047	065		
2370	011513	002	015	L7:	.BYTE 02,15,20,43,46,61,65
2371	011516	043	046		
2372	011521	065			
2373	011522	005	012	L8:	.BYTE 05,12,24,37,41,56,70,63
2374	011525	037	041		
2375	011530	070	063		

J04

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 48
 DZNCB.P11 VTO1 INTENSITY POINTERS & DATA

2376	011532	000	013	016	L9:	.BYTE 00,13,16,31,35,43,57,61,74
2377	011535	031	035	043		
2378	011540	057	061	074		
2379	011543	000	013	016	L10:	.BYTE 00,13,16,31,43,55,57,61,74,35
2380	011546	031	043	055		
2381	011551	057	061	074		
2382	011554	035				
2383	011555	012	015	027	L11:	.BYTE 12,15,27,33,41,45,62,57,74,00,76
2384	011560	033	041	045		
2385	011563	062	057	074		
2386	011566	000	076			
2387	011570	000	003	016	L12:	.BYTE 00,03,16,21,24,37,42,45,57,61,64,76
2388	011573	021	024	037		
2389	011576	042	045	057		
2390	011601	061	064	076		
2391	011604	007	012	015	L13:	.BYTE 07,12,15,20,34,36,42,50,54,56,71,73,75
2392	011607	020	034	036		
2393	011612	042	050	054		
2394	011615	056	071	073		
2395	011620	075				
2396	011621	001	005	007	L14:	.BYTE 01,05,07,13,20,26,32,34,46,40,52,60,65,73
2397	011624	013	020	026		
2398	011627	032	034	046		
2399	011632	040	052	060		
2400	011635	065	073			
2401	011637	003	006	027	L15:	.BYTE 03,00,27,11,25,30,33,45,47,50,54,52,71,74,66
2402	011642	011	025	030		
2403	011645	033	045	047		
2404	011650	050	054	052		
2405	011653	071	074	066		
2406	011656	006	010	023	L16:	.BYTE 06,10,23,25,27,30,32,44,46,50,53,64,71,77,02,04
2407	011661	025	027	030		
2408	011664	032	044	046		
2409	011667	050	057	064		
2410	011672	071	077	002		
2411	011675	004				
2412	011676	002	004	006	L17:	.BYTE 02,04,06,10,22,25,27,30,33
2413	011701	010	022	025		
2414	011704	027	030	033		
2415	011707	045	047	050		.BYTE 45,47,50,52,65,71,73,77
2416	011712	052	065	071		
2417	011715	073	077			
2418	011717	002	014	006	L18:	.BYTE 02,14,06,10,22,25,27,30,33
2419	011722	010	022	025		
2420	011725	027	030	033		
2421	011730	045	047	050		.BYTE 45,47,50,52,64,66,71,73,77
2422	011733	052	064	066		
2423	011736	071	073	077		
2424	011741	002	004	006	L19:	.BYTE 02,04,06,10,22,25,27,30,34,41,43,46,50,57
2425	011744	010	022	025		
2426	011747	027	030	034		
2427	011752	041	043	046		
2428	011755	050	057			
2429	011757	063	065	071		.BYTE 63,65,71,74,77
2430	011762	074	077			
2431	011764	001	003	005	L20:	.BYTE 01,03,05,07,12,30,23,64,25,42,27,34,36,51

K04

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 49
 DZNCB.P11 VTO1 INTENSITY POINTERS & DATA

2432	011767	007	012	030	
2433	011772	023	064	025	
2434	011775	042	027	034	
2435	012000	036	051		
2436	012002	055	057	070	.BYTE 55,57,70,53,66,72
2437	012005	053	066	072	
2438	012010	001	003	005	L21: .BYTE 01,03,05,07,10,40,16,21,23,25,42,27,34,45
2439	012013	007	010	040	
2440	012016	016	021	023	
2441	012021	025	042	027	
2442	012024	034	045		
2443	012026	051	055	057	.BYTE 51,55,57,60,64,66,72
2444	012031	060	064	066	
2445	012034	072			
2446	012035	001	003	005	L22: .BYTE 01,03,05,07,10,40,16,21,23,25,42,27,34,46
2447	012040	007	010	040	
2448	012043	016	021	023	
2449	012046	025	042	027	
2450	012051	034	046		
2451	012053	051	053	055	.BYTE 51,53,55,57,60,64,66,72
2452	012056	057	060	064	
2453	012061	066	072		
2454	012063	000	002	005	L23: .BYTE 00,02,05,11,13,16,22,34,27,31,25,40,42,47
2455	012066	011	013	016	
2456	012071	022	034	027	
2457	012074	031	025	040	
2458	012077	042	047		
2459	012101	051	054	056	.BYTE 51,54,56,60,63,65,71,74,76
2460	012104	060	063	065	
2461	012107	071	074	076	
2462	012112	000	003	007	L24: .BYTE 00,03,07,12,14,16,21,23,25,27,32,34,30,43
2463	012115	012	014	016	
2464	012120	021	023	025	
2465	012123	027	032	034	
2466	012126	030	043		
2467	012130	046	050	052	.BYTE 46,50,52,55,63,65,67,71,74,76
2468	012133	055	063	065	
2469	012136	067	071	074	
2470	012141	076			
2471	012142	003	005	007	L25: .BYTE 03,05,07,10,12,14,16,21,23,27,32,34,36,41
2472	012145	010	012	014	
2473	012150	016	021	023	
2474	012153	027	032	034	
2475	012156	036	041		
2476	012160	045	047	050	.BYTE 45,47,50,52,54,63,65,67,70,72,76
2477	012163	052	054	063	
2478	012166	065	067	070	
2479	012171	072	076		
2480	012173	003	005	007	L26: .BYTE 03,05,07,10,12,16,21,23,25,30,32,34,36,41
2481	012176	010	012	016	
2482	012201	021	023	025	
2483	012204	030	032	034	
2484	012207	036	041		
2485	012211	045	047	052	.BYTE 45,47,52,54,56,61,63,67,70,72,74,76
2486	012214	054	056	061	
2487	012217	063	067	070	

L04

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 50
 DZNCB.P11 VTO1 INTENSITY POINTERS & DATA

2488	012222	072	074	076	
2489	012225	001	003	007	L27: .BYTE 01,03,07,12,14,16,21,23,25,27,30,32,34,36
2490	012230	012	014	016	
2491	012233	021	023	025	
2492	012236	027	030	032	
2493	012241	034	036		
2494	012243	041	045	047	.BYTE 41,45,47,50,52,54,61,63,65,67,70,74,76
2495	012246	050	052	054	
2496	012251	061	063	065	
2497	012254	067	070	074	
2498	012257	076			
2499	012260	003	005	007	L28: .BYTE 03,05,07,10,12,14,16,21,23,27,30,32,34,36
2500	012263	010	012	014	
2501	012266	016	021	023	
2502	012271	027	030	032	
2503	012274	034	036		
2504	012276	043	045	047	.BYTE 43,45,47,50,52,54,56,61,63,65,67,70,72,76
2505	012301	050	052	054	
2506	012304	056	061	063	
2507	012307	065	067	070	
2508	012312	072	076		
2509	012314	001	003	005	L29: .BYTE 01,03,05,07,10,12,16,14,21,23,27,30,32,34,36,43
2510	012317	007	010	012	
2511	012322	016	014	021	
2512	012325	023	027	030	
2513	012330	032	034	036	
2514	012333	043			
2515	012334	045	047	050	.BYTE 45,47,50,52,54,56,61,63,67,70,72,74,76
2516	012337	052	054	056	
2517	012342	061	063	067	
2518	012345	070	072	074	
2519	012350	076			
2520	012351	001	003	005	L30: .BYTE 01,03,05,07,10,12,14,16,21,23,25,27,30,32,36,41
2521	012354	007	010	012	
2522	012357	014	016	021	
2523	012362	023	025	027	
2524	012365	030	032	036	
2525	012370	041			
2526	012371	043	045	047	.BYTE 43,45,47,50,54,56,61,63,65,67,70,72,74,76
2527	012374	050	054	056	
2528	012377	061	063	065	
2529	012402	067	070	072	
2530	012405	074	076		
2531	012407	001	003	005	L31: .BYTE 01,03,05,07,10,12,14,16,21,23,25,27,30,32,36,41
2532	012412	007	010	012	
2533	012415	014	016	021	
2534	012420	023	025	027	
2535	012423	030	032	036	
2536	012426	041			
2537	012427	043	045	047	.BYTE 43,45,47,50,52,54,56,61,63,65,67,70,72,74,76
2538	012432	050	052	054	
2539	012435	056	061	063	
2540	012440	065	067	070	
2541	012443	072	074	076	
2542	012446	001	003	005	L32: .BYTE 01,03,05,07,10,12,14,16,21,23,25,27,30,32,34,36
2543	012451	007	010	012	

MO4

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 51
DZNCB.P11 VT01 INTENSITY POINTERS & DATA

2544	012454	014	016	021
2545	012457	023	025	027
2546	012462	030	032	034
2547	012465	036		
2548	012466	041	043	045
2549	012471	047	050	052
2550	012474	054	056	061
2551	012477	063	065	067
2552	012502	070	072	074
2553	012505	076		
2554				

.BYTE 41,43,45,47,50,52,54,56,61,63,65,67,70,72,74,76

.EVEN

N04

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 52
DZNCB.P11 DISPLAY MESSAGES

2555						.SBTTL DISPLAY MESSAGES
2556						;THIS IS THE MESSAGE FOR THE BOTTOM OF THE SCOPE
2557		000000				TERM=C
2558	012506	012536				MESS1: WD001
2559	012510	000000				TERM
2560	012512	012561				MESS2: WD002
2561	012514	000000				0
2562	012516	012604				MESS3: WD003
2563	012520	000000				0
2564	012522	012617				MESS4: WD004
2565	012524	000000				0
2566	012526	012636				MESS5: WD005
2567	012530	000000				0
2568	012532	012645				MESS6: WD006
2569	012534	000000				0
2570	012536	005015	047514	042527		WD001: .ASCII <15><12>/LOWER THRESHOLD /<END>
2571	012544	020122	044124	042522		
2572	012552	044123	046117	020104		
2573	012560	200				
2574	012561	015	052412	050120		WD002: .ASCII <15><12>/UPPER THRESHOLD /<END>
2575	012566	051105	052040	051110		
2576	012574	051505	047510	042114		
2577	012602	100040				
2578	012604	005015	020132	047503		WD003: .ASCII <15><12>/Z COUNT=/<END>
2579	012612	047125	036524	200		
2580	012617	040	040515	051124		WD004: .ASCII / MATRIX COUNT=/<END>
2581	012624	054111	041440	052517		
2582	012632	052116	100075			
2583	012636	020040	047532	046517		WD005: .ASCII / ZOOM/<END>
2584	012644	200				
2585	012645	040	026502	040507		WD006: .ASCII / B-GAMMA/<END>
2586	012652	046515	100101			
2587						.EVEN
2588		000200				END=200 ;MSG TERMINATOR FOR DISPLAY CHARS

```

.SBTTL  VTO1 CHARACTER TABLE
:THIS IS THE TABLE OF CODED LETTERS AVAILABLE FOR THE
:VTO1 SCOPE.

```

TABLE:		
77061		:A
35061		
45117		:B
37116		
02056		:C
34041		
45117		:D
37116		
02056		:E
34041		
45117		:F
37116		
02056		:G
34041		
45117		:H
37116		
02056		:I
34041		
45117		:J
37116		
02056		:K
34041		
45117		:L
37116		
02056		:M
34041		
45117		:N
37116		
02056		:O
34041		
45117		:P
37116		
02056		:Q
34041		
45117		:R
37116		
02056		:S
34041		
45117		:T
37116		
02056		:U
34041		
45117		:V
37116		
02056		:W
34041		
45117		:X
37116		
02056		:Y
34041		
45117		:Z
37116		

2645	013026	002057	02057	:	1
2646	013030	036041	36041	:	2
2647	013032	010400	10400	:	3
2648	013034	020202	20202	:	4
2649	013036	020417	20417	:	5
2650	013040	036410	36410	:	6
2651	013042	001740	01740	:	7
2652	013044	000037	00037	:	8
2653	013046	000200	00200	:	9
2654	013050	000004	00004	:	0
2655	013052	000000	00000	:	SPACE
2656	013054	000000	00000	:	1
2657	013056	010004	10004	:	2
2658	013060	010204	10204	:	3
2659	013062	000000	00000	:	4
2660	013064	024512	24512	:	5
2661	013066	076512	76512	:	6
2662	013070	025752	25752	:	7
2663	013072	035217	35217	:	8
2664	013074	011705	11705	:	9
2665	013076	005470	05470	:	0
2666	013100	046544	46544	:	1
2667	013102	010100	10100	:	2
2668	013104	004210	04210	:	3
2669	013106	000000	00000	:	4
2670	013110	004101	04101	:	5
2671	013112	004104	04104	:	6
2672	013114	010102	10102	:	7
2673	013116	020404	20404	:	8
2674	013120	010410	10410	:	9
2675	013122	010004	10004	:	0
2676	013124	035050	35050	:	1
2677	013126	010200	10200	:	2
2678	013130	010237	10237	:	3
2679	013132	004101	04101	:	4
2680	013134	000000	00000	:	5
2681	013136	000000	00000	:	6
2682	013140	000017	00017	:	7
2683	013142	000004	00004	:	8
2684	013144	000000	00000	:	9
2685	013146	010101	10101	:	0
2686	013150	001010	01010	:	1
2687	013152	047056	47056	:	2
2688	013154	075465	75465	:	3
2689	013156	010216	10216	:	4
2690	013160	010304	10304	:	5
2691	013162	004077	04077	:	6
2692	013164	035054	35054	:	7
2693	013166	041056	41056	:	8
2694	013170	035054	35054	:	9
2695	013172	076410	76410	:	0
2696	013174	020612	20612	:	1
2697	013176	041017	41017	:	2
2698	013200	076057	76057	:	3
2699	013202	037056	37056	:	4
2700	013204	034041	34041	:	5

2701	013206	010102	10102	:7
2702	013210	077010	77010	
2703	013212	043056	43056	:8
2704	013214	035056	35056	
2705	013216	041016	41016	:9
2706	013220	035077	35077	

.SBTTL ASCII MESSAGES

2710	013222	005015	051012	051505	PWRMSG: .ASCIZ <15><12><12>/RESTARTED AFTER POWER FAILURE/
2711	013230	040524	052122	042105	
2712	013236	040440	052106	051105	
2713	013244	050040	053517	051105	
2714	013252	043040	044501	052514	
2715	013260	042522	000		
2716	013263	015	005012	042115	MSG1: .ASCIZ <15><12><12>/MD-11-DZNCB-C GAMMA 11 EXERCISER/
2717	013270	030455	026461	055104	
2718	013276	041516	026502	020103	
2719	013304	020040	040507	046515	
2720	013312	020101	030461	042440	
2721	013320	042530	041522	051511	
2722	013326	051105	000		
2723	013331	015	042412	052116	MSG2: .ASCIZ <15><12>/ENTER THRESHOLD VALUE IN OCTAL - THEN RETURN =/
2724	013336	051105	052040	051110	
2725	013344	051505	047510	042114	
2726	013352	053040	046101	042525	
2727	013360	044440	020116	041517	
2728	013366	040524	020114	020055	
2729	013374	044124	047105	051040	
2730	013402	052105	051125	020116	
2731	013410	000075			
2732	013412	005015	052502	020123	MSG3: .ASCIZ <15><12>/BUS TIMEOUT ER - VSV01 DISPLAY/
2733	013420	044524	042515	052517	
2734	013426	020124	051105	026440	
2735	013434	053040	053123	030460	
2736	013442	042040	051511	046120	
2737	013450	054501	000		
2738	013453	015	041012	051525	MSG4: .ASCIZ <15><12>/BUS TIMEOUT ER - VTO1 DISPLAY/
2739	013460	052040	046511	047505	
2740	013466	052125	042440	020122	
2741	013474	020055	052126	030460	
2742	013502	042040	051511	046120	
2743	013510	054501	000		
2744	013513	015	041012	051525	MSG5: .ASCIZ <15><12>/BUS TIMEOUT ER - NC11/
2745	013520	052040	046511	047505	
2746	013526	052125	042440	020122	
2747	013534	020055	041516	030461	
2748	013542	000			
2749	013543	015	042412	052116	MSG6: .ASCIZ <15><12>/ENTER KEYBOARD COMMAND(S)/
2750	013550	051105	045440	054505	
2751	013556	047502	051101	020104	
2752	013564	047503	046515	047101	
2753	013572	024104	024523	000	
2754	013577	015	041012	051525	MSG7: .ASCIZ <15><12>/BUS TIMEOUT ER - AR11/
2755	013604	052040	046511	047505	
2756	013612	052125	042440	020122	

E05

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 56
DZNCB.P11 ASCII MESSAGES

2757 013620 020055 051101 030461
2758 013626 000
2759 000001

.END

ADUMMY	007105	1689	1694	1752*					
ARADR	001174	617*	1386						
ARBUF	001244	652*	1507	1512					
ARCHAN	001350	689*	1289*	1292*	1295*	1508			
ARCONV	006024	1290	1293	1296	1507*				
ARCSR	001242	651*	1369	1387	1393*	1508*	1509*	1510	
ASCIZ	006736	1719*							
ASCLP	006770	1726*	1735						
ASCRES	007072	1724	1751*						
ASCZSP	006732	1692	1718*						
AWRIT	006700	1118	1122	1691*					
BASX	001326	680*	1113*	1589*	1658	1673	1676*		
BASY	001330	681*	1114*	1593*	1659				
BELLEN	001354	691*	1253*	1254					
BIT0 =	000001	501*							
BIT00 =	000001	491*	501						
BIT01 =	000002	490*	500						
BIT02 =	000004	489*	499						
BIT03 =	000010	488*	498						
BIT04 =	000020	487*	497						
BIT05 =	000040	486*	496						
BIT06 =	000100	485*	495						
BIT07 =	000200	484*	494						
BIT08 =	000400	483*	493						
BIT09 =	001000	482*	492						
BIT1 =	000002	500*	1406						
BIT10 =	002000	481*							
BIT11 =	004000	480*							
BIT12 =	010000	479*							
BIT13 =	020000	478*							
BIT14 =	040000	477*							
BIT15 =	100000	476*							
BIT2 =	000004	499*							
BIT3 =	000010	498*							
BIT4 =	000020	497*							
BIT5 =	000040	496*	851						
BIT6 =	000100	495*							
BIT7 =	000200	494*							
BIT8 =	000400	493*	879	910					
BIT9 =	001000	492*							
BLANK	006674	1683*	1696						
B00B00	002064	790	793	796	830*				
BCX	002626	927*							
B0X1	003012	928	960*						
BPTVEC=	000014	508*							
BRAN	006462	1597	1633*						
BRANL	006470	1634*	1639						
CHANGE	002616	836	838	841	859	874	894	900	919*
CHARCT	001324	679*	986*	995*					
CHDN	003436	1022	1050	1059*					
CH72	006416	1601	1610*						
CH74	006430	1599	1612*						
CH75	006422	1600	1611*						
CH76	006410	1602	1603*						
CH77	006402	1603	1608*						
CKSWR =	104405	2273*							

CKVALU	003270	1014	1017*
CLALL =	000100	521*	
CLZ =	000020	520*	1421
COL	006672	1656*	1671* 1683*
COLK	006670	1656	1682*
COMSAV	001316	676*	884 1152 1412*
COPERL	001334	683*	1003* 1025
CR =	000015	416*	2099 2109
CRLF =	000200	417*	2070 2109
DCHAR	006106	1526*	1527 1585
DCONT	005646	935	942 1173 1180 1454*
DDISP =	177570	423*	563 712
DISBG	004462	1192	1216 1241*
DISBG1	004730	1242	1288*
DISDAT	002622	923*	
DISDON	003500	1002	1067*
DISLOP	003234	1011*	1061 1066 1091 1103
DISPLA	001142	563*	712* 720*
DISPRE	000174	531*	720
DISPY	005634	931	938 949 952 1089 1169 1176 1184 1187 1452*
DISPY1	005676	962	966 971 976 1199 1204 1209 1213 1221 1226 1231 1235 1282
		1318	1464* 1465 1565 1652
DISPY2	005726	1284	1320 1475* 1476
DIVCNT	007070	1723*	1734* 1750*
DIVIDE	007120	1728	1771*
DIVJUS	007126	1774*	1776
DIVLOO	007146	1780*	1789
DMATRX	003212	1008*	
DSWR =	177570	422*	562 711
DTYPE	001254	659*	1021 1027 1045 1107 1163 1265 1269 1303 1306 1351* 1374*
		1431	1583
DUMMY1	006724	1117*	1121* 1125* 1138* 1693 1715*
DUMMY2	006726	1126*	1139* 1691* 1716*
DVLQOP	003342	1031*	1041
DVT01	006304	1584	1587*
EMTVEC=	000030	511*	
END =	000200	2570	2574 2578 2580 2583 2585 2588*
ERASE	005536	842	860 919 1162 1431*
ERRVEC=	000004	504*	709 710* 721* 737* 743* 1343* 1354* 1371* 1376* 1392* 1396*
FREERN	001342	686*	782* 844* 1149
GAIN	001320	677*	747* 886* 914* 915 1141 1405
GET	006212	1056	1556*
GNS =	***** U	530	2266 2267 2268 2269 2271 2273 2274 2275 2276
GOCN	003376	1043	1045*
GSG	003166	990	995*
GTSWR =	104404	2271*	
HORIZ	001274	667*	980* 1060* 1062*
HT =	000011	414*	2068 2109 1670 1675
INCXY	001332	682*	1111* 1666
INTENS	001256	660*	1352* 1373*
INTLUT	001260	661*	1346
IOTVEC=	000020	509*	
JBOX	004160	1167*	
JBOX1	004306	1164	1195*
JBCX2	004400	1196	1219*
JTYPE	001346	688*	867* 1195 1241 1394*

NAME	ADDRESS	START	END	LEN	TYPE	UNIT	STATUS	REMARKS
K808UF	001304	671#	1595*	1638				
K8INT	006122	751	1534*					
K8UFF	001306	672#	784*	786	788*	951*	852	1250 1252* 1534* 1535* 1536
LARGE1	003150	989#	996					
LARGE5	003132	959	986#					
LDIMGE	005756	893	1489#					
LEV TAB	011366	1048	2323#					
LF =	000012	415#	2103	2109				
LISEN	001712	782#	856	1538				
LISN	001732	786#	787	833	840	843	866	880 885 888 891 903 911 916
L1	011456	1153	2323	2361#				
L10	011543	2332	2379#					
L11	011555	2333	2383#					
L12	011570	2334	2387#					
L13	011604	2335	2391#					
L14	011621	2336	2396#					
L15	011637	2337	2401#					
L16	011656	2338	2406#					
L17	011676	2339	2412#					
L18	011717	2340	2418#					
L19	011741	2341	2424#					
L2	011467	2324	2362#					
L20	011764	2342	2431#					
L21	012010	2343	2438#					
L22	012035	2344	2446#					
L23	012063	2345	2454#					
L24	012112	2346	2462#					
L25	012142	2347	2471#					
L26	012173	2348	2480#					
L27	012225	2349	2489#					
L28	012260	2350	2499#					
L29	012314	2351	2509#					
L3	011471	2325	2363#					
L30	012351	2352	2520#					
L31	012407	2353	2531#					
L32	012446	2354	2542#					
L4	011474	2326	2364#					
L5	011500	2327	2366#					
L6	011505	2328	2368#					
L7	011513	2329	2370#					
L8	011522	2330	2373#					
L9	011532	2331	2376#					
MACHER	006530	1633*	1635*	1636	1638	1644*		
MAPADR	001266	664#	955*	1071	1094*	1095	1102*	
MAPLO	003504	1024	1046	1071#				
MATRIX=	020000	522#	685	746	748	835	837	1145 1422 1424
MESS1	012506	1116	2558#					
MESS2	012512	1120	2560#					
MESS3	012516	1124	2562#					
MESS4	012522	1129	2564#					
MESS5	012526	1144	2566#					
MESS6	012532	1148	2568#					
MLOOP	003416	1054#	1058					
MOVAL	007062	1747#	1748					
MCVALL	007052	1738	1743	1745#				

[illegible]

RDLIN = 104407	2207	2275*											
RDOCT = 104410	871	897	2276*										
RESVEC= 000010	505*												
RLJL 007174	1791*	1793											
RMJUST 007206	1778*	1792*	1795*										
ROUTA 002104	901	835*											
ROUTB 002116	802	837*											
ROUTC 002130	803	839*											
ROUTD 002140	804	841*											
ROUTE 002144	805	842*											
ROUTF 002154	806	844*	1151										
ROUTG 002250	807	860*											
ROUTJ 002260	810	862*											
ROUTK 002302	811	867*											
ROUTL 002312	812	869*											
ROUTM 002342	813	875*											
ROUTN 002402	814	883*											
ROUTR 002420	818	886*											
ROUTS 002436	819	889*											
ROUTT 002454	820	892*											
ROUTU 002470	821	895*											
ROUTV 002520	822	901*											
ROUTW 002540	823	906*											
ROUTZ 002576	826	914*											
ROW 006666	1655*	1667*	1669*	1681*									
ROWCNT 001336	684*	1010*	1013*	1015*									
ROWK 006664	1655	1669	1680*										
RTABLE 002000	795	797	801*										
RO =%000000	426*	731*	733	734*	786*	789	791*	792	794*	795	797	847*	849*
	878*	873*	909*	910*	929*	932*	933	937*	939*	940	944*	945*	947
	951*	960*	963*	965*	972*	973	975*	987*	989	993*	1000*	1003	1054*
	1071*	1095*	1096	1099*	1100	1167*	1170*	1171	1175*	1177*	1178	1182*	1186*
	1188*	1189	1197*	1200*	1201	1203*	1210*	1212*	1219*	1222*	1223	1225*	1232*
	1234*	1273*	1275*	1276*	1277*	1309*	1311*	1312*	1313*	1329*	1331	1332*	1335*
	1333*	1339	1340*	1346*	1347	1348*	1349	1365*	1367	1368*	1386*	1388	1389*
	1422*	1423*	1424	1453	1466*	1468	1489*	1496*	1559*	1658*	1666*	1673*	1675*
	1676	1733	1745*	1747*	1772*	1781*	1800	1801*	1803	1810	1813*	1814*	1815
	1816*	1819*	1822*	1826*	1827*	1828*	1829*	1830*	2060	2061*	2062	2065*	2204
	2208*	2209	2212	2232*	2235*	2250	2251*	2252	2253*	2254*	2255*	2256*	2284
	2309*												
R1 =%000001	427*	732*	733*	735	848*	857*	930*	946*	950*	961*	967*	969	970*
	977*	978	988*	989	991	993	994	1025*	1032	1034	1037	1039*	1042
	1055*	1088*	1133*	1134	1168*	1183*	1185*	1198*	1205*	1206	1208*	1214*	1220*
	1227*	1228	1230*	1236*	1274*	1278*	1279*	1280*	1310*	1314*	1315*	1316*	1330*
	1331*	1333	1339*	1341	1366*	1367*	1369	1387*	1388*	1390	1457	1467*	1469
	1490*	1493*	1562*	1564*	1659*	1670*	1719*	1721	1722	1732	1773*	1780*	1785*
	2205	2210*	2218*	2220*	2222*	2225*	2228	2231*	2285	2308*			
R2 =%000002	428*	1011*	1017	1019*	1032	1037*	1042*	1131*	1135*	1139	1491*	1494*	1499*
	1556*	1557*	1558*	1559	1560*	1561*	1562	1563*	1564	1657*	1660*	1665*	1722*
	1730*	1731	1733*	1782	1784*	1787*	1791*	1802*	1803*	1804	1806*	1807	1808*
	1811*	1821	1823*	1824	2206	2211*	2219*	2221*	2223*	2229	2230*	2286	2307*
	429*	1026*	1029*	1040*	1132*	1134*	1138	1291	1294	1297	1492*	1493	1498*
R3 =%000003	1512*	1513*	1514	1518*	1520*	1528	1581*	1587	1591	1595	1612*	1613*	1614*
	1615*	1616*	1657	1665	1721*	1732*	1786*	1979	1981*	1982	1985*	1986	1993*
	1994	1996	2004	2008	2010*	2016	2018	2020*	2023*	2143	2152*	2158*	2159*
	2162*	2167*	2168*	2169	2178*	2287	2306*						

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 63
DZNCB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

R4	=%000004	430*	1023*	1030*	1031*	1036*	1044*	1047*	1049	1049*	1057*	1072	1074*	1075*
		1076*	1077*	1078*	1079*	1085	1580*	1581	1726*	1775*	1777*	1782	1794	2144
R5	=%000005	2146*	2147*	2148*	2149	2150*	2164	2166*	2174*	2177*	2298	2305*		
		431*	1048*	1556	1560	1576*	1578	1580	1724*	1725*	1727	1729*	1731*	1740
		1742	1744*	1747	1771*	1774*	1779	1779*	1788*	1820*	1821*	1822	2145	2151*
		2153*	2155*	2156*	2157*	2158	2176*	2289	2304*					
R6	=%000006	432*	434	697*	698*	699								
R7	=%000007	433*	435											
SCLCEL	003320	1020	1025*											
SELCTA	005112	724	876	907	1329*									
SELCTB	005262	726	901	1365*										
SELCTC	005356	862	1386*											
SHFTCT	006662	1654*	1663*	1679*										
SHFTK	006660	1654	1678*											
SKZE	007036	1740*	1741											
SP	=%000006	434*	701*	709*	717*	721	740	783*	872	898	1356	1378	1576	1577*
		1633	1634*	1640*	1719	1720*	1727*	1729	1745	1746*	1800*	1810*	1814	1816
		1984*	1885*	1886	1893*	1896*	1897*	1901*	1902*	1906	1909*	1913	1915	1917
		1918*	1925	1927	1929*	1930	1932*	1933*	1934*	1935*	1936*	1951*	1952*	1955*
		1956*	1957	1961*	1962*	1963	1966	1968	1970*	1979*	1980*	1985	1998	1992*
		1999	2003*	2022	2023	2024*	2025*	2026*	2060*	2061	2062*	2064	2065	2066*
		2068	2070	2072	2078	2080*	2082*	2090*	2094	2098	2099	2103	2135*	2136
		2137	2138*	2143*	2144*	2145*	2151	2176	2177	2178	2179*	2180*	2202*	2203*
		2204*	2205*	2206*	2208	2212*	2214	2216	2224*	2225	2227	2228*	2230	2231
		2232	2234	2250*	2251	2284*	2285*	2286*	2287*	2288*	2289*	2290*	2291	2299*
		2303	2304	2305	2306	2307	2308	2309	2314*					
STACK =	001100	409*	701	783										
START	001356	534	694*											
START1	001616	739	743*	861	2315									
STKLMT=	177774	420*												
SWR	001140	562*	699	711*	713	719*	1880	1917*	2290	2303*				
SWREG	000176	532*	719	1880	1893									
SW0	= 000001	473*												
SW00	= 000001	463*	473											
SW01	= 000002	462*	472											
SW02	= 000004	461*	471											
SW03	= 000010	460*	470											
SW04	= 000020	459*	469											
SW05	= 000040	458*	468											
SW06	= 000100	457*												

SW8	=	000400	465*
SW9	=	001000	464*
TABLE		012656	1616
TABLEX		001340	685*
TBITVE=		000014	506*
TEMCHR		006460	1596*
TEMPO		001246	656*
			1308*
TEMP1		001250	657*
TEMP2		001252	658*
TERM	=	000000	2557*
THHI		001314	675*
THLO		001312	674*
TIME		001344	687*
TIXBNO		006576	1661
TIXDIS		006532	1617
TIXDON		006646	1672
TIXL		006566	1660*
TIXWOK		006606	1664
TJOY		004146	864
TKVEC =		000060	513*
TOTSIZ		001322	678*
TPVEC =		000064	514*
TRAPVE=		000034	512*
TRTVEC=		000014	507*
TTYOUT		001310	673*
TYPCR		006154	832
TYPE =		104400	723
			1895
			2073
TYPO		006174	831
TYPC =		104401	1894
TYPON =		104403	2269*
TYPOS =		104402	2268*
VERT		001276	668*
VTADR		001172	613*
VTCR		001234	645*
VTEXCP		006340	1592
VTEXT		006374	1598
VTMADR		001170	609*
VTNOSP		006434	1604
VTPL		006256	1578*
VTVADR		001166	605*
VTVCHP		001216	632*
VTVCRG		001214	631*
VTVCSR		001222	637*
			1458*
VTVINT		001232	641*
VTVMAP		001224	638*
VTVPOS		001220	633*
VTVPX		001226	639*
VTVPX1		001230	640*
VTVSAY		001262	662*
VTWDON		006456	1579
VTWL		006264	1581*
VTWRIT		006246	1115

VTXDAC	001236	646#	1468*						
VTYDAC	001240	647#	1469*						
WD001	012536	2558	2570#						
WD002	012561	2560	2574#						
WD003	012604	2562	2578#						
WD004	012617	2564	2580#						
WD005	012636	2566	2583#						
WD006	012645	2568	2585#						
XREF	001300	669#	982*	1054	1059*	1063*			
XYAVE	007210	1261	1300	1800#					
XYBUF	007336	1806	1808	1811	1833#	1867			
XYBUFE=	007436	1804	1824	1866#					
XYBUFP	007436	1802	1807*	1867#					
YREF	001302	670#	983*	1055	1064*				
ZESUP	006730	1717#	1718*	1737	1739*				
\$AUTOB	001134	559#	1889	2037					
\$BDAOR	001122	554#							
\$BDDAT	001126	556#							
\$CHARC	010556	2075*	2085*	2092	2101*	2106#			
\$CKSWR	007440	1880#	2273						
\$CMTAG	001100	542#	696	697					
\$CM3 =	000000	572#							
\$CNTLG	010312	1891	2032#						
\$CNTLU	010305	1908	2006	2031#					
\$CRLF	001161	573#	1919	2011	2031	2074	2109	2241	
\$ERFLG	001103	545#							
\$ERMAX	001115	551#							
\$ERRPC	001116	552#							
\$ERRTB	001164	590#							
\$ERTTL	001112	549#							
\$FILLC	001156	570#	2078	2109					
\$FILLS	001155	569#	2109						
\$GDADR	001120	553#							
\$GDDAT	001124	555#							
\$GTSWR	007510	1892#	2271						
\$HD =	000003	404	405						
\$HIOCT	011146	2229*	2240#						
\$ICNT	001104	546#							
\$ILLUP	011360	2282	2298	2317#					
\$INTAG	001135	560#	1920	2037					
\$ITEMB	001114	550#							
\$LF	001162	574#	2021	2031	2109	2241			
\$LPADR	001106	547#							
\$LPERR	001110	548#							
\$MAIL =	***** U	723	2062						
\$MNEW	010330	1895	2035#						
\$MSWR	01037	1892	2033#						
\$NULL	001154	568#	2080	2109					
\$OCNT	011004	2142*	2171*	2184#					
\$OMODE	011006	2137*	2141*	2146	2149*	2160*	2186#		
\$PASS	001100	543#							
\$PWAD	011354	2315#							
\$PWADN	011214	705	2282#	2310					
\$PWARMG	011350	2313#							
\$PWRUP	011266	2292	2298#						
\$QUES	001160	572#	1938	2014	2031	2109	2238	2241	

[illegible]

ADC	1135														
ADD	1334	945	951	1009	1012	1059	1085	1102	1134	1186	1332	1338	1340	1348	1368
	1389	1559	1562	1564	1577	1608	1609	1611	1616	1634	1666	1670	1675	1720	1730
	1746	1779	1822	1909	1918	2066	2138	2148	2225						
ASL	1794	1031	1047	1076	1077	1078	1079	1269	1275	1276	1277	1278	1279	1280	1311
	1312	1313	1314	1315	1316	1563	1614	1615	1775	1780	1786	1836	1827	1828	1932
	1933	1934	2218	2220	2222	2254									
ASR	1049	1558													
BCC	725	737	793	1661											
BCCS	790	863	877	902	908										
BFC	787	796	934	941	948	1035	1046	1050	1073	1142	1150	1172	1179	1249	1270
	1307	1337	1350	1497	1579	1637	1672	1738	1741	1889	1916	1931	2000	2069	2104
	2165	2213													
BGT	1928	1969	2172	2215											
BHI	992	1071	1018	1020											
BHIS	990														
BIC	791	85	854	879	887	889	910	1099	1260	1272	1413	1444	1452	1454	1456
	1467	1535	1557	1561	1613	1830	1885	1902	1929	1956	1962	1970	2162	2224	
BIS	915	1405	1406	1441	1458	1479	1509	1936	2167	2168					
BISB	1814	1921													
BIT	1248	1349	1514												
BITB	2092														
BLO	1033	1805	1825	1995											
BLOS	1043	1993													
BLT	1926	1957	2083	2173	2217										
BMI	1108	1196	1242	1515	1582	1793									
BNE	700	714	736	850	853	858	928	969	974	979	996	1014	1022	1028	1041
	1058	1061	1066	1087	1097	1101	1137	1146	1164	1190	1202	1207	1224	1229	1251
	1255	1256	1302	1304	1334	1342	1370	1391	1425	1437	1439	1495	1516	1537	1584
	1588	1592	1639	1664	1668	1735	1789	1793	1881	1887	1907	1914	1921	1958	1964
	1987	1999	2005	2009	2019	2063	2071	2079	2093	2100	2163	2302			
BPL	964	1091	1211	1215	1233	1237	1247	1264	1432	1443	1456	1465	1476	1511	1527
	1548	1743	1748	1776	1883	1899	1954	1960	2057	2097	2161				
BR	716	739	833	882	905	913	936	943	953	1110	1174	1191	1192	1216	1258
	1267	1283	1285	1305	1319	1321	1358	1390	1399	1440	1500	1517	1519	1586	1590
	1594	1604	1618	1674	1910	1937	1939	1965	1998	2007	2013	2015	2059	2076	2096
	2095	2102	2139	2154	2175	2226	2239	2294	2318						
CLC	1038	1080	1353	1375	1395	1641	1790								
CLN	1642														
CLR	698	747	749	782	784	785	788	847	855	867	873	886	890	899	906
	957	982	1023	1030	1063	1092	1131	1132	1167	1197	1199	1219	1220	1252	1351
	1393	1414	1423	1434	1518	1691	1739	1771	1772	1773	1819	1820	1896	1897	1980
	2003	2152	2210	2211	2300										
CLRB	2010	2020	2075	2101	2235										
CMP	699	713	735	740	789	792	852	933	940	947	968	973	978	939	991
	1017	1032	1096	1100	1145	1171	1178	1189	1201	1206	1223	1228	1250	1333	1341
	1356	1369	1378	1390	1424	1536	1587	1591	1782	1804	1824	1890	1886	1906	1913
	1925	1927	1957	1963	1966	1968	1992	1994							
CMFB	1638	1740	1888	1920	1986	2004	2008	2018	2068	2070	2078	2099	2103	2214	2216
COM	1253	1271	1308												
COMB	1477														
DEC	849	857	963	967	975	995	1013	1029	1040	1057	1060	1065	1074	1090	1136
	1203	1208	1210	1214	1225	1230	1232	1236	1436	1438	1494	1496	1653	1667	1671
	1734	1788	1792	1993											
DECB	2082	2085	2160	2171											
EXT	410														

DZNCB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

HALT	530	730	742	2058	2293	2317								
INC	932	939	965	970	972	977	036	1044	1094	1170	1177	1188	1200	1205
	1222	1227	1234	1610	1635	1718	1744	1774	1785	1823	1935	2156	2174	2301
INCB	1292	1295	2105											
IOT	411													
JMP	534	797	836	838	840	841	843	856	859	861	864	866	868	874
	885	888	891	894	900	903	911	916	958	1002	1024	1067	1098	1103
	1153	1538												
JSR	724	726	745	831	832	839	842	845	846	860	862	976	883	892
	901	907	919	923	931	935	938	942	949	952	962	966	971	976
	1089	1115	1118	1119	1122	1123	1127	1128	1140	1143	1147	1162	1169	1173
	1180	1184	1187	1199	1204	1209	1213	1221	1226	1231	1235	1257	1261	1282
	1290	1293	1296	1300	1318	1320	1420	1545	1565	1585	1597	1617	1662	1692
	1728	1809	1912	1924	2077	2084	2091							
MOV	697	701	703	704	705	706	709	710	711	712	717	719	720	721
	732	733	737	743	746	748	750	751	752	753	793	786	830	835
	844	848	869	872	875	878	884	895	898	909	914	929	930	937
	946	950	954	955	956	960	961	980	981	983	986	987	988	993
	1008	1010	1011	1015	1025	1026	1048	1054	1055	1062	1071	1089	1093	1095
	1111	1112	1113	1114	1117	1121	1125	1126	1130	1133	1138	1139	1152	1158
	1182	1183	1185	1191	1243	1244	1245	1256	1259	1262	1281	1289	1291	1294
	1317	1329	1330	1331	1335	1339	1343	1346	1347	1352	1354	1365	1366	1367
	1372	1373	1374	1376	1386	1387	1388	1392	1394	1396	1404	1412	1421	1422
	1435	1445	1453	1457	1468	1469	1478	1489	1490	1491	1492	1493	1499	1508
	1520	1534	1544	1546	1549	1576	1580	1589	1596	1612	1633	1640	1654	1655
	1657	1658	1659	1665	1669	1673	1676	1719	1721	1722	1723	1724	1726	1727
	1732	1733	1745	1778	1800	1801	1802	1803	1806	1807	1808	1811	1815	1816
	1917	1922	1951	1952	1979	1981	1992	2023	2024	2025	2026	2060	2061	2065
	2135	2143	2144	2145	2151	2158	2176	2177	2178	2179	2180	2202	2203	2204
	2206	2208	2209	2228	2229	2230	2231	2232	2250	2251	2255	2282	2283	2284
	2286	2287	2288	2289	2290	2291	2292	2298	2299	2303	2304	2305	2306	2307
	2309	2310	2311	2314										
MOVB	1273	1274	1299	1309	1310	1528	1556	1560	1581	1595	1725	1731	1747	1810
	1901	1955	1961	1985	1990	1996	2001	2016	2062	2090	2098	2136	2137	2140
	2142	2146	2149	2150	2169	2212	2253							
NEG	2147													
RESET	744													
ROL	1781	1787	2153	2155	2156	2157	2159	2219	2221	2223				
ROR	1039	1081	1082	1083	1084	1660	1777	1791						
RTI	718	1539	1923	1971	2027	2067	2181	2233	2316					
RTS	1355	1377	1397	1407	1415	1426	1446	1459	1470	1480	1501	1521	1529	1550
	1619	1643	1677	1697	1749	1794	1817	1831	2107	2256				1566
SEC	1357	1379	1398											
SJ8	1000	1016	1019	1037	1042	1064	1498	1513	1593	1784				
SWAB	1075	1298	1813	1829										
TRAP	2258	2267	2268	2269	2271	2273	2274	2275	2276					
TST	738	755	927	994	1021	1027	1034	1045	1072	1086	1107	1141	1149	1163
	1241	1254	1263	1265	1269	1303	1306	1336	1344	1345	1431	1475	1507	1526
	1583	1737	1915	1930	1988	1999	2022	2064	2072	2094	2164	2227	2234	2252
TSTB	1246	1301	1442	1455	1464	1510	1547	1636	1742	1882	1898	1953	1959	2056
.ASCII	572	573	2570	2574	2578	2580	2583	2585						
.ASCIIZ	574	2031	2032	2033	2035	2710	2716	2723	2732	2738	2744	2749	2754	
.BLKB	2030													
.BYTE	544	545	550	551	559	560	568	569	570	571	1605	2028	2029	2182
	2184	2185	2361	2362	2363	2364	2366	2368	2370	2373	2376	2379	2383	2397
	2396	2401	2406	2412	2415	2418	2421	2424	2429	2431	2436	2438	2443	2446

	2454	2459	2462	2467	2471	2476	2490	2485	2489	2494	2499	2504	2509	2515	2520
.DSABL	2526	2531	2537	2542	2548										
.ENABL	1940														
.END	394	1873													
.ENDC	399														
	707	410	502	516	535	538	542	544	572	576	694	701	702	703	705
	1431	723	1324	1329	1361	1365	1383	1386	1402	1404	1410	1412	1418	1420	1429
	1542	1449	1452	1462	1464	1473	1475	1483	1489	1504	1507	1524	1526	1532	1534
	1771	1544	1553	1555	1569	1575	1623	1633	1647	1652	1686	1689	1700	1715	1756
	2014	1793	1800	1870	1873	1874	1876	1904	1940	1944	1972	1973	1981	1983	1986
	2254	2031	2037	2039	2042	2062	2110	2113	2188	2191	2197	2241	2242	2245	2251
	2293	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2281
.EQUIV	410	2291	2297	2303	2304	2314	2316	2320							
	492	411	419	434	435	464	465	466	467	468	463	470	471	472	473
.EVEN	1607	493	494	495	496	497	498	499	500	501					
.IF	395	1753	2037	2554	2587										
	705	408	474	502	533	537	541	543	572	575	576	694	696	701	703
	1428	707	723	1323	1328	1360	1364	1382	1385	1401	1403	1409	1411	1417	1419
	1533	1430	1448	1451	1461	1463	1472	1482	1488	1503	1506	1523	1525	1531	1531
	1755	1541	1543	1552	1555	1568	1574	1622	1632	1646	1651	1685	1688	1699	1714
	1986	1770	1797	1799	1869	1872	1874	1875	1876	1904	1943	1944	1972	1980	1982
	2244	1987	2030	2031	2037	2038	2041	2062	2109	2112	2187	2190	2193	2209	2241
	2290	2250	2254	2258	2267	2268	2269	2270	2271	2273	2274	2275	2276	2277	2280
.IFF	408	2291	2296	2303	2304	2312	2314	2316	2320	2324	2329	1361	1365	1383	1404
	1410	538	541	543	572	576	701	1324	1329	1361	1365	1383	1386	1402	1404
	1507	1412	1418	1420	1429	1431	1449	1452	1462	1464	1473	1475	1483	1489	1504
	1686	1524	1526	1532	1534	1542	1544	1553	1556	1569	1575	1623	1633	1647	1652
	1973	1689	1700	1715	1756	1771	1798	1800	1870	1873	1876	1944	1946	1951	1972
	2297	1982	2014	2030	2039	2042	2110	2113	2188	2191	2242	2245	2251	2278	2281
.IFT	1946	2312													
.IFTF	1891	1951	2214	2234	2241										
.IIF	394	1944	1947	2210	2218	2240									
	2267	399	404	405	530	575	702	1873	1894	2022	2031	2037	2109	2241	2256
.IRP	694	2268	2269	2271	2273	2274	2275	2276							
.LIST	394	2204	2230	2284	2290	2303	2304								
	2273	516	530	572	694	707	1972	2258	2266	2267	2268	2269	2270	2271	2272
.MACRO	535	2274	2275	2276	2277										
.MCALL	394	2258													
.NLIST	394	516	707												
	2273	516	530	572	694	707	1972	2258	2266	2267	2268	2269	2270	2271	2272
.PAGE	535	2274	2275	2276	2277										
.REM	1	576	619	654	692	755	917	1322	2320	2555	2589				
.REPT	530														
.SBTTL	406	1823													
	2320	524	533	535	576	692	695	1322	1870	2039	2110	2188	2242	2258	2279
.TITLE	394	2555	2589	2708											
.WORD	530														
	561	531	532	543	546	547	548	549	552	553	554	555	556	557	558
		562	563	2106	2186	2237	2240	2313	2315						

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

F06

MAINDEC-11-DZNCB-C GAMMA 11 EXERCISER MACY11 27(732) 21-SEP-76 15:32 PAGE 73
DZNCB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

*.DZNCB.SEO/SOL/CRF=DZNCB.P11
RUN-TIME: 32 22 4 SECONDS
RUN-TIME RATIO: 332/60=5.5
CORE USED: 19K (37 PAGES)

Spooler runtime 11 Seconds, 328 KCS, 297 disk reads, 3 disk writes, 70 pages.

Shooter Funtime 11 seconds, 568 KCS, 537 disk reads, 9 disk writes, 70 pages
 1974 Date 17-Oct-76 15:51:25 Modeler IPC-0 8978 (103) endrun

[illegible]