

AD11-K

PERFORMANCE TEST
MD-11-DZADL-B

EP-DZADL-B-DL-B
COPYRIGHT © 1976
FICHE 1 OF 1

DEC 1976
digital
MADE IN USA

This microfiche card contains a grid of frames on the left side, with a large blank area on the right. The frames are arranged in 10 rows and 8 columns. Each frame contains a small, high-contrast image, likely a test pattern or a document page. The images are too small to read clearly but appear to be variations of a document or test pattern. The right side of the card is a large, solid black area, which is the standard format for microfiche cards to allow for easy identification and handling.

801

MAINT-11-DZADL-B

MAINT-11-DZADL-B 27 665 2-DEC-76 16:29 PAGE :

.REM

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZADL-B
PRODUCT NAME: ADI1K PERFORMANCE TEST
DATE: DECEMBER 1976
MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT (C) 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 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 IN EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1.0 ABSTRACT

THIS DIAGNOSTIC HAS TWO STARTING ADDRESSES: 200 FOR STANDARD TOLERANCES AND 210 FOR TIGHTER OPTION TEST AREA TOLERANCES.

THIS DIAGNOSTIC TESTS THE AD11K WITH OR WITHOUT A WRAPAROUND MODULE (G5036).

WHEN STARTING THE DIAGNOSTIC, A SET OF TESTS IS LISTED AND THIS STATEMENT IS PRINTED OUT: "TYPE THE LETTER AND CARRIAGE RETURN OF THE DESIRED TEST:". THE FOLLOWING CHART INDICATES WHICH LETTER CORRESPONDS TO WHICH TEST:

- W: THE ENTIRE WRAPAROUND TEST (REQUIRES G5036 MODULE)
 - A. ANALOG SUBTESTS
 - B. NOISE TEST
 - C. INTERCHANNEL SETTling TEST
 - D. DIFFERENTIAL LINEARITY AND RELATIVE ACCURACY TEST
- C: CALIBRATION TEST ONLY
- N: NOISE TEST ONLY
- S: INTERCHANNEL SETTling ONLY
- L: LOGIC SUBTESTS ONLY
- A: AUTO TEST (REQUIRES G5036 MODULE)
 - A. LOGIC SUBTESTS
 - B. ANALOG SUBTESTS
 - C. NOISE TEST
 - D. INTERCHANNEL SETTling TEST
 - E. DIFFERENTIAL LINEARITY AND RELATIVE ACCURACY TEST

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 FAMILY COMPUTER WITH 8K OF MEMORY
TELETYPE
AD11K MODULE
VT55 TERMINAL SUPPORTED FOR GRAPHIC OUTPUT
G5036 WRAPAROUND MODULE

2.2 STORAGE

THIS PROGRAM USES ALL BK OF MEMORY AND IS NOT "CHAINABLE" ON AN 8K CPU. THE PROGRAM IS "CHAINABLE" ON 12K OR GREATER. THE PROGRAM WILL DESTROY "ABSOLUTE LOADER" ON AN 8K CPU, IF "W" OR "A" IS SELECTED.

3.C LOADING PROCEDURE

PROCEDURE FOR LOADING NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4.3 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

STANDARD POP-11 FORMAT

SW15=1	HALT ON ERROR
SW14=1	LOOP ON TEST
SW13=1	INHIBIT ERROR TYPEOUTS
SW12=1	HALT FOR VTES DISPLAY
SW11=1	INHIBIT ITERATIONS
SW10=1	BELL ON ERROR
SW9 =1	LOOP ON ERROR
SW8 =1	LOOP ON TEST IN SWR <7:0>

200 IS THE STARTING ADDRESS OF THE DIAGNOSTIC FOR STANDARD TOLERANCES. 204 IS THE RESTART ADDRESS. 210 IS THE STARTING ADDRESS OF THE DIAGNOSTIC FOR THE OPTION TEST AREA'S TIGHTER TOLERANCES.

5.0 OPERATING PROCEDURE

START THE DIAGNOSTIC AT 200 OR 210. THE PROGRAM HEADING AND THE LIST OF TESTS AVAILABLE, WILL BE PRINTED OUT FOLLOWED BY A MESSAGE "TYPE THE LETTER AND CARRIAGE RETURN FOR THE DESIRED

E01

MAINDEC-11-DZADL-B
DZADLB.P11

MACY:1 27(665) 2-DEC-76 16:29 PAGE 4

161
162

TEST:". THEN TYPE THE LETTER YOU WANT, ACCORDING TO THE TABLE
LISTED AND HIT CARRIAGE RETURN.

163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203

TWO CONTROL CHARACTERS, $\uparrow$ A AND $\uparrow$ C, ARE SET ASIDE FOR INTERRUPTING A TEST AND TRANSFERRING CONTROL TO EITHER THE BEGINNING OF THE DIAGNOSTIC ($\uparrow$ C) OR TO THE BEGINNING OF THE SPECIFIC TEST WHICH WAS IN PROGRESS ($\uparrow$ A). DURING THE LOGIC TESTS WHILE A RESET IS BEING PERFORMED, $\uparrow$ C OR $\uparrow$ A WILL NOT BE EXECUTED UNTIL AFTER THE RESET HAS BEEN COMPLETED, THEREFORE HIT $\uparrow$ C OR $\uparrow$ A UNTIL IT IS SUCCESSFUL.

FOR MACHINES WITHOUT A HARDWARE SWITCH REGISTER, LOCATION SWREG (176) IS USED AS A SOFTWARE SWITCH REGISTER. TO MODIFY THE CONTENTS OF SWREG, TYPE $\uparrow$ G. THE PROGRAM RESPONDS WITH THE CURRENT CONTENTS OF SWREG AND A SLASH. TYPE THE DESIRED NEW CONTENTS OF SWREG FOLLOWED BY A CARRIAGE RETURN.

IF "W" IS TYPED, THE PROGRAM WILL TYPE "XX AD11K'S FOUND". WHERE XX IS THE NUMBER OF AD11K'S IN OCTAL. IF THE NUMBER IS GREATER THAN 1, THE TEST WILL BE RUN SUCCESSIVELY ON EACH AD11K. THE PROGRAM WILL RUN THROUGH THE LOGIC SUBTESTS, THE NOISE TEST ON 8 EDGES, THE INTERCHANNEL SETTLING TEST ON 8 EDGES, AND THE DIFFERENTIAL LINEARITY AND RELATIVE ACCURACY TEST. A G5036 WRAPAROUND MODULE IS REQUIRED. THE PROGRAM SUPPORTS AD11K EXPANSION BEYOND 16 CHANNELS. TO RUN THIS TEST ON A GROUP OF CHANNELS OTHER THAN 0-17, LOAD 20, 40, OR 60 INTO LOCATION BASECH (1336) FOR CHANNELS 20-37, 40-57, 60-77.

IF "C" IS TYPED, THE PROGRAM WILL RUN THE CALIBRATION TEST AND WILL LOOP ON THAT TEST UNTIL THE OPERATOR HALTS IT. IF A CERTAIN AD11K IS TO BE TESTED, ITS STATUS REGISTER ADDRESS MUST BE LOADED INTO \$BASE (1250), AND ITS VECTOR ADDRESS MUST BE LOADED INTO THE LOW BYTE OF \$VECT1 (1244) (THE HIGH BYTE CONTAINING THE PRIORITY).

IF "N" IS TYPED, THE PROGRAM WILL RUN THE NOISE TEST TAGGED "BEGINN" AND WILL LOOP ON THIS TEST UNTIL THE OPERATOR HALTS IT. IF A CERTAIN AD11K IS TO BE TESTED ITS STATUS REGISTER ADDRESS MUST BE LOADED INTO \$BASE (1250), AND ITS VECTOR ADDRESS MUST BE LOADED INTO THE LOW BYTE OF \$VECT1 (1244) (THE HIGH BYTE CONTAINING THE PRIORITY).

GO1

MAINDEC-11-DZADL-B
DZADLB.P11

MACY11 27(665) 2-DEC-76 16:29 PAGE 6

204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231

IF "S" IS TYPED, THE PROGRAM WILL RUN THE INTERCHANNEL SETTling TEST TAGGED "BEGINS" AND WILL LOOP ON THIS TEST UNTIL THE OPERATOR HALTS IT. AT THE BEGINNING OF THIS TEST, THE OPERATOR MUST RESPOND TO THE STATEMENTS ASKING FOR THE "FROM" CHANNEL AND THE "TO" CHANNEL BY TYPING IN THE CHANNEL VALUE IN OCTAL AND HITTING CARRIAGE RETURN. IF A CERTAIN AD11K IS TO BE TESTED ITS STATUS REGISTER ADDRESS MUST BE LOADED INTO \$BASE (1250), AND ITS VECTOR ADDRESS MUST BE LOADED INTO \$VECT1 (1244) (THE HIGH BYTE CONTAINING THE PRIORITY).

IF "A" IS TYPED, THE PROGRAM WILL EXECUTE THE LOGIC TESTS, ANALOG TESTS, NOISE, SETTLE AND DIFFERENTIAL LINEARITY. AT THE BEGINNING OF THE TEST THE PROGRAM WILL TYPE "XX AD11K'S FOUND". WHERE XX IS THE NUMBER OF AD11K'S IN OCTAL IF THE NUMBER IS GREATER THAN 1, THE TEST WILL BE RUN SUCCESSIVELY ON EACH AD11K. THE PROGRAM SUPPORTS AD11K EXPANSION BEYOND 16. CHANNELS. TO RUN THIS TEST ON A GROUP OF CHANNELS OTHER THAN 0-17, LOAD 20,40, OR 60 INTO LOCATION BASECH (1336) FOR CHANNELS 20-37, 40-57, 60-77.

IF "L" IS TYPED, THE PROGRAM WILL EXECUTE THE LOGIC TESTS, PRINTING "END PASS" WHEN IT HAS COMPLETED AN ENTIRE PASS. AT THE BEGINNING OF THE TEST THE PROGRAM WILL TYPE "XX AD11K'S FOUND". WHERE XX IS THE NUMBER OF AD11K'S IN OCTAL IF THE NUMBER IS GREATER THAN 1, THE TEST WILL BE RUN SUCCESSIVELY ON EACH AD11K.

2
1

6.0 ERRORS

THIS PROGRAM USES THE DIAGNOSTIC "SYSMAC" PACKAGE FOR ERROR REPORTING AND TYPEOUT. THE ERROR INFORMATION CONSISTS OF THE FOLLOWING:

ERRPC: LOCATION AT WHICH AN ERROR WAS DETECTED.
STREG: ADDRESS OF THE STATUS REGISTER.
ADBUFF: ADDRESS OF THE BUFFER
CHANL: CHANNEL VALUE
NOMINAL: EXPECTED CORRECT DATA
TOLERANCE: THE ACCEPTABLE DEVIATION FROM THE NOMINAL
ACTUAL: ACTUAL DATA
EXPECTED: EXPECTED CORRECT DATA

7.0 MISCELLANEOUS

7.1 EXECUTION TIME

EXECUTION TIME FOR EACH OF THE TESTS IS:

CALIBRATION:	8 CONVERSIONS/5 SECONDS @ 110 BAUD
WRAPAROUND TEST:	17 MINUTES FIRST PASS; 35 MINUTES FOR SUCCESSIVE PASSES
SETTLING TEST:	1 MINUTE
NOISE TEST:	1 MINUTE
LOGIC TEST:	1 MINUTE
AUTO TEST:	18 MINUTES FIRST PASS, 36 MINUTES FOR SUCCESSIVE PASSES

7.2 STATUS REGISTER AND VECTOR ADDRESSES AND PRIORITY

WHEN TESTING MORE THAN ONE AD11K, THE DIFFERENCE IN ADDRESSES IS PRESENTLY 40 FOR BUS ADDRESS AND VECTOR ADDRESS. THESE VALUES ARE IN VADR (BUS ADDRESS) (1332) AND VVCT (VECTOR ADDRESS) (1334). THE FIRST AD11K'S STATUS REGISTER ADDRESS MUST BE IN \$BASE (1250), ITS VECTOR ADDRESS MUST BE IN THE LOW BYTE OF \$VECT1 (1244), AND THE PRIORITY MUST BE IN THE HIGH BYTE OF \$VECT1.

7.3 AD11K PRIORITY

IF AD11K IS SET FOR A PRIORITY OTHER THAN 6, THE HIGH BYTE OF \$VECT1 (1244) MUST BE ADJUSTED ACCORDINGLY (THE LOW BYTE

I01

MAINDEC-11-02ADL-B
02ADLB.P11

MACY11 27(665) 2-DEC-76 16:29 PAGE 8

296
287

CONTAINING THE VECTOR ADDRESS). IF MORE THAN ONE AD11K IS BEING
TESTED, ALL MUST BE SET AT THE SAME PRIORITY.

7.4 SWITCH REGISTER

IF A HARDWARE SWITCH REGISTER IS PRESENT AND THE OPERATOR DESIRES TO USE A SOFTWARE SWITCH REGISTER AND THE IG FEATURE; IT IS NECESSARY TO LOAD THE STARTING ADDRESS, SET THE HARDWARE SWITCH REGISTER TO ALL ONES (-1), AND HIT START. THE PROGRAM WILL THEN RUN WITH THE SOFTWARE SWITCH REGISTER.

7.5 VT55 GRAPHIC OUTPUT

THE SCREEN DISPLAY MAY BE HALTED FOR EXAMINATION BY SETTING BIT 12. AND THEN JUST HIT CONTINUE 'O' COMPLETE THE PROGRAM'S EXECUTION.

8.0 RESTRICTIONS

8.1 A G5036 WRAPAROUND MODULE MUST BE PRESENT WHEN RUNNING THE AUTO TEST AND THE WRAPAROUND TEST.

SWITCH ON G5036 MUST BE IN 'O' POSITION

THE WRAPAROUND (G5036) MODULE MUST BE CONNECTED AS FOLLOWS:
AD11K TO BC08R CONNECTION A-A, VV-VV
BC08R TO G5036 CONNECTION "UPSIDE-DOWN" A-VV, VV-A

9.0 PROGRAM DESCRIPTION

9.1 LOGIC TESTS

THESE 14 LOGIC SUBTESTS RUN SEQUENTIALLY WITHOUT FURTHER OPERATOR INTERVENTION AFTER HE/SHE HAS TYPED IN THE NUMBER OF AD11K'S TO BE TESTED. ITS PURPOSE IS TO CHECK THAT EACH OF THE MUX BITS CAN BE LOADED AND PROPERLY READ BACK; THAT INITIALIZE CLEARS THE EXTERNAL START ENABLE BIT, THE DONE BIT, THE INTERRUPT ENABLE BIT, THE OVERFLOW BIT, THE ERROR FLAG, AND THE A/D START BIT. IT ALSO CHECKS THAT THE A/D DONE FLAG SETS AT END OF CONVERSION AND CLEARS WHEN THE CONVERTED VALUE IS READ. IT CHECKS THE INTERRUPT

K01

MAINDEC-11-DZADL-B
DZADLB.P11

MACY11 27(665) 2-DEC-76 16:29 PAGE 10

342

LOGIC AND THE CORRECT SETTING OF THE ERROR FLAG.

9.2 CALIBRATION TEST

THIS TEST BEGINS WHEN THE OPERATOR TYPES "C". IT THEN LOADS THE CHANNEL FROM THE SWITCH REGISTER BITS 0-7 AND DOES A CONVERSION ON THAT CHANNEL. IF SWR BIT 13 IS DOWN, IT PRINTS OUT THE CONVERTED VALUE ON THE TELETYPE; OTHERWISE, IF SWR BIT 13 IS UP, IT PUTS THE CONVERTED VALUE IN THE DISPLAY REGISTER. THE OPERATOR MAY CHANGE THE CHANNEL AT ANY TIME DURING THE TEST, HOWEVER THE NEW VALUES FROM THE NEW CHANNEL WILL NOT BE PRINTED UNTIL THE NEXT LINE OF 8 VALUES IS PRINTED. THE 8 VALUES ON EACH LINE CORRESPOND TO ONLY ONE CHANNEL.

9.3 DIFFERENTIAL LINEARITY

THIS TEST IS TO DETERMINE IF A CHANGE IN THE INPUT VOLTAGE REPRESENTS A SIMILAR CHANGE IN THE RESULTING CONVERTED BINARY VALUE.

9.4 SETTLING TEST

THE PURPOSE OF THIS TEST IS TO CHECK THAT THE TIME NEEDED TO SETTLE AND CORRECTLY REPORT A NEW INPUT VALUE AFTER SWITCHING CHANNELS DOES NOT EXCEED THE EXPECTED AMOUNT OF TIME FOR SUCH A CHANGE.

9.5 NOISE TEST

THIS TEST MEASURES THE INTERNAL SHORT-TERM REPEATABILITY NOISE WITHIN THE A/D. R'S NOISE EQUALS 1 STANDARD DEVIATION OF THE GAUSSIAN CURVE. PEAK NOISE EQUALS 2.3 STANDARD DEVIATION OF THE GAUSSIAN CURVE.

9.6 ANALOG TESTS

THESE 11 SUBTESTS CHECK THE CHANNELS AND THEIR OUTPUT.

```

390      .TITLE MAINDEC-11-DZADL-B
391      ;*COPYRIGHT (C) 1976
392      ;*DIGITAL EQUIPMENT CORP.
393      ;*MAYNARD, MASS. 01754
394      ;*
395      ;*PROGRAM BY VERA BREUER
396      ;*
397      ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
398      ;*PACKAGE (MAINDEC-11-DZQAC-CO), MAR 21, 1976.
399      ;*
400      .SBTTL BASIC DEFINITIONS
401
402      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
403      STACK= 1100
404      001100
405      .EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
406      .EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
407
408      ;*MISCELLANEOUS DEFINITIONS
409      HT= 11                ;;CODE FOR HORIZONTAL TAB
410      LF= 12                ;;CODE FOR LINE FEED
411      CR= 15                ;;CODE FOR CARRIAGE RETURN
412      CRLF= 200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
413      PS= 177776            ;;PROCESSOR STATUS WORD
414      .EQUIV PS,PSW
415      STKLM= 177774          ;;STACK LIMIT REGISTER
416      PIRQ= 177772           ;;PROGRAM INTERRUPT REQUEST REGISTER
417      DSWR= 177570          ;;HARDWARE SWITCH REGISTER
418      DDISP= 177570         ;;HARDWARE DISPLAY REGISTER
419
420      ;*GENERAL PURPOSE REGISTER DEFINITIONS
421      R0= %0                ;;GENERAL REGISTER
422      R1= %1                ;;GENERAL REGISTER
423      R2= %2                ;;GENERAL REGISTER
424      R3= %3                ;;GENERAL REGISTER
425      R4= %4                ;;GENERAL REGISTER
426      R5= %5                ;;GENERAL REGISTER
427      R6= %6                ;;GENERAL REGISTER
428      R7= %7                ;;GENERAL REGISTER
429      .EQUIV R6,SP           ;;STACK POINTER
430      .EQUIV R7,PC           ;;PROGRAM COUNTER
431
432      ;*PRIORITY LEVEL DEFINITIONS
433      PR0= 0                ;;PRIORITY LEVEL 0
434      PR1= 40               ;;PRIORITY LEVEL 1
435      PR2= 100              ;;PRIORITY LEVEL 2
436      PR3= 140              ;;PRIORITY LEVEL 3
437      PR4= 200              ;;PRIORITY LEVEL 4
438      PR5= 240              ;;PRIORITY LEVEL 5
439      PR6= 300              ;;PRIORITY LEVEL 6
440      PR7= 340              ;;PRIORITY LEVEL 7
441
442      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
443      SW15= 100000
444      SW14= 40000

```

444	020000	SW13=	20000
445	010000	SW12=	10000
446	004000	SW11=	4000
447	002000	SW10=	2000
448	001000	SW09=	1000
449	000400	SW08=	400
450	000200	SW07=	200
451	000100	SW06=	100
452	000040	SW05=	40
453	000020	SW04=	20
454	000010	SW03=	10
455	000004	SW02=	4
456	000002	SW01=	2
457	000001	SW00=	1
458		.EQUIV	SW09,SW9
459		.EQUIV	SW08,SW8
460		.EQUIV	SW07,SW7
461		.EQUIV	SW06,SW6
462		.EQUIV	SW05,SW5
463		.EQUIV	SW04,SW4
464		.EQUIV	SW03,SW3
465		.EQUIV	SW02,SW2
466		.EQUIV	SW01,SW1
467		.EQUIV	SW00,SW0

468		; *DATA BIT DEFINITIONS (BIT00 TO BIT15)	
469	100000	BIT15=	100000
470	040000	BIT14=	40000
471	020000	BIT13=	20000
472	010000	BIT12=	10000
473	004000	BIT11=	4000
474	002000	BIT10=	2000
475	001000	BIT09=	1000
476	000400	BIT08=	400
477	000200	BIT07=	200
478	000100	BIT06=	100
479	000040	BIT05=	40
480	000020	BIT04=	20
481	000010	BIT03=	10
482	000004	BIT02=	4
483	000002	BIT01=	2
484	000001	BIT00=	1
485		.EQUIV	BIT09,BIT9
486		.EQUIV	BIT08,BIT8
487		.EQUIV	BIT07,BIT7
488		.EQUIV	BIT06,BIT6
489		.EQUIV	BIT05,BIT5
490		.EQUIV	BIT04,BIT4
491		.EQUIV	BIT03,BIT3
492		.EQUIV	BIT02,BIT2
493		.EQUIV	BIT01,BIT1
494		.EQUIV	BIT00,BIT0

; *BASIC "CPU" TRAP VECTOR ADDRESSES

MINOEC-11-02ADL-B MACV: 27.665) 2-DEC-76 16:29 PAGE 14
 02ADL.B.P11 BASIC DEFINITIONS

```

498      000004      ERRVEC= 4      :: TIME OUT AND OTHER ERRORS
499      000010      RESVEC= 10     :: RESERVED AND ILLEGAL INSTRUCTIONS
500      000014      TBITVEC= 14    :: "T" BIT
501      000014      TRIVEC= 14     :: TRACE TRAP
502      000014      BPTVEC= 14     :: BREAKPOINT TRAP (BPT)
503      000020      IOTVEC= 20      :: INPUT/OUTPUT TRAP (IOT) **SCOPE**
504      000024      PWRVEC= 24      :: POWER FAIL
505      000030      EMTVEC= 30      :: EMULATOR TRAP (EMT) **ERROR**
506      000034      TRAPVEC= 34     :: "TRAP" TRAP
507      000060      TKVEC= 60       :: TTY KEYBOARD VECTOR
508      000064      TPVEC= 64      :: TTY PRINTER VECTOR
509      000240      PIRQVEC= 240    :: PROGRAM INTERRUPT REQUEST VECTOR
510      .SBTTL      OPERATIONAL SWITCH SETTINGS
511      :
512      :          SWITCH          USE
513      :          -----
514      :          15      HALT ON ERROR
515      :          14      LOOP ON TEST
516      :          13      INHIBIT ERROR TYPEOUTS
517      :          12      HALT FOR VTSS DISPLAY
518      :          11      INHIBIT ITERATIONS
519      :          10      BELL ON ERROR
520      :          9       LOOP ON ERROR
521      :          8       LOOP ON TEST IN SWR 7:0
522      170400      ABASE= 170400
523      140340      AVECT1= 140340
524      000300      APRIOP= 300
525      .SBTTL      TRAP CATCHER
526      :
527      :          =0
528      :          *ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
529      :          *SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
530      :          *LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
531      :          =174
532      000174      000000      DISPREG: .WORD 0      :: SOFTWARE DISPLAY REGISTER
533      000176      000000      SWREG: .WORD 0      :: SOFTWARE SWITCH REGISTER
534      .SBTTL      STARTING ADDRESS(ES)
535      JMP      @BEGIN      :: JUMP TO STARTING ADDRESS OF PROGRAM
536      JMP      @BEG2      :: RESTART ADDRESS
537      JMP      @BEGIN2     :: START ADDRESS FOR OPTION TEST AREA
538      000200      000137      001620
539      000204      000137      002256
540      000210      000137      001626

```


MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 15
 DZADL.B.P11 ACT11 HOOKS

539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571

```
.SBTTL ACT11 HOOKS

:*****
:HOOKS REQUIRED BY ACT11
      $SVPC=.          ;SAVE PC
      =46
      $ENDAD          ::1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SECF
      =52
      .WORD 0          ::2)SET LOC.52 TO ZERO
      = $SVPC          :: RESTORE PC
.=100C
.SBTTL APT PARAMETER BLOCK

:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
      .SX=.          ::SAVE CURRENT LOCATION
      =24          ::SET POWER FAIL TO POINT TO START OF PROGRAM
      200          ::FOR APT START UP
      =44          ::POINT TO APT INDIRECT ADDRESS PNTR.
      $APTHDR       ::POINT TO APT HEADER BLOCK
      =.SX          ::RESET LOCATION COUNTER
:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.

$APTHD:
$MIBTS: .WORD 0          ::TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBAOR: .WORD $MAIL      ::ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM: .WORD 1200.       ::RUN TIM OF LONGEST TEST
$PASTM: .WORD 500.       ::RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 1700.      ::ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
      .WORD $ETEND-$MAIL/2 ::LENGTH MAILBOX-ETABLE(WORDS)
```

572			
573			
574			
575			
576			
577			
578			
579			
580	001100	001100	
581	001100	000000	
582	001102	000	
583	001103	000	
584	001104	000000	
585	001106	000000	
586	001110	000000	
587	001112	000000	
588	001114	000	
589	001115	001	
590	001116	000000	
591	001120	000000	
592	001122	000000	
593	001124	000000	
594	001126	000000	
595	001130	000000	
596	001132	000000	
597	001134	000	
598	001135	000	
599	001136	000000	
600	001140	177570	
601	001142	177570	
602	001144	177560	
603	001146	177562	
604	001150	177564	
605	001152	177566	
606	001154	000	
607	001155	002	
608	001156	012	
609	001157	000	
610	001160	000000	
611	001162	000000	
612	001164	177607	000377
613	001170	077	
614	001171	015	
615	001172	000012	
616			
617			
618			
619			
620	001174		
621	001174	000000	
622	001176	000000	
623	001200	000000	
624	001202	000000	
625	001204	000000	

```

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

```

STSTNM:	.BYTE	00
SERFLG:	.BYTE	00
SICNT:	.WORD	00
SLPADR:	.WORD	00
SLPERR:	.WORD	00
SERTTL:	.WORD	00
SITEMB:	.BYTE	00
SERMAX:	.BYTE	01
SERRPC:	.WORD	00
SGDADR:	.WORD	00
SBDADR:	.WORD	00
SGDDAT:	.WORD	00
SBDDAT:	.WORD	00
	.WORD	00
	.WORD	00
SAUTOB:	.BYTE	00
SINTAG:	.BYTE	00
	.WORD	00
SWR:	.WORD	DSWR
DISPLAY:	.WORD	DDISP
STKS:	177560	
STKB:	177562	
STPS:	177564	
STPB:	177566	
SNUL:	.BYTE	00
SFILLS:	.BYTE	20
SFILLC:	.BYTE	12
STPFLG:	.BYTE	00
STIMES:	.CO	
SESCAPE:	.CO	
SBELL:	.ASCIIZ	(207)
SGUES:	.ASCII	"/
SCALF:	.ASCII	(15)
SLE:	.ASCIIZ	(12)

```

::CONTAINS THE TEST NUMBER
::CONTAINS ERROR FLAG
::CONTAINS SUBTEST ITERATION COUNT
::CONTAINS SCOPE LOOP ADDRESS
::CONTAINS SCOPE RETURN FOR ERRORS
::CONTAINS TOTAL ERRORS DETECTED
::CONTAINS ITEM CONTROL BYTE
::CONTAINS MAX. ERRORS PER TEST
::CONTAINS PC OF LAST ERROR INSTRUCTION
::CONTAINS ADDRESS OF 'GOOD' DATA
::CONTAINS ADDRESS OF 'BAD' DATA
::CONTAINS 'GOOD' DATA
::CONTAINS 'BAD' DATA
::RESERVED--NOT TO BE USED

::AUTOMATIC MODE INDICATOR
::MANUAL MODE INDICATOR

::ADDRESS OF SWITCH REGISTER
::ADDRESS OF DISPLAY REGISTER
::TTY KBD STATUS
::TTY KBD BUFFER
::TTY PRINTER STATUS REG. ADDRESS
::TTY PRINTER BUFFER REG. ADDRESS
::CONTAINS NULL CHARACTER FOR FILLS
::CONTAINS # OF FILLER CHARACTERS REQUIRED
::INSERT FILL CHARS. AFTER A "LINE FEED"
::"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
::MAX. NUMBER OF ITERATIONS
::ESCAPE ON ERROR ADDRESS
::CODE FOR BELL
::QUESTION MARK
::CARRIAGE RETURN
::LINE FEED

```

SBTIL APT MAILBOX-ETABLE

EVEN

```

:..EVEN
$MAIL:                                ::APT MAILBOX
$MSGTY: .WORD      MSGTY             ::MESSAGE TYPE CODE
$FATAL: .WORD      AFATAL            ::FATAL ERROR NUMBER
$TESTN: .WORD      ATESTN           ::TEST NUMBER
$PASS:  .WORD      APASS             ::PASS COUNT
$DEVT:  .WORD      ADEVCT           ::DEVICE COUNT

```

E02

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 17
 DZADLB.P11 APT MAILBOX-ETABLE

626	001206	000000	\$UNIT:	.WORD	AUNIT	:: I O UNIT NUMBER
627	001210	000000	\$MSGAD:	.WORD	AMSGAD	:: MESSAGE ADDRESS
628	001212	000000	\$MSGLG:	.WORD	AMSGLG	:: MESSAGE LENGTH
629	001214		\$ETABLE:			:: APT ENVIRONMENT TABLE
630	001214	000	\$ENV:	.BYTE	AENV	:: ENVIRONMENT BYTE
631	001215	000	\$ENVM:	.BYTE	AENVM	:: ENVIRONMENT MODE BITS
632	001216	000000	\$SWREG:	.WORD	ASWREG	:: APT SWITCH REGISTER
633	001220	000000	\$USWR:	.WORD	AUSWR	:: USER SWITCHES
634	001222	000000	\$CPUOP:	.WORD	ACPUOP	:: CPU TYPE, OPTIONS
635			::			BITS 15-11=CPU TYPE
636			::			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
637			::			11/70=06, PDG=07, G=10
638			::			BIT 10=REAL TIME CLOCK
639			::			BIT 9=FLOATING POINT PROCESSOR
640			::			BIT 8=MEMORY MANAGEMENT
641	001224	000	\$MAMS1:	.BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
642	001225	000	\$MTYP1:	.BYTE	AMTYP1	:: MEM. TYPE, BLK#1
643			::			MEM. TYPE BYTE -- (HIGH BYTE)
644			::			900 NSEC CORE=001
645			::			300 NSEC BIPOLAR=002
646			::			500 NSEC MOS=003
647	001226	000000	\$MADR1:	.WORD	AMADR1	:: HIGH ADDRESS, BLK#1
648			::			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
649	001230	000	\$MAMS2:	.BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
650	001231	000	\$MTYP2:	.BYTE	AMTYP2	:: MEM. TYPE, BLK#2
651	001232	000000	\$MADR2:	.WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
652	001234	000	\$MAMS3:	.BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
653	001235	000	\$MTYP3:	.BYTE	AMTYP3	:: MEM. TYPE, BLK#3
654	001236	000000	\$MADR3:	.WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
655	001240	000	\$MAMS4:	.BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
656	001241	000	\$MTYP4:	.BYTE	AMTYP4	:: MEM. TYPE, BLK#4
657	001242	000000	\$MADR4:	.WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
658	001244	140340	\$VECT1:	.WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
659	001246	000000	\$VECT2:	.WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
660	001250	170400	\$BASE:	.WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
661	001252	000000	\$DEVN:	.WORD	ADEVN	:: DEVICE MAP
662	001254	000000	\$CDW1:	.WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
663	001256		\$ETEND:			
664			.MEXIT			

MAINDEC-11-DZADL-B MACY11 27(605) 2-DEC-76 16:29 PAGE 18
 DZADLB.P11 ERROR POINTER TABLE

.SETTL 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:

:ITEM 1
 EM1 :STATUS REG. ERROR
 DH1 :ERRPC STREG EXPECTED ACTUAL
 DT1 :\$ERRPC, STREG, \$GDDAT, \$BDDAT
 DF1

:ITEM 2
 EM2 :FAILED TO INTERRUPT
 DH3 :ERRPC STREG ACTUAL
 DT3 :\$ERRPC, STREG, \$BDDAT
 DF1

:ITEM 3
 EM3 :UNEXPECTED INTERRUPT
 DH3 :ERRPC STREG
 DT3 :\$ERRPC, STREG
 DF1

:ITEM 4
 EM4 :ERROR ON A/D CHANNEL
 DH2 :ERRPC STREG CHAN NOMINAL TOL ACTUAL
 DT2 :\$ERRPC, STREG, CHANL, \$GDDAT, SPREAD, \$BDDAT
 DF1

665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679 001256
 680
 681
 682
 683
 684 001256 014146
 685 001260 014306
 686 001262 014472
 687 001264 014532
 688
 689
 690
 691 001266 014174
 692 001270 014427
 693 001272 014522
 694 001274 014532
 695
 696
 697 001276 014224
 698 001300 014427
 699 001302 014522
 700 001304 014532
 701
 702
 703 001306 014255
 704 001310 014344
 705 001312 014504
 706 001314 014532
 707
 708

G02

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 19
MISCELLANEOUS, TEMPORARY, AND STORAGE LOCATIONS

709				.SBTTL	MISCELLANEOUS, TEMPORARY, AND STORAGE LOCATIONS	
710	001316	170400		STREG:	ABASE	: ADDRESS OF STATUS REGISTER
711	001320	170401		ADST1:	ABASE+1	: UPPER BYTE OF STATUS REG.
712	001322	170402		ADBLFF:	ABASE+2	: ADDRESS OF A/D BUFFER
713	001324	140340		VECTOR:	AVECT1	: VECTOR ADDRESS
714	001326	000300		BASEBR:	APRIOR	: INTERRUPT PRIORITY LEVEL
715	001330	140342		VECTR1:	AVECT1+2	
716	001332	000040		VADR:	40	: INCREMENT FOR BUS ADDRESS
717	001334	000040		VVCT:	40	: INCREMENT FOR VECTOR ADDRESS
718	001336	000000		BASECH:	0	: BASE CHANNEL
719	001340	000060		KBVECT:	60	
720	001342	000000		WIDE:	00	: NO. OF WIDE STATES
721	001344	000000		NARROW:	00	: NO. OF NARROW STATES
722	001346	000000		FIRST:	00	
723	001350	000000		SKIPST:	00	: NO. OF SKIPPED STATES
724	001352	000000		TEMP:	00	: WORK AREA
725	001354	000000		CH1:	00	: FIRST CHANNEL
726	001356	000000		CH2:	00	: SECOND CHANNEL
727	001360	000000		NBEXT:	00	: NO. OF AD11K'S TO BE TESTED
728	001362	000000		NMBEXT:	00	: NO. OF AD11K'S TO BE TESTED
729	001364	000000		DUMMY:	00	: DUMMY CHANNEL
730	001366	000000		CHANL:	00	: CHANNEL VALUE
731	001370	000000		TADDR:	00	: TEST ADDRESS
732	001372	000000		RNA:	00	: RANDOM
733	001374	000000		RNB:	00	: NUMBER
734	001376	000000		RNC:	00	: VALUES
735	001400	000000		RMS:	00	: RMS NOISE VALUE
736	001402	000000		PEAK:	00	: PEAK NOISE VALUE
737	001404	000000		FLAG:	00	: VTSS FLAG
738	001406	000000		SPREAD:	00	: DEVIATION FROM THE NOMINAL
739	001410	000000		DAC:	00	: SAR VALUE
740	001412	000000		DELAY:	00	: TIME DELAY COUNTER
741	001414	000000		EDGE:	00	: EDGE VALUE
742	001416	000000		BITPNT:	00	
743	001420	000000		MIN:	00	: MIN VALUE
744	001422	000000		WFTST:	00	: OPTION TEST AREA FLAG
745	001424	000000		MAX:	00	: MAX VALUE
746	001426	000000		PERCNT:	00	: PERCENT FOR SAR ROUTINE
747	001430	000000		OUT:	00	
748						
749	001432			UNEXP:		
750	001432	012737	001446	MOV	#15, \$ESCAPE	: ;ESCAPE TO 15 ON ERROR
751	001440	005237	001103	INC	\$ERFLG	
752	001444	104003		ERROR	3	
753	001446	005037	001162	15:	\$ESCAPE	: RETURN ESCAPE TO NORMAL
754	001452	000002		RTI		: UNEXPECTED INTERRUPT

H02

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 20
CONTROL A AND C DECODERS

755				:SBTTL	CONTROL A AND C DECODERS	
756	001454	010046		:SERV:	MOV RO, -(SP)	:SAVE RO
757	001456	017700	177464		MOV 2\$TKB, RO	:GET CHARACTER
758	001462	042700	177600		BIC #177600, RO	
759	001466	120027	000003		CMPB RO, #3	:IS IT 1C?
760	001472	001010			BNE 1\$	
761	001474	104400	012146		TYPE ,CMMSG	:ECHO CHARACTER
762	001500	012706	001100		MOV 1\$STACK, SP	
763	001504	004737	011266		JSR PC, RST	:RESET & SET INTRPT. EN.
764	001510	000137	002256		JMP BEG2	
765	001514	120027	000001	1\$:	CMPB RO, #1	:IS IT 1A?
766	001520	001010			BNE 2\$	
767	001522	104400	012141		TYPE ,AMMSG	:ECHO CHARACTER
768	001526	012706	001100		MOV 1\$STACK, SP	
769	001532	004737	011266		JSR PC, RST	:RESET & SET INTRPT. EN.
770	001536	000177	177626		JMP 2\$ADDR	:RETURN TO TEST
771	001542	120027	000007	2\$:	CMPB RO, #7	:IS IT 1G?
772	001546	001021			BNE NONE	
773	001550	023727	001140	177570	CMP SWR, #177570	:HARDWARE SWREG?
774	001556	001415			BEQ NONE	
775	001560	104400	012153		TYPE ,CMMSG	:ECHO CHARACTER
776	001564	017746	177350		MOV 2\$SWR, -(SP)	:SAVE 2\$SWR FOR TYPEOUT
777						:TYPE SWREG
778	001570	104402			TYPOS	:GO TYPE--OCTAL ASCII
779	001572	006			.BYTE 6	:TYPE 6 DIGITS
780	001573	001			.BYTE 1	:TYPE LEADING ZEROS
781	001574	104400	012333		TYPE ,SLASH	
782	001600	104406			RDOCT	:READ NEW VALUE
783	001602	012677	177332		MOV (SP)+, 2\$SWR	:LOAD NEW SWREG VALUE
784	001606	012600		POPRO:	MOV (SP)+, RO	
785	001610	000002		RETURN:	RTI	
786	001612	104400	012137	NONE:	TYPE ,QUEST	:TYPE ""
787	001616	000773			BR POPRO	

MAINDEC-11-02ADL-5
02ADL8.P11

MAG 11 27 6651 2-DEC-76 16:29 PAGE 21
INITIAL START-UP, HOUSEKEEPING, AND DIALOGUE

```

788      .SBTTL      INITIAL START-UP, HOUSEKEEPING, AND DIALOGUE
789      BEGIN:     CLR      WFTST
790      001620      005037      001422      BR      RBEG
791      001624      000403
792      001626      012737      000001      001422      BEGIN2:  MOV      #1, WFTST
793      001634      000005      RBEG:     RESET
794      .SBTTL      INITIALIZE THE COMMON TAGS
795      001636      012706      001100      ::CLEAR THE COMMON TAGS ($CMTAG) AREA
796      001642      005026      MOV      #CMTAG, R6      ;; FIRST LOCATION TO BE CLEARED
797      001644      022706      001140      CLR      (R6)+      ;; CLEAR MEMORY LOCATION
798      001650      001374      CMP      #SWR, R6      ;; DONE?
799      001652      012706      001100      BNE      -6      ;; LOOP BACK IF NO
800      ::INITIALIZE A FEW VECTORS
801      001656      012737      015130      000020      MOV      #SCOPE, @IOTVEC      ;; IOT VECTOR FOR SCOPE ROUTINE
802      001664      012737      000340      000022      MOV      #340, @IOTVEC+2      ;; LEVEL 7
803      001672      012737      015406      000030      MOV      #ERROR, @EMTVEC      ;; EMT VECTOR FOR ERROR ROUTINE
804      001700      012737      000340      000032      MOV      #340, @EMTVEC+2      ;; LEVEL 7
805      001706      012737      016714      000034      MOV      #TRAP, @TRFVEC      ;; TRAP VECTOR FOR TRAP CALLS
806      001714      012737      000340      000036      MOV      #340, @TRFVEC+2      ;; LEVEL 7
807      001722      012737      016754      000024      MOV      #SPWRDN, @PWRVEC      ;; POWER FAILURE VECTOR
808      001730      012737      000340      000026      MOV      #340, @PWRVEC+2      ;; LEVEL 7
809      001736      013737      011756      011750      MOV      SENDCT, SEOPCT      ;; SETUP END-OF-PROGRAM COUNTER
810      001744      005037      001160      CLR      $TIMES      ;; INITIALIZE NUMBER OF ITERATIONS
811      001750      005037      001162      CLR      $ESCAPE      ;; CLEAR THE ESCAPE ON ERROR ADDRESS
812      001754      112737      000001      001115      MOV      #1, $ERMAX      ;; ALLOW ONE ERROR PER TEST
813      001762      012737      001762      001106      MOV      #, $LPADR      ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
814      001770      012737      001770      001110      MOV      #, $LPERR      ;; SETUP THE ERROR LOOP ADDRESS
815      ::SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
816      ::EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
817      001776      013746      000004      MOV      @ERRVEC, -(SP)      ;; SAVE ERROR VECTOR
818      002002      012737      002036      000004      MOV      #64$, @ERRVEC      ;; SET UP ERROR VECTOR
819      002010      012737      177570      001140      MOV      #DSWR, SWR      ;; SETUP FOR A HARDWARE SWITCH REGISTER
820      002016      012737      177570      001142      MOV      #DDISP, DISPLAY      ;; AND A HARDWARE DISPLAY REGISTER
821      002024      022777      177777      177106      CMP      #-1, @SWR      ;; TRY TO REFERENCE HARDWARE SWR
822      002032      001012      BNE      66$      ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
823      002034      000403      BR      65$      ;; AND THE HARDWARE SWR IS NOT = -1
824      002036      012716      002044      64$:   MOV      #65$, (SP)      ;; BRANCH IF NO TIMEOUT
825      002042      000002      RTI      ;; SET UP FOR TRAP RETURN
826      002044      012737      000176      001140      65$:   MOV      #SWREG, SWR      ;; POINT TO SOFTWARE SWR
827      002052      012737      000174      001142      MOV      #DISPREG, DISPLAY
828      002060      012637      000004      66$:   MOV      (SP)+, @ERRVEC      ;; RESTORE ERROR VECTOR
829      830
831      002064      005037      001202      CLR      $PASS      ;; CLEAR PASS COUNT
832      002070      132737      000200      001215      BITB      #APTSIZE, $ENVN      ;; TEST USER SIZE UNDER APT
833      002076      001403      BEQ      67$      ;; YES, USE NON-APT SWITCH
834      002100      012737      001216      001140      MOV      #SSWREG, SWR      ;; NO, USE APT SWITCH REGISTER
835      002106
836      67$:

```

J02

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 22
 DZADLB.P11 INITIALIZE THE COMMON TAGS

836	002106	005037	001404		CLR	FLAG	:CLEAR VT55 FLAG
837	002112	005737	000042		TST	#42	:IS IT CHAINED?
838	002116	001033			BNE	REST1	
839				.SBTTL		DETERMINE IF VT55	TYPE TERMINAL IS PRESENT
840	002120	042777	000100	177016	BIC	#100,25TKS	
841	002126	104400	013573		TYPE	CO	:TYPE ASCIZ STRING
842	002132	004737	002530		JSR	PC,VTFLG	:GET A CHARACTER
843	002136	020027	000033		CMP	RO,#33	
844	002142	001017			BNE	NOVT55	:NO VT55 PRESENT
845	002144	004737	002530		JSR	PC,VTFLG	:GET A CHARACTER
846	002150	020027	000057		CMP	RO,#57	
847	002154	001012			BNE	NOVT55	:NO VT55 PRESENT
848	002156	004737	002530		JSR	PC,VTFLG	:GET A CHARACTER
849	002162	020027	000103		CMP	RO,#103	
850	002166	001403			BEQ	VT55	:VT55 IS PRESENT
851	002170	020027	000105		CMP	RO,#105	
852	002174	001002			BNE	NOVT55	
853	002176	005237	001404	VT55:	INC	FLAG	

K02

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27.665) 2-DEC-76 16:29 PAGE 23
DIALOGUE TO DETERMINE WHICH TEST TO RUN

854					.SBTTL	DIALOGUE TO DETERMINE WHICH TEST TO RUN
855	002202	104400	013736		NOVT55: TYPE	.HEAD1
856	002206	000005			REST1: RESET	
857	002210	004737	005504		JSR	PC, FIXONE ;INITIALIZE ADDRESSES
858	002214	013700	001340		MOV	KBVECT, RO
859	002220	012720	001454		MOV	#ISERV, (RO)+
960	002224	012710	000340		MOV	#340, (RO)
861	002230	012737	062341	001372	MOV	#62341, RNA ;RANDOM NO, VARIABLES
862	002236	012737	142315	001374	MOV	#142315, RNB
863	002244	012737	127623	001376	MOV	#127623, RNC
864	002252	004737	011552		JSR	PC, WFAJ ;STANDARD OR OPTION TEST TOLERANCES?
865	002256	000005			BEG2: RESET	;RESTART ADDRESS
866	002260	012706	001100		MOV	#STACK, SP ;RESET STACK IN CASE RESTARTED
867	002264	005737	000042		TST	#42 ;IS IT CHAINED?
868	002270	001402			BEG	15
869	002272	000137	005234		JMP	BEG1 ;GO TO LOGIC TESTS
870	002276	104400	013401		15: TYPE	, MSG71
871	002302	104405			TRYAG: RDLIN	
872	002304	052777	000100	176632	BIS	#100, #STKS
873	002312	005037	177776		CLR	PSW
874	002316	012600			MOV	(SP)+, RO ;READ ANSWER
875	002320	142710	000040		BICB	#40, (RO)
876	002324	121027	000101		CMPB	(RO), #'A ;IS IT A?
877	002330	001002			BNE	15 ;;NO, TRY C
878	002332	000137	005272		JMP	BEGINA ;GO TO AUTO TEST
879	002336	121027	000103		15: CMPB	(RO), #'C ;IS IT C?
880	002342	001002			BNE	25 ;;NO, TRY L
881	002344	000137	005050		JMP	BEGINC ;GO TO CALIBRATION TEST
882	002350	121027	000114		25: CMPB	(RO), #'L ;IS IT L?
883	002354	001002			BNE	35 ;;NO, TRY N
884	002356	000137	005234		JMP	BEG1 ;GO TO LOGIC TESTS
885	002362	121027	000116		35: CMPB	(RO), #'N ;IS IT N?
886	002366	001002			BNE	45 ;;NO, TRY S
887	002370	000137	005660		JMP	BEGINN ;GO TO NOISE TEST
888	002374	121027	000123		45: CMPB	(RO), #'S ;IS IT S?
889	002400	001002			BNE	55 ;;NO, TRY W
890	002402	000137	005730		JMP	BEGINS ;GO TO SETTLE TEST
891	002406	121027	000127		55: CMPB	(RO), #'W ;IS IT W?
892	002412	001002			BNE	65 ;;NO, TRY AGAIN
893	002414	000137	005360		JMP	BEGINW ;GO TO WRAPAROUND TEST
894	002420	104400	012137		65: TYPE	QUEST
895	002424	000726			BR	TRYAG ;WAIT FOR CHARACTER
896	002426	013737	001250	001126	TESTAD: MOV	\$BASE, \$BDDAT ;SETUP TO TEST FOR AD11K'S
897	002434	013746	000004		MOV	#ERRVEC, -(SP) ;SAVE ERRVEC
898	002440	012737	002472	000004	MOV	#25, ERRVEC ;SET UP FOR TIME OUT ERROR
899	002446	005037	001360		CLR	NBEXT ;CLEAR AD11K COUNTER
900	002452	005777	176450		15: TST	\$BDDAT ;ADDRESS AD11K
901	002456	005237	001360		INC	NBEXT ;INCREMENT AD11K COUNTER
902	002462	063737	001332	001126	ADD	VAOR, \$BDDAT ;GET NEXT AD11K
903	002470	000770			BR	15 ;;TRY NEXT AD11K

L02

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 24
DIALOGUE TO DETERMINE WHICH TEST TO RUN

904	002472	022626		25:	CMP	(SP)+,(SP)+	:POP 2 WORDS OFF STACK
905	002474	013746	001360		MOV	NBEXT,-(SP)	::SAVE NBEXT FOR TYPEOUT
906							::TYPE NUMBER OF AD11K'S
907	002500	104402			TYPOS		::GO TYPE--OCTAL ASCII
908	002502	002			.BYTE	2	::TYPE 2 DIGIT(S)
909	002503	000			.BYTE	0	::SUPPRESS LEADING ZEROS
910	002504	104400	012741		TYPE	MSG50	
911	002510	005337	001360		DEC	NBEXT	:ADJUST AD11K COUNT
912	002514	013737	001360	001362	MOV	NBEXT,NMBEXT	:KEEP COUNT OF NUMBER
913	002522	012637	000004		MOV	(SP)+,ERRVEC	:RESTORE ERRVEC
914	002526	000207			RTS	PC	
915							
916	002530	005000		VTFLG:	CLR	RO	:TEST FOR PRESENCE
917	002532	105777	176406	15:	TSTB	2\$TKS	:OF VT55
918	002536	100404			BMI	2\$	::VT55 RESPONDS WITH <33><57>[<103> OR <105>]
919	002540	005300			DEC	RO	
920	002542	001373			BNE	1\$	::
921	002544	005726			TST	(SP)+	:POP A WORD OFF STACK
922	002546	000615			BR	NOVT55	:NO VT55 PRESENT
923	002550	017700	176372	25:	MOV	2\$TKB,RO	
924	002554	042700	177600		BIC	#177600,RO	:TEST VT55 CODE
925	002560	000207			RTS	PC	

M02

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 25
DIALOGUE TO DETERMINE WHICH TEST TO RUN

```

926 002562 BEGINL:
927 ::*****
928 ::*TEST 1      FLOAT A ONE THRU MULTIPLEXER BITS
929 ::*****
930 002562 012737 002562 001106 TST1:  MOV  #TST1,$LPADR
931 002570 012737 002562 001110      MOV  #TST1,$LPERR
932 002576 012737 000400 001124      MOV  #BIT8,$GDDAT      ;LOAD FIRST BIT
933 002604 004737 003660 2$:      JSR  PC,TESTIT
934 002610 104001      ERROR 1      ;FAILED TO LOAD + READ BIT
935 002612 006137 001124 1$:      ROL  $GDDAT      ;GET NEXT BIT
936 002616 023727 001124 040000      CMP  $GDDAT,#BIT14      ;FINISHED?
937 002624 001367      BNE  2$      ;;NO,GO TO NEXT TEST
938
939 ::*****
940 ::*TEST 2      LOAD AND READ BACK INTERRUPT ENABLE BIT6
941 ::*****
942 002626 000004 TST2:  SCOPE
943 002630 012777 001432 176466      MOV  #UNEXP,$VECTOR      ;SETUP FOR UNEXPECTED INTERRUPT
944 002636 012737 000100 001124      MOV  #BIT6,$GDDAT      ;LOAD EXPECTED DATA
945 002644 004737 003660      JSR  PC,TESTIT
946 002650 104001      ERROR 1      ;FAILED TO LOAD + READ INTERRUPT ENABLE
947
948 ::*****
949 ::*TEST 3      LOAD AND READ BACK CLOCK OVERFLOW START ENABLE BITS
950 ::*****
951 002652 000004 TST3:  SCOPE
952 002654 012737 000040 001124      MOV  #BIT5,$GDDAT      ;LOAD EXPECTED DATA
953 002662 004737 003660      JSR  PC,TESTIT
954 002666 104001      ERROR 1      ;FAILED TO LOAD + READ CLOCK OVERFLOW START ENAB
955
956 ::*****
957 ::*TEST 4      LOAD AND READ BACK EXTERNAL START ENABLE BIT4
958 ::*****
959 002670 000004 TST4:  SCOPE
960 002672 012737 000020 001124      MOV  #BIT4,$GDDAT      ;LOAD EXPECTED DATA
961 002700 004737 003660      JSR  PC,TESTIT
962 002704 104001      ERROR 1      ;FAILED TO LOAD + READ EXT. START ENABLE
963
964 ::*****
965 ::*TEST 5      LOAD AND READ BACK ERROR FLAG BIT15
966 ::*****
967 002706 000004 TST5:  SCOPE
968 002710 012737 100000 001124      MOV  #BIT15,$GDDAT      ;LOAD EXPECTED DATA
969 002716 004737 003660      JSR  PC,TESTIT
970 002722 104001      ERROR 1      ;FAILED TO LOAD + READ ERROR FLAG

```


N02

MAINDEC-11-DZADL-8
DZADLB.P11 T6

MACY11 27(665) 2-DEC-76 16:29 PAGE 26
TEST INIT CLEARS BITS 1,4-6,8-13,15

```

970
971
972
973 002724 000004
974 002726 012737 000300 001160
975 002734 005037 011124
976 002740 012777 137562 176350
977 002746 012737 000010 001352
978 002754 005337 001352
979 002760 001375
980 002762 000005
981 002764 052777 000100 176152
982 002772 017737 176320 001126
983 003000 001401
984 003002 104001
985
986
987
988
989 003004 000004
990 003006 012700 001000
991 003012 005277 176300
992 003016 012737 000200 001124
993 003024 005300
994 003026 001376
995 003030 042777 100000 176260
996 003036 004737 003666
997 003042 104001
998 003044 017700 176252
999
1000
1001
1002
1003 003050 000004
1004 003052 012737 000300 001160
1005 003060 005037 001124
1006 003064 005277 176226
1007 003070 105777 176222
1008 003074 100375
1009 003076 000005
1010 003100 004737 003666
1011 003104 104001
1012 003106 052777 000100 176030
1013
1014
1015
1016
1017 003114 000004
1018 003116 012777 000001 176172
1019 003124 105777 176166
1020 003130 100375
1021 003132 017700 176164
1022 003136 004737 003666
1023 003142 104001

*****
;TEST 6 TEST INIT CLEARS BITS 1,4-6,8-13,15
*****
;STE: SCOPE
;MOV #300,$TIMES ;DO 300 ITERATIONS
;CLR $GDDAT ;LOAD EXPECTED DATA
2$: MOV #137562,$STREG ;SET STATUS REGISTER
;MOV #10,TEMP
1$: DEC TEMP ;DELAY FOR ONE-SHOT
;BNE 1$ ;IN ANALOG PACKAGE
;RESET ;INITIALIZE
;BIS #100,$STKS ;SET INTRPT. ENABLE
;MOV $STREG,$BDDAT ;READ STATUS REGISTER
;BEQ TST7 ;NEXT TEST
;ERROR 1 ;RESET FAILED TO CLEAR AD ST. REG. BITS

*****
;TEST 7 TEST DONE FLAG SETS AND BIT0 CLEARS ON END OF CONV.
*****
;TST7: SCOPE
;MOV #BIT9,RO ;STALL TIME COUNTER
;INC $STREG ;START CONVERSION
;MOV #BIT7,$GDDAT ;LOAD EXPECTED
1$: DEC RO ;STALL
;BNE 1$ ;TIME
;BIC #BIT15,$STREG ;MASK OUT ERROR BIT
;JSR PC,TEST
;ERROR 1 ;A/D DONE FLAG FAILED TO SET;BIT0 FAILED TO CLEAR
;MOV $ADBUFF,RO ;CLEAR DONE FLAG FOR ITERATIONS

*****
;TEST 10 TEST INIT CLEARS DONE FLAG
*****
;TST10: SCOPE
;MOV #300,$TIMES ;DO 300 ITERATIONS
;CLR $GDDAT ;CLEAR EXPECTED
;INC $STREG ;START CONVERSION
2$: TSTB $STREG
;BPL 2$
;RESET
;JSR PC,TEST
;ERROR 1 ;DONE FLAG FAILED TO CLEAR
;BIS #100,$STKS ;SET INTRPT. EN. BIT

*****
;TEST 11 TEST A/D DONE FLAG CLEARS WHEN READ CONVERTED VALUE
*****
;TST11: SCOPE
;MOV #BIT0,$STREG ;SET A/D START CONVERSION BIT
1$: TSTB $STREG ;WAIT FOR FLAG
;BPL 1$
;MOV $ADBUFF,RO ;READ CONVERTED VALUE
;JSR PC,TEST
;ERROR 1 ;DONE FLAG FAILED TO CLEAR

```

B03

MAINDEC-11-DZADL-B
DZADL.B.P11 T12MACY: 27.665 2-DEC-76 16:29 PAGE 27
GENERATE INTERRUPT WHEN DONE FLAG SETS AFTER CONVERSION

```

1024 *****
1025 *TEST 12 GENERATE INTERRUPT WHEN DONE FLAG SETS AFTER CONVERSION
1026 *****
1027 *12: SCOPE
1028 * "ENTERING TEST 12" TYPED OUT TO TELL YOU THE NEXT
1029 *TEST THAT IS GOING TO BE EXECUTED. IT IS ONLY TYPED ON PASS 0.
1030 *THERE IS DANGER THAT THE UNIBUS COULD GET "HUNG" WHILE
1031 *EXECUTING TEST "12".
1032 MOV #12,RO ;GET TEST NO.
1033 JSR PC,DUMW ;PRINT MESSAGE
1034 CLR PSW ;RESET PRIORITY
1035 MOV #15,2VECTOR ;INTERRUPT VECTOR ADDRESS
1036 MOV #BIT6!BIT0,2STREG ;SET INTERRUPT ENABLE BIT + START CONVERSION
1037 25: TSTB 2STREG ;WAIT FOR DONE
1038 BPL 25 ;FLAG TO SET
1039 MOV 2STREG,$BDDAT ;READ STATUS REGISTER
1040 MOV #BIT7!BIT6,$GDDAT ;GOOD DATA
1041 ERROR 2 ;FAILED TO INTERRUPT ON DONE
1042 JSR PC,DUMC ;TYPE COMPLETED
1043 BR 13 ;BRANCH TO NEXT TEST
1044 15: CMP (SP)+,(SP)+ ;RESET STACK POINTER
1045 JSR PC,DUMC ;TYPE COMPLETED
1046 *****
1047 *TEST 13 TEST FOR CORRECTNESS OF PRIORITY OF INTERRUPT
1048 *****
1049 *13: SCOPE
1050 * "ENTERING TEST 13" TYPED OUT TO TELL YOU THE NEXT
1051 *TEST THAT IS GOING TO BE EXECUTED. IT IS ONLY TYPED ON PASS 0.
1052 *THERE IS DANGER THAT THE UNIBUS COULD GET "HUNG" WHILE
1053 *EXECUTING TEST "13".
1054 MOV #13,RO ;GET TEST NO.
1055 JSR PC,DUMW ;PRINT MESSAGE
1056 MOV BASEBR,PSW ;GET ONE LEVEL LESS
1057 SUB #40,PSW ;PRIORITY
1058 MOV #15,2VECTOR ;INTERRUPT VECTOR ADDRESS
1059 MOV #BIT6!BIT0,2STREG ;SET INTERRUPT ENABLE BIT + START CONVERSION
1060 25: TSTB 2STREG ;WAIT FOR
1061 BPL 25 ;DONE FLAG
1062 MOV 2STREG,$BDDAT ;READ ST. REG.
1063 ERROR 2 ;INTERRUPT NOT SERVICED AT LEVEL 5
1064 BR 35
1065 15: CMP (SP)+,(SP)+ ;RESET STACK POINTER
1066 35: MOV #45,2VECTOR ;INTERRUPT VECTOR ADDRESS
1067 MOV BASEBR,PSW ;SET PRIORITY TO 6
1068 MOV #BIT6!BIT0,2STREG ;SET INTERRUPT ENABLE + START CONVERSION
1069 55: TSTB 2STREG ;WAIT FOR
1070 BPL 55 ;DONE FLAG
1071 65: MOV #UNEXP,2VECTOR ;UNEXPECTED INTERRUPT VECTOR
1072 CLR 2STREG ;CLEAR BIT 6
1073 CLR PSW ;RESET PRIORITY
1074 MOV 2ADDBUFF,RO ;CLEAR DONE FLAG
1075 JSR PC,DUMC ;TYPE COMPLETED
1076 BR 14 ;BRANCH TO NEXT TEST
1077

```

C03

MAINDEC-11-DZADL-B
DZADLB.P11

T13

MACY11 27(665) 2-DEC-76 16:29 PAGE 28
TEST FOR CORRECTNESS OF PRIORITY OF INTERRUPT

```

1078 003402 022626      45:  CMP      (SP)+,(SP)+      ;RESET STACK POINTER
1079 003404 104003      ERROR 3      ;INTERRUPTED AT LEVEL 6 - WRONG
1080 003406 000761      BR      65
1081
1082
1083
1084
1085 003410 000004      ;*****
1086 003412 012737 000300 001160      ;TEST 14      TEST DONE FLAG CLEARS AFTER INTERRUPT
1087 003420 012737 000100 001124      ;*****
1088 003426 012777 003462 175670      ;ST14:  SCOPE
1089 003434 012777 000101 175654      MOV      #300,STIMES      ;DO 300 ITERATIONS
1090 003442 105777 175650      MOV      #BIT6,$GDDAT      ;LOAD GOOD DATA
1091 003446 100375      MOV      #15,$VECTOR      ;INTERRUPT VECTOR ADDRESS
1092 003450 017737 175642 001126      MOV      #BIT6!BIT0,$STREG      ;SET INTERRUPT ENABLE BIT + START CONVERSION
1093 003456 104002      TSTB      $STREG      ;WAIT FOR
1094 003460 000412      BPL      25      ;DONE FLAG
1095 003462 022626      MOV      $STREG,$BDDAT      ;READ STATUS REGISTER
1096 003464 004737 003666      ERROR 2      ;FAILED TO GENERATE INTERRUPT
1097 003470 104001      BR      TST15      ;GO TO NEXT TEST
1098 003472 017737 175624 001352      15:  CMP      (SP)+,(SP)+      ;RESET STACK POINTER
1099 003500 012777 001432 175616      JSR      PC,TEST      ;DONE FLAG NOT CLEARED AFTER AN INTERRUPT
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
003506 000004      ;*****
003510 012777 000001 175600      ;TEST 15      TEST ERROR FLAG SETS IF 2ND CONVERSION ENDS BEFORE READING BUFFER
003516 105777 175574      ;*****
003522 100375      ;ST15:  SCOPE
003524 012737 100200 001124      MOV      #BIT0,$STREG      ;START CONVERSION
003532 012777 000001 175556      15:  TSTB      $STREG      ;WAIT FOR
003540 012700 001000      BPL      15
003544 005300      MOV      #BIT15!BIT7,$GDDAT      ;LOAD EXPECTED VALUE
003546 001376      MOV      #BIT0,$STREG      ;START 2ND CONVERSION
003550 004737 003666      MOV      #BIT9,R0      ;WAIT FOR 2ND
003554 104001      35:  DEC      R0      ;CONVERSION TO END
003556 017700 175540      45:  BNE      35
      JSR      PC,TEST      ;ERROR FLAG NOT SET WHEN 2ND
      ERROR 1      ;CONVERT ENDS BEFORE READ BUFFER FROM FIRST
      MOV      $ADDBUFF,R0      ;CLEAR DONE FLAG

```

D03

MAINDEC-11-DZADL-8
DZADLB.P11 716

MACY11 27(665) 2-DEC-76 16:29 PAGE 29
TEST ERROR FLAG SETS IF START 2ND CONV. BEFORE DONE FLAG SETS

```

1117
1118
1119
1120 003562 000004
1121 003564 012737 100000 001124
1122 003572 012700 000100
1123 003576 012777 000001 175512
1124 003604 112777 000001 175504
1125 003612 005300
1126 003614 001376
1127 003616 017737 175474 001126
1128 003624 042737 077777 001126
1129 003632 023737 001124 001126
1130 003640 001401
1131 003642 104001
1132 003644 017700 175452
1133 003650 005077 175442
1134 003654 000004
1135 003656 000207
1136
1137
1138
1139
1140 003660 013777 001124 175430
1141 003666 017737 175424 001126
1142 003674 023737 001124 001126
1143 003702 001002
1144 003704 062716 000002
1145 003710 000207

::*****
::TEST 16 TEST ERROR FLAG SETS IF START 2ND CONV. BEFORE DONE FLAG SETS
::*****
↑ST16: SCOPE
MOV #BIT15,$GDDAT ;LOAD EXPECTED DATA
MOV #100,R0 ;STALL TIME COUNTER
MOV #BIT0,$STREG ;START CONVERSION
MOVB #BIT0,$STREG ;START NEXT CONVERSION
DECS: DEC R0 ;STALL TIME
BNE DECS
MOV $STREG,$BDDAT ;READ STATUS REGISTER
BIC #7777,$BDDAT ;MASK OUT BIT 15
CMP $GDDAT,$BDDAT ;COMPARE RESULTS
BEQ IS ;BRANCH OVER ERROR
ERROR 1 ;ERROR FLAG NOT SET WHEN 2ND
;CONVERT BEGINS BEFORE FIRST DONE
;READ CONVERTED VALUE
;CLEAR STATUS REGISTER
IS: MOV $ADDBUFF,R0
CLR $STREG
SCOPE
RTS PC ;RETURN TO TEST SECTION

::SUBROUTINE FOR LOGIC TESTS::
TEST1: MOV $GDDAT,$STREG ;LOAD EXPECTED VALUE
TEST: MOV $STREG,$BDDAT ;READ ST. REG.
CMP $GDDAT,$BDDAT ;COMPARE RESULTS
BNE RETERR ;ERROR RETURN
ACD #2,$SP
RETERR: RTS PC ;BUMP RETURN ADDRESS TO GET AROUND ERROR

```

E03

MAINDEC-11-DZADL-B
DZADLB.P11MACY: 27-665 2-DEC-76 16:29 PAGE 30
WRAPAROUND TEST SECTION

```

1146      .SECTL      WRAPAROUND TEST SECTION
1147      WRAP:
1148      ::*****
1149      :+TEST 17      TEST CH14 GROUND
1150      ::*****
1151      :+ST17:  NOP
1152      :          MOV      #10,STIMES      ::DO 10 ITERATIONS
1153      :          MOV      #STN-1,STSTNM
1154      :          MOV      #TST17,SLPERR
1155      :          MOV      #TST17,SLPADP
1156      :          JSR      RS,CONVRT      ;DO 8 CONVERSIONS
1157      :          JSR      RS,COMPAR      ;COMPARE RESULTS
1158      :          4000      ;NOMINAL
1159      :          V50      ;TOLERANCE
1160      :          ERROR 4      ;ERROR-CH14 NOT GROUND-AD11K MUST BE IN
1161      :                      ;SINGLE-ENDED CONFIGURATION, G5036 WRAPAROUND
1162      :                      ;MODULE MUST BE PRESENT, CHECK CONNECTION A-VV, VV-A
1163
1164      ::*****
1165      :+TEST 20      TEST CONVERSION FROM EXT. START
1166      ::*****
1167      :+ST20:  SCOPE
1168      :          MOV      #10,STIMES      ::DO 10 ITERATIONS
1169      :          TST      BASECH      ;TESTING AN AM?
1170      :          BNE      TST21      ::YES, GOTO NEXT TEST
1171      :          MOV      #BIT4,STREG      ;ENABLE EXTERNAL START
1172      :          MOV      #BIT9,R0      ;TIME DELAY COUNTER
1173      :          MOV      #BIT7,BIT4,$GDDAT      ;LOAD EXPECTED
1174      :          MOV      #200,$ADDBUFF      ;SIMULATE EXT. START WITH G5036
1175      :          ;WRAPAROUND MODULE PRESENT
1176
1177      :+S:  DEC      R0
1178      :          BNE      IS
1179      :          JSR      PC,TEST
1180      :          ERROR 1      ;FAILED TO DO CONVERSION FROM EXT. START
1181      :          MOV      $ADDBUFF,R0
1182      :          CLR      $STREG      ;CLEAR ST. REG.
1183
1184      ::*****
1185      :+TEST 21      TEST CH0 GROUND
1186      ::*****
1187      :+ST21:  SCOPE
1188      :          MOV      #10,STIMES      ::DO 10 ITERATIONS
1189      :          JSR      RS,CONVRT      ;CONVERT 8 TIMES
1190      :          0
1191      :          JSR      RS,COMPAR      ;COMPARE RESULTS
1192      :          4000      ;NOMINAL
1193      :          V1      ;TOLERANCE
1194      :          ERROR 4      ;ERROR ON A/D CHANNEL
1195
1146      003712      000240
1147
1148      003714      012737      000010      001160
1149
1150      003722      012737      000017      001102
1151
1152      003730      012737      003712      001110
1153
1154      003736      012737      003712      001106
1155
1156      003744      004537      011004
1157
1158      003750      000014
1159
1160      003752      004537      011116
1161
1162      003756      004000
1163
1164      003760      011630
1165
1166      003762      104004
1167
1168      003764      000004
1169
1170      003766      012737      000010      001160
1171
1172      003774      005737      001336
1173
1174      004000      001024
1175
1176      004002      012777      000020      175306
1177
1178      004010      012700      001000
1179
1180      004014      012737      000220      001124
1181
1182      004022      012777      000200      175272
1183
1184      004030      005300
1185
1186      004032      001376
1187
1188      004034      004737      003666
1189
1190      004040      104001
1191
1192      004042      017700      175254
1193
1194      004046      005077      175244
1195
1189      004052      000004
1190
1191      004054      012737      000010      001160
1192
1193      004062      004537      011004
1194
1195      004066      000000
1196
1197      004070      004537      011116
1198
1199      004074      004000
1200
1201      004076      011622
1202
1203      00-100      104004

```

F03

MAINDEC-11-DZADL-B
DZADLB.P11 722MACY:1 27(665) 2-DEC-76 16:29 PAGE 31
TEST CH1 GROUND

```

1197
1198
1199
1200 004102 000004
1201 004104 012737 000010 001160
1202 004112 004537 011004
1203 004116 000001
1204 004120 004537 011116
1205 004124 004000
1206 004126 011626
1207 004130 104004
1208
1209
1210
1211
1212 004132 000004
1213 004134 012737 000010 001160
1214 004142 004537 011004
1215 004146 000002
1216 004150 004537 011116
1217 004154 004632
1218 004156 011630
1219 004160 104004
1220
1221
1222
1223
1224
1225 004162 000004
1226 004164 012737 000010 001160
1227 004172 004537 011004
1228 004176 000003
1229 004200 004537 011116
1230 004204 006000
1231 004206 011636
1232 004210 104004
1233
1234
1235
1236
1237 004212 000004
1238 004214 012737 000010 001160
1239 004222 004537 011004
1240 004226 000004
1241 004230 004537 011116
1242 004234 002000
1243 004236 011636
1244 004240 104004

*****
*TEST 22 TEST CH1 GROUND
*****
*ST22: SCOPE
      MOV      #10,STIMES      ;;DO 10 ITERATIONS
      JSR      RS,CONVRT      ;CONVERT 8 TIMES
                                ;CHANNEL 1
      JSR      RS,COMPAR      ;COMPARE RESULTS
      4000      ;NOMINAL
      VIC      ;TOLERANCE
      ERROR    4              ;ERROR ON A/D CHANNEL

*****
*TEST 23 TEST CH2 +1 VOLT
*****
*ST23: SCOPE
      MOV      #10,STIMES      ;;DO 10 ITERATIONS
      JSR      RS,CONVRT      ;CONVERT 8 TIMES
                                ;CHANNEL 2
      JSR      RS,COMPAR      ;COMPARE RESULTS
      4632      ;NOMINAL
      V50      ;TOLERANCE
      ERROR    4              ;ERROR ON A/D CHANNEL
                                ;AD11K MUST BE SET UP FOR +CR- 5. CR +CR- 5.12V

*****
*TEST 24 TEST CH3 +2.5 VOLTS
*****
*ST24: SCOPE
      MOV      #10,STIMES      ;;DO 10 ITERATIONS
      JSR      RS,CONVRT      ;CONVERT 8 TIMES
                                ;CHANNEL 3
      JSR      RS,COMPAR      ;COMPARE RESULTS
      6000      ;NOMINAL
      V240     ;TOLERANCE
      ERROR    4              ;ERROR ON A/D CHANNEL

*****
*TEST 25 TEST CH4 -2.5 VOLTS
*****
*ST25: SCOPE
      MOV      #10,STIMES      ;;DO 10 ITERATIONS
      JSR      RS,CONVRT      ;CONVERT 8 TIMES
                                ;CHANNEL 4
      JSR      RS,COMPAR      ;COMPARE RESULTS
      2000      ;NOMINAL
      V240     ;TOLERANCE
      ERROR    4

```


G03

MAINDEC-11-DZADL-B
DZADLB.P11 T26MACY11 27(665) 2-DEC-76 16:29 PAGE 32
TEST VERNIER OFFSET DAC ON CH12

```

1245      ::*****
1246      ::*TEST 26      TEST VERNIER OFFSET DAC ON CH12
1247      ::*****
1248      TST26: SCOPE
1249      004242 000004      MOV      #1,STIMES      ;;DO 1 ITERATION
1250      004244 012737 000001 001160      CLR      JADBUFF      ;VERNIER DAC=0
1251      004252 005077 175044      JSR      PC,DAWAIT      ;DELAY FOR DAC SETTling
1252      004256 004737 005040      JSR      R5,CONVRT      ;CONV. CH12, DIRECT VERNIER DAC
1253      004262 004537 011004      JSR      R5,CONVRT
1254      004266 000012      JSR      R5,CONVRT
1255      004270 013704 001352      MOV      TEMP,R4      ;SAVE VALUE IN R4
1256      004274 004537 011116      JSR      R5,COMPAR      ;COMPARE RESULTS
1257      004300 002376      JSR      R5,COMPAR      ;WITH -1.875 VOLTS
1258      004302 011634      JSR      R5,COMPAR      ;TOLERANCE OF 10%
1259      004304 104004      JSR      R5,COMPAR
1260      004306 005037 001424      CLR      MAX
1261      004312 012702 000001      MOV      #1,R2
1262      004316 010277 175000      15:      MOV      R2,JADBUFF      ;SET UP NEXT VERNIER DAC VALUE
1263      004322 004737 005040      JSR      PC,DAWAIT      ;DELAY FOR DAC SETTling
1264      004326 004537 011004      JSR      R5,CONVRT      ;CONVERT IT
1265      004332 000012      JSR      R5,CONVRT
1266      004334 005737 001424      TST      MAX
1267      004340 001010      BNE      25
1268      004342 023727 001352 004000      CMP      TEMP,#4000
1269      004350 002404      BLT      25
1270      004352 005237 001424      INC      MAX
1271      004356 010237 001420      MOV      R2,MIN
1272      004362 020227 000200      25:      CMP      R2,#200
1273      004366 001003      BNE      35
1274      004370 013737 001352 004460      MOV      TEMP,45
1275      004376 013703 001352      35:      MOV      TEMP,R3
1276      004402 160437 001352      SUB      R4,TEMP
1277      004410 004537 011116      MOV      R3,R4
1278      004414 000006      JSR      R5,COMPAR
1279      004416 011640      JSR      R5,COMPAR
1280      004420 104004      JSR      R5,COMPAR
1281      004422 005202      JSR      R5,COMPAR
1282      004424 020227 000400      INC      R2
1283      004430 001332      CMP      R2,#400
1284      004432 000005      BNE      15
1285      004434 052777 000100 174502      RESET
1286      004442 004737 005040      BIS      #100,STKS
1287      004446 004537 011004      JSR      PC,DAWAIT
1288      004452 000012      JSR      R5,CONVRT
1289      004454 004537 011116      JSR      R5,COMPAR
1290      004460 000000      JSR      R5,COMPAR
1291      004462 011624      JSR      R5,COMPAR
1292      004464 104004      JSR      R5,COMPAR

```


H03

MAINDEC-11-DZADL-B
DZADLB.P11 727MACY:1 27(665) 2-DEC-76 16:29 PAGE 33
TEST CH13 +2.5 VOLTS

```

1293
1294
1295
1296 004466 000004
1297 004470 012737 000010 001160
1298 004476 004537 011004
1299 004502 000013
1300 004504 004537 011116
1301 004510 006000
1302 004512 011632
1303 004514 104004
1304
1305
1306
1307 004516 000004
1308 004520 012737 000010 001160
1309 004526 004537 011004
1310 004532 000017
1311 004534 004537 011116
1312 004540 007146
1313 004542 011636
1314 004544 104004

```

```

*****
*TEST 27      TEST CH13 +2.5 VOLTS
*****
TST27: SCOPE
      MOV      #10,STIMES      ;;DO 10 ITERATIONS
      JSR      RS,CONVRT      ;CONVERT 8 TIMES
      13
      JSR      RS,COMPAR      ;COMPARE RESULTS
      6000      ;NOMINAL
      V144      ;TOLERANCE
      ERROR 4
*****
*TEST 30      TEST CH17 +4V
*****
TST30: SCOPE
      MOV      #10,STIMES      ;;DO 10 ITERATIONS
      JSR      RS,CONVRT      ;CONVERT 8 TIMES
      17      ;CHANNEL 17
      JSR      RS,COMPAR      ;COMPARE RESULTS
      7146      ;NOMINAL
      V240      ;TOLERANCE
      ERROR 4      ;ERROR ON A/D CHANNEL

```

I03

MAINDEC-11-DZADL-B
DZADLB.P11 T31

MACY11 27(665)
OFFSET ON CHO

2-DEC-76 16:29 PAGE 34

```

1315
1316
1317
1318 004546 000004
1319 004550 012737 000001 001160
1320 004556 013737 001336 001366
1321 004564 013737 001336 001364
1322 004572 012737 004001 001414
1323 004600 004537 006516
1324 004604 000062
1325 004606 013737 001410 001352
1326 004614 004537 006516
1327 004620 000062
1328 004622 063737 001410 001352
1329 004630 162737 000062 001352
1330 004636 013700 001420
1331 004642 006300
1332 004644 160037 001352
1333 004650 104400 013605
1334 004654 013702 001352
1335 004660 004737 011406
1336 004664 104400 013620
1337 004670 004537 011116
1338 004674 000000
1339 004676 011642
1340 004700 000401
1341 004702 000403
1342 004704 104400 012407
1343 004710 000402
1344 004712 104400 012376

;*****
;TEST 31      OFFSET ON CHO
;*****
TST31:  SCOPE
        MOV     #1,STIMES      ;;DO 1 ITERATION
        MOV     BASECH,CHANL   ;LOAD CHANNEL
        MOV     BASECH,DUMMY   ;LOAD DUMMY
        MOV     #4001,EDGE
        JSR     RS,SARSUB
        SO.
        MOV     DAC,TEMP
        JSR     RS,SARSUB
        SO.
        ADD     DAC,TEMP
        SUB     #62,TEMP
        MOV     MIN,RO
        ASL     RO
        SUB     RO,TEMP
        TYPE    MOFSET         ;TYPE ASCIZ STRING
        MOV     TEMP,R2
        JSR     PC,DECTYP
        TYPE    MLSB          ;TYPE ASCIZ STRING
        JSR     RS,COMPAR      ;IS RESULT WITHIN LIMITS?
        O
        VSOD
        BR      OFFERR         ;NO-ERROR
        BR      OFFOK          ;YES-OK
        BR      TYPE           ;ERMSG
        BR      TST32          ;GO TO NEXT TEST
        BR      TYPE           ;OKMSG

```

J03

MAINDEC-11-DZADL-B
DZADLB.P11 T32MACY11 27(665) 2-DEC-76 16:29 PAGE 35
NOISE TEST ON 8 EDGES

```

1345      ;*****
1346      ;*TEST 32      NOISE TEST ON 8 EDGES
1347      ;*****
1348      TST32: SCOPE
1349      MOV      #1,STIMES      ;;DO 1 ITERATION
1350      MOV      #116,TEMP      ;DAC VALUE
1351      JSR      R5,NOI8      ;NOISE AT -FULL SCALE
1352      IS
1353      JSR      R5,NOI8      ;NOISE AT MID-RANGE
1354      JSR      R5,NOI8      ;NOISE AT +FULL SCALE
1355      IS
1356      IS
1357      IS
1358      ;*****
1359      ;*TEST 33      SETTLE TEST ON 8 EDGES
1360      ;*****
1361      TST33: SCOPE
1362      MOV      #1,STIMES      ;;DO 1 ITERATION
1363      JSR      R5,SET8      ;SETTLE-POSITIVE DIRECTION
1364      IS
1365      IS
1366      MOV      #116,TEMP
1367      JSR      R5,SET8      ;SETTLE-NEGATIVE DIRECTION
1368      IS
1369      IS
1370      ;*****
1371      ;*TEST 34      DIFFERENTIAL LINEARITY AND RELATIVE ACCURACY TEST
1372      ;*****
1373      TST34: SCOPE
1374      MOV      #1,STIMES      ;;DO 1 ITERATION
1375      TST      SPASS      ;FIRST TIME-SKIP DIFLIN
1376      BEQ      LEND
1377      JSR      PC,DIFLIN
1378      LEND: RTS      PC      ;RETURN TO TEST SECTION
1379
1380      DAWAIT: CLR      RO
1381      IS:      DECB      RO
1382      BNE      IS
1383      RTS      PC

```

K03

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 36
CALIBRATION TEST

1384					.SBTTL	CALIBRATION TEST	
1385	005050	012737	005050	001370	BEGINC:	MOV #BEGINC,TADDR	;TEST ADDRESS IN TADDR
1386	005056	005077	174234		CLR	%STREG	;CLEAR STATUS REGISTER
1387	005062	104400	013515		TYPE	HEADS	;TYPE OUT HEADING
1388	005066	005037	177776		CLR	PSW	
1389	005072	017700	174042	15:	MOV	%SWR,R0	;READ CHANNEL FROM SWITCH REG.
1390	005076	042700	177700		BIC	#177700,R0	;ISOLATE MUX BITS
1391	005102	032777	020000	174030	BIT	#BIT13,%SWR	;IS BIT 13 SET?
1392	005110	001005			BNE	25	;;YES,SKIP TYPEOUT
1393	005112	104400	012221		TYPE	CH	
1394	005116	010046			MOV	R0,-(SP)	;;SAVE R0 FOR TYPEOUT
1395							;;TYPE CHANNEL
1396	005120	104402			TYPOS		;;GO TYPE--OCTAL ASCII
1397	005122	002			.BYTE	2	;;TYPE 2 DIGIT(S)
1398	005123	000			.BYTE	0	;;SUPPRESS LEADING ZEROS
1399	005124	012777	001610	174172	25:	MOV #RETURN,%VECTOR	;ADDRESS AFTER INTRPT.
1400	005132	000300			SWAB	R0	;SWITCH BYTES
1401	005134	052700	000100		BIS	#BIT6,R0	
1402	005140	013077	174152		MOV	R0,%STREG	;LOAD THE CHANNEL
1403	005144	012702	000010		MOV	#10,R2	;TYPEOUT COUNTER
1404	005150	005277	174142	35:	INC	%STREG	;START CONVERSION
1405	005154	000001			WAIT		;WAIT FOR INTRPT.
1406	005156	017700	174140		MOV	%ADBUFF,R0	;READ CONVERTED VALUE
1407	005162	032777	020000	173750	BIT	#BIT13,%SWR	;IS BIT 13 SET?
1408	005170	001403			BEQ	45	;NOT SET, TYPE OUT LIST
1409	005172	010077	173744		MOV	R0,%DISPLAY	;PUT VALUE IN DISPLAY FOR DISPLAY CONTR
1410	005176	000735			BR	15	;REPEAT CONVERSION
1411	005200	104400	012224	45:	TYPE	SPACE	
1412	005204	010046			MOV	R0,-(SP)	;;SAVE R0 FOR TYPEOUT
1413							;;PRINT OCTAL CONVERTED VALUE
1414	005206	104402			TYPOS		;;GO TYPE--OCTAL ASCII
1415	005210	004			.BYTE	4	;;TYPE 4 DIGIT(S)
1416	005211	001			.BYTE	1	;;TYPE LEADING ZEROS
1417	005212	012701	010000		MOV	#10000,R1	
1418	005216	005301		55:	DEC	R1	
1419	005220	001376			BNE	55	
1420	005222	005302			DEC	R2	;DECREMENT THE COUNTER
1421	005224	001351			BNE	35	;NO CARRIAGE RETURN
1422	005226	104400	001171		TYPE	%CRLF	;CARRIAGE RETURN
1423	005232	000717			BR	15	;REPEAT CONVERSION

L03

MAINDEC-11-DZADL-B
DZADLB.P11MACY:1 27.6651 2-DEC-76 16:29 PAGE 37
LOGIC TEST SECTION

1424					.SBTTL		LOGIC TEST SECTION	
1425	005234	012737	005234	001370	BEG1:	MOV	#BEG1, TADDR	; TEST ADDRESS
1426	005242	004737	002426			JSR	PC, TESTAD	; NO OF ADDITIONAL AD'S
1427	005246	004737	002562		1\$:	JSR	PC, BEGINL	; LOGIC TESTS
1428	005252	004737	005422			JSR	PC, BUMPAD	; MORE TO TEST?
1429	005256	000773				BR	1\$	; TEST NEXT A/D
1430	005260	012737	005246	011720		MOV	#1\$, AGTST	; ADDRESS FOR EOP
1431	005266	000137	011722			JMP	SEOP	; TYPE END OF PASS
1432								
1433					.SBTTL		AUTO TEST	
1434	005272	012737	005272	001370	BEGINA:	MOV	#BEGINA, TADDR	; TEST ADDRESS
1435	005300	005037	001202			CLR	\$PASS	; CLEAR PASS COUNTER
1436	005304	004737	002426			JSR	PC, TESTAD	; NO. OF AD'S TO BE TESTED
1437	005310	004737	002562		1\$:	JSR	PC, BEGINL	; LOGIC TESTS
1438	005314	104400	012677			TYPE	MEND	; TYPE END OF LOGIC TEST
1439	005320	013746	001316			MOV	\$STREG, -(SP)	; SAVE STREG FOR TYPEOUT
1440	005324	104402				TYPOS		; TYPE OCTAL NUMBER
1441	005326	006				.BYTE	6	; TYPE 6 DIGITS
1442	005327	001				.BYTE	1	; TYPE LEADING ZEROS
1443	005330	104400	001171			TYPE	\$SCRLF	; TYPE A CR, LF
1444	005334	004737	003712			JSR	PC, WRAP	
1445	005340	004737	005422			JSR	PC, BUMPAD	; TEST NEXT A/D
1446	005344	000761				BR	1\$	; TEST NEXT AD
1447	005346	012737	005310	011720		MOV	#1\$, AGTST	; ADDRESS FOR EOP
1448	005354	000137	011722			JMP	SEOP	; TYPE END OF PASS
1449								
1450					.SBTTL		WRAPAROUND TEST	
1451	005360	012737	005360	001370	BEGINW:	MOV	#BEGINW, TADDR	; TEST ADDRESS
1452	005366	005037	001202			CLR	\$PASS	; CLEAR PASS COUNT
1453	005372	004737	002426			JSR	PC, TESTAD	; NO. OF AD'S TO BE TESTED
1454	005376	004737	003712		1\$:	JSR	PC, WRAP	; WRAPAROUND TESTS
1455	005402	004737	005422			JSR	PC, BUMPAD	; MORE A/D'S TO BE TESTED?
1456	005406	000773				BR	1\$	; YES-GO TEST NEXT AD11K
1457	005410	012737	005376	011720		MOV	#1\$, AGTST	
1458	005416	000137	011722			JMP	SEOP	; INCREMENTS \$PASS

M03

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 38
DETERMINE IF MORE AD11K'S TO BE TESTED

1459					.SBTTL	DETERMINE IF MORE AD11K'S TO BE TESTED	
1460	005422	005737	001360		BUMPAD: TST	NBEXT	;ADDITIONAL AD'S?
1461	005426	001424			BEQ	FIXADR	;NO-INITIALIZE ADDRESSES
1462	005430	063737	001332	001316	ADD	VADR,STREG	;SET UP NEW ST. REG.
1463	005436	063737	001332	001320	ADD	VADR,ADST1	;SET UP NEW ADST1
1464	005444	063737	001332	001322	ADD	VADR,ADBUFF	;SET UP NEW BUFFER ADDRESS
1465	005452	063737	001334	001324	ADD	VVCT,VECTOR	;SET UP NEW VECTOR
1466	005460	063737	001334	001330	ADD	VVCT,VECTR1	
1467	005466	005077	173636		CLR	2VECTR1	
1468	005472	005337	001360		DEC	NBEXT	;ONE LESS AD11K
1469	005476	000446			BR	BYPASS	
1470	005500	062716	000002		FIXADR: ADD	#2, (SP)	
1471	005504	013737	001250	001316	FIXONE: MOV	\$BASE,STREG	;RELOAD INITIAL ADDRESSES
1472	005512	013737	001250	001320	MOV	\$BASE,ADST1	
1473	005520	013737	001250	001322	MOV	\$BASE,ADBUFF	
1474	005526	005237	001320		INC	ADST1	
1475	005532	062737	000002	001322	ADD	#2,ADBUFF	
1476	005540	013737	001244	001324	MOV	\$VECT1,VECTOR	
1477	005546	042737	170000	001324	BIC	#170000,VECTOR	
1478	005554	113737	001245	001326	MOVB	\$VECT1+1,BASEBR	
1479	005562	105037	001327		CLRB	BASEBR+1	;CLEAR HIGH BYTE
1480	005566	013737	001324	001330	MOV	VECTOR,VECTR1	
1481	005574	062737	000002	001330	ADD	#2,VECTR1	
1482	005602	005077	173522		CLR	2VECTR1	
1483	005606	013737	001362	001360	MOV	NMBEXT,NBEXT	;RESET COUNTER
1484					;;LOAD .+2 AND HALT TRAP CATCH;;		
1485	005614	012700	000216		BYPASS: MOV	#216,R0	;FILL .+2
1486	005620	012701	000214		MOV	#214,R1	;LOAD HALT
1487	005624	020137	001340		1\$: CMP	R1,K8VECT	
1488	005630	001410			BEQ	2\$	
1489	005632	010021			MOV	R0,(R1)+	
1490	005634	005021			CLR	(R1)+	
1491	005636	010100			MOV	R1,R0	
1492	005640	005720			TST	(R0)+	
1493	005642	020027	001002		CMP	R0,#1002	
1494	005646	001366			BNE	1\$	
1495	005650	000207			RTS	PC	;TEST NEXT A/D
1496	005652	022021			2\$: CMP	(R0)+,(R1)+	
1497	005654	022021			CMP	(R0)+,(R1)+	
1498	005656	000762			BR	1\$	
1499							
1500							
1501					.SBTTL	NOISE TEST 1 EDGE	
1502	005660	012737	005660	001370	BEGINN: MOV	#BEGINN,TADDR	;TEST ADDRESS IN TADDR
1503	005666	104400	012030		TYPE	,NOIMSG	;ASK FOR CHANNEL
1504	005672	104400	013534		TYPE	,ASKCH	
1505	005676	017737	173236	001354	1\$: MOV	JSWR,CH1	;LOAD CHANNEL
1506	005704	042737	177700	001354	BIC	#177700,CH1	
1507	005712	012737	000200	001352	MOV	#200,TEMP	;LOAD DAC VALUE
1508	005720	004537	010312		JSR	R5,NOITST	;GO TO NOISE SUBROUTINE
1509	005724	001354			CH1		
1510	005726	000763			BR	1\$	

N03

MAINDEC-11-DZADL-B
DZADLB.P11

MACY11 27(665) 2-DEC-76 16:29 PAGE 39
INTERCHANNEL SETTling TEST, 1 EDGE

```

1511          SBTTL      INTERCHANNEL SETTling TEST, 1 EDGE
1512 005730 012737 005730 001370 BEGINS: MOV      *BEGINS.TADDR      ;TEST ADDRESS IN TADDR
1513 005736 104400 012050          TYPE      ,SETMSG      ;ASK FOR CHANNELS
1514 005742 104406          RDOCT
1515 005744 012637 001354          MOV      (SP)+,CH1
1516 005750 104400 012335          TYPE      ,TOMSG
1517 005754 104406          RDOCT
1518 005756 012637 001356          MOV      (SP)+,CH2
1519 005762 012737 000200 001352 BK3: MOV      *200,TEMP      ;LOAD DAC
1520 005770 013737 001356 001366          MOV      CH2,CHANL
1521 005776 004737 006336          JSR      PC,GETEDG      ;GET EDGE VALUES
1522 006002 005002          CLR      R2
1523 006004 004737 006170          JSR      PC,SET1A      ;SCALING = .02 LSB
1524 006010 004737 006170          JSR      PC,SET1A      ;MAKE IT .01 LSB
1525 006014 100001          BPL      POSR2
1526 006016 005402          NEG      R2
1527 006020 010204          MOV      R2,R4
1528 006022 012737 000001 006514 POSR2: MOV      *1,EDGFLG
1529 006030 004737 006036          JSR      PC,TYPSET
1530 006034 000752          BR      BK3
1531 006036 004737 011406          TYPSE1: JSR      PC,DECTYP
1532 006042 104400 012231          TYPE      LSB
1533 006046 013746 001356          MOV      CH2,-(SP)      ;;SAVE CH2 FOR TYPEOUT
1534          ;;TYPE CH
1535 006052 104402          TYPOS      ;;GO TYPE--OCTAL ASCII
1536 006054 002          .BYTE      2      ;;TYPE 2 DIGIT(S)
1537 006055 000          .BYTE      0      ;;SUPPRESS LEADING ZEROS
1538 006056 104400 013626          TYPE      ,MAT      ;;TYPE ASCII STRING
1539 006062 004737 006452          JSR      PC,TYPEDG
1540 006066 104400 012244          TYPE      ,SETCH
1541 006072 013746 001354          MOV      CH1,-(SP)      ;;SAVE CH1 FOR TYPEOUT
1542          ;;TYPE CH
1543 006076 104402          TYPOS      ;;GO TYPE--OCTAL ASCII
1544 006100 002          .BYTE      2      ;;TYPE 2 DIGIT(S)
1545 006101 000          .BYTE      0      ;;SUPPRESS LEADING ZEROS
1546 006102 104400 012266          TYPE      ,ATMSG
1547 006106 013737 001354 006134          MOV      CH1,15
1548 006114 163737 001336 006134          SJB      BASECH,15
1549 006122 012777 000200 173172          MOV      *200,ADDBUFF
1550 006130 004537 011004          JSR      R5,CNVRT
1551 006134 000000          15: 0
1552 006136 013746 001352          MOV      TEMP,-(SP)      ;;SAVE TEMP FOR TYPEOUT
1553          ;;TYPE VALUE
1554 006142 104402          TYPOS      ;;GO TYPE--OCTAL ASCII
1555 006144 004          .BYTE      4      ;;TYPE 4 DIGIT(S)
1556 006145 001          .BYTE      1      ;;TYPE LEADING ZEROS
1557 006146 020437 011650          CMP      R4,VSET
1558 006152 003003          BGT      ERR
1559 006154 104400 012376          TYPE      ,OKMSG
1560 006160 000207          RTS      PC

```


B04

NOEC-11-2240-8
2240.B.P.11NOEC-11-27 665 2-DEC-76 16:29 PAGE 40
INTERCHANNEL SETTLING TEST, 1 EDGE

	006162	004400	012407	ERR:	TYPE	ERMSG
	006166	000207			R'S	PC
561						
562						
563						
564						
565						
566						
567	006170	013737	001356	001364	;; SUBROUTINE FOR SETTLING TESTS:;	
568	006176	004537	006516		SETIA: MOV CH2 DUMMY	:LOAD DUMMY
569	006202	000062			JSR R5, SAR SUB	:DO SAR ROUTINE AT 50%
570	006204	063732	001410		SO.	
571	006210	013737	001354	001364	ADD DAC, R2	:ADD RESULT TO R2
572	006216	004537	006516		MOV CH1 DUMMY	:CHANGE DUMMY VALUE
573	006222	000062			JSR R5, SAR SUB	:DO SAR ROUTINE AT 50%
574	006224	163732	001410		SO.	
575	006230	000207			SUB DAC, R2	:SUBTRACT RESULT FROM R2
576					RTS PC	:RETURN
577	006232	012537	001354		SETB: MOV (R5)+, CH1	:GET FIRST CHANNEL
578	006236	012537	001356		MOV (R5)+, CH2	:GET SECOND CHANNEL
579	006242	063737	001336	001354	ADD BASECH, CH1	
580	006250	063737	001336	001356	ADD BASECH, CH2	
581	006256	004737	006336		JSR PC, GETEDG	:GET EDGE VALUES
582	006262	005002			CLR R2	
583	006264	012703	000010		MOV #10, R3	:SET UP COUNTER
584	006270	004737	006170		JSR PC, SETIA	:GET SETTLE VALUES
585	006274	005237	001414		INC EDGE	
586	006300	005303			DEC R3	
587	006302	001372			BNE SETAA	:REPEAT 8 TIMES
588	006304	162737	000010	001414	SUB #10, EDGE	
589	006312	005702			TST R2	
590	006314	100001			BPL R2POS	
591	006316	005402			NEG R2	
592	006320	013204			MOV R2, R4	
593	006322	012737	000010	006514	MOV #8, EDGE FLG	
594	006330	004737	006336		JSR PC, TYPSET	:TYPE OUT RESULTS
595	006334	000207			RTS R5	:RETURN

C04

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27.6651 2-DEC-76 16:29 PAGE 41
INTERCHANNEL SETTLING TEST, 1 EDGE

```

1596      :SUBROUTINE TO GET EDGE VALUE
1597      :CALL=JSR PC,GETEDG
1598      :CONVERSIONS ON A/D CHANNEL 'CHANL'
1599      :RESULT IN EDGE, USES R0
1600      006336 013777 001352 172756 GETEDG: MOV TEMP, DADBUFF
1601      006344 113700 001366      MOVB CHANL, R0      :GET CHANNEL
1602      006350 000300      SWAB R0      :SET UP A.D STATUS REG.
1603      006352 052700 000100      BIS #100, R0      :ENABLE INTRPT.
1604      006356 010077 172734      MOV R0, DSTREG
1605      006362 012700 000100      MOV #100, R0      :DAC SETTLING DELAY
1606      006366 005300      IS: DEC R0
1607      006370 001376      BNE IS
1608      006372 005037 001414      CLR EDGE
1609      006376 012700 000010      MOV #10, R0
1610      006402 012777 001610 172714      MOV #RETURN, DVECTOR      :RETURN ADDRESS
1611      006410 005277 172702      CONV: INC DSTREG      :START CONVERSION
1612      006414 000001      WAIT      :WAIT FOR INTERRUPT
1613      006416 067737 172700 001414      ADD DADBUFF, EDGE
1614      006424 005300      DEC R0
1615      006426 001370      BNE CONV
1616      006430 006237 001414      ASR EDGE
1617      006434 006237 001414      ASR EDGE
1618      006440 006237 001414      ASR EDGE
1619      006444 005537 001414      ADC EDGE
1620      006450 000207      RTS PC
1621
1622      :SUBROUTINE TO TYPE EDGE VALUES::
1623      006452 013703 001414      TYPE: MOV EDGE, R3
1624      006456 010346      MOV R3, -1(SF)      :SAVE R3 FOR TYPEOUT
1625      006460 104402      TYP0S      :TYPE OCTAL VALUE OF EDGE
1626      006462 004      .BYTE 4      :GO TYPE--OCTAL ASCII
1627      006463 001      .BYTE 1      :TYPE 4 DIGIT(S)
1628      006464 023727 006514 000001      CMP EDGEFLG, #1      :TYPE LEADING ZEROS
1629      006472 001407      BEQ RET
1630      006474 062703 000007      ADC #7, R3
1631      006500 104400 013576      TYPE C1      :TYPE ASCII STRING
1632      006504 010346      MOV R3, -1(SF)      :SAVE R3 FOR TYPEOUT
1633      006506 104402      TYP0S      :TYPE EDGE VALUE
1634      006510 004      .BYTE 4      :GO TYPE--OCTAL ASCII
1635      006511 001      .BYTE 1      :TYPE 4 DIGIT(S)
1636      006512 000207      RET: RTS PC      :TYPE LEADING ZEROS
1637      006514 000000      EDGEFLG: 0

```

D04

MAINDEC-11-DZADL-8
DZADL.B.P11

MAY11 27(665) 2-DEC-76 16:29 PAGE 42
INTERCHANNEL SETTLING TEST, 1 EDGE

```

1640      :SUBROUTINE TO DO SUCCESSIVE APPROXIMATION ROUTINE
1641      :CALL=JSR  R5,SARSUB
1642      :XXX:XXX=PERCENT
1643      :RESULT RETURNED IN 'DAC' USES R0,R1,R4
1644      SARSUB: MOV  (R5),+PERCNT      :GET PERCENT
1645      ASL  PERCNT
1646      ASL  PERCNT
1647      ASL  PERCNT
1648      ASL  PERCNT      :RESCALE PERCENT FOR 1600.
1649      SAR1: MOV  #200,BITPNT      :POINTS PER BURST
1650      CLR  DAC      :INITIALIZE BIT POINTER AT MSB
1651      TRY: CLR  RO      :INITIALIZE DAC VALUE
1652      ADD  BITPNT,DAC      :TRY BIT
1653      MOV  DAC,DABUFF
1654      MOV  #100,DELAY
1655      IS:  DEC  DELAY      :STALL TIME
1656      BNE  IS
1657      MOV  #1600,R1      :SET UP FOR 1600. CONVERSIONS
1658      NXTCVT: MOVB DUMMY,DAST1      :PRESET MUX TO DUMMY CHANNEL
1659      MOV  #RETURN,DVECTOR      :RETURN ADDRESS
1660      BIS  #101,DSTREG      :CONVERSION ON DUMMY CHANNEL
1661      WAIT
1662      MOV  DABUFF,R4      :WAIT FOR INTERRUPT
1663      MOV  CHANL,R4      :DUMMY READ
1664      SWAB R4
1665      BIS  #101,R4      :INTERRUPT ENABLE START
1666      MOV  R4,DSTREG      :JUMP TO CHANNEL + START CONVERT
1667      WAIT      :WAIT FOR INTERRUPT
1668      CMP  DABUFF,EDGE
1669      BGE  ZS
1670      INC  RO      :COUNT RESULTS .LT. EDGE
1671      2S:  DEC  R1
1672      BNE  NXTCVT
1673      CMP  RO,PERCNT
1674      BGT  SHIFT
1675      SUB  BITPNT,DAC      :TAKE THE BIT OUT
1676      ASR  BITPNT
1677      BNE  TRY
1678      RTS

```

E04

MAYDEC-11-02ADL-B
02ADL.F11MACY11 27(665) 2-DEC-76 16:29 PAGE 43
INTERCHANNEL SETTLING TEST, 1 EDGE

```

1679          006724 004400 013022          :: DIFFERENTIAL LINEARITY SUBROUTINE::
1680          006730 005037 001430          DIFFLIN: TYPE MSG20
1681          006734 012700 017756          CLR OUT
1682          006740 012701 010000          MOV #BUFFER, R0
1683          006744 005020          MOV #4096, R1          ; 4096 WORDS FOR HISTOGRAM
1684          006746 005301          CLEAR1: CLR (R0)+          ; CLEAR BUFFER AREA
1685          006750 001375          DEC R1
1686          006752 012700 017136          BNE CLEAR1
1687          006756 012701 000310          MOV #DIST, R0          ; DISTRIBUTION BUFFER PCINTER
1688          006762 005003          MOV #200, R1          ; 200. WORDS FOR DISTRIBUTION
1689          006764 005037 001430          CLR R3
1690          006770 005037 001342          CLR OUT
1691          006774 005037 001344          CLR WIDE
1692          007000 005037 001346          CLR NARROW
1693          007004 005037 001350          CLR FIRST
1694          007010 005020          CLEAR2: CLR SKIPST
1695          007012 005301          (R0)+          ; CLEAR DISTRIBUTION BUFFER AREA
1696          007014 001375          DEC R1
1697          007016 012700 000011          BNE CLEAR2
1698          007022 063700 001336          CHANNL: MOV #11, R0          ; CHANNEL 11
1699          007026 000300          ADD BASECH, R0
1700          007030 052700 000100          SWAB R0          ; LOAD MUX BITS
1701          007034 010077 172256          BIS #100, R0
1702          007040 012700 001440          MOV R0, R2STREG
1703          007044 012777 001610          MOV #800, R0          ; NOMINAL STATE WIDTH - 1 LSB
1704          007052 012701 007776          MOV #RETURN, R2VECTOR
1705          007056 004737 010722          AGAIN: MOV #4094, R1
1706          007062 013702 001372          NEXT: JSR PC, RANDY          ; GET RANDOM NUMBER
1707          007066 042702 177760          MOV R1A, R2
1708          007072 001402          BIC #177760, R2          ; MASK IT TO 4 BITS ONLY
1709          007074 005302          BEQ CONVR
1710          007076 001376          DELAY3: DEC R2          ; STALL
1711          007100 005277 172212          BNE DELAY3          ; TIME
1712          007104 000001          CONVR: INC R2STREG          ; START CONVERSION
1713          007106 017702 172210          WAIT MOV #ADDBUFF, R2          ; GET CONVERTED VALUE
1714          007112 001413          BEQ DELAY1          ; IGNORE IF =0
1715          007114 020227 007777          CMP R2, #7777          ; IGNORE IF =7777
1716          007120 001413          BEQ DELAY2
1717          007122 006302          ASL R2
1718          007124 005262 017756          INC BUFFER(R2)          ; MAKE HISTOGRAM
1719          007130 100013          BPL OKAY
1720          007132 012762 077777 017756          MOV #077777, BUFFER(R2)          ; PREVENT OVERFLOW
1721          007140 000407          BR OKAY
1722          007142 020227 007777          DELAY1: CMP R2, #7777          ; EQUALIZE LOOP TIME
1723          007146 001400          BEQ DELAY2          ; WITH DUMMY INSTR.
1724          007150 005201          DELAY2: INC R1
1725          007152 005263 001352          INC TEMP(R3,
1726          007156 100403          BMI NOTOK
1727          007160 005301          OKAY: DEC R1
1728          007162 001335          BNE NEXT
1729          007164 000403          BR AROUND
1730          007166 005037 001352          NOTOK: CLR TEMP
1731          007172 000772          BR OKAY

```

F04

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 44
INTERCHANNEL SETTLING TEST, 1 EDGE

1733	007174	005300		AROUND:	DEC	R0	
1734	007176	001325			BNE	AGAIN	
1735	007200	012700	007776		MOV	#4094, R0	
1736	007204	012701	017760		MOV	#BUFFER+2, R1	
1737	007210	012102		READ:	MOV	(R1)+, R2	; GET STATE WIDTH
1738	007212	006202			ASR	R2	; 1 LSB = 800.
1739	007214	006202			ASR	R2	
1740	007216	006202			ASR	R2	
1741	007220	005502			ADC	R2	; 1 LSB = 100.
1742	007222	020227	000310		CMP	R2, #200.	; OUT OF RANGE?
1743	007226	002403			BLT	INRNGE	
1744	007230	005237	001430		INC	OUT	; YES - INCREMENT COUNTER
1745	007234	000423			BR	TYPBAD	
1746	007236	006302		INRNGE:	ASL	R2	
1747	007240	005262	017136		INC	DIST(R2)	; MAKE STATE WIDTH DISTRIBUTION
1748	007244	006202			ASR	R2	
1749	007246	020227	000062		CMP	R2, #50.	; IS IT 1/2 LSB?
1750	007252	002007			BGE	NOTNAR	
1751	007254	005237	001344		INC	NARROW	
1752	007260	005702			TST	R2	; IS IT A SKIPPED STATE?
1753	007262	001002			BNE	31\$	
1754	007264	005237	001350		INC	SKIPST	
1755	007270	000405		31\$:	BR	TYPBAD	
1756	007272	020227	000226	NOTNAR:	CMP	R2, #150.	; IS IT 1.5 LSB?
1757	007276	003426			BLE	LAST	
1758	007300	005237	001342		INC	WIDE	
1759	007304	005737	001346	TYPBAD:	TST	FIRST	
1760	007310	001004			BNE	60\$	
1761	007312	005237	001346		INC	FIRST	
1762	007316	104400	012201		TYPE	STATE	
1763	007322	010103		60\$:	MOV	R1, R3	
1764	007324	162703	017760		SUB	#BUFFER+2, R3	
1765	007330	006203			ASR	R3	
1766	007332	010346			MOV	R3, -(SP)	; SAVE R3 FOR TYPEOUT
1767							; TYPE STATE
1768	007334	104402			TYPOS		; GO TYPE--OCTAL ASCII
1769	007336	004			.BYTE	4	; TYPE 4 DIGIT(S)
1770	007337	001			.BYTE	1	; TYPE LEADING ZEROS
1771	007340	104400	012175		TYPE	DASH	
1772	007344	004737	011406		JSR	PC, DECTYP	
1773	007350	104400	012166		TYPE	LSBMSG	
1774	007354	005300		LAST:	DEC	R0	
1775	007356	001314			BNE	READ	
1776	007360	112737	000177	014465	MOVB	#177, DECPNT	
1777	007366	013702	001350		MOV	SKIPST, R2	; GET NO. OF SKIPPED STATES
1778	007372	004737	011406		JSR	PC, DECTYP	; TYPE IT
1779	007376	104400	012424		TYPE	SKPMSG	; TYPE MESSAGE
1780	007402	005737	001350		TST	SKIPST	
1781	007406	001403			BEQ	1\$	
1782	007410	104400	012407		TYPE	ERMSG	; TYPE "ERROR"
1783	007414	000402			BR	NAR	
1784	007416	104400	012376	1\$:	TYPE	.OKMSG	; TYPE #OK#

G04

MAINDEC-11-DZADL-B
DZADLB.P11MACY:1 27(665) 2-DEC-76 16:29 PAGE 45
INTERCHANNEL SETTLING TEST. 1 EDGE

1785	007422	013702	001344	NAR:	MOV	NARROW R2	:GET NO. OF NARROW STATES
1796	007426	004737	011406		JSR	PC,DECTYP	:TYPE IT
1787	007432	104400	012446		TYPE	,NARMSG	;TYPE MESSAGE
1788	007436	013702	001342		MOV	WIDE R2	
1789	007442	063702	001430		ADD	OUT R2	
1790	007446	004737	011406		JSR	PC,DECTYP	:TYPE NO. OF WIDE STATES
1791	007452	104400	012505		TYPE	,WIDMSG	;TYPE MESSAGE
1792	007456	013702	001430		MOV	OUT R2	
1793	007462	004737	011406		JSR	PC,DECTYP	:TYPE NO. OF STATES OUTSIDE 2 LSB
1794	007466	104400	012544		TYPE	,OUTMSG	;TYPE MESSAGE
1795	007472	005737	001430		TST	OUT	
1796	007476	001403			BEQ	11\$	
1797	007500	104400	012407		TYPE	,ERMSG	;TYPE "ERROR"
1798	007504	000402			BR	HALF	
1799	007506	104400	012376	11\$:	TYPE	,OKMSG	;TYPE "OK"
1800	007512	013702	001344	HALF:	MOV	NARROW R2	
1801	007516	063702	001342		ADD	WIDE R2	
1802	007522	063702	001430		ADD	OUT R2	
1803	007526	010200			MOV	R2,R0	
1804	007530	004737	011406		JSR	PC,DECTYP	:TYPE NO. OF STATES OUTSIDE LIMITS
1805	007534	112737	000056	014465	MOVB	#56,DECPNT	
1806	007542	104400	012577		TYPE	,HAFMSG	
1807	007546	020027	000051		CMP	R0,#41.	;COMPARE IT TO NOMINAL
1808	007552	003403			BLE	21\$	
1809	007554	104400	012407		TYPE	,ERMSG	;TYPE "ERROR"
1810	007560	000402			BR	\$WDIST.	
1811	007562	104400	012376	21\$:	TYPE	,OKMSG	;TYPE "OK"
1812	007566	005737	001404	\$WDIST:	TST	FLAG	;VT55?
1813	007572	011426			BEQ	RELACC	
1814	007574	004737	010254		JSR	PC,DELCLR	;WAIT AWHILE, THEN CLEAR VT55
1815	007600	104400	013054		TYPE	,MSG16	
1816	007604	104400	013655		TYPE	,BUFF1	;TYPE BUFF1-PRINT GRID
1817	007610	012700	017136		MOV	\$DIST,R0	;POINTER TO STATE WIDTH DISTRIBUTION
1818	007614	012701	000310		MOV	#200.,R1	;GO 200. TIMES UP TO 2 LSB
1819	007620	012002		NXTY1:	MOV	(R0)+,R2	
1820	007622	004737	011304		JSR	PC,LOADY	
1821	007626	005002			CLR	R2	
1822	007630	004737	011304		JSR	PC,LOADY	
1823	007634	005301			DEC	R1	
1824	007636	001370			BNE	NXTY1	
1825	007640	104400	013600		TYPE	,C2	;TYPE ASCII STRING
1826	007644	004737	010254		JSR	PC,DELCLR	

H04

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 46
INTERCHANNEL SETTLING TEST, 1 EDGE

```

1828      :CHANGE HISTOGRAM ERROR TO RELATIVE ACCURACY ERROR
1829
1830      RELACC: CLR      R1      ;RUNNING ERROR = 0
1831             CLR      R3      ;MAXIMUM ERROR = 0
1832             TYPE     ,MSG21
1833             MOV      #BUFFER+2,R0
1834      NXTSTA: MOV      (R0),R2   ;STATE WIDTH = R2
1835             SUB      #800.,R2   ;STATE WIDTH ERROR IN R2
1836             ADD      R2,R1      ;UPDATE RUNNING ERROR
1837             MOV      R1,(R0)+   ;SAVE IN BUFFER
1838             MOV      R1,R4      ;SAVE IN R4 ALSO
1839             BPL      PLUS       ;IS IT POSITIVE?
1840             NEG      R4         ;NO - MAKE IT POSITIVE
1841      PLUS:  CMP      R4,R3      ;CHECK AGAINST PREVIOUS MAX. ERROR
1842             BLE      NOTNEW     ;NOT A NEW MAXIMUM
1843             MOV      R4,R3      ;UPDATE MAXIMUM IN R3
1844             MOV      R0,R5
1845             SUB      #BUFFER+2,R5
1846             ASR      R5         ;R5=EDGE VALUE AT MAX. RELACC
1847      NOTNEW: CMP      R0,#BUFFER+8190. ;DONE
1848             BNE      NXTSTA     ;NO - REPEAT
1849             ASR      R3         ;RESCALE FROM 1 LSB = 800. SCALING
1850             ASR      R3         ;TO 1 LSB = 100. SCALING
1851             ASR      R3
1852             ADC      R3
1853             MOV      R3,R2
1854             JSR      PC,DECTYP
1855             TYPE     ,LINEA
1856             MOV      R5,-(SP)   ;:SAVE R5 FOR TYPEOUT
1857                                     ;:TYPE VALUE
1858             TYP0S
1859             .BYTE    4          ;:GO TYPE--OCTAL ASCII
1860             .BYTE    1          ;:TYPE 4 DIGIT(S)
1861             TYPE     ,SLASH     ;:TYPE LEADING ZEROS
1862             INC      R5         ;:PRINT '/'
1863             MOV      R5,-(SP)   ;:SAVE R5 FOR TYPEOUT
1864                                     ;:TYPE VALUE
1865             TYP0S
1866             .BYTE    4          ;:GO TYPE--OCTAL ASCII
1867             .BYTE    1          ;:TYPE 4 DIGIT(S)
1868             .BYTE    1          ;:TYPE LEADING ZEROS
1869             CMP      R3,VLIN
1870             BLE      41$
1871             TYPE     ,ERMSG
1872             BR       42$
1873      41$:  TYPE     ,OKMSG
1874             TST      FLAG      ;VT55?
1875             BEQ      L02
1876             MOV      #BUFFER,R0
1877             MOV      #4096.,R1

```

MAINDEC-11-DZADL-B
DZADLB.P11

MACV: 27(665) 2-DEC-76 16:29 PAGE 47
INTERCHANNEL SETTLING TEST, 1 EDGE

1877	010032	011002	GETDAT:	MOV	(R0),R2	:GET RELATIVE ACCURACY ERROR SCALED 1LSB = 800.
1878	010034	006202		ASR	R2	:RESCALE IT TO 1 LSB = 100.
1879	010036	006202		ASR	R2	
1880	010040	006202		ASR	R2	
1891	010042	005502		ADC	R2	
1892	010044	062702	000166	ADD	#118.,R2	:AND MOVE IT TO MID-SCREEN
1893	010050	010220		MOV	R2,(R0)+	:PUT IT BACK INTO BUFFER
1894	010052	005301		DEC	R1	
1895	010054	001366		BNE	GETDAT	
1896	010056	012700	017756	MOV	#BUFFER,R0	
1897	010062	012704	017756	MOV	#BUFFER,R4	
1898	010066	012705	017760	MOV	#BUFFER+2,R5	
1899	010072	012701	001000	MOV	#512.,R1	
1890	010076	012702	000007	NXTB:	MOV	#7.,R2
1891	010102	012003		MOV	(R0)+,R3	
1892	010104	010337	001420	MOV	R3,MIN	:MINIMUM
1893	010110	010337	001424	MOV	R3,MAX	:MAXIMUM
1894	010114	012003		NXTCMP:	MOV	(R0)+,R3
1895	010116	020337	001420	CMP	R3,MIN	
1896	010122	002002		BGE	MAXTST	
1897	010124	010337	001420	MOV	R3,MIN	:NEW MINIMUM
1898	010130	020337	001424	MAXTST:	CMP	R3,MAX
1899	010134	003402		BLE	TSTB	
1900	010136	010337	001424	MOV	R3,MAX	:NEW MAXIMUM
1901	010142	005302		TSTB:	DEC	R2
1902	010144	001363		BNE	NXTCMP	
1903	010146	013724	001420	MOV	MIN,(R4)+	
1904	010152	013725	001424	MOV	MAX,(R5)+	
1905	010156	022425		CMP	(R4)+,(R5)+	:BUMP EACH ONCE MORE
1906	010160	005301		DEC	R1	
1907	010162	001345		BNE	NXTB	
1908	010164	104400	012762	TYPE	,MSG18	
1909	010170	104400	013703	TYPE	,BUFF2	:TYPE BUFF2
1910	010174	012700	017756	MOV	#BUFFER,R0	
1911	010200	004737	010232	JSR	PC,LOAD	
1912	010204	104400	013603	TYPE	,C3	:TYPE ASCIZ STRING
1913	010210	012700	017760	MOV	#BUFFER+2,R0	
1914	010214	004737	010232	JSR	PC,LOAD	
1915	010220	104400	013600	TYPE	,C2	:TYPE ASCIZ STRING
1916	010224	004737	010254	JSR	PC,DELCLR	
1917	010230	000207		LO2:	RTS	
1918	010232	012701	001000	LOAD:	MOV	#512.,R1
1919	010236	012002		LOADC:	MOV	(R0)+,R2
1920	010240	005720		TST	(R0)+	
1921	010242	004737	011304	JSR	PC,LOADY	
1922	010246	005301		DEC	R1	
1923	010250	001372		BNE	LOADC	
1924	010252	000207		RTS	PC	

J04

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 48
INTERCHANNEL SETTLING TEST, 1 EDGE

```

1925 010254 005000 DELCLR: CLR RO
1926 010256 012701 000020 MOV #20,R1 ;DELAY BEFORE CLEANING SCREEN
1927 010262 005300 15: DEC RO
1928 010264 001376 BNE 15
1929 010266 005301 DEC R1
1930 010270 001374 BNE 15
1931 010272 032777 010000 170640 BIT #BIT12,2SWR ;TEST FOR HALT FOR DISPLAY
1932 010300 001401 BEQ 25 ;:DON'T HALT FOR DISPLAY
1933 010302 000000 HALT
1934 010304 104400 013723 25: TYPE VTINIT
1935 010310 000207 RTS PC
1936 010312 013537 001366 ;:NOISE SUBROUTINE:;
1937 010316 013737 001366 001364 NOITST: MOV 2(R5)+,CHANL ;LOAD CHANNEL
1938 010324 004737 006336 MOV CHANL,DUMMY ;LOAD DUMMY CHANNEL
1939 010330 004737 010504 JSR PC,GETEDG ;GET EDGE VALUE
1940 010334 012737 000001 006514 JSR PC,NOIA ;GET RMS AND PEAK VALUES
1941 010342 004737 010350 JSR PC,EDGFLG ;TYPE RMS AND PEAK VALUES
1942 010346 000205 RTS R5
1943
1944
1945
1946
1947
1948
1949
1950 010350 104400 012273 ;:TYPE RMS AND PEAK VALUES:;
1951 010354 005737 001400 TYPRP: TYPE NOI
1952 010360 100002 TST RMS
1953 010362 005037 001400 BPL POSRMS ;RMS<0.SET RMS=0
1954 010366 005737 001402 CLR RMS
1955 010372 100002 POSRMS: TST PEAK
1956 010374 005037 001402 BPL POSPEA ;PEAK<0.SET PEAK=0
1957 010400 013702 001400 CLR PEAK
1958 010404 004737 011406 POSPEA: MOV RMS,R2
1959 010410 104400 012646 JSR PC,DECTYP
1960 010414 013702 001402 TYPE MESR
1961 010420 004737 011406 MOV PEAK,R2
1962 010424 104400 012661 JSR PC,DECTYP
1963 010430 004737 006452 TYPE MESP
1964 010434 104400 012303 JSR PC,TYPEDG
1965 010440 013746 001366 TYPE CHAN
1966 010444 104402 TYPOS CHANL,-(SP) ;:SAVE CHANL FOR TYPEOUT
1967 010446 002 .BYTE 2 ;:TYPE CHANL
1968 010447 000 .BYTE 0 ;:GO TYPE--OCTAL ASCII
1969 010450 023737 001400 011644 CMP RMS,VNR ;:TYPE 2 DIGIT(S)
1970 010456 003007 001402 011646 BGT ER ;:SUPPRESS LEADING ZEROS
1971 010460 023737 001402 011646 CMP PEAK,VNP ;:WITHIN LIMITS?
1972 010466 003003 012376 BGT ER ;:WITHIN LIMITS?
1973 010470 104400 TYPE OKMSG
1974 010474 000207 RTS PC
1975 010476 104400 012407 ER: TYPE ERMSG
1976 010502 000207 RTS PC

```

K04

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 49
INTERCHANNEL SETTLING TEST, 1 EDGE

```

1978      ::SUBROUTINES FOR NOISE TEST::
1979      NOIA: CLR      RMS      ;CLEAR RMS VLAUE
1980      010504 005037 001400      CLR      PEAK      ;CLEAR PEAK VALUE
1981      010510 005037 001402      JSR      R5,SARSUB    ;DO SAR ROUTINE AT 16%
1982      010514 004537 006516      16.
1983      010520 000020      ADD      DAC,RMS      ;ADD RESULT TO RMS
1984      010522 063737 001410 001400 JSR      R5,SARSUB    ;DO SAR ROUTINE AT 84%
1985      010530 004537 006516      84.
1986      010534 000124      SUB      DAC,RMS      ;SUBTRACT RESULT FROM RMS
1987      010536 163737 001410 001400 JSR      R5,SARSUB    ;DO SAR ROUTINE AT 1%
1988      010544 004537 006516      1.
1989      010550 000001      ADD      DAC,PEAK      ;ADD RESULT TO PEAK
1990      010552 063737 001410 001402 JSR      R5,SARSUB    ;DO SAR ROUTINE AT 99%
1991      010556 004537 006516      99.
1992      010564 000143      SUB      DAC,PEAK      ;SUBTRACT RESULT FROM PEAK
1993      010566 163737 001410 001402 RTS      PC      ;RETURN
1994      010574 000207
1995      010576 012537 001366      NOIB: MOV      (R5)+,CHANL    ;GET CHANNEL VALUE
1996      010602 063737 001336 001366 ADD      BASECH,CHANL
1997      010610 013737 001366 001364 MOV      CHANL,DUMMY    ;LOAD DUMMY CHANNEL
1998      010616 004737 006236      JSR      PC,GETEDG    ;GET EDGE VALUES
1999      010622 005037 001400      CLR      RMS      ;CLEAR RMS VALUE
2000      010626 005037 001402      CLR      PEAK      ;CLEAR PEAK VALUE
2001      010632 012737 000010 010720 MOV      #10,10$    ;SET UP COUNTER
2002      010640 004737 010514      1$: JSR      PC,NOI1    ;GET NOISE VALUES
2003      010644 005237 001414      INC      EDGE
2004      010650 005337 010720      DEC      10$
2005      010654 001371      BNE      1$      ;REPEAT 8 TIMES
2006      010656 162737 000010 001414 SUB      #10,EDGE    ;SCALE IT TO 1 LSB=100.
2007      010664 006237 001400      ASR      RMS
2008      010670 005537 001400      ADC      RMS
2009      010674 006237 001402      ASR      PEAK
2010      010700 005537 001402      ADC      PEAK
2011      010704 012737 000010 006514 MOV      #8,EDGEFLG
2012      010712 004737 010350      JSR      PC,TYPRP
2013      010716 000205      RTS      R5      ;RETURN
2014      010720 000000      1CS: 0      ;COUNTER
2015
2016
2017      ::RANDOM NUMBER GENERATOR::
2018      010722 063737 001374 001372 RANDY: ADD      RNB,RNA
2019      010730 063737 001376 001372 ADD      RNC,RNA
2020      010736 005537 001372      ADC      RNA
2021      010742 063737 001372 001374 ADD      RNA,RNB
2022      010750 063737 001376 001374 ADD      RNC,RNB
2023      010756 005537 001374      ADC      RNB
2024      010762 063737 001372 001376 ADD      RNA,RNC
2025      010770 063737 001374 001376 ADD      RNB,RNC
2026      010776 005537 001376      ADC      RNC
2027      011002 000207      RTS      PC

```

TYPE RESULTS

;RETURN
;COUNTER

L04

MAINDEC-11-DZADL-B
DZADL.B.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 50
INTERCHANNEL SETTLING TEST, 1 EDGE

```

2028      ::ROUTINE TO AVERAGE 8 CONVERSIONS::
2029      CONVRT: MOV      (R5)+,RO      ;GET CHANNEL VALUE
2030      ADD      BASECH,RO
2031      MOV      RO,CHANL
2032      SWAB     RO
2033      CLR      TEMP
2034      MOV      RO,ASTREG      ;LOAD CHANNEL INTO MIX BITS
2035      MOV      #10000,RO
2036      2$: DEC     RO
2037      BNE     2$
2038      MOV      #RETURN,AVECTOR      ;LOAD VECTOR
2039      MOV      #10,RO      ;SET UP COUNTER
2040      BISB     #101,ASTREG      ;SET INTRPT. EN., START CONV.
2041      WAIT
2042      ADD      @ADBUFF,TEMP      ;WAIT FOR CONVERSION
2043      DEC     RO      ;READ BUFFER
2044      BNE     1$
2045      ASR      TEMP      ;DO 8 TIMES
2046      ASR      TEMP      ;AVERAGE VALUE
2047      ASR      TEMP
2048      ASR      TEMP
2049      ADC      TEMP
2050      RTS      R5      ;RETURN
2051
2052      ;COMPARE $GDDAT AND $BDDAT::
2053      COMPAR: MOV      (R5)+,$GDDAT      ;GET GOOD DATA
2054      MOV      @R5+,SPREAD      ;GET SPREAD
2055      MOV      TEMP,$BDDAT      ;GET BAD(ACTUAL) DATA
2056      MOV      $BDDAT,R1
2057      MOV      $GDDAT,RO
2058      SUB      R1,RO      ;GET DIFFERENCE
2059      BPL      7$
2060      NEG      RO
2061      7$: CMP      RO,SPREAD      ;COMPARE IT TO SPREAD
2062      BGT      10$      ;GO TO ERROR PRINTOUT
2063      TST      (R5)+      ;BUMP RETURN POINTER AROUND ERROR CALL
2064      RTS      R5

```

M04

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665) 2-DEC-76 16:29 PAGE 51
INTERCHANNEL SETTLING TEST, 1 EDGE

```

2064      011164 005737 001202      ::SUBROUTINE TO TYPE INTRPT. TST MSG.:
2065      011170 001021      DUMW:  TST      $PASS
2066      011172 012737 011234 001110      BNE      20$
2067      011200 012737 011234 001106      MOV      #20$, $LPERR
2068      011206 104400 013633      MOV      #20$, $LPADR
2069      011212 010046      TYPE      METST      ;TYPE ASCII STRING
2070      011214 104402      MOV      R0, -(SP)      ;SAVE R0 FOR TYPEOUT
2071      011216 002      TYPOS      ;TYPE TEST NO.
2072      011217 000      .BYTE      2      ;GO TYPE--OCTAL ASCII
2073      011220 104400 012723      .BYTE      0      ;TYPE 2 DIGIT(S)
2074      011224 013746 001316      TYPE      ONAD      ;SUPPRESS LEADING ZEROS
2075      011230 104402      MOV      $STREG, -(SP)      ;SAVE STREG FOR TYPEOUT
2076      011232 006      TYPOS      ;TYPE BUS ADDRESS
2077      011233 001      .BYTE      6      ;GO TYPE--OCTAL ASCII
2078      011234 000207      .BYTE      1      ;TYPE 6 DIGITS
2079      011236 005737 001202      20$:  RTS      PC      ;TYPE LEADING ZEROS
2080      011242 001010      DUMC:  TST      $PASS
2081      011244 012737 011264 001110      BNE      30$
2082      011252 012737 011264 001106      MOV      #30$, $LPERR
2083      011260 104400 012320      MOV      #30$, $LPADR
2084      011264 000207      TYPE      DONE
2085      011266 000207      30$:  RTS      PC

```

N04

MAINDEC-11-DZADL-B
DZADLB.P11

MACY11 27(665) 2-DEC-76 16:29 PAGE 52
INTERCHANNEL SETTLING TEST, 1 EDGE

```

2089 ;SUBROUTINE TO RESET & SET INTRPT. EN.;
2090 RST: RESET
2091 011266 000005 000100 167646 BIS #100,2STKS
2092 011270 052777 177776 CLR PSW
2093 011276 005037 RTS PC
2094 011302 000207
2095 ;SUBROUTINE LOADY:
2096 011304 005702 LOADY: TST R2 ;ROUTINE TO LOAD VLAUE INTO R2
2097 011306 100001 BPL PLUSR2 ;AS A VTSS Y-VALUE
2098 011310 005002 CLR R2
2099 011312 020227 000353 PLJSR2: CMP R2,#235.
2100 011316 002402 BLT LESS
2101 011320 012702 000353 MOV #235.,R2
2102 011324 010203 LESS: MOV R2,R3
2103 011326 042702 177740 BIC #177740,R2
2104 011332 052702 000040 BIS #40,R2
2105 011336 105777 167606 B10: TSTB 2STPS ;PRINT CHARACTER
2106 011342 100375 BPL B10
2107 011344 110277 167602 MOVB R2,2STPB
2108 011350 006203 ASR R3
2109 011352 006203 ASR R3
2110 011354 006203 ASR R3
2111 011356 006203 ASR R3
2112 011360 006203 ASR R3
2113 011362 042703 177770 BIC #177770,R3
2114 011366 052703 000040 BIS #40,R3
2115 011372 105777 167552 B11: TSTB 2STPS ;PRINT CHARACTER
2116 011376 100375 BPL B11
2117 011400 110377 167546 MOVB R3,2STPB
2118 011404 000207 RTS PC
2119
2120

```

B05

NOV-11-0240L-B
C240.B.F.11

NOV-11 27:6651 2-DEC-76 16:29 PAGE 53
INTERCHANNEL SETTLING TEST, 1 EDGE

0011406 005702
0011410 100003
0011412 104400
0011416 005402
0011420 020227
0011424 003402
0011428 012702
0011432 105037
0011436 105037
0011440 105037
0011444 005702
0011448 001424
0011452 005302
0011456 105237
0011460 123727
0011464 001367
0011468 105037
0011472 105237
0011476 123727
0011480 001357
0011484 105037
0011488 105237
0011492 000752
0011496 152737
0011500 152737
0011504 152737
0011508 104400
0011512 000207
0011552 012701
0011556 005737
0011560 001403
0011564 012702
0011568 000410
0011572 005737
0011576 001003
0011580 012702
0011584 000402
0011588 012702
0011592 012221
0011596 005711
0011600 100375
0011604 000207

012135
001747
001747
014467
014466
014464
014467
014467 000012
014467
014466 000012
014466
014464
000060 014464
000060 014466
000060 014467
014464
011644
001336
011676
001422
011656
011666

```

::SUBROUTINE TO TYPE DECIMAL VALUE::
::IN R2 AS X.XX:
DECTYP: TST R2
BPL POS
TYPE MINUS
NEG R2
POS: CMP R2, #999.
BLE OKAYD
MOV #999., R2
OKAYD: CLRB ONES
CLRB TENS
CLRB HUNS
TESTR2: TST R2
BEQ TYP0UT
DEC R2
INCB ONES
CMPB ONES, #10.
BNE TESTR2
CLRB ONES
INCB TENS
CMPB TENS, #10.
BNE TESTR2
CLRB TENS
INCB HUNS
BR TESTR2
TYP0UT: B1SB #60, HUNS
B1SB #60, TENS
B1SB #60, ONES
TYPE HUNS
RTS PC
WFA0J: MOV #VNR, R1
TST BASECH
BEQ 1$
MOV #VARLT3, R2
BR 3$
1$: TST WFTST
BNE 2$
MOV #VARLT1, R2
BR 3$
2$: MOV #VARLT2, R2
3$: MOV (R2)+, (R1)+
TST (R1)
BPL 3$
RTS PC

```

```

;TEST VALUE TO BE TYPED
;TYPE MINUS SIGN
;>999. REPLACE IT WITH 999.
;CLEAR ONES
;CLEAR TENS
;CLEAR HUNS
;CONVERT VALUE TO A DECIMAL VALUE
;PREPARE FOR TYP0UT
;TYPE VALUE
;SUBROUTINE TO SET UP LIMITS
;TESTING AN AMI1K'
;BASECH NOT ZERO, USE AMI1K LIMITS
;WFTST=0, USE NORMAL LIMITS
;WFTST=1, USE OPTION AREA LIMITS

```


C05

MAINDEC-11-02AOL-B
02AOLB.P11

MACY!!! 27.665) 2-DEC-76 16:29 PAGE 54
INTERCHANNEL SETTLING TEST. 1 EDGE

:TOLERANCE VALUES FOR FUNCTIONAL TESTS

011622 000001
011624 000002
011626 000010
011630 000050
011632 000144
011634 000115
011636 000240
011640 000005
011642 000062

011644 000000
011646 000000
011650 000000
011652 000000
011654 100000

011656 000031
011660 000310
011662 000144
011664 000144

011666 000027
011670 000226
011672 000132
011674 000132

011676 000062
011700 000310
011702 000226
011704 000226

011706 052777 000100 167230
011708 000177 000000
011720 001620

V1: 1
V2: 2
V10: 10
V50: 50
V144: 144
V115: 115
V240: 240
V5: 5
V500: 50.

VNR: 0
VNP: 0
VSET: 0
VLIN: 0
BIT15

VARLT1: 25.
200.
100.
100.

VARLT2: 23.
150.
90.
90.

VARLT3: 50.
200.
150.
150.

AGATST: BIS
JMP
AGTST: BEGIN

:RMS NOISE LIMIT
:PEAK NOISE LIMIT
:INTER-CHANNEL SETTLING LIMIT
:RELATIVE ACCURACY ERROR LIMIT

:.25 LSB, NORMAL LIMITS FOR SYSTEM
:.2. LSB, INTEGRATION AND FIELD USE ON SPEC TESTS
:.1 LSB
:.1 LSB

:.23 LSB, TIGHTER LIMITS FOR OPTION
:.1.5 LSB, AREA USE ON SPEC TESTS
:.9 LSB
:.9 LSB

:.5 LSB, LIMITS FOR AMIUK TESTING
:.2. LSB
:.1.5 LSB
:.1.5 LSB

8100.25TKS
2AGTST

005

MAINDEC-11-DZADL-B MACY:1 27(665) 2-DEC-76 16:29 PAGE 55
DZADL.B.P11 ENC OF PASS ROUTINE

2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210 011722
2211 011722 000240
2212 011724 005037 001102
2213 011730 005037 001160
2214 011734 005237 001202
2215 011740 042737 100000 001202
2216 011746 005327
2217 011750 000001
2218 011752 003015
2219 011754 012737
2220 011756 000001
2221 011760 011750
2222 011762 104400 012015
2223 011766 013700 000042
2224 011772 001405
2225 011774 000005
2226 011776 004710
2227 012000 000240
2228 012002 000240
2229 012004 000240
2230 012006
2231 012006 000137
2232 012010 011706
2233 012012 377 377 000
2234 012015 015 042412 042116
2235 012022 050040 051501 000123
2236

.SBTTL END OF PASS ROUTINE

: INCREMENT THE PASS NUMBER (SPASS)
: TYPE "END PASS"
: IF THERES A MONITOR GO TO IT
: IF THERE ISN'T JUMP TO AGATST
: IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
: SENDMG CAN BE CHANGED TO 7.

SECP:

NOP
CLR \$TSTNM : ZERO THE TEST NUMBER
CLR \$TIMES : ZERO THE NUMBER OF ITERATIONS
INC \$PASS : INCREMENT THE PASS NUMBER
BIC #100000,\$PASS : DON'T ALLOW A NEG. NUMBER
DEC (PC)+ : LOOP?
SEOPCT: .WORD 1
BGT \$DOAGN : YES
MOV (PC)+,2(PC)+ : RESTORE COUNTER
SENDCT: .WORD 1
SEOPCT
TYPE \$SENDMG : TYPE "END PASS"
\$GET42: MOV \$42,R0 : GET MONITOR ADDRESS
BEQ \$DOAGN : BRANCH IF NO MONITOR
RESET : CLEAR THE WORLD
SENDAD: JSR PC,(R0) : GO TO MONITOR
NOP : SAVE ROOM
NOP : FOR
NOP : ACT11
\$DOAGN:
JMP 2(PC)+ : RETURN
\$RTNAD: .WORD AGATST
\$ENULL: .BYTE -1,-1,0 : NULL CHARACTER STRING
\$SENDMG: .ASCIIZ <15><12>/END PASS

E05

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665)
ASCII MESSAGES

2-DEC-76 16:29 PAGE 56

2237				
2238	012030	005015	047516	051511
2239	012036	020105	042524	052123
2240	012044	026455	000040	
2241	012050	005015	042523	052124
2242	012056	044514	043516	052040
2243	012064	051505	026524	020055
2244	012072	054524	042520	042040
2245	012100	051505	051111	042105
2246	012106	023440	051106	046517
2247	012114	020047	044103	047101
2248	012122	042516	020114	020046
2249	012130	051103	020072	000
2250	012137	055	000	
2251	012141	077	000	
2252	012144	136	101	040
2253	012144	040	000	
2254	012146	136	103	040
2255	012151	040	000	
2256	012153	136	107	015
2257	012156	055	123	127
2258	012161	122	105	107
2259	012164	072	000	
2260	012166	046040	041123	005015
2261	012174	000		
2262	012175	055	020055	000
2263	012201	123	040524	042524
2264	012206	026455	053440	042111
2265	012214	044124	005015	000
2266	012221	103	000110	
2267	012224	020040	020040	000
2268	012231	040	051514	020102
2269	012236	047117	041440	000110
2270	012244	051440	052105	046124
2271	012252	047111	020107	051106
2272	012260	046517	041440	050110
2273	012266	040440	020124	000
2274	012273	116	044517	042523
2275	012300	020072	000	
2276	012303	040	047117	041440
2277	012310	040510	047116	046105
2278	012316	000040		
2279	012320	020040	020040	047504
2280	012326	042516	005015	000
2281	012333	057	000	
2282	012335	124	050131	020105
2283	012342	042504	044523	042522
2284	012350	020104	052047	023517
2285	012356	041440	040510	047116
2286	012364	046105	023040	041440
2287	012372	035122	000040	
2288	012376	020040	020040	045517
2289	012404	005015	000	

.SBTTL ASCII MESSAGES

NOIMSG: .ASCIZ (15)(12)/NOISE TEST--

SETMSG: .ASCIZ (15)(12)/SETTLING TEST-- TYPE DESIRED 'FROM' CHANNEL & CR: /

MINUS: .BYTE 55,0

QUEST: .BYTE 77,0

RMSG: .BYTE 136,101,40,40,0

CMMSG: .BYTE 136,103,40,40,0

GMSG: .BYTE 136,107,15,12,123,127,122,105,107,72,0

LSBMSG: .ASCIZ / LSB/(15)(12)

DASH: .ASCIZ /--/

STATE: .ASCIZ /STATE-- WIDTH/(15)(12)

CH: .ASCIZ /CH/

SPACE: .ASCIZ / /

LSB: .ASCIZ / LSB ON CH/

SETCH: .ASCIZ / SETTLING FROM CH/

ATMSG: .ASCIZ / AT /

NOI: .ASCIZ /NOISE: /

CHAN: .ASCIZ / ON CHANNEL /

DONE: .ASCIZ / DONE/(15)(12)

SLASH: .ASCIZ #/

TOMSG: .ASCIZ /TYPE DESIRED 'TO' CHANNEL & CR: /

OKMSG: .ASCIZ / OK/(15)(12)

2290	012407	040	025052	051105
2291	012414	047522	025122	006452
2292	012422	000012		
2293	012424	051440	044513	050120
2294	012432	042105	051440	040524
2295	012440	042524	051450	000051
2296	012446	047040	051101	047522
2297	012454	020127	026050	030440
2298	012462	031057	046040	041123
2299	012470	020051	052123	052101
2300	012476	024105	024523	005015
2301	012504	000		
2302	012505	040	044527	042504
2303	012512	024040	020076	020061
2304	012520	027461	020062	051514
2305	012526	024502	051440	040524
2306	012534	042524	051450	006451
2307	012542	000012		
2308	012544	051440	040524	042524
2309	012552	051450	020051	044527
2310	012560	042504	020122	044124
2311	012566	047101	031040	046040
2312	012574	041123	000	
2313	012577	040	052123	052101
2314	012604	026505	044527	052104
2315	012612	024110	024523	047440
2316	012620	052125	044523	042504
2317	012626	025440	047440	020122
2318	012634	020055	027461	020062
2319	012642	051514	000102	
2320	012646	046040	041123	051040
2321	012654	051515	020054	000
2322	012661	040	051514	020102
2323	012666	042520	045501	040440
2324	012674	020124	000	
2325	012677	015	042412	042116
2326	012704	047440	020106	047514
2327	012712	044507	020103	042524
2328	012720	052123	123	
2329	012723	040	047117	040440
2330	012730	030504	045461	040440
2331	012736	020124	000	
2332	012741	040	042101	030461
2333	012746	023513	020123	047506
2334	012754	047125	036504	000012
2335	012762	005012	025412	027461
2336	012770	020062	051514	006502
2337	012776	005012	005012	005012
2338	013004	005012	005012	005012
2339	013012	030455	031057	051514
2340	013020	000102		

ERMSG: .ASCIZ **ERROR**/(15)(12)

SKPMSG: .ASCIZ / SKIPPED STATE(S)/

NARMSG: .ASCIZ * NARROW (< 1/2 LSB) STATE(S)*(<15>(<12>

WIDMSG: .ASCII * WIDE (1 1/2 LSB) STATE(S) * <15><12>

CJTMMSG: .ASCIZ / STATE(S) WIDER THAN 2 LSB/

HAFMSG: .ASCIZ * STATE-WIDTH(S) OUTSIDE + OR - 1/2 LSB*

MESR: .ASCIZ / LSB RMS, /

MESP: .ASCII / LSB PEAK AT /

MENC: .ASCII (15)(12)/END OF LOGIC TESTS/

CNAD: .ASCIZ / ON ADLIK AT /

MSGSC: .ASCIZ / AD11K'S FOUND/ <15> <12>

MSG18: .ASCII (12)(12)(12) *+1/2 LSB(15)(12)(12)(12)(12)(12)(12)(12)(12)(12)(12)(12)(12)(1

.ASCIZ \-1/2LSB\

G05

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27(665)
ASCII MESSAGES

2-DEC-76 16:29 PAGE 58

2342
 2343 013022 044504 043106 051105
 2344 013030 047105 044524 046101
 2345 013036 046040 047111 040505
 2346 013044 044522 054524 006472
 2347 013052 000012
 2348 013054 020040 020040 020040
 2349 013062 020040 020040 020040
 2350 013070 020040 020040 020040
 2351 013076 020040 052123 052101
 2352 013104 026505 044527 052104
 2353 013112 020110 044504 052123
 2354 013120 044522 052502 044524
 2355 013126 047117 005015 005012
 2356 013134 020040 020043 043117
 2357 013142 051440 040524 042524
 2358 013150 005123 005012 005012
 2359 013156 005012 005012 005012
 2360 013164 005012 005012 005012
 2361 013172 005012
 2362 013174 020040 020040 020040
 2363 013202 020040 020040 020040
 2364 013210 020040 020040 020040
 2365 013216 020040 020040 020040
 2366 013224 020040 020040 020040
 2367 013232 020040 020040 020040
 2368 013240 020040 020040 020040
 2369 013246 020040 020040 020040
 2370 013254 051440 040524 042524
 2371 013262 053440 042111 044124
 2372 013270 024040 051514 024502
 2373 013276 005015
 2374 013300 030040 020040 020040
 2375 013306 020040 020040 020040
 2376 013314 020040 020040 027461
 2377 013322 020062 020040 020040
 2378 013330 020040 020040 020040
 2379 013336 020040 020061 020040
 2380 013344 020040 020040 020040
 2381 013352 020040 030440 030440
 2382 013360 031057 020040 020040
 2383 013366 020040 020040 020040
 2384 013374 020040 031040 000
 2385 013401 015 052012 050131
 2386 013406 020105 042514 052124
 2387 013414 051105 023040 041440
 2388 013422 020122 047506 020122
 2389 013430 042504 044523 042522
 2390 013436 020104 042524 052123
 2391 013444 020072 000
 2392 013447 122 046105 052101
 2393 013454 053111 020105 041501
 2394 013462 052503 040522 054503
 2395 013470 006472 000012

MSG20: .EVEN
.ASCIIZ /DIFFERENTIAL LINEARITY:/(15)<12>

MSG16: .ASCII / STATE-WIDTH DISTRIBUTION/(15)<12><12><12>

.ASCII # OF STATES/(12)<12><12><12><12><12><12><12><12><12><12><12><12><12><

.ASCII / STATE WIDTH (LSB)/(15)

.ASCIIZ # 0 1/2 1 1 1/2 2

MSG71: .ASCIIZ (15)<12>/TYPE LETTER & CR FOR DESIRED TEST: /

MSG21: .ASCIIZ /RELATIVE ACCURACY:/(15)<12>

H05

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27.665)
ASCII MESSAGES

2-DEC-76 16:29 PAGE 59

2396	013474	046040	041123	046440	LINEA: .ASCIIZ / LSB MAXIMUM AT /
2397	013502	054101	046511	046525	
2398	013510	040440	020124	000	
2399	013515	015	041412	046101	HEADS: .ASCII <15><12>/CALIBRATION--/
2400	013522	041111	040522	044524	
2401	013530	047117	026455		
2402	013534	051440	052105	041440	ASKCH: .ASCIIZ / SET CHANNEL IN SWR LOW BYTE/<15><12>
2403	013542	040510	047116	046105	
2404	013550	044440	020116	053523	
2405	013556	020122	047514	020127	
2406	013564	054502	042524	005015	
2407	013572	000			
2408	013573	033	000132		CO: .ASCIIZ <33><132>
2409	013576	000055			C1: .ASCIIZ <55>
2410	013600	03!033	000		C2: .ASCIIZ <33><62>
2411	013603	112	000		C3: .ASCIIZ <112>
2412	013605	015	047412	043106	MOFSET: .ASCIIZ <15><12>/OFFSET =/
2413	013612	042523	020124	000075	
2414	013620	046040	041123	000040	MLSB: .ASCIIZ / LSB /
2415	013626	040440	020124	000	MAT: .ASCIIZ / AT /
2416	013633	015	020012	047105	METST: .ASCIIZ <15><12>/ ENTERING TEST /
2417	013640	042524	044522	043516	
2418	013646	052040	051505	020124	
2419	013654	000			
2420	013655	033	061	101	BUFF1: .BYTE 33,61,101,61,111,62,114,41,60,45,63,51,66,55,71,61,74,110,41,40,112,0
2421	013660	061	111	062	
2422	013663	114	041	060	
2423	013666	045	063	051	
2424	013671	066	055	071	
2425	013674	061	074	110	
2426	013677	041	040	112	
2427	013702	000			
2428	013703	033	061	101	BUFF2: .BYTE 33,61,101,47,111,61,104,50,65,44,62,110,40,40,102,0
2429	013706	047	111	061	
2430	013711	104	050	065	
2431	013714	044	062	110	
2432	013717	040	040	102	
2433	013722	000			
2434	013723	033	110	033	VTINIT: .BYTE 33,110,33,112,33,61,101,40,33,62,0
2435	013726	112	033	061	
2436	013731	101	040	033	
2437	013734	062	000		
2438	013736	005015	046412	026504	HEAD1: .ASCII <15><12><12>/MD-11-DZADL-B AD11K DIAGNOSTIC/<15><12>
2439	013744	030461	042055	040532	
2440	013752	046104	041055	020040	
2441	013760	020040	042101	030461	
2442	013766	020113	044504	043501	
2443	013774	047516	052123	041511	
2444	014002	005015			
2445	014004	040412	020072	052501	.ASCII <12>/A: AUTO TEST/
2446	014012	047524	052040	051505	
2447	014020	124			
2448	014021	015	041412	020072	.ASCII <15><12>/C: CALIBRATION/
2449	014026	040503	044514	051102	

MAINDEC-11-DZADL-B
DZADLB.P11

MACY11 27(665) 2-DEC-76 16:29 PAGE 60
ASCII MESSAGES

2450	014034	052101	047511	116
2451	014041	015	046012	020072
2452	014046	047514	044507	020103
2453	014054	042524	052123	
2454	014060	005015	035116	047040
2455	014066	044517	042523	052040
2456	014074	051505	124	

.ASCII <15><12>/L: LOGIC TEST/

.ASCII <15><12>/N: NOISE TEST/

J05

MAINDEC-11-DZADL-B
DZADLB.P11MACY11 27.665)
ASCII MESSAGES

2-DEC-76 16:29 PAGE 61

```

2457 014077 015 051412 020072
2458 014104 042523 052124 042514
2459 014112 052040 051505 124
2460 014117 015 053412 020072
2461 014124 051127 050101 051101
2462 014132 052517 042116 052040
2463 014140 051505 006524 000012
2464 014146 005015 052123 052101
2465 014154 051525 051040 043505
2466 014162 020056 051105 047522
2467 014170 006522 000012
2468 014174 005015 040506 046111
2469 014202 042105 052040 020117
2470 014210 047111 042524 051122
2471 014216 050125 006524 000012
2472 014224 005015 047125 054105
2473 014232 042520 052103 042105
2474 014240 044440 052116 051105
2475 014246 052522 052120 005015
2476 014254 000
2477 014255 015 042412 051122
2478 014262 051117 047440 020116
2479 014270 027501 020104 044103
2480 014276 047101 042516 006514
2481 014304 000012
2482 014306 051105 050122 020103
2483 014314 052123 042522 020107
2484 014322 054105 042520 052103
2485 014330 042105 040440 052103
2486 014336 040525 006514 000012
2487 014344 051105 050122 020103
2488 014352 051440 051124 043505
2489 014360 020040 041440 040510
2490 014366 047116 046105 020040
2491 014374 047516 044515 040516
2492 014402 020114 052040 046117
2493 014410 051105 047101 042503
2494 014416 020040 041501 052524
2495 014424 046101 000
2496 014427 105 051122 041520
2497 014434 020040 020040 020040
2498 014442 052123 042522 020107
2499 014450 020040 040440 052103
2500 014456 040525 006514 000012

```

.ASCII <15><12>/S: SETTLE TEST/

.ASCIZ <15><12>/W: WRAPAROUND TEST/<15><12>

EM1: .ASCIZ <15><12>/STATUS REG. ERROR/<15><12>

EM2: .ASCIZ <15><12>/FAILED TO INTERRUPT/<15><12>

EM3: .ASCIZ <15><12>/UNEXPECTED INTERRUPT/<15><12>

EM4: .ASCIZ <15><12>#ERROR ON A/D CHANNEL#<15><12>

DH1: .ASCIZ /ERRPC STREG EXPECTED ACTUAL/<15><12>

DH2: .ASCIZ /ERRPC STREG CHANNEL NOMINAL TOLERANCE ACTUAL/

DH3: .ASCIZ /ERRPC STREG ACTUAL/<15><12>

K05

MAINDEC-11-DZADL-B
DZADLB.P11

MACY11 27(665)
ASCII MESSAGES

2-DEC-76 16:29 PAGE 62

2501	014464	000				HUNS:	.BYTE	0
2502	014465	056				DECPNT:	.BYTE	56
2503	014466	000				TENS:	.BYTE	0
2504	014467	000	000			ONES:	.BYTE	0,0
2505		014472				.EVEN		
2506								
2507	014472	001116	001316	001124	DT1:	SERRPC, STREG, SGDDAT, SBDDAT, 0		
2508	014500	001126	000000					
2509	014504	001116	001316	001366	DT2:	SERRPC, STREG, CHANL, SGDDAT, SPREAD, SBDDAT, 0		
2510	014512	001124	001406	001126				
2511	014520	000000						
2512	014522	001116	001316	001126	DT3:	SERRPC, STREG, SBDDAT, 0		
2513	014530	000000						
2514								
2515	014532	000000			DF1:	0		
2516								
2517								
2518								
2519								

L05

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 63
DZADLB.P11 *TV INPUT ROUTINE

```

2520 .SBTTL TTY INPUT ROUTINE
2521
2522 ::*****
2523 .ENABL LSB
2524
2525 .DSABL LSB
2526
2527
2528 ::*****
2529 ::THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
2530 ::CALL:
2531 ::      RDCHR          :: INPUT A SINGLE CHARACTER FROM THE TTY
2532 ::      RETURN HERE    :: CHARACTER IS ON THE STACK
2533 ::                     :: WITH PARITY BIT STRIPPED OFF
2534 ::
2535
2536 014534 011646          SRDCHR: MOV      (SP),-(SP)      :: PUSH DOWN THE PC
2537 014536 016666 000004 000002 15:      MOV      4(SP),2(SP)      :: SAVE THE PS
2538 014544 105777 164374          15:      TSTB      2$TKS          :: WAIT FOR
2539 014550 100375          BPL      15          :: A CHARACTER
2540 014552 117766 164370 000004          MOVB      2$TKB,4(SP)      :: READ THE TTY
2541 014560 042766 177600 000004          BIC      #177,4(SP)      :: GET RID OF JUNK IF ANY
2542 014566 026627 000004 000023          CMP      4(SP),#23      :: IS IT A CONTROL-S?
2543 014574 001013          BNE      3$          :: BRANCH IF NO
2544 014576 105777 164342          25:      TSTB      2$TKS          :: WAIT FOR A CHARACTER
2545 014602 100375          BPL      25          :: LOOP UNTIL ITS THERE
2546 014604 117746 164336          MOVB      2$TKB,-(SP)          :: GET CHARACTER
2547 014610 042716 177600          BIC      #177,4(SP)          :: MAKE IT 7-BIT ASCII
2548 014614 022627 000021          CMP      (SP)+,#21          :: IS IT A CONTROL-Q?
2549 014620 001366          BNE      25          :: IF NOT DISCARD IT
2550 014622 000750          BR      15          :: YES, RESUME
2551 014624 026627 000004 000140 35:      CMP      4(SP),#140      :: IS IT UPPER CASE?
2552 014632 002407          BLT      45          :: BRANCH IF YES
2553 014634 026627 000004 000175          CMP      4(SP),#175      :: IS IT A SPECIAL CHAR?
2554 014642 003003          BGT      45          :: BRANCH IF YES
2555 014644 042766 000040 000004          BIC      #40,4(SP)      :: MAKE IT UPPER CASE
2556 014652 000002          45:      RTI          :: GO BACK TO USER
2557
2558 ::*****
2559 ::THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2560 ::CALL:
2561 ::      RDLIN          :: INPUT A STRING FROM THE TTY
2562 ::      RETURN HERE    :: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2563 ::                     :: TERMINATOR WILL BE A BYTE OF ALL 0'S
2564
2565 014654 010346          SRDLIN: MOV      R3, -(SP)          :: SAVE R3
2566 014656 012703 014762 15:      MOV      $TTYIN,R3          :: GET ADDRESS
2567 014662 022703 014772 25:      CMP      $TTYIN+8.,R3      :: BUFFER FULL?
2568 014666 101405          BLOS      45          :: BR IF YES
2569 014670 104404          RDCHR          :: GO READ ONE CHARACTER FROM THE TTY
2570 014672 112613          MOVB      (SP)+,(R3)          :: GET CHARACTER
2571 014674 122713 000177 105:      CPB      #177,(R3)          :: IS IT A RUBOUT
2572 014700 001003          BNE      35          :: SKIP IF NOT
2573 014702 104400 001170 45:      TYPE      $QUES          :: TYPE A .
2574 014706 000763          BR      15          :: CLEAR THE BUFFER AND LOOP

```


M05

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 64
 DZADLB.P11 TTY INPUT ROUTINE

2574	014710	111337	014760	35:	MOVB	(R3),95	::ECHO THE CHARACTER
2575	014714	104400	014760		TYPE	95	
2576	014720	122723	000015		CMPB	#15,(R3)+	::CHECK FOR RETURN
2577	014724	001356			BNE	25	::LOOP IF NOT RETURN
2578	014726	105063	177777		CLRB	-1(R3)	::CLEAR RETURN (THE 15)
2579	014732	104400	001172		TYPE	\$LF	::TYPE A LINE FEED
2580	014736	012603			MOV	(SP)+,R3	::RESTORE R3
2581	014740	011646			MOV	(SP)-,(SP)	::ADJUST THE STACK AND PUT ADDRESS OF THE
2582	014742	016666	000004 000002		MOV	4(SP),2(SP)	::FIRST ASCII CHARACTER ON IT
2583	014750	012766	014762 000004		MOV	#TTYIN,4(SP)	
2584	014756	000002			RTI		::RETURN
2585	014760	000		95:	.BYTE	0	::STORAGE FOR ASCII CHAR. TO TYPE
2586	014761	000			.BYTE	0	::TERMINATOR
2587	014762	000010		\$TTYIN:	.BLKB	8.	::RESERVE 8 BYTES FOR TTY INPUT
2588	014772	052536	005015 000	\$CNTLU:	.ASCIZ	/U<15><12>	::CONTROL "U"
2589	014777	136	006507 000012	\$CNTLG:	.ASCIZ	/G<15><12>	::CONTROL "G"
2590	015004	005015	053523 020122	\$MSWR:	.ASCIZ	<15><12>/SWR = /	
2591	015012	020075	000				
2592	015015	040	047040 053505	\$MNEW:	.ASCIZ	/ NEW = /	
2593	015022	036440	000040				

N05

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 65
DZADLB.P11 READ AN OCTAL NUMBER FROM THE TTY

2594				.SBTTL	READ AN OCTAL NUMBER FROM THE TTY	
2595						
2596						
2597						
2598						
2599						
2600						
2601						
2602						
2603						
2604	015026	011646				
2605	015030	016666	000004 000002			
2606	015036	010046				
2607	015040	010146				
2608	015042	010246				
2609	015044	104405				
2610	015046	012600				
2611	015050	005001				
2612	015052	005002				
2613	015054	112046				
2614	015056	001412				
2615	015060	006301				
2616	015062	006102				
2617	015064	006301				
2618	015066	006102				
2619	015070	006301				
2620	015072	006102				
2621	015074	042716	177770			
2622	015100	052601				
2623	015102	030764				
2624	015104	005726				
2625	015106	010166	000012			
2626	015112	010237	015126			
2627	015116	012602				
2628	015120	012601				
2629	015122	012600				
2630	015124	000002				
2631	015126	000000				

.SBTTL SCOPE HANDLER ROUTINE

```

*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG.(DISPLAY:7:0)
*AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY:15:08.
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1      LOOP ON TEST
*SW11=1      INHIBIT ITERATIONS
*SW09=1      LOOP ON ERROR
*SW08=1      LOOP ON TEST IN SWR:7:0)
*CALL
*          SCOPE          ::SCOPE=107

```

2632	015130	032777	040000	164002	SSCOPE:	BIT	BIT14,DSWR	::LOOP ON PRESENT TEST?
2633	015130	001114			15:	BNE	SOVER	::YES IF SW14=1
2634					*****	START OF	CODE FOR THE XOR	TESTER*****
2635	015140	000416			XTSTR:	BR	65	::IF RUNNING ON THE "XOR" TESTER CHANGE
2636								::THIS INSTRUCTION TO A "NOP" (NOP=240)
2637	015142	013746	000004			MOV	2ERRVEC, -(SP)	::SAVE THE CONTENTS OF THE ERROR VECTOR
2638	015146	012737	015166	000004		MOV	SS, 2ERRVEC	::SET FOR TIMEOUT
2639	015154	005737	177060			TST	2177060	::TIME OUT ON XOR?
2640	015160	012637	000004			MOV	(SP)+, 2ERRVEC	::RESTORE THE ERROR VECTOR
2641	015164	000463				BR	SSVLAD	::GO TO THE NEXT TEST
2642	015166	022626			55:	CMP	(SP)+, (SP)+	::CLEAR THE STACK AFTER A TIME OUT
2643	015170	012637	000004			MOV	(SP)+, 2ERRVEC	::RESTORE THE ERROR VECTOR
2644	015174	000423				BR	75	::LOOP ON THE PRESENT TEST
2645	015176				65: *****	END OF	CODE FOR THE XOR	TESTER*****
2646	015176	032777	000400	163734		BIT	BIT08,DSWR	::LOOP ON SPEC. TEST?
2647	015204	001404				BEQ	25	::BR IF NO
2648	015206	127737	163726	001102		CMPB	DSWR, STSTNM	::ON THE RIGHT TEST? SWR:7:0
2649	015214	001465				BEQ	SOVER	::BR IF YES
2650	015216	105737	001103		25:	STB	SERFLG	::HAS AN ERROR OCCURRED?
2651	015222	001421				BEQ	35	::BR IF NO
2652	015224	123737	001115	001103		CMPB	SERMAX, SERFLG	::MAX. ERRORS FOR THIS TEST OCCURRED?
2653	015232	101015				BHI	35	::BR IF NO
2654	015234	032777	001000	163676		BIT	08 *09, DSWR	::LOOP ON ERROR?
2655	015242	001404				BEQ	45	::BR IF NO
2656	015244	013737	001110	001106	75:	MOV	SLPERF, SLPCR	::SET LOOP ADDRESS TO LAST SCOPE
2657	015252	000446				BR	SOVER	
2658	015254	105037	001103		45:	CLRB	SERFLG	::ZERO THE ERROR FLAG
2659	015260	005037	001160			CLR	STIMES	::CLEAR THE NUMBER OF ITERATIONS TO MAKE
2660	015264	000415				BR	15	::ESCAPE TO THE NEXT TEST
2661	015266	032777	004000	163644	35:	BIT	BIT11, DSWR	::INHIBIT ITERATIONS?
2662	015274	001011				BNE	15	::BR IF YES
2663	015276	005737	001202			TST	SPASS	::IF FIRST PASS OF PROGRAM
2664	015302	001406				BEQ	15	::INHIBIT ITERATIONS
2665	015304	005237	001104			INC	SICNT	::INCREMENT ITERATION COUNT
2666	015310	023737	001160	001104		CMP	STIMES, SICNT	::CHECK THE NUMBER OF ITERATIONS MADE
2667	015316	002024				BGE	SOVER	::BR IF MORE ITERATION REQUIRED
2668	015320	012737	000001	001104	15:	MOV	01, SICNT	::REINITIALIZE THE ITERATION COUNTER
2669	015326	013737	015404	001160		MOV	SMXNT, STIMES	::SET NUMBER OF ITERATIONS TO DO
2670	015334	105237	001102		SSVLAD:	INCB	STSTNM	::COUNT TEST NUMBERS

C06

MACRO-11-D2ADL-B MACY11 27.665) 2-DEC-76 16:29 PAGE 67
 CAROLE.P11 SCOPE HANDLER ROUTINE

```

0015340 113737 001102 001200      MOV      $TSTNM,$TESTN      :: SET TEST NUMBER IN APT MAILBOX
0015346 011637 001106      MOV      (SP),SLPADR      :: SAVE SCOPE LOOP ADDRESS
0015352 011637 001110      MOV      (SP),SLPERR      :: SAVE ERROR LOOP ADDRESS
0015356 005037 001162      CLR      $ESCAPE      :: CLEAR THE ESCAPE FROM ERROR ADDRESS
0015362 112737 000001 001115      MOV      #1,$ERMAX      :: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
0015368 013777 001102 163544 $OVER: MOV      $TSTNM,$DISPLAY  :: DISPLAY TEST NUMBER
0015374 013716 001106      MOV      SLPADR,(SP)      :: FUDGE RETURN ADDRESS
0015380 000003      RTI      :: FIXES PS
0015386 003720      SMXCNT: 2000      :: MAX. NUMBER OF ITERATIONS
                                :: *****
                                :: THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
                                :: SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                :: AND GO TO $ERRTYP ON ERROR
                                :: THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                :: $SW15=1      HALT ON ERROR
                                :: $SW13=1      INHIBIT ERROR TYPEOUTS
                                :: $SW10=1      BELL ON ERROR
                                :: $SW09=1      LOOP ON ERROR
                                :: $CALL      ERROR      N      :: ERROR=EMT AND N=ERROR ITEM NUMBER
                                :: *****
0015406 105237 001103      $ERROR: INC      $ERFLG      :: SET THE ERROR FLAG
0015412 001775      75:      BEQ      75      :: DON'T LET THE FLAG GO TO ZERO
0015418 013777 001102 163520      MOV      $TSTNM,$DISPLAY  :: DISPLAY TEST NUMBER AND ERROR FLAG
0015424 032777 002000 163510      BIT      #BIT10,$SWR      :: BELL ON ERROR?
0015430 001402      15:      BEQ      15      :: NO - SKIP
0015436 104400      TYPE      $BELL      :: RING BELL
0015442 005237      INC      $ERTTL      :: COUNT THE NUMBER OF ERRORS
0015448 011637 001112      MOV      (SP),$ERRPC      :: GET ADDRESS OF ERROR INSTRUCTION
0015454 162737 000002 001116      SUB      #2,$ERRPC
0015460 177737 163436 001114      MOV      $ERRPC,$ITEMB      :: STRIP AND SAVE THE ERROR ITEM CODE
0015466 032777 020000 163450      BIT      #BIT13,$SWR      :: SKIP TYPEOUT IF SET
0015472 001004      BNE      205      :: SKIP TYPEOUTS
0015478 004737 015602      JSR      PC,$ERRTYP      :: GO TO USER ERROR ROUTINE
0015484 104400      TYPE      $SCLF
0015502 122737 000001 001214      205:      CMPB      #APTENV,$ENV      :: RUNNING IN APT MODE
0015508 001007      BNE      25      :: NO SKIP APT ERROR REPORT
0015514 113737 001114 015524      MOV      $ITEMB,215      :: SET ITEM NUMBER AS ERROR NUMBER
0015520 004737 016236      JSR      PC,$ATTN      :: REPORT FATAL ERROR TO APT
0015526 000      215:      .BYTE      0
0015532 000      .BYTE      0
0015538 000777      225:      BR      225      :: APT ERROR LOOP
0015544 005777 163404      25:      TST      25      :: HALT ON ERROR
0015550 100001      BPL      35      :: SKIP IF CONTINUE
0015556 032777 001000 163372      35:      HALT      :: HALT ON ERROR!
0015562 001402      BIT      #BIT09,$SWR      :: LOOP ON ERROR SWITCH SET?
0015568 013716 001110      BEQ      45      :: BR IF NO
0015574 005737 001162      MOV      SLPERR,(SP)      :: FUDGE RETURN FOR LOOPING
0015580 001402      TST      $ESCAPE      :: CHECK FOR AN ESCAPE ADDRESS
0015586      EMT      55      :: BR IF NONE

```

D06

MACRODEC-11-DIADL-8 MACY:1 27(665) 2-DEC-76 16:29 PAGE 68
 CAROLB.P:11 ERROR HANDLER ROUTINE

2740 015562 013716 001162
 2741 015566
 2742 015566 022737 011776 000042
 2743 015574 001001
 2744 015576 000000
 2745 015600
 2746 015600 000002

MOV \$ESCAPE,(SP) ::FUDGE RETURN ADDRESS FOR ESCAPE
 5\$: CMP \$SENDAD,2842 ::ACT-11 AUTO-ACCEPT?
 BNE 6\$::BRANCH IF NO
 HALT ::YES
 6\$: RTI ::RETURN
 .SBTTL ERROR MESSAGE TYPEOUT ROUTINE

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

\$ERRTYP:

015602
 015602 104400 001171
 015606 010046
 015610 005300
 015612 153700 001114
 015616 001004
 015620 013746 001116
 015624 104401
 015626 000426
 015630 005300
 015632 006300
 015634 006300
 015636 006300
 015640 062700 001256
 015644 012037 015654
 015650 001404
 015652 104400
 015654 000000
 015656 104400 001171
 015660 012037 015672
 015666 001404
 015670 104400
 015672 000000
 015674 104400 001171
 015700 011000
 015702 001004
 015704 012600
 015706 104400 001171
 015712 000207
 015714
 015714 013046
 015716 104401
 015720 005710
 015722 001770
 015724 104400 015732
 015730 000771
 015732 020040 000
 015736

TYPE \$SCRLF ::"CARRIAGE RETURN" & "LINE FEED"
 MOV \$RO,-(SP) ::SAVE \$RO
 CLR \$RO ::PICKUP THE ITEM INDEX
 BISB 28\$ITEMB,\$RO
 BNE 1\$::IF ITEM NUMBER IS ZERO, JUST
 TYPE THE PC OF THE ERROR
 MOV \$ERRPC,-(SP) ::SAVE \$ERRPC FOR TYPEOUT
 ERROR ADDRESS
 GO TYPE--OCTAL ASCII(ALL DIGITS)
 GET OUT
 1\$: ADJUST THE INDEX SO THAT IT WILL
 WORK FOR THE ERROR TABLE
 TYPEOC
 BR 6\$
 2\$: DEC \$RO
 ASL \$RO
 ASL \$RO
 ASL \$RO
 ADD \$ERRTB,\$RO ::FORM TABLE POINTER
 MOV \$RO,2\$::PICKUP "ERROR MESSAGE" POINTER
 BEQ 3\$::SKIP TYPEOUT IF NO POINTER
 TYPE "ERROR MESSAGE"
 2\$: TYPE "ERROR MESSAGE" POINTER GOES HERE
 TYPE \$SCRLF ::"CARRIAGE RETURN" & "LINE FEED"
 3\$: MOV (\$RO)+,4\$::PICKUP "DATA HEADER" POINTER
 BEQ 5\$::SKIP TYPEOUT IF 0
 TYPE "DATA HEADER"
 4\$: TYPE "DATA HEADER" POINTER GOES HERE
 TYPE \$SCRLF ::"CARRIAGE RETURN" & "LINE FEED"
 5\$: MOV (\$RO), \$RO ::PICKUP "DATA TABLE" POINTER
 BNE 7\$::GO TYPE THE DATA
 6\$: MOV (SP)+, \$RO ::RESTORE \$RO
 TYPE \$SCRLF ::"CARRIAGE RETURN" & "LINE FEED"
 RTS PC ::RETURN
 7\$: MOV 2(\$RO)+, -(SP) ::SAVE 2(\$RO)+ FOR TYPEOUT
 TYPEOC
 TST (\$RO)
 BEQ 6\$::IS THERE ANOTHER NUMBER?
 TYPE 2\$::TYPE TWO(2) SPACES
 BR 7\$
 8\$: .ASCIZ /
 .EVEN

```

*****
: *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.

```

```

: *CALL:
: *1) USING A TRAP INSTRUCTION
: *      TYPE      .MESADR      :: MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
: *OR
: *      TYPE
: *      MESADR
: *

```

```

2794 .SBTTL TYPE ROUTINE
2795
2796 *****
2797 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2798 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2799 *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
2800 *NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
2801 *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2802
2803 *
2804 *CALL:
2805 *1) USING A TRAP INSTRUCTION
2806 *
2807 *CR TYPE .MESADR ;; MESADR IS FIRST ADDRESS OF AN ASCIZ ST
2808 *
2809 * TYPE
2810 * MESADR
2811 *
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847

```


F06

 MAINDEC-11-DZADL-B MACY11 27.665) 2-DEC-76 16:29 PAGE 70
 DZADL8.P11 TYPE ROUTINE

```

2848 016112 004737 016150      JSR    PC,$TYPEC      ;;GO TYPE A NULL
2849 016116 105337 016214      DECB   $CHARCNT      ;;DO NOT COUNT AS A COUNT
2850 016122 000770              BR      75          ;;LOOP

2851
2852
2853      :HORIZONTAL TAB PROCESSOR
2854
2855 016124 112716 000040      85:    MOVB   #' (SP)      ;; REPLACE TAB WITH SPACE
2856 016130 004737 016150      95:    JSR    PC,$TYPEC      ;; TYPE A SPACE
2857 016134 132737 000007 016214  BITB   #',$CHARCNT      ;; BRANCH IF NOT AT
2858 016142 001372              BNE     95          ;; TAB STOP
2859 016144 005726              TST     (SP)+      ;; POP SPACE OFF STACK
2860 016146 000724              BR      25          ;; GET NEXT CHARACTER
2861 016150 105777 162774      $TYPEC: TSTB   $STPS      ;; WAIT UNTIL PRINTER IS READY
2862 016154 100375              BPL     $TYPEC
2863 016156 116677 000002 162766  MOVB   2(SP), $STPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.
2864 016164 122766 000015 000002  CMPB   #'CR, 2(SP)      ;; IS CHARACTER A CARRIAGE RETURN?
2865 016172 001003              BNE     15          ;; BRANCH IF NO
2866 016174 105037 016214      CLRB   $CHARCNT      ;; YES--CLEAR CHARACTER COUNT
2867 016200 000406              BR      $TYPEX      ;; EXIT
2868 016202 122766 000012 000002 15:    CMPB   #'LF, 2(SP)      ;; IS CHARACTER A LINE FEED?
2869 016210 001402              BEQ     $TYPEX      ;; BRANCH IF YES
2870 016212 105227              INCB   (PC)+      ;; COUNT THE CHARACTER
2871 016214 000000              $CHARCNT: .WORD 0      ;; CHARACTER COUNT STORAGE
2872 016216 000207              $TYPEX: RTS    PC

2873
2874      .SBTTL  APT COMMUNICATIONS ROUTINE
2875
2876 016220 112737 000001 016464  $ATY1:  MOVB   #'1, $FFLG      ;; TO REPORT FATAL ERROR
2877 016226 112737 000001 016462  $ATY3:  MOVB   #'1, $MFLG      ;; TO TYPE A MESSAGE
2878 016234 000403              BR      $ATYC
2879 016236 112737 000001 016464  $ATY4:  MOVB   #'1, $FFLG      ;; TO ONLY REPORT FATAL ERROR
2880 016244              $ATYC:
2881 016244 010046              MOV     R0, -(SP)      ;; PUSH R0 ON STACK
2882 016246 010146              MOV     R1, -(SP)      ;; PUSH R1 ON STACK
2883 016250 105737 016462      TSTB   $MFLG      ;; SHOULD TYPE A MESSAGE?
2884 016254 001450              BEQ     55          ;; IF NOT: BR
2885 016256 122737 000001 001214  CMPB   #'APTENV, $ENV      ;; OPERATING UNDER APT?
2886 016264 001031              BNE     35          ;; IF NOT: BR
2887 016266 132737 000100 001215  BITB   #'APTSPOOL, $ENV      ;; SHOULD SPOOL MESSAGES?
2888 016274 001425              BEQ     35          ;; IF NOT: BR
2889 016276 017600 000004              MOV     $4(SP), R0      ;; GET MESSAGE ADDR.
2890 016302 062766 000002 000004  ADD     #'2, 4(SP)      ;; BUMP RETURN ADDR.
2891 016310 005737 001174      15:    TST     $MSGTYPE      ;; SEE IF DONE W/ LAST XMISSION?
2892 016314 001375              BNE     15          ;; IF NOT: WAIT
2893 016316 010037 001210      MOV     R0, $MSGAD      ;; PUT ADDR IN MAILBOX
2894 016322 105720      25:    TSTB   (R0)+      ;; FIND END OF MESSAGE
2895 016324 001376              BNE     25
2896 016326 163700 001210      SUB     $MSGAD, R0      ;; SUB START OF MESSAGE
2897 016332 006200              ASR     R0          ;; GET MESSAGE LGTH IN WORDS
2898 016334 010037 001212      MOV     R0, $MSGGLT      ;; PUT LENGTH IN MAILBOX
2899 016340 012737 000004 001174  MOV     #'4, $MSGTYPE      ;; TELL APT TO TAKE MSG.
2900 016346 000413              BR      55
2901 016350 017637 000004 016374 35:    MOV     $4(SP), 45      ;; PUT MSG ADDR IN JSR LINKAGE

```

G06

MAINDEC-11-DZADL-B MACY!! 27(665) 2-DEC-76 16:29 PAGE 71
 DZADLB.P11 APT COMMUNICATIONS ROUTINE

2902	016356	062766	000002	000004	ADD	#2,4(SP)	::BUMP RETURN ADDRESS
2903	016364	013746	177776		MOV	177776,-(SP)	::PUSH 177776 ON STACK
2904	016370	004737	015736		JSR	PC,\$TYPE	::CALL TYPE MACRO
2905	016374	000000			45:	.WORD	0
2906	016376				55:		
2907	016376	105737	016464		105:	TSTB	\$FFLG
2908	016402	001416				BEQ	125
2909	016404	005737	001214			SENV	::SHOULD REPORT FATAL ERROR?
2910	016410	001413				TST	::IF NOT: BR
2911	016412	005737	001174			BEQ	::RUNNING UNDER APT?
2912	016416	001375			113:	TST	::IF NOT: BR
2913	016420	017637	000004	001176		\$MSGTYPE	::FINISHED LAST MESSAGE?
2914	016426	062766	000002	000004		115	::IF NOT: WAIT
2915	016434	005237	001174			MOV	::GET ERROR #
2916	016440	105037	016464			24(SP), \$FATAL	::BUMP RETURN ADDR.
2917	016444	105037	016463		125:	INC	::TELL APT TO TAKE ERROR
2918	016450	105037	016462			CLRB	::CLEAR FATAL FLAG
2919	016454	012601				\$FFLG	::CLEAR LOG FLAG
2920	016456	012600				\$LFLG	::CLEAR MESSAGE FLAG
2921	016460	000207				\$MFLG	::POP STACK INTO R1
2922	016462	000				MOV	::POP STACK INTO R0
2923	016463	000				(SP)+,R1	::RETURN
2924	016464	000				MOV	::MESSG. FLAG
2925	016466					(SP)+,R0	::LOG FLAG
2926	000200					RTS	::FATAL FLAG
2927	000001					PC	
2928	000100					0	
2929	000040					0	

\$MFLG: .BYTE
 \$LFLG: .BYTE
 \$FFLG: .BYTE
 .EVEN
 APTSIZE=200
 APTENV=00!
 APTSPCCL=100
 APTCSUP=040

H06

MAINDEC-11-02ADL-B MACY: 27.665) 2-DEC-76 16:29 PAGE 72
 02ADLB.P11 BINARY TO OCTAL (ASCII) AND TYPE

2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983

016466	017646	000000	
016472	116637	000001	016711
016500	112637	016713	
016504	062716	000002	
016510	000406		
016512	112737	000001	016711
016520	112737	000006	016713
016526	112737	000005	016710
016534	010346		
016536	010446		
016540	010546		
016542	113704	016713	
016546	005404		
016550	062704	000006	
016554	110437	016712	
016560	113704	016711	
016564	016605	000012	
016570	005003		
016572	006105		
016574	000404		
016576	006105		
016600	006105		
016602	006105		
016604	010503		
016606	006103		
016610	105337	016712	
016614	100016		
016616	042703	177770	
016622	001002		

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

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

STYPOS: MOV      2(SP), -(SP)      ;; PICKUP THE MODE
        MOVB     1(SP), $OFILL    ;; LOAD ZERO FILL SWITCH
        MOVB     (SP)+, $OMODE+1  ;; NUMBER OF DIGITS TO TYPE
        ADD      #2, (SP)         ;; ADJUST RETURN ADDRESS
        BR       STYON

STYPOC: MOVB     #1, $OFILL        ;; SET THE ZERO FILL SWITCH
        MOVB     #6, $OMODE+1     ;; SET FOR SIX(6) DIGITS
STYPON: MOVB     #5, $OCNT        ;; SET THE ITERATION COUNT
        MOV      R3, -(SP)        ;; SAVE R3
        MOV      R4, -(SP)        ;; SAVE R4
        MOV      R5, -(SP)        ;; SAVE R5
        MOVB     $OMODE+1, R4     ;; GET THE NUMBER OF DIGITS TO TYPE
        NEG      R4
        ADD      #6, R4           ;; SUBTRACT IT FOR MAX. ALLOWED
        MOVB     R4, $OMODE       ;; SAVE IT FOR USE
        MOVB     $OFILL, R4      ;; GET THE ZERO FILL SWITCH
        MOV      12(SP), R5      ;; PICKUP THE INPUT NUMBER
        CLR      R3              ;; CLEAR THE OUTPUT WORD
        ROL      R5              ;; ROTATE MSB INTO "C"
        BR       3$              ;; GO DO MSB
        ROL      R5              ;; FORM THIS DIGIT
        ROL      R5
        ROL      R5
        MOV      R5, R3
        ROL      R3
        DECB     $OMODE
        BPL      7$
        BIC      #177770, R3     ;; GET RID OF JUNK
        BNE     4$              ;; TEST FOR 0

1$: ROL      R5
2$: ROL      R5
3$: ROL      R5
4$:
5$:
6$:
7$:
8$:
9$:

```

MACRODEC-11-02ADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 73
 02ADLB.F11 BINARY TO OCTAL (ASCII) AND TYPE

2994	016624	005704		TST	R4	:: SUPPRESS THIS 0'
2995	016626	001403		BEQ	55	:: BR IF YES
2996	016630	005204		INC	R4	:: DON'T SUPPRESS ANYMORE 0'S
2997	016632	052703	000060	BIS	*'0,R3	:: MAKE THIS DIGIT ASCII
2998	016636	052703	000040	BIS	*',R3	:: MAKE ASCII IF NOT ALREADY
2999	016642	110337	016706	MOVB	R3,R5	:: SAVE FOR TYPING
2990	016646	104400	016706	TYPE	R5	:: GO TYPE THIS DIGIT
2991	016652	105337	016710	DECB	\$OCNT	:: COUNT BY 1
2992	016656	003347		BGT	25	:: BR IF MORE TO DO
2993	016660	002402		BLT	65	:: BR IF DONE
2994	016662	005204		INC	R4	:: INSURE LAST DIGIT ISN'T A BLANK
2995	016664	000744		BR	25	:: GO DO THE LAST DIGIT
2996	016666	012605		MOV	(SP)+,R5	:: RESTORE R5
2997	016670	012604		MOV	(SP)+,R4	:: RESTORE R4
2998	016672	012603		MOV	(SP)+,R3	:: RESTORE R3
2999	016674	016666	000002 000004	MOV	2(SP),4(SP)	:: SET THE STACK FOR RETURNING
3000	016702	012616		MOV	(SP)+,(SP)	
3001	016704	000002		RTI		:: RETURN
3002	016706	000		85: .BYTE	0	:: STORAGE FOR ASCII DIGIT
3003	016707	000		.BYTE	00	:: TERMINATOR FOR TYPE ROUTINE
3004	016710	000		\$OCNT: .BYTE	00	:: OCTAL DIGIT COUNTER
3005	016711	000		\$OFILL: .BYTE	00	:: ZERO FILL SWITCH
3006	016712	000000		\$CMODE: .WORD	0	:: NUMBER OF DIGITS TO TYPE

```

3007          .SBTTL TRAP DECODER
3008
3009          ;*****
3010          ;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3011          ;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3012          ;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3013          ;GO TO THAT ROUTINE.
3014
3015          016714 010046          STRAP:  MOV  RO, -(SP)          ;;SAVE RO
3016          016716 016600 000002  MOV  2(SP),RO          ;;GET TRAP ADDRESS
3017          016722 005740          TST  -(RO)          ;;BACKUP BY 2
3018          016724 111000          MOVB (RO),RO          ;;GET RIGHT BYTE OF TRAP
3019          016726 006300          ASL  RO          ;;POSITION FOR INDEXING
3020          016730 016000 016736  MOV  STRPAD(RO),RO      ;;INDEX TO TABLE
3021          016734 000200          RTS   RO          ;;GO TO ROUTINE
3022
3023          .SBTTL TRAP TABLE
3024
3025          ;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3026          ;BY THE "TRAP" INSTRUCTION.
3027
3028          :          ROUTINE
3029          :          -----
3030          016736          STRPAD:
3031          016736 015736          STYPE  ;;CALL=TYPE          TRAP+0(104400) TTY TYPEOUT ROUTINE
3032          016740 016512          STYPOC ;;CALL=TYPOC        TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3033          016742 016466          STYPOS  ;;CALL=TYPOS        TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
3034          016744 016526          STYPON  ;;CALL=TYPON        TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
3035
3036
3037          016746 014534          SRDCHR  ;;CALL=RDCHR        TRAP+4(104404) TTY TYPEIN CHARACTER ROUTINE
3038          016750 014654          SRDLIN  ;;CALL=RDLIN        TRAP+5(104405) TTY TYPEIN STRING ROUTINE
3039          016752 015026          SRDOCT  ;;CALL=RDOCT        TRAP+6(104406) READ AN OCTAL NUMBER FROM TTY

```

K06

MAINDEC-11-DZADL-B MACY11 27.665) 2-DEC-76 16:29 PAGE 75
 DZADLB.P11 POWER DOWN AND UP ROUTINES

.SBTTL POWER DOWN AND UP ROUTINES

;*****

:POWER DOWN ROUTINE

```

$PWRDN: MOV    $ILLUP,2*PWRVEC    ;;SET FOR FAST UP
        MOV    #340,2*PWRVEC+2    ;;PRIO:7
        MOV    R0,-(SP)            ;;PUSH R0 ON STACK
        MOV    R1,-(SP)            ;;PUSH R1 ON STACK
        MOV    R2,-(SP)            ;;PUSH R2 ON STACK
        MOV    R3,-(SP)            ;;PUSH R3 ON STACK
        MOV    R4,-(SP)            ;;PUSH R4 ON STACK
        MOV    R5,-(SP)            ;;PUSH R5 ON STACK
        MOV    2SWR,-(SP)          ;;PUSH 2SWR ON STACK
        MOV    SP,$SAVR6          ;;SAVE SP
        MOV    $PWRUP,2*PWRVEC    ;;SET UP VECTOR
        HALT
        BR      .-2                ;;HANG UP

```

;*****

:POWER UP ROUTINE

```

$PWRUP: MOV    $ILLUP,2*PWRVEC    ;;SET FOR FAST DOWN
        MOV    $SAVR6,SP          ;;GET SP
        CLR    $SAVR6            ;;WAIT LOOP FOR THE TTY
        IS:    INC    $SAVR6      ;;WAIT FOR THE INC
        BNE    IS              ;;OF WORD
        MOV    (SP)+,2SWR        ;;POP STACK INTO 2SWR
        MOV    (SP)+,R5          ;;POP STACK INTO R5
        MOV    (SP)+,R4          ;;POP STACK INTO R4
        MOV    (SP)+,R3          ;;POP STACK INTO R3
        MOV    (SP)+,R2          ;;POP STACK INTO R2
        MOV    (SP)+,R1          ;;POP STACK INTO R1
        MOV    (SP)+,R0          ;;POP STACK INTO R0
        MOV    $PWRDN,2*PWRVEC    ;;SET UP THE POWER DOWN VECTOR
        MOV    #340,2*PWRVEC+2    ;;PRIO:7
        TYPE    $POWER            ;;REPORT THE POWER FAILURE
        SPWRMG: .WORD    $POWER    ;;POWER FAIL MESSAGE PCINTER
        MOV    (PC)+,(SP)        ;;RESTART AT BEG2
        SPWRAD: .WORD    BEG2      ;;RESTART ADDRESS
        RTI
        $ILLUP: HALT              ;;THE POWER UP SEQUENCE WAS STARTED
        BR      .-2              ;;BEFORE THE POWER DOWN WAS COMPLETE
        $SAVR6: 0                  ;;PUT THE SP HERE
        $POWER: .ASCIIZ    <15><12>"POWER"

```

.EVEN

.EVEN

```

DIST:  .BLKW    200.
BUFFER: .BLKW    4096.

```

```

;STATE-WIDTH DISTRIBUTION
;BUFFER AREA

```

.END

```

3040
3041
3042
3043
3044 016754 012737 017120 000024
3045 016762 012737 000340 000026
3046 016770 010046
3047 016772 010146
3048 016774 010246
3049 016776 010346
3050 017000 010446
3051 017002 010546
3052 017004 017746 162130
3053 017010 010637 017124
3054 017014 012737 017026 000024
3055 017022 000000
3056 017024 000776
3057
3058
3059
3060 017026 012737 017120 000024
3061 017034 013706 017124
3062 017040 005037 017124
3063 017044 005237 017124
3064 017050 001375
3065 017052 012677 162062
3066 017056 012605
3067 017060 012604
3068 017062 012603
3069 017064 012602
3070 017066 012601
3071 017070 012600
3072 017072 012737 016754 000024
3073 017100 012737 000340 000026
3074 017106 104400
3075 017110 017126
3076 017112 012716
3077 017114 002256
3078 017116 000002
3079 017120 000000
3080 017122 000776
3081 017124 000000
3082 017126 005015 047520 042527
3083 017134 000122
3084
3085
3086 017136 000310
3087 017756 010000
3088
3089 000001

```

L06

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 76
DZADLB.P11 CROSS REFERENCE TABLE

[illegible]

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 77
DZADLB.P11 CROSS REFERENCE TABLE

[illegible]

BYPASS	005614	1469	1485																
B10	011336	2105	2106																
B11	011372	2115	2116																
CH	012221	1393	2256																
CHN	012303	1964	2276																
CHNL	001366	730	1320	1520*	1601	1663	1937*	1938	1965	1995*	1996*	1997	2031*	2509					
CHNL	007016	1698																	
CH1	001354	225	1505*	1506*	1509	1515*	1541	1547	1571	1577*	1579*								
CH2	001356	226	1518*	1520	1533	1567*	1578*	1580*											
COLERR1	006744	1694	1686																
COLERR2	007010	1695	1697																
COMP	012146	761	2254																
CONVR	001116	1158	1193	1204	1216	1229	1241	1255	1277	1289	1300	1311	1337	2052					
CONV	006410	1611	1615																
CONVR	007100	1709	1712																
CONVRT	011004	1156	1191	1202	1214	1227	1239	1252	1263	1287	1298	1309	1550	2029					
CR	000015	410	2863	2873															
CRLF	000200	411	2834	2873															
CT	013573	841	2408																
CT	013576	1632	2409																
CT2	013600	1825	1915	2410															
CT3	013603	1912	2411																
DAC	001410	739	1325	1328	1570	1574	1650*	1652*	1653	1675*	1983	1986	1989	1992					
DASH	012175	1771	2262																
DAWAIT	005040	1251	1262	1296	1380														
DECTSP	177570	417	600	820															
DECPNT	014465	1776	1805	2502															
DECTYP	011406	1335	1531	1772	1778	1786	1790	1793	1804	1854	1958	1961	2123						
DECS	003612	1125	1126																
DELAY	001412	740	1654*	1655*															
DELAY1	007142	1715	1723																
DELAY2	007150	1717	1724	1725															
DELAY3	007074	1710	1711																
DELCLR	010254	1814	1826	1916	1925														
EX1	014532	687	694	700	706	2515													
EX1	014306	685	2482																
EX2	014344	704	2487																
EX3	014427	692	698	2496															
EXFLIN	006724	1377	1680																
EXSP	001142	600	820*	828*	1409*	2691*	2712*												
EXSPRM	000174	533	828																
EXIST	017136	1687	1747*	1817	3086														
EXC	012320	2087	2279																
EXC	177570	416	599	819															
EXC	014472	686	2507																
EXC	014524	705	2509																
EXC	014522	693	699	2512															
EXC	011236	1042	1045	1076	2083														
EXC	001364	729	1321*	1567*	1571*	1658	1938*	1997*											
EXC	011164	1033	1056	2065															
EXC	001414	741	1322*	1585*	1588*	1608*	1613*	1616*	1617*	1618*	1619*	1623	1668	2003*					
EDGEFLG	006514	2006*																	
ENTVEC	000030	1529*	1593*	1629	1639	1941*	2011*												
		505	803*	804*															

B07

MAINDEC-11-D2ADL-B MACY:1 27.665 2-DEC-76 16:29 PAGE 79
 D2ADL.B.P11 CROSS REFERENCE TABLE

[illegible]

MAINDEC-11-DZADL-B MACY: 27.665) 2-DEC-76 16:29 PAGE 80
DZADLB.P11 CROSS REFERENCE TABLE

NAME	001422	1783	1785																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
------	--------	------	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

[illegible]

		1765*	1766	1831*	1841	1843*	1849*	1850*	1851*	1852*	1853	1868	1891*	1892
		1893	1894*	1895	1897	1898	1900	2102*	2108*	2109*	2110*	2111*	2112*	2113*
		2114*	2117	2564	2565*	2566	2565*	2570	2574	2575	2576*	2580*	2962	2972*
R4	=%.0000034	2978*	2979*	2982*	2987*	2988*	2989	2998*	3049	3068*				
		424*	1254*	1275	1276*	1527*	1557	1592*	1662*	1663*	1664*	1665*	1666	1838*
		1840*	1841	1843	1867*	1903*	1905	2964	2966*	2967*	2968*	2969	2970*	2984
		2986*	2994*	2997*	3050	3067*								
R5	=%.0000005	425*	1156*	1158*	1191*	1193*	1202*	1204*	1214*	1216*	1227*	1229*	1239*	1241*
		1252*	1255*	1263*	1277*	1287*	1289*	1298*	1300*	1309*	1311*	1323*	1326*	1337*
		1351*	1353*	1355*	1363*	1367*	1509*	1550*	1569*	1572*	1577	1578	1595*	1644
		1678*	1844*	1845*	1846*	1856	1862*	1863	1888*	1904*	1905	1937	1943*	1981*
		1984*	1987*	1990*	1995	2013*	2029	2049*	2052	2053	2062	2063*	2965	2971*
		2973*	2975*	2976*	2977*	2978	2996*	3051	3066*					
R6	=%.0000006	426*	428	795*	796*	797								
R7	=%.0000007	427*	429											
SARSUB	006516	1323	1326	1568	1572	1644*	1981	1984	1987	1990				
SAR1	006542	1649*												
SETAF	006270	1594*	1587											
SETCH	012244	1540	2270*											
SETMSG	012050	1513	2241*											
SET1A	006170	1523	1524	1567*	1584									
SETB	006232	1363	1367	1577*										
SHIFT	006714	1674	1676*											
SKIPST	001350	723*	1694*	1754*	1777	1780								
SKPM5G	012424	1779	2293*											
SLASH	012333	781	1861	2281*										
SP	=%.0000006	428*	756*	762*	768*	776*	783	784	799*	817*	825*	829	866*	874
		897*	904	905*	913	921	1044	1066	1078	1095	1144*	1394*	1412*	1439*
		1470*	1515	1518	1533*	1541*	1552*	1624*	1633*	1766*	1856*	1863*	1965*	2070*
		2076*	2536*	2537*	2540*	2541*	2542	2546*	2547*	2548	2551	2553	2555*	2564*
		2569	2580	2581*	2582*	2583*	2604*	2605*	2606*	2607*	2608*	2610	2613*	2621*
		2622	2624	2625*	2627	2628	2629	2652*	2655	2657	2658	2687	2688	2692*
		2717	2737*	2740*	2756*	2761*	2782	2786*	2815*	2816	2826*	2828	2829	2830*
		2832	283	2836	2842	2844*	2846*	2854*	2858	2862	2863	2867	2881*	2882*
		2889	2890*	2901	2902*	2903*	2913	2914*	2919	2920	2955*	2956	2957	2958*
		2963*	2964*	2965*	2971	2996	2997	2998	2999*	3000*	3015*	3016	3046*	3047*
		3048*	3049*	3050*	3051*	3052*	3053	3061*	3065	3066	3067	3068	3069	3070
		3071	3076*											
SPACE	012224	1411	2267*											
SPREAC	001406	738*	2053*	2060	2509									
STACK =	001100	403*	762	768	799	866								
STATE	012201	1762	2263*											
STKLMT=	177774	414*												
STREG	001316	710*	976*	982	991*	995*	1006*	1007	1018*	1019	1036*	1037	1039	1060*
		1061	1063	1069*	1070	1073*	1089*	1090	1092	1105*	1106	1109*	1123*	1124*
		1127	1134*	1140*	1141	1173*	1183*	1386*	1402*	1404*	1439	1462*	1471*	1604*
		1611*	1660*	1666*	1702*	1712*	2034*	2040*	2076	2507	2509	2512		
SACIST	007566	1810	1812*											
SAP	001140	599*	773	776	783*	797	819*	821	827*	834*	1389	1391	1407	1505
		1931	2647	2661	2663	2669	2676	2713	2720	2732	2735	3052	3065*	
SAREG	000176	534*	827											
SAC	= 000001	467*												
SAC0	= 000001	457*	467											
SAC1	= 000002	456*	466											

F07

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 83
DZADLB.P11 CROSS REFERENCE TABLE

[illegible]

[illegible]

MAINDEC-11-DZADL-B MACY!! 27(665) 2-DEC-76 16:29 PAGE 85
DZADLB.P11 CROSS REFERENCE TABLE

VSO	011630	1160	1218	2169*										
VSOO	011642	1339	2174*											
WFAOI	011552	864	2152*											
WFTEST	001422	744*	789*	791*	2157									
WIDE	001342	720*	1691*	1758*	1788	1801								
WIDMSG	012505	1791	2302*											
WRAP	003712	1147*	1444	1454										
SAPTDO	00100.	559	565*											
SASTAT=	*****	2907	2922											
SATYC	016244	2878	2880*											
SATYI	016220	2876*												
SATY3	016226	2822	2877*											
SATY4	016236	2728	2879*											
SAUTOB	001134	596*												
SBASE	001250	660*	896	1471	1472	1473								
SBOADR	001122	591*												
SBODAT	001126	593*	896*	900	902*	982*	1039*	1063*	1092*	1127*	1128*	1129	1141*	1142
		2054*	2055	2507	2509	2512								
SBELL	001164	611*	2715	2747										
SCDWI	001254	662*												
SCHARC	016214	2839*	2849*	2856	2865*	2870*								
SCKSWR=	*****	3037												
SCMTAG	001100	579*	794	795	803	809	810	811						
SCM3 =	000000	609*												
SCNTLG	014777	2589*												
SCNTHU	014772	2588*												
SCPUOP	001222	634*												
SCRLE	001171	613*	1422	1443	2598	2723	2747	2755	2774	2779	2783	2938	2973	
SDEVCT	001204	625*												
SDEVM	001252	661*												
SODAGN	012006	2218	2224	2230*										
SENDAD	011776	545	2226*	2742										
SENDOCT	011756	809	2220*											
SENOING	012015	2222	2234*											
SENULL	012012	2233*												
SENV	001214	630*	2725	2817	2885	2909								
SENYM	001215	631*	832	2819	2824	2887								
SEOP	011722	1431	1448	1458	2210*									
SEOPTC	011750	809*	2217*	2221										
SERFLG	001103	582*	751*	2637	2665	2667	2673*	2695	2710*	2747				
SERMAX	001115	598*	812*	2667	2690*	2695								
SERROR	015406	803	2709*											
SERRPC	001116	589*	2507	2509	2512	2717*	2718*	2719	2747	2761				
SERRTB	001256	679*	2769											
SERTTY	015602	2722	2754*											
SERTTL	001112	586*	2716*	2747										
SESCAP	001162	610*	750*	753*	811*	2689*	2738	2740	2747					
SETABL	001214	629*												
SETEND	001256	571	663*											
SFATAL	001176	622*	2913*											
SFFLG	016464	2876*	2879*	2907	2916*	2924*								
SFILEC	001156	607*	2842	2873										
SFILES	001155	606*	2873											
SGACFR	001120	590*												

SGDDAT	001124	592*	932*	935*	936	944*	952*	960*	967*	975*	992*	1005*	1040*	1087*
SGET42	011766	1108*	1121*	1129	1140	1142	1175*	2052*	2056	2507	2509			
SGTSWR=	***** J	2223*												
SHC	= 000000	3036												
SHIBTS	001000	400												
SHIOCT	015126	566*												
SICNT	001104	2626*	2631*											
SILLUP	017120	583*	2680*	2681	2683*	2694								
SINTAG	001135	3044	3060	3079*										
SITEMB	001114	597*												
SLE	001172	587*	2719*	2727	2747	2758								
SLFLG	016463	614*	2579	2588	2747	2873								
SLPAOR	001106	2917*	2923*											
SLPERR	001110	594*	813*	930*	1155*	2068*	2086*	2671*	2687*	2692	2694			
SMADR1	001226	585*	814*	931*	1154*	2067*	2085*	2671	2688*	2694	2737			
SMADR2	001232	647*												
SMADR3	001236	651*												
SMADR4	001242	654*												
SMAL	001174	657*												
SMAMS1	001224	567	571	620*	831	2686	2725	2817						
SMAMS2	001230	641*												
SMAMS3	001234	649*												
SMAMS4	001240	652*												
SMBAOR	001002	655*												
SMFLG	016462	567*												
SMNEW	015015	2877*	2683	2918*	2922*									
SMSGAD	001210	2592*												
SMSGLG	001212	627*	2893*	2896										
SMSGTY	001174	628*	2898*											
SMSWR	015004	621*	2891	2899*	2911	2915*								
SMTYP1	001225	2590*												
SMTYP2	001231	642*												
SMTYP3	001235	650*												
SMTYP4	001241	653*												
SMXCNT	015404	656*												
SNJLL	001154	2684	2694*											
SNWTS=	000001	605*	2844	2873										
		927*	939*	948*	956*	963*	970*	986*	1000*	1014*	1024*	1047*	1082*	1101*
		1117*	1148*	1166*	1186*	1197*	1209*	1222*	1234*	1245*	1293*	1304*	1315*	1345*
		1358*	1370*											
SOCNT	016710	2962*	2991*	3004*										
SOMODE	016712	2957*	2961*	2966	2969*	2980*	3006*							
SOVER	015370	2648	2664	2672	2682	2691*								
SPASS	001202	624*	831*	1375	1435*	1452*	2065	2083	2214*	2215*	2233	2678	2695	
SPASTH	001006	569*												
SPOWER	017126	3075	3082*											
SPWPAO	017114	3077*												
SPWPDN	016754	807	3044*	3072										
SPWPMG	017110	3075*												
SPWRUP	017026	3054	3060*											
SQJES	001170	612*	2572	2588	2747	2873								
SPDCHA	014534	2536*	3037											
SPDDEC=	***** U	3040												
SFOLIN	014654	2564*	3038											

[illegible]

K07

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 88
DZADLB.P11 CROSS REFERENCE TABLE

```
SSGET4= 000000
SOFILL 016711
$4OCAT= ***** U
.       = 037756
```

[illegible]

.SASTA= ***** U
.SX = 001000

MAINDEC-11-DZADL-B MACY11 27 665) 2-DEC-76 16:29 PAGE 89
DZADLB.P11 CROSS REFERENCE TABLE

[illegible]

M07

MAINDEC-11-DZADL-B MACY:1 27(665) 2-DEC-76 16:29 PAGE 90
DZADLB.P11 CROSS REFERENCE TABLE

.SAPT8	18	3908	6168
.SAPTH	18	3908	550
.SAPTY	18	3908	2873
.SASTA	18		
.SCATC	18	3908	526
.SCMTA	18	3908	572
.SDB2D	18		
.SDB20	18		
.SDIV	18		
.SECP	18	3908	2200
.SEERR0	18	3908	2695
.SEERR1	18	3908	2747
.SMULT	18		
.SPARM	3908		
.SPOWE	18	3908	3040
.SRAND	18	3908	
.SRDCE	18		
.SRDCC	18	3908	2594
.SREAO	18	3908	2520
.SR2A2	18		
.SSAVE	18	3908	
.SSB2D	18		
.SSB20	18		
.SSCOP	18	3908	2632
.SSIZE	18		
.SSPAC	3908		
.SSJPR	18		
.SSWDC	3908		
.STRAP	18	3908	3007
.STYP8	18		
.STYPC	18	3908	
.STYPE	18	3908	2794
.STYPC	18	3908	2930
.S4CCA	18		
.117C	18		

200	1619	1741	1852	1881	2008	2010	2020	2023	2026	2048					
200	902	1144	1328	1462	1463	1464	1465	1466	1470	1475	1481	1570	1579	1590	1613
	1631	1652	1699	1789	1801	1802	1836	1882	1983	1989	1995	2018	2019	2021	2022
2024	2024	2025	2030	2042	2622	2769	2830	2890	2902	2914	2958	2968			
25	1331	1645	1646	1647	1648	1718	1746	2615	2617	2619	2766	2767	2768	3019	
25R	1616	1617	1618	1676	1738	1739	1740	1748	1765	1846	1849	1850	1851	1878	1879
	1880	2007	2009	2045	2046	2047	2108	2109	2110	2111	2112	2897			
250	1784	833	850	868	993	1130	1376	1408	1461	1488	1630	1709	1715	1717	1724
	1781	1796	1813	1874	1932	2134	2154	2224	2614	2662	2664	2666	2670	2679	2711
250	2714	2736	2739	2771	2776	2789	2820	2833	2868	2884	2988	2908	2910	2985	
250	1669	1750	1896	2692											
250	1556	1674	1971	1973	2061	2218	2554	2992							
250	2668														
250	758	840	924	995	1128	1390	1477	1506	1708	2103	2113	2215	2541	2547	2555
250	2621	2982													
250	625														
250	204	981	1012	1285	1401	1603	1660	1665	1701	2091	2104	2114	2197	2987	2988
250	1391	2146	2147	2148	2758										
250	832	1407	1931	2647	2661	2669	2676	2713	2720	2735					
250	1757	2819	2824	2856	2887										
250	2567	1808	1812	1869	1899	2128									
250	1743		2100	2552	2847	2993									
250	1723														
250	766	772	798	822	838	844	847	852	877	880	880	883	886	889	892
250	174	936	979	994	1112	1126	1143	1172	1179	1266	1272	1283	1322	1392	1419
250	174	1494	1587	1607	1615	1656	1673	1677	1686	1697	1711	1729	1734	1753	1760
250	174	1824	1848	1885	1902	1907	1923	1929	1930	2005	2037	2044	2066	2084	2109
250	174	2024	2043	2049	2071	2077	2048	2077	2072	2076	2074	2059	2081	2018	2023
250	174	2024	2043	2057	2064	2096	2092	2095	2092	2093	3064				
250	174	2024	2043	2062	2071	2091	2107	2125	2130	2120	1839	1952	1955	2058	2097
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					
250	174	2024	2043	2164	2139	2145	2133	2112	2061	2081					

[illegible]

980	963	965	967	970	972	974	975	983	986	988	990	1000	1002	1004	1004
1020	1094	1016	1018	1024	1026	1028	1043	1047	1049	1051	1077	1082	1082	1084	1084
1168	1194	1101	1103	1105	1117	1119	1121	1130	1143	1143	1150	1152	1152	1153	1156
1214	1220	1171	1172	1186	1189	1190	1191	1197	1199	1201	1202	1209	1209	1211	1213
1258	1260	1224	1226	1227	1234	1236	1238	1239	1245	1247	1249	1250	1250	1293	1295
1358	1360	1304	1306	1308	1309	1315	1317	1319	1320	1343	1345	1347	1347	1349	1350
1544	1545	1362	1363	1370	1372	1374	1375	1392	1397	1398	1415	1416	1416	1536	1537
1688	1689	1555	1556	1627	1628	1636	1637	1769	1770	1859	1860	1866	1866	1867	1932
2017	2017	2033	2034	2079	2080	2154	2156	2202	2203	2204	2205	2206	2206	2207	2211
2565	2566	2222	2223	2225	2231	2233	2234	2236	2522	2524	2525	2528	2528	2529	2557
2661	2662	2570	2571	2587	2588	2594	2596	2599	2611	2634	2637	2642	2642	2647	2659
2700	2700	2663	2665	2666	2667	2676	2678	2686	2688	2693	2694	2695	2695	2697	2700
2700	2700	2720	2722	2723	2725	2732	2735	2742	2746	2747	2749	2764	2764	2780	2796
3036	3037	3038	3039	3040	3042	3052	3053	3058	3019	3023	3032	3033	3033	3034	3035
402	517	520	521	522	542	546	548	553	555	562	575	578	578	581	609
616	619	750	779	781	799	877	880	893	886	889	892	903	903	909	918
920	922	927	928	929	930	931	937	940	941	942	943	949	949	950	951
988	989	958	959	960	964	965	966	967	971	972	973	974	974	984	987
1044	1048	990	1001	1002	1003	1004	1015	1016	1017	1018	1025	1026	1026	1027	1028
1118	1119	1049	1050	1051	1078	1083	1084	1085	1086	1095	1102	1103	1103	1104	1105
1173	1187	1120	1121	1130	1143	1148	1149	1150	1151	1152	1167	1168	1168	1169	1170
1225	1226	1186	1189	1190	1198	1199	1200	1201	1210	1211	1212	1213	1213	1223	1224
1305	1306	1235	1236	1237	1238	1246	1247	1248	1249	1250	1294	1295	1295	1296	1297
1355	1360	1307	1308	1316	1317	1318	1319	1320	1344	1346	1347	1348	1348	1349	1350
1545	1556	1361	1362	1363	1371	1372	1373	1374	1375	1392	1398	1416	1416	1417	1537
2074	2079	1557	1628	1629	1637	1638	1770	1771	1860	1861	1867	1869	1869	1932	1969
2536	2557	2081	2154	2156	2203	2206	2211	2218	2221	2233	2523	2525	2525	2529	2531
2700	2713	2558	2567	2571	2588	2597	2635	2660	2663	2664	2667	2694	2694	2695	2698
2531	2536	2742	2615	2631	2632	2675	2723	2794	2663	3010					
2525	2529	2532	2611	2615	2631	2673	2722	2797	2663						
390	395	400	514	515	516	518	521		532	615	619	777	777	800	903
809	810	811	813	814	906	1395	1413		1534	1542	1553	1625	1634	1767	1857
1864	1966	2071	2077	2204	2205	2212	2213		2233	2236	2523	2580	2588	2594	2638
2633	2640	2641	2642	2643	2647	2674	2675		2691	2694	2695	2701	2702	2703	2704
2705	2710	2735	2742	2747	2762	2787	2873		3031	3032	3033	3034	3037	3038	3039
680	927	939	948	956	963	970	986		1000	1014	1024	1047	1082	1101	1117
1148	1166	1186	1197	1209	1222	1234	1245		1293	1304	1315	1345	1358	1370	2211
2506	2627	2881	2882	2903	2919	2920	3046		3052	3065	3066				
	390	510	521	532	609	616	619		680	815	927	931	939	943	948
95	956	960	963	967	970	974	986		990	1000	1004	1014	1018	1024	1028
1047	1051	1082	1086	1101	1105	1117	1121		1148	1152	1166	1170	1186	1190	1197
1201	1209	1213	1222	1234	1238	1238	1245		1249	1293	1297	1304	1308	1315	1319
1345	1349	1358	1362	1370	1374	2212	2225		2557	2642	2742	3023	3031	3032	3033
3034	3035	3037	3038	3039	3040										
	522	572	683	831											
	510	616	815												
MACRO															
MCALL															
MEFIT															
MLIST															
	390	510	521	532	609	616	619		680	815	927	931	939	943	948
	956	960	963	967	970	974	986		990	1000	1004	1014	1018	1024	1028
	1051	1082	1086	1101	1105	1117	1121		1148	1152	1166	1170	1186	1190	1197
	1209	1213	1222	1234	1238	1238	1245		1249	1293	1297	1304	1308	1315	1319
	1349	1358	1362	1370	1374	2212	2225		2557	2642	2742	3023	3031	3032	3033

E08

MAINDEC-11-DZADL-B MACY11 27(665) 2-DEC-76 16:29 PAGE 95
 DZADLB.P11 CROSS REFERENCE TABLE

	3034	3035	3037	3038	3039	3040									
.PAGE	572	665													
.REN	9														
.REPT	532														
.SBTTL	400	510	526	535	539	550	572	616	665	709	755	788	793	839	854
	927	939	948	956	963	970	986	1000	1014	1024	1047	1082	1101	1117	1146
	1148	1166	1186	1197	1209	1222	1234	1245	1293	1304	1315	1345	1358	1370	1384
	1424	1433	1450	1459	1501	1511	2200	2237	2520	2594	2632	2695	2747	2794	2873
	2930	3007	3023	3040											
.TITLE	390														
.WORD	532	533	534	547	566	567	568	569	570	571	580	583	584	595	586
	589	590	591	592	593	594	595	598	599	600	621	622	623	624	625
	626	627	628	632	633	634	647	651	654	657	658	659	660	661	662
	2217	2220	2232	2631	2773	2779	2823	2870	2905	3006	3075	3077			

ERRORS DETECTED: 0

* DZADLB.SEG/SC-ORF/ML:TOC=DSKZ:SYSMAC.SML.DZADLB.P11
 RUN-TIME: 05 62 8 SECONDS
 CORE USED: 33K

F08

Spooler runtime 14 Seconds, 62 KCS, 340 disk reads, 3 disk writes, 95 pages

~~ACTION - MONITORING~~, BIRMINGHAM File # 70-86190 Date 03-Dec-76 09:44:21 Monitor IPC-B 602 (600) ENDOWN

[illegible]

10			...B1	2130	011432	105037	...B5
63			...C1	2175	011642	000062	...C5
116			...D1	2209			...D5
			...E1	2246	012106	023440	...E5
172			...F1	2299	012470	020051	...F5
213			...G1	2351	013076	020040	...G5
241			...H1	2405	013556	020122	...H5
			...I1				...I5
297			...J1	2466	014162	020056	...J5
			...K1	2510	014512	001124	...K5
352			...L1	2529			...L5
399			...M1	2583	014750	012766	...M5
453		000020	...N1	2603			...N5
507		000060	...B2	2641			...B6
549		000214	...C2	2695			...C6
591	001102	000	...D2	2749			...D6
635			...E2	2803			...E6
674			...F2	2857	016142	001372	...F6
719	001336	000000	...G2	2911	016412	005737	...G6
764	001510	000137	...H2	2939			...H6
797	001644	022706	...I2	2993	016660	002402	...I6
845	002144	004737	...J2	3016	016716	016600	...J6
863	002244	012737	...K2	3049	016776	010346	...K6
913	002522	012637	...L2	ADDW11=	000000		...L6
935	002612	006137	...M2	BASECH	001336		...M6
979	002760	001375	...N2	CLEAR1	006744		...N6
1033	003152	004737	...B3	FIXADR	005500		...B7
1087	003420	012737	...C3	NOITST	010312		...C7
1126	003614	001376	...D3	RANDY	010722		...D7
1155	003736	012737	...E3				...E7
1206	004126	011626	...F3	SW10 =	002000		...F7
1254	004270	013704	...G3	TST34	005014		...G7
1302	004512	011632	...H3	\$ATYC	016244		...H7
1324	004604	000062	...I3	\$INTAG	001135		...I7
1354	004746	000007	...J3	\$STUP =	177777		...J7
1393	005112	104400	...K3				...K7
1433			...L3	NEWTST	1#	510#	...L7
1468	005472	005337	...M3	.SEOP	1#	390#	...M7
1520	005770	013737	...N3		2714	2736	...N7
1570	006204	063702	...B4		1216	1227	...B8
1605	006362	012700	...C4	.ASCIZ	611	614	...C8
1649	006542	012737	...D4		2217	2220	...D8
1688	006756	012701	...E4	.TITLE	390	3007	...E8
1742	007222	020227	...F4	**END**	USER DAVIES, TOM		...F8
1794	007466	104400	...G4				
1837	007674	010120	...H4				
1886	010056	012700	...I4				
1934	010304	104400	...J4				
1987	010544	004537	...K4				
2037	011036	001376	...L4				
2073	011216	002	...M4				
2098	011310	005002	...N4				