

.REM *

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZLPC-C
PRODUCT NAME:	LABORATORY PERIPHERAL SYSTEM DIAGNOSTIC TEST I
DATE:	MAY 21, 1976
MAINTAINER:	DIAGNOSTIC GROUP

FIRST PRINTING, 1973

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1973,1974,1976 BY DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

THIS DIAGNOSTIC TESTS AND EXERCISES THE "LPS-11". THE PROGRAM IS SELF-STARTING AND WHEN LOADED WILL TYPE OUT THE PROGRAM TITLE. A MESSAGE IS THEN PRINTED GIVING THE LETTER DESIGNATORS TO BE TYPED TO RUN ANY ONE OF THE SIX (6) SEPERATE TESTS OF WHICH THIS PROGRAM IS COMPRISED. THE PROGRAM THEN TYPES A 'CR .' AND THEN WAITS IN A KEYBOARD MONITOR MODE FOR A LETTER TO BE TYPED. ALTHOUGH THESE TESTS MAY BE RUN IN ANY ORDER IT IS IMPERATIVE THAT THE 'LOGIC' TESTS ARE RUN FIRST AND PROVED FULLY OPERATIONAL.

THE PROGRAM IS SET UP TO GIVE THE OPERATOR AS MUCH CONTROL OVER THE PROGRAM AS POSSIBLE VIA THE TELETYPE. TYPING A '!C' (OBTAINED VIA TYPING THE 'CNTR' AND 'C' KEYS SIMULTANEOUSLY) WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND AWAIT A NEW LETTER DESIGNATOR TO BE TYPED. TYPING A '!A' WHILE IN MONITOR MODE WILL ENABLE THE LETTER DESIGNATORS TO BE RETYPED. IF RUNNING ON A NON-SWITCH REGISTER CPU, TYPING A 'CTRL G' WILL ALLOW THE CHANGING OF A SOFTWARE SWITCH REGISTER.

2. REQUIREMENTS (EQUIPMENT)

- A. PDP-11 COMPUTER WITH 4K OF MEMORY
- B. TELETYPE
- C. LPS11 OPTION BOX WITH:
 - LPSAD12 SIMPLE A TO D CONTROL AND
 - LPSADNP DMA A TO D CONTROL AND/OR
 - LPSSH DUAL SAMPLE AND HOLD CONTROL.

3. LOADING PROCEDURE

- A. USE STANDARD PROCEDURE FOR LOADING BINARY TAPES.

4. STARTING PROCEDURE

THE PROGRAM IS SELF STARTING WITH A RESTART ADDRESS OF '174'
RE-INITIALIZATION ADDRESS OF '200'.

5. CONSOLE SWITCH SETTINGS

- A. ALL SWITCHES SHOULD BE DOWN (0) WHEN THE PROGRAM IS STARTED.
- B. REFER TO THE INDIVIDUAL TEST DESCRIPTIONS FOR APPLICABLE CONSOLE SWITCH SETTINGS
- C. REFER TO 15. FOR SOFTWARE SWITCH REGISTER OPERATION.

* TYPE 'CARRIAGE RETURN' (CR) TO TERMINATE ALL INPUT DATA.

B. A TO D LOGIC TEST

A. THE LOGIC TEST CONSISTS OF 26 SUBTESTS WHICH CHECK OUT THE 'LPS A TO D' LOGIC. EACH TEST IS LOOPEO '2048' TIMES TO TEST LOGIC RELIABILITY.

B. STARTING SEQUENCE

1. TYPE 'A' TO RUN THE A TO D LOGIC TEST.
2. THE PROGRAM WILL THEN EXECUTE THE A TO D LOGIC TEST.

C. CONTROL SWITCHES1. ↑C (CONTROL C)

TYPING A ↑C AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT THE 'A TO D LOGIC' TEST AND RETURN TO THE MONITOR.

2. CONSOLE SWITCHFUNCTION

CONSOLE SW11=0	NORMAL RUN (2048 PASSES/TEST)
CONSOLE SW11=1	SUPPRESS SUBPROGRAM ITERATIONS
CONSOLE SW13=0	PRINT ERROR MESSAGE
CONSOLE SW13=1	INHIBIT ERROR MESSAGES
CONSOLE SW14=1	RUN SCOPE MODE
CONSOLE SW14=0	INHIBIT SCOPE MODE
CONSOLE SW15=0	CONTINUE AFTER ERROR
CONSOLE SW15=1	HALT ON ERROR

D. LOGIC ERRORS

ON ENCOUNTERING AN ERROR (DATA SWITCHES DOWN) THE ERROR ADDRESS AND THE CONTENTS OF THE A/D STATUS REGISTER AND A/D BUFFER AND DMA STATUS REGISTER ARE TYPED OUT.

E. RESTRICTIONS

DO 'NOT' CONNECT EXTERNAL SYNC FOR THE LOGIC TEST.

F. TEST TIME

IT TAKES APPROXIMATELY 90 SEC. TO RUN THE LOGIC TEST AND RING THE TELETYPE BELL.

7. A TO D DMA LOGIC TEST

A. THE LOGIC TEST CONSISTS OF 18 PARTS WHICH CHECK OUT THE LOGIC USED ONLY IN THE DIRECT MEMORY ACCESS MODE. IT IS IMPORTANT THAT THE A TO D BASIC LOGIC TEST ALSO RUN

B. STARTING SEQUENCE

1. TYPE 'B' TO RUN THE A TO D DMA LOGIC TEST.
2. THE PROGRAM WILL THEN EXECUTE THE A TO D DMA LOGIC TEST.

C. CONTROL SWITCHES1. ↑C (CONTROL C)

TYPING A ↑C AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT THE 'A TO D DMA LOGIC' TEST AND RETURN TO THE MONITOR.

2. CONSOLE SWITCH FUNCTION

CONSOLE SW11=0	NORMAL RUN (2048) PASSES/TEST)
CONSOLE SW11=1	SUPPRESS SUBPROGRAM INTERACTIONS
CONSOLE SW13=0	PRINT ERROR MESSAGE
CONSOLE SW13=1	INHIBIT ERROR MESSAGES
CONSOLE SW14=0	INHIBIT SCOPE MODE
CONSOLE SW14=1	RUN SCOPE MODE
CONSOLE SW15=0	CONTINUE AFTER ERROR
CONSOLE SW15=1	HALT ON ERROR

D. LOGIC ERRORS

ON ENCOUNTERING AN ERROR (DATA SWITCHES DOWN) THE ERROR ADDRESS AND THE CONTENTS OF THE DMA REGISTER ARE TYPED OUT.

E. RESTRICTIONS

NONE

F. TEST TIME

IT TAKES APPROXIMATELY 10 SECONDS TO RUN THE DMA LOGIC TEST AND RING THE TELETYPE BELL.

3. A TO D CALIBRATION TEST

A. THE 'A/D CALIBRATION' TEST IS DESIGNED TO ACCEPT AN INPUT FROM THE TELETYPE TO INDICATE THE TYPE OF SYNC (EXTERNAL, INTERNAL OR LPS CLOCK TO BE USED AND THEN TAKES CONTINUOUS CONVERSIONS USING THE 'CH.' SELECTED VIA THE CONSOLE SWITCHES. THESE SETTINGS MAY BE CHANGED AT ANY TIME.

B. STARTING SEQUENCE

1. TYPE 'C' TO RUN THE A/D CALIBRATION TEST.
2. THE TEST HEADER PLUS A REQUEST FOR A SYNC TYPE WILL THEN BE TYPED.
3. TYPE IN THE DESIRED SYNC, 'I' FOR INTERNAL, 'E' FOR EXTERNAL 'C' FOR LPS CLOCK (IF INSTALLED) FOLLOWED BY 'CR'.
4. THE TEST WILL START.

C. CONTROL SWITCHES1. ↑A (CONTROL A)

TYPING ↑A WILL ENABLE A NEW SYNC TYPE TO BE ENTERED.

2. ↑C (CONTROL C)

TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT THE CALIBRATION TEST AND RETURN TO THE MONITOR.

3. CONSOLE SWITCH FUNCTION

CONSOLE SW '0-5'	CHANNEL SELECT
CONSOLE SW '10=0'	DISPLAY VALUE IN LED DISPLAY
CONSOLE SW '10=1'	PRINT CONVERSION VALUE

D. CALIBRATION ERRORS

THE PRINTOUT 'ERROR BIT SET' WILL OCCUR WHILE RUNNING WITH EXTERNAL SYNC IF THE SYNC FREQUENCY IS TOO FAST.

9. A TO D REPEATIBILITY TEST

A. THIS TEST REQUESTS A CH.(S) AND A COUNT SPREAD OF '1-4' AND MODE OF OPERATION TO BE TYPED IN BY THE OPERATOR. A SERIES OF '512' CONVERSIONS ARE THEN TAKEN ON THE INPUT CH.(S) CONVERSIONS ARE THEN AVERAGED OUT AND IF THE COUNT SPREAD IS FOUND TO BE GREATER THAN REQUEST, THE RESULTS OF THE CONVERSIONS ARE TYPED OUT. A SINGLE CHANNEL OR A SERIES OF CHANNELS MAY BE TESTED VIA TYPING EITHER 'N(CR)' TO SELECT A SINGLE CHANNEL OR 'N.N(CR)' TO TEST A SERIES OF CHANNELS.

B. STARTING SEQUENCE

1. TYPE 'D' TO RUN THE 'REPEATIBILITY' TEST.
2. A REQUEST IS THEN MADE FOR CH.(S) TO BE TESTED AND COUNT SPREAD (RANGE IN WHICH ALL 512 COUNTS MUST FALL FOR THE CH. TO BE CONSIDERED ACCEPTABLE).
3. A REQUEST IS THEN MADE FOR MODE OF OPERATION (NPR OR BR) TYPE 'I' FOR PROGRAM SAMPLING OR TYPE 'B' FOR BURST NPR OR 'D' FOR DUAL MODE SAMPLING IF INSTALLED.
4. IF THE CHANNEL IS FOUND TO BE WITHIN THE SELECTED COUNT SPREAD, THE PROGRAM WILL EITHER CONTINUE TO THE NEXT CHANNEL IF SELECTED OR RETEST THE CURRENT CHANNEL.

C. CONTROL SWITCHES
-----1. ↑A (CONTROL A)

TYPING A ↑A WHILE THE PROGRAM IS RUNNING WILL ENABLE A NEW CH.(S) AND COUNT SPREAD TO BE SELECTED.

2. ↑C (CONTROL)

TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT THE 'REPEATIBILITY' TEST AND RETURN TO THE MONITOR.

3. CONSOLE SWITCHES
-----FUNCTION

CONSOLE SW 10=0	PRINT ERRORS ONLY
CONSOLE SW 10=1	PRINT OUT ALL CONVERSIONS
CONSOLE SW 13=0	PRINT ERRORS
CONSOLE SW 13=1	INHIBIT ERROR PRINTOUTS
CONSOLE SW 15=0	DO NOT HALT UPON REPEATIBILITY REPORT
CONSOLE SW 15=1	HALT UPON REPEATIBILITY REPORT

D. REPEATIBILITY ERRORS

ON ENCOUNTERING AN ERROR (CONSOLE SWITCHES DOWN) THE ERROR DATA IS TYPED OUT.

1. ERROR FORMAT

CH.														
A														
	HI													
	B													
		AV												
		C												
			LO											
			D											
LO	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	HI		
E	F	G	H	I	J	K	L	M	N	O	P	Q		

WHERE:

A=CHANNEL BEING TESTED
B=THE HIGHEST READING OF THE '512' CONVERSIONS
C=THE AVERAGE READING OF THE '512' CONVERSIONS
D=THE LOWEST READING OF THE '512' CONVERSIONS
E=NUMBER OF COUNTS 'OUT OF RANGE' LOWER THAN 5 COUNTS
F-J=NUMBER OF COUNTS IN EACH PART LOWER THAN AVERAGE.
K=NUMBER OF COUNTS AT AVERAGE OF THE '512'
L-P=NUMBER OF COUNTS IN EACH PART HIGHER THAN AVERAGE.
Q=NUMBER OF COUNTS 'OUT OF RANGE' HIGHER THAN 5 COUNTS

E. RESTRICTIONS

1. IF A SECOND CH. IS ENTERED, IT MUST BE LARGER THAN THE FIRST CH. OR THE INPUT WILL NOT BE ACCEPTED.

10. A TO D RECOVERY TEST

A. THE "RECOVERY TEST" IS DESIGNED TO DETERMINE THE RECOVERY CAPABILITY OF THE 'LPS A TO D'. THE TEST REQUESTS FOR TWO (2) CH. INPUTS TO BE TYPED IN. THE TEST THEN TAKES A SERIES OF SIXTEEN (16) CONVERSIONS (8 ON EACH CH.) AND THEN TYPES OUT THE 'B' CONVERSION VALUES IN THE ORDER THEY WERE TAKEN ON THE SECOND CH.

B. STARTING SEQUENCE

1. TYPE 'E' TO RUN THE RECOVERY TEST.
2. A REQUEST IS THEN MADE FOR THE CH.S TO BE TESTED.
3. TYPE 'N,N (CR)' WHERE 'N' IS ANY CH.
4. THE PROGRAM WILL THEN TAKE CONTINUOUS CONVERSIONS TYPING OUT THE CONVERSION VALUES FOR THE SECOND CH.

EXAMPLE:

CH. A XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX

WHERE:

A=TO THE SECOND CH.
X=TO THE 'B' CONVERSIONS TAKEN ON THAT CH.

C. CONTROL SWITCHES

1. ↑A (CONTROL A)

TYPING A '↑A' WILL ENABLE A NEW SET OF CH.S BE ENTERED.

2. ↑C (CONTROL C)

TYPING A '↑C' WILL ENABLE THE PROGRAM TO RETURN TO THE MONITOR.

3. CONSOLE SWITCHES FUNCTION

CONSOLE SW 10 = 0	PRINT RECOVERY REPORT
CONSOLE SW 10 = 1	INHIBIT PRINTING OF RECOVERY REPORT

D. RESTRICTIONS

DO NOT EXECUTE THIS TEST IF THE WIDE BAND JUMPERS HAVE BEEN REMOVED.

11. DUAL MODE DYNAMIC LOGIC TEST

A. THE 'DUAL MODE' TEST IS DESIGNED TO TEST THE DYNAMIC DUAL SAMPLE AND HOLD LOGIC OF THE LPS. A SAMPLE IS TAKEN FROM A CHANNEL PAIR (IE. CHANNEL 0 AND 10) NOT IN DUAL MODE. ANOTHER SAMPLE IS TAKEN FROM A CHANNEL PAIR USING 'DUAL MODE'. THE RESULTING CONVERSIONS ARE THEN COMPARED FOR VALUE AND SIMILARITY.

B. STARTING SEQUENCE

1. TYPE 'F' TO RUN THE 'DUAL MODE LOGIC TEST'.
2. THE PROGRAM WILL THEN EXECUTE THE DUAL MODE TEST ON CHANNEL PAIRS 0-3.

C. CONTROL SWITCHES1. ↑C (CONTROL C)

TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT TO THE MONITOR.

2. ↑A (CONTROL A)

TYPING ↑A WILL CAUSE THE RESTARTING OF THE DUAL MODE LOGIC TEST.

3. CONSOLE SWITCHES FUNCTION

CONSOLE SW13=0	PRINT ERROR MESSAGES
CONSOLE SW13=1	INHIBIT ERROR MESSAGES
CONSOLE SW15=0	CONTINUE AFTER ERROR
CONSOLE SW15=1	HALT ON ERROR

12. MANUAL STARTING ADDRESSES

EXTERNAL STARTING ADDRESSES OF THE BASIC TEST HAVE BEEN PROVIDED FOR THE USER TO MANUAL START THESE TESTS.

<u>STARTING ADDRESS</u>	<u>BASIC TEST</u>
204	A TO D LOGIC TEST
210	DMA LOGIC TEST
214	DUAL SAMPLE LOGIC TEST
220	A TO D CALIBRATION TEST

13. MISC. INFORMATION

THE PROGRAM CAN BE CHAINED UNDER XXDP/ACT-11 IF THE PROGRAM WAS LOADED FROM ACT-11 OR DDP, THE A TO D LOGIC TEST AND (IF INSTALLED) DMA LOGIC TEST WILL BE RUN. THIS PROGRAM DOES NOT HAVE THE "APT" HOOKS.

14. OPERATOR VARIABLE LOCATIONS

LOCATION 1000 CONTAINS THE LPS STARTING DEVICE ADDRESS <170400>
 LOCATION 1002 CONTAINS THE LPS STARTING VECTOR <340>
 LOCATION 1004 CONTAINS THE LPS A TO D BR LEVEL <6><300>
 LOCATION 1010 CONTAINS THE + OR - 37 COUNT SPREAD FOR THE DUAL SAMPLE TEST.
 LOCATION 1012 CONTAINS THE TTY FILLER CHARACTER COUNT
 LOCATION 1014 CONTAINS THE TTY FILLER CHARACTER
 LOCATION 170 CONTAINS THE SOFTWARE SWITCH REGISTER VALUE
 LOCATION 172 CONTAINS THE SOFTWARE DISPLAY REGISTER VALUE.

NOTE: IF LOCATIONS 1000 OR 1002 ARE CHANGED, THE TEST MUST BE RE-INITIALIZED AT 200.

15. SOFTWARE SWITCH REGISTER OPERATION

THE PROGRAM SUPPORTS NON-SWITCH REGISTER CPU TYPES. THIS IS ACCOMPLISHED BY TYPING A "CTRL G". THE RESPONSE WILL REPORT THE VALUE OF THE OLD SOFTWARE SWITCH REGISTER AND WAIT FOR A NEW VALUE. THE OPERATOR NOW INPUTS THE NEW VALUE AND TERMINATES IT WITH A "CR". IF THE OPERATOR TYPES A "CR" WITH NO INPUT, THE SWITCH REGISTER IS SET TO 0. UPON TERMINATING, THE PROGRAM WILL RESUME RUNNING THE APPROPATE TEST.

488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529

.TITLE LPS-11 DIAGNOSTIC TEST I MAINDEC-11-DZLPC-C
.ENABL ABS,AMA
.LIST ME
.NLIST MD,MC,CND

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1

;SWITCH REGISTER DEFINITIONS AND FUNCTIONS:

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

SW15=100000
SW14=40000
SW13=20000
SW12=10000
SW11=4000
SW10=2000
SW09=1000
SW08=400
SW07=200
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

=1, HALT ON ERROR
=1, LOG ON CURRENT TEST
=1, SUPPRESS ERROR TYPEOUT

=1, SUPPRESS 'SUBPROGRAM' ITERATIONS
=1, FORCE TYPEOUT (REPEATABILITY)

;REGISTER DEFINITIONS

000000
000001
000002
000003
000004
000005
000006
000007

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7

```

540          ;LOAD TRAP CATCHER INTO LOC'S 0-1000
541          .=0
542 000000 000000 HALT
543 000002 000000 HALT
544          .=24
545 000024 011676 PWRFAL          ;POWER FAIL HANDLER
546 000026 000340 340
547          .=60
548 000060 010414 XTTYIN          ;TELEPRINTER KEYBOARD ROUTINE
549 000062 000340 340
550          .=30
551 000030 011776 EMTSRV          ;EMT TRAP, EMT DISPATCH SERVICE
552 000032 000340 340
553 000034 013304 LOGERR          ;TRAP TRAP, LOGIC ERROR TRAP
554 000036 000340 340
555          .=46
556 000046 001740 LOGICAL
557          .=52
558 000052 000000 0
559          .=170
560 000170 000000 SOFTSW: 0          ;SOFTWARE SWITCH VALUE
561 000172 000000 SOFTDI: 0          ;SOFTWARE DISPLAY VALUE
562 000174 000137 001426 JMP MONITR          ;PROGRAM 'RESTART' ADDRESS
563 000200 000137 001060 JMP INIT          ;INITIALIZATION ADDRESS
564 000204 000137 001764 JMP ADTEST          ;MANUAL START OF A TO D LOGIC
565 000210 000137 004050 JMP DMATST          ;MANUAL START OF DMA LOGIC
566 000214 000137 007504 JMP DSHTST          ;MANUAL START OF DUAL SAMPLE
567 000220 000137 006004 JMP MANCAL          ;MANUAL START OF THE A TO D CALIBRATION
568
569          ;TRAP EQUIVALENCE TABLE:
570
571          104400 ERROR=TRAP          ;LOGIC TEST ERROR ROUTINE
572          104000 PRINT=EMT          ;MESSAGE PRINTER ROUTINE
573          104001 DECOCT=EMT+1          ;OCTAL CONVERSIN ROUTINE
574          104002 SCOPE=EMT+2          ;LOGIC TEST SCOPE SUBROUTINE (4000)
575          104003 SCOPE1=EMT+3          ;LOGIC TEST SCOPE SUBROUTINE (10)
576          104004 CMPUTE=EMT+4          ;A/D AVERAGING ROUTINE
577          104005 CATORIZ=EMT+5          ;ROUTINE TO CALCULATE THE COUNT SPREAD
578          104006 BINDEC=EMT+6          ;BINARY TO DECIMAL CONVERSION ROUTINE
579          104007 SPACE=EMT+7          ;TYPE 'N' SPACES
580          104010 PRTOCT=EMT+10          ;OCTAL PRINT ROUTINE
581          104011 TTYIN=EMT+11          ;TELETYPE INPUT ROUTINE
582          104012 PRTAVG=EMT+12          ;GAIN AVERAGE PRINT ROUTINE
583          104013 TSTTKS=EMT+13          ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
584          104014 SCOPE0=EMT+14          ;SCOPE SUBROUTINE (1)
585
586          001000 .=1000
587 001000 170400 LPSADD: 170400          ;LPS STARTING DEVICE ADDRESS
588 001002 000340 LPSVCT: 340          ;LPS STARTING DEVICE VECTOR
589 001004 000300 ADBRL: 300          ;LPS A TO D BR LEVEL
590 001006 000001 POPDLY: 1          ;CHANGED BY THE PROGRAM
591 001010 000037 AMASK: 37          ;+ OR - 37 COUNTS FOR DUAL SAMPLE LOGIC TEST
592 001012 000002 FILLS: 2          ;NUMBER OF FILLER CHARS AFTER LF
593 001014 000000 FILCHR: 0          ;FILLER CHARACTER

```

```

594
595 001016 177776 PSW: 177776
596 001020 177560 TKS: 177560
597 001022 177562 TKB: 177562
598 001024 177564 TPS: 177564
599 001026 177566 TPB: 177566
600 001030 177570 SWR: 177570 ;OR 170 IF NO SWR CPU TYPE
601 001032 177570 DISPLA: 177570 ;OR 172 IF NO SWR CPU TYPE
602 001034 000000 PASSCT: 0
603 001036 004000 ICCUNT: 4000
604
605 001040 170400 ADCS: 170400 ;A TO D STATUS/CONTROL REGISTER
606 001042 170401 ADCS1: 170401 ;A TO D STATUS REGISTER (HIGH BYTE)
607 001044 170402 ADDBR: 170402 ;A TO D CONVERTED VALUE (READ)
608 ;A TO D LED DISPLAY LIGHTS (WRITE)
609 001046 170404 CSR: 170404 ;CLOCK STATUS REGISTER
610 001050 170406 CSB: 170406 ;CLOCK PRESET BUFFER
611
612 001052 170436 ADMR: 170436 ;A TO D DIRECT MEMORY ACCESS ADDRESS
613
614 001054 000340 ADINT: 340 ;A TO D INTERRUPT VECTOR
615 001056 000342 ADINT1: 342
616
617 ; THIS ROUTINE IS EXECUTED ON LOADING THE PROGRAM
618 ; OR RESTARTING AT LOCATION 174
619
620 001060 013706 015444 INIT: MOV STACK, SP ; INIT STACK POINTER=1000
621 001064 012777 000340 177724 MOV #340, @PSW
622 001072 000005 RESET ; CLEAR THE WORLD
623 001074 013737 001000 001040 MOV LPSADD, ADCS ; HOUSEKEEP THE ADDRESS AND VECTOR
624 001102 013737 001000 001042 MOV LPSADD, ADCS1
625 001110 013737 001000 001044 MOV LPSADD, ADDBR
626 001116 013737 001000 001046 MOV LPSADD, CSR
627 001124 013737 001000 001050 MOV LPSADD, CSB
628 001132 013737 001000 001052 MOV LPSADD, ADMR
629 001140 005237 001042 INC ADCS1
630 001144 062737 000002 001044 ADD #2, ADDBR
631 001152 062737 000004 001046 ADD #4, CSR
632 001160 062737 000006 001050 ADD #6, CSB
633 001166 062737 000006 001052 ADD #36, ADMR
634 001174 013737 001002 001054 MOV LPSVCT, ADINT
635 001202 013737 001002 001056 MOV LPSVCT, ADINT1
636 001210 062737 000002 001056 ADD #2, ADINT1
637
638 ; NOW TEST IF A SWITCH REGISTER EXISTS
639
640 001216 012737 001232 000004 MOV #15, @#4 ; LOAD BUS TRAP
641 001224 005777 177600 TST @SWR ; TEST IF REAL SWITCH REGISTER
642 001230 000407 BR CPUTYP ; BR IF NO TRAP
643 001232 022626 1$: CMP (SP)+, (SP)+ ; POP STACK
644 001234 012737 000170 001030 MOV #170, SWR ; CHANGE POINTER TO LOC. 170
645 001242 012737 000172 001032 MOV #172, DISPLA ; AND 172

```

```

001250 000002 000006 000006
001256 000006 000004 000004
001264 000003 000003 000003
001272 000261 177772 000261
001278 005600 005600 005600
001284 000261 177772 000261
001290 105737 177777 105737
001296 005600 005600 005600
001302 005037 177700 005037
001308 005037 000006 005037
001314 005037 000006 005037
001320 006300 012070 006300
001326 016037 012070 016037
001332 010037 001006 010037
001338 001002 001006 001002
001344 005237 001006 005237
001350 012702 000242 012702
001356 012701 000240 012701
001362 010221 000240 010221
001368 005021 000242 005021
001374 010102 001002 010102
001380 005722 001002 005722
001386 020227 001002 020227
001392 001371 000042 001371
001398 005737 000042 005737
001404 001402 001454 001402
001410 000137 001454 000137
001416 012777 000100 012777
001422 005037 015442 005037
001428 104000 015442 104000
001434 014012 015442 014012
001440 104000 015442 104000
001446 014133 015442 014133
001452 000413 015442 000413
001458 000005 015444 000005
001464 013706 000340 013706
001470 012777 000100 012777
001476 012777 000100 012777
001482 104000 177354 104000
001488 014432 177350 014432
001494 014432 011676 014432
001500 012737 000024 012737
001506 012737 000006 012737
001512 005037 000004 005037
001518 013777 001056 013777
001524 005077 177352 005077
001530 012737 001056 012737
001536 004537 015446 004537
001542 013664 013664 013664

```

```

: DETERMINE CPU TYPE
:
: RD=3 IF 11'45
: RD=2 IF 11'40
: RD=1 IF 11'20
: RD=0 IF 11'05
: PIRQ=177772
: PS=177776
CPUYTP: MOV #RTI,2#6
MOV #6,2#4
MOV #3,RC ;LOAD RC
SEC ;SET C BIT
TST #PIRQ ;TEST PIRQ
SBC RC
SEC
TSTB #PS+1
SBC RC
CLR #177700
CLR #6
ASL RC
MOV CPDLAY(RO),CPTIME ;GET TIME DELAY
MOV RO,POPDLY ;LOAD CPU DELAY
BNE IS
INC POPDLY ;UPDATE IT
: TRAP CATCHER
IS: MOV #242,R2 ;LOAD R2
MOV #240,R1 ;LOAD R1
SS: MOV R2,(R1)+ ;LOAD .+2
CLR (R1)+ ;LOAD HALT
MOV R1,R2 ;LOAD R2
TST (R2)+ ;BUMP R2
CMP R2,#1002 ;TEST FOR LAST
BNE SS ;BR UNTIL DONE
TST #42
BEG +6
JMP INIT2
MOV #100,3TKS ;ENABLE TTY INTERRUPTS
CLR PRINT!
PRINT ;CALL MESSAGE PRINTER VIA 'EMT'
TITLE ;TYPE PROGRAM HEADER.
INITA: PRINT ;PRINT THE TEST CALL LETTERS.
MES4 ;GO AND AWAIT COMMAND.
BR INIT2 ;OR A RESTART AT LOCATION '200'.
: MONITOR SUBROUTINE. ENTER VIA 'C' OR 'A'
MONTR: RESET ;INITIALIZE ON ENTRY
MOV STACK,SP ;RESET STACK POINTER
MOV #340,3PSW
MOV #100,3TKS ;ENABLE TTY INTERRUPTS
PRINT ;CALL MESSAGE PRINTER
CNTRLC ;TYPE 'C'
INIT2: MOV #PWFAL,2#24 ;SET UP POWER FAIL
MOV #6,2#4 ;SET UP BUSS ERROR
CLR #6
MOV ADINT1,3ADINT
CLR 3ADINT!
MOV #INITA,AVECTR ;SET UP 'IA' VECTOR ADDRESS.
JSR R5,LEDS

```

702	001520	000006		6		
703	001522	005737	000042	TST	2042	
704	001526	001402		BEQ	.+5	
705	001530	000137	001674	JMP	WHAT	
706	001534	012737	000100	MOV	#130,2TKS	:ENABLE KEYBOARD INTERRUPT
707	001542	104000		PRINT		
708	001544	014472		DOT		:PRINT '.' TO INDICATE MONITOR READY
709	001546	104011		TTYIN		:WAIT FOR TTY ENTRY
710	001550	042737	000040	BIC	#BITS,INBUF	:CLEAR LOWER CASE BIT
711	001556	122737	000101	CMPB	#'A,INBUF	:TEST FOR 'A'
712	001564	001002		BNE	.+6	:NOT 'A'
713	001566	000137	001764	JMP	ADTEST	:YES RUN 'A TO D LOGIC' TEST
714	001572	122737	000102	CMPB	#'B,INBUF	:TEST FOR 'B'
715	001600	001002		BNE	.+6	:NOT 'B'
716	001602	000137	004050	JMP	DMATST	:YES RUN 'A TO D DMA LOGIC TEST'
717	001606	122737	000103	CMPB	#'C,INBUF	:TEST FOR 'C'
718	001614	001002		BNE	.+6	:NOT 'C'
719	001616	000137	005742	JMP	CALBRT	:YES RUN 'A TO D CALIBRATION' TEST
720	001622	122737	000104	CMPB	#'D,INBUF	:TEST FOR 'D'
721	001630	001002		BNE	.+6	:NOT 'D'
722	001632	000137	006252	JMP	REPTST	:YES RUN 'A TO D REPEATABILITY' TEST
723	001636	122737	000105	CMPB	#'E,INBUF	:TEST FOR 'E'
724	001644	001002		BNE	.+6	:NOT 'E'
725	001646	000137	007070	JMP	RECVRY	:YES RUN 'A TO D RECOVERY TEST
726	001652	122737	000106	CMPB	#'F,INBUF	:TEST FOR 'F'
727	001660	001002		BNE	.+6	:NOT 'F'
728	001662	000137	007504	JMP	DSHTST	:YES RUN 'DUAL MODE' TEST
729	001666	104000		PRINT		:ILLEGAL ENTRY
730	001670	014475		QMARK		:TYPE '?'
731	001672	000670		BR	INIT2	:WAIT AGAIN
732	001674	005037	001762	CLR	NODMA	:CLEAR DMA POINTER
733	001700	012737	001754	MOV	#WHAT1,204	
734	001706	005077	177140	CLR	2ADMA	:TEST DMA OPTION
735	001712	000137	001770	JMP	ADTST1	:RUN A TO D TEST
736	001716	005737	001762	TST	NODMA	:TEST FOR DMA OPTION
737	001722	001002		BNE	.+6	
738	001724	000137	004054	JMP	DMAD1	
739	001730	000005		RESET		
740	001732	000005		RESET		
741	001734	013700	000042	MOV	2042,PC	
742	001740	004710		JSR	PC,(0)	
743	001742	000240		NOP		
744	001744	000240		NOP		
745	001746	000240		NOP		
746	001750	000137	001674	JMP	WHAT	
747	001754	005137	001762	COM	NODMA	
748	001760	000002		RTI		
749	001762	000000		C		

750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803

001764 104000
001766 014500
001770 000005
001772 005037 001034
001776 012737 002042 013600
002004 013777 001034 177020
002012 013706 015444
002016 005077 176774
002022 052777 000100 176770

002030 005077 177004
002034 005077 177004

002040 104003
002042 012777 000002 176770
002050 022777 000002 176762
002056 001401
002060 104400
002062 005077 176752
002066 005777 176746
002072 001401
002074 104400

002076 104002
002100 012777 000004 176732
002106 022777 000004 176724
002114 001401
002116 104400
002120 005077 176714
002124 005777 176710
002130 001401
002132 104400

002134 104002
002136 012777 000010 176674
002144 022777 000010 176666
002152 001401
002154 104400
002156 005077 176656
002162 005777 176652
002166 001401
002170 104400

```
*****  
: A TO D LOGIC TEST  
*****  
ADTEST: PRINT  
MESS  
ADTST1: RESET  
CLR PASSCT  
ADTST0: MOV #ADT1+2, RETURN  
MOV PASSCT, @DISPLA  
MOV STACK, SP  
CLR @PSW  
BIS #BIT6, @TKS  
  
:TEST FOR NO BUSS ERRORS  
CLR @ADCS  
CLR @ADDBR  
  
:DOES (BIT 1) SET AND CLEAR  
ADT1: SCOPE1  
MOV #BIT1, @ADCS :SET BIT 1  
CMP #BIT1, @ADCS  
BEQ .+4  
ERROR :ERROR, ADCS NOT = 2  
CLR @ADCS  
TST @ADCS  
BEQ .+4  
ERROR :ERROR, ADCS NOT = 0  
  
:DOES (BIT 2) SET AND CLEAR  
ADT2: SCOPE  
MOV #BIT2, @ADCS :SET BIT 2  
CMP #BIT2, @ADCS  
BEQ .+4  
ERROR :ERROR, ADCS NOT = 4  
CLR @ADCS  
TST @ADCS  
BEQ .+4  
ERROR :ERROR, ADCS NOT = 3  
  
:DOES BURST (BIT 3) SET AND CLEAR  
ADT3: SCOPE  
MOV #BIT3, @ADCS :SET BIT 3  
CMP #BIT3, @ADCS  
BEQ .+4  
ERROR :ERROR, ADCS NOT = 10  
CLR @ADCS  
TST @ADCS  
BEQ .+4  
ERROR :ERROR, ADCS NOT = 0
```

804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852

002172 104003
002174 012777 000020 176636
002202 022777 000020 176630
002210 001401
002212 104400
002214 005077 176620
002220 005777 176614
002224 001401
002226 104400

002230 104002
002232 012777 000040 176600
002240 022777 000040 176572
002246 001401
002250 104400
002252 005077 176562
002256 005777 176556
002262 001401
002264 104400

002266 104002
002270 012777 000100 176542
002276 022777 000100 176534
002304 001401
002306 104400
002310 005077 176524
002314 005777 176520
002320 001401
002322 104400

002324 104002
002326 012777 000400 176504
002334 022777 000400 176476
002342 001401
002344 104400
002346 005077 176466
002352 005777 176462
002356 001401
002360 104400

:DOES SCHMITT TRIGGER (BIT 4) SET AND CLEAR
ADT4: SCOPE1
MOV #BIT4,ADCS ;SET BIT 4
CMP #BIT4,ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 20
CLR ADCS ;CLEAR ADCS
TST ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 0

:DOES CLOCK OVERFLOW (BIT 5) SET AND CLEAR
ADT5: SCOPE
MOV #BIT5,ADCS ;SET BIT 5
CMP #BIT5,ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 40
CLR ADCS
TST ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 0

:DOES INTERRUPT ENABLE (BIT 6) SET AND CLEAR
ADT6: SCOPE
MOV #BIT6,ADCS ;SET BIT 6
CMP #BIT6,ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 100
CLR ADCS
TST ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 100

:DOES MUX CHANNEL (BIT 8) SET AND CLEAR
ADT7: SCOPE
MOV #BIT8,ADCS ;SET BIT 8
CMP #BIT8,ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 400
CLR ADCS
TST ADCS
BEQ .+4
ERROR ;ERROR ADCS NOT = 0

```

853                                     ;DOES MUX CHANNEL (BIT 9) SET AND CLEAR
854
855 002362 104002 ADT10: SCOPE
856 002364 012777 MOV #BIT9,ADCS ;SET BIT 9
857 002372 022777 001000 176446 CMP #BIT9,ADCS
858 002400 001401 BEQ .+4
859 002402 104400 ERROR ;ERROR ADCS NOT = 10000
860 002404 005077 176430 CLR ADCS
861 002410 005777 176424 TST ADCS
862 002414 001401 BEQ .+4
863 002416 104400 ERROR ;ERROR ADCS NOT = 0
864
865                                     ;DOES MUX CHANNEL (BIT 10) SET AND CLEAR
866
867 002420 104002 ADT11: SCOPE
868 002422 012777 MOV #BIT10,ADCS ;SET BIT 10
869 002430 022777 002000 176410 CMP #BIT10,ADCS
870 002436 001401 BEQ .+4
871 002440 104400 ERROR ;ERROR ADCS NOT 2000
872 002442 005077 176372 CLR ADCS
873 002446 005777 176366 TST ADCS
874 002452 001401 BEQ .+4
875 002454 104400 ERROR ;ERROR ADCS NOT = 0
876
877                                     ;DOES MUX CHANNEL (BIT 11) SET AND CLEAR
878
879 002456 104002 ADT12: SCOPE
880 002460 012777 MOV #BIT11,ADCS ;SET BIT 11
881 002466 022777 004000 176352 CMP #BIT11,ADCS
882 002474 001401 BEQ .+4
883 002476 104400 ERROR ;ERROR ADCST NOT = 4000
884 002500 005077 176334 CLR ADCS
885 002504 005777 176330 TST ADCS
886 002510 001401 BEQ .+4
887 002512 104400 ERROR ;ERROR ADCST NOT = 0
888
889                                     ;DOES MUX CHANNEL (BIT 12) SET AND CLEAR
890
891 002514 104002 ADT13: SCOPE
892 002516 012777 MOV #BIT12,ADCS
893 002524 022777 010000 176314 CMP #BIT12,ADCS
894 002532 001401 BEQ .+4
895 002534 104400 ERROR ;ERROR ADCS NOT = 10000
896 002536 005077 176276 CLR ADCS
897 002542 005777 176272 TST ADCS
898 002546 001401 BEQ .+4
899 002550 104400 ERROR ;ERROR ADCS NOT = 0
900
901
    
```

```

902                                     :DOES MUX CHANNEL (BIT 13) SET AND CLEAR
903
904 002552 104002 ADT14: SCOPE
905 002554 012777 020000 176256 MOV #BIT13,ADCS ;SET BIT 13
906 002562 022777 020000 176250 CMP #BIT13,ADCS
907 002570 001401 BEQ .+4
908 002572 104400 ERROR ;ERROR ADCS NOT = 20000
909 002574 005077 176240 CLR ADCS
910 002600 005777 176234 TST ADCS
911 002604 001401 BEQ .+4
912 002606 104400 ERROR ;ERROR ADCS NOT = 0
913
914                                     :DOES DUAL MODE (BIT14) SET AND CLEAR
915
916 002610 104002 ADT15: SCOPE
917 002612 012777 040000 176220 MOV #BIT14,ADCS ;SET BIT 14
918 002620 022777 040000 176212 CMP #BIT14,ADCS
919 002626 001401 BEQ .+4
920 002630 104400 ERROR ;ERROR, ADCS NOT 40000
921 002632 005077 176202 CLR ADCS ;CLEAR BIT 14
922 002636 005777 176176 TST ADCS
923 002642 001401 BEQ .+4
924 002644 104400 ERROR ;ERROR, ADCS NOT = 0
925
926                                     :DOES ERROR (BIT 15) SET AND CLEAR
927
928 002646 104002 ADT16: SCOPE
929 002650 012777 100000 176162 MOV #BIT15,ADCS
930 002656 022777 100000 176154 CMP #BIT15,ADCS
931 002664 001401 BEQ .+4
932 002666 104400 ERROR ;ERROR, ERROR BIT FAILED TO SET
933 002670 005077 176144 CLR ADCS
934 002674 005777 176140 TST ADCS
935 002700 001401 BEQ .+4
936 002702 104400 ERROR ;ERROR, ERROR BIT FAILED TO CLEAR
937
938                                     :DOES DONE (BIT 7) SET AND CLEAR
939
940 002704 104003 ADT20: SCOPE1
941 002706 017700 176132 MOV @ADDSR,R0
942 002712 005037 015464 CLR KSTOR4
943 002716 012777 000001 176114 MOV #BIT0,ADCS
944 002724 105777 176110 ADT20A: TSTB ADCS
945 002730 100404 BMI ADT20B
946 002732 005237 015464 INC KSTOR4
947 002736 001372 BNE ADT20A
948 002740 104400 ERROR ;ERROR, ADCS NOT = 200
949 002742 017700 176076 ADT20B: MOV @ADDSR,R0
950 002746 105777 176066 TSTB ADCS
951 002752 100001 BPL .+4
952 002754 104400 ERROR ;ERROR, DONE FAILED TO CLEAR
953 002756 032777 000001 176054 BIT #BIT0,ADCS
954 002764 001401 BEQ .+4
955 002766 104400 ERPOR ;ERROR, AD TO FAILED TO CLEAR

```

```

956
957
958
959 002770 104003
960 002772 017700 176046
961 002776 012777 000001 176034
962 003004 105777 176030
963 003010 100375
964 003012 013737 012100 015502
965 003020 006237 015502
966 003024 006237 015502
967 003030 006237 015502
968 003034 012777 000001 175776
969 003042 005337 015502
970 003046 100375
971 003050 005777 175764
972 003054 100401
973 003056 104400
974 003060 005077 175754
975 003064 013737 012100 015502
976 003072 005337 015502
977 003076 100375
978 003100 000240
979 003102 000240
980 003104 000240
981 003106 005777 175726
982 003112 100001
983 003114 104400
984
985
986
987 003116 104003
988 003120 005077 175714
989 003124 017700 175714
990 003130 052777 000020 175702
991 003136 005000
992 003140 005200
993 003142 001376
994 003144 105777 175670
995 003150 100001
996 003152 104400
997
998
999
1000 003154 104003
1001 003156 005077 175656
1002 003162 017700 175656
1003 003166 052777 000040 175644
1004 003174 005037 015464
1005 003200 005237 015464
1006 003204 001375
1007 003206 105777 175626
1008 003212 100001
1009 003214 104400

```

```

:DOES ERROR (BIT 15) SET
ADT21: SCOPE1
MOV @ADDBR,RO
MOV #BIT0,@ADCS
1$: TSTB @ADCS ;WAIT FOR FLAG
BPL 1$
MOV CPTIME,DELAY1
ASR DELAY1
ASR DELAY1
ASR DELAY1
MOV #BIT0,@ADCS
2$: DEC DELAY1
BPL 2$
TST @ADCS
BMI .+4
ERROR ;ERROR ADCS NOT = 100000
CLR @ADCS
MOV CPTIME,DELAY1
3$: DEC DELAY1
BPL 3$
NOP
NOP
NOP
TST @ADCS
BPL .+4
ERROR ;ERROR ADCS NOT = 0
:TEST THAT NO EXTERNAL CONVERSIONS INPUT
ADT22: SCOPE1
CLR @ADCS
MOV @ADDBR,RO
BIS #BIT4,@ADCS
CLR RO
INC RO
BNE -2
TSTB @ADCS
BPL .+4
ERROR ;ERROR EXTERNAL CONVERSION
:TEST THAT NO CLOCK OVERFLOWS START CONVERSIONS
ADT23: SCOPE1
CLR @ADCS
MOV @ADDBR,RO
BIS #BIT5,@ADCS
CLR KSTOR4
1$: INC KSTOR4
BNE 1$
TSTB @ADCS
BPL .+4
ERROR ;ERROR OVERFLOW STARTED A CONVERSION

```



```

1056 ;TEST THAT THE A TO D DOES NOT INTERRUPT AT LEVEL INDICATED +1
1057
1058 003444 104003 ADT26: SCOPE1
1059 003446 005077 175366 CLR @ADCS ;CLEAR A TO D STATUS
1060 003452 017700 175366 MOV @ADDBR,R0
1061 003456 012777 003516 175370 MOV #ADT26A,@ADINT
1062 003464 013777 015530 175324 MOV BRLEV3,@PSW ;BR LEVEL +1
1063 003472 052777 000101 175340 BIS #BIT6!BIT0,@ADCS ;SET THE AD GO AND INTERRUPT ENABLE BIT
1064 003500 012737 001000 015464 MOV #1000,KSTOR4
1065 003506 005337 015464 1$: DEC KSTOR4 ;DELAY
1066 003512 001375 BNE 1$
1067 003514 000401 BR ADT26B ;NO INTERRUPT
1068 003516 104400 ADT26A: ERROR ;ERROR, A TO D INTERRUPTED IN ERROR
1069 003520 013777 001056 175326 ADT26B: MOV ADINT1,@ADINT
1070 003526 005077 175324 CLR @ADINT1
1071
1072 ;TEST THAT RESET CLEARS MUX BITS
1073
1074 003532 104003 ADT27: SCOPE1
1075 003534 012777 037400 175276 MOV #BIT13!BIT12!BIT11!BIT10!BIT9!BIT8,@ADCS
1076 003542 000005 RESET
1077 003544 005777 175270 TST @ADCS
1078 003550 001401 BEQ .+4
1079 003552 104400 ERROR ;ERROR, RESET FAILED TO CLEAR MUX BITS
1080
1081 ;TEST THAT RESET CLEARS ST, CLOCK OVERFLOW AND INTERRUPT ENABLE BITS
1082
1083 003554 104003 ADT30: SCOPE1
1084 003556 012777 000160 175254 MOV #BIT6!BIT5!BIT4,@ADCS
1085 003564 000005 RESET
1086 003566 005777 175246 TST @ADCS
1087 003572 001401 BEQ .+4
1088 003574 104400 ERROR ;ERROR, RESET FAILED TO CLEAR ENABLES
1089
1090 ;TEST THAT RESET CLEARS DONE AND ERROR BITS
1091
1092 003576 104003 ADT31: SCOPE1
1093 003600 012777 000001 175232 MOV #BIT0,@ADCS
1094 003606 105777 175226 TSTB @ADCS
1095 003612 100375 BPL .-4
1096 003614 012777 100000 175216 MOV #BIT15,@ADCS ;SET ERROR BIT
1097 003622 000005 RESET
1098 003624 005777 175210 TST @ADCS
1099 003630 001401 BEQ .+4
1100 003632 104400 ERROR ;ERROR, RESET FAILED TO CLEAR ERROR OR DONE FLAG

```

```

1101
1102           ;TEST THAT RESET CLEARS BURST AND MODE BITS
1103
1104 003634 104003 ADT32: SCOPE1
1105 003636 012777 000016 175174 MOV #BIT3!BIT2!BIT1,ADCS ;SET BITS 1-3
1106 003644 000005 RESET
1107 003646 005777 175166 TST ADCS
1108 003652 001401 BEQ .+4
1109 003654 104400 ERROR ;ERROR, ADCS NOT = 0
1110
1111
1112           ;LED TEST
1113
1114 003656 104003 TEST15: SCOPE1
1115 003660 005737 000042 TST #42
1116 003664 001054 BNE TEST16
1117 003666 005737 001034 TST PASSCT ;TEST IF FIRST PASS
1118 003672 001451 BEQ TEST16 ;BR IF YES
1119 003674 005077 175116 CLR ADPSW
1120 003700 052777 000100 175112 BIS #BIT6,ADTKS
1121 003706 005037 015516 CLR TEMP1
1122 003712 013737 015516 015520 TST15A: MOV TEMP1,TEMP2
1123 003720 012737 000006 015522 MOV #6,TEMP3
1124 003726 013777 015520 175110 TST15B: MOV TEMP2,ADDBR
1125 003734 013737 001006 015462 MOV POPDLY,KSTOR3
1126 003742 012700 000000 MOV #0,RO
1127 003746 032777 004000 175054 BIT #BIT11,ADSWR ;TEST IF NO INTERATIONS
1128 003754 001020 BNE TEST16 ;BR IF NO INTERATIONS
1129 003756 005300 TST15C: DEC RO
1130 003760 001376 BNE TST15C
1131 003762 005337 015462 DEC KSTOR3
1132 003766 001373 BNE TST15C
1133 003770 105237 015521 INCB TEMP2+1
1134 003774 005337 015522 DEC TEMP3
1135 004000 001352 BNE TST15B
1136 004002 005237 015516 INC TEMP1
1137 004006 022737 000040 015516 CMP #40,TEMP1
1138 004014 001336 BNE TST15A
1139
1140 004016 104014 TEST16: SCOPED ;LOGICAL END OF A TO D LOGIC TEST
1141 004020 000005 RESET
1142 004022 005737 000042 TST #42
1143 004026 001402 BEQ .+6
1144 004030 000137 001716 JMP WHATA
1145 004034 004737 013656 JSR PC,BELL ;REPORT END OF PASS
1146 004040 005237 001034 INC PASSCT
1147 004044 000137 001776 JMP ADTSTO

```



```

1148
1149
1150 ;*****
1151 ; DMA LOGIC TEST
1152 ;*****
1152 004050 104000 DMATST: PRINT
1153 004052 014653 MES11
1154 004054 000005 DMA01: RESET
1155 004056 005037 001034 CLR PASSCT
1156
1157 004062 013706 015444 DMA00: MOV STACK,SP
1158 004066 052777 000100 174724 BIS #BIT6,@TKS
1159 004074 305077 174716 CLR @PSW
1160 004100 013777 001034 174724 MOV PASSCT,@DISPLA
1161 004106 012737 004122 013600 MOV #DMAT0+2,RETURN
1162
1163 ;TEST FOR NO BUSS ERRORS
1164
1165 004114 005077 174732 CLR @ADMR
1166
1167 ;TEST THAT CURRENT ADDRESS REGISTER (BITS 1-15) SET AND CLEAR
1168
1169 004120 104002 DMAT0: SCOPE
1170 004122 005077 174712 CLR @ADCS ;CLEAR AD STATUS
1171 004126 012777 000006 174704 MOV #6,@ADCS ;SET UP CA POINTER
1172 004134 012777 000000 174710 MOV #0,@ADMR ;LOAD CURRENT ADDRESS
1173 004142 022777 000000 174702 CMP #0,@ADMR ;
1174 004150 001401 BEQ .+4 ;
1175 004152 104400 ERROR ;ERROR, CURRENT ADDRESS NOT = 0
1176
1177 ;TEST THAT CURRENT ADDRESS REGISTER (BITS 1-15) SET AND CLEAR
1178
1179 004154 104002 DMAT1: SCOPE
1180 004156 005077 174656 CLR @ADCS ;CLEAR AD STATUS
1181 004162 012777 000006 174650 MOV #6,@ADCS ;SET UP CA POINTER
1182 004170 012777 177776 174654 MOV #177776,@ADMR ;LOAD CURRENT ADDRESS
1183 004176 022777 177776 174646 CMP #177776,@ADMR ;
1184 004204 001401 BEQ .+4 ;
1185 004206 104400 ERROR ;ERROR, CURRENT ADDRESS NOT = 177776
1186
1187 ;TEST THAT CURRENT ADDRESS REGISTER (BITS 1-15) SET AND CLEAR
1188
1189 004210 104002 DMAT2: SCOPE
1190 004212 005077 174622 CLR @ADCS ;CLEAR AD STATUS
1191 004216 012777 000006 174614 MOV #6,@ADCS ;SET UP CA POINTER
1192 004224 012777 125252 174620 MOV #125252,@ADMR ;LOAD CURRENT ADDRESS
1193 004232 022777 125252 174612 CMP #125252,@ADMR ;
1194 004240 001401 BEQ .+4 ;
1195 004242 104400 ERROR ;ERROR, CURRENT ADDRESS NOT = 125252
1196

```

```

1197
1198 ;TEST THAT CURRENT ADDRESS REGISTER (BITS 1-15) SET AND CLEAR
1199
1200 004244 104002 DMAT3: SCOPE
1201 004246 005077 174566 CLR @ADCS ;CLEAR AD STATUS
1202 004252 012777 000006 174560 MOV #6,@ADCS ;SET UP CA POINTER
1203 004260 012777 052524 174564 MOV #52524,@ADMR ;LOAD CURRENT ADDRESS
1204 004266 022777 052524 174556 CMP #52524,@ADMR ;
1205 004274 001401 BEQ .+4 ;
1206 004276 104400 ERROR ;ERROR, CURRENT ADDRESS NOT = 52525
1207
1208 ;TEST THAT WORD COUNT REGISTER (BITS 0-11) CAN BE SET AND CLEARED
1209
1210 004300 104002 DMAT4: SCOPE
1211 004302 005077 174532 CLR @ADCS ;CLEAR AD STATUS
1212 004306 012777 000004 174524 MOV #4,@ADCS ;LOAD WC POINTER
1213 004314 012777 000000 174530 MOV #0,@ADMR ;LOAD WORD COUNT REGISTER
1214 004322 022777 000000 174522 CMP #0,@ADMR ;
1215 004330 001401 BEQ .+4 ;
1216 004332 104400 ERROR ;ERROR, WORD COUNT NOT = 0
1217
1218 ;TEST THAT WORD COUNT REGISTER (BITS 0-11) CAN BE SET AND CLEARED
1219
1220 004334 104002 DMAT5: SCOPE
1221 004336 005077 174476 CLR @ADCS ;CLEAR AD STATUS
1222 004342 012777 000004 174470 MOV #4,@ADCS ;LOAD WC POINTER
1223 004350 012777 177777 174474 MOV #-1,@ADMR ;LOAD WORD COUNT REGISTER
1224 004356 022777 007777 174466 CMP #7777,@ADMR ;
1225 004364 001401 BEQ .+4 ;
1226 004366 104400 ERROR ;ERROR, WORD COUNT NOT = 7777
1227
1228 ;TEST THAT WORD COUNT REGISTER (BITS 0-11) CAN BE SET AND CLEARED
1229
1230 004370 104002 DMAT6: SCOPE
1231 004372 005077 174442 CLR @ADCS ;CLEAR AD STATUS
1232 004376 012777 000004 174434 MOV #4,@ADCS ;LOAD WC POINTER
1233 004404 012777 125252 174440 MOV #125252,@ADMR ;LOAD WORD COUNT REGISTER
1234 004412 022777 005252 174432 CMP #5252,@ADMR ;
1235 004420 001401 BEQ .+4 ;
1236 004422 104400 ERROR ;ERROR, WORD COUNT NOT = 5252
1237
1238 ;TEST THAT WORD COUNT REGISTER (BITS 0-11) CAN BE SET AND CLEARED
1239
1240 004424 104002 DMAT7: SCOPE
1241 004426 005077 174406 CLR @ADCS ;CLEAR AD STATUS
1242 004432 012777 000004 174400 MOV #4,@ADCS ;LOAD WC POINTER
1243 004440 012777 052525 174404 MOV #52525,@ADMR ;LOAD WORD COUNT REGISTER
1244 004446 022777 002525 174376 CMP #2525,@ADMR ;
1245 004454 001401 BEQ .+4 ;
1246 004456 104400 ERROR ;ERROR, WORD COUNT NOT = 2525
1247
1248

```

```

1249                                     ;TEST THAT DMA STATUS (BIT 0) CAN BE SET AND CLEARED
1250
1251 004460 104002                      DMAT10: SCOPE
1252 004462 005077 174352                CLR      @ADCS           ;CLEAR AD STATUS
1253 004466 012777 000002 174344        MOV      #2,@ADCS       ;POINTER TO DMA MODE STATUS
1254 004474 012777 010000 174350        MOV      @BIT12,@ADMR   ;LOAD ENABLE
1255 004502 022777 010000 174342        CMP      @BIT12,@ADMR   ;
1256 004510 001401                      BEQ      .+4
1257 004512 104400                      ERROR                    ;ERROR, DMA STATUS NOT = 10000
1258
1259                                     ;TEST THAT DMA STATUS (BIT 1) CAN BE SET AND CLEARED
1260
1261 004514 104002                      DMAT11: SCOPE
1262 004516 005077 174316                CLR      @ADCS
1263 004522 012777 000002 174310        MOV      #2,@ADCS
1264 004530 012777 020000 174314        MOV      @BIT13,@ADMR  ;LOAD EXT 1
1265 004535 022777 020000 174306        CMP      @BIT13,@ADMR
1266 004544 001401                      BEQ      .+4
1267 004546 104400                      ERROR                    ;ERROR, DMA STATUS NOT = 20000
1268
1269                                     ;TEST THAT DMA STATUS (BIT 2) CAN SET AND CLEAR
1270
1271 004550 104002                      DMAT12: SCOPE
1272 004552 005077 174262                CLR      @ADCS
1273 004556 012777 000002 174254        MOV      #2,@ADCS
1274 004564 012777 040000 174260        MOV      @BIT14,@ADMR  ;LOAD EXT 2
1275 004572 022777 040000 174252        CMP      @BIT14,@ADMR
1276 004600 001401                      BEQ      .+4
1277 004602 104400                      ERROR                    ;ERROR, DMA STATUS NOT = 40000
1278
1279                                     ;TEST THAT RESET CLEARS CURRENT ADDRESS REGISTER
1280
1281 004604 104002                      DMAT13: SCOPE
1282 004606 005077 174226                CLR      @ADCS
1283 004612 012777 000006 174220        MOV      #6,@ADCS
1284 004620 012777 177777 174224        MOV      #-1,@ADMR     ;SET CA = -1
1285 004626 000005                      RESET
1286 004630 012777 000006 174202        MOV      #6,@ADCS
1287 004636 005777 174210                TST      @ADMR
1288 004642 001401                      BEQ      .+4
1289 004644 104400                      ERROR                    ;ERROR, RESET FAILED TO CLEAR CURRENT ADDRESS
1290
1291                                     ;TEST THAT RESET CLEARS WORD COUNT REGISTER
1292
1293 004646 104003                      DMAT14: SCOPE1
1294 004650 005077 174164                CLR      @ADCS
1295 004654 012777 000004 174156        MOV      #4,@ADCS
1296 004662 012777 177777 174162        MOV      #-1,@ADMR     ;SET WC = -1
1297 004670 000005                      RESET
1298 004672 012777 000004 174140        MOV      #4,@ADCS
1299 004700 005777 174146                TST      @ADMR
1300 004704 001401                      BEQ      .+4
1301 004706 104400                      ERROR                    ;ERROR, RESET FAILED TO CLEAR WORD COUNT REGISTER
1302

```

120
110
100
90
80
70
60
50
40
30
20
10
0

004710 104003
004712 005077 174123
004714 012777 000100 174114
004716 012777 174120 174120
004718 000006
004720 012777 000002 174076
004722 005077 174104
004724 001401
004726 104400

004752 104003
004754 005077 174060
004756 012777 000100 174052
004758 032777 000100 174044
004760 001001
004762 104400

005000 052777 000002 174032
005002 012777 010000 174036
005004 032777 000100 174016
005006 001401
005008 104400

005026 005077 174020

:TEST THAT RESET CLEARS DMA STATUS REGISTER

DMAT15: SCOPE1
CLR 2ADCS
MOV #2,2ADCS
MOV #-1,2ADMR
RESET
MOV #2,2ADCS
TST 2ADMR
BEQ .+4
ERROR

:ERROR. RESET FAILED TO CLEAR DMA STATUS

:TEST THAT THE DMA ENABLE CLEARS INTERRUPT ENABLE

DMAT16: SCOPE1
CLR 2ADCS
MOV #BIT6,2ADCS
BIT #BIT6,2ADCS
BNE .+4
ERROR

:CLEAR
:ENABLE INTERRUPT

:ERROR. BIT 6 FAILED TO SET

BIS #BIT1,2ADCS
MOV #BIT12,2ADMR
BIT #BIT6,2ADCS
BEQ .+4
ERROR

:SET DMA POINTER
:SET DMA ENABLE
:TEST BIT 6

:ERROR. SETTING DMA ENABLE FAILED
: TO CLEAR INTERRUPT ENABLE

CLR 2ADMR

```

1332
1333      :TEST THAT THE DMA ACCESS A CORRECT MEMORY LOCATION
1334      : AND CAN INTERRUPT
1335
1336      005032 104003      DMAT17: SCOPE1
1337      005034 012777 000340 173754      MOV      #340, @PSW
1338      005042 012777 000100 173750      MOV      @BIT6, @TKS
1339      005050 012777 005264 173776      MOV      @DMT17A, @ADINT
1340      005056 005077 173734      CLR      @PSW
1341      005062 012737 177777 005734      MOV      #-1, @MKO
1342      005070 005077 173744      CLR      @ADCS      :CLEAR AD STATUS
1343      005074 012777 000006 173736      MOV      #6, @ADCS  :LOAD CA POINTER
1344      005102 012777 015654 173742      MOV      @ADBUFF, @ADMR :LOAD CURRENT ADDRESS
1345      005110 012777 000004 173722      MOV      #4, @ADCS  :LOAD WC POINTER
1346      005116 012777 177777 173726      MOV      #-1, @ADMR
1347      005124 012777 000002 173706      MOV      #2, @ADCS  :LOAD DMA STATUS POINTER
1348      005132 012777 010000 173712      MOV      @BIT12, @ADMR :LOAD DMA ENABLE
1349      005140 012777 000001 173672      MOV      #1, @ADCS  :START A TO D CONVERSION
1350      005146 012737 002000 015464      MOV      #2000, KSTOR4 :SET UP A DELAY
1351      005154 012777 000000 173634      MOV      #0, @PSW
1352      005162 005337 015464      IS:      DEC      KSTOR4
1353      005166 001375      BNE      IS      :DELAY
1354      005170 012777 000006 173642      MOV      #6, @ADCS  :LOAD CA POINTER
1355      005176 012737 015656 015514      MOV      @ADBUFF+2, TEMP :SET UP COMPARE
1356      005204 023777 015514 173640      CMP      TEMP, @ADMR
1357      005212 001401      BEQ      .+4
1358      005214 104400      ERROR      :ERROR, CURRENT ADDRESS FAILED TO INCREMENT
1359      : BY 2
1360      005216 012777 000004 173614      MOV      #4, @ADCS  :LOAD WC POINTER
1361      005224 005777 173622      TST      @ADMR      :TEST WORD COUNT
1362      005230 001401      BEQ      .+4
1363      005232 104400      ERROR      :ERROR, WORD COUNT FAILED TO INCREMENT BY 1
1364
1365      005234 012777 000002 173576      MOV      #2, @ADCS  :LOAD DMA STATUS POINTER
1366      005242 005777 173604      TST      @ADMR      :TEST DMA STATUS
1367      005246 001401      BEQ      .+4
1368      005250 104400      ERROR      :ERROR, DMA STATUS FAILED TO CLEAR
1369
1370      005252 005737 005734      TST      @MKO
1371      005256 001401      BEQ      .+4
1372      005250 104400      ERROR      :ERROR, DMA FAILED TO INTERRUPT
1373      005262 000403      BR      DMAT20
1374
1375      005264 005037 005734      DMT17A: CLR      @MKO
1376      005270 000002      RTI      :EXIT

```

1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415

005272 104003
005274 012777 000340 173514
005302 013706 015444
005306 013777 001056 173540
005314 005077 173536
005320 012737 000777 005734
005326 012737 007001 005736
005334 012737 015656 005740
005342 005077 173472
005346 012777 000006 173464
005354 012777 015654 173470
005362 012777 000004 173450
005370 012777 177000 173454
005376 012777 000002 173434
005404 012777 010000 173440
005412 012777 000001 173420
005420 012737 000400 015464
005426 005337 015464
005432 001375
005434 012777 000006 173376
005442 013737 005740 015514
005450 023777 015514 173374
005456 001401
005460 104400
005462 013737 005736 015514
005470 012777 000004 173342
005476 023777 015514 173346
005504 001402
005506 104400
005510 000410
005512 062737 000002 005740
005520 005237 005736
005524 005337 005734
005530 001330

:TEST THAT THE DMA CAN INCREMENT THRU MEMORY

DMAT20: SCOPE1

MOV #340, @PSW
MOV STACK, SP
MOV @DINT1, @ADINT
CLR @ADINT1
MOV #777, DMK0 :SET UP COUNTER
MOV #7001, DMK1
MOV @ADBUFF+2, DMK2 :SET UP POINTER
CLR @ADCS :CLEAR STATUS
MOV #6, @ADCS :SET UP CA POINTER
MOV @ADBUFF, @ADMR :LOAD CURRENT ADDRESS BUFFER
MOV #4, @ADCS :LOAD WC POINTER
MOV #-1000, @ADMR :LOAD WORD COUNT REGISTER
MOV #2, @ADCS :LOAD STATUS POINTER
MOV @BIT12, @ADMR :LOAD DMA ENABLE
DMT20A: MOV #1, @ADCS :LOAD A TO D GO
MOV #400, KSTOR4
IS: DEC KSTOR4 :DELAY
BNE IS
MOV #6, @ADCS :LOAD CA POINTER
MOV DMK2, TEMP :SET UP COMPARE
CMP TEMP, @ADMR :TEST
BEQ .+4 :ERROR, CURRENT ADDRESS FAILED TO INCREMENT
ERROR : BY 2 PROPERLY
MOV DMK1, TEMP :SET UP COMPARE
MOV #4, @ADCS :LOAD WC POINTER
CMP TEMP, @ADMR :TEST
BEQ .+6 :ERROR, WORD COUNT FAILED TO INCREMENT
ERROR : BY 1 PROPERLY
BR DMAT21 :UPDATE CA
ADD #2, DMK2
INC DMK1
DEC DMK0
BNE DMT20A :MORE TEST

```

1416
1417
1418
1419
1420 005532 104003
1421 005534 012777 000340 173254
1422 005542 012777 000100 173250
1423 005550 012737 177777 005734
1424 005556 005077 173256
1425 005562 012777 000006 173250
1426 005570 012777 160000 173254
1427 005576 012777 000004 173234
1428 005604 012777 177777 173240
1429 005612 012777 000002 173220
1430 005620 012777 070000 173224
1431 005626 012777 000001 173204
1432 005634 012737 002000 015464
1433 005642 005337 015464
1434 005646 001375
1435 005650 012777 000002 173162
1436 005656 005777 173170
1437 005662 100401
1438 005664 104400
1439
1440 005666 005077 173160
1441 005672 005777 173154
1442 005676 001401
1443 005700 104400
1444
1445 005702 104003
1446 005704 000005
1447 005706 005737 000042
1448 005712 001402
1449 005714 000137 001730
1450 005720 004737 013656
1451 005724 005237 001034
1452 005730 000137 004062
1453
1454 005734 000000
1455 005736 000000
1456 005740 000000

```

:TEST THAT THE DMA ERROR BIT (BIT 15) CAN BE SET

DMAT21: SCOPE1

15:

DMK0: 0
DMK1: 0
DMK2: 0

```

MOV #340, @PSW
MOV #BIT6, @TKS
MOV #-1, DMK0
CLR @ADCS ;CLEAR AD STATUS
MOV #6, @ADCS ;LOAD CA POINTER
MOV #160000, @ADMR ;LOAD NON-EXISTENT MEMORY LOCATION
MOV #4, @ADCS ;LOAD WC POINTER
MOV #-1, @ADMR ;LOAD 1 WRD
MOV #2, @ADCS ;LOAD DMA STATUS POINTER
MOV #70000, @ADMR ;LOAD DMA ENABLE
MOV #1, @ADCS ;START CONVERSION
MOV #2000, KSTOR4
DEC KSTOR4
BNE 15
MOV #2, @ADCS ;LOAD DMA STATUS POINTER
TST @ADMR ;TEST BIT 15
BMI .+4 ;BRANCH IF SET
ERROR ;ERROR, DMA ERROR BIT FAILED TO SET
CLR @ADMR ;CLEAR DMA STATUS
TST @ADMR ;TEST IT
BEQ .+4
ERROR ;ERROR, DMA STATUS FAILED TO CLEAR
SCOPE1
RESET
TST @#42
BEQ .+6
JMP WHATB
JSR PC, BELL ;REPORT END OF PASS
INC PASSCT
JMP DMA00

```

:LOGICAL END OF DMA LOGIC TEST

:REPORT END OF PASS

```

1457 :*****
1458 :CALIBRATION ROUTINE
1459 :*****
1460
1461 :ROUTINE REQUESTS THE TYPE OF 'SYNC' TO BE USED ('I' INTERNAL OR 'E' EXTERNAL).
1462 :OR 'C' LPS CLOCK)
1463 :THE PROGRAM THEN TAKES CONTINUOUS CONVERSIONS USING DATA SW'S 5-0
1464 :TO SELECT THE CH.
1465 :OR SW '10' TO PRINT THE CONVERSION VALUE.
1466
1467 005742 012737 005754 015446 CALBRT: MOV #CALBT1.AVECTR :SET UP '1A' RESTART ADDRESS
1468 005750 104000 PRINT
1469 005752 014527 MES7 ;TYPE TEST HEADER
1470 005754 104000 CALBT1: PRINT
1471 005756 014606 MES10 ;TEXT 'SYNC I OR E OR C'
1472 005760 104011 CALB1A: TTYIN ;WAIT FOR INPUT.
1473 005762 042737 000040 011032 BIC #BITS.INBUF
1474 005770 013737 011032 015450 MOV INBUF,PROC ;SAVE IT IN TEMP STORAGE
1475 005776 104000 PRINT
1476 006000 014470 CRLF
1477 006002 000402 BR .+6
1478 006004 005037 015450 MANCAL: CLR PROC
1479 006010 005037 01550F CLR USEDMA ;CLEAR DMA FLAG
1480 006014 005037 015511 CLR USECLK ;CLEAR CLOCK FLAG
1481 006020 005037 0155C4 CLR USED5H ;CLEAR DUAL MODE
1482 006024 012737 000001 015454 MOV #1,COUNT ;SET UP FOR '1' CONVERSION
1483 006032 117737 172772 015441 CALBT2: MOVB @SWR,ADWRD2+1 ;GET CH. FROM THE SW REG.
1484 006040 042737 140377 015440 BIC #140377,ADWRD2 ;CLR UNWANTED BITS, A/D WORD COMPLETE
1485 006046 017737 172756 015460 MOV @SWR,KSTOR2 ;SAVE ORIGINAL SWITCH SETTING.
1486 006054 022737 000105 015450 CMP #105,PROC ;TEST SYNC SELECT
1487 006062 021004 BNE CALB2 ;BRANCH IF NOT 'E'
1488 006064 052737 000020 015440 BIS #20,ADWRD2 ;OTHERWISE ADD 'EXT' SYNC BIT TO A/D.
1489 006072 000416 BR CALB2A
1490 006074 122737 000103 015450 CALB2: CMPB #103,PROC
1491 006102 001007 BNE CALB2C ;BRANCH IF NOT 'C'
1492 006104 052737 000040 015440 BIS #40,ADWRD2 ;OTHERWISE ADD 'CLOCK' SYNC TO A/D
1493 006112 012737 177777 015510 MOV #-1,USECLK
1494 006120 000403 BR CALB2A
1495 006122 052737 000001 015440 CALB2C: BIS #1,ADWRD2
1496 006130 005737 015510 CALB2A: TST USECLK ;CHECK CLOCK FLAG
1497 006134 001410 BEQ CALB2D ;NOT LPS CLOCK
1498 006136 005077 172704 CLR @CSR ;USE CLOCK, CLEAR CLOCK STATUS
1499 006142 012777 176000 172700 MOV #-2000,@CSB ;SET UP PRESET BUFFER
1500 006150 012777 000403 172670 MOV #403,@CSR ;SET UP RATE, START CLOCK
1501 006156 013706 015444 CALB2D: MOV STACK,SP
1502 006162 004737 011212 JSR PC,ADCVT ;TAKE AND STORE THE CONVERSIONS
1503 006166 012737 000001 015522 MOV #1,TEMP3 ;SETUP TO PRINT '1' VALUE
1504 006174 004537 013664 JSR RS,LEDS
1505 006200 015654 ADBUFF
1506 006202 032777 002000 172620 BIT #SW10,@SWR
1507 006210 001404 BEQ CALB2B
1508 006212 104012 PRTAVG ;PRINT IT
1509 006214 104000 PRINT
1510 006216 014470 CRLF
1511 006220 000406 BR CALBT4 ;TEST FOR LOOP
1512 006222 013700 015654 CALB2B: MOV ADBUFF,RO ;SET UP A-D BUFFER.

```



```

1513 006226 010077 172600      MOV      RD,DISPLA      ;LOAD DISPLAY REG. (11/45)
1514 006232 104013              TSTTKS                  ;TEST FOR TTY FLAG
1515 006234 000240      ARESET: NOP
1516 006236 104013      CALBT4: TSTTKS          ;TEST FOR KEYBOARD INTERRUPT
1517 006240 023777 015450 172562    CMP      KSTOR2,DSWR    ;TEST IF SWITCH REGISTER HAS CHANGED
1518 006246 001730      BEQ      CALB2A        ;BRANCH AND TAKE NEXT CONVERSION
1519 006250 000670      BR      CALBT2        ;YES, COMPUTE NEW INPUT
1520
1521 :*****
1522 :REPEATABILITY TEST
1523 :*****
1524
1525 ;THIS ROUTINE IS DESIGNED TO SHOW REPEATABILITY BY TAKING A SERIES OF
1526 ;'512' CONVERSIONS, AVEPAGING THEM AND THEN CATORIZING
1527 ;THEM IN BINS FROM THE AVERAGE PLUS & MINUS 6 COUNTS. THE ROUTINE
1528 ;REQUESTS FOR A CHANNEL OR CHANNELS, A COUNT SPREAD AND A A/D MODE TO BE TYPED
1529 ;IN VIA THE OPERATOR. A CONTINUOUS SERIES OF CONVERSIONS ARE THEN TAKEN
1530 ;AND COMPARED AGAINST THE INPUT COUNT SPREAD. IF ALL '512' CONVERSIONS
1531 ;ARE FOUND TO BE WITHIN THE SPREAD THE NEXT CH. IS EXERCISED OTHERWISE
1532 ;THE COUNTS ARE TYPED OUT. SETTING SWITCH '10' TO A '1' WILL FORCE A PRINTCUT
1533 ;OF THE CH (S).
1534 006252 012737 006264 015446    REPTST: MOV      #REPT1,AVECTR ;SET UP CNTR 'A' VECTOR ADDRESS
1535 006260 104000      PRINT
1536 006262 014726      MES13                  ;TEXT 'REPEATABILITY TEST'
1537 006264 005037 015532    REPT1:  CLR      MESPRT
1538 006270 005037 015505      CLR      USEDMA
1539 006274 005037 015504      CLR      USED5H
1540 006300 005037 015512      CLR      OPS1
1541 006304 104000      PRINT
1542 006306 014756      MES14                  ;REQUEST CHANNEL (S)
1543 006310 104011      TTYIN                  ;WAIT FOR INPUT
1544 006312 104001      DECOCT                 ;CONVERT TO OCTAL
1545 006314 013737 011202 015456    MOV      BCDTAB,KSTOR1 ;SAVE AS INTIAL CH.
1546 006322 013737 015456 015460    MOV      KSTOR1,KSTOR2 ;ALSO SAVE AS 2ND CH. ENTRY
1547 006330 005737 011204      TST      BCDTAB+2      ;TEST FOR SECOND ENTRY
1548 006334 001407      BEQ      REPT2         ;BRANCH IF NO SECOND ENTRY
1549 006336 023737 011204 015456    CMP      BCDTAB+2,KSTOR1 ;COMPARE ENTRY 1 TO ENTRY 2
1550 006344 100747      BMI     REPT1         ;BRANCH AND RESTART IF ILLEGAL
1551 006346 013737 011204 015460    MOV      BCDTAB+2,KSTOR2 ;OTHERWISE SAVE AS SECOND CH.
1552 006354 104000      REPT2: PRINT
1553 006356 015016      MES16                  ;TEXT 'COUNT SPREAD '
1554 006360 104011      TTYIN                  ;WAIT FOR ENTRY
1555 006362 104001      DECOCT                 ;DECODE TO OCTAL
1556 006364 013737 011202 015462    MOV      BCDTAB,KSTOR3 ;SAVE IT
1557 006372 104000      PRINT
1558 006374 015035      MES18
1559 006376 104011      TTYIN
1560 006400 122737 000102 011032    CMPB    #'B,INBUF      ;REQUEST MODE
1561 006406 001003      BNE     REPT2B        ;TEST FOR B
1562 006410 005137 015506      COM
1563 006414 000411      BR      REPT2A        ;NOT
1564 006416 122737 000104 011032    REPT2B: CMPB    #'D,INBUF ;TEST FOR DUAL MODE
1565 006424 001005      BNE     REPT2A        ;BRANCH IF NOT
1566 006426 005137 015504      COM      USED5H        ;SET DUAL MODE
1567 006432 052737 000100 015456    BIS     #BIT6,KSTOR1   ; BIT
1568 006440 013706 015444    REPT2A: MOV      STACK,SP

```

H03

LPS-11 DIAGNOSTIC TEST I MAINDEC-11-DZLPC-C
 DZLPC.P11 04-APR-76 00:00

MACY11 27(1006) 16-SEP-76 21:50 PAGE 33

1569	006444	013737	015456	015464		MOV	KSTOR1,KSTOR4	;SAVE STARTING CH.
1570	006452	104013			REPT3:	TSTTKS		;TEST FOR KEYBOARD FLAG
1571	006454	012737	001000	015454		MOV	#1000,COUNT	;SET FOR '512' CONVERSIONS
1572	006462	113737	015464	015441		MOV	KSTOR4,ADWORD2+1	;MOV SELECTED CH. TO HIGH BYTE OF ADWORD
1573	006470	042737	100377	015440		BIC	#100377,ADWORD2	;MASK
1574	006476	052737	000001	015440		BIS	#1,ADWORD2	
1575	006504	004737	311212			JSR	PC,ADCNVT	;TAKE THE CONVERSIONS
1576	006510	104004				COMPUTE		;AVERAGE & COMPUTE DISTRIBUTION
1577	006512	104005				CATORIZ		
1579	006514	013777	015554	172310		MOV	AVERAGE,ADISPLA	;LOAD DISPLAY REG. (11/45)
1579	006522	032777	002000	172300		BIT	#SW10,ASWR	;TEST DATA SW10
1580	006530	301047				BNE	REPT4	;IF SET, FORCE TYPE OUT
1581	006532	032777	020000	172270	TSTCT4:	BIT	#SW13,ASWR	;TEST FOR INHIBIT TYPEOUT
1582	006540	001135				BNE	REPT7	;BRANCH IF SW SET
1583	006542	022737	000004	015462		CMP	#4,KSTOR3	;WAS 4 TYPED
1584	006550	001005				BNE	TSTCT3	;NO, TEST FOR '3'
1585	006552	022737	001000	015530		CMP	#1000,XSPRD4	;TOTAL COUNTS WITHIN 4 COUNTS
1586	006560	001033				BNE	REPT4	;BRANCH IF NO.
1587	006562	000524				BR	REPT7	;YES, TEST NEXT CH.
1589	006564	022737	000003	015462	TSTCT3:	CMP	#3,KSTOR3	;COUNT = TO 3
1589	006572	001005				BNE	TSTCT2	;NO TEST COUNT 2
1590	006574	022737	001000	015626		CMP	#1000,XSPRD3	
1591	006602	001022				BNE	REPT4	;BRANCH IF COUNT NOT WITHIN 3
1592	006604	000513				BR	REPT7	;YES, TEST NEXT CH.
1593								
1594	006606	022737	000002	015462	TSTCT2:	CMP	#2,KSTOR3	;COUNT =TO 2
1595	006614	001005				BNE	TSTCT1	;NO, TEST COUNT 1
1596	006616	022737	001000	015624		CMP	#1000,XSPRD2	
1597	006624	001011				BNE	REPT4	;BRANCH IF NOT WITHIN 2
1598	006626	000502				BR	REPT7	;YES, TEST NEXT CH.
1599	006630	022737	000001	015462	TSTCT1:	CMP	#1,KSTOR3	;COUNT = TO 1
1600	006636	001004				BNE	REPT4	;NO, REPORT EVEN IF NOT '0'
1601	006640	022737	001000	015622		CMP	#1000,XSPRD1	
1602	006646	001472				BEQ	REPT7	;BRANCH IF TOTAL WITHIN 1 COUNT
1603	006650	104000			REPT4:	PRINT		
1604	006652	014470				CRLF		
1605	006654	005737	015532			TST	MESPR1	;TEST IF HEADER HAS BEEN TYPED
1606	006660	001002				BNE	REPT5	;BRANCH IF YES
1607	006662	104000				PRINT		
1608	006664	015114				MES19		;TEXT 'CH. HIGH AVG. LOW'
1609	006666	104013			REPT5:	TSTTKS		;TEST FOR KEYBOARD INTERRUPT
1610	006670	104000				PRINT		
1611	006672	014470				CRLF		;CARRIAGE RETURN, LINE FEED
1612	006674	013737	015464	007066		MOV	KSTOR4,REPT8A	;MOV. CH.
1613	006702	042737	177700	007066		BIC	#177700,REPT8A	
1614	006710	005737	015504			TST	USED5H	;TEST FOR DUAL MODE
1615	006714	001403				BEQ	REPT8	;BRANCH IF NOT
1616	006716	062737	000010	007066		ADD	#10,REPT8A	
1617	006724	104010			REPT9:	PRTOCT		
1618	006726	007066				REPT8A		
1619	006730	104007				SPACE		
1620	006732	104010				PRTOCT		;PRINT LOW VALJE
1621	006734	015540				LOW		
1622	006736	104007				SPACE		
1623	006740	104010				PRTOCT		;PRINT AVERAGE VALUE
1624	006742	015554				AVERAGE		

```

1625 006744 104007 SPACE
1626 006746 104010 PRTCT :PRINT HIGH VALUE
1627 006750 015536 HIGH
1628 006752 005737 015532 TST MESPRT
1629 006756 001002 BNE REPT6
1630 006760 104000 PRINT
1631 006762 015147 MES20 :PRINT 'COUNT SPREAD' HEADER
1632 006764 052737 000007 015532 REPT6: BIS #7,MESPRT :INHIBIT OTHER HEADERS
1633 006772 022737 001000 015604 CMP #1000,AVGCNT :TEST IF ALL COUNTS WERE AT AVG.
1634 007000 000240 NOP :<BEQ REPT7> BRANCH TO NEXT CH. IF YES.
1635 007002 104000 PRINT
1636 007004 014470 CRLF
1637 007006 012704 015570 MOV #ORLOW,R4
1638 007012 012402 REPT6A: MOV (R4)+,R2
1639 007014 104006 BINDEC :TYPE OUT COUNT SPREAD
1640 007016 022704 015622 CMP #XSPRD1,R4 :TEST FOR DONE
1641 007022 001373 BNE REPT6A :BRANCH IF NO AND TYPE NEXT COUNT
1642 007024 005777 172000 TST @SWR
1643 007030 100001 BPL REPT7
1644 007032 000000 HALT :REPEATIBILITY ERROR
1645 007034 013704 015464 REPT7: MOV KSTOR4,R4
1646 007040 042704 177700 BIC #177700,R4
1647 007044 023704 015460 CMP KSTOR2,R4 :TESTED ALL CH.(S)?
1648 007050 001404 BEQ REPT7A
1649 007052 005237 015464 INC KSTOR4 :TEST NEXT CHANNEL
1650 007056 000137 006452 JMP REPT3
1651 007062 000137 006440 REPT7A: JMP REPT2A
1652
1653 007066 000000 REPT8A: 0

```

```

1654
1655
1656
1657
1658
1659
1660
1661 007070 012737 007112 015446
1662 007076 005037 015506
1663 007102 005037 015504
1664 007106 104000
1665 007110 014556
1666 007112 104000
1667 007114 014756
1668 007116 104011
1669 007120 104001
1670 007122 013737 011202 015456
1671 007130 013737 011204 015460
1672 007136 013706 015444
1673 007142 104013
1674 007144 012737 000020 007456
1675 007152 005037 007460
1676 007156 012737 000010 015522
1677 007164 012737 000010 015454
1678 007172 113737 015456 015441
1679 007200 042737 140377 015440
1680 007206 052737 000001 015440
1681 007214 005237 015500
1682 007220 004737 011212
1683 007224 113737 015460 015441
1684 007232 042737 140377 015440
1685 007240 052737 000001 015440
1686 007246 005237 015500
1687 007252 004737 011212

```

```

:*****
:RECOVERY TEST
:*****
:THIS TEST IS DESIGNED TO TEST RECOVERY OF THE A/D CONVERTER VIA ACCEPT-
:ING TWO (2) CHANNEL FROM THE TELETYPE AND THEN TAKE A
:SERIES OF EIGHT CONVERSIONS ON EACH CHANNEL AND TYPING OUT THE CON-
:VERSION VALUES OF THE 2ND CHANNEL IN THE ORDER THEY WERE TAKEN.
RECVY: MOV #RECVY1,AVECTR ;SET UP THE 'A' RETURN ADDRESS
      CLR USEDMA
      CLR USED5H
      PRINT
      MESB ;TEXT 'RECOVERY TEST'
RECVY1: PRINT ;REQUEST CHANNELS
      MES14
      TTYIN ;WAIT FOR INPUT
      DECOCT ;CONVERT TO OCTAL
      MOV BCDTAB,KSTOR1 ;SAVE 1ST CH.
      MOV BCDTAB+2,KSTOR2 ;SAVE 2ND CH.
RECVY2: MOV STACK,SP
      TSTTKS ;CHECK FOR KEYBOARD FLAG
      MOV #16..10$ ;LOAD COUNT
      CLR 11$ ;CLEAR POINTER
      MOV #10,TEMP3 ;SET UP TO PRINT 10
      MOV #10,COUNT ;SETUP TO TAKE '8' CONVERSIONS ON 1ST CH.
1$: MOVB KSTOR1,ADWRD2+1 ;LOAD 1ST CH.
      BIC #140377,ADWRD2
      BIS #1,ADWRD2
      INC DELAY ;SET FLAG
      JSR PC,ADCNVT ;TAKE THE CONVERSIONS
      MOVB KSTOR2,ADWRD2+1 ;SET UP 2ND CH.
      BIC #140377,ADWRD2
      BIS #1,ADWRD2
      INC DELAY ;SET FLAG
      JSR PC,ADCNVT ;TAKE 2ND SERIES OF CONVERSIONS

```

K03

LPS-11 DIAGNOSTIC TEST I MAINDEC-11-DZLPC-C
DZLPC.C.P11 04-APR-76 00:00

MACY11 27(1006) 16-SEP-76 21:50 PAGE 36

1688	007256	013701	007460		MOV	11\$,R1		;LOAD A POINTER
1689	007262	012700	015654		MOV	#ADBUFF,R0		;LOAD POINTER
1690	007266	012061	015716		MOV	(R0)+,ADTB0(R1)		
1691	007272	012061	015760		MOV	(R0)+,ADTB1(R1)		
1692	007276	012061	016022		MOV	(R0)+,ADTB2(R1)		
1693	007302	012061	016064		MOV	(R0)+,ADTB3(R1)		
1694	007306	012061	016126		MOV	(R0)+,ADTB4(R1)		
1695	007312	012061	016170		MOV	(R0)+,ADTB5(R1)		
1696	007316	012061	016232		MOV	(R0)+,ADTB6(R1)		
1697	007322	012061	016274		MOV	(R0)+,ADTB7(R1)		
1698	007326	062737	000002	007460	ADD	#2,11\$;UPDATE POINTER
1699	007334	005337	007456		DEC	10\$		
1700	007340	001314			SNE	1\$		
1701								
1702	007342	012700	000010		MOV	#8,R0		;LOAD COUNT
1703	007346	012702	015654		MOV	#ADBUFF,R2		;LOAD POINTER
1704	007352	012701	000000		MOV	#0,R1		
1705	007356	012737	000017	015516	MOV	#15,TEMP1		;LOAD TEMP
1706	007364	012737	000004	012612	MOV	#4,CMP CNT		;LOAD AVRG. COUNT
1707	007372	016104	007464		MOV	LADTB(R1),R4		;GET POINTER
1708	007376	012746	000340		MOV	#340,-(SP)		;PUSH
1709	007402	004737	012446		JSR	PC,CMPTEA		;CONVERT AND AVERAGE
1710	007406	013722	015554		MOV	AVRAGE,(R2)+		;SAVE AVERAGE
1711	007412	005721			TST	(R1)+		;UPDATE POINTER
1712	007414	005300			DEC	R0		
1713	007416	001357			BNE	2\$;BR IF NOT DONE
1714								
1715	007420	032777	002000	171402	BIT	#SW10,JSWR		;TEST BIT 10
1716	007426	001243			BNE	RECVY2		
1717	007430	104000			PRINT			
1718	007432	014601			MES9			;TEXT 'CH.'
1719	007434	104013			TSTTKS			;CHECK AGAIN FOR KEYBOARD FLAG
1720	007436	013737	015460	007066	MOV	KSTOR2,REPT8A		
1721	007444	104010			PRTCT			
1722	007446	007066			REPT8A			
1723	007450	104007			SPACE			
1724	007452	104012			PRTAVG			;PRINT VALUES OF 2ND CH.
1725	007454	000630			BR	RECVY2		;DO IT AGAIN
1726	007456	000000						
1727	007460	000000						
1728	007462	000000						
1729								
1730	007464	015716			LADTB:	ADTB0		
1731	007466	015760				ADTB1		
1732	007470	016022				ADTB2		
1733	007472	016064				ADTB3		
1734	007474	016126				ADTB4		
1735	007476	016170				ADTB5		
1736	007500	016232				ADTB6		
1737	007502	016274				ADTB7		

1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779

007504 012737 007512 015446
007512 104000
007514 014771
007516 005037 001034
007522 005037 015474
007526 004537 013664
007532 015474
007534 000240
007536 052777 000100 171254
007544 013706 015444
007550 013777 001034 171254
007556 113777 015474 171256
007564 005277 171250
007570 105777 171244
007574 100375
007576 017737 171242 015544
007604 152777 000010 171230
007612 005277 171222
007616 105777 171216
007622 100375
007624 017737 171214 015546
007632 004537 010120
007636 015544
007640 015546
007642 001014
007644 032777 020000 171156
007652 001004
007654 104000
007656 014470
007660 104000
007662 015255
007664 005777 171140
007670 100001
007672 000000

```
*****  
: DUAL SAMPLE AND HOLD TEST <CH 0-3> AND <CH 10-13> ONLY  
*****  
DSHTST: MOV #DSHTD,AVECTR ;SET UP '1A' RETURN  
DSHT0: PRINT ;TYPE HEADER  
MES15  
DSHT1: CLR PASSCT ;CLEAR PASS COUNT  
DSHT2: CLR STCHAN ;CLEAR STARTING CHANNEL  
DSHT3: JSR R5,LEDS  
STCHAN  
NOP ;RESET'''  
BIS #BIT6,@TKS ;ENABLE INTERRUPT  
MOV STACK,SP ;LOAD THE STACK POINTER  
MOV PASSCT,@DISPLA ;LOAD PASS COUNT  
MOV STCHAN,@ADCS1 ;LOAD CHANNEL <0-3>  
INC @ADCS ;START CONVERSION  
TSTB @ADCS ;WAIT FOR DONE  
BPL -4  
MOV @ADDBR,AVERM4 ;SAVE THE CONVERTED VALUE  
BISB #10,@ADCS1 ;UPDATE THE CHANNEL  
INC @ADCS ;START CONVERSION  
TSTB @ADCS ;WAIT FOR DONE  
BPL -4  
MOV @ADDBR,AVERM3 ;SAVE THE CONVERTED VALUE  
JSR R5,COMPR ;COMPARE TWO NUMBERS  
AVERM4  
AVERM3  
BNE DSHT6 ;BRANCH IF NOT EQUAL <DIFFERENT INPUT>  
BIT #BIT13,@SWR  
BNE DSHT4  
PRINT  
CRLF  
PRINT  
MES22 ;PRINT MESSAGE  
DSHT4: TST @SWR  
BPL DSHT6  
HALT ;ERROR, CHANNEL PAIRS HAVE SAME INPUT VOLTAGE  
;THE FIRST CHANNEL NUMBER OF THE PAIR IS IN THE LEDES
```

1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1810
 1811

007674 113777 015474 171140
 007702 052777 040000 171130
 007710 005277 171124
 007714 105777 171120
 007720 100375
 007722 005277 171112
 007726 017737 171112 015550
 007734 105777 171100
 007740 100375
 007742 017737 171076 015552
 007750 004537 010120
 007754 015550
 007756 015552
 007760 001015
 007762 032777 020000 171040
 007770 001004
 007772 104000
 007774 014470
 007776 104000
 010000 015333
 010002 005777 171022
 010006 100014
 010010 000000
 010012 000412

:NOW TEST THAT DUAL SAMPLE REALLY WORKS
 :SET DUAL MODE AND CONVERT A CHANNEL THE FIRST CONVERSION
 :SHOULD NOT EQUAL THE SECOND CONVERSION
 :IF IT DOES THE DUAL SAMPLE AND HOLD IS NOT WORKING CORRECTLY

DSHT6: MOVB STCHAN, @ADCS1 ;LOAD 1ST MUX CHANNEL
 BIS #BIT14, @ADCS ;SET 'DUAL MODE'
 INC @ADCS ;START CONVERSION
 TSTB @ADCS ;WAIT FOR DONE
 BPL -4
 INC @ADCS ;START SECOND CONVERSION
 MOV @ADDBR, AVERM2 ;SAVE 1ST CHANNEL CONVERSION
 TSTB @ADCS ;WAIT FOR DONE
 BPL -4
 MOV @ADDBR, AVERM1 ;SAVE 2ND CHANNEL CONVERSION

JSR R5, COMPR
 AVERM2
 AVERM1
 BNE DSHT7 ;BRANCH IF NOT EQUAL
 BIT #BIT13, @SWR
 BNE DSHT6A

MES23 ;PRINT ERROR
 DSHT6A: TST @SWR
 BPL DSHT7A
 HALT ;ERROR, FIRST CONVERSION ON CHANNEL X (STCHAN) WAS EQUAL
 BR DSHT7A ;TO THE SECOND CONVERSION IN DUAL MODE

```

1812
1813 010014 004537 010120          DSHT7: JSR      R5,COMPR
1814 010020 015544                      AVERM4          ;COMPARE TWO NUMBERS
1815 010022 015550                      AVERM2          ;TEST IF NOT EQUAL
1816 010024 001017                      BNE            DSHT8
1817 010026 004537 010120          JSR      R5,COMPR          ;COMPARE TWO SAMPLES
1818 010032 015546                      AVERM3
1819 010034 015552                      AVERM1
1820 010036 001012                      BNE            DSHT8          ;BRANCH IF NOT EQUAL
1821 010040 005237 015474          DSHT7A: INC      STCHAN          ;UPDATE CHANNEL
1822 010044 022737 000004 015474    CMP      #4,STCHAN          ;FINISHED ?
1823 010052 001225                      BNE            DSHT3          ;NO
1824 010054 005037 015474          CLR      STCHAN          ;CLEAR CHANNEL
1825 010060 104013                      TSTTKS
1826 010062 000414                      BR         DSHT8B          ;TEST KEYBOARD
1827
1828 010064 032777 020000 170736    DSHT8: BIT      #BIT13,DSWR
1829 010072 001004                      BNE            DSHT8A
1830 010074 104000                      PRINT
1831 010076 014470                      CRLF
1832 010100 104000                      PRINT
1833 010102 015412                      MES24
1834 010104 005777 170720          DSHT8A: TST      DSWR
1835 010110 100001                      BPL      DSHT8B
1836 010112 000000                      HALT
1837 010114 000137 010040          DSHT8B: JMP      DSHT7A          ;ERROR, PREVIOUS CONVERSTION ON CHANNEL NOT EQUAL
1838                                     ; CURRENT CONVERSION
1839 010120 013537 010210          COMPR: MOV      @ (R5)+,CMPRA          ;SAVE 1ST NUMBER
1840 010124 013537 010212          MOV      @ (R5)+,CMPRB          ;SAVE 2ND NUMBER
1841 010130 013737 010212 010214    MOV      CMPRB,CMPRC          ;SAVE 2ND AGAIN
1842 010136 163737 010210 010214    SUB      CMPRA,CMPRC          ;SUBTRACT 2ND FROM 1ST
1843 010144 001413                      BEQ      CMPR2          ;BRANCH IF EQUAL
1844 010146 100005                      BPL      CMPR1          ;BRANCH IF + <A-B=+>
1845 010150 063737 001010 010214    ADD      AMASK,CMPRC          ;ADD SPREAD
1846 010156 100006                      BPL      CMPR2          ;BRANCH IF <A=B+MASK>
1847 010160 000410                      BR       CMPR3          ;BRANCH IF
1848 010162 163737 001010 010214    CMPR1: SUB      AMASK,CMPRC          ;SUBTRACT AMASK
1849 010170 003401                      BLE      CMPR2          ;BRANCH IF EQUAL *
1850 010172 000403                      BR       CMPR3          ;BRANCH IF NOT-EQUAL *
1851
1852 010174 005037 010214          CMPR2: CLR      CMPRC          ;A AND B ARE WITHIN AMASK
1853 010200 000205                      RTS      R5
1854
1855 010202 005137 010214          CMPR3: COM      CMPRC          ;A AND B ARE NOT WITHIN AMASK
1856 010206 000205                      RTS      R5
1857
1858 010210 000000          CMPRA: 0
1859 010212 000000          CMPRB: 0
1860 010214 000000          CMPRC: 0

```


:ROUTINE TO LOOP THRU A SINGLE LOGIC SUBTEST. ENTERED FROM THE 'MONITOR'
:VIA TYPING 'TNN' WHERE 'NN' IS EQUATED TO THE 'PC' OF A SUBTEST.
:NOTE THAT 'SW11' MUST BE '0' (DOWN) TO RUN THIS TEST.

010216
010220
010224
010228
010232
010236
010240
010244
010248
010252
010256
010260

010264
010268
010272
010276
010304
010308
010312

010324
010328
010332
010336
010340
010344
010348
010352

010356
010360
010364
010368
010372
010376
010380
010384
010388
010392
010396
010400
010404
010408
010412

017701 011032
005027 015456
047711 177770
062713 015456
005337 015452
001407
006337 015456
006337 015456
006337 015456
000762

022737 005742 015456
100302
000137 001666
062737 000002 015456
005037 013576
012737 010304 013600
000177 005134

105777 170476
100375
012777 000240 170470
005337 010352
003367
005037 010352
000002
000000

012737 015654 010364
104010
015654
062737 000002 010364
012737 000002 010352
104007
005337 015522
001364
000002

TESTX: MOV #INBUF,R1
CLR KSTOR1
TSTA: SIC #177770,(R1) :MASK TO OCTAL
ADD (R1)+,KSTOR1 :ADD TO LAST RESULT
DEC CHRCNT
BEQ TS*XB
ASL KSTOR1
ASL KSTOR1
ASL KSTOR1
BR TSTA

TS*XB: CMP #CALBRT,KSTOR1 :IS NO. WITHIN LIMITS OF THE LOGIC TEST
BPL +6 :CONTINUE IF YES
JMP INIT3 :OTHERWISE RETURN TO MONITOR
ADD #2,KSTOR1 :ADD '2' TO POINT TO INSTRUCTION AFTER SCOPE
XLOOP: CLR SCOPEF :KEEP COUNT AT ZERO
MOV #XLOOP,RETURN :LOAD SCOPE LOOP RETURN POINTER
JMP #KSTOR1 :JUMP TO TEST

:SUBROUTINE TO ISSUE N SPACES
:N IS ONE PLUS VALUE CONTAINED IN SPACEX
:SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
:SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE

XSPACE: TSTB #TPS :WAIT FOR TTY READY
BPL -4
MOV #240,TPB :OUTPUT A SPACE
DEC SPACEX :DECREMENT COUNT
BGT XSPACE :LOOP IF NOT DONE
CLR SPACEX :RESET COUNT TO ZERO
RTI :RETURN
SPACEX: 0

:SUBROUTINE TO TYPE OUT AVERAGES FOR THE RECOVERY TEST

XPRTAV: MOV #ADBUFF,AVGTAB
XPTA1: PRTCT
AVGTAB: ADBUFF
ADD #2,AVGTAB
MOV #2,SPACEX
SPACE
DEC TEMP3
BNE XPTA1
RTI

```

1903          :KEYBOARD SERVICE ROUTINE
1904          XTTYIN: MOV      R0, -(SP)
1905          MOV      R1, -(SP)
1906          MOV      R2, -(SP)
1907          MOV      R3, -(SP)
1908          MOV      R4, -(SP)
1909          MOV      R5, -(SP)
1910          NEWIN: MOV      @INBUF, R4          :SETUP CHARACTER BUFFER
1911          BIC      @BIT6, @TKS
1912          CLR      @CHARCNT          :CLEAR CHARACTER COUNTER
1913          CLR      @INBUF
1914          CLR      @INBUF+2
1915          CLR      @INBUF+4
1916          CLR      @INBUF+6
1917          CLR      @INBUF+10
1918          CLR      @INBUF+12
1919          CLR      @INBUF+14
1920          INPUTA: TSTB     @TKS          :CHARACTER READY?
1921          BPL      @INPUTA          :NO, WAIT IT OUT
1922          MOV      @TKB, R1          :SAVE CHARACTER
1923          BIC      @177600, R1       :STRIP PARITY BIT
1924          CPB     R1, @6C          :IS IT A SPECIAL CHARACTER
1925          BMI     @SPCHR          :YES, TEST IT
1926          BMI     @SPCHR
1927          BMI     @SPCHR
1928          MOV      R1, @R4)+       :SAVE CHARACTER
1929          INC      @CHARCNT          :INCREMENT THE CHARACTER COUNT.
1930          CMP     @6, @CHARCNT
1931          BMI     @SPCHR5
1932          CLTPTA: TSTB     @TPS          :ECHO CHARACTER
1933          BPL      @OUTPTA
1934          MOV     R1, @TPB
1935          BR      @INPUTA          :WAIT FOR NEXT CHARACTER
1936          :SUBROUTINE TO TEST FOR SPECIAL CHARACTERS : 'A', 'C', 'G', 'CR', '.' OR 'RUBOUT'
1937          SPCHR:  CMPB     @1, R1          :CHAR. = 'A'
1938          BNE     @SPCHR1          :NO, NOT 'A'
1939          PRINT   @SPCHR1          :ECHO 'A'
1940          CNTRLA
1941          MOV     (SP)+, R5
1942          MOV     (SP)+, R4
1943          MOV     (SP)+, R3
1944          MOV     (SP)+, R2
1945          MOV     (SP)+, R1
1946          MOV     (SP)+, R0
1947          CMP     (SP)+, (SP)+
1948          BIS     @BIT6, @TKS          :ENABLE KEYBOARD INTR.
1949          JMP     @VECTA          :YES, EXIT VIA 'A' VECTOR ADDRESS.
1950          SPCHR1: CMPB     @3, R1          :CHAR. = 'C'
1951          BNE     @IS          :NO, NOT 'C'
1952          JMP     @MONITR          :YES, EXIT TO MONITOR
1953          IS:    CMPB     @177, R1       :CHAR. = 'RUBOUT'
1954          BNE     @SPCHR3          :NO TRY ANOTHER CHAR
1955          TST     @CHARCNT          :IS RUBOUT LEGAL?
1956          0103414 010046
1957          0103416 010146
1958          0103420 010246
1959          0103422 010346
1960          0103424 010446
1961          0103426 010546
1962          0103428 012704 011032
1963          0103430 042777 000100 170356
1964          0103432 005037 015452
1965          0103434 005037 011032
1966          0103436 005037 011034
1967          0103438 005037 011036
1968          0103440 005037 011040
1969          0103442 005037 011042
1970          0103444 005037 011044
1971          0103446 005037 011046
1972          0103448 105777 170312
1973          0103450 100375
1974          0103452 017701 170306
1975          0103454 042701 177600
1976          0103456 120127 000060
1977          0103458 100420
1978          0103460 122701 000173
1979          0103462 100415
1980          0103464 010124
1981          0103466 005237 015452
1982          0103468 022737 000006 015452
1983          0103470 100524
1984          0103472 105777 170246
1985          0103474 100375
1986          0103476 110177 170242
1987          0103478 000746
1988          010566 122701 000001
1989          010572 001016
1990          010574 104000
1991          010576 014436
1992          010600 012605
1993          010602 012604
1994          010604 012603
1995          010606 012602
1996          010610 012601
1997          010612 012600
1998          010614 022626
1999          010616 052777 000100 170174
2000          010624 000177 004616
2001          010630 122701 000003
2002          010634 001002
2003          010636 000137 001426
2004          010642 122701 000177
2005          010646 001011
2006          010650 005737 015452

```

1965	010654	001712			BEQ	INPUTA		:NO, IGNORE IT
1966	010656	005337	015452		DEC	CHARCNT		
1967	010662	012701	000134		MOV	#134,R1		:TYPE '\ ' TO INDICATE RUBOUT
1968	010666	005744			TST	-(R4)		:POP OFF LAST CHARACTER
1969	010670	000730			BR	OUTPTA		:WAIT FOR NEXT CHARACTER
1970	010672	122701	000054	SPCHR3:	CMPB	#54,R1		:TEST FOR ' '
1971	010676	001716			BEQ	INPUTB		:LEGAL CHAR. SAVE IT
1972	010700	122701	000015	SPCHR4:	CMPB	#15,R1		:TO 'CARRIAGE RETURN' TO TERMINATE
1973	010704	001014			BNE	IS		:NO, CONTINUE
1974	010706	104000			PRINT			:YES, TYPE 'CR-LF'
1975	010710	014470			CRLF			
1976	010712	012605		4\$:	MOV	(SP)+,R5		
1977	010714	012604			MOV	(SP)+,R4		
1978	010716	012603			MOV	(SP)+,R3		
1979	010720	012602			MOV	(SP)+,R2		
1980	010722	012601			MOV	(SP)+,R1		
1981	010724	012600			MOV	(SP)+,R0		
1982	010726	052777	000100	170054	BIS	#BIT6,BTKS		:ENABLE KEYBOARD INTR.
1983	010734	000002			RTI			:EXIT
1984	010736	122701	000007	1\$:	CMPB	#7,R1		:TEST IF CTRL G
1985	010742	001027			BNE	SPCHRS		:BR IF NOT
1986	010744	104000			PRINT			
1987	010746	014441			CNTRLG			
1988	010750	104010			PRTOCT			:PRINT OLD VALUE
1989	010752	000170			SOFTSW			
1990	010754	104000			PRINT			
1991	010756	014457			NEWSWR			:ASK FOR NEW VALUE
1992	010760	104011			TTYIN			
1993	010762	005001			CLR	R1		
1994	010764	012700	011032		MOV	#INBUF,R0		
1995	010770	005710		2\$:	TST	(R0)		:TEST FOR TERM
1996	010772	001410			BEQ	3\$:BR IF TERM
1997	010774	012002			MOV	(R0)+,R2		:GET CHAR
1998	010776	042702	177770		BIC	#177770,R2		:MASK OUT
1999	011002	006301			ASL	R1		
2000	011004	006301			ASL	R1		
2001	011006	006301			ASL	R1		
2002	011010	060201			ADD	R2,R1		
2003	011012	000766			BR	2\$		
2004	011014	010137	000170	3\$:	MOV	R1,SOFTSW		:SAVE NEW SWITCH VALUE
2005	011020	000734			BR	4\$		

```

2006 011022 104000          SPCHRS: PRINT          ;OTHERWISE TYPE '?'
2007 011024 014475          QMARK
2008 011026 000137 010414  JMP          XTTYIN    ; WAIT FOR NEW ENTRY
2009 011032 000000          INBUF: 0              ;CHARACTER STORAGE BUFFER
2010
2011
2012          011050
2013          =.+14
2014          :SUBROUTINE WILL CONVERT 'N' WORDS (SEPARATED VIA COMMA'S)
2015          :WHICH WERE STORED IN A TABLE VIA 'TTYIN' TO OCTAL AND STORE THEM.
2016 011050 012704 011032  BCDBIN: MOV          #INBUF,R4      ;SETUP ASCII STORAGE TABLE
2017 011054 012703 011202      MOV          #BCDTAB,R3      ;TABLE FOR STORAGE OF CONVERTED WORDS
2018 011060 005037 011204      CLR          BCDTAB+2
2019 011064 005001          BCDBN1: CLR          R1      ;REG. TO STORE RUNNING TOTAL
2020 011066 005002          CLR          R2      ;TEMP. STORAGE FOR 'R1'
2021 011070 005737 015452  BCDBN2: TST          CHRCNT    ;END OF DATA?
2022 011074 003424          BLE          BCDEND      ;YES, EXIT
2023 011076 005337 015452  DEC          CHRCNT      ;DECREMENT CHARACTER COUNTER
2024 011102 122714 000054  CMPB        #54,(R4)      ;IS CHARACTER = TO '?'
2025 011106 001417          BEQ          BCDEND      ;YES, DECODE NEW WORD
2026 011110 121427 000060  CMPB        (R4),#60
2027 011114 002425          BLT          BCDERR      ;TEST FOR LEGAL NO.
2028 011116 021427 000067  CMP          (R4),#67
2029 011122 003022          BGT          BCDERR
2030 011124 042714 177770  BIC          #177770,(R4) ;STRIP NO.
2031 011130 012400          MOV          (R4)+,R0      ;SAVE NO. IN R0.
2032 011132 010102          MOV          R1,R2      ;SAVE CURRENT TOTAL
2033 011134 006301          ASL          R1      ;NX2
2034 011136 006301          ASL          R1      ;NX4
2035 011140 006301          ASL          R1      ;NX8
2036 011142 060001          ADD          R0,R1      ;N+NEW NO.
2037 011144 000751          BR          BCDBN2
2038 011146 020127 000077  BCDEND: CMP          R1,#77
2039 011152 003006          BGT          BCDERR
2040 011154 005724          TST          (R4)+      ;UPDATE BUFFER
2041 011156 010123          MOV          R1,(R3)+    ;SAVE CONVERTED VALUE & SETUP TO SAVE NEXT
2042 011160 005737 015452  TST          CHRCNT      ;FINISHED?
2043 011164 001337          BNE          BCDBN1      ;NO, CONVERT NEXT WORD
2044 011166 000002          RTI          ;YES, EXIT
2045 011170 104000          BCDERR: PRINT
2046 011172 014475          QMARK
2047 011174 104011          TTYIN
2048 011176 000137 011050  JMP          BCDBIN
2049 011202 000000          BCDTAB: 0
2050 011204 000000          0
2051 011206 000000          0
2052 011210 000000          0

```

```

2053          ;SUBROUTINE TO TAKE 'N' CONVERSIONS AND STORE THEM IN AN A/D BUFFER. ROUTINE
2054          ;IS ENTERED WITH 'N' IN COUNT AND THE CH TO BE CONVERTED IN 'ADWORD'.
2055
2056 011212 005077 167600 ADCNVT: CLR      QPSW
2057 011216 013737 015454 015516 MOV      COUNT,TEMP1 ;SET UP # OF CONVERSIONS TO BE TAKEN
2058 011224 012704 015654 MOV      #ADBUFF,R4  ;LOAD BUFFER ADDRESS
2059 011230 005037 015476 CLR      FIRST      ;CLEAR TEMP
2060 011234 005737 015500 TST      DELAY      ;TEST DELAY
2061 011240 001031 BNE      6$         ;IF SET, SKIP
2062 011242 117737 167574 005740 MOV      QADCS1,DMK2 ;READ MUX CHANNELS
2063 011250 123737 015441 005740 CMP      ADWORD2+1,DMK2 ;TEST
2064 011256 001411 BEQ      1$         ;BR IF SAME
2065 011260 113777 015441 167554 MOV      ADWORD2+1,QADCS1 ;LOAD MUX CHANNELS
2066 011266 013737 012100 015502 MOV      CPTIME,DELAY1
2067 011274 005337 015502 2$: DEC      DELAY1
2068 011300 001375 BNE      2$         ;DELAY
2069 011302 005037 015500 1$: CLR      DELAY
2070 011306 005737 015506 TST      US_DMA     ;TEST DMA FLAG
2071 011312 001051 BNE      DMACVT    ;USE DMA
2072 011314 113777 015440 167516 4$: MOV      ADWORD2,QADCS ;LOAD CONTROL BYTE
2073 011322 000403 BR       5$
2074 011324 013777 015440 167506 6$: MOV      ADWORD2,QADCS ;LOAD CONTROL WORD
2075          ;LOAD CH. & START CONVERT IF NOT 'EXT'
2076 011332 105777 167502 5$: TST      QADCS
2077 011336 100375 BPL      5$         ;WAIT FOR DONE
2078 011340 005777 167474 TST      QADCS
2079 011344 100014 BPL      11$        ;TEST ERROR BIT
2080 011346 005077 167466 CLR      QADCS
2081 011352 037727 167452 020000 BIT      QSWR,#BIT13 ;BRANCH IT NOT SET
2082 011360 001002 BNE      10$
2083 011362 104000 PRINT
2084 011364 014706 MES12
2085 011366 005777 167436 10$: TST      QSWR
2086 011372 100001 BPL      11$
2087 011374 000000 HALT
2088          ;ERROR BIT SET
2089 011376 005737 015504 11$: TST      USED$H
2090 011402 001407 BEQ      12$        ;BRANCH IF NOT DUAL MODE
2091 011404 005137 015476 COM      FIRST
2092 011410 001404 BEQ      12$
2093 011412 017737 167426 015524 MOV      QADDBR,BRLEV1 ;EJECT THIS VALUE
2094 011420 000735 BR       4$
2095 011422 017724 167416 12$: MOV      QADDBR,(R4)+ ;SAVE DATA
2096 011426 005337 015516 DEC      TEMP1      ;DECREMENT COUNTER
2097 011432 003330 BGT      4$
2098 011434 000207 RTS      PC
2099

```

```

2100
2101 011436 012777 011606 167410 DMACVT: MOV #DACUTC,ADINT ;SET UP RETURN ADDRESS
2102 011444 012704 030000 MOV #30000,R4
2103 011450 105077 167364 CLRB ADACS
2104 011454 112777 000006 167356 MOVB #6,ADACS ;LOAD CA POINTER
2105 011462 012777 015654 167362 MOV #ADBUFF,ADMR ;LOAD CURRENT ADDRESS
2106 011470 112777 000004 167342 MOVB #4,ADACS ;LOAD WO POINTER
2107 011476 012777 177000 167346 MOV #-1000,ADMR ;LOAD WORD COUNT
2108 011504 112777 000002 167326 MOVB #2,ADACS ;LOAD DMA STATUS POINTER
2109 011512 012777 010000 167332 MOV #BIT12,ADMR ;LOAD DMA ENABLE
2110 011520 052737 000010 015440 BIS #BIT3,ADWRD2
2111 011526 113777 015440 167304 MOVB ADWRD2,ADACS
2112 011534 005304 DACUTA: DEC R4
2113 011536 001376 BNE DACUTA
2114 011540 005777 167274 TST ADACS
2115 011544 100404 BMI DACUTB
2116 011546 005077 167266 CLR ADACS
2117 011552 104400 ERROR
2118 011554 000000 HALT ;A TO D FAILED TO INTERRUPT IN BURST MODE
2119 011556 005077 167256 DACUTB: CLR ADACS
2120 011562 037727 167242 002000 BIT ASWR,#2000
2121 011570 001002 BNE DACUTD
2122 011572 104000 PRINT
2123 011574 014706 MES12 ;PRINT 'ERROR BIT SET'
2124 011576 005777 167226 DACUTD: TST ASWR
2125 011602 100001 BPL DACUTC
2126 011604 000000 HALT ;ERROR BIT SET IN BURST MODE
2127
2128 011606 005077 167226 DACUTC: CLR ADACS
2129 011612 012777 000006 167220 MOV #6,ADACS
2130 011620 022777 017654 167224 CMP #ADBUFF+2000,ADMR
2131 011626 001401 BEQ .+4
2132 011630 104400 ERROR ;ERROR, DMA CURRENT ADDRESS IN ERROR
2133
2134 011632 012777 000004 167200 MOV #4,ADACS
2135 011640 005777 167206 TST ADMR
2136 011644 001401 BEQ .+4
2137 011646 104400 ERROR ;ERROR, DMA WORD COUNT IN ERROR
2138
2139 011650 012777 000002 167162 MOV #2,ADACS
2140 011656 005777 167170 TST ADMR
2141 011662 001401 BEQ .+4
2142 011664 104400 ERROR ;ERROR, DMA STATUS IN ERROR
2143
2144 011666 005237 015512 INC OPS1
2145 011672 022626 CMP (SP)+,(SP)+
2146 011674 000207 RTS PC

```

2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175

011676 010046
011700 010146
011702 010246
011704 010346
011706 010446
011710 010546
011712 013746 000024
011716 010637 015450
011722 012737 011732 000024
011730 000000

011732 012777 000340 167056
011740 013706 015450
011744 012637 000024
011750 012605
011752 012604
011754 012603
011756 012602
011760 012601
011762 012600
011764 000005
011766 104000
011770 015235
011772 000137 001426

:POWER FAIL HANDLER

PWRFAL: MOV R0,-(SP)
MOV R1,-(SP)
MOV R2,-(SP)
MOV R3,-(SP)
MOV R4,-(SP)
MOV R5,-(SP)
MOV 24,-(SP)
MOV SP,PROC
MOV #PWRUP,24
HALT

:POWER UP HANDLER

PWRUP: MOV #340,PSW
MOV PROC,SP
MOV (SP)+,24
MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
RESET
PRINT
MES21
JMP MONITR

```

2176
2177
2178
2179
2180
2181
2182 011776 011646
2183 012000 162716 000002
2184 012004 017616 000000
2185 012010 005716
2186 012012 001001
2187 012014 000000
2188 012016 006316
2189 012020 042716 177001
2190 012024 062716 012036
2191 012030 017616 000000
2192 012034 000136
2193
2194
2195
2196 012036 012102
2197 012040 011050
2198 012042 013506
2199 012044 013602
2200 012046 012426
2201 012050 012614
2202 012052 012230
2203 012054 010322
2204 012056 013154
2205 012060 010414
2206 012062 010354
2207 012064 013272
2208 012066 013642
2209
2210 012070 002000
2211 012072 002400
2212 012074 003500
2213 012076 006000
2214
2215 012100 002600

```

```

;EMT DISPATCH SERVICE ROUTINE
;ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
;TO THE SELECTED SUBROUTINE.

```

```

EMTSRV: MOV      (SP), -(SP)      ;GET PC FOR TO RETURN
        SUB      #2, (SP)       ;PC OF EMT
        MOV      @ (SP), (SP)   ;GET EMT
        TST      (SP)          ;IS EMT VALID?
        BNE      EMTOK
        HALT
EMTOK:  ASL      (SP)           ;INVALID EMT
        BIC      #177001, (SP)  ;MULTIPLY EMT ARG BY '2'
        ADD      #EMTTAB, (SP)  ;CLEAR UNWANTED BITS
        MOV      @ (SP), (SP)  ;POINTER TO SUBROUTINE ADDRESS
        JMP      @ (SP)+       ;SUBROUTINE ADDRESS
        ;GO TO SUBROUTINE

```

```

;EMT DISPATCH TABLE

```

```

EMTTAB: TYPMES      ;MESSAGE PRINT ROUTINE
        BCOBIN     ;BINARY CONVERSION ROUTINE
        SCOPEC    ;LOGIC TEST SCOPE ROUTINE
        SCOPEH    ;LOGIC TEST SCOPE LOOP (10)
        CMPTE     ;SUBROUTINE TO COMPUTE THE AVG
        CATORZ    ;SUBROUTINE TO COMPUTE 'COUNT SPREAD'
        DECPRT   ;SUBROUTINE TO CONVERT OCT TO DEC + PRINT
        XSPACE   ;SUBROUTINE TO TYPE SPACES
        OCTPRT   ;OCTAL PRINT ROUTINE
        XTTYIN   ;TELEPRINTER SERVICE ROUTINE
        XPRTAV   ;SUBROUTINE TO PRINT OUT THE GAIN AVERAGES
        TKSFLG   ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
        SCOPEI   ;SCOPE ROUTINE

```

```

CPDLAY: 2000      ;PDP-11/05
        2400      ;PDP-11/20
        3500      ;PDP-11/40
        6000      ;PDP-11/45

```

```

CPTIME: 2000

```


2216
2217
2218
2219
2220 012102 005077 166710
2221 012106 017605 000000
2222 012112 062716 000002
2223 012116 105777 166702
2224 012122 100375
2225 012124 122715 000100
2226 012130 001001
2227 012132 000002
2228 012134 122715 000045
2229 012140 001403
2230 012142 112577 166660
2231 012146 000763
2232 012150 012777 000015 166650
2233 012156 105777 166642
2234 012162 100375
2235 012164 012777 000012 166634
2236 012172 013737 001012 012226
2237 012200 105777 166620
2238 012204 100375
2239 012206 113777 001014 166612
2240 012214 005337 012226
2241 012220 100367
2242 012222 105725
2243 012224 000734
2244 012226 000002
2245
2246
2247
2248 012230 005077 166562
2249 012234 012737 177774 012410
2250 012242 012737 012416 012414
2251 012250 012737 000240 012412
2252 012256 012737 177777 012406
2253 012264 005237 012406
2254 012270 167702 000120
2255 012274 100373
2256 012276 067702 000112
2257 012302 004737 012326
2258 012306 005237 012410
2259 012312 001001
2260 012314 000002
2261 012316 062737 000002 012414
2262 012324 000754
2263 012326 005737 012406
2264 012332 001010
2265 012334 022737 177777 012410
2266 012342 001404
2267 012344 013737 012412 012406
2268 012352 000406
2269 012354 012737 000260 012412
2270 012362 052737 000260 012406
2271 012370 105777 166430

:MESSAGE PRINT ROUTINE, ENTERED IA EMT DISPATCH HANDLER.
:ROUTINE PICKS UP CONTENTS OF THE 'PC' AND USES THIS AS
:THE ADDRESS OF MESSAGE TO BE TYPED.

TYPMES: CLR @PSW ;ENABLE KEYBOARD INTR.
MOV @ (SP), R5 ;GET THE MESSAGE ADDRESS FROM START
ADD #2, (SP) ;SET UP STACK TO EXIT
TYPERA: TSTB @TPS
BPL TYPERA ;WAIT FOR TTY DONE
CMPB #100, (R5) ;TEST FOR 'a'
BNE TYPER1 ;BRANCH IF NO EQUAL
RTI ;OTHERWISE EXIT
TYPER1: CMPB #45, (R5) ;TEST FOR '%'
BEQ TYPECL ;IF = TYPE 'CR-LF'
TYPER2: MOVB (R5)+, @TPB ;OUTPUT CHAR.
BR TYPERA
TYPECL: MOV #15, @TPB ;TYPE 'CR'
TSTB @TPS
BPL .-4
MOV #12, @TPB
MOV FILLS, 2\$;LOAD FILL COUNT
1\$: TSTB @TPS
BPL 1\$
MOVB FILCHR, @TPB ;FILL CHAR
DEC 2\$
BPL 1\$
TSTB (R5)+ ;INCREMENT BUFFER
BR TYPERA
2\$: 2

:PRINT DECIMAL VALUE IN R2

DECPRT: CLR @PSW
MOV #-4, DIGCNT
MOVB #DECPNT+2, DECPNT
TYPT1: MOV #240, ZERO
TYPT2: MOV #-1, DIGIT
INC DIGIT
SUB @DECPNT, %2
BPL TYPT2
ADD @DECPNT, %2
JSR PC, DECOU
INC DIGCNT
BNE TYPT3
RTI
TYPT3: ADD #2, DECPNT
BR TYPT1
DECOU: TST DIGIT
BNE DEC1
CMP #-1, DIGCNT
BEQ DEC1
MOV ZERO, DIGIT
BR DEC2
DEC1: MOV #260, ZERO
BIS #260, DIGIT
DEC2: TSTB @TPS

```

2272 012374 100375          BPL      .-4
2273 012376 013777 012406 166422  MOV     DIGIT,ATPB
2274 012404 000207          RTS     7
2275
2276 012406 000000          DIGIT:  0
2277 012410 000000          DIGCNT: 0
2278 012412 000240          ZERO:   240
2279 012414 012416          DECPNT: .+2
2280 012416 001750          1000.
2281 012420 000144          100.
2282 012422 000012          10.
2283 012424 000001          1.
2284
2285          ;COMPUTE THE RESULTS OF 512 CONVERSIONS AS HIGH, LOW AND AVERAGE
2286
2287 012426 012737 000777 015516  CMPTE:  MOV     #777,TEMP1          ;SET UP TO COMPARE '511' NUMBERS
2288 012434 012704 015654          MOV     #ADBUFF,R4          ;SET UP DATA BUFFER ADDRESS
2289 012440 012737 000011 012612  MOV     #11,CMPCNT          ;LOAD AVRG. COUNT
2290 012446 005037 015534          CMPTEA: CLR     HIORDV
2291 012452 012437 015554          MOV     (R4)+,AVRAGE          ;STORE 1ST VALUE AS AVERAGE
2292 012456 013737 015554 015536  MOV     AVRAGE,HIGH          ;HIGH
2293 012464 013737 015554 015540  MOV     AVRAGE,LOW          ;& LOW
2294 012472 012437 015520          GETDAT: MOV     (R4)+,TEMP2
2295 012476 023737 015520 015536  CMP     TEMP2,HIGH          ;IS NEW NO. GREATER THAN OLD NO.
2296 012504 003403          BLE     TSLO                ;BRANCH IF NOT GREATER
2297 012506 013737 015520 015536  MOV     TEMP2,HIGH          ;OTHERWISE SAVE AS NEW HIGH
2298 012514 023737 015520 015540  TSLO:  CMP     TEMP2,LOW
2299 012522 003003          BGT     TAGA
2300 012524 013737 015520 015540  MOV     TEMP2,LOW          ;OTHERWISE SAVE AS NEW LOW
2301 012532 063737 015520 015554  TAGA:  ADD     TEMP2,AVRAGE          ;ADD LOW ORDER
2302 012540 005537 015534          ADC     HIORDV              ;ADD CARRY TO HI ORDER
2303 012544 005337 015516          DEC     TEMP1
2304 012550 001350          BNE     GETDAT
2305 012552 013737 012612 015516  MOV     CMPCNT,TEMP1          ;512 OR 16 ADDITIONS?
2306 012560 006237 015534          AVGDAT: ASR     HIORDV          ;YES. DIVIDE/512 OR 16
2307 012564 006037 015554          ROR     AVRAGE              ;SHIFT CARRY BIT INTO LO ORDER
2308 012570 005337 015516          DEC     TEMP1
2309 012574 001371          BNE     AVGDAT              ;DONE?
2310 012576 005537 015554          ADC     AVRAGE              ;YES. ADD REMAINDER TO LO ORDER
2311 012602 004537 013664          JSR     S,LEDS
2312 012606 015554          AVRAGE
2313 012610 000002          RTI
2314
2315 012612 000011          CMPCNT: 11                  ;11 OR 4

```

```

2316          :SUBROUTINE TO CALCULATE THE PLUS & MINUS 5 COUNT LIMITS FROM AN AVERAGE
2317
2318 012614 012737 000005 015516 CATORZ: MOV      #5,TEMP1
2319 012622 013737 015554 015520      MOV      AVERAGE,TEMP2      :MOV AVER. TO WORK AREA
2320 012630 012703 015556          :MOV      #AVERP1,R3          :SETUP DISTRIBUTION TABLE (POS.)
2321 012634 005237 015520 FILE1:  INC      TEMP2          :A=A+1
2322 012640 013723 015520      MOV      TEMP2,(R3)+      :SAVE A+1
2323 012644 005337 015516      DEC      TEMP1          :SAVED '5' COUNTS?
2324 012650 001371          BNE      FILE1          :BRANCH IF NO
2325          :SET UP TABLE OF AVG. -1 TO -5
2326 012652 012737 000005 015516      MOV      #5,TEMP1
2327 012660 013737 015554 015520      MOV      AVERAGE,TEMP2      :MOV AVG. TO WORK AREA.
2328 012666 012703 015554          :MOV      #AVERAGE,R3          :SET UP DISTRIBUTION TABLE NEG.
2329 012672 005337 015520 FILE2:  DEC      TEMP2          :A=1-1
2330 012676 013743 015520      MOV      TEMP2,-(R3)      :SAVE 'A-1'
2331 012702 005337 015516      DEC      TEMP1          :SAVED '5' COUNTS?
2332 012706 001371          BNE      FILE2          :BRANCH IF NO
2333
2334          :CATEGORIZE THE COUNT SPREAD AS '+6 & -6' COUNTS FROM THE AVERAGE
2335
2336 012710 012703 015570 CATR1:  MOV      #ORLOW,R3          :CLEAR COUNTS
2337 012714 005023          CLR      (R3)+
2338 012716 022703 015622          CMP      #ORHIGH+2,R3      :FINISHED?
2339 012722 001374          BNE      CATR1          :NO. CLEAR NEXT COUNTER
2340 012724 012737 001001 015516      MOV      #1001,TEMP1      :COMPARE '512' COUNTS
2341 012732 012700 015654          MOV      #ADBUFF,R0      :SET UP A/C BUFFER
2342 012736 005337 015516 CATR2:  DEC      TEMP1
2343 012742 001437          BEQ      CATR5          :EXIT IF '0'
2344 012744 012037 015520          MOV      (R0)+,TEMP2
2345 012750 023737 015566 015520      CMP      AVERP5,TEMP2
2346 012756 100423          BMI      OVRHI
2347 012760 023737 015520 015542      CMP      TEMP2,AVERM5
2348 012766 100422          BMI      OVRLO
2349 012770 005001          CLR      R1
2350 012772 012702 015542 CATR3:  MOV      #AVERM5,R2
2351 012776 022237 015520          CMP      (R2)+,TEMP2
2352 013002 001405          BEQ      CATR4
2353 013004 005201          INC      R1
2354 013006 022701 000013          CMP      #13,R1
2355 013012 001371          BNE      CATR3
2356 013014 000000          HALT
2357 013016 006301 CATR4:  ASL      R1          :FATAL ERROR
2358 013020 005261 015572          INC      MINUS5(R1)      :MULTIPLY 'OFFSET' X2
2359 013024 000744          BR      CATR2
2360 013026 005237 015620 OVRHI:  INC      ORHIGH
2361 013032 000741          BR      CATR2
2362 013034 005237 015570 OVRLO:  INC      ORLOW
2363 013040 000736          BR      CATR2

```

```

2364
2365           ;ADD THE COUNTS AND SAVE TOTAL IN SPREADS OF '1-4'
2366
2367 013042 013737 015604 015622 CATRS: MOV     AVGCNT,XSPRD1
2368 013050 063737 015606 015622      ADD     PLUS1,XSPRD1
2369 013056 063737 015602 015622      ADD     MINUS1,XSPRD1      ;=TO NO. COUNTS AT SPREAD OF '1'
2370 013064 013737 015622 015624      MOV     XSPRD1,XSPRD2
2371 013072 063737 015610 015624      ADD     PLUS2,XSPRD2
2372 013100 063737 015600 015624      ADD     MINUS2,XSPRD2     ;=TO NO. COUNTS AT SPREAD OF '2'
2373 013106 013737 015624 015626      MOV     XSPRD2,XSPRD3
2374 013114 063737 015612 015626      ADD     PLUS3,XSPRD3
2375 013122 063737 015576 015626      ADD     MINUS3,XSPRD3     ;=TO NO. COUNTS AT SPREAD OF '3'
2376 013130 013737 015626 015630      MOV     XSPRD3,XSPRD4
2377 013136 063737 015614 015630      ADD     PLUS4,XSPRD4
2378 013144 063737 015574 015630      ADD     MINUS4,XSPRD4     ;=TO NO. COUNTS AT SPREAD OF '4'
2379 013152 000002      RTI      ;EXIT
2380
2381           ;SUBROUTINE TO TYPEOUT A '6' DIGIT OCTAL NO. THE 'PC' CONTAINS
2382           ;THE ADDRESS OF 'WORD' TO BE TYPED
2383
2384 013154 017605 000000      OCTPRT: MOV     @ (SP),R5      ;THE ADDRESS OF WORD TO BE TYPED
2385 013160 062716 000002      ADD     #2,(SP)      ;SET UP STACK TO EXIT
2386 013164 012737 000006 013264      MOV     #6,10$      ;LOAD COUNTER
2387 013172 012737 000376 013270      MOV     #376,MASK    ;MASK FOR FIRST BIT
2388 013200 000401      BR      .+4
2389 013202 006115      1$:     ROL     (R5)
2390 013204 006115      ROL     (R5)
2391 013206 006115      ROL     (R5)
2392 013210 111537 013266      MOVB    (R5),11$
2393 013214 143737 013270 013266      BICB    MASK,11$
2394 013222 052737 000260 013266      BIS     #260,11$
2395 013230 132777 000200 165566      BITB    #200,@TPS
2396 013236 100374      BPL     .-6      ;WAIT FOR PRINTER READY
2397 013240 113777 013266 165560      MOVB    11$,@TPB     ;PRINT CHAR.
2398 013246 012737 000370 013270      MOV     #370,MASK    ;MASK FOR NEXT '5' DIGITS
2399 013254 005337 013264      DEC     10$
2400 013260 001350      BNE     1$
2401 013262 000002      RTI
2402 013264 000000      10$:    0
2403 013266 000000      11$:    0
2404 013270 000376      MASK:   376
2405
2406           ;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET
2407
2408 013272 105777 165522      TKSFLG: TSTB    @TKS      ;FLAG SET?
2409 013276 100001      BPL     .+4      ;NO, EXIT
2410 013300 104011      TTYIN   ;YES, INQUIRE
2411
2412
2413 013302 000002      RTI
2414           ;ENTERED WITH SYSTEM TRAP CALL (ERROR)
2415 013304 104013      LOGERR: TSTKS
2416 013306 037727 165516 020000      BIT     @SWR,#20000    ;TEST FOR KEYBOARD INTERRUPT
2417 013314 001067      BNE     CK      ;TEST FOR INHIBIT PRINT OUT
2418 013316 012737 000002 000006      MOV     #RTI,@#6      ;INHIBIT,CHECK FOR HALT
2419 013324 011637 015462      MOV     (SP),KSTOR3   ;SET UP FOR BUSS ERROR
                        ;PC OF FAILING ROUTINE

```

2420	013330	162737	000002	015462	SUB	#2,KSTOR3	
2421	013336	004537	013664		JSR	5,LEDS	
2422	013342	015462			KSTOR3		
2423	013344	017737	165470	015464	MOV	@ADCS,KSTOR4	;SAVE A/D STATUS TO BE TYPED
2424	013352	017737	165466	015466	MOV	@ADDBR,KSTOR5	;SAVE A/D CONVERTED VALUE
2425	013360	017705	165466		MOV	@ADMR,R5	;SAVE DMA REG.
2426	013364	010537	015472		MOV	R5,KSTR12	;SAVE R5 IN KSTR12
2427	013370	013737	015514	015470	MOV	TEMP,KSTR11	;SAVE TEMP
2428	013376	012737	000000	000006	MOV	#0,@#6	;RESET BUSS ERROR
2429	013404	005737	015442		TST	PRINT1	
2430	013410	001006			BNE	LGERR1	
2431	013412	104000			PRINT		
2432	013414	014470			CRLF		
2433	013416	104000			PRINT		
2434	013420	014071			MES!		
2435	013422	005237	015442		INC	PRINT1	
2436	013426	104000			LGERR1: PRINT		;OUTPUT CARRIAGE RETURN AND LINE FEED
2437	013430	014470			CRLF		
2438	013432	104010			PRTOCT		;PRINT FAILING PC+2
2439	013434	015462			KSTOR3		
2440	013436	104007			SPACE		;OUTPUT A SPACE
2441	013440	104010			PRTOCT		;PRINT
2442	013442	015464			KSTOR4		;CONTENTS OF A/D STATUS REG.
2443	013444	104007			SPACE		
2444	013446	104010			PRTOCT		
2445	013450	015466			KSTOR5		
2446	013452	104007			SPACE		
2447	013454	104010			PRTOCT		
2448	013456	015472			KSTR12		
2449	013460	104007			SPACE		
2450	013462	104010			PRTOCT		
2451	013464	015470			KSTR11		
2452	013466	105777	165332		TSTB	@TPS	
2453	013472	100375			BPL	.-4	
2454	013474	005777	165330		CK: TST	@SWR	;CHECK SR FOR HALT SWITCH
2455	013500	100001			BPL	."+	;BRANCH IF NOT SET
2456	013502	000000			HALT		;HALT ON ERROR UP
2457	013504	000002			RTI		;RETURN TO MAIN LINE

013526
013510
013516
013520
013526
013530
013534
013536
013544
013548
013552
013560
013564
013570
013574
013578
013602
013604
013612
013614
013622
013624
013630
013632
013640

013642
013644
013652
013654

104013
032777
001015
032777
001016
005737
001413
023737
001407
005237
022606
012677
000177
000014
005037
013576
011637
000002
000000
002042

104013
032777
001357
032777
001360
005737
001755
023727
000741

104013
032777
001337
000742

:SCOPE AND/OR ITERATION LOOP FOR SOME TEST 4000 TIMES

```
SCOPEC: TSTTKS          :TEST FOR KEYBOARD INIT
          BIT            #40000,JSWR  :TEST SR FOR SCOPE
          BNE           SCOPEB       :YES SCOPE
          BIT            #4000,JSWR   :NO-TEST FOR ITERATION
          BNE           SCOPEC       :INHIBIT ITERATION
          TST           PASSCT       :TEST IF FIRST PASS
          BEQ           SCOPEG       :BR IF FIRST PASS -- QUICK PASS
          CMP           SCOPEF,ICOUNT :COMPARE CURRENT COUNT TO MAX NUMBER
SCOPEJ: BEQ           SCOPEJ         :EXIT-DONE
          INC           SCOPEF        :INCREMENT COUNT
SCOPEB: CMP           (6)+,SP        :REPOSITION STACK
          MOV           (6)+,JPSW     :RESTORE PREVIOUS PROCESSOR STATUS
SCOPEG: JMP           JRETURN        :REPEAT TEST
          CLR           SCOPEF        :CLEAR COUNT
          MOV           JSP,RETURN    :SAVE SCOPE RETURN POINTER
          RTI                    :RETURN INLINE-NEXT TEST
SCOPEF: 0                    :COUNT LOCATION FOR ITERATION LOOP
RETURN: ADT1+2              :ADDRESS OF LAST TEST
```

:SCOPE AND/OR ITERATION LOOP FOR SOME TESTS 10 TIMES

```
SCOPEH: TSTTKS          :TEST IF FIRST PASS
          BIT            #40000,JSWR  :BR IF YES
          BNE           SCOPEB
          BIT            #4000,JSWR
          BNE           SCOPEG
          TST           PASSCT
          BEQ           SCOPEG
          CMP           SCOPEF,#10
          BR            SCOPEJ
```

:SCOPE LOOP FOR SOME TEST 1 TIME

```
SCOPEI: TSTTKS          :TEST KEYBOARD
          BIT            #40000,JSWR
          BNE           SCOPEB
          BR            SCOPEG
```

```

013656 104000
013660 013772
013662 000207

013664 012737 000006 013762
013672 005037 013770
013676 013537 013764
013702 013737 013764 013756
013710 042737 177770 013766
013716 113737 013770 013767
013724 013777 013766 155112
013732 006237 013764
013736 006237 013764
013742 006237 013764
013746 005237 013770
013752 005337 013762
013756 001351
013760 000205

013762 000006
013764 000000
013766 000000
013770 000000

013772 007 007
013774 042445 042116 050040
014002 051501 020123 020040
014010 100

```

```

BELL: PRINT
ENDPAS
RTS PC :EXIT

:LOAD THE LPS DISPLAY LIGHTS

LEDS: MOV #6,CNTLED
CLR LEDSV3
MOV #5+,LEDSV1
LEDSA: MOV LEDSV1,LEDSV2
BIC #177770,LEDSV2
MOVB LEDSV3,LEDSV2+1
MOV LEDSV2,JACDDBR
ASR LEDSV1
ASR LEDSV1
ASR LEDSV1
INC LEDSV3
DEC CNTLED
BNE LEDSA
RTS 5

CNTLED: 6
LEDSV1: 0
LEDSV2: 0
LEDSV3: 0

ENDPAS: .BYTE 7,7
.ASCII ".END PASS ."

```

2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584

014011 000
014012 022445 046045 051520
014020 042040 040511 047107
014026 051517 044524 020103
014034 042524 052123 044440
014042 020054 046450 044501
014050 042116 041505 030455
014056 026461 055104 050114
014064 026503 024503 100

014071 040 050040 020103
014076 020040
014100 042101 052123 052101
014106 040
014107 101 041104 043125
014114 020106
014116 040440 042104 040515
014124 040

014125 040 042524 050115
014132 100

014133 045 054524 042520
014140 046040 052105 042524
014146 020122 020047 020047
014154 047524 051040 047125
014162 042040 051505 051111
014170 042105 052040 051505
014176 035124 045
014201 047 023501 040475
014206 052040 020117 020104
014214 047514 044507 020103
014222 042524 052123 045
014227 047 023502 040475
014234 052040 020117 020104
014242 046504 020101 047514
014250 044507 020103 042524
014256 052123 045
014261 047 023503 040475
014266 052040 020117 020104
014274 040503 044514 051102
014302 052101 047511 022516
014310 042047 036447 020101
014316 047524 042040 051040
014324 050105 040505 040524
014332 044502 044514 054524
014340 045
014341 047 023505 040475
014346 052040 020117 020104
014354 042522 047503 042526
014362 054522 045
014365 047 023506 040475
014372 052040 020117 020104
014400 052504 046101 046440

:MESSAGES

BYTE
TITLE: .ASCII '%:ZLPS DIAGNOSTIC TEST I, (MAINDEC-11-DZLPC-C)@'

MES1: .ASCII " PC "
.ASCII "ADSTAT "
.ASCII "ADBUFF "
.ASCII " ADDMA "
.ASCII " TEMP@"

MES4: .ASCII "%:TYPE LETTER ' ' TO RUN DESIRED TEST:%"
.ASCII "'A'=A TO D LOGIC TEST%"
.ASCII "'B'=A TO D DMA LOGIC TEST%"
.ASCII "'C'=A TO D CALIBRATION%"
.ASCII "'D'=A TO D REPEATABILITY%"
.ASCII "'E'=A TO D RECOVERY%"
.ASCII "'F'=A TO D DUAL MODE DYNAMIC LOGIC %@"

2585	014406	042117	020105	054504	
2586	014414	040516	044515	020103	
2587	014422	047514	044507	020103	
2588	014430	040045			
2589					
2590	014432	041536	040045		CNTRLC: .ASCII '1C%2'
2591					
2592	014436	040536	100		CNTRLA: .ASCII '1A2'
2593	014441	136	022507	046117	CNTRLG: .ASCII '1G%OLD SWR = 2'
2594	014446	020104	053523	020122	
2595	014454	020075	100		
2596	014457	040	047040	053505	NEWSWR: .ASCII ' NEW = 2'
2597	014464	036440	040040		
2598					
2599	014470	040045			CRLF: .ASCII '%2'
2600					
2601	014472	027045	100		DOT: .ASCII '%.2'
2602					
2603	014475	077	040040		QMARK: .ASCII '? 2'
2604					
2605	014500	021045	020101	047524	MES5: .ASCII '%A TO D LOGIC TEST% 2'
2606	014506	042040	046040	043517	
2607	014514	041511	052040	051505	
2608	014522	021124	020045	100	
2609					
2610	014527	045	040442	052040	MES7: .ASCII '%A TO D CALIBRATION%2'
2611	014534	020117	020104	040503	
2612	014542	044514	051102	052101	
2613	014550	047511	021116	040045	
2614	014556	021045	020101	047524	MES8: .ASCII '%A TO D RECOVERY2'
2615	014564	042040	051040	041505	
2616	014572	053117	051105	021131	
2617	014600	100			
2618					
2619	014601	045	044103	040040	MES9: .ASCII '%CH 2'
2620					
2621	014606	023445	023505	052130	MES10: .ASCII '%E'XT. OR 'I'NT. OR 'C'LOCK. SYNC' 2"
2622	014614	020056	051117	023440	
2623	014622	023511	052116	020056	
2624	014630	051117	023440	023503	
2625	014636	047514	045503	020056	
2626	014644	054523	041516	020077	
2627	014652	100			
2628					
2629					
2630	014653	045	040442	052040	MES11: .ASCII '%A TO D DMA LOGIC TEST% 2'
2631	014660	020117	020104	046504	
2632	014666	020101	047514	044507	
2633	014674	020103	042524	052123	
2634	014702	022442	040040		
2635	014706	051105	047522	020122	MES12: .ASCII 'ERROR BIT SET!%2'
2636	014714	044502	020124	042523	
2637	014722	020524	040045		
2638					
2639	014726	021045	020101	047524	MES13: .ASCII '%A TO D REPEATIBILITY2'
2640	014734	042040	051040	050105	

2641	014742	040505	044524	044502	
2642	014750	044514	054524	040042	
2643					
2644	014756	022445	044103	024056	MES14: .ASCII '%CH.(S)? 3'
2645	014764	024523	020077	100	
2646	014771	045	040442	052040	MES15: .ASCII '%A TO D DUAL MODE"%3'
2647	014776	020117	020104	052504	
2648	015004	046101	046440	042117	
2649	015012	021105	040045		
2650					
2651	015016	047503	047125	020124	MES16: .ASCII 'COUNT SPREAD? 3'
2652	015024	050123	042522	042101	
2653	015032	020077	100		
2654	015035	045	044447	047047	MES18: .ASCII "%I'INTERNAL OR 'B'URST OR 'D'UAL SAMPLE MODE? 3"
2655	015042	042524	047122	046101	
2656	015050	047440	020122	041047	
2657	015056	052447	051522	020124	
2658	015064	051117	023440	023504	
2659	015072	040525	020114	040523	
2660	015100	050115	042514	046440	
2661	015106	042117	037505	040040	
2662	015114	020045	041440	027110	MES19: .ASCII '% CH. LO AV HI3'
2663	015122	020040	020040	047514	
2664	015130	020040	020040	040440	
2665	015136	020126	020040	020040	
2666	015144	044510	100		
2667					
2668	015147	045	020040	047514	MES20: .ASCII '% LO -5 -4 -3 -2 -1 AV +1 +2 +3 +4 +5 HI3'
2669	015154	020040	032455	020040	
2670	015162	032055	020040	031455	
2671	015170	020040	031055	020040	
2672	015176	030455	020040	053101	
2673	015204	020040	030453	020040	
2674	015212	031053	020040	031453	
2675	015220	020040	032053	020040	
2676	015226	032453	020040	044510	
2677	015234	100			
2678					
2679	015235	045	047520	042527	MES21: .ASCII '%POWER FAILURE 3'
2680	015242	020122	040506	046111	
2681	015250	051125	020105	100	
2682	015255	111	050116	052125	MES22: .ASCII 'INPUT CHANNEL PAIR HAVE THE SAME INPUT VALUE'.3'
2683	015262	041440	040510	047116	
2684	015270	046105	050040	044501	
2685	015276	020122	040510	042526	
2686	015304	052040	042510	051440	
2687	015312	046501	020105	047111	
2688	015320	052520	020124	040526	
2689	015326	052514	022505	100	
2690	015333	105	051122	051117	MES23: .ASCIZ 'ERROR - FIRST CONVERSION EQUAL TO THE SECOND'.3'
2691	015340	026440	043040	051111	
2692	015346	052123	041440	047117	
2693	015354	042526	051522	047511	
2694	015362	020116	050505	040525	
2695	015370	020114	047524	052040	
2696	015376	042510	051440	041505	

2697	015404	047117	022504	000100
2698	015412	051461	020124	047503
2699	015420	053116	020056	047516
2700	015426	020124	020075	047062
2701	015434	022504	000100	
2702				
2703				
2704	015440	000000		
2705	015442	000000		
2706	015444	001000		
2707	015446	001420		
2708	015450	000000		
2709	015452	000000		
2710	015454	000000		
2711	015456	000000		
2712	015460	000000		
2713	015462	000000		
2714	015464	000000		
2715	015466	000000		
2716	015470	000000		
2717	015472	000000		
2718	015474	000000		
2719	015476	000000		
2720	015500	000000		
2721	015502	000000		
2722	015504	000000		
2723	015506	000000		
2724	015510	000000		
2725	015512	000000		
2726	015514	000000		
2727	015516	000000		
2728	015520	000000		
2729	015522	000000		
2730	015524	000000		
2731	015526	000000		
2732	015530	000000		
2733	015532	000000		
2734	015534	000000		
2735	015536	000000		
2736	015540	000000		
2737	015542	000000		
2738	015544	000000		
2739	015546	000000		
2740	015550	000000		
2741	015552	000000		
2742	015554	000000		
2743	015556	000000		
2744	015560	000000		
2745	015562	000000		
2746	015564	000000		
2747	015566	000000		
2748	015570	000000		
2749	015572	000000		
2750	015574	000000		
2751	015576	000000		
2752	015600	000000		

MES24: .ASCIZ '1ST CONV. NOT = 2ND%Q'

.EVEN
: ADDRESS AND CONSTANTS TABLE

```

ADWORD: 0 ;LOW BYTE OF 'ADWORD'
PRINT1: 0
STACK: 1000 ;INITIAL SP. ADDRESS
AVECTR: INITA ;'1A' VECTOR ADDRESS
PROC: 0 ;TEMP STORAGE FOR 'PSW'
CHRCNT: 0 ;TEMP STORAGE
COUNT: 0 ;TEMP STORAGE
KSTOR1: 0 ;PERMANENT STORAGE
KSTOR2: 0 ;PERMANENT STORAGE
KSTOR3: 0 ;PERMANENT STORAGE
KSTOR4: 0 ;PERMANENT STORAGE
KSTOR5: 0
KSTR11: 0
KSTR12: 0
STCHAN: 0
FIRST: 0
DELAY: 0
DELAY1: 0
USEDSH: 0
USEDMA: 0
USECLK: 0
CPS1: 0
TEMP: 0
TEMP1: 0 ;TEMPORARY STORAGE
TEMP2: 0 ;TEMPORARY STORAGE
TEMP3: 0 ;TEMPORARY STORAGE
BRLEV1: 0
BRLEV2: 0
BRLEV3: 0
MESPRT: 0
HIORDV: 0
HIGH: 0
LOW: 0
AVERM5: 0
AVERM4: 0
AVERM3: 0
AVERM2: 0
AVERM1: 0
AVRAGE: 0
AVERP1: 0
AVERP2: 0
AVERP3: 0
AVERP4: 0
AVERP5: 0
ORLOW: 0
MINUS5: 0
MINUS4: 0
MINUS3: 0
MINUS2: 0

```

```

2753 015602 000000
2754 015604 000000
2755 015606 000000
2756 015610 000000
2757 015612 000000
2758 015614 000000
2759 015616 000000
2760 015620 000000
2761 015622 000000
2762 015624 000000
2763 015626 000000
2764 015630 000000
2765 015632 000000
2766 015634 000000
2767 015636 000000
2768 015640 015642
2769 015642 023420
2770 015644 001750
2771 015646 000144
2772 015650 000012
2773 015652 000001
2774
2775 015654 000000
2776
2777 015716 015716
2778 015716 000000
2779 015760 015760
2780 015760 000000
2781 016022 016022
2782 016022 000000
2783 016064 016064
2784 016064 000000
2785 016126 016126
2786 016126 000000
2787 016170 016170
2788 016170 000000
2789 016232 016232
2790 016232 000000
2791 016274 016274
2792 016274 000000
2793 016336 016336
2794 0101060

```

```

MINUS1: 0
AVGCNT: 0
PLUS1: 0
PLUS2: 0
PLUS3: 0
PLUS4: 0
PLUS5: 0
ORHIGH: 0
XSPRD1: 0
XSPRD2: 0
XSPRD3: 0
XSPRD4: 0
DIAB: 0
DIAC: 0
DIAD: 0
      .+2
      10000.
      1000.
      100.
      10.
      1.
:HERE STARTS THE '512' WORD A/D DATA BUFFER.
ADBUFF: 0
ADTB0: 0 .+.40
ADTB1: 0 .+.40
ADTB2: 0 .+.40
ADTB3: 0 .+.40
ADTB4: 0 .+.40
ADTB5: 0 .+.40
ADTB6: 0 .+.40
ADTB7: 0 .+.40
      .END      INIT

```


J05

ADT25	003364	1043#												
ADT25A	003436	1047	1053#											
ADT26	003444	1052	1058#											
ADT26A	003516	1061	1068#											
ADT26B	003520	1067	1069#											
ADT27	003532	1074#												
ADT3	002134	795#												
ADT30	003554	1083#												
ADT31	003576	1092#												
ADT32	003634	1104#												
ADT4	002172	808#												
ADT5	002230	820#												
ADT6	002266	832#												
ADT7	002324	844#												
ADWRD2	015440	1483*	1484*	1488*	1492*	1495*	1572*	1573*	1574*	1678*	1679*	1680*	1683*	1684*
		1685*	2063	2065	2072	2074	2110*	2111	2704*					
		591#	1645	1848										
AMASK	001010	1515#												
ARESET	006234	700*	1467*	1534*	1661*	1743*	1958	2707*						
ALECTR	015446	1795*	1799	1819	2741#									
AVERM1	015552	1792*	1798	1815	2740#									
AVERM2	015550	1763*	1767	1818	2739#									
AVERM3	015546	1758*	1766	1814	2738#									
AVERM4	015544	2347	2350	2737#										
AVERM5	015542	2320	2743#											
AVERP1	015556	2744#												
AVERP2	015560	2745#												
AVERP3	015562	2746#												
AVERP4	015564	2345	2747#											
AVERP5	015566	1633	2367	2754#										
AVGCNT	015604	2306#	2309											
AVGDAT	012560	1900*	1902#	1903*										
AVGTAB	010364	1578	1624	1710	2291*	2292	2293	2301*	2307*	2310*	2312	2319	2327	2328
AVERAGE	015554	2742#												
BCDBIN	011050	2016#	2048	2197										
BCDBN1	011064	2019#	2043											
BCDBN2	011070	2021#	2037											
BCDEND	011146	2022	2025	2038#										
BCDERR	011170	2027	2029	2039	2045#									
BCDTAB	011202	1545	1547	1549	1551	1556	1670	1671	2017	2018*	2049#			
BELL	013656	1145	1449	2499#										
BINDEC=	104006	578#	1639											
BIT0 =	000001	508#	943	953	961	968	1033	1048	1063	1093				
BIT1 =	000002	507#	772	773	1105	1325								
BIT10 =	002000	498#	869	870	1075									
BIT11 =	004000	497#	881	882	1075	1127								
BIT12 =	010000	496#	893	894	1075	1254	1255	1326	1348	1394	2109			
BIT13 =	020000	495#	905	906	1075	1264	1265	1769	1801	1828	2081			
BIT14 =	040000	494#	917	918	1274	1275	1787							
BIT15 =	100000	493#	929	930	1096									
BIT2 =	000004	506#	784	785	1105									
BIT3 =	000010	505#	796	797	1105	2110								
BIT4 =	000020	504#	809	810	990	1084								
BIT5 =	000040	503#	710	821	822	1003	1084	1473						
BIT6 =	000100	502#	762	833	834	1033	1048	1063	1084	1120	1158	1320	1321	1327
		1338	1421	1567	1751	1918	1957	1982						

L05

DEC2	012370	2268	2271*											
DELAY	015500	1681*	1686*	2060	2069*	2720*								
DELAY1	015502	964*	965*	966*	967*	969*	975*	976*	2066*	2067*	2721*			
DIAB	015632	2765*												
DIAB	015634	2766*												
DIAC	015636	2767*												
DIAD	015640	2768*												
DIGCNT	012410	2249*	2258*	2265	2277*									
DIGIT	012406	2252*	2253*	2263	2267*	2270*	2273	2276*						
DISPLA	001032	601*	645*	759*	1160*	1513*	1578*	1753*						
DMACVT	011436	2071	2101*											
DMATST	004050	565	716	1152*										
DMATO	004120	1161	1169*											
DMATI	004154	1179*												
DMATI0	004460	1251*												
DMATI1	004514	1261*												
DMATI2	004550	1271*												
DMATI3	004604	1281*												
DMATI4	004646	1293*												
DMATI5	004710	1306*												
DMATI6	004752	1318*												
DMATI7	005032	1336*												
DMAT2	004210	1189*												
DMAT20	005272	1373	1380*											
DMAT21	005532	1410	1419*											
DMAT3	004244	1200*												
DMAT4	004300	1210*												
DMAT5	004334	1220*												
DMAT6	004370	1230*												
DMAT7	004424	1240*												
DMA00	004062	1157*	1451											
DMA01	004054	738	1154*											
DMK0	005734	1341*	1370	1375*	1385*	1413*	1422*	1454*						
DMK1	005736	1386*	1405	1412*	1455*									
DMK2	005740	1387*	1400	1411*	1456*	2062*	2053							
DMT17A	005264	1339	1375*											
DMT20A	005412	1395*	1414											
DOT	014472	708	2601*											
DSHTST	007504	566	728	1743*										
DSHT0	007512	1743	1744*											
DSHT1	007516	1746*												
DSHT2	007522	1747*												
DSHT3	007526	1748*	1823											
DSHT4	007664	1770	1775*											
DSHT6	007674	1768	1776	1786*										
DSHT6A	010002	1802	1807*											
DSHT7	010014	1800	1813*											
DSHT7A	010040	1808	1810	1821*	1837									
DSHT8	010064	1816	1820	1828*										
DSHT8A	010104	1829	1834*											
DSHT8B	010114	1826	1835	1837*										
EMTOK	012016	2186	2188*											
EMTSRV	011776	551	2182*											
EMTTAB	012036	2190	2196*											
ENOPAS	013772	2500	2525*											
ERROR =	104400	571*	775	779	797	791	799	803	812	816	824	828	936	940

TEMP2	015520	1122*	1124	1133*	2294*	2295	2297	2298	2300	2301	2319*	2321*	2322	2327*
TEMP3	015522	2329*	2330	2344*	2345	2347	2351	2728*						
TEST1	010216	1123*	1134*	1503*	1676*	1906*	2729*							
TEST15	003656	1114*												
TEST16	004016	1116	1118	1128	1140*									
FILE	014012	684	2532*											
TKB	00:022	597*	1929											
TKS	001020	596*	681*	692*	706*	762*	1120*	1158*	1338*	1421*	1751*	1918*	1927	1957*
TKSFLG	013272	1982*	2408											
TPE	001036	2207	2408*											
TPS	001024	599*	1891*	1941*	2230*	2232*	2235*	2239*	2273*	2397*				
TSLO	012514	598*	1889	1939	2223	2233	2237	2271	2395	2452				
TSTA	010226	2296	2298*											
TSTCT1	006630	1967*	1874											
TSTCT2	006606	1595	1599*											
TSTCT3	006564	1589	1594*											
TSTCT4	006532	1584	1588*											
TSTTKS=	104013	1581*												
TSTXB	010262	583*	1514	1516	1570	1609	1673	1719	1825	2415	2461	2482	2494	
TST154	003712	1870	1876*											
TST158	003726	1122*	1138											
TST15C	003756	1124*	1135											
TYIN =	104011	1129*	1130	1132										
TYPECL	012150	581*	709	1472	1543	1554	1559	1668	1992	2047	2410			
TYPERA	012116	2229	2232*											
TYPER1	012134	2223*	2224	2231	2243									
TYPER2	012142	2226	2229*											
TYPER3	012102	2230*												
TYPT1	012256	2196	2220*											
TYPT2	012264	2252*	2262											
TYPT3	012316	2253*	2255											
USECLK	015510	2259	2261*											
USEDMA	015506	1480*	1493*	1496	2724*									
USEDSH	015504	1479*	1538*	1562*	1662*	2070	2723*							
WHAT	001674	1481*	1539*	1566*	1614	1663*	2099	2722*						
WHATA	001716	705	732*	746										
WHATB	001730	736*	1144											
WHATI	001754	739*	1448											
XLOOP	010304	733	747*											
XPARTAV	010354	1880*	1881											
XPTAI	010362	1900*	2206											
XSPACE	010322	1901*	1907											
XSPRD1	015622	1889*	1893	2203										
XSPRD2	015624	1601	1640	2367*	2368*	2369*	2370	2761*						
XSPRD3	015626	1596	2370*	2371*	2372*	2373	2762*							
XSPRD4	015630	1590	2373*	2374*	2375*	2376	2763*							
X*TYIN	010414	1585	2376*	2377*	2378*	2764*								
ZERO	012412	548	1911*	2008	2205									
.	= 016336	2251*	2267	2269*	2278*									
		541*	544*	547*	550*	555*	557*	559*	586*	679	704	712	715	718
		721	724	727	737	774	778	786	790	798	902	911	915	918
		827	835	839	847	851	859	863	871	875	983	997	999	999
		907	911	919	923	931	935	951	954	972	982	993	995	1008
		1013	1016	1078	1087	1095	1099	1108	1143	1174	1184	1194	1205	1215
		1225	1235	1245	1256	1266	1276	1288	1300	1313	1322	1329	1357	1362

D06

PS-11 DIAGNOSTIC TEST I MAINDEC-11-DZLPC-C MACY11 27(1006) 16-SEP-76 21:50 PAGE 69
DZLPC.P11 04-APR-76 00:00 CROSS REFERENCE TABLE -- USER SYMBOLS

1367	1371	1402	1408	1436	1441	1447	1477	1757	1762	1790	1794	1877
1990	2012*	2131	2136	2141	2234	2272	2279	2389	2396	2409	2453	2455
2768	2777*	2779*	2781*	2783*	2785*	2787*	2789*	2791*	2793*			

. ABS. 016336 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZLPC DZLPC.SEO SOL CRF DS:ERFZ EN:ABS=DSKM:DZLPC.P11
RUN-TIME: 10 19 2 SECONDS
RUN-TIME RATIO: 365 33=10.9
CORE USED: 9% (17 PAGES)

