

.REM ..

IDENTIFICATION

PRODUCT CODE: AC-E730F-MC
PRODUCT NAME: CXLPDF0 LPS-AD,NP MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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 DIGITALS 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) 1973,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT:

LPD IS AN INTERRUPT IOMOD THAT EXERCISES THE LPS-AD OR LPS-NP A TO D CONTROL. A CONVERSION IS TAKEN ON EACH CHANNEL AND STORED IN TABLE A. THEN REPEATED AND STORED IN TABLE B. AFTER THE LAST CHANNEL HAS BEEN SAMPLED A SECOND TIME, EACH ENTRY IN TABLE A IS COMPARED TO TABLE B. THE DIFFERENCE MAY VARY TO THE VALUE OF LOCATION (WIDE). IF THE DIFFERENCE IS GREATER, A DATA ERROR WILL BE REPORTED. LOCATION (CHOUTA) CONTAINS THE FAILING CHANNEL.

2. REQUIREMENTS:

HARDWARE: LPS-11 INTERFACE WITH A LPS-AD OR LPS-NP A TO D CONTROLLER INSTALLED.

STORAGE: LPD REQUIRES:

1. DECIMAL WORDS: 811
2. OCTAL WORDS: 1453
3. OCTAL BYTES: 3126

3. PASS DEFINITION:

ONE PASS OF THE LPD MODULE CONSISTS OF SEQUENCING THRU FOUR CHANNELS 512 TIMES FOR A TOTAL CONVERSION COUNT OF 4096. THIS IS REPEATED 50 OCTAL TIMES FOR EACH PASS. THIS CAN BE ALTERED BY CHANGING THE "#50" AT LOCATION TIMES.

4. EXECUTION TIME:

VARIABLES WITH NUMBER OF CHANNELS BUT SHOULD TAKE AN AVERAGE OF ONE MINUTE TO COMPLETE ONE PASS WHEN RUNNING ALONE.

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 170400, VCT: 1, BR1: 6

REQUIRED PARAMETERS:

AT CONFIGURATION TIME THE USER MUST SPECIFY:

VCT: VECTOR ADDRESS OF LPS-AD
SR1: BIT15=1 DMA CONVERSIONS (LPSADNP)
BIT15=0 BR CONVERSIONS (LPSAD12)

6. DEVICE OPTION SETUP:

NONE REQUIRED

7.0 MODULE OPERATION:

7.1 TEST SEQUENCE:

- A. START: USING THE DEVICE ADDRESS, THIS SECTION OF CODE DETERMINES THE CONTROL/STATUS, DATA BUFFER REGISTERS AND VECTOR.
- B. TESTAD: IN THIS CODE, A CONFIDENCE REGISTER TEST IS PERFORMED ON THE BASIC CONTROL/STATUS REGISTER.
- C. TSTCAR: IF THE NPR CONTROLLER WAS SELECTED, A CONFIDENCE TEST OF THE CURRENT ADDRESS REGISTER IS EXECUTED <15 BIT REGISTER>.
- D. TSTWCR: IF THE NPR CONTROLLER WAS SELECTED, THIS SECTION OF CODE EXECUTES A CONFIDENCE TEST OF THE WORD COUNT REGISTER <12 BIT REGISTER>.
- E. TSTEXT: IF THE NPR CONTROLLER WAS SELECTED, THIS SECTION OF CODE EXECUTES A CONFIDENCE TEST OF THE THREE BIT EXTENDED STATUS/CONTROL REGISTER.

- F. PRIME: THIS SECTION TESTS 'SR1' FOR THE TYPE OF A TO D CONTROLLER. AN A TO D CHANNEL IS SELECTED AND THE A TO D IS ENABLED TO INTERRUPT.
- G.0 SMPAD: IF THE SIMPLE
 A TO D CONTROL WAS SELECTED, THE INTERRUPT WILL RETURN HERE AND THE ERROR BIT IS TESTED.
- G.1 SMPAD1: THIS SECTION OF CODE STORES THE CONVERTED VALUE IN ONE OF TWO TABLES. IF THE LAST CHANNEL WAS SAMPLED FOR THE SECOND TIME, A JSR IS EXECUTED TO 'ADCK' FOR DATA COMPARISON.
- H.0 NPRAD: IF THE COMPLEX <NPRG A TO D CONTROL WAS SELECTED, THE INTERRUPT WILL RETURN HERE AND THE ERROR BIT IS TESTED.
- H.1 NPRADA: THIS SECTION IS SIMILAR TO SMPAD1
- I. ADCK: THIS CODE IS THE ACTUAL DATA COMPARISON ROUTINE. EACH ENTRY IN 'TABLE A' IS COMPRED TO 'TABLE B'. THE VALUES MAY VARY IN DIFFERENCE, BEFORE AN ERROR IS REPORTED, BY THE SIZE OF LOCATION 'WIDE'. AFTER ALL CONVERSIONS HAVE BEEN TESTED THE 'LED' DISPLAY IS LOADED.
- J. LOADS: THIS SECTION OF CODE LOADS A SEQUENCE OF NUMBERS (0-37) INTO EACH L.E.D. THIS RESULTS IN THE NUMBERS 0 THRU 9, TEST <ALL BITS>, BLANK <NO BITS>, DASH AND DEC. POINT IN EACH OF THE SIX LED'S.

8. OPERATOR OPTIONS:

- A. LOCATION (WIDE) CAN BE MODIFIED TO VARY THE ALLOWABLE DIFFERENCE BETWEEN CONVERSIONS ON EACH CHANNEL
- B. LOCATION (ENDCH) CAN BE MODIFIED TO VARY THE NUMBER OF A TO D CHANNELS TO BE SAMPLED AND TESTED.
- C. LOCATION (WHYCNT) CAN BE MODIFIED TO INCREASE THE NUMBER OF CONVERSIONS BEFORE TESTING RESULTS, FOR A HIGHER NPR/BR RATE ON THE BUSS. THIS WILL ALSO RESULT IN A LONGER DELAY BEFORE LOADED A NEW PATTERN INTO THE LED DISPLAY.

9. NON-STANDARD PRINTOUTS:

NONE: ALL PRINTOUTS HAVE THE STANDARD FORMATS.

```
216 .LIST
217 ;LPS-11 AD DEC/X11 EXERCISER MODULE
218
219 000000 10MOD <LPDF >,170400,1,6,,50,,50
220 000000 MODULE 140000,LPDF ,170400,1,6,,50,,50
221 .TITLE LPDF DEC/X11 SYSTEM EXERCISER MODULE
222 ; DDXCOM VERSION 6 23-MAY-78
223 .LIST BIN
224 ;*****
225 000000 BEGIN;
226 000000 050114 043104 040 MODNAM; .ASCII /LPDF / ;MODULE NAME.
227 000005 000 XFLAG; .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
228 000006 170400 ADDR; 170400+0 ;1ST DEVICE ADDR.
229 000010 000001 VECTOR; 1+0 ;1ST DEVICE VECTOR.
230 000012 300 BR1; .BYTE PRTY6+0 ;1ST BR LEVEL.
231 000013 000 BR2; .BYTE PRTY+0 ;2ND BR LEVEL.
232 000014 000001 DVID1; +1 ;DEVICE INDICATOR 1.
233 000016 000000 SR1; OPEN ;SWITCH REGISTER 1
234 000020 000000 SR2; OPEN ;SWITCH REGISTER 2
235 000022 000000 SR3; OPEN ;SWITCH REGISTER 3
236 000024 000000 SR4; OPEN ;SWITCH REGISTER 4
237 ;*****
238 000026 140000 STAT; 140000 ;STATUS WORD.
239 000030 000250 INIT; START ;MODULE START ADDR.
240 000032 000224 SPOINT; MODSP ;MODULE STACK POINTER.
241 000034 000000 PASCNT; 0 ;PASS COUNTER.
242 000036 000062 ICOUNT; 50 ;# OF ITERATIONS PER PASS=50.
243 000040 000000 ICOUNT; 0 ;LOC TO COUNT ITERATIONS
244 000042 000000 SOPCNT; 0 ;LOC TO SAVE TOTAL SOFT ERRORS
245 000044 000000 HRDCNT; 0 ;LOC TO SAVE TOTAL HARD ERRORS
246 000046 000000 SOPPAS; 0 ;LOC TO SAVE SOFT ERRORS PER PASS
247 000050 000000 HPDPAS; 0 ;LOC TO SAVE HARD ERRORS PER PASS
248 000052 000000 SYSCNT; 0 ;# OF SYS ERRORS ACCUMULATED
249 000054 000000 RANNUM; 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
250 000056 000000 CONFIG;
251 000056 000000 RES1; 0 ;RESERVED FOR MONITOR USE
252 000060 000000 RES2; 0 ;RESERVED FOR MONITOR USE
253 000062 000000 SVR0; OPEN ;LOC TO SAVE R0.
254 000064 000000 SVR1; OPEN ;LOC TO SAVE R1.
255 000066 000000 SVR2; OPEN ;LOC TO SAVE R2.
256 000070 000000 SVR3; OPEN ;LOC TO SAVE R3.
257 000072 000000 SVR4; OPEN ;LOC TO SAVE R4.
258 000074 000000 SVR5; OPEN ;LOC TO SAVE R5.
259 000076 000000 SVR6; OPEN ;LOC TO SAVE R6.
260 000100 000000 CSRA; OPEN ;ADDR OF CURRENT CSR.
261 000102 000000 SBADR; ;ADDR OF GOOD DATA, OR
262 000102 000000 ACSR; OPEN ;CONTENTS OF CSR.
263 000104 000000 WASADR; ;ADDR OF BAD DATA, OR
264 000104 000000 ASTAT; OPEN ;STATUS REG CONTENTS.
265 000106 000000 ERRTP; ;TYPE OF ERROR
266 000106 000000 ASB; OPEN ;EXPECTED DATA.
267 000110 000000 AWAS; OPEN ;ACTUAL DATA.
268 000112 000264 RSTR; RESTR ;RESTART ADDRESS AFTER END OF PASS
269 000114 000000 WDTO; OPEN ;WORDS TO MEMORY PER ITERATION
270 000116 000000 WDFR; OPEN ;WORDS FROM MEMORY PER ITERATION
```

```
271 000120 000000 INTR; OPEN ;# OF INTERRUPTS PER ITERATION
272 000122 000050 IDNUM; 50 ;MODULE IDENTIFICATION NUMBER=50
273 000040 .REPT SPSIZ ;MODULE STACK STARTS HERE.
274 .NLIST
275 .WORD 0
276 .LIST
277 .ENDR
278 .LIST
279 .LIST
280 .LIST
281 .LIST
282 .LIST
283 .LIST
284 .LIST
285 .LIST
286 .LIST
287 .LIST
288 .LIST
289 .LIST
290 .LIST
291 .LIST
292 .LIST
293 .LIST
294 .LIST
295 .LIST
296 .LIST
297 .LIST
298 .LIST
299 .LIST
300 .LIST
301 .LIST
302 .LIST
303 .LIST
304 .LIST
305 .LIST
306 .LIST
307 .LIST
308 .LIST
309 .LIST
310 000224 MODSP;
311 ;*****
312 .LIST
313
```

```

314
315
316
317
318 000224 001000 ADPASS: 1000 ;NUMBER ON TIMES THRU ALL CHANNELS BEFORE "ENDPASS"
319
320 000226 000004 WIDE: 4 ; + OR - COUNT SPREAD ON AN A TO D CONVERSION
321 000230 000003 ENDCH: 3 ;HIGHEST CHANNEL TO BE TESTED <77
322 000232 000001 WHYCNT: 1 ;COUNTER BEFORE CONVERSIONS ARE TESTED
323
324 ;LPS-AD COMMON DEVICE ADDRESSES
325
326 000234 170400 ADCS: 170400 ;A TO D STATUS REGISTER
327 000236 170402 ADDBR: 170402 ;A TO D BUFFER
328
329 000240 170436 ADMR: 170436 ;NPR A TO D OPTION ADDRESS
330
331 000242 170401 ADCSHB: 170401 ;A TO D STATUS <HIGH BYTE>
332
333 ;LPS-AD COMMON DEVICE INTERRUPT VECTOR
334
335 000244 000300 ADINT: 300 ;A TO D VECTOR
336 000246 000302 ADINT1: 302
337
338
339 ;NOW SET UP THE ADDRESS AND VECTOR DISPATCH LOC.
340 000250 012767 010000 177636 START: MOV #10000,WDT0 ;10000 WDS TO MEM
341 000256 012767 010000 177634 MOV #10000,INTR ;10000 INTERRUPTS/ITERATION
342
343 000264 016767 177516 177742 RESTRT: MOV ADDR,ADCS ;LOAD DEVICE ADDRESS
344 000272 016767 177510 177736 MOV ADDR,ADDBR
345 000300 062767 000002 177730 ADD #2,ADDBR
346 000306 016767 177474 177724 MOV ADDR,ADMR ;LOAD DMA WORD
347 000314 062767 000036 177716 ADD #36,ADMR ;TO CORRECT VALUE
348 000322 016767 177460 177712 MOV ADDR,ADCSHB ;LOAD ADC HIGH BYTE LOC
349 000330 005267 177706 INC ADCSHB ;MAKE HIGH BYTE
350 000334 016767 177450 177702 MOV VECTOR,ADINT
351 000342 016767 177442 177676 MOV VECTOR,ADINT1
352 000350 062767 000002 177670 ADD #2,ADINT1
353 000356 016767 177652 177514 MOV ADCS,CSRA
354 000364 016767 177634 002076 MOV ADPASS,ADPAS
355 000372 012767 002522 002106 MOV #ADTBLA,TABA ; GET TABLE A ADDRESS
356 000400 012767 002724 002106 MOV #ADTBLB,TABB ; GET TABLE B ADDRESS
357 000406 104415 000000 002506 GETPAS,BEGIN,TABA ;GET PHYSICAL ADDRESS FROM 16-BIT TABA
358 000414 104415 000000 002514 GETPAS,BEGIN,TABB ;GET PHYSICAL ADDRESS FROM 16-BIT TABB
359 000422 016700 002064 MOV TABAEA,R0 ; GET EXTENDED MEMORY BITS
360 000426 000300 SWAB R0 ;MOVE TO HIGH BYTE
361 000430 006100 ROL R0 ;MOVE LEFT
362 000432 042700 117777 BIC #11777,R0 ;MASK
363 000436 052700 010000 BIS #BIT12,R0 ;SET DMA ENABLE
364 000442 010067 002016 MOV R0,MVEA ;SAVE EA BITS AND DMA ENABLE

```

```

365 ;LPS-AD LOGIC TEST
366
367
368 000446 005077 177562 TESTAD: CLR @ADCS ;CLEAR STATUS
369 000452 005777 177556 TST @ADCS ;TEST STATUS
370 000456 001025 BNE 18 ;BR IF ERROR
371
372 000460 012777 052524 177546 MOV #52524,@ADCS ;LOAD STATUS
373 000466 012767 052524 177410 MOV #52524,ASTAT ;LOAD POINTER
374 000474 022777 052524 177532 CMP #52524,@ADCS ;TEST STATUS
375 000502 001013 BNE 18 ;BR IF ERROR
376
377 000504 012777 125012 177522 MOV #125012,@ADCS ;LOAD STATUS
378 000512 012767 125012 177364 MOV #125012,ASTAT ;LOAD POINTER
379 000520 022777 125012 177506 CMP #125012,@ADCS ;TEST STATUS
380 000526 001001 BNE 18 ;BR IF ERROR
381 000530 000420 BR PRINTS
382
383 000532 017767 177476 177342 18: MOV @ADCS,ACSR ;READ STATUS
384 000540 016767 177470 177332 MOV ADCS,CSRA ;LOAD ADDRESS
385 000546 005077 177462 CLR @ADCS ;ENSURE CLEAR STATUS
386 000552 012767 000025 177326 MOV #25,ERRRTP ;BIT STUCK IN C/S RFG
387 ;*****
388 HRDRS,BEGIN,NULL ;CONTROL/STATUS BIT IN ERROR
389 ;*****
390 000566 104410 000000 ENDS,REGIN
391

```

```
392
393 000572 005767 177220
394 000576 100402
395 000600 000167 000522
396
397
398
399 000604 005077 177424
400 000610 012777 000006 177416
401 000616 016767 177416 177254
402
403 000624 005077 177410
404 000630 005067 177250
405 000634 005777 177400
406 000640 001037
407
408 000642 012777 125212 177370
409 000650 012767 125212 177226
410 000656 022777 125212 177354
411 000664 001025
412
413 000666 012777 052524 177344
414 000674 012767 052524 177202
415 000702 022777 052524 177330
416 000710 001013
417
418 000712 012777 000001 177320
419 000720 005067 177160
420 000724 005777 177310
421 000730 001003
422
423 000732 005077 177302
424 000736 000417
425
426 000740 017767 177274 177134 181
427 000746 005077 177266
428 000752 005077 177256
429 000756 012767 000025 177122
430
431 000764 104405 000000 000000
432
433 000772 104410 000000
PRIMS: TST SRI ;TEST SRI
      BMI TSTCAR ;BR IF NPR
      JMP PRIME
;TEST THE LPS-NP CURRENT ADDRESS REGISTER
TSTCAR: CLR @ADCS ;CLEAR CONTROL
        MOV #6,@ADCS ;LOAD MODE
        MOV ADMR,CSRA ;LOAD ERROR DEVICE ADDRESS
        CLR @ADMR ;CLEAR C.A.
        CLR ASTAT ;CLEAR POINTER
        TST @ADMR ;TEST C.A.
        BNE 18 ;BR IF ERROR
        MOV #125212,@ADMR ;LOAD C.A.
        MOV #125212,ASTAT ;LOAD POINTER
        CMP #125212,@ADMR ;TEST IT
        BNE 18 ;BR IF ERROR
        MOV #52524,@ADMR ;LOAD C.A.
        MOV #52524,ASTAT ;LOAD POINTER
        CMP #52524,@ADMR ;TEST IT
        BNE 18 ;BR IF ERROR
        MOV #1,@ADMR ;LOAD ODD ADDRESS <NO ODD BIT>
        CLR ASTAT ;CLEAR POINTER
        TST @ADMR ;TEST IT
        BNE 18 ;BR IF ODD ADDRESS
        CLR @ADMR ;CLEAR EXT STATUS
        BR TSTWCR ;TEST W.C. REGISTERS
        MOV @ADMR,ACSR ;READ BAD VALUE
        CLR @ADMR ;CLEAR REGISTER
        CLR @ADCS ;CLEAR CONTROL
        MOV #25,ERRTP ;BIT STUCK IN CUR ADDR REGISTER
        HRDERS,BEGIN,NULL ;CURRENT ADDRESS REGISTER BIT IN ERROR
        ENDS,BEGIN ;
```

```
434
435
436
437
438 000776 005077 177232
439 001002 012777 000004 177224
440 001010 005077 177224
441 001014 005067 177064
442 001020 005777 177214
443 001024 001024
444
445 001026 012777 005252 177204
446 001034 012767 005252 177042
447 001042 022777 005252 177170
448 001050 001012
449
450 001052 012777 002525 177160
451 001060 012767 002525 177016
452 001066 022777 002525 177144
453 001074 001413
454
455 001076 017767 177136 176776 181
456 001104 012767 000025 176774
457
458 001112 104405 000000 000000
459
460 001120 104410 000000
;TEST THE LPS-NP WORD COUNT REGISTER
TSTWCR: CLR @ADCS ;ENSURE CLEAR CONTROL
        MOV #4,@ADCS ;LOAD POINTER TO W.C.
        CLR @ADMR ;CLEAR W.C.
        CLR ASTAT ;LOAD POINTER
        TST @ADMR ;TEST W.C.
        BNE 18 ;BR IF ERROR
        MOV #5252,@ADMR ;LOAD W.C.
        MOV #5252,ASTAT ;LOAD POINTER
        CMP #5252,@ADMR ;TEST W.C.
        BNE 18 ;BR IF ERROR
        MOV @ADMR,ACSR ;READ W.C. VALUE
        MOV #25,ERRTP ;BIT STUCK IN WORD COUNT REGISTER
        HRDERS,BEGIN,NULL ;WORD COUNT REGISTER BIT IN ERROR
        ENDS,BEGIN ;
```



```

461
462
463
464
465 001124* 005077 177104 TSTEXT: CLR @ADCS ;ENSURE CLEAR CONTROL
466 001130* 012777 000002 177076 MOV #2,@ADCS ;LOAD MODE
467 001136* 005077 177176 CLR @ADMR ;CLEAR EXT STATUS
468 001142* 005067 176736 CLR ASTAT ;CLEAR POINTER
469 001146* 005777 177066 TST @ADMR ;TEST IT
470 001152* 001046 BNE 18 ;BR IF ERROR
471
472 001154* 012777 050000 177056 MOV #BIT14|BIT12,@ADMR ;LOAD EXT STATUS
473 001162* 012767 050000 176714 MOV #BIT13,ASTAT ;LOAD POINTER
474 001170* 022777 050000 177042 CMP #BIT14|BIT12,@ADMR ;TEST IT
475 001176* 001034 BNE 18 ;BR IF ERROR
476
477 001200* 012777 020000 177032 MOV #BIT13,@ADMR ;LOAD EXT STATUS
478 001206* 012767 020000 176670 MOV #BIT13,ASTAT ;LOAD POINTER
479 001214* 022777 020000 177016 CMP #BIT13,@ADMR ;TEST IT
480 001222* 001022 BNE 18 ;BR IF ERROR
481
482 001224* 012777 000100 177002 MOV #100,@ADCS ;LOAD INT ENABLE
483 001232* 052777 000002 176774 BTS #2,@ADCS ;SET MODE
484 001240* 012777 010000 176772 MOV #BIT12,@ADMR ;ENABLE DMA
485 001246* 032777 000100 176760 BIT #100,@ADCS ;DMA ENR. SHOULD CLEAR INT ENABLE
486 001254* 001005 BNE 18 ;BR IF BIT SIX SET
487
488 001256* 005077 176756 CLR @ADMR ;CLEAR EXT STATUS
489 001262* 005077 176746 CLR @ADCS ;CLEAR CONTROL
490 001266* 000417 BR PRIME ;START TEST
491
492 001270* 017767 176744 176604 181 MOV @ADMR,ACSR ;READ VALUE
493 001276* 005077 176736 CLR @ADMR ;CLEAR EXT STATUS
494 001302* 005077 176726 CLR @ADCS ;ENSURE CLEAR CONTROL
495 001306* 012767 000025 176572 MOV #25,ERRTP ;BIT STUCK IN EXT. STAT REGISTER
496 ;*****
497 HRDRS,REGIN,NULL ;EXTENDED STATUS REGISTER BIT IN ERROR
498 ;*****
499 ENDS,REGIN ;
500

```

```

501
502
503
504
505 001326* 005077 176702 PRIME: CLR @ADCS
506 001332* 005767 176400 TST SRI ;TEST BIT 15
507 001336* 100026 HPL TSMPAD ;BRANCH IF TEST SIMPLE A TO D
508 001340* 012777 001712* 176676 MOV #NPRAD,@ADINT ;SET UP NPR AD VECTOR
509 001346* 012777 000006 176660 MOV #6,@ADCS ;LOAD CA POINTER
510 001354* 016777 001130 176656 MOV TABAPA,@ADMR ;LOAD CA REGISTER
511 001362* 012777 000004 176644 MOV #4,@ADCS ;LOAD WC POINTER
512 001370* 012777 177777 176642 MOV #-1,@ADMR ;LOAD WC REGISTER
513 001376* 012777 000002 176630 MOV #2,@ADCS ;LOAD STATUS POINTER
514 001404* 016777 001054 176626 MOV MYEA,@ADMR ;LOAD EXT. STATUS REGISTER
515 001412* 000406 HR LPAD
516 001414* 012777 001464* 176622 TSMPAD: MOV #SMPAD,@ADINT ;SET UP SIMPLE AD VECTOR
517 001422* 052777 000100 176604 HIS #BIT6,@ADCS ;ENABLE INTERRUPT
518 001430* 116777 176356 176610 LPAD: MOVR BRI,@ADINT1
519 001436* 005067 001136 CLR CHOUT
520 001442* 005067 001021 CLR ADRB ;CLEAR BUFFER POINTER
521 001446* 012767 002522* 001020 MOV #ADTBIA,TRPTR ;LOAD TABLE POINTER
522 001454* 005277 176554 INC @ADCS ;START A TO D
523
524 001460* 104400 000000* EXITS,REGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
525

```

```

526
527
528
529
530
531 001464* SMPADI
532
533 001464* 000004 000000* 001472*
534
535 001472* 005777 176536 18: TST @ADCS ;TEST A TO D STATUS
536 001476* 100020 BPL SMPADI
537 001500* 017767 176530 176374 MOV @ADCS,ACSR ;LOAD VALUE
538 001506* 005077 176522 CLR @ADCS ;CLEAR STATUS
539 001512* 017767 176520 000756 MOV @ADDBR,ADTEMP
540 001520* 012767 000025 176360 MOV #2,ERRTYP ;DATA LATE
541
542 001526* 104406 000000* 000000
543
544 001534* 104410 000000*
545
546 001540* 017777 176472 000726 SMPADI: MOV @ADDBR,TBPTR ;READ THE DATA INTO A BUFFER
547 001546* 026767 176456 000724 CMP ENDCH,CHOUT ;TEST FOR LAST CHANNEL
548 001554* 001031 BNE AD2 ;BRANCH IF NOT
549 001556* 005767 000710 TST AORB ;TEST BUFFER
550 001562* 001011 BNE AD1
551 001564* 012767 002724* 000702 MOV #ADTB, TBPTR ;END OF A, RESET TO B
552 001572* 012767 177777 000672 MOV #1,AORB
553 001600* 005067 000674 CLR CHOUT
554 001604* 000422 BR AD3
555
556 001606* 004767 000366 AD1: JSR PC,ADCK ;TEST THE DATA
557 001612* 012767 002522* 000654 MOV #ADTB, TBPTR ;RESET TO A BUFFER
558 001620* 005067 000654 CLR CHOUT ;CLEAR STARTING CHANNEL
559 001624* 005067 000642 CLR AORB
560 001630* 005367 000634 DEC ADPAS ;FINISHED A PASS ?
561 001634* 001006 BNE AD3 ;NO
562 001636* 000417 BR DONE ;FINISHED A PASS
563
564 001640* 026767 000002 000626 AD2: ADD #2,TBPTR ;UP DATE THE POINTER
565 001646* 005267 000626 INC CHOUT ;UPDATE THE CHANNEL
566
567 001652* 116777 000622 176362 AD3: MOVB CHOUT,@ADCSHB ;LOAD CHANNEL INTO HIGH BYTE
568 001660* 112777 000100 176346 MOVB #BIT6,@ADCS
569 001666* 105277 176342 INCR @ADCS
570 001672* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
571
572 001676* 005077 176332 DONE: CLR @ADCS ;CLEAR STATUS/CONTROL
573 001702* TIMES:
574 001702* 104413 000000* ENDS,BEGIN ;SIGNAL END OF ITERATION,
575 ;MONITOR SHALL TEST END OF PASS
576 001706* 000167 176352 JMP RSTRT ;GO BACK AND DO IT AGAIN

```

```

577
578
579
580 001712* NPRADI
581
582 001712* 000004 000000* 001720*
583
584 001720* 005777 176310 18: TST @ADCS ;TEST FOR ERROR
585 001724* 100025 BPL NPRADI
586 001726* 017767 176302 176146 MOV @ADCS,ACSR ;LOAD VALUE
587 001734* 012777 000002 176272 MOV #2,@ADCS
588 001742* 005077 176272 CLR @ADMR
589 001746* 005077 176262 CLR @ADCS ;CLEAR STSUA
590 001752* 017767 176260 000516 MOV @ADDBR,ADTEMP
591 001760* 012767 000002 176120 MOV #2,ERRTYP ;DATA LATE
592
593 001766* 104406 000000* 000000
594
595 001774* 104410 000000*
596
597 002000* 017767 176232 000470 NPRADI: MOV @ADDBR,ADTEMP ;READ BUFFER
598 002006* 026767 176216 000464 CMP ENDCH,CHOUT ;TEST FOR LAST CHANNEL
599 002014* 001017 BNE NPRAD3 ;BRANCH IF NOT
600 002016* 005767 000450 TST AORB ;TEST FOR A OR B BUFFER
601 002022* 001041 BNE NPRAD2 ;BRANCH IF B
602 002024* 012777 000006 176202 MOV #6,@ADCS ;LOAD CA POINTER
603 002032* 016777 000450 176200 MOV TABBPA,@ADMR ;LOAD CURRENT ADDRESS
604 002040* 012767 177777 000424 MOV #1,AORB
605 002046* 005067 000426 CLR CHOUT ;CLEAR CHANNEL
606 002052* 000402 BR NPRADI
607 002054* 005267 000420 NPRAD3: INC CHOUT ;UPDATE CHANNEL
608 002060* 012777 000004 76146 NPRADI: MOV #4,@ADCS ;LOAD WORD COUNT POINTER
609 002066* 012777 177777 176144 MOV #1,@ADMR ;LOAD WORD COUNT
610 002074* 012777 000002 176132 MOV #2,@ADCS ;LOAD DMA STATUS
611 002102* 016777 000356 176130 MOV MYEA,@ADMR ;LOAD EA BITS AND DMA ENABLE
612 002110* 116777 000364 176124 MOVB CHOUT,@ADCSHB ;LOAD CHANNEL INTO THE HIGH BYTE
613 002116* 105277 176112 INCR @ADCS
614 002122* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
615
616 002126* 112777 000002 176100 NPRAD2: MOVB #2,@ADCS
617 002134* 005077 176100 CLR @ADMR
618 002140* 004767 000034 JSR PC,ADCK ;CHECK THE CONVERTED VALUE
619 002144* 005367 000320 DEC ADPAS ;FINISHED A PASS ?
620 002150* 001652 BEQ DONE ;YES
621 002152* 012777 000006 176054 MOV #6,@ADCS ;LOAD CA POINTER
622 002160* 016777 000324 176052 MOV TABBPA,@ADMR ;LOAD CA
623 002166* 005067 000300 CLR AORB ;SET POINTER TO A BUFFER
624 002172* 005067 000302 CLR CHOUT ;CLEAR CHANNEL
625 002176* 000730 BR NPRADI ;BRANCH TO START AT CH 0

```

```

620 002200 005367 000262 ANCKI DEC WHYCT ;DECREMNT COUNTER
621 002204 001401 BEQ 18 ;BRANCH IF DONE
622 002206 000207 RTS PC ;EXIT
623 002210 016767 176016 000250 18: MOV WHYCNT,WHYCT ;RELOAD COUNTER
624 002216 012767 002522 000256 MOV #ADTBLA,POINTA ;LOAD POINTER TO TABLE A
625 002224 012767 002724 000252 MOV #ADTBLS,POINTB ;LOAD POINTER TO TABLE B
626 002232 005067 000140 CLR CHOUTA
627 002236 017767 000234 000232 ADLOOP: MOV @POINTA,ADTEMP ;SAVE DATA
628 002244 167767 000234 000224 SUB @POINTB,ADTEMP ;SUBTRACT SECOND FROM THE FIRST
629 002252 001413 BEQ ADLOPA ;BR IF EQUAL
630 002254 100005 BPL ADLOPB ;BR IF PLUS
631 002256 066767 175744 000212 ADD WIDE,ADTEMP ;CHECK FOR A=B<-WIDE
632 002264 100006 BPL ADLOPA ;BR IF OK
633 002266 000422 BR DATBAD ;BR IF ERROR
634 002270 166767 175732 000200 ADLOPB: SUB WIDE,ADTEMP ;CHECK FOR A=B>+WIDE
635 002276 003401 BLE ADLOPA ;BR IF OK
636 002300 000415 BR DATBAD ;BR IF ERROR
637 002302 026767 000070 175720 ADLOPA: CMP CHOUTA,ENDCH ;TEST FOR LOAS CHANNEL
638 002310 001433 BEQ LOADLS ;BRANCH IF LAST
639 002312 005267 000000 INC CHOUTA ;POINTER TO CURRENT CHANNEL IF SIMPLE A TO D
640 002316 062767 000002 000156 ADD #2,POINTA ;MOVE POINTER
641 002324 062767 000002 000152 ADD #2,POINTB ;MOVE POINTER
642 002332 000741 BR ADLOOP ;TEST MORE CHANNELS
651
652
653 002334 016767 000142 175540 DATBAD: MOV POINTA,SBADR ;LOAD "GOOD" ADDRESS
654 002342 017767 000134 175536 MOV @POINTA,ASB ;LOAD "GOOD" DATA
655 002350 016767 000130 175526 MOV @POINTB,WASADR ;LOAD "BAD" ADDRESS
656 002356 017767 000122 175524 MOV @POINTB,AWAS ;LOAD "RAD" DATA
657 002364 005077 175644 CLR @ADCS
658 ;*****
659 002370 104404 000000 DATER$,BEGIN ;DATA ERROR!!!
660 ;*****
661 002374 000742 BR ADLOPA
662
663 002376 000000 CHOUTA: 0 ;BAD CHANNEL
  
```

```

664
665
666 002400 016767 000052 000052 LOADLS: MOV TEMP1,TEMP2
667 002406 012767 000006 000046 MOV #6,TEMP3
668 002414 016777 000040 175614 28: MOV TEMP2,@ADDBR
669 002422 105267 000033 INCB TEMP2+1
670 002426 005367 000030 DEC TEMP3
671 002432 001370 BNE 28
672 002434 005267 000016 INC TEMP1
673 002440 022767 000040 000010 CMP #40,TEMP1
674 002446 001002 BNE 38
675 002450 005067 000002 CLR TEMP1
676 002454 000207 38: RTS PC
677 002456 000000 TEMP1: 0
678 002460 000000 TEMP2: 0
679 002462 000000 TEMP3: 0
680
681 002464 010000 MYEAI 10000 ;EA IN 14-13 DMA ENABLE IN 12
682 002466 000001 WHYCT: 1
683 002470 000000 ADPAS: 0
684 002472 000000 AORBI: 0
685 002474 002522 TRPTR: ADTBLA
686 002476 000000 ADTEMP: 0
687 002500 000000 CHOUT: 0
688 002502 000000 POINTA: 0
689 002504 000000 POINTB: 0
690 002506 000000 TABA: 0
691 002510 000000 TABAPA: 0
692 002512 000000 TABAEA: 0
693 002514 000000 TABR: 0
694 002516 000000 TABBPA: 0
695 002520 000000 TABBEA: 0
696 002522 000000 ADTBLA: 0 ;TABLE A
697 002724 000000 ADTBLS: =.+200 ;TABLE B
698 002724 000000 ;TABLE B
699 003126 =.+200
700 000001 ;END
  
```


PRTY7	# 000340	312#					
PS	# 177776	312#					
PS#	# 177776	312#					
PUSH	# 005746	312#					
PUSH2	# 024646	312#					
RAND#	# 104417	312#					
RANNUM	000054R	249#					
RESTR	000264R	260	343#	576			
RES1	000056R	251#					
RES2	000060R	252#					
RSTRT	000112R	260#					
SBADR	000102R	261#	653#				
SMPAD	001464R	516	531#				
SMPAD1	001540R	536	546#				
SOFCONT	000042R	244#					
SOFER#	104406	312#	542	593			
SOFPAS	000046R	246#					
SPOINT	000032R	240#					
SPSIZ	# 000040	1#	273				
SR1	000016R	233#	393	506			
SR2	000020R	234#					
SR3	000022R	235#					
SR4	000024R	236#					
START	000250R	239	340#				
STAT	000026R	230#					
SVR0	000062R	253#					
SVR1	000064R	254#					
SVR2	000066R	255#					
SVR3	000070R	256#					
SVR4	000072R	257#					
SVR5	000074R	258#					
SVR6	000076R	259#					
SYSCNT	000052R	248#					
TABA	002506R	355#	357	690#			
TABAEA	002512R	359	692#				
TABADA	002510R	510	622	691#			
TABB	002514R	356#	358	693#			
TABBEA	002520R	695#					
TABBPA	002516R	603	694#				
TBPTR	002474R	521#	546#	551#	557#	564#	685#
TEMP1	002456R	666	672#	673	675#	677#	
TEMP2	002460R	666#	668	669#	678#		
TEMP3	002462R	667#	670#	679#			
TESTAD	000446R	368#					
TIMES	001702R	573#					
TRPDFD	# 000022	312#					
TSPAD	001414R	507	516#				
TSTCAR	000604R	394	399#				
TSTEXT	001124R	453	465#				
TSTMCR	000776R	424	438#				
VECTOR	000010R	229#	350	351			
WASADR	000104R	263#	655#				
WDFR	000116R	270#					
WDTO	000114R	269#	340#				
WHYCNT	000232R	322#	630				
WHYCT	002466R	627#	630#	682#			

WIDE	000226R	320#	638	641			
XFLAG	000005R	227#					
.	# 003126R	697#	699#				
. ABS.	000000	000					
	003126	001					

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XLPDF0,XLPDF0/SOL/CRF1SY#DUXCOM,XLPDF0
 RUN-TIME: 1 2 .3 SECONDS
 RUN-TIME RATIO: 26/4#6.3
 CORE USED: 7K (13 PAGES)