.REM _

IDENTIFICATION

PRODUCT CODE: AC-E730F-MC

PRODUCT NAME: CXLPDF0 LPS-AD,NP MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

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:

VARIES 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=Ø BR CONVERSIONS (LPSAD12)

6. DEVICE OPTION SETUP:

NONE REQUIRED

7.Ø 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 PREFORMED 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 COM-
PARISON 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. LOADLS: 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.

```
                                  .LIST
                                  ;LPS-11 AD DEC/X11 EXERCISER MODULE

 000000'                                  IOMOD   <LPDF  >,170400,1,6,,,50.,50
 000000'                                  MODULE  140000,LPDF  ,170400,1,6,,,50.,50
                                          .TITLE  LPDF DEC/X11 SYSTEM EXERCISER MODULE
                                  ;       DDXCOM  VERSION 6      23-MAY-78
                                          .LIST   BIN
                                  ;*****************************************************************
 000000'                          BEGIN:
 000000' 050114  043104      040  MODNAM: .ASCII  /LPDF / ;MODULE NAME.
 000005'    000                   XFLAG:  .BYTE   OPEN               ;USED TO KEEP TRACK OF WBUFF USAGE
 000006' 170400                   ADDR:   170400+0                   ;1ST DEVICE ADDR.
 000010' 000001                   VECTOR: 1+0                        ;1ST DEVICE VECTOR.
 000012'    300                   BR1:    .BYTE   PRTY6+0            ;1ST BR LEVEL.
 000013'    000                   BR2:    .BYTE   PRTY+0             ;2ND BR LEVEL.
 000014' 000001                   DVID1:  +1                         ;DEVICE INDICATOR 1.
 000016' 000000                   SR1:    OPEN                       ;SWITCH REGISTER 1
 000020' 000000                   SR2:    OPEN                       ;SWITCH REGISTER 2
 000022' 000000                   SR3:    OPEN                       ;SWITCH REGISTER 3
 000024' 000000                   SR4:    OPEN                       ;SWITCH REGISTER 4
                                  ;*****************************************************************
 000026' 140000                   STAT:   140000                     ;STATUS WORD.
 000030' 000250'                  INIT:   START                      ;MODULE START ADDR.
 000032' 000224'                  SPOINT: MODSP                      ;MODULE STACK POINTER.
 000034' 000000                   PASCNT: 0                          ;PASS COUNTER.
 000036' 000062                   ICONT:  50.                        ;# OF ITERATIONS PER PASS=50.
 000040' 000000                   ICOUNT: 0                          ;LOC TO COUNT ITERATIONS
 000042' 000000                   SOFCNT: 0                          ;LOC TO SAVE TOTAL SOFT ERRORS
 000044' 000000                   HRDCNT: 0                          ;LOC TO SAVE TOTAL HARD ERRORS
 000046' 000000                   SOFPAS: 0                          ;LOC TO SAVE SOFT ERRORS PER PASS
 000050' 000000                   HRDPAS: 0                          ;LOC TO SAVE HARD ERRORS PER PASS
 000052' 000000                   SYSCNT: 0                          ;# OF SYS ERRORS ACCUMULATED
 000054' 000000                   RANNUM: 0                          ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
 000056'                          CONFIG:                            ;RESERVED FOR MONITOR USE
 000056' 000000                   RES1:   0                          ;RESERVED FOR MONITOR USE
 000060' 000000                   RES2:   0                          ;RESERVED FOR MONITOR USE
 000062' 000000                   SVR0:   OPEN                       ;LOC TO SAVE R0.
 000064' 000000                   SVR1:   OPEN                       ;LOC TO SAVE R1.
 000066' 000000                   SVR2:   OPEN                       ;LOC TO SAVE R2.
 000070' 000000                   SVR3:   OPEN                       ;LOC TO SAVE R3.
 000072' 000000                   SVR4:   OPEN                       ;LOC TO SAVE R4.
 000074' 000000                   SVR5:   OPEN                       ;LOC TO SAVE R5.
 000076' 000000                   SVR6:   OPEN                       ;LOC TO SAVE R6.
 000100' 000000                   CSRA:   OPEN                       ;ADDR OF CURRENT CSR.
 000102'                          SBADR:                             ;ADDR OF GOOD DATA, OR
 000102' 000000                   ACSR:   OPEN                       ;CONTENTS OF CSR.
 000104'                          WASADR:                            ;ADDR OF BAD DATA, OR
 000104' 000000                   ASTAT:  OPEN                       ;STATUS REG CONTENTS.
 000106'                          ERRTYP:                            ;TYPE OF ERROR
 000106' 000000                   ASB:    OPEN                       ;EXPECTED DATA.
 000110' 000000                   AWAS:   OPEN                       ;ACTUAL DATA.
 000112' 000264'                  RSTRT:  RESTRT                     ;RESTART ADDRESS AFTER END OF PASS
 000114' 000000                   WDTO:   OPEN                       ;WORDS TO MEMORY PER ITERATION
 000116' 000000                   WDFR:   OPEN                       ;WORDS FROM MEMORY PER ITERATION
```



```
 000120' 000000                   INTR:   OPEN                       ;# OF INTERRUPTS PER ITERATION
 000122' 000050                   IDNUM:  50                         ;MODULE IDENTIFICATION NUMBER=50
         000040                           .REPT   SPSIZ              ;MODULE STACK STARTS HERE.
                                          .NLIST
                                          .WORD   0
                                          .LIST
                                          .ENDR
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
                                          .LIST
 000224'                          MODSP:
                                  ;*****************************************************************
                                          .LIST
```

```
                                        .LIST

000224' 001000                  ADPASS: 1000            ;NUMBER ON TIMES THRU ALL CHANNELS BEFORE "ENDPASS"

000226' 000004                  WIDE:   4               ; + OR - COUNT SPREAD ON AN A TO D CONVERSION
000230' 000003                  ENDCH:  3               ;HIGHEST CHANNEL TO BE TESTED <77
000232' 000001                  WHYCNT: 1               ;COUNTER BEFORE CONVERSIONS ARE TESTED

                                ;LPS-AD COMMON DEVICE ADDRESSES

000234' 170400                  ADCS:   170400          ;A TO D STATUS REGISTER
000236' 170402                  ADDBR:  170402          ;A TO D BUFFER

000240' 170436                  ADMR:   170436          ;NPR A TO D OPTION ADDRESS

000242' 170401                  ADCSHB: 170401          ;A TO D STATUS <HIGH BYTE>

                                ;LPS-AD COMMON DEVICE INTERRUPT VECTOR

000244' 000300                  ADINT:  300             ;A TO D VECTOR
000246' 000302                  ADINT1: 302

                                ;NOW SET UP THE ADDRESS AND VECTOR DISPATCH LOC.
000250' 012767  010000  177636  START:  MOV     #10000,WDTO     ;10000 WDS TO MEM
000256' 012767  010000  177634          MOV     #10000,INTR     ;10000 INTERRUPTS/ITERATION

000264' 016767  177516  177742  RESTRT: MOV     ADDR,ADCS       ;LOAD DEVICE ADDRESS
000272' 016767  177510  177736          MOV     ADDR,ADDBR
000300' 062767  000002  177730          ADD     #2,ADDBR
000306' 016767  177474  177724          MOV     ADDR,ADMR       ;LOAD DMA WORD
000314' 062767  000036  177716          ADD     #36,ADMR        ;TO CORRECT VALUE
000322' 016767  177460  177712          MOV     ADDR,ADCSHB     ;LOAD ADC HIGH BYTE LOC
000330' 005267  177706                  INC     ADCSHB          ;MAKE HIGH BYTE
000334' 016767  177450  177702          MOV     VECTOR,ADINT
000342' 016767  177442  177676          MOV     VECTOR,ADINT1
000350' 062767  000002  177670          ADD     #2,ADINT1
000356' 016767  177652  177514          MOV     ADCS,CSRA
000364' 016767  177634  002076          MOV     ADPASS,ADPAS
000372' 012767  002522' 002106          MOV     #ADTBLA,TABA    ; GET TABLE A ADDRESS
000400' 012767  002724' 002106          MOV     #ADTBLB,TABB    ; GET TABLE B ADDRESS
000406' 104415  000000' 002506'         GETPA$,BEGIN, TABA              ;GET PHYSICAL ADDRESS FROM 16-BIT TABA
000414' 104415  000000' 002514'         GETPA$,BEGIN, TABB              ;GET PHYSICAL ADDRESS FROM 16-BIT TABB
000422' 016700  002064                  MOV     TABAEA,R0       ; GET EXTENDED MEMORY BITS
000426' 000300                          SWAB    R0              ;MOVE TO HIGH BYTE
000430' 006100                          ROL     R0              ;MOVE LEFT
000432' 042700  117777                  BIC     #117777,R0      ;MASK
000436' 052700  010000                  BIS     #BIT12,R0       ;SET DMA ENABLE
000442' 010067  002016                  MOV     R0,MYEA         ;SAVE EA BITS AND DMA ENABLE
```



```
                                ;LPS-AD LOGIC TEST

000446' 005077  177562  TESTAD: CLR     @ADCS           ;CLEAR STATUS
000452' 005777  177556          TST     @ADCS           ;TEST STATUS
000456' 001025                  BNE     1$              ;BR IF ERROR

000460' 012777  052524  177546          MOV     #52524,@ADCS    ;LOAD STATUS
000466' 012767  052524  177410          MOV     #52524,ASTAT    ;LOAD POINTER
000474' 022777  052524  177532          CMP     #52524,@ADCS    ;TEST STATUS
000502' 001013                  BNE     1$              ;BR IF ERROR

000504' 012777  125012  177522          MOV     #125012,@ADCS   ;LOAD STATUS
000512' 012767  125012  177364          MOV     #125012,ASTAT   ;LOAD POINTER
000520' 022777  125012  177506          CMP     #125012,@ADCS   ;TEST STATUS
000526' 001001                  BNE     1$              ;BR IF ERROR
000530' 000420                  BR      PRIMTS

000532' 017767  177476  177342  1$:     MOV     @ADCS,ACSR      ;READ STATUS
000540' 016767  177470  177332          MOV     ADCS,CSRA       ;LOAD ADDRESS
000546' 005077  177462                  CLR     @ADCS           ;ENSURE CLEAR STATUS
000552' 012767  000025  177326          MOV     #25,ERRTYP      ;BIT STUCK IN C/S REG
                                ;*****************************************************************
000560' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CONTROL/STATUS BIT IN ERROR
                                ;*****************************************************************
000566' 104410  000000'                 END$,BEGIN              ;
```

```
                                ;TEST THE LPS-NP CURRENT ADDRESS REGISTER

000572' 005767  177220  PRIMTS: TST     SR1             ;TEST SR1
000576' 100402                  BMI     TSTCAR          ;BR IF NPR
000600' 000167  000522          JMP     PRIME

000604' 005077  177424  TSTCAR: CLR     @ADCS           ;CLEAR CONTROL
000610' 012777  000006  177416  MOV     #6,@ADCS        ;LOAD MODE
000616' 016767  177416  177254  MOV     ADMR,CSRA       ;LOAD ERROR DEVICE ADDRESS

000624' 005077  177410          CLR     @ADMR           ;CLEAR C.A.
000630' 005067  177250          CLR     ASTAT           ;CLEAR POINTER
000634' 005777  177400          TST     @ADMR           ;TEST C.A.
000640' 001037                  BNE     1$              ;BR IF ERROR

000642' 012777  125212  177370  MOV     #125212,@ADMR   ;LOAD C.A.
000650' 012767  125212  177226  MOV     #125212,ASTAT   ;LOAD POINTER
000656' 022777  125212  177354  CMP     #125212,@ADMR   ;TEST IT
000664' 001025                  BNE     1$              ;BR IF ERROR

000666' 012777  052524  177344  MOV     #52524,@ADMR    ;LOAD C.A.
000674' 012767  052524  177202  MOV     #52524,ASTAT    ;LOAD POINTER
000702' 022777  052524  177330  CMP     #52524,@ADMR    ;TEST IT
000710' 001013                  BNE     1$              ;BR IF ERROR

000712' 012777  000001  177320  MOV     #1,@ADMR        ;LOAD ODD ADDRESS <NO ODD BIT>
000720' 005067  177160          CLR     ASTAT           ;CLEAR POINTER
000724' 005777  177310          TST     @ADMR           ;TEST IT
000730' 001003                  BNE     1$              ;BR IF ODD ADDRESS

000732' 005077  177302          CLR     @ADMR           ;CLEAR EXT STATUS
000736' 000417                  BR      TSTWCR          ;TEST W.C. REGISTERS

000740' 017767  177274  177134  1$:     MOV     @ADMR,ACSR      ;READ BAD VALUE
000746' 005077  177266          CLR     @ADMR           ;CLEAR REGISTER
000752' 005077  177256          CLR     @ADCS           ;CLEAR CONTOL
000756' 012767  000025  177122  MOV     #25,ERRTYP      ;BIT STUCK IN CUR ADDR REGISTER
                                ;*****************************************************************
000764' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;CURRENT ADDRESS REGISTER BIT IN ERROR
                                ;*****************************************************************
000772' 104410  000000'         END$,BEGIN              ;
```



```
                                ;TEST THE LPS-NP WORD COUNT REGISTER

000776' 005077  177232  TSTWCR: CLR     @ADCS           ;ENSURE CLEAR CONTROL
001002' 012777  000004  177224  MOV     #4,@ADCS        ;LOAD POINTER TO W.C.
001010' 005077  177224          CLR     @ADMR           ;CLEAR W.C.
001014' 005067  177064          CLR     ASTAT           ;CLOAD POINTER
001020' 005777  177214          TST     @ADMR           ;TEST W.C.
001024' 001024                  BNE     1$              ;BR IF ERROR

001026' 012777  005252  177204  MOV     #5252,@ADMR     ;LOAD W.C.
001034' 012767  005252  177042  MOV     #5252,ASTAT     ;LOAD POINTER
001042' 022777  005252  177170  CMP     #5252,@ADMR     ;TEST W.C.
001050' 001012                  BNE     1$              ;BR IF ERROR

001052' 012777  002525  177160  MOV     #2525,@ADMR     ;LOAD W.C.
001060' 012767  002525  177016  MOV     #2525,ASTAT     ;LOAD POINTER
001066' 022777  002525  177144  CMP     #2525,@ADMR     ;TEST W.C.
001074' 001413                  BEQ     TSTEXT          ;BR IF NO ERROR

001076' 017767  177136  176776  1$:     MOV     @ADMR,ACSR      ;READ W.C. VALUE
001104' 012767  000025  176774  MOV     #25,ERRTYP      ;BIT STUCK IN WORD COUNT REGISTER
                                ;*****************************************************************
001112' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;WORD COUNT REGISTER BIT IN ERROR
                                ;*****************************************************************
001120' 104410  000000'         END$,BEGIN              ;
```

```
                                ;TEST THE EXTENDED STATUS REGISTER BITS
001124' 005077  177104  TSTEXT: CLR     @ADCS           ;ENSURE CLEAR CONTROL
001130' 012777  000002  177076          MOV     #2,@ADCS        ;LOAD MODE
001136' 005077  177076          CLR     @ADMR           ;CLEAR EXT STATUS
001142' 005067  176736          CLR     ASTAT           ;CLEAR POINTER
001146' 005777  177066          TST     @ADMR           ;TEST IT
001152' 001046                  BNE     1$              ;BR IF ERROR

001154' 012777  050000  177056          MOV     #BIT14!BIT12,@ADMR      ;LOAD EXT STATUS
001162' 012767  050000  176714          MOV     #BIT14!BIT12,ASTAT      ;LOAD POINTER
001170' 022777  050000  177042          CMP     #BIT14!BIT12,@ADMR      ;TEST IT
001176' 001034                  BNE     1$              ;BR IF ERROR

001200' 012777  020000  177032          MOV     #BIT13,@ADMR    ;LOAD EXT STATUS
001206' 012767  020000  176670          MOV     #BIT13,ASTAT    ;LOAD POINTER
001214' 022777  020000  177016          CMP     #BIT13,@ADMR    ;TEST IT
001222' 001022                  BNE     1$              ;BR IF ERROR

001224' 012777  000100  177002          MOV     #100,@ADCS      ;LOAD INT ENABLE
001232' 052777  000002  176774          BIS     #2,@ADCS        ;SET MODE
001240' 012777  010000  176772          MOV     #BIT12,@ADMR    ;ENABLE DMA
001246' 032777  000100  176760          BIT     #100,@ADCS      ;DMA ENB, SHOULD CLEAR INT ENABLE
001254' 001065                  BNE     1$              ;BR IF BIT SIX SET

001256' 005077  176756          CLR     @ADMR           ;CLEAR EXT STATUS
001262' 005077  176746          CLR     @ADCS           ;CLEAR CONTROL
001266' 000417                  BR      PRIME           ;START TEST

001270' 017767  176744  176004  1$:     MOV     @ADMR,ACSR      ;READ VALUE
001276' 005077  176736          CLR     @ADMR           ;CLEAR EXT STATUS
001302' 005077  176726          CLR     @ADCS           ;ENSURE CLEAR CONTROL
001306' 012767  000025  176572          MOV     #25,ERRTYP      ;BIT STUCK IN EXT, STAT REGISTER
                                ;*************************************************************
001314' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;EXTENDED STATUS REGISTER BIT IN ERROR
                                ;*************************************************************
001322' 104410  000000'         END$,BEGIN              ;
```



```
                                ; PRIMER ROUTINE
001326' 005077  176702  PRIME:  CLR     @ADCS
001332' 005767  176600          TST     SR1             ;TEST BIT 15
001336' 100026                  BPL     TSMPAD          ;BRANCH IF TEST SIMPLE A TO D
001340' 012777  001712' 176676          MOV     #NPRAD,@ADINT   ;SET UP NPR AD VECTOR
001346' 012777  000006  176660          MOV     #6,@ADCS        ;LOAD CA POINTER
001354' 016777  001130  176656          MOV     TABAPA,@ADMR    ;LOAD CA REGISTER
001362' 012777  000004  176644          MOV     #4,@ADCS        ;LOAD WC POINTER
001370' 012777  177777  176642          MOV     #-1,@ADMR       ;LOAD WC REGISTER
001376' 012777  000002  176630          MOV     #2,@ADCS        ;LOAD STATUS POINTER
001404' 016777  001154  176626          MOV     MYEA,@ADMR      ;LOAD EXT, STATUS REGISTER
001412' 000406                  BR      LPAD
001414' 012777  001464' 176622  TSMPAD: MOV     #SMPAD,@ADINT   ;SET UP SIMPLE AD VECTOR
001422' 052777  000100  176604          BIS     #BIT6,@ADCS     ;ENABLE INTERRUPT
001430' 116777  176356  176610  LPAD:   MOVB    BR1,@ADINT1
001436' 005067  001436          CLR     CHOUT
001442' 005067  001521          CLR     AORB            ;CLEAR BUFFER POINTER
001446' 012767  002520' 001020          MOV     #ADTBLA,TBPTR   ;LOAD TABLE POINTER
001454' 005277  176551          INC     @ADCS           ;START A TO D

001460' 104404  000000'         EXITS,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
```

```
                                ;SIMPLE A TO D TEST
001464'                         SMPAD1:
                                ;*****************************************************************
001464' 000004  000000' 001472'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                ;*****************************************************************
001472' 005777  176536  1$:     TST     @ADCS           ;TEST A TO D STATUS
001476' 100020                  BPL     SMPAD1
001500' 017767  176530  176374          MOV     @ADCS,ACSR      ;LOAD VALUE
001506' 005077  176522                  CLR     @ADCS           ;CLEAR STATUS
001512' 017767  176520  000756          MOV     @ADDBR,ADTEMP
001520' 012767  000025  176360          MOV     #25,ERRTYP      ;DATA LATE
                                ;*****************************************************************
001526' 104406  000000' 000000          SOFER$,BEGIN,NULL       ; A TO D STATUS ERROR
                                ;*****************************************************************
001534' 104410  000000'                 END$,BEGIN              ;
001540' 017777  176472  000726  SMPAD1: MOV     @ADDBR,@TBPTR   ;READ THE DATA INTO A BUFFER
001546' 026767  176456  000724          CMP     ENDCH,CHOUT     ;TEST FOR LAST CHANNEL
001554' 001031                  BNE     AD2             ;BRANCH IF NOT
001556' 005767  000710          TST     AORB            ;TEST BUFFER
001562' 001011                  BNE     AD1
001564' 012767  002724' 000702          MOV     #ADTBLB,TBPTR   ;END OF A, RESET TO B
001572' 012767  177777  000672          MOV     #-1,AORB
001600' 005067  000674                  CLR     CHOUT
001604' 000422                  BR      AD3
001606' 004767  000366  AD1:    JSR     PC,ADCK         ;TEST THE DATA
001612' 012767  002522' 000654          MOV     #ADTBLA,TBPTR   ;RESET TO A BUFFER
001620' 005067  000654                  CLR     CHOUT           ;CLEAR STARTING CHANNEL
001624' 005067  000642                  CLR     AORB
001630' 005367  000634                  DEC     ADPAS           ;FINISHED A PASS ?
001634' 001006                  BNE     AD3             ;NO
001636' 000417                  BR      DONE            ;FINISHED A PASS
001640' 062767  000002  000626  AD2:    ADD     #2,TBPTR        ;UP DATE THE POINTER
001646' 005267  000626          INC     CHOUT           ;UPDATE THE CHANNEL
001652' 116777  000622  176362  AD3:    MOVB    CHOUT,@ADCSHB   ;LOAD CHANNEL INTO HIGH BYTE
001660' 112777  000100  176346          MOVB    #BIT6,@ADCS
001666' 105277  176342                  INCB    @ADCS
001672' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
001676' 005077  176332  DONE:   CLR     @ADCS           ;CLEAR STATUS/CONTROL
001702'                 TIMES:
001702' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION,
                                                                ;MONITOR SHALL TEST END OF PASS
001706' 000167  176352                  JMP     RESTRT          ;GO BACK AND DO IT AGAIN
```



```
                                ;NPR A TO D TEST
001712'                         NPRAD1:
                                ;*****************************************************************
001712' 000004  000000' 001720'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                ;*****************************************************************
001720' 005777  176310  1$:     TST     @ADCS           ;TEST FOR ERROR
001724' 100025                  BPL     NPRADA
001726' 017767  176302  176146          MOV     @ADCS,ACSR      ;LOAD VALUE
001734' 012777  000002  176272          MOV     #2,@ADCS
001742' 005077  176272                  CLR     @ADMR
001746' 005077  176262                  CLR     @ADCS           ;CLEAR STSTUA
001752' 017767  176260  000516          MOV     @ADDBR,ADTEMP
001760' 012767  000002  176120          MOV     #2,ERRTYP       ;DATA LATE
                                ;*****************************************************************
001766' 104406  000000' 000000          SOFER$,BEGIN,NULL       ;NPR A TO D STATUS ERROR
                                ;*****************************************************************
001774' 104410  000000'                 END$,BEGIN              ;
002000' 017767  176232  000470  NPRADA: MOV     @ADDBR,ADTEMP   ;READ BUFFER
002006' 026767  176216  000464          CMP     ENDCH,CHOUT     ;TEST FOR LAST CHANNEL
002014' 001017                  BNE     NPRAD3          ;BRANCH IF NOT
002016' 005767  000450          TST     AORB            ;TEST FOR A OR B BUFFER
002022' 001041                  BNE     NPRAD2          ;BRANCH IF B
002024' 012777  000006  176202          MOV     #6,@ADCS        ;LOAD CA POINTER
002032' 016777  000460  176200          MOV     TABBPA,@ADMR    ;LOAD CURRENT ADDRESS
002040' 012767  177777  000424          MOV     #-1,AORB
002046' 005067  000426                  CLR     CHOUT           ;CLEAR CHANNEL
002052' 000402                  BR      NPRAD1
002054' 005267  000420  NPRAD3: INC     CHOUT           ;UPDATE CHANNEL
002060' 012777  000004   76146  NPRAD1: MOV     #4,@ADCS        ;LOAD WORD COUNT POINTER
002066' 012777  177777  176144          MOV     #-1,@ADMR       ;LOAD WORD COUNT
002074' 012777  000002  176132          MOV     #2,@ADCS        ;LOAD DMA STATUS
002102' 016777  000356  176130          MOV     MYEA,@ADMR      ;LOAD EA BITS AND DMA ENABLE
002110' 116777  000364  176124          MOVB    CHOUT,@ADCSHB   ;LOAD CHANNEL INTO THE HIGH BYTE
002116' 105277  176112                  INCB    @ADCS
002122' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
002126' 112777  000002  176100  NPRAD2: MOVB    #2,@ADCS
002134' 005077  176100                  CLR     @ADMR
002140' 004767  000034                  JSR     PC,ADCK         ;CHECK THE CONVERTED VALUE
002144' 005367  000320                  DEC     ADPAS           ;FINISHED A PASS ?
002150' 001652                  BEQ     DONE            ;YES
002152' 012777  000006  176054          MOV     #6,@ADCS        ;LOAD CA POINTER
002160' 016777  000324  176052          MOV     TABAPA,@ADMR    ;LOAD CA
002166' 005067  000300                  CLR     AORB            ;SET POINTER TO A BUFFER
002172' 005067  000302                  CLR     CHOUT           ;CLEAR CHANNEL
002176' 000730                  BR      NPRAD1          ;BRANCH TO START AT CH 0
```

```
                ;DECREMENT COUNTER
002200' 005367 000262      ADCK:  DEC     WHYCT
002204' 001401                    BEQ     1$
002206' 000207                    RTS     PC
002210' 016767 176016 000250 1$:  MOV     WHYCNT,WHYCT
002216' 012767 002522' 000256     MOV     #ADTBLA,POINTA
002224' 012767 002724' 000252     MOV     #ADTBLB,POINTB
002232' 005067 000140             CLR     CHOUTA
002236' 017767 000244 000232 ADLOOP: MOV  @POINTA,ADTEMP
002244' 167767 000234 000224     SUB     @POINTB,ADTEMP
002252' 001413                    BEQ     ADLOPA
002254' 100005                    BPL     ADLOPB
002256' 066767 175744 000212     ADD     WIDE,ADTEMP
002264' 100006                    BPL     ADLOPA
002266' 000422                    BR      DATBAD
002270' 166767 175732 000200 ADLOPB: SUB  WIDE,ADTEMP
002276' 003401                    BLE     ADLOPA
002300' 000415                    BR      DATBAD
002302' 026767 000070 175720 ADLOPA: CMP  CHOUTA,ENDCH
002310' 001433                    BEQ     LOADLS
002312' 005267 000060             INC     CHOUTA
                                  ;WORD POINTER IF NPR A TO D
002316' 062767 000002 000156     ADD     #2,POINTA
002324' 062767 000002 000152     ADD     #2,POINTB
002332' 000741                    BR      ADLOOP
                ;
                ;
002334' 016767 000142 175540 DATBAD: MOV  POINTA,SBADR
002342' 017767 000134 175536     MOV     @POINTA,ASB
002350' 016767 000130 175526     MOV     POINTB,WASADR
002356' 017767 000122 175524     MOV     @POINTB,AWAS
002364' 005077 175644             CLR     @ADCS
                ;*****************************************************************
002370' 104404 000000'            DATER$,BEGIN          ;DATA ERROR!!!
                ;*****************************************************************
002374' 000742                    BR      ADLOPA
                ;
002376' 000000             CHOUTA: 0                     ;BAD CHANNEL
```



```
                ;
                ;
002400' 016767 000052 000052 LOADLS: MOV  TEMP1,TEMP2
002406' 012767 000006 000046     MOV     #6,TEMP3
002414' 016777 000040 175614 2$: MOV     TEMP2,@ADDBR
002422' 105267 000033             INCB    TEMP2+1
002426' 005367 000030             DEC     TEMP3
002432' 001370                    BNE     2$
002434' 005267 000016             INC     TEMP1
002440' 022767 000040 000010     CMP     #40,TEMP1
002446' 001002                    BNE     3$
002450' 005067 000002             CLR     TEMP1
002454' 000207             3$:    RTS     PC
002456' 000000             TEMP1: 0
002460' 000000             TEMP2: 0
002462' 000000             TEMP3: 0
                ;
002464' 010000             MYEA:  10000                 ;EA IN 14-13 DMA ENABLE IN 12
002466' 000001             WHYCT: 1
002470' 000000             ADPAS: 0
002472' 000000             AORB:  0
002474' 002522'            TRPTR: ADTBLA
002476' 000000             ADTEMP: 0
002500' 000000             CHOUT: 0
002502' 000000             POINTA: 0
002504' 000000             POINTB: 0
002506' 000000             TABA:  0
002510' 000000             TABAPA: 0
002512' 000000             TABAEA: 0
002514' 000000             TABB:  0
002516' 000000             TABBPA: 0
002520' 000000             TABBEA: 0
002522' 000000             ADTBLA: 0                    ;TABLE A
        002724'                    .=.+200
002724' 000000             ADTBLB: 0                    ;TABLE B
        003126'                    .=.+200
        000001                     .END
```

```
ACSR    000102R          262#    383*    426*    455*    492*    537*    586*
ADCK    002200R          556     618     627#
ADCS    000234R          326#    343*    353     368*    369     372*    374     377*    379     383     384     385*    399*
                         400*    428*    438*    439*    465*    466*    482*    483*    485     489*    494*    505*    509*
                         511*    513*    517*    522*    535     537     538*    568*    569*    572*    584     586     587*
                         589*    602*    608*    610*    613*    616*    621*    657*
ADCSHB  000242R          331#    348*    349*    567*    612*
ADDBR   000236R          327#    344*    345*    539     546     590     597     668*
ADDR    000006R          228#    343     344     346     348
ADDR22= 001000           312#
ADINT   000244R          335#    350*    508*    516*
ADINT1  000246R          336#    351*    352*    518*
ADLOOP  002236R          634#    650
ADLOPA  002302R          636     639     642     644#    661
ADLOPB  002270R          637     641#
ADMR    000240R          329#    346*    347*    401     403*    405     408*    410     413*    415     418*    420     423*
                         426     427*    440*    442     445*    447     450*    452     455     467*    469     472*    474
                         477*    479     484*    488*    492     493*    510*    512*    514*    588*    603*    609*    611*
                         617*    622*
ADPAS   002470R          354*    560*    619*    683#
ADPASS  000224R          318#    354
ADTBLA  002522R          355     521     557     631     685     696#
ADTBLB  002724R          356     551     632     698#
ADTEMP  002476R          539*    590*    597*    634*    635*    638*    641*    686#
AD1     001606R          550     556#
AD2     001640R          548     564#
AD3     001652R          554     561     567#
AORB    002472R          520*    549     552*    559*    600     604*    623*    684#
ASB     000106R          266#    654*
ASTAT   000104R          264#    373*    378*    404*    409*    414*    419*    441*    446*    451*    468*    473*    478*
AWAS    000110R          267#    656*
BEGIN   000000R          225#    357     358     388     390     431     433     458     460     497     499     524     533
                         542     544     570     574     582     593     595     614     659
BIT0  = 000001           312#    315#
BIT1  = 000002           312#    315#
BIT10 = 002000           312#    315#
BIT11 = 004000           312#    315#
BIT12 = 010000           312#    315#    363     472     473     474     484
BIT13 = 020000           312#    315#    477     478     479
BIT14 = 040000           312#    315#    472     473     474
BIT15 = 100000           312#    315#
BIT2  = 000004           312#    315#
BIT3  = 000010           312#    315#
BIT4  = 000020           312#    315#
BIT5  = 000040           312#    315#
BIT6  = 000100           312#    315#    517     568
BIT7  = 000200           312#    315#
BIT8  = 000400           312#    315#
BIT9  = 001000           312#    315#
BREAK$= 104407           312#
BR1     000012R          230#    518
BR2     000013R          231#
BTOD$ = 104421           312#
CDATA$= 104412           312#
CHOUT   002500R          519*    547     553*    558*    565*    567     598     605*    607*    612     624*    687#
CHOUTA  002376R          633*    644     646*    663#
```



```
CONFIG  000056R          250#
CSRA    000100R          260#    353*    384*    401*
DATBAD  002334R          640     643     653#
DATCK$= 104411           312#
DATER$= 104404           312#    659
DONE    001676R          562     572#    620
DVID1   000014R          232#
ENDCH   000230R          321#    547     598     644
ENDIT$= 104413           312#    574
END$  = 104410           312#    390     433     460     499     544     595
ERRTYP  000106R          265#    386*    429*    456*    495*    540*    591*
EXIT$ = 104400           312#    524     570     614
GETPA$= 104415           312#    357     358
GWBUF$= 104414           312#
HRDCNT  000044R          245#
HRDER$= 104405           312#    388     431     458     497
HRDPAS  000050R          247#
ICONT   000036R          242#
ICOUNT  000040R          243#
IDNUM   000122R          272#
INIT    000030R          239#
INTR    000120R          271#    341*
LOADLS  002400R          645     666#
LPAD    001430R          515     518#
MAP22$= 104416           312#
MODNAM  000000R          226#
MODSP   000224R          240     310#
MSGN$ = 104403           312#
MSGS$ = 104402           312#
MSG$  = 104401           312#
MYEA    002464R          364*    514     611     681#
NPRAD   001712R          508     580#
NPRADA  002000R          585     597#
NPRAD1  002060R          606     608#    625
NPRAD2  002126R          601     616#
NPRAD3  002054R          599     607#
NULL  = 000000           312#    388     431     458     497     542     593
OPEN  = 000000           227     233     234     235     236     253     254     255     256     257     258     259     260
                         262     264     266     267     269     270     271     312#
OTOA$ = 104420           312#
PASCNT  000034R          241#
PIRQ$ = 000004           312#    533     582
POINTA  002502R          631*    634     648*    653     654     688#
POINTB  002504R          632*    635     649*    655     656     689#
POPSP = 005726           312#
POPSP2= 022626           312#
PRIME   001326R          395     490     505#
PRIMTS  000572R          381     393#
PRTY  = 000000           231     312#
PRTY0 = 000000           312#
PRTY1 = 000040           312#
PRTY2 = 000100           312#
PRTY3 = 000140           312#
PRTY4 = 000200           312#
PRTY5 = 000240           312#
PRTY6 = 000300           230     312#
```

```
PRTY7 = 000340          312#
PS    = 177776          312#
PSW   = 177776          312#
PUSH  = 005746          312#
PUSH2 = 024646          312#
RAND$ = 104417          312#
RANNUM  000054R         249#
RESTRT  000264R         268     343#    576
RES1    000056R         251#
RES2    000060R         252#
RSTRT   000112R         268#
SBADR   000102R         261#    653#
SMPAD   001464R         516     531#
SMPAD1  001540R         536     546#
SOFCNT  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         238#
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#
TABAPA  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#
TSMPAD  001414R         507     516#
TSTCAR  000604R         394     399#
TSTEXT  001124R         453     465#
TSTWCR  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/CRF:SYM=OBXCOM,XLPDF0
 RUN-TIME:  1  2  .3 SECONDS
 RUN-TIME RATIO:  26/4=6.3
 CORE USED:  7K  (13 PAGES)
```