.REM _

IDENTIFICATION

PRODUCT CODE: AC-E788G-MC

PRODUCT NAME: CXCDAG0 CD11 MOD

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

CDA IS AN IOMOD THAT EXERCISES THE CD11 CARD READER. IT EXERCISES THE READER BY READING A PRE-PUNCHED ALPHANUMERIC CARD DECK. FOR EACH CARD READ A CHECKSUM IS CALCULATED AND COMPARED AGAINST A PREDEFINED VALUE. BOTH THE IMAGE AND PACKED MODES ARE VERIFIED BY CHANGING MODES ON EVERY OTHER CARD. ALL ERRORS DETECTED ARE PRINTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: ONE CD11 CARD READER WITH CONTROLLER
ONE PRE-PUNCHED ALPHANUMERIC CARD DECK (80 CARDS),
MAINDEC-89-D1B1-C

STORAGE:: CDA REQUIRES:
1. DECIMAL WORDS: 454
2. OCTAL WORDS: 0706
3. OCTAL BYTES: 1614

3. PASS DEFINITION

ONE PASS OF THE CDA MODULE CONSISTS OF READING 80 80-COLUMN CARDS (6400 CHARACTERS). FOR MULTPLE PASSES, THE SAME 80 CARD DECK MAY BE RELOADED AFTER EACH PASS OR SEVERAL DECKS MAY BE STACKED IN THE HOPPER.

4. EXECUTION TIME

ONE PASS OF CDA RUNNING ALONE ON A PDP11/05 WITH AN 80 CARD DECK TAKES APPROXIMATELY .08 MINUTES.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172460, VECTOR: 230, BR1:6, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SET-UP

A. POWER UP THE READER
B. LOAD AN ALPHANUMERIC DECK
C. DEPRESS RESET TO CLEAR ANY ERROR CONDITIONS AND

PLACE READER ON LINE.

7. MODULE OPERATION

TEST SEQUENCE:

A. SET UP VECTOR, DEVICE REGISTER ADDRESSES, AND MODULE VARIABLES
B. IF OFF-LINE REPORT ERROR AND WAIT
C. READ 80 COLUMNS IN IMAGE MODE INTO A BUFFER
D. REPORT ANY ERRORS
E. SUM UP BUFFER AND COMPARE AGAINST KNOWN CKSUM
F. READ 80 COLUMNS IN PACKED MODE INTO A BUFFER
G. REPORT ANY ERRORS
H. SUM UP BUFFER AND COMPARE AGAINST KNOWN CKSUM
I. REPEAT B THROUGH H UNTIL OUT OF CARDS (OFF-LINE AND/OR END OF FILE)
J. AFTER 80 CARDS REPORT END OF PASS; RESTART AT A

AFTER HOPPER IS EMPTY:

A. RELOAD CARD DECK
B. DEPRESS RESET TO BEGIN NEXT PASS

8. OPERATION OPTIONS

NONE

9. NON-STANDARD PRINTOUTS

NONE: ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT.

```
000000'            IOMOD   <CDAG  >,172460,230,6,,,1,14
000000'            MODULE  140000,CDAG  ,172460,230,6,,,1,14
                   .TITLE  CDAG DEC/X11 SYSTEM EXERCISER MODULE
        ;          DDXCOM  VERSION 6        23-MAY-78
                                           .LIST   BIN
                                  ;*********************************************************************
000000'                           BEGIN:
000000' 042103  043501    040     MODNAM: .ASCII  /CDAG / ;MODULE NAME.
000005'    000                    XFLAG:  .BYTE   OPEN              ;USED TO KEEP TRACK OF MBUFF USAGE
000006' 172460                    ADDR:   172460+0                  ;1ST DEVICE ADDR.
000010' 000230                    VECTOR: 230+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' 000224'                   INIT:   START                     ;MODULE START ADDR.
000032' 000224'                   SPOINT: MODSP                     ;MODULE STACK POINTER.
000034' 000000                    PASCNT: 0                         ;PASS COUNTER.
000036' 000001                    ICONT:  1                         ;# OF ITERATIONS PER PASS=1
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' 000224'                   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' 000014                    IDNUM:  14                        ;MODULE IDENTIFICATION NUMBER=14
        000040                            .REPT   SPSIZ             ;MODULE STACK STARTS HERE.
                                          .NLIST
                                          .WORD   0
                                          .LIST
                                          .ENDR
000224'                           MODSP:
                                  ;*********************************************************************
```

```
000224'                            START:
000224' 012767  001346' 001354     RESTRT: MOV     #BUFF,RDADR      ; GET VIRTUAL ADDRESS OF READ BUFFER
000232' 104415  000000' 001506'            GETPAS,BEGIN, RDADR      ;GET PHYSICAL ADDRESS FROM 16-BIT RDADR
000240' 016700  177542                     MOV     ADDR,R0          ; GET DEVICE ADDRESS
000244' 010067  001044                     MOV     R0,CDST          ; LOAD ADDR OF STATUS AND CONTROL REG.
000250' 005720                             TST     (R0)+
000252' 010067  001040                     MOV     R0,CDCC          ; LOAD ADDR OF COLUMN COUNT REG.
000256' 005720                             TST     (R0)+
000260' 010067  001034                     MOV     R0,CDBA          ; LOAD ADDR OF CURRENT BUFFER ADR.REG.
000264' 005720                             TST     (R0)+
000266' 010067  001030                     MOV     R0,CDDB          ; LOAD ADDR OF DATA BUFFER REG.
000272' 004767  000772                     JSR     PC,ECO14         ;++ GO SET UP FOR ECO # C011-00014
000276' 016700  177506                     MOV     VECTOR,R0        ; GET VECTOR
000302' 012720  000566'                    MOV     #INTER,(R0)+     ; SET POINTER TO INTERUPT SERVICE
000306' 116710  177500                     MOVB    BR1,(R0)         ; LOAD PRIORITY

000312' 005067  001010                     CLR     CRDCNT           ; ZERO CARD COUNT, FLAG
000316' 005067  001002                     CLR     STATUS           ; SET IMAGE MODE

000322' 122767  000120  000776     NUCARD: CMPB    #80.,CRDCNT      ; DONE ?
000330' 001001                             BNE     1$               ; NO, CONTINUE
000332' 000460                             BR      PASS             ; YES, GO END PASS
000334' 004567  000140             1$:     JSR     R5,READY         ; CONTROLLER AND READER READY ?
000340' 000412                             BR      READ             ; YES, CONTINUE
000342' 004767  000460                     JSR     PC,ERSUB         ; NO, LOAD ERROR INFORMATION
000346' 012767  000003  177532             MOV     #3,ERRTYP        ;NOT READY
                                           ;*************************************************************
000354' 104405  000000' 000000             HRDERS,BEGIN,NULL        ; READER STILL NOT READY ... BYE
                                           ;*************************************************************


000362'                            FINI:
000362' 104410  000000'                    ENDS,BEGIN               ; DROP THE MODULE


000366' 012777  177660  000722     READ:   MOV     #-80.,@CDCC      ; SET TO READ 80 COLUMNS
000374' 016777  001210  000716             MOV     RDPA,@CDBA       ; SET BUFFER ADDRESS
000402' 056767  001204  000714             BIS     RDEA,STATUS      ; SET EXTENDED MEMORY BITS
000410' 052767  000101  000706             BIS     #101,STATUS      ; SET ENABLE INTERUPT AND READ
000416' 016777  000702  000670             MOV     STATUS,@CDST     ; GO
000424' 104400  000000'                    EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                   ;       ------------------------
```



```
000430' 132767  000001  000671     BACK:   BITB    #BIT0,FLAG       ; END OF PASS ?
000436' 001016                             BNE     PASS             ; YES
000440' 105267  000662                     INCB    CRDCNT           ; NO, COUNT A CARD
000444' 032767  000002  000652             BIT     #BIT1,STATUS     ; PACKING MODE ?
000452' 001004                             BNE     1$               ; YES
000454' 052767  000002  000642             BIS     #BIT1,STATUS     ; NO, SET PACKING MODE
000462' 000717                             BR      NUCARD           ; GO FOR ANOTHER CARD
000464' 042767  000002  000632     1$:     BIC     #BIT1,STATUS     ; SET IMAGE MODE
000472' 000713                             BR      NUCARD           ; GO FOR ANOTHER CARD
                                   ;       ------------------------




000474'                            PASS:
000474' 104413  000000'                    ENDITS,BEGIN             ;SIGNAL END OF ITERATION.
                                                                    ;MONITOR SHALL TEST END OF PASS
                                   ;       ------------------------




000500' 012767  177777  000622     READY:  MOV     #177777,CLK      ; SET THE TIMER
000506' 012777  000400  000600     1$:     MOV     #BIT8,@CDST      ; ISSUE A POWER CLEAR
000514' 105777  000574                     TSTB    @CDST            ; CONTROLLER READY ?
000520' 100011                             BPL     2$               ; NO, WAIT
000522' 032777  010000  000564             BIT     #BIT12,@CDST     ; OFF-LINE ?
000530' 001005                             BNE     2$               ; YES, WAIT
000532' 032777  000004  000554             BIT     #BIT2,@CDST      ; BUSY ?
000540' 001001                             BNE     2$               ; YES, WAIT
000542' 000205                             RTS     R5               ; READY, RETURN
000544'                            2$:
000544' 104407  000000'                    BREAKS,BEGIN             ;TEMPORARY RETURN TO MONITOR....
000550' 104407  000000'                    BREAKS,BEGIN             ;THEN CONTINUE AT NEXT INSTRUCTION.
000554' 005367  000550                     DEC     CLK              ; IS WAIT TIME EXPENDED ?
000560' 001352                             BNE     1$               ; NO, CONTINUE TO WAIT
000562' 005725                             TST     (R5)+            ; YES, SKIP INSTRUCTION FOLLOWING CALL
000564' 000205                             RTS     R5               ; RETURN, TIME-OUT
                                   ;       ------------------------
```

```
000566' 004767  000234          INTER:  JSR     R7,ERSUB                ; SAVE ADR AND CONTENTS OF CONTROL REG.
000572' 042777  000100  000514          BIC     #BIT6,@CDST             ; DISABLE INTERUPT
                                        ;----------------------------------------------------------------
000600' 000004  000000' 000606'         PIRQS,BEGIN,1$                  ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;----------------------------------------------------------------

000606' 005767  177270          1$:     TST     ACSR                    ; ANY ERRORS ?
000612' 100003                          BPL     2$                      ; NO, CONTINUE
000614' 004767  000224                  JSR     PC,ERRORS               ; YES, GO TO ERROR ROUTINE
000620' 000500                          BR      7$                      ; RETURN
000622' 032767  000010  177252  2$:     BIT     #BIT3,ACSR              ; TRANSITION TO ON-LINE ?
000630' 001074                          BNE     7$                      ; YES, GO GET A CARD
000632' 105777  000456                  TSTB    @CDST                   ; READY ?
000636' 100407                          BMI     3$                      ; YES, CONTINUE
000640' 012767  000011  177240          MOV     #11,ERRTYP              ;ILLEGAL INTERRUPT
                                        ;****************************************************************
000646' 104405  000000' 000000          HRDERS,BEGIN,NULL               ; INTERUPT OCCURED BUT NO REASON FOUND
                                        ;****************************************************************
000654' 000462                          BR      7$                      ; GO TRY ANOTHER CARD
000656' 005002                  3$:     CLR     R2                      ; ZERO REG. 2
000660' 012703  000120                  MOV     #80.,R3                 ; LOAD COUNTER
000664' 012701  001346'                 MOV     #BUFF,R1                ; GET BUFFER ADDRESS
000670' 032767  000002  177204          BIT     #BIT1,ACSR              ; PACKING MODE ?
000676' 001024                          BNE     5$                      ; YES, GO CHECK
                                ;       ------------------------
000700' 062102                  4$:     ADD     (R1)+,R2                ; SUM UP THE BUFFER
000702' 005303                          DEC     R3                      ; DONE ?
000704' 001375                          BNE     4$                      ; NO, KEEP ADDING
000706' 020277  000420                  CMP     R2,@ISUM                ;++  IS SUM CORRECT ?
000712' 001443                          BEQ     7$                      ; YES, GET ANOTHER CARD
                                ;       ------------------------
000714' 016767  000412  177160          MOV     ISUM,SBADR              ;++ NO, LOAD GOOD SUM ADDRESS
000722' 017767  000404  177156          MOV     @ISUM,ASB               ;++ LOAD GOOD SUM
000730' 012767  177702  177146          MOV     #177702,WASADR          ; LOAD ADDRESS OF REG. 2
000736' 010267  177146                  MOV     R2,AWAS                 ; LOAD BAD SUM
                                        ;****************************************************************
000742' 104404  000000'                 DATERS,BEGIN                    ;DATA ERROR!!!
                                        ;****************************************************************
                                                                        ; BAD CHECKSUM IN IMAGE MODE
000746' 000425                          BR      7$                      ; GO TRY ANOTHER CARD
                                ;       ------------------------


000750' 005000                  5$:     CLR     R0                      ; ZERO REG. 0
000752' 112100                  6$:     MOVB    (R1)+,R0                ; GET A BYTE
000754' 060002                          ADD     R0,R2                   ; SUM UP THE BUFFER
000756' 005303                          DEC     R3                      ; DONE ?
000760' 001374                          BNE     6$                      ; NO, KEEP ADDING
000762' 020277  000346                  CMP     R2,@PSUM                ;++ IS SUM CORRECT ?
000766' 001415                          BEQ     7$                      ; YES, GET ANOTHER CARD
```



```
000770' 016767  000340  177104          MOV     PSUM,SBADR              ;++ NO, LOAD GOOD SUM ADDRESS
000776' 017767  000332  177102          MOV     @PSUM,ASB               ;++ LOAD GOOD SUM
001004' 012767  177702  177072          MOV     #177702,WASADR          ; LOAD ADDRESS OF REG. 2
001012' 010267  177072                  MOV     R2,AWAS                 ; LOAD BAD SUM
                                        ;****************************************************************
001016' 104404  000000'                 DATERS,BEGIN                    ;DATA ERROR!!!
                                        ;****************************************************************
                                                                        ; BAD CHECKSUM IN PACKING MODE
001022' 000167  177402          7$:     JMP     BACK                    ; RETURN TO MAIN PGM
                                ;       ------------------------


001026' 016767  000262  177044  ERSUB:  MOV     CDST,CSRA               ; SAVE ADDRESS OF CONT. STAT. REG.
001034' 017767  000254  177040          MOV     @CDST,ACSR              ; SAVE CONTENTS OF CONT. STAT. REG.
001042' 000207                          RTS     PC                      ; RETURN
                                ;       ------------------------


001044' 016700  177032          ERRORS: MOV     ACSR,R0                 ; LOAD REG. 0 WITH STATUS
001050' 032700  040000                  BIT     #BIT14,R0               ; READER CHECK ?
001054' 001023                          BNE     2$                      ; YES
001056' 032700  004000                  BIT     #BIT11,R0               ; DATA ERROR ?
001062' 001053                          BNE     6$                      ; YES
001064' 032700  002000                  BIT     #BIT10,R0               ; DATA LATE ?
001070' 001057                          BNE     7$                      ; YES
001072' 032700  001000                  BIT     #BIT9,R0                ; NON-EXISTENT MEMORY ?
001076' 001063                          BNE     8$                      ; YES
001100' 032700  010000                  BIT     #BIT12,R0               ; OFF-LINE ?
001104' 001401                          BEQ     1$                      ; NO, REPORT ERROR
001106' 000467                          BR      10$                     ; YES, GO TRY ANOTHER CARD
001110' 005067  176772          1$:     CLR     ERRTYP                  ;UNKNOWN
                                        ;****************************************************************
001114' 104405  000000' 000000          HRDERS,BEGIN,NULL               ; ERROR BIT WAS SET ... OTHERS WEREN'T
                                        ;****************************************************************
001122' 000461                          BR      10$                     ; GO TRY ANOTHER CARD
001124' 032700  020000          2$:     BIT     #BIT13,R0               ; END OF FILE ?
001130' 001416                          BEQ     5$                      ; NO, MUST BE READ CHECK
001132' 122767  000117  000166          CMPB    #79.,CRDCNT             ; YES, ALL CARDS READ ?
001140' 001003                          BNE     4$                      ; NO, REPORT THE ERROR
001142' 105267  000161          3$:     INCB    FLAG                    ; SET END OF PASS FLAG
001146' 000447                          BR      10$                     ; RETURN
001150' 012767  000012  176730  4$:     MOV     #12,ERRTYP              ;EOF
                                        ;****************************************************************
001156' 104405  000000' 000000          HRDERS,BEGIN,NULL               ; END OF FILE ENCOUNTERED BUT NOT ENOUGH TEST CARDS READ
                                        ;****************************************************************
001164' 000436                          BR      9$                      ; GO TRY ANOTHER CARD
```

```
                                ;*****************************************************************
001166' 122767  000117  000132  5$:     CMPB    #79.,CRDCNT     ; END OF PASS ?
001174' 001762                  BEQ     3$              ; YES, GO SET THE FLAG
001176' 005087  176704          CLR     ERRTYP          ;UNKNOWN
                                ;*****************************************************************
001202' 104405  000000' 000000  MRDER$,BEGIN,NULL       ; HOPPER,PICK,STACK,AND/OR READ CHECK
                                ;*****************************************************************
001210' 000424                  BR      9$              ; GO TRY ANOTHER CARD
001212' 012767  000001  176666  6$:     MOV     #1,ERRTYP       ;DATA ERROR
                                ;*****************************************************************
001220' 104405  000000' 000000  MRDER$,BEGIN,NULL       ; PACKING MODE DATA ERROR
                                ;*****************************************************************
001226' 000417                  BR      10$             ; GO TRY ANOTHER CARD
001230' 012767  000002  176650  7$:     MOV     #2,ERRTYP       ;DATA LATE
                                ;*****************************************************************
001236' 104405  000000' 000000  MRDER$,BEGIN,NULL       ; DATA LATE ERROR
                                ;*****************************************************************
001244' 000410                  BR      10$             ; GO TRY ANOTHER CARD
001246' 012767  000010  176632  8$:     MOV     #10,ERRTYP      ;NON-EXISTENT MEMORY
                                ;*****************************************************************
001254' 104405  000000' 000000  MRDER$,BEGIN,NULL       ; NON-EXISTENT MEMORY
                                ;*****************************************************************
001262' 105367  000040          9$:     DECB    CRDCNT          ; DON'T COUNT A CARD
001266' 000207                  10$:    RTS     PC              ; RETURN
                                ;       ------------------------
                                ;++THIS ROUTINE IS CALLED FROM THE INITAL START-UP CODE TO CHECK
                                ;++FOR ECO #CD11-00014 THAT USES THE UPPER BITS IN THE DATA BUFFER
                                ;++REGISTER FOR ADDITIONAL ERROR FLAGS. IT TESTS BIT15 IN THE "CDDB"
                                ;++REGISTER AND IF FOUND ON A ONE THE ECO IS ASSUMED TO BE
                                ;++INSTALLED AND THE POINTERS TO THE CORRECT CHECKSUMS ARE
                                ;++CHANGED TO POINT TO DIFFERENT CHECKSUMS.

001270' 005777  000026          ECO14:  TST     @CDDB           ;++ IS THE ECO INSTALLED ??
001274' 100006                  BPL     1$              ;++ BR IF NOT
001276' 012767  001342' 000026          MOV     #ISUM8,ISUM     ;++ CHANGE THE CHECKSUM POINTERS
001304' 012767  001344' 000022          MOV     #PSUM8,PSUM     ;++
001312' 000207                  1$:     RTS     PC              ;++ RETURN TO CONTINUE START-UP

001314' 000000                  CDST:   0                       ; HOLDS ADDR OF CONTROL STATUS REG.
001316' 000000                  CDCC:   0                       ; HOLDS ADDR OF COLUMN COUNT REG.
001320' 000000                  CDBA:   0                       ; HOLDS ADDR OF CURRENT ADDR REG.
001322' 000000                  CDDA:   0                       ; HOLDS ADDR OF DATA BUFFER REG.
001324' 000000                  STATUS: 0                       ; HOLDS STATUS OF THE READER
001326'     000                 CRDCNT: .BYTE                   ; CARD COUNT
001327'     000                 FLAG:   .BYTE                   ; HOLDS FLAG BITS
001330' 000000                  CLK:    0                       ; CLOCK COUNTER
001332' 001336'                 ISUM:   ISUMA                   ;++ ADDRESS POINTER TO CKSUM
001334' 001340'                 PSUM:   PSUMA                   ;++ ADDRESS POINTER TO CKSUM
001336' 067443                  ISUMA:  67443                   ;++ IMAGE SUM FOR 80 COLUMNS
001340' 174173                  PSUMA:  174173                  ;++ PACKED SUM FOR 80 COLUMNS
001342' 117443                  ISUMB:  117443                  ;++ IMAGE SUM FOR 80 COLUMNS (ECO #14 INSTALLED)
001344' 174173                  PSUMB:  174173                  ;++ PACKED SUM FOR 80 COLUMNS (ECO #14 INSTALLED)
001346' 000120                  BUFF:   .BLKW   80.             ; INPUT BUFFER --- 80 WORDS LONG
001606' 000000                  RDADR:  0                       ; READ BUFFER VIRTUAL ADDRESS
```



```
001610' 000000                  RDPA:   0                       ; READ BUFFER PHYSICAL ADDRESS
001612' 000000                  RDEA:   0                       ; EXTENDED MEMORY BITS
                                ;       ------------------------
000001                          .END
```

```
ACSR      001102R       186#   205    299    311    356*   363
ADDR      000052R       152#   207
ADDR2 =   001000        204#
ASB       001106R       190#   321*   342*
ASTAT     001104R       188#
ATAB      001110R       191#   323*   344*
BACK      000430R       247#   349
BEGIN     000000R       119#   226    231    235    243    252    279    290    291    305    325    346    377
                        336    397    402    407    412
BIT0   =  000001        204#   247
BIT1   =  000002        204#   250    252    254    311
BIT10  =  002000        204#   368
BIT11  =  004000        204#   356
BIT12  =  010000        204#   273    372
BIT13  =  020000        204#   330
BIT14  =  040000        204#   364
BIT15  =  100000        204#
BIT2   =  000004        204#   275
BIT3   =  000010        204#   294
BIT4   =  000020        204#
BIT5   =  000040        204#
BIT6   =  000100        204#   239
BIT7   =  000200        204#
BIT8   =  000400        204#   270
BIT9   =  001000        204#   370
BREAKS=   104407        204#   279    290
BS1       000012R       154#   216
BS2       000013R       155#
BTODS  =  104421        204#
BUFF      001346R       205    310    445#
CDATAS=   104412        204#
CDBA      001320R       212*   239*   433#
CDCC      001316R       210*   238*   432#
CDDB      001322R       214*   425    434#
CDSR      001314R       235*   242*   270*   271    273    275    289*   301    355    356    431#
CLK       001330R       259*   281*   438#
CINFIG    000055R       174#
CPOCNT    001326R       220*   223    249*   382    393    414*   436#
CSRA      000100R       134#   355*
DATCKS=   104411        204#
DATERS=   104404        204#   325    346
DVID1     000014R       150#
ECO14     001270R       215    425#
ENDITS=   104413        204#   262
ENDS   =  104410        204#   235
ERRORS    001044R       297    353#
ERRTYP    000106R       199#   229*   303*   375*   386*   395*   400*   405*   410*
ERSUB     001026R       228    286    355#
EXITS  =  104100        204#   243
FINI      000362R       234#
FLAG      001327R       247    384*   437#
GETPAS=   104415        204#   206
GVBUFS=   104414        204#
HRDCNT    000044R       159#
HRDERS=   104405        204#   231    305    377    388    397    402    407    412
HRDPAS    000050R       171#
```



```
ICONT     000036R       156#
ICOUNT    000040R       157#
IDNUM     000122R       196#
INIT      000030R       153#
INTER     000566R       217    288#
INTR      000120R       195#
ISUM      001332R       317    320    321    427*   439#
ISUMA     001336R       439    441#
ISUMB     001342R       427    443#
MAP22S=   104416        204#
MODMAP    000000R       150#
MODSP     000224R       154    202#
MSGNS  =  104403        204#
MSGSS  =  104402        204#
MSGS   =  104401        204#
NUCARD    000322R       223#   253    255
NULL   =  000000        204#   231    305    377    388    397    402    407    412
OPEN   =  000000        151    157    158    159    160    177    178    179    180    181    182    183    184
                        186    188    190    191    193    194    195    204#
OTUAS  =  104420        204#
PASCNT    000034R       165#
PASS      000474R       225    240    261#
PIRQS  =  000004        204#   291
POPSP  =  005726        204#
POPSP2=   022626        204#
PRTY   =  000000        155    204#
PRTY0  =  000000        204#
PRTY1  =  000040        204#
PRTY2  =  000100        204#
PRTY3  =  000140        204#
PRTY4  =  000200        204#
PRTY5  =  000240        204#
PRTY6  =  000300        154    204#
PRTY7  =  000340        204#
PS     =  177776        204#
PSUM      001334R       337    341    342    428*   440#
PSUMA     001340R       440    442#
PSUMB     001344R       428    444#
PSW    =  177776        204#
PUSH   =  005746        204#
PUSH2  =  024646        204#
RANDS  =  104417        204#
RANNUM    000054R       173#
RDADR     001606R       205*   206    416#
RDEA      001612R       240    448#
RDPA      001610R       239    447#
READ      000366R       227    236#
READY     000500R       226    269#
RESTRT    000224R       192    205#
RES1      000056R       175#
RES2      000060R       176#
RSTRT     000112R       132#
SRADR     000102R       185#   320*   341*
SOFCNT    000042R       158#
SOFERS=   104406        204#
SOFPAS    000046R       170#
```

```
SPOINT  000032R          164#
SPSIZ = 000040             1#     197
SR1     000016R          157#
SR2     000020R          158#
SR3     000022R          159#
SR4     000024R          160#
START   000224R          193      204#
STAT    000026R          162#
STATUS  001324R          221*     240*    241*    242     250     252*    254*    435#
SVR0    000062R          177#
SVR1    000064R          178#
SVR2    000066R          179#
SVR3    000070R          180#
SVR4    000072R          181#
SVR5    000074R          182#
SVR6    000076R          183#
SYSCNT  000052R          172#
TRPDFN= 000022           204#
VECTOR  000010R          153#     216
WASADR  000104R          197#     322*    343*
WDFR    000116R          194#
WDTO    000114R          193#
XFLAG   000005R          151#
.       = 001614R        445#

. ABS.  000000     000
        001614     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XCDAGO,XCDAGO/SQL/CRF:SYM=DDXCOM,XCDAGO
 RUN-TIME:  1  1.3 SECONDS
 RUN-TIME RATIO: 10/3=2.8
 CORE USED:  7K  (13 PAGES)
```