```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E917B-MC

PRODUCT NAME: CXAACR0 AAV11 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) 1976,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT:

"AAC" IS A BKMOD THAT EXERCISES THE AAV11 DIGITAL TO ANALOG CONVERTER. A CONFIDENCE LOGIC TEST IS EXECUTED ON THE DAC0, DAC1, DAC2 AND DAC3 REGISTERS. ALL LOGIC ERRORS ARE REPORTED TO THE CONSOLE TELETYPE.

2. REQUIREMENTS:

HARDWARE: AAV11 INTERFACE MODULE

STORAGE:: AAC REQUIRES:
1. DECIMAL WORDS: 399
2. OCTAL WORDS: 0617
3. OCTAL BYTES: 1436

3. PASS DEFINITION:

ONE PASS OF THE AAC MODULE CONSISTS OF FLOATING A 1 AND A 0 ACCROSS THE FOUR D TO A REGISTERS 3000(8) TIMES. THIS RESULTS IN 340,000 BUS REFERENCES TO THE AAV11 OPTION

4. EXECUTION TIME:

VARIES WITH THE NUMBER OF OTHER DEVICES BEING RUN. THIS SHOULD TAKE AN AVERAGE OF FIVE SECONDS TO COMPLETE ONE PASS WHEN RUNNING ALONE.

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 170440, VCT: N/A, BR1: N/A

REQUIRED PARAMETERS:

NONE

6. DEVICE OPTION SETUP:

NONE.

SEQ 0003

7. MODULE OPERATION:

START/RESTART:

THIS CODE WILL USE THE VALUE CONTAINED IN LOCATION "ADDR" TO BE THE BASE ADDRESS OF THE AAV11. THE BUS ADDRESS OF EACH DAC IS PRIMED IN THIS ROUTINE. THE INITIAL PASS COUNTER IS ALSO PRESET.

TSDAC0:

THE ABILITY OF DAC 0 REGISTER TO HOLD A FLOATING 1 PATTERN IS VERIFIED IN THIS CODE. BIT 11 OF THE REGISTER IS INITILLY SET (4000) AND THEN ROTATED TO THE RIGHT. UPON COMPLETION, THE SAME PROCEDURE IS REPEATED EXCEPT THE INITIAL VALUE IS 3777.

TSDAC1:, TSDAC2:, TSDAC3:

SAME AS TSDAC0

DUAL:

THIS ROUTINE WILL LOAD DIFFERENT DATA INTO EACH REGISTER AND VERIFY INDEPENDANT ADDRESS SELECTION.

DONE:

IF NOT THE PROGRAM
WILL LOOP TO LOCATION "LOOPA" AND REPEAT THE SEQUENCE.
WHEN THE PASS COUNT HAS BEEN COMPLETED, THE "END OF PASS" IS REPORTED.

8. OPERATOR OPTIONS:

A. LOCATION (ICONT) CAN BE MODIFIED TO VARY THE NO. LOOPS THRU TEST BEFORE END OF PASS IS REPORTED.

9. NON-STANDARD PRINTOUTS:

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

.LIST SEQ,BIN

SEQ 0004

```
                        BKMOD   <AACB >,170440,,,,3000,140
                        MODULE  40020,AACB ,170440,,,,3000,140
                        .TITLE  AACB DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6       23-MAY-78
                        .LIST   BIN
                ;*****************************************************************************
000000'         BEGIN:
000000' 040501  MODNAM: .ASCII  /AACB / ;MODULE NAME.
000005'     000 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 170440  ADDR:   170440+0                ;1ST DEVICE ADDR.
000010' 000000  VECTOR: +0                      ;1ST DEVICE VECTOR.
000012'     000 BR1:    .BYTE   PRTY+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' 040020  STAT:   40020                   ;STATUS WORD.
000030' 000234' INIT:   START                   ;MODULE START ADDR.
000032' 000224' SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000  PASCNT: 0                       ;PASS COUNTER.
000036' 003000  ICONT:  3000                    ;# OF ITERATIONS PER PASS=3000
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  ASR:    OPEN                    ;EXPECTED DATA.
000110' 000000  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000234' 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' 000140  IDNUM:  140                     ;MODULE IDENTIFICATION NUMBER=140
000040                  .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                        .NLIST
```


SEQ 0005

```
                        .WORD   0
                        .LIST
                        .ENDR
000224'         MODSP:
                ;*****************************************************************************

                ;DEVICE BUS ADDRESS

000224' 000006' DAC0:   ADDR
000226' 000010' DAC1:   ADDR+2                  ;BUS ADDRESS OF DAC 0
000230' 000012' DAC2:   ADDR+4                  ;BUS ADDRESS FOR DAC 1
000232' 000014' DAC3:   ADDR+6
```

```
                        ;INITILIZATION CODE
                        ;
000234'                 START:
000234' 016767 177546 177762 RESTRT: MOV     ADDR,DAC0               ;LOAD BUS ADDRESS
000242' 016767 177756 177756         MOV     DAC0,DAC1               ; FOR
000250' 062767 000002 177750         ADD     #2,DAC1
000256' 016767 177744 177744         MOV     DAC1,DAC2               ;  DIFFERENT
000264' 062767 000002 177736         ADD     #2,DAC2
000272' 016767 177732 177732         MOV     DAC2,DAC3               ;   DAC
000300' 062767 000002 177724         ADD     #2,DAC3                 ;BUS ADDRESSES
                        ;
000306'                 LOOPA:
000306' 012767 010000 177566 TSDAC0: MOV     #BIT12,ACSR             ;LOAD EXPECTED
000314' 012777 004000 177702         MOV     #BIT11,@DAC0            ;LOAD DAC0 REGISTER
000322' 016767 177676 177550         MOV     DAC0,CSRA               ;LOAD BUS ADDRESS
000330' 006267 177546       1$:      ASR     ACSR                    ;SHIFT THE EXPECTED
000334' 001051                       BNE     4$                      ;BR IF DONE
000336' 017767 177662 177540         MOV     @DAC0,ASTAT             ;READ THE REGISTER
000344' 026767 177532 177532         CMP     ACSR,ASTAT              ;COMPARE
000352' 001406                       BEQ     2$                      ;BR IF SAME
000354' 012767 000025 177524         MOV     #25,ERRTYP
                        ;*************************************************************
000362' 104405 000000' 000000        HRDERS,BEGIN,NULL       ;DAC0 FAILED TO HOLD THE FLOATING 1 PATTERN
                        ;*************************************************************
000370' 005167 177506       2$:      COM     ACSR                    ;COMPLEMENT DATA
000374' 005177 177624                COM     @DAC0                   ;COMPLEMENT DATA IN DAC0
000400' 042767 170000 177474         BIC     #170000,ACSR            ;MASK OFF UNUSED BITS
000406' 017767 177612 177470         MOV     @DAC0,ASTAT             ;READ DAC0
000414' 026767 177462 177462         CMP     ACSR,ASTAT              ;COMPARE
000422' 001406                       BEQ     3$                      ;BR IF SAME
000424' 012767 000025 177454         MOV     #25,ERRTYP
                        ;*************************************************************
000432' 104405 000000' 000000        HRDERS,BEGIN,NULL       ;DAC0 FAILED TO HOLD THE FLOATING 0 PATTERN
                        ;*************************************************************
000440' 005167 177436       3$:      COM     ACSR                    ;COMPLEMENT EXPECTED
000444' 005177 177554                COM     @DAC0                   ;COMPLEMENT DATA
000450' 042767 170000 177424         BIC     #170000,ACSR            ;MASK EXPECTED DATA
000456' 000724                       BR      1$                      ;TEST MORE BITS
                        ;
000460'                 4$:
```



```
000460' 012767 010000 177414 TSDAC1: MOV     #BIT12,ACSR             ;LOAD EXPECTED
000466' 012777 004000 177532         MOV     #BIT11,@DAC1            ;LOAD DAC1 REGISTER
000474' 016767 177526 177376         MOV     DAC1,CSRA               ;LOAD BUS ADDRESS
000502' 006267 177374       1$:      ASR     ACSR                    ;SHIFT THE EXPECTED
000506' 001051                       BNE     4$                      ;BR IF DONE
000510' 017767 177512 177366         MOV     @DAC1,ASTAT             ;READ THE REGISTER
000516' 026767 177360 177360         CMP     ACSR,ASTAT              ;COMPARE
000524' 001406                       BEQ     2$                      ;BR IF SAME
000526' 012767 000025 177352         MOV     #25,ERRTYP
                        ;*************************************************************
000534' 104405 000000' 000000        HRDERS,BEGIN,NULL       ;DAC1 FAILED TO HOLD THE FLOATING 1 PATTERN
                        ;*************************************************************
000542' 005167 177334       2$:      COM     ACSR                    ;COMPLEMENT DATA
000546' 005177 177454                COM     @DAC1                   ;COMPLEMENT DATA IN DAC1
000552' 042767 170000 177322         BIC     #170000,ACSR            ;MASK OFF UNUSED BITS
000560' 017767 177442 177316         MOV     @DAC1,ASTAT             ;READ DAC1
000566' 026767 177310 177310         CMP     ACSR,ASTAT              ;COMPARE
000574' 001406                       BEQ     3$                      ;BR IF SAME
000576' 012767 000025 177302         MOV     #25,ERRTYP
                        ;*************************************************************
000604' 104405 000000' 000000        HRDERS,BEGIN,NULL       ;DAC1 FAILED TO HOLD THE FLOATING 0 PATTERN
                        ;*************************************************************
000612' 005167 177264       3$:      COM     ACSR                    ;COMPLEMENT EXPECTED
000616' 005177 177404                COM     @DAC1                   ;COMPLEMENT DATA
000622' 042767 170000 177252         BIC     #170000,ACSR            ;MASK EXPECTED DATA
000630' 000724                       BR      1$                      ;TEST MORE BITS
                        ;
000632'                 4$:
000632' 012767 010000 177242 TSDAC2: MOV     #BIT12,ACSR             ;LOAD EXPECTED
000640' 012777 004000 177362         MOV     #BIT11,@DAC2            ;LOAD DAC2 REGISTER
000646' 016767 177356 177224         MOV     DAC2,CSRA               ;LOAD BUS ADDRESS
000654' 006267 177222       1$:      ASR     ACSR                    ;SHIFT THE EXPECTED
000660' 001051                       BNE     4$                      ;BR IF DONE
000662' 017767 177342 177214         MOV     @DAC2,ASTAT             ;READ THE REGISTER
000670' 026767 177206 177206         CMP     ACSR,ASTAT              ;COMPARE
000676' 001406                       BEQ     2$                      ;BR IF SAME
000700' 012767 000025 177200         MOV     #25,ERRTYP
                        ;*************************************************************
000706' 104405 000000' 000000        HRDERS,BEGIN,NULL       ;DAC2 FAILED TO HOLD THE FLOATING 1 PATTERN
                        ;*************************************************************
000714' 005167 177162       2$:      COM     ACSR                    ;COMPLEMENT DATA
000720' 005177 177304                COM     @DAC2                   ;COMPLEMENT DATA IN DAC2
000724' 042767 170000 177150         BIC     #170000,ACSR            ;MASK OFF UNUSED BITS
000732' 017767 177272 177144         MOV     @DAC2,ASTAT             ;READ DAC2
000740' 026767 177136 177136         CMP     ACSR,ASTAT              ;COMPARE
000746' 001406                       BEQ     3$                      ;BR IF SAME
000750' 012767 000025 177130         MOV     #25,ERRTYP
                        ;*************************************************************
000756' 104405 000000' 000000        HRDERS,BEGIN,NULL       ;DAC2 FAILED TO HOLD THE FLOATING 0 PATTERN
                        ;*************************************************************
000764' 005167 177112       3$:      COM     ACSR                    ;COMPLEMENT EXPECTED
000770' 005177 177234                COM     @DAC2                   ;COMPLEMENT DATA
000774' 042767 170000 177100         BIC     #170000,ACSR            ;MASK EXPECTED DATA
001002' 000724                       BR      1$                      ;TEST MORE BITS
                        ;
001004'                 4$:
```

```
001004' 012767  010000  177070  TSDAC3: MOV     #BIT12,ACSR             ;LOAD EXPECTED
001012' 012777  004000  177212          MOV     #BIT11,@DAC3            ;LOAD DAC3 REGISTER
001020' 016767  177206  177052          MOV     DAC3,CSRA               ;LOAD BUS ADDRESS
001026' 006267  177050          1$:     ASR     ACSR                    ;SHIFT THE EXPECTED
001032' 001051                          BNE     4$                      ;BR IF DONE
001034' 017767  177172  177042          MOV     @DAC3,ASTAT             ;READ THE REGISTER
001042' 026767  177034  177034          CMP     ACSR,ASTAT              ;COMPARE
001050' 001406                          BEQ     2$                      ;BR IF SAME
001052' 012767  000025  177026          MOV     #25,ERRTYP
                                        ;*************************************************************
001060' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DAC3 FAILED TO HOLD THE FLOATING 1 PATTERN
                                        ;*************************************************************
001066' 005167  177010          2$:     COM     ACSR                    ;COMPLEMENT DATA
001072' 005177  177134                  COM     @DAC3                   ;COMPLEMENT DATA IN DAC3
001076' 042767  170000  176776          BIC     #170000,ACSR            ;MASK OFF UNUSED BITS
001104' 017767  177122  176772          MOV     @DAC3,ASTAT             ;READ DAC3
001112' 026767  176764  176764          CMP     ACSR,ASTAT              ;COMPARE
001120' 001406                          BEQ     3$                      ;BR IF SAME
001122' 012767  000025  176756          MOV     #25,ERRTYP
                                        ;*************************************************************
001130' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DAC3 FAILED TO HOLD THE FLOATING 0 PATTERN
                                        ;*************************************************************
001136' 005167  176740          3$:     COM     ACSR                    ;COMPLEMENT EXPECTED
001142' 005177  177064                  COM     @DAC3                   ;COMPLEMENT DATA
001146' 042767  170000  176726          BIC     #170000,ACSR            ;MASK EXPECTED DATA
001154' 000724                          BR      1$                      ;TEST MORE BITS

001156'                         4$:

                                ;TEST FOR DUAL ADDRESSING

001156' 012777  000000  177040  DUAL:   MOV     #0,@DAC0                ;LOAD DAC 0
001164' 012777  002525  177034          MOV     #2525,@DAC1             ;LOAD DAC 1
001172' 012777  005252  177030          MOV     #5252,@DAC2             ;LOAD DAC 2
001200' 012777  007777  177024          MOV     #7777,@DAC3             ;LOAD DAC 3

001206' 016767  177012  176664          MOV     DAC0,CSRA               ;LOAD DAC 0 BUS ADDRESS
001214' 017767  177004  176662          MOV     @DAC0,ASTAT             ;READ DAC 0
001222' 012767  000000  176652          MOV     #0,ACSR                 ;LOAD EXPECTED
001230' 026767  176646  176646          CMP     ACSR,ASTAT              ;COMPARE
001236' 001405                          BEQ     1$                      ;BR IF SAME
001240' 005067  176642                  CLR     ERRTYP
                                        ;*************************************************************
001244' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DUAL ADDRESS ERROR, ON DAC 0
                                        ;*************************************************************

001252' 016767  176750  176620  1$:     MOV     DAC1,CSRA               ;LOAD DAC 1 BUS ADDRESS
001260' 017767  176742  176616          MOV     @DAC1,ASTAT             ;READ DAC 1
001266' 012767  002525  176606          MOV     #2525,ACSR              ;LOAD EXPECTED
001274' 026767  176602  176602          CMP     ACSR,ASTAT              ;COMPARE
001302' 001405                          BEQ     2$                      ;BR IF SAME
001304' 005067  176576                  CLR     ERRTYP
                                        ;*************************************************************
001310' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DUAL ADDRESS ERROR, ON DAC 1
                                        ;*************************************************************
```



```
001316' 016767  176706  176554  2$:     MOV     DAC2,CSRA               ;LOAD DAC 2 BUS ADDRESS
001324' 017767  176700  176552          MOV     @DAC2,ASTAT             ;READ DAC 2
001332' 012767  005252  176542          MOV     #5252,ACSR              ;LOAD EXPECTED
001340' 026767  176536  176536          CMP     ACSR,ASTAT              ;COMPARE
001346' 001405                          BEQ     3$                      ;BR IF SAME
001350' 005067  176532                  CLR     ERRTYP
                                        ;*************************************************************
001354' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DUAL ADDRESS ERROR, ON DAC 2
                                        ;*************************************************************

001362' 016767  176644  176510  3$:     MOV     DAC3,CSRA               ;LOAD DAC 3 BUS ADDRESS
001370' 017767  176636  176506          MOV     @DAC3,ASTAT             ;READ DAC 3
001376' 012767  007777  176476          MOV     #7777,ACSR              ;LOAD EXPECTED
001404' 026767  176472  176472          CMP     ACSR,ASTAT              ;COMPARE
001412' 001405                          BEQ     4$                      ;BR IF SAME
001414' 005067  176466                  CLR     ERRTYP                  ;UNDEFINED ERROR
                                        ;*************************************************************
001420' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DUAL ADDRESS ERROR, ON DAC 3
                                        ;*************************************************************

001426'                         4$:

001426'                         DONE:
001426' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
001432' 000167  176650                  JMP     LOOPA                   ;JUMP IF NOT

        000001                          .END
```

```
INIT    000030R          169#
INTR    000120R          201#
LOOPA   000306R          230#      396
MAP22S= 104416           210#
MODNAM  000000R          156#
MODSP   000224R          170       208#
MSGN$ = 104403           210#
MSGS$ = 104402           210#
MSG$  = 104401           210#
NULL  = 000000           210#      241   251   269   279   297   307   325   335   358   368   378   388
OPEN  = 000000           157       163   164   165   166   183   184   185   186   187   188   189   190
                         192       194   196   197   199   200   201   210#
OTOA$ = 104420           210#
PASCNT  000034R          171#
PIRQ$ = 000004           210#
POPSP = 005726           210#
POPSP2= 022626           210#
PRTY  = 000000           160       161   210#
PRTY0 = 000000           210#
PRTY1 = 000040           210#
PRTY2 = 000100           210#
PRTY3 = 000140           210#
PRTY4 = 000200           210#
PRTY5 = 000240           210#
PRTY6 = 000300           210#
PRTY7 = 000340           210#
PS    = 177776           210#
PSW   = 177776           210#
PUSH  = 005746           210#
PUSH2 = 024646           210#
RAND$ = 104417           210#
RANNUM  000054R          179#
RESTRT  000234R          198       222#
RES1    000056R          181#
RES2    000060R          182#
RSTRT   000112R          196#
SRADR   000102R          191#
SOFCNT  000042R          174#
SOFERS= 104406           210#
SOPPAS  000046R          176#
SPOINT  000032R          170#
SPSIZ = 000040             1#      203
SR1     000016R          163#
SR2     000020R          164#
SR3     000022R          165#
SR4     000024R          166#
START   000234R          169       221#
STAT    000026R          168#
SVR0    000062R          183#
SVR1    000064R          184#
SVR2    000066R          185#
SVR3    000070R          186#
SVR4    000072R          187#
SVR5    000074R          188#
SVR6    000076R          189#
SVSCNT  000052R          178#
```



```
TRPDFD= 000022           210#
TSDAC0  000306R          231#
TSDAC1  000460R          259#
TSDAC2  000632R          287#
TSDAC3  001004R          315#
VECTOR  000010R          159#
WASADR  000104R          193#
WDER    000116R          200#
WDTO    000114R          199#
XFLAG   000005R          157#

. ABS.  000000     000
        001436     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XAACB0,XAACB0/SOL/CRF:SYM=DDXCOM,XAACB0
 RUN-TIME:  1  1  .3 SECONDS
 RUN-TIME RATIO: 11/3=3.3
 CORE USED:  7K  (13 PAGES)
```