```
1
          .REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E667J-MC

PRODUCT NAME: CXCPBJ0 EIS EXER MOD

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT:

CPB IS A BKMOD THAT EXERCISES THE EIS IN THE PDP11/45 AND THE PDP11/40.

2. REQUIREMENTS:

HARDWARE: ANY PDP11/45 OR PDP11/40 WITH EIS OPTION.
STORAGE:: CPB REQUIRES:
1. DECIMAL WORDS: 471
2. OCTAL WORDS: 0727
3. OCTAL BYTES: 1656

3. PASS DEFINITION:

ONE PASS OF THE CPB MODULE CONSISTS OF EXECUTING EACH INSTRUCTION 12500 TIMES.

4. EXECUTION TIME:

CPB RUNNING ALONE ON A PDP11/45 TAKES APPROXIMATELY 30 SECONDS.

5. CONFIGURATION REQUIREMENTS:

11/40 OR LSI WITHOUT EIS OPTION, SEE SECTION 8.

6. DEVICE/OPTION SETUP:

MAKE SURE EIS IS INSTALLED.

7. MODULE OPERATION:

A. SETUP CYCLE COUNTER.
B. TEST ALL EIS INSTRUCTIONS UNLESS SR1 BIT#1 = 1.
   IF SR1 BIT#1 = 1, CHECK ONLY XOR,SOB,MARK,SXT, AND RTT.
C. IF NOT EOP,GO TO B.
D. ELSE DO EOP AND GO TO A.

8. OPERATING OPTIONS:

SR1

BIT #0=0	DO ALL EIS INSTRUCTIONS
BIT #0=1	11/40, LSI WITHOUT EIS (11V03) DO ONLY XOR,SOB,MARK,SXT, AND RTT.
BIT #2=0	ACCESS PROCESSOR STATUS DIRECTLY.
BIT #2=1	USE MFPS INSTRUCTION.

I.E.

SR1=0	ALL EIS, PSW DIRECT.
SR1=1	NO EIS, PSW DIRECT.
SR1=4	ALL EIS, USE MFPS
SR1=5	NO EIS, USE MFPS.

9. NON-STANDARD PRINTOUTS:

NONE

```
                        BKMOD <CPBJ >,,,,,,12500,2
                MODULE  40020,CPBJ ,,,,,,12500,2
                .TITLE  CPBJ DEC/X11 SYSTEM EXERCISER MODULE
        ;       DDXCOM  VERSION  6      23-MAY-76
                        .LIST     BIN
                ;*****************************************************************************
                BEGIN:
000000  050103  045102      040   MODNAM: .ASCII  /CPBJ / ;MODULE NAME.
000005  000                     XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006  000000                  ADDR:   +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  000224'                 INIT:   START                   ;MODULE START ADDR.
000032  000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034  000000                  PASCNT: 0                       ;PASS COUNTER.
000036  012500                  ICONT:  12500                   ;# OF ITERATIONS PER PASS=12500
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                          SRADR:                          ;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  000002                  IDNUM:  2                       ;MODULE IDENTIFICATION NUMBER=2
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
```



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

```
                                ;
000224'                         START:
000224'                         RPSTRT:
000224'                                 ENDIT
000224' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
                                ;
000230' 012702  000010          XOR1:   MOV     #10,R2          ;SET UP INDEX REGISTER
000234' 012701  052525                  MOV     #52525,R1       ;LOAD SOURCE OPERAND
000240' 012767  177777  001342          MOV     #-1,TEMP        ;LOAD DESTINATION OPERAND
000246' 074162  001600'                 XOR     R1,C(R2)        ;XOR SOURCE,DEST.
000252' 022767  125252  001330          CMP     #125252,TEMP
000260' 001403                          BEQ     1$
000262'                                 HRDER ,                 <XOR INSTRUCTION FAILED>
                                        ;*****************************************************************
            001                         .IF B <>
000262' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;XOR INSTRUCTION FAILED
                                        .IFF
                                        HRDERS,BEGIN,   ;XOR INSTRUCTION FAILED
            000                         .ENDC
                                        ;*****************************************************************
                                ;
                                ;SBTTL  TEST XOR USING INDEX DEFERRED
                                ;
000270' 012703  125252          1$:     MOV     #125252,R3
000274' 012704  000010                  MOV     #10, R4
000300' 012767  001624' 001330          MOV     #Y, X+10
000306' 012767  177777  001310          MOV     #-1, Y
000314' 074374  001626'                 XOR     R3, @X(R4)
000320' 022767  052525  001276          CMP     #52525, Y
000326' 001403                          BEQ     2$
000330'                                 HRDER ,                 <XOR -1,125252 FAILED>
                                        ;*****************************************************************
            001                         .IF B <>
000330' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;XOR -1,125252 FAILED
                                        .IFF
                                        HRDERS,BEGIN,   ;XOR -1,125252 FAILED
            000                         .ENDC
                                        ;*****************************************************************
                                ;
                                ;SBTTL  TEST SOB FOR BRANCH
                                ;
000336' 000401                  2$:     BR      SOB1
000340' 000406                  SOB2:   BR      SOB3
000342' 012701  000010          SOB1:   MOV     #10,R1
000346' 077104                          SOB     R1,SOB2
000350'                                 HRDER ,                 <SOB SHOULD HAVE BRANCHED>
                                        ;*****************************************************************
            001                         .IF B <>
000350' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SOB SHOULD HAVE BRANCHED
                                        .IFF
                                        HRDERS,BEGIN,   ;SOB SHOULD HAVE BRANCHED
            000                         .ENDC
                                        ;*****************************************************************
                                ;
                                ;SBTTL  TEST SOB FOR NO BRANCH
```



```
                                ;
000356' 000404                  SOB3:   BR      SOB4
000360'                         SOB5:
                                        ;*****************************************************************
            001                         .IF B <>
000360' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SOB SHOULD NOT HAVE BRANCHED
                                        .IFF
                                        HRDERS,BEGIN,   ;SOB SHOULD NOT HAVE BRANCHED
            000                         .ENDC
                                        ;*****************************************************************
000366' 000403                          BR      MA
000370' 012701  000001          SOB4:   MOV     #1,R1
000374' 077107                          SOB     R1,SOB5
                                ;
                                ;SBTTL  TEST MARK INSTRUCTION
                                ;
000376' 016706  177430          MA:     MOV     SPOINT,SP       ;INITIALIZE STACK POINTER
000402' 005001                          CLR     R1
000404' 012746  000007                  MOV     #7,-(SP)
000410' 012746  000001                  MOV     #1,-(SP)        ;PUSH A PARAMETER ON THE STACK
000414' 012746  006401                  MOV     #MARK1,-(SP)    ;PUSH MARK      1 ON THE STACK
000420' 010605                          MOV     SP,R5           ;LOAD PARAMETER POINTER
000422' 004767  000002                  JSR     PC,MARK0        ;GO TO SUBROUTINE
000426' 000403                          BR      MARK0A
000430' 016501  000002          MARK0:  MOV     2(R5),R1        ;GET THE PARAMETER
000434' 000205                          RTS     R5              ;EXIT SUBROUTINE
000436' 022701  000001          MARK0A: CMP     #1,R1           ;DID SUBROUTINE GET THE PARAMETER
000442' 001403                          BEQ     1$
000444'                                 HRDER ,                 <SUBROUTINE DID NOT GET PUSHED PARAMETER>
                                        ;*****************************************************************
            001                         .IF B <>
000444' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SUBROUTINE DID NOT GET PUSHED PARAMETER
                                        .IFF
                                        HRDERS,BEGIN,   ;SUBROUTINE DID NOT GET PUSHED PARAMETER
            000                         .ENDC
                                        ;*****************************************************************
000452' 026706  177354          1$:     CMP     SPOINT,SP       ;IS STACK POINTER CORRECT?
000456' 001403                          BEQ     2$
000460'                                 HRDER ,                 <STACK POINTER INCORRECT FOR MARK>
                                        ;*****************************************************************
            001                         .IF B <>
000460' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;STACK POINTER INCORRECT FOR MARK
                                        .IFF
                                        HRDERS,BEGIN,   ;STACK POINTER INCORRECT FOR MARK
            000                         .ENDC
                                        ;*****************************************************************
000466' 022705  000007          2$:     CMP     #7,R5           ;IS R5 CORRECT?
000472' 001403                          BEQ     3$
000474'                                 HRDER ,                 <MARK DID NOT LOAD R5>
                                        ;*****************************************************************
            001                         .IF B <>
000474' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;MARK DID NOT LOAD R5
                                        .IFF
                                        HRDERS,BEGIN,   ;MARK DID NOT LOAD R5
            000                         .ENDC
```

```
                                        ;*****************************************************************
                                ;SBTTL  TEST THE SXT INSTRUCTION
000502' 012702  000010          3$:     MOV     #10,R2          ;SET UP INDEX REGISTER
000506' 012767  177777  001074          MOV     #-1,TEMP
000514' 000257                          CCC                     ;CLEAR ALL CONDITION CODES
000516' 006762  001600'                 SXT     C(R2)           ;EXTEND 0'S INTO TEMP [C(R2)]
000522' 005767  001062                  TST     TEMP
000526' 001403                          BEQ     4$
000530'                                 HRDER   <0'S DOID NOT EXTEND INTO TEMP>
                                        ;*****************************************************************
                                        .IF B <>
000530' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;0'S DOID NOT EXTEND INTO TEMP
                                        .IFF
                                        HRDERS,BEGIN,   ;0'S DOID NOT EXTEND INTO TEMP
                                        .ENDC
                                        ;*****************************************************************
000536' 005067  001046          4$:     CLR     TEMP
000542' 000277                          SCC                     ;SET ALL CONDITION CODES
000544' 006767  001040                  SXT     TEMP            ;EXTEND 1'S INTO TEMP
000550' 022767  177777  001032          CMP     #-1,TEMP
000556' 001403                          BEQ     5$
000560'                                 HRDER   <-1 DID NOT EXTEND INTO TEMP>
                                        ;*****************************************************************
                                        .IF B <>
000560' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;-1 DID NOT EXTEND INTO TEMP
                                        .IFF
                                        HRDERS,BEGIN,   ;-1 DID NOT EXTEND INTO TEMP
                                        .ENDC
                                        ;*****************************************************************
                                ;SBTTL  TEST THE RTT INSTRUCTION
000566' 016706  177240          5$:     MOV     SPOINT,SP       ;RESET THE STACK POINTER
000572' 032767  000004  177216          BIT     #4,SR1          ;ACCESS PSW DIRECT?
000600' 001403                          BEQ     6$              ;YES
000602' 000257                          CCC                     ;CLEAR STATUS
000604' 106746                          MFPS    -(SP)           ;PUSH ON STACK FOR RTT
000606' 000403                          BR      7$              ;SET UP PC FOR RTT
000610' 000257                  6$:     CCC                     ;CLR COND CODES
000612' 016746  177776          7$:     MOV     PS,-(SP)        ;PUSH IT ON STACK AS PS FOR RTT
000616' 012746  000642'                 MOV     #RTTA,-(SP)     ;PUSH RETURN PC ONTO STACK
000622' 000277                          SCC                     ;INSURE N=Z=V=C=1 BEFORE RTT
000624' 000006                          RTT
000626'                                 HRDER   <RTT FAILED>
                                        ;*****************************************************************
                                        .IF B <>
000626' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;RTT FAILED
                                        .IFF
                                        HRDERS,BEGIN,   ;RTT FAILED
                                        .ENDC
                                        ;*****************************************************************
000634' 005726                          TST     (SP)+           ;POP STACK
000636' 005726                          TST     (SP)+           ;POP STACK
```



```
000640' 000407                          BR      RTTB            ;GO ON WITH TEST
000642'                         RTTA:
000642' 100403                          BMI     10$             ;IF N SET,REPORT ERROR
000644' 101402                          BLOS    10$             ;IF C OR Z SET,REPORT ERROR
000646' 102401                          BVS     10$             ;IF V SET,REPORT ERROR
000650' 000403                          BR      RTTB            ;IF YES GO ON WITH TEST
000652'                         10$:
000652'                                 HRDER   <INCORRECT STATUS>
                                        ;*****************************************************************
                                        .IF B <>
000652' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT STATUS
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT STATUS
                                        .ENDC
                                        ;*****************************************************************
000660' 032767  000001  177130  RTTB:   BIT     #1,SR1
000666' 001402                          BEQ     MUL1
000670' 000167  177330                  JMP     START

000674' 005003                  MUL1:   CLR     R3              ;CLEAR R3
000676' 012702  000005                  MOV     #5,R2           ;LOAD MULTIPICAND
000702' 005067  000746                  CLR     PLIER
000706' 012767  000002  000740          MOV     #2,PLIER        ;LOAD MULTIPLIER
000714' 000277                          SCC                     ;PRESET CONDITION CODES
000716' 070267  000732                  MUL     PLIER,R2        ;MULTIPLY 2X5 RESULT IN R2 (MSH);R3 (LSH)
000722' 100402                          BMI     10$             ;IF N SET,REPORT ERROR
000724' 101401                          BLOS    10$             ;IF C OR Z SET,REPORT ERROR
000726' 102003                          BVC     1$              ;IF V = 0,GO ON WITH TEST
000730'                         10$:
000730'                                 HRDER   <CC'S NOT = TO 0'S>
                                        ;*****************************************************************
                                        .IF B <>
000730' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CC'S NOT = TO 0'S
                                        .IFF
                                        HRDERS,BEGIN,   ;CC'S NOT = TO 0'S
                                        .ENDC
                                        ;*****************************************************************
000736' 022703  000012          1$:     CMP     #12,R3          ;RESULT =12?
000742' 001403                          BEQ     2$              ;BRANCH IF THE RESULT (LSH) IS CORRECT
000744'                                 HRDER   <INCORRECT RESULT (LSH)>
                                        ;*****************************************************************
                                        .IF B <>
000744' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT RESULT (LSH)
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT RESULT (LSH)
                                        .ENDC
                                        ;*****************************************************************
000752' 005702                  2$:     TST     R2              ;CORRECT RESULT (MSH)?
000754' 001403                          BEQ     MUL2
000756'                                 HRDER   <INCORRECT RESULT (MSH)>
                                        ;*****************************************************************
                                        .IF B <>
000756' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT RESULT (MSH)
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT RESULT (MSH)
```

```
                        000                             .ENDC
                                                        ;*****************************************************************
                                                        ;
                                                        ;
000764' 005003                          MUL2:   CLR     R3
000766' 005067  000662                          CLR     PLIER
000772' 012702  125252                          MOV     #125252,R2      ;LOAD MULTIPICAND
000776' 012767  000002  000650                  MOV     #2,PLIER
001004' 070267  000644                          MUL     PLIER,R2        ;MULTIPLY 2X125252 RESULT=-1(R2)
                                                                        ;52524(R3),N=1,Z=0,V=0,C=1
001010' 103003                                  BCC     10$             ;IF C = 0,REPORT ERROR
001012' 100002                                  BPL     10$             ;IF N = 0,REPORT ERROR
001014' 102401                                  BVS     10$             ;IF V SET,REPORT ERROR
001016' 001003                                  BNE     1$              ;IF Z = 0,GO ON WITH TEST
001020'                                 10$:
001020'                                         HRDER ,                 <INCORRECT CONDITION CODES>
                                                        ;*****************************************************************
                        001                             .IF B <>
001020' 104405  000000' 000000                  HRDERS,BEGIN,NULL       ;INCORRECT CONDITION CODES
                                                .IFF
                                                HRDERS,BEGIN,   ;INCORRECT CONDITION CODES
                        000                     .ENDC
                                                        ;*****************************************************************
001026' 022702  177777          1$:     CMP     #-1,R2          ;CORRECT MSH RESULT?
001032' 001403                          BEQ     2$
001034'                                 HRDER ,                 <INCORRECT MSH RESULT>
                                                        ;*****************************************************************
                        001                             .IF B <>
001034' 104405  000000' 000000                  HRDERS,BEGIN,NULL       ;INCORRECT MSH RESULT
                                                .IFF
                                                HRDERS,BEGIN,   ;INCORRECT MSH RESULT
                        000                     .ENDC
                                                        ;*****************************************************************
001042' 022703  052524          2$:     CMP     #52524,R3       ;CORRECT LSH RESULT?
001046' 001403                          BEQ     ASH1
001050'                                 HRDER ,                 <INCORRECT LSH RESULT>
                                                        ;*****************************************************************
                        001                             .IF B <>
001050' 104405  000000' 000000                  HRDERS,BEGIN,NULL       ;INCORRECT LSH RESULT
                                                .IFF
                                                HRDERS,BEGIN,   ;INCORRECT LSH RESULT
                        000                     .ENDC
                                                        ;*****************************************************************
                                                        ;
                                        .SBTTL  TEST ASH INSTRUCTION
                                                        ;
001056' 005067  000572          ASH1:   CLR     PLIER
001062' 012767  000001  000564          MOV     #1,PLIER        ;LOAD SHIFT VALUE (+1 OR 1 PLACE LEFT)
001070' 012702  125252                  MOV     #125252,R2      ;GET VALUE TO BE SHIFTED (#125252)
001074' 000257                          CCC                     ;PRE-CLEAR ALL CC BITS
001076' 000264                          SEZ                     ;PRE-SET Z BIT
001100' 000270                          SEN                     ;PRE-SET N BIT
001102' 072267  000546                  ASH     PLIER,R2        ;SHIFT 1 PLACE LEFT RESULT = 52524
                                                                ;N=0,Z=0,V=1,C=1
001106' 100403                          BMI     10$             ;IF N SET,REPORT ERROR
001110' 001402                          BEQ     10$             ;IF Z SET,REPORT ERROR
```



```
001112' 102001                          BVC     10$             ;IF V = 0,REPORT ERROR
001114' 103403                          BCS     1$              ;IF C SET,GO ON WITH TEST
001116'                                 10$:
001116'                                 HRDER ,                 <INCORRECT CONDITION CODES>
                                                        ;*****************************************************************
                        001                             .IF B <>
001116' 104405  000000' 000000                  HRDERS,BEGIN,NULL       ;INCORRECT CONDITION CODES
                                                .IFF
                                                HRDERS,BEGIN,   ;INCORRECT CONDITION CODES
                        000                     .ENDC
                                                        ;*****************************************************************
001124' 022702  052524          1$:     CMP     #52524,R2       ;RESULT CORRECT?
001130' 001403                          BEQ     ASH2
001132'                                 HRDER ,                 <INCORRECT RESULT>
                                                        ;*****************************************************************
                        001                             .IF B <>
001132' 104405  000000' 000000                  HRDERS,BEGIN,NULL       ;INCORRECT RESULT
                                                .IFF
                                                HRDERS,BEGIN,   ;INCORRECT RESULT
                        000                     .ENDC
                                                        ;*****************************************************************
                                                        ;
001140' 005067  000510          ASH2:   CLR     PLIER
001144' 012767  177777  000502          MOV     #-1,PLIER       ;LOAD SHIFT VALUE (-1 OR 1 PLACE RIGHT)
001152' 012701  052525                  MOV     #52525,R1       ;GET VALUE TO BE SHIFTED (#52525)
001156' 072167  000472                  ASH     PLIER,R1        ;SHIFT 1 PLACE RIGHT RESULT = #25252
001162' 020127  025252                  CMP     R1,#25252       ;RESULT CORRECT?
001166' 001403                          BEQ     ASHC1
001170'                                 HRDER ,                 <INCORRECT RESULT>
                                                        ;*****************************************************************
                        001                             .IF B <>
001170' 104405  000000' 000000                  HRDERS,BEGIN,NULL       ;INCORRECT RESULT
                                                .IFF
                                                HRDERS,BEGIN,   ;INCORRECT RESULT
                        000                     .ENDC
                                                        ;*****************************************************************
                                                        ;
                                        ;SBTTL  TEST ASHC INSTRUCTION
                                                        ;
001176' 005002                          ASHC1:  CLR     R2              ;CLEAR MSH RESULT REGISTER
001200' 005067  000450                  CLR     PLIER
001204' 012767  000020  000442          MOV     #16.,PLIER      ;LOAD SHIFT COUNT (16 PLACES LEFT)
001212' 012703  125252                  MOV     #125252,R3      ;GET VALUE TO BE SHIFTED (#125252)
001216' 000257                          CCC                     ;PRE-CLEAR ALL CC BITS
001220' 000264                          SEZ                     ;PRE-SET Z BIT
001222' 000261                          SEC                     ;PRE-SET C BIT
001224' 073267  000424                  ASHC    PLIER,R2        ;SHIFT # IN R3 TO R2
                                                                ;CC: N=1,Z=0,V=1,C=0
001230' 100003                          BPL     10$             ;IF N = 0,REPORT ERROR
001232' 001402                          BEQ     10$             ;IF Z SET,REPORT ERROR
001234' 102001                          BVC     10$             ;IF V = 0,REPORT ERROR
001236' 103003                          BCC     1$              ;IF C = 0,GO ON WITH TEST
001240'                                 10$:
001240'                                 HRDER ,                 <INCORRECT CONDITION CODES>
                                                        ;*****************************************************************
                        001                             .IF B <>
```

```
                                        HRDERS,BEGIN,NULL       ;INCORRECT CONDITION CODES
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT CONDITION CODES
                    000                 .ENDC
                                        ;*****************************************************************
001246' 022702  125252          1$:     CMP     #125252,R2      ;R2=#125252?
001252' 001403                          BEQ     2$              ;BRANCH IF CORRECT RESULT
001254'                                 HRDER ,                 <INCORRECT RESULT FOR ASHC LEFT SHIFT>
                                        ;*****************************************************************
                    001                 .IF B <>
001254' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT RESULT FOR ASHC LEFT SHIFT
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT RESULT FOR ASHC LEFT SHIFT
                    000                 .ENDC
                                        ;*****************************************************************
001262' 005703                  2$:     TST     R3              ;WAS # SHIFTED OUT OF R3?
001264' 001403                          BEQ     ASHC2
001266'                                 HRDER ,                 <# WAS NOT SHIFTED OUT OF R3 PROPERLY>
                                        ;*****************************************************************
                    001                 .IF B <>
001266' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;# WAS NOT SHIFTED OUT OF R3 PROPERLY
                                        .IFF
                                        HRDERS,BEGIN,   ;# WAS NOT SHIFTED OUT OF R3 PROPERLY
                    000                 .ENDC
                                        ;*****************************************************************
001274' 005003                  ASHC2:  CLR     R3              ;CLEAR RESULT REGISTER
001276' 005067  000352                  CLR     PLIER
001302' 012767  177760  000344          MOV     #-16.,PLIER     ;LOAD SHIFT COUNT (16 PLACES RIGHT)
001310' 012702  125252                  MOV     #125252,R2      ;LOAD # TO BE SHIFTED (#125252)
001314' 000250                          CLN                     ;PRE-CLEAR N BIT
001316' 000264                          SEZ                     ;PRE-SET Z BIT
001320' 000262                          SEV                     ;   "     V  "
001322' 000261                          SEC                     ;   "     C  "
001324' 073267  000324                  ASHC    PLIER,R2        ;SHIFT R2 16 PLACES RIGHT INTO R3
                                                                ;CC: N=1,Z=V=C=0
001330' 100002                          BPL     10$             ;IF N = 0,REPORT ERROR
001332' 101401                          BLOS    10$             ;IF C OR Z SET,REPORT ERROR
001334' 102003                          BVC     1$              ;IF V = 0,GO ON WITH TEST
001336'                         10$:
001336'                                 HRDER ,                 <INCORRECT CONDITION CODES>
                                        ;*****************************************************************
                    001                 .IF B <>
001336' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT CONDITION CODES
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT CONDITION CODES
                    000                 .ENDC
                                        ;*****************************************************************
001344' 022702  177777          1$:     CMP     #-1,R2          ;DID SIGN EXTEND IN R2?
001350' 001403                          BEQ     2$
001352'                                 HRDER ,                 <SIGN FAILED TO EXTEND>
                                        ;*****************************************************************
                    001                 .IF B <>
001352' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SIGN FAILED TO EXTEND
                                        .IFF
                                        HRDERS,BEGIN,   ;SIGN FAILED TO EXTEND
```



```
                    000                 .ENDC
                                        ;*****************************************************************
001360' 022703  125252          2$:     CMP     #125252,R3      ;DID R2 SHIFT TO R3
001364' 001403                          BEQ     DIV1
001366'                                 HRDER ,                 <R2 DID NOT SHIFT INTO R3>
                                        ;*****************************************************************
                    001                 .IF B <>
001366' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;R2 DID NOT SHIFT INTO R3
                                        .IFF
                                        HRDERS,BEGIN,   ;R2 DID NOT SHIFT INTO R3
                    000                 .ENDC
                                        ;*****************************************************************

                                ;SBTTL  TEST THE DIVIDE INSTRUCTION

001374' 012701  000004          DIV1:   MOV     #4,R1           ;LOAD INDEX REGISTER
001400' 005002                          CLR     R2              ;CLEAR QUOTIENT REGISTER
001402' 012703  052525                  MOV     #52525,R3       ;LOAD LSH DIVIDEND
001406' 000277                          SCC                     ;PRE SET THE CONDITION CODES
001410' 071261  001556'                 DIV     8(R1),R2        ;DIVIDE #52525 BY 8(R1) (#52525)
                                                                ;QUOTIENT=1,REM=0,C=N=V=Z=0
001414' 100402                          BMI     10$             ;IF N SET,REPORT ERROR
001416' 101401                          BLOS    10$             ;IF C OR Z SET,REPORT ERROR
001420' 102003                          BVC     1$              ;IF V = 0,GO ON WITH TEST
001422'                         10$:
001422'                                 HRDER ,                 <INCORRECT CONDITION CODES>
                                        ;*****************************************************************
                    001                 .IF B <>
001422' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT CONDITION CODES
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT CONDITION CODES
                    000                 .ENDC
                                        ;*****************************************************************
001430' 020227  000001          1$:     CMP     R2,#1           ;QUOTIENT CORRECT?
001434' 001403                          BEQ     2$              ;BRANCH IF THE QUOTIENT IS CORRECT
001436'                                 HRDER ,                 <INCORRECT QUOTIENT>
                                        ;*****************************************************************
                    001                 .IF B <>
001436' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT QUOTIENT
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT QUOTIENT
                    000                 .ENDC
                                        ;*****************************************************************
001444' 005703                  2$:     TST     R3              ;REMAINDER CORRECT?
001446' 001403                          BEQ     DIV2            ;BRANCH IF THE REMAINDER IS CORRECT
001450'                                 HRDER ,                 <INCORRECT REMAINDER>
                                        ;*****************************************************************
                    001                 .IF B <>
001450' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INCORRECT REMAINDER
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT REMAINDER
                    000                 .ENDC
                                        ;*****************************************************************

001456' 005067  000172          DIV2:   CLR     PLIER
001462' 012704  157777                  MOV     #157777,R4      ;LOAD MSH DIVIDEND
```

```
                                        MOV     #100001,R5      ;LOAD LSH DIVIDEND
                                        MOV     #100000,PLIER   ;LOAD DIVISOR INTO TEMP
                                        SCC                     ;PRE-SET ALL CC BITS
                                        DIV     PLIER,R4        ;DIVIDE #157777 100001 BY 100000
                                                                ;QUOTIENT=40000,REM=100001,C=N=V=Z=0
                                        BMI     10$             ;IF N SET,REPORT ERROR
                                        BLOS    10$             ;IF C OR Z SET,REPORT ERROR
                                        BVC     1$              ;IF V = 0,GO ON WITH TEST
                                10$:
                                        HRDER                   <INCORRECT CONDITION CODES>
                                        ;*****************************************************************
                                        .IF B <>
                                        HRDERS,BEGIN,NULL       ;INCORRECT CONDITION CODES
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT CONDITION CODES
                                        .ENDC
                                        ;*****************************************************************
                                1$:     CMP     R4,#40000       ;QUOTIENT CORRECT?
                                        BEQ     2$              ;BRANCH IF THE QUOTIENT IS CORRECT
                                        HRDER                   <INCORRECT QUOTIENT>
                                        ;*****************************************************************
                                        .IF B <>
                                        HRDERS,BEGIN,NULL       ;INCORRECT QUOTIENT
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT QUOTIENT
                                        .ENDC
                                        ;*****************************************************************
                                2$:     CMP     R5,#100001      ;REMAINDER CORRECT?
                                        BEQ     3$              ;BRANCH IF THE REMAINDER IS CORRECT
                                        HRDER                   <INCORRECT REMAINDER>
                                        ;*****************************************************************
                                        .IF B <>
                                        HRDERS,BEGIN,NULL       ;INCORRECT REMAINDER
                                        .IFF
                                        HRDERS,BEGIN,   ;INCORRECT REMAINDER
                                        .ENDC
                                        ;*****************************************************************


001552' 000167  176446          3$:     JMP START
001556' 125252                  B:      125252
001560' 001556'                         B                       ;ADDRESS OF B
001562' 052525                          052525

                001566'                 .=B+10
001566' 177777                  A:      -1
001570' 001572'                         A+4

                001572'                 .=A+4
001572' 125252                          125252
001574' 001576'                         A+10                    ;ADDRESS OF A+10
001576' 052525                          052525
001600' 000000                  C:      0
001602' 001600'                         C                       ;ADDRESS OF C

                001610'                 .=C+10
```



```
001610' 000000                  TEMP:   0
001612' 001610'                         TEMP                    ;ADDRESS OF TEMP

                001616'                 .=TEMP+6
001616' 001620'                         TEMP+10                 ;ADDRESS OF TEMP+10 OR "D"
001620' 000000                  D:0
        006401                  MARK1=  6401
001622' 000000                  CNT:    0
        140000                  UM=140000

        177776                          PS=177776               ;PROCESSOR STATUS WORD
001624' 000000                          V:0
001626' 000013                  X:      .BLKW   11.
001654' 000000                          PLIER:0
        000001                          .END
```

```
A       001566R           666#    667     669     671
ACSR    000102R           155#
ADDR    000006R           121#
ADDR22= 001000            173#
ASB     000106R           159#
ASHC1   001176R           480     492#
ASHC2   001274R           525     535#
ASH1    001056R           431     443#
ASH2    001140R           465     475#
ASTAT   000104R           157#
AWAS    000110R           160#
B       001556R           584     661#    662     665
BEGIN   000000R           118#    177     189     207     222     234     261     271     281     298     312     334     351
                          374     384     394     415     425     435     459     469     484     509     519     529     552
                          562     572     593     603     613     633     643     653
BIT0  = 000001            173#
BIT1  = 000002            173#
BIT10 = 002000            173#
BIT11 = 004000            173#
BIT12 = 010000            173#
BIT13 = 020000            173#
BIT14 = 040000            173#
BIT15 = 100000            173#
BIT2  = 000004            173#
BIT3  = 000010            173#
BIT4  = 000020            173#
BIT5  = 000040            173#
BIT6  = 000100            173#
BIT7  = 000200            173#
BIT8  = 000400            173#
BIT9  = 001000            173#
BREAK$= 104407            173#
BR1     000012R           123#
BR2     000013R           124#
BTOD$ = 104421            173#
C       001600R           183*    292*    673#    674     676
CDATA$= 104412            173#
CNT     001622R           684#
CONFIG  000056R           143#
CSRA    000100R           153#
D       001620R           682#
DATCK$= 104411            173#
DATER$= 104404            173#
DIV1    001374R           568     580#
DIV2    001456R           609     619#
DVID1   000014R           125#
ENDIT$= 104413            173#    177
END$  = 104410            173#
ERRTYP  000106R           158#
EXIT$ = 104400            173#
GETPA$= 104415            173#
GWBUF$= 104414            173#
HRDCNT  000044R           138#
HRDER$= 104405            173#    189     207     222     234     261     271     281     298     312     334     351     374
                          384     394     415     425     435     459     469     484     509     519     529     552     562
                          572     593     603     613     633     643     653
```



```
HRDPAS  000050R           140#
ICONT   000036R           135#
ICOUNT  000040R           136#
IDNUM   000122R           165#
INIT    000036R           132#
INTR    000120R           164#
MA      000376R           239     246#
MAP22$= 104416            173#
MARK0   000430R           252     254#
MARK0A  000436R           253     256#
MARK1 = 006401            250     683#
MODNAM  000000R           119#
MODSP   000224R           133     171#
MSGN$ = 104403            173#
MSGS$ = 104402            173#
MSG$  = 104401            173#
MUL1    000674R           357     361#
MUL2    000764R           390     401#
NULL  = 000000            173#    189     207     222     234     261     271     281     298     312     334     351     374
                          384     394     415     425     435     459     469     484     509     519     529     552     562
                          572     593     603     613     633     643     653
OPEN  = 000000            120     126     127     128     129     146     147     148     149     150     151     152     153
                          155     157     159     160     162     163     164     173#
OTOA$ = 104420            173#
PASCNT  000034R           134#
PIRQ$ = 000004            173#
PLIER   001654R           363*    364*    366     402*    404*    405     443*    444*    449     475*    476*    478     493*
                          494*    499     536*    537*    543     619*    622*    624     690#
POPSP = 005726            173#
POPSP2= 022626            173#
PRTY  = 000000            123     124     173#
PRTY0 = 000000            173#
PRTY1 = 000040            173#
PRTY2 = 000100            173#
PRTY3 = 000140            173#
PRTY4 = 000200            173#
PRTY5 = 000240            173#
PRTY6 = 000300            173#
PRTY7 = 000340            173#
PS    = 177776            173#    327     687#
PSW   = 177776            173#
PUSH  = 005746            173#
PUSH2 = 024646            173#
RAND$ = 104417            173#
RANNUM  000054R           142#
RESTRT  000224R           161     175#
RES1    000056R           144#
RES2    000060R           145#
RSTRT   000112R           161#
RTTA    000642R           328     342#
RTTB    000660R           341     346     356#
SBADR   000102R           154#
SOB1    000342R           215     217#
SOB2    000340R           216#    218
SOB3    000356R           216     230#
SOB4    000370R           230     240#
```

```
SOB5    000360R          231#    241
SOFCNT  000042R          137#
SOFERS= 104406           173#
SOFPAS  000046R          139#
SPOINT  000032R          133#    246     266     370
SPSIZ = 000040             1#    166
SR1     000016R          126#    321     356
SR2     000020R          127#
SR3     000022R          128#
SR4     000024R          129#
START   000224R          132     174#    358     660
STAT    000026R          131#
SVR0    000062R          146#
SVR1    000064R          147#
SVR2    000066R          148#
SVR3    000070R          149#
SVR4    000072R          150#
SVR5    000074R          151#
SVR6    000076R          152#
SYSCNT  000052R          141#
TEMP    001610R          182*    184     290*    293     304*    306*    307     677#    678     680     681
TRPDFD= 000022           173#
UM    = 140000           685#
VECTOR  000010R          122#
WASADR  000104R          156#
WDFR    000116R          163#
WDTO    000114R          162#
X       001626R          199*    201*    689#
XFLAG   000005R          120#
XOR1    000230R          180#
Y       001624R          199     200*    202     688#
.     = 001656R          665#    669#    676#    680#    689#

. ABS.  000000      000
        001656      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XCPBJ0,XCPBJ0/SOL/CRF:SYM=DDXCOM,XCPBJ0
 RUN-TIME:  1  2  .3 SECONDS
 RUN-TIME RATIO: 12/4=3.0
 CORE USED:  7K  (13 PAGES)
```