

SXT

INSTRUCTION TEST
MD-11-DCKBA-B

EP-DCKBA-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBA TO DCKBE-B-D
 PRODUCT NAME: PDP11 45-11/40 BASIC CP TESTS
 DATE CREATED: 15 NOV 1972
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHORS: JOHN ADAMS

COPYRIGHT(c) 1972
 DIGITAL EQUIPMENT CORPORATION
 MAYNARD, MASS

THIS GROUP OF TESTS CONSIST OF:

MAINDEC NO.	TEST FUNCTION	PROCESSOR
DCKBA-A	SXT INSTRUCTION	11/45, 11/40
DCKBB-A	SOB INSTRUCTION	11/45, 11/40
DCKBC-A	XOR INSTRUCTION	11/45, 11/40
DCKBD-B	MARK INSTRUCTION	11/45, 11/40
DCKBE-B	RTT INSTRUCTION	11/45, 11/40

1.0 Abstract

This is the first 5 of 15 tests that incrementally test and isolate simple malfunctions in the PDP 11/45, 11/40. The tests should be run in the indicated alphabetic sequence. There are additional tests for more complex malfunctions. All tests are executed in kernel mode only except for test DCKBE (11/45 only).

2.0 Requirements

2.1 Equipment

PDP11/45 or PDP 11/40

2.2 Storage

Program storage - The programs use all of a 4KW memory with the exception of 17500 to 17776 (which is reserved for the boot and absolute loader).

2.3 Preliminary Programs

Tests T0-T13(DDAA-DJMA)

3.0 Loading Procedure

Load program into memory using ABS loader.

4.0 Starting Procedure

Load address 200. Press start. The program will loop, and ring bell on completion. Pass count may be monitored in the display register (11/45 only), and is stored in address 1000.

5.0 Operating Procedure

5.1 Operational Switch Settings

SW<08>=1 Load PDP11/45 MICRO BREAK register with value in SW<00-07>. (At start of test only).

5.2 Subroutine Abstracts

5.2.1 SCOPE

SCOPE is a MOVE PC, R1(010701) and stores the PC+2 in R1 (or R11 if the register set bit is set in the PDP11/45 Processor Status Word.) Thus, R1 (R11) may be used as a tag to determine the last test successfully completed.

5.2.2 HLT

HLT is a HALT instruction and is executed whenever a hardware malfunction is detected by the diagnostic. The address lights display the PC-PC+2 of the halt instruction if an 11/40-11/45.

5.3 Program and/or Operator Action

5.3.1 PASS COUNT (ICNT)

The number of program passes completed is contained in address ICNT (1000). This address may be examined to determine in which pass the error occurred. Note, the pass count is displayed in the display register in the PDP11/45.

6.0 Errors

6.1 Test Errors Will Cause a Halt

False trap/interrupt errors - the program will halt at the trap vector address +2.

6.2 Error Recovery

Test errors - press continue or loop test (see sec 6.3)

Trap errors - determine where error occurred (see sec 8)

6.3 Error Looping

To loop on an error replace the HLT instruction with a branch back to the previous SCOPE instruction. Note that if the error is intermittent that the test will drop through the HLT and proceed to the next test, therefore to loop the test continuously replace the BEQ .+4 instruction immediately preceding the HLT with the branch back to the previous SCOPE.

To loop trap failures patch in the following routine at the address of the trap vector:

TRAPVEC: TRAPVEC+4
TRAPVEC+2: 0
TRAPVEC+4: 012716 ;move 'SCOPE'
;address to stack
TRAPVEC+6: address ;address of previous
;SCOPE inst.
TRAPVEC+10: 000006 ;return to test at 'SCOPE'

Restore all locations before proceeding to next test(s).

7.0 Restrictions

These programs must be loaded in the lower 4K of memory.

7.1 Starting restriction

All programs must be initially started at 200 and may be started at a SCOPE instruction thereafter.

7.2 Operational restriction

None

8.0 Miscellaneous

If a halt occurs in the trap or interrupt vector area, examine register 6 (the stack pointer). The contents of R6 contains the address where the PC of the instruction that caused the trap is stored.

8.1 Execution time

All tests take approximately 1 minute each on an 11/45 with core memory.

8.2 Stack pointer

All programs initially set the stack pointer at 500

9.0 Program description

DCKBA This is a test of the SXT instruction and insures correct results and condition code operation. The SXT instruction is tested in all address modes in a general register and the PC.

DCKBB This is a test of the SOB instruction and insures correct branching and condition code operation.

DCKBC This is a test of the XOR instruction and insures correct results and condition code operation. The XOR instruction is executed using various operands and is executed using all address modes using a general register and the PC.

DCKBD This is a test of the MARK instruction. The test executes the MARK instruction using all values of 'N' and checks results. Correct condition code operation is also tested.

DCKBE This is a test of the RTT and RTI instructions and uses 'T' bit traps in the test. Proper stack operation is tested and also proper status changes are tested.

274

```
.TITLE MAINDEC-11-DCKBA-B PDP11/45 SXT INST TEST
.NLIST MC,MD,SEQ
.LIST ME
.ABS
;TEST DCKBAA- TEST OF THE SXT INSTRUCTION AND INCLUDES DATA CHECKS AND
;CONDITION CODES CHECKS.
```


000064	000066	.+2
000066	000000	HALT
000070	000072	.+2
000072	000000	HALT
000074	000076	.+2
000076	000000	HALT
000100	000102	.+2
000102	000000	HALT
000104	000106	.+2
000106	000000	HALT
000110	000112	.+2
000112	000000	HALT
000114	000116	.+2
000116	000000	HALT
000120	000122	.+2
000122	000000	HALT
000124	000126	.+2
000126	000000	HALT
000130	000132	.+2
000132	000000	HALT
000134	000136	.+2
000136	000000	HALT
000140	000142	.+2
000142	000000	HALT
000144	000146	.+2
000146	000000	HALT
000150	000152	.+2
000152	000000	HALT
000154	000156	.+2
000156	000000	HALT
000160	000162	.+2
000162	000000	HALT
000164	000166	.+2
000166	000000	HALT
000170	000172	.+2
000172	000000	HALT
000174	000176	.+2
000176	000000	HALT
000200	000202	.+2
000202	000000	HALT
000204	000206	.+2
000206	000000	HALT
000210	000212	.+2
000212	000000	HALT
000214	000216	.+2
000216	000000	HALT
000220	000222	.+2
000222	000000	HALT
000224	000226	.+2
000226	000000	HALT
000230	000232	.+2
000232	000000	HALT
000234	000236	.+2
000236	000000	HALT
000240	000242	.+2
000242	000000	HALT

000244	000246		.+2	
000246	000000		HALT	
000250	000252		.+2	
000252	000000		HALT	
000254	000256		.+2	
000256	000000		HALT	
000260	000262		.+2	
000262	000000		HALT	
000264	000266		.+2	
000266	000000		HALT	
000270	000272		.+2	
000272	000000		HALT	
000274	000276		.+2	
000276	000000		HALT	
000300	000302		.+2	
000302	000000		HALT	
000304	000306		.+2	
000306	000000		HALT	
000310	000312		.+2	
000312	000000		HALT	
000314	000316		.+2	
000316	000000		HALT	
000320	000322		.+2	
000322	000000		HALT	
000324	000326		.+2	
000326	000000		HALT	
000330	000332		.+2	
000332	000000		HALT	
000334	000336		.+2	
000336	000000		HALT	
000340	000342		.+2	
000342	000000		HALT	
000344	000346		.+2	
000346	000000		HALT	
000350	000352		.+2	
000352	000000		HALT	
000354	000356		.+2	
000356	000000		HALT	
000360	000362		.+2	
000362	000000		HALT	
000364	000366		.+2	
000366	000000		HALT	
000370	000372		.+2	
000372	000000		HALT	
000374	000376		.+2	
000376	000000		HALT	
000000	000000	001634	.=0	
	000167		JMP	SXT07
000046	000046		.=46	
	002604		LOGICAL	
000052	000052		.=52	
	040000		40000	

```

000200 000000 DEST=%0
000200 000200
000200 000167 000576      . =200
001000 001000      JMP START
001000 000000      . =1000
001002 005067 177772 ICNT: 0 ;PASS COUNT
001006 012706 000500 START: CLR ICNT ;CLEAR PASS COUNT
001012 016737 177762 177570 BEGIN: MOV #STKPTR,%6 ;SET STACK POINTER
001020 032737 000400 177570 MOV ICNT,%#DISPLAY ;DISPLAY PASS COUNT
001026 001403 BEQ #400,%#SWR ;LOAD MICRO BREAK REGISTER?
001030 113737 177570 177770 MOVB @#SWR,%#UBREAK ;BRANCH IF NOT
;LOAD MICRO BREAK WITH SRO-7

;TEST THAT WHEN N IS CLEAR SXT CLEARS THE DESTINATION
001036 010701
001040 012700 177777 SCOPE
001044 000277 MOV #-1,DEST
001046 000250 SCC ;SET ALL CONDITION CODES
001050 006700 CLN ;CLEAR N
001052 005700 SXT DEST ;EXTEND N BIT (0) INTO DESTINATION
001054 001401 TST DEST ;DESTINATION=0?
001056 000000 BEQ .+4
HLT ;DESTINATION OTHER THAN 0

;TEST THAT WHEN N IS SET THAT SXT SETS ALL BITS IN THE DESTINATION
001060 010701
001062 005000
001064 000257
001066 000270
001070 006700
001072 020027 177777 SCOPE
001076 001401 CMP DEST,#-1 ;CLEAR ALL CONDITON CODES
001100 000000 BEQ .+4 ;SET N
;EXTEND N BIT (1) INTO DESTINATION
;DID ALL BITS SET IN DESTINATION
;DESTINATION OTHER THAN 177777

;TEST THAT N IS CLEAR AFTER SXT EXTENDS 0'S INTO THE DESTINATION
001102 010701
001104 012700 177777 SCOPE
001110 000250 MOV #-1,DEST
001112 006700 CLN
001114 100001 SXT DEST
001116 000000 BPL .+4
HLT

;TEST THAT N IS SET AFTER SXT EXTENDS 1'S INTO THE DESTINATION
001120 010701
001122 005000
001124 000270
001126 006700
001130 100401
001132 000000 SCOPE
CLR DEST
SEN
SXT DEST
BMI .+4
HLT

;TEST THAT SIGN EXTENDS PROPERLY AFTER A MOVE INSTRUCTION (N SET)
001134 010701
001136 005000
001140 012767 177777 177774 SCOPE
001146 006700 MOV #-1,%+2
001150 020027 177777 SXT DEST
001154 001401 CMP DEST,#-1
BEQ .+4

```

MAINDEC-11-DCKBA-B FDP11/45 SXT INST TEST
DCKBAB.P11

001156 000000

K01

MACY11 27(732) 16-SEP-76 15:47 PAGE 10

HLT

;TEST THAT SIGN EXTENDS PROPERLY AFTER A MOVE INSTRUCTION (N CLEAR)

001160	010701			SCOPE	
001162	012700	177777		MOV	#-1,DEST
001166	012767	000000	177774	MOV	#0,+2
001174	006700			SXT	DEST
001176	020027	000000		CMP	DEST,#0
001202	001401			BEQ	.+4
001204	000000			HLT	

;TEST THAT SIGN EXTENDS PROPERLY AFTER VARIOUS BYTE INSTRUCTIONS
 ;NOTE: THESE TESTS MUST BE RUN SEQUENTIALLY OR TEMP & TEMP+2 MUST
 ;BE MANUALLY LOADED BEFORE STARTING THE TEST. THE COMMENT AT THE BE -
 ;INNING OF THE TEST SHOWS CONTENTS OF TEMP & TEMP+2.

001206	005000			CLR	DEST	;(TEMP+1),(TEMP)/(TEMP+3),(TEMP+2)
001210	005067	000226		CLR	TEMP	;'?'/?'
001214	105167	000223		COMB	TEMP+1	;'0'/'?'/?'
001220	006700			SXT	DEST	;-1,0/'?'/?'
001222	022700	177777		CMP	#-1,DEST	
001226	001401			BEQ	.+4	
001230	000000			HLT		;ERROR! SIGN EXTEND FAILED (N=1)
001232	010701			SCOPE		

001234	005000			CLR	DEST	;-1,0/'?'/?'
001236	105367	000200		DECB	TEMP	;-1,-1/'?'/?'
001242	006700			SXT	DEST	
001244	022700	177777		CMP	#-1,DEST	
001250	001401			BEQ	.+4	
001252	000000			HLT		;ERROR! SIGN EXTEND FAILED (N=1)
001254	010701			SCOPE		

001256	005000			CLR	DEST	;-1,-1/'?'/?'
001260	156767	000157	000156	BISB	TEMP+1,TEMP+2	;-1,-1/'?'/?',-1
001266	006700			SXT	DEST	
001270	022700	177777		CMP	#-1,DEST	
001274	001401			BEQ	.+4	
001276	000000			HLT		;ERROR! SIGN EXTEND FAILED (N=1)
001300	010701			SCOPE		

001302	005000			CLR	DEST	;-1,-1/'?'/?',-1
001304	116767	000134	000133	MOVB	TEMP+2,TEMP+3	;-1,-1/'?'/?',-1
001312	006700			SXT	DEST	
001314	022700	177777		CMP	#-1,DEST	
001320	001401			BEQ	.+4	
001322	000000			HLT		;ERROR! SIGN EXTEND FAILED (N=1)
001324	010701			SCOPE		

001326	012700	177777		MOV	#-1,DEST	;-1,-1/'?'/?',-1
001332	146767	000105	000105	BICB	TEMP+1,TEMP+3	;-1,-1/'0',-1
001340	006700			SXT	DEST	
001342	005700			TST	DEST	
001344	001401			BEQ	.+4	
001346	000000			HLT		;ERROR! SIGN EXTEND FAILED (N=0)
001350	010701			SCOPE		

```

001352 012700 177777      MOV      #-1,DEST      ; -1,-1/0,-1
001356 000261             SEC                ; SET CARRY
001360 105567 000057      ADCB     TEMP+1      ; 0,-1/0,-1
001364 006700             SXT     DEST
001366 005700             TST     DEST
001370 001401             BEQ     .+4
001372 000000             HLT
001374 010701             SCOPE                ; ERROR! SIGN EXTEND FAILED (N=0)

001376 012700 177777      MOV      #-1,DEST      ; 0,-1/0,-1
001402 105267 000034      INCB    TEMP         ; 0,0/0,-1
001406 006700             SXT     DEST
001410 005700             TST     DEST
001412 001401             BEQ     .+4
001414 000000             HLT
001416 010701             SCOPE                ; ERROR! SIGN EXTEND FAILED (N=0)

001420 012700 177777      MOV      #-1,DEST      ; 0,0/0,-1
001424 105167 000014      COMB    TEMP+2      ; 0,0/0,0
001430 006700             SXT     DEST
001432 005700             TST     DEST
001434 001404             BEQ     TZ
001436 000000             HLT
001440 010701             SCOPE                ; ERROR! SIGN EXTEN FAILED (N=0)
001442 000000             TEMP:0
001446 001446             .=.+2

;TEST THAT Z BIT IS SET AFTER SXT EXTENDS 0'S
TZ:
001446 010701             SCOPE
001450 012700 177777      MOV      #-1,DEST
001454 000250             CLN
001456 006700             SXT     DEST          ; CLEAR N
001460 001401             BEQ     .+4          ; EXTEND N BIT (0) INTO DESTINATION
001462 000000             HLT                ; BRANCH IF Z IS SET
                                ; Z NOT SET AFTER 0'S WERE EXTENDED

;TEST THAT Z BIT IS CLEAR AFTER SXT EXTENDS 1'S
001464 010701             SCOPE
001466 005000             CLR     DEST
001470 000270             SEN
001472 006700             SXT     DEST
001474 001001             BNE    .+4
001476 000000             HLT

;TEST THAT THE CARRY BIT (C) IS UNCHANGED BY SXT (C=0,N=0)
001500 010701             SCOPE
001502 000257             CCC
001504 006700             SXT     DEST          ; SLAER ALL CONDITION CODES
001506 103001             BCC    .+4          ; EXTEND SIGN
001510 000000             HLT                ; BRANCH IF CARRY IS CLEAR
                                ; ERROR CARRY SET

;TEST THAT THE CARRY BIT IS UNCHANGED BY SXT (C=0,N=1)
001512 010701             SCOPE
001514 000257             CCC
001516 000270             SEN
001520 006700             SXT     DEST          ; C=0,N=1
                                ; EXTEND SIGN

```



```

;TEST THAT V IS CLEARED BY SXT (C=1,N=1)
001616 010701          SCOPE
001620 000277          SCC
001622 006700          SXT          DEST          ;V=1,N=1
001624 102001          BVC          .+4          ;EXTEND SIGN
001626 000000          HLT          ;BRANCH IF V IS CLEAR
;ERROR! V REMAINED SET

000007
DEST=%7
;TEST THAT SIGN EXTENDS INTO R7 (N=0).
001630 010701          SCOPE
001632 000257          SCC
001634 006707          SXT          DEST          ;N=0
001636 000000          HLT          ;EXTEND 0'S INTO THE PC
001640 000402          SXT07: BR          MODE67          ;ERROR! PC SHOULD'VE GONE TO 0

001642
001644 000000          DEST=.
;=+2
IDEST: 0
;TEST DESTINATION MODE 67
MODE67: SCOPE
001646 010701          CLR          DEST
001650 005067          177766          SEN
001654 000270          SXT          DEST          ;SET N
001656 006767          177760          SXT          DEST          ;EXTEND 1'S INTO DEST
001662 026727          177754 177777          CMP          DEST,#-1          ;DID 1'S EXTEND
001670 001401          BEQ          .+4
001672 000000          HLT          ;1'S FAILED TO EXTEND

;TEST DESTINATION MODE 27
001674 010701          SCOPE
001676 005067          000004          CLR          MODE27
001702 000270          SEN
001704 006727          SXT          (7)+          ;SET N
001706 000000          MODE27: 0          ;EXTEND 1'S INTO NEXT LOCATION
001710 026727          177772 177777          CMP          MODE27,#-1          ;DID 1'S EXTEND
001716 001401          BEQ          .+4
001720 000000          HLT

;TEST DESTINATION MODE 37
001722 010701          SCOPE
001724 012737          177777 001642          MOV          #-1,2*DEST
001732 000250          CLN
001734 006737          001642          SXT          2*DEST          ;CLEAR N
001740 023727          001642 000000          CMP          2*DEST,#0          ;EXTEND 0'S
001746 001401          BEQ          .+4          ;DID 0'S EXTEND
001750 000000          HLT

;TEST DESTINATION MODE 77
001752 010701          SCOPE
001754 012767          001644 177660          MOV          #IDEST,DEST
001762 012777          177777 177652          MOV          #-1,2*DEST
001770 000250          CLN
001772 006777          177644          SXT          2*DEST
001776 027727          177640 000000          CMP          2*DEST,#0
002004 001401          BEQ          .+4
002006 000000          HLT

```

```

;TEST DESTINATION MODE 1
002010 010701
002012 012700 002320      SCOPE
002016 005002              MOV      #MODE1,%0
002020 000261              CLR      %2
002022 005602              SEC
002024 006710              SBC      %2
002026 022767 177777 000264  SXT      (0)
002034 001401              CMP      #-1,MODE1
002036 000000              BEQ      .+4
;ERROR! INCORRECT RESULT

;TEST DESTINATION MODE2
002040 010701
002042 005046
002044 012746 002056      SCOPE
002050 012702 002320      CLR      -(6)
002054 000002              MOV      #MODE2,-(6)
002056 006722              MOV      #MODE1,%2
002060 005767 000234      MODE2:  SXT      (2)+
002064 001401              TST      MODE1
002066 000000              BEQ      .+4
002070 022702 002322      HLT
002074 001401              CMP      #MODE1+2,%2
002076 000000              BEQ      .+4
;ERROR! INCORRECT RESULT
;CHECK AUTO-INCREMENT
;ERROR! AUTO INCREMENT FAILED

;TEST DESTINATION MODE 3
002100 010701
002102 012767 002322 000210  SCOPE
002110 012702 002320      MOV      #MODE3,MODE1
002114 000270              MOV      #MODE1,%2
002116 006732              SEN
002120 022767 177777 000174  SXT      2(2)+
002126 001401              CMP      #-1,MODE3
002130 000000              BEQ      .+4
;ERROR! INCORRECT RESULT

;TEST DESTINATION MODE 4
002132 010701
002134 012703 002322      SCOPE
002140 012767 177777 000152  MOV      #MODE3,%3
002146 005067 000150      MOV      #-1,MODE1
002152 006743              CLR      MODE3
002154 005767 000140      SXT      -(3)
002160 001401              TST      MODE1
002162 000000              BEQ      .+4
002164 022703 002320      HLT
002170 001401              CMP      #MODE3-2,%3
002172 000000              BEQ      .+4
;ERROR! INCORRECT RESULT
;CHECK AUTO-DECREMENT
;ERROR! AUTO-DECREMENT FAILED

;TEST DESTINATION MODE 5
002174 010701
002176 012767 002320 000116  SCOPE
002204 012704 002324      MOV      #MODE1,MODE3
002210 000270              MOV      #MODE3+2,%4
002212 006754              SEN
002214 022767 177777 000076  SXT      2-(4)
002222 001401              CMP      #-1,MODE1
;ERROR! INCORRECT RESULT
;CHECK AUTO-DECREMENT
;ERROR! AUTO-DECREMENT FAILED
    
```



```

002224 000000          HLT
002226 022704 002322  CMP      #MODE3,%4      ;ERROR! INCORRECT RESULT
002232 001401          BEQ      .+4             ;CHECK AUTO-DECREMENT
002234 000000          HLT             ;ERROR! AUTO-DECREMENT DEFERRED FAILED

```

```

;TEST DESTINATION MODE 6
002236 010701          SCOPE
002240 012705 002322  MOV      #MODE3,%5
002244 006765 177776  SXT     -2(5)
002250 005767 000044  TST     MODE1
002254 001401          BEQ      .+4
002256 000000          HLT             ;ERROR! INCORRECT RESULT

```

```

;TEST DESTINATION MODE 7
002260 010701          SCOPE
002262 012767 177777 000030  MOV     #-1,MODE1
002270 012704 177770  MOV     #-10,%4
002274 012767 002320 000020  MOV     #MODE1,MODE3
002302 006774 002332  SXT     #MODE3+10(4)
002306 005767 000006  TST     MODE1
002312 001401          BEQ      .+4
002314 000000          HLT             ;ERROR INCORRECT RESULT
002316 000402          BR       .+6             ;GO TO NEXT TEST
002320 000000  MODE1: 0
002322 000000  MODE3: 0

```

```

;TEST CONDITION CODES WHEN THE DESTINATION MODE IS NOT 0
;N=0,Z=1,V=0,C=0
002324 010701          SCOPE
002326 012767 177777 177764  MOV     #-1,MODE1
002334 000257          CCC
002336 006767 177756  SXT     MODE1
002342 016700 175430  MOV     PSW,%0
002346 022700 000004  CMP     #4,%0
002352 001401          BEQ     .+4
002354 000000          HLT             ;ERROR! INCORRECT CONDITION CODES

```

```

;N=0,Z=1,V=0,C=1
002356 010701          SCOPE
002360 012767 177777 177732  MOV     #-1,MODE1
002366 012702 002320  MOV     #MODE1,%2
002372 012767 000003 175376  MOV     #3,PSW
002400 006712          SXT     (2)
002402 016700 175370  MOV     PSW,%0
002406 022700 000005  CMP     #5,%0
002412 001401          BEQ     .+4
002414 000000          HLT             ;ERROR! INCORRECT CONDITION CODES

```

```

;N=1,Z=0,V=0,C=0
002416 010701          SCOPE
002420 012702 002320  MOV     #MODE1,%2
002424 005067 177670  CLR     MODE1
002430 012767 000016 175340  MOV     #16,PSW
002436 006722          SXT     (2)
002440 016700 175332  MOV     PSW,%0
002444 022700 000010  CMP     #10,%0

```

```

002450 001401      BEQ      .+4
002452 000000      HLT
                                ;ERROR! INCORRECT CONDITION CODES

                                ;N=1,Z=0,V=0,C=1
002454 010701      SCOPE
002456 012702 002322  MOV      #MODE1+2,%2
002462 005067 177E32  CLR      MODE1
002466 012767 000017 175302  MOV      #17,PSW
002474 006742      SXT      -(2)
002476 016700 175274  MOV      PSW,%0
002502 022700 000011  CMP      #11,%0
002506 001401      BEQ      .+4
002510 000000      HLT
                                ;ERROR! INCORRECT CONDITION CODES

                                ;TEST THAT SXT EXTENDS 0'S INTO THE PSW
002512 010701      SCOPE
002514 012767 000357 175254  MOV      #357,PSW
002522 000250      CLN
002524 006767 175246  SXT      PSW
002530 016700 175242  MOV      PSW,%0
                                ;GET & TEST PSW CONTENTS
002534 001401      BEQ      .+4
002536 000000      HLT

002540 005267 176234      INC      ICNT
                                ;INCREMENT PASS COUNT
002544 026727 176230 177777  CMP      ICNT,#-1
                                ;GO TO DONE IF 100. PASSES COMPLETED
002552 001402      BEQ      DONE
002554 000167 176226 175000  JMP      BEGIN
002560 012767 000007 175000  MOV      #7,TPBUF
                                ;RING BELL
002566 105767 174772  TSTB    TPCSR
                                ;WAIT FOR THE
002572 100375      BPL      -4
                                ;BELL TO RING
002574 013702 000042  MOV      #42,%2
                                ;GET DECTAPE MONITOR ADDRESS
002600 001405      BEQ      DONE1
                                ;DO NOT RETURN IF (42)=0
002602 000005      RESET
002604 004712  LOGICAL: JSR     7,(2)
                                ;START OF
002606 000240      NOP
                                ;ACT11
002610 000240      NOP
                                ;OVERLAY
002612 000240      NOP
                                ;AREA
002614 000167 176162  DONE1: JMP      START
                                ;AND REPEAT TEST

                                .END
000001

```


H02

MAINDEC-11-DCKBA-B PDP11/45 SXT INST TEST MACY!! 27(732) 16-SEP-76 15:47 PAGE 22
DCKBAB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DCKBAB/SOL/CRF/PAGNUM=DCKBAB
RUN-TIME: 13.5 SECONDS
RUN-TIME RATIO: 58/6=8.6
CORE USED: 6K (11 PAGES)

