

PDP11/45

REGISTERS

MD-11-DCKBH-A

EP-DCKBH-A-DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.

This microfiche strip contains 16 frames of technical information. The frames are arranged in two columns of eight. The left column contains diagrams and tables, while the right column contains more detailed data tables. The content is too small to read clearly but appears to be related to the registers mentioned in the header.

90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

1.0 ABSTRACT

THIS IS A TEST OF ALL THE 11/45 HARDWARE REGISTERS (R10-R15, SUPERVISOR STACK POINTER(R16), USER STACK POINTER(R17), AND THE MICRO BREAK REGISTER. THIS TEST INSURES THAT ALL BITS IN EACH OF THE REGISTERS CAN BE SET AND CLEARED PROPERLY.

2.0 REQUIREMENTS

2.1 EQUIPMENT

BASIC 11/45 SYSTEM

2.2 STORAGE

THIS PROGRAM USES 0 THRU 17500

2.3 PRELIMINARY PROGRAMS

DDAA THRU DDMA

3.0 LOADING PROCEDURE

LOAD PROGRAM USING ABS LOADER

4.0 STARTING PROCEDURE

LOAD ADDRESS 200. PRESS START. THE PROGRAM WILL LOOP AND RING BELL ON PASS COMPLETION.

5.0 OPERATING PROCEDURE

5.1 SWITCH SETTINGS

NONE

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

SCOPE IS A MOVE PC,R1 AND STORES THE PC+2 IN R1.

5.2.2 HLT

HLT IS A HALT INSTRUCTION.

6.0 ERRORS

ALL ERRORS WILL CAUSE A HALT
TRAP AND INTERRUPT ERRORS WILL CAUSE A HALT AT VECTOR+2.

6.1 ERROR RECOVERY

PRESS CONTINUE TO PROCEED TO NEXT TEST

6.2 ERROR LOOPING

TO LOOP ON AN ERROR, PLACE A BRANCH TO THE PREVIOUS SCOPE INSTRUCTION IN PLACE OF THE HALT INSTRUCTION.
NOTE THAT IF THE ERROR IS INTERMITTANT THAT THE TEST WILL DROP THRU THE HALT AND PROCEED TO THE NEXT TEST.
THEREFORE, TO LOOP THE TEST CONTINUOUSLY REPLACE THE BEQ .+4 INSTRUCTION IMMEDIATELY PRECEDING THE HALT WITH A BRANCH BACK TO THE PREVIOUS SCOPE.

TO LOOP ON TRAP FAILURES, PATCH IN THE FOLLOWING ROUTINE AT THE ADDRESS OF THE TRAP VECTOR.

010000
020000
040000
100000

BIT12=10000
BIT13=20000
BIT14=40000
BIT15=100000

: - 14 -
: - 15 -
: - 16 -
: - 17 -

000000
000001
000002
000003
000004
000005
000006
000007
000006
000006
000006
000007

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
R6=%6
R7=%7
SP=%6
KSP=%6
SSP=%6
USP=%6
PC=%7

:
: KERNEL'S STACK POINTER
: SUPERVISOR'S STACK POINTER
: USER'S STACK POINTER

000000
000001
000002
000003
000004
000005
000006
000007

R10=%0
R11=%1
R12=%2
R13=%3
R14=%4
R15=%5
R16=%6
R17=%7

000500
000600
000700

KPTR=500
SPTR=600
UPTR=700

: KERNEL'S INITIAL STACK POINTER
: SUPERVISOR'S INITIAL STACK POINTER
: USER'S INITIAL STACK POINTER

177570
177776
177770
177564
177566
177570
022626
010701
000000

; REGISTER ADDRESSES
DISPLAY=177570
PSW=177776
UE EAK=177770
TPCSR=177564
TPBUF=177566
SWR=177570
POP2=22626
SCOPE=010701
HLT=0

: ADDRESS OF CONSOLE DISPLAY REGISTER
: ADDRESS OF PROCESSOR STATUS WORD
: ADDRESS OF MICRO BREAK REGISTER

: ADDRESS OF CONSOLE SWITCH REGISTER
: CMP (6)+ (6)+ POPS 2 WORDS OFF STACK
: MOV PC,R1 (R11)

000004
000014
000030
000034

; VECTOR ADDRESSES
ERRVEC=4
TBITVEC=14
EMTVEC=30
TRAPVEC=34

: ADDRESS OF ERROR TRAP VECTOR
: ADDRESS OF 'T' BIT TRAP VECTOR
: ADDRESS OF EMT TRAP VECTOR
: ADDRESS OF TRAP TRAP VECTOR

000000
040000
140000

; BIT ASSIGNMENTS IN PSW
KM=0
SM=40000
UM=140000

: KERNEL MODE
: SUPERVISORY MODE
: USER MODE

000000
000002
000002

.=0
.+2
HALT


```

000344 000346
000346 000350
000350 000352
000352 000000
000354 000356
000356 000000
000360 000362
000362 000000
000364 000366
000366 000000
000370 000372
000372 000000
000374 000376
000376 000000

```

```

.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT

```

```

000200 000200 000604
000167

```

```

.=200
JMP START

```

```

001000 001000
001002 000000
001010 001010

```

```

.=1000
ICNT:0 ;PASS COUNT
TEMP:0

```

```

001010 005067 177764
001014 016737 177760 177570
001022 012706 000500
001026 032737 000400 177570
001034 001403
001036 113737 177570 177770
001044 005067 176726

```

```

.=. +4
START: CLR ICNT
BEGIN: MOV ICNT, #0 DISPLAY ;DISPLAY PASS COUNT
MOV #KPTR, KSP ;INITIALIZE THE STACK POINTER
BIT #400, #SMR ;LOAD MICRO BREAK REGISTER?
BEQ .+10
MOVB #SMR, #SUBBREAK ;LOAD MICRO BREAK REG WITH SRO-7
CLR PSM
;LOAD EACH REGISTER WITH ITS IDENTIFIER

```

```

001050 012700 000001
001054 012701 000002
001060 012702 000004
001064 012703 000010
001070 012704 000020
001074 012705 000040
001100 012767 004000 176670
001106 012700 000400
001112 012701 001000
001116 012702 002000
001122 012703 004000
001126 012704 010000
001132 012705 020000

```

```

TO: MOV #BIT0, R0
MOV #BIT1, R1
MOV #BIT2, R2
MOV #BIT3, R3
MOV #BIT4, R4
MOV #BIT5, R5
MOV #BIT11, PSM ;SET REGISTER SET BIT
MOV #BIT8, R10
MOV #BIT9, R11
MOV #BIT10, R12
MOV #BIT11, R13
MOV #BIT12, R14
MOV #BIT13, R15

```

```

001136 005067 176634

```

```

CLR PSM ;SWITCH TO CONVENTIONAL REGISTERS

```

```

;TEST THAT ALL REGISTERS WERE PROPERLY LOADED
001142 022700 000001      CMP      #BIT0,R0
001146 001401      BEQ      .+4
001150 000000      HLT
001152 022701 000002      CMP      #BIT1,R1
001156 001401      BEQ      .+4
001160 000000      HLT
001162 022702 000004      CMP      #BIT2,R2
001166 001401      BEQ      .+4
001170 000000      HLT
001172 022703 000010      CMP      #BIT3,R3
001176 001401      BEQ      .+4
001200 000000      HLT
001202 022704 000020      CMP      #BIT4,R4
001206 001401      BEQ      .+4
001210 000000      HLT
001212 022705 000040      CMP      #BIT5,R5
001216 001401      BEQ      .+4
001220 000000      HLT
001222 022706 000500      CMP      #KPTR,KSP
001226 001401      BEQ      .+4
001230 000000      HLT

;SWTICH TO UPPER REGISTERS
001232 012767 004000 176536      MOV      #BIT11,PSW
001240 022700 000400      CMP      #BIT8,R10
001244 001401      BEQ      .+4
001246 000000      HLT
001250 022701 001000      CMP      #BIT9,R11
001254 001401      BEQ      .+4
001256 000000      HLT
001260 022702 002000      CMP      #BIT10,R12
001264 001401      BEQ      .+4
001266 000000      HLT
001270 022703 004000      CMP      #BIT11,R13
001274 001401      BEQ      .+4
001276 000000      HLT
001300 022704 010000      CMP      #BIT12,R14
001304 001401      BEQ      .+4
001306 000000      HLT
001310 022705 020000      CMP      #BIT13,R15
001314 001401      BEQ      .+4
001316 000000      HLT

;R7 CHECK
001320 012707 001330      MOV      #TOX,R17
001324 000000      HLT
001326 000000      HLT

```

001330	005067	176442		TOX:	CLR	PSW
001334	010701				SCOPE	
001336	012767	004000	176432	:CHECK THAT ALL BITS IN R10 CAN BE SET/CLEARED.		
001344	012767	000020	177432	A0:	MOV	#BIT11,PSW
001352	012700	000001			MOV	#20,TEMP+2
001356	010067	177420			MOV	#1,R10
001362	006367	177414		AOA:	MOV	R10,TEMP
001366	006300				ASL	TEMP
001370	020067	177406			ASL	R10
001374	001401				CMP	R10,TEMP
001376	000000				BEQ	.+4
001400	005367	177400			HLT	
001404	001366				DEC	TEMP+2
001406	010701				BNE	AOA
					SCOPE	
001410	012767	004000	176350	:CHECK THAT ALL BITS IN R11 CAN BE SET/CLEARED.		
001416	012767	000020	177360	A1:	MOV	#BIT11,PSW
001424	012701	000001			MOV	#20,TEMP+2
001430	010167	177346			MOV	#1,R11
001434	006367	177342		A1A:	MOV	R11,TEMP
001440	006301				ASL	TEMP
001442	020167	177334			ASL	R11
001446	001401				CMP	R11,TEMP
001450	000000				BEQ	.+4
001452	005367	177326			HLT	
001456	001366				DEC	TEMP+2
001460	010701				BNE	A1A
					SCOPE	
001462	012767	004000	176306	:CHECK THAT ALL BITS IN R12 CAN BE SET/CLEARED.		
001470	012767	000020	177306	A2:	MOV	#BIT11,PSW
001476	012702	000001			MOV	#20,TEMP+2
001502	010267	177274			MOV	#1,R12
001506	006367	177270		A2A:	MOV	R12,TEMP
001512	006302				ASL	TEMP
001514	020267	177262			ASL	R12
001520	001401				CMP	R12,TEMP
001522	000000				BEQ	.+4
001524	005367	177254			HLT	
001530	001366				DEC	TEMP+2
001532	010701				BNE	A2A
					SCOPE	
001534	012767	004000	176234	:CHECK THAT ALL BITS IN R13 CAN BE SET/CLEARED.		
001542	012767	000020	177234	A3:	MOV	#BIT11,PSW
001550	012703	000001			MOV	#20,TEMP+2
001554	010367	177222			MOV	#1,R13
001560	006367	177216		A3A:	MOV	R13,TEMP
001564	006303				ASL	TEMP
001566	020367	177210			ASL	R13
001572	001401				CMP	R13,TEMP
001574	000000				BEQ	.+4
					HLT	

K01

001576 005367 177202
001602 001366
001604 010701

DEC TEMP+2
BNE A3A
SCOPE

001606 012767 004000 176162
001614 012767 000020 177162
001622 012704 000001
001626 010467 177150
001632 006367 177144
001636 006304
001640 020467 177136
001644 001401
001646 000000
001650 005367 177130
001654 001366
001656 010701

:CHECK THAT ALL BITS IN R14 CAN BE SET/CLEARED.
A4: MOV #BIT11,PSW
MOV #20,TEMP+2
MOV #1,R14
MOV R14,TEMP
A4A: ASL TEMP
ASL R14
CMP R14,TEMP
BEQ .+4
HLT
DEC TEMP+2
BNE A4A
SCOPE

001660 012767 004000 176110
001666 012767 000020 177110
001674 012705 000001
001700 010567 177076
001704 005367 177072
001710 006305
001712 020567 177064
001716 001401
001720 000000
001722 005367 177056
001726 001366
001730 010701

:CHECK THAT ALL BITS IN R15 CAN BE SET/CLEARED.
A5: MOV #BIT11,PSW
MOV #20,TEMP+2
MOV #1,R15
MOV R15,TEMP
ASA: ASL TEMP
ASL R15
CMP R15,TEMP
BEQ .+4
HLT
DEC TEMP+2
BNE ASA
SCOPE

001732 012767 004000 176036
001740 012700 000400
001744 012701 001000
001750 012702 002000
001754 012703 004000
001760 012704 010000
001764 012705 020000
001770 005067 176002
001774 010701

MOV #BIT11,PSW
MOV #BIT8,R10
MOV #BIT9,R11
MOV #BIT10,R12
MOV #BIT11,R13
MOV #BIT12,R14
MOV #BIT13,R15
CLR PSW
SCOPE

001776 012767 004000 175772
002004 010067 175766
002010 016767 175762 176764
002016 032767 000001 176756
002024 001401
002026 000000
002030 005767 176746
002034 001401
002036 000000
002040 010701

T1: MOV #BIT11,PSW
T1A: MOV R10,PSW
MOV PSW,TEMP
BIT #BIT0,TEMP
BEQ .+4
HLT
TST TEMP
BEQ .+4
HLT
SCOPE
:GO UPPER SET
:MOVE BIT8 TO PSW (SWITCHES
:TO 11/20 REGS.)
:WAS CORRECT %0 LOADED IN PSW
:IF BIT 0 WAS SET 11/20 REG 0
:WAS INCORRECTLY SELECTED
:IF TRULY R10 PSW WILL BE
:CLEAR
:ERROR! DID INST AT T1A CLR PSW?

002042 012767 004000 175726
002050 010067 175722

T2: MOV #BIT11,PSW
MOV R10,PSW ;SEL 11/20 REGS.

```

002054 010067 175716      MOV      R0,PSW      ;MOVE BIT0 TO PSW (SETS C)
002060 103401      BCS      .+4        ;IS CARRY SET?
002062 000000      HLT
002064 010701      SCOPE      ;RO NOT SELECTED

002066 012767 004000 175702 T3:  MOV      #BIT11,PSW
002074 040367 175676      BIC      R13,PSW    ;CLEAR BIT11 IN PSW
002100 010267 175672      MOV      R2,PSW    ;MOVE BIT1 TO PSW
002104 016767 175666 176670  MOV      PSW,TEMP
002112 022767 000004 176662  CMP      #BIT2,TEMP
002120 001401      BEQ      .+4
002122 000000      HLT
002124 010701      SCOPE

002126 012767 004000 175642 T4:  MOV      #BIT11,PSW
002134 160367 175636      SUB      R13,PSW
002140 074267 175632      XOR      R2,PSW
002144 016767 175626 176630  MOV      PSW,TEMP
002152 022767 000004 176622  CMP      #BIT2,TEMP
002160 001401      BEQ      .+4
002162 000000      HLT
002164 010701      SCOPE

002166 012767 004000 175602 T5:  MOV      #BIT11,PSW
002174 074367 175576      XOR      R13,PSW
002200 074367 175572      XOR      R3,PSW
002204 016767 175566 176570  MOV      PSW,TEMP
002212 022767 000010 176562  CMP      #BIT3,TEMP
002220 001401      BEQ      .+4
002222 000000      HLT
002224 010701      SCOPE

002226 012767 004000 175542 T6:  MOV      #BIT11,PSW
002234 012767 002276 175552  MOV      #T6A,TBITVEC
002242 005067 175550      CLR      TBITVEC+2
002246 052767 000020 175522  BIS      #BIT4,PSW    ;ATTEMPT TO SET 'T' BIT
002254 016767 175516 176520  MOV      PSW,TEMP
002262 022767 004004 176512  CMP      #BIT11+BIT2,TEMP ;'T' BIT SHOULD NOT HAVE SET
002270 001404      BEQ      T6X        ;'Z' WAS SET BY CLR TBITVEC+2 INST.
002272 000000      HLT                ;ERROR! ONLY BIT11 & Z SHOULD BE SET
002274 000402      BR      T6X
002276 000000      T7A:  HLT
002300 022626      POP2
002302 010701      T6X:  SCOPE

002304 012767 002344 175502 T7:  MOV      #T7A,TBITVEC
002312 012767 004000 175456  MOV      #BIT11,PSW
002320 012705 004020      MOV      #BIT11+BIT4,R15
002324 074567 175446      XOR      R15,PSW
002330 016767 175442 176444  MOV      PSW,TEMP
002336 001404      BEQ      T7X
002340 000000      HLT
002342 000402      BR      T7X
002344 000000      T7A:  HLT
002346 022626      POP2
002350 012767 000016 175436 T7X:  MOV      #16,14
002356 010701      SCOPE

```

```

002360 012767 004000 175410 T10:  MOV  #BIT11,PSW
002366 012705 004000          MOV  #BIT11,R15
002372 074567 175400          XOR  R15,PSW
002376 074567 175374          XOR  R5,PSW
002402 016767 175370 176372  MOV  PSW,TEMP
002410 022767 000040 176364  CMP  #BITS,TEMP
002416 001401          BEQ  .+4
002420 000000          HLT
002422 010701          SCOPE

002424 012767 004000 175344 T11:  MOV  #BIT11,PSW
002432 005000          CLR  R10
002434 010067 176342  MOV  R10,TEMP
002440 001401          BEQ  .+4
002442 000000          HLT
002444 010701          SCOPE

002446 012767 004000 175322 T12:  MOV  #BIT11,PSW
002454 005000          CLR  R10
002456 012767 000200 176316  MOV  #BIT7,TEMP
002464 116700 176312  MOVB TEMP,R10 ;MOVB TO A REG. EXTENDS SIGN
002470 020027 177600  CMP  R10,#177600 ;DID SIGN EXTEND
002474 001401          BEQ  .+4
002476 000000          HLT
002500 012700 000400  MOV  #BIT8,R10
002504 010701          SCOPE

002506 012767 004000 175262 T13:  MOV  #BIT11,PSW
002514 012701 177777          MOV  #-1,R11
002520 010167 176256          MOV  R11,TEMP
002524 026727 176252 177777  CMP  TEMP,#-1
002532 001401          BEQ  .+4
002534 000000          HLT
002536 010701          SCOPE

002540 012767 004000 175230 T14:  :TEST THAT MOV B TO A REGISTER EXTENDS THE SIGN (SIGN = 0)
002546 012701 177777          MOV  #BIT11,PSW
002552 012767 017777 176222  MOV  #-1,R11
002560 116701 176217  MOVB #17777,TEMP
002564 022701 000037  MOVB TEMP+1,R11
002570 001401          CMP  #37,R11
002572 000000          BEQ  .+4
002574 012701 001000          HLT
002600 010701          MOV  #BIT9,R11
SCOPE

002602 012767 004000 175166 T15:  :TEST THAT XOR %R,%R OPERATES PROPERLY
002610 074203          MOV  #BIT11,PSW
002612 010367 176164          XOR  R12,R13
002616 026727 176160 006000  MOV  R13,TEMP
002624 001401          CMP  TEMP,#BIT10+BIT11
002626 000000          BEQ  .+4
002630 012703 004000          HLT
002634 010701          MOV  #BIT11,R13
SCOPE

```

```

002636 012767 004000 175132 ;TEST SOB WITH UPPER REGISTER SET
002644 012704 000001 †16:  MOV  #BIT11,PSW
002650 000402          BR    R14
002652 000000          T16A: HLT  T16B
002654 000401          BR    T16X
002656 077403          T16B: SOB  R14,T16A
002660 012704 010000          T16X: MOV  #BIT12,R14
002664 010701          SCOPE

002666 060302          ;TEST ADD %R,%R USING UPPER REGISTER SET
002670 010267 176106 †17:  ADD  R13,R12
002674 022767 006000 176100          MOV  R12,TEMP
002702 001401          CMP  #BIT10+BIT11,TEMP
002704 000000          BEQ  .+4
002706 012702 002000          HLT
002712 010701          MOV  #BIT10,R12
          SCOPE

002714 005067 176064          ;TEST UPPER REGISTER SET (R13) IN AUTO INCREMENT MODE.
002720 012767 177777 176054 †20:  CLR  TEMP+2          ;PRE SET MEMORY ADDRESS
002726 012767 004000 175042          MOV  #-1,TEMP
002734 012703 001002          MOV  #BIT11,PSW          ;SWITCH TO UPPER REG. SET
002740 012367 176040          MOV  #TEMP,R13          ;LOAD REGISTER
002744 022767 177777 176032          MOV  (R13)+,TEMP+2      ;MOVE TEMP TO TEMP+2
002752 001401          CMP  #-1,TEMP+2          ;WAS TEMP MOVED
002754 000000          BEQ  .+4
002756 022703 001004          HLT          ;ERROR!
002762 001401          CMP  #TEMP+2,R13          ;DID R13 INCREMENT
002764 000000          BEQ  .+4
002766 010701          HLT          ;ERROR! R13 DID NOT AUTO-INCREMENT
          SCOPE

002770 012767 001002 176006          ;TEST UPPER REG. SET (R14) IN AUTO-DECREMENT MODE
002776 005067 176000 †20A: MOV  #TEMP,TEMP+2
003002 012767 004000 174766          CLR  TEMP
003010 012704 001006          MOV  #BIT11,PSW
003014 014467 175762          MOV  #TEMP+4,R14
003020 022767 001002 175754          MOV  -(R14),TEMP
003026 001401          CMP  #TEMP,TEMP
003030 000000          BEQ  .+4
003032 022704 001004          HLT
003036 001401          CMP  #TEMP+2,R14          ;DID REGISTER AUTO-DECREMENT
003040 000000          BEQ  .+4
003042 010701          HLT
          SCOPE

;TESTS 21 AND 22 HAVE BEEN DELETED.

003044 000240          ;TEST UPPER REGISTER SET REGISTERS AS INDEX REGISTERS
†23:  NOP          ;THIS LOCATION MAY BE USED TO CLEAR
          ;REGISTER SET BIT.

003046 012767 001002 175730          MOV  #TEMP,TEMP+2
003054 012767 177777 175720          MOV  #-1,TEMP
003062 012702 177776          MOV  #-2,R12          ;LOAD INDEX

```

```

003066 010205      MOV      R12,R15      ;REGISTERS
003070 067275 001006 001006  ADD      @TEMP+4(2),@TEMP+4(5) ;ADD TEMP [(TEMP+4(5))] TO ITSELF
003076 103401      BCS     .+4          ;-1,+1 RESULTS IN CARRY
003100 000000      HLT
003102 022767 177776 175672  CMP      @177776,TEMP ;RESULT CORRECT
003110 001401      BEQ     .+4
003112 000000      HLT
003114 010701      SCOPE
    
```

```

;TEST THAT ALL THREE STACK POINTERS (KSP,SSP AND USP CAN BE SELECTED.
T24:
003116 012767 000000 174652  MOV      @KM,PSW      ;KERNEL MODE!!!
003124 012706 000500      MOV      @KPTR,KSP    ;LOAD KERNEL STACK POINTER
003130 052767 040000 174640  BIS      @SM,PSW      ;SUPERVISORY MODE!!!
003136 012706 000600      MOV      @SPTR,SSP    ;LOAD SUPERVISOR STACK POINTER
003142 052767 140000 174626  SIS      @UM,PSW      ;USER MODE!!!
003150 012706 000700      MOV      @UPTR,USP    ;LOAD USER STACK POINTER
003154 010667 175622      MOV      USP,TEMP     ;GET USER STACK POINTER
003160 042767 100000 174610  BIC      @BIT15,PSW   ;SUPERVISORY MODE!!!
003166 010667 175612      MOV      SSP,TEMP+2   ;GET SUPERVISOR STACK POINTER
003172 005067 174600      CLR     PSW          ;KERNEL MODE
003176 022706 000500      CMP      @KPTR,KSP    ;CHECK KERNEL STACK POINTER
003202 001401      BEQ     .+4
003204 000000      HLT                ;ERROR! KERNEL STACK POINTER NOT LOADED

003206 022767 000600 175570  CMP      @SPTR,TEMP+2 ;CHECK SUPERVISOR STACK POINTER
003214 001401      BEQ     .+4
003216 000000      HLT                ;ERROR! SUPERVISOR STACK NOT LOADED

003220 022767 000700 175554  CMP      @UPTR,TEMP   ;CHECK USER STACK POINTER
003226 001401      BEQ     .+4
003230 000000      HLT                ;ERROR! USER STACK POINTER NOT LOADED
003232 010701      SCOPE
    
```

```

;TEST THAT JSR INST. OPERATES PROPERLY WITH REG. SET BIT SET
T25:
003234 012706 000500      MOV      @KPTR,KSP    ;INITIALIZE THE STACK POINTER
003240 012700 000001      MOV      @BIT0,R0     ;PRE SET R0
003244 012767 004000 174524  MOV      @BIT11,PSW   ;SET REG. SET BIT
003252 012700 000400      MOV      @BIT8,R10    ;PRE SET R10
003256 004067 000002      JSR     R10,T25B      ;GO TO T25B & SAVE R10 ON THE STACK
003262 000000      HLT                ;JSR DID NOT GO
003264 022706 000476  T25A:      CMP      @KPTR-2,KSP  ;STACK POINTER DID NOT DECREMENT
003270 001401      BEQ     .+4
003272 000000      HLT
003274 022767 000400 175174  CMP      @BIT8,KPTR-2 ;WAS OLD CONTENTS OF R10 SAVED?
003302 001401      BEQ     .+4
003304 000000      HLT                ;ERROR! R10 NOT SAVED ON THE STACK
003306 022700 003262  T25B:      CMP      @T25A,R10   ;IS RETURN ADDRESS IN R10
003312 001401      BEQ     .+4
003314 000000      HLT                ;ERROR! RETURN ADRS. NOT SAVED IN R10
003316 005067 174454      CLR     PSW          ;RO LEFT UNCHANGED?
003322 022700 000001      CMP      @BIT0,R0
003326 001401      BEQ     .+4
003330 000000      HLT                ;RO GOT CHANGED
003332 010701      SCOPE
    
```

;TEST JSR INSTRUCTION USING UPPER REG SET IN DEST. CALCULATION

```

003334 012706 000500          T26:  MOV    #KPTR,KSP
003340 012705 000040          MOV    #BITS,R5          ;PRE SET R5
003344 012767 004000 174424    MOV    #BIT11,PSW        ;SWITCH TO UPPER REG. SET
003352 005005          CLR    R15              ;PRE SET R15
003354 004565 003362          JSR    R15,T26B(5)      ;GO TO T26B
003360 000000          T26A: HLT
003362 005767 175110    T26B: TST    KPTR-2
003366 001401          BEQ    .+4              ;ERROR! JSR FAILED
003370 000000          HLT
003372 022705 003360          CMP    #T26A,R15        ;OLD CONTENTS OF R15 SAVED?
003376 001401          BEQ    .+4              ;ERROR! OLD CONTENTS OF R15 NOT SAVED
003400 000000          HLT
003402 005067 174370          CLR    PSW              ;RETURN ADDRESS IN R15?
003406 022705 000040          CMP    #BITS,R5
003412 001401          BEQ    .+4              ;ERROR! R15 DID NOT GET RETURN ADDRESS
003414 000000          HLT
003416 010701          SCOPE                  ;SWITCH TO LOWER REGISTERS
                                ;DID R5 CHANGE?
                                ;ERROR! R5 GOT CHANGED

                                ;TEST RTS WITH UPPER REGISTERS
003420 012706 000500          T27:  MOV    #KPTR,KSP
003424 012716 001000          MOV    #BIT9,(KSP)
003430 012702 000004          MOV    #BIT2,R2
003434 012767 004000 174334    MOV    #BIT11,PSW
003442 012702 003452          MOV    #T27A,R12
003446 000202          RTS    R12
003450 000000          HLT
003452 022706 000502    T27A: CMP    #KPTR+2,KSP
003456 001401          BEQ    .+4              ;PRE SET R2
003460 000000          HLT
003462 022702 001000          CMP    #BIT9,R12        ;SWITCH TO UPPER REG. SET
003466 001401          BEQ    .+4              ;LOAD REG. WITH RETURN ADDRESS
003470 000000          HLT
003472 005067 174300          CLR    PSW              ;RETURN TO T27A [(R12)]
003476 022702 000004          CMP    #BIT2,R2
003502 001401          BEQ    .+4              ;ERROR! RTS FAILED
003504 000000          HLT
003506 010701          SCOPE                  ;WAS STACK POINTER INCREMENTED
                                ;ERROR! STACK POINTER WAS NOT INCREMENTED
                                ;WAS R12 RESTORED?
                                ;ERROR! RTS DID NOT RESTORE R12
                                ;SWITCH TO LOWER REG. SET
                                ;DID R2 CHANGE?
                                ;ERROR! R2 CHANGED

                                ;CHECK THAT ALL BITS IN KSP CAN BE SET/CLEARED.
003510 012700 000001          MOV    #1,R0
003514 012767 000000 174254    T30:  MOV    #KH,PSW          ;GET '1' BIT
003522 010006          MOV    R0,KSP           ;LOAD KSP
003524 010602          MOV    KSP,R2          ;GET RESULT
003526 005067 174244          CLR    PSW              ;KERNEL MODE!!!

003532 020200          CMP    R2,R0            ;WAS KSP LOADED CORRECTLY?
003534 001401          BEQ    .+4
003536 000000          HLT
003540 006300          ASL    R0
003542 103364          BCC   T30
003544 010701          SCOPE
                                ;SHIFT '1' BIT THRU KSP
                                ;UNTIL ALL BITS ARE TESTED

                                ;CHECK THAT ALL BITS IN SSP CAN BE SET/CLEARED.
003546 012700 000001          MOV    #1,R0
003552 012767 040000 174216    T31:  MOV    #SM,PSW
003560 010006          MOV    R0,SSP          ;LOAD SSP

```

```

003562 010602      MOV      SSP,R2 ;GET RESULT
003564 005067 174206    CLR      PSH      ;KERNEL MODE!!!

003570 020200      CMP      R2,R0    ;WAS SSP LOADED CORRECTLY?
003572 001401      BEQ     .+4
003574 000000      HLT
003576 006300      RSL     R0        ;SHIFT '1' BIT THRU SSP
003600 103364      BCC     T31      ;UNTIL ALL BITS ARE TESTED
003602 010701      SCOPE

;CHECK THAT ALL BITS IN USP CAN BE SET/CLEARED.
003604 012700 000001      MOV      #1,R0    ;GET '1' BIT
003610 012767 140000 174160 T32:    MOV      #14,R0
003616 010006      MOV     R0,USP   ;LOAD USP
003620 010602      MOV     USP,R2   ;GET RESULT
003622 005067 174150    CLR      PSH      ;KERNEL MODE!!!

003626 020200      CMP      R2,R0    ;WAS USP LOADED CORRECTLY?
003630 001401      BEQ     .+4
003632 000000      HLT
003634 006300      RSL     R0        ;SHIFT '1' BIT THRU USP
003636 103364      BCC     T32      ;UNTIL ALL BITS ARE TESTED
003640 010701      SCOPE

```

```

:CHECK THAT ALL BITS IN THE MICRO BREAK REGISTER (177770) CAN BE SET
:AND CLEARED
003642 012700 000001
003646 012702 177770
003652 010012
003654 011203
003656 020003
003660 001401
003662 000000
003664 040012
003666 011203
003670 001401
003672 000000
003674 105300
003676 103365
003700 010701

T33:  MOV      #1,R0
      MOV      #UBREAK,R2
T33A: MOV      R0,(R2)
      MOV      (R2),R3
      CMP      R0,R3
      BEQ      .+4
      HLT
      :GET ADDRESS OF MICRO BREAK REGISTER
      :LOAD TEST BIT INTO REGISTER
      :GET RESULT
      :COMPARE RESULT & TEST BIT
      :ERROR! TEST BIT (R0) DID NOT SET INTO
      :REGISTER
      :CLEAR TEST BIT IN REGISTER
      :GET RESULT
      BIC      R0,(R2)
      MOV      (R2),R3
      BEQ      .+4
      HLT
      :ERROR! TEST BIT DID NOT GET CLEARED
      ASLB     R0
      BCC      T33A
      SCOPE
      :SHIFT TEST BIT UNTIL DONE

:CHECK THAT RESET DOES NOT CLEAR MICRO BREAK REGISTER
003702 012737 177777 177770
003710 122737 177777 177770
003716 001401
003720 000000
003722 000005
003724 122737 177777 177770
003732 001401
003734 000000
003736 010701

T34:  MOV      #-1,#UBREAK
      CMPB     #-1,#UBREAK
      BEQ      .+4
      HLT
      :ERROR!
      :RESET DOES NOT CLEAR REGISTER
      :CHECK THAT RESET DID NOT CLEAR ANY BITS
      RESET
      CMPB     #-1,#UBREAK
      BEQ      .+4
      HLT
      :ERROR!
      SCOPE

003740 005267 175034
003744 026727 175037 001000
003752 001402
003754 000167 175034
003760 012767 000007 173600
003766 105767 173572
003772 100375
003774 013702 000042
004000 001404
004002 004712
004004 000240
004006 000240
004010 000240
004012 000167 174772

END:  INC      ICNT
      CMP      ICNT,#1000
      BEQ      DONE
      JMP      BEGIN
DONE: MOV      #7,TPBUF
      TSTB    TPCSR
      BPL     .-4
      MOV     #42,%2
      BEQ     DONE1
      JSR     7,(2)
      NOP
      NOP
      NOP
      JMP     START

:INCREM ENT PASS COUNT
:1000 PASSES?
:RING THE BELL
:WAIT FOR THE BELL
:TO RING
:GET DECTAPE MONITOR RETURN ADDRESS
:DO NOT RETURN IF (42)=0
:RETURN TO DECTAPE MONITOR
:ACT11
:OVERLAY
:AREA
:RESTART TEST

000001 .END

```

RO	001336	415#																				
ROA	001362	419#	425																			
AI	001410	429#																				
A1A	001434	433#	439																			
A2	001462	443#																				
A2A	001506	447#	453																			
A3	001534	457#																				
A3A	001560	461#	467																			
A4	001606	471#																				
A4A	001632	475#	481																			
A5	001660	485#																				
A5A	001704	489#	495																			
BEGIN	001014	340#	859																			
BIT0	== 000001	134#	348	365	512	725	740															
BIT1	== 000002	135#	349	366																		
BIT10	== 002000	146#	357	391	503	629	648	651														
BIT11	== 004000	147#	354	388	502	397	415	429	443	457	471	485	499	503								
		509	520	526	526	544	553	558	567	568	579	580	589	596								
		606	615	626	626	632	636	648	657	671	726	748	768									
BIT12	== 010000	148#	359	400	501	642																
BIT13	== 020000	149#	360	403	505																	
BIT14	== 040000	150#																				
BIT15	== 100000	151#	707																			
BIT2	== 000004	136#	350	371	531	540	558	767	779													
BIT3	== 000010	137#	351	374	548																	
BIT4	== 000020	138#	352	377	556	568																
BIT5	== 000040	139#	353	380	584	747	759															
BIT6	== 000100	140#																				
BIT7	== 000200	141#	598																			
BIT8	== 000400	144#	355	388	500	603	727	733														
BIT9	== 001000	145#	356	391	501	622	766	775														
DISPLA=	177570	180#	340#																			
DONE	003760	858	860#																			
DONE1	004012	864	869#																			
ENTVEC=	000030	192#																				
END	003740	856#																				
ERRVEC=	000004	190#																				
HLT	== 000000	183#	367	370	373	376	379	382	385	390	393	396	399	402								
		405	408	409	423	437	451	465	479	493	514	517	524	533								
		542	550	560	562	572	574	576	593	602	611	621	631	639								
		650	662	665	676	679	693	696	712	716	720	729	732	735								
		738	742	751	754	757	761	771	774	777	781	793	807	821								
		835	840	849	853																	
ICNT	001000	336#	339#	340	856#	857																
KM	== 000000	195#	700	786																		
KPTR	== 000500	176#	341	383	701	710	724	730	733	746	752	765	772	788								
KSP	== 000006	162#	341#	383	701#	710	724*	730	746*	765*	766*	772	787*									
PC	== 000007	165#																				
POP2	== 022626	186#	563	575																		
PSM	== 177776	181#	345#	354#	362#	387#	411#	415#	429#	443#	457#	471#	485#	499#								
		506#	509#	510#	511	520#	521#	522#	527#	528#	529#	530	536#	537#								
		538#	539	544#	545#	546#	547	553#	556#	557	567#	569#	570	579#								
		581#	582#	583	589#	596#	606#	615#	626#	636#	657#	671#	700#	702#								
		704#	707#	709#	726#	739#	748#	758#	768#	778#	786#	789#	800#	803#								
		814#	817#																			
RO	== 000000	153#	348#	365	522	725#	740	785#	787	791	794#	799#	801	805								

ROTREG	199#	414	428	442	456	470	484
ROTSP	200#	784	798	812			

K02

MAINDEC-11-DCKBH-A ALL PDP11/45 REGISTERS MACY11 27(732) 07-SEP-76 09:51 PAGE 26
DCKBHA P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.REN 1
.REPT 202
.TITLE 113

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DCKBHA, DCKBHA. SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DCKBHA.P11
RUN-TIME: 2 4 1 SECONDS
RUN-TIME RATIO: 33/8=3.8
CORE USED: 6K (12 PAGES)

