

PDP11/45

TRAPS TEST
MD-11-DCKBM-E

EP-DCKBM-E-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames, each displaying a different set of data. The data is organized into columns and rows, with some frames showing numerical values and others showing text or graphical representations. The frames are arranged in a regular grid pattern, typical of microfiche storage.

MACINDEX-11-DCKBM-E PDF11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 1
DCKBME.DOC

ՀԱՅԿԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ՎԵՐԱԴԱՐՄԱՆ ԱԶԳԱՅԻՆ ԲԱՆԿԻ ՓԵՏԱԿԱՆ ԲՆԱԿԱՆ ԿՈՌՊՈՐԱՑԻՆ

PRODUCT CODE:	MAINDEC-11-DCKBM-E-D
PRODUCT NAME:	TRAPS TEST
DATE:	APRIL, 1976
MAINTAINER:	DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

- 1.0 ABSTRACT
THIS PROGRAM TESTS ALL TRAP INSTRUCTIONS AND ERROR TRAPS. (TIMEOUT, ODD ADDRESS AND OVERFLOW) INTERRUPT LOGIC IS ALSO TESTED USING TELETYPE.
- 2.0 REQUIREMENTS
- 2.1 EQUIPMENT
BASIC 11/45 SYSTEM
- 2.2 STORAGE
THIS PROGRAM USES 0 THRU 17500
- 2.3 PRELIMINARY PROGRAMS
DOAA THRU DOMA
- 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.
R1 CONTAINS THE PC OF THE LAST TEST SUCCESSFULLY COMPLETED.
- 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.

TRAPVEC: TRAPVEC+4

DO1

MAINDEC-11-DCKBM-E FDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 3
DCKBME.DOC

104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123

TRAPVEC+2: 0
TRAPVEC+4: 012716 ;MOVE SCOPE ADDRESS TO STACK
TRAPVEC+6: ADDRESS ;ADDRESS OF PREVIOUS SCOPE
TRAPVEC+10: 000006 ;RETURN TO TEST AT SCOPE

RESTORE ALL LOCATIONS BEFORE PROCEEDING TO NEXT TEST.

7.0 RESTRICTIONS
NONE

8.0 MISCELLANEOUS
ON TRAP ERRORS THE STACK POINTER(R6) WILL CONTAIN THE
ADDRESS WHERE THE TRAP OCCURED.

8.1 EXECUTION TIME
THIS PROGRAM TAKES ABOUT 1 MINUTE.

8.2 STACK POINTER
THIS PROGRAM INITIALLY SETS THE STACK POINTER AT 500.

!

E01

MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 5
DCKBME.P11

124
125

.TITLE MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST.
.LIST ME
.NLIST MC,SEQ
.ABS

:TEST DCKBME-PDP11/45 TRAPS TEST. ALL PROCESSOR TRAPS INCLUDING ERROR TRAPS ARE
:TESTED IN THIS PROGRAM (KERNEL MODE ONLY).
:NOTE: THE SEGMENTATION ABORT TRAPS ARE NOT TESTED.

:STARTING PROCEEDURE
:LOAD ADDRESS 0200
:PRESS START
:STACK POINTER IS AT 500
:BELL WILL RING WHEN TEST IS COMPLETE
:TEST WILL TYPE 'NULL' CHARACTERS IN THE FIRST 8. PASSES AND THEN
:OMIT TYPING 'NULLS' FOR THE REST OF THE TEST.

000500

STKPTR=0500 ;INITIAL STACK POINTER

:MACRO CALL
.MACR SETTBIT A
MOV #20,-(6)
MOV #.+6,-(6)
'A'
.ENDM

:PUSH 'T'BIT ON THE STACK
:PUSH PC ON THE STACK
:SET 'T' BIT

000000
000003
000002
010701
000000

TAB=%0
LAST=%3
FIRST=%2
SCOPE=010701
HLT=HALT

:MOVE PC TO R1

:TRAP VECTOR ADDRESSES

ERRVEC=4
RESVEC=10
BPTVEC=14
TBITVEC=14
IOTVEC=20
PFVEC=24
EMTVEC=30
TRAPVEC=34
TPVEC=64
PIRVEC=240
FPEVEC=244

:RESERVED INST AND ILLEGAL ADDRESSES
:RESERVED INSTRUCTION TRAP VECTOR
: 'BREAK-POINT' TRAP VECTOR
: 'T' BIT TRAP VECTOR ADDRESS
: IOT TRAP VECTOR ADDRESS
: POWER FAIL TRAP VECTOR ADDRESS
: EMT TRAP VECTOR ADDRESS
: TRAP TRAP VECTOR ADDRESS
: ADDRESS OF TELEPRINTER INT VECTOR
: PROG INT RQST TRAP VECTOR
: FLOATING POINT TRAP VECTOR

000004
000010
000014
000014
000020
000024
000030
000034
000064
000240
000244

:REGISTER ADDRESSES

TTCR=177564
TRCSR=177560
SWR=177570
DISPLAY=177570
PSW=177776
PIRQ=177772
UBREAK=177770
PIR4=10000
RESINST=76000
RES=75000
ILLA=004700

:ADDRESS OF DISPLAY REGISTER
:ADDRESS OF PROCESSOR STATUS WORD
:ADDRESS OF PROC. INT. RQST. REGISTER
:ADDRESS OF MICRO BREAK REGISTER
:LEVEL 4 REQUEST
:A RESERVED INSTRUCTION
:A RESERVED INSTRUCTION
:AN ILLEGAL INSTRUCTION (JSR 7,%0)

177564
177560
177570
177570
177776
177772
177770
010000
076000
075000
004700

	000100	ILLB=100	;AN ILLEGAL INSTRUCTION (JMP %0)
	000000	.+0	
000000	000002	.+2	
000002	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000004	000006	.+2	
000006	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000010	000012	.+2	
000012	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000014	000016	.+2	
000016	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000020	000022	.+2	
000022	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000024	000026	.+2	
000026	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000030	000032	.+2	
000032	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000034	000036	.+2	
000036	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000040	000042	.+2	
000042	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000044	000046	.+2	
000046	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000050	000052	.+2	
000052	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000054	000056	.+2	
000056	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000060	000062	.+2	
000062	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000064	000066	.+2	
000066	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000070	000072	.+2	
000072	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000074	000076	.+2	
000076	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000100	000102	.+2	
000102	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000104	000106	.+2	
000106	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000110	000112	.+2	
000112	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000114	000116	.+2	
000116	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000120	000122	.+2	
000122	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000124	000126	.+2	
000126	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000130	000132	.+2	
000132	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000134	000136	.+2	
000136	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000140	000142	.+2	
000142	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000144	000146	.+2	
000146	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000150	000152	.+2	
000152	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION

000154	000156	.+2	
000156	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000160	000162	.+2	
000162	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000164	000166	.+2	
000166	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000170	000172	.+2	
000172	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000174	000176	.+2	
000176	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000200	000202	.+2	
000202	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000204	000206	.+2	
000206	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000210	000212	.+2	
000212	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000214	000216	.+2	
000216	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000220	000222	.+2	
000222	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000224	000226	.+2	
000226	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000230	000232	.+2	
000232	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000234	000236	.+2	
000236	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000240	000242	.+2	
000242	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000244	000246	.+2	
000246	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000250	000252	.+2	
000252	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000254	000256	.+2	
000256	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000260	000262	.+2	
000262	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000264	000266	.+2	
000266	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000270	000272	.+2	
000272	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000274	000276	.+2	
000276	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000300	000302	.+2	
000302	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000304	000306	.+2	
000306	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000310	000312	.+2	
000312	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000314	000316	.+2	
000316	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000320	000322	.+2	
000322	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000324	000326	.+2	
000326	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000330	000332	.+2	
000332	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION

H01

MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 8
DCKBME.P11

000334	000336	.+2	
000336	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000340	000342	.+2	
000342	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000344	000346	.+2	
000346	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000350	000352	.+2	
000352	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000354	000356	.+2	
000356	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000360	000362	.+2	
000362	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000364	000366	.+2	
000366	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000370	000372	.+2	
000372	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000374	000376	.+2	
000376	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000400	000402	.+2	
000402	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000404	000406	.+2	
000406	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000410	000412	.+2	
000412	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000414	000416	.+2	
000416	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000420	000422	.+2	
000422	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000424	000426	.+2	
000426	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000430	000432	.+2	
000432	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000434	000436	.+2	
000436	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000440	000442	.+2	
000442	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000444	000446	.+2	
000446	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000450	000452	.+2	
000452	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000454	000456	.+2	
000456	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000460	000462	.+2	
000462	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000464	000466	.+2	
000466	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000470	000472	.+2	
000472	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000474	000476	.+2	
000476	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000500	000502	.+2	
000502	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000504	000506	.+2	
000506	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000510	000512	.+2	
000512	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION

000514	000516	.+2	
000516	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000520	000522	.+2	
000522	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000524	000526	.+2	
000526	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000530	000532	.+2	
000532	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000534	000536	.+2	
000536	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000540	000542	.+2	
000542	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000544	000546	.+2	
000546	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000550	000552	.+2	
000552	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000554	000556	.+2	
000556	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000560	000562	.+2	
000562	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000564	000566	.+2	
000566	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000570	000572	.+2	
000572	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000574	000576	.+2	
000576	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000600	000602	.+2	
000602	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000604	000606	.+2	
000606	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000610	000612	.+2	
000612	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000614	000616	.+2	
000616	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000620	000622	.+2	
000622	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000624	000626	.+2	
000626	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000630	000632	.+2	
000632	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000634	000636	.+2	
000636	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000640	000642	.+2	
000642	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000644	000646	.+2	
000646	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000650	000652	.+2	
000652	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000654	000656	.+2	
000656	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000660	000662	.+2	
000662	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000664	000666	.+2	
000666	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION
000670	000672	.+2	
000672	000000	HALT	;ERROR! TRAPPED TO PREVIOUS LOCATION

000674	000676	.+2		
000676	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000700	000702	.+2		
000702	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000704	000706	.+2		
000706	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000710	000712	.+2		
000712	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000714	000716	.+2		
000716	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000720	000722	.+2		
000722	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000724	000726	.+2		
000726	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000730	000732	.+2		
000732	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000734	000736	.+2		
000736	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000740	000742	.+2		
000742	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000744	000746	.+2		
000746	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000750	000752	.+2		
000752	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000754	000756	.+2		
000756	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000760	000762	.+2		
000762	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000764	000766	.+2		
000766	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000770	000772	.+2		
000772	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
000774	000776	.+2		
000776	000000	HALT		;ERROR! TRAPPED TO PREVIOUS LOCATION
		.NLIST MC		

000046	000046	.=46		
	016036	LOGICAL		
	000052	.=52		
000052	040000	40000		
	000200	.=0200		
000200	000167	JMP	START	
	000260	.=260		

001000	001000	ICNT:	0	.=1000				;CONTAINS THE PASS COUNT
	000000							
001002	005067	177772	START:	CLR	ICNT			
001006	012706	000500	BEGIN:	MOV	#STKPTR,%6			;SET STACK PTR
001012	016737	177762		MOV	ICNT,%DISP			;DISPLAY THE PASS COUNT
001020	032737	000400		BIT	#400,%SWR			;LOAD MICRO BREAK REGISTER?
001026	001403			BEG	+10			
001030	113737	177570		MOVB	%SWR,%UBREAK			;LOAD MICRO BREAK REG WITH SR0-7
001036	010701			SCOPE				;SCOPE STORES PC IN R1

;CHECK THAT A TRAP OCCURS ON A RESERVED INSTRUCTION (076000)

K01

MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 11
DCKBME.P11

001040	012737	001056	000010	MOV	#RETA, @#RESVEC	; RETURN LOCATION
001046	005037	000012		CLR	@#RESVEC+2	
001052	076000			RESINST		; RESERVED INSTRUCTION (SHOULT TRAP)
001054	000000			HLT		; RESERVED INSTRUCTION FAILED TO TRAP
001056	010701			RETA:	SCOPE	; SCOPE STORES PC IN R1
; TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION						
001060	012706	000500		MOV	#STKPTR, %6	; SET STACK PTR
001064	012737	001074	000010	MOV	#RETB, @#RESVEC	; RETURN POINTER
001072	076000			RESINST		; RESERVED INSTRUCTION
001074	020627	000474		RETB:	CMP %6, #STKPTR-4	; TEST DECREMENT OF %6
001100	001401			BEQ	.+4	
001102	000000			HLT		; NOT DECREMENTED TWO WORDS
001104	010701			SCOPE		; SCOPE STORES PC IN R1
; TEST THAT PROPER P.C. IS SAVED						
001106	012706	000500		MOV	#STKPTR, %6	; SET STACK PTR
001112	012737	001122	000010	MOV	#RETC, @#RESVEC	; RETURN FROM TRAP POINTER
001120	076000			RESINST		; TRAP ON THIS INSTRUCTION
001122	022737	001122	000474	RETC:	CMP #, @#STKPTR-4	; CHECK RETURN PC ON THE STACK
001130	001401			BEQ	.+4	
001132	000000			HLT		; INCORRECT P.C.
001134	010701			SCOPE		; SCOPE STORES PC IN R1
; TEST THAT "OLD" PSW AND PRIORITY ARE PLACED ON STACK						
001136	010701			SCOPE		; SCOPE STORES PC IN R1
001140	012706	000500		MOV	#STKPTR, %6	; SET STACK PTR
001144	012737	001160	000010	MOV	#RETD, @#RESVEC	; SET UP
001152	005037	177776		CLR	@#PSW	; CLEAR PSW AND PRIORITY
001156	076000			RESINST		; TRAP ON RESERVED INSTRUCTION
001160	016700	177312		RETD:	MOV STKPTR-2, %0	; GET SAVED STATUS & TEST FOR ALL 0'S
001164	001401			BEQ	.+4	; BRANCH IF 0
001166	000000			HLT		; ERROR! INCORRECT STATUS SAVED ON STACK
001170	010701			SCOPE		; SCOPE STORES PC IN R1
; TEST THAT "NEW" STATUS IS CORRECT						
001172	012706	000500		MOV	#STKPTR, %6	; SET STACK PTR
001176	012737	001214	000010	MOV	#RETE, @#RESVEC	; SET UP
001204	112737	000357	177776	MOVB	#357, @#PSW	; PRE SET THE STATUS WORD
001212	076000			RESINST		; RESERVED INSTRUCTION TRAP
001214	016700	177256		RETE:	MOV STKPTR-2, %0	; GET SAVED STATUS
001220	122700	000357		CMPB	#357, %0	; SAVED STATUS CORRECT?
001224	001401			BEQ	.+4	; BRANCH IF CORRECT
001226	000000			HLT		; ERROR! INCORRECT STATUS WAS SAVED
001230	010701			SCOPE		; SCOPE STORES PC IN R1
; TEST THAT "NEW" STATUS IS CORRECT						
001232	012706	000500		MOV	#STKPTR, %6	; INITIALIZE THE STACK POINTER
001236	012737	001260	000010	MOV	#RETF, @#RESVEC	; SET UP VECTOR
001244	005037	000012		CLR	@#RESVEC+2	; CLEAR 'NEW' STATUS WORD
001250	112737	000357	177776	MOVB	#357, @#PSW	; PRE SET THE STATUS WORD
001256	076000			RESINST		; DO A RESERVED INSTRUCTION
001260	013700	177776		RETF:	MOV @#PSW, %0	; GET & TEST THE 'NEW' STATUS WORD
001264	001401			BEQ	.+4	; BRANCH IF ALL 0'S
001266	000000			HLT		; ERROR! 'NEW' STATUS WAS INCORRECT
001270	005037	177776		CLR	@#PSW	
001274	010701			SCOPE		; SCOPE STORES PC IN R1

L01

MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 12
DCKBME.P11

001276	012706	000500		MOV	#STKPTR,%6	;SET UP THE STACK POINTER
001302	012737	001324	000010	MOV	#RETG,%#RESVEC	
001310	012737	004357	000012	MOV	#4357,%#RESVEC+2	;LOAD THE NEW STATUS WORD
001316	005037	177776		CLR	%PSW	;PRE SET THE STATUS WORD
001322	076000			RESINST		;DO A RESERVED INSTRUCTION
				*****NOTE: R0 IS UPPER R0 *****		
001324	013700	177776		RETG:	MOV	%PSW,%0
001330	022700	004357		CMP	#4357,%0	;GET THE 'NEW' STATUS WORD
001334	001401			BEQ	.+4	;WAS 'NEW' STATUS CORRECTLY LOADED
001336	000000			HLT		;BRANCH IF CORRECT
001340	010701			SCOPE		;ERROR! 'NEW' STATUS WAS INCORRECT
						;SCOPE STORES PC IN R1
001342	012737	000006	000010	MOV	#6,%#RESVEC	;RESTORE RESERVED INSTRUCTION
001350	005037	000012		CLR	%#RESVEC+2	;TO HALT AT 6


```

001354 010701          ;TEST THAT A TRAP OCCURES FOR A "TRAP" INSTRUCTION
001356 012706 000500          SCOPE          ;SCOPE STORES PC IN R1
001362 012737 001400 000034  MOV      #STKPTR,%6          ;STACK POINTER SETUP
001370 005037 000036          MOV      #RETA1,@#TRAPVEC      ;RETURN LOCATION
001374 104400          CLR      @#TRAPVEC+2
001376 000000          TRAP          ;DO A TRAP INSTRUCTION
001400 010701          HLT          ;SCOPE STORES PC IN R1
RETA1:  SCOPE

001402 012706 000500          ;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
001406 012737 001416 000034  MOV      #STKPTR,%6          ;STACK POINTER SETUP
001414 104400          MOV      #RETB1,@#TRAPVEC      ;RETURN POINTER
001416 020627 000474          TRAP          ;DO A TRAP INSTRUCTION
001422 001401          RETB1:  CMP      %6,#STKPTR-4          ;TEST DECREMENT OF %6
001424 000000          BEQ      .+4
001426 010701          HLT          ;NOT DECREMENTED TWO WORDS
SCOPE          ;SCOPE STORES PC IN R1

001430 012706 000500          ;TEST THAT PROPER P.C. IS SAVED
001434 012737 001444 000034  MOV      #STKPTR,%6          ;STACK POINTER SETUP
001442 104400          MOV      #RETC1,@#TRAPVEC      ;RETURN FROM TRAP POINTER
001444 022767 001444 177022  TRAP          ;TRAP ON THIS INSTRUCTION
001452 001401          RETC1:  CMP      #.,STKPTR-4          ;CHECK INCREMENTED P.C.
001454 000000          BEQ      .+4
001456 010701          HLT          ;INCORRECT P.C.
SCOPE          ;SCOPE STORES PC IN R1

001460 010701          ;TEST THAT "OLD" PSW & PRIORITY ARE PLACED ON STACK
001462 012706 000500          SCOPE          ;SCOPE STORES PC IN R1
001466 012737 001502 000034  MOV      #STKPTR,%6          ;SET UP
001474 005037 177776          MOV      #RETD1,@#TRAPVEC      ;SET UP
001500 104400          CLR      @#PSW          ;CLEAR PSW AND PRIORITY
001502 016700 176770          TRAP          ;DO A TRAP INSTRUCTION
001506 001401          RETD1:  MOV      STKPTR-2,%0          ;GET & TEST THE SAVED STATUS
001510 000000          BEQ      .+4          ;BRANCH IF ALL 0'S
001512 010701          HLT          ;ERROR! SAVED STATUS IS INCORRECT
SCOPE          ;SCOPE STORES PC IN R1

001514 012706 000500          ;INITIALIZE THE STACK POINTER
001520 012737 001536 000034  MOV      #STKPTR,%6          ;SET UP
001526 012737 000357 177776  MOV      #RETE1,@#TRAPVEC      ;PRE SET THE STATUS WORD
001534 104400          MOV      #357,@#PSW
001536 016700 176734          TRAP          ;DO A TRAP INSTRUCTION
001542 022700 000357          RETE1:  MOV      STKPTR-2,%0          ;GET THE SAVED STATUS WORD
001546 001401          CMP      #357,%0          ;WAS CORRECT STATUS SAVED ON THE STACK
001550 000000          BEQ      .+4          ;BRANCH IF CORRECT
001552 010701          HLT          ;ERROR! INCORRECT STATUS SAVED ON STACK
SCOPE          ;SCOPE STORES PC IN R1

001554 012706 000500          ;TEST THAT "NEW" STATUS IS CORRECT
001560 012737 001602 000034  MOV      #STKPTR,%6
001566 005037 000036          MOV      #RETF1,@#TRAPVEC
001572 012737 000357 177776  CLR      @#TRAPVEC+2
001600 104400          MOV      #357,@#PSW          ;PRE SET THE STATUS WORD
001602 013700 177776          TRAP          ;GET THE 'NEW' STATUS WORD
001606 001401          RETF1:  MOV      @#PSW,%0          ;BRANCH IF ALL 0'S
          BEQ      .+4

```


Address	Hex	Dec	Label	Instruction	Comment
001610	000000			HLT	;ERROR! INCORRECT 'NEW' STATUS LOADED
001612	005037	177776		CLR	@#PSW
001616	010701			SCOPE	;SCOPE STORES PC IN R1
001620	012706	000500		MOV	#STKPTR,%6
001624	012737	001646	000034	MOV	#RETG1,@#TRAPVEC
001632	012737	004357	000036	MOV	#4357,@#TRAPVEC+2 ;LOAD 'NEW' STATUS WORD
001640	005037	177776		CLR	@#PSW ;PRE SET THE STATUS WORD
001644	104400			TRAP	;DO A TRAP INSTRUCTION
001646	013700	177776	RETG1:	MOV	@#PSW,%0 ;GET THE 'NEW' STATUS
001652	022700	004357		CMP	#4357,%0 ;IS IT CORRECT?
001656	001401			BEQ	+.4 ;BRANCH IF CORRECT
001660	000000			HLT	;ERROR! 'NEW' STATUS WAS INCORRECT
001662	005037	177776		CLR	@#PSW
001666	010701			SCOPE	;SCOPE STORES PC IN R1
001670	012767	104400	000012		;TEST THAT ALL COMBINATION OF "TRAP" WILL CAUSE A TRAP
001676	012767	001714	176130	MOV	#TRAP,RB1 ;INITIALIZE BASE TRAP INSTRUCTION
001704	012706	000500		MOV	#RA1,%4 ;RETURN FROM TRAP TO RA1
001710	104400		RC1:	MOV	#STKPTR,%6 ;SET UP STACK POINTER
001712	000000		RB1:	TRAP	;TRAP INST WILL BE MODIFIED TO TRAP+377
001714	005267	177770		HLT	;PREVIOUS INST FAILED TO TRAP
001720	022767	104777	177762	RA1:	INC RB1 ;INCREMENT TRAP INSTRUCTION
001726	103366			CMP	#104777,RB1 ;TRAP+377 TO UPPER LIMIT
001730	010701			BHIS	RC1 ;HAVE WE TESTED ALL
				SCOPE	;SCOPE STORES PC IN R1
001732	012737	000036	000034	MOV	#TRAPVEC+2,@#TRAPVEC ;RESTORE TRAP VECTOR TO
001740	005037	000036		CLR	@#TRAPVEC+2 ;HALT AT '36


```

;TEST THAT A TRAP OCCURES ON AN "IOT" INSTRUCTION
001744 010701          SCOPE
001746 012706 000500  MOV      #STKPTR,%6      ;SCOPE STORES PC IN R1
001752 012737 001770 000020  MOV      #RETA2,%IOTVEC  ;STACK POINTER SETUP
001760 005037 000022          CLR      %IOTVEC+2      ;RETURN LOCATION
001764 000004          IOT
001766 000000          HLT
001770 010701          RETA2: SCOPE      ;SCOPE STORES PC IN R1

;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
001772 012706 000500  MOV      #STKPTR,%6      ;STACK POINTER SETUP
001776 012737 002006 000020  MOV      #RETB2,%IOTVEC  ;RETURN POINTER
002004 000004          IOT
002006 020627 000474  RETB2:  CMP      %6,%STKPTR-4      ;TEST DECREMENT OF %6
002012 001401          BEQ      .+4
002014 000000          HLT
002016 010701          SCOPE      ;NOT DECREMENTED TWO WORDS
                                ;SCOPE STORES PC IN R1

;TEST THAT PROPER P.C. IS SAVED
002020 012706 000500  MOV      #STKPTR,%6      ;STACK POINTER SETUP
002024 012737 002034 000020  MOV      #RETC2,%IOTVEC  ;RETURN FROM TRAP POINTER
002032 000004          IOT
002034 022767 002034 176432  RETC2:  CMP      #STKPTR-4      ;TRAP ON THIS INSTRUCTION
002042 001401          BEQ      .+4      ;CHECK FOR INCREMENTED P.C.
002044 000000          HLT
002046 010701          SCOPE      ;INCORRECT P.C.
                                ;SCOPE STORES PC IN R1

;TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
002050 010701          SCOPE
002052 012706 000500  MOV      #STKPTR,%6      ;SCOPE STORES PC IN R1
002056 012737 002072 000020  MOV      #RETD2,%IOTVEC  ;SET UP
002064 005037 177776          CLR      %PSW          ;SET UP
002070 000004          IOT
002072 016700 176400  RETD2:  MOV      STKPTR-2,%0      ;CLEAR CC AND PRIORITY
002076 001401          BEQ      .+4      ;TRAP
002100 000000          HLT
002102 010701          SCOPE      ;GET & TEST SAVED STATUS
                                ;BRANCH IF ALL 0'S
                                ;ERROR! INCORRECT STATUS SAVED ON STACK
                                ;SCOPE STORES PC IN R1

002104 012706 000500  MOV      #STKPTR,%6      ;SET UP
002110 012737 002126 000020  MOV      #RETE2,%IOTVEC  ;SET UP
002116 012737 000357 177776  MOV      #357,%PSW      ;PRE SET STATUS
002124 000004          IOT
002126 016700 176344  RETE2:  MOV      STKPTR-2,%0      ;GET SAVED STATUS
002132 022700 000357  CMP      #357,%0      ;SAVED STATUS CORRECT?
002136 001401          BEQ      .+4      ;BRANCH IF CORRECT
002140 000000          HLT
002142 010701          SCOPE      ;ERROR! INCORRECT STATUS SAVED ON STACK
                                ;SCOPE STORES PC IN R1

;TEST THAT "NEW" STATUS IS CORRECT
002144 012706 000500  MOV      #STKPTR,%6
002150 012737 002172 000020  MOV      #RETF2,%IOTVEC
002156 005037 000022          CLR      %IOTVEC+2
002162 012737 004357 177776  MOV      #4357,%PSW      ;CLEAR FUTURE PSW
002170 000004          IOT
002172 013700 177776  RETF2:  MOV      %PSW,%0      ;PRE SET STATUS
002176 001401          BEQ      .+4      ;GET & TEST 'NEW' STATUS
                                ;BRANCH IF CORRECT

```


Address	Op Code	Register	Constant	Instruction	Comment
002200	000000			HLT	;ERROR! 'NEW' STATUS IS INCORRECT
002202	005037	177776		CLR	@#PSW
002206	010701			SCOPE	;SCOPE STORES PC IN R1
002210	012706	000500		MOV	#STKPTR,%6
002214	012737	002236	000020	MOV	#RETG2,@#IOTVEC
002222	012737	000357	000022	MOV	#357,@#IOTVEC+2
002230	005037	177776		CLR	@#PSW
002234	000004			IOT	
002236	013700	177776		MOV	@#PSW,%0
002242	022700	000357		CMP	#357,%0
002246	001401			BEQ	.+4
002250	000000			HLT	;ERROR! 'NEW' STATUS IS INCORRECT
002252	010701			SCOPE	;SCOPE STORES PC IN R1
002254	012737	000022	000020	MOV	#IOTVEC+2,@#IOTVEC
002262	005037	000022		CLR	@#IOTVEC+2

002

```

;TEST THAT A TRAP OCCURES ON AN EMT INSTRUCTION
002266 012706 000500      MOV      #STKPTR,%6      ;STACK POINTER SETUP
002272 012737 002310 000030  MOV      #RETA3,%6EMTVEC      ;RETURN LOCATION
002300 005037 000032      CLR      %6EMTVEC+2
002304 104000      EMT
002306 000000      HLT
002310 010701      RETA3:  SCOPE      ;SCOPE STORES PC IN R1

;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
002312 012706 000500      MOV      #STKPTR,%6      ;STACK POINTER SETUP
002316 012737 002326 000030  MOV      #RETB3,%6EMTVEC      ;RETURN POINTER
002324 104000      EMT
002326 020627 000474      RETB3:  CMP      %6,%STKPTR-4      ;TEST DECREMENT OF %6
002332 001401      BEQ      .+4
002334 000000      HLT
002336 010701      SCOPE      ;NOT DECREMENTED TWO WORDS
                                ;SCOPE STORES PC IN R1

;TEST THAT PROPER P.C IS SAVED
002340 012706 000500      MOV      #STKPTR,%6      ;STACK POINTER SETUP
002344 012737 002354 000030  MOV      #RETC3,%6EMTVEC      ;RETURN FROM TRAP POINTER
002352 104000      EMT      ;TRAP ON THIS INSTRUCTION
002354 022767 002354 176112  RETC3:  CMP      %6,%STKPTR-4      ;CHECK FOR INCREMENTED P.C.
002362 001401      BEQ      .+4
002364 000000      HLT
002366 010701      SCOPE      ;INCORRECT P.C.
                                ;SCOPE STORES PC IN R1

;TEST THAT "OLD" PSW AND PRIORITY ARE PLACED ON STACK
002370 012706 000500      MOV      #STKPTR,%6      ;SET UP
002374 012737 002410 000030  MOV      #RETD3,%6EMTVEC      ;SET UP
002402 005037 177776      CLR      %6PSW      ;CLEAR PSW
002406 104000      EMT      ;TRAP
002410 016700 176062      RETD3:  MOV      STKPTR-2,%0      ;GET THE SAVED STATUS
002414 001401      BEQ      .+4      ;BRANCH IF SAVED STATUS IS CORRECT
002416 000000      HLT      ;ERROR! SAVED STATUS IS INCORRECT
002420 010701      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT "OLD" PSW AND PRIORITY ARE PLACED ON STACK
002422 012706 000500      MOV      #STKPTR,%6      ;SET UP
002426 012737 002444 000030  MOV      #RETE3,%6EMTVEC      ;SET UP
002434 012737 000357 177776  MOV      #357,%6PSW      ;PRE SET STATUS
002442 104000      EMT
002444 016700 176026      RETE3:  MOV      STKPTR-2,%0      ;GET THE SAVED STATUS
002450 022700 000357      CMP      #357,%0      ;SAVED STATUS CORRECT?
002454 001401      BEQ      .+4      ;BRANCH IF CORRECT
002456 000000      HLT      ;ERROR! SAVED STATUS IS INCORRECT
002460 010701      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT "NEW" STATUS IS CORRECT
002462 012706 000500      MOV      #STKPTR,%6
002466 012737 002510 000030  MOV      #RETF3,%6EMTVEC
002474 005037 000032      CLR      %6EMTVEC+2      ;CLEAR FUTURE PSW
002500 012737 000357 177776  MOV      #357,%6PSW      ;PRE SET THE STATUS WORD
002506 104000      EMT
002510 013700 177776      RETF3:  MOV      %6PSW,%0      ;GET THE 'NEW' STATUS
002514 001401      BEQ      .+4      ;BRANCH IF 'NEW' STATUS IS CORRECT
002516 000000      HLT      ;ERROR! 'NEW' STATUS IS INCORRECT
002520 010701      SCOPE      ;SCOPE STORES PC IN R1

```


002522	012706	000500		MOV	#STKPTR,%6	
002526	012737	002550	000030	MOV	#RETG3,%EMTVEC	
002534	012737	000357	000032	MOV	#357,%EMTVEC+2	;LOAD 'NEW' STATUS
002542	005037	177776		CLR	%PSW	;PRE SET THE STATUS
002546	104000			EMT		
002550	013700	177776		RETG3: MOV	%PSW,%0	;GET THE 'NEW' STATUS
002554	022700	000357		CMP	#357,%0	;IS IT CORRECT
002560	001401			BEQ	+.4	;BRANCH IF CORRECT
002562	000000			HLT		;ERROR! 'NEW' STATUS IS INCORRECT
002564	010701			SCOPE		;SCOPE STORES PC IN R1
;TEST THAT ALL COMBINATION OF EMT WILL CAUSE A TRAP						
002566	010701			SCOPE		;SCOPE STORES PC IN R1
002570	012767	104000	000012	MOV	%EMT,RB	;INITIALIZE BASE EMT INSTRUCTION
002576	012767	002614	175224	MOV	%RA,30	;RETURN FROM TRAP TO RA
002604	012706	000500		RC: MOV	#STKPTR,%6	;SET UP STACK POINTER
002610	104000			RB: EMT		;TRAP INST. WILL BE MODIFIED TO EMT+377
002612	000000			HLT		;PREVIOUS INST FAILED TO TRAP
002614	005267	177770		RA: INC	RB	;INCREMENT TRAP INSTRUCTION
002620	022767	104377	177762	CMP	#104377,RB	;EMT+377 TO EMT?
002626	103366			BHIS	RC	;HAVE WE TESTED ALL
002630	010701			SCOPE		;SCOPE STORES PC IN R1
002632	012737	000032	000030	MOV	#EMTVEC+2,%EMTVEC	;RESTORE EMT TRAP TO HALT
002640	005037	000032		CLR	%EMTVEC+2	;AT 32


```

;TEST THAT A TRAP OCCURS ON AN BPT INSTRUCTION
002644 012706 000500      ;TEST THAT A TRAP OCCURS ON AN BPT INSTRUCTION
002650 012767 002666 175136  MOV    #STKPTR,%6      ;STACK POINTER SETUP
002656 005067 175134      MOV    #RETA4,BPTVEC    ;RETURN LOCATION
002662 000003      CLR    BPTVEC+2
002664 000000      BPT                      ;TRACE TRAP
002666 010701      HLT
RETA4:  SCOPE                      ;SCOPE STORES PC IN R1

;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
002670 012706 000500      ;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
002674 012767 002704 175112  MOV    #STKPTR,%6      ;STACK POINTER SETUP
002702 000003      MOV    #RETB4,BPTVEC    ;RETURN POINTER
002704 020627 000474      BPT                      ;RESERVED INSTRUCTION
002710 001401      RETB4:  CMP    %6,#STKPTR-4    ;TEST DECREMENT OF %6
002712 000000      BEQ    .+4
002714 010701      HLT
SCOPE                      ;NOT DECREMENTED TWO WORDS
;SCOPE STORES PC IN R1

;TEST THAT PROPER P.C. IS SAVED
002716 012706 000500      ;TEST THAT PROPER P.C. IS SAVED
002722 012767 002732 175064  MOV    #STKPTR,%6      ;STACK POINTER SETUP
002730 000003      MOV    #RETC4,BPTVEC    ;RETURN FROM TRAP POINTER
002732 022767 002732 175534  BPT                      ;TRAP ON THIS INSTRUCTION
002740 001401      RETC4:  CMP    #.#STKPTR-4    ;CHECK FOR INCREMENTED P.C.
002742 000000      BEQ    .+4
002744 010701      HLT
SCOPE                      ;INCORRECT P.C.
;SCOPE STORES PC IN R1

;TEST THAT "OLD" PSW IS PLACED ON THE STACK
002746 010701      ;TEST THAT "OLD" PSW IS PLACED ON THE STACK
002750 012706 000500      SCOPE                      ;SCOPE STORES PC IN R1
002754 012767 002770 175032  MOV    #STKPTR,%6      ;SET UP
002762 005037 177776      MOV    #RETD4,BPTVEC    ;LOAD VECTOR
002766 000003      CLR    @#PSW                ;CLEAR PSW
002770 016700 175502      RETD4:  MOV    STKPTR-2,%0    ;GET SAVED STATUS OFF STACK
002774 001401      BEQ    .+4                  ;BRANCH IF ALL 0'S
002776 000000      HLT                          ;ERROR! SAVED STATUS IS INCORRECT
003000 010701      SCOPE                      ;SCOPE STORES PC IN R1

003002 012706 000500      ;SET UP
003006 012767 003024 175000  MOV    #STKPTR,%6      ;SET UP
003014 012737 000357 177776  MOV    #RETE4,BPTVEC    ;PRE SET STATUS
003022 000003      MOV    #357,@#PSW
003024 016700 175446      RETE4:  MOV    STKPTR-2,%0    ;GET THE SAVED STATUS
003030 022700 000357      CMP    #357,%0          ;IS IT CORRECT?
003034 001401      BEQ    .+4                  ;BRANCH IF CORRECT
003036 000000      HLT                          ;ERROR! SAVED STATUS IS INCORRECT
003040 010701      SCOPE                      ;SCOPE STORES PC IN R1

;TEST THAT "NEW" STATUS IS CORRECT
003042 012706 000500      ;TEST THAT "NEW" STATUS IS CORRECT
003046 012767 003070 174740  MOV    #STKPTR,%6      ;SET UP
003054 005067 174736      MOV    #RETF4,BPTVEC    ;SET UP
003060 012737 000357 177776  CLR    BPTVEC+2      ;CLEAR FUTURE PSW
003066 000003      MOV    #357,@#PSW          ;PRE SET STATUS
003070 013700 177776      RETF4:  MOV    @#PSW,%0    ;GET & TEST 'NEW' STATUS
003074 001401      BEQ    .+4                  ;BRANCH IF ALL 0'S

```


PC	OP	RS	RT	DATA	COMMENT
003076	000000				
003100	005037	177776		HLT	;ERROR! 'NEW' STATUS IS INCORRECT
003104	010701			CLR	@#PSW
				SCOPE	;SCOPE STORES PC IN R1
003106	012706	000500		MOV	#STKPTR,%6
003112	012767	003134	174674	MOV	#RETG4,BPTVEC
003120	012767	000357	174670	MOV	#357,BPTVEC+2
003126	005037	177776		CLR	@#PSW
003132	000003			BPT	
003134	013700	177776		MOV	@#PSW,%0
003140	022700	000357		CMP	#357,%0
003144	001401			BEQ	.+4
003146	000000			HLT	
003150	005037	177776		CLR	@#PSW
003154	010701			SCOPE	;SCOPE STORES PC IN R1
003156	012767	000016	174630	MOV	#BPTVEC+2,BPTVEC
003164	005037	000016		CLR	@#BPTVEC+2
					;RESTORE 'BPT' TRAP


```

;TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION
003170 010701          SCOPE          ;SCOPE STORES PC IN R1
003172 012706 000500  MOV      #STKPTR,%6 ;STACK POINTER SETUP
003176 012737 003214 000010  MOV      #RETAS,%RESVEC ;RETURN LOCATION
003204 005037 000012          CLR      @#RESVEC+2
003210 000100          JMP      %0          ;ILLEGAL INSTRUCTION, SHOULD TRAP
003212 000000          HLT
003214 010701  RETAS:  SCOPE          ;SCOPE STORES PC IN R1

;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
003216 012706 000500  MOV      #STKPTR,%6 ;STACK POINTER SETUP
003222 012737 003232 000010  MOV      #RETB5,%RESVEC ;RETURN POINTER
003230 000101          JMP      %1          ;RESERVED INSTRUCTION
003232 020627 000474  RETB5:  CMP      %6,%STKPTR-4 ;TEST DECREMENT OF %6
003236 001401          BEQ      .+4
003240 000000          HLT
003242 010701          SCOPE          ;NOT DECREMENTED TWO WORDS
;SCOPE STORES PC IN R1

;TEST THAT PROPER P.C. IS SAVED
003244 012706 000500  MOV      #STKPTR,%6 ;STACK POINTER SETUP
003250 012737 003260 000010  MOV      #RETC5,%RESVEC ;RETURN FROM TRAP POINTER
003256 000102          JMP      %2          ;TRAP ON THIS INSTRUCTION
003260 022767 003260 175206  RETC5:  CMP      #,%STKPTR-4 ;CHECK FOR INCREMENTED P.C.
003266 001401          BEQ      .+4
003270 000000          HLT
003272 010701          SCOPE          ;INCORRECT P.C.
;SCOPE STORES PC IN R1

;TEST THAT "OLD" PSW IS PLACED ON THE STACK
003274 010701          SCOPE          ;SCOPE STORES PC IN R1
003276 012706 000500  MOV      #STKPTR,%6 ;SET UP
003302 012737 003316 000010  MOV      #RETD5,%RESVEC ;SET UP
003310 005037 177776          CLR      @#PSW          ;CLEAR PSW
003314 000103          JMP      %3          ;TRAP
003316 016700 175154  RETD5:  MOV      STKPTR-2,%0 ;GET THE SAVED STATUS
003322 001401          BEQ      .+4          ;BRANCH IF ALL 0'S
003324 000000          HLT
003326 010701          SCOPE          ;ERROR! SAVED STATUS IS INCORRECT
;SCOPE STORES PC IN R1

003330 012706 000500  MOV      #STKPTR,%6 ;SET UP
003334 012737 003352 000010  MOV      #RETE5,%RESVEC ;SET UP
003342 012737 000357 177776  MOV      #357,%PSW          ;PRE SET STATUS
003350 000104          JMP      %4
003352 016700 175120  RETE5:  MOV      STKPTR-2,%0 ;GET THE SAVED STATUS
003356 022700 000357  CMP      #357,%0 ;IS IT CORRECT
003362 001401          BEQ      .+4          ;BRANCH IF SAVED STATUS IS CORRECT
003364 000000          HLT
003366 010701          SCOPE          ;ERROR! SAVED STATUS IS INCORRECT
;SCOPE STORES PC IN R1

;TEST THAT "NEW" STATUS IS CORRECT
003370 012706 000500  MOV      #STKPTR,%6
003374 012737 003416 000010  MOV      #RETF5,%RESVEC
003402 005037 000012          CLR      @#RESVEC+2
003406 012737 000357 177776  MOV      #357,%PSW          ;PRE SET STATUS
003414 000105          JMP      %5
003416 013700 177776  RETF5:  MOV      @#PSW,%0 ;GET & TEST 'NEW' STATUS
003422 001401          BEQ      .+4

```


Address	Hex	Dec	Label	Op	Opnd	Comment
003424	000000			HLT		;ERROR! 'NEW' STATUS IS INCORRECT
003426	005037	177776		CLR	@PSW	
003432	010701			SCOPE		;SCOPE STORES PC IN R1
003434	012706	000500		MOV	#STKPTR,%6	
003440	012737	003462	000010	MOV	#RETG5,@#RESVEC	
003446	012737	000357	000012	MOV	#357,@#RESVEC+2	;LOAD 'NEW' STATUS
003454	005037	177776		CLR	@PSW	;PRE SET STATUS
003460	000106			JMP	%6	
003462	013700	177776	RETG5:	MOV	@PSW,%0	;GET THE 'NEW' STATUS
003466	022700	000357		CMP	#357,%0	;IS IT CORRECT
003472	001401			BEQ	+.4	;BRANCH IF CORRECT
003474	000000			HLT		;ERROR! 'NEW' STATUS IS INCORRECT
003476	005037	177776		CLR	@PSW	
003502	010701			SCOPE		;SCOPE STORES PC IN R1


```

003504 012706 000500      ;TEST THAT A TRAP OCCURES ON ALL ILLEGAL INSTRUCTION
003510 012737 003526 000010      MOV      #STKPTR,%6      ;STACK POINTER SETUP
003516 005037 000012      MOV      #RETH5,%RESVEC      ;RETURN LOCATION
003522 004000      CLR      @#RESVEC+2
003524 000000      JSR      %0,%0      ;ILLEGAL INSTRUCTION (SHOULD TRAP)
003526 010701      HLT
RETH5:  SCOPE      ;SCOPE STORES PC IN R1

003530 012706 000500      ;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
003534 012737 003544 000010      MOV      #STKPTR,%6      ;STACK POINTER SETUP
003542 004101      MOV      #RETJ,%RESVEC      ;RETURN POINTER
003544 020627 000474      JSR      %1,%1      ;RESERVED INSTRUCTION
003550 001401      RETJ:  CMP      %6,%STKPTR-4      ;TEST DECREMENT OF %6
003552 000000      BEQ      .+4
003554 010701      HLT      ;NOT DECREMENTED TWO WORDS
SCOPE      ;SCOPE STORES PC IN R1

003556 012706 000500      ;TEST THAT PROPER P.C. IS SAVED
003562 012737 003572 000010      MOV      #STKPTR,%6      ;STACK POINTER SETUP
003570 004202      MOV      #RETK,%RESVEC      ;RETURN FROM TRAP POINTER
003572 022767 003572 174674      JSR      %2,%2      ;TRAP ON THIS INSTRUCTION
003600 001401      RETK:  CMP      #INSTK+2,STKPTR-4      ;CHECK FOR INCREMENTED P.C.
003602 000000      BEQ      .+4
003604 010701      HLT      ;INCORRECT P.C.
SCOPE      ;SCOPE STORES PC IN R1

003606 010701      ;TEST THAT "OLD" PSW IS SAVED ON THE STACK
003610 012706 000500      SCOPE      ;SCOPE STORES PC IN R1
003614 012737 003630 000010      MOV      #STKPTR,%6      ;SET UP
003622 005037 177776      MOV      #RETL,%RESVEC      ;SET UP
003626 004303      CLR      @#PSW      ;CLEAR PSW
003630 016700 174642      JSR      %3,%3
RETL:  MOV      STKPTR-2,%0      ;GET & TEST SAVED STATUS
003634 001401      BEQ      .+4      ;BRANCH IF ALL 0'S
003636 000000      HLT      ;ERROR! INCORRECT STATUS SAVED ON STACK
003640 010701      SCOPE      ;SCOPE STORES PC IN R1

003642 012706 000500      MOV      #STKPTR,%6      ;SET UP
003646 012737 003664 000010      MOV      #RETM,%RESVEC      ;SET UP
003654 012737 000357 177776      MOV      #357,%PSW      ;PRE SET STATUS
003662 004404      JSR      %4,%4
003664 016700 174606      RETM:  MOV      STKPTR-2,%0      ;GET SAVED STATUS
003670 022700 000357      CMP      #357,%0      ;IS IT CORRECT
003674 001401      BEQ      .+4
003676 000000      HLT      ;ERROR! INCORRECT STATUS SAVED ON STACK
003700 010701      SCOPE      ;SCOPE STORES PC IN R1

003702 012706 000500      ;TEST THAT "NEW" STATUS IS CORRECT
003706 012737 003730 000010      MOV      #STKPTR,%6
003714 005037 000012      MOV      #RETN,%RESVEC
003720 012737 000357 177776      CLR      @#RESVEC+2
003726 004505      MOV      #357,%PSW      ;PRE SET STATUS
003730 013700 177776      JSR      %5,%5
RETN:  MOV      @#PSW,%0      ;GET & TEST 'NEW' STATUS
003734 001401      BEQ      .+4      ;BRANCH IF ALL 0'S

```


MAINDEC-11-DCKBM-E PDP11/45 TRAFS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 24
DCKBME.P11

Address	Hex	Dec	Label	Op	Opnd	Comment
003736	000000			HLT		;ERROR! 'NEW' STATUS IS INCORRECT
003740	005037	177776		CLR	@PSW	
003744	010701			SCOPE		;SCOPE STORES PC IN R1
003746	012706	000500		MOV	#STKPTR,%6	
003752	012737	003774	000010	MOV	#RETO,@RESVEC	
003760	012737	000357	000012	MOV	#357,@RESVEC+2	;LOAD 'NEW' STATUS
003766	005037	177776		CLR	@PSW	;PRE SET STATUS
003772	004606			JSR	%6,%6	
003774	013700	177776	RETO:	MOV	@PSW,%0	;GET THE 'NEW' STATUS
004000	022700	000357		CMP	#357,%0	;IS IT CORRECT
004004	001401			BEG	+.4	;BRANCH IF CORRECT
004006	000000			HLT		;ERROR! 'NEW' STATUS IS INCORRECT
004010	012737	000012	000010	MOV	#RESVEC+2,@RESVEC	
004016	005037	000012		CLR	@RESVEC+2	
004022	010701			SCOPE		;SCOPE STORES PC IN R1

Address	Offset	PC	Instruction	Comment
004024	012706	000500		
004030	012737	004050	000004	
004036	005037	000006		
004042	005767	173733		
004046	000000			
004050	010701			
			RETP: SCOPE	;TEST THAT A TRAP OCCURS ON AN ILLEGAL ADDRESS ;STACK POINTER SETUP ;SET TRAP VECTOR ;ILL.ADRS. (ODD ADDRESS ON WORD INST.) ;ILLEGAL ADDRESS DID NOT TRAP ;SCOPE STORES PC IN R1
004052	012706	000500		
004056	012737	004070	000004	
004064	005767	173711		
004070	020627	000474		
004074	001401			
004076	000000			
004100	010701			
			RETQ: CMP %6, #STKPTR-4 BEQ .+4 HLT SCOPE	;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION ;STACK POINTER SETUP ;RETURN POINTER ;RESERVED INSTRUCTION ;TEST DECREMENT OF %6 ;NOT DECREMENTED TWO WORDS ;SCOPE STORES PC IN R1
004102	012706	000500		
004106	012737	004120	000004	
004114	005767	173661		
004120	022767	004120	174346	
004126	001401			
004130	000000			
004132	010701			
			RETR: CMP #, STKPTR-4 BEQ .+4 HLT SCOPE	;TEST THAT PROPER P.C. IS SAVED ;STACK POINTER SETUP ;RETURN FROM TRAP POINTER ;TRAP ON THIS INSTRUCTION ;CHECK FOR INCREMENTED P.C. ;INCORRECT P.C. ;SCOPE STORES PC IN R1
004134	012706	000500		
004140	012737	004156	000004	
004146	005037	177776		
004152	005767	173623		
004156	016700	174314		
004162	001401			
004164	000000			
004166	010701			
			RETS: MOV STKPTR-2, %0 BEQ .+4 HLT SCOPE	;TEST THAT "OLD" PSW IS PLACED ON THE STACK ;SET UP ;SET UP ;GET & TEST SAVED STATUS ON STACK ;BRANCH IF SAVED STATUS IS CORRECT ;ERROR! SAVED STATUS IS INCORRECT ;SCOPE STORES PC IN R1
004170	012706	000500		
004174	012737	004214	000004	
004202	012737	000357	177776	
004210	005767	173565		
004214	016700	174256		
004220	022700	000357		
004224	001401			
004226	000000			
004230	010701			
			RETT: MOV STKPTR-2, %0 CMP #357, %0 BEQ .+4 HLT SCOPE	;SET UP ;SET UP ;PRE SET STATUS ;GET THE SAVED STATUS OFF STACK ;IS IT CORRECT ;BRANCH IF CORRECT ;ERROR! SAVED STATUS ON STACK IS INC. ;SCOPE STORES PC IN R1
004232	012706	000500		
004236	012737	004262	000004	
004244	005067	173536		
004250	012737	000357	177776	
004256	005767	173517		
004262	013700	177776		
004266	001401			
004270	000000			
			RETU: MOV #PSW, %0 BEQ .+4 HLT	;TEST THAT "NEW" STATUS IS CORRECT ;CLEAR FUTURE PSW ;PRE SET STATUS ;GET & TEST THE 'NEW' STATUS ;BRANCH IF CORRECT ;ERROR! 'NEW' STATUS IS INCORRECT


```

004272 005037 177776          CLR      @#PSW
004276 010701                SCOPE          ;SCOPE STORES PC IN R1

004300 012706 000500          MOV      #STKPTR,%6
004304 012737 004324 000004    MOV      #RETV,@#ERRVEC
004312 012737 000357 000006    MOV      #357,@#ERRVEC+2 ;LOAD 'NEW' STATUS
004320 005767 173455          TST      1
004324 013700 177776          RETV:    MOV      @#PSW,%0          ;GET THE 'NEW' STATUS
004330 022700 000357          CMP      #357,%0          ;IS IT CORRECT
004334 001401                BEQ      .+4          ;BRANCH IF CORRECT
004336 000000                HLT
004340 005037 177776          CLR      @#PSW
004344 010701                SCOPE          ;SCOPE STORES PC IN R1

;TEST THAT AN ODDSOURCE INTERMEDIATE ADDRESS CAUSES AN ODDD ADDRESS ERROR
004346 012706 000500          MOV      #STKPTR,%6          ;SET UP STACK POINTER
004352 012767 004376 173424    MOV      #OAERO,4          ;LOAD ERROR VECTOR
004360 005067 173422          CLR      6
004364 012702 000001          MOV      #1,%2          ;LOAD INDEX REGISTER
004370 067200 000000          ADD      @0(2),%0          ;SRC ADRS [0(2)] IS ODD
004374 000000                HLT          ;ERROR! ODDD ADRES ERROR FAILED TO TRAP
004376 010701                OAERO:    SCOPE          ;SCOPE STORES PC IN R1

004400 012706 000500          MOV      #STKPTR,%6
004404 012767 004424 173372    MOV      #OAER1,4
004412 012703 000477          MOV      #STKPTR-1,%3
004416 147300 000000          BICB      @0(3),%0          ;DST INT ADRS [0(3)] IS ODD
004422 000000                HLT          ;ERROR! ODD ADRS IN DST FAILED TO TRAP
004424 010701                OAER1:    SCOPE          ;SCOPE STORES PC IN R1

;TEST THAT AN ODD SOURCE FINAL ADRES WILL CAUSE AN ERROR IF A WORD INST.
004426 012706 000500          MOV      #STKPTR,%6
004432 012767 004456 173344    MOV      #OAER2,4
004440 012767 000001 173432    MOV      #1,100
004446 005004                CLR      %4
004450 017400 000100          MOV      @100(4),%0          ;SRC FINAL ADRS IS ODD
004454 000000                HLT          ;ERROR! ODD FINAL SRC ADRS FAILED TO TRAP
004456 010701                OAER2:    SCOPE          ;SCOPE STORES PC IN R1

;TEST THAT AN ODD DEST INTERMEDIATE ADDRESS CAUSES AN ODD ADRS ERROR
004460 012706 000500          MOV      #STKPTR,%6
004464 012767 004504 173312    MOV      #OAER3,4
004472 012705 000001          MOV      #1,%5
004476 074075 000000          XOR      %0,@0(5)          ;DST INT ADRS [0(5)] IS ODD
004502 000000                HLT          ;ERROR! ODD ADRS ERROR FAILED TO TRAP
004504 010701                OAER3:    SCOPE          ;SCOPE STORES PC IN R1

004506 012706 000500          MOV      #STKPTR,%6
004512 012767 004526 173264    MOV      #OAER4,4
004520 005000                CLR      %0
004522 122030                CMPB      (0)+,@(0)+          ;DST INT ADRS IS ODD [(0) =1]
004524 000000                HLT          ;ERROR! ODD ADRS ERROR FAILED TO TRAP
004526 010701                OAER4:    SCOPE          ;SCOPE STORES PC IN R1

;TEST THAT AN ODD DEST FINAL ADRS WILL CAUSE AN ODD ADRS ERROR TRAP
004530 012706 000500          MOV      #STKPTR,%6

```


004534	012767	004560	173242	MOV	#OAERS,4	
004542	012767	000001	173330	MOV	#1,100	
004550	005002			CLR	%2	
004552	006772	000100		SXT	@100(2)	; DST FINAL ADRS IS ODD
004556	000000			HLT		; ERROR! DST ODD ADRS ERR FAILED TO TRAP
004560	010701			OAERS: SCOPE		; SCOPE STORES PC IN R1
; TEST THAT CMPB @ (R)+, @ (R)+ DOES NOT FAIL						
004562	012706	000500		MOV	#STKPTR,%6	; INITIALIZE THE STACK POINTER
004566	012767	004646	173210	MOV	#OAER6,4	
004574	005002			CLR	%2	
004576	012767	016117	173174	MOV	#TEMP+1,0	
004604	012767	016120	173170	MOV	#TEMP+2,2	
004612	012767	123000	011276	MOV	#123000,TEMP	
004620	012767	177246	011272	MOV	#177246,TEMP+2	
004626	123232			CMPB	@(2)+,@(2)+	; COMP. LOC. (BYTE) TEMP+1 & TEMP+2
004630	001401			BEQ	+.4	
004632	000000			HLT		; COMPARE INSTRUCTION FAILED
004634	022702	000004		CMP	#4,%2	; DID REGISTER INCREMENT PROPERLY
004640	001401			BEQ	+.4	
004642	000000			HLT		; ERROR! R1 DID NOT INC. BY +2 TWICE
004644	000401			BR	+.4	
004646	000000			OAER6: HLT		; ERROR! ODD ADDRESS ERROR
004650	010701			SCOPE		; SCOPE STORES PC IN R1
; TEST THAT SWAB ODD ADDRESS CAUSES AN ODD ADDRESS ERROR TRAP						
004652	012706	000500		MOV	#STKPTR,%6	
004656	012767	004672	173120	MOV	#OAER7,4	
004664	000367	011227		SWAB	TEMP+1	; DO SWAB USING AN ODD ADDRESS
004670	000000			HLT		; ERROR! FAILED TO TRAP
004672	010701			OAER7: SCOPE		; SCOPE STORES PC IN R1
; TEST TIME OUT TRAP						
004674	012706	000500		MOV	#STKPTR,%6	; SET STACK PTR
004700	012737	004720	000004	MOV	#TOERO,@#ERRVEC	; LOAD ERROR VECTOR
004706	000277			SCC		; PRESET CC'S
004710	013767	177700	011200	MOV	@#177700,TEMP	; ADRS 177700 IS NOT A BUS ADDRESS
004716	000000			TOEROA: HLT		; ERROR! TIME OUT FAILED
004720	022706	000474		TOERO: CMP	#STKPTR-4,%6	; CHECK STACK PTR
004724	001401			BEQ	+.4	
004726	000000			HLT		; ERROR! INCORRECT STACK PTR AFTER TRAP
004730	022726	004714		CMP	#TOEROA-2,(6)+	; CHECK PC ON STPCK
004734	001401			BEQ	+.4	
004736	000000			HLT		; ERROR! INCORRECT PC ON STACK
004740	122716	000017		CMPB	#17,(6)	; SHECK SAVED STACTUS ON STACK
004744	001401			BEQ	+.4	
004746	000000			HLT		; ERROR! INCORRECT STATUS ON STACK
004750	010701			SCOPE		; SCOPE STORES PC IN R1
; SET STACK PTR						
004752	012706	000500		MOV	#STKPTR,%6	
004756	012737	004774	000004	MOV	#TOER1,@#ERRVEC	; LOAD ERROR VECTOR
004764	012767	177777	166006	MOV	#-1,173000	; ADRS 173000 IS A TIME OUT ON DATIP
004772	000000			TOER1A: HLT		; ERROR! FAILED TO TRAP ON TIME OUT
004774	022716	004772		TOER1: CMP	#TOER1A,(6)	; CHECK PC ON STACK
005000	001401			BEQ	+.4	
005002	000000			HLT		; ERROR! PC NOT ON STACK ON TIME OUT


```

005004 010701          SCOPE          ;SCOPE STORES PC IN R1

005006 012737 000006 000004      MOV    #ERRVEC+2, @#ERRVEC
005014 005037 000006          CLR    @#ERRVEC+2

;TEST THAT THE 'T' BIT CANNOT BE DIRECTLY SET
005020 012737 005046 000014      MOV    #NOTBIT, @#TBITVEC          ;LOAD 'T' BIT TRAP VECTOR
005026 005037 000016          CLR    @#TBITVEC+2
005032 012706 000500          MOV    #STKPTR, %6          ;SET THE STACK PTR
005036 012737 000020 177776      MOV    #20, @#PSW          ;TRY TO SET THE 'T' BIT
005044 000402          BR    .+6
005046 000000      NOTBIT: HLT
005050 010701          SCOPE          ;ERROR! 'T' BIT WAS SET BY MOVE
;SCOPE STORES PC IN R1

;TEST THAT SETTING THE 'T' BIT WILL CAUSE A TRAP TO 14
005052 012706 000500          MOV    #STKPTR, %6
005056 012767 005104 172730      MOV    #RETAT, @#PTVEC          ;SET UP TO TRAP TO 14
005064 005067 172726          CLR    @#PTVEC+2
005070 012746 000020          MOV    #20, -(6)
005074 012746 005102          MOV    #.+6, -(6)          ;PUSH 'T' BIT ON THE STACK
;PUSH PC ON THE STACK
005100 000002          RTI          ;SET 'T' BIT
005102 000000      RETAT: HLT          ;ERROR! 'T' BIT FAILED TO TRAP
005104 010701          SCOPE          ;SCOPE STORES PC IN R1

;TEST STACK POINTER DECREMENTS
005106 012706 000500          MOV    #STKPTR, %6
005112 012767 005132 172674      MOV    #RETBT, @#PTVEC
005120 012746 000020          MOV    #20, -(6)
005124 012746 005132          MOV    #.+6, -(6)          ;PUSH 'T' BIT ON THE STACK
;PUSH PC ON THE STACK
005130 000002          RTI          ;SET 'T' BIT
005132 020627 000474      RETBT: CMP    %6, #STKPTR-4
005136 001401          BEQ    .+4
005140 000000          HLT
005142 010701          SCOPE          ;STACK POINTER WAS NOT PUSHED BY TRAP
;SCOPE STORES PC IN R1

;TEST FOR PROPER PC ON STACK
005144 012706 000500          MOV    #STKPTR, %6
005150 012767 005172 172636      MOV    #RETCT, @#PTVEC
005156 012746 000020          MOV    #20, -(6)
005162 012746 005170          MOV    #.+6, -(6)          ;PUSH 'T' BIT ON THE STACK
;PUSH PC ON THE STACK
005166 000002          RTI          ;SET 'T' BIT
005170 000000      RETCT: HLT          ;ERROR! 'T' BIT TRAP NOT ACKNOWLEDGED
;BEFORE 1ST INST. AFTER AN RTI

005172 022767 005170 173274      RETCT: CMP    #.-2, STKPTR-4
005200 001401          BEQ    .+4
005202 000000          HLT
005204 010701          SCOPE          ;CORRECT PC WAS NOT SAVED ON STACK
;SCOPE STORES PC IN R1

;TEST FOR PROPER PC ON THE STACK (RTT)
005206 012706 000500          MOV    #STKPTR, %6          ;INITIALIZE THE STACK POINTER
005212 012767 005240 172574      MOV    #RETCT1, @#PTVEC          ;SET UP 'T' BIT TRAP VECTOR
005220 012746 000020          MOV    #20, -(6)
005224 012746 005232          MOV    #.+6, -(6)          ;PUSH 'T' BIT ON THE STACK
;PUSH PC ON THE STACK
005230 000006          RTT
005232 012707 005236          MOV    #.+4, %7          ;ADVANCE PC TO HLT INSTRUCTION

```



```

; 'T' BIT TRAP SHOULD OCCUR HERE
005236 000000      005236 173226 RETCT1: HLT      ;ERROR! 'T' BIT TRAP NOT ACKNOWLEDGED
005240 022767      ;CMP      #.-2,STKPTR-4 ;PROPER PC ON THE STACK
005246 001401      ;BEQ      .+4
005250 000000      ;HLT
005252 010701      ;SCOPE      ;ERROR! IMPROPER PC ON THE STACK
                                ;SCOPE STORES PC IN R1

005254 012737 000016 000014      MOV      #16,3#BPTVEC ;RESTORE 'T' TRAP
005262 005037 000016      CLR      3#BPTVEC+2 ;TO HALT AT 16

;TEST THAT IF FLOATING POINT IS NOT AVAILABLE THAT A TRAP TO 10 (RESVEC)
;OCCURS.
005266 012706 001000      FP: MOV      #1000,%6
005272 012737 005322 000010      MOV      #FPA,3#RESVEC ;LOAD RESERVED INST TRAP VECTOR
005300 005002      CLR      %2
005302 170127 147157      FPAA: LDFPS   #147157 ;DO A FLOATING POINT INST.
005306 170202      STFPS   %2 ;GET RESULT
005310 022702 147157      CMP      #147157,%2 ;CHECK RESULT
005314 001401      BEQ      .+4
005316 000000      HLT
                                ;WORKING OR IS NOT AVAILABLE AND NOT TRAPPING!!!
                                ;ERROR! EITHER FPU IS AVAILABLE AND NOT
005320 000405      BR      FPB ;EXIT TEST
005322 022767 005304 173444      FPA: CMP      #FPAA+2,774 ;WAS RETURN PC SAVED ON TRAP
005330 001401      BEQ      .+4
005332 000000      HLT
                                ;ERROR! RETURN PC NOT SAVED ON STACK
005334 012737 000012 000010      FPB: MOV      #RESVEC+2,3#RESVEC
005342 010701      SCOPE      ;SCOPE STORES PC IN R1

```



```

;TEST THAT WHEN A DATI TO A STACK LOCATION LESS THAN 400 DOES NOT CAUSE
;AN OVERFLOW TRAP.
005344 012706 000376 000004      MOV    #376,%6      ;SET STACK LESS THAN 400
005350 012737 005366      MOV    #TDEC1,3#ERRVEC
005356 005037 000006      CLR    3#ERRVEC+2
005362 005716      TST    (6)
005364 000401      BR     .+4
005366 000000      TDEC1:  HLT
005370 010701      SCOPE      ;ERROR! OVERFLOW TRAP
                                ;SCOPE STORES PC IN R1

;TEST THAT A DATIP/DATO TO A STACK LOCATION LESS THAN 400 CAUSES AN OVER-
;FLOW TRAP.
005372 012706 000376 000004      MOV    #376,%6
005376 012737 005412      MOV    #TDEC2,3#ERRVEC
005404 005016      CLR    (6)
005406 000240      NOP
005410 000000      HLT
005412 010701      TDEC2:  SCOPE      ;ERROR! OVERFLOW FAILED TO TRAP
                                ;SCOPE STORES PC IN R1

;TEST THAT A DATIP/DATOB(EVEN) CAUSES AN OVERFLOW TRAP,
005414 012706 000376 000004      MOV    #376,%6
005420 012737 005434      MOV    #TDEC2A,3#ERRVEC
005426 152716 177777      BISB   #-1,(6)
005432 000000      HLT
005434 010701      TDEC2A: SCOPE      ;ERROR! NO OVERFLOW TRAP
                                ;SCOPE STORES PC IN R1

;TEST THAT A SOURCE REFERENCE TO A 'YELLOW' ADDRESS DOES NOT OVERFLOW.
005436 012706 001000 000004      MOV    #1000,%5
005442 012706 000376      MOV    #376,%6
005446 012737 005460      MOV    #TDEC3,3#ERRVEC
005454 124645      CMPB   -(6),-(5)
005456 000401      BR     .+4
005460 000000      TDEC3:  HLT      ;ERROR! OVERFLOW TRAP
005462 010701      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT A DESTINATION DATI (USING A BINARY INST.) DOES NOT OVERFLOW
005464 012706 000400 000004      MOV    #400,%6
005470 012737 005502      MOV    #TDEC4,3#ERRVEC
005476 134546      BITB   -(5),-(6)
005500 000401      BR     .+4
005502 000000      TDEC4:  HLT      ;ERROR! OVERFLOW TRAP
005504 010701      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT OVERFLOW TRAP DOES NOT LOSE INFORMATION
005506 012706 000400      MOV    #400,%6
005512 005067 172660      CLR    376      ;STATUS WORD OF LOC 10
005516 005067 172652      CLR    374
005522 012737 005544 000004      MOV    #TDEC5,3#ERRVEC
005530 012737 000017 177776      MOV    #17,3#PSW      ;PRE SET STATUS
005536 005246      INC    -(6)
005540 000000      TDEC5A: HLT
005542 000412      BR     TDEC5B      ;ERROR! TRAP FAILED
005544 022767 000001 172624      TDEC5: CMP    #1,376      ;GO TO SCOPE
005552 001401      BEQ    .+4      ;WAS INC -(6) EXECUTED      ;SCOPE STORES PC IN R1

```



```

005554 000000          HLT          ;ERROR! INC -(6) WAS NOT EXECUTED
005556 022767 000001 172610      CMP      #1,374      ;WAS STATUS SAVED NOTE: INC DOES NOT
                                          ;AFFECT 'C' BIT IN STATUS.

005564 001401          BEQ          .+4
005566 000000          HLT
005570 010701      TDEC5B: SCOPE      ;ERROR! INCORRECT STATUS SAVED
                                          ;SCOPE STORES PC IN R1

      .MACR VTRP INST A,B,DEST
;TEST THAT INST CAUSES AN OVERFLOW TRAP
      MOV      #1000,%6      ;SET UP STACK TO OVERFLOW
      MOV      #VDEC'B,2#'DEST ;SET TRAP VECTOR
      MOV      #VDEC'A,2#ERRVEC ;SET UP OVERFLOW VECTOR
      INST
      HLT          ;SHOULD OVERFLOW
                                          ;NO TRAP OCCURRED
VDEC'B':      HLT          ;TRAP FLAG OVERFLOW DID NOT OCCUR
VDEC'A':      CMP      #VDEC'B'-2,4(6) ;CHECK RETURN PC ON THE STACK
      BEQ          .+4
      HLT          ;ERROR! RETURN PC NOT ON THE STACK
      CMP      #770,%6      ;CHECK STACK POINTER
      BEQ          .+4      ;4 PUSHES (2 FOR TRAP & 2 FOR OVERFLOW)
      HLT          ;ERROR! INCORRECT STACK PTR
      MOV      #'DEST+2,2#'DEST
      SCOPE      ;SCOPE STORES PC IN R1

      .ENDM

005572 112737 000001 177775      MOV      #1,2#177775 ;SET STACK LIMIT =1000
;TEST THAT 75000 CAUSES AN OVERFLOW TRAP
005600 012706 001000          MOV      #1000,%6      ;SET UP STACK TO OVERFLOW
005604 012737 005624 000010      MOV      #VDEC2,2#RESVEC ;SET TRAP VECTOR
005612 012737 005626 000004      MOV      #VDEC1,2#ERRVEC ;SET UP OVERFLOW VECTOR
005620 075000          75000      ;SHOULD OVERFLOW
005622 000000          HLT          ;NO TRAP OCCURRED
005624 000000          HLT          ;TRAP FLAG OVERFLOW DID NOT OCCUR
005626 022766 005622 000004      VDEC2: CMP      #VDEC2-2,4(6) ;CHECK RETURN PC ON THE STACK
005634 001401          BEQ          .+4
005636 000000          HLT          ;ERROR! RETURN PC NOT ON THE STACK
005640 022706 000770          CMP      #770,%6      ;CHECK STACK POINTER
005644 001401          BEQ          .+4      ;4 PUSHES (2 FOR TRAP & 2 FOR OVERFLOW)
005646 000000          HLT          ;ERROR! INCORRECT STACK PTR
005650 012737 000012 000010      MOV      #RESVEC+2,2#RESVEC
005656 010701          SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT IOT CAUSES AN OVERFLOW TRAP
005660 012706 001000          MOV      #1000,%6      ;SET UP STACK TO OVERFLOW
005664 012737 005704 000020      MOV      #VDEC4,2#IOTVEC ;SET TRAP VECTOR
005672 012737 005706 000004      MOV      #VDEC3,2#ERRVEC ;SET UP OVERFLOW VECTOR
005700 000004          IOT          ;SHOULD OVERFLOW
005702 000000          HLT          ;NO TRAP OCCURRED
005704 000000          HLT          ;TRAP FLAG OVERFLOW DID NOT OCCUR
005706 022766 005702 000004      VDEC4: CMP      #VDEC4-2,4(6) ;CHECK RETURN PC ON THE STACK
005714 001401          BEQ          .+4
005716 000000          HLT          ;ERROR! RETURN PC NOT ON THE STACK
005720 022706 000770          CMP      #770,%6      ;CHECK STACK POINTER
005724 001401          BEQ          .+4      ;4 PUSHES (2 FOR TRAP & 2 FOR OVERFLOW)
005726 000000          HLT          ;ERROR! INCORRECT STACK PTR

```



```

005730 012737 000022 000020      MOV      #IOTVEC+2,2#IOTVEC
005736 010701                      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT EMT CAUSES AN OVERFLOW TRAP
005740 012706 001000                      MOV      #1000,%6      ;SET UP STACK TO OVERFLOW
005744 012737 005764 000030      MOV      #VDEC6,2#EMTVEC      ;SET TRAP VECTOR
005752 012737 005766 000004      MOV      #VDEC5,2#ERRVEC      ;SET UP OVERFLOW VECTOR
005760 104000                      EMT                      ;SHOULD OVERFLOW
005762 000000                      HLT                      ;NO TRAP OCCURRED
005764 000000                      VDEC6: HLT                      ;TRAP FLAG OVERFLOW DID NOT OCCUR
005766 022766 005762 000004      VDEC5: CMP      #VDEC6-2,4(6) ;CHECK RETURN PC ON THE STACK
005774 001401                      BEQ      .+4
005776 000000                      HLT
006000 022706 000770                      CMP      #770,%6      ;ERROR! RETURN PC NOT ON THE STACK
006004 001401                      BEQ      .+4      ;CHECK STACK POINTER
006006 000000                      HLT                      ;4 PUSHES (2 FOR TRAP & 2 FOR OVERFLOW)
006010 012737 000032 000030      MOV      #EMTVEC+2,2#EMTVEC ;ERROR! INCORRECT STACK PTR
006016 010701                      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT TRAP CAUSES AN OVERFLOW TRAP
006020 012706 001000                      MOV      #1000,%6      ;SET UP STACK TO OVERFLOW
006024 012737 006044 000034      MOV      #VDEC8,2#TRAPVEC      ;SET TRAP VECTOR
006032 012737 006046 000004      MOV      #VDEC7,2#ERRVEC      ;SET UP OVERFLOW VECTOR
006040 104400                      TRAP                     ;SHOULD OVERFLOW
006042 000000                      HLT                      ;NO TRAP OCCURRED
006044 000000                      VDEC8: HLT                      ;TRAP FLAG OVERFLOW DID NOT OCCUR
006046 022766 006042 000004      VDEC7: CMP      #VDEC8-2,4(6) ;CHECK RETURN PC ON THE STACK
006054 001401                      BEQ      .+4
006056 000000                      HLT
006060 022706 000770                      CMP      #770,%6      ;ERROR! RETURN PC NOT ON THE STACK
006064 001401                      BEQ      .+4      ;CHECK STACK POINTER
006066 000000                      HLT                      ;4 PUSHES (2 FOR TRAP & 2 FOR OVERFLOW)
006070 012737 000036 000034      MOV      #TRAPVEC+2,2#TRAPVEC ;ERROR! INCORRECT STACK PTR
006076 010701                      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT BPT CAUSES AN OVERFLOW TRAP
006100 012706 001000                      MOV      #1000,%6      ;SET UP STACK TO OVERFLOW
006104 012737 006124 000014      MOV      #VDEC10,2#BPTVEC      ;SET TRAP VECTOR
006112 012737 006126 000004      MOV      #VDEC9,2#ERRVEC      ;SET UP OVERFLOW VECTOR
006120 000003                      BPT                     ;SHOULD OVERFLOW
006122 000000                      HLT                      ;NO TRAP OCCURRED
006124 000000                      VDEC10: HLT                      ;TRAP FLAG OVERFLOW DID NOT OCCUR
006126 022766 006122 000004      VDEC9: CMP      #VDEC10-2,4(6) ;CHECK RETURN PC ON THE STACK
006134 001401                      BEQ      .+4
006136 000000                      HLT
006140 022706 000770                      CMP      #770,%6      ;ERROR! RETURN PC NOT ON THE STACK
006144 001401                      BEQ      .+4      ;CHECK STACK POINTER
006146 000000                      HLT                      ;4 PUSHES (2 FOR TRAP & 2 FOR OVERFLOW)
006150 012737 000016 000014      MOV      #BPTVEC+2,2#BPTVEC ;ERROR! INCORRECT STACK PTR
006156 010701                      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT ILLA CAUSES AN OVERFLOW TRAP
006160 012706 001000                      MOV      #1000,%6      ;SET UP STACK TO OVERFLOW
006164 012737 006204 000010      MOV      #VDEC11,2#RESVEC      ;SET TRAP VECTOR
006172 012737 006206 000004      MOV      #VDEC12,2#ERRVEC      ;SET UP OVERFLOW VECTOR
006200 004700                      ILLA                     ;SHOULD OVERFLOW

```



```

006202 000000          HLT                ;NO TRAP OCCURRED
006204 000000          VDEC11: HLT          ;TRAP FLAG OVERFLOW DID NOT OCCUR
006206 022766 006202 000004 VDEC12: CMP    #VDEC11-2,4(6) ;CHECK RETURN PC ON THE STACK
006214 001401          BEQ    .+4
006216 000000          HLT                ;ERROR! RETURN PC NOT ON THE STACK
006220 022706 000770    CMP    #770,%6    ;CHECK STACK POINTER
006224 001401          BEQ    .+4          ;4 PUSHES (2 FOR TRAP & 2 FOR OVERFLOW)
006226 000000          HLT                ;ERROR! INCORRECT STACK PTR
006230 012737 000012 000010 MOV    #RESVEC+2,2#RESVEC
006236 010701          SCOPE              ;SCOPE STORES PC IN R1

```

```

.MACR VTRPO INST,A
;TEST THAT THE INSTRUCTION ('INST') CAUSES AN OVERFLOW CONDITION.
MOV    #1000,%6          ;SET STACK POINTER
MOV    #VDEC'A,2#ERRVEC  ;LOAD ERROR VECTOR
INST
HLT
VDEC'A: CMP    #772,%6    ;CAUSE OVERFLOW
BEQ    .+4               ;ERROR! OVERFLOW FAILED TO TRAP
HLT
SCOPE                    ;HAS STACK POINTER MOVED BY 6
                          ; (2 FOR THE AUTO DECREMENT + 4 FOR
                          ; THE ERROR TRAP)
                          ;SCOPE STORES PC IN R1

```

.ENDM

```

; INSTRUCTION EQUATE STATEMENTS
:
: 4510    =JSR    5,(0)
: 005046  =CLR    -(6)
: 010046  =MOV    %0,-(6)
: 006746  =SXT    -(6)
: 074046  =XOR    %0,-(6)
:

```

```

;TEST THAT THE INSTRUCTION (4510) CAUSES AN OVERFLOW CONDITION.
006240 012706 001000          MOV    #1000,%6          ;SET STACK POINTER
006244 012737 006256 000004 MOV    #VDEC15,2#ERRVEC  ;LOAD ERROR VECTOR
006252 004510          4510          ;CAUSE OVERFLOW
006254 000000          HLT                ;ERROR! OVERFLOW FAILED TO TRAP
006256 022706 000772    VDEC15: CMP    #772,%6        ;HAS STACK POINTER MOVED BY 6
006262 001401          BEQ    .+4          ; (2 FOR THE AUTO DECREMENT + 4 FOR
006264 000000          HLT                ; THE ERROR TRAP)
006266 010701          SCOPE              ;SCOPE STORES PC IN R1

```

```

;TEST THAT THE INSTRUCTION (005046) CAUSES AN OVERFLOW CONDITION.
006270 012706 001000          MOV    #1000,%6          ;SET STACK POINTER
006274 012737 006306 000004 MOV    #VDEC16,2#ERRVEC  ;LOAD ERROR VECTOR
006302 005046          005046          ;CAUSE OVERFLOW
006304 000000          HLT                ;ERROR! OVERFLOW FAILED TO TRAP
006306 022706 000772    VDEC16: CMP    #772,%6        ;HAS STACK POINTER MOVED BY 6
006312 001401          BEQ    .+4          ; (2 FOR THE AUTO DECREMENT + 4 FOR
006314 000000          HLT                ; THE ERROR TRAP)
006316 010701          SCOPE              ;SCOPE STORES PC IN R1

```

```

;TEST THAT THE INSTRUCTION (010046) CAUSES AN OVERFLOW CONDITION.
006320 012706 001000          MOV    #1000,%6          ;SET STACK POINTER
006324 012737 006336 000004 MOV    #VDEC17,2#ERRVEC  ;LOAD ERROR VECTOR
006332 010046          010046          ;CAUSE OVERFLOW

```



```

006334 000000
006336 022706 000772
006342 001401
006344 000000
006346 010701
VDEC17: HLT
CMP #772,%6
BEQ .+4
HLT
SCOPE
;ERROR! OVERFLOW FAILED TO TRAP
;HAS STACK POINTER MOVED BY 6
;(2 FOR THE AUTO DECREMENT + 4 FOR
;THE ERROR TRAP)
;SCOPE STORES PC IN R1

;TEST THAT THE INSTRUCTION (006746) CAUSES AN OVERFLOW CONDITION.
006350 012706 001000
006354 012737 006366 000004
006362 006746
006364 000000
006366 022706 000772
006372 001401
006374 000000
006376 010701
VDEC18: MOV #1000,%6
MOV #VDEC18,%ERRVEC
006746
HLT
CMP #772,%6
BEQ .+4
HLT
SCOPE
;SET STACK POINTER
;LOAD ERROR VECTOR
;CAUSE OVERFLOW
;ERROR! OVERFLOW FAILED TO TRAP
;HAS STACK POINTER MOVED BY 6
;(2 FOR THE AUTO DECREMENT + 4 FOR
;THE ERROR TRAP)
;SCOPE STORES PC IN R1

;TEST THAT THE INSTRUCTION (074046) CAUSES AN OVERFLOW CONDITION.
006400 012706 001000
006404 012737 006416 000004
006412 074046
006414 000000
006416 022706 000772
006422 001401
006424 000000
006426 010701
VDEC19: MOV #1000,%6
MOV #VDEC19,%ERRVEC
074046
HLT
CMP #772,%6
BEQ .+4
HLT
SCOPE
;SET STACK POINTER
;LOAD ERROR VECTOR
;CAUSE OVERFLOW
;ERROR! OVERFLOW FAILED TO TRAP
;HAS STACK POINTER MOVED BY 6
;(2 FOR THE AUTO DECREMENT + 4 FOR
;THE ERROR TRAP)
;SCOPE STORES PC IN R1

;TEST THAT INTERRUPT CAUSES AN OVERFLOW
006430 012706 001000
006434 012737 006500 000004
006442 012737 006526 000064
006450 000237
006452 105737 177564
006456 100375
006460 012737 000100 177564
006466 005037 177776
006472 000237
006474 000000
006476 000754
VDEC20: MOV #1000,%6
MOV #VDC20,%ERRVEC
MOV #VDC20B,%TPVEC
SPL 7
IS: TSTB %TTCSR
BPL IS
MOV #100,%TTCSR
CLR %PSW
VDC20A: SPL 7
HLT
BR VDEC20
;SET STACK PTR
;LOAD OVERFLOW VECTOR
;LOAD TTY PRINTER VECTOR
;LOCK OUT INTERRUPTS
;WAIT FOR TTY TO BE READY
;SET IE BIT IN TTY PRINTER
;ALLOW INTERRUPTS
;LOCK OUT INTERRUPTS
;ERROR! NO TTY INTERRUPT
;LOOP TEST IF NO TTY INTERRUPT

;TEST THE STACK
006500 022767 006472 172266
006506 001401
006510 000000
006512 022767 006526 172250
006520 001401
006522 000000
006524 000401
006526 000000
006530 005037 177564
006534 012737 000066 000064
006542 012737 000006 000004
006550 105037 177775
006554 010701
VDC20: CMP #VDC20A,774
BEQ .+4
HLT
CMP #VDC20B,770
BEQ .+4
HLT
BR .+4
VDC20B: HLT
CLR %TTCSR
MOV #TPVEC+2,%TPVEC
MOV #ERRVEC+2,%ERRVEC
CLRB %177775
SCOPE
;IS RETURN PC (FROM INT.) ON THE STACK?
;ERROR!
;IS RETURN PC (FROM OVERFLOW) ON STACK
;ERROR!
;ERROR! NO OVERFLOW
;CLEAR IE BIT
;SET STACK LIMIT =400
;SCOPE STORES PC IN R1

```

```

;TEST FOR FALSE OVERFLOW TRAP
;PROGRAM MAY HAVE RELOADED IF OVERFLOW FAILS

```


006556	012767	006632	171220	MOV	#FOVER,4	;SET UP OVERFLOW POINTER
006564	012706	000402		MOV	#402,%6	
006570	014626			MOV	-(6),(6)+	;SHOULD NOT OVERFLOW
006572	012706	001002		MOV	#1002,%6	
006576	014626			MOV	-(6),(6)+	;SHOULD NOT OVERFLOW
006600	012706	002002		MOV	#2002,%6	
006604	014626			MOV	-(6),(6)+	;SHOULD NOT OVERFLOW
006606	012706	004002		MOV	#4002,%6	
006612	014626			MOV	-(6),(6)+	;SHOULD NOT OVERFLOW
006614	012706	010002		MOV	#10002,%6	
006620	014626			MOV	-(6),(6)+	;SHOULD NOT OVERFLOW
006622	012706	020000		MOV	#20000,%6	;SHOULD NOT OVERFLOW
006626	005746			TST	-(6)	
006630	000401			BR	+.4	
006632	000000			HLT		;FALSE OVERFLOW OCCURRED
006634	010701			SCOPE		;SCOPE STORES PC IN R1

THIS TEST SETS THE STACK IN THE 'YELLOW' ZONE AND THEN DOES EACH OF
THE TRAP INSTRUCTIONS CAUSING AN OVERFLOW CONDITION AFTER EACH.
WHAT THE STACK LOOKS LIKE WHEN INSTRUCTION AT 04 IS BEING EXECUTED

OVFLW	PC=05,STAT=0	04	342
		4	344
BPT	PC=04,STAT=4	03A	346
		0	350
OVFLW	PC=05,STAT=0	03	352
		3	354
IOT	PC=03,STAT=3	02A	356
		0	360
OVFLW	PC=05,STAT=0	02	362
		2	364
TRAP	PC=02,STAT=2	01A	366
		0	370
OVFLW	PC=05,STAT=0	01	372
		1	374
EMT	PC=01,STAT=1	00A	376
		17	400
			402 INITIAL STACK POINTER

006636	012706	000402		MOV	#402,%6	;INITIALIZE STACK POINTER
006642	012737	006746	000030	MOV	#01,%#EMTVEC	
006650	012737	000001	000032	MOV	#1,%#EMTVEC+2	
006656	012737	006752	000034	MOV	#02,%#TRAPVEC	
006664	012737	000002	000036	MOV	#2,%#TRAPVEC+2	
006672	012737	006756	000020	MOV	#03,%#IOTVEC	
006700	012737	000003	000022	MOV	#3,%#IOTVEC+2	
006706	012767	006762	171100	MOV	#04,%#BPTVEC	
006714	012767	000004	171074	MOV	#4,%#BPTVEC+2	
006722	012737	007200	000004	MOV	#05,%#ERRVEC	
006730	005037	000006		CLR	%#ERRVEC+2	
006734	012737	000017	177776	MOV	#17,%#PSW	
006742	104000			EMT		
006744	000000			HLT		;EMT NOT EXECUTED
006746	104400			TRAP		
006750	000000			HLT		;TRAP NOT EXECUTED
006752	000004			IOT		
006754	000000			HLT		;IOT NOT EXECUTED
006756	000003			BPT		

006760	000000		03A:	HLT			
006762	022706	000342	04:	CMP	#342,%6		;BPT NOT EXECUTED
006766	001401			BEQ	.+4		;IS STACK POINTER
006770	000000			HLT			;POSITIONED PROPERLY
006772	012700	000402		MOV	#402,%0		;ERROR! INCORRECT STACK POINTER
006776	022740	000017		CMP	#17,-(0)		
007002	001401			BEQ	.+4		
007004	000000			HLT			;STACK ERROR
007006	022740	006744		CMP	#00A,-(0)		
007012	001401			BEQ	.+4		
007014	000000			HLT			
007016	022740	000001		CMP	#1,-(0)		
007022	001401			BEQ	.+4		
007024	000000			HLT			
007026	022740	006746		CMP	#01,-(0)		
007032	001401			BEQ	.+4		
007034	000000			HLT			
007036	022740	000000		CMP	#0,-(0)		
007042	001401			BEQ	.+4		
007044	000000			HLT			
007046	022740	006750		CMP	#01A,-(0)		
007052	001401			BEQ	.+4		
007054	000000			HLT			
007056	022740	000002		CMP	#2,-(0)		
007062	001401			BEQ	.+4		
007064	000000			HLT			
007066	022740	006752		CMP	#02,-(0)		
007072	001401			BEQ	.+4		
007074	000000			HLT			
007076	022740	000000		CMP	#0,-(0)		
007102	001401			BEQ	.+4		
007104	000000			HLT			
007106	022740	006754		CMP	#02A,-(0)		
007112	001401			BEQ	.+4		
007114	000000			HLT			
007116	022740	000003		CMP	#3,-(0)		
007122	001401			BEQ	.+4		
007124	000000			HLT			
007126	022740	006756		CMP	#03,-(0)		
007132	001401			BEQ	.+4		
007134	000000			HLT			
007136	022740	000000		CMP	#0,-(0)		
007142	001401			BEQ	.+4		
007144	000000			HLT			
007146	022740	006760		CMP	#03A,-(0)		
007152	001401			BEQ	.+4		
007154	000000			HLT			
007156	022740	000004		CMP	#4,-(0)		
007162	001401			BEQ	.+4		
007164	000000			HLT			
007166	022740	006762		CMP	#04,-(0)		
007172	001401			BEQ	.+4		
007174	000000		STKERR:	HLT			;INCORRECT VALUE ON THE STACK
007176	000402			BR	06		;RO GIVES LOCATION OF STACK ERROR
007200	000176	000000	05:	JMP	0(6)		;RETURN FROM ERROR

Address	Instruction	Comment
007204	010701	06: SCOPE ;SCOPE STORES PC IN R1
;TEST BPT TRAP WHEN STACK IS IN YELLOW AREA.		
007206	005067	006704 CLR TEMP
007212	012706	000400 MOV #400,%6 ;SET UP STACK POINTER
007216	012767	007342 170570 MOV #BPTC,BPTVEC ;LOAD BPT VECTOR
007224	005067	170566 CLR BPTVEC+2
007230	012737	007266 000004 MOV #BPTB,2#ERRVEC ;LOAD OVFLW. ERROR VECTOR
007236	005067	170544 CLR 6
007242	016767	000012 006650 MOV BPTA,TEMP+2 ;SAVE FIRST WORD OF INST. AT BPTA
007250	012767	000003 000002 MOV #BPT,BPTA ;REPLACE WITH BPT INSTRUCTION
007256	000277	SCC ;SET CONDITION CODES
007260	005167	006632 BPTA: COM TEMP ;NOTE: THIS INSTRUCTION WAS REPLACED BY BPT
007264	000475	BPTAA: BR BPTF
007266	022767	000017 171102 BPTB: CMP #17,376 ;WERE CONDITION CODES SAVED
007274	001401	BEQ .+4
007276	000000	HLT ;ERROR! CONDITION CODES NOT SAVED
007300	022767	007262 171066 CMP #BPTA+2,374 ;WAS RETURN FROM BPT SAVED
007306	001402	BEQ .+6
007310	000000	HLT ;ERROR! RETURN PC NOT SAVED
007312	000467	BR BPTEX ;EXIT TEST
007314	005767	171052 TST 372 ;WAS BPT STATUS SAVED?
007320	001401	BEQ .+4
007322	000000	HLT ;ERROR! BPT STATUS (BPTVEC+2) NOT SAVED
007324	022767	007342 171036 CMP #BPTC,370 ;BPT VECTOR SAVED
007332	001402	BEQ .+6
007334	000000	HLT ;ERROR! BPT VECTOR (BPTVEC) NOT SAVED
007336	000455	BR BPTEX ;EXIT TEST
007340	000002	RTI ;RETURN FROM OVERFLOW TRAP
007342	010605	BPTC: MOV %6,%5 ;SAVE STACK POINTER
007344	162715	000002 SUB #2,(5) ;POINT PC TO TRAPPED INSTRUCTION
007350	016735	006544 MOV TEMP+2,2(5)+ ;RESTORE ORIGINAL INSTRUCTION (COM TEMP)
007354	052715	000020 BIS #20,(5) ;SET 'T' BIT ON STACK
007360	012737	007376 000004 MOV #BPTD,2#ERRVEC ;CHANGE ERROR VECTOR
007366	012767	007452 170420 MOV #BPTC,TBITVEC ;LOAD 'T' BIT VECTOR
007374	000006	RTI ;GO DO ORIGINAL INSTRUCTION AND TRAP
;('T' BIT TRAP) WHEN FINISHED		
;CHECK THE STACK		
007376	022767	000031 170772 BPTD: CMP #31,376 ;IS 'N','C',&'T' BITS SET ON THE STACK
007404	001401	BEQ .+4 ;(RESULT OF COM TEMP IN BPTA
007406	000000	HLT ;ERROR!STATUS NOT SAVED
007410	022767	007264 170756 CMP #BPTAA,374 ;WAS RETURN PC FROM 'T' TRAP SAVED?
007416	001402	BEQ .+6
007420	000000	HLT ;ERROR! RETURN PC NOT SAVED
007422	000423	BR BPTEX ;EXIT TEST
007424	005767	170742 TST 372 ;'T' BIT TRAP STATUS SAVED?
007430	001401	BEQ .+4
007432	000000	HLT ;ERROR! 'T' BIT STATUS (TBITVEC+2) NOT SAVED
007434	022767	007452 170726 CMP #BPTC,370 ;'T' BIT VECTOR SAVED (TBITVRC)
007442	001402	BEQ .+6
007444	000000	HLT ;ERROR! 'T' BIT VECTOR NOT SAVED
007446	000411	BR BPTEX ;EXIT TEST
007450	000002	RTI ;RETURN FROM ERROR TRAP


```

007452 005067 170720      BPTC: CLR 376      ;CLEAR SAVED STATUS
007456 000002              RTI              ;RETURN FROM 'T' BIT TRAP

007460 022767 177777 006430 BPTF: CMP #-1,TEMP ;WAS ORIGINAL INSTRUCTION EXECUTED?
007466 001401              BEQ .+4
007470 000000              HLT              ;TRACED INSTRUCTION (CCM TEMP) NOT EXEC.
007472 012767 000016 170314 BPTX: MOV #BPTVEC+2,BPTVEC;RESTORE TRAP
007500 012737 000006 000004      MOV #ERRVEC+2,#ERRVEC ;VECTORS
007506 010701              SCOPE            ;SCOPE STORES PC IN R1

;TEST THAT INDEX MODE (6) USING R6 AS THE REGISTER OVERFLOWS IF THE AD-
;DRESS IS BETWEEN 376-340.
007510 012706 001000      INDEX: MOV #1000,%6 ;SET UP STACK POINTER
007514 012737 007534 000004      MOV #IND1,#ERRVEC
007522 005067 170260      CLR 6
007526 005066 177340      CLR -440(6) ;FINAL ADDRESS IS 340
007532 000000              HLT              ;ERROR! DID NOT TRAP
007534 010701      IND1: SCOPE            ;SCOPE STORES PC IN R1

;TEST THAT THE TRAP SEQUENCE IS EXECUTED PROPERLY ON OVERFLOW.
007536 012706 001000      MOV #1000,%6
007542 012737 007600 000004      MOV #IND2,#ERRVEC
007550 005067 171222      CLR 776
007554 005067 171214      CLR 774
007560 012767 000001 170610      MOV #1,376 ;PRE SET DEST ADDRESS
007566 012766 000000 177376      MOV #0,-402(6) ;FINAL ADDRESS IS 376
007574 000000      IND2A: HLT              ;ERROR! FAILED TO TRAP
007576 000422      BR IND2X              ;GO TO SCOPE ;SCOPE STORES PC IN R1
007600 026727 171172 000004 IND2: CMP 776,#4 ;STATUS SAVED ON THE STACK (Z BIT SET)
007606 001401      BEQ .+4 ;[RESULT OF MOV #0,-402(6)]
007610 000000      HLT              ;ERROR! INCORRECT STATUS SAVED
007612 022767 007574 171154      CMP #IND2A,774 ;RETURN PC SAVED?
007620 001401      BEQ .+4
007622 000000      HLT              ;ERROR! RETURN PC NOT SAVED
007624 020627 000774      CMP %6,#774 ;CORRECT STACK POINTER?
007630 001401      BEQ .+4
007632 000000      HLT              ;ERROR! INCORRECT STACK POINTER
007634 005767 170536      TST 376
007640 001401      BEQ .+4 ;WAS MOVE INST. EXECUTED
007642 000000      HLT              ;ERROR! MOVE NOT EXECUTED
007644 010701      IND2X: SCOPE            ;SCOPE STORES PC IN R1

;THIS TEST IS THE SAME AS ABOVE EXCEPT THAT THE STACK POINTER IS INIT-
;IALLY IN THE YELLOW ZONE.
007646 012706 000376      MOV #376,%6 ;SET UP STACK IN 'YELLOW' AREA
007652 012737 007676 000004      MOV #IND3,#ERRVEC ;SET UP ERROR RETURN
007660 005067 170510      CLR 374
007664 005067 170502      CLR 372
007670 005066 177742      CLR -36(6) ;FINAL ADDRESS IS 340 (YELLOW)
007674 000000      IND3A: HLT              ;FAILED TO TRAP
007676 022767 000004 170470 IND3: CMP #4,374 ;SAVED STATUS CORRECT?
007704 001401      BEQ .+4
007706 000000      HLT              ;SAVED STATUS WAS INCORRECT (Z SET)
007710 022767 007674 170454      CMP #IND3A,372 ;SAVED RETURN ADDRESS CORRECT?
007716 001401      BEQ .+4
007720 000000      HLT              ;ERROR! SAVED RETURN ADDRESS INCORRECT

```



```

007722 022706 000372      CMP      #372,%6      ;STACK POINTER CORRECT
007726 001401      BEQ      .+4
007730 000000      HLT
007732 005767 170402      TST      340      ;ERROR! STACK POINTER INCORRECT
007736 001401      BEQ      .+4      ;WAS CLEAR INSTRUCTION EXECUTED?
007740 000000      HLT      ;ERROR! CLEAR NOT EXECUTED
007742 010701      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT AN OVERFLOW OCCURS IF THE ADDRESS IS IN THE 'RED' ZONE AND ALSO
;THAT THE INSTRUCTION [CLRB -442(6)] IS ABORTED.
007744 012706 001000      MOV      #1000,%6      ;SET STACK PTR
007750 012737 010016 000004      MOV      #IND4,%ERRVEC ;SET ERROR TRAP VECTOR
007756 012767 177777 170014      MOV      #-1,0
007764 012767 177777 170010      MOV      #-1,2
007772 005067 171000      CLR      776
007776 005037 177776      CLR      %PSW
010002 012767 177777 170326      MOV      #-1,336      ;PRE SET 336
010010 105066 177336      CLRB      -442(6)      ;CLEAR 'RED' LOCATION (336)
010014 000000      HLT      ;ERROR! FAILED TO TRAP
010016 005706      IND4A:      TST      %6      ;STACK POINTER CORRECT?
010020 001401      IND4:      BEQ      .+4
010022 000000      HLT      ;ERROR! INCORRECT STACK POINTER
010024 022767 177777 170304      CMP      #-1,336      ;WAS CLEAR INSTRUCTION ABORTED?
010032 001401      BEQ      .+4
010034 000000      HLT      ;ERROR! CLEAR INSTRUCTION NOT ABORTED
010036 022767 000010 167736      CMP      #10,2      ;WAS PROPER STATUS SAVED? (N SET)
010044 001401      BEQ      .+4      ;RESULT OF MOV #-1,336
010046 000000      HLT      ;ERROR! INCORRECT STATUS WAS SAVED
010050 022767 010014 167722      CMP      #IND4A,0      ;WAS RETURN PC SAVED?
010056 001401      BEQ      .+4
010060 000000      HLT
010062 005767 170710      TST      776      ;WAS NOTHING PUT ON THE OLD STACK?
010066 001401      BEQ      .+4
010070 000000      HLT      ;ERROR! RED TRAP DID NOT STACK AT 2 & 0
010072 010701      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT INDEX MODE USING R6 IN SOURCE DOES NOT TRAP.
010074 012706 001000      MOV      #1000,%6      ;SET UP STACK
010100 012737 010124 000004      MOV      #IND5,%ERRVEC ;SET UP ERROR TRAP
010106 012767 177777 170234      MOV      #-1,350
010114 016667 177350 005774      MOV      -430(6),TEMP
010122 000401      BR      .+4
010124 000000      IND5:      HLT      ;ERROR! TRAPPED
010126 010701      SCOPE      ;SCOPE STORES PC IN R1

;THIS TEST IS THE SAME AS ABOVE EXCEPT THAT LOCATION IS 'RED'.
010130 012706 001000      MOV      #1000,%6
010134 012737 010152 000004      MOV      #IND6,%ERRVEC ;SET ERROR TRAP VECTOR
010142 016667 177336 005746      MOV      -442(6),TEMP
010150 000401      BR      .+4
010152 000000      IND6:      HLT      ;ERROR! TRAPPED
010154 010701      SCOPE      ;SCOPE STORES PC IN R1

;TEST THAT TRAP DOES NOT OCCUR IF INDEX DEFERRED USING R6 IN DEST IF
;INTERMEDIATE ADDRESS IS 'YELLOW' AND FINAL ADDRESS IS NOT 'YELLOW' OR 'RED'.
010156 012706 001000      MOV      #1000,%6

```


010162	012737	010204	000004	MOV	#IND7,2#ERRVEC	;SET ERROR TRAP VECTOR
010170	012767	016116	170152	MOV	#TEMP,350	
010176	005076	177350		CLR	2-430(6)	;CLEAR TEMP
010202	000401			BR	.+4	
010204	000000			IND7: HLT		;ERROR! TRAPPED
010206	010701			SCOPE		;SCOPE STORES PC IN R1
;THIS TEST IS THE SAME AS ABOVE EXCEPT THE INTERMEDIATE ADDRESS IS 'RED'.						
010210	012706	001000		MOV	#1000,%6	
010214	012737	010240	000004	MOV	#IND10,2#ERRVEC	
010222	012767	016116	170106	MOV	#TEMP,336	
010230	012776	177777	177336	MOV	#-1,2-442(6)	;MOVE #-1 TO TEMP
010236	000401			BR	.+4	
010240	000000			IND10: HLT		;ERROR TRAPPED
010242	010701			SCOPE		;SCOPE STORES PC IN R1
;TEST THAT INDEX DEFERRED DOES NOT OVERFLOW IF THE FINAL ADDRESS IS 'YELLOW'						
010244	012706	000776		MOV	#776,%6	
010250	012737	010272	000004	MOV	#IND11,2#ERRVEC	
010256	012767	000376	170512	MOV	#376,776	
010264	005076	000000		CLR	2(6)	;THIS INSTRUCTION CLEARS 376
010270	000401			BR	.+4	;ADDRESS MODE 7 USING R6 DOES NOT TRAP
010272	000000			IND11: HLT		;ERROR! TRAPPED
010274	010701			SCOPE		;SCOPE STORES PC IN R1
;THIS TEST IS THE SAME AS ABOVE EXCEPT THAT THE FINAL ADDRESS IS 'RED'.						
010276	012706	000776		MOV	#776,%6	
010302	012737	010330	000004	MOV	#IND12,2#ERRVEC	
010310	012767	000336	170460	MOV	#336,776	
010316	005067	170014		CLR	336	
010322	005176	000000		COM	2(6)	
010326	000401			IND12A: BR	.+4	;ADDRESS MODE 7 USING R6 DOES NOT TRAP
010330	000000			IND12: HLT		
010332	010701			SCOPE		;SCOPE STORES PC IN R1
;TEST THAT OVERFLOW OPERATES PROPERLY WHEN THE STACK POINTER IS 'YELLOW' ;AND A 'RED' ADDRESS IS REFERENCED.						
010334	012706	000376		MOV	#376,%6	;SET STACK IN 'YELLOW' ZONE
010340	005067	167772		CLR	336	
010344	012737	010360	000004	MOV	#IND13,2#ERRVEC	
010352	005166	177740		COM	-40(6)	;ATTEMPT TO COMPLEMENT 336
010356	000000			IND13A: HLT		;ERROR! NO OVERFLOW TRAP
010360	022767	010356	167412	IND13: CMP	#IND13A,0	;WAS STACK REPOSITIONED
010366	001401			BEQ	.+4	
010370	000000			HLT		
010372	005767	167740		TST	336	;WAS INST. COM -40(6) ABORTED?
010376	001401			GEQ	.+4	
010400	000000			HLT		;ERROR! INSTRUCTION NOT ABORTED
010402	010701			SCOPE		;SCOPE STORES PC IN R1
;TEST THAT ADDRESS MODE 3 USING R6 DOES NOT OVERFLOW IF ADDRESS IS 'YELLOW'						
010404	012737	010430	000004	MOV	#IND14,2#ERRVEC	
010412	012706	000776		MOV	#776,%6	
010416	012767	000376	170352	MOV	#376,776	
010424	005036			CLR	2(6)+	;CLEAR LOCATION 376
010426	000401			BR	.+4	
010430	000000			IND14: HLT		;ERROR! ADDRESS MODE 3 CAUSED OVERFLOW TRAP


```

010432 010701                                SCOPE                                ;SCOPE STORES PC IN R1

;TEST THAT ADDRESS MODE 5 DOES NOT OVERFLOW
010434 012737 010460 000004                MOV      #IND15,2#ERRVEC
010442 012706 001000                        MOV      #1000,%6
010446 012767 000336 170322                MOV      #336,776
010454 005056                                CLR      2-(6)                                ;CLEAR LOCATION 336 (A 'RED' LOCATION)
010456 000401                                BR        .+4
010460 000000                                IND15:    HLT
010462 010701                                SCOPE                                ;ERROR! TRAPPED
                                           ;SCOPE STORES PC IN R1

;TEST OVERFLOW WITH STACK POINTER IN THE 'RED' ZONE'
010464 012706 000340                        MOV      #340,%6                                ;SET UP STACK POINTER IN 'RED' AREA
010470 012737 010540 000030                MOV      #RED1,2#EMTVEC
010476 005037 000032                        CLR      2#EMTVEC+2
010502 012737 010542 000004                MOV      #RED1A,2#ERRVEC ;SET ERROR TRAP VECTOR
010510 005067 167272                        CLR      6
010514 005067 167260                        CLR      0
010520 012767 136336 167610                MOV      #136336,336 ;PRESET 'RED' LOCATION
010526 012767 177777 167246                MOV      #-1,2
010534 104000                                EMT
010536 000000                                RED1B:   HLT
010540 000000                                RED1:   HLT                                ;DID NOT ABORT EMT
010542 005767 167234                                RED1A:  TST      2                                ;WAS STATUS SAVED?
010546 001401                                BEQ      .+4                                ; (2#EMTVEC+2)
010550 000000                                HLT
010552 022767 010540 167220                CMP      #RED1,0 ;WAS RETURN PC SAVED
010560 001401                                BEQ      .+4                                ; (2#EMTVEC)
010562 000000                                HLT
010564 005706                                TST      %6                                ;STACK POINTER=0?
010566 001401                                BEQ      .+4
010570 000000                                HLT
010572 022767 136336 167536                CMP      #136336,336 ;WAS 'RED' LOCATION UNDISTURBED?
010600 001401                                BEQ      .+4
010602 000000                                HLT
010604 013737 000032 000030                MOV      2#EMTVEC+2,2#EMTVEC
010612 010701                                SCOPE                                ;SCOPE STORES PC IN R1

010614 012706 000200                        MOV      #200,%6                                ;SET UP STACK IN 'RED' AREA
010620 012767 176176 167350                MOV      #176176,176 ;PRE SET 'RED' LOCATION
010626 012737 010670 000034                MOV      #RED2,2#TRAPVEC
010634 005037 000036                        CLR      2#TRAPVEC+2
010640 005067 167142                        CLR      6
010644 005067 167130                        CLR      0
010650 012737 010672 000004                MOV      #RED2A,2#ERRVEC
010656 012767 177777 167116                MOV      #-1,2
010664 104400                                TRAP
010666 000000                                RED2B:   HLT                                ;ERROR! FAILED TO TRAP
010670 000000                                RED2:   HLT                                ;DID NOT ABORT TRAP
010672 005767 167104                                RED2A:  TST      2                                ;WAS STATUS SAVED?
010676 001401                                BEQ      .+4                                ; (TRAPVEC+2)
010700 000000                                HLT
010702 022767 010670 167070                CMP      #RED2,0 ; (TRAPVEC)
010710 001401                                BEQ      .+4
010712 000000                                HLT
010714 005706                                TST      %6

```


C04

MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 42
DCKBME.P11

010716	001401			BEQ	.+4	
010720	000000			HLT		
010722	022767	176176	167246	CMP	#176176,176	;WAS 'RED' LOCATION LEFT UNDISTURBED?
010730	001401			BEQ	.+4	
010732	000000			HLT		;ERROR! 'RED' LOCATION WAS CHANGED
010734	012737	000036	000034	MOV	#TRAPVEC+2,2#TRAPVEC	
010742	010701			SCOPE		;SCOPE STORES PC IN R1
010744	012706	000100		MOV	#100,%6	
010750	012737	011012	000020	MOV	#RED3,2#IOTVEC	
010756	005037	000022		CLR	2#IOTVEC+2	
010762	005067	167020		CLR	6	
010766	012737	011014	000004	MOV	#RED3A,2#ERRVEC	
010774	005067	167000		CLR	0	
011000	012767	177777	166774	MOV	#-1,2	
011006	000004			IOT		
011010	000000			RED3B: HLT		;DID NOT OVERFLOW
011012	000000			RED3: HLT		;DID NOT ABORT IOT
011014	005767	166762		RED3A: TST	2	;WAS STATUS SAVED?
011020	001401			BEQ	.+4	; (2#IOTVEC+2)
011022	000000			HLT		
011024	022767	011012	166746	CMP	#RED3,0	;WAS NEW PC SAVED?
011032	001401			BEQ	.+4	; (2#IOTVEC)
011034	000000			HLT		
011036	005706			TST	%6	
011040	001401			BEQ	.+4	
011042	000000			HLT		
011044	013737	000022	000020	MOV	2#IOTVEC+2,2#IOTVEC	
011052	010701			SCOPE		;SCOPE STORES PC IN R1
011054	005006			CLR	%6	
011056	012767	011120	166730	MOV	#RED4,BPTVEC	
011064	005067	166726		CLR	BPTVEC+2	
011070	005067	166712		CLR	6	
011074	012737	011122	000004	MOV	#RED4A,2#ERRVEC	
011102	005067	166672		CLR	0	
011106	012767	177777	166666	MOV	#-1,2	
011114	000003			BPT		;TRACE TRAP
011116	000000			RED4B: HLT		;DID NOT OVERFLOW
011120	000000			RED4: HLT		;DID NOT ABORT BPT
011122	005767	166654		RED4A: TST	2	;WAS NEW STATUS SAVED?
011126	001401			BEQ	.+4	; (BPTVEC+2)
011130	000000			HLT		
011132	022767	011120	166640	CMP	#RED4,0	;WAS NEW PC SAVED?
011140	001401			BEQ	.+4	; (BPTVEC)
011142	000000			HLT		
011144	005706			TST	%6	
011146	001401			BEQ	.+4	
011150	000000			HLT		
011152	012767	000016	166634	MOV	#BPTVEC+2,BPTVEC	
011160	010701			SCOPE		;SCOPE STORES PC IN R1
;TEST TRANSITION FROM 'YELLOW' TO 'RED' AREAS. THE TRANSITION OCCURS						
;AFTER THE EMT HAS PUSHED ITS RETURN ADDRESS AND STATUS.						
011162	012706	000044		MOV	#344,%6	;SET UP STACK TO ALLOW 2 'PUSHES'
011166	012737	011246	000030	MOV	#RED5,2#EMTVEC	;LOAD EMT VECTOR


```

011174 005037 000032          CLR      2#EMTVEC+2      ;AND STATUS
011200 012737 011252 000004  MOV      #RED5A,2#ERRVEC ;LOAD OVERFLOW VECTOR
011206 012767 000001 166572  MOV      #1,6      ;AND STATUS
011214 005067 166562          CLR      2
011220 005067 166554          CLR      0
011224 012767 136336 167104  MOV      #136336,336 ;PRE SET 'RED' LOCATION
011232 012737 000004 177776  MOV      #4,2#PSW
011240 104000          EMT
011242 000000          HLT
011244 000240          NOP
011246 000000          RED5: HLT
011250 000240          NOP
011252 022767 000004 167062 RED5A: CMP      #4,342      ;WAS STATUS SAVED?
011260 001401          BEQ      .+4
011262 000000          HLT
011264 022767 011242 167046  CMP      #RED5B,340 ;ERROR! EMT DID NOT SAVE STATUS
011272 001401          BEQ      .+4      ;WAS RETURN PC SAVED?
011274 000000          HLT
011276 022767 136336 167032  CMP      #136336,336 ;ERROR! RETURN PC NOT SAVED
011304 001401          BEQ      .+4      ;WAS 'RED' LOCATION LEFT UNDISTURBED?
011306 000000          HLT
011310 022767 000001 166464  CMP      #1,2      ;ERROR! 'RED' AREA WAS CHANGED
011316 001401          BEQ      .+4      ;WAS OVERFLOW'S STATUS SAVED?
011320 000000          HLT
011322 022767 011252 166450  CMP      #RED5A,0   ;ERROR! OVERFLOW'S STATUS (6) NOT SAVED
011330 001401          BEQ      .+4      ;WAS OVERFLOW'S VECTOR SAVED?
011332 000000          HLT
011334 005706          TST      %6      ;ERROR! OVERFLOW'S VECTOR (4) NOT SAVED
011336 001401          BEQ      .+4
011340 000000          HLT
011342 010701          SCOPE          ;SCOPE STORES PC IN R1

```

```

011344 013737 000032 000030  MOV      2#EMTVEC+2,2#EMTVEC
011352 005037 000032          CLR      2#EMTVEC+2

```

;TEST THAT AN ODD ADDRESS ERROR USING THE STACK POINTER RESULTS IN A 'RED' TRAP

```

011356 012706 001000          MOV      #1000,%6
011362 012737 011402 000004  MOV      #RED6,2#ERRVEC
011370 005037 000006          CLR      2#ERRVEC+2
011374 005066 177777          CLR      -1(6) ; -1(6) IS ODD
011400 000000          HLT ;ERROR! FAILED TO TRAP
011402 022767 011400 166370 RED6A: CMP      #RED6A,0 ;CHECK RETURN PC IN ADDRESS 0
011410 001401          BEQ      .+4
011412 000000          HLT ;ERROR! 'RED' TRAP DID NOT SAVE PC IN 0
011414 010701          SCOPE ;SCOPE STORES PC IN R1

```

;THIS TEST IS THE SAME AS ABOVE EXCEPT THAT INDEX DEFERRED MODE IS USED.
;NOTE: A 'RED TRAP' SHOULD NOT OCCUR.

```

011416 012706 001000          MOV      #1000,%6
011422 005067 166352          CLR      0
011426 012767 000777 167342  MOV      #777,776
011434 012737 011450 000004  MOV      #RED7,2#ERRVEC
011442 005076 177776          CLR      2-2(6)
011446 000000          HLT RED7A: HLT ;ERROR! FAILED TO TRAP
011450 022767 011446 167316 RED7: CMP      #RED7A,774 ;RETURN PC IN 774?
011456 001401          BEQ      .+4

```



```

011460 000000          HLT          ;ERROR! RED TRAP FAILED
011462 005767 166312   TST          0          ;TEST THAT RED TRAP DID NOT OCCUR
011466 001401          BEQ          .+4
011470 000000          HLT          ;ERROR! 'RED TRAP' ON STACK RELATED REFERENCE
011472 010701          SCOPE        ;SCOPE STORES PC IN R1

;TEST THAT 'A TIME OUT' WHEN THE STACK POINTER IS USED DOES NOT CAUSE A 'RED' TRAP
;IF MODE 5.
011474 012706 001000   MOV          #1000,%6
011500 005067 166274   CLR          0          ;PRE SET ADDRESS 0
011504 012767 177700 167264   MOV          #177700,776 ;776 CONTAINS ADDRESS OF R0
011512 012737 011524 000004   MOV          #RED10,3#ERRVEC
011520 005056          CLR          3-(6) ;R0 IS NOT A BUS ADDRESS
011522 000000          HLT          ;ERROR! FAILED TO TRAP
011524 005767 166250   RED10A: TST          0          ;WAS ADDRESS 0 UNCHANGED?
011530 001401          BEQ          .+4
011532 000000          HLT          ;ERROR! A 'RED' TRAP OCCURRED
011534 010701          SCOPE        ;SCOPE STORES PC IN R1

;TEST THAT A 'TIME OUT' CAN CAUSE A 'RED' TRAP IF MODE 6.
011536 012706 001000   MOV          #1000,%6
011542 005067 166232   CLR          0          ;PRE SET ADDRESS 0
011546 012737 011562 000004   MOV          #RED11,3#ERRVEC
011554 005066 176700   CLR          176700(6) ;FINAL ADDRESS IS 177700 (R0)
011560 000000          HLT          ;ERROR! NO OVERFLOW TRAP
011562 022767 011560 166210 RED11A: CMP          #RED11A,0 ;WAS 'RED' TRAP TAKEN?
011570 001401          BEQ          .+4
011572 000000          HLT          ;ERROR! NO 'RED' TRAP
011574 010701          SCOPE        ;SCOPE STORES PC IN R1

;CHECK 'T' BIT TRAP IN RED ZONE
011576 012737 011642 000004   MOV          #RED12B,3#ERRVEC ;LOAD ERROR (OVERFLOW) TRAP VECTOR
011604 012737 011634 000014   MOV          #RED12C,3#TBITVEC ;LOAD 'T' BIT TRAP VECTOR
011612 012706 000334   MOV          #334,%6 ;SET STACK IN RED ZONE
011616 010605          MOV          %6,%5
011620 012725 011636   MOV          #RED12A,(5)+ ;SET UP TO SET 'T' BIT
011624 012725 000020   MOV          #20,(5)+
011630 000006          RTT          ;SET 'T' BIT & GO TO RED12A
011632 000000          HLT          ;ERROR! TRAP ON RTT
011634 000000          RED12C: HLT          ;ERROR! NO RED OVERFLOW BEFORE 'T' TRAP
011636 005046          RED12A: CLR          -(6) ;RESULTS IN 'RED' OVERFLOW TRAP
011640 000000          HLT          ;ERROR! NO TRAP
011642 022767 011640 166130 RED12B: CMP          #RED12A+2,0 ;CHECK 'T' BIT TRAP VECTOR ON STACK
011650 001401          BEQ          .+4
011652 000000          HLT          ;ERROR!
011654 022767 000020 166120   CMP          #20,2 ;CHECK 'T' BIT ON STACK
011662 001401          BEQ          .+4
011664 000000          HLT          ;ERROR!
011666 012737 000016 000014   MOV          #TBITVEC+2,3#TBITVEC ;SCOPE STORES PC IN R1
011674 010701          SCOPE

;TEST TRANSITION FROM 'YELLOW' TO 'RED' ZONES. THE TRANSITION OCCURS AFTER
;THE JSR HAS 'PUSHED' ITS OLD R5.
011676 012706 000342   MOV          #342,%6 ;SET UP STACK ON THE 'HAIRY EDGE'
011702 012737 011764 000004   MOV          #HAIR1,3#ERRVEC ;LOAD OVERFLOW VECTOR

```


011710	012737	000357	000006	MOV	#357, @#ERRVEC+2	; AND OVERFLOW STATUS
011716	005037	000000		CLR	@#0	
011722	005037	000002		CLR	@#2	
011726	005037	000336		CLR	@#336	
011732	005037	000340		CLR	@#340	
011736	012705	000007		MOV	#7, %5	; PRE SET R5
011742	012737	000017	177776	MOV	#17, @#PSW	; PRE SET THE STATUS WORD
011750	004567	000004		JSR	5, HAIR	; THIS SHOULD RAISE SOME HAIR
011754	000000		HAIR1B:	HLT		; ERROR! DID NOT TRAP
011756	000240			NOP		
011760	000000		HAIR:	HLT		; ERROR! DID NOT TRAP
011762	000240			NOP		
011764	022737	000007	000340	HAIR1: CMP	#7, @#340	; WAS R5 SAVED ON THE STACK
011772	001401			BEQ	.+4	
011774	000000			HLT		; ERROR! R5 NOT SAVED ON THE STACK
011776	022705	011754		CMP	#HAIR1B, %5	; DOES R5 CONTAIN RETURN ADDRESS?
012002	001401			BEQ	.+4	
012004	000000			HLT		; ERROR! R5 DID NOT GET RETURN ADRS.
012006	022737	000357	000002	CMP	#357, @#2	; DID OVERFLOW SAVE THE STATUS?
012014	001401			BEQ	.+4	
012016	000000			HLT		; OVERFLOW DID NOT SAVE STATUS
012020	022737	011764	000000	CMP	#HAIR1, @#0	; WAS RETURN ADDRESS SAVED ON THE STACK
012026	001401			BEQ	.+4	
012030	000000			HLT		; OVERFLOW TRAP DID NOT SAVE RETURN PC
012032	005737	000336		TST	@#336	; CHECK THAT 'RED' LOCATION WAS NOT CHANGED
012036	001401			BEQ	.+4	
012040	000000			HLT		; ERROR! 'RED' LOCATION WAS CHANGED
012042	010701			SCOPE		; SCOPE STORES PC IN R1

; THIS TEST IS THE SAME AS ABOVE EXCEPT THAT THE TRANSITION IS AFTER THE
; OVERFLOW PUSHES ONE WORD.

012044	012706	000344		MOV	#344, %6	
012050	012737	012116	000004	MOV	#HAIR2, @#ERRVEC	; SET ERROR TRAP VECTOR
012056	005037	000006		CLR	@#6	
012062	005037	000342		CLR	@#342	
012066	012737	136336	000336	MOV	#136336, @#336	; PRE SET 'RED' LOCATION
012074	012705	000007		MOV	#7, %5	; PRE SET R5
012100	012737	000357	177776	MOV	#357, @#PSW	; PRE SET STATUS
012106	004567	000002		JSR	5, HAIR2A	; HERE WE GO AGAIN
012112	000000		HAIR2B:	HLT		; ERROR! DID NOT TRAP
012114	000000		HAIR2A:	HLT		; ERROR! DID NOT TRAP
012116	022737	000007	000342	HAIR2: CMP	#7, @#342	; WAS R5 SAVED
012124	001401			BEQ	.+4	
012126	000000			HLT		; ERROR! R5 NOT SAVED ON THE STACK
012130	022705	012112		CMP	#HAIR2B, %5	; DOES R5 CONTAIN RETURN PC?
012134	001401			BEQ	.+4	
012136	000000			HLT		; ERROR! JSR DID NOT LOAD R5
012140	022737	000357	000340	CMP	#357, @#340	; WAS STATUS SAVED?
012146	001401			BEQ	.+4	
012150	000000			HLT		; ERROR! STATUS NOT SAVED
012152	005737	000002		TST	@#2	; WAS STATUS SAVED?
012156	001401			BEQ	.+4	
012160	000000			HLT		; ERROR! STATUS NOT SAVED
012162	022737	012116	000000	CMP	#HAIR2, @#0	; RETURN PC SAVED?
012170	001401			BEQ	.+4	
012172	000000			HLT		; ERROR! RETURN PC NOT SAVED


```

012174 022737 136336 000336      CMP      #136336,0#336      ;WAS 'RED' LOCATION UNDISTURBED?
012202 001401      BEQ      .+4
012204 000000      HLT
012206 010701      SCOPE      ;ERROR! 'RED' LOCATION WAS CHANGED
                                ;SCOPE STORES PC IN R1

                                ;TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
012210 012737 000340 177776      MOV      #340,0#PSW      ;LOCK OUT INTERRUPT
012216 012706 000400      MOV      #400,%6      ;SET UP STACK TO OVERFLOW
012222 012737 012262 000004      MOV      #TDEC7,0#ERRVEC      ;SET UP OVERFLOW TRAP
012230 012737 012260 000064      MOV      #TDEC8,0#TPVEC      ;SET UP INTERRUPT VECTOR
012236 105737 177564      1$:      TSTB      0#TTCSR      ;WAIT FOR TTY TO BE READY
012242 100375      BPL      1$
012244 012737 000100 177564      MOV      #100,0#TTCSR      ;SET INTERRUPT ENABLE
012252 005037 177776      CLR      0#PSW      ;ALLOW INTERRUPT TO OCCUR
012256 000000      HLT      ;NO INTERRUPT OCCURRED
012260 000000      TDEC8:      HLT      ;TRAP FLAG OVERFLOW DID NOT OCCUR
012262 005037 177564      TDEC7:      CLR      0#TTCSR      ;CLEAR INTERRUPT ENABLE
012266 012737 000066 000064      MOV      #TPVEC+2,0#TPVEC
012274 012737 000006 000004      MOV      #ERRVEC+2,0#ERRVEC
012302 010701      SCOPE      ;SCOPE STORES PC IN R1

                                ;DOES THE PROCESSOR TRAP WHEN %7 IS ODD?
012304 012706 000500      R7TRX:      MOV      #STKPTR,%6      ;SET UP STACK POINTER
012310 012737 012326 000004      MOV      #R7TR1,0#ERRVEC      ;RETURN FROM TRAP
012316 012707 000001      MOV      #1,%7      ;PC EQUALS ONE
012322 000000      HLT
012324 000000      HLT
012326 022767 000003 166140      R7TR1:      CMP      #3,STKPTR-4
012334 001401      BEQ      .+4      ;CORRECT PC WAS NOT SAVED ON STACK
012336 000000      HLT
012340 010701      SCOPE      ;SCOPE STORES PC IN R1

012342 012706 000500      MOV      #STKPTR,%6      ;STACK POINTER
012346 012737 012362 000004      MOV      #R7TR2,0#ERRVEC      ;PC BECOMES ODD
012354 005207      INC      %7
012356 000000      R7TR2A:      HLT
012360 000000      HLT
012362 022767 012361 166104      R7TR2:      CMP      #R7TR2A+3,STKPTR-4
012370 001401      BEQ      .+4      ;CORRECT PC NOT ON STACK
012372 000000      HLT
012374 010701      SCOPE      ;SCOPE STORES PC IN R1

012376 012706 000500      MOV      #STKPTR,%6
012402 012737 012414 000004      MOV      #R7TR3,0#ERRVEC
012410 005307      DEC      %7      ;MAKE PC ODD
012412 000000      HLT      ;SHOULD TRAP
012414 022767 012413 166052      R7TR3:      CMP      #-1,STKPTR-4      ;CHECK VALUE OF PC ON STACK
012422 001401      BEQ      .+4
012424 000000      HLT      ;WRONG VALUE ON STACK
012426 010701      SCOPE      ;SCOPE STORES PC IN R1

012430 012706 000500      MOV      #STKPTR,%6
012434 012737 012452 000004      MOV      #R7TR4,0#ERRVEC
012442 000261      SEC
012444 006107      ROL      %7      ;CARRY EQUALS A 1
                                ;PC BECOMES ODD

```


H04

MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 47
DCKBME.P11

012446	000000		TR4A:	HLT	
012450	000000			HLT	
012452	022767	025117 166014	R7TR4:	CMP	#TR4A+TR4A+3,STKPTR-4 ;CHECK FOR VALUE ON STACK
012460	001401			BEG	.+4
012462	000000			HLT	;WRONG VALUE ON STACK
012464	012737	000006 000004		MOV	#ERRVEC+2,0#ERRVEC
012472	010701			SCOPE	;SCOPE STORES PC IN R1


```

;TEST THAT THE 'T' BIT WILL CAUSE A TRAP
012474 012706 000500      MOV      #STKPTR,%6      ;SET UP STACK POINTER
012500 012737 012530 000014  MOV      #TRC1,%7      ;TRACE TRAP RETURN
012506 005037 000016      CLR      %7      ;
012512 012746 000020      MOV      #20,-(6)      ;PUSH 'T' BIT ON THE STACK
012516 012746 012524      MOV      #.+6,-(6)      ;PUSH PC ON THE STACK
012522 000006      RTT      ;SET 'T' BIT
012524 000240      NOP      ;'T' TRAP ACKNOWLEDGED AFTER NOP
012526 000000      HLT      ;ERROR! 'T' TRAP FAILED TO TRAP
012530 036727 165742 000020 TRC1:  BIT      STKPTR-2,%20 ;CHECK FOR T BIT ON STACK
012536 001001      BNE      .+4
012540 000000      HLT      ;T BIT NOT SAVED ON STACKED
012542 010701      SCOPE   ;SCOPE STORES PC IN R1

;TEST THAT AN RTI POPS THE T BIT
012544 012706 000500      MOV      #STKPTR,%6      ;SET UP THE STACK
012550 012746 000020      MOV      #20,-(6)      ;FUTURE T BIT ON STACK
012554 012746 012570      MOV      #TRC2,%7      ;RTI RETURN
012560 012737 012574 000014  MOV      #TRC3,%7      ;TRACE TRAP INTERRUPT POINTER
012566 000002      RTI      ;
012570 000000      HLT      ;TRACE IS SET SHOULD TRAP TO 14
012572 000000      HLT      ;DID NOT TRACE TRAP
012574 010701      TRC3:  SCOPE   ;SCOPE STORES PC IN R1

;TEST RTT POPS TBIT
012576 012706 000500      MOV      #STKPTR,%6      ;
012602 012746 000020      MOV      #20,-(6)      ;
012606 012746 012622      MOV      #TRC4,%7      ;
012612 012737 012634 000014  MOV      #TRC5,%7      ;
012620 000006      RTT      ;
012622 000240      TRC4:  NOP      ;
012624 000000      HLT      ;ERROR! 'T' BIT NOT SET
012626 012737 000016 000014  MOV      #TBITVEC+2,%7 ;SCOPE STORES PC IN R1
012634 010701      TRC5:  SCOPE   ;SCOPE STORES PC IN R1

;TEST THAT INTERRUPT OCCURS BEFORE TRAP
012636 012706 000500      MOV      #STKPTR,%6      ;
012642 012737 000340 177776  MOV      #340,%7      ;HIGHEST PRIORITY LEVEL
012650 012737 000100 177564  MOV      #100,%7      ;INTERRUPT FOR TTY PUNCH/PRINTER
012656 012737 012706 000034  MOV      #TR1,%7      ;TRAP VECTOR
012664 012737 012710 000064  MOV      #TR2,%7      ;TTY VECTOR
012672 012737 000340 000036  MOV      #340,%7      ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
012700 005037 177776      CLR      %7      ;SHOULD TRAP AT END OF CLR INST
012704 104400      TRAP      ;TTY INTERRUPT SHOULD OVERRIDE TRAP
012706 000000      TR1:  HLT      ;TRAP OCCUR FIRST
012710 005037 177564      TR2:  CLR      %7
012714 010701      SCOPE   ;SCOPE STORES PC IN R1

;WILL INTERRUPTS OCCURE BETWEEN TRAPS
012716 012706 000500      MOV      #STKPTR,%6      ;
012722 012737 000340 177776  MOV      #340,%7      ;
012730 012737 000100 177564  MOV      #100,%7      ;
012736 012737 012774 000034  MOV      #TR3,%7      ;TRAP
012744 005037 000036      CLR      %7
012750 012737 013000 000064  MOV      #TR4,%7      ;TTY OUTPUT

```



```

012756 012737 012776 000020      MOV      #TR5, @#IOTVEC      ; IOT
012764 012737 000340 000022      MOV      #340, @#IOTVEC+2    ; IOT PRIORITY
012772 104400                      TRAP                      ; THE ACT OF TRAPPING LOWERS PRIORITY
012774 000004                      TR3: IOT                      ; INTERRUPT SHOULD OCCURE INPLACE OF IOT TRAP
012776 000000                      TR5: HLT                      ; NO INTERRUPT BETWEEN TRAPS
013000 005037 177564              TR4: CLR      @#TTCSR
013004 012737 000036 000034      MOV      #TRAPVEC+2, @#TRAPVEC
013012 012737 000022 000020      MOV      #IOTVEC+2, @#IOTVEC
013020 005037 000022              CLR      @#IOTVEC+2      ; CLR IOT PRIORITY
013024 010701                      SCOPE                      ; SCOPE STORES PC IN R1

;TEST THAT 'T' BIT TRAP OCCURS AFTER AN INTERRUPT IS ACKNOWLEDGED.
013026 012737 000340 177776      MOV      #340, @#PSW      ; SET PRIORITY =7
013034 012737 000100 177564      MOV      #100, @#TTCSR    ; ENABLE INTERRUPT ON TTY PRINTER
013042 012737 013074 000014      MOV      #INT, @#TBITVEC  ; LOAD 'T' BIT VECTOR
013050 012737 013076 000064      MOV      #INT+2, @#TPVEC    ; LOAD TTY INT. VECTOR
013056 012706 000500              MOV      #STKPTR, %6      ; SET STACK POINTER
013062 012746 000020              MOV      #20, -(6)        ; PUSH 'T' BIT ON THE STACK
013066 012746 013074              MOV      #.+6, -(6)       ; PUSH PC ON THE STACK
013072 000006                      RTT                      ; SET 'T' BIT
013074 000000                      INT: HLT                      ; ERROR! TTY FAILED TO INTERRUPT BEFORE 'T' TRAP
013076 005037 177564              CLR      @#TTCSR          ; DISABLE TTY INTERRUPT
013102 012737 000016 000014      MOV      #TBITVEC+2, @#TBITVEC
013110 010701                      SCOPE                      ; SCOPE STORES PC IN R1

;TEST THAT "RESET" GOES TO OUTSIDE WORLD
013112 010701                      SCOPE                      ; SCOPE STORES PC IN R1
013114 012706 000500              MOV      #STKPTR, %6
013120 012737 000340 177776      MOV      #340, @#PSW      ; LOCK OUT INTERRUPTS
013126 012737 013160 000064      MOV      #TR6, @#TPVEC    ; LOAD TELEPRINTER VECTOR
013134 012737 000100 177564      MOV      #100, @#TTCSR    ; SET INTERRUPT ENABLE
013142 000005                      RESET                     ; SHOULD CLEAR INTERRUPT ENABLE
013144 032737 000100 177564      BIT      #100, @#TTCSR    ; TEST FOR CLEAR
013152 001401                      BEQ      .+4
013154 000000                      HLT
013156 000402                      BR      TR6X              ; RESET FAILED TO CLEAR @#TTCSR
013160 000000                      TR6: HLT                  ; GO TO SCOPE ; SCOPE STORES PC IN R1
; ERROR! TELETYPE INTERRUPTED WHEN
; PROCESSOR WAS AT LEVEL 7

013162 022626                      TR6X: CMP      (6)+, (6)+
013164 010701                      SCOPE                      ; SCOPE STORES PC IN R1

;TEST THAT RESET DOES NOT HANG THE SYSTEM
013166 012706 000500              MOV      #STKPTR, %6      ; SET STACK
013172 005037 177776              CLR      @#PSW          ; ALLOW INTERRUPT
013176 012737 013212 000064      MOV      #RESET1, @#TPVEC  ; TTY INTERRUPT VECTOR
013204 052737 000100 177564      BIS      #100, @#TTCSR    ; SET INTERRUPT ENABLE
013212 000005                      RESET1: RESET           ; IF THIS HANGS CHECK SACK
013214 012737 000066 000064      MOV      #66, @#TPVEC      ; FOR FALSE INTERRUPT
013222 010701                      SCOPE                      ; SCOPE STORES PC IN R1

;TEST RESET WITH TRACE ON
013224 012706 000500              MOV      #STKPTR, %6      ; SET STACK
013230 012737 013262 000014      MOV      #RESET2, @#TBITVEC  ; SET UP TRACE VECTOR
013236 005037 000016              CLR      @#TBITVEC+2
013242 012746 000020              MOV      #20, -(6)        ; PUSH 'T' BIT ON THE STACK
013246 012746 013254              MOV      #.+6, -(6)       ; PUSH PC ON THE STACK

```



```

013252 000006          RIT
013254 000005          RESET
013256 000005          RESET
013260 000000          HLT
013262 012737 000016 000014 RESET2: MOV #TBITVEC+2,2#TBITVEC
013270 010701          SCOPE

;TEST THAT WHEN TTY INTERRUPTS IT POPS NEW STATUS
013272 000005          RESET
013274 012706 000500    MOV #STKPTR,%6          ;SET UP STACK
013300 012737 013324 000064 MOV #TTY3,2#TPVEC      ;INTERRUPT VECTOR
013306 005037 177776    CLR 2#PSW              ;DROP PROCESSOR PRIORITY
013312 012767 000357 164546 MOV #357,66        ;HIGH PRIORITY ON INTERRUPT
013320 005137 177564    COM 2#TTCSR             ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
013324 013727 177776    TTY3: MOV 2#PSW,(7)+    ;SAVE PROCESSOR STATUS
013330 000000          .WORD 0                  ;IS SAVED HERE
013332 022767 000357 177770 CMP #357,-2
013340 001401          BEQ .+4
013342 000000          HLT
013344 000005          RESET
013346 010701          SCOPE

;INTERRUPT DID NOT POP CORRECT STATUS
;CLR INTERRUPT ENABLE
;SCOPE STORES PC IN R1

013350 012706 000500    MOV #STKPTR,%6          ;STACK SET UP
013354 012737 013400 000064 MOV #TTY4,2#TPVEC      ;INTERRUPT VECTOR
013362 005067 164500    CLR 66                  ;CLR NEW STATUS
013366 012737 000157 177776 MOV #157,2#PSW        ;PROCESSOR STATUS
013374 005137 177564    COM 2#TTCSR             ;SET INTERRUPT ENABLE
013400 013727 177776    TTY4: MOV 2#PSW,(7)+    ;SAVE NEW STATUS
013404 000000          .WORD 0                  ;IS SAVED HERE
013406 005767 177772    TST .-2
013412 001401          BEQ .+4
013414 000000          HLT
013416 005037 177564    CLR 2#TTCSR
013422 010701          SCOPE

;INTERRUPT DID NOT POP NEW STATUS
;SCOPE STORES PC IN R1

;TEST THAT A PROGRAMMED INTERRUPT REQUEST IS HONORED BEFORE A BUS
;REQUEST AT THE SAME INTERRUPT LEVEL
013424 000237          SPL 7                    ;SET PRIORITY LEVEL 7
013426 000005          RESET
013430 012737 013506 000240 MOV #PIRINT,2#PIRVEC    ;SET PIRQ INTERRUPT VECTOR
013436 012737 000340 000242 MOV #340,2#PIRVEC+2    ;ASSUME PRIORITY LEVEL 7 ON INT
013444 012737 013532 000064 MOV #TTY5,2#TPVEC      ;SET TELEPRINTER INT VECTOR
013452 012737 000340 000066 MOV #340,2#TPVEC+2    ;PRIORITY LEVEL 7 ON TTY INT
013460 005137 177564    COM 2#TTCSR             ;ENABLE INTERRUPT
013464 005067 002426    CLR TEMP
013470 012737 010000 177772 MOV #PIR4,2#PIRQ
013476 005037 177776    CLR 2#PSW
013502 000000          HLT
013504 000000          HLT
013506 012737 013542 000064 PIRINT: MOV #TTY6,2#TPVEC
013514 005267 002376    INC TEMP
013520 005037 177772    CLR 2#PIRQ
013524 062716 000002    ADD #2,(6)
013530 000002          RTI
013532 000000          HLT
013534 005037 177772    CLR 2#PIRQ

;BOOK REQUEST AT LEVEL 4
;ALLOW INTERRUPTS
;ERROR! NO INTERRUPT
;ERROR! TTY DID NOT INTERRUPT AFTER PIRQ
;RESET TTY INTERRUPT VECTOR
;SET INDICATOR
;DISABLE REQUEST
;ADJUST RETURN PC TO SECOND HLT
;GO HONOR TTY INTERRUPT
;ERROR! DID NOT HONOR PIR
;DISABLE REQUEST

```



```

013540 000406
013542 005367 002350
013546 005767 002344
013552 001401
013554 000000
013556 000005
013560 012737 000242 000240
013566 012706 000500
013572 010701

TTY6: BR
DEC
TST
BEQ
HLT
RESET
MOV #PIRVEC+2, %PIRVEC
MOV #STKPTR, %6
SCOPE

TTYEX:
;EXIT TEST
;ERROR! BOTH INTERRUPTS WERE NOT HONORED
;RESTORE STACK POINTER
;SCOPE STORES PC IN R1

;CHECK THAT 'T' BIT DOES NOT TRAP OUT OF WAIT
013574 022767 000010 165176
013602 003457
013604 000230
013606 012706 000500
013612 012737 013650 000064
013620 012737 013644 000014
013626 005002
013630 052737 000100 177564

CMP #8, ICNT
BLE TTYEND
SPL 0
MOV #STKPTR, %6
MOV #TTY7A, %TPVEC
MOV #TTY7B, %BPTVEC
CLR %2
BIS #100, %TTCSR

WAITI: WAIT
INC %2
HLT
TTY7B: HLT
TTY7A: BR
MOV #40, %177566
MOV #TTY7C, %TPVEC
MOV #20, 2(6)
RTT

TTY7C: MOV #TTY7D, %BPTVEC
CLR %TTCSR
MOV #15, %177566
RTT

TTY7D: NOP
TTY7EX: MOV #BPTVEC+2, %BPTVEC
MOV #66, %TPVEC
DEC %2
BEQ .+4
HLT
TTYEND: SCOPE

;CHECK THAT PIRQ WILL INTERRUPT OUT OF WAIT
013744 012706 000500
013750 000237
013752 012737 010000 177772
013760 012737 000340 000242
013766 012737 014002 000240
013774 000230
013776 000001

MOV #STKPTR, %6
SPL 7
MOV #PIR4, %PIRQ
MOV #340, %PIRVEC+2
MOV #PIRQA, %PIRVEC
SPL 0
WAITI: WAIT
;
PIRQA: CLR %PIRQ

;SET STACK PTR
;SET PRIORITY LEVEL 7
;BOOK REQUEST AT LEVEL 4
;LEVEL 7 ON INTERRUPT
;GO TO PIRQ ON INTERRUPT
;SET PRIORITY LEVEL 0
;PIRQ SHOULD INTERRUPT OUT OF WAIT
;ERROR! DID NOT DO INC INST AFTER INTERRUPT
;SCOPE STORES PC IN R1
;RETURN TO INST FOLLOWING WAIT WITH 'T'
;BIT SET
;RESTORE VECTORS TO HALT AT
;VECTOR +2
;CHECK INDICATOR
;CAUTION PROGRAM COULD HANG HERE IF TEST FAILS
;ERROR!
;CLEAR REQUEST

```


Address	Hex	Dec	Label	Instruction	Comment
014006	022716	014000		CMP	#WAIT1+2,(6) ;CHECK RETURN PC ON STACK
014012	001404			BEQ	1\$
014014	105737	177564		TSTB	@#TTCSR
014020	100375			BPL	.-4
014022	000000			HLT	;ERROR! INCORRECT RETURN PC ON STACK AFTER ;PIRQ INTERRUPT
014024	012737	000242	000242	1\$: MOV	#PIRVEC+2,@#PIRVEC+2
014032	005037	000242		CLR	@#PIRVEC+2
014036	010701			SCOPE	;SCOPE STORES PC IN R1
;CONTIGUOUS MEMORY ADDRESS TEST					
;THIS TEST CHECKS THAT ALL MEMORY (UP TO 28K) IS CONTIGUOUS					
014040	012706	000500		MOV	#STKPTR,%6 ;SET STACK PTR
014044	012737	014060	000004	MOV	#MEMEND,@#ERRVEC ;SET TIME OUT TRAP VECTOR
014052	005000			CLR	%0 ;SET STARTING ADDRESS FOR TEST
014054	005720			TST	(0)+ ;BEGIN
014056	000776			BR	.-2 ;LOOP UNTIL TIMES OUT
014060	022626		MEMEND:	CMP	(6)+,(6)+ ;RESTORE STACK PTR
014062	022700	160002		CMP	#160002,%0 ;AT END OF 28K MEMORY
014066	001402			BEQ	MEMEX ;EXIT TEST IF AT END OF 28K
014070	005720			TST	(0)+ ;SHOULD TIME OUT
014072	000000			HALT	;ERROR! FAILED TO TIME OUT
014074	012737	000006	000004	MEMEX: MOV	#ERRVEC+2,@#ERRVEC
014102	005037	000006		CLR	@#ERRVEC+2
014106	010701			SCOPE	;SCOPE STORE PC IN R1
014110	000167	000024		JMP	R6TST
014114	000000		K1:	0	
014116	000000		K2:	0	
014120	000000		K3:	0	
014122	000000		K4:	0	
014124	000000		K5:	0	
014126	000000		K6:	0	
014130	052525		K7:	052525	
014132	052400		K10:	052400	
014134	000000		K11:	0	
014136	000000		K12:	0	
;TEST AUTO INCREMENT AND DECREMENT OF R6 FOR .WORD AND .BYTES					
014140	012706	000500	R6TST:	MOV	#STKPTR,%6 ;SET UP THE STACK
014144	112667	001746		MOVB	(6)+,TEMP ;SIX SHOULD INCREMENT BY TWO
014150	020627	000502		CMP	%6,#STKPTR+2
014154	001401			BEQ	+.4
014156	000000			HLT	;R6 DID NOT AUTO INCREMENT BY TWO
014160	010701			SCOPE	;SCOPE STORES PC IN R1
014162	012706	020000		MOV	#20000,%6
014166	114627	000000		MOVB	-(6),#0 ;SHOULD DECREMENT BY TWO
014172	020627	017776		CMP	%6,#17776
014176	001401			BEQ	+.4
014200	000000			HLT	;R6 DID NOT AUTO DECREMENT BY 2
014202	010701			SCOPE	;SCOPE STORES PC IN R1
014204	012706	000500		MOV	#STKPTR,%6
014210	112626			MOVB	(6)+,(6)+ ;DOUBLE AUTO INCREMENT OF R6
014212	020627	000504		CMP	%6,#STKPTR+4
014216	001401			BEQ	+.4
014220	000000			HLT	;WRONG AUTO INCREMENT OF R6

014222	010701		SCOPE		;SCOPE STORES PC IN R1
014224	012706	000500	MOV	#STKPTR,%6	
014230	005004		CLR	%4	
014232	122624		CMPB	(6)+,(4)+	;TEST INCREMENT OF R6
014234	020627	000502	CMP	%6,#STKPTR+2	
014240	001401		BEQ	.+4	
014242	000000		HLT		;WRONG INCREMENT OF R6
014244	010701		SCOPE		;SCOPE STORES PC IN R1
014246	012706	000500	MOV	#STKPTR,%6	
014252	005004		CLR	%4	
014254	122426		CMPB	(4)+,(6)+	;TEST INCREMENT OF R6
014256	020627	000502	CMP	%6,#STKPTR+2	
014262	001401		BEQ	.+4	
014264	000000		HLT		;WRONG INCREMENT OF R6
014266	010701		SCOPE		;SCOPE STORES PC IN R1
014270	012706	000500	MOV	#STKPTR,%6	
014274	005004		CLR	%4	
014276	122624		CMPB	(6)+,(4)+	;TEST INCREMENT OF R4
014300	020427	000001	CMP	%4,#1	
014304	001401		BEQ	.+4	
014306	000000		HLT		;WRONG INCREMENT OF R4
014310	010701		SCOPE		;SCOPE STORES PC IN R1

014312	012706	000500		MOV	#STKPTR,%6	
014316	005004			CLR	%4	
014320	122446			CMPB	(4)+-(6)	;TEST DECREMENT OF R6
014322	020627	000476		CMP	%6,#STKPTR-2	
014326	001401			BEQ	+.4	
014330	000000			HLT		;WRONG INCREMENT OF R6
014332	010701			SCOPE		;SCOPE STORES PC IN R1
014334	012706	000500		MOV	#STKPTR,%6	
014340	005004			CLR	%4	
014342	122426			CMPB	(4)+(6)+	;TEST INCREMENT OF R4
014344	020427	000001		CMP	%4,#1	
014350	001401			BEQ	+.4	
014352	000000			HLT		;WRONG INCREMENT OF R4
014354	010701			SCOPE		;SCOPE STORES PC IN R1
014356	012706	010000		MOV	#10000,%6	
014362	124527	000000		CMPB	-(6),#0	;TEST DECREMENT OF R6
014366	022706	007776		CMP	#7776,%6	
014372	001401			BEQ	+.4	
014374	000000			HLT		;WRONG DECREMENT OF R6
014376	010701			SCOPE		;SCOPE STORES PC IN R1
014400	012706	000500		MOV	#STKPTR,%6	;INITIAL POSITION OF R6
014404	012767	177777	164064	MOV	#-1,STKPTR-2	;HIGH BYTE WILL BE DATA
014412	012767	025252	177474	MOV	#25252,K1	
014420	114667	177471		MOVB	-(6),K1+1	;MOV TO ODD ADDRESS
014424	022767	177652	177462	CMP	#177652,K1	;DID MOVB WORK CORRECTLY
014432	001401			BEQ	+.4	
014434	000000			HLT		;MOVB -(6), ODD: FAILED
014436	010701			SCOPE		;SCOPE STORES PC IN R1
014440	012706	000500		MOV	#STKPTR,%6	;INITIAL POSITION OF R6
014444	012767	177777	164024	MOV	#-1,STKPTR-2	;HIGH BYTE SOURCE DATA
014452	012767	125252	177434	MOV	#125252,K1	;INITIAL SET UP OF DESTINATION
014460	114667	177430		MOVB	-(6),K1	;HIGH BYTE OF STKPTR-2, TO LOW BYTE OF K1
014464	022767	125377	177422	CMP	#125377,K1	;TEST RESULTS
014472	001401			BEQ	+.4	
014474	000000			HLT		;MOVB -(6), EVEN, FAILED
014476	010701			SCOPE		;SCOPE STORES PC IN R1


```

;TEST TRANSFER OF .BYTE USING R6
014500 012767 123456 177416 MOV #123456,K5
014506 012767 050505 177400 MOV #050505,K1
014514 012705 014114 MOV #K1,%5 ;%5=(050505)K1
014520 012706 014124 MOV #K5,%6 ;%6(123456)K5
014524 112625 MOVB (6)+,(5)+ ;LOW .BYTE OF R6 TO R5
014526 022767 050456 177360 CMP #050456,K1
014534 001401 BEQ .+4
014536 000000 HLT
014540 010701 SCOPE ;FALSE TRANSFER OF .BYTE
;SCOPE STORES PC IN R1

014542 012767 123456 177354 MOV #123456,K5
014550 012767 050505 177336 MOV #050505,K1
014556 012705 014114 MOV #K1,%5 ;%5(050505)K1
014562 012706 014126 MOV #K6,%6 ;%6(123456)K5
014566 114625 MOVB -(6),(5)+ ;LOW .BYTE OF R6 TO R5 (DECREMENT)
014570 026727 177320 050456 CMP K1,#050456
014576 001401 BEQ .+4
014600 000000 HLT
014602 010701 SCOPE ;FALSE R6 .BYTE TRANSFER
;SCOPE STORES PC IN R1

014604 012767 123456 177302 MOV #123456,K1
014612 012767 050505 177304 MOV #050505,K5
014620 012705 014114 MOV #K1,%5 ;(123456)
014624 012706 014124 MOV #K5,%6 ;(050505)
014630 112526 MOVB (5)+,(6)+ ;LOW OF R5 TO LOW OF R6
014632 022767 050456 177264 CMP #050456,K5
014640 001401 BEQ .+4
014642 000000 HLT
014644 010701 SCOPE ;FALSE R6 .BYTE TRANSFER
;SCOPE STORES PC IN R1

014646 012767 123456 177240 MOV #123456,K1
014654 012767 050505 177242 MOV #050505,K5
014662 012705 014115 MOV #K1+1,%5 ;123456
014666 012706 014124 MOV #K5,%6 ;050505
014672 112526 MOVB (5)+,(6)+ ;HIGH OF R5 TO LOW OF R6
014674 026727 177224 050647 CMP K5,#050647
014702 001401 BEQ .+4
014704 000000 HLT
014706 010701 SCOPE ;FALSE R6 .BYTE TRANSFER
;SCOPE STORES PC IN R1

014710 012767 123456 177176 MOV #123456,K1
014716 012767 050505 177200 MOV #050505,K5
014724 012705 014115 MOV #K1+1,%5 ;R5-123456--ODD ADDRESS
014730 012706 014124 MOV #K5,%6 ;R6-050505--EVEN ADDRESS
014734 112625 MOVB (6)+,(5)+ ;LOW OF R6 TO HIGH OF R5
014736 022767 042456 177150 CMP #042456,K1
014744 001401 BEQ .+4
014746 000000 HLT
014750 010701 SCOPE ;FAILED LOW OF 6 TO HIGH OF 5
;SCOPE STORES PC IN R1

```



```

;TEST .BYTE OPERATION WITH SEQUENTIAL ODD .EVEN ADDRESS
014752 126767 177152 177151 CMPB K7,K7+1 ;SAME .WORD LOW TO HIGH
014760 001401 BEQ .+4
014762 000000 HLT
014764 010701 SCOPE ;SHOULD COMPARE LOW TO HIGH
;SCOPE STORES PC IN R1

014766 126767 177137 177134 CMPB K7+1,K7 ;COMPARE ODD TO .EVEN SAME .WORD
014774 001401 BEQ .+4
014776 000000 HLT ;ODD TO .EVEN .BYTE FAILURE
015000 010701 SCOPE ;SCOPE STORES PC IN R1

015002 126767 177125 177120 CMPB K10+1,K7 ;SEQUENTIAL .BYTES
015010 001401 BEQ .+4 ;DIFFERENT .WORDS
015012 000000 HLT ;ODD TO .EVEN FAILED
015014 010701 SCOPE ;SCOPE STORES PC IN R1

015016 126767 177110 177102 CMPB K10,K6
015024 001401 BEQ .+4
015026 000000 HLT ;.EVEN TO EVEN FAILED
015030 010701 SCOPE ;SCOPE STORES PC IN R1

015032 126767 177073 177073 CMPB K7+1,K10+1
015040 001401 BEQ .+4
015042 000000 HLT ;ODD TO ODD FAILED
015044 010701 SCOPE ;SCOPE STORES PC IN R1

015046 126767 177060 177057 CMPB K10,K10+1
015054 001001 BNE .+4
015056 000000 HLT ;LOW TO HIGH IN SAME .WORD FAILED
015060 010701 SCOPE ;SCOPE STORES PC IN R1

015062 126767 177045 177042 CMPB K10+1,K10
015070 001001 BNE .+4
015072 000000 HLT ;HIGH TO LOW IN SAME .WORD FAILED
015074 010701 SCOPE ;SCOPE STORES PC IN R1

015076 126767 177030 177025 CMPB K10,K7+1
015104 001001 BNE .+4
015106 000000 HLT ;.EVEN TO ODD FAILED
015110 010701 SCOPE ;SCOPE STORES PC IN R1

```



```

;TEST THAT MOV B %R MOVES ONLY THE LSH OF THE REGISTER.
015112 012767 177777 177014 MOV #1,K11
015120 012700 000125 MOV #125,%0 ;LOAD R0
015124 110067 177004 MOV B %0,K11
015130 026727 177000 177525 CMP K11,#177525 ;WAS ONLY LSH MOVED?
015136 001401 BEQ .+4
015140 000000 HLT ;ERROR! MOV B %R FAILED
015142 010701 SCOPE ;SCOPE STORES PC IN R1

015144 012700 014134 MOV #K11,%0
015150 010067 176760 MOV %0,K11
015154 110020 MOV B %0,(0)+
015156 022767 014134 176750 CMP #K11,K11
015164 001401 BEQ .+4
015166 000000 HLT
015170 010701 SCOPE ;SCOPE STORES PC IN R1

015172 012706 014134 MOV #K11,%6
015176 010667 176732 MOV %6,K11
015202 110626 MOV B %6,(6)+
015204 026727 176724 014134 CMP K11,#K11
015212 001401 BEQ .+4
015214 000000 HLT ;FAILED MOV B %6,(6)+
015216 010701 SCOPE ;SCOPE STORES PC IN R1

015220 012706 014134 MOV #K11,%6
015224 010626 MOV %6,(6)+
015226 026727 176702 014134 CMP K11,#K11
015234 001401 BEQ .+4
015236 000000 HLT ;FAILED MOV %6,(6)+
015240 010701 SCOPE ;SCOPE STORES PC IN R1

015242 000277 SCC ;SET STATUS
015244 005037 177776 CLR @#PSW ;CLEAR STATUS
015250 103001 BCC .+4
015252 000000 HLT ;C NOT CLEAR
015254 102001 BVC .+4
015256 000000 HLT ;V NOT CLEAR
015260 001001 BNE .+4
015262 000000 HLT ;Z NOT CLEAR
015264 100001 BPL .+4
015266 000000 HLT ;N NOT CLEAR
015270 010701 SCOPE ;SCOPE STORES PC IN R1

015272 000257 CCC ;CLEAR CONDITION CODES
015274 052737 000017 177776 BIS #17,@#PSW ;SET STATUS TO ONES
015302 103401 BCS .+4

015304 000000 HLT ;C NOT SET
015306 102401 BVS .+4
015310 000000 HLT ;V NOT SET
015312 001401 BEQ .+4
015314 000000 HLT ;Z NOT SET
015316 100401 BMI .+4
015320 000000 HLT ;N NOT SET

```


015322	010701		SCOPE		;SCOPE STORES PC IN R1
015324	012706	000500	MOV	#STKPTR,%6	;SET UP STKPTRER POINTER
015330	005067	163146	CLR	STKPTR+2	;SET UP NEW STATUS
015334	012767	015344	MOV	#+10,STKPTR	;SET UP, RETURN
015342	000006		RTT		;RETURN NEXT INSTRUCTION
015344	020627	000504	CMP	%6,#STKPTR+4	;%6 SHOULD BE PLUS 4
015350	001401		BEQ	+.4	
015352	000000		HLT		;%6 NOT INCREMENTED
015354	010701		SCOPE		;SCOPE STORES PC IN R1
015356	012767	000357	MOV	#357,STKPTR+2	;RETURN STATUS ALL ONES
015364	012706	000500	MOV	#STKPTR,%6	;SET UP STKPTRER POINTER
015370	012767	015402	MOV	#+12,STKPTR	;SET UP RETURN POINTER
015376	000257		CCC		;CLEAR CONDITION CODES
015400	000006		RTT		;RETURN
015402	001401		BEQ	+.4	
015404	000000		HLT		;Z NOT SET
015406	102401		BVS	+.4	
015410	000000		HLT		;V NOT SET
015412	100401		BMI	+.4	
015414	000000		HLT		;N NOT SET
015416	103401		BCS	+.4	
015420	000000		HLT		;C NOT SET
015422	010701		SCOPE		;SCOPE STORES PC IN R1
015424	005067	163052	CLR	STKPTR+2	;ZERO TO STATUS
015430	012706	000500	MOV	#STKPTR,%6	
015434	012767	015446	MOV	#+12,STKPTR	
015442	000277		SCC		;SET CONDITION CODES
015444	000006		RTT		;RETURN
015446	001001		BNE	+.4	
015450	000000		HLT		;Z NOT CLEARED
015452	102001		BVC	+.4	
015454	000000		HLT		;V NOT CLEARED
015456	100001		BPL	+.4	
015460	000000		HLT		;N NOT CLEARED
015462	103001		BCC	+.4	
015464	000000		HLT		;C NOT CLEARED
015466	010701		SCOPE		;SCOPE STORES PC IN R1


```

;TEST THAT ALL RESERVED INSTRUCTIONS TRAP
015470 012700 016052      GIN1: MOV #TABLE,TAB      ;TABLE POINTER
015474 012002              MOV (TAB)+,FIRST    ;FIRST OR CURRENT INSTRUCTION
015476 012003              MOV (TAB)+,LAST     ;LAST INSTRUCTION OR GROUP
015500 020267 000376      CMP FIRST,FINISH    ;TESTED ALL
015504 001440              BEQ SINS1           ;GO TO MULT TRAPS TEST IF FIN.
015506 010267 000372      GIN2: MOV FIRST,INST ;SET UP INST
015512 012737 015534 000010 MOV #RET,#RESVEC ;SET UP RETURN FROM TRAP
015520 012706 000500      MOV #STKPTR,%6      ;SET UP STACK POINTER
015524 005037 177776      CLR #PSW           ;CLEAR PRIORITY
015530 000167 000350      JMP INST            ;EXECUTE RESERVED INSTRUCTION

;TRAPPING SHOULD SEND YOU HERE
015534 020627 000474      RET:  CMP %6,#STKPTR-4 ;TEST DECREMENT OF %6
015540 001401              BEQ RET1            ;WRONG DECREMENT
015542 000000              HLT                  ;LOC OF INST UNINCREMENTED
015544 026727 162724 016106 RET1: CMP STKPTR-4,#INST+2 ;INST INC ON TRAP
015552 001401              BEQ RET2            ;CONDITION CODES SET ON TRAP
015554 000000              HLT
015556 005767 162714      RET2: TST STKPTR-2
015562 001401              BEQ .+4
015564 000000              HLT
015566 005267 000312      INC INST
015572 005202              INC FIRST
015574 026703 000304      CMP INST,LAST
015600 001735              BEQ GIN1
015602 000167 177700      JMP GIN2            ;SET UP NEW GROUP
;FINISH OLD GROUP
;END OF INSTRUCTION GROUP

;THE FOLOWING TESTS ARE WRITTEN TO CHECK MULTIPLE TRAP CONDITIONS
;TO TEST THAT TRAPS ARE PROCESSED PROPERLY.
015606 010701 000012 000010 SINS1: SCOPE      ;SCOPE STORES PC IN R1
015610 012737 000012 000010      MOV #RESVEC+2,#RESVEC
015616 012706 000500 162164      MOV #STKPTR,%6 ;SET UP STACK POINTER
015622 012767 015666 162146      MOV #SINSD,TBITVEC ;LOAD 'T' BIT VECTOR
015630 012767 015700 162146      MOV #SINSE,4 ;LOAD ERROR VECTOR
015636 012737 015673 000030      MOV #SINSF+1,#EMTVEC ;LOAD EMT VECTOR (ODD)
015644 005037 000032      CLR #EMTVEC+2
015650 005000      CLR %0 ;PRE SET R0
015652 012746 000020      MOV #20,-(6) ;SET UP
015656 012746 015664      MOV #SINSDD,-(6) ;TO SET 'T' BIT
015662 000006      RTT ;SET 'T' BIT
015664 104000      SINSDD: EMT ;DO EMT PICK UP ODD PC
015666 000000      SINSDD: HLT ;ERROR! 'T' TRAP OCCURRED BEFORE ERROR
015670 000431      SINSF: BR SINSI ;GO TO SCOPE ;SCOPE STORES PC IN R1
015672 000240      SINSF: NOP ;THE EMT TRIES TO COME HERE
015674 000000      HLT ;ERROR! NO ERROR TRAP ON ODD EMT VECTOR
015676 000426 015714 162562 SINSE: BR SINSI ;GO TO SCOPE ;SCOPE STORES PC IN R1
015700 012767 015714 162562 SINSE: MOV #SINSG,STKPTR-10 ;CORRECT BAD EMT VECTOR ON THE STACK
015706 000006      RTT ;GO TO SINSG (SERVICE EMT)
015710 000000      HLT ;DIDN'T GO
015712 000420 015734 162552 SINSG: BR SINSI ;GO TO SCOPE ;SCOPE STORES PC IN R1
015714 012767 015744 162064 SINSG: MOV #SINSH,STKPTR-4 ;RESTORE RETURN FROM EMT (ON STACK)
015722 012767 015744 162064 SINSG: MOV #SINSJ,TBITVEC ;CHANGE 'T' BIT VECTOR
015730 000006      RTT ;GO TO SINSH

```


015732	000000				HLT			
015734	012700	000001		SINSH:	MOV	#1,%0		;LOAD INDICATOR
015740	000000				HLT			;ERROR! SHOULD HAVE 'T' TRAPPED
015742	000404				BR	SINSH		;GO TO SCOPE
015744	022700	000001		SINSJ:	CMP	#1,%0		;WAS INST. AT SINSH EXECUTED? ;SCOPE STORES PC IN R1
015750	001401				BEQ	SINSH		
015752	000000				HLT			;ERROR! INST AT SINSH WAS NOT EXECUTED
015754	010701			SINSI:	SCOPE			;SCOPE STORES PC IN R1
015756	005267	163016		END:	INC	ICNT		
015762	022767	000100	163010		CMP	#100,ICNT		
015770	001402				BEQ	DONE		
015772	000167	163010			JMP	BEGIN		
015776	012737	000007	177566	DONE:	MOV	#7,2#177566		
016004	105737	177564		2\$:	TSTB	2#177564		
016010	100375				BPL	2\$		
016012	012737	000000	177566		MOV	#0,2#177566		
016020	105737	177564		1\$:	TSTB	2#177564		
016024	100375				BPL	1\$		
016026	013702	000042			MOV	2#42,%2		;GET DETAPE MONITOR RETURN ADDRESS
016032	001405				BEQ	DONE1		;DO NOT RETURN TO MON IF (42)=0
016034	000005				RESET			
016036	004712			LOGICAL:	JSR	7,(2)		;GO TO MONITOR
016040	000240				NOP			;ACT11
016042	000240				NOP			;OVERLAY
016044	000240				NOP			;AREA
016046	000167	162730		DONE1:	JMP	START		;RESTART TEST
016052	000007			TABLE:	7			;START OF RESVD INTRUCTION TABLE
016054	000077				77			
016056	000210				210			;RTS,RT1,JMP
016060	000227				227			
016062	007000				7000			
016064	007777				7777			
016066	075000				75000			
016070	076777				76777			

016072 106400
016074 106477
016076 106700
016100 107777
016102 016102
016104 000000
016106 000000
016110 000000
016112 000000
016114 000000
016116 000000
016122
000001

106400
106477
106700
107777
FINISH:
INST: HALT
HALT
HALT
HALT
HALT
TEMP: 0
=.+2
.END

;END FLAG
;WILL CONTAIN RESERVED INST
;SHOULD TRAP TO LOC 10
;LOC 10 SHOULD SEND YOU TO
;RET

BEGIN	001006	452*	2801															
BPTA	007260	1642	1643*	1645*	1650													
BPTAA	007264	1646*	1676															
BPTB	007266	1640	1647*															
BPTC	007342	1638	1657	1663*														
BPTD	007376	1667	1673*															
BPTE	007452	1668	1683	1689*														
BPTEx	007472	1653	1660	1679	1686	1695*												
BPTF	007460	1646	1692*															
BPTVEC=	000014	158*	773*	774*	781*	790*	800*	809*	820*	821*	831*	832*	842*	843*				
		1172*	1173*	1182*	1193*	1207*	1220*	1221*	1390*	1401*	1565*	1566*	1638*	1639*				
		1695*	1944*	1945*	1962*	2384*	2399*	2406*										
DISPLA=	177570	172*	453*															
DONE	015776	2800	2803*															
DONE1	016046	2810	2816*															
EMTVEC=	000030	162*	691*	692*	699*	708*	717*	726*	737*	738*	747*	748*	768*	769*				
		1358*	1369*	1559*	1560*	1870*	1871*	1892*	1968*	1969*	2001*	2002*	2770*	2771*				
END	015756	2798*																
ERRVEC=	000004	156*	991*	992*	999*	1008*	1017*	1026*	1037*	1038*	1048*	1049*	1134*	1150*				
		1158*	1159*	1246*	1247*	1256*	1264*	1272*	1280*	1290*	1327*	1343*	1359*	1375*				
		1391*	1407*	1443*	1453*	1463*	1473*	1483*	1493*	1514*	1567*	1568*	1640*	1667*				
		1696*	1702*	1710*	1734*	1756*	1783*	1792*	1801*	1810*	1818*	1827*	1839*	1851*				
		1860*	1872*	1901*	1925*	1947*	1970*	2006*	2007*	2020*	2036*	2047*	2056*	2080*				
		2081*	2113*	2145*	2155*	2161*	2171*	2181*	2190*	2198*	2436*	2445*	2446*					
FINISH	016102	2738	2831*															
FIRST =	%000002	151*	2736*	2738	2740	2757*												
FOVER	006632	1521	1535*															
FP	005266	1225*																
FPA	005322	1226	1235*															
FPAA	005302	1228*	1235															
FPB	005334	1234	1238*															
FPEVEC=	000244	166*																
GIN1	015474	2736*	2759															
GIN2	015506	2740*	2760															
HAIR	011760	2088	2091*															
HAIR1	011764	2080	2093*	2102														
HAIR1B	011754	2089*	2096															
HAIR2	012116	2113	2122*	2134														
HAIR2A	012114	2119	2121*															
HAIR2B	012112	2120*	2125															
HLT =	000000	153*	463	472	481	492	502	513	526	538	547	556	566	576				
		587	599	608	622	631	640	651	661	672	684	694	703	712				
		722	732	743	754	763	776	785	794	805	815	826	838	851				
		860	869	880	890	901	913	922	931	940	951	961	972	984				
		994	1003	1012	1022	1032	1043	1054	1064	1071	1080	1088	1095	1104				
		1117	1120	1122	1129	1137	1140	1143	1146	1152	1155	1167	1177	1188				
		1197	1201	1213	1216	1232	1237	1250	1259	1266	1275	1283	1293	1297				
		1301	1329	1330	1333	1336	1345	1346	1349	1352	1361	1362	1365	1368				
		1377	1378	1381	1384	1393	1394	1397	1400	1409	1410	1413	1416	1445				
		1448	1455	1458	1465	1468	1475	1478	1485	1488	1501	1506	1509	1511				
		1535	1571	1573	1575	1577	1580	1584	1587	1590	1593	1596	1599	1602				
		1605	1608	1611	1614	1617	1620	1623	1626	1629	1649	1652	1656	1659				
		1675	1678	1682	1685	1694	1705	1715	1719	1722	1725	1728	1738	1741				
		1744	1747	1750	1763	1766	1769	1772	1775	1778	1787	1795	1805	1814				
		1822	1832	1841	1844	1847	1856	1865	1878	1879	1882	1885	1888	1891				
		1904	1905	1908	1911	1914	1917	1929	1930	1933	1936	1939	1951	1952				

		1955	1958	1961	1977	1979	1983	1986	1989	1992	1995	1998	2009	2012
		2022	2025	2028	2038	2041	2049	2052	2063	2064	2066	2069	2072	2089
		2091	2095	2098	2101	2104	2107	2120	2121	2124	2127	2130	2133	2136
		2139	2151	2152	2163	2164	2167	2173	2174	2177	2183	2186	2193	2194
		2197	2209	2212	2221	2222	2232	2245	2260	2276	2290	2292	2315	2330
		2343	2359	2360	2366	2372	2390	2391	2410	2422	2428	2465	2472	2479
		2487	2495	2503	2510	2518	2525	2534	2543	2553	2563	2573	2583	2593
		2599	2604	2609	2614	2619	2624	2629	2634	2644	2652	2660	2667	2673
		2675	2677	2679	2686	2688	2690	2692	2701	2710	2712	2714	2716	2725
		2727	2729	2731	2749	2752	2755	2777	2780	2784	2789	2791	2795	
ICNT	001000	449#	451*	453	2379	2798*	2799							
ILLA	= 004700	179#	1408											
ILLB	= 000100	180#												
INDEX	007510	1701#												
IND1	007534	1702	1706#											
IND10	010240	1810	1814#											
IND11	010272	1818	1822#											
IND12	010330	1827	1832#											
IND12A	010326	1831#												
IND13	010360	1839	1842#											
IND13A	010356	1841#	1842											
IND14	010430	1851	1856#											
IND15	010460	1860	1865#											
IND2	007600	1710	1717#											
IND2A	007574	1715#	1720											
IND2X	007644	1716	1729#											
IND3	007676	1734	1739#											
IND3A	007674	1738#	1742											
IND4	010016	1756	1764#											
IND4A	010014	1763#	1773											
IND5	010124	1783	1787#											
IND6	010152	1792	1795#											
IND7	010204	1801	1805#											
INST	016104	2740*	2744	2750	2756*	2758	2832#							
INSTC	001120	478#												
INSTK	003570	937#	938											
INT	013074	2270	2271	2276#										
IOTVEC=	000020	160#	619*	620*	627*	636*	646*	655*	666*	667*	677*	678*	687*	688*
		1342*	1353*	1563*	1564*	1922*	1923*	1940*	2256*	2257*	2263*	2264*	2558	2561
K1	014114	2450#	2530*	2531*	2532	2539*	2540*	2541	2547*	2548	2551	2557*		
		2566*	2568	2576*	2578	2586*	2588	2591						
		2457#	2607	2612	2617	2622	2627	2632						
		2458#	2639*	2641*	2642	2647	2648*	2650	2655	2656*	2658	2663	2665	
		2459#												
K2	014116	2451#												
K3	014120	2452#												
K4	014122	2453#												
K5	014124	2454#	2546*	2549	2556*	2567*	2569	2571	2577*	2579	2581	2587*	2589	
K6	014126	2455#	2559	2612										
K7	014130	2456#	2597	2602	2607	2617	2632							
LAST	=%000003	150#	2737*	2758										
LOGICA	016036	441	2812#											
MEMEND	014060	2436	2440#											
MEMEX	014074	2442	2445#											
NOTBIT	005046	1162	1167#											
OAERO	004376	1060	1065#											

0AER1	004424	1068	1072#
0AER2	004456	1076	1081#
0AER3	004504	1085	1089#
0AER4	004526	1092	1096#
0AER5	004560	1100	1105#
0AER6	004646	1109	1122#
0AER7	004672	1127	1130#
00	006742	1570#	
00A	006744	1571#	1585
01	006746	1559	1572# 1591
01A	006750	1573#	1597
02	006752	1561	1574# 1603
02A	006754	1575#	1609
03	006756	1563	1576# 1615
03A	006760	1577#	1621
04	006762	1565	1578# 1627
05	007200	1567	1632#
06	007204	1631	1633#
PFVEC =	000024	161#	
PIRINT	013506	2351	2361#
PIRQ =	177772	174#	2357# 2363* 2367* 2416* 2423*
PIRQA	014002	2418	2423#
PIRVEC=	000240	165#	2351# 2352* 2374* 2417* 2418* 2430* 2431*
PIR4 =	010000	176#	2357 2416
PSW =	177776	173#	488* 497* 509* 511 514* 520* 523 562* 571* 583* 585 588*
		594#	596 600* 647* 656* 668* 670 673* 679* 681 718* 727* 739*
		741	749* 751 801* 810* 822* 824 827* 833* 835 839* 876* 885*
		897*	899 902* 908* 910 914* 947* 956* 968* 970 973* 979* 981
		1018*	1027* 1039* 1041 1044* 1051 1055* 1165* 1291* 1499* 1569* 1760* 1975*
		2087*	2118* 2143* 2150* 2238* 2243* 2251* 2268* 2284* 2299* 2323* 2326 2337*
		2339	2358* 2671* 2683* 2743*
		760	764#
RA	002614	605	609#
RA1	001714	759*	762# 764* 765
RB	002610	604*	607# 609* 610
RB1	001710	761#	766
RC	002604	606#	611
RC1	001704	1870	1879# 1883
RED1	010540	1872	1880#
RED1A	010542	1878#	
RED1B	010536	2036	2039#
RED1C	011524	2038#	
RED1D	011522	2047	2050#
RED1E	011562	2049#	2050
RED1F	011560	2060	2065# 2067
RED1G	011636	2056	2067#
RED1H	011642	2057	2064#
RED1I	011634	1897	1905# 1909
RED2	010670	1901	1906#
RED2A	010672	1904#	
RED2B	010666	1922	1930# 1934
RED3	011012	1925	1931#
RED3A	011014	1929#	
RED3B	011010	1944	1952# 1956
RED4	011120	1947	1953#
RED4A	011122	1951#	
RED4B	011116		

[illegible]

[illegible]


```
SWR      = 177570
TAB      =X000000
TABLE    016052
TBITVE= 000014
```

TDEC1	005366
TDEC2	005412
TDEC2A	005434
TDEC3	005450
TDEC4	005502
TDEC5	005544
TDEC5A	005540
TDEC5B	005570
TDEC7	012262
TDEC8	012260
TEMP	016116

TOERO	004720
TOERDA	004716
TOERI	004774
TOER1A	004772
TPVEC =	000064

TRAPVE= 000034

```
TRCSR = 177560
TRC1 012530
TRC2 012570
TRC3 012574
TRC4 012622
TRC5 012634
TR1 012706
TR2 012710
TR3 012774
TR4 013000
TR4A 012446
TR5 012776
TR6 013160
TR6X 013164
TTC SR = 177564

TTYEND 013742
TTYEX 013556
TTY3 013324
```

[illegible]

TTY4	013400
TTY5	013532
TTY6	013542
TTY7A	013650
TTY7B	013644
TTY7C	013674
TTY7D	013716
TTY7EX	013720
UBREAK=	177770
VDC20	006500
VDC20A	006472
VDC20B	006526
VDEC1	005626
VDEC10	006124
VDEC11	006204
VDEC12	006206
VDEC15	006256
VDEC16	006306
VDEC17	006336
VDEC18	006366
VDEC19	006416
VDEC2	005624
VDEC20	006430
VDEC3	005706
VDEC4	005704
VDEC5	005766
VDEC6	005764
VDEC7	006046
VDEC8	006044
VDEC9	006126
WAITI	013636
WAIT1	013776
.	= 016122

2335	2339#											
2353	2366#											
2361	2369#											
2383	2393#											
2384	2391#											
2394	2399#											
2399	2405#											
2392	2406#											
175#	456*											
1493	1504#											
1500#	1504											
1494	1507	1511#										
1327	1331#											
1390	1394#	1395										
1406	1410#	1411										
1407	1411#											
1443	1446#											
1453	1456#											
1463	1466#											
1473	1476#											
1483	1486#											
1326	1330#	1331										
1492#	1502											
1343	1347#											
1342	1346#	1347										
1359	1363#											
1358	1362#	1363										
1375	1379#											
1374	1378#	1379										
1391	1395#											
2388#												
2420#	2424											
181#	182	184	186	188	190	192	194	196	198	200	202	204
206	208	210	212	214	216	218	220	222	224	226	228	230
232	234	236	238	240	242	244	246	248	250	252	254	256
258	260	262	264	266	268	270	272	274	276	278	280	282
284	286	288	290	292	294	296	298	300	302	304	306	308
310	312	314	316	318	320	322	324	326	328	330	332	334
336	338	340	342	344	346	348	350	352	354	356	358	360
362	364	366	368	370	372	374	376	378	380	382	384	386
388	390	392	394	396	398	400	402	404	406	408	410	412
414	416	418	420	422	424	426	428	430	432	434	436	440#
442#	444#	446#	448#	455	471	479	480	491	501	512	525	546
554	555	565	575	586	598	630	638	639	650	660	671	683
702	710	711	721	731	742	753	784	792	793	804	814	825
837	859	867	868	879	889	900	912	930	939	950	960	971
983	1002	1010	1011	1021	1031	1042	1053	1116	1119	1121	1139	1142
1145	1154	1166	1175	1184	1187	1195	1199	1200	1209	1211	1214	1215
1231	1236	1249	1274	1282	1296	1300	1332	1335	1348	1351	1364	1367
1380	1383	1396	1399	1412	1415	1447	1457	1467	1477	1487	1505	1508
1510	1534	1579	1583	1586								

MAINDEC-11-DCKBM-E PDP11/45 TRAFS TEST. MACY11 27(732) 09-APR-76 10:45 PAGE 70
DCKBME.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

MAINDEC-11-DCKBM-E PDP11/45 TRAFS TEST. MACY:1 27(732)
DCKBME.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

09-APR-76 10:45 PAGE 70

[illegible]

E06

MAINDEC-11-DCKBM-E PDP11/45 TRAFS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 72
DCKBME.P11 CROSS REFERENCE TABLE -- MACRO NAMES

SETTBI	143#	1174	1183	1194	1208	2205	2273	2310
VTRP	1304#	1324	1340	1356	1372	1388	1404	
VTRPD	1421#	1441	1451	1461	1471	1481		

ADD	1063	2364													
BCC	2672	2730													
BCS	2684	2715													
BEQ	455	471	480	491	501	512	525	546	555	565	575	586	598	630	639
	650	660	671	683	702	711	721	731	742	753	784	793	804	814	825
	837	859	868	879	889	900	912	930	939	950	960	971	983	1002	1011
	1021	1031	1042	1053	1116	1119	1139	1142	1145	1154	1187	1200	1215	1231	1236
	1296	1300	1332	1335	1348	1351	1364	1367	1380	1383	1396	1399	1412	1415	1447
	1457	1467	1477	1487	1505	1508	1579	1583	1586	1589	1592	1595	1598	1601	1604
	1607	1610	1613	1616	1619	1622	1625	1628	1648	1651	1655	1658	1674	1677	1681
	1684	1693	1718	1721	1724	1727	1740	1743	1746	1749	1765	1768	1771	1774	1777
	1843	1846	1881	1884	1887	1890	1907	1910	1913	1916	1932	1935	1938	1954	1957
	1960	1982	1985	1988	1991	1994	1997	2011	2024	2027	2040	2051	2068	2071	2094
	2097	2100	2103	2106	2123	2126	2129	2132	2135	2138	2166	2176	2185	2196	2289
	2329	2342	2371	2409	2425	2442	2464	2471	2478	2486	2494	2502	2509	2517	2524
	2533	2542	2552	2562	2572	2582	2592	2598	2603	2608	2613	2618	2643	2651	2659
	2666	2689	2700	2709	2739	2748	2751	2754	2759	2794	2800	2810			
BHIS	611	766													
BICB	1070														
BIS	1666	2301	2386	2683											
BISB	1265														
BIT	454	2210	2288												
BITB	1281														
BLE	2380														
BMI	2691	2713													
BNE	2211	2623	2628	2633	2676	2724									
BPL	1497	2148	2427	2678	2728	2805	2808								
BPT	775	782	791	802	811	823	834	1392	1576	1643	1950				
BR	1121	1166	1234	1249	1274	1282	1294	1502	1510	1534	1631	1646	1653	1660	1679
	1686	1716	1786	1794	1804	1813	1821	1831	1855	1864	2291	2368	2392	2439	2778
	2781	2785	2792												
BVC	2674	2726													
BVS	2687	2711													
CCC	2682	2707													
CLR	451	461	488	508	514	520	530	536	562	582	588	594	600	615	620
	647	667	673	679	688	692	718	738	749	769	774	801	821	827	833
	839	843	849	876	896	902	908	914	920	947	967	973	979	986	992
	1018	1038	1044	1055	1061	1078	1093	1102	1110	1159	1163	1173	1221	1227	1247
	1257	1288	1289	1499	1512	1568	1636	1639	1641	1689	1703	1704	1711	1712	1735
	1736	1737	1759	1760	1803	1820	1829	1838	1854	1863	1871	1873	1874	1898	1899
	1900	1923	1924	1926	1943	1945	1946	1948	1969	1972	1973	2002	2007	2008	2018
	2021	2034	2037	2046	2048	2065	2082	2083	2084	2085	2114	2115	2150	2153	2204
	2243	2246	2254	2261	2264	2277	2299	2309	2323	2336	2344	2356	2358	2363	2367
	2385	2401	2423	2431	2437	2446	2483	2491	2499	2506	2514	2671	2696	2719	2743
	2771	2772													
CLRB	1515	1762													
CMP	470	479	524	545	554	574	597	610	629	638	659	682	701	710	730
	752	765	783	792	813	836	858	867	888	911	929	938	959	982	1001
	1010	1030	1052	1118	1138	1141	1153	1186	1199	1214	1230	1235	1295	1298	1331
	1334	1347	1350	1363	1366	1379	1382	1395	1398	1411	1414	1446	1456	1466	1476
	1486	1504	1507	1578	1582	1585	1588	1591	1594	1597	1600	1603	1606	1609	1612
	1615	1618	1621	1624	1627	1647	1650	1657	1673	1676	1683	1692	1717	1720	1723
	1739	1742	1745	1767	1770	1773	1842	1883	1889	1909	1915	1934	1956	1981	1984
	1987	1990	1993	2010	2023	2050	2067	2070	2093	2096	2099	2102	2122	2125	2128
	2134	2137	2165	2175	2184	2195	2294	2328	2379	2424	2440	2441	2463	2470	2477
	2485	2493	2501	2508	2516	2523	2532	2541	2551	2561	2571	2581	2591	2642	2650

CMPB	2658	2665	2699	2738	2747	2750	2758	2793	2799	2515	2522	2597	2602	2607	2612
	500	1094	1115	1144	1273	2484	2492	2500	2507						
COM	2617	2622	2627	2632	2338	2355									
DEC	1645	1830	1840	2325											
EMT	2182	2369	2408												
HALT	693	700	709	719	728	740	750	759	762	1360	1570	1677	1976	2776	
	153	183	185	187	189	191	193	195	197	199	201	203	205	207	209
	211	213	215	217	219	221	223	225	227	229	231	233	235	237	239
	241	243	245	247	249	251	253	255	257	259	261	263	265	267	269
	271	273	275	277	279	281	283	285	287	289	291	293	295	297	299
	301	303	305	307	309	311	313	315	317	319	321	323	325	327	329
	331	333	335	337	339	341	343	345	347	349	351	353	355	357	359
	361	363	365	367	369	371	373	375	377	379	381	383	385	387	389
	391	393	395	397	399	401	403	405	407	409	411	413	415	417	419
	421	423	425	427	429	431	433	435	437	2444	2832	2833	2834	2835	2836
INC	609	764	1292	2172	2362	2389	2756	2757	2798						
IOT	621	628	637	648	657	669	680	1344	1574	1928	2259				
JMP	445	850	857	866	877	886	898	909	1632	2449	2744	2760	2801	2816	
JSR	921	928	937	948	957	969	980	2088	2119	2812					
LDFPS	1228														
MOV	452	453	460	467	468	476	477	486	487	490	495	496	499	506	507
	511	517	518	519	523	529	534	535	542	543	551	552	560	561	564
	569	570	571	573	580	581	583	585	591	592	593	596	604	605	606
	614	618	619	626	627	635	636	645	646	649	654	655	656	658	665
	666	668	670	676	677	678	681	687	690	691	698	699	707	708	716
	717	720	725	726	727	729	736	737	739	741	746	747	748	751	759
	760	761	768	772	773	780	781	789	790	799	800	803	808	809	810
	812	819	820	822	824	830	831	832	835	842	847	848	855	856	864
	865	874	875	878	883	884	885	887	894	895	897	899	905	906	907
	910	918	919	926	927	935	936	945	946	949	954	955	956	958	965
	966	968	970	976	977	978	981	985	990	991	998	999	1007	1008	1016
	1017	1020	1025	1026	1027	1029	1036	1037	1039	1041	1047	1048	1049	1051	1059
	1060	1062	1067	1068	1069	1075	1076	1077	1079	1084	1085	1086	1091	1092	1099
	1100	1101	1108	1109	1111	1112	1113	1114	1126	1127	1133	1134	1136	1149	1150
	1151	1158	1162	1164	1165	1171	1172	1174	1175	1181	1182	1183	1184	1192	1193
	1194	1195	1206	1207	1208	1209	1211	1220	1225	1226	1238	1245	1246	1255	1256
	1263	1264	1270	1271	1272	1279	1280	1287	1290	1291	1325	1326	1327	1337	1341
	1342	1343	1353	1357	1358	1359	1369	1373	1374	1375	1385	1389	1390	1391	1401
	1405	1406	1407	1417	1442	1443	1452	1453	1462	1463	1472	1473	1482	1483	1492
	1493	1494	1498	1513	1514	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530
	1531	1532	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1569	1581	1637
	1638	1640	1642	1643	1663	1665	1667	1668	1695	1696	1701	1702	1709	1710	1713
	1714	1733	1734	1755	1756	1757	1758	1761	1782	1783	1784	1785	1791	1792	1793
	1800	1801	1802	1809	1810	1811	1812	1817	1818	1819	1826	1827	1828	1837	1839
	1851	1852	1853	1860	1861	1862	1869	1870	1872	1875	1876	1892	1895	1896	1897
	1901	1902	1918	1921	1922	1925	1927	1940	1944	1947	1949	1962	1967	1968	1970
	1971	1974	1975	2001	2005	2006	2017	2019	2020	2033	2035	2036	2045	2047	2056
	2057	2058	2059	2060	2061	2073	2079	2080	2081	2086	2087	2112	2113	2116	2117
	2118	2143	2144	2145	2146	2149	2154	2155	2160	2161	2162	2170	2171	2180	2181
	2189	2190	2198	2202	2203	2205	2206	2216	2217	2218	2219	2226	2227	2228	2229
	2233	2237	2238	2239	2240	2241	2242	2250	2251	2252	2253	2255	2256	2257	2262
	2263	2268	2269	2270	2271	2272	2273	2274	2278	2283	2284	2285	2286	2298	2300
	2303	2307	2308	2310	2311	2316	2321	2322	2324	2326	2334	2335	2337	2339	2351
	2352	2353	2354	2357	2361	2374	2375	2382	2383	2384	2393	2394	2395	2399	2402
	2406	2407	2414	2416	2417	2418	2430	2435	2436	2445	2461	2468	2475	2482	2490
	2498	2505	2513	2521	2528	2529	2530	2537	2538	2539	2546	2547	2548	2549	2556

H06

MAINDEC-11-DCKBM-E PDP11/45 TRAPS TEST. MACY11 27(732) 08-APR-76 10:45 PAGE 76
 DCKBME.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	2557	2558	2559	2566	2567	2568	2569	2576	2577	2578	2579	2586	2587	2588	2589
	2639	2640	2647	2648	2655	2656	2663	2664	2695	2697	2704	2705	2706	2720	2721
	2735	2736	2737	2740	2741	2742	2766	2767	2768	2769	2770	2773	2774	2782	2786
	2787	2790	2803	2806	2809										
MOVB	456	497	509	1323	2462	2469	2476	2531	2540	2550	2560	2570	2580	2590	2641
	2649	2657													
NOP	1258	1978	1980	2090	2092	2208	2231	2405	2779	2813	2814	2815			
RESET	2287	2302	2313	2314	2320	2331	2350	2373	2811						
ROL	2192														
RTI	1176	1185	1196	1661	1687	1690	2220	2365							
RTT	1210	1669	2062	2207	2230	2275	2312	2396	2403	2658	2708	2723	2775	2783	2789
SCC	1135	1644	2670	2722											
SEC	2191														
SPL	1495	1500	2349	2381	2415	2419									
STFPS	1229														
SUB	1664														
SWAB	1128														
SXT	1103														
TRAP	537	544	553	563	572	584	595	604	607	1376	1572	1903	2244	2258	
TST	993	1000	1009	1019	1028	1040	1050	1248	1533	1654	1680	1726	1748	1764	1776
	1845	1880	1886	1906	1912	1931	1937	1953	1959	1996	2026	2039	2105	2131	2341
	2370	2438	2443	2753											
TSTB	1496	2147	2426	2804	2807										
WAIT	2388	2420													
XOR	1087														
.ABS	127														
.END	2839														
.LIST	125														
.MACR	143	1304	1421												
.NLIST	126	438													
.REM	1														
.REPT	182														
.TITLE	124														
.WORD	2327	2340													

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*,DCKBME.SEO/CRF/SOL=DCKBME.DOC,DCKBME.P11/DS:ERFZ
 RUN-TIME: 7 16 3 SECONDS
 RUN-TIME RATIO: 122/28=4.2
 CORE USED: 9K (17 PAGES)

Spooler runtime 10 Seconds, 44 KCS, 269 disk reads, 4 disk writes, 72 pages

4/24/00 Mueller IPC-0 0070 (100) 000000

[illegible]