

# KJ11

STACK LIMIT TEST  
MD-11-DCKBF-C

EP-DCKBF-C-DL-B  
COPYRIGHT © 1977  
FICHE 1 OF 1

APR 1977  
**digital**  
MADE IN USA

*[Faint, illegible text and graphics visible along the left edge of the page, likely bleed-through from the reverse side.]*

B01

EOF108KTFDSEQ

00010000

770323

PDP10 411

67HDR1DCKBFCSEQ

00010000

770323

CO1

DCKBF-C STACK LIMIT REGISTER TEST  
DCKBFC.P11 25-OCT-76 22:27

MACY11 27(1006) 25-OCT-76 22:30 PAGE 2

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBF-C-D  
PRODUCT NAME: STACK LIMIT TEST  
STANDARD ON 11/45  
KJ11-A OPTION ON 11/40  
DATE RELEASED: MARCH, 1977  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHORS: ;JOHN ADAMS

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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1972, 1977 BY DIGITAL EQUIPMENT CORPORATION

## 1.0 ABSTRACT

THIS PROGRAM INCREMENTLY TESTS THE STACK LIMIT FUNCTION.

THE PROGRAM USES THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES IF NO HARDWARE SWITCH REGISTER IS FOUND. THE OPERATOR IS RESPONSIBLE FOR LOADING THE DESIRED VALUE BEFORE STARTING THE PROGRAM. LOCATION 174 WILL BE USED AS THE SOFTWARE DISPLAY REGISTER.

## 2.0 REQUIREMENTS

## 2.1 EQUIPMENT

BASIC 11/45 SYSTEM OR  
11/40 WITH STACK LIMIT OPTION

## 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 AND PRINT '\*' ON PASS COMPLETION.

## 5.0 OPERATING PROCEDURE

## 5.1 SWITCH SETTINGS

NONE

## 5.2 SUBROUTINE ABSTRACTS

## 5.2.1 SCOPE

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

## 5.2.2 HLT

HLT IS A HALT INSTRUCTION.

## 6.0 ERRORS

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

## 6.1 ERROR RECOVERY

PRESS CONTINUE TO PROCEED TO NEXT TEST.

## 6.2 ERROR LOOPING

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

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

```
TRAPVEC:      TRAPVEC+4
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 ADDRESS WHERE THE TRAP OCCURRED.

## 8.1 EXECUTION TIME

EACH PROGRAM TAKES ABOUT 1 MINUTE.

## 9.0 PROGRAM DESCRIPTIONS

THIS IS A TEST OF THE STACK LIMIT REGISTER AND INSURES CORRECT OPERATION OF THE RED AND YELLOW ZONE BOUNDARIES. OVERFLOW TRAPS ARE TESTED FOR ALL VALUES OF THE STACK LIMIT REGISTER.

```

151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167      000000
168      000001
169      000002
170      000003
171      000004
172      000005
173      000006
174      000007
175
176
177      177776
178      177770
179      177570
180      177564
181      177566
182      177570
183      000000
184      022626
185      010701
186      000340
187      000004
188
189      000000
190
191
192      000046
193      000046 002230
194      000052
195      000052 000000
196
197      000174
198      000174 000000
199      000176 000000
200
201      000200
202      000200 000167 000604
203
204      000500
205
206      000500 000000
    
```

```

;TEST DCKBFC- STACK LIMIT REGISTER TEST
;THE STACK LIMIT REGISTER ALLOWS THE 'OVERFLOW' BOUNDARIES TO BE CHANGED.
;FOR EXAMPLE IF THE STACK LIMIT REGISTER IS CLEAR THE BOUNDARY IS AT
;400 (YELLOW ZONE) AND 340 (RED ZONE). IN ALL CASES THE YELLOW ZONE
;BOUNDARY IS AT 400(8) PLUS THE VALUE IN THE STACK LIMIT REGISTER, AND
;THE RED ZONE BEGINS 20(8) WORDS BELOW THE YELLOW ZONE. THIS TEST
;CHECKS THAT THE STACK LIMIT IS 400 GREATER THAN THE CONTENTS OF THE
;STACK LIMIT REGISTER (CORE PERMITTING), AND CHECKS THE LENGTH OF THE
;YELLOW ZONE AND THE BEGINNING OF THE RED ZONE.
    
```

```

;STARTING PROCEDURE
;      LOAD ADDRESS=200
;      PRESS START
;      BELL WILL RING WHEN TEST IS COMPLETE
    
```

```

;EQUATE STATEMENTS
RO=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7
    
```

```

;REGISTER ADDRESSES
PSW=177776
UBREAK=177770
DSWR=177570
TPS=177564
TPB=177566
DDISP=177570
    
```

```

;ADDRESS OF PROCESSOR STATUS WORD
;ADDRESS OF PDP11/45 MICRO BREAK REGISTER
;ADDRESS CONSOLE SWITCH REGISTER
    
```

```

;ADDRESS OF CONSOLE DISPLAY REGISTER
    
```

```

HLT=HALT
POP2=22626
SCOPE=010701
PRTY7=340
ERRVEC=4
    
```

```

;MOVE PC TO R1
;PRIORITY LEVEL 7
;ADDRESS OF ERROR VECTOR
    
```

```

.=0
.=46
$ENDAD
.=52
000000
.=174
DISPREG:0
SWREG: 0
    
```

```

.=200
JMP START
    
```

```

.=500
;TAGS
ICNT: 0
    
```

```

;CONTAINS PASS COUNT
    
```



```

263 001270 016706 177214 BEGIN: MOV SPBOT,%6 ;INITIALIZE STACK POINTER
264 001274 016777 177200 177204 MOV ICNT,%DISPLAY ;DISPLAY PASS COUNT
265 001302 032777 000400 177174 BIT #400,%SWR ;LOAD PDP11/45 MICRO BREAK REGISTER?
266 001310 001403 BEQ .+10
267 001312 117737 177166 177770 MOVB %SWR,%UBREAK ;LOAD MICRO BREAK REG WITH SPO-7
268
269 ;CHECK THAT CP CAN TIME OUT TRAP
270 001320 012737 001336 000004 MOV #TORET,%ERRVEC ;LOAD TIMEOUT TRAP VECTOR
271 001326 005037 173000 CLR %173000 ;ADDRESS 173000 ALWAYS TIMES OUT ON
272 ;DATIP/DATO BUS CYCLE
273 001332 000000 HLT ;ERROR! FAILED TO TIME OUT TRAP
274 001334 000755 BR BEGIN ;LOOP TEST
275 001336 022626 TORET: CMP (6)+,(6)+ ;RESTORE THE STACK
276
277 ;TEST THAT THE STACK LIMIT REGISTER CAN BE REFERENCED USING DATI,
278 ;DATIP/DATO
279 001340 010701 TO: SCOPE
280 001342 012767 001364 176434 MOV #TOA,ERRVEC ;LOAD ERROR VECTOR
281 001350 005067 176432 CLR ERRVEC+2
282 001354 017737 177132 177774 MOV %SLR,%177774 ;REFERENCE STACK LIMIT REGISTER
283 001362 000403 BR TOB ;GO TO NEXT TEST
284 001364 022626 TOA: POP2
285 001366 000000 HLT ;ERROR: CANNOT REFERENCE STACK LIMIT REG.
286 001370 000763 BR TO ;LOOP TEST IF ERROR
287
288 ;USING DATI, DATIP/DATOB
289
290 001372 012767 001410 176404 TOB: MOV #TOC,ERRVEC ;LOAD ERROR VECTOR
291 001400 117777 177106 177106 MOVB %SLR,%SLH ;REFERENCE ODD BYTE
292 001406 000403 BR TOD
293 001410 022626 TOC: POP2
294 001412 000000 HLT ;ERROR! CANNOT REFERENCE STACK LIMIT
295 001414 000766 BR TOB ;USING BYTE INSTRUCTION.
296
297 001416 012767 000006 176360 TOD: MOV #6,ERRVEC ;RESTORE ERROR TRAP VECTOR
298 001424 000400 BR T1 ;GO TO NEXT TEST
299
300 ;TEST THAT EACH BIT OF THE STACK LIMIT REGISTER BITS CAN BE SET
301 ;AND CLEARED. THIS TEST ROTATES A BIT THROUGH THE STACK LIMIT REGISTER.
302
303 001426 010701 T1: SCOPE
304 001430 012702 000400 MOV #400,R2 ;LOAD TEST VALUE
305 001434 010277 177052 T1A: MOV R2,%SLR ;LOAD TEST VALUE INTO STACK LIM. REG.
306 001440 017700 177046 MOV %SLR,R0 ;GET RESULT
307 001444 020002 CMP R0,R2 ;CHECK RESULT
308 001446 001401 BEQ .+4 ;BRANCH IF RESULT IS CORRECT
309 001450 000000 HLT ;ERROR! INCORRECT RESULT. R2 HAS CORRECT
310 ;RESULT AND R0 HAS INCORRECT RESULT.
311 001452 005077 177034 CLR %SLR ;CLEAR STACK LIM. REG.
312 001456 017700 177030 MOV %SLR,R0 ;GET AND CHECK RESULT
313 001462 001401 BEQ .+4
314 001464 000000 HLT ;ERROR! INCORRECT RESULT
315 001466 006302 ASL R2 ;SHIFT TEST VALUE
316 001470 103361 BCC T1A ;BRANCH IF NOT DONE
317 001472 000746 BR T2
318 001410 .=1410

```

```

319
320 ;THIS TEST INCREMENTS THE STACK LIMIT REGISTER
321
322 001410 010701 T2: SCOPE
323 001412 005067 177100 CLR TEMP
324 001416 005077 177070 CLR @SLR ;CLEAR STACK LIMIT REGISTER
325 001422 017700 177064 T2A: MOV @SLR,R0 ;GET RESULT
326 001426 020067 177064 CMP R0,TEMP ;CHECK RESULT
327 001432 001401 BEQ .+4
328 001434 000000 HLT ;ERROR! STACK LIM. REG. WAS INCORRECT DATA
329 ;TEMP # CORRECT RESULT
330 001436 105277 177052 INCB @SLH ;INCREMENT VALUE IN STACK LIM. REG.
331 001442 105267 177051 INCB TEMP+1 ;INCREMENT TEST VALUE
332 001446 001365 BNE T2A ;BRANCH IF ALL VALUES NOT TESTED
333
334 ;TEST THAT RESET CLEARS THE STACK LIMIT REGISTER
335
336 001450 010701 T3: SCOPE
337 001452 012777 177777 177032 MOV #-1,@SLR ;PRESET SLR
338 001460 017700 177026 MOV @SLR,R0 ;SAVE SLR
339 001464 022700 177400 CMP #177400,R0 ;CHECK THAT SLR WAS LOADED
340 001470 001401 BEQ .+4
341 001472 000000 HLT ;! SLR FAILED TO LOAD
342 001474 000005 RESET ;RESET CLEARS SLR
343 001476 017700 177010 MOV @SLR,R0 ;GET RESULT OF RESET AND CHECK RESULT
344 001502 001402 BEQ T4 ;GO TO NEXT TEST IF RESET CLEARED SLR
345 001504 000000 HLT ;ERROR! RESET FAILED TO CLEAR SLR
346 001506 000760 BR T3 ;LOOP TEST IF ERROR
347
348 ;TEST THAT THE CLEAR INSTRUCTION CLEARS THE STACK LIMIT REGISTER
349
350 001510 010701 T4: SCOPE
351 001512 112777 177777 176774 MOVB #-1,@SLH ;PRESET ODD BYTE
352 001520 017700 176766 MOV @SLR,R0 ;GET RESULT
353 001524 022700 177400 CMP #177400,R0 ;CHECK RESULT
354 001530 001401 BEQ .+4
355 001532 000000 HLT ;ERROR! SLR DID NOT PRESET
356 001534 005077 176752 CLR @SLR
357 001540 017700 176746 MOV @SLR,R0 ;GET RESULT OF CLEAR & BRANCH IF CLEAR
358 001544 001402 BEQ T5 ;GO TO NEXT TEST
359 001546 000000 HLT ;ERROR! CLR INST FAILED TO CLEAR SLR
360 001550 000757 BR T4 ;LOOP TEST IF ERROR
361
362
363 ;TEST THAT AN OVERFLOW ERROR OCCURS FOR ALL STACK LIMIT REGISTER VALUES.
364 ;(PROVIDED CORE IS AVAILABLE).
365
366 001552 010701 T5: SCOPE
367 001554 012702 000010 MOV #10,R2 ;INITIALIZE STACK VALUE AND
368 001560 012703 177400 MOV #-400,R3 ;STACK LIMIT REGISTER VALUE
369 001564 062702 000400 T5A: ADD #400,R2 ;LOAD NEW STACK VALUE
370 001570 062703 000400 ADD #400,R3 ;AND NEW STACK LIM. REG. VALUE
371 001574 005037 000000 CLR @#0 ;CLEAR ADDRESS 0
372 001600 016706 176704 MOV SPBOT,SP ;INITIALIZE THE STACK POINTER
373 001604 012767 002072 176172 T5B: MOV #LIMX,ERRVEC ;LOAD TIME OUT TRAP
374 001612 016204 177776 MOV -2(2),R4 ;SAVE STACK LOCATIONS

```

375	001616	016205	177774		MOV	-4(2),R5	;EXIT TEST IF EITHER INST TIMES OUT
376	001622	012767	001646	176154	MOV	#LIMA,ERRVEC	;LOAD OVERFLOW VECTOR
377	001630	010206			MOV	R2,SP	;LOAD STACK POINTER
378	001632	010377	176654		MOV	R3,@SLR	;LOAD STACK LIM. REG.
379	001636	016666	177770	177770	MOV	-10(6),-10(6)	;REFERENCE LIMIT ADDRESS
380	001644	000401			BR	.+4	;SHOULD NOT HAVE TRAPPED
381	001646	000000			LIMA: HLT		;ERROR! REFERENCE TO LIMIT ADDRESS
382							;CAUSED AN OVERFLOW
383	001650	010206			MOV	R2,SP	;REPOINT STACK POINTER
384	001652	012767	001670	176124	MOV	#LIMB,ERRVEC	;REPOINT OVERFLOW VECTOR
385	001660	016666	177766	177766	MOV	-12(6),-12(6)	;REFERENCE FIRST 'YELLOW' ADDRESS
386	001666	000000			HLT		;ERROR! SHOULD HAVE TRAPPED
387	001670	005737	000000		LIMB: TST	@#0	;WAS IT 'RED' OVERFLOW?
388	001674	001401			BEQ	.+4	
389	001676	000000			HLT		;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
390							;A 'YELLOW' ADDRESS WAS REFERENCED.
391	001700	010206			MOV	R2,SP	;REPOINT STACK POINTER
392	001702	005037	000000		CLR	@#0	;CLEAR ADDRESS 0
393	001706	012767	001724	176070	MOV	#LIMC,ERRVEC	;REPOINT OVERFLOW VECTOR
394	001714	016666	177730	177730	MOV	-50(6),-50(6)	;REFERENCE LAST 'YELLOW' ADDRESS
395	001722	000000			HLT		;ERROR! SHOULD HAVE TRAPPED
396	001724	005737	000000		LIMC: TST	@#0	;WAS IT 'RED' OVERFLOW
397	001730	001401			BEQ	.+4	
398	001732	000000			HLT		;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
399							;THE LAST 'YELLOW' ADDRESS WAS REFERENCED
400	001734	005037	000000		CLR	@#0	;CLEAR ADDRESS 0
401	001740	010206			MOV	R2,SP	;REPOINT THE STACK POINTER
402	001742	012767	002010	176034	MOV	#LIMD,ERRVEC	;REPOINT THE OVERFLOW TRAP
403	001750	016267	177726	176540	MOV	-52(2),TEMP	;GET 'RED' LOCATION
404	001756	005166	177726		COM	-52(6)	;REFERENCE 'RED' ADDRESS
405	001762	000000			LIMCC: HLT		;ERROR! NO OVERFLOW TRAP WHEN 'RED' ADDRESS
406	001764	000411			BR	LIMD	;WAS REFERENCED
407		002010				=2010	
408	002010	026267	177726	176500	LIMD: CMP	-52(2),TEMP	;WAS INSTRUCTION ABORTED?
409	002016	001401			BEQ	.+4	
410	002020	000000			HLT		;ERROR! COM -52(6) WAS ALLOWED TO CHANGE
411							;A 'RED' ADDRESS.
412	002022	016762	176470	177726	MOV	TEMP,-52(2)	;RESTORE 'RED' LOCATION IN ANY EVENT
413	002030	005706			TST	%6	;WAS STACK POINTER CHANGED TO 0?
414	002032	001401			BEQ	.+4	
415	002034	000000			HLT		; 'RED' OVERFLOW DID NOT ASSUME NEW STACK
416	002036	022737	001762	000000	CMP	#LIMCC,@#0	;IS RETURN ADDRESS ON NEW STACK?
417	002044	001401			BEQ	.+4	
418	002046	000000			HLT		;ERROR! RETURN ADDRESS NOT SAVED ON NEW
419							;STACK
420	002050	005077	176436		CLR	@SLR	;GET READY TO GET NEW VALUES FOR TEST
421	002054	016706	176430		MOV	SPBOT,SP	;INITIALIZE THE STACK POINTER
422	002060	010462	177776		MOV	R4,-2(2)	;RESTORE STACK ADDRESS DATA
423	002064	010562	177774		MOV	R5,-4(2)	
424	002070	000635			BR	T5A	;GET NEW VALUES
425	002072	012767	000006	175704	LIMX: MOV	#6,ERRVEC	;RESTORE TIME OUT TRAP VECTOR
426							
427							
428	002100	005767	176376		TST	ICNTA	;TEST FOR
429	002104	001013			BNE	END	
430	002106	022767	177777	000120	CMP	#-1,SENDAD+4	;SCRIPT

# K01

DCKBF-C STACK LIMIT REGISTER TEST  
 DCKBFC.P11 25-OCT-76 22:27

MACY11 27(1006) 25-OCT-76 22:30 PAGE 10

431	002114	001417				BEQ	DONE		;CONDITION
432	002116	026767	175724	175716		CMP	46,42		;AND
433	002124	001003				BNE	END		;SET
434	002126	005267	176350			INC	ICNTA		;FLAG
435	002132	000410				BR	DONE		
436	002134	005267	176340		END:	INC	ICNT		
437	002140	026727	176334	001500		CMP	ICNT,#1500		;HAVE 1500 PASSES BEEN COMPLETED
438	002146	001402				BEQ	DONE		
439	002150	000167	177114			JMP	BEGIN		
440	002154	012767	000007	175404	DONE:	MOV	#7,TPB		;RING BELL
441	002162	105767	175376			TSTB	TPS		
442	002166	100375				BPL	.-4		
443	002170	012767	000052	175370		MOV	#52,TPB		;PRINT '*'
444	002176	105767	175362			TSTB	TPS		
445	002202	100375				BPL	.-4		
446	002204	012767	000000	175354		MOV	#0,TPB		;INSERT
447	002212	105767	175346			TSTB	TPS		;NULL
448	002216	100375				BPL	.-4		;CHARACTER
449	002220	013702	000042			MOV	#42,%2		;GET MONITOR RETURN ADDRESS
450	002224	001410				BEQ	DONE1		;DO NOT RETURN IF (42)=0
451	002226	000005				RESET			
452	002230	004712			SENDAD:	JSR	7,(2)		;RETURN TO MONITOR
453	002232	000240				NOP			;ACT11
454	002234	000240				NOP			;OVERLAY
455	002236	000240				NOP			;AREA
456	002240	005000				CLR	RO		;DELAY FOR ACT11
457	002242	005200				INC	RO		
458	002244	001376				BNE	.-2		
459	002246	000167	176536		DONE1:	JMP	START		;RESTART
460									
461		000001					.END		

BEGIN	001270	ICNT	000500	SCOPE =	010701	TORET	001336	T2A	001422
DDISP =	177570	ICNTA	000502	SLH	000514	TPB =	177566	T3	001450
DISPLA	000506	LIMA	001646	SLR	000512	TPS =	177564	T4	001510
DISPRE	000174	LIMB	001670	SPBOT	000510	TO	001340	T5	001552
DONE	002154	LINC	001724	START	001010	TOA	001364	T5A	001564
DONE1	002246	LIMCC	001762	STARTA	001136	TOB	001372	T5B	001604
DSMR =	177570	LIMD	002010	STARTB	001232	TOC	001410	UBREAK=	177770
END	002134	LIMX	002072	SWR	000504	TOD	001416	SENDAD	002230
ERRVEC=	000004	POP2 =	022626	SWREG	000176	T1	001426	.	= 002252
FTITLE	000520	PRTY7 =	000340	TEMP	000516	T1A	001434		
HLT =	000000	PSW =	177776	TITLE	001064	T2	001410		

. ABS. 002252 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

DSKZ:DCKBFC,DSKZ:DCKBFC/SOL=DSKZ:DCKBFC.SML,DSKZ:DCKBFC.P11  
RUN-TIME: 20 21 .1 SECONDS  
RUN-TIME RATIO: 78/42=1.8  
CORE USED: 31K (61 PAGES)