

KJ11-A

11/40,45 STACK LIMIT
CCKBFD0

AH-7804D-MC

COPYRIGHT © 72-78

FICHE 1 OF 1

MAR 1977

digital

MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE: AC-7802D-MC
PRODUCT NAME: CCKBFD0 11/40,45 STACK LIMIT
PRODUCT DATE: FEB 1978
MAINTAINER: PRODUCT ENHANCEMENT

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) 1972,1978 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" WHICH 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 PRECEEDING 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.

```

148 :CCKBFD0 11/40.45 STACK LIMIT
149 :THE STACK LIMIT REGISTER ALLOWS THE 'OVERFLOW' BOUNDARIES TO BE CHANGED.
150 :FOR EXAMPLE IF THE STACK LIMIT REGISTER IS CLEAR THE BOUNDARY IS AT
151 :400 (YELLOW ZONE) AND 340 (RED ZONE). IN ALL CASES THE YELLOW ZONE
152 :BOUNDARY IS AT 400(8) PLUS THE VALUE IN THE STACK LIMIT REGISTER, AND
153 :THE RED ZONE BEGINS 20(8) WORDS BELOW THE YELLOW ZONE. THIS TEST
154 :CHECKS THAT THE STACK LIMIT IS 400 GREATER THAN THE CONTENTS OF THE
155 :STACK LIMIT REGISTER (CORE PERMITTING), AND CHECKS THE LENGTH OF THE
156 :YELLOW ZONE AND THE BEGINNING OF THE RED ZONE.
157
158 :STARTING PROCEDURE
159 :   LOAD ADDRESS=200
160 :   PRESS START
161 :   BELL WILL RING WHEN TEST IS COMPLETE
162
163 :EQUATE STATEMENTS
164 :   R0=%0
165 :   R1=%1
166 :   R2=%2
167 :   R3=%3
168 :   R4=%4
169 :   R5=%5
170 :   SP=%6
171 :   PC=%7
172
173 ;REGISTER ADDRESSES
174 :   PSW=177776 ;ADDRESS OF PROCESSOR STATUS WORD
175 :   UBREAK=177770 ;ADDRESS OF PDP11/45 MICRO BREAK REGISTER
176 :   DSWR=177570 ;ADDRESS CONSOLE SWITCH REGISTER
177 :   TPS=177564
178 :   TPB=177566
179 :   DDISP=177570 ;ADDRESS OF CONSOLE DISPLAY REGISTER
180 :   HLT=HALT
181 :   POP2=22626
182 :   SCOPE=010701 ;MOVE PC TO R1
183 :   PRTY7=340 ;PRIORITY LEVEL 7
184 :   ERRVEC=4 ;ADDRESS OF ERROR VECTOR
185
186 :   . =0
187
188 :   . =46
189 :   $ENDAD
190 :   . =52
191 :   000000
192 :   000052
193 :   000000
194 :   . =174
195 :   DISPREG:0
196 :   SWREG: 0
197
198 :   . =200
199 :   JMP START
200 :   . =500
201 :   . =500
202 :   . =500
203 :   TAGS
204 :   ICNT: 0 ;CONTAINS PASS COUNT

```

```

204 000502 000000          ICNTA: 0
205 000504 177570          SWR: DSWR
206 000506 177570          DISPLAY:DDISP
207 000510 000760          SPBOT: 760
208 000512 177774          SLR: 177774          ;ADDRESS OF STACK LIMIT REGISTER
209 000514 177775          SLH: 177775          ;HIGH (ODD BYTE)
210 000516 000000          TEMP: 0
211 000520 000000          FTITLE: 0          ;TITLE PRINTED = 1
212
213
214 001010 016706 177474          START: . =1010          ;INITIALIZE STACK POINTER
215 001014 012737 000340 177776          MOV #PRTY,%6          ;LOCK OUT INTERRUPTS
216 001022 023737 000042 000046          CMP #42,%46          ;ARE WE IN ACT11 AUTOMATIC MODE?
217 001030 001475          BEQ STARTB          ;YES, SKIP THE TITLE
218 001032 005767 177462          TST FTITLE          ;TITLE PRINTED YET?
219 001036 001034          BNE STARTA          ;YES, SKIP TITLE
220 001040 012700 001064          MOV #TITLE,RO          ;GET MESSAGE ADDRESS
221 001044 105710          1$: TSTB (0)          ;END OF MESSAGE?
222 001046 001430          BEQ STARTA          ;YES, GET OVER THE ASCII
223 001050 105767 176510          TSTB TPS
224 001054 100375          BPL .-4
225 001056 112067 176504          MOVB (0)+,TPB          ;PRINT CHARACTER
226 001062 000770          BR 1$
227 001064 005015 041577 045503          TITLE: .ASCIZ <15><12><177>'CCKBFDD 11/40,45 STACK LIMIT?'<15><12><177>
228 001072 043102 030104 030440
229 001100 027461 030064 032054
230 001106 020065 052123 041501
231 001114 020113 044514 044515
232 001122 006524 077412 000
233 001130 001130
234 001130 012767 000001 177362          STARTA: .EVEN          ;SET TITLE PRINTED FLAG
235          MOV #1,FTITLE          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
236          ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
237 001136 013746 000004          MOV #ERRVEC -(SP)          ;;SAVE ERROR VECTOR
238 001142 012737 001176 000004          MOV #64,%ERRVEC          ;;SET UP ERROR VECTOR
239 001150 012767 177570 177326          MOV #DSWR,SWR          ;;SETUP FOR A HARDWARE SWICH REGISTER
240 001156 012767 177570 177322          MOV #DDISP,DISPLAY          ;;AND A HARDWARE DISPLAY REGISTER
241 001164 022777 177777 177312          CMP #-1,%DSWR          ;;TRY TO REFERENCE HARDWARE SWR
242 001172 001012          BNE 66$          ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
243          ;;AND THE HARDWARE SWR IS NOT = -1
244          BR 65$          ;;BRANCH IF NO TIMEOUT
245 001176 012716 001204          64$: MOV #65$, (SP)          ;;SET UP FOR TRAP RETURN
246 001202 000002
247 001204 012767 000176 177272          65$: MOV #SWREG,SWR          ;;POINT TO SOFTWARE SWR
248 001212 012767 000174 177266          MOV #DISPREG,DISPLAY
249 001220 012637 000004          66$: MOV (SP)+,%ERRVEC          ;;RESTORE ERROR VECTOR
250 001224 005067 177250          STARTB: CLR ICNT          ;CLEAR PASS COUNT
251 001230 005767 177246          TST ICNTA
252 001234 001412          BEQ BEGIN
253 001236 022767 177777 001104          CMP #-1,$ENDAD+4
254 001244 001006          BNE BEGIN
255 001246 022767 001477 177224          CMP #1477,ICNT
256 001254 001002          BNE BEGIN
257 001256 005067 177220          CLR ICNTA
258
259 001262 016706 177222          BEGIN: MOV SPBOT,%6          ;INITIALIZE STACK POINTER
    
```

```

260 001266 016777 177206 177212      MOV      ICNT, @DISPLAY      ;DISPLAY PASS COUNT
261 001274 032777 000400 177202      BIT      #400, @SWR        ;LOAD PDP11/45 MICRO BREAK REGISTER?
262 001302 001403                BEQ      .+10
263 001304 117737 177174 177770      MOV      @SWR, @UBREAK      ;LOAD MICRO BREAK REG WITH SPO-7
264
265                ;CHECK THAT CP CAN TIME OUT TRAP
266 001312 012737 001330 000004      MOV      #TRET, @ERRVEC     ;LOAD TIMEOUT TRAP VECTOR
267 001320 005037 173000                CLR      @#173000          ;ADDRESS 173000 ALWAYS TIMES OUT ON
268                ;DATIP/DATO BUS CYCLE
269 001324 000000                HLT
270 001326 000755                BR      BEGIN              ;ERROR! FAILED TO TIME OUT TRAP
271 001330 022626      TRET:    CMP      (6)+, (6)+ ;LOOP TEST
272                ;RESTORE THE STACK
273                ;TEST THAT THE STACK LIMIT REGISTER CAN BE REFERENCED USING DATI,
274                ;DATIP/DATO
275 001332 010701      TO:      SCOPE
276 001334 012767 001356 176442      MOV      #TOA, ERRVEC      ;LOAD ERROR VECTOR
277 001342 005067 176440                CLR      ERRVEC+2
278 001346 017737 177140 177774      MOV      @SLR, @#177774    ;REFERENCE STACK LIMIT REGISTER
279 001354 000415                BR      TOB                ;GO TO NEXT TEST
280 001356 022626      TOA:    POP2
281 001360 000000                HLT
282 001362 000763                BR      TO                  ;ERROR: CANNOT REFERENCE STACK LIMIT REG.
283                ;LOOP TEST IF ERROR
284                001364      STKLC1=.
285
286
287                ;USING DATI, DATIP/DATOB
288
289                001410      .=1410
290
291 001410 012767 001426 176366      TOB:    MOV      #TDC, ERRVEC ;LOAD ERROR VECTOR
292 001416 117777 177070 177070      MOV      @SLR, @SLH        ;REFERENCE ODD BYTE
293 001424 000403                BR      TOD
294 001426 022626      TDC:    POP2
295 001430 000000                HLT
296 001432 000766                BR      TOB                ;ERROR! CANNOT REFERENCE STACK LIMIT
297                ;USING BYTE INSTRUCTION.
298 001434 012767 000006 176342      TOD:    MOV      #6, ERRVEC   ;RESTORE ERROR TRAP VECTOR
299 001442 000400                BR      T1                  ;GO TO NEXT TEST
300
301                ;TEST THAT EACH BIT OF THE STACK LIMIT REGISTER BITS CAN BE SET
302                ;AND CLEARED. THIS TEST ROTATES A BIT THROUGH THE STACK LIMIT REGISTER.
303
304 001444 010701      T1:      SCOPE
305 001446 012702 000400                MOV      #400, R2          ;LOAD TEST VALUE
306 001452 010277 177034      T1A:    MOV      R2, @SLR     ;LOAD TEST VALUE INTO STACK LIM. REG.
307 001456 017700 177030                MOV      @SLR, R0          ;GET RESULT
308 001462 020002                CMP      R0, R2            ;CHECK RESULT
309 001464 001401                BEQ      .+4              ;BRANCH IF RESULT IS CORRECT
310 001466 000000                HLT                       ;ERROR! INCORRECT RESULT. R2 HAS CORRECT
311                ;RESULT AND R0 HAS INCORRECT RESULT.
312 001470 005077 177016                CLR      @SLR              ;CLEAR STACK LIM. REG.
313 001474 017700 177012                MOV      @SLR, R0          ;GET AND CHECK RESULT
314 001500 001401                BEQ      .+4
315 001502 000000                HLT                       ;ERROR! INCORRECT RESULT

```

```

316 001504 006302          ASL      R2          :SHIFT TEST VALUE
317 001506 103361          BCC     T1A         :BRANCH IF NOT DONE
318 001510 000400          BR      T2
319
320
321          ;THIS TEST INCREMENTS THE STACK LIMIT REGISTER
322
323
324 001512 010701          T2:     SCOPE
325 001514 005067 176776      CLR     TEMP
326 001520 005077 176766      CLR     @SLR        :CLEAR STACK LIMIT REGISTER
327 001524 017700 176762      MOV     @SLR,R0     :GET RESULT
328 001530 020067 176762      CMP     R0,TEMP     :CHECK RESULT
329 001534 001401          BEQ     .+4
330 001536 000000          HLT
331          ;ERROR! STACK LIM. REG. WAS INCORRECT DATA
332 001540 105277 176750      INCB   @SLH        :TEMP # CORRECT RESULT
333 001544 105267 176747      INCB   TEMP+1     :INCREMENT VALUE IN STACK LIM. REG.
334 001550 001365          BNE    T2A        :INCREMENT TEST VALUE
335          ;BRANCH IF ALL VALUES NOT TESTED
336
337          ;TEST THAT RESET CLEARS THE STACK LIMIT REGISTER
338
339 001552 010701          T3:     SCOPE
340 001554 012777 177777 176730  MOV     #-1,@SLR   :PRESET SLR
341 001562 017700 176724          MOV     @SLR,R0   :SAVE SLR
342 001566 022700 177400          CMP     #177400,R0 :CHECK THAT SLR WAS LOADED
343 001572 001401          BEQ     .+4
344 001574 000000          HLT
345          ;! SLR FAILED TO LOAD
346 001576 000005          RESET :RESET CLEARS SLR
347 001600 017700 176706      MOV     @SLR,R0   :GET RESULT OF RESET AND CHECK RESULT
348 001604 001402          BEQ     T4        :GO TO NEXT TEST IF RESET CLEARED SLR
349 001606 000000          HLT
350 001610 000760          BR      T3        :ERROR! RESET FAILED TO CLEAR SLR
351          ;LOOP TEST IF ERROR
352
353          ;TEST THAT THE CLEAR INSTRUCTION CLEARS THE STACK LIMIT REGISTER
354
355 001612 010701          T4:     SCOPE
356 001614 112777 177777 176672  MOVVB  #-1,@SLH   :PRESET ODD BYTE
357 001622 017700 176664          MOV     @SLR,R0   :GET RESULT
358 001626 022700 177400          CMP     #177400,R0 :CHECK RESULT
359 001632 001401          BEQ     .+4
360 001634 000000          HLT
361          ;ERROR! SLR DID NOT PRESET
362 001636 005077 176650      CLR     @SLR
363 001642 017700 176644          MOV     @SLR,R0   :GET RESULT OF CLEAR & BRANCH IF CLEAR
364 001646 001402          BEQ     T5        :GO TO NEXT TEST
365 001650 000000          HLT
366 001652 000757          BR      T4        :ERROR! CLR INST FAILED TO CLEAR SLR
367          ;LOOP TEST IF ERROR
368
369          ;TEST THAT AN OVERFLOW ERROR OCCURS FOR ALL STACK LIMIT REGISTER VALUES.
370          ;(PROVIDED CORE IS AVAILABLE).
371
372 001654 010701          T5:     SCOPE
373 001656 012702 000010      MOV     #10,R2    :INITIALIZE STACK VALUE AND
374 001662 012703 177400      MOV     #-400,R3  :STACK LIMIT REGISTER VALUE
375 001666 062702 000400      ADD     #400,R2   :LOAD NEW STACK VALUE
376 001672 062703 000400      ADD     #400,R3   :AND NEW STACK LIM. REG. VALUE
  
```

372	001676	005037	000000			CLR	Q#0	; CLEAR ADDRESS 0
373	001702	016706	176602			MOV	SPBOT, SP	; INITIALIZE THE STACK POINTER
374	001706	012767	002206	176070	TSB:	MOV	#LIMX, ERRVEC	; LOAD TIME OUT TRAP
375	001714	016204	177776			MOV	-2(2), R4	; SAVE STACK LOCATIONS
376	001720	016205	177774			MOV	-4(2), R5	; EXIT TEST IF EITHER INST TIMES OUT
377	001724	012767	001750	176052		MOV	#LIMA, ERRVEC	; LOAD OVERFLOW VECTOR
378	001732	010206				MOV	R2, SP	; LOAD STACK POINTER
379	001734	010377	176552			MOV	R3, QSLR	; LOAD STACK LIM. REG.
380	001740	016666	177770	177770		MOV	-10(6), -10(6)	; REFERENCE LIMIT ADDRESS
381	001746	000420				BR	LIMAA	; SHOULD NOT HAVE TRAPPED
382	001750	000000			LIMA:	HLT		; ERROR! REFERENCE TO LIMIT ADDRESS
383								; CAUSED AN OVERFLOW
384	001752	000416				BR	LIMAA	
385								
386		001754						STKLO2=.
387								
388								
389		002010						.=2010
390								
391	002010	010206			LIMAA:	MOV	R2, SP	; REPOINT STACK POINTER
392	002012	012767	002030	175764		MOV	#LIMB, ERRVEC	; REPOINT OVERFLOW VECTOR
393	002020	016666	177766	177766		MOV	-12(6), -12(6)	; REFERENCE FIRST 'YELLOW' ADDRESS
394	002026	000000				HLT		; ERROR! SHOULD HAVE TRAPPED
395	002030	005737	000000		LIMB:	TST	Q#0	; WAS IT 'RED' OVERFLOW?
396	002034	001401				BEQ	+.4	
397	002036	000000				HLT		; ERROR! A 'RED' OVERFLOW OCCURRED WHEN
398								; A 'YELLOW' ADDRESS WAS REFERENCED.
399	002040	010206				MOV	R2, SP	; REPOINT STACK POINTER
400	002042	005037	000000			CLR	Q#0	; CLEAR ADDRESS 0
401	002046	012767	002064	175730		MOV	#LIMC, ERRVEC	; REPOINT OVERFLOW VECTOR
402	002054	016666	177730	177730		MOV	-50(6), -50(6)	; REFERENCE LAST 'YELLOW' ADDRESS
403	002062	000000				HLT		; ERROR! SHOULD HAVE TRAPPED
404	002064	005737	000000		LIMC:	TST	Q#0	; WAS IT 'RED' OVERFLOW?
405	002070	001401				BEQ	+.4	
406	002072	000000				HLT		; ERROR! A 'RED' OVERFLOW OCCURRED WHEN
407								; THE LAST 'YELLOW' ADDRESS WAS REFERENCED
408	002074	005037	000000			CLR	Q#0	; CLEAR ADDRESS 0
409	002100	010206				MOV	R2, SP	; REPOINT THE STACK POINTER
410	002102	012767	002124	175674		MOV	#LIMD, ERRVEC	; REPOINT THE OVERFLOW TRAP
411	002110	016267	177726	176400		MOV	-52(2), TEMP	; GET 'RED' LOCATION
412	002116	005166	177726			COM	-52(6)	; REFERENCE 'RED' ADDRESS
413	002122	000000			LIMCC:	HLT		; ERROR! NO OVERFLOW TRAP WHEN 'RED' ADDRESS
414								; WAS REFERENCED
415	002124	026267	177726	176364	LIMD:	CMP	-52(2), TEMP	; WAS INSTRUCTION ABORTED?
416	002132	001401				BEQ	+.4	
417	002134	000000				HLT		; ERROR! COM -52(6) WAS ALLOWED TO CHANGE
418								; A 'RED' ADDRESS.
419	002136	016762	176354	177726		MOV	TEMP, -52(2)	; RESTORE 'RED' LOCATION IN ANY EVENT
420	002144	005706				TST	%6	; WAS STACK POINTER CHANGED TO 0?
421	002146	001401				BEQ	+.4	
422	002150	000000				HLT		; 'RED' OVERFLOW DID NOT ASSUME NEW STACK
423	002152	022737	002122	000000		CMP	#LIMCC, Q#0	; IS RETURN ADDRESS ON NEW STACK?
424	002160	001401				BEQ	+.4	
425	002162	000000				HLT		; ERROR! RETURN ADDRESS NOT SAVED ON NEW
426								; STACK
427	002164	005077	176322			CLR	QSLR	; GET READY TO GET NEW VALUES FOR TEST

```

428 002170 016706 176314      MOV      SPBOT,SP      ;INITIALIZE THE STACK POINTER
429 002174 010462 177776      MOV      R4,-2(2)     ;RESTORE STACK ADDRESS DATA
430 002200 010562 177774      MOV      R5,-4(2)
431 002204 000630          BR       TSA          ;GET NEW VALUES
432 002206 012767 000006 175570 LIMX:  MOV      #6,ERRVEC    ;RESTORE TIME OUT TRAP VECTOR
433
434
435 002214 005767 176262      TST      ICNTA        ;TEST FOR
436 002220 001013          BNE     END          ;
437 002222 022767 177777 000120      CMP      #-1,$ENDAD+4 ;SCRIPT
438 002230 001417          BEQ     DONE        ;CONDITION
439 002232 026767 175610 175602      CMP      46,42      ;AND
440 002240 001003          BNE     END          ;SET
441 002242 005267 176234      INC      ICNTA        ;FLAG
442 002246 000410          BR      DONE
443 002250 005267 176224      END:    INC      ICNT
444 002254 026727 176220 001500      CMP      ICNT,#1500  ;HAVE 1500 PASSES BEEN COMPLETED
445 002262 001402          BEQ     DONE
446 002264 000167 176772      JMP     BEGIN
447 002270 012767 000007 175270 DONE:  MOV      #7,TPB      ;RING BELL
448 002276 105767 175262      TSTB   TPS
449 002302 100375          BPL     -4
450 002304 012767 000052 175254      MOV      #52,TPB    ;PRINT '*'
451 002312 105767 175246      TSTB   TPS
452 002316 100375          BPL     -4
453 002320 012767 000000 175240      MOV      #0,TPB    ;INSERT
454 002326 105767 175232      TSTB   TPS        ;NULL
455 002332 100375          BPL     -4        ;CHARACTER
456 002334 013702 000042      MOV      @#42,%2    ;GET MONITOR RETURN ADDRESS
457 002340 001410          BEQ     DONE1      ;DO NOT RETURN IF (42)=0
458 002342 000005          RESET
459 002344 004712      SENDAD: JSR      7,(2)  ;RETURN TO MONITOR
460 002346 000240          NOP
461 002350 000240          NOP              ;ACT11
462 002352 000240          NOP              ;OVERLAY
463 002354 005000          CLR     RO         ;AREA
464 002356 005200          INC     RO         ;DELAY FOR ACT11
465 002360 001376          BNE     -2
466 002362 000167 176422      DONE1:  JMP      START    ;RESTART
467
468          002366      STKL03=.
469
470
471          000001      .END
    
```


COMMEN	1#		
ENDCOM	1#		
ESCAPE	1#		
GETPRI	1#		
GETSWR	1#		
MULT	1#		
NEWTST	1#		
POP	1#		
PUSH	1#		
REPORT	1#		
SETPRI	1#		
SETUP	1#		
SKIP	1#		
SLASH	1#		
STARS	1#		
SWRSU	1#	148#	235
TYPBIN	1#		
TYPDEC	1#		
TYPNAM	1#		
TYPNUM	1#		
TYPOCS	1#		
TYPOCT	1#		
TYPTXT	1#		
\$\$ESCA	1#		
\$\$NEWT	1#		
\$\$SKIP	1#		
.EQUAT	1#		
.H.ODE	1#		
.KTI1	1#		
.SETUP	1#		
.SWRHI	1#		
.SACT1	1#		
.SAPT8	1#		
.SAPTH	1#		
.SAPTY	1#		
.SASTA	1#		
.SCATC	1#		
.SCMTA	1#		
.SOB2D	1#		
.SOB2O	1#		
.SOIV	1#		
.SEOP	1#		
.SERRO	1#		
.SERRT	1#		
.SMULT	1#		
.SPOWE	1#		
.SRAND	1#		
.SRDDE	1#		
.SRDOC	1#		
.SREAD	1#		
.SR2AZ	1#		
.\$SAVE	1#		
.\$SB2D	1#		
.\$SB2O	1#		
.\$SCOP	1#		
.\$SIZE	1#		

CCKBFD 11/40.45 STACK LIMIT
CCKBFD.P11 07-DEC-77 15:50

MACY11 30(1046) 07-DEC-77 15:55 PAGE 16
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0013

.\$SUPR 1#
.\$STRAP 1#
.\$STYPB 1#
.\$STYPD 1#
.\$STYPE 1#
.\$STYPO 1#
.\$4DCA 1#
.\$1170 1#

. ABS. 002366 000

ERRORS DETECTED: 0

CCKBFD,CCKBFD/CRF/SOL/PAGNUM/NL:TOC=CCKBFD.SML,CCKBFD.P11
RUN-TIME: 6 6 .1 SECONDS
RUN-TIME RATIO: 447/13=32.3
CORE USED: 31K (61 PAGES)

