

TEST POINT	TEST POINT	TEST POINT	TEST POINT
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48
49	50	51	52
53	54	55	56
57	58	59	60
61	62	63	64
65	66	67	68
69	70	71	72
73	74	75	76
77	78	79	80
81	82	83	84
85	86	87	88
89	90	91	92
93	94	95	96
97	98	99	100

B01

EOF1CZRBPSPBQ411
DCKTBC.P11

00818800
28-NOV-77 11:17

MA8822330(1046) 28-NOV-77 11:26 PAGE RWD R1 DCKTBC SEQ

00010000

780223
SEQ 0001

.REM *

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKTB-C-D
PRODUCT NAME: KT11-C BASIC LOGIC TEST TWO
DATE RELEASED: FEB 1978
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: RICK FADDEN
MODIFIED BY: BILL SCHLITZKUS

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.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1972, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

MAINDEC-11-DCKTB-C
PAGE 02

1.0 ABSTRACT

THIS PROGRAM AND THE PREVIOUS ONE (DCKTA) INCREMENTALLY TEST THE BASIC LOGIC FUNCTIONS OF THE KT11-C MEMORY MANAGEMENT OPTION FOR THE PDP-11/45. THEY FULLY TEST RELOCATION, DIRECT AND INDIRECT ADDRESSING OF THE MEMORY MANAGEMENT REGISTERS, AND CORRECT OPERATION OF ALL THE BITS IN THE REGISTERS. THE VARIOUS ABORTS ARE TESTED, AS IS PROPER "LOCKING" AND "UNLOCKING" OF THE ERROR TRACKING LOGIC.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11/45 WITH KT11-C OPTION

2.2 STORAGE

THE PROGRAM REQUIRES MEMORY LOCATIONS 0 TO 17474.

3.0 LOADING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

4.0 STARTING PROCEDURE

LOAD ADDRESS 200.
SET DESIRED SWITCH REGISTER SETTINGS (ALL DOWN FOR WORST CASE).
PRESS START.
THE PROGRAM WILL DISPLAY THE NUMBER OF THE CURRENT SUBTEST IN THE DISPLAY REGISTER, AND WILL RING THE BELL ON COMPLETION OF A PASS.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SW 15=1 OR UP -- HALT ON ERROR
 SW 14=1 OR UP -- SCOPE LOOP
 SW 13=1 OR UP -- INHIBIT PRINTOUT
 SW 11=1 OR UP -- INHIBIT ITERATIONS
 SW 08=1 OR UP -- LOAD MICROBREAK REGISTER WITH VALUE IN
 SW 00-07.

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 1024 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS.

5.2.2 HLT

THIS EMT CALLS THE SUBROUTINE PRINT, WHICH PRINTS OUT THE LOCATION COUNTER AT THE TIME OF FAILURE AND THE CONTENTS OF THE PROCESSOR STATUS REGISTER. NOTE THAT THE LOCATION COUNTER WILL BE THE ADDRESS OF THE HLT PLUS TWO.

5.2.3 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH A HALT (000000). THUS AN ILLEGAL TRAP OR INTERRUPT WILL CAUSE A HALT AT THE TRAP LOCATION PLUS TWO.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA EXAMINE REGISTER SIX. IT WILL CONTAIN THE CURRENT STACK ADDRESS. THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VALUE OF THE LOCATION COUNTER WHEN THE TRAP OR INTERRUPT OCCURRED.

5.2.4 EMTSRV (EMT DECODER)

THIS ROUTINE DECODES ALL EMT CALLS, INCLUDING PATCHES AND THE HLT CALL WHICH PASSES CONTROL TO THE PRINT ROUTINE.

5.2.5 CLRALL

THIS ROUTINE CLEARS ALL THE PAR'S AND PDR'S OF THE KT11-C, AS WELL AS SRD.

5.2.6 RWALL

THIS ROUTINE MAPS ALL PAGES TO BANK 0 BY CLEARING ALL THE PAR'S. ALL PAGES ARE MADE 4K READ-WRITE BY LOADING ALL THE PDR'S WITH THE VALUE 77406.

5.2.7 RWISP

THIS ROUTINE MAPS ALL I-SPACE PAGES RW,4K, BANK 0.

5.2.8 RWDSP

THIS ROUTINE MAPS ALL D-SPACE PAGES RW,4K, BANK 0.

5.3 PROGRAM AND/OR OPERATOR ACTION

THIS TEST CONTINUES THE SERIES OF TESTS OF THE KT11-C OPTION STARTED IN MAINDEC-11-DCKTA. NO OPERATOR INTERVENTION IS REQUIRED. THE BELL IS RUNG AT THE END OF EACH PASS, AND THE CURRENT SUBTEST NUMBER IS DISPLAYED IN THE DISPLAY REGISTER.

6.0 ERRORS

6.1 ERROR PRINTOUT

PRINTOUTS ARE IN A STANDARD TWO-WORD FORMAT. THE FIRST WORD IS THE OCTAL VALUE OF THE PC+2 OF THE DETECTED ERROR. THE SECOND IS THE CONTENTS OF THE PROCESSOR STATUS REGISTER WHEN THE ERROR WAS DETECTED.

6.2 ERROR RECOVERY

IN GENERAL, TEST FAILURES WILL PRINTOUT AN ERROR MESSAGE AND CONTINUE. IF THE "HALT ON ERROR" SWITCH IS SET, HITTING CONTINUE WILL RECOVER. IF THE PROGRAM HANGS UP IN A LOOP, THE ERROR IS LIKELY TO BE A SIGNAL WHICH WAS NEVER RECEIVED. IF A HALT OCCURS IN THE TRAP AND VECTOR AREA THE PROGRAM MUST BE RESTARTED, IF THE PROGRAM HALTS IN THE MAIN FLOW, CONSULT THE LISTING IF NO MESSAGE IS TYPED OUT.

6.3 BRANCH SELF

A BRANCH TO SELF IS USED IN THE KT11-C DIAGNOSTICS TO INDICATE A FAILURE WHEN A HALT OR A HLT TRAP CALL COULD CAUSE A PROBLEM.

7.0 RESTRICTIONS

PROGRAM MUST BE LOADED INTO LOWER 4K OF MEMORY.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

EACH PASS TAKES APPROXIMATELY 1 MINUTE WITH CORE MEMORY.

8.2 STACK POINTERS

THE KERNEL STACK POINTER IS USUALLY INITIALIZED TO 1000. HOWEVER, IN CERTAIN TESTS IT MAY BE INITIALIZED TO A LOWER ADDRESS (VIRTUAL) TO MAKE UP FOR RELOCATION OF THE BANK.

THE SUPERVISOR STACK POINTER IS INITIALIZED TO 2000.

THE USER STACK POINTER IS INITIALIZED TO 3000.

8.3 DISPLAY REGISTER

THE NUMBER OF THE CURRENT SUBTEST IS DISPLAYED.

8.4 EXECUTION ORDER CHECKING

SINCE THE KT11-C MAY CAUSE AN INCORRECT FETCH IF IT IS NOT WORKING CORRECTLY, THE ORDER OF EXECUTION OF ALL SUBTESTS IS CHECKED. THE SCOPE ROUTINE, WHEN IT CHANGES FROM ONE SUBTEST TO THE NEXT, INCREMENTS A COUNTER CALLED TESTCT. AT THE START OF EACH SUBTEST, THIS COUNTER IS CHECKED FOR THE CORRECT VALUE FOR THAT SUBTEST. IF TESTS ARE NOT EXECUTED IN THE CORRECT ORDER, TESTCT WILL NOT CONTAIN THE EXPECTED VALUE, AND AN ERROR PRINTOUT WILL OCCUR.

9.0 PROGRAM DESCRIPTION

THIS PROGRAM COMPLETES THE SERIES OF TESTS OF THE KT11-C OPTION STARTED IN MAINDEC-11-DCKTA (BASIC LOGIC TEST ONE). THE BELL IS RUNG AT THE END OF EACH PASS, AND THE CURRENT SUBTEST NUMBER IS DISPLAYED IN THE DISPLAY REGISTER.

*

H01

DCKTB-C MACY11 30(1046) 28-NOV-77 11:26 PAGE 7
DCKTBC.P11 28-NOV-77 11:17

SEQ 0007

;SECOND BASIC LOGIC TEST OF KT11-C

;OPERATING INSTRUCTIONS

- 1. LOAD TEST USING THE ABSOLUTE LOADER
- 2. LOAD SA 200
- 3. SET SP TO INITIAL SETTINGS
- 4. PRESS START

;SWITCH REGISTER SETTINGS ARE:

- SW15=1 CAUSES HALT ON ERROR
- SW14=1 CAUSES SCOPE LOOPING
- SW13=1 INHIBITS ERROR PRINTOUT
- SW11=1 INHIBITS ITERATIONS
- SW08=1 LOAD MICROBREAK REGISTER WITH LOW BYTE OF SWITCH REGISTER

;DEFINITIONS

104400
000240
000000
000001
000002
000003
000004
000005
000006
000007
000006
000007
177570
177776

SCOPE=TRAP
NOP=240
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
R6=%6
R7=%7
SP=%6
PC=%7
SR=177570
PS=177776

;LOAD TRAP CATCHER INTO 0 THRU 777
;LOAD EACH VECTOR ADDRESS WITH THE ADDRESS OF THE NEXT
;LOCATION, AND LOAD EACH LOCATION IMMEDIATELY FOLLOWING
;A VECTOR ADDRESS WITH A HALT INSTRUCTION
;CODE NOT LISTED TO MAKE LISTING EASIER TO READ

;LOAD VECTOR AREA

000030 000030
000032 015614
000034 000340
000036 000034
000046 015056
000052 000000
000052 000046
000052 014500
000052 000052
000052 040000

.=30
EMTSRV
340
.=34
SCOPEC
0
.=46
LOGIC
.=52
40000

;LOAD STARTING AREA

000200 000200
000200 000167 003216

.=200
JMP START

;LOAD DATA AREA

001000 001000
001000 000000

.=1000
KSTACK: 0

;KERNEL STACK BUFFER AREA

002000	002000		SSTACK: 0	.=.+776		;SUPERVISOR STACK BUFFER AREA
003000	000000		USTACK: 0	.=.+776		;USER STACK BUFFER AREA
003002	000000	000000 000000		.WORD 0,0,0,0		
003010	000000		K200:	200		;CONSTANTS
003012	000200		K1000:	1000		
003014	001000		TCSR:	177564		;TELEPRINTER REGISTERS
003016	177564		TDBR:	177566		
003020	177566		TEMP:	0		;TEMPORARY STORAGE LOCATIONS
003022	000000		TEMPX:	0		
003024	000000		TEMP1:	0		
003026	000000		TEMP2:	0		
003030	000000		SR0:	177572		;KT11-C STATUS REGISTER ADDRESSES
003032	177572		SR0H:	177573		
003034	177573		SR1:	177574		
003036	177574		SR1H:	177575		
003040	177575		SR2:	177576		
003042	177576		SR2H:	177577		
003044	177577		SR3:	172516		
003046	172516		SR3H:	172517		
003050	172517					
003052			ADRTAB:			
003052	177600		UIPDR0:	177600		;USER I-SPACE PAGE DESCRIPTOR REGISTER ADDRESSES
003054	177602		UIPDR1:	177602		
003056	177604		UIPDR2:	177604		
003060	177606		UIPDR3:	177606		
003062	177610		UIPDR4:	177610		
003064	177612		UIPDR5:	177612		
003066	177614		UIPDR6:	177614		
003070	177616		UIPDR7:	177616		
003072	177620		UDPDR0:	177620		;USER D-SPACE PAGE DESCRIPTOR REGISTER ADDRESSES
003074	177622		UDPDR1:	177622		
003076	177624		UDPDR2:	177624		
003100	177626		UDPDR3:	177626		
003102	177630		UDPDR4:	177630		
003104	177632		UDPDR5:	177632		
003106	177634		UDPDR6:	177634		
003110	177636		UDPDR7:	177636		
003112	177640		UIPAR0:	177640		;USER I-SPACE PAGE ADDRESS REGISTER ADDRESSES
003114	177642		UIPAR1:	177642		
003116	177644		UIPAR2:	177644		
003120	177646		UIPAR3:	177646		
003122	177650		UIPAR4:	177650		
003124	177652		UIPAR5:	177652		
003126	177654		UIPAR6:	177654		
003130	177656		UIPAR7:	177656		
003132	177660		UDPAR0:	177660		;USER D-SPACE PAGE ADDRESS REGISTER ADDRESSES
003134	177662		UDPAR1:	177662		
003136	177664		UDPAR2:	177664		
003140	177666		UDPAR3:	177666		
003142	177670		UDPAR4:	177670		
003144	177672		UDPAR5:	177672		
003146	177674		UDPAR6:	177674		
003150	177676		UDPAR7:	177676		

003152 172200
 003154 172202
 003156 172204
 003160 172206
 003162 172210
 003164 172212
 003166 172214
 003170 172216
 003172 172220
 003174 172222
 003176 172224
 003200 172226
 003202 172230
 003204 172232
 003206 172234
 003210 172236
 003212 172240
 003214 172242
 003216 172244
 003220 172246
 003222 172250
 003224 172252
 003226 172254
 003230 172256
 003232 172260
 003234 172262
 003236 172264
 003240 172266
 003242 172270
 003244 172272
 003246 172274
 003250 172276

003252 172300
 003254 172302
 003256 172304
 003260 172306
 003262 172310
 003264 172312
 003266 172314
 003270 172316
 003272 172320
 003274 172322
 003276 172324
 003300 172326
 003302 172330
 003304 172332
 003306 172334
 003310 172336
 003312 172340
 003314 172342
 003316 172344
 003320 172346
 003322 172350
 003324 172352

SIPDR0: 172200
 SIPDR1: 172202
 SIPDR2: 172204
 SIPDR3: 172206
 SIPDR4: 172210
 SIPDR5: 172212
 SIPDR6: 172214
 SIPDR7: 172216
 SDPDR0: 172220
 SDPDR1: 172222
 SDPDR2: 172224
 SDPDR3: 172226
 SDPDR4: 172230
 SDPDR5: 172232
 SDPDR6: 172234
 SDPDR7: 172236
 SIPAR0: 172240
 SIPAR1: 172242
 SIPAR2: 172244
 SIPAR3: 172246
 SIPAR4: 172250
 SIPAR5: 172252
 SIPAR6: 172254
 SIPAR7: 172256
 SDPAR0: 172260
 SDPAR1: 172262
 SDPAR2: 172264
 SDPAR3: 172266
 SDPAR4: 172270
 SDPAR5: 172272
 SDPAR6: 172274
 SDPAR7: 172276

KIPDR0: 172300
 KIPDR1: 172302
 KIPDR2: 172304
 KIPDR3: 172306
 KIPDR4: 172310
 KIPDR5: 172312
 KIPDR6: 172314
 KIPDR7: 172316
 KDPDR0: 172320
 KDPDR1: 172322
 KDPDR2: 172324
 KDPDR3: 172326
 KDPDR4: 172330
 KDPDR5: 172332
 KDPDR6: 172334
 KDPDR7: 172336
 KIPAR0: 172340
 KIPAR1: 172342
 KIPAR2: 172344
 KIPAR3: 172346
 KIPAR4: 172350
 KIPAR5: 172352

;SUPERVISOR I-SPACE PAGE DESCRIPTOR REGISTER ADDRESSES

;SUPERVISOR D-SPACE PAGE DESCRIPTOR REGISTER ADDRESSES

;SUPERVISOR I-SPACE PAGE ADDRESS REGISTER ADDRESSES

;SUPERVISOR D-SPACE PAGE ADDRESS REGISTER ADDRESSES

;KERNEL I-SPACE PAGE DESCRIPTOR REGISTER ADDRESSES

;KERNEL D-SPACE PAGE DESCRIPTOR REGISTER ADDRESSES

;KERNEL I-SPACE PAGE ADDRESS REGISTER ADDRESSES

```

003326 172354 KIPAR6: 172354
003330 172356 KIPAR7: 172356
003332 172360 KDPAR0: 172360 ;KERNEL D-SPACE PAGE ADDRESS REGISTER ADDRESSES
003334 172362 KDPAR1: 172362
003336 172364 KDPAR2: 172364
003340 172366 KDPAR3: 172366
003342 172370 KDPAR4: 172370
003344 172372 KDPAR5: 172372
003346 172374 KDPAR6: 172374
003350 172376 KDPAR7: 172376
003350 ADREND= .-2
    
```

```

003352 177600 PDRTAB: 177600 ;STARTING ADDRESSES OF PDR'S FOR EACH MODE
003354 172200 172200
003356 172300 172300
003360 177640 PARTAB: 177640 ;STARTING ADDRESSES OF PAR'S FOR EACH MODE
003362 172240 172240
003364 172340 172340
    
```

```

003366 003252 STATAB: KIPDR0 ;ADDRESS OF KERNEL TABLE OF PDR'S AND PAR'S
003370 000000 0
003372 003152 SIPDR0 ;ADDRESS OF SUPERVISOR TABLE OF PDR'S AND PAR'S
003374 040000 40000
003376 003052 UIPDR0 ;ADDRESS OF USER TABLE OF PDR'S AND PAR'S
003400 140000 STAEND: 140000
    
```

```

003402 000000 STAPNT: 0
003404 000000 SAVEA: 0
003406 000000 SAVEB: 0
003410 000250 KTVEC: 250 ;KT11-C TRAP AND ABORT VECTOR ADDRESS
003412 000252 KTSTA: 252
003414 177770 UBRK: 177770 ;MICROBREAK REGISTER ADDRESS
003416 177770 MSKB: 177770
003420 000000 TESTCT: 0 ;INDICATES NUMBER OF CURRENT TEST
    
```

```

003422 005037 177776 ;SET UP FOR START OF BASIC LOGIC TESTS
003426 012706 001000 START: CLR @#PS ;INITIALIZE STATUS
003432 012737 040000 177776 MOV #KSTACK, SP ;SETUP KERNEL STACK
003440 012706 002000 MOV #40000, @#PS ;INITIALIZE SUPERVISOR STACK
003444 012737 140000 177776 MOV #SSTACK, SP
003452 012706 003000 MOV #140000, @#PS ;INITIALIZE USER STACK
003456 005037 177776 MOV #USTACK, SP
003462 012767 002000 011514 CLR @#PS ;INITIALIZE ITERATION COUNT
003470 012767 003510 011512 MOV #2000, ICOUNT ;SETUP SCOPE AND ITERATION LOOP RETURN
003476 012767 000001 177714 MOV #TEST1+2, RETURN ;INITIALIZE TEST COUNTER
003504 000401 BR #1, TESTCT ;SKIP SCOPE INSTRUCTION
        .+4
    
```

```

003506 104400 ;SHOW THAT INIT CLEARS SRO<9,11-15>
003510 012737 000001 177570 TEST1: SCOPE
003516 005037 177776 MOV #1, @#SR ;DISPLAY TEST NUMBER
003522 012706 001000 CLR @#PS ;INITIALIZE PROCESSOR STATUS
003526 005077 177314 MOV #KSTACK, SP ;INITIALIZE KERNEL STACK POINTER
003532 005077 177274 CLR @SR3 ;INITIALIZE SR3
        CLR @SRO ;INITIALIZE SRO
    
```



```

004160 000000          ADR3:  HALT          ;NO PL ABORT OCCURRED
004162 005037 177776          CLR          @#PS          ;RESTORE STATUS
004166 000415          BR          DONE3
004170 022777 040201 176634 RET3:  CMP          #40201,@SR0      ;CHECK SR0
004176 001401          BEQ          .+4
004200 104006          HLT
;SR0 INCORRECT - SHOULD SHOW
;REFERENCE TO KERNEL I-SPACE 0.
;INSTRUCTION COMPLETE SHOULD BE SET,
;AND PL ABORT SHOULD BE SET
;WAIT FOR ANY TTY OUTPUT TO FINISH

004202 105777 176610          TSTB         @TCSR
004206 100375          BPL          .-4
004210 000005          RESET
004212 005777 176614          TST          @SR0
004216 001401          BEQ          .+4
004220 104006          HLT
;SR0 INCORRECT AFTER INIT
004222 005077 176604          CLR          @SR0          ;REINITIALIZE SR0
004226 016777 177160 177154 DONE3:  MOV          KTSTA,@KTVEC
004234 012737 000016 000014          MOV          #16,@#14      ;RESTORE T-BIT TRAP CATCHER

;SHOW THAT INIT CLEARS SR0(0-6)
;REFERENCE NR USER PAGE 7 D-SPACE TO SET ALL BITS(0-6)
;THEN ISSUE INIT
TEST4:  SCOPE
004242 104400          MOV          #4,@#SR
004244 012737 000004 177570          CLR          @#PS
004252 005037 177776          MOV          #KSTACK,SP
004256 012706 001000          CLR          @SR3
004262 005077 176560          CLR          @SR0
004266 005077 176540          CMP          TESTCT,#4
004272 026727 177122 000004          BEQ          .+4
004300 001401          HLT
;DISPLAY TEST NUMBER
;INITIALIZE PROCESSOR STATUS
;INITIALIZE KERNEL STACK POINTER
;INITIALIZE SR3
;INITIALIZE SR0
;IS THIS TEST BEING EXECUTED IN THE
;CORRECT SEQUENCE? - BRANCH IF YES
;TEST EXECUTED OUT OF ORDER - TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

004304 004767 010234          JSR          %7,RWALL
004310 012777 077407 176572          MOV          #77407,@JUDPDR7
004316 012777 007600 177024          MOV          #7600,@KDPAR7
;MAP ALL PAGES INITIALLY RW,4K,BANK 0
;MAKE USER 7 D-SPACE NR
;MAP KERNEL 7 D-SPACE TO THE.
;EXTERNAL BANK
;ENABLE ALL D-SPACES
;SETUP ABORT RETURN

004324 012777 000007 176514          MOV          #7,@SR3
004332 012777 004370 177050          MOV          #RET4,@KTVEC
004340 005077 177046          CLR          @KTSTA
004344 012737 140000 177776          MOV          #140000,@#PS
004352 012706 003000          MOV          #USTACK,R6
004356 005277 176450          INC          @SR0
004362 005737 160000          TST          @#160000
004366 000777          BR          .
;SET MODE TO USER
;SETUP USER STACK IN CASE NEEDED
;TURN ON KT11-C
;REFERENCE NR PAGE 7
;NO ABORT ON NR REFERENCE TO USER D-SPACE
;PAGE 7
;CHECK SR0

004370 022777 100177 176434 RET4:  CMP          #100177,@SR0
004376 001401          BEQ          .+4
004400 104006          HLT
;SR0 INCORRECT - SHOULD HAVE LOCKED
;ON NR REFERENCE TO USER 7 D-SPACE
;WAIT FOR ANY TTY OUPUT TO FINISH

004402 105777 176410          TSTB         @TCSR
004406 100375          BPL          .-4
004410 000005          RESET
004412 005777 176414          TST          @SR0
004416 001401          BEQ          .+4
004420 104006          HLT
;ISSUE INIT
;CHECK SR0
;SR0 INCORRECT AFTER INIT
    
```

```

004422 005077 176404          CLR      @SR0          ;REINITIALIZE SR0
004426 012767 000010 010550  MOV      #10,ICOUNT   ;DROP ITERATION COUNT
004434 016777 176752 176746  MOV      KTSTA,@KTVEC

; SHOW THAT BYTE ADDRESSING OF SR0 WORKS
004442 104400          TESTS:  SCOPE
004444 012737 000005 177570  MOV      #5,@#SR      ;DISPLAY TEST NUMBER
004452 005037 177776          CLR      @#PS         ;INITIALIZE PROCESSOR STATUS
004456 012706 001000          MOV      #KSTACK,SP   ;INITIALIZE KERNEL STACK POINTER
004462 005077 176360          CLR      @SR3         ;INITIALIZE SR3
004466 005077 176340          CLR      @SR0         ;INITIALIZE SR0
004472 026727 176722 000005  CMP      TESTCT,#5    ;IS THIS TEST BEING EXECUTED IN THE
004500 001401          BEQ      .+4         ;CORRECT SEQUENCE? - BRANCH IF YES
004502 104006          HLT                    ;TEST EXECUTED OUT OF ORDER - TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

004504 004767 010034          JSR      %7,RWALL     ;MAP ALL PAGES RW,4K,BANK 0
004510 012777 007600 176612  MOV      #7600,@KIPAR7 ;MAP KERNEL 7 TO THE EXTERNAL BANK
004516 012777 170001 176306  MOV      #170001,@SR0 ;TURN ON KT11-C AND SET ERROR FLAGS
004524 105077 176302          CLR      @SR0         ;DATOB (LOW) TO SR0
004530 022777 170000 176274  CMP      #170000,@SR0 ;CHECK SR0
004536 001401          BEQ      .+4
004540 104006          HLT                    ;SR0 INCORRECT AFTER CLRB (LOW)
004542 012777 170001 176262  MOV      #170001,@SR0
004550 105077 176260          CLR      @SR0H       ;DATOB (HIGH) TO SR0
004554 017701 176252          MOV      @SR0,R1     ;SAVE CONTENTS OF SR0
004560 005077 176246          CLR      @SR0         ;TURN OFF KT11-C
004564 022701 000001          CMP      #1,R1       ;CHECK SAVED CONTENTS OF SR0
004570 001401          BEQ      .+4
004572 104006          HLT                    ;SR0 INCORRECT AFTER DATOB
                                ;(SEE CONTENTS SAVED IN R1)

; SHOW THAT SR0 <1-3> TRACK THE PAGE REFERENCED IF
; KT11-C IS ON AND THE REFERENCE IS NOT TO A KT11-C REGISTER
; SHOW THAT EACH VALUE IS CORRECTLY "LOCKED" IN SR0 AFTER AN ABORT
004574 104400          TEST6: SCOPE
004576 012737 000006 177570  MOV      #6,@#SR      ;DISPLAY TEST NUMBER
004604 005037 177776          CLR      @#PS         ;INITIALIZE PROCESSOR STATUS
004610 012706 001000          MOV      #KSTACK,SP   ;INITIALIZE KERNEL STACK POINTER
004614 005077 176226          CLR      @SR3         ;INITIALIZE SR3
004620 005077 176206          CLR      @SR0         ;INITIALIZE SR0
004624 026727 176570 000006  CMP      TESTCT,#6    ;IS THIS TEST BEING EXECUTED IN THE
004632 001401          BEQ      .+4         ;CORRECT SEQUENCE? - BRANCH IF YES
004634 104006          HLT                    ;TEST EXECUTED OUT OF ORDER - TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

004636 004767 007702          JSR      %7,RWALL     ;INITIALLY SET ALL PAGES 4K,RW,BANK 0
004642 012777 007600 176500  MOV      #7600,@KOPAR7 ;MAP KERNEL 7 D-SPACE TO THE EXTERNAL BANK
004650 012777 000006 176170  MOV      #6,@SR3      ;ENABLE KERNEL AND SUPERVISOR D-SPACES
004656 012777 004734 176524  MOV      #RET6,@KTVEC ;SET UP ABORT RETURN
004664 005077 176522          CLR      @KTSTA
004670 016701 176276          MOV      SDPDR,R1    ;LOAD R1 WITH THE ADDRESS OF THE FIRST
                                ;SUPERVISOR D-SPACE PDR
004674 005002          CLR      R2          ;R2 WILL BE USED TO ADDRESS THE NR PAGE
004676 012703 100061          MOV      #100061,R3  ;R3 CONTAINS THE EXPECTED CONTENTS OF SR0
004702 012704 000010          MOV      #10,R4      ;R4 IS A COUNTER
    
```

```

004706 012711 077400 LOOP6: MOV #77400,@R1 ;MAP SUPERVISOR PDR BEING TESTED 4K,NR
004712 012737 040000 177776 MOV #40000,@#PS ;SET MODE TO SUPERVISOR
004720 005277 176106 INC @SR0 ;TURN ON KT11-C
004724 005712 TST @R2 ;ADDRESS NON-RESIDENT PAGE
004726 000777 BR . ;REFERENCE TO NR PAGE DIDN'T ABORT
004730 000005 RESET ;AFTER ERROR, TURN OFF KT11-C
004732 000416 BR DONE6
004734 017705 176072 RET6: MOV @SR0,R5 ;SAVE CONTENTS OF SR0
004740 005077 176066 CLR @SR0 ;TURN OFF KT11-C
004744 020503 CMP R5,R3 ;CHECK SAVED CONTENTS
;OF SR0 (IN R5)

004746 001401 BEQ .+4
004750 104006 HLT ;SR0 INCORRECT AFTER NR ABORT
;R3 CONTAINS THE EXPECTED CONTENTS
;R5 CONTAINS THE ACTUAL CONTENTS

004752 022626 CMP (R6)+,(R6)+ ;RESTORE STACK POINTER
004754 005721 TST (R1)+ ;MOVE POINTER TO ADDRESS NEXT PDR
004756 062703 000002 ADD #2,R3 ;CHANGE R3 TO EXPECTED CONTENTS OF SR0
;AFTER A NR REFERENCE TO THE NEXT PAGE
004762 062702 020000 ADD #20000,R2 ;CHANGE R2 TO ADDRESS THE NEXT SUPERVISOR PAGE
004766 077431 SOB R4,LOOP6 ;CHECK REFERENCE TO ALL SUPERVISOR D-SPACE PAGES
004770 016777 176416 176412 DONE6: MOV KTSTA,@KTVEC ;RESTORE TRAP CATCHER
004776 005077 176410 CLR @KTSTA

;SHOW THAT SR0 <4> TRACKS PAGE REFERENCES (I-SPACE VS. D-SPACE) IF
;KT11-C IS ON AND REFERENCE IS NOT TO A KT11-C REGISTER
;SHOW THAT EACH VALUE IS CORRECTLY "LOCKED"
;IN SR0 AFTER AN ABORT
005002 104400 TEST7: SCOPE
005004 012737 000007 177570 MOV #7,@#SR ;DISPLAY TEST NUMBER
005012 005037 177776 CLR @#PS ;INITIALIZE PROCESSOR STATUS
005016 012706 001000 MOV #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
005022 005077 176020 CLR @SR3 ;INITIALIZE SR3
005026 005077 176000 CLR @SR0 ;INITIALIZE SR0
005032 026727 176362 000007 CMP TESTCT,#7 ;IS THIS TEST BEING EXECUTED IN THE
005040 001401 BEQ .+4 ;CORRECT SEQUENCE? - BRANCH IF YES
005042 104006 HLT ;TEST EXECUTED OUT OF ORDER - TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

005044 004767 007474 JSR %7,RWALL ;INITIALLY MAKE ALL PAGES 4K,RW,BANK 0
005050 012777 007600 176272 MOV #7600,@KDPAR7 ;MAP KERNEL D-SPACE PAGE 7 TO THE EXTERNAL BANK
005056 012777 077400 176170 MOV #77400,@KIPDR1 ;MAKE KERNEL PAGE 1 NR
005064 012777 077400 176202 MOV #77400,@KDPDR1
005072 012777 005126 176310 MOV #RET7A,@KTVEC ;SETUP ABORT RETURN
005100 012777 000004 175740 MOV #4,@SR3 ;ENABLE KERNEL D-SPACE
005106 005277 175720 INC @SR0 ;TURN ON KT11-C
005112 005737 020000 TST @#20000 ;REFERENCE KERNEL 1 D-SPACE
;SHOULD ABORT SINCE IT'S MAPPED NR
;TURN OFF KT11-C
;REFERENCE TO NR PAGE DIDN'T ABORT

005116 005077 175710 CLR @SR0
005122 104006 HLT
005124 000435 BR DONE7
005126 017701 175700 RET7A: MOV @SR0,R1 ;SAVE CONTENTS OF SR0
005132 005077 175674 CLR @SR0 ;TURN OFF KT11-C
005136 022701 100023 CMP #100023,R1 ;CHECK SAVED CONTENTS OF SR0
005142 001401 BEQ .+4
005144 104006 HLT ;SR0 INCORRECT (SAVED IN R1) - SHOULD SHOW

```



```

005146 012777 005200 176234      MOV      #RET7B, &KTVEC      ; NR ABORT, KERNEL D-SPACE PAGE 1
005154 005277 175652              &SR0      ; SETUP NEW ABORT RETURN
005160 012707 025164              MOV      #ADR7+20000, PC    ; TURN ON KT11-C
005164 000000      ADR7:  HALT                ; CHANGE TO KERNEL PAGE 1 PC
005166 042707 160000              BIC      #160000, PC        ; NR FETCH FROM THIS ADDRESS SHOULD ABORT
005172 005077 175634              CLR      &SR0              ; BESTORE TO BANK 0 PC
005176 000410      BR      DONE7              ; TURN OFF KT11-C
005200 017701 175626      RET7B:  MOV      &SR0, R1    ; SAVE CONTENTS OF SR0
005204 005077 175622              CLR      &SR0              ; TURN OFF KT11-C
005210 022701 100003              CMP      #100003, R1        ; CHECK SAVED CONTENTS OF SR0
005214 001401              BEQ      .+4
005216 104006              HLT
                                ; SR0 INCORRECT - (CONTENTS SAVED IN R1)
                                ; SHOULD SHOW NR ABORT, KERNEL PAGE 1 I-SPACE
005220 016777 176166 176162  DONE7:  MOV      KTSTA, &KTVEC    ; RESTORE TRAP CATCHER

; SHOW THAT SR0 <5-6> TRACK PAGE REFERENCED (MODE) IF
; KT11-C IS ON AND THE REFERENCE IS NOT TO A KT11-C REGISTER
; SHOW THAT EACH VALUE IS CORRECTLY "LOCKED" IN SR0 AFTER AN ABORT
005226 104400      TEST10: SCOPE
005230 012737 000010 177570      MOV      #10, &#SR          ; DISPLAY TEST NUMBER
005236 005037 177776              CLR      &#PS              ; INITIALIZE PROCESSOR STATUS
005242 012706 001000              MOV      #KSTACK, SP       ; INITIALIZE KERNEL STACK POINTER
005246 005077 175574              CLR      &SR3              ; INITIALIZE SR3
005252 005077 175554              CLR      &SR0              ; INITIALIZE SR0
005256 026727 176136 000010      CMP      TESTCT, #10        ; IS THIS TEST BEING EXECUTED IN THE
005264 001401              BEQ      .+4                ; CORRECT SEQUENCE? - BRANCH IF YES
005266 104006              HLT                          ; TEST EXECUTED OUT OF ORDER - TESTCT
                                ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

005270 004767 007250      JSR      %?7, RWALL         ; MAP ALL PAGES RW 4K, BANK 0
005274 012777 007600 176046      MOV      #7600, &KDPAR7     ; MAP KERNEL 7 D-SPACE TO EXTERNAL BANK
005302 012777 077400 175764      MOV      #77400, &KDPR1     ; MAKE PAGE 1 IN EACH MODE'S D-SPACE
005310 012777 077400 175656      MOV      #77400, &SDPR1     ; NON-RESIDENT
005316 012777 077400 175550      MOV      #77400, &JDPDR1
005324 012777 005360 176056      MOV      #RET10A, &KTVEC    ; SETUP ABORT RETURN
005332 012777 000007 175506      MOV      #7, &SR3          ; ENABLE ALL D-SPACES
005340 005277 175466              INC      &SR0              ; TURN ON KT11-C
005344 005737 020000              TST      &#20000           ; REFERENCE KERNEL PAGE 1 D-SPACE (NR) - SHOULD ABORT
005350 005077 175456              CLR      &SR0              ; TURN OFF KT11-C
005354 104006              HLT                          ; NR REFERENCE DIDN'T ABORT
005356 000464      BR      DONE10
005360 017701 175446      RET10A: MOV      &SR0, R1    ; SAVE SR0 CONTENTS IN R1
005364 005077 175442              CLR      &SR0              ; TURN OFF KT11-C
005370 022701 100023              CMP      #100023, R1        ; CHECK SAVED CONTENTS OF SR0
005374 001401              BEQ      .+4
005376 104006              HLT                          ; SR0 INCORRECT (CONTENTS SAVED IN R1) - SHOULD SHOW NR
                                ; ERROR, KERNEL D-SPACE PAGE 1
005400 012777 005434 176002      MOV      #RET10B, &KTVEC    ; SETUP NEXT ABORT RETURN
005406 012737 040000 177776      MOV      #40000, &#PS       ; CHANGE MODE TO SUPERVISOR
005414 005277 175412              INC      &SR0              ; TURN ON KT11-C
005420 005737 020000              TST      &#20000           ; REFERENCE SUPERVISOR PAGE 1
                                ; D-SPACE (NR)-SHOULD ABORT
005424 005077 175402              CLR      &SR0              ; TURN OFF KT11-C
005430 104006              HLT                          ; NR REFERENCE DIDN'T ABORT
005432 000436      BR      DONE10
    
```

```

005434 017701 175372      RET108: MOV      @SRO,R1      ;SAVE CONTENTS OF SRO
005440 005077 175366      CLR      @SRO      ;TURN OFF KT11-C
005444 022701 100063      CMP      #100063,R1 ;CHECK SAVED CONTENTS OF SRO
005450 001401      BEQ      .+4
005452 104006      HLT
;SRO INCORRECT (CONTENTS SAVED IN R1) - SHOULD SHOW NR
;ERROR, SUPERVISOR D-SPACE PAGE 1

005454 012777 005510 175726      MOV      #RET10C,@KTVEC ;SETUP NEXT ABORT RETURN
005462 012737 140000 177776      MOV      #140000,@#PS  ;CHANGE MODE TO USER
005470 005277 175336      INC      @SRO      ;TURN ON KT11-C
005474 005737 020000      TST      @#20000    ;REFERENCE USER PAGE 1 D-SPACE (NR)
005500 005077 175326      CLR      @SRO      ;TURN OFF KT11-C
005504 104006      HLT              ;NR REFERENCE DIDN'T ABORT
005506 000410      BR
005510 017701 175316      RET10C: MOV      @SRO,R1      ;SAVE CONTENTS OF SRO
005514 005077 175312      CLR      @SRO      ;TURN OFF KT11-C
005520 022701 100163      CMP      #100163,R1 ;CHECK SAVED CONTENTS OF SRO
005524 001401      BEQ      .+4
005526 104006      HLT
;SRO INCORRECT - SHOULD SHOW NR
;ERROR, USER D-SPACE PAGE 1
;(CONTENTS SAVED IN R1)

005530 016777 175656 175652  DONE10: MOV      KTSTA,@KTVEC ;RESTORE TRAP CATCHER

;SHOW THAT SRO <1-6> DOESN'T TRACK IF KT11-C IS OFF OR IF REFERENCE IS TO
;AN INTERNAL (KT11-C) REGISTER
TEST11: SCOPE
005536 104400      MOV      #11,@#SR    ;DISPLAY TEST NUMBER
005540 012737 000011 177570      CLR      @#PS      ;INITIALIZE PROCESSOR STATUS
005546 005037 177776      MOV      #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
005552 012706 001000      CLR      @SR3      ;INITIALIZE SR3
005556 005077 175264      CLR      @SRO      ;INITIALIZE SRO
005562 005077 175244      CMP      TESTCT,#11 ;IS THIS TEST BEING EXECUTED IN THE
005566 026727 175626 000011      BEQ      .+4        ;CORRECT SEQUENCE? - BRANCH IF YES
005574 001401      HLT              ;TEST EXECUTED OUT OF ORDER - TESTCT
005576 104006      ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

005600 004767 006740      JSR      %7,RWALL   ;SET ALL PAGES RW, 4K, BANK 0
005604 012777 007600 175436      MOV      #7600,@SDPAR7 ;MAP SUPERVISOR 7 D-SPACE TO THE
;EXTERNAL BANK
005612 012777 000007 175226      MOV      #7,@SR3    ;ENABLE ALL D-SPACES
005620 012737 040000 177776      MOV      #40000,@#PS ;SET MODE TO SUPERVISOR
005626 005277 175200      INC      @SRO      ;TURN ON KT11-C
005632 042777 000001 175172      BIC      #1,@SRO    ;TURN OFF KT11-C - SHOULD
;NOT TRACK REFERENCE TO SUPERVISOR
;D-SPACE 7 WHICH IS AN INTERNAL
;REFERENCE (TO SRO)
;CHANGE TO KERNEL MODE
;CHECK SRO

005640 005037 177776      CLR      @#PS
005644 022777 000060 175160      CMP      #60,@SRO
005652 001401      BEQ      .+4
005654 104006      HLT
;SRO INCORRECT - SHOULD SHOW REFERENCE
;TO SUPERVISOR 0 D-SPACE
;IF IT SHOWS SUPERVISOR 7 D-SPACE
;IT TRACKED THE INTERNAL REFERENCE
;IF IT SHOWS KERNEL 0, IT IS
;TRACKING WITH KT11-C OFF

005656 005077 175150      CLR      @SRO
005662 005077 175160      CLR      @SR3

```

;SHOW THAT ALL BITS IN SR2 WORK BY ROTATING A BIT THRU SR2. MAP USER NR
 ;(ALL PAGES) THEN SET UP THE DESIRED VALUE FOR SR2 ON THE KERNEL
 ;STACK AND RTI TO USER. THIS SHOULD GIVE A NR ABORT WITH THE DESIRED
 ;VALUE IN SR2

```

005666 104400
005670 012737 000012 177570
005676 005037 177776
005702 012706 001000
005706 005077 175134
005712 005077 175114
005716 026727 175476 000012
005724 001401
005726 104006
    TEST12: SCOPE
            MOV #12, @#SR
            CLR @#PS
            MOV #KSTACK, SP
            CLR @SR3
            CLR @SR0
            CMP TESTCT, #12
            BEQ .+4
            HLT
    ;DISPLAY TEST NUMBER
    ;INITIALIZE PROCESSOR STATUS
    ;INITIALIZE KERNEL STACK POINTER
    ;INITIALIZE SR3
    ;INITIALIZE SR0
    ;IS THIS TEST BEING EXECUTED IN THE
    ;CORRECT SEQUENCE? - BRANCH IF YES
    ;TEST EXECUTED OUT OF ORDER - TESTCT
    ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
    
```

```

005730 004767 006560
005734 012777 077406 175310
005742 012777 007600 175360
005750 012777 077406 175312
005756 012700 000002
005762 012777 006012 175420
005770 005077 175416
005774 012746 140000
006000 010046
006002 005277 175024
006006 000002
    LOOP12: MOV #140000, -(R6)
            MOV RO, -(R6)
            INC @SR0
            RTI
    ;INITIALLY CLEAR ALL KT11-C REGISTERS
    ;MAP KERNEL 0 I-SPACE 4K, RW, BANK 0
    ;MAP KERNEL 7 I-SPACE 4K, RW,
    ;EXTERNAL BANK
    ;SETUP FIRST VALUE TO BE CHECKED
    ;SETUP ABORT RETURN
    ;PUSH USER MODE ON STACK
    ;PUSH VALUE TO BE CHECKED ON STACK
    ;TURN ON KT11-C
    ;POP STACK - NEW PC SHOULD
    ;GIVE NR ABORT
    ;SHOULDN'T BE ANY WAY TO ARRIVE HERE
    ;TURN OFF KT11-C
    ;CHECK VALUE IN SR2
    
```

```

006010 000777
006012 042777 000001 175012
006020 020077 175016
006024 001401
006026 104006
    RET12: BR .
            BIC #1, @SR0
            CMP RO, @SR2
            BEQ .+4
            HLT
    ;SR2 INCORRECT - SHOULD CONTAIN
    ;THE PC POPPED OFF THE STACK
    ;WHOSE VALUE IS IN RO
    
```

```

006030 005077 174776
006034 022626
006036 006300
006040 103355
006042 016777 175344 175340
    CLR @SR0
    CMP (R6)+, (R6)+
    ASL RO
    BCC LOOP12
    MOV KTSTA, @KTVEC
    ;RESTORE STACK POINTER
    ;SHIFT BIT TO BE TESTED
    ;BRANCH IF NOT DONE
    ;RESTORE TRAP CATCHER
    
```

;SHOW THAT A MEMORY MANAGEMENT TRAP CONDITION WILL NOT CAUSE A TRAP IF
 ;THE MANAGEMENT TRAP FLAG IN SR0 IS ALREADY SET
 ;SHOW THAT HAVING THE ABORT ERROR
 ;BITS SET WILL NOT PREVENT A MEMORY MANAGEMENT TRAP

```

006050 104400
006052 012737 000013 177570
006060 005037 177776
006064 012706 001000
006070 005077 174752
006074 005077 174732
006100 026727 175314 000013
006106 001401
006110 104006
    TEST13: SCOPE
            MOV #13, @#SR
            CLR @#PS
            MOV #KSTACK, SP
            CLR @SR3
            CLR @SR0
            CMP TESTCT, #13
            BEQ .+4
            HLT
    ;DISPLAY TEST NUMBER
    ;INITIALIZE PROCESSOR STATUS
    ;INITIALIZE KERNEL STACK POINTER
    ;INITIALIZE SR3
    ;INITIALIZE SR0
    ;IS THIS TEST BEING EXECUTED IN THE
    ;CORRECT SEQUENCE? - BRANCH IF YES
    ;TEST EXECUTED OUT OF ORDER - TESTCT
    ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
    
```

```

006112 004767 006426 JSR %7,RWALL ;INITIALIZE ALL PAGES RW,4K, BANK 0
006116 012777 077405 175132 MOV #77405,@KIPDR2 ;SET KERNEL PAGE 2 RRWTW,4K
006124 012777 007600 175176 MOV #7600,@KIPAR7 ;MAP KERNEL PAGE 7 TO EXTERNAL BANK
006132 012777 006174 175250 MOV #RET13A,@KTVEC ;SETUP MEMORY MANAGEMENT ABORT RETURN
006140 005077 175246 CLR @KTSTA
006144 005277 174662 INC @SRO ;TURN ON KT11-C
006150 012777 161001 174654 MOV #161001,@SRO ;ENABLE MEMORY MANAGEMENT TRAPPING
;AND SET THE ABORT ERROR BITS
;WRITE KERNEL PAGE 2 (RRWTW)-SHOULD TRAP

006156 013737 007000 047000 MOV @#7000,@#47000
006164 005077 174642 CLR @SRO
006170 104006 HLT ;NO TRAP OCCURRED
006172 000416 BR
006174 022626 RET13A: CMP (SP)+,(SP)+ ;RESTORE THE STACK POINTER
006176 017701 174630 MOV @SRO,R1 ;SAVE CONTENTS OF SRO
006202 005077 174624 CLR @SRO ;TURN OFF KT11-C
006206 022701 171001 CMP #171001,R1 ;CHECK SAVED CONTENTS OF SRO
006212 001401 BEQ .+4
006214 104006 HLT ;SAVED CONTENTS OF SRO INCORRECT (CONTAINED IN R1)
006216 022777 077705 175032 CMP #77705,@KIPDR2 ;CHECK THE PDR CORRESPONDING TO THE TRAP REFERENCE
006224 001401 BEQ .+4
006226 104006 HLT ;THE PDR CORRESPONDING TO THE TRAP REFERENCE
;IS INCORRECT
;SETUP MEMORY MANAGEMENT TRAP RETURN
;TURN ON KT11-C, ENABLE MMGT TRAPPING,
;AND SET MEMORY MANAGEMENT TRAP

006230 012777 006314 175152 CONT13: MOV #RET13B,@KTVEC
006236 012777 011001 174566 MOV #11001,@SRO
;CLEAR A AND W BITS
;WRITE KERNEL PAGE 2 (RRWTW) -
;SHOULDN'T TRAP SINCE MEMORY MANAGEMENT
;TRAP HASN'T BEEN CLEARED YET
;TURN OFF KT11-C
;CHECK SRO

006244 012777 077405 175004 MOV #77405,@KIPDR2
006252 013737 007000 047000 MOV @#7000,@#47000
;SRO INCORRECT-SHOULD SHOW MEMORY MANAGEMENT TRAP,
;MMGT ENABLE, AND PAGE 0 I-SPACE
;CHECK PDR CORRESPONDING TO THE TRAP REFERENCE

006260 042777 000001 174544 BIC #1,@SRO
006266 022777 011000 174536 CMP #11000,@SRO
006274 001401 BEQ .+4
006276 104006 HLT ;PDR CORRESPONDING TO THE TRAP REFERENCE IS INCORRECT

006300 022777 077705 174750 CMP #77705,@KIPDR2
006306 001401 BEQ .+4
006310 104006 HLT
006312 000405 BR
006314 022626 RET13B: CMP (SP)+,(SP)+ ;RESTORE THE STACK POINTER
006316 042777 000001 174506 BIC #1,@SRO ;TURN OFF KT11-C
006324 104006 HLT ;MMGT TRAP ACCESS TRAPPED BEFORE
;PREVIOUS MMGT TRAP WAS CLEARED
;RESTORE MEMORY MANAGEMENT TRAP RETURN
;TO CAUSE A HALT ON A FALSE TRAP OR ABORT

006326 016777 175060 175054 DONE13: MOV KTSTA,@KTVEC
006334 005077 174472 CLR @SRO ;REINITIALIZE SRO

;IF MEMORY MANAGEMENT ENABLE IS SET WITH AN ATTENTION (A) BIT ALREADY SET, NO TRAP
;WILL OCCUR UNTIL ANOTHER MEMORY MANAGEMENT FAULT OCCURS

006340 104400 TEST14: SCOPE
006342 012737 000014 177570 MOV #14,@#SR ;DISPLAY TEST NUMBER
006350 005037 177776 CLR @#PS ;INITIALIZE PROCESSOR STATUS
006354 012706 001000 MOV #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
006360 005077 174462 CLR @SR3 ;INITIALIZE SR3
006364 005077 174442 CLR @SRO ;INITIALIZE SRO
006370 026727 175024 000014 CMP TESTCT,#14 ;IS THIS TEST BEING EXECUTED IN THE

```

```

006376 001401      BEQ      .+4      ;CORRECT SEQUENCE? - BRANCH IF YES
006400 104006      HLT
;TEST EXECUTED OUT OF ORDER - TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

006402 004767 006136      JSR      %7,RWALL      ;MAP ALL PAGES RW, 4K, BANK 0
006406 012777 007600 174714      MOV      #7600,@KIPAR7 ;MAP KERNEL 7 I-SPACE TO THE EXTERNAL BANK
006414 012777 077404 174632      MOV      #77404,@KIPDR1 ;MAP KERNEL 1 I-SPACE RW
006422 012777 006472 174760      MOV      #ERR14,@KTVEC ;SETUP ERROR RETURN
006430 005077 174756      CLR      @KTSTA
006434 012777 000001 174370      MOV      #1,@SR0      ;TURN ON KT11-C (DON'T ENABLE TRAPPING)
006442 005737 030000      TST     @#30000      ;SET THE A BIT IN KERNEL 1 I-SPACE
006446 012777 001001 174356      MOV      #1001,@SR0   ;SET MEMORY MANAGEMENT ENABLE AND
;CLEAR PREVIOUS MANAGEMENT TRAP FLAG
006454 012777 006516 174726      ADD14:  MOV      #RET14,@KTVEC ;SETUP TRAP RETURN
006462 005737 030000      TST     @#30000      ;ACCESS RWRT PAGE WITH TRAP ENABLE SET-SHOULD TRAP
006466 000000      ADR14:  HALT        ;NO MEMORY MANAGEMENT TRAP ON REFERENCING
;KERNEL 1 I-SPACE MAPPED READ-WRITE AND TRAP
;WITH MGMT TRAP ENABLE SET
006470 000426      BR      DONE14      ;TURN OFF KT11-C AFTER ERROR
006472 042777 000001 174332      ERR14:  BIC      #1,@SR0      ;CHECK PC AT TIME OF ERROR
006500 022716 006454      CMP     #ADD14,(SP)
006504 001002      BNE
006506 104006      HLT
;MEMORY MANAGEMENT TRAP OCCURRED ON SETTING MEMORY
;MANAGEMENT ENABLE WITH AN A BIT ALREADY SET

006510 000416      BR      DONE14
006512 104006      HLT
;MEMORY MANAGEMENT TRAP OCCURED AT THE WRONG TIME
;CHECK PC ON STACK

006514 000414      BR      DONE14
006516 042777 000001 174306      RET14:  BIC      #1,@SR0
006524 022716 006466      CMP     #ADR14,(SP) ;CHECK PC ON STACK
006530 001401      BEQ
006532 104006      HLT
;INCORRECT PC ON STACK AFTER MEMORY
;MANAGEMENT TRAP
006534 022777 011000 174270      CMP     #11000,@SR0 ;CHECK SR0
006542 001401      BEQ
006544 104006      HLT
;SR0 INCORRECT AFTER MEMORY
;MANAGEMENT TRAP

006546 005077 174260      DONE14: CLR      @SR0
006552 016777 174634 174630      MOV      KTSTA,@KTVEC ;RESTORE TRAP CATCHER

;SHOW THAT SETTING MEMORY MANAGEMENT FAULT (SR0<12>) DOESN'T LOCK OUT
;ABORTS. SET MEMORY MANAGEMENT FAULT, THEN ACCESS A NR PAGE.
TEST15: SCOPE
006560 104400      MOV      #15,@#SR
006562 012737 000015 177570      CLR      @#PS
006570 005037 177776      MOV      #KSTACK,SP ;INITIALIZE PROCESSOR STATUS
006574 012706 001000      CLR      @SR3      ;INITIALIZE KERNEL STACK POINTER
006600 005077 174242      CLR      @SR0      ;INITIALIZE SR3
006604 005077 174222      CLR      @SR0      ;INITIALIZE SR0
006610 026727 174604 000015      CMP     TESTCT,#15 ;IS THIS TEST BEING EXECUTED IN THE
006616 001401      BEQ      .+4      ;CORRECT SEQUENCE? - BRANCH IF YES
006620 104006      HLT        ;TEST EXECUTED OUT OF ORDER - TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

006622 004767 005716      JSR      %7,RWALL      ;INITIALLY MAP ALL PAGES RW, BANK 0
006626 012777 007600 174474      MOV      #7600,@KIPAR7 ;MAP KERNEL 7 TO THE EXTERNAL BANK
006634 005077 174414      CLR      @KIPDR1
006640 012777 006676 174542      MOV      #RET15,@KTVEC ;MAP KERNEL 1 NR
;SETUP ABORT RETURN

```

```

006646 005077 174540 CLR @KTSTA
006652 012777 010001 174152 MOV #10001,@SR0 ;TURN ON KT11-C AND SET
;MEMORY MANAGEMENT FAULT
006660 005737 020000 TST @#20000 ;REFERENCE KERNEL PAGE 1-SHOULD
;ABORT SINCE IT'S MAPPED NR
006664 042777 000001 174140 BIC #1,@SR0 ;TURN OFF KT11-C IF NO ABORT
006672 104006 HLT ;NO ABORT ON REFERENCE TO NR
;KERNEL PAGE 1 AFTER SETTING
006674 000403 BR DONE15 ;MEMORY MANAGEMENT FAULT
006676 005077 174130 RET15: CLR @SR0 ;TURN OFF KT11-C AFTER ABORT
006702 022626 CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
006704 016777 174502 174476 DONE15: MOV KTSTA,@KTVEC ;RESTORE TRAP CATCHER

;SHOW THAT IF THE INSTRUCTION SETTING MEMORY MANAGEMENT TRAP ENABLE
;CAUSES A TRAP REFERENCE BEFORE SETTING ENABLE, THE TRAP WILL NOT OCCUR
;ALSO SHOW THAT MEMORY MANAGEMENT WILL NOT TRAP ON AN INTERNAL REFERENCE
006712 104400 TEST16: SCOPE
006714 012737 000016 177570 MOV #16,@#SR ;DISPLAY TEST NUMBER
006722 005037 177776 CLR @#PS ;INITIALIZE PROCESSOR STATUS
006726 012706 001000 MOV #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
006732 005077 174110 CLR @SR3 ;INITIALIZE SR3
006736 005077 174070 CLR @SR0 ;INITIALIZE SR0
006742 026727 174452 000016 CMP TESTCT,#16 ;IS THIS TEST BEING EXECUTED IN THE
006750 001401 BEQ .+4 ;CORRECT SEQUENCE? - BRANCH IF YES
006752 104006 HLT ;TEST EXECUTED OUT OF ORDER - TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

006754 004767 005564 JSR %7,RWALL ;MAP ALL PAGES 4K, RW, BANK 0
006760 012777 007600 174342 MOV #7600,@KIPAR7 ;MAP KERNEL 7 TO THE EXTERNAL BANK, RRWT
006766 012777 077404 174274 MOV #77404,@KIPDR7
006774 012777 077404 174252 MOV #77404,@KIPDR1 ;MAP KERNEL 1 RRWT, 4K
007002 012777 007100 174400 MOV #RET16,@KTVEC ;SETUP TRAP RETURN IN CASE
007010 005077 174376 CLR @KTSTA
007014 005277 174012 INC @SR0 ;TURN ON KT11-C
007020 053777 023014 174004 BIS @#K1000+20000,@SR0 ;SET MEMORY MANAGEMENT TRAP ENABLE
;AND REFERENCE A RRWT PAGE (KERNEL PAGE 1) FIRST

007026 012777 007112 174354 MOV #RET16A,@KTVEC
007034 032777 001000 173770 BIT #1000,@SR0 ;TRAP REFERENCE TO A KT11-C REGISTER
007042 001001 BNE .+4
007044 104006 HLT ;MEMORY MANAGEMENT TRAP ENABLE WASN'T SET
007046 005077 173760 CLR @SR0 ;TURN OFF KT11-C
007052 022777 077604 174174 CMP #77604,@KIPDR1 ;CHECK KERNEL PDR1
007060 001401 BEQ .+4
007062 104006 HLT ;KERNEL PDR 1 WRONG-SHOULD SHOW THAT
;PAGE WAS REFERENCED BUT NOT WRITTEN
007064 022777 077404 174176 CMP #77404,@KIPDR7 ;CHECK KERNEL PDR 7
007072 001401 BEQ .+4
007074 104006 HLT ;KERNEL PDR 7 INCORRECT-SHOULDN'T HAVE
;TRACKED REFERENCE TO KT11-C REGISTER

007076 000411 BR DONE16
007100 042777 000001 173724 RET16: BIC #1,@SR0 ;TURN OFF KT11-C
007106 104006 HLT ;TRAP OCCURRED ON THE INSTRUCTION SETTING TRAP ENABLE
007110 000404 BR DONE16
007112 042777 000001 173712 RET16A: BIC #1,@SR0
007120 104006 HLT ;TRAP OCCURRED ON A REFERENCE TO AN INTERNAL
;KT11-C REGISTER
    
```

```

007122 005077 173704 174254 DONE16: CLR  @SR0
007126 016777 174260 174254 MOV  KTSTA,@KTVEC

;SHOW THAT IF AN INSTRUCTION WHICH TURNS OFF MEMORY MANAGEMENT
;TRAPPING MAKES A TRAP REFERENCE, A TRAP WILL OCCUR
TEST17: SCOPE
007134 104400 MOV  #17,@#SR ;DISPLAY TEST NUMBER
007136 012737 000017 177570 CLR  @#PS ;INITIALIZE PROCESSOR STATUS
007144 005037 177776 MOV  #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
007150 012706 001000 CLR  @SR3 ;INITIALIZE SR3
007154 005077 173666 CLR  @SR0 ;INITIALIZE SRO
007160 005077 173646 CMP  TESTCT,#17 ;IS THIS TEST BEING EXECUTED IN THE
007164 026727 174230 000017 BEQ  .+4 ;CORRECT SEQUENCE? - BRANCH IF YES
007172 001401 HLT ;TEST EXECUTED OUT OF ORDER - TESTCT
007174 104006 ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

007176 004767 005342 JSR  %7,RWALL ;MAP ALL PAGES 4K, RW, BANK 0
007202 012777 077404 174044 MOV  #77404,@KIPDR1 ;MAP KERNEL 1 RRWT
007210 012777 007600 174112 MOV  #7600,@KIPAR7 ;MAP KERNEL 7 TO THE EXTERNAL BANK
007216 012777 007256 174164 MOV  #RET17,@KTVEC ;SETUP TRAP RETURN
007224 005077 174162 CLR  @KTSTA
007230 012777 001001 173574 MOV  #1001,@SR0 ;TURN ON KT11-C AND ENABLE MEMORY MANAGEMENT TRAPPING
007236 043777 023014 173566 BIC  @#K1000+20000,@SR0 ;PICK UP VALUE TO DO BIT CLEAR
;WITH THRU KERNEL 1(RRWT)
;CAUSES A TRAP REFERENCE WHICH SHOULD TRAP
;IF NO TRAP, TURN OFF KT11-C
;TRAP REFERENCE TO KERNEL PAGE 1 WHEN CLEARING
;TRAP ENABLE DIDN'T TRAP

007244 042777 000001 173560 BIC  #1,@SR0
007252 104006 HLT

007254 000410 BR  DONE17
007256 042777 000001 173546 RET17: BIC  #1,@SR0 ;TURN OFF KT11-C AFTER TRAP
007264 022777 010000 173540 CMP  #10000,@SR0 ;CHECK SRO
007272 001401 BEQ  .+4
007274 104006 HLT ;SRO INCORRECT AFTER MEMORY MANAGEMENT TRAP
007276 016777 174110 174104 DONE17: MOV  KTSTA,@KTVEC

;SHOW THAT SETTING PROGRAMMER'S AID SYSTEM TRAP (BIT 11) WON'T PREVENT POTENTIAL
;MEMORY MANAGEMENT TRAPS
TEST20: SCOPE
007304 104400 MOV  #20,@#SR ;DISPLAY TEST NUMBER
007306 012737 000020 177570 CLR  @#PS ;INITIALIZE PROCESSOR STATUS
007314 005037 177776 MOV  #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
007320 012706 001000 CLR  @SR3 ;INITIALIZE SR3
007324 005077 173516 CLR  @SR0 ;INITIALIZE SRO
007330 005077 173476 CMP  TESTCT,#20 ;IS THIS TEST BEING EXECUTED IN THE
007334 026727 174060 000020 BEQ  .+4 ;CORRECT SEQUENCE? - BRANCH IF YES
007342 001401 HLT ;TEST EXECUTED OUT OF ORDER - TESTCT
007344 104006 ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

007346 004767 005172 JSR  %7,RWALL ;INITIALIZE ALL PAGES RW, 4K, BANK 0
007352 012777 077404 173674 MOV  #77404,@KIPDR1 ;MAP KERNEL PAGE 1 RRWT, 4K, BANK 0
007360 012777 007600 173742 MOV  #7600,@KIPAR7 ;MAP KERNEL PAGE 7 TO THE EXTERNAL BANK
007366 012777 007424 174014 MOV  #RET20,@KTVEC ;SETUP MEMORY MANAGEMENT TRAP RETURN
007374 005077 174012 CLR  @KTSTA
007400 012777 005001 173424 MOV  #5001,@SR0 ;TURN ON KT11-C, MMGT TRAP ENABLE
;AND SET PROGRAMMER'S AID SYSTEM
;TRAP (BIT 11)
    
```

```

007406 005737 021000          TST      @#21000      ;REFERENCE KERNEL PAGE 1 (RRWT)-SHOULD TRAP
007412 042777 000001 173412  BIC      #1,@SR0      ;TURN OFF KT11-C
007420 104006          HLT                      ;SETTING SRO <11> INHIBITED
007422 000411          BR       DONE20        ;A MEMORY MANAGEMENT TRAP
007424 022626          RET20: CMP      (SP)+,(SP)+ ;RESTORE STACK POINTER
007426 042777 000001 173376  BIC      #1,@SR0      ;TURN OFF KT11-C
007434 022777 015000 173370  CMP      #15000,@SR0  ;CHECK SRO
007442 001401          BEQ                      ;
007444 104006          HLT                      ;SRO INCORRECT AFTER MEMORY
                                           ;MANAGEMENT TRAP
007446 005077 173360          DONE20: CLR     @SR0      ;INITIALIZE SRO
007452 016777 173734 173730  MOV     KTSTA,@KTVEC ;RESTORE TRAP RETURN TO CAUSE A
                                           ;HALT IN CASE OF A FALSE ABORT OR TRAP

                                           ;SETTING SRO<11> SHOULD NOT PREVENT ABORTS OR LOCK UP TRACKING
007460 104400          TEST21: SCOPE
007462 012737 000021 177570  MOV     #21,@#SR      ;DISPLAY TEST NUMBER
007470 005037 177776          CLR     @#PS          ;INITIALIZE PROCESSOR STATUS
007474 012706 001000          MOV     #KSTACK,SP   ;INITIALIZE KERNEL STACK POINTER
007500 005077 173342          CLR     @SR3         ;INITIALIZE SR3
007504 005077 173322          CLR     @SR0         ;INITIALIZE SRO
007510 026727 173704 000021  CMP     TESTCT,#21    ;IS THIS TEST BEING EXECUTED IN THE
007516 001401          BEQ                      ;CORRECT SEQUENCE? - BRANCH IF YES
007520 104006          HLT                      ;TEST EXECUTED OUT OF ORDER - TESTCT
                                           ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

007522 004767 005016          JSR     %7,RWALL      ;INITIALIZE ALL PAGES RW, 4K, BANK 0
007526 012777 077400 173520  MOV     #77400,@KIPDR1 ;MAP KERNEL PAGE 1 NR
007534 012777 007600 173566  MOV     #7600,@KIPAR7 ;MAP KERNEL PAGE 7 TO EXTERNAL BANK
007542 012777 007616 173640  MOV     #RET21,@KTVEC ;SETUP MEMORY MANAGEMENT ABORT
007550 005077 173636          CLR     @KTSTA       ;RETURN
007554 012777 004001 173250  MOV     #4001,@SR0    ;TURN ON KT11-C, AND SET PROGRAMMER'S AID
007562 000240          NOP                      ;SYSTEM TRAP (BIT 11)
007564 017701 173252          AD21:  MOV     @SR2,R1 ;CHECK TO SEE THAT SR2 IS STILL TRACKING
007570 022701 007564          CMP     #AD21,R1
007574 001401          BEQ                      ;
007576 104006          HLT                      ;SR2 NOT TRACKING AFTER SRO<11> SET
007600 005737 021000          TST     @#21000      ;READ NR PAGE-SHOULD ABORT
007604 042777 000001 173220  BIC     #1,@SR0      ;TURN OFF KT11-C IF NO ABORT
007612 104006          HLT                      ;SRO <11> SET PREVENTED NR REFERENCE FROM ABORTING
007614 000410          BR       DONE21
007616 042777 000001 173206  RET21: BIC     #1,@SR0 ;TURN OFF KT11-C
007624 022777 104002 173200  CMP     #104002,@SR0 ;CHECK SRO
007632 001401          BEQ                      ;
007634 104006          HLT                      ;SRO INCORRECT - SHOULD SHOW NR REFERENCE TO KERNEL
                                           ;PAGE 1, AND BIT 11 SHOULD STILL BE SET
007636 016777 173550 173544  DONE21: MOV     KTSTA,@KTVEC ;RESTORE TRAP CATCHER

                                           ;SHOW THAT IF THE STACK IS MAPPED TO A RRWTW PAGE AN IMPLICIT
                                           ;STACK WRITE WILL CAUSE A MEMORY MANAGEMENT TRAP TO OCCUR
007644 104400          TEST22: SCOPE
007646 012737 000022 177570  MOV     #22,@#SR      ;DISPLAY TEST NUMBER
007654 005037 177776          CLR     @#PS          ;INITIALIZE PROCESSOR STATUS
007660 012706 001000          MOV     #KSTACK,SP   ;INITIALIZE KERNEL STACK POINTER
007664 005077 173156          CLR     @SR3         ;INITIALIZE SR3
007670 005077 173136          CLR     @SR0         ;INITIALIZE SRO

```



```

007674 026727 173520 000022      CMP      TESTCT, #22      ; IS THIS TEST BEING EXECUTED IN THE
007702 001401      BEQ      .+4          ; CORRECT SEQUENCE? - BRANCH IF YES
007704 104006      HLT                               ; TEST EXECUTED OUT OF ORDER - TESTCT
                                           ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

007706 004767 004602      JSR      %7, CLRALL    ; INITIALLY CLEAR ALL KT11-C REGISTERS
007712 012777 077405 173352      MOV      #77405, @KDPDR0 ; SETUP KERNEL 0 D-SPACE RRWTW
007720 012777 007600 173422      MOV      #7600, @KDPAR7  ; MAP KERNEL 7 D-SPACE 4K, RW,
007726 012777 077406 173354      MOV      #77406, @KDPDR7 ; EXTERNAL BANK
007734 012777 077406 173310      MOV      #77406, @KIPDR0 ; SETUP KERNEL 0 I-SPACE RW
007742 012777 000004 173076      MOV      #4, @SR3       ; ENABLE KERNEL D-SPACE
007750 012777 010002 173432      MOV      #RET22, @KTVEC  ; SETUP TRAP RETURN
007756 012777 001001 173046      MOV      #1001, @SR0    ; TURN ON KT11-C AND ENABLE MEMORY
                                           ; MANAGEMENT TRAPPING
                                           ; IMPLICIT STACK WRITE TO RRWTW SPACE

007764 004767 000002      JSR      PC, ADR22
007770 000240      NOP
007772 005077 173034      ADR22:  CLR      @SR0    ; TURN OFF KT11-C
007776 104006      HLT                               ; IMPLICIT STACK WRITE OF RRWTW SPACE
010000 000425      BR      DONE22                ; VIA JSR DIDN'T TRAP
010002 042777 000001 173022      RET22:  BIC      #1, @SR0    ; TURN OFF KT11-C
010010 022706 000772      CMP      #KSTACK-6, SP    ; CHECK STACK POINTER
010014 001401      BEQ      .+4
010016 104006      HLT                               ; STACK POINTER INCORRECT - SHOULD HAVE BEEN
                                           ; PUSHED ONCE BY JSR AND TWICE BY MMGT TRAP
                                           ; CHECK SR0

010020 022777 011020 173004      CMP      #11020, @SR0
010026 001401      BEQ      .+4
010030 104006      HLT                               ; SR0 INCORRECT - SHOULD SHOW KERNEL
                                           ; I-SPACE 0, WITH TRAP ENABLE AND MMGT TRAP SET
                                           ; CHECK CONTENTS OF STACK

010032 022766 007770 000004      CMP      #ADR22-2, 4(R6)
010040 001401      BEQ      .+4
010042 104006      HLT                               ; ADDRESS OF RETURN FROM JSR NOT
                                           ; ON STACK CORRECTLY
                                           ; CHECK CONTENTS OF STACK

010044 022716 007772      CMP      #ADR22, (SP)
010050 001401      BEQ      .+4
010052 104006      HLT                               ; ADDRESS IN PC AT TIME OF
                                           ; MEMORY MANAGEMENT TRAP NOT
                                           ; CORRECTLY STORED ON STACK

010054 005077 172766 173322      DONE22: CLR      @SR3       ; DISABLE D-SPACES
010060 016777 173326      MOV      KTSTA, @KTVEC    ; RESTORE TRAP CATCHER

; TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN NOT SET)
; KERNEL PAGE 1 IS USED WITH ALL PAGE LENGTH VALUES
; SHOW THAT REFERENCES TO BOTH BOUNDARIES OF THE ALLOWED AREA DON'T TRAP OR ABORT
; SHOW THAT A REFERENCE TO THE FIRST WORD BEYOND THE ALLOWABLE AREA DOES TRAP
010066 104400      TEST23: SCOPE
010070 012737 000023 177570      MOV      #23, @#SR
010076 005037 177776      CLR      @#PS
010102 012706 001000      MOV      #KSTACK, SP
010106 005077 172734      CLR      @SR3
010112 005077 172714      CLR      @SR0
010116 026727 173276 000023      CMP      TESTCT, #23
010124 001401      BEQ      .+4
010126 104006      HLT                               ; DISPLAY TEST NUMBER
                                           ; INITIALIZE PROCESSOR STATUS
                                           ; INITIALIZE KERNEL STACK POINTER
                                           ; INITIALIZE SR3
                                           ; INITIALIZE SR0
                                           ; IS THIS TEST BEING EXECUTED IN THE
                                           ; CORRECT SEQUENCE? - BRANCH IF YES
                                           ; TEST EXECUTED OUT OF ORDER - TESTCT
                                           ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
    
```

M02

010130	004767	004410		JSR	%7,RWALL	: INITIALIZE ALL PAGES TO RW, 4K, BANK 0
010134	012777	007600	173166	MOV	#7600,@KIPAR7	: MAP KERNEL PAGE 7 TO THE EXTERNAL BANK
010142	012702	000006		MOV	#6,R2	: R2 CONTAINS VALUE TO BE LOADED IN THE
						: PDR BEING CHECKED (INCLUDING PLF)
010146	012701	020076		MOV	#20076,R1	: R1 IS USED TO REFERENCE THE TOP ADDRESS
						: WITHIN THE ALLOWED AREA
010152	012777	010232	173230	MOV	#RET23A,@KTVEC	: SETUP ABORT RETURN IN CASE REFERENCE
010160	005077	173226		CLR	@KTSTA	: WITHIN ALLOWED AREA ABORTS
010164	005277	172642		INC	@SRO	: TURN ON KT11-C
010170	010277	173060	LOOP23:	MOV	R2,@KIPDR1	: SET KERNEL PAGE 1 TO NEW PAGE LENGTH
010174	005727	020000		TST	#20000	: READ LOWER BOUNDARY-SHOULDN'T ABORT
010200	005711			TST	@R1	: READ UPPER ALLOWED BOUNDARY-SHOULDN'T
						: ABORT
010202	012777	010252	173200	MOV	#RET23B,@KTVEC	: SETUP ABORT RETURN
010210	020127	037776		CMP	R1,#37776	: CHECK FOR DONE (TO AVOID REFERENCING
						: NEXT PAGE)
010214	103041			BHIS	DONE23	: EXIT LOOP IF DONE
010216	005761	000002		TST	2(R1)	: REFERENCE OUTSIDE ALLOWED AREA -
						: SHOULD ABORT
010222	005077	172604		CLR	@SRO	: TURN KT11-C OFF IF NO ABORT
010226	104006			HLT		: NO ABORT OCCURRED ON A REFERENCE
						: OUTSIDE THE ALLOWED PAGE LENGTH
						: THE ADDRESS REFERENCED WAS THE VALUE CONTAINED IN
						: R1 PLUS 2
010230	000426			BR	CONT23	
010232	042777	000001	172572	RET23A:	BIC	#1,@SRO
010240	022626			CMP	(SP)+,(SP)+	: TURN OFF KT11-C
010242	104006			HLT		: RESTORE STACK POINTER
						: REFERENCE WITHIN ALLOWED AREA
						: CAUSED A TRAP OR ABORT
						: CLEAR ERROR BITS
010244	005077	172562		CLR	@SRO	
010250	000416			BR	CONT23	
010252	022626			RET23B:	CMP	(SP)+,(SP)+
010254	017703	172552		MOV	@SRO,R3	: RESTORE STACK POINTER
010260	005077	172546		CLR	@SRO	: SAVE CURRENT SRO
010264	022703	040003		CMP	#40003,R3	: TURN OFF KT11-C
010270	001401			BEQ	+.4	: CK SAVED SRO
010272	104006			HLT		: CONTENTS OF SRO INCORRECT AFTER
						: PAGE LENGTH ERROR ABORT - SHOULD SHOW PL ERROR
						: AND KERNEL PAGE 1
						: CHECK SRO TO BE SURE PL BIT CLEARED
010274	022777	000002	172530	CMP	#2,@SRO	
010302	001401			BEQ	+.4	: SRO INCORRECT AFTER CLEARING IT
010304	104006			HLT		: ONLY KERNEL PAGE 1 SHOULD STILL BE SET
						: SETUP R1 TO REFERENCE BOUNDARY OF
						: NEXT PAGE
010306	062701	000100	CONT23:	ADD	#100,R1	: ADD 1 TO VALUE TO BE LOADED IN
010312	062702	000400		ADD	#400,R2	: PAGE LENGTH FIELD
						: CHECK NEXT PAGE LENGTH VALUE
010316	000722			BR	LOOP23	: TURN OFF KT11-C
010320	005077	172506	DONE23:	CLR	@SRO	: RESTORE MEMORY MANAGEMENT ABORT RETURN
010324	016777	173062	173056	MOV	KTSTA,@KTVEC	: TO CAUSE HALT ON A FALSE TRAP
010332	005077	173054		CLR	@KTSTA	: OR ABORT

: TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN SET)
 : KERNEL PAGE 1 IS TESTED WITH ALL VALUES OF PAGE LENGTH FIELD
 : SHOW THAT REFERENCES TO BOTH BOUNDARIES OF THE ALLOWED AREA DON'T TRAP OR ABORT


```

010566 000722          BR      LOOP24          ;CHECK NEXT PAGE LENGTH VALUE
010570 005077 172236 172606 DONE24: CLR      @SR0          ;TURN OFF KT11-C
010574 016777 172612          MOV      KTSTA,@KTVEC ;RESTORE MEMORY MANAGEMENT ABORT RETURN
010602 005077 172604          CLR      @KTSTA      ;TO CAUSE A HALT ON A FALSE TRAP
                                           ;OR ABORT

;TEST ALL COMBINATIONS OF VALUES FOR THE PAGE LENGTH COMPARATORS-
;USE KERNEL I-SPACE PAGE 1
TEST25: SCOPE
010606 104400          MOV      #25,@#SR      ;DISPLAY TEST NUMBER
010610 012737 000025 177570 CLR      @#PS          ;INITIALIZE PROCESSOR STATUS
010616 005037 177776          MOV      #KSTACK,SP   ;INITIALIZE KERNEL STACK POINTER
010622 012706 001000          CLR      @SR3         ;INITIALIZE SR3
010626 005077 172214          CLR      @SR0         ;INITIALIZE SR0
010632 005077 172174          CMP      TESTCT,#25   ;IS THIS TEST BEING EXECUTED IN THE
010636 026727 172556 000025 BEQ      .+4           ;CORRECT SEQUENCE? - BRANCH IF YES
010644 001401          HLT                     ;TEST EXECUTED OUT OF ORDER - TESTCT
010646 104006          ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

010650 012767 000020 004326 MOV      #20,ICOUNT   ;DROP ITERATION COUNT
010656 004767 003662          JSR      %7,RWALL     ;INITIALIZE ALL PAGES RW, BANK 0
010662 012777 007600 172440 MOV      #7600,@KIPAR7 ;MAP KERNEL PAGE 7 EXTERNAL
010670 012777 011000 172512 MOV      #RET25,@KTVEC ;SETUP ABORT RETURN
010676 005077 172510          CLR      @KTSTA
010702 012701 000006          MOV      #6,R1        ;R1 CONTAINS THE VALUE TO BE
                                           ;LOADED INTO THE PDR
                                           ;TURN ON KT11-C
010706 012777 000001 172116 L25A: MOV      #1,@SR0   ;R3 CONTAINS VA USED TO REFERENCE PAGE 1
010714 012703 020000          MOV      #20000,R3   ;LOAD NEW PAGE LENGTH FIELD
010720 010177 172000          MOV      R1,@KIPDR1  ;R2 IS A COPY OF R1 USED FOR CALCULATIONS
010724 010102          L25B: MOV      R1,R2   ;R4 IS A COPY OF R3 USED FOR CALCULATIONS
010726 010304          MOV      R3,R4       ;CLEAR ALL BUT ADDRESS OFFSET IN R4
010730 042704 160000          BIC      #160000,R4  ;USE VA IN R3 TO REFERENCE PAGE 1
010734 005713          TST      (R3)        ;NO TRAP-CHECK TO MAKE SURE
010736 000302          SWAB      R2         ;THAT THE ADDRESS REFERENCED WAS
010740 042702 177400          BIC      #177400,R2  ;WITHIN THE ALLOWED PAGE LENGTH
010744 072427 177772          ASH      #-6,R4
010750 020402          CMP      R4,R2
010752 003401          BLE      .+4
010754 104006          HLT

;REFERENCE OUTSIDE ALLOWED PAGE LENGTH
;DIDN'T ABORT - R3 CONTAINS VA USED
;R1 CONTAINS VALUE LOADED INTO THE PDR
;CHANGE R3 TO REFERENCE NEXT BLOCK
;CHECK FOR ALL BLOCKS REFERENCED
;BRANCH IF NOT
;INCREMENT VALUE TO BE LOADED INTO
;THE PAGE LENGTH FIELD
;BRANCH IF NOT DONE
;EXIT
;RESTORE STACK POINTER
;CHECK TO MAKE SURE VIRTUAL
;ADDRESS WAS OUTSIDE ALLOWED
;PAGE LENGTH

010756 062703 000100          C25: ADD      #100,R3
010762 020327 037776          CMP      R3,#37776
010766 003756          BLE      L25B
010770 062701 000400          ADD      #400,R1

010774 100347          BPL      L25A
010776 000412          BR      DONE25
011000 022626          RET25: CMP      (SP)+,(SP)+
011002 000302          SWAB      R2
011004 042702 177400          BIC      #177400,R2
011010 072427 177772          ASH      #-6,R4
011014 020402          CMP      R4,R2
011016 003001          BGT      .+4
011020 104006          HLT
011022 000755          BR      C25

;REFERENCE WITHIN ALLOWED
;PAGE LENGTH ABORTED-R3 CONTAINS
    
```

```

011024 016777 172362 172356 DONE25: MOV      KTSTA, @KTVEC      ;VA USED, R1 CONTAINS VALUE
011032 005077 171774          CLR      @SR0          ;LOADED INTO THE PDR
                                ;RESTORE TRAP CATCHER
                                ;TURN OFF KT11-C

                                ;SHOW THAT THE W BIT DOESN'T SET IF THE KT11-C IS OFF
011036 104400          ;TEST26: SCOPE
011040 012737 000026 177570 MOV      #26, @#SR      ;DISPLAY TEST NUMBER
011046 005037 177776          CLR      @#PS          ;INITIALIZE PROCESSOR STATUS
011052 012706 001000          MOV      #KSTACK, SP   ;INITIALIZE KERNEL STACK POINTER
011056 005077 171764          CLR      @SR3          ;INITIALIZE SR3
011062 005077 171744          CLR      @SR0          ;INITIALIZE SRO
011066 026727 172326 000026 CMP      TESTCT, #26    ;IS THIS TEST BEING EXECUTED IN THE
011074 001401          BEQ      .+4          ;CORRECT SEQUENCE? - BRANCH IF YES
011076 104006          HLT                    ;TEST EXECUTED OUT OF ORDER - TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

011100 012767 002000 004076 MOV      #2000, ICOUNT ;RESTORE ITERATION COUNT
011106 004767 003402          JSR      %7, CLRAL     ;CLEAR ALL KT11-C REGISTERS
011112 013737 010000 010000 MOV      @#10000, @#10000 ;WRITE BANK 0
011120 005777 172126          TST      @KIPDR0      ;CHECK CORRESPONDING PDR
011124 001401          BEQ      .+4
011126 104006          HLT                    ;W BIT SET OR ANOTHER BIT INCORRECT
                                ;IN KERNEL I-SPACE 0 PDR

                                ;SHOW THAT THE W BIT IS CLEARED BY WRITING (VIA DATO) THE CORRESPONDING PAR
                                ;CHECK EACH PDR
011130 104400          ;TEST27: SCOPE
011132 012737 000027 177570 MOV      #27, @#SR      ;DISPLAY TEST NUMBER
011140 005037 177776          CLR      @#PS          ;INITIALIZE PROCESSOR STATUS
011144 012706 001000          MOV      #KSTACK, SP   ;INITIALIZE KERNEL STACK POINTER
011150 005077 171672          CLR      @SR3          ;INITIALIZE SR3
011154 005077 171652          CLR      @SR0          ;INITIALIZE SRO
011160 026727 172234 000027 CMP      TESTCT, #27    ;IS THIS TEST BEING EXECUTED IN THE
011166 001401          BEQ      .+4          ;CORRECT SEQUENCE? - BRANCH IF YES
011170 104006          HLT                    ;TEST EXECUTED OUT OF ORDER - TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

011172 004767 003346          JSR      %7, RWALL     ;INITIALLY MAP ALL PAGES RW, 4K, BANK 0
011176 012777 000007 171642 MOV      #7, @SR3      ;ENABLE ALL D-SPACES
011204 012777 007600 172136 MOV      #7600, @KDPAR7 ;MAP I-SPACE 7, ALL MODES,
011212 012777 007600 172030 MOV      #7600, @SDPAR7 ;TO THE EXTERNAL BANK
011220 012777 007600 171722 MOV      #7600, @UDPAR7
011226 012737 040000 177776 MOV      #40000, @#PS   ;SET MODE TO SUPERVISOR
011234 012706 002000          MOV      #SSTACK, R6   ;SETUP SUPERVISOR STACK POINTER
011240 012737 140000 177776 MOV      #140000, @#PS  ;SET MODE TO USER
011246 012706 003000          MOV      #USTACK, R6   ;SETUP USER STACK POINTER
011252 012700 003366          MOV      #STATAB, R0   ;POINT R0 TO THE TABLE OF MODE INFORMATION
011256 012001          LOP27: MOV      (R0)+, R1 ;POINT R1 TO THE ADDRESS OF
                                ;THE CURRENT I-SPACE PDR
011260 012702 011276          MOV      #ADR27A, R2  ;R2 CONTAINS VIRTUAL ADDRESS TO
                                ;REFERENCE DESIRED PAGE
011264 012037 177776          MOV      (R0)+, @#PS   ;SETUP STATUS FOR CURRENT MODE
011270 005277 171536          LOP27A: INC      @SR0  ;TURN ON KT11-C
011274 010207          MOV      R2, PC      ;CHANGE CURRENT PC TO REFERENCE
                                ;DESIRED PAGE
    
```

```

011276 005027 000000          ADR27A: CLR      #0          ;WRITE I-SPACE
011302 042707 160000          BIC      #160000,PC ;CHANGE TO PAGE 0 PC
011306 005077 171520          CLR      @SR0      ;TURN OFF KT11-C
011312 004767 000056          JSR      %7,CKWBIT ;TEST W BIT
011316 005721                TST      (R1)+      ;MOVE POINTER
011320 062702 020000          ADD      #20000,R2 ;CHANGE VA TO REFERENCE NEXT PAGE
011324 103361                BCC      LOP27A    ;LOOP UNTIL ALL I-SPACE PDR'S HAVE BEEN
                                ;CHECKED IN THE CURRENT MODE
011326 012702 017776          MOV      #17776,R2 ;SETUP R2 TO REFERENCE DESIRED PAGE
011332 005277 171474          LOP27B: INC      @SR0 ;TURN ON KT11-C
011336 011212                MOV      (R2),(R2) ;WRITE IN D-SPACE
011340 005077 171466          CLR      @SR0      ;TURN OFF KT11-C
011344 004767 000024          JSR      %7,CKWBIT ;TEST W BIT
011350 005721                TST      (R1)+      ;MOVE POINTER
011352 062702 020000          ADD      #20000,R2 ;CHANGE VA TO REFERENCE NEXT PAGE
011356 103365                BCC      LOP27B    ;LOOP UNTIL ALL D-SPACE PDR'S HAVE BEEN
                                ;CHECKED IN THE CURRENT MODE
011360 020027 003400          CMP      R0,#STAEND ;HAVE ALL MODES BEEN TESTED?
011364 002734                BLT      LOP27     ;NO, BRANCH
011366 005077 171454          CLR      @SR3      ;YES, CLEAR ALL D-SPACE ENABLES
011372 000415                BR       TEST30    ;EXIT
011374 032771 000100 000000 CKWBIT: BIT      #100,@(R1) ;CHECK W BIT
011402 001001                BNE      .+4
011404 104006                HLT
011406 005071 000040          CLR      @40(R1)   ;W BIT DIDN'T SET IN PDR WHOSE
                                ;ADDRESS IS POINTED TO BY R1
011412 032771 000100 000000          BIT      #100,@(R1) ;CLEAR W BIT BY WRITING CORRESPONDING
011420 001401                BEQ      .+4        ;PAR VIA DATO
011422 104006                HLT                ;CHECK W BIT
011424 000207                RTS      %7        ;W BIT DIDN'T CLEAR IN PDR WHOSE
                                ;ADDRESS IS POINTED TO BY R1

                                ;SHOW THAT THE W BIT IS CLEARED BY A DATOB TO THE PDR
                                ;CHECK BOTH HIGH AND LOW DATOB'S, ON KERNEL 0 I-SPACE ONLY
011426 104400                TEST30: SCOPE
011430 012737 000030 177570          MOV      #30,@#SR  ;DISPLAY TEST NUMBER
011436 005037 177776          CLR      @#PS      ;INITIALIZE PROCESSOR STATUS
011442 012706 001000          MOV      #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
011446 005077 171374          CLR      @SR3      ;INITIALIZE SR3
011452 005077 171354          CLR      @SR0      ;INITIALIZE SR0
011456 026727 171736 000030          CMP      TESTCT,#30 ;IS THIS TEST BEING EXECUTED IN THE
011464 001401                BEQ      .+4        ;CORRECT SEQUENCE? - BRANCH IF YES
011466 104006                HLT                ;TEST EXECUTED OUT OF ORDER - TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

011470 004767 003050          JSR      %7,RWALL   ;MAP ALL PAGES 4K, RW, BANK 0
011474 012777 007600 171626          MOV      #7600,@KIPAR7 ;MAP KERNEL 7 TO THE EXTERNAL BANK
011502 005277 171324          INC      @SR0      ;TURN ON KT11-C
011506 013737 000000 000000          MOV      @#0,@#0   ;WRITE INTO PAGE 0
011514 005077 171312          CLR      @SR0      ;TURN OFF KT11-C
011520 032777 000100 171524          BIT      #100,@KIPDRO ;CHECK W BIT
011526 001001                BNE      .+4
011530 104006                HLT
011532 112777 000106 171512          MOVB    #106,@KIPDRO ;W BIT NOT SET AFTER WRITING PAGE 0
011540 032777 000100 171504          BIT      #100,@KIPDRO ;DATOB SHOULD CLEAR W BIT
    
```

```

011546 001401      BEQ      .+4
011550 104006      HLT
;W BIT DIDN'T CLEAR VIA DATOB (LOW)
;TO THE PDR
011552 005277 171254      INC      @SR0      ;TURN ON KT11-C
011556 013737 017776 017776      MOV      @#17776,@#17776 ;WRITE INTO PAGE 0 AGAIN
011564 005077 171242      CLR      @SR0      ;TURN OFF KT11-C
011570 032777 000100 171454      BIT      #100,@KIPDR0 ;CHECK W BIT
011576 001001      BNE      .+4
011600 104006      HLT
;W BIT NOT SET AFTER WRITING PAGE 0
011602 016701 171444      MOV      KIPDR0,R1 ;SETUP R1 TO REFERENCE HIGH BYTE
011606 005201      INC      R1         ;OF KIPDR0
011610 112711 000177      MOVB    #177,@R1   ;DATOB TO HIGH BYTE OF KIPDR0
011614 032777 000100 171430      BIT      #100,@KIPDR0 ;CHECK W BIT
011622 001401      BEQ      .+4
011624 104006      HLT
;W BIT DIDN'T CLEAR VIA DATOB
;TO HIGH BYTE OF PDR

;SHOW THAT THE W BIT IS CLEARED BY A DATOB TO THE PAR
;CHECK BOTH HIGH AND LOW DATOB'S, ON KERNEL 0 I-SPACE ONLY
TEST31: SCOPE
011626 104400      MOV      #31,@SR   ;DISPLAY TEST NUMBER
011630 012737 000031 177570      CLR      @#PS      ;INITIALIZE PROCESSOR STATUS
011636 005037 177776      MOV      #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
011642 012706 001000      CLR      @SR3      ;INITIALIZE SR3
011646 005077 171174      CLR      @SR0      ;INITIALIZE SR0
011652 005077 171154      CMP     TESTCT,#31 ;IS THIS TEST BEING EXECUTED IN THE
011656 026727 171536 000031      BEQ     .+4        ;CORRECT SEQUENCE? - BRANCH IF YES
011664 001401      BEQ     .+4        ;TEST EXECUTED OUT OF ORDER - TESTCT
011666 104006      HLT              ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

011670 004767 002650      JSR     %7,RWALL   ;MAP ALL PAGES 4K, RW, BANK 0
011674 012777 007600 171426      MOV     #7600,@KIPAR7 ;MAP KERNEL 7 TO THE EXTERNAL BANK
011702 005277 171124      INC     @SR0      ;TURN ON KT11-C
011706 013737 000000 000000      MOV     @#0,@#0    ;WRITE INTO PAGE 0
011714 005077 171112      CLR     @SR0      ;TURN OFF KT11-C
011720 032777 000100 171324      BIT     #100,@KIPDR0 ;CHECK W BIT
011726 001001      BNE     .+4
011730 104006      HLT
;W BIT NOT SET AFTER WRITING PAGE 0
011732 112777 000000 171352      MOVB   #0,@KIPAR0 ;DATOB TO THE PAR
011740 032777 000100 171304      BIT     #100,@KIPDR0 ;CHECK W BIT
011746 001401      BEQ     .+4
011750 104006      HLT
;W BIT DIDN'T CLEAR VIA DATOB
;(LOW) TO THE PAR
011752 005277 171054      INC     @SR0      ;TURN ON KT11-C
011756 013737 017776 017776      MOV     @#17776,@#17776 ;WRITE INTO PAGE 0 AGAIN
011764 005077 171042      CLR     @SR0      ;TURN OFF KT11-C
011770 032777 000100 171254      BIT     #100,@KIPDR0 ;CHECK W BIT
011776 001001      BNE     .+4
012000 104006      HLT
;W BIT NOT SET AFTER WRITING PAGE 0
012002 016701 171304      MOV     KIPAR0,R1 ;SETUP R1 TO REFERENCE HIGH BYTE
012006 005201      INC     R1         ;OF KIPAR0
012010 112711 000000      MOVB   #0,@R1    ;DATOB TO HIGH BYTE OF KIPAR0
012014 032777 000100 171230      BIT     #100,@KIPDR0 ;CHECK W BIT
012022 001401      BEQ     .+4
012024 104006      HLT
;W BIT DIDN'T CLEAR VIA DATOB
;TO HIGH BYTE OF PAR

```

```

:SHOW THAT THE W BIT IS NOT CLEARED BY INIT
:INITIALLY SET ALL THE W BITS, THEN DO A RESET AND CHECK THE W BITS
TEST32: SCOPE
012026 104400
012030 012737 000032 177570      MOV     #32, @#SR      ;DISPLAY TEST NUMBER
012036 005037 177776             CLR     @#PS          ;INITIALIZE PROCESSOR STATUS
012042 012706 001000             MOV     #KSTACK, SP  ;INITIALIZE KERNEL STACK POINTER
012046 005077 170774             CLR     @SR3         ;INITIALIZE SR3
012052 005077 170754             CLR     @SRO         ;INITIALIZE SRO
012056 026727 171336 000032      CMP     TESTCT, #32  ;IS THIS TEST BEING EXECUTED IN THE
012064 001401                    BEQ     .+4          ;CORRECT SEQUENCE? - BRANCH IF YES
012066 104006                    HLT                    ;TEST EXECUTED OUT OF ORDER - TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

012070 004767 002450             JSR     %7, RWALL    ;INITIALIZE ALL PAGES RW, 4K, BANK 0
012074 012777 000007 170744      MOV     #7, @SR3    ;ENABLE ALL D-SPACES
012102 012777 007600 171240      MOV     #7600, @KDPAR7 ;MAP D-SPACE PAGE 7, ALL MODES,
012110 012777 007600 171132      MOV     #7600, @SDPAR7 ;TO THE EXTERNAL BANK
012116 012777 007600 171024      MOV     #7600, @UDPAR7
012124 012737 040000 177776      MOV     #40000, @#PS ;SET MODE TO SUPERVISOR
012132 012706 002000             MOV     #SSTACK, R6 ;SETUP SUPERVISOR STACK
012136 012737 140000 177776      MOV     #140000, @#PS ;SET MODE TO USER
012144 012706 003000             MOV     #USTACK, R6 ;SETUP USER STACK
012150 012700 003366             MOV     #STATAB, R0 ;R0 POINTS TO INFORMATION FOR
                                ;CURRENT MODE
012154 005720                    TST     (R0)+        ;MOVE POINTER
012156 012037 177776             MOV     (R0)+, @#PS ;SETUP MODE TO REFERENCE NEXT
                                ;SET OF REGISTERS
012162 012702 012174             MOV     #ADR32, R2  ;R2 CONTAINS VA TO REFERENCE
                                ;DESIRED PAGE
012166 005277 170640             LOP32A: INC @SRO    ;TURN ON KT11-C
012172 010207                    LOP32B: MOV R2, PC  ;CHANGE PC TO REFERENCE NEXT PAGE
012174 005027 000000            ADR32:  CLR #0      ;WRITE IN I-SPACE
012200 062702 020000            ADR32:  ADD #20000, R2 ;SETUP NEXT VIRTUAL ADDRESS
012204 103372                    BCC    LOP32B'     ;LOOP TILL ALL I-SPACE PAGES IN
                                ;THIS MODE HAVE BEEN WRITTEN INTO
012206 042707 160000            BIC     #160000, PC ;CHANGE TO BANK ZERO PC
012212 012702 017776            MOV     #17776, R2 ;SETUP R2 TO REFERENCE DESIRED
                                ;D-SPACE PAGE

012216 011212                    LOP32C: MOV (R2), (R2) ;WRITE IN D-SPACE
012220 062702 020000            ADD     #20000, R2  ;CHANGE VA TO REFERENCE NEXT PAGE
012224 103374                    BCC    LOP32C'     ;SET ALL W-BITS IN CURRENT MODE
012226 005077 170600            CLR     @SRO       ;TURN OFF KT11-C
012232 020027 003400            CMP     R0, #STAEND ;CHECK FOR DONE SETTING THE W BITS
012236 002746                    BLT    LOOP32     ;IF NOT, LOOP TO DO NEXT MODE
012240 012701 003052            MOV     #ADRTAB, R1 ;SETUP R1 TO REFERENCE ADDRESSES
                                ;OF PDR'S
012244 012702 000020            LOP32D: MOV #20, R2 ;USE R2 AS COUNTER TO CHANGE ADDRESS
                                ;AT END OF EACH SET OF REGISTERS
012250 032771 000100 000000      LOP32E: BIT #100, @ (R1) ;CHECK W BIT
012256 001001                    BNE    .+4
012260 104006                    HLT
;W BIT NOT SET IN PDR WHOSE
;ADDRESS IS POINTED TO BY R1-
;SHOULD HAVE BEEN SET WHEN
;PAGE WAS WRITTEN INTO
    
```



```

012262 005721          TST      (R1)+          ; MOVE POINTER
012264 077207          SOB      R2,LOP32E      ; CHECK ALL PDR'S IN THIS SET
012266 062701 000040  ADD      #40,R1        ; CHANGE R1 TO REFERENCE NEXT
                                ; SET OF PDR ADDRESSES
012272 020127 003350  CMP      R1,#ADREND     ; CHECK FOR DONE
012276 002762          BLT      LOP32D         ; IF NOT, CHECK NEXT SET OF PDR'S
012300 005037 177776  CLR      @#PS          ; SET MODE TO KERNEL
012304 005277 170522  INC      @SR0         ; TURN KT11-C ON
012310 105777 170502  TSTB    @TCSR         ; WAIT FOR ANY TTY OUTPUT TO FINISH
012314 100375          BPL      .-4
012316 000005          RESET
012320 000005          RESET
012322 012701 003052  MOV      #ADRTAB,R1    ; INIT WITH KT11-C ON
012326 012702 000020  LOP32F: MOV     #20,R2  ; INIT WITH KT11-C OFF
                                ; R1 REFERENCES ADDRESS OF PDR
                                ; R2 KEEPS TRACK OF WHEN TO CHANGE
                                ; REGISTER SETS
                                ; CHECK W BIT
012332 032771 000100 000000 LOP32G: BIT     #100,@(R1)
012340 001001          BNE     .+4
012342 104006          HLT
                                ; INIT CLEARED W BIT IN PDR WHOSE
                                ; ADDRESS IS POINTED TO BY R1
012344 005721          TST      (R1)+          ; MOVE POINTER
012346 077207          SOB      R2,LOP32G      ; CHECK ALL PDR'S IN THIS SET
012350 062701 000040  ADD      #40,R1        ; CHANGE R1 TO REFERENCE NEXT SET
                                ; OF PDR ADDRESSES
012354 020127 003350  CMP      R1,#ADREND     ; CHECK FOR DONE
012360 002762          BLT      LOP32F         ; IF NOT, CHECK NEXT SET OF PDR'S
012362 005077 170444  CLR      @SR0         ; REINITIALIZE SR0, SR3
012366 005077 170454  CLR      @SR3

; SHOW THAT EACH "A" BIT IN THE I-SPACE PDR'S CAN BE SET BY REFERENCING
; THE CORRESPONDING PAGE MAPPED READ WRITE AND TRAP ON WRITE WITHOUT
; MEMORY MANAGEMENT ENABLE SET
; SHOW THAT ONLY ONE "A" BIT SETS ON EACH REFERENCE
; SHOW THAT EACH "A" BIT IN THE I-SPACE PDR'S IS CLEARED BY A DATO TO THE PDR,
; AND BY A DATO TO THE CORRESPONDING PAR
012372 104400          TEST33: SCOPE
012374 012737 000033 177570  MOV     #33,@#SR      ; DISPLAY TEST NUMBER
012402 005037 177776  CLR     @#PS          ; INITIALIZE PROCESSOR STATUS
012406 012706 001000  MOV     #KSTACK,SP   ; INITIALIZE KERNEL STACK POINTER
012412 005077 170430  CLR     @SR3         ; INITIALIZE SR3
012416 005077 170410  CLR     @SR0         ; INITIALIZE SR0
012422 026727 170772 000033  CMP     TESTCT,#33   ; IS THIS TEST BEING EXECUTED IN THE
012430 001401          BEQ     .+4         ; CORRECT SEQUENCE? - BRANCH IF YES
012432 104006          HLT      ; TEST EXECUTED OUT OF ORDER - TESTCT
                                ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
012434 004767 002252          JSR     %7,RWTWAL    ; MAP ALL PAGES 4K, RWTW
                                ; MAP ALL EXCEPT D-SPACE 7 (ALL MODES)
                                ; TO BANK 0 - MAP D-SPACE 7 (ALL MODES)
                                ; TO EXTERNAL BANK
012440 012737 040000 177776  MOV     #40000,@#PS  ; SET MODE TO SUPERVISOR
012446 012706 002000          MOV     #SSTACK,R6   ; SETUP SUPERVISOR STACK POINTER
012452 012737 140000 177776  MOV     #140000,@#PS ; SET MODE TO USER
012460 012706 003000          MOV     #USTACK,R6   ; SETUP USER STACK POINTER
012464 012777 000007 170354  MOV     #7,@SR3      ; ENABLE ALL D-SPACES
012472 012700 003366          MOV     #STATAB,R0   ; POINT R0 TO TABLE OF MODE INFORMATION
012476 012001          LOP33A: MOV    (R0)+,R1 ; R1 POINTS TO ADDRESS OF PDR BEING TESTED

```

012500	012037	177776		MOV	(R0)+, @#PS	;	SET STATUS TO REFERENCE DESIRED MODE
012504	012702	012522		MOV	#ADR33A, R2	;	R2 AND R3 CONTAIN VIRTUAL ADDRESSES
012510	012703	012642		MOV	#ADR33B, R3	;	USED TO REFERENCE CURRENT PAGE
012514	005277	170312		LOP33B: INC	@SR0	;	TURN ON KT11-C
012520	010207			MOV	R2, PC	;	CHANGE ADDRESS TO REFERENCE DESIRED PAGE
012522	005027	000000		ADR33A: CLR	#0	;	WRITE IN I-SPACE
012526	042707	160000		BIC	#160000, PC	;	CHANGE TO BANK 0 ADDRESS
012532	005077	170274		CLR	@SR0	;	TURN OFF KT11-C
012536	032771	000200	000000	BIT	#200, @ (R1)	;	CHECK "A" BIT
012544	001001			BNE	+.4	;	"A" BIT NOT SET IN PDR WHOSE ADDRESS IS POINTED TO
012546	104006			HLT		;	BY R1 AFTER WRITING CORRESPONDING PAGE
						;	MAPPED RWTW
012550	012704	003052		MOV	#ADRTAB, R4	;	SETUP TO CHECK ALL OTHER "A" BITS
012554	012705	000020		LOP33C: MOV	#20, R5	;	R5 KEEPS TRACK OF WHEN TO CHANGE REGISTER SETS
012560	020104			LOP33D: CMP	R1, R4	;	DON'T CHECK IF ITS THE PDR BEING
012562	001405			BEQ	CNT33A	;	TESTED
012564	032774	000200	000000	BIT	#200, @ (R4)	;	OTHERWISE MAKE SURE THE "A" BIT DIDN'T SET
012572	001401			BEQ	+.4	;	
012574	104006			HLT		;	THE "A" BIT WAS SET IN THE PDR WHOSE
						;	ADDRESS IS POINTED TO BY R4, WITHOUT THE
						;	CORRESPONDING PAGE HAVING BEEN REFERENCED
						;	R2 CONTAINS THE ADDRESS THAT WAS REFERENCED
012576	005724			CNT33A: TST	(R4)+	;	MOVE POINTER
012600	077511			SOB	R5, LOP33D	;	CHECK ALL PDR'S IN THIS SET
012602	062704	000040		ADD	#40, R4	;	CHANGE ADDRESS TO REFERENCE NEXT SET OF PDR'S
012606	020427	003350		CMP	R4, #ADREND	;	HAVE ALL REGISTERS BEEN CHECKED?
012612	002760			BLT	LOP33C	;	NO-BRANCH
012614	012771	077605	000000	MOV	#77605, @ (R1)	;	YES-CLEAR "A" BIT BY DATO TO PDR
012622	032771	000200	000000	BIT	#200, @ (R1)	;	CHECK THE "A" BIT
012630	001401			BEQ	+.4	;	
012632	104006			HLT		;	DATO TO PDR WHOSE ADDRESS IS POINTED
						;	TO BY R1 DIDN'T CLEAR THE A BIT
012634	005277	170172		INC	@SR0	;	TURN ON KT11-C
012640	010307			MOV	R3, PC	;	SET "A" BIT AGAIN
012642	005027	000000		ADR33B: CLR	#0	;	
012646	042707	160000		BIC	#160000, PC	;	
012652	005077	170154		CLR	@SR0	;	TURN OFF KT11-C
012656	032771	000200	000000	BIT	#200, @ (R1)	;	CHECK THE "A" BIT
012664	001001			BNE	+.4	;	
012666	104006			HLT		;	A BIT NOT SET AFTER WRITING I-SPACE
						;	PAGE WHOSE ADDRESS IS POINTED TO BY R1-
						;	MAPPED RWTW
012670	017171	000040	000040	MOV	@40(R1), @40(R1)	;	CLEAR "A" BIT BY DATO TO PAR
012676	032771	000200	000000	BIT	#200, @ (R1)	;	CHECK THE "A" BIT
012704	001401			BEQ	+.4	;	
012706	104006			HLT		;	DATO TO CORRESPONDING PAR DIDN'T CLEAR "A" BIT
						;	IN PDR WHOSE ADDRESS IS POINTED TO BY R1
012710	005721			TST	(R1)+	;	MOVE POINTER
012712	062702	020000		ADD	#20000, R2	;	CHANGE VIRTUAL ADDRESSES TO REFERENCE
012716	062703	020000		ADD	#20000, R3	;	NEXT PAGE
012722	103274			BCC	LOP33B	;	CHECK ALL I-SPACE PDR'S IN THIS MODE
012724	020067	170450		CMP	RO, STAEND	;	
012730	002662			BLT	LOP33A	;	CHECK ALL 3 MODES
012732	005077	170110		CLR	@SR3	;	CLEAR ALL D-SPACE ENABLES

```

;SHOW THAT EACH "A" BIT IN THE D-SPACE PDR'S CAN BE SET BY REFERENCING
;THE CORRESPONDING PAGE MAPPED READ WRITE AND TRAP ON WRITE
;WITHOUT MEMORY MANAGEMENT ENABLE SET
;SHOW THAT EACH "A" BIT IN THE D-SPACE PDR'S CAN BE CLEARED BY A DATO TO
;THE PDR, AND BY A DATO TO THE CORRESPONDING PAR
012736 104400          000034 177570          TEST34: SCOPE
012740 012737          000034 177570          MOV      #34, @#SR
012746 005037          177776          CLR      @#PS
012752 012706          001000          MOV      #KSTACK, SP
012756 005077          170064          CLR      @SR3
012762 005077          170044          CLR      @SR0
012766 026727          170426 000034          CMP      TESTCT, #34
012774 001401          170426          BEQ      .+4
012776 104006          170426          HLT

013000 004767 001706          JSR      %7, RWTWAL
;DISPLAY TEST NUMBER
;INITIALIZE PROCESSOR STATUS
;INITIALIZE KERNEL STACK POINTER
;INITIALIZE SR3
;INITIALIZE SR0
;IS THIS TEST BEING EXECUTED IN THE
;CORRECT SEQUENCE? - BRANCH IF YES
;TEST EXECUTED OUT OF ORDER - TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

013004 012737 040000 177776          MOV      #40000, @#PS
013012 012706 002000 177776          MOV      #SSTACK, R6
013016 012737 140000 177776          MOV      #140000, @#PS
013024 012706 003000 177776          MOV      #USTACK, R6
013030 012777 000007 170010          MOV      #7, @SR3
013036 012700 003366          MOV      #STATAB, R0
;MAP ALL PAGES 4K, RWTW, BANK 0
;EXCEPT D-SPACE 7, MAPPED TO THE
;EXTERNAL BANK
;SET MODE TO SUPERVISOR
;SET UP SUPERVISOR STACK POINTER
;SET MODE TO USER
;SET UP USER STACK POINTER
;ENABLE ALL D-SPACES
;R0 POINTS TO TABLE OF MODE INFORMATION
;R1 POINTS TO ADDRESS OF PDR BEING
;TESTED
;SETUP MODE TO BE TESTED
;USE R2 TO REFERENCE PAGE TO SET THE A BIT
;TURN ON KT11-C
;WRITE IN D-SPACE PAGE TO SET "A" BIT
;TURN OFF KT11-C
;CHECK THE "A" BIT
LOP34A: 013042 012001          MOV      (R0)+, R1
013044 062701 000020          ADD      #20, R1
013050 012037 177776          MOV      (R0)+, @#PS
013054 012702 017776          MOV      #17776, R2
;CHECK THE "A" BIT
; "A" BIT NOT SET AFTER WRITING D-SPACE
;PAGE MAPPED RWTW
;ADDRESS REFERENCED IS IN R2
;CLEAR "A" BIT BY DATO TO PDR
;CHECK THE "A" BIT
LOP34B: 013060 005277 167746          @SR0
013064 011212          MOV      @R2, @R2
;TURN ON KT11-C
;SET "A" BIT AGAIN
;TURN OFF KT11-C
;CHECK "A" BIT
013066 005077 167740          CLR      @SR0
013072 032771 000200 000000          BIT      #200, @ (R1)
013100 001001          BNE      .+4
013102 104006          HLT

013104 017171 000000 000000          MOV      @ (R1), @ (R1)
013112 032771 000200 000000          BIT      #200, @ (R1)
013120 001401          BEQ      .+4
013122 104006          HLT

;DATO TO PDR DIDN'T CLEAR "A" BIT -
;PDR ADDRESS IS POINTED TO BY R1
013124 005277 167702          INC      @SR0
013130 011212          MOV      @R2, @R2
013132 005077 167674          CLR      @SR0
013136 032771 000200 000000          BIT      #200, @ (R1)
013144 001001          BNE      .+4
013146 104006          HLT
; "A" BIT NOT SET AFTER WRITING D-SPACE
    
```

```

013150 017171 000040 000040      MOV      @40(R1),@40(R1)      ;PAGE MAPPED RWTW
013156 032771 000200 000000      BIT      #200,@(R1)         ;ADDRESS REFERENCED IS IN R2
013164 001401                      BEQ      .+4                 ;CLEAR THE "A" BIT BY DATO TO PAR
013166 104006                      HLT                               ;CHECK THE "A" BIT

                                ;DATO TO PAR DIDN'T CLEAR "A" BIT
013170 005721                      TST      (R1)+               ;IN CORRESPONDING PDR WHOSE ADDRESS IS POINTED TO BY R1
013172 062702 020000      ADD      #20000,R2          ;MOVE POINTER
013176 103330                      BCC      LOP34B              ;CHANGE R2 TO REFERENCE NEXT PAGE
013200 020027 003400      CMP      RO,#STAEND         ;CHECK ALL D-SPACE PDR'S IN THIS MODE
013204 002716                      BLT      LOP34A
013206 005077 167634      CLR      @SR3                ;CHECK ALL 3 MODES
                                ;CHECK ALL D-SPACE ENABLES

;SHOW THAT THE A BIT IS CLEARED BY A DATOB TO THE PDR AND ALSO BY
;A DATOB TO THE CORRESPONDING PAR. CHECK BOTH HIGH AND LOW BYTES, ON
;KERNEL PAGE 1 I-SPACE
TEST35: SCOPE
013212 104400                      MOV      #35,@#SR           ;DISPLAY TEST NUMBER
013214 012737 000035 177570      CLR      @#PS               ;INITIALIZE PROCESSOR STATUS
013222 005037 177776                      MOV      #KSTACK,SP        ;INITIALIZE KERNEL STACK POINTER
013226 012706 001000                      CLR      @SR3              ;INITIALIZE SR3
013232 005077 167610                      CLR      @SR0              ;INITIALIZE SR0
013236 005077 167570                      CMP      TESTCT,#35         ;IS THIS TEST BEING EXECUTED IN THE
013242 026727 170152 000035      BEQ      .+4                 ;CORRECT SEQUENCE? - BRANCH IF YES
013250 001401                      HLT                               ;TEST EXECUTED OUT OF ORDER - TESTCT
013252 104006                      ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

013254 004767 001264                      JSR      %7,RWALL           ;MAP ALL PAGES 4K, RW, BANK 0
013260 012777 007600 170042      MOV      #7600,@KIPAR7     ;MAP KERNEL 7 TO THE EXTERNAL BANK
013266 012777 077404 167760      MOV      #77404,@KIPDR1    ;MAP KERNEL 1 RW
013274 005277 167532                      INC      @SR0               ;TURN ON KT11-C
013300 005737 020000                      TST      @#20000           ;SET THE A BIT
013304 005077 167522                      CLR      @SR0               ;TURN OFF KT11-C
013310 032777 000200 167736      BIT      #200,@KIPDR1      ;MAKE SURE THAT THE A BIT SET
013316 001001                      BNE      .+4
013320 104006                      HLT
013322 117777 167726 167724      MOV      @KIPDR1,@KIPDR1   ;THE A BIT DIDN'T SET IN KIPDR1
013330 032777 000200 167716      BIT      #200,@KIPDR1      ;DATOB TO THE LOW BYTE OF THE PDR
013336 001401                      BEQ      .+4                 ;CHECK THE A BIT
013340 104006                      HLT                               ;THE A BIT WASN'T CLEARED BY DATOB
                                ;TO THE LOW BYTE OF THE PDR
013342 005277 167464                      INC      @SR0               ;TURN ON KT11-C
013346 005737 020000                      TST      @#20000           ;SET THE A BIT AGAIN
013352 005077 167454                      CLR      @SR0               ;TURN OFF KT11-C
013356 016701 167672                      MOV      KIPDR1,R1         ;SETUP R1 TO REFERENCE HIGH BYTE
013362 005201                      INC      R1                  ;OF THE PDR
013364 111111                      MOV      @R1,@R1           ;DATOB TO HIGH BYTE OF PDR
013366 032777 000200 167660      BIT      #200,@KIPDR1      ;CHECK A BIT
013374 001401                      BEQ      .+4
013376 104006                      HLT                               ;THE A BIT WASN'T CLEARED BY A
                                ;DATOB TO THE HIGH BYTE OF THE PDR
013400 005277 167426                      INC      @SR0               ;TURN ON KT11-C
013404 005737 020000                      TST      @#20000           ;SET THE A BIT AGAIN
013410 005077 167416                      CLR      @SR0               ;TURN OFF KT11-C
013414 117777 167674 167672      MOV      @KIPAR1,@KIPAR1   ;DATOB TO LOW BYTE OF THE PAR
    
```

```

013422 032777 000200 167624 BIT #200, @KIPDR1 ;CHECK THE A BIT
013430 001401 BEQ .+4
013432 104006 HLT ;THE A BIT WASN'T CLEARED BY A DATOB
; TO THE LOW BYTE OF THE PAR
013434 005277 167372 INC @SRO ;TURN ON KT11-C
013440 005737 020000 TST @#20000 ;SET THE A BIT AGAIN
013444 005077 167362 CLR @SRC ;TURN OFF KT11-C
013450 016701 167640 MOV KIPAR1, R1 ;SETUP R1 TO ADDRESS THE HIGH
013454 005201 INC R1 ;BYTE OF THE PAR
013456 111111 MOVB @R1, @R1 ;DATOB TO HIGH BYTE OF PAR
013460 032777 000200 167566 BIT #200, @KIPDR1
013466 001401 BEQ .+4
013470 104006 HLT ;THE A BIT WAS NOT CLEARED BY A
; DATOB TO THE HIGH BYTE OF THE PAR

;SHOW THAT INIT DOESN'T CLEAR THE "A" BITS, WITH KT11-C ON OR OFF
;ALL "A" BITS ARE CHECKED
013472 104400 TEST36: SCOPE
013474 012737 000036 177570 MOV #36, @#SR ;DISPLAY TEST NUMBER
013502 005037 177776 CLR @#PS ;INITIALIZE PROCESSOR STATUS
013506 012706 001000 MOV #KSTACK, SP ;INITIALIZE KERNEL STACK POINTER
013512 005077 167330 CLR @SR3 ;INITIALIZE SR3
013516 005077 167310 CLR @SRO ;INITIALIZE SRO
013522 026727 167672 000036 CMP TESTCT, #36 ;IS THIS TEST BEING EXECUTED IN THE
013530 001401 BEQ .+4 ;CORRECT SEQUENCE? - BRANCH IF YES
013532 104006 HLT ;TEST EXECUTED OUT OF ORDER - TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

013534 004767 001152 JSR %? RWTWAL ;MAKE ALL PAGES READ WRITE AND TRAP ON WRITE
013540 012777 007600 167602 MOV #7600, @KDPAR7 ;MAP D-SPACE PAGE 7, ALL MODES,
013546 012777 007600 167474 MOV #7600, @SDPAR7 ;TO THE EXTERNAL BANK
013554 012777 007600 167366 MOV #7600, @UDPAR7
013562 012777 000007 167256 MOV #7, @SR3 ;ENABLE ALL D-SPACES
013570 012700 003366 MOV #STATAB, R0 ;POINT R0 TO THE TABLE OF MODE INFORMATION
013574 005720 LOOP36: TST (R0)+
013576 012037 177776 MOV (R0)+, @#PS ;SET THE STATUS TO REFERENCE THE DESIRED SET OF
; REGISTERS
013602 012702 013616 MOV #ADR36, R2 ;R2 REFERENCES THE DESIRED PAGE
013606 012777 000001 167216 LOP36A: MOV #1, @SRO ;TURN ON THE KT11-C
013614 010207 ADR36: MOV R2, PC ;CHANGE PC TO THE DESIRED PAGE
013616 005027 000000 CLR #0 ;WRITE I-SPACE
013622 062702 020000 ADD #20000, R2 ;CHANGE R2 TO REFERENCE THE NEXT PAGE
013626 103372 LOP36A: BCC LOP36A ;WRITE ALL I-SPACES IN THIS MODE
013630 042707 160000 BIC #160000, PC ;CHANGE TO BANK 0 PC
013634 012702 017776 MOV #17776, R2 ;SETUP R2 TO REFERENCE THE FIRST D-SPACE PAGE
013640 011212 LOP36B: MOV @R2, @R2 ;WRITE INTO D-SPACE
013642 062702 020000 ADD #20000, R2 ;CHANGE R2 TO REFERENCE THE NEXT PAGE
013646 103374 LOP36B: BCC LOP36B ;REFERENCE ALL PAGES IN THIS MODE
013650 005077 167156 CLR @SRO ;TURN OFF KT11-C
013654 020027 003400 CMP R0, #STAEND
013660 002745 BLT LOOP36 ;SET "A" BITS IN ALL MODES
013662 005037 177776 CLR @#PS ;RETURN TO KERNEL
013666 005277 167140 INC @SRO ;TURN ON KT11-C
013672 105777 167120 TSTB @TCSR ;WAIT FOR ANY TTY OUTPUT TO FINISH
013676 100375 BPL .-4
013700 000005 RESET ;ISSUE INIT WITH KT11-C ON

```

```

013702 000005          RESET          ;ISSUE INIT WITH KT11-C OFF
013704 005077 167122  CLR          @SR0          ;MAKE SURE KT11-C IS OFF
013710 012701 003052  MOV          #ADRTAB,R1      ;R1 POINTS TO ADDRESS OF 1ST PDR
013714 012702 000020  LOP36C: MOV      #20,R2      ;R2 INDICATES WHEN TO CHANGE REGISTER SETS
013720 032771 000200 000000 LOP36D: BIT      #200,@(R1)  ;CHECK THE "A" BIT
013726 001001          BNE          .+4          ;OK IF STILL SET
013730 104006          HLT          ;RESET CLEARED "A" BIT IN
                                ;PDR WHOSE ADDRESS IS
                                ;POINTED TO BY R1
                                ;MOVE POINTER
013732 005721          TST          (R1)+        ;CHECK ALL PDR'S IN THIS SET
013734 077207          SOB          R2,LOP36D
013736 062701 000040  ADD          #40,R1
013742 020127 003350  CMP          R1,#ADREND
013746 002762          BLT          LOP36C
013750 005077 167072  CLR          @SR3          ;CHECK ALL 3 SETS
                                ;DISABLE ALL D-SPACES

;SHOW THAT A DATO TO A PDR WILL CLEAR THE A AND W BITS
;EVEN WHEN THE INSTRUCTION ALSO CAUSES A TRAP REFERENCE TO
;THE CORRESPONDING PAGE
;MAP KERNEL PAGE 1 RRWT AND MAKE A WRITE ACCESS TO PAGE 1
;TO SET THE A AND W BITS
;THEN LOAD THE PDR, MAKING A TRAP REFERENCE TO PAGE 1 IN THE SOURCE
;FETCH OF THE SAME INSTRUCTION-THE A AND W BITS SHOULD BE CLEARED DUE
;TO THE DATO TO THE PDR
013754 104400          TEST37: SCOPE
013756 012737 000037 177570  MOV          #37,@#SR      ;DISPLAY TEST NUMBER
013764 005037 177776  CLR          @#PS          ;INITIALIZE PROCESSOR STATUS
013770 012706 001000  MOV          #KSTACK,SP    ;INITIALIZE KERNEL STACK POINTER
013774 005077 167046  CLR          @SR3          ;INITIALIZE SR3
014000 005077 167026  CLR          @SR0          ;INITIALIZE SR0
014004 026727 167410 000037  CMP          TESTCT,#37    ;IS THIS TEST BEING EXECUTED IN THE
014012 001401          BEQ          .+4          ;CORRECT SEQUENCE? - BRANCH IF YES
014014 104006          HLT          ;TEST EXECUTED OUT OF ORDER - TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

014016 004767 000522          JSR          %7,RWALL      ;INITIALIZE ALL PAGES RW, BANK 0
014022 012777 007600 167300  MOV          #7600,@KIPAR7 ;MAP KERNEL PAGE 7 TO THE EXTERNAL BANK
014030 012777 077404 167216  MOV          #77404,@KIPDR1 ;MAKE KERNEL PAGE 1 RRWT
014036 012777 000001 166766  MOV          #1,@SR0       ;TURN ON KT11-C
014044 013737 020000 020000  MOV          @#20000,@#20000 ;READ AND WRITE PAGE 1
014052 022777 077704 167174  CMP          #77704,@KIPDR1 ;CHECK PDR OF PAGE 1
014060 001401          BEQ          .+4
014062 104006          HLT          ;KERNEL I-SPACE PAGE 1 PDR
                                ;INCORRECT - A AND W BITS SHOULD
                                ;BE SET DUE TO PREVIOUS MOVE INSTRUCTION
014064 012767 077704 166730  MOV          #77704,TEMP    ;LOAD TEMP WITH VALUE TO BE MOVED TO KIPDR1
014072 016777 006724 167154  MOV          TEMP+20000,@KIPDR1 ;PAGE 1 REFERENCE SHOULD SET
                                ;THE A BIT, BUT DATO TO THE PDR CLEARS A AND W BITS
014100 022777 077404 167146  CMP          #77404,@KIPDR1 ;CHECK PAGE 1 PDR
014106 001401          BEQ          .+4
014110 104006          HLT          ;PDR INCORRECT - A AND W BITS
014112 005077 166714  CLR          @SR0          ;SHOULD HAVE BEEN CLEARED

;CHECK TO SEE THAT MULTIPLE ACCESSES TO A PAGE AFTER SETTING THE
;A AND W BITS DON'T CLEAR THE A AND W BITS
014116 104400          TEST40: SCOPE

```

```

014120 012737 000040 177570      MOV    #40, @#SR      ; DISPLAY TEST NUMBER
014126 005037 177776             CLR    @#PS          ; INITIALIZE PROCESSOR STATUS
014132 012706 001000             MOV    #KSTACK, SP  ; INITIALIZE KERNEL STACK POINTER
014136 005077 166704             CLR    @SR3         ; INITIALIZE SR3
014142 005077 166664             CLR    @SR0         ; INITIALIZE SR0
014146 026727 167246 000040      CMP    TESTCT, #40   ; IS THIS TEST BEING EXECUTED IN THE
014154 001401             BEQ    .+4          ; CORRECT SEQUENCE? - BRANCH IF YES
014156 104006             HLT                ; TEST EXECUTED OUT OF ORDER - TESTCT
                                ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
    
```

```

014160 012767 000010 001016      MOV    #10, ICOUNT   ; DROP ITERATION COUNT
014166 004767 000352             JSR    %7, RWALL     ; INITIALIZE ALL PAGES 4K, RW, BANK 0
014172 012777 007600 167130      MOV    #7600, @KIPAR7 ; MAP KERNEL PAGE 7 TO THE EXTERNAL BANK
014200 012777 077405 167046      MOV    #77405, @KIPDR1 ; MAP KERNEL PAGE 1 RRWTW
014206 012777 000001 166616      MOV    #1, @SR0      ; TURN ON KT11-C
014214 013737 020000 020000      MOV    @#20000, @#20000 ; READ AND WRITE PAGE 1
014222 022777 077705 167024      CMP    #77705, @KIPDR1 ; CHECK THE PDR
014230 001401             BEQ    .+4
014232 104006             HLT                ; KERNEL I-SPACE PDR1 INCORRECT
                                ; A AND W BITS SHOULD BE SET
                                ; POINT R1 TO PAGE 1
    
```

```

014234 012701 020000             MOV    #20000, R1    ; POINT R1 TO PAGE 1
014240 012702 000100             MOV    #100, R2
014244 005721             L40:  TST    (R1)+    ; READ PAGE 1 REPEATEDLY
014246 077202             SOB    R2, L40
014250 005077 166556             CLR    @SR0          ; TURN OFF KT11-C
014254 022777 077705 166772      CMP    #77705, @KIPDR1 ; CHECK A AND W BITS AGAIN
014262 001401             BEQ    .+4
014264 104006             HLT                ; KERNEL I-SPACE PDR 1
                                ; INCORRECT AFTER REPEATEDLY READING PAGE 1
    
```

; SHOW THAT IF KT11-C IS ON, SETTING THE CURRENT MODE TO 10 WILL
 ; CAUSE A MEMORY MANAGEMENT ABORT. NON RESIDENT SHOULD BE SET, AND ALSO PL SHOULD
 ; BE SET IF THE REFERENCE IS OUTSIDE THE FIRST BLOCK

```

014266 104400             TEST41: SCOPE
014270 012737 000041 177570      MOV    #41, @#SR      ; DISPLAY TEST NUMBER
014276 005037 177776             CLR    @#PS          ; INITIALIZE PROCESSOR STATUS
014302 012706 001000             MOV    #KSTACK, SP  ; INITIALIZE KERNEL STACK POINTER
014306 005077 166534             CLR    @SR3         ; INITIALIZE SR3
014312 005077 166514             CLR    @SR0         ; INITIALIZE SR0
014316 026727 167076 000041      CMP    TESTCT, #41   ; IS THIS TEST BEING EXECUTED IN THE
014324 001401             BEQ    .+4          ; CORRECT SEQUENCE? - BRANCH IF YES
014326 104006             HLT                ; TEST EXECUTED OUT OF ORDER - TESTCT
                                ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
    
```

```

014330 012767 002000 000646      MOV    #2000, ICOUNT ; RESTORE ITERATION COUNT
014336 004767 000202             JSR    %7, RWALL     ; INITIALIZE ALL PAGES 4K, RW, BANK 0
014342 012777 007600 166760      MOV    #7600, @KIPAR7 ; MAP KERNEL 7 TO EXTERNAL BANK
014350 012777 014416 167032      MOV    #RET41, @KTVEC ; SETUP MEMORY MANAGEMENT ABORT RETURN
014356 005077 167030             CLR    @KTSTA
014362 012777 000001 166442      MOV    #1, @SR0      ; TURN ON KT11-C
014370 012737 100000 177776      MOV    #100000, @#PS ; SET MODE TO 10-FETCH OF NEXT
014376 000240             ADD41: NOP           ; INSTRUCTION SHOULD ABORT
014400 005037 177776             CLR    @#PS          ; RESTORE MODE TO KERNEL
014404 042777 000001 166420      BIC    #1, @SR0      ; TURN OFF KT11-C
014412 104006             HLT                ; NO ABORT WHEN MODE WAS SET
014414 000415             BR     CONT41       ; TO 10 (ILLEGAL) WITH KT11-C ON
    
```

```

014416 042777 000001 166406 RET41: BIC      #1,SR0      ;TURN OFF KT11-C AFTER ABORT
014424 022777 140100 166400      CMP      #140100,SR0 ;CK SR0
014432 001401      BEQ      .+4
014434 104006      HLT
                                ;SR0 INCORRECT AFTER MODE 10 ABORT
                                ;NR,PL,AND MODE 10 SHOULD BE SET
014436 022777 014376 166376      CMP      #ADD41,SR2 ;CHECK SR2
014444 001401      BEQ      .+4
014446 104006      HLT
                                ;SR2 INCORRECT - SHOULD CONTAIN
                                ;ADDRESS OF THE INSTRUCTION
                                ;IMMEDIATELY AFTER THE ONE SETTING
                                ;THE MODE TO 10
                                ;REINITIALIZE SR0
                                ;RESTORE TRAP CATCHER
014450 005077 166356      CONT41: CLR      SR0
014454 016777 166732 166726      MOV      KTSTA,KTVEC
014462 104400      SCOPE
014464 004767 001174      JSR      %7,BELL
014470 013701 000042      MOV      #42,R1      ;MONITOR HOOK
014474 001405      BEQ      END
014476 000005      RESET
014500 004711      LOGIC: JSR      %7,SR1
014502 000240      NOP
014504 000240      NOP
014506 000240      NOP
014510 000167 166706      END:    JMP      START
                                ;SUBROUTINE TO CLEAR ALL KT11-C REGISTERS (EXCEPT SR1,SR2)
014514 005077 166312      CLRALL: CLR      SR0
014520 005077 166322      CLR      SR3
014524 005000      CLR      R0
014526 012701 000140      MOV      #96,R1      ;R0 IS USED TO INDEX THRU THE ADDRESS TABLE
014532 005070 003052      CLRLP: CLR      @ADRTAB(R0) ;COUNT OF REGISTERS TO BE CLEARED
014536 005720      TST      (R0)+      ;CLEAR REGISTERS THRU ADDRESS TABLE
014540 077104      SOB      R1,CLRLP   ;MOVE POINTER
014542 000207      RTS      %7        ;LOOP TILL DONE

                                ;SUBROUTINE TO MAKE ALL PAGES RW, BANK 0, 4K, UP
014544 005077 166262      RWALL: CLR      SR0
014550 012701 003052      MOV      #ADRTAB,R1 ;R1 POINTS TO ADDRESS OF PDR
014554 012700 000020      RWL1:  MOV      #20,R0 ;R0 KEEPS TRACK OF WHEN TO CHANGE REGISTER SETS
014560 005071 000040      RWL2:  CLR      @40(R1) ;CLEAR PAR
014564 012731 077406      MOV      #77406,@(R1)+ ;SET PDR TO 4K,RW
014570 077005      SOB      R0,RWL2
014572 062701 000040      ADD      #40,R1
014576 020127 003350      CMP      R1,#ADREND ;CHECK FOR END OF TABLE
014602 002764      BLT      RWL1
014604 000207      RTS      %7

                                ;SUBROUTINE TO MAKE ALL I SPACE PAGES RW, BANK 0,4K,UP
014606 005077 166220      RWISP: CLR      SR0
014612 012701 003052      MOV      #ADRTAB,R1 ;R1 POINTS TO ADDRESS OF I-SPACE PDR
014616 012700 000010      RWI1:  MOV      #10,R0
014622 005071 000040      RWI2:  CLR      @40(R1) ;CLEAR PAR
014626 012731 077406      MOV      #77406,@(R1)+ ;MAP PDR RW, 4K
    
```


014632 077005
 014634 062701 000060
 014640 020127 003350
 014644 002764
 014646 000207

SOB RO,RW12
 ADD #60,R1
 CMP R1,#ADREND ;CHECK FOR DONE
 BLT RW11 ;BRANCH IF NOT
 RTS %7

014650 005077 166156
 014654 012701 003072
 014660 012700 000010
 014664 005071 000040
 014670 012731 077406
 014674 077005
 014676 062701 000060
 014702 020127 003350
 014706 002764
 014710 000207

;SUBROUTINE TO MAKE ALL D-SPACE PAGES RW, BANK 0, 4K, UP
 RWDSP: CLR @SRO
 MOV #ADRTAB+20,R1 ;R1 POINTS TO ADDRESS OF D-SPACE PDR
 RWD1: MOV #10,RO
 RWD2: CLR @40(R1) ;CLEAR PAR
 MOV #77406,@(R1)+ ;MAP PDR RW,4K
 SOB RO,RWD2
 ADD #60,R1
 CMP R1,#ADREND ;CHECK FOR DONE
 BLT RWD1 ;BRANCH IF NOT
 RTS %7

014712 005077 166114
 014716 012701 003052
 014722 012700 000020
 014726 005071 000040
 014732 012731 077405
 014736 077005
 014740 062701 000040
 014744 020127 003350
 014750 002764
 014752 012777 007600 166370
 014760 012777 007600 166262
 014766 012777 007600 166154
 014774 000207

;ROUTINE TO MAKE ALL PAGES READ/WRITE AND TRAP ON WRITE
 ;ALL PAGES ARE MAPPED 4K, BANK 0 EXCEPT FOR D-SPACE PAGE 7,
 ;ALL OF WHICH ARE MAPPED TO THE EXTERNAL BANK
 RWTWAL: CLR @SRO
 MOV #ADRTAB,R1 ;R1 POINTS TO ADDRESS OF PDR
 RWTW1: MOV #20,RO
 RWTW2: CLR @40(R1) ;CLEAR PAR
 MOV #77405,@(R1)+ ;MAP PDR RWTW,4K
 SOB RO,RWTW2
 ADD #40,R1
 CMP R1,#ADREND
 BLT RWTW1
 MOV #7600,@KDPAR7 ;MAP PAGE 7, ALL MODES,
 MOV #7600,@SDPAR7 ;TO THE EXTERNAL BANK
 MOV #7600,@UDPAR7
 RTS %7

014776 005037 177776
 015002 012706 001000
 015006 000000
 015010 016767 162554 000036
 015016 062767 000002 000030
 015024 000000
 015026 005067 000154
 015032 012767 015044 000150
 015040 000177 000010
 015044 005067 000136
 015050 000177 000000
 015054 000000

;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
 ;LOAD THE STARTING ADDRESS OF THE TEST
 ;YOU WISH TO RUN (THE ADDRESS OF THE TESTXX
 ;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
 ;OPTIONS AT THE 2ND HALT.
 ;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
 TESTXX: CLR @#PS
 MOV #KSTACK,SP
 HALT ;WAIT FOR STARTING ADDRESS
 MOV SR,RETRNX ;LOAD STARTING ADDRESS IN RETRNX
 ADD #2,RETRNX ;ADD 2 TO POINT TO INSTRUCTION AFTER
 HALT ;SET SR OPTIONS
 CLR SCOPEF ;KEEP COUNT AT ZERO
 MOV #XLOOP,RETURN ;LOAD SCOPE LOOP RETURN POINTER
 JMP @RETRNX ;JUMP TO TEST
 XLOOP: CLR SCOPEF ;KEEP COUNT AT ZERO
 JMP @RETRNX ;JUMP TO TEST
 RETRNX: 0

;SCOPE AND/OR ITERATION LOOP FOR EACH TEST

```

015056 032737 040000 177570 SCOPEC: BIT #40000,@#SR ;TEST SR FOR SCOPE
015064 001015 BNE SCOPEB ;YES SCOPE
015066 032737 004000 177570 BIT #4000,@#SR ;NO-TEST FOR ITERATION
015074 001025 BNE SCOPEG ;INHIBIT ITERATION
015076 026767 000104 000100 CMP SCOPEF,ICOUNT ;COMPARE CURRENT COUNT TO MAX NUMBER
015104 100021 BPL SCOPEG ;EXIT-DONE
015106 005267 000074 INC SCOPEF ;INCREMENT COUNT
015112 012737 000340 177776 MOV #340,@#PS ;PREVENT TRAPPING WHILE MOVING STACK
015120 022606 SCOPEB: CMP (6)+,%6 ;REPOSITION STACK
015122 012637 177776 MOV (6)+,@#PS ;RESTORE PREVIOUS PROCESSOR STATUS
015126 032737 000400 177570 BIT #400,@#SR ;LOAD MICROBREAK REGISTER?
015134 001403 BEQ .+10 ;NO-BRANCH
015136 113777 177570 166250 MOVB @#SR,@UBRK ;YES-LOAD FROM LOW BYTE OF SR
015144 000177 000040 JMP @RETURN ;REPEAT TEST
015150 005067 000032 SCOPEG: CLR SCOPEF ;CLEAR COUNT
015154 005267 166240 INC TESTCT ;STEP TEST COUNTER TO ALLOW CHECKING ORDER OF EXECUTION
015160 011667 000024 MOV @%6,RETURN ;SAVE SCOPE RETURN POINTER
015164 032737 000400 177570 BIT #400,@#SR
015172 001403 BEQ .+10
015174 113777 177570 166212 MOVB @#SR,@UBRK
015202 000002 RTI ;RETURN INLINE-NEXT TEST
015204 002000 ICOUNT:2000 ;ITERATION COUNT
015206 000000 SCOPEF:0 ;COUNT LOCATION FOR ITERATION LOOP
015210 000000 RETURN:0 ;ADDRESS OF LAST TEST
    
```

```

;ENTERED WITH SYSTEM TRAP CALL (HLT)
;PRINT OUT THE ERROR PC+2 AND STATUS REGISTER
015212 012767 000340 162556 PRINT: MOV #340,PS ;SET PRIORITY TO 7
015220 036727 162344 020000 BIT SR,#20000 ;TEST FOR INHIBIT PRINT OUT
015226 001401 BEQ .+4 ;BRANCH TO PRINT
015230 000432 BR CK ;INHIBIT, CHECK FOR HALT
015232 012667 000100 MOV (6)+,SAVPC ;PC OF FAILING ROUTINE
015236 012667 000076 MOV (6)+,SAVPSR ;PSR OF ERROR CONDITION
015242 024646 CMP -(6),-(6) ;RESTORE STACK
015244 012767 000200 162524 MOV #200,PS
015252 004767 000426 JSR %7,CALF ;OUTPUT CARRIAGE RETURN AND LINE FEED
015256 016767 000054 000322 MOV SAVPC,PTEMP1 ;LOAD WITH FAILING PC+2
015264 004767 000104 JSR %7,PROCT ;PRINT FAILING PC+2
015270 105777 165522 TSTB @TCSR ;WAIT FOR TTY READY
015274 100375 BPL .-4
015276 012777 000240 165514 MOV #240,@TDBR ;OUTPUT A SPACE
015304 016767 000030 000274 MOV SAVPSR,PTEMP1 ;LOAD PROCESSOR STATUS
015312 004767 000056 JSR %7,PROCT ;PRINT PROCESSOR STATUS
015316 005767 162246 CK: TST SR ;CHECK SR FOR HALT SWITCH
015322 100001 BPL .+4 ;BRANCH IF NOT SET
015324 000000 HALT ;HALT ON ERROR UP
015326 000002 RTI ;RETURN TO MAIN LINE
015330 000000 SAVR2:0
015332 000000 SAVR3:0
015334 000000 SAVR4:0
015336 000000 SAVPC:0
015340 000000 SAVPSR:0
    
```

;SUBROUTINE TO PRINT OUT OCTAL NUMBER
 ;PRSHRT DELETES LEADING ZEROS

```

;PROCT PRINTS OUT 6 OCTAL DIGITS
015342 012767 000001 000232 PRSHRT: MOV #1,PRSFLG ;SET FLAG TO INDICATE SHORT PRINTOUT
015350 005767 000232 TST PTEMP1 ;CHECK FOR ZERO
015354 001011 BNE PROCT+4 ;BRANCH IF NOT ZERO
015356 105777 165434 TSTB @TCSR ;WAIT FOR TTY READY
015362 100375 BPL .-4
015364 012777 000260 165426 MOV #260,@TDBR ;OUTPUT A SINGLE ZERO
015372 000207 RTS %7 ;RETURN
015374 005067 000202 PROCT: CLR PRSFLG ;CLEAR FLAG TO INDICATE FULL PRINTOUT
015400 005067 000206 CLR PTEMP3 ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
015404 005067 000174 CLR PRFLG ;INITIALIZE CARRY FLAG FOR ROTATES
015410 012767 000260 000172 MOV #260,PTEMP2 ;SETUP R3
015416 005767 000164 TST PTEMP1 ;CHECK BIT 15 OF NUMBER
015422 100002 BPL .+6 ;BRANCH IF ZERO
015424 005267 000160 INC PTEMP2 ;INCREMENT R3 IF ONE
015430 006167 000152 ROL PTEMP1 ;ROTATE LEFT MOST OCTAL TO RIGHT END
015434 006167 000146 ROL PTEMP1
015440 005567 000140 ADC PRFLG ;STORE CARRY
015444 005767 000132 P.CK: TST PRSFLG ;CHECK FOR SHORT PRINTOUT
015450 001404 BEQ P.WAIT ;BRANCH IF NOT SET
015452 026727 000132 000260 CMP PTEMP2,#260 ;CHECK FOR ZERO IF SET
015460 001410 BEQ P.CONT ;IF SET, GO TO NEXT CHARACTER
015462 105777 165330 P.WAIT: TSTB @TCSR ;WAIT FOR TTY READY
015466 100375 BPL P.WAIT
015470 016777 000114 165322 MOV PTEMP2,@TDBR ;OUTPUT NEXT CHARACTER
015476 005067 000100 CLR PRSFLG ;PRINT REST OF NUMBER AFTER A NON-ZERO DIGIT
015502 005267 000104 P.CONT: INC PTEMP3 ;COUNT
015506 005727 000100 000006 CMP PTEMP3,#6 ;CHECK FOR DONE
015514 001001 BNE P.CNT1 ;BRANCH IF NOT DONE
015516 000207 RTS %7
015520 000241 P.CNT1: CLC ;CLEAR CARRY
015522 005767 000056 TST PRFLG ;CHECK FOR PREVIOUS CARRY
015526 001403 BEQ .+10 ;BRANCH IF PREVIOUSLY ZERO
015530 005067 000050 CLR PRFLG ;INITIALIZE FLAG
015534 000261 SEC ;SET CARRY
015536 006167 000044 ROL PTEMP1 ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTER
015542 006167 000040 ROL PTEMP1
015546 006167 000034 ROL PTEMP1
015552 005567 000026 ADC PRFLG ;STORE CARRY
015556 016767 000024 000024 MOV PTEMP1,PTEMP2 ;LOAD DATA INTO R3
015564 042767 177770 000016 BIC #177770,PTEMP2 ;CLEAR ALL BUT LOWEST OCTAL DIGIT
015572 052767 000260 000010 BIS #260,PTEMP2 ;SET TO ASCII EQUIVALENT
015600 000721 BR P.CK ;LOOP
015602 000000 PRSFLG: 0
015604 000000 PRFLG: 0
015606 000000 PTEMP1: 0 ;CONTAINS VALUE TO BE OUTPUT
015610 000000 PTEMP2: 0 ;SCRATCH
015612 000000 PTEMP3: 0 ;USED TO COUNT CHARACTERS OUTPUT

;EMT HANDLER
;FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES
015614 011667 000032 EMTSRV: MOV @SP,EPC ;GET CALL
015620 162767 000002 000024 SUB #2,EPC
015626 017767 000020 000016 MOV @EPC,EPC
015634 105067 000013 CLRB EPC+1 ;SAVE OFFSET ONLY
015640 062767 015654 000004 ADD #EMTAB,EPC ;POINT TO TABLE OF ADDRESSES
    
```

```

015646 017707 000000
015652 000000
          104000
          104002
          104004
          104006
015654 104000
015656 104002
015660 104004
015662 015212

          MOV      @EPC,PC      ;JUMP TO DESIRED ROUTINE
EPC:    0
          PATCH1=EMT+0
          PATCH2=EMT+2
          PATCH3=EMT+4
          HLT      =EMT+6
EMTAB:  PATCH1
          PATCH2
          PATCH3
          PRINT

          ;BELL ON PASS COMPLETE
015664 105777 165126
015670 100375
015672 012777 000207 165120
015700 077001
015702 000207

          BELL:   TSTB      @TCSR
          BPL      .-4
          MOV      #207,@TDBR
1$:     SOB      %0,      #1$
          RTS      %7

          ;SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED
015704 105777 165106
015710 100375
015712 012777 000215 165100
015720 105777 165072
015724 100375
015726 012777 000212 165064
015734 000207
          017712

          CRLF:   TSTB      @TCSR      ;WAIT FOR TTY READY
          BPL      .-4
          MOV      #215,@TDBR      ;ROUTPUT CARRIAGE RETURN
          TSTB      @TCSR      ;WAIT FOR TTY READY
          BPL      .-4
          MOV      #212,@TDBR      ;OUTPUT LINEFEED
          RTS      %7
          .-17712
          ;RETURN

017712 125252
          000001

          DESTAD: 125252
          .END

```

ADD14	006454	971#	977												
ADD41	014376	2009#	2019												
ADREND=	003350	421#	1634	1653	1712	1914	2060	2072	2084	2098					
ADRTAB	003052	322#	1620	1642	1699	1904	2047*	2054	2066	2078	2092				
ADR14	006466	973#	986												
ADR22	007772	1196	1198#	1210	1214										
ADR27A	011276	1456	1462#												
ADR3	004160	560	566#												
ADR32	012174	1602	1606#												
ADR33A	012522	1687	1691#												
ADR33B	012642	1688	1721#												
ADR36	013616	1883	1886#												
ADR7	005164	737	738#												
AD21	007564	1159#	1160												
BELL	015664	2030	2250#												
CK	015316	2157	2170#												
CKWBIT	011374	1465	1474	1483#											
CLRALL	014514	863	1187	1424	2043#										
CLRLP	014532	2047#	2049												
CNT33A	012576	1702	1709#												
CONT13	006230	914	925#												
CONT2	004022	521	536#												
CONT23	010306	1262	1268	1281#											
CONT24	010556	1329	1335	1346#											
CONT41	014450	2013	2025#												
CRLF	015704	2162	2257#												
C25	010756	1391#	1405												
DESTAD	017712	2266#													
DONE10	005530	775	789	802	810#										
DONE13	006326	940	945#												
DONE14	006546	975	981	984	994#										
DONE15	006704	1022	1025#												
DONE16	007122	1063	1066	1070#											
DONE17	007276	1098	1103#												
DONE20	007446	1129	1136#												
DONE21	007636	1166	1172#												
DONE22	010054	1200	1220#												
DONE23	010320	1254	1286#												
DONE24	010570	1321	1351#												
DONE25	011024	1397	1408#												
DONE3	004222	568	581#												
DONE6	004770	684	699#												
DONE7	005220	728	741	748#											
EMTAB	015654	2237	2244#												
EMTSRV	015614	282	2233#												
END	014510	2033	2040#												
EPC	015652	2233*	2234*	2235*	2236*	2237*	2238	2239#							
ERR14	006472	965	976#												
HLT =	104006	469	475	481	499	525	530	534	549	571	580	596	614	621	
		635	644	651	665	690	714	727	733	746	761	774	780	788	
		794	801	807	822	838	860	879	900	913	920	923	935	939	
		943	959	979	982	988	992	1007	1020	1038	1053	1057	1061	1065	
		1068	1083	1096	1102	1115	1128	1134	1149	1162	1165	1170	1184	1199	
		1204	1208	1212	1216	1235	1258	1265	1274	1279	1303	1325	1332	1341	
		1345	1366	1388	1404	1420	1428	1441	1485	1491	1505	1515	1519	1526	
		1532	1545	1555	1559	1566	1572	1585	1626	1647	1672	1696	1705	1717	

UDPAR4	003142	351#																		
UDPAR5	003144	352#																		
UDPAR6	003146	353#																		
UDPAR7	003150	354#	1448*	1592*	1877*	2102*														
UDPDR0	003072	331#																		
UDPDR1	003074	332#	768*																	
UDPDR2	003076	333#																		
UDPDR3	003100	334#																		
UDPDR4	003102	335#																		
UDPDR5	003104	336#																		
UDPDR6	003106	337#																		
UDPDR7	003110	338#	600*																	
UIPAR0	003112	339#																		
UIPAR1	003114	340#																		
UIPAR2	003116	341#																		
UIPAR3	003120	342#																		
UIPAR4	003122	343#																		
UIPAR5	003124	344#																		
UIPAR6	003126	345#																		
UIPAR7	003130	346#																		
UIPDR0	003052	323#	434																	
UIPDR1	003054	324#																		
UIPDR2	003056	325#																		
UIPDR3	003060	326#																		
UIPDR4	003062	327#																		
UIPDR5	003064	328#																		
UIPDR6	003066	329#																		
UIPDR7	003070	330#																		
USTACK	003000	302#	453	607	1452	1596	1682	1764												
XLOOP	015044	2118	2120#																	
.	= 017714	279#	281#	284#	287#	289#	293#	297#	299#	301#	421	458	468	474						
		477	480	498	511	519	524	529	533	548	570	576	579	595						
		610	613	617	620	634	643	650	664	682	689	713	732	745						
		760	779	793	806	821	837	859	875	878	899	919	922	934						
		938	958	978	987	991	1006	1037	1052	1056	1060	1082	1101	1114						
		1133	1148	1161	1169	1183	1203	1207	1211	1215	1234	1273	1278	1302						
		1340	1344	1365	1387	1403	1419	1427	1440	1484	1490	1504	1514	1518						
		1525	1531	1544	1554	1558	1565	1571	1584	1625	1639	1646	1671	1695						
		1704	1716	1725	1731	1754	1775	1781	1788	1794	1814	1825	1829	1839						
		1847	1857	1870	1900	1907	1933	1943	1951	1964	1975	1984	1998	2016						
		2020	2137	2144	2156	2166	2171	2187	2195	2214	2251	2258	2261	2264#						

M04

DCKTB-C MACY11 30(1046) 28-NOV-77 11:26 PAGE 53
DCKTBC.P11 28-NOV-77 11:17

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0051

TESTNO	446#	462	492	542	589	628	658	707	754	815	853	893	952	1000	1031
	1076	1109	1142	1177	1228	1296	1359	1413	1434	1498	1538	1578	1665	1748	1808
	1864	1927	1958	1992											

. ABS. 017714 000

ERRORS DETECTED: 0

DCKTBC.BIN,DCKTBC.LST/CRF/SOL/NL:TOC=DCKTBC.P11
RUN-TIME: 24.7 SECONDS
RUN-TIME RATIO: 251/8=29.9
CORE USED: BK (15 PAGES)

N04