

# KT11-D

LOGIC TEST  
MD-11-DBKTA-D

EP-DBKTA-D-DL-B  
COPYRIGHT © 1976  
FICHE 1 OF 1

DEC 1976  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames, each displaying logic test data. The data is organized into columns and rows, with some frames containing numerical values and others containing graphical representations of test results. The text is small and difficult to read due to the low resolution of the scan.



B01

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 1  
DBKTAD.P11 13-SEP-76 10:29

SEQ 0001

.REM\*

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DBKTA-D-D  
PRODUCT NAME: KT11-D BASIC LOGIC TEST  
DATE REVISED: DECEMBER 1976  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: ROBERT WHITTON/STAN HARACKIEWICZ

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH A SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1972, 1976 BY DIGITAL EQUIPMENT CORPORATION



MAINDEC-11-DBKTA-D  
PAGE 02

## 1.0 ABSTRACT

THIS PROGRAM INCREMENTALLY TESTS THE BASIC LOGIC FUNCTIONS OF THE KT11-D MEMORY MANAGEMENT OPTION FOR THE PDP-11/40. 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-11/40 WITH KT11-D 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 12=1 OR UP -- INHIBIT BELL AT END OF PASS, TYPE ASTERICK  
 SW 12=0 OR DOWN -- RING BELL AT END OF EACH PASS  
 SW 11=1 OR UP -- INHIBIT ITERATIONS  
 SW 10=1 OR UP-- HALT AT END OF CURRENT TEST  
 WITH NEXT TEST NUMBER IN DATA LIGHTS

## 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.



# E01

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 4  
DBKTAD.P11 13-SEP-76 10:29

SEQ 0004

MAINDEC-11-DBKTA-D  
PAGE 04

## 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-D, AS WELL AS SRO.

## 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.

## 5.3 PROGRAM AND/OR OPERATOR ACTION

THE PROGRAM FIRST CHECKS THOSE PROPERTIES OF THE KT11-D WHICH CAN BE TESTED WITH MEMORY MANAGEMENT TURNED OFF. THEN, DESTINATION ONLY RELOCATION IS USED TO SHOW THAT BASIC RELOCATION IS WORKING CORRECTLY. FINALLY, FULL RELOCATION IS ENABLED AND MISCELLANEOUS ASPECTS OF THE KT11-D'S OPERATION ARE CHECKED.



## 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-D DIAGNOSTICS TO INDICATED A FAILURE WHEN A HALT OR A HLT WORD TRAP CALL COULD LEAD TO 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 USER STACK POINTER IS INITIALIZED TO 400.



MAINDEC-11-DBKTA-D  
PAGE 06

## 8.4 EXECUTION ORDER CHECKING

SINCE THE KT11-D 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

THE PROGRAM INITIALLY TESTS THOSE FEATURES OF THE KT11-D OPTION WHICH CAN BE TESTED WITHOUT TURNING ON MEMORY MANAGEMENT. IT THEN USES THE MAINTENANCE MODE (DESTINATION ONLY RELOCATION) TO TEST TURNING MEMORY MANAGEMENT ON AND OFF AND TO FULLY CHECK OUT RELOCATION. ONCE RELOCATION HAS BEEN FULLY TESTED, FULL PAGING IS USED TO TEST THE REMAINING OPERATIONS OF THE OPTION.

\*



```
:BASIC LOGIC TEST OF THE KT11-D
:COPYRIGHT 1972, 1973, 1974, 1975, 1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
:THIS PROGRAM WAS REVISED ON SEPTEMBER 30, 1974 TO CHECK FOR THE
:IMPLEMENTATION OF ECO #M-7236-00005. THE ECO WAS NEEDED TO ALLOW THE
:RELOCATED REFERENCE TO THE USER PAGE ADDRESS AND PAGE DESCRIPTOR REGISTERS
:WITH BIT SIX OF THE VIRTUAL ADDRESS EQUAL TO A ONE. THE REVISION
:WAS ACCOMPLISHED BY ADDING TEST NUMBER 61.
```

```
:OPERATING INSTRUCTIONS
: 1. LOAD TEST USING THE ABSOLUTE LOADER
: 2. LOAD SA 200
: 3. SET SR TO INITIAL SETTINGS
: 4. PRESS START
```

```
:SW15=1 CAUSES HALT ON ERROR
:SW14=1 CAUSES SCOPE LOOPING
:SW13=1 INHIBITS ERROR PRINTOUT
:SW11=1 INHIBITS ITERATIONS
:SW10=1 HALT AT END OF CURRENT TEST WITH TEST NUMBER IN DATA LIGHTS OF NEXT
:TEST. PRESS CONTINUE TO ADVANCE TO NEXT TEST. (WITH SW11=1)
```

```
:DEFINITIONS
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
STATUS=PS
HLT=104006
BIT0=1
BIT1=2
BIT2=4
BIT3=10
BIT4=20
BIT5=40
BIT6=100
BIT7=200
BIT8=400
BIT9=1000
BIT10=2000
BIT11=4000
BIT12=10000
BIT13=20000
BIT14=40000
BIT15=100000
```

```
104400
000240
000000
000001
000002
000003
000004
000005
000006
000007
000006
000007
177570
177776
177776
104006
000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000
```



# IO1

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 8  
 DBKTAD.P11 13-SEP-76 10:28

SEQ 0008

```

;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
  
```

```

;LOAD VECTOR AREA
  
```

000030	000030				
	016246				.=30
000032	000340				EMTSRV
	000034				340
000034	015456				.=34
000036	000000				SCOPEC
	000046				0
000046	015156				.=46
					LOGIC

```

;LOAD STARTING AREA
  
```

000200	000200				
	000167	000774			.=200
	000210				JMP START
000210	000167	015144			.=210
					JMP TESTX

```

;LOAD DATA AREA
  
```

000400	000400				
	000000				.=400
	001000				USTACK: 0
001000	000000				.+.376
001002	000000	000000	000000		KSTACK: 0
001010	000000				.WORD 0,0,0,0
001012	123456				K123: 123456
001014	134567				K134: 134567
001016	177564				TCSR: 177564
001020	177566				TDBR: 177566
001022	000000				TEMP: 0

```

;KT11-D STATUS REGISTER ADDRESSES
SR0: 177572
SR0H: 177573
SR1: 177574
SR2: 177576
  
```

001024	177572
001026	177573
001030	177574
001032	177576

```

;ADRTAB:
  
```

001034	177600				
001034	177600				UPDR0: 177600
001036	177602				UPDR1: 177602
001040	177604				UPDR2: 177604
001042	177606				UPDR3: 177606
001044	177610				UPDR4: 177610
001046	177612				UPDR5: 177612
001050	177614				UPDR6: 177614
001052	177616				UPDR7: 177616

```

;USER PAGE DESCRIPTOR REGISTERS
  
```

```

;
  
```

001054	177640				
001054	177640				UPAR0: 177640
001056	177642				UPAR1: 177642
001060	177644				UPAR2: 177644
001062	177646				UPAR3: 177646
001064	177650				UPAR4: 177650
001066	177652				UPAR5: 177652
001070	177654				UPAR6: 177654

```

;USER PAGE ADDRESS REGISTERS
  
```



```

001072 177656          UPAR7: 177656
001074 172300          KPDR0: 172300          ;KERNEL PAGE DESCRIPTOR REGISTERS
001076 172302          KPDR1: 172302
001100 172304          KPDR2: 172304
001102 172306          KPDR3: 172306
001104 172310          KPDR4: 172310
001106 172312          KPDR5: 172312
001110 172314          KPDR6: 172314
001112 172316          KPDR7: 172316
001114 172340          KPARD: 172340          ;KERNEL PAGE ADDRESS REGISTERS
001116 172342          KPAR1: 172342
001120 172344          KPAR2: 172344
001122 172346          KPAR3: 172346
001124 172350          KPAR4: 172350
001126 172352          KPAR5: 172352
001130 172354          KPAR6: 172354
001132 172356          KPAR7: 172356
001132 001132          ADREND= .-2
001134 177600          PDRTAB: 177600          ;STARTING ADDRESSES OF PDR'S FOR EACH MODE
001136 172300          PDREND: 172300
001140 177640          PARTAB: 177640          ;STARTING ADDRESSES OF PAR'S FOR EACH MODE
001142 172340
001144 001074          STATAB: KPDR0          ;ADDRESS OF KERNEL TABLE OF PDR'S AND PAR'S
001146 000000          0
001150 001034          UPDR0
001152 140000          STAEND: 140000          ;ADDRESS OF USER TABLE OF PDR'S AND PAR'S

001154 000000          STAPNT: 0
001156 000000          PAGES: 0
001160 000000          SAVER: 0
001162 000000          SAVEB: 0
001164 000250          KTVEC: 250
001166 000252          KTSTA: 252
001170 100361          PDRM2: 100361
001172 000000          FTITLE: 0
001174 000000          TESTCT: 0
001176 000000          BLOCKS: 0

;SET UP FOR START OF BASIC LOGIC TESTS
001200 005037 177776          START: CLR          @#PS          ;INITIALIZE STATUS
001204 012706 001000          MOV          #KSTACK,SP          ;SETUP KERNEL STACK
001210 012767 002000 014352          MOV          #2000,ICOUNT          ;INITIALIZE ITERATION COUNT
001216 012767 001256 014350          MOV          #TEST1+2,RETURN          ;SETUP SCOPE AND ITERATION LOOP RETURN
001224 012767 000001 177742          MOV          #1,TESTCT          ;INITIALIZE TEST COUNT
001232 005767 177734          TST          FTITLE          ;DID TITLE PRINT
001236 001007          BNE          TEST1+2          ;YES, START TEST
001240 004767 014450          JSR          PC,TYPE          ;NO, PRINT TITLE
001244 015172          MTIT
001246 005267 177720          INC          FTITLE
001252 000401          BR          .+4          ;SKP SCOPE INSTRUCTION
    
```



# K01

DBKTR.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 10  
 DBKTRD.P11 13-SEP-76 10:29

SEQ 0010

```

;SRO AND SR1 SHOULD BE INITIALIZED TO 0
TEST1: SCOPE
001254 104400                                MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
001256 012706 001000                          JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
001262 004767 015030                          1        ;TEST NUMBER
001266 000001                                HLT                               ;TEST EXECUTED OUT OF SEQUENCE
001270 104006                                RESET                          ;ISSUE INIT
001272 000005                                TST      @SRO              ;CHECK SRO
001274 005777 177524                          BEQ      .+4
001300 001401                                HLT                               ;SRO WAS NOT INITIALIZED TO ZERO
001302 104006                                TST      @SR1              ;CHECK SR1
001304 005777 177520                          BEQ      .+4
001310 001401                                HLT                               ;SR1 WAS NOT INITIALIZED TO ZERO
001312 104006                                MOV      #10,ICOUNT        ;DROP ITERATION COUNT SINCE RESET IS USED
001314 012767 000010 014246

;CHECK READ/WRITE PROPERTIES OF ALL BITS IN SRO EXCEPT 0 AND 8
;BY ROTATING A ONE THRU THE BIT POSITIONS BEING CHECKED
TEST2: SCOPE
001322 104400                                MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
001324 012706 001000                          JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
001330 004767 014762                          2        ;TEST NUMBER
001334 000002                                HLT                               ;TEST EXECUTED OUT OF SEQUENCE
001336 104006                                TST      @SRO              ;CHECK SRO INITIALLY
001340 005777 177460                          BEQ      .+6
001344 001402                                HLT                               ;SRO NOT ZERO AT START OF TEST
001346 104006                                BR       EXIT2
001350 000422                                MOV      #1,R0            ;R0 CONTAINS BIT INDICATING POSITION BEING TESTE
001352 012700 000001  LOOP2: MOV      R0,R1
001356 010001                                MOV      R1,R2
001360 010102                                BIC      #401,R1          ;DON'T SET THE BIT IN SRO IF IT'S BIT 0 OR BIT 8
001362 042701 000401                                BIC      #17777,R2       ;CLEAR THE BIT IN R2 IF IT SHOULDN'T SET IN SRO
001366 042702 017777                                MOV      R1,@SRO
001372 010177 177426                                CMP      R2,@SRO
001376 020277 177422                                BEQ      .+4
001402 001401                                HLT                               ;SRO INCORRECT WHEN VALUE IN R1
001404 104006                                ;WAS LOADED INTO IT

001406 006300                                ASL      R0
001410 103362                                BCC     LOOP2
001412 005077 177406                                CLR      @SRO
001416

EXIT2:

;BITS 0-11 OF ALL PAR'S SHOULD BE READ/WRITE
;TEST BY ROTATING A BIT THRU EACH PAR
;ALSO SHOWS THAT OUTPUT PATHS FROM PAR'S ARE OK
;AND THAT EVERY PAR ADDRESS IS RESPONDED TO
TEST3: SCOPE
001416 104400                                MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
001420 012706 001000                          JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
001424 004767 014666                          3        ;TEST NUMBER
001430 000003                                HLT                               ;TEST EXECUTED OUT OF SEQUENCE
001432 104006                                MOV      #2000,ICOUNT     ;RESTORE ICOUNT
001434 012767 002000 014126                    JSR      %7,CLRAL        ;INITIALIZE KT!!-D REGISTERS
001442 004767 013624                                MOV      #PARTAB,R3      ;R3 POINTS TO TABLE OF PAR ADDRESSES
001446 012703 001140                                MOV      #2,R0           ;R0 IS COUNTER OF STATES LEFT TO TEST
001452 012700 000002  LOOP3: MOV      (R3)+,R1    ;PUT ADDRESS OF 1ST PAR IN SET IN R1
001456 012301

```



L01

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 11  
 DBKTAD.P11 13-SEP-76 10:28

SEQ 0011

```

001460 012702 000010      MOV      #10,R2      ;R2 IS COUNTER OF PAR'S LEFT TO TEST IN SET
001464 012704 000001      LOOP3A: MOV      #1,R4      ;R4 IS BIT OF PAR BEING TESTED
001470 010411      LOOP3B: MOV      R4,@R1      ;SET BIT IN PAR
001472 020411      CMP      R4,@R1      ;CHECK PAR
001474 001401      BEQ      .+4         ;BRANCH IF OK
001476 104006      HLT                      ;PAR WHOSE ADDRESS IS IN R1
                                ;FAILED WHEN THE VALUE IN R4
                                ;WAS LOADED INTO IT

001500 006304      ASL      R4
001502 020427 010000      CMP      R4,#10000
001506 001370      BNE      LOOP3B
001510 005011      CLR      @R1
001512 005721      TST      (R1)+       ;MOVE POINTER
001514 077215      SOB      R2,LOOP3A   ;TEST ALL PAR'S IN SET
001516 077021      SOB      R0,LOOP3    ;TEST ALL 3 REGISTER SETS

;BITS 1-3, 8-14 OF ALL PDR'S SHOULD BE READ/WRITE
;BITS 0,4,5,7 AND 15 SHOULD ALWAYS BE ZERO
;BIT 6 SHOULD BE ZERO IF PDR IS WRITTEN
;ACTUAL CLEARING AND SETTING OF 6 TESTED LATER
;ALSO SHOWS THAT OUTPUT PATHS FROM PDR'S ARE OK
;AND THAT EVERY PDR ADDRESS IS RESPONDED TO
TEST4: SCOPE
001520 104400      MOV      #KSTACK,SP  ;INITIALIZE KERNEL STACK POINTER
001522 012706 001000      JSR      PC,ORDER    ;CHECK TEST SEQUENCE + INIT SRO
001526 004767 014564      4          ;TEST NUMBER
001532 000004      HLT                      ;TEST EXECUTED OUT OF SEQUENCE
001534 104006      JSR      %7,CLRALL    ;INITIALIZE KT11-D REGISTERS
001536 004767 013530      MOV      #PDRTAB,R3
001542 012703 001134      LOOP4: MOV      (R3)+,R1 ;LOAD ADDRESS OF 1ST PDR IN STATE
001546 012301      MOV      #10,R2      ;USE R2 AS A COUNTER OF PDR'S
001550 012702 000010      ;LEFT TO TEST
                                ;SETUP R0 TO ROTATE A BIT THRU
001554 012700 000001      LOOP4A: MOV      #1,R0
001560 010005      LOOP4B: MOV      R0,R5
001562 046705 177402      BIC      PDRM2,R5    ;R5 CONTAINS EXPECTED RESULTING CONTENTS OF PDR
001566 010011      MOV      R0,@R1      ;LOAD PDR
001570 021105      CMP      @R1,R5      ;CHECK RESULTING CONTENTS OF PDR
001572 001401      BEQ      .+4
001574 104006      HLT                      ;PDR WHOSE ADDRESS IS IN R1
                                ;WAS INCORRECT AFTER VALUE IN R0
                                ;WAS LOADED INTO IT
                                ;ROTATE BIT
001576 006300      ASL      R0          ;BRANCH IF NOT DONE WITH THIS PDR
001600 103367      BCC      LOOP4B      ;IF DONE WITH THIS PDR, CLEAR IT
001602 005011      CLR      @R1        ;MOVE POINTER TO ADDRESS NEXT PDR
001604 005721      TST      (R1)+       ;TEST ALL PDR'S IN THIS GROUP
001606 077216      SOB      R2,LOOP4A   ;TEST ALL 2 GROUPS OF PDRS-USER, KERNEL
001610 020327 001136      CMP      R3,#PDREND
001614 003754      BLE      LOOP4

;NO DUAL ADDRESSING TEST FOR PAR'S AND PDR'S
TEST5: SCOPE
001616 104400      MOV      #KSTACK,SP  ;INITIALIZE KERNEL STACK POINTER
001620 012706 001000      JSR      PC,ORDER    ;CHECK TEST SEQUENCE + INIT SRO
001624 004767 014466      5          ;TEST NUMBER
001630 000005      HLT                      ;TEST EXECUTED OUT OF SEQUENCE
001632 104006      JSR      %7,CLRALL    ;CLEAR ALL PAR'S AND PDR'S
001634 004767 013432
    
```



# MO1

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 12  
 DBKTAD.P11 13-SEP-76 10:28

SEQ 0012

001640	012701	001034		MOV	#ADRTAB,R1	
001644	012702	001034		LOP5AA: MOV	#ADRTAB,R2	:R1 POINTS TO ADDRESS OF LOCATION
001650	012703	000040		MOV	#32,R3	:LOADED WITH 1 BIT SET IN EACH 4 BITS
001654	012771	010421	000000	MOV	#10421,@(R1)	:R2 USED AS A POINTER TO CYCLE THRU
001662	020201			LOP5B: CMP	R2,R1	:ALL OTHER ADDRESSES OF PAR/PDR PAIR'S TO
001664	001406			BEQ	CONT5	:CHECK FOR DUAL ADDRESSING
001666	005772	000000		TST	@(R2)	:R3 USED AS A COUNTER
001672	001403			BEQ	CONT5	:LOAD A PAR OR PDR - SET ONE BIT
001674	104006			HLT		:IN EACH CHIP (4 BITS PER CHIP) IF R/W
001676	005072	000000		CONT5: CLR	@(R2)	:SKIP CHECKING THIS ADDRESS TO SEE IF
001702	005722			TST	(R2)+	:IT'S A DUAL, SINCE IT WAS THE ONE LOADED
001704	077312			SOB	R3,LOP5B	:OTHERWISE, CHECK TO SEE IF THIS
001706	022701	001132		CMP	#ADREND,R1	:REGISTER RESPONDED TO THE ADDRESS
001712	001402			BEQ	DONE5A	:OF THE ONE LOADED AS A DUAL
001714	005031			CLR	@(R1)+	:BRANCH IF OK
001716	000752			BR	LOP5AA	:DUAL ADDRESSING - ADDRESS POINTED
001720	012767	000100	013642	DONE5A: MOV	#100,ICOUNT	:TO BY R2 RESPONDED TO THE ADDRESS
001726	104400			TEST6: SCOPE		:POINTED TO BY R1 IN AT LEAST ONE
001730	012706	001000		MOV	#KSTACK,SP	:4 BIT SECTION (1 CHIP)
001734	004767	014356		JSR	PC,ORDER	:REINITIALIZE FAULTY LOCATION
001740	000006					:MOVE POINTER R2
001742	104006			HLT		:CHECK ALL PAR'S AND PDR'S
001744	012767	002000	013616	MOV	#2000,ICOUNT	:TO SEE IF THEY RESPONDED TO THE
001752	004767	013314		JSR	%7,CLRALL	:ADDRESS POINTED TO BY R1
001756	012703	001140		MOV	#PARTAB,R3	:HAVE ALL ADDRESSES BEEN CHECKED
001762	012700	000002		MOV	#2,R0	:FOR DUALS?
001766	012301			LOP6: MOV	(R3)+,R1	:YES - GO TO NEXT TEST
001770	012702	000010		MOV	#10,R2	:NO - MOVE POINTER R1
001774	012711	177777		LOP6A: MOV	#-1,@R1	:CHECK TO SEE IF ANY OTHER ADDRESS
002000	105011			CLRB	@R1	:ALSO RESPONDS TO THE ADDRESS POINTED
002002	022711	007400		CMP	#7400,@R1	:TO BY R1
002006	001401			BEQ	+.4	:DROP ITERATION COUNT
002010	104006			HLT		:SHOW THAT BYTE ADDRESSING OF PAR'S WORKS FOR HIGH AND LOW BYTES
002012	012711	177777		MOV	#-1,@R1	:INITIALIZE KERNEL STACK POINTER
002016	105061	000001		CLRB	1(R1)	:CHECK TEST SEQUENCE + INIT SR0
002022	022711	000377		CMP	#377,@R1	:TEST NUMBER
002026	001401			BEQ	+.4	:TEST EXECUTED OUT OF SEQUENCE
002030	104006			HLT		:RESTORE ITERATION COUNT

:DATOB TO HIGH BYTE OF PAR WHOSE  
 :ADDRESS IS IN R1 FAILED



# NO1

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 13  
 DBKTAD.P11 13-SEP-76 10:28

SEQ 0013

002032	005721		TST	(R1)+	; MOVE POINTER
002034	077221		SOB	R2, LOOP6A	; TEST ALL PAR'S IN SET
002036	077025		SOB	R0, LOOP6	; TEST ALL 2 REGISTER SETS
; SHOW THAT BYTE ADDRESSING OF PDR'S WORKS FOR HIGH AND LOW BYTES					
002040	104400		TEST7:	SCOPE	; INITIALIZE KERNEL STACK POINTER
002042	012706	001000		MOV	; CHECK TEST SEQUENCE + INIT SRO
002046	004767	014244		JSR	; TEST NUMBER
002052	000007			7	; TEST EXECUTED OUT OF SEQUENCE
002054	104006			HLT	; INITIALIZE KT11-D REGISTERS
002056	004767	013210		JSR	; R3 POINTS TO TABLE OF PDR ADDRESSES
002062	012703	001134		MOV	; R0 IS COUNTER OF STATES LEFT TO TEST
002066	012700	000002		MOV	; PUT ADDRESS OF 1ST PDR IN SET INTO R1
002072	012301		LOOP7:	MOV	; R2 IS COUNTER OF PDR'S LEFT TO TEST IN SET
002074	012702	000010		MOV	; SET UP PDR BEING TESTED
002100	012711	177777	LOOP7A:	MOV	; CLEAR LOW BYTE OF PDR
002104	105011			CLRB	; CHECK PDR
002106	022711	077400		CMP	; BRANCH IF OK
002112	001401			BEQ	; DATOB TO PDR WHOSE ADDRESS IS
002114	104006			HLT	; IN R1 FAILED
002116	012711	177777		MOV	; SET UP PDR TO TEST HIGH BYTE
002122	105061	000001		CLRB	; CLEAR HIGH BYTE
002126	022711	000016		CMP	; CHECK PDR
002132	001401			BEQ	
002134	104006			HLT	; DATOB TO HIGH BYTE OF PDR WHOSE
002136	005721			TST	; ADDRESS IS IN R1 FAILED
002140	077221			SOB	; MOVE POINTER
002142	077025			SOB	; TEST ALL PDR'S IN SET
; INIT SHOULD HAVE NO EFFECT ON PAR'S					
002144	104400		TEST10:	SCOPE	; INITIALIZE KERNEL STACK POINTER
002146	012706	001000		MOV	; CHECK TEST SEQUENCE + INIT SRO
002152	004767	014140		JSR	; TEST NUMBER
002156	000010			10	; TEST EXECUTED OUT OF SEQUENCE
002160	104006			HLT	; DROP ITERATION COUNT
002162	012767	000010	013400	MOV	
002170	005067	000104		CLR	
002174	012704	005252		MOV	
002200	012703	001140	TST10:	MOV	
002204	012700	000002		MOV	
002210	012301		LOOP10:	MOV	
002212	012702	000010		MOV	; COUNTER TO LOAD PAR'S
002216	010421		LOP10A:	MOV	; LOAD PAR WITH PATTERN
002220	077202			SOB	; LOAD ALL 16 IN THIS SET
002222	077006			SOB	; INITIALIZE ALL 2 SETS
002224	000005			RESET	; ISSUE INIT
002226	012703	001140		MOV	
002232	012700	000002		MOV	
002236	012301		LOP10B:	MOV	
002240	012702	000010		MOV	; COUNTER TO CHECK PAR'S
002244	020411		LOP10C:	CMP	; CHECK DATA
002246	001401			BEQ	
002250	104006			HLT	; PAR WHOSE ADDRESS IS IN R1
; WAS INCORRECT AFTER INIT					



```

002252 005721 TST (R1)+ ;MOVE POINTER
002254 077205 SOB R2,LOP10C ;TEST ALL 8 PAR'S IN THIS SET
002256 077011 SOB R0,LOP10B ;TEST ALL 2 REGISTER SETS
002260 005767 000014 TST TST10F ;CHECK FOR BOTH PATTERNS USED
002264 001006 BNE EXIT10 ;IF DONE, GO TO NEXT TEST
002266 005267 000006 INC TST10F ;IF NOT, SET FLAG
002272 012704 002525 MOV #2525,R4 ;LOAD OTHER PATTERN
002276 000740 BR TST10 ;REPEAT TEST WITH 2ND PATTERN
002300 000000 TST10F: 0
002302 EXIT10:
    
```

```

;INIT SHOULDN'T CLEAR OR SET ANY OF THE R/W BITS IN THE PDR'S
002302 104400 TEST11: SCOPE
002304 012706 001000 MOV #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
002310 004767 014002 JSR PC,ORDER ;CHECK TEST SEQUENCE + INIT SRO
002314 000011 I1 ;TEST NUMBER
002316 104006 HLT ;TEST EXECUTED OUT OF SEQUENCE
002320 005067 000104 CLR TST11F
002324 012704 025012 MOV #25012,R4 ;LOAD PATTERN IN R4
002330 012703 001134 TST11: MOV #PDRTAB,R3
002334 012700 000002 MOV #2,R0
002340 012301 LOOP11: MOV (R3)+,R1
002342 012702 000010 MOV #10,R2 ;COUNTER TO LOAD PDR'S
002346 010421 LOP11A: MOV R4,(R1)+ ;LOAD PDR WITH PATTERN
002350 077202 SOB R2,LOP11A ;LOAD ALL 8 IN THIS SET
002352 077006 SOB R0,LOOP11 ;INITIALIZE ALL 2 SETS
002354 000005 RESET ;ISSUE INIT
002356 012703 001134 MOV #PDRTAB,R3
002362 012700 000002 MOV #2,R0
002366 012301 LOP11B: MOV (R3)+,R1
002370 012702 000010 MOV #10,R2 ;COUNTER TO CHECK PDR'S
002374 020411 LOP11C: CMP R4,R1 ;CHECK DATA
002376 001401 BEQ .+4
002400 104006 HLT ;PDR WHOSE ADDRESS IS IN R1
    
```

```

002402 005721 TST (R1)+ ;MOVE POINTER
002404 077205 SOB R2,LOP11C ;TEST ALL 8 PDR'S IN THIS SET
002406 077011 SOB R0,LOP11B ;TEST ALL 2 REGISTER SETS
002410 005767 000014 TST TST11F ;CHECK FOR BOTH PATTERNS USED
002414 001006 BNE EXIT11 ;IF DONE, GO TO NEXT TEST
002416 005267 000006 INC TST11F ;IF NOT, SET FLAG
002422 012704 052404 MOV #52404,R4 ;LOAD 2ND PATTERN
002426 000740 BR TST11
002430 000000 TST11F: 0
002432 000240 EXIT11: NOP
    
```

```

;SHOW THAT SRI IS ONLY = 0 AND CANNOT BE LOADED
002434 104400 TEST12: SCOPE
002436 012706 001000 MOV #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
002442 004767 013650 JSR PC,ORDER ;CHECK TEST SEQUENCE + INIT SRO
002446 000012 I2 ;TEST NUMBER
002450 104006 HLT ;TEST EXECUTED OUT OF SEQUENCE
002452 012767 002000 013110 MOV #2000,ICOUNT ;RESTORE ITERATION COUNT
002460 012777 177777 176342 MOV #-1,JSR1 ;TRY TO LOAD SRI
002466 005777 176336 TST JSR1
002472 001401 BEQ .+4
    
```



```

002474 104006          HLT          ;SR1 INCORRECT - SHOULD HAVE TRACKED
;SR2 SHOULD CONTAIN ADDRESS OF LAST FETCH WITH KT11-D TURNED OFF
;CHECK THAT ABORT FREEZES SR2
TEST13: SCOPE
002476 104400          MOV          #KSTACK,SP          ;INITIALIZE KERNEL STACK POINTER
002500 012706 001000     JSR          PC,ORDER          ;CHECK TEST SEQUENCE + INIT SR0
002504 004767 013606     13          ;TEST NUMBER
002510 000013          HLT          ;TEST EXECUTED OUT OF SEQUENCE
002512 104006          MOV          @SR2,R1          ;PICK UP SR2 - SHOULD CONTAIN ADDRESS
002514 017701 176312     AD13:      CMP          #AD13,R1          ;OF THIS INSTRUCTION
002520 022701 002514     BEQ          .+4
002524 001401          HLT          ;SR2 DID NOT CONTAIN FETCH ADDRESS
002526 104006          BIS          #BIT15,@SR0     ;SET NR ABORT
002530 052777 100000 176266 AD13A:    NOP
002536 000240          CMP          #AD13A,@SR2     ;CHECK IF SR2 FROZE
002540 022777 002530 176264     BEQ          .+4
002546 001401          HLT          ;SR2 NOT BEING DISABLED BY NR ABORT
002550 104006          BIC          #BIT15,@SR0     ;CLEAR NR ABORT
002552 042777 100000 176244     AD13B:    BIS          #BIT14,@SR0     ;SET PL ABORT
002560 052777 040000 176236     NOP
002566 000240          CMP          #AD13B,@SR2     ;DID SR2 FREEZE
002570 022777 002560 176234     BEQ          .+4
002576 001401          HLT          ;SR2 NOT BEING DISABLED BY PL ABORT
002600 104006          BIC          #BIT14,@SR0     ;CLEAR PL ABORT
002602 042777 040000 176214     AD13C:    BIS          #BIT13,@SR0     ;SET RO ABORT
002610 052777 020000 176206     NOP
002616 000240          CMP          #AD13C,@SR2     ;DID SR2 FREEZE
002620 022777 002610 176204     BEQ          .+4
002626 001401          HLT          ;SR2 NOT BEING DISABLED BY RO ABORT
002630 104006

```

```

;SHOW THAT DESTINATION ONLY RELOCATION DOESN'T RELOCATE AN INSTRUCTION
;FETCH (ONE CASE), AND THAT RESET CLEARS SR0<8>
;AND TURNS OFF DESTINATION ONLY RELOCATION
;IF THAT MUCH WORKS, YOU'LL GET THRU TO THE NEXT TEST

```

```

TEST14: SCOPE
002632 104400          MOV          #KSTACK,SP          ;INITIALIZE KERNEL STACK POINTER
002634 012706 001000     JSR          PC,ORDER          ;CHECK TEST SEQUENCE + INIT SR0
002640 004767 013452     14          ;TEST NUMBER
002644 000014          HLT          ;TEST EXECUTED OUT OF SEQUENCE
002646 104006          JSR          %7,CLRALL        ;THIS TEST SHOULDN'T GO THRU ANY PAR/PDR PAIR'S
002650 004767 012416     MOV          #1006,@KPDRO     ;SO MAKE THEM ALL GIVE NON-RESIDENT
;AND PAGE LENGTH ERRORS IF ACCESSED
002654 012777 001006 176212     MOV          #10,ICOUNT        ;3 BLOCKS OF KERNEL PDR0 MUST BE MAPPED
;TO ALLOW TRAPS AND ABORTS
002662 012767 000010 012700     MOV          #400,@SR0        ;DROP THE ITERATION COUNT
002670 012777 000400 176126     MOV          #400,@SR0        ;TURN ON DESTINATION ONLY RELOCATION
002676 000005     RESET        ;SHOULD CLEAR DEST ONLY BIT, AND A
;SOLID PLACE TO START
;IF THE FETCH IS RELOCATED
;THIS WILL GIVE A PL ABORT
002700 032777 000400 176116     BIT          #400,@SR0        ;IF KT11-D STILL ON, THIS SHOULD CAUSE
002706 001401          BEQ          .+4              ;PL AND NR ERRORS
002710 000000          HALT                          ;IF KT11-D IS OFF, BIT 8 OF SR0 READS
;AS STILL SET OR ANOTHER BIT IS INCORRECT
;IF KT11-D IS ON, NO NR OR SL ABORT
;OCCURRED AND RESET FAILED TO TURN KT11-D OFF

```



002712 005077 176106

CLR 2SR0

:SHOW THAT DESTINATION ONLY RELOCATION DOESN'T RELOCATE THE SOURCE  
:ADDRESS AND DOES RELOCATE THE DESTINATION002716 104400  
002720 012706 001000  
002724 004767 013366  
002730 000015  
002732 104006  
002734 012767 000010 012626  
002742 004767 012324  
002746 012777 000001 176140  
002754 012777 077406 176112  
002762 012701 003034  
002766 012777 000400 176030  
002774 021111  
002776 001001  
003000 000000  
003002 000005  
003004 012701 002734  
003010 012702 003034  
003014 012777 000400 176002  
003022 021211  
003024 001401  
003026 000000

TEST15: SCOPE

MOV #KSTACK,SP  
JSR PC,ORDER  
15  
HLT  
MOV #10,ICOUNT  
JSR %7,CLRALL  
MOV #1,2KPAR0  
MOV #77406,2KPDRO  
MOV #DATA16,R1  
MOV #400,2SR0  
CMP 2R1,2R1  
BNE .+4  
HALT  
RESET  
MOV #DATA16-100,R1  
MOV #DATA16,R2  
MOV #400,2SR0  
CMP 2R2,2R1  
BEQ .+4  
HALT:INITIALIZE KERNEL STACK POINTER  
:CHECK TEST SEQUENCE + INIT SR0  
:TEST NUMBER  
:TEST EXECUTED OUT OF SEQUENCE  
:KEEP THE NUMBER OF LOOPS DOWN  
:OFFSET KERNEL PAR/PDR PAIR 0 ONE BLOCK FROM BAN  
:LOAD A BANK 0 ADDRESS  
:TURN ON DESTINATION ONLY RELOCATION  
:THIS TEST WILL FAIL IF BOTH ARE  
:RELOCATED OR BOTH ARE NOT RELOCATED  
:SOURCE AND DESTINATION BOTH ADDRESSED SAME LOCA  
:TURN OFF DESTINATION-ONLY RELOCATION  
:LOAD DESTINATION ADDRESS MINUS RELOCATION FACTO  
:LOAD SOURCE ADDRESS  
:TURN ON DESTINATION-ONLY RELOCATION  
:USE SAME INSTRUCTION AND ADDRESS  
:MODES AS BEFORE  
:DESTINATION NOT RELOCATED OR INCORRECTLY  
:RELOCATED OR SOURCE RELOCATED  
:TURN OFF RELOCATION003030 000005  
003032 000401  
003034 132465RESET  
BR .+4

DATA16: 132465

:SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU KERNEL PAGE 7 WILL  
:CLEAR THE DESTINATION ONLY RELOCATION BIT AND TURN OFF DESTINATION ONLY RELOCATION003036 104400  
003040 012706 001000  
003044 004767 013246  
003050 000016  
003052 104006  
003054 004767 012212  
003060 012777 000001 176026  
003066 012777 077406 176000  
003074 012701 003034  
003100 004767 013276  
003104 016702 175714  
003110 012777 000400 175706  
003116 005012  
003120 021111  
003122 001401  
003124 000000  
003126 032777 000400 175670  
003134 001402  
003136 104006  
003140 000005

TEST16: SCOPE

MOV #KSTACK,SP  
JSR PC,ORDER  
16  
HLT  
JSR %7,CLRALL  
MOV #1,2KPAR0  
MOV #77406,2KPDRO  
MOV #DATA16,R1  
JSR PC,KERN7  
MOV SR0,R2  
MOV #400,2SR0  
CLR 2R2  
CMP 2R1,2R1  
BEQ .+4  
HALT  
BIT #400,2SR0  
BEQ .+6  
HLT  
RESET:INITIALIZE KERNEL STACK POINTER  
:CHECK TEST SEQUENCE + INIT SR0  
:TEST NUMBER  
:TEST EXECUTED OUT OF SEQUENCE  
:INITIALIZE  
:MAP KERNEL PAR/PDR PAIR 0  
:TO BANK 0 OFFSET BY 1 PAGE  
:USED TO PROVE KT11-D IS  
:TURNED OFF AFTER CLEARING BIT 8, SR0  
:SETUP R1 TO REFERENCE KERNEL PAR/PDR PAIR 0  
:MAP KERNEL PAR/PDR 7 TO EXT BANK  
:SETUP R2 TO ADDRESS SR0  
:TURN ON DESTINATION ONLY RELOCATION  
:CLEAR SR0 THRU KERNEL PAR/PDR PAIR7  
:SHOW THAT KT11-D IS OFF  
:KT11-D STILL ON  
:SHOW THAT BIT 8, SR0 IS NOW ZERO  
:DESTINATION ONLY RELOCATION BIT IS STILL ON  
:MAKE SURE THAT KT11-D IS OFF

003142 004767 012124

JSR %7,CLRALL

:SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU USER PAGE 7  
:WILL TURN OFF DESTINATION - ONLY PAGING

:INITIALLY CLEAR ALL PAR/PDR PAIRS



003146	012777	000001	175700	MOV	#1, @UPAR0	:MAP USER 0 TO
003154	012777	077406	175652	MOV	#77406, @UPDR0	:BANK 0 OFFSET BY 1 PAGE, RW
003162	012701	003034		MOV	#DATA16, R1	:SETUP R1 TO REFERENCE USER 0
003166	012777	007600	175676	MOV	#7600, @UPAR7	:MAP USER 7 TO THE
003174	012777	077406	175650	MOV	#77406, @UPDR7	:EXTERNAL BANK
003202	016702	175616		MOV	SRO, R2	:SETUP R2 TO ADDRESS SRO
003206	012737	140000	177776	MOV	#140000, @#PS	:SET MODE TO USER
003214	012777	000400	175602	MOV	#400, @SRO	:TURN ON DESTINATION - ONLY PAGING
003222	005012			CLR	@R2	:CLEAR SRO THRU USER ASR7
003224	021111			CMP	@R1, @R1	:SHOW THAT KT11-D IS OFF
003226	001401			BEG	.+4	
003230	000777			BR	.	:RELOCATION STILL ON

:SHOW THAT ALL PAGE BOUNDARY REFERENCES REFERENCE THE CORRECT PAR  
:AND RELOCATE CORRECTLY  
:USE DESTINATION - ONLY PAGING  
:MAP ALL PAR/PDR PAIR'S RESIDENT READ WRITE

:  
:R0 - POINTS TO THE ADDRESS OF THE CURRENT PAR IN THE ADDRESS TABLE  
:R1 - CONTAINS VIRTUAL ADDRESS BEING USED TO REFERENCE START OF PAGE  
:R2 - CONTAINS VIRTUAL ADDRESS BEING USED TO REFERENCE END OF PAGE  
:R3  
:R4  
:R5 - USED TO REFERENCE SRO TO TURN OFF DESTINATION ONLY PAGING

003232	104400			TEST17:	SCOPE		
003234	012706	001000		MOV	#KSTACK, SP	:INITIALIZE KERNEL STACK POINTER	
003240	004767	013052		JSR	PC, ORDER	:CHECK TEST SEQUENCE + INIT SRO	
003244	000017			17		:TEST NUMBER	
003246	104006			HLT		:TEST EXECUTED OUT OF SEQUENCE	
003250	004767	012016		JSR	%7, CLRALL	:INITIALIZE	
003254	004767	012036		JSR	%7, RWALL	:MAKE ALL PAR/PDR PAIR'S RW, BANK 0, 4K	
003260	013767	017700	175672	MOV	@#17700, SAVEA	:SAVE CONTENTS OF LOCATIONS TO BE USED	
003266	013767	017776	175666	MOV	@#17776, SAVEB		
003274	012737	123456	017700	MOV	#123456, @#17700	:SET UP LOCATIONS TO BE REFERENCED	
003302	012737	134567	017776	MOV	#134567, @#17776		
003310	012703	001012		MOV	#K123, R3		
003314	012704	001014		MOV	#K134, R4		
003320	012767	000100	012242	MOV	#100, ICOUNT	:CHANGE ITERATION COUNT	
003326	012737	140000	177776	MOV	#140000, @#PS	:CHANGE TO USER	
003334	012706	000400		MOV	#USTACK, SP	:SET UP USER STACK POINTER	
003340	005037	177776		CLR	@#PS	:RETURN TO KERNEL	
003344	012767	001144	175602	MOV	#STATAB, STAPNT	:SET UP TO REFERENCE STATE TABLE	
003352	017700	175576		STAT20:	MOV @STAPNT, R0	:PICK UP ADDRESS OF START OF	
003356	062700	000020		ADD	#20, R0	:ADDRESS TABLE FOR NEW STATE	
003362	062767	000002	175564	ADD	#2, STAPNT		
003370	017737	175560	177776	MOV	@STAPNT, @#PS	:SET UP NEW STATE	
003376	062767	000002	175550	ADD	#2, STAPNT		
003404	012767	000010	175544	MOV	#8, PAGES	:SET UP COUNTER OF ASR'S LEFT TO TEST	
003412	012770	007600	000016	MOV	#7600, @16(R0)	:SET UP SEGMENTED REFERENCE TO SRO	
003420	016705	175400		MOV	SRO, R5	:USED TO TURN DESTINATION - ONLY PAGING OFF	
003424	005001			CLR	R1		
003426	012702	000076		MOV	#76, R2		
003432	012767	000200	175536	PAG20:	MOV #128, BLOCKS	:SET UP BLOCK COUNT	
003440	012770	000177	000000	MOV	#177, @ (R0)	:SET UP PAR	
003446	022767	000001	175502	CMP	#1, PAGES	:IS THIS PAGE 7? (WAS USED FOR REFERENCE TO SRO)	



003454	001005			BNE	BLK20				: IF NOT, BRANCH
003456	012770	007600	177776	MOV	#7600, @-2(R0)				: YES, SET UP PAGE 6 FOR REFERENCES TO SR0
003464	042705	020000		BIC	#20000, R5				: CHANGE R5 TO POINT TO SR0 THRU PAR/PDR PAIRS
003470	012777	000400	175326	BLK20: MOV	#400, @SR0				: TURN ON DESTINATION ONLY PAGING
003476	021311			CMP	@R3, @R1				: CK BOTTOM PAGE BOUNDARY
003500	001401			BEG	.+4				
003502	000000			HALT					: RELOCATION FAILED
003504	021412			CMP	@R4, @R2				: CK UPPER PAGE BOUNDARY
003506	001401			BEG	.+4				
003510	000000			HALT					: RELOCATION FAILED
003512	005015			CLR	@R5				: TURN OFF KT11-D
003514	005370	000000		DEC	@(R0)				: MAP PAR 1 PAGE LOWER
003520	062701	000100		ADD	#100, R1				: SET UP R1 AND R2 TO REFERENCE
003524	062702	000100		ADD	#100, R2				: NEXT VIRTUAL PAGE
003530	005367	175442		DEC	BLOCKS				: DECREMENT COUNT OF PAGES LEFT
003534	001355			BNE	BLK20				: BRANCH IF NOT DONE WITH THIS PAR/PDR PAIR
003536	005070	000000		CLR	@(R0)				
003542	005367	175410		DEC	PAGES				: DECREMENT COUNT OF PAR/PDR PAIR'S LEFT
003546	001402			BEG	END20				: BRANCH IF ALL PAR/PDR PAIR'S IN THIS STATE DONE
003550	005720			TST	(R0)+				: MOVE ADDRESS TABLE POINTER
003552	000727			BR	PAG20				
003554	026727	175374	001152	END20: CMP	STAPNT, #STAEND				: CHECK FOR ALL STATES TESTED
003562	003673			BLE	STAT20				: IF NOT, BRANCH
003564	005037	177776		CLR	@#PS				: IF DONE, REINITIALIZE
003570	005077	175230		CLR	@SR0				
003574	016727	175360	017700	MOV	SAVEA, #17700				
003602	016727	175354	017776	MOV	SAVEB, #17776				
003610	016777	175352	175346	MOV	KTSTA, @KTVEC				
003616	005077	175344		CLR	@KTSTA				

: SHOW THAT THE INSTRUCTIONS USED IN THE NEXT TEST RELOCATE CORRECTLY IN  
 : DESTINATION ONLY RELOCATION

003622	104400			TEST20: SCOPE					
003624	012706	001000		MOV	#KSTACK, SP				: INITIALIZE KERNEL STACK POINTER
003630	004767	012462		JSR	PC, ORDER				: CHECK TEST SEQUENCE + INIT SR0
003634	000020			20					: TEST NUMBER
003636	104006			HLT					: TEST EXECUTED OUT OF SEQUENCE
003640	012767	002000	011722	MOV	#2000, ICOUNT				: RESTORE ITERATION COUNT
003646	004767	011420		JSR	%7, CLRALL				: CLEAR ALL KT11-D REGISTERS
003652	012777	000001	175234	MOV	#1, @KPAR0				: OFFSET KERNEL I-SPACE PAGE 0
003660	012777	077406	175206	MOV	#77406, @KPDRO				: BY 1 BLOCK FROM BANK 0
003666	004767	012510		JSR	PC, KERN7				: MAP KERNEL PAR/PDR 7 TO EXT BANK
003672	016701	175126		MOV	SR0, R1				: SETUP R1 TO REFERENCE SR0
003676	016746	000054		MOV	DST21A-100, -(SP)				
003702	016746	000052		MOV	DST21B-100, -(SP)				
003706	016746	000050		MOV	DST21C-100, -(SP)				
003712	005067	000140		CLR	DST21A				: INITIALIZE LOCATIONS TO BE
003716	012767	177777	000134	MOV	#-1, DST21B				: WRITTEN INTO
003724	012767	177777	000130	MOV	#-1, DST21C				
003732	012777	000400	175064	MOV	#400, @SR0				: TURN ON DESTINATION - ONLY RELOCATION
003740	022737	176543	003642	CMP	#176543, @AD21A-100				: COMPARE THE CONTENTS OF AD21A
003746	003742			AD21A=.	-4				: WITH ITSELF, RELOCATED THRU KERNEL 0
003750	001401			BEG	.+4				
003750	104006			HLT					: DESTINATION - ONLY RELOCATION FAILED
									: TO RELOCATE ONLY THE LAST CALCULATION
									: OF THE CMP INSTRUCTION



```

003752 122737 165432 003654      CMPB      #165432,3#AD21B-100      ;COMPARE THE CONTENTS OF AD21B
                                AD21B=-4      ;WITH ITSELF, RELOCATED THRU KERNEL 0
003760 003754                      BEQ      .+4
003762 001401                      HLT
                                ;DESTINATION - ONLY RELOCATION
                                ;FAILED TO RELOCATE ONLY THE FINAL
                                ;CALCULATION OF THE CMPB INSTRUCTION
                                ;EXECUTE REMAINING INSTRUCTIONS

003764 012737 077711 003756      MOV      #77711,3#DST21A-100
003772 005077 000066      CLR      3#AD21C
003776 105037 003762      CLRB    3#DST21C-100
004002 005011      CLR      3#R1
                                ;TURN OFF KT11-D
004004 022767 077711 000044      CMP      #77711,DST21A
                                ;CHECK LOCATION ADDRESSED BY MOV
004012 001401                      BEQ      .+4
004014 104006                      HLT
                                ;MOV INSTRUCTION FAILED TO RELOCATE
                                ;ONLY THE FINAL ADDRESS CALCULATION
                                ;CHECK LOCATION ADDRESSED BY CLR

004016 005767 000036      TST     DST21B
004022 001401                      BEQ      .+4
004024 104006                      HLT
                                ;CLR INSTRUCTION FAILED TO RELOCATE
                                ;CORRECTLY IN DESTINATION - ONLY RELOCATION
                                ;CHECK LOCATION ADDRESSED BY CLRB

004026 022767 177400 000026      CMP      #177400,DST21C
004034 001401                      BEQ      .+4
004036 104006                      HLT
                                ;CLRB INSTRUCTION FAILED TO RELOCATE
                                ;CORRECTLY IN DESTINATION - ONLY RELOCATION
                                ;RESTORE LOCATIONS IN CASE OF ERROR

004040 012667 177716      MOV      (SP)+,DST21C-100
004044 012667 177710      MOV      (SP)+,DST21B-100
004050 012667 177702      MOV      (SP)+,DST21A-100
004054 000404      BR      EXIT2!
004056 000000
004060 000000      DST21A: 0
004062 000000      DST21B: 0
004064 003760      DST21C: 0
004066 000240      AD21C: DST21B-100
                                EXIT21: NOP

```

```

;TEST OF RELOCATION ADDERS - CHECK CORRECT PROPAGATION OF CARRY, AND CORRECT
;OUTPUT FOR EACH POSSIBLE COMBINATION FOR EACH BIT POSITION
;USE DESTINATION - ONLY RELOCATION, KERNEL
;TEST BY USING THE NECESSARY VALUE IN KERNEL PAR 1, WITH THE SECOND
;NECESSARY VALUE BEING THE VIRTUAL ADDRESS REFERENCE TO KERNEL PAR 1
;CHECK THE RESULTING PHYSICAL ADDRESS BY READING THE CONTENTS OF THE LOCATION,
;AND WRITING INTO THE LOCATION
;NOTE THAT THIS INCLUDES CHECKS OF ADDRESS WRAP AROUND

```

```

004070 104400      TEST21: SCOPE
004072 012706 001000      MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
004076 004767 012214      JSR     PC,ORDER      ;CHECK TEST SEQUENCE + INIT SRO
004102 000021      HLT      ;TEST NUMBER
004104 104006      HLT      ;TEST EXECUTED OUT OF SEQUENCE
004106 004767 011160      JSR     %7,CLRALL      ;CLEAR ALL KT11-D REGISTERS
004112 012777 077406 174754      MOV      #77406,3#KPDRO      ;MAP KERNEL 0 TO BANK 0, 4K, RW
004120 012777 077406 174750      MOV      #77406,3#KPDRI      ;MAKE KERNEL 1 4K, RW
004126 004767 012250      JSR     PC,KERN7      ;MAP KERNEL PAR/PDR 7 TO EXT BANK

```

```

;CHECK VIRTUAL ADDRESS OF 0 ADDED TO PAR OF -1 (FOR BIT POSITIONS
;RELEVANT TO THE ADDERS ONLY)

```

```

004132 012777 007777 174756      MOV      #7777,3#KPAR1      ;SET PAR TO -1
004140 012737 030000 177776      MOV      #30000,3#PS      ;SET UP LOCATION TO BE REFERENCED
004146 012777 000400 174650      MOV      #400,3#SRO      ;TURN ON DESTINATION - ONLY PAGING
004154 122737 000060 020077      CMPB    #60,3#20077      ;CHECK HIGH BYTE OF RESULTING ADDRESS

```



```

004162 001011 BNE ERR22A ;(PS) - REFERENCED THRU PAR/PDR PAIR 1
004164 105037 020077 CLR B #20077 ;BRANCH ON FAILURE
004170 005077 174630 CLR @SR0 ;CLEAR PA 77777 THRU KERNEL 1
004174 105737 177777 TSTB @PS+1 ;TURN OFF KT11-D
004200 001401 BEQ .+4 ;CHECK TO SEE IF CORRECT LOCATION
004202 104006 HLT ;WAS REFERENCED
004204 000405 BR CNT22B ;RELOCATION FAILED
004206 005077 174612 ERR22A: CLR @SR0 ;GO TO NEXT CHECK
004212 104006 HLT ;TURN OFF KT11-D
;RELOCATION FAILED IN THE COMPARE
;AT LOCATION ADR22A
004214 005037 177776 CLR @PS ;REINITIALIZE PROCESSOR STATUS

;CHECK VIRTUAL ADDRESS OF -1 ADDED TO PAR OF 0 (VALUES FOR BIT
;POSITIONS RELEVANT TO THE ADDERS ONLY). RESULT SHOULD BE PA 17712
004220 005077 174672 CNT22B: CLR @KPAR1 ;SET PAR TO 0
004224 012737 125252 017712 MOV #125252,@DESTAD ;LOAD PHYSICAL LOCATION TO BE REFERENCED
;ADDRESS 17712
004232 012777 000400 174564 MOV #400,@SR0 ;TURN ON DESTINATION - ONLY PAGING
004240 022737 125252 037712 CMP #125252,@37712 ;RELOCATE THRU KERNEL PAR/PDR PAIR1
004246 001011 BNE ERR22B ;BRANCH ON FAILURE
004250 005037 037712 CLR @37712 ;CLEAR THRU KERNEL PAR/PDR PAIR1
004254 005077 174544 CLR @SR0 ;TURN OFF KT11-D
004260 005737 017712 TST @17712 ;CHECK TO SEE IF CORRECT LOCATION
004264 001401 BEQ .+4 ;WAS CLEARED
004266 104006 HLT ;RELOCATION FAILED
004270 000403 BR CNT22C ;GO TO NEXT CHECK
004272 005077 174526 ERR22B: CLR @SR0 ;TURN OFF KT11-D
004276 104006 HLT ;RELOCATION FAILED IN THE COMPARE
;AT LOCATION ADR22B

;CHECK VIRTUAL ADDRESS OF 1 (BIT 6) ADDED TO PAR OF -1
;RESULTING PHYSICAL ADDRESS SHOULD BE ZERO
;NOTE THAT THIS IS A CHECK OF ADDRESS WRAP AROUND
004300 012777 007777 174610 CNT22C: MOV #7777,@KPAR1 ;SET UP PAR TO -1
004306 012737 034343 000000 MOV #34343,@#0 ;SET UP A VALUE IN LOCATION TO
;BE REFERENCED (0)
004314 012777 000400 174502 MOV #400,@SR0 ;TURN ON DESTINATION-ONLY PAGING
004322 022737 034343 020100 CMP #34343,@#20100 ;EFFECTIVELY ADDS 1 TO PAR ADDRESS
;TO GET PHYSICAL ADDRESS OF 0
004330 001013 BNE ERR22C ;BRANCH ON FAILURE
004332 012737 000002 020100 MOV #2,@#20100 ;WRITE SAME LOCATION
004340 005077 174460 CLR @SR0 ;TURN OFF KT11-D
004344 022737 000002 000000 CMP #2,@#0 ;CHECK LOCATION WHICH SHOULD HAVE
;BEEN REFERENCED
004352 001401 BEQ .+4 ;RELOCATION FAILED WHEN WRITING PA 0
004354 104006 HLT ;GO TO NEXT CHECK
004356 000406 BR CNT22D ;TURN OFF KT11-D
004360 005077 174440 ERR22C: CLR @SR0 ;RELOCATION FAILED IN THE COMPARE
004364 104006 HLT ;AT LOCATION ADR22C
004366 012737 000002 000000 MOV #2,@#0

;CHECK VIRTUAL ADDRESS OF -1 (BITS 6-12) ADDED TO PAR OF 1
;(PLUS HIGH BITS SET, BUT THEY DON'T ALTER CARRY CONDITION TESTED FOR)
;RESULTING PHYSICAL ADDRESS SHOULD BE ZERO
004374 012777 007601 174514 CNT22D: MOV #7601,@KPAR1 ;SET UP PAR TO 1, WITH HIGH BITS SET
004402 012737 043434 000000 MOV #43434,@#0 ;SET UP A VALUE IN LOCATION TO

```







```

004650 000000          ADD23: HALT          ;WITH RELOCATION ON, SHOULD FETCH
004652 000000          HALT          ;FROM 1 BLOCK ABOVE THIS
004654 000000          HALT          ;(ADD23A)
004656 000000          HALT
004660 000000          HALT
004662 000000          HALT
004664 032777 000001 174132  BIT      #1,SR0      ;WHEN KT11-D IS TURNED OFF, NEXT
                                ;FETCH SHOULD BE FROM HERE -
004672 001401          BEQ      .+4        ;CHECK BIT 0, SR0
004674 104006          HLT
                                ;KT11-D OFF BUT SR0<0> STILL SET
                                ;AFTER AN INIT
004676 005077 174122    CLR      SR0
004702 000432          BR      EXIT23
                                ;
004750 004750          .=ADD23+100
004750 022737 052525 017612  CMP      #52525,SR0-100      ;WHEN KT11-D IS TURNED ON, NEXT
                                ;INSTRUCTION EXECUTED SHOULD
                                ;BE HERE - CK RELOCATION OF SOURCE
                                ;AND DESTINATION CALCULATIONS
                                ;RELOCATION FAILED IN A SOURCE OR
                                ;DESTINATION ADDRESS CALCULATION
                                ;WITH FULL RELOCATION ON (SR0<0>SET)
                                ;ISSUE INIT TO TURN OFF KT11-D
                                ;INIT DIDN'T TURN OFF KT11-D
004756 001401          BEQ      .+4
004760 000000          HALT
004762 000005          RESET
004764 000000          HALT
004766 000777          BR
004770 000240          EXIT23: NOP
                                ;
                                ;SHOW THAT A DATO OF 0 TO SR0<0> WILL CLEAR SR0<0> AND
                                ;TURN OFF RELOCATION
004772 104400          TEST23: SCOPE
004774 012706 001000    MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
005000 004767 011312    JSR      PC,ORDER      ;CHECK TEST SEQUENCE + INIT SR0
005004 000023          23          ;TEST NUMBER
005006 104006          HLT          ;TEST EXECUTED OUT OF SEQUENCE
005010 012767 002000 010552  MOV      #2000,ICOUNT      ;RESTORE ITERATION COUNT
005016 004767 010250    JSR      %7,CLRALL      ;INITIALLY CLEAR ALL KT11-D REGISTERS
005022 012777 000001 174064  MOV      #1,AKPAR0      ;MAP KERNEL PAGE 0 TO
005030 012777 077406 174036  MOV      #77406,AKPDR0    ;BANK 0 OFFSET BY 1 BLOCK
005036 004767 011340    JSR      PC,KERN7      ;MAP KERNEL PAR/PDR 7 TO EXT BANK
005042 012777 000001 173754  MOV      #1,SR0      ;TURN ON KT11-D
005050 000000          ADD24: HALT      ;WHEN KT11-D IS TURNED ON, SHOULD
005052 000000          HALT      ;FETCH FROM THIS ADDRESS PLUS
005054 000000          HALT      ;ONE BLOCK (ADD24A)
005056 000240          NOP
005060 000240          NOP
005062 032777 000001 173734  BIT      #1,SR0      ;AFTER KT11-D IS TURNED OFF, CHECK
005070 001401          BEQ      .+4        ;SR0<0>
005072 104006          HLT          ;KT11-D OFF BUT SR0<0> STILL
                                ;SET AFTER A BIC #1,SR0
005074 000433          BR      EXIT24
                                ;
005150 005150          .=ADD24+100
005150 042777 000001 173646  BIC      #1,SR0      ;WHEN KT11-D IS TURNED ON, SHOULD
                                ;RELOCATE FETCH TO HERE - TURN
                                ;OFF KT11-D VIA BIC OF SR0<0>
                                ;KT11-D STILL RELOCATING AFTER
                                ;BIC OF SR0<0>
005156 000000          HALT
005160 000005          RESET
    
```



005162 000777  
 005164 000240

BR  
 EXIT24: NOP

:SHOW THAT A REFERENCE TO A NON-RESIDENT PAGE  
 :WILL ABORT TO THE KT11-D ABORT VECTOR ADDRESS (250)  
 :WITH BIT 15 OF SR0 SET. SR0 AND SR2 ARE CHECKED FOR  
 :THE CORRECT VALUES, AS ARE KPDR0 AND KPDR1  
 :SHOW THAT BIT 15 OF SR0 CAN BE CLEARED  
 :SHOW THAT SR2 IS READ ONLY

005166 104400  
 005170 012706 001000  
 005174 004767 011116  
 005200 000024  
 005202 104006  
 005204 004767 010062  
 005210 012777 077406 173656  
 005216 004767 011160  
 005222 012777 005256 173734  
 005230 005077 173732  
 005234 012704 020000  
 005240 005277 173560  
 005244 005724  
 005246 000000  
 005250 005077 173550  
 005254 000442  
 005256 017701 173542  
 005262 005377 173536  
 005266 022701 100003  
 005272 001401  
 005274 104006

TEST24: SCOPE

MOV #KSTACK, SP  
 JSR PC, ORDER  
 24  
 HLT  
 JSR %7, CLRALL  
 MOV #77406, @KPDR0  
 JSR PC, KERN7  
 MOV #INT25, @KTVEC  
 CLR @KTSTA  
 MOV #20000, R4  
 INC @SR0  
 ADR25: TST (R4)+  
 ADR25A: HALT  
 CLR @SR0  
 BR DON25  
 INT25: MOV @SR0, R1  
 DEC @SR0  
 CMP #100003, R1  
 BEQ .+4  
 HLT

:INITIALIZE KERNEL STACK POINTER  
 :CHECK TEST SEQUENCE + INIT SR0  
 :TEST NUMBER  
 :TEST EXECUTED OUT OF SEQUENCE  
 :CLEAR ALL KT11-D REGISTERS  
 :MAP KERNEL 0 TO BANK 0, RW, 4K  
 :MAP KERNEL PAR/PDR 7 TO EXT BANK  
 :SETUP RETURN VECTOR

:USE R4 TO REFERENCE NR KERNEL 1  
 :TURN ON KT11-D  
 :REFERENCE NR KERNEL 1  
 :SHOULD HAVE ABORTED ALREADY  
 :TURN OFF KT11-D

:SAVE CONTENTS OF SR0  
 :TURN OFF KT11-D  
 :CHECK SAVED CONTENTS OF SR0

:SR0 INCORRECT AFTER NR ABORT  
 :(SEE SAVED CONTENTS IN R1)  
 :CK SR2

:SR2 INCORRECT-SHOULD CONTAIN ADDRESS  
 :OF LAST FETCH BEFORE THE ABORT  
 :TRY TO WRITE INTO SR2  
 :SR2 SHOULD BE READ ONLY

:SR2 NOT READ ONLY

:KERNEL PDR 0 INCORRECT  
 :W BIT SHOULD HAVE BEEN SET BY THE STACK WRITE

005276 022777 005244 173526  
 005304 001401  
 005306 104006  
 005310 005077 173516  
 005314 022777 005244 173510  
 005322 001401  
 005324 104006  
 005326 022777 077506 173540  
 005334 001401  
 005336 104006

CMP #ADR25, @SR2  
 BEQ .+4  
 HLT  
 CLR @SR2  
 CMP #ADR25, @SR2  
 BEQ .+4  
 HLT  
 CMP #77506, @KPDR0  
 BEQ .+4  
 HLT

005340 005777 173532  
 005344 001401  
 005346 104006  
 005350 021627 005246  
 005354 001401  
 005356 104006  
 005360 022626  
 005362 005077 173600  
 005366 016777 173574 173570

TST @KPDR1  
 BEQ .+4  
 HLT  
 CMP (R6), #ADR25A  
 BEQ .+4  
 HLT  
 CMP (R6)+, (R6)+  
 DON25: CLR @KTSTA  
 MOV KTSTA, @KTVEC

:KERNEL PDR 1 INCORRECT  
 :CHECK VALUE PUSHED ON STACK

:INCORRECT VALUE ON STACK  
 :RESTORE STACK

:CHANGE TRAP VECTOR TO CAUSE A  
 :HALT ON A FALSE TRAP

:SHOW THAT WRITING A PAGE WILL SET THE W BIT IN THE CORRESPONDING  
 :PDR, AND THAT NO OTHER W BITS SET AT THE SAME TIME  
 :SHOW THAT WRITING THE PDR (VIA A DAT0) WILL CLEAR THE W BIT



: SINCE THIS IS DONE FOR ALL PDR'S, THIS IS ALSO  
 : A TEST OF INDIRECT ADDRESSING (VIA A VIRTUAL ADDRESS) OF THE PDR'S  
 TEST25: SCOPE

005374	104400			MOV	#KSTACK, SP		: INITIALIZE KERNEL STACK POINTER
005376	012706	001000		JSR	PC, ORDER		: CHECK TEST SEQUENCE + INIT SRD
005402	004767	010710		25			: TEST NUMBER
005406	000025			HLT			: TEST EXECUTED OUT OF SEQUENCE
005410	104006			MOV	#400, ICOUNT		: LOAD ITERATION COUNT
005412	012767	000400	010150	JSR	%7, RWALL		: MAP ALL PAR/PDR PAIR'S 4K, BANK 0, RW
005420	004767	007672		JSR	PC, KERN7		: MAP KERNEL PAR/PDR 7 TO EXT BANK
005424	004767	010752		MOV	#7600, @UPAR7		: MAP USER 7 TO EXTERNAL BANK
005430	012777	007600	173434	MOV	#140000, @#PS		: SET MODE TO USER
005436	012737	140000	177776	MOV	#USTACK, R6		: SET UP USER STACK
005444	012706	000400		CLR	@#PS		: REINITIALIZE STATUS TO KERNEL MODE
005450	005037	177776		MOV	#ADRTAB, R4		
005454	012704	001034		MOV	#10, R5		: LOAD R4 WITH ADDRESS OF ADR TABLE
005460	012705	000010		LOP31A: MOV	#77406, @ (R4) +		: INIT COUNTER OF PDR'S LEFT TO CHECK
005464	022734	077406		LOP31B: CMP	. +4		: CHECK ALL PDR W BITS BITS CLEAR
005470	001401			BEQ			: PDR INCORRECT - W BIT SET OR ANOTHER
005472	104006			HLT			: BIT INCORRECT IN PDR WHOSE ADDRESS
							: IS IN THE LOCATION POINTED TO BY R4
							: MOVE POINTER TO FIRST ADR OF NEXT SET
005474	077505			SOB	R5, LOP31B		
005476	062704	000020		ADD	#20, R4		
005502	020427	001132		CMP	R4, #ADREND		: BRANCH IF DONE
005506	003001			BGT	CNT31A		
005510	000763			BR	LOP31A		
005512	012700	001144		CNT31A: MOV	#STATAB, R0		: SET UP START OF STATE TABLE
005516	012001			LOP31C: MOV	(R0) +, R1		: R1 CONTAINS ADDRESS OF PDR OF ADDRESS
005520	012702	017776		MOV	#17776, R2		: SET UP VIRTUAL ADDRESS TO BE REFERENCED
005524	012037	177776		MOV	(R0) +, @#PS		: SET UP STATUS FOR CURRENT MODE
005530	005277	173270		LOP31D: INC	@SR0		: TURN ON KT11-D
005534	011212			MOV	(R2), (R2)		: REFERENCE PAGE TO SET W BIT
005536	005077	173262		CLR	@SR0		: TURN OFF KT11-D
005542	032771	000100	000000	BIT	#100, @ (R1)		: CHECK W BIT
005550	001001			BNE	. +4		
005552	104006			HLT			: W BIT NOT SET IN PDR AFTER PAGE WRITTEN
005554	012703	001034		MOV	#ADRTAB, R3		: SET UP ADDRESS OF ADDRESS TABLE
005560	012704	000010		LOP31E: MOV	#10, R4		: NOW CHECK ALL PDR TO SHOW NO OTHER
005564	020103			LOP31F: CMP	R1, R3		: W BITS WERE SET
005566	001405			BEQ	CNT31B		
005570	032773	000100	000000	BIT	#100, @ (R3)		
005576	001401			BEQ	. +4		
005600	104006			HLT			: W BIT SET IN THE PDR WHOSE ADDRESS IS POINTED T
							: AS WELL AS THE W BIT IN THE PDR
							: FOR THE PAGE THAT WAS WRITTEN
							: UPDATE ADDRESS POINTER
005602	005723			CNT31B: TST	(R3) +		: TEST NEW PDW
005604	077411			SOB	R4, LOP31F		: UPDATE POINTER TO NEXT SET
005606	062703	000020		ADD	#20, R3		
005612	020327	001132		CMP	R3, #ADREND		
005616	002760			BLT	LOP31E		
005620	012771	077406	000000	MOV	#77406, @ (R1)		: CLEAR W BIT VIA DATO TO PDR
005626	032771	000100	000000	BIT	#100, @ (R1)		: CHECK W BIT
005634	001401			BEQ	. +4		
005636	104006			HLT			: W BIT DIDN'T CLEAR WHEN PDR
							: WAS WRITTEN (ADDRESS OF ADDRESS
							: OF PDR IS IN R1)
							: UPDATE POINTER
005640	005721			TST	(R1) +		



005642	062702	020000		ADD	#20000,R2		:CHANGE VA TO REFERENCE NEXT PAGE
005646	103330			BCC	LOP31D		:BRANCH TO TEST NEXT PAGE IN THIS MODE
005650	020027	001152		CMP	RO,#STAEND		:IF DONE THIS MODE, CHECK NEXT MODE
005654	002720			BLT	LOP31C		:LOOP UNTIL ALL STATES HAVE BEEN TESTED
005656	005077	173142		CLR	ASRO		:REINITIALIZE SRO

:SHOW THAT A REFERENCE TO A NR PAGE WILL SET BOTH THE NR AND PL  
 :ERROR BITS IF IT IS OUTSIDE THE MAPPED PAGE LENGTH

005662	104400						TEST26: SCOPE
005664	012706	001000		MOV	#KSTACK,SP		:INITIALIZE KERNEL STACK POINTER
005670	004767	010422		JSR	PC,ORDER		:CHECK TEST SEQUENCE + INIT SRO
005674	000026			26			:TEST NUMBER
005676	104006			HLT			:TEST EXECUTED OUT OF SEQUENCE
005700	004767	007412		JSR	%7,RWALL		:MAP ALL PAGES RW,4K,BANK 0
005704	012777	000004	173164	MOV	#4,AKPDR1		:MAP KERNEL 1 NR, 1 PAGE
005712	004767	010464		JSR	PC,KERN7		:MAP KERNEL PAR/PDR 7 TO EXT BANK
005716	012777	005742	173240	MOV	#RET33,AKTVEC		:SETUP ABORT RETURN
005724	005077	173236		CLR	AKTSTA		
005730	005277	173070		INC	ASRO		:TURN ON KT11-D
005734	005737	030000		TST	AS30000		:REFERENCE NR KERNEL 1 - SHOULD ABORT
005740	000000			HALT			:NO NR ABORT
005742	022777	140003	173054	RET33: CMP	#140003,ASRO		:CHECK SRO
005750	001401			BEG	.+4		
005752	104006			HLT			:SRO INCORRECT - SHOULD SHOW KERNEL :PAGE 1, AND BOTH NR + PL ERRORS SET
005754	005077	173044		CLR	ASRO		
005760	016777	173202	173176	MOV	KTSTA,AKTVEC		:RESTORE TRAP CATCHER

:SHOW THAT KERNEL AND USER STACKS ARE ACCESSED CORRECTLY. AN IOT IS DONE TO  
 :EACH MODE. THE LOCATION WRITTEN INTO WHEN THE STACK IS PUSHED  
 :SHOWS WHICH STACK WAS USED.

005766	104400						TEST27: SCOPE
005770	012706	001000		MOV	#KSTACK,SP		:INITIALIZE KERNEL STACK POINTER
005774	004767	010316		JSR	PC,ORDER		:CHECK TEST SEQUENCE + INIT SRO
006000	000027			27			:TEST NUMBER
006002	104006			HLT			:TEST EXECUTED OUT OF SEQUENCE
006004	004767	007262		JSR	%7,CLRALL		:INITIALIZE ALL KT11-D REGISTERS
006010	012706	000500		MOV	#500,SP		:SET THE KERNEL STACK TO VIRTUAL ADDRESS 500
006014	012737	140000	177776	MOV	#140000,ASPS		
006022	012706	000100		MOV	#100,SP		:SET THE USER STACK TO VA 100
006026	005037	177776		CLR	ASPS		
006032	012777	077406	173034	MOV	#77406,AKPDR0		:MAP KERNEL, AND USER TO BANK 0,4K,RW
006040	012777	077406	172766	MOV	#77406,AKUPDR0		
006046	012737	006114	000020	MOV	#KRET34,AS20		:TEST USING IOT TRAP (THRU KERNEL SPACE)
006054	005037	000022		CLR	AS22		:RETURN FROM TRAP IN KERNEL MODE
006060	016701	172740		MOV	SRO,R1		:REFERENCE SRO THRU R1
006064	004767	010312		JSR	PC,KERN7		:MAP KERNEL PAR/PDR 7 TO EXT BANK
006070	012777	077406	172754	MOV	#77406,AKUPDR7		:MAP USER PAGE 7
006076	012777	007600	172766	MOV	#7600,AKUPAR7		:TO THE EXTERNAL BANK
006104	005277	172714		INC	ASRO		:TURN ON KT11-D
006110	000004			IOT			:SHOULD USE STACK IN KERNEL ADDRESS SPACE
006112	000240			NOP			
006114	005011			KRET34: CLR	AR1		:TURN OFF KT11-D
006116	012737	006150	000020	MOV	#URET34,AS20		:SETUP FOR IOT TO USER
006124	012737	140000	000022	MOV	#140000,AS22		
006132	012737	140000	177776	MOV	#140000,ASPS		



N02

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 26  
DBKTAD.P11 13-SEP-76 10:28

SEQ 0026

006140	005277	172660		INC	@SR0		;TURN ON KT11-D
006144	000004			IOT			;SHOULD USE STACK IN USER SPACE
006146	000240			NOP			
006150	005011		URET34:	CLR	@R1		;TURN OFF KT11-D
006152	022737	006112	000474	CMP	#KRET34-2,@#474		
006160	001401			BEG	.+4		
006162	104006			HLT			;KERNEL STACK CONTENTS WRONG. PC NOT WHERE IT
006164	022737	000000	000476	CMP	#0,@#476		;SHOULD HAVE BEEN PUSHED OR



006172	001401			BEQ	.+4	: VALUE WRONG
006174	104006			HLT		: KERNEL STACK WRONG-TRAP STATUS NOT
006176	022737	006146	000074	CMP	#URET34-2, @#74	: NOT WHERE IT SHOULD HAVE BEEN PUSHED
006204	001401			BEQ	.+4	: OR VALUE WRONG
006206	104006			HLT		: USER STACK WRONG-PC NOT WHERE
006210	022737	140000	000076	CMP	#140000, @#76	: IT SHOULD HAVE BEEN PUSHED
006216	001401			BEQ	.+4	: OR VALUE WRONG
006220	104006			HLT		: USER STACK WRONG-TRAP STATUS
						: NOT WHERE IT SHOULD HAVE BEEN
						: PUSHED OR VALUE WRONG
						: REINITIALIZE LOCATIONS CHECKED
006222	012737	000076	000074	MOV	#76, @#74	
006230	005037	000076		CLR	@#76	
006234	012737	000476	000474	MOV	#476, @#474	
006242	005037	000476		CLR	@#476	
006246	012706	001000		MOV	#KSTACK, SP	

: SHOW THAT TRAP, EMT, AND INTERRUPTS TAKE VECTORS FROM KERNEL  
 : IRREGARDLESS OF THE MODE AT THE TIME OF THE TRAP SEQUENCE  
 : ALSO SHOW THAT ODD-ADDRESS TRAP (AN "INTERNAL"  
 : TRAP) TAKES ITS VECTOR FROM KERNEL  
 : NOTE THAT IF DUAL ADDRESSING OCCURS, THE ERROR  
 : ADDRESS WILL BE USED (THE 0 OVERRIDES THE 1)

006252	104400			TEST30: SCOPE		
006254	012706	001000		MOV	#KSTACK, SP	: INITIALIZE KERNEL STACK POINTER
006260	004767	010032		JSR	PC, ORDER	: CHECK TEST SEQUENCE + INIT SR0
006264	000030			30		: TEST NUMBER
006266	104006			HLT		: TEST EXECUTED OUT OF SEQUENCE
006270	005077	172530		CLR	@SR0	
006274	004767	007016		JSR	%7, RWALL	: MAP ALL PAR/PDR PAIR'S RW, 4K, BANK 0
006300	012777	000001	172546	MOV	#1, @UPAR0	: OFFSET USER 0 1 PAGE
006306	004767	010070		JSR	PC, KERN7	: MAP KERNEL PAR/PDR 7 TO EXT BANK
006312	012777	007600	172552	MOV	#7600, @UPAR7	: MAP USER 7 TO THE EXTERNAL BANK
006320	016701	172500		MOV	SR0, R1	: SETUP R1 TO REFERENCE SR0
006324	012737	140000	177776	MOV	#140000, @#PS	: SETUP USER STACK
006332	012706	000400		MOV	#USTACK, SP	
006336	005037	177776		CLR	@#PS	
006342	012706	001000		MOV	#KSTACK, SP	: SETUP THE KERNEL STACK POINTER
006346	012737	006506	000130	MOV	#NG35B, @#130	: SETUP FAILURE RETURN
006354	012737	006524	000030	MOV	#OK35B, @#30	: SETUP SUCCESS RETURN
006362	005037	000132		CLR	@#132	
006366	005037	000032		CLR	@#32	
006372	012737	140000	177776	MOV	#140000, @#PS	: SET MODE TO USER
006400	005277	172420		INC	@SR0	: TURN ON KT11-D
006404	000000			0		
006406	000000			0		
006410	000000			0		
006412	000000			0		
006414	000000			0		
006416	000000			0		
006420	000000			0		
006422	000000			0		
006424	000000			0		
006426	000000			0		
006430	000000			0		
006432	000000			0		
006434	000000			0		











```

007100 000000
007102 000000
007104 000000
007106 000000
007110 000000
007112 000000
007114 012737 000100 177564 MOV #100, @#177564 ;SET TTY INTERRUPT ENABLE-SHOULD
007122 000240 NOP ;INTERRUPT IMMEDIATELY
007124 000240 NOP
007126 005011 CLR @R1 ;TURN OFF KT11-D
007130 005077 171662 CLR @TCSR ;CLEAR TTY IE
007134 104006 HLT ;TTY FAILED TO INTERRUPT
007136 000412 BR ODDAD
007140 022626 NG35D: CMP (SP)+, (SP)+ ;RESTORE STACK POINTER
007142 005011 CLR @R1 ;TURN OFF KT11-D
007144 005077 171646 CLR @TCSR ;CLEAR TTY IE
007150 104006 HLT ;TTY INTERRUPT DIDN'T GO THRU KERNEL
007152 000404 BR ODDAD
007154 022626 OK35D: CMP (SP)+, (SP)+ ;RESTORE STACK POINTER
007156 005011 CLR @R1 ;TURN OFF KT11-D
007160 005077 171632 CLR @TCSR
007164 012737 000066 000064 ODDAD: MOV #66, @#64 ;RESTORE TTY VECTOR RETURN TO CAUSE
007172 005037 000066 CLR @#66 ;A HALT ON A FALSE INTERRUPT
007176 012737 000162 000160 MOV #162, @#160
007204 005037 000162 CLR @#162
007210 005037 177776 CLR @#PS
007214 012737 007360 000104 MOV #NG35E, @#104 ;SETUP INTERNAL TRAP FAILURE RETURN
007222 012737 000340 000106 MOV #340, @#106
007230 012737 007370 000004 MOV #OK35E, @#4 ;SETUP INTERNAL TRAP SUCCESS RETURN
007236 005037 000006 CLR @#6
007242 012737 140000 177776 MOV #140000, @#PS ;SET MODE TO USER
007250 005277 171550 INC @SRO ;TURN ON KT11-D
007254 000000
007256 000000
007260 000000
007262 000000
007264 000000
007266 000000
007270 000000
007272 000000
007274 000000
007276 000000
007300 000000
007302 000000
007304 000000
007306 000000
007310 000000
007312 000000
007314 000000
007316 000000
007320 000000
007322 000000
007324 000000
007326 000000
007330 000000
007332 000000

```



```

007334 000000          0
007336 000000          0
007340 000000          0
007342 000000          0
007344 000000          0
007346 000000          0
007350 000000          0
007352 000000          0
007354 005737 000001  TST      @#1          ; ODD ADDRESS REFERENCE - AN "INTERNAL
                                          ; TRAP" SHOULD OCCUR
007360 022626          NG35E:  CMP      (SP)+,(SP)+  ; RESTORE STACK POINTER
007362 005011          CLR      @R1          ; TURN OFF KT11-D
007364 104006          HLT                    ; ODD ADDRESS TRAP DIDN'T TAKE
                                          ; VECTOR FROM KERNEL
007366 000407          OK35E:  BR       END35
007370 022626          CMP      (SP)+,(SP)+  ; RESTORE STACK POINTER
007372 005011          CLR      @R1          ; TURN OFF KT11-D
007374 032737 000340 177776  BIT      @340,@#PS  ; WAS CORRECT STATUS PICKED UP?
007402 001401          BEQ      .+4          ; YES- BRANCH
007404 104006          HLT                    ; PICKED UP NEW STATUS WORD FROM USER SPACE
007406 012737 000006 000004  END35:  MOV      @6,@#4
007414 012737 000106 000104  MOV      @106,@#104  ; RESTORE TRAP CATCHER
    
```

; SHOW THAT THE ABORT LOGIC "LOCKS" SRO AND SR2 AFTER A NR  
 ; ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SRO, WHEN  
 ; THEY RESUME TRACKING. A NR ERROR SHOULD STILL ABORT TO 250 EVEN  
 ; WHEN BIT 15 (SRO) IS ALREADY SET

```

007422 104400          TEST31: SCOPE
007424 012706 001000  MOV      #KSTACK,SP  ; INITIALIZE KERNEL STACK POINTER
007430 004767 006662  JSR      PC,ORDER    ; CHECK TEST SEQUENCE + INIT SRO
007434 000031          31          ; TEST NUMBER
007436 104006          HLT                    ; TEST EXECUTED OUT OF SEQUENCE
007440 004767 005626  JSR      %7,CLRALL   ; CLEAR ALL KT11-D REGISTERS
007444 004767 006732  JSR      PC,KERN7    ; MAP KERNEL PAR/PDR 7 TO EXT BANK
007450 012777 077406 171416  MOV      @77406,@KPDRO ; MAP KERNEL 0 RW,RK,BANK0
007456 012777 077400 171412  MOV      @77400,@KPDR1 ; MAP KERNEL 1 NR,4 K,BANK0
007464 012777 007520 171472  MOV      @INT36,@KTVEC ; SETUP RETURN VECTOR
007472 005077 171470  CLR      @KTSTA
007476 005277 171322  INC      @SRO
007502 013737 037776 037776  ADR36:  MOV      @#37776,@#37776  ; TURN ON KT11-D
007510 005077 171310  CLR      @SRO          ; REFERENCE KERNEL 1 - 1ST ABORT
007514 104006          HLT                    ; TURN OFF KT11-D
007516 000510          BR       DONE36       ; REFERENCE TO KERNEL 1
007520 042777 000001 171276  INT36:  BIC      #1,@SRO    ; DIDN'T ABORT
007526 022777 100002 171270  CMP      #100002,@SRO ; TURN OFF KT11-D
007534 001401          BEQ      .+4          ; CHECK SRO
007536 104006          HLT                    ; SRO INCORRECT AFTER NR ABORT
007540 012777 007574 171416  MOV      #INT36A,@KTVEC ; SETUP NEW RETURN VECTOR
007546 022626          CMP      (R6)+,(R6)+  ; RESTORE STACK POINTER
007550 012702 037776  MOV      @37776,R2    ; SETUP R2 TO REFERENCE KERNEL 1
007554 052777 000001 171242  BIS      #1,@SRO    ; TURN ON KT11-D
007562 012242          MOV      (R2)+,-(R2) ; REFERENCE KERNEL 1 -2ND ABORT
007564 005077 171234  ADR36A: CLR      @SRO  ; TURN OFF KT11-D
007570 104006          HLT                    ; 2ND REFERENCE TO KERNEL 1
007572 000462          BR       DONE36       ; DIDN'T ABORT
    
```



```

007574 042777 000001 171222 INT36A: BIC #1,SR0 ;TURN OFF KT11-D
007602 022777 100002 171214 CMP #100002,SR0 ;CHECK SR0
007610 001401 BEQ .+4
007612 104006 HLT ;SR0 INCORRECT AFTER 2ND NR ABORT
007614 022777 007502 171210 CMP #ADR36,SR2 ;CHECK SR2
007622 001401 BEQ .+4
007624 104006 HLT ;SR2 DOESN'T CONTAIN VALUE FROM 1ST ABORT
007626 021627 007564 CMP (R6),#ADR36A ;CHECK ADDRESS PUSHED ON STACK
007632 001401 BEQ .+4
007634 104006 HLT ;INCORRECT ADDRESS ON STACK
007636 022626 CMP (R6)+,(R6)+ ;RESTORE STACK POINTER
007640 012777 007674 171316 MOV #INT36B,@KTVEC ;CHANGE RETURN ADDRESS
007646 005077 171152 CLR SR0 ;CLEAR NR ERROR BIT-SHOULD
; "UNLOCK" ERROR TRACKING
; SETUP R2 TO REFERENCE KERNEL 1
; TURN ON KT11-D
007652 012702 037776 MOV #37776,R2 ;3RD NR REFERENCE, ERROR BIT WAS CLEARED
007656 005277 171142 INC SR0 ;TURN OFF KT11-D
007662 012242 ADR36B: MOV (R2)+,-(R2) ;3RD REFERENCE TO KERNEL 1
007664 005077 171134 ADR36C: CLR SR0 ;DIDN'T ABORT
;TURN OFF KT11-D
007670 104006 HLT ;3RD REFERENCE TO KERNEL 1
007672 000422 BR DONE36 ;DIDN'T ABORT
007674 042777 000001 171122 INT36B: BIC #1,SR0 ;TURN OFF KT11-D
007702 022777 100002 171114 CMP #100002,SR0 ;CHECK SR0
007710 001401 BEQ .+4
007712 104006 HLT ;SR0 INCORRECT
007714 022777 007662 171110 CMP #ADR36B,SR2 ;CHECK SR2
007722 001401 BEQ .+4
007724 104006 HLT ;SR2 INCORRECT - SHOULD CONTAIN
;LAST FETCH ADDRESS BEFORE ABORT
;CHECK STACK
007726 022716 007664 CMP #ADR36C,(SP)
007732 001401 BEQ .+4
007734 104006 HLT ;PC ON STACK INCORRECT
007736 022626 CMP (R6)+,(R6)+ ;RESTORE STACK POINTER
007740 005077 171060 DONE36: CLR SR0 ;CLEAR ERROR BIT
007744 005077 171216 CLR @KTSTA ;CHANGE TRAP RETURN TO CAUSE A HALT
007750 016777 171212 171206 MOV KTSTA,@KTVEC ;ON A FALSE INTERRUPT

;SHOW THAT THE ABORT LOGIC "LOCKS" SR0 AND SR2 AFTER A PL
;ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0, WHEN
;THEY RESUME TRACKING. A PL ERROR SHOULD STILL ABORT TO 250 EVEN
;WHEN BIT 14 (SR0) IS ALREADY SET
TEST32: SCOPE
007756 104400 MOV #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
007760 012706 001000 JSR PC,ORDER ;CHECK TEST SEQUENCE + INIT SR0
007764 004767 006326 JSR 32 ;TEST NUMBER
007770 000032 HLT ;TEST EXECUTED OUT OF SEQUENCE
007772 104006 JSR %7,CLRALL ;CLEAR ALL KT11-D REGISTERS
007774 004767 005272 JSR PC,KERN7 ;MAP KERNEL PAR/PDR 7 TO EXT BANK
010000 004767 006376 JSR #77406,@KPDRO ;MAP KERNEL 0 RW,RK,BANK0
010004 012777 077406 171062 MOV #17406,@KPDRI ;MAP KERNEL 1 PL,1 K,BANK0
010012 012777 017406 171056 MOV #INT37,@KTVEC ;SETUP RETURN VECTOR
010020 012777 010054 171136 CLR @KTSTA
010026 005077 171134 INC SR0 ;TURN ON KT11-D
010032 005277 170766 ADR37: MOV @#37776,@#37776 ;REFERENCE KERNEL 1 - 1ST ABORT
010036 013737 037776 037776 CLR SR0 ;TURN OFF KT11-D
010044 005077 170754 HLT ;REFERENCE TO KERNEL 1
010050 104006 BR DONE37 ;DIDN'T ABORT
010052 000510

```



```

010054 042777 000001 170742 INT37: BIC #1, @SR0 ;TURN OFF KT11-D
010062 022777 040002 170734 CMP #40002, @SR0 ;CHECK SR0
010070 001401 BEQ .+4
010072 104006 HLT ;SR0 INCORRECT AFTER PL ABORT
010074 012777 010130 171062 MOV #INT37A, @KTVEC ;SETUP NEW RETURN VECTOR
010102 022626 CMP (R6)+, (R6)+ ;RESTORE STACK POINTER
010104 012702 037776 MOV #37776, R2 ;SETUP R2 TO REFERENCE KERNEL 1
010110 052777 000001 170706 BIS #1, @SR0 ;TURN ON KT11-D
010116 012242 MOV (R2)+, -(R2) ;REFERENCE KERNEL 1 -2ND ABORT
010120 005077 170700 ADR37A: CLR @SR0 ;TURN OFF KT11-D
010124 104006 HLT ;2ND REFERENCE TO KERNEL 1
010126 000462 BR DONE37 ;DIDN'T ABORT
010130 042777 000001 170666 INT37A: BIC #1, @SR0 ;TURN OFF KT11-D
010136 022777 040002 170660 CMP #40002, @SR0 ;CHECK SR0
010144 001401 BEQ .+4
010146 104006 HLT ;SR0 INCORRECT AFTER 2ND PL ABORT
010150 022777 010036 170654 CMP #ADR37, @SR2 ;CHECK SR2
010156 001401 BEQ .+4
010160 104006 HLT ;SR2 DOESN'T CONTAIN VALUE FROM 1ST ABORT
010162 021627 010120 CMP (R6), #ADR37A ;CHECK ADDRESS PUSHED ON STACK
010166 001401 BEQ .+4
010170 104006 HLT ;INCORRECT ADDRESS ON STACK
010172 022626 CMP (R6)+, (R6)+ ;RESTORE STACK POINTER
010174 012777 010230 170762 MOV #INT37B, @KTVEC ;CHANGE RETURN ADDRESS
010202 005077 170616 CLR @SR0 ;CLEAR PL ERROR BIT-SHOULD
; "UNLOCK" ERROR TRACKING
; SETUP R2 TO REFERENCE KERNEL 1
; TURN ON KT11-D
; 3RD PL REFERENCE, ERROR BIT WAS CLEARED
; TURN OFF KT11-D
; 3RD REFERENCE TO KERNEL 1
; DIDN'T ABORT
; TURN OFF KT11-D
; CHECK SR0
; SR0 INCORRECT
; CHECK SR2
; SR2 INCORRECT - SHOULD CONTAIN
; LAST FETCH ADDRESS BEFORE ABORT
; CHECK STACK
; PC ON STACK INCORRECT
; RESTORE STACK POINTER
; CLEAR ERROR BIT
; CHANGE TRAP RETURN TO CAUSE A HALT
; ON A FALSE INTERRUPT

```

```

; SHOW THAT THE ABORT LOGIC "LOCKS" SR0, AND SR2 AFTER A ACC
; ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0, WHEN
; THEY RESUME TRACKING. A ACC ERROR SHOULD STILL ABORT TO 250 EVEN
; WHEN BIT 13 (SR0) IS ALREADY SET

```

```

010312 104400 TEST33: SCOPE
010314 012706 001000 MOV #KSTACK, SP ;INITIALIZE KERNEL STACK POINTER
010320 004767 005772 JSR PC, ORDER ;CHECK TEST SEQUENCE + INIT SR0
010324 000033 33 ;TEST NUMBER

```







```

010630 005077 170170      DONE40: CLR      JSRO      ;CLEAR ERROR BIT
010634 005077 170326      CLR      ;CHANGE TRAP RETURN TO CAUSE A HALT
010640 016777 170322 170316  MOV      KTSTA,JSRVEC ;ON A FALSE INTERRUPT

;SHOW THAT INIT CLEARS SRO<13-15>
TEST34: SCOPE
010646 104400      MOV      #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
010650 012706 001000      JSR      PC,ORDER    ;CHECK TEST SEQUENCE + INIT SRO
010654 004767 005436      34      ;TEST NUMBER
010660 000034      HLT      ;TEST EXECUTED OUT OF SEQUENCE
010662 104006      MOV      #340,JSROH ;SET SRO BITS 13-15
010664 112777 000340 170134  CMP      #340,JSROH ;MAKE SURE THEY SET CORRECTLY
010672 122777 000340 170126      BEQ      .+4
010700 001401      HLT      ;SRO INCORRECT (HIGH BYTE)
010702 104006      RESET    ;ISSUE INIT
010704 000005      CMP      #0,JSROH   ;CHECK SRO HIGH BYTE
010706 122777 000000 170112      BEQ      .+4
010714 001401      HLT      ;SRO INCORRECT AFTER INIT
010716 104006      MOV      #10,ICOUNT ;DROP ITERATION COUNT
010720 012767 000010 004642

;SHOW THAT INIT CLEARS SRO AFTER ABORT
TEST35: SCOPE
010726 104400      MOV      #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
010730 012706 001000      JSR      PC,ORDER    ;CHECK TEST SEQUENCE + INIT SRO
010734 004767 005356      35      ;TEST NUMBER
010740 000035      HLT      ;TEST EXECUTED OUT OF SEQUENCE
010742 104006      JSR      %7,RWALL   ;MAP ALL PAR/PDR PAIR'S 4K,RW,BANK 0
010744 004767 004346      MOV      #416,JKPDR0 ;MAP KERNEL 0 RW,4K LESS 1 PAGE
010750 012777 000416 170116      PC,KERN7 ;MAP KERNEL PAR/PDR 7 TO EXT BANK
010756 004767 005420      MOV      #77400,JKPDR1 ;MAP KERNEL PAGE 1 NR
010762 012777 077400 170106      MOV      #RET2,JSRVEC ;SETUP ABORT RETURN
010770 012777 011024 170166      CLR      JSRVEC
010776 005077 170164      MOV      #20,-(SP) ;SET T BIT IN STATUS ON STACK
011002 012746 000020      MOV      #ADR2,-(SP) ;SETUP ADDRESS ON STACK
011006 012746 011020      INC      JSRO ;TURN ON KT11-D
011012 005277 170006      RTI     ;SHOULD TRACE TRAP IMMEDIATELY SINCE T-BIT
011016 000002      ;IS SET - SINCE T-BIT VECTOR IS OUTSIDE ALLOWED
;PAGE LENGTH, SHOULD DO A MEMORY
;MANAGEMENT ABORT
;NO PL ABORT OCCURRED

011020 000000      ADR2: HALT
011022 000412      BR      DONE2
011024 022777 040001 167772 RET2:  CMP      #40001,JSRO ;CHECK SRO
011032 001401      BEQ      .+4
011034 104006      HLT      ;SRO INCORRECT - SHOULD SHOW
;REFERENCE TO KERNEL 0
;AND PL ABORT SHOULD BE SET
;ISSUE INIT - SHOULD CLEAR SRO
;CHECK SRO

011036 000005      RESET
011040 005777 167760      TST      JSRO
011044 001401      BEQ      .+4
011046 104006      HLT      ;SRO INCORRECT AFTER INIT
011050 005077 167750      DONE2: CLR      JSRO ;REINITIALIZE SRO
011054 016777 170106 170102      MOV      KTSTA,JSRVEC
011062 012737 000016 000014      MOV      #16,JSR14 ;RESTORE T-BIT TRAP CATCHER

;SHOW THAT INIT CLEARS SRO<0-3,5-6>
    
```



# K03

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 36  
 DBKTAD.P11 13-SEP-76 10:28

SEQ 0036

```

;REFERENCE NR USER PAGE 7 TO SET ALL BITS(0-6)
;THEN ISSUE INIT
TEST36: SCOPE
011070 104400
011072 012706 001000      MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
011076 004767 005214      JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
011102 000036              36                      ;TEST NUMBER
011104 104006              HLT
011106 004767 004204      JSR      %7,RWALL        ;TEST EXECUTED OUT OF SEQUENCE
;MAP ALL PAR/PDR PAIR'S INITIALLY RW,4K,
;BANK 0
;MAKE USER 7 NR
;MAP KERNEL PAR/PDR 7 TO EXT BANK
011112 012777 077400 167732  MOV      #77400,%UPDR7
011120 004767 005256      JSR      PC,KERN7
011124 012777 011162 170032  MOV      #RET3,%KTVEC    ;SETUP ABORT RETURN
011132 005077 170030      CLR      %KTSTA
011136 012737 140000 177776  MOV      #140000,%#PS    ;SET MODE TO USER
011144 012706 000400      MOV      #USTACK,R6     ;SETUP USER STACK IN CASE NEEDED
011150 005277 167650      INC      %SRO           ;TURN ON KT11-D
011154 005737 160000      TST      %#160000       ;REFERENCE PAGE 7
011160 000777              BR
011162 022777 100157 167634  RET3:    CMP      #100157,%SRO    ;NO ABORT ON NR REFERENCE
011170 001401              BEQ      .+4            ;CHECK SRO
011172 104006              HLT                    ;SRO INCORRECT - SHOULD HAVE TRACKED
;NR REFERENCE TO USER 7
;ISSUE INIT
;CHECK SRO
011174 000005              RESET
011176 005777 167622      TST      %SRO
011202 001401              BEQ      .+4
011204 104006              HLT                    ;SRO INCORRECT AFTER INIT
011206 005077 167612      CLR      %SRO
011212 012767 000010 004350  MOV      #10,%ICOUNT    ;DROP ITERATION COUNT
011220 016777 167742 167736  MOV      KTSTA,%KTVEC

;SHOW THAT BYTE ADDRESSING OF SRO WORKS
TEST37: SCOPE
011226 104400
011230 012706 001000      MOV      #KSTACK,SP    ;INITIALIZE KERNEL STACK POINTER
011234 004767 005056      JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
011240 000037              37                      ;TEST NUMBER
011242 104006              HLT
011244 004767 004046      JSR      %7,RWALL        ;MAP ALL PAR/PDR PAIRS RW,4K,BANK 0
011250 004767 005126      JSR      PC,KERN7        ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011254 012777 160001 167542  MOV      #160001,%SRO    ;TURN ON KT11-D AND SET ERROR FLAGS
011262 105077 167536      CLRB    %SRO           ;DATOB (LOW) TO SRO
011266 032777 160000 167530  BIT      #160000,%SRO    ;CHECK SRO
011274 001001              BNE      .+4
011276 104006              HLT                    ;SRO INCORRECT AFTER DATOB
011300 012777 160001 167516  MOV      #160001,%SRO    ;DATOB (HIGH) TO SRO
011306 105077 167514      CLRB    %SROH
011312 022777 000017 167504  CMP      #17,%SRO        ;CHECK SRO
011320 001401              BEQ      .+4
011322 104006              HLT                    ;SRO INCORRECT AFTER DATOB
011324 005077 167474      CLR      %SRO

;SHOW THAT SRO <1-3> TRACK PAGE REFERENCED IF
;KT11-D IS ON AND REFERENCE IS NOT TO A KT11-D REGISTER
;SHOW THAT EACH VALUE IS CORRECTLY "LOCKED" IN SRO AFTER AN ABORT
TEST40: SCOPE
011330 104400
011332 012706 001000      MOV      #KSTACK,SP    ;INITIALIZE KERNEL STACK POINTER
011336 004767 004754      JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO

```



```

011342 000040          40          ;TEST NUMBER
011344 104006          HLT          ;TEST EXECUTED OUT OF SEQUENCE
011346 004767 003744   JSR          %7,RWALL          ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011352 004767 005024   JSR          PC,KERN7
011356 012777 011434 167600   MOV          #RETS,@KTVEC
011364 005077 167576   CLR          @KTSTA
011370 016701 167440   MOV          UPDR0,R1
011374 005002          CLR          R2
011376 012703 100141   MOV          #100141,R3
011402 012704 000010   MOV          #10,R4
011406 012711 077400   MOV          #77400,@R1          ;MAKE USER NR
011412 012737 140000 177776   MOV          #140000,@#PS      ;ENTER USER MODE
011420 005277 167400   INC          @SRO
011424 005712          TST          @R2
011426 000777          BR          .          ;REFERENCE TO NR PAGE DIDN'T ABORT
011430 000005          RESET        ;AFTER ERROR, TURN OFF KT11-D
011432 000423          BR          .
011434 017705 167364   RETS: MOV      @SRO,R5          ;SAVE CONTENTS OF SRO
011440 005077 167360   CLR          @SRO          ;TURN OFF KT11-D
011444 020503          CMP          R5,R3          ;CHECK SAVED CONTENTS OF SRO
011446 001401          BEQ          .+4
011450 104006          HLT          ;SRO INCORRECT
011452 020167 167356   CMP          R1,UPDR0        ;IS USER 0 UNDER TEST
011456 001002          BNE          LOP5A          ;NO, CONTINUE
011460 012711 077406   LOP5A: MOV      #77406,@R1      ;MAKE USER 0 RESIDENT
011464 022626          CMP          (R6)+,(R6)+
011466 005721          TST          (R1)+
011470 062703 000002   ADD          #2,R3
011474 062702 020000   ADD          #20000,R2
011500 077436          SOB          R4,LOP5
011502 016777 167460 167454   DONES: MOV      KTSTA,@KTVEC
011510 005077 167452   CLR          @KTSTA

;SHOW THAT SRO <5-6> TRACK PAGE REFERENCED (MODE) IF
;KT11-D IS ON AND THE REFERENCE IS NOT TO A KT11-D REGISTER
;SHOW THAT EACH VALUE IS CORRECTLY "LOCKED" IN SRO AFTER AN ABORT
TEST41: SCOPE
011514 104400          MOV          #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
011516 012706 001000   JSR          PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
011522 004767 004570   41          ;TEST NUMBER
011526 000041          HLT          ;TEST EXECUTED OUT OF SEQUENCE
011530 104006          JSR          %7,RWALL        ;MAP ALL PAGES RW,4K, BANK 0
011532 004767 003560   JSR          PC,KERN7        ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011536 004767 004640   MOV          #77400,@KPDR1   ;SETUP PAGE 1 IN EACH MODE TO BE NR
011542 012777 077400 167326   MOV          #77400,@UPDR1
011550 012777 077400 167260   MOV          #RET7A,@KTVEC
011556 012777 011604 167400   INC          @SRO          ;SETUP ABORT RETURN
011564 005277 167234   TST          @#20000        ;TURN ON KT11-D
011570 005737 020000   CLR          @SRO          ;REFERENCE PAGE 1 (NR)
011574 005077 167224   HLT          ;TURN OFF KT11-D
011600 104006          BR          .          ;NR REFERENCE DIDN'T ABORT
011602 000436          RET7A: MOV      @SRO,R1      ;SAVE SRO CONTENTS IN R1
011604 017701 167214   CLR          @SRO          ;TURN OFF KT11-D
011610 005077 167210   CMP          #100003,R1     ;CHECK SAVED CONTENTS OF SRO
011614 022701 100003   BEQ          .+4
011620 001401          HLT          ;SRO INCORRECT SHOULD SHOW NR ERR, KERNEL PAGE 1
011622 104006

```



```

011624 012777 011660 167332      MOV      #RET7C, @KTVEC      ; SETUP NEXT ABORT RETURN
011632 012737 140000 177776      MOV      #140000, @#PS      ; CHANGE MODE TO USER
011640 005277 167160                INC      @SRO                ; TURN ON KT11-D
011644 005737 020000                TST      @#20000             ; REFERENCE USER PAGE 1 (NR)
011650 005077 167150                CLR      @SRO                ; TURN OFF KT11-D
011654 104006                HLT                               ; NR REFERENCE DIDN'T ABORT
011656 000410                BR                               ;
011660 017701 167140      RET7C: MOV      @SRO, R1          ; SAVE CONTENTS OF SRO
011664 005077 167134                CLR      @SRO                ; TURN OFF KT11-D
011670 022701 100143                CMP      #100143, R1         ; CHECK SAVED CONTENTS OF SRO
011674 001401                BEQ                               ;
011676 104006                HLT                               ; SRO INCORRECT - SHOULD SHOW NR
                                           ; ERROR, USER PAGE 1
011700 016777 167262 167256  DONE7: MOV      KTSTA, @KTVEC      ; RESTORE TRAP CATCHER

; SHOW THAT SRO <1-3,5-6> DOESN'T TRACK IF KT11-D IS OFF BUT DOES IF REFERENCE IS TO
; AN INTERNAL (KT11-D) REGISTER
TEST42: SCOPE
011706 104400                MOV      #KSTACK, SP        ; INITIALIZE KERNEL STACK POINTER
011710 012706 001000                JSR      PC, ORDER          ; CHECK TEST SEQUENCE + INIT SRO
011714 004767 004376                42                          ; TEST NUMBER
011720 000042                HLT                               ; TEST EXECUTED OUT OF SEQUENCE
011722 104006                JSR      %7, RWALL           ; SET ALL PAR/PDR PAIRS RW, 4K, BANK 0
011724 004767 003366                MOV      #7600, @UPAR7      ; MAP USER 7 TO THE EXT. BANK
011730 012777 007600 167134                MOV      #140000, @#PS      ; SET MODE TO USER
011736 012737 140000 177776                INC      @SRO                ; TURN ON KT11-D
011744 005277 167054                BIC      #1, @SRO           ; TURN OFF KT11-D
011750 042777 000001 167046                CLR      @#PS               ; CHANGE TO KERNEL MODE
011756 005037 177776                CMP      #156, @SRO         ; CHECK SRO
011762 022777 000156 167034                BEQ      .+4                 ;
011770 001401                HLT                               ; SRO INCORRECT - SHOULD SHOW REFERENCE
011772 104006                HLT                               ; TO USER 7
                                           ; IF IT SHOWS USER 0
                                           ; IT DID NOT TRACK THE INTERNAL REFERENCE
                                           ; IF IT SHOWS KERNEL 0, IT IS
                                           ; TRACKING WITH KT11-D OFF
011774 005077 167024                CLR      @SRO

; SHOW THAT IF AN INSTRUCTION IS COMPLETED BEFORE A MEMORY MANAGEMENT FAULT
; OCCURS, SR2 WILL CONTAIN THE ADDRESS OF LAST FETCH BEFORE ABORT
; TO TEST THIS, TRACE TRAP IS USED. THE VECTOR IS MADE NON-RESIDENT BY MAKING
; KERNEL PAGE 0 MAPPED DOWN FROM 17776 TO 100. THUS THE MEMORY MANAGEMENT
; VECTOR IS RESIDENT WHILE THE TRACE TRAP VECTOR IS OUTSIDE THE ALLOWED
; PAGE LENGTH.
TEST43: SCOPE
012000 104400                MOV      #KSTACK, SP        ; INITIALIZE KERNEL STACK POINTER
012002 012706 001000                JSR      PC, ORDER          ; CHECK TEST SEQUENCE + INIT SRO
012006 004767 004304                43                          ; TEST NUMBER
012012 000043                HLT                               ; TEST EXECUTED OUT OF SEQUENCE
012014 104006                JSR      %7, RWALL           ; INITIALIZE ALL PAGES RW, 4K, BANK 0
012016 004767 003274                MOV      #416, @KPDRO       ; MAP KERNEL TO EXCLUDE
012022 012777 000416 167044                PC, KERN7                   ; LOCATIONS 0 TO 77
                                           ; MAP KERNEL PAR/PDR 7 TO EXT BANK
012030 004767 004346                MOV      #RET11, @KTVEC     ; SETUP MEMORY MANAGEMENT ABORT RETURN
012034 012777 012072 167122                CLR      @KTSTA             ;
012042 005077 167120                MOV      #20, -(SP)         ; PREPARE STACK TO TURN ON T-BIT
012046 012746 000020

```



```

012052 012746 012060      MOV      #.+6,-(SP)      ;SET T-BIT VIA RTT
012056 000006      RTT
012060 012777 000001 166736 ADR11: MOV      #1,SR0      ;TURN ON KT11-D - SHOULD
;ATTEMPT TO TRACE TRAP AT END OF
;INSTRUCTION - SHOULD GET A PAGE
;LENGTH ERROR ON THAT ATTEMPT
;NO PAGE LENGTH ERROR ON TRACE TRAP

012066 000000      HALT
012070 000415      BR
012072 042777 000001 166724 RET11: BIC      #1,SR0      ;TURN OFF KT11-D
012100 022777 040000 166716      CMP      #40000,SR0    ;CK SR0
012106 001401      BEQ      .+4
012110 104006      HLT
012112 022777 012060 166712      CMP      #ADR11,SR2    ;SR0 INCORRECT - PL FAULT,KERNEL 0 REFERENCE COMPLETED
012120 001401      BEQ      .+4          ;CK SR2
012122 104006      HLT
;SR2 INCORRECT - SHOULD CONTAIN
;ADDRESS OF LAST FETCH BEFORE ABORT

012124 005077 166674      CONT11: CLR     SR0      ;REINITIALIZE SR0
012130 016777 167032 167026      MOV     KTSTA,KTVEC   ;RESTORE TRAP CATCHER

;SHOW THAT HAVING THE ABORT ERROR
;BITS SET WILL NOT PREVENT A MEMORY MANAGEMENT TRAP
TEST44: SCOPE
012136 104400      MOV     #KSTACK,SP    ;INITIALIZE KERNEL STACK POINTER
012140 012706 001000      JSR     PC,ORDER      ;CHECK TEST SEQUENCE + INIT SR0
012144 004767 004146      44
;TEST NUMBER
012150 000044      HLT
;TEST EXECUTED OUT OF SEQUENCE
012152 104006      JSR     %7,RWALL      ;INITIALIZE ALL PAR/PDR PAIRS TO RW,4K, BANK 0
012154 004767 003136      MOV     #77402,PKPDR2 ;SET KERNEL PAR/PDR PAIR 2 RRO,4K
012160 012777 077402 166712      JSR     PC,KERN7
;MAP KERNEL PAR/PDR 7 TO EXT BANK
012166 004767 004210      MOV     #RET13A,KTVEC ;SETUP MEMORY MANAGEMENT ABORT RETURN
012172 012777 012234 166764      CLR     KTSTA
012200 005077 166762      INC     SR0
;TURN ON KT11-D
012204 005277 166614      MOV     #160001,SR0   ;SET ABORT ERROR BITS
012210 012777 160001 166606      MOV     #7000,SR0    ;WRITE KERNEL PAR/PDR PAIR 2 (RRO)-SHOULD TRAP
012216 013737 007000 047000      CLR     SR0
;NO TRAP OCCURRED
012224 005077 166574      HLT
012230 104006      BR
012232 000416      RET13A: CMP     (SP)+,(SP)+ ;RESTORE THE STACK POINTER
012234 022626      MOV     SR0,R1        ;SAVE CONTENTS OF SR0
012236 017701 166562      CLR     SR0          ;TURN OFF KT11-D
012242 005077 166556      CMP     #160017,R1
012246 022701 160017      BEQ     .+4
;SAVED CONTENTS OF SR0 INCORRECT
012252 001401      HLT
;CHECK THE PDR CORRESPONDING TO THE TRAP REFERENCE
012254 104006      CMP     #77402,PKPDR2 ;THE PDR CORRESPONDING TO THE TRAP REFERENCE IS INCORREC
012256 022777 077402 166614      BEQ     .+4          ;RESTORE MEMORY MANAGEMENT TRAP RETURN
012264 001401      HLT
;TO CAUSE A HALT ON A FALSE TRAP OR ABORT
012266 104006      MOV     KTSTA,KTVEC   ;REINITIALIZE SR0
012270 016777 166672 166666      CLR     SR0

;SHOW THAT MEMORY MANAGEMENT WILL NOT TRAP ON AN INTERNAL REFERENCE
TEST45: SCOPE
012302 104400      MOV     #KSTACK,SP    ;INITIALIZE KERNEL STACK POINTER
012304 012706 001000      JSR     PC,ORDER      ;CHECK TEST SEQUENCE + INIT SR0
012310 004767 004002      45
;TEST NUMBER
012314 000045      HLT
;TEST EXECUTED OUT OF SEQUENCE
012316 104006

```



```

012320 004767 002772 JSR %7,RWALL ;MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
012324 004767 004052 JSR PC,KERN7 ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012330 012777 012372 166626 MOV #RET16,@KTVEC ;SETUP TRAP RETURN IN CASE
012336 005077 166624 CLR @KTSTA
012342 005277 166456 INC @SRO ;TURN ON KT11-D
012346 005777 166452 TST @SRO ;TRAP REFERENCE TO A KT11-D REGISTER
012352 005077 166446 CLR @SRO
012356 022777 077406 166526 CMP #77406,@KPDR7
012364 001401 BEQ .+4
012366 104006 HLT
012370 000404 BR
012372 042777 000001 166424 RET16: BIC #1,@SRO
012400 104006 HLT
012402 005077 166416 DONE16: CLR @SRO
012406 016777 166554 166550 MOV KTSTA,@KTVEC

;TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN NOT SET)
;KERNEL PAR/PDR PAIR1 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
TEST46: SCOPE
012414 104400 MOV #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
012416 012706 001000 JSR PC,ORDER ;CHECK TEST SEQUENCE + INIT SRO
012422 004767 003670 46 ;TEST NUMBER
012426 000046 HLT ;TEST EXECUTED OUT OF SEQUENCE
012430 104006 JSR %7,RWALL ;INITIALIZE ALL PAR/PDR PAIRS TO RW,4K, BANK 0
012432 004767 002660 JSR PC,KERN7 ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012436 004767 003740 MOV #6,R2 ;R2 CONTAINS VALUE TO BE LOADED IN THE
012442 012702 000006 ;PDR BEING CHECKED (INCLUDING PLF)
012446 012701 020076 MOV #20076,R1 ;R1 IS USED TO REFERENCE THE TOP ADDRESS
012452 012777 012532 166504 LOOP23: MOV #RET23A,@KTVEC ;SETUP ABORT RETURN IN CASE REFERENCE
012460 005077 166502 CLR @KTSTA ;WITHIN ALLOWED AREA ABORTS
012464 005277 166334 INC @SRO ;TURN ON KT11-D
012470 010277 166402 MOV R2,@KPDR1 ;SET KERNEL PAR/PDR PAIR 1 TO NEW PAGE LENGTH
012474 005737 020000 TST @#20000 ;READ LOWER BOUNDARY-SHOULDN'T ABORT
012500 005711 TST @R1 ;READ UPPER ALLOWED BOUNDARY-SHOULDN'T
;ABORT
012502 012777 012552 166454 MOV #RET23B,@KTVEC ;SETUP ABORT RETURN
012510 020127 037776 CMP R1,#37776 ;CHECK FOR DONE (TO AVOID REFERENCING
;NEXT PAR/PDR PAIR)
012514 103041 BHS DONE23 ;EXIT LOOP IF DONE
012516 005761 000002 TST 2(R1) ;REFERENCE OUTSIDE ALLOWED AREA -
;SHOULD ABORT
012522 005077 166276 CLR @SRO ;TURN KT11-D OFF
012526 104006 HLT ;NO ABORT OCCURRED ON A REFERENCE
012530 000426 BR CONT23 ;OUTSIDE THE ALLOWED PAGE LENGTH
012532 042777 000001 166264 RET23A: BIC #1,@SRO ;TURN OFF KT11-D
012540 022626 CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
012542 104006 HLT ;REFERENCE WITHIN ALLOWED AREA
012544 005077 166254 CLR @SRO ;CLEAR ERROR BITS
012550 000416 BR CONT23 ;CAUSED A TRAP OR ABORT
012552 022626 RET23B: CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
012554 017703 166244 MOV @SRO,R3 ;SAVE CURRENT SRO
012560 005077 166240 CLR @SRO ;TURN OFF KT11-D
012564 022703 040003 CMP #40003,R3 ;CK SAVED SRO

```



```

012570 001401      BEQ      .+4
012572 104006      HLT
                                :CONTENTS OF SRO INCORRECT AFTER
                                :PAGE LENGTH ERROR ABORT
                                :CHECK SRO TO BE SURE PL BIT CLEARED

012574 022777 000002 166222  CMP      #2, @SRO
012602 001401      BEQ      .+4
012604 104006      HLT
                                :SRO INCORRECT AFTER CLEARING IT
                                :ONLY KERNEL PAGE 1 SHOULD STILL BE SET
                                :SETUP R1 TO REFERENCE BOUNDARY OF
                                :NEXT PAGE
012606 062701 000100      CONT23: ADD      #100, R1
012612 062702 000400      ADD      #400, R2
                                :ADD 1 TO VALUE TO BE LOADED IN
                                :PAGE LENGTH FIELD
012616 000715      BR       LOOP23
                                :CHECK NEXT PAGE LENGTH VALUE
012620 005077 166200      DONE23: CLR      @SRO
                                :TURN OFF KT11-D
012624 016777 166336 166332  MOV      KTSTA, @KTVEC
                                :RESTORE MEMORY MANAGEMENT ABORT RETURN
012632 005077 166330      CLR      @KTSTA
                                :TO CAUSE HALT ON A FALSE TRAP
                                :OR ABORT

                                :TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN SET)
                                :KERNEL PAR/PDR PAIR1 IS TESTED WITH ALL VALUES OF PAGE LENGTH FIELD
                                :SHOW THAT REFERENCES TO BOTH BOUNDARIES OF THE ALLOWED AREA DON'T TRAP OR ABORT
                                :SHOW THAT A REFERENCE TO THE WORD IMMEDIATELY BELOW THE ALLOWED AREA DOES TRAP
                                TEST47: SCOPE
012636 104400      MOV      #KSTACK, SP
                                :INITIALIZE KERNEL STACK POINTER
012640 012706 001000      JSR      PC, ORDER
                                :CHECK TEST SEQUENCE + INIT SRO
012644 004767 003446      47
                                :TEST NUMBER
012650 000047      HLT
                                :TEST EXECUTED OUT OF SEQUENCE
012652 104006      JSR      %7, RWALL
                                :INITIALIZE ALL PAR/PDR PAIRS TO RW 4K BANK 0
012654 004767 002436      JSR      PC, KERN7
                                :MAP KERNEL PAR/PDR 7 TO EXT BANK
012660 004767 003516      MOV      #77416, R2
                                :R2 CONTAINS VALUE TO BE LOADED IN THE
012664 012702 077416      :PDR BEING CHECKING (INCLUDING PLF)
                                :R1 IS USED TO REFERENCE THE LOWEST
                                :ALLOWED ADDRESS IN THE PAGE
012670 012701 037700      MOV      #37700, R1
                                :SETUP ABORT RETURN IN CASE REFERENCE
                                :WITHIN ALLOWED AREA ABORTS
012674 012777 012754 166262  LOOP24: MOV      #RET24A, @KTVEC
                                :TURN ON KT11-D
012702 005077 166260      CLR      @KTSTA
                                :SET KERNEL PAR/PDR PAIR 1 TO NEW PAGE LENGTH
012706 005277 166112      INC      @SRO
                                :REFERENCE UPPER ALLOWED BOUNDARY
012712 010277 166160      MOV      R2, @KPDR1
                                :REFERENCE LOWER ALLOWED BOUNDARY
012716 005737 037776      TST      @#37776
                                :- NEITHER REFERENCE SHOULD ABORT
012722 005711      TST      @R1
                                :SETUP ABORT RETURN
012724 012777 012766 166232  MOV      #RET24B, @KTVEC
                                :CHECK FOR DONE
012732 020127 020000      CMP      R1, #20000
                                :EXIT LOOP IF DONE
012736 001436      BEQ      DONE24
                                :REFERENCE BELOW ALLOWED AREA -
012740 005761 177776      TST      -2(R1)
                                :SHOULD ABORT
                                :TURN KT11-D OFF
012744 005077 166054      CLR      @SRO
                                :NO ABORT OCCURRED ON A REFERENCE
012750 104006      HLT
                                :OUTSIDE THE ALLOWED PAGE LENGTH
012752 000423      BR       CONT24
                                :TURN OFF KT11-D AND CLEAR
012754 005077 166044      RET24A: CLR      @SRO
                                :ERROR BITS
                                :RESTORE STACK POINTER
012760 022626      CMP      (SP)+, (SP)+
                                :REFERENCE WITHIN ALLOWED AREA CAUSED
012762 104006      HLT
                                :A TRAP OR ABORT
012764 000416      BR       CONT24
                                :RESTORE STACK POINTER
012766 022626      RET24B: CMP      (SP)+, (SP)+
                                :SAVE CURRENT SRO
012770 017703 166030      MOV      @SRO, R3
                                :TURN OFF KT11-D
012774 005077 166024      CLR      @SRO
                                :CK SAVED SRO
013000 022703 040003      CMP      #40003, R3

```







```

013232 006304          ASL      R4
013234 000304          SWAB   R4
013236 020402          CMP    R4,R2          ;PAGE LENGTH
013240 003001          BGT   .+4
013242 104006          HLT
013244 000754          BR    C25          ;REFERENCE WITHIN ALLOWED
                                ;PAGE LENGTH ABORTED-R3 CONTAINS
                                ;VA USED, R1 CONTAINS VALUE
                                ;LOADED INTO THE PDR

013246 016777 165714 165710 DONE25: MOV   KTSTA, @KTVEC
013254 005077 165544          CLR   @SR0

;SHOW THAT THE W BIT DOESN'T SET IF THE KT11-D IS OFF
TEST51: SCOPE
013260 104400          MOV   #KSTACK, SP          ;INITIALIZE KERNEL STACK POINTER
013262 012706 001000          JSR   PC, ORDER          ;CHECK TEST SEQUENCE + INIT SR0
013266 004767 003024          S1
013272 000051          HLT          ;TEST NUMBER
013274 104006          HLT          ;TEST EXECUTED OUT OF SEQUENCE
013276 012767 002000 002264          MOV   #2000, ICOUNT          ;RESTORE ITERATION COUNT
013304 004767 001762          JSR   %7, CLRALL          ;CLEAR ALL KT11-D REGISTERS
013310 013737 010000 010000          MOV   @#10000, @#10000          ;WRITE BANK 0
013316 005777 165552          TST   @KPDR0
013322 001401          BEQ  .+4
013324 104006          HLT          ;W BIT SET OR ANOTHER BIT INCORRECT
                                ;IN KERNEL 0 PDR

;SHOW THAT THE W BIT IS CLEARED BY WRITING (VIA DAT0) THE CORRESPONDING PAR
;CHECK EACH PDR
TEST52: SCOPE
013326 104400          MOV   #KSTACK, SP          ;INITIALIZE KERNEL STACK POINTER
013330 012706 001000          JSR   PC, ORDER          ;CHECK TEST SEQUENCE + INIT SR0
013334 004767 002756          S2
013340 000052          HLT          ;TEST NUMBER
013342 104006          HLT          ;TEST EXECUTED OUT OF SEQUENCE
013344 004767 001746          JSR   %7, RWALL
013350 004767 003026          JSR   PC, KERN7          ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013354 012777 007600 165510          MOV   #7600, @UPAR7          ;MAP USER 7 TO EXTERNAL BANK
013362 012737 140000 177776          MOV   #140000, @#PS          ;SET MODE TO USER
013370 012706 000400          MOV   #USTACK, R6          ;SETUP USER STACK
013374 012700 001144          MOV   #STATAB, R0          ;SET UP KT REG TABLE POINTER
013400 012001          LOP27: MOV   (R0)+, R1          ;R1 CONTAINS ADDRESS OF
                                ;ADDRESS OF CURRENT PDR
013402 012702 017776          MOV   #17776, R2          ;R2 CONTAINS VIRTUAL ADDRESS TO
                                ;REFERENCE DESIRED PAGE
013406 012037 177776          MOV   (R0)+, @#PS          ;SETUP STATUS FOR CURRENT MODE
013412 005277 165406          LOP27A: INC   @SR0          ;TURN ON KT11-D
013416 011212          MOV   (R2), (R2)          ;WRITE
013420 005077 165400          CLR   @SR0          ;TURN OFF KT11-D
013424 004767 000016          JSR   %7, CKWBIT          ;TEST W BIT
013430 062702 020000          ADD   #20000, R2          ;CHANGE VA TO REFERENCE NEXT PAGE
013434 103366          BCC   LOP27A          ;LOOP UNTIL ALL PDR'S HAVE BEEN
                                ;CHECKED IN THE CURRENT MODE

013436 020027 001152          CMP   R0, #STAEND
013442 002756          BLT  LOP27
013444 000416          BR   EXT27
013446 032771 000100 000000 CKWBIT: BIT   #100, @ (R1)          ;CHECK W BIT
013454 001001          BNE  .+4
013456 104006          HLT          ;W BIT DIDN'T SET IN PDR WHOSE

```



```

013460 005071 000020          CLR      320(R1)          ;ADDRESS IS POINTED TO BY R1
                                ;CLEAR W BIT BY WRITING CORRESPONDING
013464 032771 000100 000000    BIT      #100,3(R1)      ;PAR VIA DATO
013472 001401                    BEQ      .+4              ;CHECK W BIT
013474 104006                    HLT
013476 005721                    TST      (R1)+           ;W BIT DIDN'T CLEAR IN PDR WHOSE
013500 000207                    RTS      %7              ;ADDRESS IS POINTED TO BY R1
013502
    
```

EXT27:

:SHOW THAT THE W BIT IS CLEARED BY A DATOB TO THE PDR  
 :CHECK BOTH HIGH AND LOW DATOB'S, ON KERNEL 0

```

TEST53: SCOPE
013502 104400                    MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
013504 012706 001000                JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
013510 004767 002602                53                      ;TEST NUMBER
013514 000053                    HLT                      ;TEST EXECUTED OUT OF SEQUENCE
013516 104006                    JSR      %7,RWALL        ;MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
013520 004767 001572                JSR      PC,KERN7        ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013524 004767 002652                JSR      PC,KERN7
013530 005277 165270                INC      3SR0           ;TURN ON KT11-D
013534 013737 000000 000000        MOV      3#0,3#0        ;WRITE INTO PAGE 0
013542 005077 165256                CLR      3SR0           ;TURN OFF KT11-D
013546 032777 000100 165320        BIT      #100,3KPDR0    ;CHECK W BIT
013554 001001                    BNE      .+4
013556 104006                    HLT                      ;W BIT NOT SET AFTER WRITING PAGE
013560 112777 000106 165306        MOVB     #106,3KPDR0    ;DATOB SHOULD CLEAR W BIT
013566 032777 000100 165300        BIT      #100,3KPDR0
013574 001401                    BEQ      .+4
013576 104006                    HLT                      ;W BIT DIDN'T CLEAR VIA DATOB (LOW)
                                ;TO THE PDR
013600 005277 165220                INC      3SR0           ;TURN ON KT11-D
013604 013737 017776 017776        MOV      3#17776,3#17776 ;WRITE INTO PAGE 0 AGAIN
013612 005077 165206                CLR      3SR0           ;TURN OFF KT11-D
013616 032777 000100 165250        BIT      #100,3KPDR0    ;CHECK W BIT
013624 001001                    BNE      .+4
013626 104006                    HLT                      ;W BIT NOT SET AFTER WRITING PAGE
013630 016701 165240                MOV      KPDR0,R1      ;SETUP R1 TO REFERENCE HIGH BYTE
013634 005201                    INC      R1             ;OF KPDR0
013636 112711 000177                MOVB     #177,3R1       ;DATOB TO HIGH BYTE OF KPDR0
013642 032777 000100 165224        BIT      #100,3KPDR0    ;CHECK W BIT
013650 001401                    BEQ      .+4
013652 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

```

TEST54: SCOPE
013654 104400                    MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
013656 012706 001000                JSR      PC,ORDER        ;CHECK TEST SEQUENCE + INIT SRO
013662 004767 002430                54                      ;TEST NUMBER
013666 000054                    HLT                      ;TEST EXECUTED OUT OF SEQUENCE
013670 104006                    JSR      %7,RWALL        ;MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
013672 004767 001420                JSR      PC,KERN7        ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013676 004767 002500                JSR      PC,KERN7
                                ;EXTERNAL BANK
013702 005277 165116                INC      3SR0           ;TURN ON KT11-D
013706 013737 000000 000000        MOV      3#0,3#0        ;WRITE INTO PAGE 0
013714 005077 165104                CLR      3SR0           ;TURN OFF KT11-D
    
```



```

013720 032777 000100 165146 BIT #100, &KPDRD ;CHECK W BIT
013726 001001 BNE .+4
013730 104006 HLT ;W BIT NOT SET AFTER WRITING PAGE
013732 112777 000000 165154 MOVB #0, &KPARD ;DATOB TO THE PAR
013740 032777 000100 165126 BIT #100, &KPDRD ;CHECK W BIT
013746 001401 BEQ .+4
013750 104006 HLT ;W BIT DIDN'T CLEAR VIA DATOB
; (LOW) TO THE PAR
; TURN ON KT11-D
013752 005277 165046 INC &SRD ;WRITE INTO PAGE 0 AGAIN
013756 013737 017776 017776 MOV &#17776, &#17776 ;TURN OFF KT11-D
013764 005077 165034 CLR &SRD ;CHECK W BIT
013770 032777 000100 165076 BIT #100, &KPDRD
013776 001001 BNE .+4
014000 104006 HLT ;W BIT NOT SET AFTER WRITING PAGE
014002 016701 165106 MOV KPAR0, R1 ;SETUP R1 TO REFERENCE HIGH BYTE
014006 005201 INC R1 ;OF KPARD
014010 112711 000000 MOVB #0, &R1 ;DATOB TO HIGH BYTE OF KPARD
014014 032777 000100 165052 BIT #100, &KPDRD ;CHECK W BIT
014022 001401 BEQ .+4
014024 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
TEST55: SCOPE
014026 104400 MOV #KSTACK, SP ;INITIALIZE KERNEL STACK POINTER
014030 012706 001000 JSR PC, ORDER ;CHECK TEST SEQUENCE + INIT SRC
014034 004767 002256 55 ;TEST NUMBER
014040 000055 HLT ;TEST EXECUTED OUT OF SEQUENCE
014042 104006
014044 012767 000020 001516 MOV #20, ICOUNT
014052 004767 001240 JSR %7, RWALL ;INITIALIZE ALL PAGES RW, 4K, BANK 0
014056 004767 002320 JSR PC, KERN7 ;MAP KERNEL PAR/PDR 7 TO EXT BANK
014062 012777 007600 165002 MOV #7600, &UPAR7 ;MAP USER 7 TO THE EXTERNAL BANK
014070 012737 140000 177776 MOV #140000, &#PS ;SET MODE TO USER
014076 012706 000400 MOV #USTACK, R6 ;SETUP USER STACK
014102 012700 001144 MOV #STATAB, R0 ;RO POINTS TO INFORMATION FOR
; CURRENT MODE
; MOVE POINTER
; SETUP MODE TO REFERENCE NEXT SET OF REGS
; SETUP R2 TO REFERENCE DESIRED PAGE
014106 005720 LOOP32: TST (R0)+
014110 012037 177776 MOV (R0)+, &#PS
014114 012702 017776 MOV #17776, R2
014120 005277 164700 INC &SRD
014124 011212 LOP32C: MOV (R2), (R2) ;WRITE IN
014126 062702 020000 ADD #20000, R2 ;CHANGE VA TO REFERENCE NEXT PAGE
014132 103374 BCC LOP32C ;SET ALL W-BITS IN CURRENT MODE
014134 005077 164664 CLR &SRD ;TURN OFF KT11-D
014140 020027 001152 CMP R0, #STAEND ;CHECK FOR DONE SETTING THE W BITS
014144 002760 BLT LOOP32 ;IF NOT, LOOP TO DO NEXT MODE
014146 012701 001034 MOV #ADRTAB, R1 ;SETUP R1 TO REFERENCE ADDRESSES OF PDR'S OF PDR'S
014152 012702 000010 LOP32D: MOV #10, R2 ;USE R2 AS COUNTER TO CHANGE ADDRESS
; AT END OF EACH SET OF REGISTERS
; CHECK W BIT
014156 032771 000100 000000 LOP32E: BIT #100, &(R1)
014164 001001 BNE .+4
014166 104006 HLT ;W BIT NOT SET IN PDR WHOSE
; ADDRESS IS POINTED TO BY R1-
; SHOULD HAVE BEEN SET WHEN
; PAGE WAS WRITTEN INTO
    
```







;CHECK TO SEE THAT MULTIPLE ACCESSES TO A PAGE AFTER SETTING THE  
 ;W BIT DON'T CLEAR THE W BIT  
 TEST57: SCOPE

014410	104400			MOV	#KSTACK, SP		; INITIALIZE KERNEL STACK POINTER
014412	012706	001000		JSR	PC, ORDER		; CHECK TEST SEQUENCE + INIT SR0
014416	004767	001674		57			; TEST NUMBER
014422	000057			HLT			; TEST EXECUTED OUT OF SEQUENCE
014424	104006			MOV	#10, ICOUNT		
014426	012767	000010	001134	JSR	%7, RWALL		; INITIALIZE ALL PAGES 4K, RW, BANK 0
014434	004767	000656		JSR	PC, KERN7		; MAP KERNEL PAR/PDR 7 TO EXT BANK
014440	004767	001736		MOV	#77406, @KPDR1		; MAP KERNEL PAGE 1 RRW
014444	012777	077406	164424	MOV	#1, @SR0		; TURN ON SEGMENTATION
014452	012777	000001	164344	MOV	@#20000, @#20000		; READ AND WRITE PAGE 1
014460	013737	020000	020000	CMP	#77506, @KPDR1		; CHECK THE PDR
014466	022777	077506	164402	BEQ	.+4		
014474	001401			HLT			; KERNEL PDR1 INCORRECT
014476	104006						; W BIT SHOULD BE SET
014500	012701	020000		MOV	#20000, R1		
014504	012702	000100		MOV	#100, R2		
014510	005721			TST	(R1)+		; READ PAGE 1 REPEATEDLY
014512	077202			SOB	R2, L40		
014514	022777	077506	164354	CMP	#77506, @KPDR1		; CHECK W BIT AGAIN
014522	001401			BEQ	.+4		
014524	104006			HLT			; KERNEL PDR 1
014526	005077	164272		CLR	@SR0		; INCORRECT AFTER REPEATEDLY READING PAGE 1
							; TURN OFF SEGMENTATION

;SHOW THAT IF KT11-D IS ON, SETTING THE CURRENT MODE TO 01 WILL  
 ;CAUSE A MEMORY MANAGEMENT ABORT. NON RESIDENT SHOULD BE SET, AND ALSO PL SHOULD  
 ;BE SET  
 TEST60: SCOPE

014532	104400			MOV	#KSTACK, SP		; INITIALIZE KERNEL STACK POINTER
014534	012706	001000		JSR	PC, ORDER		; CHECK TEST SEQUENCE + INIT SR0
014540	004767	001552		60			; TEST NUMBER
014544	000060			HLT			; TEST EXECUTED OUT OF SEQUENCE
014546	104006			MOV	#2000, ICOUNT		; RESTORE ITERATION COUNT
014550	012767	002000	001012	JSR	%7, RWALL		
014556	004767	000534		JSR	PC, KERN7		; MAP KERNEL PAR/PDR 7 TO EXT BANK
014562	004767	001614		MOV	#RET42, @KTVEC		; SETUP MEMORY MANAGEMENT ABORT RETURN
014566	012777	014634	164370	CLR	@KTSTA		
014574	005077	164366		MOV	#1, @SR0		; TURN ON KT11-D
014600	012777	000001	164216	MOV	#40000, @#PS		; SET MODE TO 01-FETCH OF NEXT
014606	012737	040000	177776	NOP			; INSTRUCTION SHOULD ABORT
014614	000240			CLR	@PS		; RESTORE MODE TO KERNEL
014616	005077	163154		CLR	#1, @SR0		; TURN OFF KT11-D
014622	042777	000001	164174	BIC			; NO ABORT WHEN MODE WAS SET
014630	104006			HLT			; TO 01 (ILLEGAL)
014632	000415			BR	CONT42		; TURN OFF KT11-D AFTER ABORT
014634	042777	000001	164162	BIC	#1, @SR0		; CK SR0
014642	022777	100040	164154	CMP	#100040, @SR0		
014650	001401			BEQ	.+4		
014652	104006			HLT			; SR0 INCORRECT AFTER MODE 01 ABORT
014654	022777	014606	164150	CMP	#ADD42, @SR2		; NR, AND MODE 01 SHOULD BE SET
014662	001401			BEQ	.+4		; CHECK SR2
014664	104006			HLT			; SR2 INCORRECT - SHOULD CONTAIN



```

                                ; ADDRESS OF THE INSTRUCTION
                                ; IMMEDIATELY AFTER THE ONE SETTING
                                ; THE MODE TO 01
                                ; REINITIALIZE SRO
                                ; RESTORE TRAP CATCHER
014666 005077 164132          CONT42: CLR    @SRO
014672 016777 164270 164264  MOV    KTSTA,@KTVEC

; *THIS TEST WAS WRITTEN TO CHECK-OUT ECO #M-7236-00005. IT USES KPAR'S 0
; *AND 2 TO REFERENCE KPAR1 AND UPAR1 RESPECTIVELY. A COUNT PATTERN IS
; *RUN THROUGH THE VIRTUAL ADDRESS STARTING AT BIT6 AND THE RECIPROCAL
; *COUNT PATTERN IS SIMULTANEOUSLY RUN THROUGH THE PAR'S. AFTER A
; *RELOCATED REFERENCE IS MADE THE KT-11 IS TURNED OFF AND THE DATA IS
; *CHECKED TO ENSURE THAT, WHATEVER THE CONDITION OF THE BITS IS IN THE
; *VIRTUAL ADDRESS, THE DECODING FOR USER AND KERNAL PAR'S IS DONE BY
; *THE PHYSICAL ADDRESS.
014700 104400
014702 012706 001000
014706 004767 001404
014712 000061
014714 104006
014716 004767 000374
014722 004767 001454
014726 012777 007723 164160
014734 012777 007776 164156
014742 005000
014744 012701 000042
014750 012702 040042
014754 052777 000400 164042 3$:
014762 012711 005252
014766 005077 164032
014772 027727 164120 005252
015000 001401
015002 104006
                                MOV    #KSTACK,SP          ; INITIALIZE KERNEL STACK POINTER
                                JSR    PC,ORDER            ; CHECK TEST SEQUENCE + INIT SRO
                                61                       ; TEST NUMBER
                                HLT                       ; TEST EXECUTED OUT OF SEQUENCE
                                JSR    %7,RWALL           ; SETUP ALL PDR'S FOR 4K R/W
                                JSR    PC,KERN7           ; SET UP KERNAL 7 REGISTERS
                                MOV    #7723,@KPAR0      ; LOAD KPAR0 WITH ADDR OF KPAR1
                                MOV    #7776,@KPAR2      ; LOAD KPAR2 WITH ADDR OF UPAR1
                                CLR    R0                ; CLEAR COUNTER REGISTER
                                MOV    #42,R1            ; LOAD OFFSET & BIT TO SELECT KPAR0
                                MOV    #40042,R2         ; LOAD OFFSET & BIT TO SELECT KPAR2
                                BIS    #400,@SRO         ; TURN ON MAINTENANCE MODE
                                MOV    #5252,(R1)         ; LOAD PATTERN IN KERNAL PAR1
                                CLR    @SRO              ; TURN OFF MAINTENANCE MODE
                                CMP    @KPAR1,#5252      ; DID DATA GET STORED IN KPAR1?
                                BEQ    1$                ; BRANCH IF DATA STORED CORRECTLY
                                HLT                       ; A HALT HERE INDICATES THAT THE
                                ; RELOCATION TO KPAR1 WAS NOT
                                ; SUCCESSFUL R1 HAS VIRTUAL ADDR AND
                                ; KPAR0 HAS THE BASE.
                                CLR    @KPAR1            ; CLEAR KPAR1 FOR NEXT TEST
                                BIS    #400,@SRO         ; TURN ON MAINTENANCE MODE
                                MOV    #5252,(R2)         ; LOAD PATTERN IN USER PAR1
                                CLR    @SRO              ; TURN OFF MAINTENANCE MODE
                                CMP    @UPAR1,#5252      ; DID DATA GET STORED IN UPAR1?
                                BEQ    2$                ; BRANCH IF DATA STORED CORRECTLY
                                HLT                       ; A HALT HERE INDICATES THAT THE
                                ; RELOCATION TO UPAR1 DID NOT WORK
                                ; R2 HAS THE VIRTUAL ADDR AND KPAR2
                                ; HAS THE BASE
                                CLR    @UPAR1            ; CLEAR UPAR1 FOR NEXT TEST
                                CMP    #10000,R0         ; CHECK TO SEE IF TEST IS DONE
                                BEQ    EOP               ; BRANCH IF TEST IS OVER
                                ADD    #100,R0            ; ADD BIT6 TO COUNTER
                                ADD    #100,R1            ; ADD BIT6 TO KPAR1'S VIRTUAL ADDR
                                ADD    #100,R2            ; ADD BIT6 TO UPAR1'S VIRTUAL ADDR
                                SUB    #1,@KPAR0         ; SUBTRACT BIT1 FROM KPAR1'S BASE
                                SUB    #1,@KPAR2         ; SUBTRACT BIT1 FROM UPAR1'S BASE
                                BR    3$                 ; CONTINUE TEST
                                BEQ    1$
                                HLT
015004 005077 164106          1$: CLR    @KPAR1
015010 052777 000400 164006  BIS    #400,@SRO
015016 012712 005252          MOV    #5252,(R2)
015022 005077 163776          CLR    @SRO
015026 027727 164024 005252  CMP    @UPAR1,#5252
015034 001401
015036 104006
                                BEQ    2$
                                HLT
                                CLR    @UPAR1
                                CMP    #10000,R0
                                BEQ    EOP
                                ADD    #100,R0
                                ADD    #100,R1
                                ADD    #100,R2
                                SUB    #1,@KPAR0
                                SUB    #1,@KPAR2
                                BR    3$
015040 005077 164012          2$: CLR    @UPAR1
015044 022700 010000          CMP    #10000,R0
015050 001415
015052 062700 000100          BEQ    EOP
015056 062701 000100          ADD    #100,R0
015062 062702 000100          ADD    #100,R1
015066 162777 000001 164020  ADD    #100,R2
015074 162777 000001 164016  SUB    #1,@KPAR0
015102 000724
                                SUB    #1,@KPAR2
                                BR    3$

```



```

015104 104400          EOP:   SCOPE
015106 032767 010000 162454   BIT      #BIT12,SR
015114 001003          BNE      1$
015116 012700 015261   MOV      #BELL,RO      ; INHIBIT BELL?
015122 000402          BR       2$            ; BRANCH IF BELL IS INHIBITED
015124 012700 015265   1$:   MOV      #ASTER,RO  ; PUT ADDRESS OF BELL CHARS IN RO
015130 112001          2$:   MOVB     (RO)+,R1    ; PUT ADDRESS OF BELL CHARS IN RO
015132 001405          BEQ      LOGICT        ; CHECK FOR TERMINATOR CODE
015134 010177 163660   3$:   MOV      R1,@TDBR   ; BRANCH IF BYTE IS ZERO
015140 105777 163652   TSTB    @TCSR          ; OUTPUT CHARACTER TO BUFFER
015144 100373          BPL      3$            ; SEE IF STATUS REG GETS SET
015146 013701 000042   LOGICT: MOV     @#42,R1  ; BRANCH UNTIL IT DOES
015152 001405          BEQ      END           ; MONITOR HOOK
015154 000005          RESET
015156 004711          LOGIC: JSR      PC,@R1
015160 000240          NOP
015162 000240          NOP
015164 000240          NOP
015166 000167 164006   END:   JMP      START

; MESSAGE AREA
015172 005015 052113 030461   MTIT:   .ASCII <15><12>'KT11-D LOGIC TEST MAINDEC-11-DBKTA-D'<15><12>'@'
015200 042055 046040 043517
015206 041511 052040 051505
015214 020124 040515 047111
015222 042504 026503 030461
015230 042055 045502 040524
015236 042055 005015 100
015243 015 050012 036503   MPC:   .ASCII <15><12>'PC= @'
015250 040040
015252 020040 051520 020075   MPS:   .ASCII ' PS= @'
015260 100
015261 207 177777 000   BELL:  .ASCIZ <207><377><377>
015265 052 177777 000   ASTER: .ASCIZ /*/<377><377>
015272 015272          .EVEN

; SUBROUTINE TO CLEAR ALL KT11-D REGISTERS (EXCEPT SR1,SR2)
015272 005077 163526   CLRALL: CLR     @SR0
015276 005000          CLR     RO
015300 012701 000040          MOV     #32,R1        ; COUNT OF REGISTERS TO BE CLEARED
015304 005070 001034   CLRLP: CLR     @ADRTAB(RO) ; CLEAR REGISTERS THRU ADDRESS TABLE
015310 005720          TST     (RO)+         ; MOVE POINTER
015312 077104          SOB     R1,CLRLP     ; LOOP TILL DONE
015314 000207          RTS     %7

; SUBROUTINE TO MAKE ALL PAGES RW, BANK 0, 4K, UP
015316 005077 163502   RWALL: CLR     @SR0
015322 012701 001034          MOV     #ADRTAB,R1   ; R1 POINTS TO ADDRESS TABLE
015326 012700 000010   RWL1:  MOV     #10,RO   ; RO IS COUNTER
015332 005071 000020   RWL2:  CLR     @20(R1)  ; CLEAR PAR
015336 012731 077406          MOV     #77406,@(R1)+ ; SET PDR RW, 4K
015342 077005          SOB     RO,RWL2
015344 062701 000020          ADD     #20,R1
015350 020127 001132          CMP     R1,#ADREND  ; POINTER TO NEXT GROUP
    
```



```

015354 002764          BLT   RWL1
015356 000207          RTS   %7

;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
;LOAD THE STARTING ADDRESS OF THE TEST
;YOU WISH TO RUN (THE ADDRESS OF THE TESTX
;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
;OPTIONS AT THE 2ND HALT.
;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
015360 005037 177776    TESTX: CLR   @#PS
015364 012706 001000    MOV   #KSTACK,SP
015370 012737 140000 177776    MOV   #140000,@#PS          ;SETUP USER TRAP
015376 012706 000400    MOV   #USTACK,SP
015402 005037 177776    CLR   @#PS
015406 000000          HALT   ;WAIT FOR STARTING ADDRESS
015410 016767 162154 000036    MOV   SR,RETRNX          ;LOAD STARTING ADDRESS IN RETRNX
015416 062767 000002 000030    ADD   #2,RETRNX          ;ADD 2 TO POINT TO INSTRUCTION AFTER
015424 000000          HALT   ;SET SR OPTIONS
015426 005067 000140    CLR   SCOPEF            ;KEEP COUNT AT ZERO
015432 012767 015444 000134    MOV   #XLOOP,RETURN     ;LOAD SCOPE LOOP RETURN POINTER
015440 000177 000010    JMP   @RETRNX           ;JUMP TO TEST
015444 005067 000122    XLOOP: CLR  SCOPEF       ;KEEP COUNT AT ZERO
015450 000177 000000    JMP   @RETRNX          ;JUMP TO TEST
015454 000000    RETRNX: 0
;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 4000 TIMES
015456 032737 040000 177570    SCOPEC: BIT  #BIT14,@#SR   ;TEST SR FOR SCOPE
015464 001015          BNE   SCOPEB            ;YES SCOPE
015466 032737 004000 177570    BIT  #BIT11,@#SR       ;NO-TEST FOR ITERATION
015474 001020          BNE   SCOPEG           ;INHIBIT ITERATION
015476 026767 000070 000064    CMP   SCOPEF, ICOUNT   ;COMPARE CURRENT COUNT TO MAX NUMBER
015504 100014          BPL   SCOPEG           ;EXIT-DONE
015506 005267 000060    INC   SCOPEF           ;INCREMENT COUNT
015512 012737 000340 177776    MOV   #340,@#PS        ;PREVENT TRAPPING WHILE MOVING STACK
015520 022626          SCOPEB: CMP  (6)+,(6)+   ;REPOSITION STACK
015522 005037 177776    CLR   @#PS
015526 005077 163272    CLR   @SR0
015532 000177 000036    JMP   @RETURN          ;REPEAT TEST
015536 005067 000030    SCOPEG: CLR  SCOPEF    ;CLEAR COUNT
015542 005267 163426    INC   TESTCT           ;STEP TEST COUNTER TO ALLOW CHECKING
;ORDER OF EXECUTION.
;SAVE SCOPE RETURN POINTER
;RETURN INLINE-NEXT TEST
015546 011667 000022          MOV   @%6,RETURN
015552 022626          CMP   (6)+,(6)+
015554 005037 177776    CLR   @#PS
015560 005077 163240    CLR   @SR0
015564 000177 000004    JMP   @RETURN
015570 004000    ICOUNT: 4000          ; ITERATION COUNT
015572 000000    SCOPEF: 0             ; COUNT LOCATION FOR ITERATION LOOP
015574 000000    RETURN: 0            ; ADDRESS OF LAST TEST

;ENTERED WITH SYSTEM TRAP CALL (HLT)
;PRINT OUT THE ERROR PC+2 AND STATUS REGISTER
015576 012767 000340 162172    PRINT: MOV  #340,PS      ;SET PRIORITY TO 7
015604 032737 020000 177570    BIT  #BIT13,@#SR       ;TEST FOR INHIBIT PRINT OUT
015612 001401          BEQ   .+4              ;BRANCH TO PRINT
015614 000430          BR   CK                ;INHIBIT, CHECK FOR HALT
015616 012667 000066          MOV   (6)+,SAVPC      ;PC OF FAILING ROUTINE

```



M04

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 51  
DBKTAD.P11 13-SEP-76 10:28

SEQ 0051

015622	012667	000064		MOV	(6)+, SAVPSR	:PSR OF ERROR CONDITION
015626	024646			CMP	-(6), -(6)	:RESTORE STACK
015630	012767	000200	162140	MOV	#200, PS	
015636	016767	000046	000374	MOV	SAVPC, PTEMP1	:LOAD WITH FAILING PC+2



```

015644 004767 000044      JSR      PC,TYPE
015650 015243      MPC
015652 004767 000116      JSR      PC,PRSHRT      ;PRINT FAILING PC+2
015656 004767 000032      JSR      PC,TYPE
015662 015252      MPS
015664 016767 000022 000346      MOV      SAVPSR,PTEMP1  ;LOAD PROCESSOR STATUS
015672 004767 000130      JSR      PC,PROCT      ;PRINT PROCESSOR STATUS
015676 005767 161666      TST      SR              ;CHECK SR FOR HALT SWITCH
015702 100001      BPL      .+4             ;BRANCH IF NOT SET
015704 000000      HALT
015706 000002      RTI                    ;HALT ON ERROR UP
015710 000000      SAVPC: 0                ;RETURN TO MAIN LINE
015712 000000      SAVPSR: 0

;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE
015714 010067 000052      TYPE:  MOV      %0,SAVR0  ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
015720 011600      MOV      (6),%0         ;SET UP EXIT
015722 062716 000002      ADD      #2,%0
015726 011000      MOV      %0,%0
015730 112067 000034      TYP A:  MOV B      (0)+,TYPDAT ;GET CHARACTER
015734 122767 000100 000026      CMP B      #100,TYPDAT ;CHECK FOR "a" CHARACTER
015742 001003      BNE      TYPB           ;BRANCH IF NOT "a"
015744 016700 000022      MOV      SAVRO,%0      ;RESTORE RO
015750 000207      RTS      PC            ;TERMINATOR CHAR. EXIT
015752 116777 000012 163040      TYP B:  MOV B      TYPDAT,%TDBR ;OUTPUT CHAR TO PRINTER
015760 105777 163032      TST B      %TCSR
015764 100375      BPL      .-4
015766 000760      BR      TYP A
015770 000000      TYPDAT: 0
015772 000000      SAVRO: 0

;SUBROUTINE TO PRINT OUT OCTAL NUMBER
;PRSHRT DELETES LEADING ZEROS
;PROCT PRINTS OUT 6 OCTAL DIGITS
015774 012767 000001 000232      PRSHRT: MOV      #1,PRFLG  ;SET FLAG TO INDICATE SHORT PRINTOUT
016002 005767 000232      TST      PTEMP1        ;CHECK FOR ZERO
016006 001011      BNE      PROCT+4       ;BRANCH IF NOT ZERO
016010 012777 000260 163002      MOV      #260,%TDBR    ;OUTPUT A SINGLE ZERO
016016 105777 162774      TST B      %TCSR
016022 100375      BPL      .-4
016024 000207      RTS      %7
016026 005067 000202      PROCT:  CLR      PRFLG  ;CLEAR FLAG TO INDICATE FULL PRINTOUT
016032 005067 000206      CLR      PTEMP3        ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
016036 005067 000174      CLR      PRFLG         ;INITIALIZE CARRY FLAG FOR ROTATES
016042 012767 000260 000172      MOV      #260,PTEMP2   ;SETUP R3
016050 005767 000164      TST      PTEMP1        ;CHECK BIT 15 OF NUMBER
016054 100002      BPL      .+6
016056 005267 000160      INC      PTEMP2        ;BRANCH IF ZERO
016062 006167 000152      ROL      PTEMP1        ;INCREMENT R3 IF ONE
016066 006167 000146      ROL      PTEMP1        ;ROTATE LEFT MOST OCTAL TO RIGHT END
016072 005567 000140      ADC      PRFLG
016076 005767 000132      P.CK:  TST      PRFLG
016102 001404      BEQ      P.WAIT
016104 026727 000132 000260      CMP      PTEMP2,#260
016112 001410      BEQ      P.CONT
016114 016777 000122 162676      P.WAIT: MOV     PTEMP2,%TDBR ;STORE CARRY
;CHECK FOR SHORT PRINTOUT
;BRANCH IF NOT SET
;CHECK FOR ZERO IF SET
;IF SET, GO TO NEXT CHARACTER
;OUTPUT NEXT CHARACTER

```



```

016122 105777 162670          TSTB   @TCSR           ;WAIT FOR TTY READY
016126 100375                    BPL     -4
016130 005067 000100          CLR     PRSFLG        ;PRINT REST OF NUMBER AFTER A NON-ZERO DIGIT
016134 005267 000104          P.CONT: INC    PTEMP3   ;COUNT
016140 026727 000100 000006  CMP     PTEMP3, #6    ;CHECK FOR DONE
016146 001001                    BNE     P.CNT1        ;BRANCH IF NOT DONE
016150 000207                    RTS
016152 000241          P.CNT1: CLC           ;CLEAR CARRY
016154 005767 000056          TST     PRFLG         ;CHECK FOR PREVIOUS CARRY
016160 001403                    BEQ     +10           ;BRANCH IF PREVIOUSLY ZERO
016162 005067 000050          CLR     PRFLG        ;INITIALIZE FLAG
016166 000261                    SEC
016170 006167 000044          ROL     PTEMP1       ;SET CARRY
016174 006167 000040          ROL     PTEMP1       ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTE
016200 006167 000034          ROL     PTEMP1
016204 005567 000026          ADC     PRFLG        ;STORE CARRY
016210 016767 000024 000024  MOV     PTEMP1, PTEMP2 ;LOAD DATA INTO R3
016216 042767 177770 000016  BIC     #177770, PTEMP2 ;CLEAR ALL BUT LOWEST OCTAL DIGIT
016224 052767 000260 000010  BIS     #260, PTEMP2   ;SET TO ASCII EQUIVALENT
016232 000721                    BR
016234 000000          PRSFLG: 0
016236 000000          PRFLG:  0
016240 000000          PTEMP1: 0
016242 000000          PTEMP2: 0
016244 000000          PTEMP3: 0
                                ;CONTAINS VALUE TO BE OUTPUT
                                ;SCRATCH
                                ;USED TO COUNT CHARACTERS OUTPUT

;EMT HANDLER
;FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES
016246 011667 000032          EMTSRV: MOV    @SP, EPC ;GET CALL
016252 162767 000002 000024  SUB     #2, EPC
016260 017767 000020 000016  MOV    @EPC, EPC
016266 105067 000013          CLRB   EPC+1
016272 062767 016306 000004  ADD    #EMTAB, EPC   ;SAVE OFFSET ONLY
016300 017707 000000          MOV    @EPC, PC     ;POINT TO TABLE OF ADDRESSES
016304 000000          EPC:  0            ;JUMP TO DESIRED ROUTINE
                                PATCH1=EMT+0
                                PATCH2=EMT+2
                                PATCH3=EMT+4
016306 104000          EMTAB: PATCH1
016310 104002          PATCH2
016312 104004          PATCH3
016314 015576          PRINT

;SUBROUTINE TO CHECK TEST SEQUENCE
016316 005037 177776          ORDER: CLR    @#PS    ;CLEAR PROCESSOR STATUS
016322 011667 000052          MOV    (SP), TEMPN   ;GET TEST NUMBER ADDRESS
016326 017767 000046 000044  MOV    @TEMPN, TEMPN ;GET TEST NUMBER
016334 032737 002000 177570  BIT    #BIT10, @#SR
016342 001404          BEQ    ORDERB
016344 016700 000030          MOV    TEMPN, R0
016350 000005          RESET
016352 000000          HALT
016354 026767 162614 000016  ORDERB: CMP    TESTCT, TEMPN ;IS TEST SEQUENCE CORRECT
016362 001403          BEQ    ORDERA       ;YES, CONTINUE
016364 062716 000002          ADD    #2, (SP)
016370 000207          RTS     PC         ;UPDATE FOR ERROR RETURN
    
```



DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 54  
DBKTAD.P11 13-SEP-76 10:28

SEQ 0054

016372 062716 000004  
016376 000207  
016400 000000

ORDERA: ADD #4.(SP)  
RTS PC

:UPDATE FOR GOOD RETURN

016402 012777 007600 162522  
016410 012777 077406 162474  
016416 000207  
017712

:MAP KERNEL PAR/PDR 7 TO EXTERNAL BANK  
KERN7: MOV #7600, @KPAR7  
MOV #77406, @KPDR7  
RTS PC  
. =17712

017712 125252  
000001

DESTAD: 125252  
.END











# F05

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 59  
 DBKTA.D.P11 13-SEP-76 10:28

## CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0057

		1708	1718	1723	1730	1735	1738	1741	1750	1755	1758	1762	1773	1777
		1781	1789	1808	1814	1826	1840	1845	1855	1862	1867	1877	1897	1916
		1925	1931	1937	1943	1953	1962	1980	1999	2002	2013	2023	2030	2033
		2043	2053	2056	2068	2089	2093	2101	2105	2125	2145	2150	2157	2161
		2178	2200	2216	2228	2234	2243	2267	2273	2284	2292	2296	2303	2309
		2317	2326	2330	2337	2343	2352	2376	2395	2417	2426	2434	2443	2452
		2460	2471	2482	2487	2491	2510	2523	2533					
ICOUNT	015570	403*	426*	463*	554*	562*	617*	688*	736*	757*	841*	897*	1077*	1117*
		1203*	1782*	1847*	2179*	2229*	2353*	2418*	2444*	2472*	2633	2649*		
INT25	005256	1158	1166#											
INT35	006732	1445	1448#											
INT36	007520	1577	1584#											
INT36A	007574	1588	1596#											
INT36B	007674	1607	1616#											
INT37	010054	1645	1652#											
INT37A	010130	1656	1664#											
INT37B	010230	1675	1684#											
INT40	010410	1713	1720#											
INT40A	010464	1724	1732#											
INT40B	010564	1743	1752#											
IOT35	006530	1396	1399#											
KERN7	016402	790	901	964	1081	1121	1157	1205	1267	1298	1346	1574	1642	1710
		1793	1830	1857	1879	1918	1984	2016	2045	2070	2127	2181	2245	2286
		2319	2355	2420	2446	2474	2512	2780#						
KPAR0	001114	367#	759*	785*	899*	1079*	1119*	2327*	2338	2513*	2543*			
KPAR1	001116	368#	968*	987*	1006*	1026*	1049*	2521	2527*					
KPAR2	001120	369#	2514*	2544*										
KPAR3	001122	370#												
KPAR4	001124	371#												
KPAR5	001126	372#												
KPAR6	001130	373#												
KPAR7	001132	374#	2780*											
KPDR0	001074	358#	382	734*	760*	787*	900*	962*	1080*	1120*	1156*	1180	1293*	1575*
		1643*	1711*	1791*	1982*	2232	2290	2293*	2294	2301	2304	2307	2324	2328
		2335	2341											
KPDR1	001076	359#	963*	1184	1266*	1576*	1644*	1712*	1794*	1919*	2078*	2135*	2188*	2421*
		2424	2430*	2432	2447*	2450	2458							
KPDR2	001100	360#	2015*	2031										
KPDR3	001102	361#												
KPDR4	001104	362#												
KPDR5	001106	363#												
KPDR6	001110	364#												
KPDR7	001112	365#												
KRET34	006114	1295	2051	2781*										
KSTACK	001000	324#	1304#	1312										
		727	402	415	431	459	491	519	558	586	613	649	684	696
		1339	753	780	829	893	957	1073	1113	1151	1199	1261	1284	1330
		2010	1352	1569	1637	1705	1770	1786	1823	1852	1874	1913	1950	1977
		2507	2040	2065	2122	2175	2225	2240	2281	2314	2349	2414	2440	2468
		2614												
KTSTA	001166	393#	887	888*	1159*	1191*	1192	1269*	1278	1578*	1629*	1630	1646*	1697*
		1698	1714*	1765*	1766	1796*	1816	1832*	1848	1881*	1906	1907*	1945	1986*
		2005	2018*	2034	2047*	2058	2076*	2113	2114*	2133*	2168	2169*	2183*	2220
		2476*	2496											
KTVEC	001164	392#	887*	1158*	1192*	1268*	1278*	1577*	1588*	1607*	1630*	1645*	1656*	1675*
		1698*	1713*	1724*	1743*	1766*	1795*	1816*	1831*	1848*	1880*	1906*	1921*	1932*
		1945*	1985*	2005*	2017*	2034*	2046*	2058*	2075*	2082*	2113*	2132*	2139*	2168*



# G05

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 59  
 DBKTA.D.P11 13-SEP-76 10:28 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0058

K123	001012	2182*	2220*	2475*	2496*
K134	001014	327*	839		
LOGIC	015156	328*	840		
LOGICT	015146	313	2562*		
LOOP10	002210	2555	2559*		
LOOP11	002340	622*	626		
LOOP12	001356	657*	661		
LOOP13	012456	440*	450		
LOOP14	012674	2075*	2111		
LOOP3	001456	2132*	2166		
LOOP3A	001464	467*	482		
LOOP3B	001464	469*	481		
LOOP3C	001470	470*	478		
LOOP4	014106	2361*	2370		
LOOP4A	001546	497*	515		
LOOP4B	001554	500*	513		
LOOP5	011406	501*	510		
LOOP5	001766	1886*	1905		
LOOP5A	001774	566*	582		
LOOP7	002072	568*	581		
LOOP7A	002100	593*	609		
LOOP10A	002216	595*	608		
LOOP10B	002236	624*	625		
LOOP10C	002244	630*	638		
LOOP11A	002346	632*	637		
LOOP11B	002366	659*	660		
LOOP11C	002374	665*	673		
LOOP27	013400	667*	672		
LOOP27A	013412	2250*	2263		
LOOP31A	005460	2255*	2260		
LOOP31B	005464	1211*	1220		
LOOP31C	005516	1212*	1216		
LOOP31D	005530	1222*	1255		
LOOP31E	005560	1225*	1253		
LOOP31F	005564	1232*	1244		
LOOP32C	014124	1233*	1241		
LOOP32D	014152	2265*	2367		
LOOP32E	014156	2272*	2385		
LOOP32F	014226	2274*	2381		
LOOP32G	014232	2391*	2402		
LOOP5A	011464	2393*	2398		
LOOP5A	001644	1899	1901*		
LOOP5B	001662	526*	551		
L25A	013132	532*	544		
L25B	013142	2187*	2206		
L40	014510	2189*	2204		
MPC	015243	2456*	2457		
MPS	015252	2577*	2665		
MTIT	015172	2579*	2668		
NG35B	006506	409	2570*		
NG35C	006716	1353	1392*		
NG35D	007140	1403	1442*		
NG35E	007360	1452	1497*		
NOP	000240	1510	1550*		
ODDAD	007164	268*			
		1496	1501	1505*	











TEST15	002716	752#			
TEST16	003036	779#			
TEST17	003232	828#			
TEST2	001322	430#			
TEST20	003622	892#			
TEST21	004070	956#			
TEST22	004564	1072#			
TEST23	004772	1112#			
TEST24	005166	1150#			
TEST25	005374	1198#			
TEST26	005662	1260#			
TEST27	005766	1283#			
TEST3	001416	458#			
TEST30	006252	1338#			
TEST31	007422	1568#			
TEST32	007756	1636#			
TEST33	010312	1704#			
TEST34	010646	1769#			
TEST35	010726	1785#			
TEST36	011070	1822#			
TEST37	011226	1851#			
TEST4	001520	490#			
TEST40	011330	1873#			
TEST41	011514	1912#			
TEST42	011706	1949#			
TEST43	012000	1976#			
TEST44	012136	2009#			
TEST45	012302	2039#			
TEST46	012414	2064#			
TEST47	012636	2121#			
TEST5	001616	518#			
TEST50	013052	2174#			
TEST51	013260	2224#			
TEST52	013326	2239#			
TEST53	013502	2280#			
TEST54	013654	2313#			
TEST55	014026	2348#			
TEST56	014266	2413#			
TEST57	014410	2439#			
TEST6	001726	557#			
TEST60	014532	2467#			
TEST61	014700	2506#			
TEST7	002040	585#			
TST10	002200	620#	643		
TST10F	002300	618#	639	641*	644#
TST11	002330	655#	678		
TST11F	002430	653#	674	676*	679#
TYPA	015730	2683#	2691		
TYPB	015752	2685#	2688#		
TYPDAT	015770	2683#	2684	2688	2692#
TYPE	015714	408	2664	2667	2679#
UPAR0	001054	349#	804*	1345*	
UPAR1	001056	350#	2531	2537*	
UPAR2	001060	351#			
UPAR3	001062	352#			
UPAR4	001064	353#			







# L05

DBKTA.D MACY11 27(1006) 07-OCT-76 09:10 PAGE 65

DBKTAD.P11 13-SEP-76 10:28

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0063

ABRT	1563*	1564	1632	1700											
TESTNO	399*	414	430	459	490	518	557	585	612	648	683	695	726	752	779
	828	892	956	1072	1112	1150	1198	1260	1283	1338	1563	1636	1704	1769	1785
	1822	1851	1873	1912	1949	1976	2009	2039	2064	2121	2174	2224	2239	2280	2313
	2348	2413	2439	2467	2506										

. ABS. 017714 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

DBKTAD,DBKTAD/SOL/CRF+DBKTAD  
RUN-TIME: 9 19 2 SECONDS  
RUN-TIME RATIO: 56/31=1.7  
CORE USED: BK (16 PAGES)



