

.REMI

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DBKTA-C-0
PRODUCT NAME: KT11-0 BASIC LOGIC TEST
DATE REVISED: SEPTEMBER 20, 1974
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: ROBERT WHITTON

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) 1972,1973,1974, BY DIGITAL EQUIPMENT CORPORATION

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

5.2.3 TRAPCATCHER

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

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

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

5.2.4 EMTSRV (EMT DECODER)

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

5.2.5 CLRALL

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

5.2.6 RWALL

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

5.3 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.

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.

*

H01

DBKTA.C MACY11 27(732) 08-SEP-76 09:35 PAGE 7
DBKTAC.P11

:BASIC LOGIC TEST OF THE K11-0
:COPYRIGHT 1972, 1973, 1974, 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

104400	SCOPE=TRAP
000240	NOP=240
000000	R0=%0
000001	R1=%1
000002	R2=%2
000003	R3=%3
000004	R4=%4
000005	R5=%5
000006	R6=%6
000007	R7=%7
000006	SP=%6
000007	PC=%7
177570	SR=177570
177776	PS=177776
177776	STATUS=%S
104006	HLT=104006
000001	BIT0=1
000002	BIT1=2
000004	BIT2=4
000010	BIT3=10
000020	BIT4=20
000040	BIT5=40
000100	BIT6=100
000200	BIT7=200
000400	BIT8=400
001000	BIT9=1000
002000	BIT10=2000
004000	BIT11=4000
010000	BIT12=10000
020000	BIT13=20000
040000	BIT14=40000
100000	BIT15=100000


```

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

```

;LOAD STARTING AREA
      =200
000200 000200      JMP      START
000210 000167 000774
      =210
000210 000167 015144      JMP      TESTX
  
```

```

;LOAD DATA AREA
      =400
000400 000400      USTACK: 0
001000 000000      =. +376
001002 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
  
```

```

001024 177572      SR0: 177572      ;KT11-D STATUS REGISTER ADDRESSES
001026 177573      SR0H: 177573
001030 177574      SR1: 177574
001032 177576      SR2: 177576
  
```

```

;AORTAB:
001034 177600      UPDR0: 177600      ;USER PAGE DESCRIPTOR REGISTERS
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
  
```

```

;UPAR:
001054 177640      UPAR0: 177640      ;USER PAGE ADDRESS REGISTERS
001056 177642      UPAR1: 177642
001060 177644      UPAR2: 177644
001062 177646      UPAR3: 177646
001064 177650      UPAR4: 177650
001066 177652      UPAR5: 177652
001070 177654      UPAR6: 177654
  
```

```

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      KPAR0: 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      SAVEA: 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
.
001200 005037 177776      ;SET UP FOR START OF BASIC LOGIC TESTS
001204 012706 001000      START: CLR      @#PS
001210 012767 002000 014352      MOV      @KSTACK, SP
001216 012767 001256 014350      MOV      @2000, ICOUNT
001224 012767 000001 177742      MOV      @TEST1+2, RETURN
001232 005767 177734      MOV      @1, TESTCT
001236 001007      TST      FTITLE
001240 004767 014450      BNE     TEST1+2
001244 015172      JSR     PC, TYPE
001246 005267 177720      MTIT
001252 000401      INC     FTITLE
BR      .+4      ;SKP SCOPE INSTRUCTION
.
INITIALIZE STATUS
SETUP KERNEL STACK
INITIALIZE ITERATION COUNT
SETUP SCOPE AND ITERATION LOOP RETURN
INITIALIZE TEST COUNT
DID TITLE PRINT
YES, START TEST
NO, PRINT TITLE

```

K01

DBKTR.C MACY11 27(732) 08-SEP-76 09:35 PAGE 10
 DBKTR.C.P11

```

;SRO AND SRI SHOULD BE INITIALIZED TO 0
TEST1:  SCOPE
001254 104400
001256 012706 001000      MOV      #KSTACK, SP      ;INITIALIZE KERNEL STACK POINTER
001262 004767 015030      JSR      PC, ORDER       ;CHECK TEST SEQUENCE + INIT SRO
001265 000001              1                          ;TEST NUMBER
001270 104006              HLT                       ;TEST EXECUTED OUT OF SEQUENCE
001272 000005              RESET                      ;ISSUE INIT
001274 005777 177524      TST      @SRO             ;CHECK SRO
001300 001401              BEQ      .+4
001302 104006              HLT                       ;SRO WAS NOT INITIALIZED TO ZERO
001304 005777 177520      TST      @SRI             ;CHECK SRI
001310 001401              BEQ      .+4
001312 104006              HLT                       ;SRI WAS NOT INITIALIZED TO ZERO
001314 012767 000010 014246  MOV      #10, ICOUNT      ;DROP ITERATION COUNT SINCE RESET IS USED

;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
001324 012706 001000      MOV      #KSTACK, SP      ;INITIALIZE KERNEL STACK POINTER
001330 004767 014762      JSR      PC, ORDER       ;CHECK TEST SEQUENCE + INIT SRO
001334 000002              2                          ;TEST NUMBER
001336 104006              HLT                       ;TEST EXECUTED OUT OF SEQUENCE
001340 005777 177460      TST      @SRO             ;CHECK SRO INITIALLY
001344 001402              BEQ      .+6
001346 104006              HLT                       ;SRO NOT ZERO AT START OF TEST
001350 000422              BR
001352 012700 000001      MOV      #1, R0           ;R0 CONTAINS BIT INDICATING POSITION BEING TESTE
001356 010001      LOOP2:  MOV      R0, R1
001360 010102              MOV      R1, R2
001362 042701 000401      BIC      #401, R1         ;DON'T SET THE BIT IN SRO IF IT'S BIT 0 OR BIT 8
001366 042702 017777      BIC      #17777, R2      ;CLEAR THE BIT IN R2 IF IT SHOULDN'T SET IN SRO
001372 010177 177426      MOV      R1, @SRO
001376 020277 177422      CMP      R2, @SRO
001402 001401              BEQ      .+4
001404 104006              HLT                       ;SRO INCORRECT WHEN VALUE IN R1
                                ;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
001420 012706 001000      MOV      #KSTACK, SP      ;INITIALIZE KERNEL STACK POINTER
001424 004767 014666      JSR      PC, ORDER       ;CHECK TEST SEQUENCE + INIT SRO
001430 000003              3                          ;TEST NUMBER
001432 104006              HLT                       ;TEST EXECUTED OUT OF SEQUENCE
001434 012767 002000 014126  MOV      #2000, ICOUNT    ;RESTORE ICOUNT
001442 004767 013624      JSR      %7, CLRALL       ;INITIALIZE K11-0 REGISTERS
001446 012703 001140      MOV      #PARTAB, R3     ;R3 POINTS TO TABLE OF PAR ADDRESSES
001452 012700 000002      MOV      #2, R0          ;R0 IS COUNTER OF STATES LEFT TO TEST
001456 012301      LOOP3:  MOV      (R3)+, R1    ;PUT ADDRESS OF 1ST PAR IN SET IN R1
  
```

```

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      #1,R4          ;SET BIT IN PAR
001472 020411          CMP      R4,R4          ;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
001532 000004          HLT                     ;TEST NUMBER
001534 104006          JSR     %7,CLRALL     ;TEST EXECUTED OUT OF SEQUENCE
001536 004767 013530    MOV     #PORTAB,R3    ;INITIALIZE KT11-D REGISTERS
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   ;R5 CONTAINS EXPECTED RESULTING CONTENTS OF PDR
001562 046705 177402    BIC     PDRM2,R5      ;LOAD PDR
001566 010011          MOV     R0,@R1        ;CHECK RESULTING CONTENTS OF PDR
001570 021105          CMP     @R1,R5
001572 001401          BEQ     .+4
001574 104006          HLT                     ;PDR WHOSE ADDRESS IS IN R1
                                     ;WAS INCORRECT AFTER VALUE IN R0
                                     ;WAS LOADED INTO IT
001576 006300          ASL     R0
001600 103367          BCC     LOOP4B        ;ROTATE BIT
001602 005011          CLR     @R1           ;BRANCH IF NOT DONE WITH THIS PDR
001604 005721          TST    (R1)+         ;IF DONE WITH THIS PDR, CLEAR IT
001606 077216          SOB    R2,LOOP4A     ;MOVE POINTER TO ADDRESS NEXT PDR
001610 020327 001136    CMP     R3,#PDREND   ;TEST ALL PDR'S IN THIS GROUP
001614 003754          BLE    LOOP4         ;TEST ALL 2 GROUPS OF PDRS-USER, KERNEL

;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
001630 000005          HLT                     ;TEST NUMBER
001632 104006          JSR     %7,CLRALL     ;TEST EXECUTED OUT OF SEQUENCE
001634 004767 013432    ;CLEAR ALL PAR'S AND PDR'S

```

MO1

DBKTA.C MACY11 27(732) 08-SEP-76 09:35 PAGE 12
 DBKTAC.P11

001640	012701	001034		MOV	#ADRTAB,R1						
001644	012702	001034		LOPSAA: MOV	#ADRTAB,R2						
001650	012703	000040		MOV	#32,R3						
001654	012771	010421	000000	MOV	#10421,@(R1)						
001662	020201			LOPSB: CMP	R2,R1						
001664	001406			BEQ	CONT5						
001666	005772	000000		TST	@(R2)						
001672	001403			BEQ	CONT5						
001674	104006			HLT							
001676	005072	000000		CONT5: CLR	@(R2)						
001702	005722			TST	(R2)+						
001704	077312			SQB	R3,LOPSB						
001706	022701	001132		CMP	#ADREND,R1						
001712	001402			BEQ	DONESA						
001714	005031			CLR	@(R1)+						
001716	000752			BR	LOPSAA						
001720	012767	000100	013642	DONESA: MOV	#100,ICOUNT						
001726	104400			:SHOW THAT BYTE ADDRESSING OF PAR'S WORKS FOR HIGH AND LOW BYTES							
001730	012706	001000		TEST6: SCOPE							
001734	004767	014356		MOV	#KSTACK,SP						
001740	000006			JSR	PC,ORDER						
001742	104006			6							
001744	012767	002000	013616	HLT							
001752	004767	013314		MOV	#2000,ICOUNT						
001756	012703	001140		JSR	X7,CLRAL						
001762	012700	000002		MOV	#PARTAB,R3						
001766	012301			MOV	#2,R0						
001770	012702	000010		LOOP6: MOV	(R3)+,R1						
001774	012711	177777		MOV	#10,R2						
002000	105011			LOOP6A: MOV	#-1,@R1						
002002	022711	007400		CLRB	@R1						
002006	001401			CMP	#7400,@R1						
002010	104006			BEQ	.+4						
002012	012711	177777		HLT							
002016	105061	000001		MOV	#-1,@R1						
002022	022711	000377		CLRB	1(R1)						
002026	001401			CMP	#377,@R1						
002030	104006			BEQ	.+4						
				HLT							

```

:R1 POINTS TO ADDRESS OF LOCATION
:LOADED WITH 1 BIT SET IN EACH 4 BITS
:R2 USED AS A POINTER TO CYCLE THRU
:ALL OTHER ADDRESSES OF PAR/PDR PAIR'S TO
:CHECK FOR DUAL ADDRESSING
:R3 USED AS A COUNTER
:LOAD A PAR OR PDR - SET ONE BIT
:IN EACH CHIP (4 BITS PER CHIP) IF R/W
:SKIP CHECKING THIS ADDRESS TO SEE IF
:IT'S A DUAL, SINCE IT WAS THE ONE LOADED
:OTHERWISE, CHECK TO SEE IF THIS
:REGISTER RESPONDED TO THE ADDRESS
:OF THE ONE LOADED AS A DUAL
:BRANCH IF OK
:DUAL ADDRESSING - ADDRESS POINTED
:TO BY R2 RESPONDED TO THE ADDRESS
:POINTED TO BY R1 IN AT LEAST ONE
:4 BIT SECTION (1 CHIP)
:REINITIALIZE FAULTY LOCATION
:MOVE POINTER R2
:CHECK ALL PAR'S AND PDR'S
:TO SEE IF THEY RESPONDED TO THE
:ADDRESS POINTED TO BY R1
:HAVE ALL ADDRESSES BEEN CHECKED
:FOR DUALS?
:YES - GO TO NEXT TEST
:NO - MOVE POINTER R1
:CHECK TO SEE IF ANY OTHER ADDRESS
:ALSO RESPONDS TO THE ADDRESS POINTED
:TO BY R1
:DROP ITERATION COUNT

:INITIALIZE KERNEL STACK POINTER
:CHECK TEST SEQUENCE + INIT SRO
:TEST NUMBER
:TEST EXECUTED OUT OF SEQUENCE
:RESTORE ITERATION COUNT
:INITIALIZE KT11-0 REGISTERS
:R3 POINTS TO TABLE OF PAR ADDRESSES
:R0 IS COUNTER OF STATES LEFT TO TEST
:PUT ADDRESS OF 1ST PAR IN SET IN R1
:R2 IS COUNTER OF PAR'S LEFT TO TEST IN SET
:SET UP PAR BEING TESTED
:CLEAR LOW BYTE OF PAR
:CHECK PAR
:BRANCH IF OK
:DATOB TO PAR WHOSE ADDRESS IS IN
:R1 FAILED
:SET UP PAR TO TEST HIGH BYTE
:CLEAR HIGH BYTE
:CHECK PAR

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

NO1

002032	005721			TST	(R1)+	; MOVE POINTER
002034	077221			S08	R2, LOOP6A	; TEST ALL PAR'S IN SET
002036	077025			S08	R0, LOOP6	; TEST ALL 2 REGISTER SETS
; SHOW THAT BYTE ADDRESSING OF PDR'S WORKS FOR HIGH AND LOW BYTES						
002040	104400			TEST7:	SCOPE	
002042	012706	001000		MOV	#KSTACK, SP	; INITIALIZE KERNEL STACK POINTER
002046	004767	014244		JSR	PC, ORDER	; CHECK TEST SEQUENCE + INIT SRO
002052	000007			?		; TEST NUMBER
002054	104006			HLT		; TEST EXECUTED OUT OF SEQUENCE
002056	004767	013210		JSR	%7, CLRALL	; INITIALIZE KT11-0 REGISTERS
002062	012703	001134		MOV	#PDRTAB, R3	; R3 POINTS TO TABLE OF PDR ADDRESSES
002066	012700	000002		MOV	#2, R0	; R0 IS COUNTER OF STATES LEFT TO TEST
002072	012301			LOOP7:	MOV (R3)+, R1	; PUT ADDRESS OF 1ST PDR IN SET INTO R1
002074	012702	000010		MOV	#10, R2	; R2 IS COUNTER OF PDR'S LEFT TO TEST IN SET
002100	012711	177777		LOOP7A:	MOV #1, #R1	; SET UP PDR BEING TESTED
002104	105011			CLRB	R1	; CLEAR LOW BYTE OF PDR
002106	022711	077400		CMP	#77400, R1	; CHECK PDR
002112	001401			BEQ	.+4	; BRANCH IF OK
002114	104006			HLT		; DATOB TO PDR WHOSE ADDRESS IS IN R1 FAILED
002116	012711	177777		MOV	#1, #R1	; SET UP PDR TO TEST HIGH BYTE
002122	105061	000001		CLRB	1(R1)	; CLEAR HIGH BYTE
002126	022711	000016		CMP	#16, #R1	; CHECK PDR
002132	001401			BEQ	.+4	
002134	104006			HLT		; DATOB TO HIGH BYTE OF PDR WHOSE ADDRESS IS IN R1 FAILED
002136	005721			TST	(R1)+	; MOVE POINTER
002140	077221			S08	R2, LOOP7A	; TEST ALL PDR'S IN SET
002142	077025			S08	R0, LOOP7	; TEST ALL 2 REGISTER SETS
; INIT SHOULD HAVE NO EFFECT ON PAR'S						
002144	104400			TEST10:	SCOPE	
002146	012706	001000		MOV	#KSTACK, SP	; INITIALIZE KERNEL STACK POINTER
002152	004767	014140		JSR	PC, ORDER	; CHECK TEST SEQUENCE + INIT SRO
002156	000010			?		; TEST NUMBER
002160	104006			HLT		; TEST EXECUTED OUT OF SEQUENCE
002162	012767	000010	013400	MOV	#10, ICOUNT	; DROP ITERATION COUNT
002170	005067	000104		CLR	TST10	
002174	012704	005252		MOV	#5252, R4	
002200	012703	001140		TST10:	MOV #PARTAB, R3	
002204	012700	000002		MOV	#2, R0	
002210	012301			LOOP10:	MOV (R3)+, R1	
002212	012702	000010		MOV	#10, R2	; COUNTER TO LOAD PAR'S
002216	010421			LOP10A:	MOV R4, (R1)+	; LOAD PAR WITH PATTERN
002220	077202			S08	R2, LOP10A	; LOAD ALL 16 IN THIS SET
002222	077006			S08	R0, LOOP10	; INITIALIZE ALL 2 SETS
002224	000005			RESET		; ISSUE INIT
002226	012703	001140		MOV	#PARTAB, R3	
002232	012700	000002		MOV	#2, R0	
002236	012301			LOP10B:	MOV (R3)+, R1	
002240	012702	000010		MOV	#10, R2	; COUNTER TO CHECK PAR'S
002244	020411			LOP10C:	CMP R4, R1	; CHECK DATA
002246	001401			BEQ	.+4	
002250	104006			HLT		; PAR WHOSE ADDRESS IS IN R1 WAS INCORRECT AFTER INIT

002252	005721		TST	(R1)+	: MOVE POINTER
002254	077205		SQB	R2, LOP10C	: TEST ALL 8 PAR'S IN THIS SET
002256	077011		SQB	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 SR0
002314	000011		LI		: 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 #PORTAB, 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		SQB	R2, LOP11A	: LOAD ALL 8 IN THIS SET
002352	077006		SQB	R0, LOOP11	: INITIALIZE ALL 2 SETS
002354	000005		RESET		: ISSUE INIT
002356	012703	001134	MOV	#PORTAB, 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:	CMR R4, R1	: CHECK DATA
002376	001401		BEQ	#+4	
002400	104006		HLT		: PDR WHOSE ADDRESS IS IN R1
: WAS INCORRECT AFTER INIT					
002402	005721		TST	(R1)+	: MOVE POINTER
002404	077205		SQB	R2, LOP11C	: TEST ALL 8 PDR'S IN THIS SET
002406	077011		SQB	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 SR0
002446	000012		LI		: TEST NUMBER
002450	104006		HLT		: TEST EXECUTED OUT OF SEQUENCE
002452	012767	002000	MOV	#2000, ICOUNT	: RESTORE ITERATION COUNT
002460	012777	177777	MOV	#-1, 2SR1	: TRY TO LOAD SRI
002466	005777	176336	TST	2SR1	
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 F-SET CLEARS SR0(B)
;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          X7,CLRALL        ;THIS TEST SHOULDN'T GO THRU ANY PAR/PDR PAIR'S
002650 004767 012416     MOV          @1006,@KPCDR0     ;SO MAKE THEM ALL GIVE NON-RESIDENT
;AND PAGE LENGTH ERRORS IF ACCESSED
;3 BLOCKS OF KERNEL PCRD MUST BE MAPPED
;TO ALLOW TRAPS AND ABORTS
002654 012777 001006 176212     MOV          @10,ICOUNT        ;DROP THE ITERATION COUNT
002662 012767 000010 012700     MOV          @400,@SR0        ;TURN ON DESTINATION ONLY RELOCATION
002670 012777 000400 176126     RESET
002676 000005          ;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 B 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 JSRO

: SHOW THAT DESTINATION ONLY RELOCATION DOESN'T RELOCATE THE SOURCE
: ADDRESS AND DOES RELOCATE THE DESTINATION

TEST15: SCOPE

002716 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

MOV BKSTACK, SP
 JSR PC, ORDER
 IS
 HLT
 MOV #10, ICOUNT
 JSR X7, CLRALL
 MOV #1, AKPAR0
 MOV #77406, AKPDR0
 MOV #DATA16, R1
 MOV #400, JSRO
 CMP R1, R1
 BNE .+4
 HALT
 RESET
 MOV #DATA16-100, R1
 MOV #DATA16, R2
 MOV #400, JSRO
 CMP R2, R1
 BEQ .+4
 HALT

: INITIALIZE KERNEL STACK POINTER
 : CHECK TEST SEQUENCE + INIT SRO
 : TEST NUMBER
 : TEST EXECUTED OUT OF SEQUENCE
 : KEEP THE NUMBER OF LOOPS DOWN
 : OFFSET KERNEL PAR/PDR PAIR 0 ONE BLOCK FROM BANK
 : 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 LOCAL
 : TURN OFF DESTINATION-ONLY RELOCATION
 : LOAD DESTINATION ADDRESS MINUS RELOCATION FACTOR
 : 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 RELOCATION

003030 000005
 003032 000401
 003034 132465

RESET
 BR .+4

DATA16: 132465

: SHOW THAT A DATO OF 0 TO BIT 8, SRO THRU KERNEL PAGE 7 WILL
: CLEAR THE DESTINATION ONLY RELOCATION BIT AND TURN OFF DESTINATION ONLY RELOCATION

TEST16: SCOPE

003036 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

MOV BKSTACK, SP
 JSR PC, ORDER
 IS
 HLT
 JSR X7, CLRALL
 MOV #1, AKPAR0
 MOV #77406, AKPDR0
 MOV #DATA16, R1
 JSR PC, KERN7
 MOV SRO, R2
 MOV #400, JSRO
 CLR R2
 CMP R1, R1
 BEQ .+4
 HALT
 BIT #400, JSRO
 BEQ .+6
 HALT
 RESET

: INITIALIZE KERNEL STACK POINTER
 : CHECK TEST SEQUENCE + INIT SRO
 : TEST NUMBER
 : TEST EXECUTED OUT OF SEQUENCE
 : INITIALIZE
 : MAP KERNEL PAR/PDR PAIR 0
 : TO BANK 0 OFFSET BY 1 PAGE
 : USED TO PROVE KT11-0 IS
 : TURNED OFF AFTER CLEARING BIT 8, SRO
 : SETUP R1 TO REFERENCE KERNEL PAR/PDR PAIR 0
 : MAP KERNEL PAR/PDR 7 TO EXT BANK
 : SETUP R2 TO ADDRESS SRO
 : TURN ON DESTINATION ONLY RELOCATION
 : CLEAR SRO THRU KERNEL PAR/PDR PAIR 7
 : SHOW THAT KT11-0 IS OFF
 : KT11-0 STILL ON
 : SHOW THAT BIT 8, SRO IS NOW ZERO
 : DESTINATION ONLY RELOCATION BIT IS STILL ON
 : MAKE SURE THAT KT11-0 IS OFF

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

003142 004767 012124

JSR X7, CLRALL

: INITIALLY CLEAR ALL PAR/PDR PAIRS

```
003146 012777 000001 175700
003154 012777 077406 175652
003162 012701 013034
003166 012777 077406 175676
003174 012777 077406 175650
003202 016702 175616
003206 012737 140000 177776
003214 012777 000400 175602
003222 005012
003224 021111
003226 001401
003230 000777
```

```
MOV #1,PAR0
MOV #77406,APDR0
MOV @DATA16,R1
MOV #77406,APDR7
MOV #77406,APDR7
MOV SRO,R2
MOV #140000,APPS
MOV #400,ASAO
CLR #22
CMP #R1,AR1
BEQ .+4
BR .
```

```
;MAP USER 0 TO
;BANK 0 OFFSET BY 1 PAGE, RW
;SETUP R1 TO REFERENCE USER 0
;MAP USER 7 TO THE
;EXTEND BANK
;SETUP R2 TO ADDRESS SRO
;SET MODE TO L R
;TURN ON DESTINATION - ONLY PAGING
;CLEAR SRO THRU USER ASR7
;SHOW THAT KTI1-0 IS OFF

;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
```

```
RO - 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
003234 012706 001000
003240 004767 013052
003244 000017
003246 104006
003250 004767 012016
003254 004767 012036
003260 013767 017700 175672
003266 013767 017776 175666
003274 012737 123456 017700
003302 012737 134567 017776
003310 012703 001012
003314 012704 001014
003320 012767 000100 012242
003326 012737 140000 177776
003334 012706 000400
003340 005037 177776
003344 012767 001144 175602
003350 017700 175576
003356 062700 000020
003362 062767 000002 175564
003370 017737 175650 177776
003376 062767 000002 175550
003404 012767 000010 175544
003412 012770 007600 000016
003420 016705 175400
003424 005001
003426 012702 000076
003432 012767 000200 175536
003440 012770 000177 000000
003446 022767 000001 175502
```

TEST17: SCOPE

```
MOV #KSTACK,SP
JSR PC,ORDER
HLT
JSR X7,CLRALL
JSR X7,RWALL
MOV @#17700,SAVEA
MOV @#17776,SAVEB
MOV #123456,@#17700
MOV #134567,@#17776
MOV #K123,R3
MOV #K134,R4
MOV #100,ICOUNT
MOV #140000,APPS
MOV #USTACK,SP
CLR APPS
MOV #STATAB,STAPNT
MOV #STAPNT,RO
ADD #20,RO
ADD #2,STAPNT
MOV #STAPNT,APPS
ADD #2,STAPNT
MOV #8,PAGES
MOV #7600,@16(RO)
MOV SRO,RS
CLR R1
MOV #76,R2
MOV #128,BLOCKS
MOV #177,@(RU)
CMP #1,PAGES
```

STAT20:

PAG20:

```
;INITIALIZE KERNEL STACK POINTER
;CHECK TEST SEQUENCE + INIT SRO
;TEST NUMBER
;TEST EXECUTED OUT OF SEQUENCE
;INITIALIZE
;MAKE ALL PAR/PDR PAIR'S RW, BANK 0, WK
;SAVE CONTENTS OF LOCATIONS TO BE USED

;SET UP LOCATIONS TO BE REFERENCED

;CHANGE ITERATION COUNT
;CHANGE TO USER
;SET UP USER STACK POINTER
;RETURN TO KERNEL
;SET UP TO REFERENCE STATE TABLE
;PICK UP ADDRESS OF START OF
;ADDRESS TABLE FOR NEW STATE

;SET UP NEW STATE

;SET UP COUNTER OF ASR'S LEFT TO TEST
;SET UP SEGMENTED REFERENCE TO SRO
;USED TO TURN DESTINATION - ONLY PAGING OFF

;SET UP BLOCK COUNT
;SET UP PAR
;IS THIS PAGE 7? (WAS USED
;FOR REFERENCE TO SRO)
```


003752	122737	165432	003654	CMPB	#165432, #A021B-100	:COMPARE THE CONTENTS OF A021B
003760	003754			A021B=.	-4	:WITH ITSELF, RELOCATED THRU KERNEL 0
003762	001401			BEQ	.+4	
	104006			HLT		:DESTINATION - ONLY RELOCATION
003764	012737	077711	003756	MOV	#77711, #DST21A-100	:FAILED TO RELOCATE OR Y THE FINAL
003772	005077	000066		CLR	#A021C	:CALCULATION OF THE CLR B INSTRUCTION
003776	105037	003762		CLRB	#DST21C-100	:EXECUTE REMAINING INSTRUCTIONS
004002	005011			CLR	#R1	:TURN OFF KT11-0
004004	022767	077711	000044	CMP	#77711, DST21A	:CHECK LOCATION ADDRESSED BY MOV
004012	001401			BEQ	.+4	
004014	104006			HLT		:MOV INSTRUCTION FAILED TO RELOCATE
004016	005767	000036		TST	DST21B	:ONLY THE FINAL ADDRESS CALCULATION
004022	001401			BEQ	.+4	:CHECK LOCATION ADDRESSED BY CLR
004024	104006			HLT		:CLR INSTRUCTION FAILED TO RELOCATE
004026	022767	177400	000026	CMP	#177400, DST21C	:CORRECTLY IN DESTINATION - ONLY RELOCATION
004034	001401			BEQ	.+4	:CHECK LOCATION ADDRESSED BY CLRB
004036	104006			HLT		:CLRB INSTRUCTION FAILED TO RELOCATE
004040	012667	177716		MOV	(SP)+, DST21C-100	:CORRECTLY IN DESTINATION - ONLY RELOCATION
004044	012667	177710		MOV	(SP)+, DST21B-100	:RESTORE LOCATIONS IN CASE OF ERROR
004050	012667	177702		MOV	(SP)+, DST21A-100	
004054	000404			BR	EXIT21	
004056	000000					
004060	000000					
004062	000000					
004064	003760					
004066	000240					

DST21A: 0
DST21B: 0
DST21C: 0
A021C: 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 SRD
004102	000021			Z1		:TEST NUMBER
004104	104006			HLT		:TEST EXECUTED OUT OF SEQUENCE
004106	004767	011160		JSR	#7, CLRALL	:CLEAR ALL KT11-0 REGISTERS
004112	012777	077406	174754	MOV	#77406, #KPOR0	:MAP KERNEL 0 TO BANK 0, 4K, RW
004120	012777	077406	174750	MOV	#77406, #KPOR1	:MAKE KERNEL 1 4K, RW
004126	004767	012250		JSR	PC, KERN7	:MAP KERNEL PAR/POR 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, #KPAR1	:SET PAR TO -1
004140	012737	030000	177776	MOV	#30000, #PS	:SET UP LOCATION TO BE REFERENCED
004146	012777	000400	174650	MOV	#400, #SR0	:TURN ON DESTINATION - ONLY PAGING
004154	122737	000060	020077	CMPB	#60, #20077	:CHECK HIGH BYTE OF RESULTING ADDRESS

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 SRO SET. SRO AND SR2 ARE CHECKED FOR
: THE CORRECT VALUES, AS ARE KPDR0 AND KPDR1
: SHOW THAT BIT 15 OF SRO CAN BE CLEARED
: SHOW THAT SR2 IS READ ONLY

005166 104006
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, UNDER
24
HLT
JSR %7, CLRALL
MOV #77406, %KPDR0
JSR PC, KERN7
MOV #INT25, %KTVEC
CLR %KTSTA
MOV #20000, R4
INC %SRO
ADR25: TST (R4)+
ADR25A: HALT
CLR %SRO
BR DON25
INT25: MOV %SRO, R1
DEC %SRO
CMP #100003, R1
BEQ .+4
HLT

: INITIALIZE KERNEL STACK POINTER
: CHECK TEST SEQUENCE + INIT SRO
: 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 SRO
: TURN OFF KT11-D
: CHECK SAVED CONTENTS OF SRO

: SRO 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

: 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

DON25:

CLR %KTSTA
MOV %KTSTA, %KTVEC

: 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 DATO) WILL CLEAR THE W BIT

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


```

;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
005376 012706 001000      MOV      #KSTACK, SP
005402 004767 010710      JSR      PC, ORDER
005406 000025
005410 104006      HLT
005412 012767 000400 010150      MOV      #400, ICOUNT
005420 004767 007672      JSR      X7, R4ALL
005424 004767 010752      JSR      PC, KERN7
005430 012777 007600 173434      MOV      #7600, @UPAR7
005436 012737 140000 177776      MOV      #140000, @#PS
005444 012706 000400      MOV      #USTACK, R6
005450 005037 177776      CLR      @#PS
005454 012704 001034      MOV      #ADRTAB, R4
005460 012705 000010      LOP31A: MOV      #10, R5
005464 022734 077406      LOP31B: CMP      #77406, @ (R4)+
005470 001401      BEQ
005472 104006      HLT

005474 077505      SOB      R5, LOP31B
005476 062704 000020      ADD      #20, R4
005502 020427 001132      CMP      R4, #ADREND
005506 003001      BGT
005510 000763      BR
005512 012700 001144      CNT31A: MOV      #STATAB, R0
005516 012001      LOP31C: MOV      (R0)+, R1
005520 012702 017776      MOV      #17776, R2
005524 012037 177776      MOV      (R0)+, @#PS
005530 005277 173270      LOP31D: INC      @SRO
005534 011212      MOV      (R2), (R2)
005536 005077 173262      CLR      @SRO
005542 032771 000100 000000      BIT      #100, @ (R1)
005544 001001      BNE
005546 104006      HLT
005554 012703 001034      MOV      #ADRTAB, R3
005560 012704 000010      LOP31E: MOV      #10, R4
005564 020103      LOP31F: CMP      R1, R3
005566 001405      BEQ
005570 032773 000100 000000      BIT      #100, @ (R3)
005576 001401      BEQ
005600 104006      HLT

005602 005723      CNT31B: TST      (R3)+
005604 077411      SOB      R4, LOP31F
005606 062703 000020      ADD      #20, R3
005612 020327 001132      CMP      R3, #ADREND
005616 002760      BLT
005620 012771 077406 000000      MOV      #77406, @ (R1)
005626 032771 000100 000000      BIT      #100, @ (R1)
005634 001401      BEQ
005636 104006      HLT

005640 005721      TST      (R1)+

```

```

; INITIALIZE KERNEL STACK POINTER
; CHECK TEST SEQUENCE + INIT SRO
; TEST NUMBER
; TEST EXECUTED OUT OF SEQUENCE
; LOAD ITERATION COUNT
; MAP ALL PAR/PDR PAIR'S 4K, BANK 0, R4
; MAP KERNEL PAR/PDR 7 TO EXT BANK
; MAP USER 7 TO EXTERNAL BANK
; SET MODE TO USER
; SET UP USER STACK
; REINITIALIZE STATUS TO KERNEL MODE
; LOAD R4 WITH ADDRESS OF ADR TABLE
; INIT COUNTER OF PDR'S LEFT TO CHECK
; CHECK ALL PDR W BITS CLEAR
; PDR INCORRECT - W BIT SET OR ANOTHER
; BIT INCORRECT IN PDR WHOSE ADDRESS
; IS IN THE LOCATION POINTED TO BY R4
; MOVE POINTER TO FIRST ADR OF NEXT SET
; BRANCH IF DONE
; SET UP START OF STATE TABLE
; R1 CONTAINS ADDRESS OF PDR OF ADDRESS
; SET UP VIRTUAL ADDRESS TO BE REFERENCED
; SET UP STATUS FOR CURRENT MODE
; TURN ON KT11-0
; REFERENCE PAGE TO SET W BIT
; TURN OFF KT11-0
; CHECK W BIT
; W BIT NOT SET IN PDR AFTER PAGE WRITTEN
; SET UP ADDRESS OF ADDRESS TABLE
; NOW CHECK ALL PDR TO SHOW NO OTHER
; W BITS WERE SET
; 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
; TEST NEW PDR
; UPDATE POINTER TO NEXT SET
; CLEAR W BIT VIA DATO TO PDR
; CHECK W BIT
; W BIT DIDN'T CLEAR WHEN PDR
; WAS WRITTEN (ADDRESS OF ADDRESS
; OF PDR IS IN R1)
; UPDATE POINTER

```

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,#STAREND		;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	104706			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			HLT			;NO NR ABORT
005742	022777	140003	173054	RET33: CMP	#140003,ASRO		;CHECK SRO
005750	001401			BEQ	.+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	104706			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	000000		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,AKPDR0		
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,AKPDR7		;MAP USER PAGE 7
006076	012777	007600	172766	MOV	#7600,AKPAR7		;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	ASR1		;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		


```

007314 000000
007316 000000
007320 000000
007322 000000
007324 000000
007326 000000
007328 000000
007330 000000
007332 000000
007334 000000
007336 000000
007338 000000
007340 000000
007342 000000
007344 000000
007346 000000
007348 000000
007350 000000
007352 000000
007354 005737 000001          TST      201          ; 000 ADDRESS REFERENCE - AN "INTERNAL
                                ; TRAP" SHOULD OCCUR
                                ; RESTORE STACK POINTER
                                ; TURN OFF KT11-0
                                ; 000 ADDRESS TRAP DIDN'T TAKE
                                ; VECTOR FROM KERNEL

007360 022626          NG35E:  CMP      (SP)+,(SP)+
007362 005011          CLR      2R1
007364 104006          HLT

007366 000407          BR      END35
007370 022626          OK35E:  CMP      (SP)+,(SP)+
007372 005011          CLR      2R1
007374 032737 000340 177776  BIT      2340,20PS
007402 001401          BEQ     .+4
007404 104006          HLT
007406 012737 000006 000004  END35:  MOV      26,204
007414 012737 000106 000104  MOV      2106,20104

```

```

; SHOW THAT THE ABORT LOGIC "LOCKS" SR0 AND SR2 AFTER A NR
; ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0, WHEN
; THEY RESUME TRACKING. A NR ERROR SHOULD STILL ABORT TO 250 EVEN
; WHEN BIT 15 (SR0) IS ALREADY SET

```

```

007422 104400          TEST31: SCOPE
007424 012706 001000  MOV      2KSTACK,SP
007430 004767 006662  JSR      PC,ORDER
007434 000031 31          JSR      31
007436 104006          HLT
007440 004767 000006  JSR      X7,CLRALL
007444 004767 000002  JSR      PC,KERN7
007450 012777 017406 171416  MOV      277406,2KPDRO
007456 012777 077400 171412  MOV      277400,2KPDRI
007464 012777 007400 171472  MOV      2INT36,2KTVEC
007472 000000 171470  CLR      2KTSTA
007476 000000 171322  INC      2SR0
007502 013737 037776 037776  ADR36:  MOV      2037776,2037776
007510 005077 171310  CLR      2SR0
007514 104006          HLT
007516 003510          BR      DONE36
007520 042777 000001 171276  INT36:  BIC      21,2SR0
007526 022777 100002 171270  CMP      2100002,2SR0
007534 001401          BEQ     .+4
007536 104006          HLT

```

```

; INITIALIZE KERNEL STACK POINTER
; CHECK TEST SEQUENCE + INIT SR0
; TEST NR
; TEST EXECUTED OUT OF SEQUENCE
; CLEAR ALL KT11-0 REGISTERS
; P KE L P/POR 7 TO EXT BANK
; P KE L 0 RW,RK,BANK0
; P KE L 1 NR,4 K,BANK0
; SETUP RETURN VECTOR

; TURN ON KT11-0
; REFERENCE KERNEL 1 - 1ST ABORT
; TURN OFF KT11-0
; REFERENCE TO KERNEL 1
; DIDN'T ABORT
; TURN OFF KT11-0
; CHECK SR0

; SR0 INCORRECT AFTER NR ABORT

```

F03

DBKTRAC.P11 27(732) 08-SEP-76 09:35 PAGE 31

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,@SR0	; TURN ON KT11-0
007562	012242			MOV	(R2)+,-(R2)	; REFERENCE KERNEL 1 -2ND ABORT
007564	005077	171234		ADR36A: CLR	@SR0	; TURN OFF KT11-0
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 KT11-0
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
007652	012702	037776		MOV	#37776,R2	; SETUP R2 TO REFERENCE KERNEL 1
007656	005277	171142		INC	@SR0	; TURN ON KT11-0
007662	012242			ADR36B: MOV	(R2)+,-(R2)	; 3RD NR REFERENCE, ERROR BIT WAS CLEARED
007664	005077	171134		ADR36C: CLR	@SR0	; TURN OFF KT11-0
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 KT11-0
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
007726	022716	007664		CMP	#ADR36C,(SP)	; CHECK STACK
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 (SK0) IS ALREADY SET
007756	104400			TEST32: SCOPE		
007760	012706	001000		MOV	#KSTACK,SP	; INITIALIZE KERNEL STACK POINTER
007764	004767	006326		JSR	PC,ORDER	; CHECK TEST SEQUENCE + INIT SR0
007770	000032			32		; TEST NUMBER
007772	104006			HLT		; TEST EXECUTED OUT OF SEQUENCE
007774	004767	005272		JSR	%7,CLRALL	; CLEAR ALL KT11-0 REGISTERS
010000	004767	006376		JSR	PC,KERN7	; MAP KERNEL PAR/POR 7 TO EXT BANK
010004	012777	077406	171062	MOV	#77406,@KPOR0	; MAP KERNEL 0 RW,RX,BANK0

010012	012777	017406	171056	MOV	#17406, #KPDRI	;MAP KERNEL 1 PL 1 K, BANK0
010020	012777	010054	171136	MOV	#INT37, #KTVEC	;SETUP RETURN VECTOR
010026	005077	171134		CLR	#KTSTA	
010032	005277	170766		INC	#R0	;TURN ON KT11-0
010036	013737	037776	037776	ADR37: MOV	#37776, #37776	;REFERENCE KERNEL 1 - 1ST ABORT
010044	005077	170754		CLR	#SR0	;TURN OFF KT11-0
010050	104006			HLT		;REFERENCE TO KERNEL 1
010052	000510			BR	DONE37	;DIDN'T ABORT
010054	042777	000001	170742	INT37: BIC	#1, #R0	;TURN OFF KT11-0
010062	022777	040002	170734	INT37: 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 KT11-0
010116	012242			MOV	(R2)+, -(R2)	;REFERENCE KERNEL 1 -2ND ABORT
010120	005077	170700		ADR37A: CLR	#SR0	;TURN OFF KT11-0
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 KT11-0
010136	022777	040002	170660	INT37A: 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
010206	012702	037776		MOV	#37776, R2	;SETUP R2 TO REFERENCE KERNEL 1
010212	005277	170606		INC	#SR0	;TURN ON KT11-0
010216	012242			ADR37B: MOV	(R2)+, -(R2)	;3RD PL REFERENCE, ERROR BIT WAS CLEARED
010220	005077	170600		ADR37C: CLR	#SR0	;TURN OFF KT11-0
010224	104006			HLT		;3RD REFERENCE TO KERNEL 1
010226	000422			BR	DONE37	;DIDN'T ABORT
010230	042777	000001	170566	INT37B: BIC	#1, #SR0	;TURN OFF KT11-0
010236	022777	040002	170560	INT37B: CMP	#40002, #SR0	;CHECK SR0
010244	001401			BEQ	.+4	
010246	104006			HLT		;SR0 INCORRECT
010250	022777	010216	170554	CMP	#ADR37B, #SR2	;CHECK SR2
010256	001401			BEQ	.+4	
010260	104006			HLT		;SR2 INCORRECT - SHOULD CONTAIN LAST FETCH ADDRESS BEFORE ABORT
010262	022716	010220		CMP	#ADR37C, (SP)	;CHECK STACK
010266	001401			BEQ	.+4	
010270	104006			HLT		;PC ON STACK INCORRECT
010272	022626			CMP	(R6)+, (R6)+	;RESTORE STACK POINTER
010274	005077	170524		DONE37: CLR	#SR0	;CLEAR ERROR BIT
010300	005077	170662		CLR	#KTSTA	;CHANGE TRAP RETURN TO CAUSE A HALT
010304	016777	170656	170652	MOV	KTSTA, #KTVEC	;ON A FALSE INTERRUPT


```

010604 022777 010552 170220      CMP      #ADR408,2SR2      ;CHECK SR2
010612 001401                      BEQ      .+4
010614 104006                      HLT
                                ;SR2 INCORRECT - SHOULD CONTAIN
                                ;LAST FETCH ADDRESS BEFORE ABORT
                                ;CHECK STACK
010616 022716 010554              CMP      #ADR40C,(SP)
010622 001401                      BEQ      .+4
010624 104006                      HLT
                                ;PC ON STACK INCORRECT
010626 022626                      CMP      (R6)+,(R6)+      ;RESTORE STACK POINTER
010630 010377 170170      DONE40: CLR      27,30          ;CLEAR E. R BIT
010634 010377 170326      CLR      2KTSTA          ;CHANGE TAMP RETURN TO CAUSE A HALT
010640 016777 170322 170316      MOV      KTSTA,2KTVEC    ;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
010658 010334                      HLT
                                ;TEST EXECUTED OUT OF SEQUENCE
010662 104006                      HLT
010664 112777 000340 170134      MOV      #340,2SR0H      ;SET SRO BITS 13-15
010672 122777 000340 170126      CMP      #340,2SR0H      ;MAKE SURE THEY SET CORRECTLY
010700 001401                      BEQ      .+4
010702 104006                      HLT
                                ;SRO INCORRECT (HIGH BYTE)
010704 010305                      RESET
                                ;ISSUE INIT
010706 122777 000000 170112      CMP      #0,2SR0H        ;CHECK SRO HIGH BYTE
010714 001401                      BEQ      .+4
010716 104006                      HLT
                                ;SRO INCORRECT AFTER INIT
010720 012767 000010 004642      MOV      #10,ICOUNT     ;DROP ITERATION COUNT

                                ;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                      HLT
010744 004767 004346          JSR      X7,RWALL        ;MAP ALL PAR/PDR PAIR'S 4K,RW BANK 0
010750 012777 000416 170116      MOV      #416,2KPDR0    ;MAP KERNEL 0 RW,4K LESS 1 PAGE
                                ;DOWN (100-17776 RW)
                                ;MAP KERNEL PAR/PDR 7 TO EXT BANK
010756 004767 005420          JSR      PC,KERN7
010762 012777 077400 170106      MOV      #77400,2KPDR1  ;MAP KERNEL PAGE 1 NR
010770 012777 011024 170166      MOV      #RET2,2KTVEC   ;SETUP ABORT RETURN
010776 005077 170164          CLR      2KTSTA
011002 012746 000020          MOV      #20,-(SP)      ;SET T BIT IN STATUS ON STACK
011006 012746 011020          MOV      #ADR2,-(SP)   ;SETUP ADDRESS ON STACK
011012 005277 170006          INC      2SR0          ;TURN ON KT11-D
011016 000002                      RTI
                                ;SHOULD TRACE TRAP IMMEDIATELY SINCE T-BIT
                                ;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
011024 022777 040001 167772      RET2:  CMP      #40001,2SRO  ;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
011036 000005                      RESET

```

```

011040 005777 167760          TST      @SR0          ;CHECK SR0
011044 001401          BEQ      .+4
011046 104006          HLT
011050 005077 167750          CLR      @SR0          ;SR0 INCORRECT AFTER INIT
011054 016777 170106 170102  DONE2:  MOV     @KTSTA,@KTVEC ;REINITIALIZE SR0
011062 012737 000016 000014  MOV     @16,@14        ;RESTORE T-BIT TRAP CATCHER

;SHOW THAT INIT CLEARS SR0(0-3,5-6)
;REFERENCE NR USER PAGE 7 TO SET ALL BITS(0-6)
;THEN ISSUE INIT
TEST36: SCOPE
011070 104400          MOV     @KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
011072 012706 001000          JSR     PC,ORDER        ;CHECK TEST SEQUENCE + INIT SR0
011076 004767 005214          36                    ;TEST NUMBER
011102 000036          HLT                    ;TEST EXECUTED OUT OF SEQUENCE
011104 104006          JSR     %7,RWALL        ;MAP ALL PAR/PDR PAIR'S INITIALLY RW,4K,
011106 004767 004204          BANK 0
;MAKE USER 7 NR
011112 012777 077400 167732  MOV     @77400,@UPDR7  ;MAP KERNEL PAR/PDR 7 TO EXT BANK
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,@APS   ;SET MODE TO USER
011144 012706 000400          MOV     @USTACK,R6    ;SETUP USER STACK IN CASE NEEDED
011150 005277 167650          INC     @SR0          ;TURN ON KT11-D
011154 005737 160000          TST     @160000       ;REFERENCE PAGE 7
011160 000777          BR
011162 022777 100157 167634  RET3:  CMP     @100157,@SR0 ;NO ABORT ON NR REFERENCE
011170 001401          BEQ     .+4           ;CHECK SR0
011172 104006          HLT                    ;SR0 INCORRECT - SHOULD HAVE TRACKED
;NR REFERENCE TO USER 7
011174 000005          RESET
011176 005777 167622          TST     @SR0          ;CHECK SR0
011202 001401          BEQ     .+4
011204 104006          HLT                    ;SR0 INCORRECT AFTER INIT
011206 005077 167612          CLR     @SR0
011212 012767 000010 004350  MOV     @10,ICOUNT    ;DROP ITERATION COUNT
011220 016777 167742 167736  MOV     @KTSTA,@KTVEC

;SHOW THAT BYTE ADDRESSING OF SR0 WORKS
TEST37: SCOPE
011226 104400          MOV     @KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
011230 012706 001000          JSR     PC,ORDER        ;CHECK TEST SEQUENCE + INIT SR0
011234 004767 005056          37                    ;TEST NUMBER
011240 000037          HLT                    ;TEST EXECUTED OUT OF SEQUENCE
011242 104006          JSR     %7,RWALL        ;MAP ALL PAR/PDR PAIRS RW 4K,BANK 0
011244 004767 004046          PC,KERN7              ;MAP KERNEL PAR/PDR 7 TO EXT BANK
011250 004767 005126          MOV     @160001,@SR0  ;TURN ON KT11-D AND SET ERROR FLAGS
011254 012777 160001 167542  CLR@   @SR0          ;DATOB (LOW) TO SR0
011262 105077 167536          BIT     @160000,@SR0  ;CHECK SR0
011266 032777 160000 167530  BNE     .+4
011274 001001          HLT                    ;SR0 INCORRECT AFTER DATOB
011276 104006          MOV     @160001,@SR0
011300 012777 160001 167516  CLR@   @SR0H         ;DATOB (HIGH) TO SR0
011306 105077 167514          CMP     @17,@SR0      ;CHECK SR0
011312 022777 000017 167504  BEQ     .+4
011320 001401          HLT                    ;SR0 INCORRECT AFTER DATOB
011322 104006

```



```

011574 005077 167224          CLR  @SR0          ;TURN OFF KT11-D
011600 104006          HLT              ;NR REFERENCE DIDN'T ABORT
011602 000436          BR              ;
011604 017701 167214  RET7A: MOV  @SR0,R1      ;SAVE SR0 CONTENTS IN R1
011610 005077 167210          CLR  @SR0          ;TURN OFF KT11-D
011614 022701 100003          CMP  @100003,R1   ;CHECK SAVED CONTENTS OF SR0
011620 001401          BEQ  .+4          ;
011622 104006          HLT              ;SR0 INCORRECT SHOULD SHOW NR ERR, KERNEL PAGE 1
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  @SR0          ;TURN ON KT11-D
011644 005737 020000          TST  @#20000      ;REFERENCE USER PAGE 1 (NR)
011650 005077 167150          CLR  @SR0          ;TURN OFF KT11-D
011654 104006          HLT              ;NR REFERENCE DIDN'T ABORT
011656 000410          BR              ;
011660 017701 167140  RET7C: MOV  @SR0,R1      ;SAVE CONTENTS OF SR0
011664 005077 167134          CLR  @SR0          ;TURN OFF KT11-D
011670 022701 100143          CMP  @100143,R1   ;CHECK SAVED CONTENTS OF SR0
011674 001401          BEQ  .+4          ;
011676 104006          HLT              ;SR0 INCORRECT - SHOULD SHOW NR
011700 016777 167262 167256  DONE7: MOV  KTSTA,@KTVEC ;ERROR USER PAGE 1
;RESTORE TRAP CATCHER
;SHOW THAT SR0 <1-3,5-6> DOESN'T TRACK IF KT11-D IS OFF BUT DOES IF REFERENCE IS TO
;AN INTERNAL (KT11-D) REGISTER
011706 104400          TEST42: SCOPE
011710 012706 001000          MOV  #KSTACK,SP   ;INITIALIZE KERNEL STACK POINTER
011714 004767 004376          JSR  PC,ORDER      ;CHECK TEST SEQUENCE + INIT SR0
011720 000042          42              ;TEST NUMBER
011722 104006          HLT              ;TEST EXECUTED OUT OF SEQUENCE
011724 004767 003366          JSR  %7,RWALL      ;SET ALL PAR/PDR PAIRS RW, 4K, BANK 0
011730 012777 007600 167134  MOV  @7600,@PAR7   ;MAP USER 7 TO THE EXT. BANK
011736 012737 140000 177776  MOV  @140000,@#PS  ;SET MODE TO USER
011744 005277 167054          INC  @SR0          ;TURN ON KT11-D
011750 042777 000001 167046  BIC  @1,@SR0       ;TURN OFF KT11-D
011756 005037 177776          CLR  @#PS          ;CHANGE TO KERNEL MODE
011762 022777 000156 167034  CMP  @156,@SR0    ;CHECK SR0
011770 001401          BEQ  .+4          ;
011772 104006          HLT              ;SR0 INCORRECT - SHOULD SHOW REFERENCE
;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  @SR0
;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.
012000 104400          TEST43: SCOPE
012002 012706 001000          MOV  #KSTACK,SP   ;INITIALIZE KERNEL STACK POINTER
012006 004767 004304          JSR  PC,ORDER      ;CHECK TEST SEQUENCE + INIT SR0
012012 000043          43              ;TEST NUMBER

```

M03

DBKTA.C MACY11 27(732) 08-SEP-76 09:35 PAGE 38
 DBKTAC.P11

```

012014 104006          HLT          ;TEST EXECUTED OUT OF SEQUENCE
012016 004767 003274  JSR          %7,RWALL  ;INITIALIZE ALL PAGES RW,4K,BANK 0
012022 012777 000416 167044  MOV      #416,@KPDR0 ;MAP KERNEL TO EXCLUDE
                                ;LOCATIONS 0 TO 77
012030 004767 004346  JSR          PC,KERN7  ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012034 012777 012072 167122  MOV      @RET11,@KTVEC ;SETUP MEMORY MANAGEMENT ABORT RETURN
012042 005077 167120  CLR          @KTSTA
012046 012746 000020  MOV      #20,-(SP)    ;PREPARE STACK TO TURN ON T-BIT
012052 012746 012060  MOV      @.+6,-(SP)
012056 000006  RTT
012060 012777 000001 166736  ADR11:  MOV      #1,@SRO
                                ;SET T-BIT VIA RTT
                                ;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          CONT11
012072 042777 000001 166724  RET11:  BIC      #1,@SRO
                                ;TURN OFF KT11-D
012100 022777 040000 166716  CMP      #40000,@SRO ;CK SRO
012106 001401  BEQ      .+4
012110 104006  HLT
012112 022777 012060 166712  CMP      @ADR11,@SR2 ;SRO 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      @SRO
                                ;REINITIALIZE SRO
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
012136 104400  TEST44: SCOPE
012140 012706 001000  MOV      #KSTACK,SP ;INITIALIZE KERNEL STACK POINTER
012144 004767 004146  JSR      PC,ORDER   ;CHECK TEST SEQUENCE + INIT SRO
012150 000044  44          ;TEST NUMBER
012152 104006  HLT          ;TEST EXECUTED OUT OF SEQUENCE
012154 004767 003136  JSR      %7,RWALL  ;INITIALIZE ALL PAR/PDR PAIRS TO RW,4K, BANK 0
012160 012777 077402 166712  MOV      #77402,@KPDR2 ;SET KERNEL PAR/PDR PAIR 2 RRO,4K
012166 004767 004210  JSR      PC,KERN7  ;MAP KERNEL PAR/PDR 7 TO EXT BANK
012172 012777 012234 166764  MOV      @RET13A,@KTVEC ;SETUP MEMORY MANAGEMENT ABORT RETURN
012200 005077 166762  CLR      @KTSTA
012204 005277 166614  INC      @SRO
                                ;TURN ON KT11-D
012210 012777 160001 166606  MOV      #160001,@SRO ;SET ABORT ERROR BITS
012216 013737 007000 047000  MOV      @#7000,@#47000 ;WRITE KERNEL PAR/PDR PAIR 2 (RRO)-SHOULD TRAP
012224 005077 166574  CLR      @SRO
                                ;NO TRAP OCCURRED
012230 104006  HLT
012232 000416  BR          DONE13
012234 022626  RET13A:  CMP      (SP)+,(SP)+ ;RESTORE THE STACK POINTER
012236 017701 166562  MOV      @SRO,R1    ;SAVE CONTENTS OF SRO
012242 005077 166556  CLR      @SRO
                                ;TURN OFF KT11-D
012246 022701 160017  CMP      #160017,R1
012252 001401  BEQ      .+4
012254 104006  HLT
                                ;SAVED CONTENTS OF SRO INCORRECT
012256 022777 077402 166614  CMP      #77402,@KPDR2 ;CHECK THE PDR CORRESPONDING TO THE TRAP REFERENCE
012264 001401  BEQ      .+4
012266 104006  HLT
                                ;THE PDR CORRESPONDING TO THE TRAP REFERENCE IS INCORREC
012270 016777 166672 166666  DONE13: MOV      KTSTA,@KTVEC ;RESTORE MEMORY MANAGEMENT TRAP RETURN
                                ;TO CAUSE A HALT ON A FALSE TRAP OR ABORT

```

```

012276 005077 166522          CLR      @SR0          ;REINITIALIZE 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          JSR      %7, RWALL          ; MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
012320 004767 002772      JSR      PC, KERN7          ; MAP KERNEL PAR/PDR 7 TO EXT BANK
012324 004767 004052          MOV      #RET16, @KTVEC      ; SETUP TRAP RETURN IN CASE
012330 012777 012372 166626      CLR      @KTSTA
012336 005077 166624          INC      @SR0          ; TURN ON KT11-D
012342 005277 166456          TST     @SR0          ; TRAP REFERENCE TO A KT11-D REGISTER
012346 005777 166452          CLR      @SR0
012352 005077 166446          CMP     #77406, @KPDR7
012356 022777 077406 166526      BEQ     .+4
012364 001401          HLT
012366 104006          BR      DONE16
012370 000004          RET16: BIC     #1, @SR0
012372 000001 166424          HLT
012400 100006          DONE16: CLR     @SR0
012402 000007 166416          MOV     KTSTA, @KTVEC
012406 016777 166554 166550

; TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN NOT SET)
; KERNEL PAR/PDR PAIR 1 IS USED WITH ALL PAGE LENGTH VALUES
; SHOW THAT REFERENCES TO BOTH BOUNDARIES OF THE ALLOWED AREA DON'T TRAP OR ABORT
; SHOW THAT A REFERENCE TO THE FIRST WORD BEYOND THE ALLOWABLE AREA DOES TRAP
TEST46: SCOPE
012414 104400          MOV      #KSTACK, SP          ; INITIALIZE KERNEL STACK POINTER
012416 012706 001000      JSR      PC, ORDER          ; CHECK TEST SEQUENCE + INIT SR0
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 (?) BE LOADED IN THE
012442 012702 000006          MOV     #20076, R1        ; PDR BEING CHECKED (INCLUDING PLF)
                                ; R1 IS USED TO REFERENCE THE TOP ADDRESS
                                ; WITHIN THE ALLOWED AREA
012446 012701 020076          MOV     #RET23A, @KTVEC    ; SETUP ABORT RETURN IN CASE REFERENCE
                                ; WITHIN ALLOWED AREA ABORTS
012452 012777 012532 166504      CLR     @KTSTA          ; TURN ON KT11-D
012460 005077 166502          INC     @SR0          ; SET KERNEL PAR/PDR PAIR 1 TO NEW PAGE LENGTH
012464 005277 166334          MOV     R2, @KPDR1        ; READ LOWER BOUNDARY-SHOULDN'T ABORT
                                ; READ UPPER ALLOWED BOUNDARY-SHOULDN'T
012470 010277 166402          TST     #20000
                                ; ABORT
012474 005727 020000          TST     @R1          ; SETUP ABORT RETURN
                                ; CHECK FOR DONE (TO AVOID REFERENCING
012502 012777 012552 166454      MOV     #RET23B, @KTVEC    ; NEXT PAR/PDR PAIR)
012510 020127 037776          CMP     R1, #37776        ; EXIT LOOP IF DONE
                                ; REFERENCE OUTSIDE ALLOWED AREA -
                                ; SHOULD ABORT
012514 103041          BHIS   DONE23          ; TURN KT11-D OFF
012516 005761 000002          TST     2(R1)          ; NO ABORT OCCURRED ON A REFERENCE
                                ; OUTSIDE THE ALLOWED PAGE LENGTH
012522 005077 166276          CLR     @SR0          ; TURN OFF KT11-D
012526 104006          HLT
012530 000426          BR      CONT23
012532 042777 000001 166264      RET23A: BIC     #1, @SR0

```


012540	022626			CMP	(SP)+,(SP)+	:RESTORE STACK POINTER
012542	104006			HLT		:REFERENCE WITHIN ALLOWED AREA
012544	005077	166254		CLR	2SR0	:CLEAR ERROR BITS
012546	000416			BR	CONT23	:CAUSED A TRAP OR ABORT
012548	022626			RET238:	CMP	(SP)+,(SP)+
012550	017703	166244		MOV	2SR0,R3	:RESTORE STACK POINTER
012552	005077	166240		CLR	2SR0	:SAVE CURRENT SR0
012554	022703	040003		CMP	#40003,R3	:TURN OFF KT11-D
012570	001401			BEQ	.+4	:CK SAVED SR0
012572	104006			HLT		:CONTENTS OF SR0 (INCORRECT AFTER
012574	022777	000002	166222	CMP	#2,2SR0	:PAGE LENGTH ERROR ACRAT
012602	001401			BEQ	.+4	:CHECK SR0 TO BE SURE PL BIT CLEARED
012604	104006			HLT		:SR0 INCORRECT AFTER CLEARING IT
012606	062701	000100		CONT23:	ADD	#100,R1
012612	062702	000400		ADD	#400,R2	:ONLY KERNEL PAGE 1 SHOULD STILL BE SET
012616	000722			BR	LOOP23	:SETUP R1 TO REFERENCE BOUNDARY OF
012620	005077	166200		CLR	2SR0	:NEXT PAGE
012624	016777	166336	166332	MOV	KTSTA,2KTVEC	:ADD 1 TO VALUE TO BE LOADED IN
012632	005077	166330		CLR	2KTSTA	:PAGE LENGTH FIELD
						:CHECK NEXT PAGE LENGTH VALUE
						:TURN OFF KT11-D
						:RESTORE MEMORY MANAGEMENT ABORT RETURN
						:TO CAUSE HALT ON A FALSE TRAP
						:OR ABORT
						:TEST PAGE LENGTH ERROR CHECKING (EXPAND DOWN SET)
						:KERNEL PAR/PDR PAIR 1 IS TESTED WITH ALL VALUES OF PAGE LENGTH FIELD
						:SHOW THAT REFERENCES TO BOTH BOUNDARIES OF THE ALLOWED AREA DON'T TRAP OR ABORT
						:SHOW THAT A REFERENCE TO THE WORD IMMEDIATELY BELOW THE ALLOWED AREA DOES TRAP
012636	104400			TEST47:	SCOPE	
012640	012706	001000		MOV	#KSTACK,SP	:INITIALIZE KERNEL STACK POINTER
012644	004767	003446		JSR	PC,ORDER	:CHECK TEST SEQUENCE + INIT SR0
012650	000047			47		:TEST NUMBER
012652	104006			HLT		:TEST EXECUTED OUT OF SEQUENCE
012654	004767	002436		JSR	X7,RWALL	:INITIALIZE ALL PAR/PDR PAIRS TO RW,WK, BANK 0
012660	004767	003516		JSR	PC,KERN7	:MAP KERNEL PAR/PDR 7 TO EXT BANK
012664	012702	077416		MOV	#77416,R2	:R2 CONTAINS VALUE TO BE LOADED IN THE
012670	012701	037700		MOV	#37700,R1	:PDR BEING CHECKING (INCLUDING PLF)
012674	012777	012754	166262	MOV	#RET24A,2KTVEC	:R1 IS USED TO REFERENCE THE LOWEST
012702	005077	166260		CLR	2KTSTA	:ALLOWED ADDRESS IN THE PAGE
012706	005277	166112		INC	2SR0	:SETUP ABORT RETURN IN CASE REFERENCE
012712	010277	166160		MOV	R2,2KPDR1	:WITHIN ALLOWED AREA ABORTS
012716	005727	037776		TST	#37776	:TURN ON KT11-D
012722	005711			TST	R1	:SET KERNEL PAR/PDR PAIR 1 TO NEW PAGE LENGTH
012724	012777	012766	166232	MOV	#RET24B,2KTVEC	:REFERENCE UPPER ALLOWED BOUNDARY
012732	020127	020000		CMP	R1,#20000	:REFERENCE LOWER ALLOWED BOUNDARY
012736	001436			BEQ	DONE24	:NEITHER REFERENCE SHOULD ABORT
012740	005761	177776		TST	-2(R1)	:SETUP ABORT RETURN
012744	005077	166054		CLR	2SR0	:CHECK FOR DONE
012750	104006			HLT		:EXIT LOOP IF DONE
012752	000423			BR	CONT24	:REFERENCE BELOW ALLOWED AREA -
012754	005077	166044		RET24A:	CLR	:SHOULD ABORT
						:TURN KT11-D OFF
						:NO ABORT OCCURRED ON A REFERENCE
						:OUTSIDE THE ALLOWED PAGE LENGTH
						:TURN OFF KT11-D AND CLEAR


```

013206 003755 BLE L258
013210 062701 000400 R00 #400,R1
013214 100346 BRPL L25A
013216 000413 BRP DONE25
013220 022526 RET25: CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
013222 000302 SLEB R2 ;CHECK TO MAKE SURE VIRTUAL
013224 042702 177400 BIC #177400,R2 ;ADDRESS WAS OUTSIDE ALLOWED
013230 006304 RSL R2
013232 006304 RSL R2
013234 002304 SLEB R2
013236 020402 CMP R2,R2 ;PAGE LENGTH
013240 003001 BGT .+4
013242 104006 HLT ;REFERENCE WITHIN ALLOWED
013244 000754 BR C25 ;PAGE LENGTH ABORTED-R3 CONTAINS
;VA USED, R1 CONTAINS VALUE
;LOADED INTO THE PDR

013246 016777 165714 165710 DONE23: MOV KTSTA,#KTVEC
013254 005077 165544 CLR #SR0

;SHOW THAT THE W BIT DOESN'T SET IF THE KT11-D IS OFF
TESTS1: 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 ;TEST NUMBER
013272 000051 HLT ;TEST EXECUTED OUT OF SEQUENCE
013274 104006 MOV #2000,ICOUNT ;RESTORE ITERATION COUNT
013276 012767 002000 002264 JSR X7,CLALL ;CLEAR ALL KT11-D REGISTERS
013304 004767 001762 JSR #10000,#10000 ;WRITE BANK 0
013310 013737 010000 010000 MOV #KPD0R
013316 005777 165552 TST .+4
013322 001401 BEQ
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
TESTS2: 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 ;TEST NUMBER
013340 000052 HLT ;TEST EXECUTED OUT OF SEQUENCE
013342 104006 JSR X7,R10LL ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013344 004767 001746 JSR PC,KLW7 ;MAP USER 7 TO EXTERNAL BANK
013350 004767 003026 MOV #7600,#UPAR7 ;SET MODE TO USER
013354 012777 007600 1655:0 MOV #14000,#APS ;SETUP USER STACK
013362 012737 140000 177776 MOV #USTACK,#6 ;SET UP KT REG TABLE POINTER
013370 012706 000400 MOV #STATA#R0 ;R1 CONTAINS ADDRESS OF
013374 012700 001144 LOP27: MOV (R0)+,R1 ;ADDRESS OF CURRENT PDR
013402 012702 017776 MOV #17776,R2 ;R2 CONTAINS VIRTUAL ADDRESS TO
;REFERENCE DESIRED PAGE
013406 012037 177776 LOP27A: MOV (R0)+,#APS ;SETUP STATUS FOR CURRENT MODE
013412 005277 165406 JSR #SR0 ;TURN ON KT11-D
013416 011212 MOV (R2),(R2) ;WRITE
013420 005077 165400 CLR #SR0 ;TURN OFF KT11-D
013424 004767 000016 JSR X7,CKWBIT ;TEST W BIT
013430 062702 020000 ADD #20000,R2 ;CHANGE VA TO REFERENCE NEXT PAGE

```

E04

DBKTA.C MACY11 27(732) 08-SEP-76 09:35 PAGE 43
 DBKTAC.P11

```

013434 103366          BCC      LOP27A          ;LOOP UNTIL ALL PDR'S HAVE BEEN
;CHECKED IN THE CURRENT MODE
013436 020027 001152    CMP      R0, #STREND
013442 002756          BLT      LOP27
013444 000416          BR       EXT27
013446 032771 000100 000000 CKMBIT: BIT    #100,2(R1)      ;CHECK W BIT
013454 001001          BNE     .+4
013456 104006          HLT
;W BIT DIDN'T SET IN PDR WHOSE
;ADDRESS IS POINTED TO BY R1
013460 005071 000020          CLR      220(R1)      ;CLEAR W BIT BY WRITING CORRESPONDING
;PAR VIA DATOB
013464 032771 000100 000000    BIT    #100,2(R1)      ;CHECK W BIT
013472 001401          BEQ     .+4
013474 104006          HLT
;W BIT DIDN'T CLEAR IN PDR WHOSE
013476 005721          TST     (R1)+         ;ADDRESS IS POINTED TO BY R1
013500 000207          RTS      X7
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    S3
013514 000053          HLT
;TEST NUMBER
013516 104006          HLT
;TEST EXECUTED OUT OF SEQUENCE
013520 004767 001572    JSR     X7, #MALL        ;MAP ALL PAR/PDR PAIRS 4K, RW, BANK 0
013524 004767 002652    JSR     PC, KERN7        ;MAP KERNEL PAR/PDR 7 TO EXT BANK
013530 005277 165270    INC     2SR0
;TURN ON KT11-0
013534 013737 000000 000000    MOV     2#0, 2#0
;WRITE INTO PAGE 0
013542 005077 165256    CLR     2SR0
;TURN OFF KT11-0
013546 032777 000100 165320    BIT    #100, 2KPDRO
;CHECK W BIT
013554 001001          BNE     .+4
013556 104006          HLT
;W BIT NOT SET AFTER WRITING PAGE
013560 112777 000106 165306    MOVB   #106, 2KPDRO
;DATOB SHOULD CLEAR W BIT
013566 032777 000100 165300    BIT    #100, 2KPDRO
013574 001401          BEQ     .+4
013576 104006          HLT
;W BIT DIDN'T CLEAR VIA DATOB (LOW)
;TO THE PDR
013600 005277 165220          INC     2SR0
;TURN ON KT11-0
013604 013737 017776 017776    MOV     2#17776, 2#17776 ;WRITE INTO PAGE 0 AGAIN
013612 005077 165206    CLR     2SR0
;TURN OFF KT11-0
013616 032777 000100 165250    BIT    #100, 2KPDRO
;CHECK W BIT
013624 001001          BNE     .+4
013626 104006          HLT
;W BIT NOT SET AFTER WRITING PAGE
013630 016701 165240          MOV     KPDRO, R1
;SETUP R1 TO REFERENCE HIGH BYTE
013634 005201          INC     R1
;OF KPDRO
013636 112711 000177          MOVB   #177, 2R1
;DATOB TO HIGH BYTE OF KPDRO
013642 032777 000100 165224    BIT    #100, 2KPDRO
;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

```



```

014634 042777 000001 164162 RET42: BIC      #1,SR0      ;TURN OFF KT11-D AFTER ABORT
014642 022777 100040 164154      CMP      #100040,SR0 ;CK SR0
014650 001401      BEQ      .+4
014652 104006      HLT
                                ;SI INCORRECT AFTER MODE 01 ABORT
                                ;NO MODE 01 SHOULD BE SET
                                ;CI .X SR2
014654 022777 014606 164150      CMP      #A0042,SR2
014662 001401      BEQ      .+4
014664 104006      HLT
                                ;SR2 INCORRECT - SHOULD CONTAIN
                                ;ADDRESS OF THE INSTRUCTION
                                ;IMMEDIATELY AFTER THE ONE SETTING
                                ;THE MODE TO 01
                                ;REINITIALIZE SR0
                                ;RESTORE TRAP CATCHER

014666 005077 164132 CONT42: CLR      SR0
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 P1'S. AFTER A
;RELOCATED REFERENCE IS MADE THE KT-11 IS TURNED OFF AND THE DATA IS
;CHECKED TO FIND OUT THAT WHATEVER THE CONDITION OF THE BITS IS IN THE
;VIRTUAL ADDRESS, THE DECODING FOR USER AND KERNEL PAR'S IS DONE BY
;THE PHYSICAL ADDRESS.
TEST61: SCOPE
014700 104400      MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
014702 012706 001000      JSR      PC,ORDER      ;CHECK TEST SEQUENCE + INIT SR0
014706 004767 001404      B1
014712 000061      HLT      ;TEST NUMBER
014714 104006      HLT      ;TEST EXECUTED OUT OF SEQUENCE
014716 004767 000374      JSR      X7,FULL
014722 004767 001454      JSR      PC,KERN7
014726 012777 007723 164160      MOV      #7723,2KPAR0 ;SETUP ALL PDR'S FOR 4K R/W
014734 012777 007776 164156      MOV      #7776,2KPAR2 ;SET UP KERNEL 7 REGISTERS
014742 005000      CLR      R0      ;LOAD KPAR0 WITH ADDR OF KPAR1
014744 012701 000042      MOV      #42,R1      ;LOAD KPAR2 WITH ADDR OF UPAR1
014750 012702 040042      MOV      #40042,R2 ;CLEAR COUNTER REGISTER
014754 052777 000400 164042 3S:  BIS      #400,SR0 ;LOAD OFFSET & BIT TO SELECT KPAR0
014762 012711 0J5252      MOV      #5252,(R1) ;LOAD OFFSET & BIT TO SELECT KPAR2
014766 005077 164032      CLR      R0 ;TURN ON MAINTENANCE MODE
014772 027727 164120 005252      CMP      2KPAR1,#5252 ;LOAD PATTERN IN KERNEL PAR1
015000 001401      BEQ      IS ;TURN OFF MAINTENANCE MODE
015002 104006      HLT ;DID DATA GET STORED IN KPAR1?
                                ;BRANCH IF DATA STORED CORRECTLY
                                ;A HALT HERE INDICATES THAT THE
                                ;RELOCATION TO KPAR1 WAS NOT
                                ;SUCCESSFUL R1 HAS VIRTUAL ADDR AND
                                ;KPAR0 HAS THE BASE.
015004 005077 164106 1S:      CLR      2KPAR1 ;CLEAR KPAR1 FOR NEXT TEST
015010 052777 000400 164006      BIS      #400,SR0 ;TURN ON MAINTENANCE MODE
015016 012712 005252      MOV      #5252,(R2) ;LOAD PATTERN IN USER PAR1
015022 005077 163776      CLR      SR0 ;TURN OFF MAINTENANCE MODE
015026 027727 164024 005252      CMP      2UPAR1,#5252 ;DID DATA GET STORED IN UPAR1?
015034 001401      BEQ      2S ;BRANCH IF DATA STORED CORRECTLY
015036 104006      HLT ;A HALT HERE INDICATES THAT THE
                                ;RELOCATION TO UPAR1 DID NOT WORK
                                ;R2 HAS THE VIRTUAL ADDR AND KPAR2
                                ;HAS THE BASE
015040 005077 164012 2S:      CLR      2UPAR1 ;CLEAR UPAR1 FOR NEXT TEST
015044 022700 010000      CMP      #10000,R0 ;CHECK TO SEE IF TEST IS DONE
015050 001415      BEQ      EOP ;BRANCH IF TEST IS OVER

```



```

015052 062700 000100      ADD      #100,R0      ;ADD BIT6 TO COUNTER
015056 062701 000100      ADD      #100,R1      ;ADD BIT6 TO KPARI'S VIRTUAL ADDR
015062 062702 000100      ADD      #100,R2      ;ADD BIT6 TO UPARI'S VIRTUAL ADDR
015066 162777 000001 164020  SUB      #1,@KPAR0    ;SUBTRACT BIT1 FROM KPARI'S BASE
015074 162777 000001 164016  SUB      #1,@KPAR2    ;SUBTRACT BIT1 FROM UPARI'S BASE
015102 000724                BR        3$          ;CONTINUE TEST

015104 104400                EOP:   SCOPE
015106 032767 010000 162454    BIT      #BIT12,SR
015114 001003                ENE     1$
015116 012700 015261      MOV      #FELL,R0      ;INHIBIT BELL?
015122 000402                BR        2$          ;BRANCH IF EELL IS INHIBITED
015124 012700 015265      1$:     MOV      #ASTER,R0    ;PUT ADDRESS OF EELL CHARS IN R0
015130 112001                2$:     MOV      (R0)+,R1    ;BRANCH TO OUTPUT CODE
015132 001405                2$:     MOV      #ASTER,R0    ;PUT ADDRESS OF ASTERICK CHARS IN R0
015134 010177 163660      3$:     MOV      (R0)+,R1    ;CHECK FOR TERMINATOR CODE
015140 105777 163652      3$:     MOV      R1,@DDBR    ;BRANCH IF BYTE IS ZERO
015144 100373                TSTB   @TCSR          ;OUTPUT CHARACTER TO BUFFER
015146 013701 000042      LOGICT: MOV      #42,R1    ;SEE IF STATUS REG GETS SET
015152 001405                BEQ     END           ;BRANCH UNTIL IT DOES
015154 000005                RESET
015156 004711                LOGIC: JSR      PC,@R1    ;MONITOR HOOK
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-C'<15><12>'@'
015200 042055 046040 043517
015206 041511 052040 051505
015214 020124 040515 047111
015222 042504 026503 030461
015230 042055 045502 040524
015236 041455 045015 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      R0
015300 012701 000040      CLRLP: MOV      #32,R1    ;COUNT OF REGISTERS TO BE CLEARED
015304 005070 001034      CLRLP: CLR      @ADR+TAB(R0) ;CLEAR REGISTERS THRU ADDRESS TABLE
015310 005720                TST      (R0)+
015312 077104                SOB     R1,CLRLP      ;MOVE POINTER
015314 000207                RTS      %7          ;LOOP TILL DONE

;SUBROUTINE TO MAKE ALL PAGES RW, BANK 0, 4K, UP

```

K04

DBKTA.C MACY11 27(732) 08-SEP-76 09:35 PAGE 49
DBKTAC.P11

```
015316 005077 163502
015322 012701 001034
015326 012700 000010
015332 005077 000020
015336 012731 077406
015342 077005
015344 052701 000020
015350 020127 001132
015354 002764
015356 000207
```

```
RHALL: CLR @SR0
MOV @ADRTAB,R1 ;R1 POINTS TO ADDRESS TABLE
RHL1: MOV #10,R0 ;R0 IS COUNTER
RHL2: CLR @20(R1) ;CLEAR PAR
MOV @77406,@(R1)+ ;SET PDR RW, 4K
SOB R0,RHL2
ADD #20,R1
CMP R1,@AREND ;POINTER TO NEXT GROUP
BLT RHL1
RTS X7
```

```
;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
015364 012706 001000
015370 012737 140000 177776
015376 012706 000400
015402 005037 177776
015406 000000
015410 016767 162154 000036
015416 012767 000002 000030
015424 000000
015426 000067 000140
015432 012767 015444 000134
015436 000177 000010
015444 005067 000122
015450 000177 000000
015454 000000
```

```
TESTX: CLR @#PS
MOV @KSTACK,SP
MOV #140000,@#PS ;SETUP USER TRAP
MOV @USTACK,SP
CLR @#PS
HALT ;WAIT FOR STARTING ADDRESS
MOV SR,RETRNX ;LOAD STARTING ADDRESS IN RETRNX
ADD #2,RETRNX ;ADD 2 TO POINT TO INSTRUCTION AFTER
HALT ;SET SR OPTIONS
CLR SCOPEF ;KEEP COUNT AT ZERO
MOV @XLOOP,RETURN ;LOAD SCOPE LOOP RETURN POINTER
JMP @RETRNX ;JUMP TO TEST
XLOOP: CLR SCOPEF ;KEEP COUNT AT ZERO
JMP @RETRNX ;JUMP TO TEST
RETRNX: 0
```

```
015456 032737 040000 177570
015464 001015
015466 032737 004000 177570
015474 001020
015476 026767 000070 000064
015504 100014
015506 005267 000060
015512 012737 000340 177776
015520 022626
015522 005037 177776
015526 005077 163272
015532 000177 000036
015536 005067 000030
015542 005267 163426
```

```
;SCOPE AND/OR ITERATION LOOP FOR EACH TEST 4000 TIMES
SCOPEC: BIT @BIT14,@#SR ;TEST SR FOR SCOPE
BNE SCOPEB ;YES SCOPE
BIT @BIT11,@#SR ;NO-TEST FOR ITERATION
BNE SCOPEG ;INHIBIT ITERATION
CMP SCOPEF,ICOUNT ;COMPARE CURRENT COUNT TO MAX NUMBER
BPL SCOPEG ;EXIT-DONE
INC SCOPEF ;INCREMENT COUNT
MOV @340,@#PS ;PREVENT TRAPPING WHILE MOVING STACK
SCOPEB: CMP (6)+,(6)+ ;REPOSITION STACK
CLR @#PS
CLR @SR0
JMP @RETURN
SCOPEG: CLR SCOPEF ;REPEAT TEST
INC TESTCT ;CLEAR COUNT
;STEP TEST COUNTER TO ALLOW CHECKING
;ORDER OF EXECUTION.
;SAVE SCOPE RETURN POINTER
;RETURN INLINE-NEXT TEST
```

```
015546 011667 000022
015552 022626
015554 005037 177776
015560 005077 163240
015564 000177 000004
015570 004000
015572 000000
015574 000000
```

```
MOV @%6,RETURN
CMP (6)+,(6)+
CLR @#PS
CLR @SR0
JMP @RETURN
ICOUNT: 4000
SCOPEF: 0
RETURN: 0
;ITERATION COUNT
;COUNT LOCATION FOR ITERATION LOOP
;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 EEQ .+4 ; BRANCH TO PRINT
015614 000430 BR CK ; INHIBIT, CHECK FOR HALT
015616 012667 000066 MOV (6)+,SAVPC ; PC OF FAILING ROUTINE
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,PROCT ; 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 CK: TST SR ; CHECK SR FOR HALT SWITCH
015702 100001 BPL .+4 ; BRANCH IF NOT SET
015704 000000 HALT ; HALT ON ERROR UP
015706 000002 RTI ; RETURN TO MAIN LINE
015710 000000 SAVPC: 0
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 TYPA: MOVB (0)+,TYPDAT ; GET CHARACTER
015734 122767 000100 000026 CMPB #100,TYPDAT ; CHECK FOR "3" CHARACTER
015742 001003 BNE TYPB ; BRANCH IF NOT "3"
015744 016700 000022 MOV SAVR0,%0 ; RESTORE R0
015750 000207 RTS PC ; TERMINATOR CHAR. EXIT
015752 116777 000012 163040 TYPB: MOVB TYPDAT,%TDBR ; OUTPUT CHAR TO PRINTER
015760 105777 163032 TSTB %TCSR ; WAIT FOR TTY READY
015764 100375 BPL -4
015766 000760 BR TYPA
015770 000000 TYPDAT: 0
015772 000000 SAVR0: 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 TSTB %TCSR ; WAIT FOR TTY READY
016022 100375 BPL -4
016024 000207 RTS %7 ; RETURN
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             ;BRANCH IF ZERO
016056 005267 000160              INC      PTEMP2          ;INCREMENT R3 IF ONE
016062 006167 000152              ROL     PTEMP1          ;ROTATE LEFT MOST OCTAL TO RIGHT END
016066 006167 000146              ROL     PTEMP1
016072 005567 000140              ADC     PRFLG           ;STORE CARRY
016076 005767 000132      P.CK:   TST      PRSFLG    ;CHECK FOR SHORT PRINTOUT
016102 001404              BEQ     P.WAIT         ;BRANCH IF NOT SET
016104 026727 000132 000260      CMP     PTEMP2,#260    ;CHECK FOR ZERO IF SET
016112 001410              BEQ     P.CONT        ;IF SET, GO TO NEXT CHARACTER
016114 016777 000122 162676      P.WAIT: MOV     PTEMP2,@TDBR ;OUTPUT NEXT CHARACTER
016122 105777 162670              TSTB   @TCSR          ;WAIT FOR TTY READY
016126 100375              BPL     .-4            ;PRINT REST OF NUMBER AFTER A NON-ZERO DIGIT
016130 005067 000100              CLR     PRSFLG         ;COUNT
016134 005267 000104      P.CONT: INC     PTEMP3    ;CHECK FOR DONE
016140 026727 000100 000006      CMP     PTEMP3,#6     ;BRANCH IF NOT DONE
016146 001001              BNE     P.CNT1
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           ;SET CARRY
016170 006167 000044              ROL     PTEMP1        ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTE
016174 006167 000040              ROL     PTEMP1
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      P.CK          ;LOOP
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
104000      PATCH1=EMT+0
104002      PATCH2=EMT+2
104004      PATCH3=EMT+4
016306 104000      EMTAB:  PATCH1
016310 104002      PATCH2
016312 104004      PATCH3
016314 015576      PRINT

;SUBROUTINE TO CHECK TEST SEQUENCE

```

N04

DBKTA.C MACY11 27(732) 08-SEP-76 09:35 PAGE 52
DBKTAC.P11

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:	CHP	TESTCT, TEMPN		; IS TEST SEQUENCE CORRECT
016362	001403				BEQ	ORDERA		; YES, CONTINUE
016364	062716	000002			ADD	#2, (SP)		; UPDATE FOR ERROR RETURN
016370	000207				RTS	PC		
016372	062716	000004		ORDERA:	ADD	#4, (SP)		; UPDATE FOR GOOD RETURN
016376	000207				RTS	PC		
016400	000000			TEMPN:	0			
					; MAP KERNEL PAR/PDR 7 TO EXTERNAL BANK			
016402	012777	007600	162522	KERN7:	MOV	#7600, @KPAR7		
016410	012777	077406	162474		MOV	#77406, @KPDR7		
016416	000207				RTS	PC		
	017712				.=17712			
017712	125252			DESTAD:	125252			
	000001				.END			

D05

ICOUNT	015570	1708	1718	1723	1730	1735	1738	1741	1745	1748	1753	1773	1777
INT25	005256	1781	1789	1808	1814	1823	1838	1841	1845	1848	1853	1897	1916
INT35	006732	1925	1931	1937	1943	1949	1958	1961	1965	1968	1973	2030	2033
INT36	007520	2043	2052	2056	2068	2073	2083	2087	2091	2094	2099	2157	2161
INT36A	007574	2178	2180	2186	2198	2203	2213	2217	2221	2224	2229	2303	2309
INT36B	007674	2317	2320	2326	2337	2343	2353	2357	2361	2364	2369	2443	2452
INT37	010054	2460	2471	2480	2487	2491	2500	2503	2507	2510	2515	2577	2582
INT37A	010130	403*	425*	463*	554*	563*	617*	688*	736*	757*	841*	1077*	1117*
INT37B	010230	1203*	1762*	1847*	2179*	2229*	2363*	2418*	2444*	2472*	2633*	2649*	
INT40	010410	1158	1166										
INT40A	010464	1445	1448										
INT40B	010564	1577	1584										
IOT35	006530	1573	1576										
KERN7	016402	1607	1616										
		1645	1652										
		1656	1664										
		1675	1684										
		1713	1720										
		1724	1732										
		1743	1752										
		1396	1399										
		790	901	964	1081	1121	1157	1205	1267	1298	1346	1574	1642
		1793	1870	1857	1879	1918	1984	2016	2045	2070	2127	2181	2245
		2319	2355	2420	2446	2474	2512	2780					2286
KPAR0	001114	367	759*	785*	899*	1079*	1119*	2327*	2338	2513*	2543*		
KPAR1	001116	378	968*	997*	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*										
KPOR0	001074	358	382	734*	760*	787*	900*	962*	1080*	1120*	1156*	1180	1293*
		1643*	1711*	1791*	1982*	2232	2290	2293*	2294	2301	2304	2307	2324
		2335	2341										2328
KPOR1	001076	379	963*	1184	1266*	1576*	1644*	1712*	1794*	1919*	2078*	2135*	2188*
		2424	2430*	2432	2447*	2450	2458						2421*
KPOR2	001100	360	2015*	2031									
KPOR3	001102	361											
KPOR4	001104	362											
KPOR5	001106	363											
KPOR6	001110	364											
KPOR7	001112	375	2051	2781*									
KRET34	006114	1275	1304	1312									
KSTACK	001000	324	402	415	431	459	491	519	558	586	613	649	696
		727	753	780	829	893	957	1073	1113	1151	1199	1261	1330
		1339	1352	1569	1637	1705	1770	1786	1823	1852	1874	1913	1977
		2010	2040	2065	2122	2175	2225	2240	2281	2314	2349	2414	2468
		2507	2614										
KTSTA	001166	393	887	888*	1159*	1191*	1192	1269*	1278	1578*	1629*	1630	1646*
		1698	1714*	1755*	1766	1796*	1816	1832*	1848	1881*	1906	1907*	1945
		2005	2018*	2034	2047*	2058	2076*	2113	2114*	2133*	2168	2169*	2183*
		2476*	2496										2220
KTVEC	001164	392	887*	1158*	1192*	1268*	1278*	1577*	1588*	1607*	1630*	1645*	1656*
		1698*	1713*	1724*	1743*	1775*	1795*	1816*	1831*	1848*	1880*	1906*	1921*
		1945*	1985*	2005*	2017*	2037*	2046*	2058*	2075*	2082*	2113*	2132*	2168*

K123	001012	2182#	2220#	2475#	2496#
K134	001014	227#	839#		
LOGIC	015156	228#	840#		
LOGICT	015146	213#	277#		
LOOP10	001010	255#	278#		
LOOP11	001010	622#	828#		
LOOP2	001358	657#	861#		
LOOP23	012464	440#	450#		
LOOP24	012706	2077#	2111#		
LOOP3	001456	2134#	2166#		
LOOP3A	001464	467#	472#		
LOOP3B	001464	469#	461#		
LOOP3C	001470	470#	478#		
LOOP3D	014106	2361#	2370#		
LOOP4	001536	497#	515#		
LOOP4A	001534	500#	513#		
LOOP4B	001560	501#	510#		
LOOPS	011436	186#	1905#		
LOOP6	001766	56#	52#		
LOC SA	001774	58#	581#		
LOOP7	001072	553#	609#		
LOP7A	00100	595#	608#		
LOP10A	001016	624#	625#		
LOP10B	001016	630#	638#		
LOP10C	001014	632#	637#		
LOP11A	001036	659#	660#		
LOP11B	001036	665#	673#		
LOP11C	001034	667#	672#		
LOP27	013400	2250#	2263#		
LOP27A	013412	2255#	2260#		
LOP31A	005460	1211#	1220#		
LOP31B	001464	1212#	1216#		
LOP31C	001516	1222#	1255#		
LOP31D	001530	1225#	1253#		
LOP31E	001530	1232#	1253#		
LOP31F	001530	1233#	1253#		
LOP32C	014106	2235#	2236#		
LOP32D	014106	2272#	2285#		
LOP32E	014106	2274#	2281#		
LOP32F	014238	2291#	2340#		
LOP32G	014238	2293#	2338#		
LOP5A	0011464	1899#	1901#		
LOP5B	0011644	526#	551#		
LOP5B	0011662	532#	544#		
L25A	013132	2187#	2206#		
L25B	013142	2189#	2204#		
L40	014510	2456#	2457#		
MPC	015243	2577#	2665#		
MPS	015252	2579#	2668#		
MTIT	015172	409#	2570#		
MG35B	006506	1353#	1392#		
MG35C	006716	1403#	1442#		
MG35D	007140	1452#	1497#		
MG35E	007360	1510#	1550#		
NOP	000240	268#			
0009D	007164	1496#	1501#	1505#	

TEST21	004070	956#							
TEST22	004564	1072#							
TEST23	004772	1112#							
TEST24	005166	1150#							
TEST25	005374	1198#							
TEST26	00552	1260#							
TEST27	005766	1283#							
TEST3	001416	458#							
TEST30	006252	1338#							
TEST31	007422	1568#							
TEST32	007796	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	2505#							
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#	2679#						
TYPOAT	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#							
UPAR5	001066	354#							
UPAR6	001070	355#							
UPAR7	001072	356#	807*	1206*	1300*	1347*	1955*	2246*	2356*
UPDR0	001034	340#	384	805*	1294*	1882	1898		
UPDR1	001036	341#	1920*						

N05

DBKTA.C MACY11 27(732) 08-SEP-76 09:35 PAGE 68
 DBKTAC.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

RTS	2275	2593	2605	2687	2704	2726	2775	2777	2782						
RTT	1989														
SEC	2731														
SOB	481	482	513	544	581	582	608	609	625	626	637	638	660	661	672
	673	1216	1241	1905	2381	2398	2457	2592	2601						
SUB	2162	2164	2543	2544	2749										
SWAB	2193	2197	2209	2213											
TRAP	267														
TST	406	420	423	435	480	512	534	543	580	607	636	639	671	674	690
	879	930	995	1162	1184	1240	1251	1271	1548	1812	1836	1843	1889	1902	1923
	1935	2049	2079	2080	2086	2136	2137	2142	2192	2232	2274	2361	2380	2397	2456
	2591	2671	2699	2709	2715	2728									
TSTB	976	2557	2689	2702	2720										
.ABS	1														
.ASCII	2570	2577	2579												
.ASCIZ	2581	2582													
.END	2786														
.EVEN	2583														
.LIST	1	304	399	414	430	458	490	518	557	585	612	648	683	695	726
	752	779	828	892	956	1072	1112	1150	1198	1260	1283	1338	1568	1636	1704
	1769	1785	1822	1851	1873	1912	1949	1976	2009	2039	2064	2121	2174	2224	2239
	2280	2313	2348	2413	2439	2467	2506								
.LIST	399	1563													
	1	304	399	414	430	458	490	518	557	585	612	648	683	695	726
	752	779	828	892	956	1072	1112	1150	1198	1260	1283	1338	1568	1636	1704
	1769	1785	1822	1851	1873	1912	1949	1976	2009	2039	2064	2121	2174	2224	2239
	2280	2313	2348	2413	2439	2467	2506								
.REM	1														
.REPT	304	1359	1409	1458	1516										
.TITLE	1														
.WORD	325														

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

#DBKTAC,DBKTAC.SEO/SOL/CRF/DS:ERFZ/EN:ABS=OSKM:DBKTAC.P11
 RUN-TIME: 10 19 4 SECONDS
 RUN-TIME RATIO: 124/35=3.5
 CORE USED: 10K (20 PAGES)

