

KT11-C

BASIC LOGIC TEST #1
MD-11-DCKTA-B

EP-DCKTA-B-DL-A

OCT 1978

COPYRIGHT ©1978

digital

FICHE 1 OF 1

Made In U.S.A.

This microfiche card contains a grid of 25 frames of logic test data, arranged in 5 rows and 5 columns. Each frame displays a complex logic diagram or test results, including various symbols, lines, and text. The data is presented in a structured, grid-like format typical of microfiche storage.

B01

DCKTA : MACY11 27(732) 09-SEP-76 15:07 PAGE 1
DCKTAB.P11

.REM*

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKTA-B-D
PRODUCT NAME: KT11-C BASIC LOGIC TEST ONE
DATE CREATED: AUGUST 1973
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: RICK FADDEN

1.0 ABSTRACT

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

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11/45 WITH KT11-C OPTION

2.2 STORAGE

THE PROGRAM REQUIRES MEMORY LOCATIONS 0 TO 17474.

3.0 LOADING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

4.0 STARTING PROCEDURE

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

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

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

5.2.2 HLT

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

5.2.3 TRAPCATCHER

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

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

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

5.2.4 EMTSRV (EMT DECODER)

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

5.2.5 CLRALL

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

5.2.6 RWALL

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

5.2.7 RWISP

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

5.2.8 RWDSP

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

5.3 PROGRAM AND/OR OPERATOR ACTION

THE PROGRAM FIRST CHECKS THOSE PROPERTIES OF THE KT11-C 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-C'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-C 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 1400. HOWEVER, IN CERTAIN TESTS IT MAY BE INITIALIZED TO A LOWER ADDRESS (VIRTUAL) TO MAKE UP FOR RELOCATION OF THE BANK.

THE SUPERVISOR STACK POINTER IS INITIALIZED TO 1000.

THE USER STACK POINTER IS INITIALIZED TO 400.

8.3 DISPLAY REGISTER

THE NUMBER OF THE CURRENT SUBTEST IS DISPLAYED.

8.4 EXECUTION ORDER CHECKING

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

9.0 PROGRAM DESCRIPTION

THE PROGRAM INITIALLY TESTS THOSE FEATURES OF THE KT11-C 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

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 7
DCKTAB.P11

```
*
:KT11-C BASIC LOGIC TEST ONE (RICK FADDEN)
:COPYRIGHT 1972, 1973 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
:REVISED TO REV. B BY BRUCE BURGESS
:TEST22, BELL ON PASS COMPLETE SUBROUTINE, AND
:ERROR PRINTOUT WITH ITERATIONS INHIBITED
:MODIFICATIONS MAKE UP REV. B
```

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

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

```
:DEFINITIONS
SCOPE=TRAP
NOP=240
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
R6=%6
R7=%7
SP=%6
PC=%7
SR=177570
PS=177776
STATUS=PS
HLT=104006
KTOFF=104010
```

```
: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
EMTSRV
340
.=34
SCOPEC
0
```

```
:LOAD STARTING AREA
.=200
```

104400
000240
000000
000001
000002
000003
000004
000005
000006
000007
000006
000007
177570
177776
177776
104006
104010

000030 000030
000030 017324
000032 000340
000034 000034
000034 016552
000036 000000

000200

000200 000167 001632 JMP START

```

000400 000400 ;LOAD DATA AREA
000000 ;=400
001000 001000 USTACK: 0
000000 ;=. +376
001400 001400 SSTACK: 0
000000 ;=. +376
001402 000000 KSTACK: 0
000000 000000 ;.WORD 0,0,0,0
001410 000000
    
```

```

001412 000200 K200: 200
001414 123456 K123: 123456
001416 134567 K134: 134567
001420 100000 KNR: 100000
001422 177564 TCSR: 177564
001424 177566 TDBR: 177566
001426 000000 TEMPX: 0
001430 000000 TEMP1: 0
001432 000000 TEMP2: 0
001434 177572 SR0: 177572
001436 177573 SR0H: 177573
001440 177574 SR1: 177574
001442 177575 SR1H: 177575
001444 177576 SR2: 177576
001446 177577 SR2H: 177577
001450 172516 SR3: 172516
001452 172517 SR3H: 172517
    
```

;CONSTANTS

;TELETYPE PRINTER CSR ADDRESS

;TEMPORARY STORAGE LOCATIONS

;KT11-C STATUS REGISTER ADDRESSES

```

001454 ADRTAB:
001454 177600 UIPDR0: 177600
001456 177602 UIPDR1: 177602
001460 177604 UIPDR2: 177604
001462 177606 UIPDR3: 177606
001464 177610 UIPDR4: 177610
001466 177612 UIPDR5: 177612
001470 177614 UIPDR6: 177614
001472 177616 UIPDR7: 177616
001474 177620 UDPDR0: 177620
001476 177622 UDPDR1: 177622
001500 177624 UDPDR2: 177624
001502 177626 UDPDR3: 177626
001504 177630 UDPDR4: 177630
001506 177632 UDPDR5: 177632
001510 177634 UDPDR6: 177634
001512 177636 UDPDR7: 177636
001514 177640 UIPAR0: 177640
001516 177642 UIPAR1: 177642
001520 177644 UIPAR2: 177644
001522 177646 UIPAR3: 177646
001524 177650 UIPAR4: 177650
001526 177652 UIPAR5: 177652
001530 177654 UIPAR6: 177654
001532 177656 UIPAR7: 177656
001534 177660 UDPAR0: 177660
001536 177662 UDPAR1: 177662
    
```

;USER I-SPACE PAGE DESCRIPTOR REGISTERS

;USER D-SPACE PAGE DESCRIPTOR REGISTERS

;USER I-SPACE PAGE ADDRESS REGISTERS

;USER D-SPACE PAGE ADDRESS REGISTERS

001540	177664	UDPAR2:	177664	
001542	177666	UDPAR3:	177666	
001544	177670	UDPAR4:	177670	
001546	177672	UDPAR5:	177672	
001550	177674	UDPAR6:	177674	
001552	177676	UDPAR7:	177676	
001554	172200	SIPDR0:	172200	;SUPERVISOR I-SPACE PAGE DESCRIPTOR REGISTERS
001556	172202	SIPDR1:	172202	
001560	172204	SIPDR2:	172204	
001562	172206	SIPDR3:	172206	
001564	172210	SIPDR4:	172210	
001566	172212	SIPDR5:	172212	
001570	172214	SIPDR6:	172214	
001572	172216	SIPDR7:	172216	
001574	172220	SDPDR0:	172220	;SUPERVISOR D-SPACE PAGE DESCRIPTOR REGISTERS
001576	172222	SDPDR1:	172222	
001600	172224	SDPDR2:	172224	
001602	172226	SDPDR3:	172226	
001604	172230	SDPDR4:	172230	
001606	172232	SDPDR5:	172232	
001610	172234	SDPDR6:	172234	
001612	172236	SDPDR7:	172236	
001614	172240	SIPAR0:	172240	;SUPERVISOR I-SPACE PAGE ADDRESS REGISTERS
001616	172242	SIPAR1:	172242	
001620	172244	SIPAR2:	172244	
001622	172246	SIPAR3:	172246	
001624	172250	SIPAR4:	172250	
001626	172252	SIPAR5:	172252	
001630	172254	SIPAR6:	172254	
001632	172256	SIPAR7:	172256	
001634	172260	SDPAR0:	172260	;SUPERVISOR D-SPACE PAGE ADDRESS REGISTERS
001636	172262	SDPAR1:	172262	
001640	172264	SDPAR2:	172264	
001642	172266	SDPAR3:	172266	
001644	172270	SDPAR4:	172270	
001646	172272	SDPAR5:	172272	
001650	172274	SDPAR6:	172274	
001652	172276	SDPAR7:	172276	
001654	172300	KIPDR0:	172300	;KERNEL I-SPACE PAGE DESCRIPTOR REGISTERS
001656	172302	KIPDR1:	172302	
001660	172304	KIPDR2:	172304	
001662	172306	KIPDR3:	172306	
001664	172310	KIPDR4:	172310	
001666	172312	KIPDR5:	172312	
001670	172314	KIPDR6:	172314	
001672	172316	KIPDR7:	172316	
001674	172320	KDPDR0:	172320	;KERNEL D-SPACE PAGE DESCRIPTOR REGISTERS
001676	172322	KDPDR1:	172322	
001700	172324	KDPDR2:	172324	
001702	172326	KDPDR3:	172326	
001704	172330	KDPDR4:	172330	
001706	172332	KDPDR5:	172332	
001710	172334	KDPDR6:	172334	
001712	172336	KDPDR7:	172336	

K01

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 10
 DCKTAB.P11

001714 172340
 001716 172342
 001720 172344
 001722 172346
 001724 172350
 001726 172352
 001730 172354
 001732 172356
 001734 172360
 001736 172362
 001740 172364
 001742 172366
 001744 172370
 001746 172372
 001750 172374
 001752 172376

KIPAR0: 172340
 KIPAR1: 172342
 KIPAR2: 172344
 KIPAR3: 172346
 KIPAR4: 172350
 KIPAR5: 172352
 KIPAR6: 172354
 KIPAR7: 172356
 KDPAR0: 172360
 KDPAR1: 172362
 KDPAR2: 172364
 KDPAR3: 172366
 KDPAR4: 172370
 KDPAR5: 172372
 KDPAR6: 172374
 KDPAR7: 172376
 ADREND= .-2

;KERNEL I-SPACE PAGE ADDRESS REGISTERS

;KERNEL D-SPACE PAGE ADDRESS REGISTERS

001754 177600
 001756 172200
 001760 172300
 001762 177640
 001764 172240
 001766 172340

PDRTAB: 177600
 PDREND: 172200
 PARTAB: 177640
 172240
 172340

;STARTING ADDRESSES OF PDR'S FOR EACH MODE

;STARTING ADDRESSES OF PAR'S FOR EACH MODE

001770 001654
 001772 000000
 001774 001554
 001776 040000
 002000 001454
 002002 140000

STATAB: KIPDR0
 0
 SIPDR0
 40000
 UIPDR0
 STAEND: 140000

;ADDRESS OF KERNEL TABLE OF PDR'S AND PAR'S

;ADDRESS OF SUPERVISOR TABLE OF PDR'S AND PAR'S

;ADDRESS OF USER TABLE OF PDR'S AND PAR'S

002004 000000
 002006 000000
 002010 000000
 002012 000000
 002014 000250
 002016 000252
 002020 177770
 002022 177770
 002024 000401
 002026 002777
 002030 100360
 002032 000000
 002034 000000

STAPNT: 0
 PAGES: 0
 SAVEA: 0
 SAVEB: 0
 KTVEC: 250
 KTSTA: 252
 UBRK: 177770
 MSKB: 177770
 SETMSK: 401
 EXPMSK: 2777
 PDRMSK: 100360
 TESTCT: 0
 BLOCKS: 0

;KT11-C VECTOR ADDRESS

;MICROBREAK REGISTER ADDRESS

002036 005037 177776
 002042 012706 001400
 002046 012737 040000 177776
 002054 012706 001000
 002060 012737 140000 177776
 002066 012706 000400
 002072 005037 177776
 002076 012767 002000 014574

;SET UP FOR START OF BASIC LOGIC TESTS

START: CLR @#PS
 MOV #KSTACK, SP
 MOV #40000, @#PS
 MOV #SSTACK, SP
 MOV #140000, @#PS
 MOV #USTACK, SP
 CLR @#PS
 MOV #2000, ICOUNT

;INITIALIZE STATUS
 ;SETUP KERNEL STACK
 ;INITIALIZE SUPERVISOR STACK

;INITIALIZE USER STACK

;INITIALIZE ITERATION COUNT

```

002104 012767 002124 014572      MOV      #TEST1+2,RETURN      ;SETUP SCOPE AND ITERATION LOOP RETURN
002112 012767 000001 177712      MOV      #1,TESTCT          ;INITIALIZE TEST COUNT
002120 000401                      BR       .+4                ;SKIP SCOPE INSTRUCTION

                                ;SRO AND SR1 SHOULD BE INITIALIZED TO 0
002122 104400      TEST1:  SCOPE
002124 012737 000001 177570      MOV      #1,@#SR           ;DISPLAY TEST NUMBER
002132 005037 177776                      CLR      @#PS              ;INITIALIZE PROCESSOR STATUS
002136 012706 001400                      MOV      #KSTACK,SP       ;INITIALIZE KERNEL STACK POINTER
002142 004767 014114                      JSR      %7,SETUP          ;INITIALIZE SRO,SR3
002146 026727 177660 000001      CMP      TESTCT,#1         ;IS THIS TEST BEING EXECUTED IN THE
002154 001401                      BEQ      .+4               ;CORRECT SEQUENCE?- BRANCH IF YES
002156 104006                      HLT                       ;TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

002160 105777 177236                      TSTB    @TCSR              ;WAIT FOR TTY READY TO AVOID KILLING BELL
002164 100375                      BPL     .-4                ;ISSUE INIT
002166 000005                      RESET                    ;CHECK SRO
002170 005777 177240                      TST     @SRO
002174 001401                      BEQ     .+4
002176 104006                      HLT
                                ;SRO WAS NOT INITIALIZED TO ZERO
002200 005777 177234                      TST     @SR1
002204 001401                      BEQ     .+4
                                ;CHECK SR1
002206 104006                      HLT
                                ;SR1 WAS NOT INITIALIZED TO ZERO
002210 012767 000010 014462      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
002216 104400      TEST2:  SCOPE
002220 012737 000002 177570      MOV      #2,@#SR           ;DISPLAY TEST NUMBER
002226 005037 177776                      CLR      @#PS              ;INITIALIZE PROCESSOR STATUS
002232 012706 001400                      MOV      #KSTACK,SP       ;INITIALIZE KERNEL STACK POINTER
002236 004767 014020                      JSR      %7,SETUP          ;INITIALIZE SRO,SR3
002242 026727 177564 000002      CMP      TESTCT,#2         ;IS THIS TEST BEING EXECUTED IN THE
002250 001401                      BEQ     .+4               ;CORRECT SEQUENCE?- BRANCH IF YES
002252 104006                      HLT                       ;TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

002254 012700 000001      LOOP2:  MOV      #1,R0           ;R0 CONTAINS BIT INDICATING POSITION BEING
002260 010001                      MOV      R0,R1             ;TESTED-SETUP R1 TO SET THAT BIT IN
002262 010102                      MOV      R1,R2             ;SRO UNLESS IT'S BIT 0 OR BIT 8
002264 046701 177534                      BIC     SETMSK,R1          ;R2 CONTAINS THE EXPECTED CONTENTS OF SRO
002270 046702 177532                      BIC     EXPMSK,R2          ;CLEAR THE BIT IN R2 IF IT SHOULDN'T SET IN SRO
002274 010177 177134                      MOV      R1,@SRO           ;SET THE BIT IN SRO UNLESS IT'S BIT 0 OR BIT 8
002300 020277 177130                      CMP      R2,@SRO           ;CHECK SRO
002304 001401                      BEQ     .+4                ;BRANCH IF OK
002306 104006                      HLT                       ;SRO INCORRECT AFTER VALUE IN R1
                                ;WAS LOADED INTO IT
002310 006300                      ASL     R0                 ;CHECK NEXT BIT POSITION
002312 103362                      BCC     LOOP2              ;BRANCH IF NOT ALL DONE
002314 005077 177114                      CLR     @SRO               ;REINITIALIZE SRO

```

```

;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

```

002320	104400			TEST3: SCOPE			
002322	012737	000003	177570	MOV	#3, @#SR		: DISPLAY TEST NUMBER
002330	005037	177776		CLR	@#PS		: INITIALIZE PROCESSOR STATUS
002334	012706	001400		MOV	#KSTACK, SP		: INITIALIZE KERNEL STACK POINTER
002340	004767	013716		JSR	%7, SETUP		: INITIALIZE SR0, SR3
002344	026727	177462	000003	CMP	TESTCT, #3		: IS THIS TEST BEING EXECUTED IN THE
002352	001401			BEQ	.+4		: CORRECT SEQUENCE?- BRANCH IF YES
002354	104006			HLT			: TEST EXECUTED OUT OF SEQUENCE- TESTCT
							: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
002356	012767	001000	014314	MOV	#1000, ICOUNT		: RAISE ITERATION COUNT
002364	004767	013704		JSR	%7, CLRALL		: INITIALIZE KT11-C REGISTERS
002370	012703	001762		MOV	#PARTAB, R3		: R3 POINTS TO TABLE OF PAR ADDRESSES
002374	012700	000003		MOV	#3, R0		: R0 IS COUNTER OF STATES LEFT TO TEST
002400	012301		LOOP3:	MOV	(R3)+, R1		: PUT ADDRESS OF 1ST PAR IN SET IN R1
002402	012702	000020		MOV	#20, R2		: R2 IS COUNTER OF PAR'S LEFT TO TEST IN SET
002406	012704	000001	LOOP3A:	MOV	#1, R4		: R4 IS BIT OF PAR BEING TESTED
002412	010411		LOOP3B:	MOV	R4, @R1		: SET BIT IN PAR
002414	020411			CMP	R4, @R1		: CHECK PAR
002416	001401			BEQ	.+4		: BRANCH IF OK
002420	104006			HLT			: PAR WHOSE ADDRESS IS IN R1
							: FAILED WHEN THE VALUE IN R4
							: WAS LOADED INTO IT
002422	006304			ASL	R4		: SETUP TO CHECK NEXT BIT POSITION
002424	020427	010000		CMP	R4, #10000		: ALL R/W BITS IN THIS PAR ALREADY CHECKED?
002430	001370			BNE	LOOP3B		: NO-BRANCH TO CONTINUE
002432	005011			CLR	@R1		: YES-CLEAR PAR JUST TESTED
002434	005721			TST	(R1)+		: MOVE POINTER TO NEXT PAR
002436	077215			SOB	R2, LOOP3A		: TEST ALL PAR'S IN SET
002440	077021			SOB	R0, LOOP3		: TEST ALL 3 REGISTER SETS
							: BITS 0-3 AND 8-14 OF ALL PDR'S SHOULD BE READ/WRITE
							: BITS 4,5, AND 15 SHOULD ALWAYS BE ZERO
							: BITS 6 AND 7 SHOULD BE ZERO IF PDR IS WRITTEN
							: ACTUAL CLEARING AND SETTING OF 6 AND 7 IS TESTED LATER
							: THIS TEST ALSO SHOWS THAT OUTPUT PATHS FROM PDR'S ARE OK
							: AND THAT EVERY PDR ADDRESS IS RESPONDED TO
002442	104400			TEST4: SCOPE			
002444	012737	000004	177570	MOV	#4, @#SR		: DISPLAY TEST NUMBER
002452	005037	177776		CLR	@#PS		: INITIALIZE PROCESSOR STATUS
002456	012706	001400		MOV	#KSTACK, SP		: INITIALIZE KERNEL STACK POINTER
002462	004767	013574		JSR	%7, SETUP		: INITIALIZE SR0, SR3
002466	026727	177340	000004	CMP	TESTCT, #4		: IS THIS TEST BEING EXECUTED IN THE
002474	001401			BEQ	.+4		: CORRECT SEQUENCE?- BRANCH IF YES
002476	104006			HLT			: TEST EXECUTED OUT OF SEQUENCE- TESTCT
							: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
002500	004767	013570		JSR	%7, CLRALL		: INITIALIZE KT11-C REGISTERS
002504	012703	001754		MOV	#PDRTAB, R3		: R3 POINTS TO TABLE OF PDR ADDRESSES
002510	012301		LOOP4:	MOV	(R3)+, R1		: LOAD ADDRESS OF 1ST PDR IN STATE INTO R1
002512	012702	000020		MOV	#20, R2		: USE R2 AS A COUNTER OF PDR'S
							: LEFT TO TEST
002516	012700	000001	LOOP4A:	MOV	#1, R0		: R0 INDICATES BIT POSITION BEING TESTED
002522	010005		LOOP4B:	MOV	R0, R5		
002524	046705	177300		BIC	PDRMSK, R5		: R5 CONTAINS EXPECTED RESULTING CONTENTS OF PDR
002530	010011			MOV	R0, @R1		: LOAD PDR


```

002700 000752          BR      LOP5A          ;CHECK FOR DUALS OF THE
002702 012767 000100 013770 DONES: MOV      #100,ICOUNT ;ADDRESS POINTED TO BY R1
                                                ;DROP ITERATION COUNT

;SHOW THAT BYTE ADDRESSING OF PAR'S WORKS FOR HIGH AND LOW BYTES
;CHECK ALL PAR'S
TEST6: SCOPE
002710 104400          MOV      #6,@#SR          ;DISPLAY TEST NUMBER
002712 012737 000006 177570          CLR      @#PS          ;INITIALIZE PROCESSOR STATUS
002720 005037 177776          MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
002724 012706 001400          JSR      %7,SETUP      ;INITIALIZE SR0,SR3
002730 004767 013326          CMP      TESTCT,#6    ;IS THIS TEST BEING EXECUTED IN THE
002734 026727 177072 000006          BEQ      .+4          ;CORRECT SEQUENCE?- BRANCH IF YES
002742 001401          HLT
002744 104006

002746 012767 002000 013724          MOV      #2000,ICOUNT ;RESTORE ITERATION COUNT
002754 004767 013314          JSR      %7,CLRALL    ;INITIALIZE KT11-C REGISTERS
002760 012703 001762          MOV      #PARTAB,R3  ;R3 POINTS TO TABLE OF PAR ADDRESSES
002764 012700 000003          MOV      #3,R0       ;R0 IS COUNTER OF SETS LEFT TO TEST
002770 012301          LOOP6: MOV      (R3)+,R1 ;PUT ADDRESS OF 1ST PAR IN SET IN R1
002772 012702 000020          MOV      #20,R2     ;R2 IS COUNTER OF PAR'S LEFT TO TEST IN SET
002776 012711 177777          LOOP6A: MOV     #-1,@R1 ;SET UP PAR BEING TESTED
003002 105011          CLRB   @R1          ;CLEAR LOW BYTE OF PAR
003004 022711 007400          CMP      #7400,@R1  ;CHECK PAR
003010 001401          BEQ      .+4          ;BRANCH IF OK
003012 104006          HLT          ;DATOB TO LOW BYTE OF PAR WHOSE
                                                ;ADDRESS IS IN R1 FAILED
003014 012711 177777          MOV      #-1,@R1    ;SET UP PAR TO TEST HIGH BYTE
003020 105061 000001          CLRB   1(R1)       ;CLEAR HIGH BYTE
003024 022711 000377          CMP      #377,@R1  ;CHECK PAR
003030 001401          BEQ      .+4
003032 104006          HLT          ;DATOB TO HIGH BYTE OF PAR WHOSE
                                                ;ADDRESS IS IN R1 FAILED
003034 005721          TST      (R1)+      ;MOVE POINTER TO ADDRESS NEXT PAR
003036 077221          SOB     R2,LOOP6A  ;TEST ALL PAR'S IN SET
003040 077025          SOB     R0,LOOP6    ;TEST ALL 3 REGISTER SETS

;SHOW THAT BYTE ADDRESSING OF PDR'S WORKS FOR HIGH AND LOW BYTES
TEST7: SCOPE
003042 104400          MOV      #7,@#SR          ;DISPLAY TEST NUMBER
003044 012737 000007 177570          CLR      @#PS          ;INITIALIZE PROCESSOR STATUS
003052 005037 177776          MOV      #KSTACK,SP      ;INITIALIZE KERNEL STACK POINTER
003056 012706 001400          JSR      %7,SETUP      ;INITIALIZE SR0,SR3
003062 004767 013174          CMP      TESTCT,#7    ;IS THIS TEST BEING EXECUTED IN THE
003066 026727 176740 000007          BEQ      .+4          ;CORRECT SEQUENCE?- BRANCH IF YES
003074 001401          HLT
003076 104006

003100 004767 013170          JSR      %7,CLRALL    ;INITIALIZE KT11-C REGISTERS
003104 012703 001754          MOV      #PDRTAB,R3  ;R3 POINTS TO TABLE OF PDR ADDRESSES
003110 012700 000003          MOV      #3,R0       ;R0 IS COUNTER OF SETS LEFT TO TEST
003114 012301          LOOP7: MOV      (R3)+,R1 ;PUT ADDRESS OF 1ST PDR IN SET INTO R1
003116 012702 000020          MOV      #20,R2     ;R2 IS COUNTER OF PDR'S LEFT TO TEST IN SET
003122 012711 177777          LOOP7A: MOV     #-1,@R1 ;SET UP PDR BEING TESTED
003126 105011          CLRB   @R1          ;CLEAR LOW BYTE OF PDR
    
```

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 15
DCKTAB.P11

003130	022711	077400		CMP	#77400, R1		:CHECK PDR
003134	001401			BEQ	.+4		:BRANCH IF OK
003136	104006			HLT			:DATOB TO LOW BYTE OF PDR WHOSE
							:ADDRESS IS IN R1 FAILED
003140	012711	177777		MOV	#-1, R1		:SET UP PDR TO TEST HIGH BYTE
003144	105061	000001		CLRB	1(R1)		:CLEAR HIGH BYTE
003150	022711	000017		CMP	#17, R1		:CHECK PDR
003154	001401			BEQ	.+4		
003156	104006			HLT			:DATOB TO HIGH BYTE OF PDR WHOSE
							:ADDRESS IS IN R1 FAILED
003160	005721			TST	(R1)+		:MOVE POINTER TO ADDRESS NEXT PDR
003162	077221			SOB	R2, LOOP7A		:TEST ALL PDR'S IN SET
003164	077025			SOB	R0, LOOP7		:TEST ALL 3 REGISTER SETS
							:INIT SHOULD HAVE NO EFFECT ON PAR'S
003166	104400			TEST10:	SCOPE		
003170	012737	000010	177570	MOV	#10, RSR		:DISPLAY TEST NUMBER
003176	005037	177776		CLR	RPS		:INITIALIZE PROCESSOR STATUS
003202	012706	001400		MOV	#KSTACK, SP		:INITIALIZE KERNEL STACK POINTER
003206	004767	013050		JSR	%7, SETUP		:INITIALIZE SR0, SR3
003212	026727	176614	000010	CMP	TESTCT, #10		:IS THIS TEST BEING EXECUTED IN THE
003220	001401			BEQ	.+4		:CORRECT SEQUENCE?- BRANCH IF YES
003222	104006			HLT			:TEST EXECUTED OUT OF SEQUENCE- TESTCT
							:CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
003224	012767	000010	013446	MOV	#10, ICOUNT		:DROP ITERATION COUNT
003232	005067	000112		CLR	TST10F		:CLEAR FLAG TO INDICATE FIRST PATTERN
003236	012704	005252		MOV	#5252, R4		:LOAD R4 WITH FIRST PATTERN
003242	012703	001762		TST10:	MOV #PARTAB, R3		:R3 POINTS TO TABLE OF PAR ADDRESSES
003246	012700	000003		MOV	#3, R0		:R0 IS USED AS COUNTER OF SETS LEFT TO LOAD
003252	012301			LOOP10:	MOV (R3)+, R1		:LOAD R1 WITH ADDRESS OF FIRST PAR IN SET
003254	012702	000020		MOV	#20, R2		:SETUP COUNTER TO LOAD PAR'S
003260	010421			LOP10A:	MOV R4, (R1)+		:LOAD PAR WITH PATTERN
003262	077202			SOB	R2, LOP10A		:LOAD ALL 16 PAR'S IN THIS SET
003264	077006			SOB	R0, LOOP10		:LOAD ALL 3 SETS
003266	105777	176130		TSTB	@TCSR		:WAIT FOR ANY TTY OUTPUT TO FINISH
003272	100375			BPL	.-4		
003274	000005			RESET			:ISSUE INIT
003276	012703	001762		MOV	#PARTAB, R3		:R3 POINTS TO TABLE OF PAR ADDRESSES
003302	012700	000003		MOV	#3, R0		:R0 IS USED AS A COUNTER OF SETS LEFT TO CHECK
003306	012301			LOP10B:	MOV (R3)+, R1		:LOAD R1 WITH ADDRESS OF 1ST PAR IN SET
003310	012702	000020		MOV	#20, R2		:SETUP COUNTER TO CHECK PAR'S
003314	020411			LOP10C:	CMP R4, R1		:COMPARE PATTERN LOADED TO PRESENT
003316	001401			BEQ	.+4		:CONTENTS OF PAR - BRANCH IF NOT CHANGED
003320	104006			HLT			:PAR WHOSE ADDRESS IS IN R1
							:WAS CHANGED BY INIT
							:IT WAS LOADED WITH THE VALUE IN R4
							:AND THEN INIT WAS ISSUED
003322	005721			TST	(R1)+		:MOVE POINTER TO ADDRESS NEXT PAR
003324	077205			SOB	R2, LOP10C		:TEST ALL 16 PAR'S IN THIS SET
003326	077011			SOB	R0, LOP10B		:TEST ALL 3 SETS
003330	005767	000014		TST	TST10F		:CHECK FOR BOTH PATTERNS USED
003334	001006			BNE	TEST11		:IF DONE, GO TO NEXT TEST
003336	005267	000006		INC	TST10F		:IF NOT, SET FLAG
003342	012704	002525		MOV	#2525, R4		:LOAD OTHER PATTERN INTO R4
003346	000735			BR	TST10		:REPEAT TEST WITH 2ND PATTERN

003350 000000

TST10F: 0

; INIT SHOULDN'T CLEAR OR SET ANY OF THE R/W BITS IN THE PDR'S
 ; A CHECKERBOARD PATTERN IS LOADED INTO ALL PDR'S, THEN INIT
 ; IS ISSUED AND THE PDR'S ARE CHECKED TO SEE IF THEY WERE ALTERED
 ; THEN THE COMPLEMENT OF THE FIRST PATTERN IS LOADED AND AFTER INIT IS
 ; ISSUED THE PDR'S ARE CHECKED AGAIN

003352 104400
 003354 012737 000011 177570
 003362 005037 177776
 003366 012706 001400
 003372 004767 012664
 003376 026727 176430 000011
 003404 001401
 003406 104006

TEST11: SCOPE
 MOV #11, @#SR
 CLR @#PS
 MOV #KSTACK, SP
 JSR %7, SETUP
 CMP TESTCT, #11
 BEQ .+4
 HLT

; DISPLAY TEST NUMBER
 ; INITIALIZE PROCESSOR STATUS
 ; INITIALIZE KERNEL STACK POINTER
 ; INITIALIZE SRO, SR3
 ; IS THIS TEST BEING EXECUTED IN THE
 ; CORRECT SEQUENCE? - BRANCH IF YES
 ; TEST EXECUTED OUT OF SEQUENCE - TESTCT
 ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

003410 005067 000112
 003414 012704 025012
 003420 012703 001754
 003424 012700 000003
 003430 012301
 003432 012702 000020
 003436 010421
 003440 077202
 003442 077006
 003444 105777 175752
 003450 100375
 003452 000005
 003454 012703 001754
 003460 012700 000003
 003464 012301
 003466 012702 000020
 003472 020411
 003474 001401
 003476 104006

CLR TST11F
 MOV #25012, R4
 TST11: MOV #PDRTAB, R3
 MOV #3, R0
 LOOP11: MOV (R3)+, R1
 MOV #20, R2
 LOP11A: MOV R4, (R1)+
 SOB R2, LOP11A
 SOB R0, LOOP11
 TSTB @TCSR
 BPL .-4
 RESET
 MOV #PDRTAB, R3
 MOV #3, R0
 LOP11B: MOV (R3)+, R1
 MOV #20, R2
 LOP11C: CMP R4, @R1
 BEQ .+4
 HLT

; CLEAR FLAG TO INDICATE FIRST PATTERN
 ; LOAD PATTERN IN R4
 ; R3 POINTS TO TABLE OF PDR ADDRESSES
 ; R0 IS COUNTER OF SETS LEFT TO LOAD
 ; LOAD R1 WITH ADDRESS OF FIRST PDR IN SET
 ; SETUP COUNTER TO LOAD PDR'S
 ; LOAD PDR WITH PATTERN
 ; LOAD ALL 16 IN THIS SET
 ; LOAD ALL 3 SETS OF PDR'S
 ; WAIT FOR ANY TTY OUTPUT TO FINISH

003500 005721
 003502 077205
 003504 077011
 003506 005767 000014
 003512 001006
 003514 005267 000006
 003520 012704 052405
 003524 000735
 003526 000000

TST (R1)+
 SOB R2, LOP11C
 SOB R0, LOP11B
 TST TST11F
 BNE TEST12
 INC TST11F
 MOV #52405, R4
 BR TST11
 TST11F: 0

; ISSUE INIT
 ; R3 POINTS TO TABLE OF PDR ADDRESSES
 ; R0 IS COUNTER OF SETS LEFT TO LOAD
 ; LOAD R1 WITH ADDRESS OF FIRST PDR IN SET
 ; SETUP COUNTER TO CHECK PDR'S
 ; COMPARE PATTERN LOADED INTO PDR
 ; WITH CONTENTS OF PDR AFTER INIT
 ; PDR WHOSE ADDRESS IS IN R1
 ; WAS CHANGED BY INIT
 ; MOVE POINTER TO ADDRESS NEXT PDR
 ; CHECK ALL 16 PDR'S IN THIS SET
 ; CHECK ALL 3 SETS OF PDR'S
 ; CHECK FOR BOTH PATTERNS USED
 ; IF DONE, GO TO NEXT TEST
 ; IF NOT, SET FLAG
 ; LOAD 2ND PATTERN INTO R4
 ; REPEAT TEST WITH SECOND PATTERN

; SHOW THAT SR1 TRACKS WITH KT11-C OFF AND THAT IT IS READ-ONLY
 ; SHOW THAT IF NR ERROR IS SET IN SRO, SR1 STOPS TRACKING
 ; NOTE THAT MOST OF THIS TEST IS ACTUALLY EXECUTED TWICE, FIRST
 ; WITH THE REGISTER SET BIT (PS<11>) CLEAR, AND THEN WITH IT SET

003530 104400
 003532 012737 000012 177570
 003540 005037 177776
 003544 012706 001400
 003550 004767 012506

TEST12: SCOPE
 MOV #12, @#SR
 CLR @#PS
 MOV #KSTACK, SP
 JSR %7, SETUP

; DISPLAY TEST NUMBER
 ; INITIALIZE PROCESSOR STATUS
 ; INITIALIZE KERNEL STACK POINTER
 ; INITIALIZE SRO, SR3

003554	026727	176252	000012	CMP	TESTCT, #12	: IS THIS TEST BEING EXECUTED IN THE
003562	001401			BEQ	+.4	: CORRECT SEQUENCE?- BRANCH IF YES
003564	104006			HLT		: TEST EXECUTED OUT OF SEQUENCE- TESTCT
						: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
003566	012767	002000	013104	MOV	#2000, ICOUNT	: RESTORE ITERATION COUNT
003574	005067	175630		CLR	TEMP1	: CLEAR FLAG TO INDICATE USING REGISTER SET 0
003600	016703	175630		MOV	SRO, R3	: SETUP R3 TO REFERENCE SRO
003604	005723			TST	(R3)+	
003606	012743	100000		MOV	#100000, -(R3)	: SET NR ERROR BIT
003612	022777	171427	175620	CMP	#171427, @SR1	: CHECK SR1
003620	001401			BEQ	+.4	
003622	104006			HLT		: SR1 INCORRECT - SHOULD HAVE TRACKED
						: CHANGE OF R7 BY +2 (LOW BYTE) AND
						: R3 BY -2 (HIGH BYTE) AND "LOCKED UP"
						: CONTAINING THAT VALUE
003624	005077	175604		CLR	@SRO	: CLEAR NR ERROR BIT TO RESUME TRACKING
003630	012701	001412		MOV	#K200, R1	: SETUP R1 TO SET NR BIT
003634	016702	175576		MOV	SROH, R2	: SETUP R2 TO REFERENCE HIGH BYTE
003640	005202			INC	R2	: OF SRO
003642	112142			MOVB	(R1)+, -(R2)	: SET NR ERROR BIT
003644	022777	175011	175566	CMP	#175011, @SR1	: CHECK SR1
003652	001401			BEQ	+.4	
003654	104006			HLT		: SR1 INCORRECT - SHOULD HAVE TRACKED
						: CHANGES OF R1 BY +1 (LOW BYTE) AND
						: R2 BY -1 (HIGH BYTE)
						: CLEAR NR ERROR BIT TO RESUME TRACKING
003656	005077	175552		CLR	@SRO	: SET NR ERROR BIT
003662	016777	175532	175544	MOV	KNR, @SRO	: CHECK SR1
003670	005777	175544		TST	@SR1	
003674	001401			BEQ	+.4	
003676	104006			HLT		: SR1 INCORRECT - SHOULD SHOW NO
						: REGISTERS CHANGED
						: TRY TO WRITE SR1 - SHOULD BE READ-ONLY
003700	012777	177777	175532	MOV	#-1, @SR1	
003706	005777	175526		TST	@SR1	
003712	001401			BEQ	+.4	
003714	104006			HLT		: SR1 WAS ALTERED BY WRITING IT
						: WHILE NR ERROR WAS STILL SET
						: CLEAR NR ERROR BIT TO RESUME TRACKING
003716	005077	175512		CLR	@SRO	: SETUP R4 TO SET NR BIT
003722	012704	001413		MOV	#K200+1, R4	: SETUP R5 TO REFERENCE SRO (HIGH BYTE)
003726	016705	175504		MOV	SROH, R5	: SET NR ERROR BIT
003732	114425			MOVB	-(R4), (R5)+	: CHECK SR1
003734	022777	006774	175476	CMP	#6774, @SR1	
003742	001401			BEQ	+.4	
003744	104006			HLT		: SR1 INCORRECT - SHOULD HAVE TRACKED
						: CHANGE OF R4 BY -1 (LOW BYTE)
						: AND R5 BY +1 (HIGH BYTE)
						: CLEAR NR ERROR BIT TO RESUME TRACKING
003746	005077	175462		CLR	@SRO	: SAVE STACK POINTER
003752	010667	175450		MOV	R6, TEMPX	: SETUP R6 TO SET NR BIT
003756	012706	001422		MOV	#KNR+2, R6	: SETUP R3 TO REFERENCE SRO
003762	016703	175446		MOV	SRO, R3	: SET NR BIT
003766	014623			MOV	-(R6), (R3)+	: RESTORE STACK POINTER
003770	016706	175432		MOV	TEMPX, R6	: CHECK SR1
003774	022777	011766	175436	CMP	#11766, @SR1	
004002	001401			BEQ	+.4	
004004	104006			HLT		: SR1 INCORRECT - SHOULD HAVE TRACKED
						: CHANGE OF R6 BY -2 (LOW BYTE)

H02

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 20
 DCKTAB.P11

```
004522 001401      BEQ      .+4
004524 104006      HLT
```

```
;SR3 INCORRECT AFTER A CLRB (HIGH)
```

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

```
004526 104400
004530 012737 000015 177570
004536 005037 177776
004542 012706 001400
004546 004767 011510
004552 026727 175254 000015
004560 001401
004562 104006
```

```
TEST15: SCOPE
MOV      #15, @#SR
CLR      @#PS
MOV      #KSTACK, SP
JSR      %7, SETUP
CMP      TESTCT, #15
BEQ      .+4
HLT
```

```
;DISPLAY TEST NUMBER
;INITIALIZE PROCESSOR STATUS
;INITIALIZE KERNEL STACK POINTER
;INITIALIZE SR0, SR3
;IS THIS TEST BEING EXECUTED IN THE
;CORRECT SEQUENCE?- BRANCH IF YES
;TEST EXECUTED OUT OF SEQUENCE- TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
```

```
004564 004767 011504
```

```
JSR      %7, CLRALL
```

```
;THIS TEST SHOULDN'T GO THRU ANY PAR/PDR PAIR'S
;SO MAKE THEM ALL GIVE NON-RESIDENT
;AND PAGE LENGTH ERRORS IF ACCESSED
;3 BLOCKS OF KERNEL PDR0 MUST BE MAPPED
;TO ALLOW TRAPS AND ABORTS
;TURN ON DESTINATION ONLY RELOCATION
;INIT SHOULD CLEAR DESTINATION ONLY BIT
;IF THE FETCH IS RELOCATED THIS WILL GIVE A
;PL ABORT
;IF KT11-C STILL ON, THIS SHOULD CAUSE
;PL AND NR ERRORS
;IF KT11-C IS OFF, BIT 8 OF SR0 READS
;AS STILL SET
;IF KT11-C IS ON, NO NR OR PL ABORT
;OCCURRED AND RESET FAILED TO TURN KT11-C OFF
```

```
004570 012777 001006 175056
```

```
MOV      #1006, @KIPDR0
```

```
004576 012777 000400 174630
004604 000005
```

```
MOV      #400, @SR0
RESET
```

```
004606 032777 000400 174620
004614 001401
004616 000000
```

```
BIT      #400, @SR0
BEQ      .+4
HALT
```

```
004620 005077 174610
```

```
CLR      @SR0
```

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

```
004624 104400
004626 012737 000016 177570
004634 005037 177776
004640 012706 001400
004644 004767 011412
004650 026727 175156 000016
004656 001401
004660 104006
```

```
TEST16: SCOPE
MOV      #16, @#SR
CLR      @#PS
MOV      #KSTACK, SP
JSR      %7, SETUP
CMP      TESTCT, #16
BEQ      .+4
HLT
```

```
;DISPLAY TEST NUMBER
;INITIALIZE PROCESSOR STATUS
;INITIALIZE KERNEL STACK POINTER
;INITIALIZE SR0, SR3
;IS THIS TEST BEING EXECUTED IN THE
;CORRECT SEQUENCE?- BRANCH IF YES
;TEST EXECUTED OUT OF SEQUENCE- TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
```

```
004662 004767 011406
004666 012777 000001 175020
004674 012777 077406 174752
004702 012701 004754
004706 012777 000400 174520
004714 021111
004716 001001
004720 000000
```

```
JSR      %7, CLRALL
MOV      #1, @KIPAR0
MOV      #77406, @KIPDR0
MOV      #DATA16, R1
MOV      #400, @SR0
CMP      @R1, @R1
BNE      .+4
HALT
```

```
;INITIALLY CLEAR ALL KT11-C REGISTERS
;OFFSET KERNEL PAR/PDR PAIR 0 ONE BLOCK
;FROM BANK 0, AND MAKE IT RW
;LOAD A BANK 0 ADDRESS INTO R1
;TURN ON DESTINATION ONLY RELOCATION
;THIS TEST WILL FAIL IF BOTH ARE
;RELOCATED OR BOTH ARE NOT RELOCATED
;SOURCE AND DESTINATION BOTH ADDRESSED
;SAME LOCATION
;TURN OFF DESTINATION ONLY RELOCATION
```

```
004722 000005
```

```
RESET
```

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 21
DCKTAB.P11

```

004724 012701 004654      MOV      #DATA16-100,R1      ;LOAD DESTINATION ADDRESS MINUS RELOCATION
                                ;FACTOR INTO R1
004730 012702 004754      MOV      #DATA16,R2        ;LOAD SOURCE ADDRESS INTO R2
004734 012777 000400 174472  MOV      #400,SR0         ;TURN ON DESTINATION ONLY RELOCATION
004742 021211                CMP      @R2,@R1          ;USE SAME INSTRUCTION AND ADDRESS
004744 001401                BEQ      .+4              ;MODES AS BEFORE-SHOULD RELOCATE DESTINATION
                                ;TO ADDRESS SAME LOCATION AS SOURCE
004746 000000                HALT                       ;DESTINATION NOT RELOCATED OR INCORRECTLY
                                ;RELOCATED OR SOURCE RELOCATED (IN
                                ;DESTINATION ONLY RELOCATION)
                                ;TURN OFF RELOCATION

```

```

004750 000005      RESET
004752 000401      BR      TEST17
004754 132465

```

DATA16: 132465

;SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU KERNEL PAGE 7 I-SPACE WILL
;CLEAR THE DESTINATION ONLY RELOCATION BIT AND TURN OFF DESTINATION ONLY RELOCATION
TEST17: SCOPE

```

004756 104400                MOV      #17,@#SR        ;DISPLAY TEST NUMBER
004760 012737 000017 177570  CLR      @#PS           ;INITIALIZE PROCESSOR STATUS
004766 005037 177776                MOV      #KSTACK,SP     ;INITIALIZE KERNEL STACK POINTER
004772 012706 001400                JSR      %7,SETUP        ;INITIALIZE SR0,SR3
004776 004767 011260                CMP      TESTCT,#17     ;IS THIS TEST BEING EXECUTED IN THE
005002 026727 175024 000017  BEQ      .+4            ;CORRECT SEQUENCE?- BRANCH IF YES
005010 001401                HLT                       ;TEST EXECUTED OUT OF SEQUENCE- TESTCT
005012 104006                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

```

```

005014 004767 011254                JSR      %7,CLRALL       ;INITIALLY CLEAR ALL KT11-C REGISTERS
005020 012777 000001 174666  MOV      #1,@KIPAR0     ;MAP KERNEL PAGE 0 I-SPACE
                                ;TO BANK 0 OFFSET BY 1 PAGE
005026 012777 077406 174620  MOV      #77406,@KIPDR0 ;OFFSET IS USED TO PROVE KT11-C IS
                                ;TURNED OFF AFTER CLEARING BIT 8, SR0
005034 012701 004754                MOV      #DATA16,R1     ;SETUP R1 TO REFERENCE KERNEL PAGE 0
005040 012777 007600 174664  MOV      #7600,@KIPAR7  ;MAP KERNEL PAGE 7 I-SPACE
005046 012777 077406 174616  MOV      #77406,@KIPDR7 ;TO THE EXTERNAL BANK, RW,4K
005054 016702 174354                MOV      SR0,R2         ;SETUP R2 TO ADDRESS SR0
005060 012777 000400 174346  MOV      #400,SR0       ;TURN ON DESTINATION ONLY RELOCATION
005066 005012                CLR      @R2           ;CLEAR SR0 THRU KERNEL PAGE 7
005070 021111                CMP      @R1,@R1       ;SHOW THAT KT11-C IS OFF
005072 001401                BEQ      .+4
005074 000000                HALT

```

```

005076 032777 000400 174330  BIT      #400,SR0       ;KT11-C STILL ON AFTER CLR SHOULD HAVE
005104 001402                BEQ      .+6            ;TURNED IT OFF
005106 104006                HLT                       ;SHOW THAT BIT 8, SR0 IS NOW ZERO
005110 000005      RESET                ;DESTINATION ONLY RELOCATION BIT IS STILL ON
                                ;MAKE SURE THAT KT11-C IS OFF

```

;SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU SUPERVISOR PAGE 7 I-SPACE
;WILL TURN OFF DESTINATION ONLY RELOCATION

```

005112 004767 011156                JSR      %7,CLRALL       ;INITIALLY CLEAR ALL PAR/PDR PAIRS
005116 012777 000001 174470  MOV      #1,@SIPAR0     ;MAP SUPERVISOR 0 I-SPACE
005124 012777 077406 174422  MOV      #77406,@SIPDR0 ;TO BANK 0 OFFSET BY 1 PAGE, RW
005132 012701 004754                MOV      #DATA16,R1     ;SETUP R1 TO REFERENCE SUPERVISOR 0
005136 012777 007600 174466  MOV      #7600,@SIPAR7  ;MAP SUPERVISOR 7 I-SPACE
005144 012777 077406 174420  MOV      #77406,@SIPDR7 ;TO THE EXTERNAL BANK, RW,4K
005152 016702 174256                MOV      SR0,R2         ;SETUP R2 TO ADDRESS SR0
005156 012737 040000 177776  MOV      #40000,@#PS    ;SET MODE TO SUPERVISOR

```

```

005164 012777 000400 174242      MOV    #400,SR0      ;TURN ON DESTINATION ONLY RELOCATION
005172 005012                    CLR    SR2          ;CLEAR SR0 THRU SUPERVISOR PAGE 7
005174 021111                    CMP    SR1,SR1     ;SHOW THAT KT11-C IS OFF
005176 001401                    BEQ    .+4
005200 000777                    BR     .           ;RELOCATION STILL ON AFTER CLR OF SR0
                                ;THUR SUPERVISOR PAGE 7
                                ;MAKE SURE THAT KT11-C IS OFF

005202 000005                    RESET
005204 005037 177776            CLR    SRPS
    
```

;SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU USER PAGE 7 I-SPACE
 ;WILL TURN OFF DESTINATION ONLY RELOCATION

```

005210 004767 011060            JSR    %7,CLRALL   ;INITIALLY CLEAR ALL PAR/PDR PAIRS
005214 012777 000001 174272    MOV    #1,SRIPAR0  ;MAP USER 0 I-SPACE TO
005222 012777 077406 174224    MOV    #77406,SRIPDR0 ;BANK 0 OFFSET BY 1 PAGE, RW
005230 012701 004754            MOV    #DATA16,R1  ;SETUP R1 TO REFERENCE USER 0
005234 012777 007600 174270    MOV    #7600,SRIPAR7 ;MAP USER 7 I-SPACE TO THE
005242 012777 077406 174222    MOV    #77406,SRIPDR7 ;EXTERNAL BANK, RW,4K
005250 016702 174160            MOV    SR0,R2      ;SETUP R2 TO ADDRESS SR0
005254 012737 140000 177776    MOV    #140000,SRPS ;SET MODE TO USER
005262 012777 000400 174144    MOV    #400,SR0    ;TURN ON DESTINATION ONLY RELOCATION
005270 005012                    CLR    SR2          ;CLEAR SR0 THRU USER PAGE 7
005272 021111                    CMP    SR1,SR1     ;SHOW THAT KT11-C IS OFF
005274 001401                    BEQ    .+4
005276 000777                    BR     .           ;RELOCATION STILL ON
005300 000005                    RESET              ;MAKE SURE THAT KT11-C IS OFF
005302 005037 177776            CLR    SRPS
    
```

;SHOW THAT ALL BLOCK BOUNDARY REFERENCES REFERENCE THE CORRECT
 ;I-SPACE PAR AND RELOCATE CORRECTLY WITHOUT D-SPACE ENABLED
 ;USE DESTINATION ONLY RELOCATION
 ;MAP ALL D-SPACE PAGES NON-RESIDENT
 ;MAP ALL I-SPACE PAGES RESIDENT READ WRITE

```

;
;      R0      - POINTS TO THE ADDRESS OF THE CURRENT I-SPACE PAR
;               IN THE ADDRESS TABLE
;      R1      - CONTAINS VIRTUAL ADDRESS BEING USED TO REFERENCE START OF BLOCK
;      R2      - CONTAINS VIRTUAL ADDRESS BEING USED TO REFERENCE END OF BLOCK
;      R3      - POINTS TO EXPECTED CONTENTS OF START OF BLOCK
;      R4      - POINTS TO EXPECTED CONTENTS OF END OF BLOCK
;      R5      - USED TO REFERENCE SR0 TO TURN OFF DESTINATION ONLY RELOCATION
    
```

```

005306 104400                    TEST20: SCOPE
005310 012737 000020 177570    MOV    #20,SR      ;DISPLAY TEST NUMBER
005316 005037 177776            CLR    SRPS        ;INITIALIZE PROCESSOR STATUS
005322 012706 001400            MOV    #KSTACK,SP  ;INITIALIZE KERNEL STACK POINTER
005326 004767 010730            JSR    %7,SETUP    ;INITIALIZE SR0,SR3
005332 026727 174474 000020    CMP    TESTCT,#20  ;IS THIS TEST BEING EXECUTED IN THE
005340 001401                    BEQ    .+4         ;CORRECT SEQUENCE?- BRANCH IF YES
005342 104006                    HLT               ;TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
    
```

```

005344 004767 010724            JSR    %7,CLRALL   ;INITIALLY CLEAR ALL KT11-C REGISTERS
005350 004767 011012            JSR    %7,RWISP    ;MAKE ALL I SPACE PAR/PDR PAIR'S RW,
                                ;BANK 0,4K
005354 013767 017700 174426    MOV    SR17700,SAVEA ;SAVE CONTENTS OF LOCATIONS TO BE USED
005362 013767 017776 174422    MOV    SR17776,SAVEB ;AS START AND END OF PHYSICAL BLOCK
005370 012737 123456 017700    MOV    #123456,SR17700 ;SET UP LOCATIONS TO BE REFERENCED
    
```

```

005376 012737 134567 017776      MOV      #134567, R3
005404 012703 001414                MOV      #K123, R3
005410 012704 001416                MOV      #K134, R4

```

```

005414 012767 000100 011256      MOV      #100, ICOUNT
005422 012737 040000 177776      MOV      #40000, R#PS
005430 012706 001000                MOV      #SSTACK, SP
005434 012737 140000 177776      MOV      #140000, R#PS
005442 012706 000400                MOV      #USTACK, SP
005446 005037 177776                CLR      R#PS
005452 012767 001770 174324      MOV      #STATAB, STAPNT

```

```

005460 017700 174320                STAT20: MOV    RSTAPNT, R0
005464 062700 000040                ADD     #40, R0
005470 062767 000002 174306      ADD     #2, STAPNT

```

```

005476 017737 174302 177776      MOV     RSTAPNT, R#PS
005504 062767 000002 174272      ADD     #2, STAPNT
005512 012767 000010 174266      MOV     #8, PAGES
005520 012770 007600 000016      MOV     #7600, R16(R0)
005526 016705 173702                MOV     SR0, R5
005532 005001                CLR     R1
005534 012702 000076                MOV     #76, R2
005540 012767 000200 174266      PAG20: MOV     #128, BLOCKS
005546 012770 000177 000000      MOV     #177, R(R0)

```

```

005554 022767 000001 174224      CMP     #1, PAGES

```

```

005562 001005                BNE     BLK20
005564 012770 007600 177776      MOV     #7600, R-2(R0)
005572 042705 020000                BIC     #20000, R5
005576 012777 000400 173630      BLK20: MOV     #400, SR0
005604 021311                CMP     R3, R1
005606 001401                BEQ
005610 000000                HALT

```

```

005612 021412                CMP     R4, R2
005614 001401                BEQ
005616 000000                HALT

```

```

005620 005015                CLR     R5
005622 005370 000000                DEC     R(R0)
005626 062701 000100                ADD     #100, R1
005632 062702 000100                ADD     #100, R2
005636 005367 174172                DEC     BLOCKS
005642 001355                BNE     BLK20

```

```

;LOAD R3 AND R4 WITH ADDRESSES OF
;LOCATIONS CONTAINING EXPECTED CONTENTS
;OF START AND END OF BLOCK
;THIS ALLOWS USING THE SAME
;INSTRUCTIONS AS HAVE ALREADY
;BEEN SHOWN TO WORK CORRECTLY IN
;DESTINATION ONLY RELOCATION
;CHANGE ITERATION COUNT
;CHANGE TO SUPERVISOR
;SET UP SUPERVISOR STACK POINTER
;CHANGE TO USER
;SET UP USER STACK POINTER
;RETURN TO KERNEL
;SET UP TO REFERENCE MODE TABLE
;THIS TABLE CONTAINS FIRST ADDRESS
;OF TABLE OF ADDRESSES OF PAR'S AND
;PDR'S FOR EACH MODE, AND THE
;VALUE OF PROCESSOR STATUS FOR THE
;DESIRED MODE
;PICK UP ADDRESS OF START OF
;ADDRESS TABLE FOR NEW MODE
;MOVE POINTER TO ADDRESS VALUE TO
;LOAD IN PROCESSOR STATUS
;SET PROCESSOR STATUS TO NEW MODE
;MOVE POINTER TO ADDRESS VALUES FOR NEXT MODE
;SET UP COUNTER OF PAGES LEFT TO TEST
;SET UP RELOCATED REFERENCE TO SR0
;USED TO TURN DESTINATION ONLY RELOCATION OFF
;INITIALIZE R1 TO CONTAIN VA OF START OF PAGE
;INITIALIZE R2 TO VA OF END OF PAGE
;SET UP BLOCK COUNT
;SET UP I-SPACE PAR TO RELOCATE VA
;TO LAST BLOCK IN BANK 0
;IS THIS PAGE 7? (WAS USED
;FOR REFERENCE TO SR0)
;IF NOT, BRANCH
;YES, SET UP PAGE 6 FOR REFERENCES TO SR0
;CHANGE R5 TO POINT TO SR0 THRU PAGE 6
;TURN ON DESTINATION ONLY RELOCATION
;CK BOTTOM PAGE BOUNDARY

;DESTINATION ONLY RELOCATION FAILED
;VA CONTAINED IN R1 FAILED TO RELOCATE
;TO TOP BLOCK OF BANK ZERO
;CK UPPER PAGE BOUNDARY

;DESTINATION ONLY RELOCATION FAILED
;VA CONTAINED IN R2 FAILED TO RELOCATE
;TO TOP BLOCK OF BANK ZERO
;TURN OFF KT11-C
;MAP I-SPACE PAR 1 BLOCK LOWER
;SET UP R1 AND R2 TO REFERENCE
;NEXT VIRTUAL BLOCK
;DECREMENT COUNT OF BLOCKS LEFT
;BRANCH IF NOT DONE WITH THIS PAGE

```

005644	005070	000000		CLR	@(RO)	: IF DONE CLEAR PAR
005650	005367	174132		DEC	PAGES	: DECREMENT COUNT OF PAGES LEFT
005654	001402			BEQ	END20	: BRANCH IF ALL PAGES IN THIS MODE DONE
005656	005720			TST	(RO)+	: MOVE ADDRESS TABLE POINTER TO ADDRESS NEXT PAR
005660	000727			BR	PAG20	: TEST REFERENCES TO NEXT PAGE
005662	026727	174116	002002	END20: CMP	STAPNT, #STAEND	: CHECK FOR ALL MODES TESTED
005670	003673			BLE	STAT20	: IF NOT, BRANCH
005672	005037	177776		CLR	@#PS	: IF DONE, REINITIALIZE
005676	005077	173532		CLR	@SRO	
005702	016727	174102	017700	MOV	SAVEA, #17700	
005710	016727	174076	017776	MOV	SAVEB, #17776	

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

005716	104400			TEST21: SCOPE		
005720	012737	000021	177570	MOV	#21, @#SR	: DISPLAY TEST NUMBER
005726	005037	177776		CLR	@#PS	: INITIALIZE PROCESSOR STATUS
005732	012706	001400		MOV	#KSTACK, SP	: INITIALIZE KERNEL STACK POINTER
005736	004767	010320		JSR	%7, SETUP	: INITIALIZE SRO SR3
005742	026727	174064	000021	CMP	TESTCT, #21	: IS THIS TEST BEING EXECUTED IN THE
005750	001401			BEQ	+.4	: CORRECT SEQUENCE?- BRANCH IF YES
005752	104006			HLT		: TEST EXECUTED OUT OF SEQUENCE- TESTCT
						: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

005754	012767	002000	010716	MOV	#2000, ICOUNT	: RESTORE ITERATION COUNT
005762	004767	010306		JSR	%7, CLRALL	: CLEAR ALL KT11-C REGISTERS
005766	012777	000001	173720	MOV	#1, @KIPAR0	: OFFSET KERNEL I-SPACE PAGE 0
005774	012777	077406	173652	MOV	#77406, @KIPDR0	: BY 1 BLOCK FROM BANK 0
006002	012777	007600	173722	MOV	#7600, @KIPAR7	: MAP KERNEL I-SPACE PAGE 7 TO THE
006010	012777	077406	173654	MOV	#77406, @KIPDR7	: EXTERNAL BANK, RW, 4K
006016	016701	173412		MOV	SRO, R1	: SETUP R1 TO REFERENCE SRO
006022	016746	000054		MOV	DST21A-100, -(SP)	: SAVE CONTENTS OF LOCATIONS WHICH
006026	016746	000052		MOV	DST21B-100, -(SP)	: WOULD BE DESTROYED IF DESTINATION
006032	016746	000050		MOV	DST21C-100, -(SP)	: DOESN'T RELOCATE
006036	005067	000140		CLR	DST21A	: INITIALIZE LOCATIONS TO BE
006042	012767	177777	000134	MOV	#-1, DST21B	: WRITTEN INTO
006050	012767	177777	000130	MOV	#-1, DST21C	
006056	012777	000400	173350	MOV	#400, @SRO	: TURN ON DESTINATION ONLY RELOCATION
006064	022737	176543	005766	CMP	#176543, @#AD21A-100	: COMPARE THE CONTENTS OF AD21A
	006066			AD21A=-.4		: WITH ITSELF, RELOCATED THRU KERNEL
						: I-SPACE 0

006072	001401			BEQ	+.4	
006074	104006			HLT		: DESTINATION ONLY RELOCATION FAILED
						: TO RELOCATE ONLY THE LAST CALCULATION
						: OF THE CMP INSTRUCTION
006076	122737	165432	006000	AD21B=-.4	CMPB	#165432, @#AD21B-100
	006100					: COMPARE THE CONTENTS OF AD21B (LOW BYTE)
006104	001401			BEQ	+.4	: WITH ITSELF, RELOCATED THRU KERNEL
006106	104006			HLT		: I-SPACE 0
						: DESTINATION ONLY RELOCATION
						: FAILED TO RELOCATE ONLY THE FINAL ADDRESS
						: CALCULATION OF THE CMPB INSTRUCTION
006110	012737	077711	006102	MOV	#77711, @#DST21A-100	: EXECUTE REMAINING INSTRUCTIONS - CHECK
006116	005077	000066		CLR	@AD21C	: AFTER KT11-C IS TURNED OFF
						: SHOULD CLEAR DST21B
006122	105037	006106		CLRB	@#DST21C-100	: SHOULD CLEAR LOW BYTE OF DST21C
006126	005011			CLR	@R1	: TURN OFF KT11-C

M02

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 25
 DCKTAB.P11

006130	022767	077711	000044	CMP	#77711,DST21A	;CHECK LOCATION ADDRESSED BY MOV
006136	001401			BEQ	+.4	
006140	104006			HLT		;MOV INSTRUCTION FAILED TO RELOCATE
						;ONLY THE FINAL ADDRESS CALCULATION
006142	005767	000036		TST	DST21B	;CHECK LOCATION ADDRESSED BY CLR
006146	001401			BEQ	+.4	
006150	104006			HLT		;CLR INSTRUCTION FAILED TO RELOCATE
						;CORRECTLY IN DESTINATION ONLY RELOCATION
006152	022767	177400	000026	CMP	#177400,DST21C	;CHECK LOCATION ADDRESSED BY CLRB
006160	001401			BEQ	+.4	
006162	104006			HLT		;CLRB INSTRUCTION FAILED TO RELOCATE
						;CORRECTLY IN DESTINATION ONLY RELOCATION
006164	012667	177716		MOV	(SP)+,DST21C-100	;RESTORE LOCATIONS SAVED IN CASE OF ERROR
006170	012667	177710		MOV	(SP)+,DST21B-100	
006174	012667	177702		MOV	(SP)+,DST21A-100	
006200	000404			BR	TEST22	
006202	000000					
006204	000000					
006206	000000					
006210	006104					

DST21A: 0
 DST21B: 0
 DST21C: 0
 AD21C: DST21B-100

;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 I-SPACE
 ;TEST BY LOADING ONE OF THE VALUES TO BE INPUT TO THE ADDERS INTO KERNEL I-SPACE PAR 1
 ;THE SECOND VALUE INPUT TO THE ADDERS IS THE LOWER 13 BITS OF THE
 ;VIRTUAL ADDRESS (THE UPPER 3 BITS SELECT PAGE 1)
 ;CHECK THE RESULTING PHYSICAL ADDRESS BY READING THE CONTENTS OF THE LOCATION,
 ;AND IF THIS IS CORRECT, BY WRITING INTO THE LOCATION
 ;NOTE THAT THIS TEST INCLUDES CHECKS OF ADDRESS WRAP AROUND

006212	104400			TEST22:	SCOPE	
006214	012737	000022	177570	MOV	#22,@#SR	;DISPLAY TEST NUMBER
006222	005037	177776		CLR	@#PS	;INITIALIZE PROCESSOR STATUS
006226	012706	001400		MOV	#KSTACK,SP	;INITIALIZE KERNEL STACK POINTER
006232	004767	010024		JSR	%7,SETUP	;INITIALIZE SR0,SR3
006236	026727	173570	000022	CMP	TESTCT,#22	;IS THIS TEST BEING EXECUTED IN THE
006244	001401			BEQ	+.4	;CORRECT SEQUENCE?- BRANCH IF YES
006246	104006			HLT		;TEST EXECUTED OUT OF SEQUENCE- TESTCT
						;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

006250	004767	010020		JSR	%7,CLRALL	;CLEAR ALL KT11-C REGISTERS
006254	012777	077406	173372	MOV	#77406,@KIPDR0	;MAP KERNEL 0 TO BANK 0, 4K, RW
006262	012777	077406	173366	MOV	#77406,@KIPDR1	;MAKE KERNEL 1 RW, 4K
006270	012777	077406	173374	MOV	#77406,@KIPDR7	;MAP KERNEL 7 RW, 4K, EXTERNAL BANK
006276	012777	007600	173426	MOV	#7600,@KIPAR7	

;CHECK VIRTUAL ADDRESS OF 0 ADDED TO PAR OF -1 (FOR BIT POSITIONS
 ;RELEVANT TO THE ADDERS ONLY. VA 20076, PAR 7777, RESULTING PA 777776)

006304	012777	007777	173404	MOV	#7777,@KIPAR1	;SET PAR TO -1
006312	012737	030000	177776	MOV	#30000,@#PS	;SET UP LOCATION TO BE REFERENCED
006320	012777	000400	173106	MOV	#400,@SR0	;TURN ON DESTINATION ONLY PAGING
006326	022737	030000	020076	ADR22A: CMP	#30000,@#20076	;IS PA 777776 BEING REFERENCED?
006334	001012			BNE	ERR22A	;BRANCH ON FAILURE
006336	005037	020076		ADR22A: CLR	@#20076	;CLEAR PA 777776 THRU KERNEL PAGE 1
006342	005077	173066		CLR	@SR0	;TURN OFF KT11-C
006346	032737	177740	177776	BIT	#177740,@#PS	;CHECK TO SEE IF CORRECT LOCATION


```

006602 012737 000002 037700 AD22D: MOV      #2, @#37700      ;WRITE SAME LOCATION
006610 005077 172620                CLR      @SR0      ;TURN OFF KT11-C
006614 022737 000002 000000                CMP      #2, @#0    ;CHECK LOCATION WHICH SHOULD HAVE
006622 001401                BEQ      .+4        ;BEEN REFERENCED
006624 104006                HLT                     ;RELOCATION FAILED IN MOV AT AD22D
006626 000406                BR       CNT22E      ;GO TO NEXT CHECK
006630 005077 172600                ERR22D: CLR      @SR0      ;TURN OFF KT11-C
006634 104006                HLT                     ;RELOCATION FAILED IN THE COMPARE
                                ;AT LOCATION ADR22D
006636 012737 000002 000000                MOV      #2, @#0    ;RESTORE LOCATION REFERENCED

                                ;CHECK VIRTUAL ADDRESS -1 (BITS 6 - 12) ADDED TO PAR OF -1
                                ;SHOULD GIVE RESULTING PA 17600
                                ;NOTE THAT THIS IS A CASE OF ADDRESS WRAP AROUND
006644 012777 007777 173044 CNT22E: MOV      #7777, @KIPAR1 ;SET UP PAR TO -1
006652 013746 017600                MOV      @#17600, -(SP) ;SAVE CONTENTS OF LOCATION TO BE
                                ;REFERENCED
006656 012737 076767 017600                MOV      #76767, @#17600 ;LOAD LOCATION TO BE REFERENCED
006664 012777 000400 172542                MOV      #400, @SR0    ;TURN ON DESTINATION ONLY PAGING
006672 022737 076767 037700 ADR22E: CMP      #76767, @#37700 ;READ LOCATION (VA=-1 (BITS 6 - 12) ,PAR=-1)
                                ;SHOULD GIVE PA 17600 (THRU KERNEL
                                ;PAGE 1)
006700 001011                BNE      ERR22E      ;BRANCH ON FAILURE
006702 005037 037700                AD22E: CLR      @#37700 ;WRITE SAME LOCATION
006706 005077 172522                CLR      @SR0      ;TURN OFF KT11-C
006712 005737 017600                TST      @#17600    ;CHECK TO SEE IF CORRECT LOCATION
006716 001401                BEQ      .+4        ;WAS CLEARED (HIGH BYTE)
006720 104006                HLT                     ;RELOCATION FAILED IN THE CLR AT AD22E
006722 000403                BR       END22E      ;TURN OFF KT11-C
006724 005077 172504                ERR22E: CLR      @SR0      ;RELOCATION FAILED IN THE COMPARE AT
006730 104006                HLT                     ;LOCATION ADR22E
006732 012637 017600                END22E: MOV      (SP)+, @#17600 ;RESTORE LOCATION REFERENCED

                                ;SHOW THAT SETTING SRO<0> TURNS ON FULL RELOCATION
                                ;SHOW THAT ALL ADDRESS CALCULATIONS ARE RELOCATED
                                ;SHOW THAT INIT CLEARS SRO<0> AND TURNS OFF RELOCATION
006736 104400                TEST23: SCOPE
006740 012737 000023 177570                MOV      #23, @#SR   ;DISPLAY TEST NUMBER
006746 005037 177776                CLR      @#PS        ;INITIALIZE PROCESSOR STATUS
006752 012706 001400                MOV      #KSTACK, SP ;INITIALIZE KERNEL STACK POINTER
006756 004767 007300                JSR      %7, SETUP   ;INITIALIZE SRO, SR3
006762 026727 173044 000023                CMP      TESTCT, #23 ;IS THIS TEST BEING EXECUTED IN THE
006770 001401                BEQ      .+4        ;CORRECT SEQUENCE?- BRANCH IF YES
006772 104006                HLT                     ;TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

006774 012767 000010 007676                MOV      #10, ICOUNT ;DROP ITERATION COUNT
007002 004767 007266                JSR      %7, CLRALL  ;INITIALLY CLEAR ALL KT11-C REGISTERS
007006 012777 000001 172700                MOV      #1, @KIPAR0 ;MAP KERNEL I-SPACE PAGE 0 TO
007014 012777 077406 172632                MOV      #77406, @KIPDR0 ;BANK 0 OFFSET BY 1 BLOCK
007022 012777 007600 172702                MOV      #7600, @KIPAR7 ;MAP KERNEL I-SPACE PAGE 7 TO
007030 012777 077406 172634                MOV      #77406, @KIPDR7 ;THE EXTERNAL BANK
007036 012767 052525 010646                MOV      #52525, DESTAD ;INITIALIZE LOCATION TO BE REFERENCED
007044 162706 000100                SUB      #100, SP    ;ALTER STACK POINTER DUE TO BANK 0 OFFSET
007050 005277 172360                INC      @SR0        ;TURN ON RELOCATION

```


DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 29
DCKTAB.P11

```

007406 007406 000001 172020 ADD24A: =ADD24+100 BIC #1,SR0
;WHEN KT11-C IS TURNED ON, SHOULD
;RELOCATE FETCH TO HERE - TURN
;OFF KT11-C VIA BIC OF SR0<0>
;KT11-C STILL RELOCATING AFTER
;BIC OF SR0<0>

007414 000000 HALT
007416 000005 RESET
007420 000777 BR

;SHOW THAT A REFERENCE TO A NON-RESIDENT PAGE
;WILL ABORT TO THE KT11-C ABORT VECTOR ADDRESS (250)
;WITH BIT 15 OF SR0 SET. SR0, SR1, AND SR2 ARE CHECKED FOR
;THE CORRECT VALUES, AS ARE KIPDR0 AND KIPDR1
;SHOW THAT BIT 15 OF SR0 CAN BE CLEARED AND THAT
;SR2 IS READ ONLY

007422 104400 TEST25: SCOPE
007424 012737 000025 177570 MOV #25,@SR
007432 005037 177776 CLR @PS
007436 012706 001400 MOV #KSTACK,SP
007442 004767 006614 JSR %7,SETUP
007446 026727 172360 000025 CMP TESTCT,#25
007454 001401 BEQ .+4
007456 104006 HLT

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

007460 004767 006610 JSR %7,CLRALL
007464 012777 077406 172200 MOV #77406,@KIPDR7
007472 012777 007600 172232 MOV #7600,@KIPAR7
007500 012777 077406 172146 MOV #77406,@KIPDR0
007506 012777 007542 172300 MOV #INT25,@KTVEC
007514 005077 172276 CLR @KTSTA
007520 012704 020000 MOV #20000,R4

;CLEAR ALL KT11-C REGISTERS
;MAP KERNEL PAGE 7 I-SPACE TO
;THE EXTERNAL BANK
;MAP KERNEL 0 TO BANK 0, RW, 4K
;SETUP ABORT RETURN

;USE R4 TO REFERENCE NR KERNEL
;ONE I-SPACE
;TURN ON KT11-C
;REFERENCE NR KERNEL 1 I-SPACE
;SHOULD HAVE ABORTED ALREADY
;TURN OFF KT11-C

007524 005277 171704 INC @SR0
007530 005724 ADR25: TST (R4)+
007532 000000 ADR25A: HALT
007534 005077 171674 CLR @SR0
007540 000457 BR DONE25
007542 017701 171666 INT25: MOV @SR0,R1
007546 005277 171662 INC @SR0
007552 022701 100003 CMP #100003,R1
007556 001401 BEQ .+4
007560 104006 HLT

;SAVE CONTENTS OF SR0
;TURN OFF KT11-C
;CHECK SAVED CONTENTS OF SR0

;SR0 INCORRECT AFTER NR ABORT
;(SEE SAVED CONTENTS IN R1)
;CHECK SR1

007562 022777 000024 171650 CMP #24,@SR1
007570 001401 BEQ .+4
007572 104006 HLT

;SR1 INCORRECT - SHOULD HAVE
;RECORDED AUTOINCREMENT OF R4.
;CK SR2

007574 022777 007530 171642 CMP #ADR25,@SR2
007602 001402 BEQ .+6
007604 104006 HLT
007606 000407 BR .+20
007610 005077 171630 CLR @SR2
007614 022777 007530 171622 CMP #ADR25,@SR2
007622 001401 BEQ .+4

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

```

```

007624 104006          HLT
007626 022777 077506 172020  CMP      #77506, @KIPDR0
007634 001401          BEQ      .+4
007636 104006          HLT
                                ;SR2 NOT READ ONLY
                                ;CHECK KERNEL 0 I-SPACE PDR
007640 005777 172012    TST      @KIPDR1
007644 001401          BEQ      .+4
                                ;KERNEL I-SPACE PDR 0 INCORRECT
                                ;W BIT SHOULD HAVE BEEN SET BY THE STACK WRITE
                                ;CHECK KERNEL 1 I-SPACE PDR
007646 104006          HLT
007650 021627 007532    CMP      (R6), @ADR25A
007654 001401          BEQ      .+4
                                ;KERNEL I-SPACE PDR 1 INCORRECT
                                ;CHECK VALUE PUSHED ON STACK
007656 104006          HLT
007660 022626          CMP      (R6)+, (R6)+
007662 005077 171546    CLR      @SR0
007666 032777 100000 171540  BIT      @100000, @SR0
007674 001401          BEQ      .+4
                                ;INCORRECT VALUE ON STACK
                                ;RESTORE STACK POINTER
                                ;CLEAR BIT 15
007676 104006          HLT
007700 005077 172112    CLR      @KTSTA
007704 016777 172106 172102  DONE25: MOV     @KTSTA, @KTVEC
                                ;BIT 15 OF SR0 DIDN'T CLEAR
                                ;CHANGE TRAP VECTOR TO CAUSE A
                                ;HALT ON A FALSE TRAP

                                ;SHOW THAT THE D-SPACE ENABLE BITS WORK CORRECTLY
007712 104400          ;TEST26: SCOPE
007714 012737 000026 177570  MOV     #26, @SR
007722 005037 177776    CLR     @PS
007726 012706 001400    MOV     @KSTACK, SP
007732 004767 006324    JSR     %7, SETUP
007736 026727 172070 000026  CMP     TESTCT, #26
007744 001401          BEQ     .+4
007746 104006          HLT
                                ;DISPLAY TEST NUMBER
                                ;INITIALIZE PROCESSOR STATUS
                                ;INITIALIZE KERNEL STACK POINTER
                                ;INITIALIZE SR0, SR3
                                ;IS THIS TEST BEING EXECUTED IN THE
                                ;CORRECT SEQUENCE?- BRANCH IF YES
                                ;TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                ;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

007750 004767 006320    JSR     %7, CLRALL
007754 012767 000010 006716  MOV     #10, ICOUNT
007762 012777 077406 171664  MOV     #77406, @KIPDR0
007770 012777 077406 171676  MOV     #77406, @KDPDR0
007776 012777 077406 171550  MOV     #77406, @SIPDR0
010004 012777 077406 171442  MOV     #77406, @UIPDR0
010012 012777 077406 171436  MOV     #77406, @UIPDR1
010020 012777 077406 171530  MOV     #77406, @SIPDR1
010026 012777 077406 171622  MOV     #77406, @KIPDR1
010034 012777 077406 171630  MOV     #77406, @KIPDR7
010042 012777 077406 171522  MOV     #77406, @SIPDR7
010050 012777 077406 171414  MOV     #77406, @UIPDR7
010056 012777 077406 171626  MOV     #77406, @KDPDR7
010064 012777 077406 171520  MOV     #77406, @SDPDR7
010072 012777 077406 171412  MOV     #77406, @UDPDR7
010100 012777 007600 171624  MOV     #7600, @KIPAR7
010106 012777 007600 171516  MOV     #7600, @SIPAR7
010114 012777 007600 171410  MOV     #7600, @UIPAR7
010122 012777 007600 171622  MOV     #7600, @KDPAR7
010130 012777 007600 171514  MOV     #7600, @SDPAR7
010136 012777 007600 171406  MOV     #7600, @UDPAR7
010144 005077 171646    CLR     @KTSTA
                                ;INITIALLY CLEAR ALL KT11-C REGISTERS
                                ;MAKE KERNEL 0 I-SPACE RW
                                ;MAKE KERNEL 0 D-SPACE RW, 4K
                                ;MAKE SUPERVISOR 0 I SPACE RW
                                ;MAKE USER 0 I SPACE RW
                                ;MAKE PAGE 1 OF EACH I-SPACE
                                ;RW, 4K, BANK 0
                                ;MAP PAGE 7 I AND D SPACES
                                ;OF ALL MODES 4K, RW, EXTERNAL
                                ;INITIALIZE KT11-C ABORT RETURN
                                ;STATUS
;CHECK D-SPACE REFERENCES WITH NO D-SPACE ENABLE BITS SET

```



```

;RESET
010272 000403          BR      E26C+2
010274 104010          RT26C: KTOFF
                                ;TURN OFF KT11-C VIA TRAPPING
                                ;TO KERNEL AND EXECUTING A RESET
010276 104006          HLT      ;REFERENCE TO USER D-SPACE WITH
                                ;NO D-SPACE ENABLE SET
                                ;REFERENCED USER D-SPACE INSTEAD
                                ;OF USER I-SPACE
                                ;RESTORE STACK POINTER

010300 022626          E26C:  CMP      (R6)+,(R6)+
                                ;CHECK D-SPACE REFERENCES WITH KERNEL D-SPACE ENABLE SET

010302 012777 000004 171140      MOV      #4,JSR3          ;ENABLE KERNEL D-SPACE
                                ;SHOW THAT A REFERENCE TO KERNEL D-SPACE WITH KERNEL D-SPACE ENABLE
                                ;SET WILL ACTUALLY REFERENCE KERNEL D-SPACE
010310 012777 010342 171476      MOV      #RT26D,KTVEC    ;SETUP ABORT RETURN
010316 012737 000000 177776      MOV      #0,PS          ;SET MODE TO KERNEL
010324 005277 171104          INC      JSR0            ;TURN ON KT11-C
010330 005737 020000          TST      #20000        ;D-SPACE REFERENCE - SHOULD ABORT
010334 104010          KTOFF   ;TURN OFF KT11-C VIA TRAPPING TO
                                ;KERNEL AND EXECUTING A RESET
010336 104006          HLT      ;REFERENCE TO KERNEL D-SPACE WITH
                                ;KERNEL D-SPACE ENABLE SET
                                ;DID NOT USE KERNEL D-SPACE

010340 000402          BR      E26D+2
010342 104010          RT26D: KTOFF      ;TURN OFF KT11-C VIA TRAPPING
                                ;TO KERNEL AND EXECUTING A RESET
010344 022626          E26D:  CMP      (R6)+,(R6)+    ;RESTORE STACK POINTER

010346 012777 000004 171074      MOV      #4,JSR3          ;ENABLE KERNEL D-SPACE
                                ;SHOW THAT A REFERENCE TO SUPERVISOR D-SPACE WITH KERNEL D-SPACE ENABLE
                                ;SET WILL NOT ACTUALLY REFERENCE SUPERVISOR D-SPACE
010354 012777 010404 171432      MOV      #RT26E,KTVEC    ;SETUP ABORT RETURN IN CASE
010362 012737 040000 177776      MOV      #40000,PS       ;SET MODE TO SUPERVISOR
010370 005277 171040          INC      JSR0            ;TURN ON KT11-C
010374 005737 020000          TST      #20000        ;D-SPACE REFERENCE IF D-SPACE ENABLED
                                ;D-SPACE IS NR, I-SPACE IS RW
                                ;NO ABORT SHOULD OCCUR SINCE
                                ;CORRESPONDING D-SPACE ENABLE
                                ;IS NOT SET
                                ;TURN OFF KT11-C VIA TRAPPING
                                ;TO KERNEL AND EXECUTING A
010400 104010          KTOFF   ;RESET

010402 000403          BR      E26E+2
010404 104010          RT26E: KTOFF
                                ;TURN OFF KT11-C VIA TRAPPING
                                ;TO KERNEL AND EXECUTING A RESET
010406 104006          HLT      ;REFERENCE TO SUPERVISOR D-SPACE WITH
                                ;KERNEL D-SPACE ENABLE SET
                                ;REFERENCED SUPERVISOR D-SPACE INSTEAD
                                ;OF SUPERVISOR I-SPACE
                                ;RESTORE STACK POINTER

010410 022626          E26E:  CMP      (R6)+,(R6)+

010412 012777 000004 171030      MOV      #4,JSR3          ;ENABLE KERNEL D-SPACE
                                ;SHOW THAT A REFERENCE TO USER D-SPACE WITH KERNEL D-SPACE ENABLE
                                ;SET WILL NOT ACTUALLY REFERENCE USER D-SPACE
010420 012777 010450 171366      MOV      #RT26F,KTVEC    ;SETUP ABORT RETURN IN CASE

```


DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 34
 DCKTAB.P11

```

010560 000402          BR      E26H+2
010562 104010          RT26H: KTOFF          ;TURN OFF KT11-C VIA TRAPPING
                                           ;TO KERNEL AND EXECUTING A RESET
010564 022626          E26H:  CMP      (R6)+,(R6)+      ;RESTORE STACK POINTER
010566 012777 000002 170654          MOV      #2,2SR3          ;ENABLE SUPERVISOR D-SPACE
                                           ;SHOW THAT A REFERENCE TO USER D-SPACE WITH SUPERVISOR D-SPACE ENABLE
                                           ;SET WILL NOT ACTUALLY REFERENCE USER D-SPACE
010574 012777 010624 171212          MOV      #RT26I,2KTVEC      ;SETUP ABORT RETURN IN CASE
010602 012737 140000 177776          MOV      #140000,2#PS      ;SET MODE TO USER
010610 005277 170620          INC      2SR0          ;TURN ON KT11-C
010614 005737 020000          TST      2#20000          ;D-SPACE REFERENCE IF D-SPACE ENABLED
                                           ;D-SPACE IS NR, I-SPACE IS RW
                                           ;NO ABORT SHOULD OCCUR SINCE
                                           ;CORRESPONDING D-SPACE ENABLE
                                           ;IS NOT SET
010620 104010          KTOFF          ;TURN OFF KT11-C VIA TRAPPING
                                           ;TO KERNEL AND EXECUTING A
                                           ;RESET
010622 000403          BR      E26I+2
010624 104010          RT26I: KTOFF          ;TURN OFF KT11-C VIA TRAPPING
                                           ;TO KERNEL AND EXECUTING A RESET
010626 104006          HLT          ;REFERENCE TO USER D-SPACE WITH
                                           ;SUPERVISOR D-SPACE ENABLE SET
                                           ;REFERENCED USER D-SPACE INSTEAD
                                           ;OF USER I-SPACE
010630 022626          E26I:  CMP      (R6)+,(R6)+      ;RESTORE STACK POINTER
                                           ;CHECK D-SPACE REFERENCES WITH USER D-SPACE ENABLE SET
010632 012777 000001 170610          MOV      #1,2SR3          ;ENABLE USER D-SPACE
                                           ;SHOW THAT A REFERENCE TO KERNEL D-SPACE WITH USER D-SPACE ENABLE
                                           ;SET WILL NOT ACTUALLY REFERENCE KERNEL D-SPACE
010640 012777 010670 171146          MOV      #RT26J,2KTVEC      ;SETUP ABORT RETURN IN CASE
010646 012737 000000 177776          MOV      #0,2#PS          ;SET MODE TO KERNEL
010654 005277 170554          INC      2SR0          ;TURN ON KT11-C
010660 005737 020000          TST      2#20000          ;D-SPACE REFERENCE IF D-SPACE ENABLED
                                           ;D-SPACE IS NR, I-SPACE IS RW
                                           ;NO ABORT SHOULD OCCUR SINCE
                                           ;CORRESPONDING D-SPACE ENABLE
                                           ;IS NOT SET
010664 104010          KTOFF          ;TURN OFF KT11-C VIA TRAPPING
                                           ;TO KERNEL AND EXECUTING A
                                           ;RESET
010666 000403          BR      E26J+2
010670 104010          RT26J: KTOFF          ;TURN OFF KT11-C VIA TRAPPING
                                           ;TO KERNEL AND EXECUTING A RESET
010672 104006          HLT          ;REFERENCE TO KERNEL D-SPACE WITH
                                           ;USER D-SPACE ENABLE SET
                                           ;REFERENCED KERNEL D-SPACE INSTEAD
                                           ;OF KERNEL I-SPACE
010674 022626          E26J:  CMP      (R6)+,(R6)+      ;RESTORE STACK POINTER
010676 012777 000001 170544          MOV      #1,2SR3          ;ENABLE USER D-SPACE
                                           ;SHOW THAT A REFERENCE TO SUPERVISOR D-SPACE WITH USER D-SPACE ENABLE
                                           ;SET WILL NOT ACTUALLY REFERENCE SUPERVISOR D-SPACE

```



```

011110 012701 004754          MOV      #DATA16,R1          ;SETUP R1 TO REFERENCE KERNEL 0
011114 012777 007600 170630  MOV      #7600, @KDPAR7     ;MAP KERNEL 7 D-SPACE TO THE
011122 012777 077406 170562  MOV      #77406, @KDPDR7   ;EXTERNAL BANK
011130 012777 000004 170312  MOV      #4, @SR3          ;ENABLE KERNEL D-SPACE
011136 016702 170272          MOV      SRO,R2           ;SETUP R2 TO ADDRESS SRO
011142 012777 000400 170264  MOV      #400, @SRO        ;TURN ON DESTINATION ONLY RELOCATION
011150 005012          CLR      @R2              ;CLEAR SRO THRU KERNEL 7 D-SPACE
011152 021111          CMP      @R1, @R1         ;SHOW THAT KT11-C IS OFF-IF STILL ON,
011154 001401          BEQ     .+4              ;THE CMP WILL REFERENCE 2 LOCATIONS
                                ;WHICH AREN'T EQUAL
                                ;KT11-C IS STILL ON
011156 000000          HALT     .               ;MAKE SURE THAT KT11-C IS OFF
011160 000005          RESET     .             ;REINITIALIZE SR3
011162 005077 170262          CLR      @SR3           ;REINITIALIZE PROCESSOR STATUS
011166 005037 177776          CLR      @#PS

;SHOW THAT A DATO OF 0 TO BIT 8, SRO THRU SUPERVISOR PAGE 7 D-SPACE
;WILL TURN OFF DESTINATION ONLY RELOCATION
011172 004767 005076          JSR      %7, CLRALL       ;INITIALLY CLEAR ALL KT11-C REGISTERS
011176 012777 000001 170430  MOV      #1, @SDPAR0      ;MAP SUPERVISOR 0 D-SPACE TO BANK 0,
011204 012777 077406 170362  MOV      #77406, @SDPDR0   ;RW, OFFSET BY 1 BLOCK
011212 012701 004754          MOV      #DATA16,R1       ;SETUP R1 TO REFERENCE SUPERVISOR 0
011216 012777 007600 170426  MOV      #7600, @SDPAR7     ;MAP SUPERVISOR 7 D-SPACE TO THE
011224 012777 077406 170360  MOV      #77406, @SDPDR7   ;EXTERNAL BANK
011232 012777 000002 170210  MOV      #2, @SR3          ;ENABLE SUPERVISOR D-SPACE
011240 012737 040000 177776  MOV      #40000, @#PS      ;SET MODE TO SUPERVISOR
011246 016702 170162          MOV      SRO,R2           ;SETUP R2 TO ADDRESS SRO
011252 012777 000400 170154  MOV      #400, @SRO        ;TURN ON DESTINATION ONLY RELOCATION
011260 005012          CLR      @R2              ;CLEAR SRO THRU SUPERVISOR 7 D-SPACE
011262 021111          CMP      @R1, @R1         ;SHOW THAT KT11-C IS OFF - IF STILL ON
011264 001401          BEQ     .+4              ;THE CMP WILL REFERENCE 2 LOCATIONS
                                ;WHICH AREN'T EQUAL
                                ;KT11-C IS STILL ON
011266 000777          BR       .               ;MAKE SURE KT11-C IS OFF
011270 000005          RESET     .             ;REINITIALIZE SR3
011272 005077 170152          CLR      @SR3           ;REINITIALIZE PROCESSOR STATUS
011276 005037 177776          CLR      @#PS

;SHOW THAT A DATO OF 0 TO BIT 8, SRO THRU USER PAGE 7 D-SPACE
;WILL TURN OFF DESTINATION ONLY PAGING
011302 004767 004766          JSR      %7, CLRALL       ;INITIALLY CLEAR ALL KT11-C REGISTERS
011306 012777 000001 170220  MOV      #1, @UDPAR0      ;MAP USER 0 D-SPACE TO BANK 0, RW,
011314 012777 077406 170152  MOV      #77406, @UDPDR0   ;OFFSET BY 1 BLOCK
011322 012701 004754          MOV      #DATA16,R1       ;SETUP R1 TO REFERENCE USER 0
011326 012777 007600 170216  MOV      #7600, @UDPAR7     ;MAP USER 7 D-SPACE TO THE
011334 012777 077406 170150  MOV      #77406, @UDPDR7   ;EXTERNAL BANK
011342 012777 000001 170100  MOV      #1, @SR3          ;ENABLE USER D-SPACE
011350 012737 140000 177776  MOV      #140000, @#PS     ;SET MODE TO USER
011356 016702 170052          MOV      SRO,R2           ;SETUP R2 TO ADDRESS SRO
011362 012777 000400 170044  MOV      #400, @SRO        ;TURN ON DESTINATION ONLY RELOCATION
011370 005012          CLR      @R2              ;CLEAR SRO THRU USER 7 D-SPACE
011372 021111          CMP      @R1, @R1         ;SHOW THAT KT11-C IS OFF - IF STILL ON, THE
011374 001401          BEQ     .+4              ;CMP WILL REFERENCE 2 LOCATIONS WHICH AREN'T
                                ;EQUAL
                                ;KT11-C IS STILL ON
011376 000777          BR       .               ;MAKE SURE KT11-C IS OFF
011400 000005          RESET     .             ;REINITIALIZE SR3
011402 005077 170042          CLR      @SR3           ;REINITIALIZE PROCESSOR STATUS
011406 005037 177776          CLR      @#PS

```

; SHOW THAT WITH ALL D-SPACES ENABLED ALL BLOCK BOUNDARY REFERENCES
 ; REFERENCE THE CORRECT D-SPACE PAR AND RELOCATE CORRECTLY
 ; USE DESTINATION ONLY RELOCATION AND ENABLE D-SPACE
 ; MAP ALL I-SPACE PAGES NON-RESIDENT
 ; MAP ALL D-SPACE PAGES RESIDENT READ WRITE

RO - POINTS TO THE ADDRESS OF THE CURRENT D-SPACE PAR IN THE ADDRESS TABLE
 R1 - VIRTUAL ADDRESS BEING USED TO REFERENCE START OF BLOCK
 R2 - VIRTUAL ADDRESS BEING USED TO REFERENCE END OF BLOCK
 R3 - POINTS TO EXPECTED CONTENTS OF START OF BLOCK
 R4 - POINTS TO EXPECTED CONTENTS OF END OF BLOCK
 R5 - USED TO REFERENCE SR0 TO TURN OFF DESTINATION ONLY RELOCATION

011412 104400
 011414 012737 000030 177570
 011422 005037 177776
 011426 012706 001400
 011432 004767 004624
 011436 026727 170370 000030
 011444 001401
 011446 104006

TEST30: SCOPE
 MOV #30, @#SR
 CLR @#PS
 MOV #KSTACK, SP
 JSR %7, SETUP
 CMP TESTCT, #30
 BEQ .+4
 HLT

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

011450 004767 004620
 011454 004767 004750
 011460 013767 017700 170322
 011466 013767 017776 170316
 011474 012737 123456 017700
 011502 012737 134567 017776
 011510 012777 000007 167732
 011516 012703 001414
 011522 012704 001416

JSR %7, CLRALL
 JSR %7, RWDSP
 MOV @#17700, SAVEA
 MOV @#17776, SAVEB
 MOV #123456, @#17700
 MOV #134567, @#17776
 MOV #7, @SR3
 MOV #K123, R3
 MOV #K134, R4

; INITIALIZE KT11-C - CLEAR ALL REGISTERS
 ; MAKE ALL D SPACE PAGES RW, BANK 0, 4K
 ; SAVE CONTENTS OF LOCATIONS TO BE USED
 ; AS START AND END OF PHYSICAL BLOCK
 ; SET UP LOCATIONS TO BE REFERENCED

011526 012767 000100 005144
 011534 012737 040000 177776
 011542 012706 001000
 011546 012737 140000 177776
 011554 012706 000400
 011560 005037 177776
 011564 012767 001770 170212

MOV #100, ICOUNT
 MOV #40000, @#PS
 MOV #SSTACK, SP
 MOV #140000, @#PS
 MOV #USTACK, SP
 CLR @#PS
 MOV #STATAB, STAPNT

; ENABLE ALL D-SPACES
 ; LOAD R3 AND R4 WITH ADDRESSES OF
 ; LOCATIONS CONTAINING EXPECTED CONTENTS
 ; OF START AND END OF BLOCK
 ; THIS ALLOWS USING THE SAME INSTRUCTIONS
 ; AS HAVE ALREADY BEEN SHOWN TO WORK
 ; CORRECTLY IN DESTINATION ONLY RELOCATION
 ; CHANGE ITERATION COUNT
 ; CHANGE TO SUPERVISOR
 ; SET UP SUPERVISOR STACK POINTER
 ; CHANGE TO USER
 ; SET UP USER STACK POINTER
 ; RETURN TO KERNEL
 ; SET UP TO REFERENCE MODE TABLE
 ; THIS TABLE CONTAINS THE FIRST
 ; ADDRESSES OF THE TABLES OF PAR AND PDR
 ; ADDRESSES FOR EACH MODE, AND THE
 ; VALUE OF THE PROCESSOR STATUS FOR EACH
 ; MODE.

011572 017700 170206
 011576 062700 000060
 011602 062767 000002 170174
 011610 017737 170170 177776
 011616 062767 000002 170160

STATEL: MOV @STAPNT, RO
 ADD #60, RO
 ADD #2, STAPNT
 MOV @STAPNT, @#PS
 ADD #2, STAPNT

; PICK UP ADDRESS OF START OF
 ; ADDRESS TABLE FOR NEW MODE
 ; MOVE POINTER TO ADDRESS VALUE TO
 ; LOAD INTO PROCESSOR STATUS
 ; SET PROCESSOR STATUS TO NEW MODE
 ; MOVE POINTER TO ADDRESS VALUE
 ; FOR NEXT MODE

```

011624 012767 000010 170154      MOV      #8, PAGES      ;SETUP COUNTER OF PAGES LEFT TO TEST
011632 012770 007600 000016      MOV      #7600, @16(R0) ;SET UP RELOCATED REFERENCE TO SR0
011640 016705 167570                MOV      SR0, R5        ;USED TO TURN DESTINATION ONLY
                                ;RELOCATION OFF
011644 005001                CLR      R1            ;INITIALIZE R1 TO CONTAIN VA OF
                                ;START OF PAGE
011646 012702 000076                MOV      #76, R2       ;INITIALIZE R2 TO VA OF END OF PAGE
011652 012767 000200 170154  PAGEL:  MOV      #128, BLOCKS  ;SET UP BLOCK COUNT
011660 012770 000177 000000      MOV      #177, @ (R0)  ;SET UP D-SPACE PAR TO RELOCATE VA TO LAST BLOCK
011666 022767 000001 170112      CMP      #1, PAGES     ;IS THIS PAGE 7? (WAS USED
                                ;FOR REFERENCE TO SR0)
011674 001005                BNE     BLKL          ;IF NOT, BRANCH
011676 012770 007600 177776      MOV      #7600, @-2(R0); YES, SET UP PAGE 6 FOR REFERENCES TO SR0
011704 042705 020000                BIC     #20000, R5     ;CHANGE R5 TO POINT TO SR0 THRU PAGE 6
011710 012777 000400 167516  BLKL:  MOV      #400, @SR0  ;TURN ON DESTINATION ONLY PAGING
011716 021311                CMP     @R3, @R1      ;CK BOTTOM BLOCK BOUNDARY
011720 001401                BEQ     .+4
011722 000000                HALT
                                ;DESTINATION ONLY RELOCATION FAILED
                                ;VA CONTAINED IN R1 FAILED TO RELOCATE
                                ;TO TOP BLOCK OF BANK ZERO
011724 021412                CMP     @R4, @R2     ;CK UPPER BLOCK BOUNDARY
011726 001401                BEQ     .+4
011730 000000                HALT
                                ;DESTINATION ONLY RELOCATION FAILED
                                ;VA CONTAINED IN R2 FAILED TO RELOCATE
                                ;TO TOP BLOCK OF BANK ZERO
011732 005015                CLR     @R5          ;TURN OFF KT11-C
011734 005370 000000      DEC     @ (R0)        ;MAP D-SPACE PAR 1 BLOCK LOWER
011740 062701 000100      ADD     #100, R1      ;SET UP R1 AND R2 TO REFERENCE
011744 062702 000100      ADD     #100, R2      ;NEXT VIRTUAL BLOCK
011750 005367 170060      DEC     BLOCKS       ;DECREMENT COUNT OF BLOCKS LEFT
011754 001355                BNE     BLKL          ;BRANCH IF NOT DONE WITH THIS PAGE
011756 005070 000000      CLR     @ (R0)        ;IF DONE WITH PAGE, CLEAR PAR
011762 005367 170020      DEC     PAGES        ;DECREMENT COUNT OF PAGES LEFT
011766 001402                BEQ     ENDPAG        ;BRANCH IF ALL PAGES IN THIS MODE DONE
011770 005720                TST     (R0)+         ;MOVE ADDRESS TABLE POINTER TO ADDRESS NEXT PAR
011772 000727                BR     PAGEL         ;TEST REFERENCES TO NEXT PAGE
011774 026727 170004 002002  ENDPAG:  CMP     STAPNT, #STAEND ;CHECK FOR ALL MODES TESTED
012002 003673                BLE     STATEL        ;IF NOT, BRANCH
012004 005037 177776                CLR     @#PS          ;IF DONE, REINITIALIZE
012010 005077 167420                CLR     @SR0
012014 016727 167770 017700      MOV     SAVEA, #17700
012022 016727 167764 017776      MOV     SAVEB, #17776

```

```

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

```

```

;FIRST TEST ALL I-SPACE PDR'S
;ALL D SPACES ARE DISABLED
TEST31: SCOPE

```

```

012030 104400                TEST31:  MOV     #31, @#SR      ;DISPLAY TEST NUMBER
012032 012737 000031 177570      CLR     @#PS          ;INITIALIZE PROCESSOR STATUS
012040 005037 177776                MOV     #KSTACK, SP   ;INITIALIZE KERNEL STACK POINTER
012044 012706 001400

```

012050	004767	004206		JSR	%7, SETUP	: INITIALIZE SRO, SR3
012054	026727	167752	000031	CMP	TESTCT, #31	: IS THIS TEST BEING EXECUTED IN THE
012062	001401			BEQ	.+4	: CORRECT SEQUENCE?- BRANCH IF YES
012064	104006			HLT		: TEST EXECUTED OUT OF SEQUENCE- TESTCT
						: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
012066	012767	000400	004604	MOV	#400, ICOUNT	: LOAD ITERATION COUNT
012074	004767	004224		JSR	%7, RWALL	: MAP ALL PAGES 4K, BANK 0, RW
012100	012777	007600	167624	MOV	#7600, @KIPAR7	: MAP KERNEL 7 I-SPACE TO EXTERNAL BANK
012106	012777	007600	167516	MOV	#7600, @SIPAR7	: MAP SUPERVISOR 7 I-SPACE TO EXTERNAL BANK
012114	012777	007600	167410	MOV	#7600, @UIPAR7	: MAP USER 7 I-SPACE TO EXTERNAL BANK
012122	005077	167322		CLR	@SR3	: CLEAR ALL D-SPACE ENABLES
012126	012737	040000	177776	MOV	#40000, @#PS	: SET MODE TO SUPERVISOR
012134	012706	001000		MOV	#SSTACK, R6	: SET UP SUPERVISOR STACK
012140	012737	140000	177776	MOV	#140000, @#PS	: SET MODE TO USER
012146	012706	000400		MOV	#USTACK, R6	: SET UP USER STACK
012152	005037	177776		CLR	@#PS	: REINITIALIZE STATUS TO KERNEL MODE
012156	012704	001454		MOV	#ADRTAB, R4	: LOAD R4 WITH THE ADDRESS OF THE ADDRESS TABLE
012162	012705	000020	LOP31A:	MOV	#20, R5	: R5 IS A COUNTER OF THE PDR'S LEFT TO CHECK
						: IN THE CURRENT SET
012166	022734	077406	LOP31B:	CMP	#77406, @ (R4)+	: CHECK THE CONTENTS OF THE PDR
012172	001401			BEQ	.+4	
012174	104006			HLT		: PDR INCORRECT - W BIT SET OR ANOTHER
						: BIT INCORRECT IN PDR WHOSE ADDRESS
						: IS IN THE LOCATION POINTED TO BY R4
						: TEST ALL PDR'S IN THE SET
						: MOVE POINTER TO FIRST ADDRESS OF NEXT SET
012176	077505			SOB	R5, LOP31B	
012200	062704	000040		ADD	#40, R4	
012204	020427	001752		CMP	R4, #ADREND	
012210	003001			BGT	CNT31A	: BRANCH IF DONE
012212	000763			BR	LOP31A	
012214	012700	001770	CNT31A:	MOV	#STATAB, R0	: LOAD R0 WITH THE ADDRESS OF THE STATE TABLE
012220	012001		LOP31C:	MOV	(R0)+, R1	: R1 POINTS TO THE ADDRESS
						: OF THE 1ST PDR IN THIS MODE
012222	012702	017776		MOV	#17776, R2	: SET UP VIRTUAL ADDRESS TO BE REFERENCED IN R2
012226	012037	177776		MOV	(R0)+, @#PS	: SET UP STATUS FOR THE CURRENT MODE
012232	005277	167176	LOP31D:	INC	@SRO	: TURN ON KT11-C
012236	011212			MOV	(R2), (R2)	: WRITE INTO THE PAGE BEING TESTED
012240	005077	167170		CLR	@SRO	: TURN OFF KT11-C
012244	032771	000100	000000	BIT	#100, @ (R1)	: CHECK THE W BIT
012252	001001			BNE	.+4	
012254	104006			HLT		: THE W BIT WAS NOT SET IN THE PDR
						: WHOSE ADDRESS IS POINTED TO BY R1,
						: AFTER THE CORRESPONDING PAGE WAS
						: WRITTEN
						: NOW CHECK ALL PDR'S TO SHOW THAT NO OTHER W-BITS WERE SET
012256	012703	001454		MOV	#ADRTAB, R3	: R3 POINTS TO THE ADDRESS OF THE
012262	012704	000020	LOP31E:	MOV	#20, R4	: PDR BEING CHECKED
012266	020103		LOP31F:	CMP	R1, R3	: DON'T CHECK THIS PDR IF IT IS THE
012270	001405			BEQ	CNT31B	: ONE WHOSE W-BIT SHOULD HAVE BEEN
						: SET (CORRESPONDING TO THE PAGE WRITTEN)
012272	032773	000100	000000	BIT	#100, @ (R3)	: CHECK THE W-BIT
012300	001401			BEQ	.+4	
012302	104006			HLT		: W BIT SET IN THE PDR WHOSE ADDRESS
						: IS POINTED TO BY R3, AS WELL AS
						: IN THE PDR CORRESPONDING TO THE PAGE
						: THAT WAS WRITTEN (ADDRESS OF PDR


```

012546 020427 001752          CMP      R4, #ADREND
012552 003001          BGT      CNT32A          ; BRANCH IF DONE
012554 000763          BR       LOP32A
012556 012700 001770  CNT32A: MOV    #STAB, R0      ; LOAD R0 WITH THE ADDRESS OF THE STATE TABLE
012562 012001          LOP32C: MOV    (R0)+, R1    ; R1 POINTS TO THE ADDRESS
                                ; OF THE 1ST PDR IN THIS MODE

012564 062701 000020          ADD      #20, R1
012570 012702 017776          MOV      #17776, R2      ; SET UP VIRTUAL ADDRESS TO BE REFERENCED IN R2
012574 012037 177776          MOV      (R0)+, #PS      ; SET UP STATUS FOR THE CURRENT MODE
012600 005277 166630          LOP32D: INC    @SR0      ; TURN ON KT11-C
012604 011212          MOV      (R2), (R2)      ; WRITE INTO THE PAGE BEING TESTED
012606 005077 166622          CLR     @SR0      ; TURN OFF KT11-C
012612 032771 000100 000000  BIT     #100, @ (R1)    ; CHECK THE W BIT
012620 001001          BNE     .+4
012622 104006          HLT

                                ; THE W BIT WAS NOT SET IN THE PDR
                                ; WHOSE ADDRESS IS POINTED TO BY R1,
                                ; AFTER THE CORRESPONDING PAGE WAS
                                ; WRITTEN

                                ; NOW CHECK ALL PDR'S TO SHOW THAT NO OTHER W-BITS WERE SET
012624 012703 001454          MOV      #ADRTAB, R3    ; R3 POINTS TO THE ADDRESS OF THE
012630 012704 000020          LOP32E: MOV    #20, R4    ; PDR BEING CHECKED
012634 020103          LOP32F: CMP    R1, R3    ; DON'T CHECK THIS PDR IF IT IS THE
012636 001405          BEQ     CNT32B          ; ONE WHOSE W-BIT SHOULD HAVE BEEN
                                ; SET (CORRESPONDING TO THE PAGE WRITTEN)
012640 032773 000100 000000  BIT     #100, @ (R3)    ; CHECK THE W-BIT
012646 001401          BEQ     .+4
012650 104006          HLT

                                ; W BIT SET IN THE PDR WHOSE ADDRESS
                                ; IS POINTED TO BY R3, AS WELL AS
                                ; IN THE PDR CORRESPONDING TO THE PAGE
                                ; THAT WAS WRITTEN (ADDRESS OF PDR
                                ; CORRESPONDING TO THE PAGE THAT
                                ; WAS WRITTEN IS POINTED TO BY R1)
012652 005723          CNT32B: TST    (R3)+    ; MOVE POINTER
012654 077411          SOB     R4, LOP32F      ; CHECK ALL PDR'S IN THIS GROUP
012656 062703 000040          ADD     #40, R3        ; IF DONE WITH THIS GROUP, MOVE POINTER
                                ; TO NEXT GROUP OF ADDRESSES
012662 020327 001752          CMP     R3, #ADREND    ; BRANCH IF SOME PDR'S NOT YET CHECKED
012666 002760          BLT     LOP32E

                                ; SHOW THAT WRITING THE PDR WILL CLEAR THE W BIT
012670 012771 077406 000000  MOV     #77406, @ (R1)  ; CLEAR W BIT VIA DATO TO PDR
012676 032771 000100 000000  BIT     #100, @ (R1)    ; CHECK W BIT
012704 001401          BEQ     .+4
012706 104006          HLT

                                ; W BIT DIDN'T CLEAR WHEN PDR
                                ; WAS WRITTEN (R1 POINTS TO
                                ; THE ADDRESS OF THE PDR)
012710 005721          CNT32C: TST    (R1)+    ; MOVE POINTER
012712 062702 020000          ADD     #20000, R2      ; CHANGE VA TO REFERENCE NEXT PAGE
012716 103330          BCC     LOP32D          ; BRANCH TO TEST NEXT PAGE IN THIS MODE
012720 020027 002002          CMP     R0, #STAEND    ; IF DONE IN THIS MODE, CHECK FOR ALL MODES DONE
012724 002716          BLT     LOP32C          ; LOOP UNTIL ALL MODES HAVE BEEN TESTED
012726 005077 166516          CLR     @SR3          ; REINITIALIZE SR3
012732 005077 166476          CLR     @SR0          ; REINITIALIZE SR0

                                ; 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
012736 104400          TEST33: SCOPE

```

```

012740 012737 000033 177570      MOV      #33, @#SR      ; DISPLAY TEST NUMBER
012746 005037 177776      CLR      @#PS          ; INITIALIZE PROCESSOR STATUS
012752 012706 001400      MOV      @KSTACK, SP  ; INITIALIZE KERNEL STACK POINTER
012756 004767 003300      JSR      %7, SETUP    ; INITIALIZE SR0, SR3
012762 026727 167044 000033      CMP      TESTCT, #33  ; IS THIS TEST BEING EXECUTED IN THE
012770 001401      BEQ     .+4           ; CORRECT SEQUENCE?- BRANCH IF YES
012772 104006      HLT     .             ; TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                           ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

012774 004767 003324      JSR      %7, RWALL    ; MAP ALL PAGES RW, 4K, BANK 0
013000 012777 000003 166650      MOV      #3, @KIPDR1  ; MAP KERNEL I-SPACE 1 NR, 1 PAGE
013006 012777 007600 166716      MOV      #7600, @KIPAR7 ; MAP KERNEL 7 I-SPACE TO THE EXTERNAL BANK
013014 012777 013042 166772      MOV      @RET33, @KTVEC ; SETUP ABORT RETURN
013022 005077 166770      CLR      @KTSTA      ;
013026 005277 166402      INC     @SR0         ; TURN ON KT11-C
013032 005737 030000      TST     @#30000     ; REFERENCE NR KERNEL 1 - SHOULD ABORT
013036 000000      HALT   .           ; NO NR ABORT
013040 000405      BR     DONE33      ;
013042 022777 140003 166364  RET33:  CMP      #140003, @SR0 ; CHECK SR0
013050 001401      BEQ     .+4         ;
013052 104006      HLT     .           ; SR0 INCORRECT - SHOULD SHOW REFERENCE
                                           ; TO KERNEL I-SPACE PAGE 1, AND BOTH
                                           ; NR AND PL ERRORS SHOULD BE SET

013054 005077 166354      DONE33: CLR      @SR0 ;
013060 016777 166732 166726      MOV     KTSTA, @KTVEC ; RESTORE TRAP CATCHER

; SHOW THAT KERNEL, USER, AND SUPERVISOR STACKS ARE IN THE
; RESPECTIVE D-SPACES WHEN D-SPACE ENABLES ARE SET
; D-SPACES ARE OFFSET FROM I-SPACES, AND AN IOT IS DONE TO
; EACH MODE. THE LOCATION WRITTEN INTO WHEN THE STACK IS PUSHED
; SHOWS WHICH SPACE WAS USED BY THAT STACK
TEST34: SCOPE
013066 104400      TEST34:
013070 012737 000034 177570      MOV      #34, @#SR    ; DISPLAY TEST NUMBER
013076 005037 177776      CLR      @#PS          ; INITIALIZE PROCESSOR STATUS
013102 012706 001400      MOV      @KSTACK, SP  ; INITIALIZE KERNEL STACK POINTER
013106 004767 003150      JSR      %7, SETUP    ; INITIALIZE SR0, SR3
013112 026727 166714 000034      CMP      TESTCT, #34  ; IS THIS TEST BEING EXECUTED IN THE
013120 001401      BEQ     .+4           ; CORRECT SEQUENCE?- BRANCH IF YES
013122 104006      HLT     .             ; TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                           ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

013124 004767 003144      JSR      %7, CLRALL   ; INITIALIZE ALL KT11-C REGISTERS
013130 012706 000500      MOV      #500, SP     ; SET THE KERNEL STACK TO VIRTUAL ADDRESS 500
013134 012737 040000 177776      MOV      #40000, @#PS ;
013142 012706 000100      MOV      #100, SP     ; SET THE SUPERVISOR STACK TO VIRTUAL ADDRESS 100
013146 012737 140000 177776      MOV      #140000, @#PS ;
013154 012706 000100      MOV      #100, SP     ; SET THE USER STACK TO VA 100
013160 005037 177776      CLR      @#PS          ;
013164 012777 077406 166462      MOV      #77406, @KIPDR0 ; MAP KERNEL, SUPERVISOR, AND USER PAGE 0
013172 012777 077406 166354      MOV      #77406, @SIPDR0 ; I-SPACES TO BANK 0, 4K, RW
013200 012777 077406 166246      MOV      #77406, @UIPDR0 ;
013206 012777 002006 166460      MOV      #2006, @KDPDR0 ; MAP KERNEL, SUPERVISOR, AND USER
013214 012777 000006 166352      MOV      #6, @SDPDR0  ; D SPACES TO BANK 0 (BUT OFFSET
013222 012777 000006 166244      MOV      #6, @UDPDR0  ; FROM PHYSICAL ADDRESS 0), RW
013230 012777 000007 166476      MOV      #7, @KDPAR0  ; KERNEL D SPACE STARTS AT 700
013236 012777 000007 166370      MOV      #7, @SDPAR0  ; SUPERVISOR D SPACE STARTS AT 700

```

013244	012777	000003	166262	MOV	#3, @JDPAR0	:USER D SPACE STARTS AT 300
013252	012737	013352	000720	MOV	#KRET34, @#720	:TEST USING IOT TRAP (THRU KERNEL D SPACE)
013260	005037	000722		CLR	@#722	:SETUP TO RETURN FROM TRAP IN KERNEL MODE
013264	016701	166144		MOV	SRO, R1	:LOAD R1 TO REFERENCE SRO
013270	012777	000007	166152	MOV	#7, @SR3	:TURN ON ALL D-SPACE ENABLES
013276	012777	077406	166406	MOV	#77406, @KDPDR7	:MAP KERNEL D SPACE PAGE 7
013304	012777	007600	166440	MOV	#7600, @KDPAR7	:TO THE EXTERNAL BANK, RW
013312	012777	077406	166272	MOV	#77406, @SDPDR7	:MAP SUPERVISOR D-SPACE PAGE 7
013320	012777	007600	166324	MOV	#7600, @SDPAR7	:TO THE EXTERNAL BANK, RW
013326	012777	077406	166156	MOV	#77406, @JDPDR7	:MAP USER D-SPACE PAGE 7
013334	012777	007600	166210	MOV	#7600, @JDPAR7	:TO THE EXTERNAL BANK, RW
013342	005277	166066		INC	@SRO	:TURN ON KT11-C
013346	000004			KTRP34:	IOT	:IOT TRAP SHOULD USE STACK IN KERNEL D-SPACE
013350	000240				NOP	:SINCE STATUS IN IOT VECTOR IS SET TO KERNEL
013352	005011			KRET34:	CLR	@R1
013354	012737	013400	000720	MOV	#SRET34, @#720	:TURN OFF KT11-C
013362	012737	040000	000722	MOV	#40000, @#722	:SETUP FOR IOT TO SUPERVISOR STACK
013370	005277	166040			INC	@SRO
013374	000004			STRP34:	IOT	:TURN ON KT11-C
013376	000240				NOP	:IOT TRAP SHOULD USE STACK IN SUPERVISOR D-SPACE
013400	005011			SRET34:	CLR	@R1
013402	012737	013426	000720	MOV	#URET34, @#720	:SINCE STATUS IN IOT VECTOR IS SET TO SUPERVISOR
013410	012737	140000	000722	MOV	#140000, @#722	:TURN OFF KT11-C
013416	005277	166012			INC	@SRO
013422	000004			UTRP34:	IOT	:SETUP FOR IOT TO USER
013424	000240				NOP	
013426	005011			URET34:	CLR	@R1
013430	022737	013350	001374	CMP	#KRET34-2, @#1374	:TURN ON KT11-C
013436	001401			BEQ	.+4	:IOT TRAP SHOULD USE STACK IN USER D-SPACE
013440	104006			HLT		:SINCE STATUS IN IOT VECTOR IS SET TO USER
013442	022737	000000	001376	CMP	#0, @#1376	:TURN OFF KT11-C
013450	001401			BEQ	.+4	:CHECK TO SEE IF IOT TRAP TO KERNEL PUSHED
013452	104006			HLT		:VALUE IN KERNEL D-SPACE
013454	022737	013376	000774	CMP	#SRET34-2, @#774	:KERNEL STACK CONTENTS WRONG. STACK NOT
013462	001401			BEQ	.+4	:IN KERNEL D-SPACE
013464	104006			HLT		
013466	022737	000000	000776	CMP	#0, @#776	:KERNEL STACK CONTENTS WRONG-TRAP STATUS NOT
013474	001401			BEQ	.+4	:WHERE IT SHOULD HAVE BEEN PUSHED
013476	104006			HLT		:OR VALUE WRONG
013500	022737	013424	000374	CMP	#URET34-2, @#374	:SUPERVISOR STACK CONTENTS WRONG-STACK
013506	001401			BEQ	.+4	:NOT IN SUPERVISOR D-SPACE
013510	104006			HLT		
013512	022737	040000	000376	CMP	#40000, @#376	:SUPERVISOR STACK WRONG-TRAP STATUS
013520	001401			BEQ	.+4	:NOT WHERE IT SHOULD HAVE BEEN PUSHED
013522	104006			HLT		:OR VALUE WRONG
013524	005037	000374		CLR	@#374	:USER STACK CONTENTS WRONG-STACK
013530	005037	000376		CLR	@#376	:NOT IN USER D-SPACE
013534	005037	000720		CLR	@#720	
013540	005037	000722		CLR	@#722	
013544	005037	000774		CLR	@#774	
013550	005037	000776		CLR	@#776	
013554	005037	001374		CLR	@#1374	
013560	005037	001376		CLR	@#1376	
013564	012706	001400		MOV	#KSTACK, SP	:RESTORE KERNEL STACK POINTER

```

: SHOW THAT TRAP, EMT, IOT, AND INTERRUPTS TAKE VECTORS FROM KERNEL
: D-SPACE IRREGARDLESS OF THE MODE AT THE TIME OF THE TRAP SEQUENCE
: ALSO SHOW THAT ODD-ADDRESS TRAP (AN "INTERNAL" TRAP) TAKES
: ITS VECTOR FROM KERNEL D-SPACE
: NOTE THAT IF DUAL ADDRESSING OCCURS, THE ERROR
: ADDRESS WILL BE USED (THE 0 OVERRIDES THE 1)
013570 104400 TEST35: SCOPE
013572 012737 000035 177570 MOV #35, @#SR ; DISPLAY TEST NUMBER
013600 005037 177776 CLR @#PS ; INITIALIZE PROCESSOR STATUS
013604 012706 001400 MOV #KSTACK, SP ; INITIALIZE KERNEL STACK POINTER
013610 004767 002446 JSR %7, SETUP ; INITIALIZE SR0, SR3
013614 026727 166212 000035 CMP TESTCT, #35 ; IS THIS TEST BEING EXECUTED IN THE
013622 001401 BEQ .+4 ; CORRECT SEQUENCE?- BRANCH IF YES
013624 104006 HLT ; TEST EXECUTED OUT OF SEQUENCE- TESTCT
; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

013626 004767 002472 JSR %7, RWALL ; MAP ALL PAGES RW, 4K, BANK 0
013632 012777 000001 166074 MOV #1, @KDPAR0 ; OFFSET KERNEL 0 D-SPACE 1 PAGE
013640 012777 007600 166104 MOV #7600, @KDPAR7 ; MAP KERNEL 7 TO THE EXTERNAL BANK
013646 012777 007600 165776 MOV #7600, @SDPAR7 ; MAP SUPERVISOR 7 TO THE EXTERNAL BANK
013654 012777 007600 165670 MOV #7600, @UDPAR7 ; MAP USER 7 TO THE EXTERNAL BANK
013662 016701 165546 MOV SR0, R1 ; SETUP R1 TO REFERENCE SR0
013666 012737 040000 177776 MOV #40000, @#PS ; SETUP SUPERVISOR STACK
013674 012706 001000 MOV #SSTACK, SP
013700 012737 140000 177776 MOV #140000, @#PS ; SETUP USER STACK
013706 012706 000400 MOV #USTACK, SP
013712 005037 177776 CLR @#PS
013716 012706 001300 MOV #KSTACK-100, SP ; LOWER THE KERNEL STACK POINTER TO
; KEEP IT OUT OF CODE WHEN RELOCATED
; SET ALL D-SPACE ENABLES
; IF SUCCESSFUL, WILL PICK UP VECTOR
; RETURN IN PHYSICAL ADDRESS 134
; IF NO GOOD, WILL PICK UP VECTOR RETURN
; IN PHYSICAL ADDRESS 34
; SET MODE TO SUPERVISOR
; TURN ON KT11-C
; SHOULD PICK UP RETURN ADDRESS FROM
; KERNEL D-SPACE
; RESTORE STACK POINTER
; TURN OFF KT11-C
; TRAP VECTOR WASN'T FROM KERNEL
; D-SPACE
; GO TO NEXT SUBTEST
; RESTORE STACK POINTER
; TURN OFF KT11-C
; RESTORE TRAP VECTOR CONTENTS

013722 012777 000007 165520 MOV #7, @SR3
013730 012737 014004 000134 MOV #OK35A, @#134
013736 012737 040000 000136 MOV #40000, @#136
013744 012737 013774 000034 MOV #NG35A, @#34
013752 012737 040000 000036 MOV #40000, @#36
013760 012737 040000 177776 MOV #40000, @#PS
013766 005277 165442 INC @SR0
013772 104400 TRP35: TRAP

013774 022626 NG35A: CMP (SP)+, (SP)+
013776 005011 CLR @R1
014000 104006 HLT

014002 000402 BR EMT35
014004 022626 OK35A: CMP (SP)+, (SP)+
014006 005011 CLR @R1
014010 012737 016552 000034 EMT35: MOV #SCOPEC, @#34
014016 005037 000036 CLR @#36
014022 012737 000136 000134 MOV #136, @#134
014030 005037 000136 CLR @#136
014034 012737 014124 000130 MOV #OK35B, @#130 ; SETUP SUCCESS RETURN
014042 012737 014100 000030 MOV #NG35B, @#30 ; SETUP FAILURE RETURN
014050 012737 140000 000132 MCV #140000, @#132
014056 012737 140000 000032 MOV #140000, @#32
014064 012737 140000 177776 MOV #140000, @#PS
014072 005277 165336 INC @SR0 ; SET MODE TO USER
; TURN ON KT11-C

```


DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 47
 DCKTAB.P11

014676	012702	037776		MOV	#37776,R2	;SETUP R2 TO REFERENCE KERNEL 1
014702	052777	000001	164524	BIS	#1,SR0	;TURN ON KT11-C
014710	012242			MOV	(R2)+,-(R2)	;REFERENCE KERNEL 1 D-SPACE
014712	005077	164516		ADR36A: CLR	SR0	;SHOULD CAUSE A SECOND NON-RESIDENT ABORT
014716	104006			HLT		;TURN OFF KT11-C
014720	000500			BR	DONE36	;2ND REFERENCE TO KERNEL 1 D-SPACE
						;DIDN'T ABORT - PREVIOUS ERROR FLAG
						;NOT YET CLEARED BUT ABORT SHOULD
						;STILL HAVE OCCURRED
014722	042777	000001	164504	INT36A: BIC	#1,SR0	;TURN OFF KT11-C
014730	022777	100022	164476	CMP	#100022,SR0	;CHECK SR0
014736	001401			BEQ	.+4	
014740	104006			HLT		;SR0 INCORRECT AFTER 2ND NON-RESIDENT ABORT
014742	022777	000027	164470	CMP	#27,SR1	;CHECK SR1 - SHOULD CONTAIN VALUE
014750	001401			BEQ	.+4	;FROM 1ST ABORT
014752	104006			HLT		;SR1 DOESN'T CONTAIN VALUE FROM 1ST ABORT
014754	022777	014630	164462	CMP	#ADR36,SR2	;CHECK SR2
014762	001401			BEQ	.+4	
014764	104006			HLT		;SR2 DOESN'T CONTAIN VALUE FROM 1ST ABORT
014766	021627	014712		CMP	(R6),#ADR36A	;CHECK ADDRESS PUSHED ON STACK
014772	001401			BEQ	.+4	
014774	104006			HLT		;INCORRECT ADDRESS ON STACK - SHOULD BE
						;PC AT TIME SECOND ABORT OCCURRED
014776	022626			CMP	(R6)+,(R6)+	;RESTORE STACK POINTER
015000	012777	015044	165006	MOV	#INT36B,KTVEC	;CHANGE RETURN ADDRESS
015006	005077	164422		CLR	SR0	;CLEAR NON-RESIDENT ERROR BIT-SHOULD
						;UNLOCK ERROR TRACKING
						;CHECK TO SEE THAT ERROR BIT CLEARED
015012	105777	164420		TSTB	SR0H	
015016	001401			BEQ	.+4	
015020	104006			HLT		;SR0 HIGH BYTE INCORRECT
015022	012702	037776		MOV	#37776,R2	;SETUP R2 TO REFERENCE KERNEL 1
015026	005277	164402		INC	SR0	;TURN ON KT11-C
015032	012242			ADR36B: MOV	(R2)+,-(R2)	;3RD ABORT REFERENCE, ERROR BIT WAS CLEARED
015034	005077	164374		ADR36C: CLR	SR0	;TURN OFF KT11-C
015040	104006			HLT		;3RD REFERENCE TO KERNEL 1 D-SPACE
015042	000427			BR	DONE36	;DIDN'T CAUSE NON-RESIDENT ABORT
015044	042777	000001	164362	INT36B: BIC	#1,SR0	;TURN OFF KT11-C
015052	022777	100022	164354	CMP	#100022,SR0	;CHECK SR0
015060	001401			BEQ	.+4	
015062	104006			HLT		;SR0 INCORRECT AFTER NON-RESIDENT ABORT
015064	022777	000022	164346	CMP	#22,SR1	;CHECK SR1
015072	001401			BEQ	.+4	
015074	104006			HLT		;SR1 INCORRECT - SHOULD HAVE RECORDED
						;CHANGES MADE DURING 3RD ABORTED REFERENCE
015076	022777	015032	164340	CMP	#ADR36B,SR2	;CHECK SR2
015104	001401			BEQ	.+4	
015106	104006			HLT		;SR2 INCORRECT - SHOULD CONTAIN
						;LAST FETCH ADDRESS BEFORE ABORT
015110	022716	015034		CMP	#ADR36C,(SP)	;CHECK STACK
015114	001401			BEQ	.+4	
015116	104006			HLT		;PC ON STACK INCORRECT
015120	022626			CMP	(R6)+,(R6)+	;RESTORE STACK POINTER
015122	005077	164306		CLR	SR0	;CLEAR ERROR BIT
015126	005077	164664		CLR	KTSTA	;CHANGE TRAP RETURN TO CAUSE A HALT
015132	016777	164660	164654	MOV	KTSTA,KTVEC	;ON A FALSE INTERRUPT

```

;SHOW THAT THE ABORT LOGIC "LOCKS" SR0, SR1, AND SR2 AFTER A
;PAGE LENGTH ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0,
;WHEN THEY RESUME TRACKING. A PAGE LENGTH ERROR SHOULD STILL ABORT
;TO 250 EVEN WHEN BIT 14 (SR0) IS ALREADY SET
015140 104400                                TEST37: SCOPE
015142 012737 000037 177570                MOV     #37, @SR
015150 005037 177776                        CLR     @PS
015154 012706 001400                        MOV     #KSTACK, SP
015160 004767 001076                        JSR     %7, SETUP
015164 026727 164642 000037                CMP     TESTCT, #37
015172 001401                                BEQ     .+4
015174 104006                                HLT

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

015176 004767 001076                        JSR     %7, CLRALL
015202 012777 077406 164502                MOV     #77406, @KDPDR7
015210 012777 007600 164534                MOV     #7600, @KDPAR7
015216 012777 077406 164430                MOV     #77406, @KIPDR0
015224 012777 077406 164442                MOV     #77406, @KDPDR0
015232 012777 017406 164436                MOV     #17406, @KDPDR1

;CLEAR ALL KT11-C REGISTERS
;MAP KERNEL 7 D-SPACE RW, 4K,
;EXTERNAL BANK
;MAP KERNEL 0 I AND D-SPACES
;RW, 4K, BANK 0
;MAP KERNEL 1 D-SPACE RW
;1K, BANK 0
;ENABLE ALL D-SPACES
;SETUP ABORT RETURN VECTOR

015240 012777 000007 164202                MOV     #7, @SR3
015246 012777 015302 164540                MOV     #INT37, @KTVEC
015254 005077 164536                        CLR     @KTSTA
015260 005277 164150                        INC     @SR0
015264 013737 037776 037776  ADR37: MOV     @#37776, @#37776

;TURN ON KT11-C
;REFERENCE KERNEL 1 D-SPACE
;SHOULD CAUSE PAGE LENGTH ABORT
;TURN OFF KT11-C
;REFERENCE TO KERNEL 1 D-SPACE
;DIDN'T CAUSE PAGE LENGTH ABORT
;TURN OFF KT11-C
;CHECK SR0

015272 005077 164136                        CLR     @SR0
015276 104006                                HLT
015300 000526                                BR     DONE37
015302 042777 000001 164124  INT37: BIC     #1, @SR0
015310 022777 040022 164116                CMP     #40022, @SR0
015316 001401                                BEQ     .+4
015320 104006                                HLT
;SR0 INCORRECT AFTER PAGE LENGTH ABORT
;SETUP NEW RETURN VECTOR
;RESTORE STACK POINTER
;SETUP R2 TO REFERENCE KERNEL 1
;TURN ON KT11-C
;REFERENCE KERNEL 1 D-SPACE
;SHOULD CAUSE A SECOND PAGE LENGTH ABORT
;TURN OFF KT11-C
;2ND REFERENCE TO KERNEL 1 D-SPACE
;DIDN'T ABORT - PREVIOUS ERROR FLAG
;NOT YET CLEARED BUT ABORT SHOULD
;STILL HAVE OCCURRED
;TURN OFF KT11-C
;CHECK SR0

015322 012777 015356 164464                MOV     #INT37A, @KTVEC
015330 022626                        CMP     (R6)+, (R6)+
015332 012702 037776 164070                MOV     #37776, R2
015336 052777 000001 164070                BIS     #1, @SR0
015344 012242                        MOV     (R2)+, -(R2)

;SR0 INCORRECT AFTER 2ND PAGE LENGTH ABORT
;CHECK SR1 - SHOULD CONTAIN VALUE
;FROM 1ST ABORT
;SR1 DOESN'T CONTAIN VALUE FROM 1ST ABORT
;CHECK SR2

015346 005077 164062  ADR37A: CLR     @SR0
015352 104006                                HLT
015354 000500                                BR     DONE37

;SR2 DOESN'T CONTAIN VALUE FROM 1ST ABORT
;CHECK ADDRESS PUSHED ON STACK

015356 042777 000001 164050  INT37A: BIC     #1, @SR0
015364 022777 040022 164042                CMP     #40022, @SR0
015372 001401                                BEQ     .+4
015374 104006                                HLT
015376 022777 000027 164034                CMP     #27, @SR1
015404 001401                                BEQ     .+4
015406 104006                                HLT
015410 022777 015264 164026                CMP     #ADR37, @SR2
015416 001401                                BEQ     .+4
015420 104006                                HLT
015422 021627 015346                        CMP     (R6), #ADR37A
    
```

015426	001401			BEQ	+.4			
015430	104006			HLT				: INCORRECT ADDRESS ON STACK - SHOULD BE
015432	022626			CMP	(R6)+, (R6)+			: PC AT TIME SECOND ABORT OCCURRED
015434	012777	015500	164352	MOV	#INT37B, @KTVEC			: RESTORE STACK POINTER
015442	005077	163766		CLR	@SRO			: CHANGE RETURN ADDRESS
015446	105777	163764		TSTB	@SROH			: CLEAR PAGE LENGTH ERROR BIT-SHOULD
015452	001401			BEQ	+.4			: "UNLOCK" ERROR TRACKING
015454	104006			HLT				: CHECK TO SEE THAT ERROR BIT CLEARED
015456	012702	037776		MOV	#37776, R2			: SR0 HIGH BYTE INCORRECT
015462	005277	163746		INC	@SRO			: SETUP R2 TO REFERENCE KERNEL 1
015466	012242			ADR37B: MOV	(R2)+, -(R2)			: TURN ON KT11-C
015470	005077	163740		ADR37C: CLR	@SRO			: 3RD ABORT REFERENCE, ERROR BIT WAS CLEARED
015474	104006			HLT				: TURN OFF KT11-C
015476	000427			BR	DONE37			: 3RD REFERENCE TO KERNEL 1 D-SPACE
015500	042777	000001	163726	INT37B: BIC	#1, @SRO			: DIDN'T CAUSE PAGE LENGTH ABORT
015506	022777	040022	163720	CMP	#40022, @SRO			: TURN OFF KT11-C
015514	001401			BEQ	+.4			: CHECK SR0
015516	104006			HLT				: SR0 INCORRECT AFTER PAGE LENGTH ABORT
015520	022777	000022	163712	CMP	#22, @SR1			: CHECK SR1
015526	001401			BEQ	+.4			: SR1 INCORRECT - SHOULD HAVE RECORDED
015530	104006			HLT				: CHANGES MADE DURING 3RD ABORTED REFERENCE
015532	022777	015466	163704	CMP	#ADR37B, @SR2			: CHECK SR2
015540	001401			BEQ	+.4			: SR2 INCORRECT - SHOULD CONTAIN
015542	104006			HLT				: LAST FETCH ADDRESS BEFORE ABORT
015544	022716	015470		CMP	#ADR37C, (SP)			: CHECK STACK
015550	001401			BEQ	+.4			: PC ON STACK INCORRECT
015552	104006			HLT				: RESTORE STACK POINTER
015554	022626			CMP	(R6)+, (R6)+			: CLEAR ERROR BIT
015556	005077	163652		DONE37: CLR	@SRO			: CHANGE TRAP RETURN TO CAUSE A HALT
015562	005077	164230		CLR	@KTSTA			: ON A FALSE INTERRUPT
015566	016777	164224	164220	MOV	KTSTA, @KTVEC			
								: SHOW THAT THE ABORT LOGIC "LOCKS" SR0, SR1, AND SR2 AFTER A
								: ACCESS VIOLATION ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SR0,
								: WHEN THEY RESUME TRACKING. A ACCESS VIOLATION ERROR SHOULD STILL ABORT
								: TO 250 EVEN WHEN BIT 13 (SR0) IS ALREADY SET
015574	104400			TEST40: SCOPE				
015576	012737	000040	177570	MOV	#40, @#SR			: DISPLAY TEST NUMBER
015604	005037	177776		CLR	@#PS			: INITIALIZE PROCESSOR STATUS
015610	012706	001400		MOV	#KSTACK, SP			: INITIALIZE KERNEL STACK POINTER
015614	004767	000442		JSR	%7, SETUP			: INITIALIZE SR0, SR3
015620	026727	164206	000040	CMP	TESTCT, #40			: IS THIS TEST BEING EXECUTED IN THE
015626	001401			BEQ	+.4			: CORRECT SEQUENCE?- BRANCH IF YES
015630	104006			HLT				: TEST EXECUTED OUT OF SEQUENCE- TESTCT
								: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
015632	004767	000436		JSR	%7, CLRALL			: CLEAR ALL KT11-C REGISTERS
015636	012777	077406	164046	MOV	#77406, @KDPDR7			: MAP KERNEL 7 D-SPACE RW, 4K,
015644	012777	007600	164100	MOV	#7600, @KDPAR7			: EXTERNAL BANK
015652	012777	077406	163774	MOV	#77406, @KIPDR0			: MAP KERNEL 0 I AND D-SPACES
015660	012777	077406	164006	MOV	#77406, @KDPDR0			: RW, 4K, BANK0
015666	012777	077402	164002	MOV	#77402, @KDPDR1			: MAP KERNEL 1 D-SPACE RRO


```

016154 022777 171022 163256      CMP      #171022, @SR1      ;CHECK SR1
016162 001401                      BEQ      .+4
016164 104006                      HLT
                                ;SR1 INCORRECT - SHOULD HAVE RECORDED
                                ;CHANGES MADE DURING 3RD ABORTED REFERENCE
                                ;CHECK SR2
016166 022777 016122 163250      CMP      #ADR40B, @SR2
016174 001401                      BEQ      .+4
016176 104006                      HLT
                                ;SR2 INCORRECT - SHOULD CONTAIN
                                ;LAST FETCH ADDRESS BEFORE ABORT
                                ;CHECK STACK
016200 022716 016124              CMP      #ADR40C, (SP)
016204 001401                      BEQ      .+4
016206 104006                      HLT
                                ;PC ON STACK INCORRECT
                                ;RESTORE STACK POINTER
016210 022626                      CMP      (R6)+, (R6)+
016212 005077 163216      DONE40: CLR      @SR0
016216 005077 163574          CLR      @KTSTA
016222 016777 163570 163564      MOV      KTSTA, @KTVEC
                                ;CLEAR ERROR BIT
                                ;CHANGE TRAP RETURN TO CAUSE A HALT
                                ;ON A FALSE INTERRUPT

016230 104400                      SCOPE
016232 004767 001144              JSR      %7, BELL
016236 013701 000042              MOV      @#42, R1
                                ;MONITOR HOOK
016242 001405                      BEQ      END
016244 000005                      RESET
016246 004711      LOGIC:      JSR      %7, @R1
016250 000240                      NOP
016252 000240                      NOP
016254 000240                      NOP
016256 000167 163554      END:      JMP      START
016262 005077 163146      SETUP:   CLR      @SR0
016266 005077 163156          CLR      @SR3
016272 000207                      RTS      %7

;SUBROUTINE TO CLEAR ALL KT11-C REGISTERS (EXCEPT SR1, SR2)
016274 005077 163134      CLRALL:  CLR      @SR0
016300 005077 163144          CLR      @SR3
016304 005000                      CLR      R0
016306 012701 000140      CLRLP:  MOV      #96, R1
                                ;COUNT OF REGISTERS TO BE CLEARED
016312 005070 001454          CLR      @ADRTAB(R0)
                                ;CLEAR REGISTERS THRU ADDRESS TABLE
016316 005720                      TST      (R0)+
                                ;MOVE POINTER
016320 077104                      SOB      R1, CLRLP
016322 000207                      RTS      %7
                                ;LOOP TILL DONE

;SUBROUTINE TO MAKE ALL PAGES RW, BANK 0, 4K, UP
016324 005077 163104      RWALL:  CLR      @SR0
016330 012701 001454          MOV      #ADRTAB, R1
                                ;LOAD R1 TO POINT TO ADDRESS OF 1ST PDR
016334 012700 000020      RWL1:  MOV      #20, R0
                                ;SETUP R0 AS COUNTER
016340 005071 000040      RWL2:  CLR      @40(R1)
                                ;CLEAR PAR
016344 012731 077406          MOV      #77406, @ (R1)+
                                ;SET PDR RW, 4K
016350 077005                      SOB      R0, RWL2
016352 062701 000040          ADD      #40, R1
                                ;AT END OF GROUP, MOVE POINTER
016356 020127 001752          CMP      R1, #ADREND
016362 002764                      BLT      RWL1
                                ;CHECK FOR DONE
                                ;CONTINUE UNTIL DONE

```

016364 000207
016366 005077 163042
016372 012701 001454
016376 012700 000010
016402 005071 000040
016406 012731 077406
016412 077005
016414 062701 000060
016420 020127 001752
016424 002764
016426 000207

```

:SUBROUTINE TO MAKE ALL I SPACE PAGES RW, BANK 0,4K,UP
RWISP: CLR    %7
        JSRO
        MOV    #ADRTAB,R1      ;R1 POINTS TO ADDRESS TABLE
RWI1:  MOV    #10,R0          ;USE R0 AS COUNTER
RWI2:  CLR    @40(R1)         ;CLEAR PAR
        MOV    #77406,@(R1)+  ;MAP PDR RW, 4K
        SOB   R0,RWI2
        ADD   #60,R1          ;AT END OF GROUP, MOVE POINTER
        CMP   R1,#ADREND     ;CHECK FOR DONE
        BLT  RWI1            ;CONTINUE UNTIL DONE
        RTS   %7
    
```

016430 005077 163000
016434 012701 001474
016440 012700 000010
016444 005071 000040
016450 012731 077406
016454 077005
016456 062701 000060
016462 020127 001752
016466 002764
016470 000207

```

:SUBROUTINE TO MAKE ALL D-SPACE PAGES RW, BANK 0, 4K, UP
RWDSP: CLR    JSRO
        MOV    #ADRTAB+20,R1 ;R1 POINTS TO ADDRESS TABLE
RWD1:  MOV    #10,R0          ;USE R0 AS A COUNTER
RWD2:  CLR    @40(R1)         ;CLEAR PAR
        MOV    #77406,@(R1)+  ;SET PDR RW, 4K.
        SOB   R0,RWD2
        ADD   #60,R1          ;AT END OF GROUP, MOVE POINTER
        CMP   R1,#ADREND     ;CHECK FOR DONE
        BLT  RWD1            ;BRANCH IF NOT DONE
        RTS   %7
    
```

```

;ROUTINE TO LOOP THRU A SINGLE INSTRUCTION TEST
;LOAD THE STARTING ADDRESS OF THE TEST
;YOU WISH TO RUN (THE ADDRESS OF THE TESTXX
;TAG) AT THE 1ST HALT, SET SWITCH REGISTER
;OPTIONS AT THE 2ND HALT.
;NOTE THAT SW11 MUST BE DOWN AFTER THE 2ND HALT
    
```

016472 005037 177776
016476 012706 001400
016502 000000
016504 016767 161060 000036
016512 062767 000002 000030
016520 000000
016522 005067 000154
016526 012767 016540 000150
016534 000177 000010
016540 005067 000136
016544 000177 000000
016550 000000

```

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

016552 032737 040000 177570
016560 001015
016562 032737 004000 177570
016570 001025
016572 026767 000104 000100
016600 100021
016602 005267 000074
016606 012737 000340 177776
016614 022606
016616 012637 177776

```

;SCOPE AND/OR ITERATION LOOP FOR EACH TEST
SCOPEC: BIT    #40000,@#SR    ;TEST SR FOR SCOPE
        BNE   SCOPEB        ;YES,SCOPE
        BIT    #4000,@#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
        CMP  (6)+,%6        ;REPOSITION STACK
        MOV  (6)+,@#PS      ;RESTORE PREVIOUS PROCESSOR STATUS
    
```

```

016622 032737 000400 177570          BIT      #400, @#SR      ;LOAD MICROBREAK REGISTER?
016630 001403          BEQ      .+10        ;NO-BRANCH
016632 113777 177570 163160          MOVVB   @#SR, @UBRK   ;YES-LOAD FROM SR
016640 000177 000040          JMP     @RETURN      ;REPEAT TEST
016644 005067 000032          SCOPEG: CLR   SCOPEF ;CLEAR COUNT
016650 005267 163156          INC     TESTCT      ;STEP TEST COUNTER TO ALLOW CHECKING
                                ;ORDER OF EXECUTION.
016654 011667 000024          MOV     @%6, RETURN  ;SAVE SCOPE RETURN POINTER
016660 032737 000400 177570          BIT      #400, @#SR      ;LOAD MICROBREAK REGISTER?
016666 001403          BEQ      .+10        ;NO-BRANCH
016670 113777 177570 163122          MOVVB   @#SR, @UBRK   ;YES-LOAD FROM SR
016676 000002          RTI                    ;RETURN INLINE-NEXT TEST
016700 002000          ICOUNT: 2000        ;ITERATION COUNT
016702 000000          SCOPEF: 0           ;COUNT LOCATION FOR ITERATION LOOP
016704 000000          RETURN: 0          ;ADDRESS OF LAST TEST
  
```

```

                                ;ENTERED WITH SYSTEM TRAP CALL (HLT)
                                ;PRINT OUT THE ERROR PC+2 AND STATUS REGISTER
016706 012767 000340 161062          PRINT: MOV     #340, PS      ;SET PRIORITY TO 7
016714 036727 160650 020000          BIT      SR, #20000     ;TEST FOR INHIBIT PRINT OUT
016722 001401          BEQ      .+4           ;BRANCH TO PRINT
016724 000432          BR      CK            ;INHIBIT, CHECK FOR HALT
016726 012667 000072          MOV     (6)+, SAVPC    ;PC OF FAILING ROUTINE
016732 012667 000070          MOV     (6)+, SAVPSR  ;PSR OF ERROR CONDITION
016736 024646          CMP     -(6), -(6)   ;RESTORE STACK
016740 012767 000200 161030          MOV     #200, PS      ;OUTPUT CARRIAGE RETURN AND LINE FEED
016746 004767 000470          JSR     %7, CALF      ;LOAD WITH FAILING PC+2
016752 016767 000046 000336          MOV     SAVPC, PTEMP1 ;PRINT FAILING PC+2
016760 004767 000076          JSR     %7, PROCT     ;WAIT FOR TTY READY
016764 105777 162432          TSTB   @TCSR
016770 100375          BPL     .-4
016772 012777 000240 162424          MOV     #240, @TDBR   ;OUTPUT A SPACE
017000 016767 000022 000310          MOV     SAVPSR, PTEMP1 ;LOAD PROCESSOR STATUS
017006 004767 000050          JSR     %7, PROCT     ;PRINT PROCESSOR STATUS
017012 005767 160552          CK:    TST     SR      ;CHECK SR FOR HALT SWITCH
017016 100001          BPL     .+4           ;BRANCH IF NOT SET
017020 000000          HALT                    ;HALT ON ERROR UP
017022 000002          RTI                    ;RETURN TO MAIN LINE
017024 000000          SAVPC: 0
017026 000000          SAVPSR: 0
  
```

```

                                ;SUBROUTINE TO PRINT OUT OCTAL NUMBER
                                ;PRSHRT DELETES LEADING ZEROS
                                ;PROCT PRINTS OUT 6 OCTAL DIGITS
017030 012767 000001 000254          PRSHRT: MOV     #1, PRSFLG ;SET FLAG TO INDICATE SHORT PRINTOUT
017036 005767 000254          TST     PTEMP1       ;CHECK FOR ZERO
017042 001011          BNE     PROCT+4      ;BRANCH IF NOT ZERO
017044 105777 162352          TSTB   @TCSR         ;WAIT FOR TTY READY
017050 100375          BPL     .-4
017052 012777 000260 162344          MOV     #260, @TDBR  ;OUTPUT A SINGLE ZERO
017060 000207          RTS                    ;RETURN
017062 005067 000224          PROCT: CLR     PRSFLG ;CLEAR FLAG TO INDICATE FULL PRINTOUT
017066 005067 000230          CLR     PTEMP3       ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
017072 005067 000216          CLR     PRFLG        ;INITIALIZE CARRY FLAG FOR ROTATES
017076 012767 000260 000214          MOV     #260, PTEMP2 ;SETUP R3
  
```

017104	005767	000206		TST	PTEMP1		:CHECK BIT 15 OF NUMBER
017110	100002			BPL	.+6		:BRANCH IF ZERO
017112	005267	000202		INC	PTEMP2		:INCREMENT R3 IF ONE
017116	006167	000174		ROL	PTEMP1		:ROTATE LEFT MOST OCTAL TO RIGHT END
017122	006167	000170		ROL	PTEMP1		
017126	005567	000162		ADC	PRFLG		:STORE CARRY
017132	005767	000154		TST	PRSFLG		:CHECK FOR SHORT PRINTOUT
017136	001404			BEQ	P.WAIT		:BRANCH IF NOT SET
017140	026727	000154	000260	CMP	PTEMP2, #260		:CHECK FOR ZERO IF SET
017146	001421			BEQ	P.CONT		:IF SET, GO TO NEXT CHARACTER
017150	105777	162246		TSTB	@TCSR		:WAIT FOR TTY READY
017154	100375			BPL	P.WAIT		
017156	016777	000136	162240	MOV	PTEMP2, @TDBR		:OUTPUT NEXT CHARACTER
017164	105777	162232		TSTB	@TCSR		:READY FOR NEXT CHARACTER?
017170	100375			BPL	.-4		:NO - WAIT
017172	012777	000377	162224	MOV	#377, @TDBR		:ISSUE A 'DELETE' CODE
							:TO BUY TIME!
017200	105777	162216		TSTB	@TCSR		:ALL DONE?
017204	100375			BPL	.-4		:NO - HANG IN THERE!
017206	005067	000100		CLR	PRSFLG		:PRINT REST OF NUMBER AFTER A NON-ZERO DIGIT
017212	005267	000104		INC	PTEMP3		:COUNT
017216	026727	000100	000006	CMP	PTEMP3, #6		:CHECK FOR DONE
017224	001001			BNE	P.CNT1		:BRANCH IF NOT DONE
017226	000207			RTS	%7		
017230	000241			CLC			:CLEAR CARRY
017232	005767	000056		TST	PRFLG		:CHECK FOR PREVIOUS CARRY
017236	001403			BEQ	+.10		:BRANCH IF PREVIOUSLY ZERO
017240	005067	000050		CLR	PRFLG		:INITIALIZE FLAG
017244	000261			SEC			:SET CARRY
017246	006167	000044		ROL	PTEMP1		:ROTATE NEXT CHARACTER INTO RIGHT END
							:OF REGISTER
017252	006167	000040		ROL	PTEMP1		
017256	006167	000034		ROL	PTEMP1		
017262	005567	000026		ADC	PRFLG		:STORE CARRY
017266	016767	000024	000024	MOV	PTEMP1, PTEMP2		:LOAD DATA INTO R3
017274	042767	177770	000016	BIC	#177770, PTEMP2		:CLEAR ALL BUT LOWEST OCTAL DIGIT
017302	052767	000260	000010	BIS	#260, PTEMP2		:SET TO ASCII EQUIVALENT
017310	000710			BR	P.CK		:LOOP
017312	000000			PRSFGL: 0			:CONTAINS VALUE TO BE OUTPUT
017314	000000			PRFLG: 0			:SCRATCH
017316	000000			PTEMP1: 0			:USED TO COUNT CHARACTERS OUTPUT
017320	000000			PTEMP2: 0			
017322	000000			PTEMP3: 0			
				:EMT HANDLER			
				:FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES			
017324	011667	000032		EMTSRV: MOV	@SP, EPC		:GET CALL
017330	162767	000002	000024	SUB	#2, EPC		
017336	017767	000020	000016	MOV	@EPC, EPC		
017344	105067	000013		CLRB	EPC+1		:SAVE OFFSET ONLY
017350	062767	017364	000004	ADD	#EMTAB, EPC		:POINT TO TABLE OF ADDRESSES
017356	017707	000000		MOV	@EPC, PC		:JUMP TO DESIRED ROUTINE
017362	000000			EPC: 0			
	104000			PATCH1=EMT+0			
	104002			PATCH2=EMT+2			
	104004			PATCH3=EMT+4			

017364 104000
 017366 104002
 017370 104004
 017372 016706
 017374 017376

EMTAB: PATCH1
 PATCH2
 PATCH3
 PRINT
 KTOFFS

017376 000005
 017400 000006

:ROUTINE TO TURN OFF KT11-C VIA RESET
 KTOFFS: RESET
 RTT

017402 105777 162014
 017406 100375
 017410 012777 000207 162006
 017416 105777 162000
 017422 100375
 017424 012777 000377 161772
 017432 105777 161764
 017436 100375
 017440 000207

:BELL ON PASS COMPLETE
 BELL: TSTB @TCSR
 BPL .-4
 MOV #207,@DDBR
 TSTB @TCSR
 BPL .-4
 MOV #377,@DDBR
 TSTB @TCSR
 BPL .-4
 RTS %7

:READY FOR NEXT CHARACTER?
 :NO - WAIT
 :ISSUE A 'DELETE' CODE
 :TO BUY TIME!
 :ALL DONE?
 :NO - HANG IN THERE!

017442 105777 161754
 017446 100375
 017450 012777 000215 161746
 017456 105777 161740
 017462 100375
 017464 012777 000212 161732
 017472 000207
 017712

:SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED
 CRLF: TSTB @TCSR ;WAIT FOR TTY READY
 BPL .-4
 MOV #215,@DDBR ;OUTPUT CARRIAGE RETURN
 TSTB @TCSR ;WAIT FOR TTY READY
 BPL .-4
 MOV #212,@DDBR ;OUTPUT LINEFEED
 RTS %7 ;RETURN
 .=17712

017712 125252
 000001

DESTAD: 125252
 .END

E05

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 57
 DCKTAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

ADD23	007054	1405#	1419															
ADD23A	007154	1420#																
ADD24	007306	1451#	1462															
ADD24A	007406	1463#																
ADREND=	001752	413#	616	2048	2083	2133	2169	2747	2758	2770								
ADRTAB	001454	314#	592	594	2038	2066	2123	2152	2734*	2741	2752	2764						
ADR22A	006326	1288#																
ADR22B	006414	1306#																
ADR22C	006476	1325#																
ADR22D	006572	1345#																
ADR22E	006672	1368#																
ADR25	007530	1496#	1510	1515														
ADR25A	007532	1497#	1525															
ADR36	014630	2458#	2485															
ADR36A	014712	2473#	2488															
ADR36B	015032	2501#	2513															
ADR36C	015034	2502#	2517															
ADR37	015264	2550#	2577															
ADR37A	015346	2565#	2580															
ADR37B	015466	2593#	2605															
ADR37C	015470	2594#	2609															
ADR40	015720	2642#	2669															
ADR40A	016002	2657#	2672															
ADR40B	016122	2685#	2697															
ADR40C	016124	2686#	2701															
AD13	004272	911#	912															
AD21A =	006066	1219	1220#															
AD21B =	006100	1226	1227#															
AD21C	006210	1233*	1256#															
AD22A	006336	1290#																
AD22B	006424	1308#																
AD22C	006506	1328#																
AD22D	006602	1349#																
AD22E	006702	1372#																
BELL	017402	2713	2928#															
BLKL	011710	1976	1979#	1995														
BLK20	005576	1161	1164#	1180														
BLOCKS	002034	441#	1156*	1179*	1972*	1994*												
CK	017012	2827	2840#															
CLRALL	016274	520	556	590	638	670	974	1003	1039	1061	1080	1119	1206	1277				
		1397	1445	1487	1548	1850	1870	1891	1933	2230	2447	2539	2631	2730#				
CLRLP	016312	2734#	2736															
CNT22B	006374	1295	1303#															
CNT22C	006454	1313	1321#															
CNT22D	006550	1333	1341#															
CNT22E	006644	1354	1363#															
CNT31A	012214	2049	2051#															
CNT31B	012304	2069	2079#															
CNT31C	012342	2092#																
CNT32A	012556	2134	2136#															
CNT32B	012652	2155	2165#															
CNT32C	012710	2178#																
CONT12	004060	854	863#															
CONT5	002664	602	606	612#														
CRLF	017442	2832	2940#															
DATA16	004754	1006	1013	1015	1025#	1044	1064	1083	1853	1873	1894							

K123	001414	297#	1126	1940					
K134	001416	298#	1127	1941					
K200	001412	296#	806	827	847				
LOGIC	016246	2718#							
LOOP10	003252	707#	711						
LOOP11	003430	754#	758						
LOOP12	003600	796#	895						
LOOP2	002260	492#	502						
LOOP3	002400	523#	538						
LOOP3A	002406	525#	537						
LOOP3B	002412	526#	534						
LOOP4	002510	558#	576						
LOOP4A	002516	561#	574						
LOOP4B	002522	562#	571						
LOOP6	002770	641#	657						
LOOP6A	002776	643#	656						
LOOP7	003114	673#	689						
LOOP7A	003122	675#	688						
LOP10A	003260	709#	710						
LOP10B	003306	717#	727						
LOP10C	003314	719#	726						
LOP11A	003436	756#	757						
LOP11B	003464	764#	772						
LOP11C	003472	766#	771						
LOP31A	012162	2039#	2050						
LOP31B	012166	2041#	2046						
LOP31C	012220	2052#	2096						
LOP31D	012232	2056#	2094						
LOP31E	012262	2067#	2084						
LOP31F	012266	2068#	2080						
LOP32A	012524	2124#	2135						
LOP32B	012530	2126#	2131						
LOP32C	012562	2137#	2182						
LOP32D	012600	2142#	2180						
LOP32E	012630	2153#	2170						
LOP32F	012634	2154#	2166						
LOP5A	002626	594#	621						
LOP5B	002644	601#	613						
MSKB	002022	436#							
NG35A	013774	2334	2340#						
NG35B	014100	2352	2359#						
NG35C	014214	2372	2379#						
NG35D	014332	2388	2400#						
NG35E	014450	2415	2421#						
NOP	= 000240	253#							
ODDAD	014356	2399	2404	2408#					
OK35A	014004	2332	2345#						
OK35B	014124	2351	2365#						
OK35C	014224	2371	2383#						
OK35D	014346	2387	2405#						
OK35E	014460	2413	2426#						
PAGEL	011652	1972#	2000						
PAGES	002006	430#	1151*	1159	1182*	1965*	1974	1997*	
PAG20	005540	1156#	1185						
PARTAB	001762	418#	521	639	705	715			
PATCH1=	104000	2914#	2917						

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 61
DCKTAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

PATCH2=	104002	2915#	2918											
PATCH3=	104004	2916#	2919											
PC	=%000007	263#	2912*											
PDREND	001760	417#	575											
PDRMSK	002030	439#	563											
PDRTAB	001754	415#	557	671	752	762								
PRFLG	017314	2859#	2866#	2886	2888#	2894#	2900#							
PRINT	016706	2824#	2920											
PROCT	017062	2834	2839	2852	2857#									
PRSFLG	017312	2850#	2857#	2867	2880#	2899#								
PRSHRT	017030	2850#												
PS	= 177776	265#	266	445*	447*	449*	451*	460*	483*	511*	548*	582*	629*	662*
		694#	742*	786*	864*	869*	877*	882*	894*	903*	921*	966*	995*	1031*
		1068#	1076*	1087*	1094*	1111*	1134*	1136*	1138*	1149*	1188*	1197*	1269*	1286*
		1292	1299*	1388*	1435*	1479*	1540*	1576*	1598*	1620*	1645*	1662*	1685*	1710*
		1733*	1750*	1775*	1798*	1821*	1841*	1866*	1877*	1887*	1898*	1908*	1925*	1947*
		1949*	1951*	1962*	2003*	2019*	2033*	2035*	2037*	2055*	2104*	2118*	2120*	2122*
		2141*	2190*	2222*	2232*	2234*	2236*	2310*	2324*	2326*	2328*	2336*	2355*	2375*
		2391*	2412*	2417*	2430*	2439*	2531*	2623*	2780*	2802*	2804*	2824*	2831*	
PTEMP1	017316	2833*	2838*	2851	2861	2864*	2865*	2890*	2892*	2893*	2895	2901#		
PTEMP2	017320	2860*	2863*	2869	2673	2895*	2896*	2897*	2902#					
PTEMP3	017322	2858*	2881*	2882	2903#									
P.CK	017132	2867#	2898											
P.CNT1	017230	2883	2685#											
P.CONT	017212	2870	2881#											
P.WAIT	017150	2868	2871#	2872										
RETRNK	016550	2783*	2784*	2788	2790	2791#								
RETURN	016704	453*	2787*	2808	2812*	2819#								
RET33	013042	2201	2207#											
RT26A	010200	1575	1587#											
RT26B	010236	1597	1609#											
RT26C	010274	1619	1631#											
RT26D	010342	1644	1654#											
RT26E	010404	1661	1673#											
RT26F	010450	1684	1696#											
RT26G	010514	1709	1721#											
RT26H	010562	1732	1742#											
RT26I	010624	1749	1761#											
RT26J	010670	1774	1786#											
RT26K	010734	1797	1809#											
RT26L	011002	1820	1830#											
RWALL	016324	2028	2113	2198	2318	2740#								
RWDSP	016430	1934	2763#											
RWD1	016440	2765#	2771											
RWD2	016444	2766#	2768											
RWISP	016366	1120	2751#											
RWI1	016376	2753#	2759											
RWI2	016402	2754#	2756											
RWL1	016334	2742#	2748											
RWL2	016340	2743#	2745											
RO	=%000000	254#	491*	492	501*	522*	538*	561*	562	564	570*	640*	657*	672*
		689*	706*	711*	716*	727*	753*	758*	763*	772*	847*	850	1145*	1146*
		1152*	1157*	1162*	1176*	1181*	1184	1958*	1959*	1966*	1973*	1977*	1991*	1996*
		1999	2051*	2052	2055	2095	2136*	2137	2141	2181	2732*	2734*	2735	2742*
		2745*	2753*	2756*	2765*	2768*								
R1	=%000001	255#	492*	493	494*	496	523*	526*	527	535*	536	558*	564*	565

MOS

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 65
 DCKTAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

TST11F	003526	750*	773	775*	778#									
UBRK	002020	435#	2807*	2815*										
UDPAR0	001534	339#	1892*	2245*										
UDPAR1	001536	340#												
UDPAR2	001540	341#												
UDPAR3	001542	342#												
UDPAR4	001544	343#												
UDPAR5	001546	344#												
UDPAR6	001550	345#												
UDPAR7	001552	346#	1568*	1895*	2116*	2255*	2322*							
UDPDR0	001474	323#	1893*	2242*										
UDPDR1	001476	324#												
UDPDR2	001500	325#												
UDPDR3	001502	326#												
UDPDR4	001504	327#												
UDPDR5	001506	328#												
UDPDR6	001510	329#												
UDPDR7	001512	330#	1562*	1896*	2254*									
UIPAR0	001514	331#	1081*											
UIPAR1	001516	332#												
UIPAR2	001520	333#												
UIPAR3	001522	334#												
UIPAR4	001524	335#												
UIPAR5	001526	336#												
UIPAR6	001530	337#												
UIPAR7	001532	338#	1084*	1565*	2031*									
UIPDR0	001454	315#	426	1082*	1553*	2239*								
UIPDR1	001456	316#	1554*											
UIPDR2	001460	317#												
UIPDR3	001462	318#												
UIPDR4	001464	319#												
UIPDR5	001466	320#												
UIPDR6	001470	321#												
UIPDR7	001472	322#	1085*	1559*										
URET34	013426	2266	2271#	2284										
USTACK	000400	289#	450	1137	1950	2036	2121	2327						
UTRP34	013422	2269#												
XLOOP	016540	2787	2789#											
.	= 017714	274#	276#	279#	284#	288#	290#	292#	413	455	464	469	472	475
		487	498	515	528	552	566	586	633	646	652	666	678	684
		698	713	720	746	760	767	790	800	811	818	823	831	842
		852	859	871	884	907	913	925	932	936	940	944	947	952
		957	970	984	999	1009	1018	1035	1051	1055	1072	1073	1091	1092
		1115	1166	1171	1201	1220	1222	1227	1228	1238	1242	1246	1273	1293
		1311	1331	1352	1375	1392	1413	1419#	1423	1429	1439	1457	1462#	1468
		1483	1503	1507	1511	1513	1516	1519	1523	1526	1531	1544	1845	1861
		1882	1884	1903	1905	1929	1981	1986	2023	2042	2060	2072	2088	2108
		2127	2146	2158	2174	2194	2208	2226	2273	2276	2279	2282	2285	2288
		2314	2443	2465	2480	2483	2486	2489	2497	2507	2510	2514	2518	2535
		2557	2572	2575	2578	2581	2589	2599	2602	2606	2610	2627	2649	2664
		2667	2670	2673	2681	2691	2694	2698	2702	2806	2814	2826	2836	2841
		2854	2862	2875	2879	2887	2929	2932	2936	2941	2944	2947#		

ADC	2866	2894																	
ADD	1146	1147	1150	1177	1178	1959	1960	1963	1992	1993	2047	2081	2093	2132	2139				
ASL	2167	2179	2746	2757	2769	2784	2911												
BCC	501	532	570																
BEQ	502	571	2094	2180															
	464	472	475	487	498	515	528	552	566	586	602	606	618	633	646				
	652	666	678	684	698	720	746	767	790	800	811	818	823	831	842				
	852	859	871	884	907	913	925	932	936	940	947	952	957	970	984				
	999	1018	1035	1051	1055	1072	1091	1115	1166	1171	1183	1201	1222	1228	1238				
	1242	1246	1273	1293	1311	1331	1352	1375	1392	1413	1423	1439	1457	1483	1503				
	1507	1511	1516	1519	1523	1526	1531	1544	1845	1861	1882	1903	1929	1981	1986				
	1998	2023	2042	2069	2072	2088	2108	2127	2155	2158	2174	2194	2208	2226	2273				
	2276	2279	2282	2285	2288	2314	2443	2465	2480	2483	2486	2489	2497	2507	2510				
	2514	2518	2535	2557	2572	2575	2578	2581	2589	2599	2602	2606	2610	2627	2649				
	2664	2667	2670	2673	2681	2691	2694	2698	2702	2716	2806	2814	2826	2858	2870				
	2887																		
BGT	2049	2134																	
BIC	494	495	563	869	882	1163	1463	1978	2463	2478	2505	2555	2570	2597	2647				
BIS	2662	2689	2896																
BIT	2470	2562	2654	2897															
	983	1054	1292	1411	1456	1530	2059	2071	2087	2145	2157	2173	2795	2797	2805				
	2813	2825																	
BLE	576	1187	2002																
BLT	2084	2096	2170	2182	2748	2759	2771												
BMI	891																		
BNE	534	729	774	1009	1161	1180	1289	1307	1327	1348	1371	1976	1995	2060	2146				
	2796	2798	2852	2883															
BPL	469	713	760	944	2800	2836	2841	2854	2862	2872	2875	2879	2929	2932	2936				
	2941	2944																	
BR	455	621	732	777	854	1024	1073	1092	1185	1252	1295	1313	1333	1354	1377				
	1418	1429	1460	1468	1499	1513	1586	1608	1630	1653	1672	1695	1720	1741	1760				
	1785	1808	1829	1884	1905	2000	2050	2135	2206	2344	2364	2382	2399	2404	2425				
	2462	2475	2504	2554	2567	2596	2646	2659	2688	2827	2898								
CLC	2885																		
CLR	445	451	460	483	503	511	535	548	572	582	611	619	629	662	694				
	703	742	750	786	795	805	815	826	835	846	857	863	876	889	903				
	921	966	989	995	1031	1049	1070	1076	1089	1094	1111	1138	1154	1175	1181				
	1188	1189	1197	1215	1233	1236	1269	1290	1291	1296	1299	1303	1308	1309	1314				
	1329	1334	1350	1355	1372	1373	1378	1388	1417	1435	1479	1492	1498	1514	1529				
	1533	1540	1569	1833	1835	1841	1859	1865	1866	1880	1886	1887	1901	1907	1908				
	1925	1951	1969	1990	1996	2003	2004	2019	2032	2037	2058	2097	2098	2104	2122				
	2144	2183	2184	2190	2202	2212	2222	2236	2247	2259	2265	2271	2292	2293	2294				
	2295	2296	2297	2298	2299	2310	2328	2341	2346	2348	2350	2360	2366	2370	2373				
	2374	2375	2380	2384	2396	2397	2401	2402	2406	2407	2409	2411	2412	2414	2416				
	2422	2427	2430	2439	2456	2460	2473	2494	2502	2521	2522	2531	2548	2552	2565				
	2586	2594	2613	2614	2623	2640	2644	2657	2678	2686	2705	2706	2725	2726	2730				
	2731	2732	2734	2740	2743	2751	2754	2763	2766	2780	2786	2789	2809	2857	2858				
	2859	2880	2888																
CLRB	644	650	676	682	950	955	1235	2910											
CMP	463	486	497	514	527	533	551	565	575	585	601	616	632	645	651				
	665	677	683	697	719	745	766	789	799	810	830	841	851	858	870				
	883	906	912	924	931	935	939	956	969	998	1008	1017	1034	1050	1071				
	1090	1114	1159	1165	1170	1186	1200	1219	1237	1245	1272	1288	1306	1325	1330				
	1345	1351	1368	1391	1420	1438	1482	1502	1506	1510	1515	1518	1525	1528	1543				
	1593	1615	1637	1656	1679	1702	1727	1744	1767	1792	1815	1832	1844	1860	1881				
	1902	1928	1974	1980	1985	2001	2022	2041	2048	2068	2083	2095	2107	2126	2133				

C06

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 70
 DCKTAB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	2154	2169	2181	2193	2207	2225	2272	2275	2278	2281	2284	2287	2313	2340	2345
	2359	2365	2379	2383	2400	2405	2421	2426	2442	2464	2468	2479	2482	2485	2488
	2492	2506	2509	2513	2517	2520	2534	2556	2560	2571	2574	2577	2580	2584	2598
	2601	2605	2609	2612	2626	2648	2652	2663	2666	2669	2672	2676	2690	2693	2697
	2701	2704	2747	2758	2770	2799	2803	2830	2869	2882					
CMPB	1226														
COM	893														
DEC	1176				1994	1997									
EMT	2357	1179	1182	1991											
HALT	274	2914	2915	2916											
	1451	985	1010	1020	1052	1167	1172	1405	1406	1407	1408	1409	1410	1424	1428
INC	730	1452	1453	1466	1497	1863	1982	1987	2205	2782	2785	2842			
	1751	775	808	849	1404	1495	1501	1577	1599	1621	1646	1663	1686	1711	1734
	2457	1776	1799	1822	2056	2142	2203	2256	2262	2268	2337	2356	2376	2392	2418
IOT	2257	2500	2549	2592	2641	2684	2801	2810	2863	2881					
JMP	285	2263	2269	2377											
JSR	462	895	2723	2788	2790	2808									
	905	485	513	520	550	556	584	590	631	638	664	670	696	744	788
	1271	923	968	974	997	1003	1033	1039	1061	1080	1113	1119	1120	1199	1206
	1933	1277	1390	1397	1437	1445	1481	1487	1542	1548	1843	1850	1870	1891	1927
	2539	1934	2021	2028	2106	2113	2192	2198	2224	2230	2312	2318	2441	2447	2533
MOV	446	2625	2631	2713	2718	2832	2834	2839							
	493	447	448	449	450	452	453	454	459	461	477	482	484	491	492
	559	496	510	512	519	521	522	523	524	525	526	547	549	557	558
	640	561	562	564	581	583	592	594	597	598	623	628	630	637	639
	702	641	642	643	649	661	663	671	672	673	674	675	681	693	695
	752	704	705	706	707	708	709	715	716	717	718	731	741	743	751
	806	753	754	755	756	762	763	764	765	776	785	787	794	796	798
	866	807	816	821	827	828	836	837	838	839	840	847	848	864	865
	930	867	868	877	878	879	880	881	894	902	904	911	920	922	929
	1007	934	938	942	949	954	965	967	977	979	994	996	1004	1005	1006
	1064	1013	1015	1016	1030	1032	1040	1042	1044	1045	1046	1047	1048	1062	1063
	1112	1065	1066	1067	1068	1069	1081	1082	1083	1084	1085	1086	1087	1088	1110
	1151	1122	1123	1124	1125	1126	1127	1133	1134	1135	1136	1137	1139	1145	1149
	1209	1152	1153	1155	1156	1157	1162	1164	1190	1191	1196	1198	1205	1207	1208
	1278	1210	1211	1212	1213	1214	1216	1217	1218	1232	1249	1250	1251	1268	1270
	1342	1279	1280	1281	1285	1286	1287	1304	1305	1321	1322	1324	1328	1336	1341
	1401	1344	1349	1358	1363	1364	1366	1367	1381	1387	1389	1396	1398	1399	1400
	1491	1402	1434	1436	1443	1446	1447	1448	1449	1450	1478	1480	1488	1489	1490
	1558	1493	1500	1534	1539	1541	1549	1550	1551	1552	1553	1554	1555	1556	1557
	1619	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568	1575	1576	1597	1598
	1732	1620	1641	1644	1645	1658	1661	1662	1681	1684	1685	1706	1709	1710	1729
	1840	1733	1746	1749	1750	1771	1774	1775	1794	1797	1798	1817	1820	1821	1834
	1875	1842	1849	1851	1852	1853	1854	1855	1856	1857	1858	1871	1872	1873	1874
	1926	1876	1877	1878	1879	1892	1893	1894	1895	1896	1897	1898	1899	1900	1924
	1962	1935	1936	1937	1938	1939	1940	1941	1946	1947	1948	1949	1950	1952	1958
	2030	1965	1966	1967	1971	1972	1973	1977	1979	2005	2006	2018	2020	2027	2029
	2086	2031	2033	2034	2035	2036	2038	2039	2051	2052	2054	2055	2057	2066	2067
	2137	2103	2105	2112	2114	2115	2116	2117	2118	2119	2120	2121	2123	2124	2136
	2231	2140	2141	2143	2152	2153	2172	2189	2191	2199	2200	2201	2213	2221	2223
	2248	2232	2233	2234	2235	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246
	2319	2249	2250	2251	2252	2253	2254	2255	2260	2261	2266	2267	2300	2309	2311
	2336	2320	2321	2322	2323	2324	2325	2326	2327	2329	2331	2332	2333	2334	2335
	2385	2347	2349	2351	2352	2353	2354	2355	2361	2362	2367	2368	2369	2371	2372
	2438	2386	2387	2388	2389	2390	2391	2393	2408	2410	2413	2415	2417	2428	2429
	2501	2440	2448	2449	2450	2451	2452	2454	2455	2458	2467	2469	2471	2493	2499
		2523	2530	2532	2540	2541	2542	2543	2544	2546	2547	2550	2559	2561	2563

	2585	2591	2593	2615	2622	2624	2632	2633	2634	2635	2636	2638	2639	2642	2651
	2653	2655	2677	2683	2685	2707	2715	2733	2741	2742	2744	2752	2753	2755	2764
	2765	2767	2781	2783	2787	2802	2804	2812	2824	2828	2829	2831	2833	2837	2838
	2850	2855	2860	2873	2876	2895	2907	2909	2912	2930	2933	2942	2945		
MOV8	809	829	850	2807	2815										
NOP	1454	1455	2258	2264	2270	2394	2395	2719	2720	2721					
RESET	470	714	761	945	980	1012	1023	1057	1075	1093	1427	1467	1864	1885	1906
	2717	2924													
ROL	2864	2865	2890	2892	2893										
RTI	2816	2843													
RTS	2727	2737	2749	2760	2772	2856	2884	2937	2946						
RTT	2925														
SEC	2889														
SOB	537	538	574	613	656	657	688	689	710	711	726	727	757	758	771
	772	2046	2080	2131	2166	2736	2745	2756	2768						
SUB	1403	1444	2908												
TRAP	252	2338													
TST	471	474	536	573	603	612	655	687	725	728	770	773	797	817	822
	890	946	951	1184	1241	1310	1374	1496	1522	1578	1600	1622	1647	1664	1687
	1712	1735	1752	1777	1800	1823	1999	2079	2092	2165	2178	2204	2419	2735	2840
TSTB	2851	2861	2867	2886											
	468	712	759	943	2496	2588	2680	2835	2853	2871	2874	2878	2928	2931	2935
	2940	2943													
.ABS	1														
.END	2950														
.ENDC	2017	2033	2054	2102	2118	2140									
.IFF	2017	2033	2100	2114											
.IFNZ	2054	2139													
.IFZ	2015	2029	2100	2114											
.LIST	1	274	443	459	482	510	547	581	628	661	693	741	785	902	920
	965	994	1030	1110	1196	1268	1387	1434	1478	1539	1840	1924	2015	2018	2100
	2103	2189	2221	2309	2438	2530	2622								
.MACR	443	1536	2014	2432											
.NLIST	1	274	443	459	482	510	547	581	628	661	693	741	785	902	920
	965	994	1030	1110	1196	1268	1387	1434	1478	1539	1840	1924	2015	2018	2100
	2103	2189	2221	2309	2438	2530	2622								
.REM	1														
.REPT	274														
.TITLE	1														
.WORD	294														

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*DCKTAB, DCKTAB, SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DCKTAB.P11
 RUN-TIME: 12 22 4 SECONDS
 RUN-TIME RATIO: 167/40=4.1
 CORE USED: 11K (21 PAGES)

