

# KT11-C

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

EP-DCKTA-B-DL-A

OCT 1978

COPYRIGHT ©1976

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 pattern of vertical lines and characters, representing digital logic test results. The data is organized into columns, with some frames showing a header section at the top. The overall appearance is that of a technical document used for testing and verification of digital systems.

801

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 K11-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 K11-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.

# E01

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 4  
DCKTAB.P11

MAINDEC-11-DCKTA-A  
PAGE 04

## 5.2.4 EMTSRV (EMT DECODER)

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

## 5.2.5 CLRALL

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

## 5.2.6 RWALL

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

## 5.2.7 RWISP

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

## 5.2.8 RWOSP

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.

```
*
: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
```



DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 8  
 DCKTAB.P11

000200	000167	001632	JMP	START	
					;LOAD DATA AREA
	000400				. =400
000400	000000		USTACK:	0	
	001000				. =.+376
001000	000000		SSTACK:	0	
	001400				. =.+376
001400	000000		KSTACK:	0	
001402	000000	000000 000000			.WORD 0,0,0,0
001410	000000				
001412	000200		K200:	200	;CONSTANTS
001414	123456		K123:	123456	
001416	134567		K134:	134567	
001420	100000		KNR:	100000	
001422	177564		TCSR:	177564	;TELETYPE PRINTER CSR ADDRESS
001424	177566		TDBR:	177566	
001426	000000		TEMPX:	0	;TEMPORARY STORAGE LOCATIONS
001430	000000		TEMP1:	0	
001432	000000		TEMP2:	0	
001434	177572		SRO:	177572	;KT11-C STATUS REGISTER ADDRESSES
001436	177573		SROH:	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	
001454			ADRTAB:		
001454	177600		UIPDR0:	177600	;USER I-SPACE PAGE DESCRIPTOR REGISTERS
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	;USER D-SPACE PAGE DESCRIPTOR REGISTERS
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	;USER I-SPACE PAGE ADDRESS REGISTERS
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	;USER D-SPACE PAGE ADDRESS REGISTERS
001536	177662		UDPAR1:	177662	

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	

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  
001752

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  
SAVER: 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 SRI 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,SRI
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
002166 000005                      RESET                    ;ISSUE INIT
002170 005777 177240                      TST     @SRO              ;CHECK SRO
002174 001401                      BEQ     .+4
002176 104006                      HLT                       ;SRO WAS NOT INITIALIZED TO ZERO
002200 005777 177234                      TST     @SRI              ;CHECK SRI
002204 001401                      BEQ     .+4
002206 104006                      HLT                       ;SRI 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,SRI
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
                                           ; SETUP TO CHECK NEXT BIT POSITION
002422 006304          ASL    R4            ; ALL R/W BITS IN THIS PAR ALREADY CHECKED?
002424 020427 010000          CMP    R4,#10000     ; NO-BRANCH TO CONTINUE
002430 001370          BNE    LOOP3B        ; YES-CLEAR PAR JUST TESTED
002432 005011          CLR    @R1          ; MOVE POINTER TO NEXT PAR
002434 005721          TST   (R1)+         ; TEST ALL PAR'S IN SET
002436 077215          SOB   R2,LOOP3A     ; TEST ALL 3 REGISTER SETS
002440 077021          SOB   R0,LOOP3

;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    #PORTAB,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   ; R5 CONTAINS EXPECTED RESULTING CONTENTS OF PDR
002524 046705 177300          BIC   PDRMSK,R5     ; LOAD PDR
002530 010011          MOV   R0,@R1
    
```

# NO1

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 13  
 DCKTAB.P11

002532	021105			CMP	@R1,R5	;CHECK RESULTING CONTENTS OF PDR
002534	001401			BEQ	.+4	
002536	104006			HLT		
002540	006300			ASL	R0	;PDR WHOSE ADDRESS IS IN R1
002542	103367			BCC	LOOP4B	;WAS INCORRECT AFTER VALUE IN R0
002544	005011			CLR	@R1	;WAS LOADED INTO IT
002546	005721			TST	(R1)+	;ROTATE BIT
002550	077216			SOB	R2,LOOP4A	;BRANCH IF NOT DONE WITH THIS PDR
002552	020327	001760		CMP	R3,#PDREND	;IF DONE WITH THIS PDR, CLEAR IT
002556	003754			BLE	LOOP4	;MOVE POINTER TO ADDRESS NEXT PDR
;NO DUAL ADDRESSING TEST FOR PAR'S AND PDR'S						
;SHOW THAT EACH PAR AND EACH PDR RESPONDS TO ONLY ONE ADDRESS						
002560	104400			TESTS:	SCOPE	
002562	012737	000005	177570	MOV	#5,@#SR	;DISPLAY TEST NUMBER
002570	005037	177776		CLR	@#PS	;INITIALIZE PROCESSOR STATUS
002574	012706	001400		MOV	#KSTACK,SP	;INITIALIZE KERNEL STACK POINTER
002600	004767	013456		JSR	%7,SETUP	;INITIALIZE SR0,SR3
002604	026727	177222	000005	CMP	TESTCT,#5	;IS THIS TEST BEING EXECUTED IN THE
002612	001401			BEQ	.+4	;CORRECT SEQUENCE?- BRANCH IF YES
002614	104006			HLT		;TEST EXECUTED OUT OF SEQUENCE- TESTCT
;CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE						
002616	004767	013452		JSR	%7,CLRALL	;INITIALIZE ALL PAR'S AND PDR'S
002622	012701	001454		MOV	#ADRTAB,R1	;TO ZERO
002626	012702	001454	LOP5A:	MOV	#ADRTAB,R2	;R1 POINTS TO ADDRESS OF PAR OR PDR
;LOADED						
;R2 USED AS A POINTER TO CYCLE THRU						
;ALL OTHER ADDRESSES OF PAR'S AND PDR'S						
;TO CHECK FOR DUAL ADDRESSING						
002632	012703	000140		MOV	#96,R3	;R3 USED AS A COUNTER
002636	012771	010421	000000	MOV	#10421,@(R1)	;LOAD THE PAR OR PDR WHOSE ADDRESS IS IN R1
;SET ONE BIT IN EACH CHIP (4 BITS PER CHIP)						
;IF READ/WRITE						
002644	020201		LOP5B:	CMP	R2,R1	;IF R1 CONTAINS ADDRESS OF
002646	001406			BEQ	CONT5	;LOCATION LOADED, SKIP CHECKING IT
002650	005772	000000		TST	@(R2)	;OTHERWISE, CHECK TO SEE IF THIS
;REGISTER RESPONDED TO THE ADDRESS						
;OF THE ONE LOADED AS A DUAL						
;BRANCH IF OK						
002654	001403			BEQ	CONT5	;DUAL ADDRESSING - ADDRESS POINTED
002656	104006			HLT		;TO BY R2 RESPONDED TO THE ADDRESS
;POINTED TO BY R1 IN AT LEAST ONE						
;4 BIT SECTION (1 CHIP)						
;REINITIALIZE FAULTY LOCATION						
002660	005072	000000		CLR	@(R2)	;MOVE POINTER R2
002664	005722		CONT5:	TST	(R2)+	;CHECK ALL PAR'S AND PDR'S
002666	077312			SOB	R3,LOP5B	;TO SEE IF THEY RESPONDED TO THE
;ADDRESS POINTED TO BY R1						
002670	022701	001752		CMP	#ADREND,R1	;HAVE ALL ADDRESSES BEEN CHECKED
;FOR DUALS?						
002674	001402			BEQ	DONE5	;YES - BRANCH TO DONE
002676	005031			CLR	@(R1)+	;NO - MOVE POINTER R1 TO ADDRESS OF
;NEXT PAR OR PDR						

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 14  
 DCKTAB.P11

```

002700 000752          BR      LOP5A          :CHECK FOR DUALS OF THE
002702 012767 000100 013770 DONE5: 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                    :TEST EXECUTED OUT OF SEQUENCE- TESTCT
002744 104006          :CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

002746 012767 002000 013724          MOV      #2000, ICOUNT :RESTORE ITERATION COUNT
002754 004767 013314          JSR     %7, CLRALL   :INITIALIZE KTI1-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                    :TEST EXECUTED OUT OF SEQUENCE- TESTCT
003076 104006          :CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE

003100 004767 013170          JSR     %7, CLRALL   :INITIALIZE KTI1-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

```

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 P1 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	#252, 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	ATCSR	: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



DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 16  
DCKTAB.P11

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 @TCR  
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

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 17  
 DCKTAB.P11

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 C
003600	016703	175630		LOOP12:	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)
003656	005077	175552			CLR	@SRO				: CLEAR NR ERROR BIT TO RESUME TRACKING
003662	016777	175532	175544		MOV	KNR, @SRO				: SET NR ERROR BIT
003670	005777	175544			TST	@SR1				: CHECK SR1
003674	001401				BEQ	.+4				
003676	104006				HLT					: SR1 INCORRECT - SHOULD SHOW NO
										: REGISTERS CHANGED
003700	012777	177777	175532		MOV	#-1, @SR1				: TRY TO WRITE SR1 - SHOULD BE READ-ONLY
003706	005777	175526			TST	@SR1				
003712	001401				BEQ	.+4				
003714	104006				HLT					: SR1 WAS ALTERED BY WRITING IT
										: WHILE NR ERROR WAS STILL SET
003716	005077	175512			CLR	@SRO				: CLEAR NR ERROR BIT TO RESUME TRACKING
003722	012704	001413			MOV	#K200+1, R4				: SETUP R4 TO SET NR BIT
003726	016705	175504			MOV	SROH, R5				: SETUP R5 TO REFERENCE SRO (HIGH BYTE)
003732	114425				MOVB	-(R4), (R5)+				: SET NR ERROR BIT
003734	022777	006774	175476		CMP	#6774, @SR1				: CHECK 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)
003746	005077	175462			CLR	@SRO				: CLEAR NR ERROR BIT TO RESUME TRACKING
003752	010667	175450			MOV	R6, TEMPX				: SAVE STACK POINTER
003756	012706	001422			MOV	#KNR+2, R6				: SETUP R6 TO SET NR BIT
003762	016703	175446			MOV	SRO, R3				: SETUP R3 TO REFERENCE SRO
003766	014623				MOV	-(R6), (R3)+				: SET NR BIT
003770	016706	175432			MOV	TEMPX, R6				: RESTORE STACK POINTER
003774	022777	011766	175436		CMP	#11766, @SR1				: CHECK SR1
004002	001401				BEQ	.+4				
004004	104006				HLT					: SR1 INCORRECT - SHOULD HAVE TRACKED
										: CHANGE OF R6 BY -2 (LOW BYTE)

004006	005077	175422		CLR	2SR0		;AND OF R3 BY +2 (HIGH BYTE)
004012	012700	001413		MOV	#K200+1,R0		;CLEAR NR ERROR BIT TO RESUME TRACKING
004016	016705	175414		MOV	SROH,R5		;SETUP R0 TO SET NR ERROR BIT
004022	005205			INC	R5		;SETUP R5 TO REFERENCE SRO (HIGH BYTE)
004024	114045			MOVB	-(R0),-(R5)		;SET NR ERROR BIT
004026	022777	176770	175404	CMP	#176770,2SR1		;CHECK SR1
004034	001402			BEQ	.+6		
004036	104006			HLT			;SR1 INCORRECT - SHOULD HAVE
004040	000407			BR	CONT12		;TRACKED CHANGE OF R0 BY -1
							;(LOW BYTE) AND OF R5 BY -1 (HIGH
							BYTE)
							;SHOW THAT SR1 IS READ-ONLY
							;CHECK SR1
							;SR1 NOT READ-ONLY - ALTERED BY A CLEAR
							INSTRUCTION WHILE AN ERROR BIT (NR)
							WAS STILL SET IN SRO
							;CLEAR NR ERROR BIT TO RESUME TRACKING
							;SET MODE TO SUPERVISOR
							;SAVE SUPERVISOR STACK POINTER
							;SET SUPERVISOR R6 TO REFERENCE SRO
							;SET NR BIT
							;RESTORE SUPERVISOR STACK POINTER
							;RETURN TO KERNEL
							;CHECK SR1
							;SR1 INCORRECT WHILE TRACKING IN
							SUPERVISOR MODE-SHOULD SHOW
							CHANGE OF R7 BY +2 (LOW BYTE)
							AND CHANGE OF R6 BY +2 (HIGH BYTE)
							;CLEAR NR ERROR BIT TO RESUME TRACKING
							;SET MODE TO USER
							;SAVE USER STACK POINTER
							;SET USER R6 TO REFERENCE SRO
							;SET NR ERROR BIT
							;RESTORE USER STACK POINTER
							;RETURN TO KERNEL
							;CHECK SR1
							;SR1 INCORRECT WHILE TRACKING IN
							USER MODE-SHOULD SHOW CHANGE OF
							R7 BY +2 (LOW BYTE) AND CHANGE
							OF R6 BY +2 (HIGH BYTE)
							;CLEAR NR BIT TO RESUME TRACKING
							;CHECK FLAG TO SEE IF REGISTER SET 1 WAS TESTED
							;BRANCH IF SET-BOTH REGISTER SETS
							ALREADY CHECKED
							;OTHERWISE SET FLAG
							;SET REGISTER SET BIT
							;REPEAT TEST USING OTHER SET OF
							REGISTERS

;SR2 SHOULD CONTAIN ADDRESS OF LAST FETCH  
;WITH KTI1-C TURNED OFF

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 19  
 DCKTAB.P11

```

004234 104400          TEST13: SCOPE
004236 012737 000013 177570      MOV      #13, @SR
004244 005037 177776          CLR      @PS
004250 012706 001400          MOV      *KSTACK, SP
004254 004767 012002          JSR     %7, SETUP
004260 026727 175546 000013      CMP     TESTCT, #13
004266 001401          BEQ     .+4
004270 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

004272 017701 175146          AD13:  MOV     @SR2, R1
004276 022701 004272          CMP     #AD13, R1
004302 001401          BEQ     .+4
004304 104006          HLT

; PICK UP SR2 - SHOULD CONTAIN ADDRESS
; OF THIS INSTRUCTION
; SR2 DID NOT CONTAIN ADDRESS OF
; INSTRUCTION BEING EXECUTED

; SHOW THAT INIT CLEARS SR3, AND THAT BITS 0-2 CAN BE SET AND CLEARED
; ALSO SHOW THAT BYTE ADDRESSING OF SR3 WORKS
004306 104400          TEST14: SCOPE
004310 012737 000014 177570      MOV      #14, @SR
004316 005037 177776          CLR      @PS
004322 012706 001400          MOV      *KSTACK, SP
004326 004767 011730          JSR     %7, SETUP
004332 026727 175474 000014      CMP     TESTCT, #14
004340 001401          BEQ     .+4
004342 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

004344 012767 000010 012326      MOV      #10, ICOUNT
004352 012777 000001 175070      MOV      #1, @SR3
004360 022777 000001 175062      CMP      #1, @SR3
004366 001401          BEQ     .+4
004370 104006          HLT
; SR3 INCORRECT WHEN SET TO 1

004372 012777 000002 175050      MOV      #2, @SR3
004400 022777 000002 175042      CMP      #2, @SR3
004406 001401          BEQ     .+4
004410 104006          HLT
; SR3 INCORRECT WHEN SET TO 2

004412 012777 000004 175030      MOV      #4, @SR3
004420 022777 000004 175022      CMP      #4, @SR3
004426 001401          BEQ     .+4
004430 104006          HLT
; SR3 INCORRECT WHEN SET TO 4

004432 012777 000007 175010      MOV      #7, @SR3
004440 105777 174756          TSTB   @TCR
004444 100375          BPL
004446 000005          RESET
004450 005777 174774          TST    @SR3
004454 001401          BEQ     .+4
004456 104006          HLT
; SR3 NOT CLEARED BY INIT

004460 012777 000007 174762      MOV      #7, @SR3
004466 105077 174756          CLRB   @SR3
004472 005777 174752          TST    @SR3
004476 001401          BEQ     .+4
004500 104006          HLT
; SR3 INCORRECT AFTER A CLRB (LOW)

004502 012777 000007 174740      MOV      #7, @SR3
004510 105077 174736          CLRB   @SR3H
004514 022777 000007 174726      CMP     #7, @SR3
; CLEAR HIGH BYTE OF SR3

```

004522 001401  
 004524 104006

BEQ .+4  
 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(8)  
 ;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      SR2,SR1          ;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      SRPS            ;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,SRIPAR0      ;MAP KERNEL PAGE 0 I-SPACE
                                ;TO BANK 0 OFFSET BY 1 PAGE
005026 012777 077406 174620  MOV      #77406,SRIPDR0  ;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,SRIPAR7   ;MAP KERNEL PAGE 7 I-SPACE
005046 012777 077406 174616  MOV      #77406,SRIPDR7  ;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      SR2            ;CLEAR SR0 THRU KERNEL PAGE 7
005070 021111                CMP      SR1,SR1        ;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,SRIPAR0      ;MAP SUPERVISOR 0 I-SPACE
005124 012777 077406 174422  MOV      #77406,SRIPDR0  ;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,SRIPAR7   ;MAP SUPERVISOR 7 I-SPACE
005144 012777 077406 174420  MOV      #77406,SRIPDR7  ;TO THE EXTERNAL BANK, RW,4K
005152 016702 174256                MOV      SR0,R2         ;SETUP R2 TO ADDRESS SR0
005156 012737 040000 177776  MOV      #40000,SRPS     ;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,UIPAR0   ;MAP USER 0 I-SPACE TO
005222 012777 077406 174224  MOV      #77406,UIPDR0 ;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,UIPAR7 ;MAP USER 7 I-SPACE TO THE
005242 012777 077406 174222  MOV      #77406,UIPDR7 ;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
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

- RO - 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, 2#17776
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, 2#PS
005430	012706	001000		MOV	#SSTACK, SP
005434	012737	140000	177776	MOV	#140000, 2#PS
005442	012706	000400		MOV	#USTACK, SP
005446	005037	177776		CLR	2#PS
005452	012767	001770	174324	MOV	#STATAB, STAPNT
005460	017700	174320		STAT20: MOV	2STAPNT, R0
005464	062700	000040		ADD	#40, R0
005470	062767	000002	174306	ADD	#2, STAPNT
005476	017737	174302	177776	MOV	2STAPNT, 2#PS
005504	062767	000002	174272	ADD	#2, STAPNT
005512	012767	000010	174266	MOV	#8, PAGES
005520	012770	007600	000016	MOV	#7600, 216(R0)
005526	016705	173702		MOV	SRO, 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, 2(R0)
005554	022767	000001	174224	CMP	#1, PAGES
005562	001005			BNE	BLK20
005564	012770	007600	177776	MOV	#7600, 2-2(R0)
005572	042705	020000		BIC	#20000, R5
005576	012777	000400	173630	BLK20: MOV	#400, 2SRO
005604	021311			CMP	2R3, 2R1
005606	001401			BEQ	.+4
005610	000000			HALT	
005612	021412			CMP	2R4, 2R2
005614	001401			BEQ	.+4
005616	000000			HALT	
005620	005015			CLR	2R5
005622	005370	000000		DEC	2(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 SRO
: 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 SRO)
: IF NOT, BRANCH
: YES, SET UP PAGE 6 FOR REFERENCES TO SRO
: CHANGE R5 TO POINT TO SRO 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
  
```





# 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	#22,2#SR	;DISPLAY TEST NUMBER
006222	005037	177776		CLR	2#PS	;INITIALIZE PROCESSOR STATUS
006226	012706	001400		MOV	#KSTACK,SP	;INITIALIZE KERNEL STACK POINTER
006232	004767	010024		JSR	%7,SETUP	;INITIALIZE SRO,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 PLJS ONE

006250	004767	010020		JSR	%7,CLRALL	;CLEAR ALL KT11-C REGISTERS
006254	012777	077406	173372	MOV	#77406,2KIPDR0	;MAP KERNEL 0 TO BANK 0, 4K, RW
006262	012777	077406	173366	MOV	#77406,2KIPDR1	;MAKE KERNEL 1 RW, 4K
006270	012777	077406	173374	MOV	#77406,2KIPDR7	;MAP KERNEL 7 RW, 4K, EXTERNAL BANK
006276	012777	007600	173426	MOV	#7600,2KIPAR7	

;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	#7777,2KIPAR1	;SET PAR TO -1
006312	012737	030000	177776	MOV	#30000,2#PS	;SET UP LOCATION TO BE REFERENCED
006320	012777	000400	173106	MOV	#400,2SRO	;TURN ON DESTINATION ONLY PAGING
006326	022737	030000	020076	ADR22A: CMP	#30000,2#20076	;IS PA 777776 BEING REFERENCED?
006334	001012			BNE	ERR22A	;BRANCH ON FAILURE
006336	005037	020076		ADR22A: CLR	2#20076	;CLEAR PA 777776 THRU KERNEL PAGE 1
006342	005077	173066		CLR	2SRO	;TURN OFF KT11-C
006346	032737	177740	177776	BIT	#177740,2#PS	;CHECK TO SEE IF CORRECT LOCATION

```

006354 001401      BEQ      .+4      ; WAS REFERENCED
006356 104006      HLT
006360 000405      BR       CNT22B   ; RELOCATION FAILED IN THE CLRB AT AD22A
006362 005077 173046  ERR22A: CLR     @SR0   ; GO TO NEXT CHECK
006366 104006      HLT           ; TURN OFF KT11-C
                                ; RELOCATION FAILED IN THE COMPARE
                                ; AT LOCATION ADR22A
006370 005037 177776  CLR     @PS      ; REINITIALIZE PROCESSOR STATUS

                                ; CHECK VIRTUAL ADDRESS OF -1 ADDED TO PAR OF 0 (VALUES FOR BIT
                                ; POSITIONS RELEVANT TO THE ADDERS ONLY. VA 37712, PAR 0, PA 17712)
006374 005077 173316  CNT22B: CLR     @KIPAR1 ; SET PAR TO 0
006400 012737 125252 017712  MOV     @125252,@DESTAD ; LOAD PHYSICAL LOCATION TO BE REFERENCED
006406 012777 000400 173020  MOV     @400,@SR0      ; TURN ON DESTINATION ONLY PAGING
006414 022737 125252 037712  ADR22B: CMP     @125252,@37712 ; RELOCATE THRU KERNEL PAGE 1
006422 001011      BNE     ERR22B    ; BRANCH ON FAILURE
006424 005037 037712  AD22B: CLR     @37712   ; CLEAR THRU KERNEL PAGE 1
006430 005077 173000  CLR     @SR0      ; TURN OFF KT11-C
006434 005737 017712  TST     @17712   ; CHECK TO SEE IF CORRECT LOCATION
006440 001401      BEQ      .+4      ; WAS CLEARED
006442 104006      HLT
006444 000403      BR       CNT22C   ; RELOCATION FAILED IN CLR AT AD22B
006446 005077 172762  ERR22B: CLR     @SR0   ; GO TO NEXT CHECK
006452 104006      HLT           ; TURN OFF KT11-C
                                ; RELOCATION FAILED IN THE COMPARE
                                ; AT LOCATION ADR22B

                                ; CHECK VIRTUAL ADDRESS OF 1 (BIT 6) ADDED TO PAR OF -1
                                ; RESULTING PHYSICAL ADDRESS SHOULD BE ZERO
                                ; NOTE THAT THIS IS A CHECK OF ADDRESS WRAP AROUND
006454 012777 007777 173234  CNT22C: MOV     @7777,@KIPAR1 ; SET UP PAR TO -1
006462 012737 034343 000000  MOV     @34343,@#0    ; SET UP A VALUE IN LOCATION TO
                                ; BE REFERENCED (0)
006470 012777 000400 172736  MOV     @400,@SR0    ; TURN ON DESTINATION ONLY PAGING
006476 022737 034343 020100  ADR22C: CMP     @34343,@#20100 ; EFFECTIVELY ADDS 1 TO PAR ADDRESS
                                ; TO GET PHYSICAL ADDRESS OF 0
006504 001013      BNE     ERR22C    ; BRANCH ON FAILURE
006506 012737 000002 020100  AD22C: MOV     @2,@#20100 ; WRITE SAME LOCATION
006514 005077 172714  CLR     @SR0      ; TURN OFF KT11-C
006520 022737 000002 000000  CMP     @2,@#0      ; CHECK LOCATION WHICH SHOULD HAVE
006526 001401      BEQ      .+4      ; BEEN REFERENCED
006530 104006      HLT
006532 000406      BR       CNT22D   ; RELOCATION FAILED IN MOV AT AD22C
006534 005077 172674  ERR22C: CLR     @SR0   ; GO TO NEXT CHECK
006540 104006      HLT           ; TURN OFF KT11-C
006542 012737 000002 000000  MOV     @2,@#0      ; RELOCATION FAILED IN THE COMPARE
                                ; AT LOCATION ADR22C

                                ; CHECK VIRTUAL ADDRESS OF -1 (BITS 6-12) ADDED TO PAR OF 1
                                ; (PLUS HIGH BITS SET, BUT THEY DON'T ALTER CARRY CONDITION TESTED FOR)
                                ; RESULTING PHYSICAL ADDRESS SHOULD BE ZERO
006550 012777 007601 173140  CNT22D: MOV     @7601,@KIPAR1 ; SET UP PAR TO 1, WITH HIGH BITS SET
006556 012737 043434 000000  MOV     @43434,@#0    ; SET UP A VALUE IN LOCATION TO
                                ; BE REFERENCED (0)
006564 012777 000400 172642  MOV     @400,@SR0    ; TURN ON DESTINATION ONLY PAGING
006572 022737 043434 037700  ADR22D: CMP     @43434,@37700 ; ALL HIGH BITS OF VA ARE 1, ADDED TO
                                ; A ONE IN LOWEST BIT OF PAR TO PROPAGATE
                                ; CARRY - RESULTING PHYSICAL ADDRESS 0
006600 001013      BNE     ERR22D    ; BRANCH ON FAILURE

```

```

006602 012737 000002 037700 A0220: MOV      #2,2#37700      ;WRITE SAME LOCATION
006610 005077 172620          CLR      #SR0        ;TURN OFF KT11-C
006614 022737 000002 000000          CMP      #2,2#0      ;CHECK LOCATION WHICH SHOULD HAVE
006622 001401          BEQ     .+4          ;BEEN REFERENCED
006624 104006          HLT     ;RELOCATION FAILED IN MOV AT A0220
006626 000406          BR     CNT22E        ;GO TO NEXT CHECK
006630 005077 172600          EPR220: CLR     #SR0      ;TURN OFF KT11-C
006634 104006          HLT     ;RELOCATION FAILED IN THE COMPARE
                                           ;AT LOCATION ADR220
006636 012737 000002 000000          MOV      #2,2#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      #7777,2#KIPAR1 ;SET UP PAR TO -1
006652 013746 017600          MOV      #17600,-(SP) ;SAVE CONTENTS OF LOCATION TO BE
                                           ;REFERENCED
006656 012737 076767 017600          MOV      #76767,2#17600 ;LOAD LOCATION TO BE REFERENCED
006664 012777 000400 172542          MOV      #400,2#SR0   ;TURN ON DESTINATION ONLY PAGING
006672 022737 076767 037700 ADR22E: CMP      #76767,2#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          A022E: 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 A022E
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)+,2#17600 ;RESTORE LOCATION REFERENCED

;SHOW THAT SETTING SRC(0) TURNS ON FULL RELOCATION
;SHOW THAT ALL ADDRESS CALCULATIONS ARE RELOCATED
;SHOW THAT INIT CLEARS SRC(0) AND TURNS OFF RELOCATION
006736 104400          TEST23: SCOPE
006740 012737 000023 177570          MOV      #23,2#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 SRC,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,CLRAL     ;INITIALLY CLEAR ALL KT11-C REGISTERS
007006 012777 000001 172700          MOV     #1,2#KIPAR0  ;MAP KERNEL I-SPACE PAGE 0 TO
007014 012777 077406 172632          MOV     #77406,2#KIPOR0 ;BANK 0 OFFSET BY 1 BLOCK
007022 012777 007600 172702          MOV     #7600,2#KIPAR7 ;MAP KERNEL I-SPACE PAGE 7 TO
007030 012777 077406 172634          MOV     #77406,2#KIPOR7 ;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 172260          INC     #SR0        ;TURN ON RELOCATION

```

007054	000000			ADD23:	HALT				: WITH RELOCATION ON, SHOULD FETCH
007056	000000				HALT				: FROM 1 BLOCK ABOVE THIS
007060	000000				HALT				: (ADD23A)
007062	000000				HALT				
007064	000000				HALT				
007066	000000				HALT				
007070	032777	000001	172336		BIT	#1, JSRO			: WHEN KT11-C IS TURNED OFF, NEXT
007076	001401				BEG	.+4			: FETCH SHOULD BE FROM HERE -
007100	104006				HLT				: CHECK BIT 0, SR0
									: KT11-C IS NOT RELOCATING THE FETCH BUT
									: SRC(0) STILL SET
									: AFTER AN INIT
007102	005077	172326			CLR	JSRO			
007106	000432				BR	TEST24			
	007154					. = ADD23 + 10C			
007154	022737	052525	017612	ADD23A:	CMP	#52525, #DESTAD-100			: WHEN KT11-C IS TURNED ON, NEXT
									: INSTRUCTION EXECUTED SHOULD
									: BE HERE - CK RELOCATION OF SOURCE
									: AND DESTINATION CALCULATIONS
									: FULL RELOCATION FAILED IN A SOURCE OR
									: DESTINATION ADDRESS CALCULATION
									: IN THE INSTRUCTION AT ADD23A
									: ISSUE INIT TO TURN OFF KT11-C
									: INIT DIDN'T TURN OFF FULL RELOCATION
									: PROBLEM SHOULD BE FIXED BEFORE CONTINUING
									: SHOW THAT A DATO OF 0 TO SRC(0) WILL CLEAR SRC(0) AND
									: TURN OFF RELOCATION
007174	104400				TEST24:	SCOPE			
007176	012737	000024	177570		MOV	#24, #SR			: DISPLAY TEST NUMBER
007204	005037	177776			CLR	#PS			: INITIALIZE PROCESSOR STATUS
007210	012706	001400			MOV	#KSTACK, SP			: INITIALIZE KERNEL STACK POINTER
007214	004767	007042			JSR	%7, SETUP			: INITIALIZE SRC, SR3
007220	026727	172606	000024		CMP	TESTCT, #24			: IS THIS TEST BEING EXECUTED IN THE
007226	001401				BEG	.+4			: CORRECT SEQUENCE? - BRANCH IF YES
007230	104006				HLT				: TEST EXECUTED OUT OF SEQUENCE - TESTCT
									: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
007232	012767	002000	007440		MOV	#2000, ICOUNT			: RESTORE ITERATION COUNT
007240	162706	00C100			SUB	#100, SP			: ALTER STACK POINTER DUE TO BANK 0 OFFSET
007244	004767	007024			JSR	%7, CLRALL			: INITIALLY CLEAR ALL KT11-C REGISTERS
007250	012777	000001	172436		MOV	#1, #KIPAR0			: MAP KERNEL I-SPACE PAGE 0 TO
007256	012777	077406	172370		MOV	#77406, #KIPDR0			: BANK 0 OFFSET BY 1 BLOCK
007264	012777	007600	172440		MOV	#7600, #KIPAR7			: MAP KERNEL I-SPACE PAGE 7 TO
007272	012777	077406	172372		MOV	#77406, #KIPDR7			: THE EXTERNAL BANK
007300	012777	000001	172126		MOV	#1, JSRO			: TURN ON KT11-C
007306	000000			ADD24:	HALT				: WHEN KT11-C IS TURNED ON, SHOULD
007310	000000				HALT				: FETCH FROM ONE BLOCK ABOVE
007312	000000				HALT				: THIS (ADD24A)
007314	000240				NOP				
007316	000240				NOP				
007320	032777	000001	172106		BIT	#1, JSRO			: AFTER KT11-C IS TURNED OFF, CHECK
007326	001401				BEG	.+4			: SRC(0)
007330	104006				HLT				: KT11-C NOT RELOCATING BUT SRC(0) STILL
									: SET AFTER A BIC #1, JSRO
007332	000433				BR	TEST25			

007406	007406	000001	172020	ADD24A: BIC	.=ADD24+100 #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	:DISPLAY TEST NUMBER
007432	005037	177776		CLR	SR	:INITIALIZE PROCESSOR STATUS
007436	012706	001400		MOV	#KSTACK,SP	:INITIALIZE KERNEL STACK POINTER
007442	004767	006614		JSR	%7,SETJP	:INITIALIZE SR0,SR3
007446	026727	172360	000025	CMP	TESTCT,#25	:IS THIS TEST BEING EXECUTED IN THE :CORRECT SEQUENCE?- BRANCH IF YES
007454	001401			BEQ	+.4	:TEST EXECUTED OUT OF SEQUENCE- TESTCT :CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
007456	104006			HLT		

007460	004767	006610		JSR	%7,CLRALL	:CLEAR ALL KT11-C REGISTERS
007464	012777	077406	172200	MOV	#77406,KIPDR7	:MAP KERNEL PAGE 7 I-SPACE TO
007472	012777	007600	172232	MOV	#7600,KIPDR7	:THE EXTERNAL BANK
007500	012777	077406	172146	MOV	#77406,KIPDR0	:MAP KERNEL 0 TO BANK 0, RW, 4K
007506	012777	007542	172300	MOV	#INT25,KIVEC	:SETUP ABORT RETURN
007514	005077	172276		CLR	KTSTA	
007520	012704	020000		MOV	#20000,R4	:USE R4 TO REFERENCE NR KERNEL :ONE I-SPACE

007524	005277	171704		ADR25: INC	SR0	:TURN ON KT11-C
007530	005724			TST	(R4)+	:REFERENCE NR KERNEL 1 I-SPACE
007532	000000			ADR25A: HALT		:SHOULD HAVE ABORTED ALREADY
007534	005077	171674		CLR	SR0	:TURN OFF KT11-C
007540	000457			BR	DONE25	

007542	017701	171666		INT25: MOV	SR0,R1	:SAVE CONTENTS OF SR0
007546	005277	171662		INC	SR0	:TURN OFF KT11-C
007552	022701	100003		CMP	#100003,R1	:CHECK SAVED CONTENTS OF SR0
007556	001401			BEQ	+.4	
007560	104006			HLT		:SR0 INCORRECT AFTER NR ABORT :(SEE SAVED CONTENTS IN R1)

007562	022777	000024	171650	CMP	#24,SR1	:CHECK SR1
007570	001401			BEQ	+.4	
007572	104006			HLT		:SR1 INCORRECT - SHOULD HAVE :RECORDED AUTOINCREMENT OF R4

007574	022777	007530	171642	CMP	#ADR25,SR2	:CK SR2
007602	001402			BEQ	+.6	
007604	104006			HLT		:SR2 INCORRECT-SHOULD CONTAIN ADDRESS :OF LAST FETCH BEFORE THE ABORT.
007606	000407			BR	+.20	
007610	005077	171630		CLR	SR2	:TRY TO WRITE INTO SR2
007614	022777	007530	171622	CMP	#ADR25,SR2	:SR2 SHOULD BE READ ONLY
007622	001401			BEQ	+.4	







```

;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
010400 104010          KTOFF   ;TURN OFF KT11-C VIA TRAPPING
                                ;TO KERNEL AND EXECUTING A
                                ;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

```

# H03

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 33  
 DCKTAB.P11

010426	012737	140000	177776	MOV	#140000, 2#PS	
010434	005277	170774		INC	2SR0	;SET MODE TO USER
010440	005737	020000		TST	2#20000	;TURN ON KT11-C
						;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
010444	104010			KTOFF		;TURN OFF KT11-C VIA TRAPPING
						;TO KERNEL AND EXECUTING A
						;RESET
010446	000403			BR	E26F+2	
010450	104010			RT26F: KTOFF		;TURN OFF KT11-C VIA TRAPPING
						;TO KERNEL AND EXECUTING A RESET
						;REFERENCE TO USER D-SPACE WITH
						;KERNEL D-SPACE ENABLE SET
						;REFERENCED USER D-SPACE INSTEAD
						;OF USER I-SPACE
010452	104006			HLT		;RESTORE STACK POINTER
010454	022626			E26F: CMP	(R6)+, (R6)+	
						;CHECK D-SPACE REFERENCES WITH SUPERVISOR D-SPACE ENABLE SET
010456	012777	000002	170764	MOV	#2, 2SR3	;ENABLE SUPERVISOR D-SPACE
						;WITH SUPERVISOR D-SPACE ENABLE
						;SET WILL NOT ACTUALLY REFERENCE KERNEL D-SPACE
010464	012777	010514	171322	MOV	#RT26G, 2KTVEC	;SETUP ABORT RETURN IN CASE
010472	012737	000000	177776	MOV	#0, 2#PS	;SET MODE TO KERNEL
010500	005277	170730		INC	2SR0	;TURN ON KT11-C
010504	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
010510	104010			KTOFF		;TURN OFF KT11-C VIA TRAPPING
						;TO KERNEL AND EXECUTING A
						;RESET
010512	000403			BR	E26G+2	
010514	104010			RT26G: KTOFF		;TURN OFF KT11-C VIA TRAPPING
						;TO KERNEL AND EXECUTING A RESET
						;REFERENCE TO KERNEL D-SPACE WITH
						;SUPERVISOR D-SPACE ENABLE SET
						;REFERENCED KERNEL D-SPACE INSTEAD
						;OF KERNEL I-SPACE
010516	104006			HLT		;RESTORE STACK POINTER
010520	022626			E26G: CMP	(R6)+, (R6)+	
010522	012777	000002	170720	MOV	#2, 2SR3	;ENABLE SUPERVISOR D-SPACE
						;WITH SUPERVISOR D-SPACE ENABLE
						;SET WILL ACTUALLY REFERENCE SUPERVISOR D-SPACE
010530	012777	010562	171256	MOV	#RT26H, 2KTVEC	;SETUP ABORT RETURN
010536	012737	040000	177776	MOV	#40000, 2#PS	;SET MODE TO SUPERVISOR
010544	005277	170664		INC	2SR0	;TURN ON KT11-C
010550	005737	020000		TST	2#20000	;D-SPACE REFERENCE - SHOULD ABORT
010554	104010			KTOFF		;TURN OFF KT11-C VIA TRAPPING TO
						;KERNEL AND EXECUTING A RESET
						;REFERENCE TO SUPERVISOR D-SPACE WITH
						;SUPERVISOR D-SPACE ENABLE SET
010556	104006			HLT		;DID NOT USE SUPERVISOR D-SPACE

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
010564	022626			E26H:	CMP	(R6)+,(R6)+	;TO KERNEL AND EXECUTING A RESET
010566	012777	000002	170654		MOV	#2,2SR3	;RESTORE STACK POINTER
							;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
010626	104006				HLT		;TO KERNEL AND EXECUTING A RESET
							;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
010668	000403			BR	E26J+2		
010670	104010			RT26J:	KTOFF		;TURN OFF KT11-C VIA TRAPPING
010672	104006				HLT		;TO KERNEL AND EXECUTING A RESET
							;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

```

010704 012777 010734 171102      MOV      #RT26K, @KTVEC      ; SETUP ABORT RETURN IN CASE
010712 012737 040000 177776      MOV      #40000, @#PS      ; SET MODE TO SUPERVISOR
010720 005277 170510              INC      @SR0              ; TURN ON KT11-C
010724 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
010730 104010                    KTOFF                       ; TURN OFF KT11-C VIA TRAPPING
                                ; TO KERNEL AND EXECUTING A
                                ; RESET
010732 000403                    BR       E26K+2
010734 104010      RT26K:  KTOFF                       ; TURN OFF KT11-C VIA TRAPPING
                                ; TO KERNEL AND EXECUTING A RESET
010736 104006                    HLT                          ; REFERENCE TO SUPERVISOR D-SPACE WITH
                                ; USER D-SPACE ENABLE SET
                                ; REFERENCED SUPERVISOR D-SPACE INSTEAD
                                ; OF SUPERVISOR I-SPACE
010740 022626                    E26K:  CMP      (R6)+, (R6)+      ; RESTORE STACK POINTER
010742 012777 000001 170500      MOV      #1, @SR3          ; ENABLE USER D-SPACE
                                ; SHOW THAT A REFERENCE TO USER D-SPACE WITH USER D-SPACE ENABLE
                                ; SET WILL ACTUALLY REFERENCE USER D-SPACE
010750 012777 011002 171036      MOV      #RT26L, @KTVEC    ; SETUP ABORT RETURN
010756 012737 140000 177776      MOV      #140000, @#PS     ; SET MODE TO USER
010764 005277 170444              INC      @SR0              ; TURN ON KT11-C
010770 005737 020000              TST     @#20000            ; D-SPACE REFERENCE - SHOULD ABORT
010774 104010                    KTOFF                       ; TURN OFF KT11-C VIA TRAPPING TO
                                ; KERNEL AND EXECUTING A RESET
010776 104006                    HLT                          ; REFERENCE TO USER D-SPACE WITH
                                ; USER D-SPACE ENABLE SET
                                ; DID NOT USE USER D-SPACE
011000 000402                    BR       E26L+2
011002 104010      RT26L:  KTOFF                       ; TURN OFF KT11-C VIA TRAPPING
                                ; TO KERNEL AND EXECUTING A RESET
011004 022626                    E26L:  CMP      (R6)+, (R6)+      ; RESTORE STACK POINTER
011006 005077 170436              CLR      @SR3              ; REINITIALIZE SR3
011012 016777 171000 170774      MOV      @KTSTA, @KTVEC    ; REINITIALIZE TRAP VECTOR TO CAUSE
011020 005077 170772              CLR      @KTSTA            ; HALT ON FALSE TRAP OR ABORT

                                ; SHOW THAT A DATO OF 0 TO BIT 8, SR0 THRU KERNEL PAGE 7 D-SPACE
                                ; WILL TURN OFF DESTINATION ONLY RELOCATION
011024 104400      TEST27: SCOPE
011026 012737 000027 177570      MOV      #27, @#SR        ; DISPLAY TEST NUMBER
011034 005037 177776              CLR      @#PS              ; INITIALIZE PROCESSOR STATUS
011040 012706 001400              MOV      #KSTACK, SP      ; INITIALIZE KERNEL STACK POINTER
011044 004767 005212              JSR     %7, SETUP          ; INITIALIZE SR0, SR3
011050 026727 170756 000027      CMP      TESTCT, #27      ; IS THIS TEST BEING EXECUTED IN THE
011056 001401                      BEQ     .+4                 ; CORRECT SEQUENCE?- BRANCH IF YES
011060 104006                    HLT                          ; TEST EXECUTED OUT OF SEQUENCE- TESTCT
                                ; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
011062 012767 000100 005610      MOV      #100, ICOUNT      ; INITIALLY CLEAR ALL KT11-C REGISTERS
011070 004767 005200              JSR     %7, CLRALL         ; MAP KERNEL 0 D-SPACE TO BANK 0.
011074 012777 000001 170632      MOV      #1, @KDOPRO      ; RW, OFFSET BY 1 BLOCK
011102 012777 077406 170564      MOV      #77406, @KDOPRO

```

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
011156	000000			HALT		; KT11-C IS STILL ON
011160	000005			RESET		; MAKE SURE THAT KT11-C IS OFF
011162	005077	170262		CLR	@SR3	; REINITIALIZE SR3
011166	005037	177776		CLR	@#PS	; REINITIALIZE PROCESSOR STATUS
						; 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
011266	000777			BR	.	; KT11-C IS STILL ON
011270	000005			RESET		; MAKE SURE KT11-C IS OFF
011272	005077	170152		CLR	@SR3	; REINITIALIZE SR3
011276	005037	177776		CLR	@#PS	; REINITIALIZE PROCESSOR STATUS
						; 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
011376	000777			BR	.	; KT11-C IS STILL ON
011400	000005			RESET		; MAKE SURE KT11-C IS OFF
011402	005077	170042		CLR	@SR3	; REINITIALIZE SR3
011406	005037	177776		CLR	@#PS	; REINITIALIZE PROCESSOR STATUS

```

: 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
    
```

				R0	- 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			TEST30: SCOPE	
011414	012737	000030	177570	MOV	#30, @#SR ; DISPLAY TEST NUMBER
011422	005037	177776		CLR	@#PS ; INITIALIZE PROCESSOR STATUS
011426	012706	001400		MOV	#KSTACK, SP ; INITIALIZE KERNEL STACK POINTER
011432	004767	004624		JSR	%7, SETUP ; INITIALIZE SR0 SR3
011436	026727	170370	000030	CMP	TESTCT, #30 ; IS THIS TEST BEING EXECUTED IN THE
011444	001401			BEQ	.+4 ; CORRECT SEQUENCE?- BRANCH IF YES
011446	104006			HLT	; TEST EXECUTED OUT OF SEQUENCE- TESTCT
					; CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
011450	004767	004620		JSR	%7, CLRALL ; INITIALIZE KT11-C - CLEAR ALL REGISTERS
011454	004767	004750		JSR	%7, RWDSP ; MAKE ALL D SPACE PAGES RW, BANK 0, 4K
011460	013767	017700	170322	MOV	@#17700, SAVEA ; SAVE CONTENTS OF LOCATIONS TO BE USED
011466	013767	017776	170316	MOV	@#17776, SAVEB ; AS START AND END OF PHYSICAL BLOCK
011474	012737	123456	017700	MOV	#123456, @#17700 ; SET UP LOCATIONS TO BE REFERENCED
011502	012737	134567	017776	MOV	#134567, @#17776
011510	012777	000007	167732	MOV	#7, @SR3 ; ENABLE ALL D-SPACES
011516	012703	001414		MOV	#K123, R3 ; LOAD R3 AND R4 WITH ADDRESSES OF
011522	012704	001416		MOV	#K134, R4 ; 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
011526	012767	000100	005144	MOV	#100, ICOUNT ; CHANGE ITERATION COUNT
011534	012737	040000	177776	MOV	#40000, @#PS ; CHANGE TO SUPERVISOR
011542	012706	001000		MOV	#SSTACK, SP ; SET UP SUPERVISOR STACK POINTER
011546	012737	140000	177776	MOV	#140000, @#PS ; CHANGE TO USER
011554	012706	000400		MOV	#USTACK, SP ; SET UP USER STACK POINTER
011560	005037	177776		CLR	@#PS ; RETURN TO KERNEL
011564	012767	001770	170212	MOV	#STATAB, STAPNT ; 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		STATEL: MOV	@STAPNT, R0 ; PICK UP ADDRESS OF START OF
011576	062700	000060		ADD	#60, R0 ; ADDRESS TABLE FOR NEW MODE
011602	062767	000002	170174	ADD	#2, STAPNT ; MOVE POINTER TO ADDRESS VALUE TO
					; LOAD INTO PROCESSOR STATUS
011610	017737	170170	177776	MOV	@STAPNT, @#PS ; SET PROCESSOR STATUS TO NEW MODE
011616	062767	000002	170160	ADD	#2, STAPNT ; MOVE POINTER TO ADDRESS VALUE
					; FOR NEXT MODE

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 38  
DCKTAB.P11

```

011624 012767 000010 170154      MOV      #8, PAGES      ;SETUP COUNTER OF PAGES LEFT TO TEST
011632 012770 007600 000016      MOV      #7600, @16(RO) ;SET UP RELOCATED REFERENCE TO SRO
011640 016705 167570                MOV      SRO, 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, @2(RO)  ;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 SRO)
011674 001005                BNE     BLKL          ;IF NOT, BRANCH
011676 012770 007600 177776      MOV      #7600, @-2(RO) ;YES, SET UP PAGE 6 FOR REFERENCES TO SRO
011704 042705 020000                BIC     #20000, R5     ;CHANGE R5 TO POINT TO SRO THRU PAGE 6
011710 012777 000400 167516  BLKL: MOV      #400, @SRO  ;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     @2(RO)       ;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     @2(RO)       ;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     (RO)+         ;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     @SRO
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                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



012304	005723			CNT318: TST	(R3)+				:CORRESPONDING TO THE PAGE THAT
012306	077411			SOB	R4, LOP31F				: WAS WRITTEN IS POINTED TO BY R1
012310	062703	000040		ADD	#40, R3				: MOVE POINTER
012314	020327	001752							: CHECK ALL PDR'S IN THIS GROUP
012320	002760			CMP	R3, #ADREND				: IF DONE WITH THIS GROUP, MOVE POINTER
				BLT	LOP31E				: TO NEXT GROUP OF ADDRESSES
									: BRANCH IF SOME PDR'S NOT YET CHECKED
012322	012771	077406	000000						
012330	032771	000100	000000	:SHOW THAT WRITING THE PDR WILL CLEAR THE W BIT					
012336	001401			MOV	#77406, 2(R1)				: CLEAR W BIT VIA DATA TO PDR
012340	104006			BIT	#100, 2(R1)				: CHECK W BIT
				BEQ	.+4				
				HLT					: W BIT DIDN'T CLEAR WHEN PDR
									: WAS WRITTEN (R1 POINTS TO
									: THE ADDRESS OF THE PDR)
012342	005721			CNT31C: TST	(R1)+				: MOVE POINTER
012344	062702	020000		ADD	#2000C, R2				: CHANGE VA TO REFERENCE NEXT PAGE
012350	103330			BCC	LOP31D				: BRANCH TO TEST NEXT PAGE IN THIS MODE
012352	020027	002002		CMP	R0, #STAEND				: IF DONE IN THIS MODE, CHECK FOR ALL MODES DONE
012356	002720			BLT	LOP31C				: LOOP UNTIL ALL MODES HAVE BEEN TESTED
012360	005077	167064		CLR	2SR3				: REINITIALIZE SR3
012364	005077	167044		CLR	2SR0				: REINITIALIZE SR0
				:NOW TEST ALL D-SPACE PDR'S					
				:ALL D SPACES ARE ENABLED					
012370	104400			TEST32: SCOPE					
012372	012737	000032	177570	MOV	#32, 2SR				: DISPLAY TEST NUMBER
012400	005037	177776		CLR	2PS				: INITIALIZE PROCESSOR STATUS
012404	012706	001400		MOV	#KSTACK, SP				: INITIALIZE KERNEL STACK POINTER
012410	004767	003646		JSR	%7, SETUP				: INITIALIZE SR0, SR3
012414	026727	167412	000032	CMP	TESTCT, #32				: IS THIS TEST BEING EXECUTED IN THE
012422	001401			BEQ	.+4				: CORRECT SEQUENCE? - BRANCH IF YES
012424	104006			HLT					: TEST EXECUTED OUT OF SEQUENCE - TESTCT
									: CONTAINS NUMBER OF PREVIOUS TEST PLUS ONE
012426	012767	000400	004244	MOV	#400, ICOUNT				: LOAD ITERATION COUNT
012434	004767	003664		JSR	%7, RWALL				: MAP ALL PAGES 4K, BANK 0, RW
012440	012777	007600	167304	MOV	#7600, 2KDPAR7				: MAP KERNEL 7 D-SPACE TO EXTERNAL BANK
012446	012777	007600	167176	MOV	#760C, 2SDPAR7				: MAP SUPERVISOR 7 D-SPACE TO EXTERNAL BANK
012454	012777	007600	16707C	MOV	#7600, 2UDPAR7				: MAP USER 7 D-SPACE TO EXTERNAL BANK
012462	012777	000007	16676C	MOV	#7, 2SR3				: ENABLE ALL D-SPACES
012470	012737	040000	177776	MOV	#40000, 2PS				: SET MODE TO SUPERVISOR
012476	012706	00100C		MOV	#SSTACK, R6				: SET UP SUPERVISOR STACK
012502	012737	140000	177776	MOV	#140000, 2PS				: SET MODE TO USER
012510	012706	000400		MOV	#USTACK, R6				: SET UP USER STACK
012514	005037	177776		CLR	2PS				: REINITIALIZE STATUS TO KERNEL MODE
012520	012704	001454		MOV	#ADRTAB, R4				: LOAD R4 WITH THE ADDRESS OF THE ADDRESS TABLE
012524	012705	000020		LOP32A: MOV	#20, R5				: R5 IS A COUNTER OF THE PDR'S LEFT TO CHECK
									: IN THE CURRENT SET
012530	022734	077406		LOP32B: CMP	#77406, 2(R4)+				: CHECK THE CONTENTS OF THE PDR
012534	001401			BEQ	.+4				
012536	104006			HLT					: PDR INCORRECT - W BIT SET OR ANOTHER
									: BIT INCORRECT IN PDR WHOSE ADDRESS
									: IS IN THE LOCATION POINTED TO BY R4
012540	077505			SOB	R5, LOP32B				: TEST ALL PDR'S IN THE SET
012542	062704	000040		ADD	#40, R4				: MOVE POINTER TO FIRST ADDRESS OF NEXT SET

012546	020427	001752		CMP	R4, #ADREND	
012552	003001			BGT	CNT32A	: BRANCH IF DONE
012554	000763			BR	LOP32A	
012556	012700	001770		CNT32A: MOV	#STATAB, 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, 2(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	#AORTAB, 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	E.	#100, 2(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, 2(R1)	: CLEAR W BIT VIA DATO TO PDR
012676	032771	000100	000000	BIT	#100, 2(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		

DCKTA MACY11 2:(732) 09-SEP-76 15:07 PAGE 42  
 DCKTAB.P11

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			BEG	.+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 FW 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			BEG	.+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 ENABLER 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
013066	104400			TEST34: SCOPE		
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			BEG	.+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, @KOPDR0	: MAP KERNEL, SUPERVISOR, AND USER
013214	012777	000006	166352	MOV	#6, @SOPDR0	: 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, @KOPAR0	: KERNEL D SPACE STARTS AT 700
013236	012777	000007	166370	MOV	#7, @SOPAR0	: SUPERVISOR D SPACE STARTS AT 700

013244	012777	000003	166262	MOV	#3, JUDPAR0	: 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, JSR3	: 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, #JUDPDR7	: MAP USER D-SPACE PAGE 7
013334	012777	007600	166210	MOV	#7600, #JUDPAR7	: 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	: TURN OFF KT11-C
013354	012737	013400	000720	MOV	#R1	: SETUP FOR IOT TO SUPERVISOR STACK
013362	012737	040000	000722	MOV	#SRET34, #720	
013370	005277	166040		MOV	#40000, #722	
013374	000004			INC	#SRO	: TURN ON KT11-C
013376	000240			STRP34:	IOT	: IOT TRAP SHOULD USE STACK IN SUPERVISOR D-SPACE
013400	005011				NOP	: SINCE STATUS IN IOT VECTOR IS SET TO SUPERVISOR
013402	012737	013426	000720	SRET34:	CLR	: TURN OFF KT11-C
013410	012737	140000	000722	MOV	#R1	: SETUP FOR IOT TO USER
013416	005277	166012		MOV	#URET34, #720	
013422	000004			MOV	#140000, #722	
013424	000240			INC	#SRO	: TURN ON KT11-C
013426	005011			UTRP34:	IOT	: IOT TRAP SHOULD USE STACK IN USER D-SPACE
013430	022737	013350	001374		NOP	: SINCE STATUS IN IOT VECTOR IS SET TO USER
013436	001401			URET34:	CLR	: TURN OFF KT11-C
013440	104006			CMP	#R1	: CHECK TO SEE IF IOT TRAP TO KERNEL PUSHED
013442	022737	000000	001376	BEQ	#KRET34-2, #1374	: VALUE IN KERNEL D-SPACE
013450	001401			HLT	.+4	: KERNEL STACK CONTENTS WRONG. STACK NOT
013452	104006			CMP	#0, #1376	: IN KERNEL D-SPACE
013454	022737	013376	000774	BEQ	.+4	
013462	001401			HLT		: KERNEL STACK CONTENTS WRONG-TRAP STATUS NOT
013464	104006			CMP	#SRET34-2, #774	: WHERE IT SHOULD HAVE BEEN PUSHED
013466	022737	000000	000776	BEQ	.+4	: OR VALUE WRONG
013474	001401			HLT		: SUPERVISOR STACK CONTENTS WRONG-STACK
013476	104006			CMP	#0, #776	: NOT IN SUPERVISOR D-SPACE
013500	022737	013424	000374	BEQ	.+4	
013506	001401			HLT		: SUPERVISOR STACK WRONG-TRAP STATUS
013510	104006			CMP	#URET34-2, #374	: NOT WHERE IT SHOULD HAVE BEEN PUSHED
013512	022737	040000	000376	BEQ	.+4	: OR VALUE WRONG
013520	001401			HLT		: USER STACK CONTENTS WRONG-STACK
013522	104006			CMP	#40000, #376	: NOT IN USER D-SPACE
				BEQ	.+4	
				HLT		: USER STACK WRONG-TRAP STATUS
						: NOT WHERE IT SHOULD HAVE BEEN
						: PUSHED OR VALUE WRONG
						: REINITIALIZE LOCATIONS CHECKED
013524	005037	000374		CLR	#374	
013530	005037	000376		CLR	#376	
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, 2#SR ; DISPLAY TEST NUMBER
013600 005037 177776 CLR 2#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, 2#KDPAR0 ; OFFSET KERNEL 0 D-SPACE 1 PAGE
013640 012777 007600 166104 MOV #7600, 2#KDPAR7 ; MAP KERNEL 7 TO THE EXTERNAL BANK
013646 012777 007600 165776 MOV #7600, 2#SDPAR7 ; MAP SUPERVISOR 7 TO THE EXTERNAL BANK
013654 012777 007600 165670 MOV #7600, 2#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, 2#PS ; SETUP SUPERVISOR STACK
013674 012706 001000 MOV #SSTACK, SP
013700 012737 140000 177776 MOV #140000, 2#PS ; SETUP USER STACK
013706 012706 000400 MOV #USTACK, SP
013712 005037 177776 CLR 2#PS
013716 012706 001300 MOV #KSTACK-100, SP ; LOWER THE KERNEL STACK POINTER TO
; KEEP IT OUT OF CODE WHEN RELOCATED

013722 012777 000007 165520 MOV #7, 2#SR3 ; SET ALL D-SPACE ENABLES
013730 012737 014004 000134 MOV #OK35A, 2#134 ; IF SUCCESSFUL, WILL PICK UP VECTOR
013736 012737 040000 000136 MOV #40000, 2#136 ; RETURN IN PHYSICAL ADDRESS 134
013744 012737 013774 000034 MOV #NG35A, 2#34 ; IF NO GOOD, WILL PICK UP VECTOR RETURN
013752 012737 040000 000036 MOV #40000, 2#36 ; IN PHYSICAL ADDRESS 34
013760 012737 040000 177776 MOV #40000, 2#PS ; SET MODE TO SUPERVISOR
013766 005277 165442 INC 2#SR0 ; TURN ON KT11-C
013772 104400 TRP35: TRAP ; SHOULD PICK UP RETURN ADDRESS FROM
; KERNEL D-SPACE

013774 022626 NG35A: CMP (SP)+, (SP)+ ; RESTORE STACK POINTER
013776 005011 CLR 2#R1 ; TURN OFF KT11-C
014000 104006 HLT ; TRAP VECTOR WASN'T FROM KERNEL
; D-SPACE

014002 000402 BR EMT35 ; GO TO NEXT SUBTEST
014004 022626 OK35A: CMP (SP)+, (SP)+ ; RESTORE STACK POINTER
014006 005011 CLR 2#R1 ; TURN OFF KT11-C
014010 012737 016552 000034 EMT35: MOV #SCOPEC, 2#34 ; RESTORE TRAP VECTOR CONTENTS
014016 005037 000036 CLR 2#36
014022 012737 000136 000134 MOV #136, 2#134
014030 005037 000136 CLR 2#136
014034 012737 014124 000130 MOV #OK35B, 2#130 ; SETUP SUCCESS RETURN
014042 012737 014100 000030 MOV #NG35B, 2#30 ; SETUP FAILURE RETURN
014050 012737 140000 000132 MOV #140000, 2#132
014056 012737 140000 000032 MOV #140000, 2#32
014064 012737 140000 177776 MOV #140000, 2#PS ; SET MODE TO USER
014072 005277 165336 INC 2#SR0 ; TURN ON KT11-C
    
```

014076 104000

EMT

; SHOULD PICK UP RETURN ADDRESS FROM  
; KERNEL D-SPACE  
; RESTORE STACK POINTER  
; TURN OFF KT11-C  
; RESTORE EMT SERVICE POINTER

014100 022626  
014102 005011  
014104 012737 017324 000030  
014112 012737 000340 000032

NG35B: CMP (SP)+, (SP)+  
CLR @R1  
MOV #EMTSRV, @#30  
MOV #340, @#32

; EMT VECTOR WASN'T TAKEN FROM KERNEL  
; D-SPACE  
; RESTORE STACK POINTER  
; TURN OFF KT11-C  
; RESTORE EMT SERVICE POINTER

014120 104006  
014122 000410  
014124 022626  
014126 005011

OK35B: CMP (SP)+, (SP)+  
CLR @R1  
MOV #EMTSRV, @#30  
MOV #340, @#32

014130 012737 017324 000030  
014136 012737 000340 000032  
014144 012737 000132 000130

IOT35: MOV #132, @#130  
CLR @#132  
MOV #OK35C, @#120  
MOV #NG35C, @#20

; SETUP IOT SUCCESS RETURN  
; SETUP IOT FAILURE RETURN

014152 005037 000132  
014156 012737 014224 000120  
014164 012737 014214 000020  
014172 005037 000122  
014176 005037 000022

CLR @#122  
CLR @#22  
CLR @#PS  
INC @SRO  
IOT

; SET MODE TO KERNEL  
; TURN ON KT11-C  
; SHOULD PICK UP RETURN ADDRESS FROM  
; KERNEL D - SPACE  
; RESTORE STACK POINTER  
; TURN OFF KT11-C  
; IOT VECTOR WASN'T TAKEN FROM KERNEL  
; D-SPACE  
; RESTORE STACK POINTER  
; TURN OFF KT11-C  
; RESTORE VECTOR LOCATIONS

014202 005037 177776  
014206 005277 165222  
014212 000004

014214 022626  
014216 005011  
014220 104006  
014222 000402  
014224 022626

NG35C: CMP (SP)+, (SP)+  
CLR @R1  
HLT  
BR INT35

; RESTORE STACK POINTER  
; TURN OFF KT11-C  
; RESTORE VECTOR LOCATIONS

014226 005011  
014230 012737 000022 000020  
014236 012737 000122 000120

OK35C: CMP (SP)+, (SP)+  
CLR @R1  
MOV #22, @#20  
MOV #122, @#120

; SETUP TTY SUCCESS RETURN  
; SETUP TTY FAILURE RETURN

014244 012737 014346 000164  
014252 012737 014332 000064  
014260 012737 140000 000166  
014266 012737 140000 000066

INT35: MOV #OK35D, @#164  
MOV #NG35D, @#64  
MOV #140000, @#166  
MOV #140000, @#66  
MOV #40000, @#PS

; SET MODE TO SUPERVISOR  
; TURN ON KT11-C  
; SET TTY INTERRUPT ENABLE-SHOULD  
; INTERRUPT IMMEDIATELY

014274 012737 040000 177776  
014302 005277 165126  
014306 012777 000100 165106

INC @SRO  
MOV #100, @TCSR  
NOP  
NOP  
CLR @R1  
CLR @TCSR

; TURN OFF KT11-C  
; CLEAR TTY IE  
; TTY FAILED TO INTERRUPT

014314 000240  
014316 000240  
014320 005011  
014322 005077 165074

BR ODDAD  
NG35D: CMP (SP)+, (SP)+  
CLR @R1  
CLR @TCSR

; RESTORE STACK POINTER  
; TURN OFF KT11-C  
; CLEAR TTY IE  
; TTY INTERRUPT WASN'T TAKEN FROM  
; KERNEL D-SPACE  
; RESTORE STACK POINTER  
; TURN OFF KT11-C  
; CLEAR TTY IE

014326 104006  
014330 000412  
014332 022626  
014334 005011  
014336 005077 165060

BR ODDAD  
OK35D: CMP (SP)+, (SP)+  
CLR @R1  
CLR @TCSR

; RESTORE TTY VECTOR RETURN TO CAUSE  
; A HALT ON A FALSE INTERRUPT

014342 104006  
014344 000404  
014346 022626  
014350 005011  
014352 005077 165044

MOV #62, @#60  
CLR @#62  
MOV #162, @#160  
CLR @#162  
CLR @#PS

; RESTORE STATUS TO KERNEL

014356 012737 000062 000060  
014364 005037 000062  
014370 012737 000162 000160  
014376 005037 000162  
014402 005037 177776

ODDAD: MOV #162, @#160  
CLR @#162  
CLR @#PS



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
						;SHOULD CAUSE A SECOND NON-RESIDENT ABORT
014712	005077	164516		ADR36A: CLR	SR0	;TURN OFF KT11-C
014716	104006			HLT		;2ND REFERENCE TO KERNEL 1 D-SPACE
014720	000500			BR	DONE36	;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
						;CHECK STACK
015110	022716	015034		CMP	#ADR36C,(SP)	
015114	001401			BEQ	.+4	
015116	104006			HLT		;PC ON STACK INCORRECT
015120	022626			CMP	(R6)+,(R6)+	;RESTORE STACK POINTER
015122	005077	164306		DONE36: 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





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				; SRO 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 SRO
015516	104006			HLT					; SRO 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" SRO, SR1, AND SR2 AFTER A
									; ACCESS VIOLATION ABORT UNTIL THE CORRESPONDING ABORT BIT IS CLEARED IN SRO.
									; WHEN THEY RESUME TRACKING. A ACCESS VIOLATION ERROR SHOULD STILL ABORT
									; TO 250 EVEN WHEN BIT 13 (SRO) 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 SRO, 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

015674	012777	000007	163546		MOV	#7, @SR3			:4K, BANK 0
015702	012777	015736	164104		MOV	#INT40, @KTVEC			:ENABLE ALL D-SPACES
015710	005077	164102			CLR	@KTSTA			:SETUP ABORT RETURN VECTOR
015714	005277	163514			INC	@SR0			
015720	013737	037776	037776	ADR40:	MOV	@#37776, @#37776			:TURN ON KT11-C
									:REFERENCE KERNEL 1 D-SPACE
									:SHOULD CAUSE ACCESS VIOLATION ABORT
015726	005077	163502			CLR	@SR0			:TURN OFF KT11-C
015732	104006				HLT				:REFERENCE TO KERNEL 1 D-SPACE
015734	000526				BR	DONE40			:DIDN'T CAUSE ACCESS VIOLATION ABORT
015736	042777	000001	163470	INT40:	BIC	#1, @SR0			:TURN OFF KT11-C
015744	022777	020022	163462		CMP	#20022, @SR0			:CHECK SR0
015752	001401				BEQ	.+4			
015754	104006				HLT				:SR0 INCORRECT AFTER ACCESS VIOLATION ABORT
015756	012777	016012	164030		MOV	#INT40A, @KTVEC			:SETUP NEW RETURN VECTOR
015764	022626				CMP	(R6)+, (R6)+			:RESTORE STACK POINTER
015766	012702	037776			MOV	#37776, R2			:SETUP R2 TO REFERENCE KERNEL 1
015772	052777	000001	163434		BIS	#1, @SR0			:TURN ON KT11-C
016000	012242				MOV	(R2)+, -(R2)			:REFERENCE KERNEL 1 D-SPACE
									:SHOULD CAUSE A SECOND ACCESS VIOLATION ABORT
016002	005077	163426		ADR40A:	CLR	@SR0			:TURN OFF KT11-C
016006	104006				HLT				:2ND REFERENCE TO KERNEL 1 D-SPACE
016010	000500				BR	DONE40			:DIDN'T ABORT - PREVIOUS ERROR FLAG
									:NOT YET CLEARED BUT ABORT SHOULD
									:STILL HAVE OCCURRED
016012	042777	000001	163414	INT40A:	BIC	#1, @SR0			:TURN OFF KT11-C
016020	022777	020022	163406		CMP	#20022, @SR0			:CHECK SR0
016026	001401				BEQ	.+4			
016030	104006				HLT				:SR0 INCORRECT AFTER 2ND ACCESS VIOLATION ABORT
016032	022777	013427	163400		CMP	#13427, @SR1			:CHECK SR1 - SHOULD CONTAIN VALUE
016040	001401				BEQ	.+4			:FROM 1ST ABORT
016042	104006				HLT				:SR1 DOESN'T CONTAIN VALUE FROM 1ST ABORT
016044	022777	015720	163372		CMP	#ADR40, @SR2			:CHECK SR2
016052	001401				BEQ	.+4			
016054	104006				HLT				:SR2 DOESN'T CONTAIN VALUE FROM 1ST ABORT
016056	021627	016002			CMP	(R6), #ADR40A			:CHECK ADDRESS PUSHED ON STACK
016062	001401				BEQ	.+4			
016064	104006				HLT				:INCORRECT ADDRESS ON STACK - SHOULD BE
									:PC AT TIME SECOND ABORT OCCURRED
016066	022626				CMP	(R6)+, (R6)+			:RESTORE STACK POINTER
016070	012777	016134	163716		MOV	#INT40B, @KTVEC			:CHANGE RETURN ADDRESS
016076	005077	163332			CLR	@SR0			:CLEAR ACCESS VIOLATION ERROR BIT-SHOULD
									:UNLOCK ERROR TRACKING
016102	105777	163330			TSTB	@SR0H			:CHECK TO SEE THAT ERROR BIT CLEARED
016106	001401				BEQ	.+4			
016110	104006				HLT				:SR0 HIGH BYTE INCORRECT
016112	012702	037776			MOV	#37776, R2			:SETUP R2 TO REFERENCE KERNEL 1
016116	005277	163312			INC	@SR0			:TURN ON KT11-C
016122	012242			ADR40B:	MOV	(R2)+, -(R2)			:3RD ABORT REFERENCE, ERROR BIT WAS CLEARED
016124	005077	163304		ADR40C:	CLR	@SR0			:TURN OFF KT11-C
016130	104006				HLT				:3RD REFERENCE TO KERNEL 1 D-SPACE
016132	000427				BR	DONE40			:DIDN'T CAUSE ACCESS VIOLATION ABORT
016134	042777	000001	163272	INT40B:	BIC	#1, @SR0			:TURN OFF KT11-C
016142	022777	020022	163264		CMP	#20022, @SR0			:CHECK SR0
016150	001401				BEQ	.+4			
016152	104006				HLT				:SR0 INCORRECT AFTER ACCESS VIOLATION ABORT

# M04

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 51  
 DCKTAB.P11

```

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
                                ;CLEAR ERROR BIT
016216 005077 163574      CLR      @KTSTA
                                ;CHANGE TRAP RETURN TO CAUSE A HALT
016222 016777 163570 163564      MOV      KTSTA,@KTVEC
                                ;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      MOV      #96,R1
                                ;COUNT OF REGISTERS TO BE CLEARED
016312 005070 001454      CLRLP: CLR      @ADR1AB(R0)
                                ;CLEAR REGISTERS THRU ADDRESS TABLE
016316 005720                      TST      (R0)+
                                ;MOVE POINTER
016320 077104                      SOB      R1,CLRLP
                                ;LOOP TILL DONE
016322 000207                      RTS      %7

;SUBROUTINE TO MAKE ALL PAGES RW, BANK 0, 4K, UP
016324 005077 163104      RWALL: CLR      @SR0
016330 012701 001454      MOV      #ADR1AB,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
                                ;CHECK FOR DONE
016362 002764                      BLT      RWL1
                                ;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

```

    RTS %7
;SUBROUTINE TO MAKE ALL I SPACE PAGES RW, BANK 0,4K,UP
RWISP: CLR @SR0
        MOV @AORTAB,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 RO,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 @SR0
        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 RO,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 ;WAIT FOR STARTING ADDRESS
        MOV SR,RETRNX ;LOAD STARTING ADDRESS IN RETRNX
        ADD #2,RETRNX ;ADD 2 TO POINT TO INSTRUCTION AFTER
        HALT ;SET SR OPTIONS
        CLR SCOPEF ;KEEP COUNT AT ZERO
        MOV #XLOOP,RETURN ;LOAD SCOPE LOOP RETURN POINTER
        JMP @RETRNX ;JUMP TO TEST
XLOOP: CLR SCOPEF ;KEEP COUNT AT ZERO
        JMP @RETRNX ;JUMP TO TEST
RETRNX: 0
    
```

016552 032737 040000 177570  
 016560 001015  
 016562 032737 004000 177570  
 016570 001025  
 016572 026767 000104 000100  
 016600 100C21  
 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
SCOPEB: CMP (6)+,%6 ;REPOSITION STACK
        MOV (6)+,@#PS ;RESTORE PREVIOUS PROCESSOR STATUS
    
```

```

016622 032737 000400 177570      BIT      #400,2#SR      :LOAD MICROBREAK REGISTER?
016630 001403      BEQ      .+10        :NO-BRANCH
016632 113777 177570 163160      MOVB     2#SR,2#BRK  :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      2%6,RETURN :SAVE SCOPE RETURN POINTER
016660 032737 000400 177570      BIT      #400,2#SR  :LOAD MICROBREAK REGISTER?
016666 001403      BEQ      .+10        :NO-BRANCH
016670 113777 177570 163122      MOVB     2#SR,2#BRK  :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,CRLF    :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      SAVPSP: 0
    
```

```

                                :SUBROUTINE TO PRINT OUT OCTAL NUMBER
                                :PRSHRT DELETES LEADING ZEROS
                                :PROCT PRINTS OUT 6 OCTAL DIGITS
017030 012767 000001 000254      PRSHRT: MOV      #1,PRSHRT :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      PRSHRT :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	@TCR		:WAIT FOR TTY READY
017154	100375			BPL	P.WAIT		
017156	016777	000136	162240	MOV	PTEMP2, @TCBR		:OUTPUT NEXT CHARACTER
017164	105777	162232		TSTB	@TCR		:READY FOR NEXT CHARACTER?
017170	100375			BPL	.-4		:NO - WAIT
017172	012777	000377	162224	MOV	#377, @TCBR		:ISSUE A 'DELETE' CODE
							:TO BUY TIME!
017200	105777	162216		TSTB	@TCR		: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			P.CNT1:	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			PRSFLG:	0		
017314	000000			PRFLG:	0		
017316	000000			PTEMP1:	0		:CONTAINS VALUE TO BE OUTPUT
017320	000000			PTEMP2:	0		:SCRATCH
017322	000000			PTEMP3:	0		:USED TO COUNT CHARACTERS OUTPUT
				: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		

DCKTA MACY11 27(732) 09-SEP-76 15:07 PAGE 55  
 DCKTAB.P11

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,@TDBR  
 TSTB @TCSR  
 BPL .-4  
 MOV @377,@TDBR  
 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,@TDBR ;OUTPUT CARRIAGE RETURN  
 TSTB @TCSR ;WAIT FOR TTY READY  
 BPL .-4  
 MOV @212,@TDBR ;OUTPUT LINEFEED  
 RTS %7 ;RETURN  
 .=17712

017712 125252  
 000001

DESTAD: 125252  
 .END



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				
ADR1AB	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						
LOPSA	002626	594#	621						
LOPSB	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						

PATCH2=	104002	2915#	2918															
PATCH3=	104004	2916#	2919															
PC	=%000007	263#	2912*															
PORND	001760	417#	575															
PDRMSK	002030	439#	563															
PORTAB	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	2873	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														
RET33A	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#												
RWOSP	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	01E540	2787	2789*											
.	= 017714	274*	276*	279*	284*	288*	290*	292*	413	455	464	469	472	475
		497	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	1245	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	2298
		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	2849	2864
		2854	2862	2875	2879	2887	2929	2932	2936	2941	2944	2947*	2836	2841







	2585	2591	2593	2615	2622	2624	2632	2633	2634	2635	2636	2638	2639	2642	2651
	2653	2655	2677	2693	2695	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										
TOP	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										
RTT	2816	2843													
RTS	2727	2737	2749	2760	2772	2856	2884	2937	2946						
RTT	2825														
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												
TRP	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	1697
	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														
.ENOC	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											
.MLIST	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)

