

```

C464:A8      61      tay
C465:68      62      pla
C466:AA      63      tax
C467:68      64      pla
C468:68      65      pla          ;A stored where RTS address would go
C469:68
66      pla
C46A:4C 47 FA 67      jmp newbreak ;Go to normal break routine stuff
C46D:48      68 irq9 pha          ;Save state of machine on stack
C46E:AD F8 07 69      lda mslot ;Save mslot
C471:48      70      pha
C472:A9 C3    71      lda #<irqdone ;Save return irq address
C474:48      72      pha
C475:A9 F4    73      lda #>irqdone ;so when interrupt does RTI
C477:48      74      pha          ;it returns to irqdone
C478:08      75      php          ;Status for user's RTI
C479:4C 74 FC 76      jmp irquser ;Off to the user
C47C:         77 * The user's RTI returns here
C47C:         78 * BEWARE
C47C:         79 * The rom must be reenabled with a LDA romin
C47C:         80 * This way if the LC was write protected, it still is
C47C:         81 * if it was write enabled, it still is
C47C:         82 * if it was being write enabled ( 2 lds), it still will be
C47C:         83 * The restore loop uses an INC because some of the switches are read
C47C:         84 * and some are write. It must be an INC abs,x since both the 6502 and
C47C:         85 * the 65C02 do two reads before the write.
C47C:AD 81 C0   86 irqfix lda romin ;Must be lda!
C47F:68      87      pla          ;Recover machine state
C480:10 07 C489 88      bpl irqdn1 ;Branch if main ZP
C482:8D 09 C0 89      sta setaltzp
C485:AE 01 01 90      ldx $101 ;Get alt stack pointer
C488:9A      91      txs
C489:AD 06     92 irqdn1 ldy #S06 ;Y = index into table of switch addresses
C48B:10 06 C493 93 irqdn2 bpl irqdn3 ;Branch if no change
C48D:BE C1 C4   94      ldx irqtbl,y ;Get soft switch address
C490:FE 00 C0   95      inc $C000,x ;Hit the switch. NO PAGE CROSS!
C493:88      96 irqdn3 dey
C494:30 03 C499 97      bmi irqdn4
C496:0A      98      asl A          ;Get next bit to check
C497:D0 F2 C48B 99      bne irqdn2
C499:0A      100 irqdn4 asl A          ;C = 1 if internal slot space
C49A:0A      101      asl A
C49B:68      102      pla          ;Restore the registers
C49C:A8      103      tay
C49D:8A      104      txs          ;Save the stack pointer
C49E:A9 40     105      lda #$40 ;RTI opcode
C4A0:48      106      pha
C4A1:A9 C0     107      lda #<setslotcxrom
C4A3:48      108      pha
C4A4:A9 06     109      lda #>setslotcxrom
C4A6:69 00     110      adc #0 ;Add 1 if internal slot space
C4A8:48      111      pha
C4A9:A9 8D     112      lda #$8D ;STA setslotcxrom
C4AB:48      113      pha

```

```

C4AC:9A      114      txs          ;Restore stack pointer
C4AD:8A      115      txs          ;Make return address on stack point to code on stack
C4AE:69 03    116      adc #3      ;C = 0 from earlier adc
C4B0:AA      117      tax
C4B1:38      118      sec
C4B2:E9 07    119      sbc #7      ;Point to where code starts
C4B4:9D 00 01 120      sta $100,x
C4B7:E8      121      inx
C4B8:A9 01    122      lda #S1
C4BA:9D 00 01 123      sta $100,x
C4BD:68      124      pla
C4BE:AA      125      tax
C4BF:68      126      pla
C4C0:60      127      rts          ;Go to code on stack

```

```

C4C1:83 88 8B 129 irqtbl dfb >lcbank2,>lcbank1,>lcbank1
C4C4:05 03 55 130      dfb >wrcardram,>rdcardram,>txtpage2
C4C7:         21      INCLUDE DIAGS

```

----- NEXT OBJECT FILE NAME IS REFLIST.1

```

C600:         C600      1      ORG C3000+$300
C600:         C600      2 * These routines test all 64K RAM, as well as the 64K on an Auxiliary
C600:         C600      3 * memory card (when present). With the exception of the INTXCROM switch
C600:         C600      4 * of the IOU, all combinations of the IOU switches are tested and ver-
C600:         C600      5 * ified. All configurations of the MMU switches are also tested.
C600:         C600      6 *
C600:         C600      7 * In the event of any failure, the diagnostic is halted. A message
C600:         C600      8 * is written to screen memory indicating the source of the failure.
C600:         C600      9 * When RAM fails the message is composed of "RAM ZP" (indicating failure
C600:         C600     10 * detected in the first page of RAM) or "RAM" (meaning the other 63.75K),
C600:         C600     11 * followed by a binary representation of the failing bits set to "1".
C600:         C600     12 * For example, "RAM 0 1 1 0 0 0 0 0" indicates that bits 5 and 6 were
C600:         C600     13 * detected as failing. To represent auxiliary memory, a "*" symbol is
C600:         C600     14 * printed preceding the message.
C600:         C600     15 *
C600:         C600     16 * When the MMU or IOU fail, the message is simply "MMU" or "IOU".
C600:         C600     17 *
C600:         C600     18 * The test will run continuously for as long as the Open and Closed
C600:         C600     19 * Apple keys remain depressed (or no keyboard is connected) and no
C600:         C600     20 * failures are encountered. The message "System OK" will appear in
C600:         C600     21 * the middle of the screen when a successful cycle has been run and
C600:         C600     22 * either of the Apple keys are no longer depressed. Another cycle
C600:         C600     23 * may be initiated by pressing both Apple keys again while this message
C600:         C600     24 * is on the screen. To exit diagnostics, Control-Reset must be pressed
C600:         C600     25 * without the Apple keys depressed.
C600:         C600     26 *
C600:         C600     C051 27 TEXT equ $C051
C600:         C600     0009 28 IOUIDX equ $09
C600:         C600     0001 29 MMUIDX equ $01
C600:         C600     0588 30 SCREEN equ $588
C600:         C600     C000 31 IOSPACE equ $C000
C600:         C600     32 *
C600:         C600     C600 33 DIAGS equ *

```