

```

C600:8D 50 C0      34      sta $C050
C603:              35 * Test Zero-Page, then all of memory. Report errors when encountered.
C603:              36 * Accumulator can be anything on entry. All registers used, but no stack.
C603:              37 * Addresses between $C000 and $CFFF are mapped to main $D000 bank.
C603:              38 * Auxillary 64K is also tested if present.

C603:A0 04         40 TSTZPG ldy #54
C605:A2 00         41 ldx #0
C607:18           42 zpl clc ;fill zero page with a pattern
C608:79 B4 C7     43 adc ntbl,y
C608:95 00         44 sta $00,x
C60D:E8           45 inx
C60E:D0 F7 C607   46 bne zpl ;after all bytes filled,
C610:18           47 zp2 clc ; ACC has original value again.
C611:79 B4 C7     48 adc ntbl,y ;so values can be tested
C614:D5 00         49 cmp $00,x
C616:D0 10 C628   50 bne ZPERROR ;branch if memory failed
C618:E8           51 inx
C619:D0 F5 C610   52 bne zp2 ;loop until all 256 bytes tested
C618:6A           53 ror a ;change ACC so location $FF will change
C61C:2C 19 C0     54 bit RDVBLBAR ; use RDVBLBAR for a little randomness...
C61F:10 02 C623   55 bpl zp3
C621:49 A5         56 eor #$A5
C623:88           57 zp3 dey ;use a different pattern now
C624:10 E1 C607   58 bpl zpl ;branch to retest with other value
C626:30 06 C62E   59 bmi TSTMEM ;branch always

C628:55 00         61 ZPERROR eor $00,x ;which bits are bad?
C62A:18           62 clc ;indicate zero page failure
C62B:4C CD C6     63 jmp BADBITS
C62E:          C62E 64 TSTMEM equ *
C62E:86 01         65 stx $01
C630:86 02         66 stx $02
C632:86 03         67 stx $03
C634:A2 04         68 ldx #4 ;do RAM $100-$FFFF five times
C636:86 04         69 stx $04
C638:E6 01         70 mem1 inc $01 ;point to page 1 first
C63A:A8           71 mem2 tay ;save ACC in Y for now
C63B:8D 83 C0     72 sta $C083 ;anticipate not $C000 range...
C63E:8D 83 C0     73 sta $C083
C641:A5 01         74 lda $01
C643:29 F0         75 and #$F0 ;get page address
C645:C9 C0         76 cmp #$C0 ;test for $C0-$CF range
C647:D0 0C C655   77 bne mem3 ;branch if not...
C649:AD 8B C0     78 lda $C08B
C64C:AD 8B C0     79 lda $C08B ;select primary $D000 space
C64F:A5 01         80 lda $01
C651:6F 0F         81 adc #$F ;Plus carry +=10
C653:D0 02 C657   82 bne mem4 ;branch always taken
C655:A5 01         83 mem3 lda $01
C657:85 03         84 mem4 sta $03
C659:98           85 tya ;restore pattern to ACC
C65A:A0 00         86 ldy #500 ;fill this page with the pattern

```

```

C65C:18           87 mem5 clc
C65D:7D B4 C7     88 adc ntbl,x
C660:91 02         89 sta ($02),y
C662:CA           90 dex ;keep x in the range 0-4
C663:10 02 C667   91 bpl mem6
C665:A2 04         92 ldx #4
C667:C8           93 mem6 iny ;all 256 filled yet?
C668:D0 F2 C65C   94 bne mem5 ;branch if not
C66A:E6 01         95 inc 1 ;bump page #
C66C:D0 CC C63A   96 bne mem2 ;loop through $0100 to $FFF0

C66E:E6 01         98 inc $01 ;point to page 1 again
C670:A8           99 mem7 tay ;save ACC in Y for now
C671:AD 83 C0     100 lda $C083 ;anticipate not $C000 range...
C674:AD 83 C0     101 lda $C083
C677:A5 01         102 lda $01 ;get page address
C679:29 F0         103 and #$F0 ;test for $C0-$CF range
C67B:C9 C0         104 cmp #$C0
C67D:D0 09 C688   105 bne mem8 ;branch if not...
C67F:AD 8B C0     106 lda $C08B ;select primary $D000 space
C682:A5 01         107 lda $01
C684:69 0F         108 adc #$F ;Plus carry +=10
C686:D0 02 C68A   109 bne mem9 ;branch always taken
C688:A5 01         110 mem8 lda $01
C68A:85 03         111 mem9 sta $03
C68C:98           112 tya ;restore pattern to ACC
C68D:A0 00         113 ldy #500 ;fill this page with the pattern
C68F:18           114 memA clc
C690:7D B4 C7     115 adc ntbl,x
C693:51 02         116 eor ($02),y
C695:D0 35 C6CC   117 bne MEMERROR ;if any bits are different, give up!!!
C697:B1 02         118 lda ($02),y ;restore correct pattern
C699:CA           119 dex ;keep x in the range 0-4
C69A:10 02 C69E   120 bpl memB
C69C:A2 04         121 ldx #4
C69E:C8           122 memB iny ;all 256 filled yet?
C69F:D0 EE C68F   123 bne memA ;branch if not
C6A1:E6 01         124 inc 1 ;bump page #
C6A3:D0 C8 C670   125 bne mem7 ;loop through $0100 to $FFF0
C6A5:6A           126 ror a ;change ACC for next pass
C6A6:2C 19 C0     127 bit RDVBLBAR ; use RDVBLBAR for a little randomness...
C6A9:10 02 C6AD   128 bpl memC
C6AB:49 A5         129 eor #$A5
C6AD:C6 04         130 memC dec $04
C6AF:10 87 C638   131 bpl mem1 ;have 5 passes been done yet?
;branch if not...

C6B1:AA           133 TAX ;save acc
C6B2:20 8D C9     134 JSR STAX ;set aux memory & write $EE to $C00,$800
C6B5:D0 07 C6BE   135 BNE SWCHTSTI ;=>not 128K
C6B7:DE 00 0C     136 ASL $C00 ;shift test byte
C6B8:0A           137 ASL A
C6BB:CD 00 0C     138 CMP $C00 ;check memory

```