

```

C6BE:D0 76 C736 139 SWCHTST1 BNE SWCHTST ;=>not 128K
C6C0:CD 00 08 140 CMP $800 ;look for shadowing
C6C3:F0 71 C736 141 BEQ SWCHTST ;=>not 128K
C6C5:8A 142 txa
C6C6:8D 09 C0 143 STA SETALTZP ;swap in alt zero page
C6C9:4C 03 C6 144 jmp TSTZPG ; and test it!
C6CC:38 145 MEMERRR sec ;indicate main ram failure
C6CD:AA 146 BADBITS tax ;save bit pattern in x for now
C6CE:AD 13 C0 147 lda RDRAMRD ;determine if primary or auxillary RAM
C6D1:88 148 clv ;with V-PLG
C6D2:10 03 C6D7 149 bpl bbits1 ;branch if primary bank
C6D4:2C B4 C7 150 setv ;branch if primary bank
C6D7:A9 A0 151 bbits1 lda #$A0 ;try to clear video screen
C6D9:A0 06 152 ldy #6
C6DB:9F FE BF 153 clrsts sta IOSPACE-2,y
C6DE:99 06 C0 154 sta IOSPACE+6,y
C6E1:88 155 dey
C6E2:88 156 dey
C6E3:D0 F6 C6D8 157 bne clrsts
C6E5:8D 51 C0 158 sta TEXT
C6E8:8D 54 C0 159 sta TXTPAGE1
C6EB:99 00 04 160 clrs sta $400,y
C6EE:99 00 05 161 sta $500,y
C6F1:99 00 06 162 sta $600,y
C6F4:99 00 07 163 sta $700,y
C6F7:C8 164 iny
C6F8:D0 F1 C6EB 165 bne clrs
C6FA:8A 166 txa
C6FB:F0 27 C724 167 beq BADSWTCH ;test for switch test failure
C6FD:A0 03 168 ldy #3 ;branch if it was a switch
C6FF:B0 02 C703 169 bcs badmain ;branch if ZP ok
C701:A0 05 170 ldy #5
C703:A9 AA 171 badmain lda #$AA ;mark aux report with an asterisks
C705:50 03 C70A 172 bvc badprim
C707:8D B0 05 173 sta screen-8
C70A:B9 EA C7 174 badprim lda rmess,y
C70D:99 B1 05 175 sta screen-7,y
C710:88 176 dey
C711:10 F7 C70A 177 bpl badprim ;message is either "RAM" or "RAM ZP"
C713:A0 10 178 ldy #$10 ;print bits
C715:8A 179 bbits2
C716:4A 180 lsr a
C717:AA 181 tax
C718:A9 58 182 lda #$58 ;bits are printed as ascii 0 or 1
C71A:2A 183 rol a
C71B:99 B6 05 184 sta screen-2,y
C71E:88 185 dey
C71F:88 186 dey
C720:D0 F3 C715 187 bne bbits2
C722:F0 FE C722 188 hangx beq hangx ;hang forever and ever
C724:A0 02 189 BADSWTCH ldy #2
C726:B9 F0 C7 190 bswtchl lda smess,y
C729:90 03 C72E 191 bcc bswtch2 ;branch if MMU in error
C72B:B9 F3 C7 192 lda smess+3,y ;else indicate IOU error

```

```

C72E:99 B8 05 193 bswtch2 sta screen,y
C731:88 194 dey
C732:10 F2 C726 195 bpl bswtchl ;print "MMU" or "IOU"
C734:30 FE C734 196 hangy bmi hangy ;branch forever

C736:A0 01 198 SWCHTST ldy #MMUIDX
C738:A9 7F 199 swtst1 lda #$7F
C73A:6A 200 swtst2 ror a ;set switches of the IOU/MMU to match Accumulo
C73B:BE B9 C7 201 ldx SWTBLO,y
C73E:F0 0F C74F 202 beq swtst4 ;branch if done setting switches
C740:90 03 C745 203 bcc swtst3 ;branch if setting switch to 0-state
C742:BE C9 C7 204 ldx SWTBLO,y ;else get index to set switch to 1
C745:9D FF BF 205 swtst3 sta IOSPACE-1,x ;set switch
C748:C8 206 iny
C749:D0 EF C73A 207 bne swtst2 ;branch always taken...
C74B:AE 30 C0 208 *
C74E:2A 209 click ldx $C030
C74F:88 210 rol a
C750:BE D9 C7 211 swtst4 dey
C753:F0 13 C768 212 ldx RSWTBL,y ;now verify the settings just made
C755:30 F4 C74B 213 beq swtst6 ;branch if done this pass
C757:2A 214 bmi click ;branch if this switch no to be verified.
C758:90 07 C761 215 rol a
C75A:1E 00 C0 216 bcc swtst5
C75D:90 17 C776 217 asl IOSPACE,x
C75F:80 EE C74F 218 bcc swerr
C761:1E 00 C0 219 bcs swtst4 ;branch always
C764:B0 10 C776 220 swtst5 asl IOSPACE,x
C766:90 E7 C74F 221 bcs swerr
C768: 222 bcc swtst4 ;branch always
C768: 223 *
C768:2A 224 swtst6 rol a ;restore original value
C769:C8 225 iny ; and IOU/MMU index
C76A:38 226 sec
C76B:E9 01 227 sbc #1 ;try next pattern
C76D:B0 CB C73A 228 bcs swtst2
C76F:88 229 dey
C770:D0 0B C77D 230 bne BIGLOOP ;was MMU just tested?
C772:A0 09 231 ldy #IOUIDX ;branch if IOU was just tested
C774:D0 C2 C738 232 bne swtst1 ;else, go test IOU.
C776: 233 * ;branch always taken...
C776:A2 00 234 swerr ldx #0 ;indicate switch error
C778:C0 0A 235 cpy #IOUIDX+1 ;set carry if IOU was cause
C77A:4C D7 C6 236 jmp bbits1
C77D:46 80 237 BIGLOOP lsr $80
C77F:D0 B5 C736 238 bne SWCHTST
C781:A9 A0 239 blp2 lda #$A0
C783:A0 00 240 ldy #0
C785:99 00 04 241 blp3 sta $400,y ;clear screen for success message
C788:99 00 05 242 sta $500,y
C78B:99 00 06 243 sta $600,y
C78E:99 00 07 244 sta $700,y
C791:C8 245 iny

```