```
AREA    InterruptStuff, CODE, READONLY

/* Interrupt Request Handler */

/*
    This interrupt request handler will be called by the VIC every
    0.1641417 ms. It contains a count of how many times it has been
    called, storing this in memory at address COUNT. When count has
    reached (1 second) / (0.1641417 milliseconds) = 6 092.29708 ~ 6,092
    then it has been approximately a second. Hanlder will then increment
    seconds counter at memory address SECONDS
*/

irqhan
  SUB LR, LR, #4                        //Adjust the LR to last location
  STMFD SP!,{R0-R1,LR}                  // Preserve registers on the stack

  LDR R1, =COUNT                        // Count of Interrupt calls
  LDR R0, [R1]
  ADD R0, R0, #1                        // count ++
  CMP R0, #6092                         // If count == 6092
  BLT saveCount                         // updateSeconds()

  LDR R0, =SECONDS
  LDR R1, [R0]                          // loadSeconds()
  ADD R1, R1, #1                        // seconds++
  STR R1, [R0]                          // storeSeconds()
  LDR R0, =0                            // count = 0
  LDR R1, =COUNT

saveCount
  STR R0, [R1]

  LDR   R0,=T0
  MOV   R1,#TimerResetTimeR0Interrupt
  STR   R1,[R0,#IR]                   //Remove MR0 interrupt request from timer

  LDR   R0,=VIC
  MOV   R1,#0                         //Stop VIC from making interrupt to CPU
  STR   R1,[R0,#VectAddr]             //Reset VIC

  LDMFD SP!,{R0-R1,PC}^               //Load values off stack, LR loaded into PC
                                      //And also restoring the CPSR (what the ^ does)
```