Clearing a Section of Memory

from 6502 Software Gourmet Guide & Cookbook

When setting up a program for entering data or storing the results of a calculation, it is often desirable to clear the memory locations to be used for storage. This operation is achieved by filling the memory locations with zeros. One way to do this is to store zero in the accumulator and perform a series of STA ADDR instructions in which the ADDR designates each memory location to be cleared. This method is fine if the area to be cleared consists of only two or three memory locations, and the clearing operation is required in only one or two different portions of the program. However, if a lengthy table area must be cleared, such as an input buffer (which may store 72 characters or more for a single line of input), this would be highly impractical. It would use more memory locations than necessary, even for short tables to be cleared by different routines throughout a program.

An alternative subroutine which, when called, will clear as many locations in a table area as defined by the calling program. The routine listed below will fill up to 256 memory locations with zeros. The calling routine must store the lowest address of the table in TOPNT on page zero. Also, the X index register must contain the binary count of the number of locations to be cleared. CLRMEM is the start of the subroutine. The accumulator and the X index register will be equal to zero upon returning.

TOPNT is a successive pair of memory locations set up on page zero. It will be used as storage for a temporary pointer. The low portion of the address stored in TOPNT is stored in the lower addressed byte and the page portion is stored in the next higher byte.

```
CLRMEM  LDA #$00        ;Set up zero value
        TAY             ;Initialize index pointer
CLRM1   STA (TOPNT),Y   ;Clear memory location
        INY             ;Advance index pointer
        DEX             ;Decrement counter
        BNE CLRM1       ;Not zero, continue checking
        RTS             ;Return
```

Mickael Pointier suggests an alternative method that saves one free register and runs a bit faster since you no longer have to increment one of the registers:

```
CLRMEM  LDA #$00        ;Set up zero value
CLRM1   DEY             ;Decrement counter
        STA (TOPNT),Y   ;Clear memory location
        BNE CLRM1       ;Not zero, continue checking
        RTS             ;RETURN
```