

00FE contains a 23 DEC R3 stop instruction in the event a later programming change causes the Output Manager to execute a return. (Don't you just hate runaway programs?)

Each subroutine mentioned above is detailed below. The assembly procedure is not difficult to understand if you remember that you specify the starting address at the beginning of pass 1. This address is incremented for each valid instruction line ( x 2) and every time a label is encountered, the address is recorded along with that label in the symbol table. On the second pass, the object code is created, and arguments are compared with labels in the symbol table. Addresses are inserted and that's it! All assemblers operate in pretty much the same way.

## SUBROUTINES

### PUT INSTRUCTION IN MEMORY

This is a simple routine (the best kind) that loops twice, taking a pair of ASCII codes on each loop, converting them to hex bytes and storing them in the object code memory area @ 0C00-0EFF via R9. The ASCII codes make up the Chip-8 instructions in the text area