Here we have the instruction -- an addressing type which is the Chip-8 Go-To instruction -- beginning at position #7. Position #11 is a space and at position #12 is the first character of the argument "BEGIN" which then fills out the line. Compare the argument "BEGIN" with the label in the original Format Example line #3. You will see they are the same, and in this lies one of the primary reasons for using an assembler. After assembly, when the Go-To instruction in Format Example #2 is encountered in the program, a jump to the instruction with the same label will be performed. Instead of an address, we have specified the flow of the program using an English word, BEGIN. In a long program, the use of labels and arguments greatly simplifies programming. And even BASIC does not give you this capability.

In line #3 of the Format Example, not only is there a label "BEGIN" on the left, but there is a new argument, "VARS" on the right. The rules have all been followed. The instruction is an addressing type (AXXX), and there is a space in position #11 between the instruction and the argument. Look at the sample program at the end of this section. You should be able to find the label VARS in the third column of the source listing. At this position are the seven bytes of data which will be the