

Jump versus call instructions

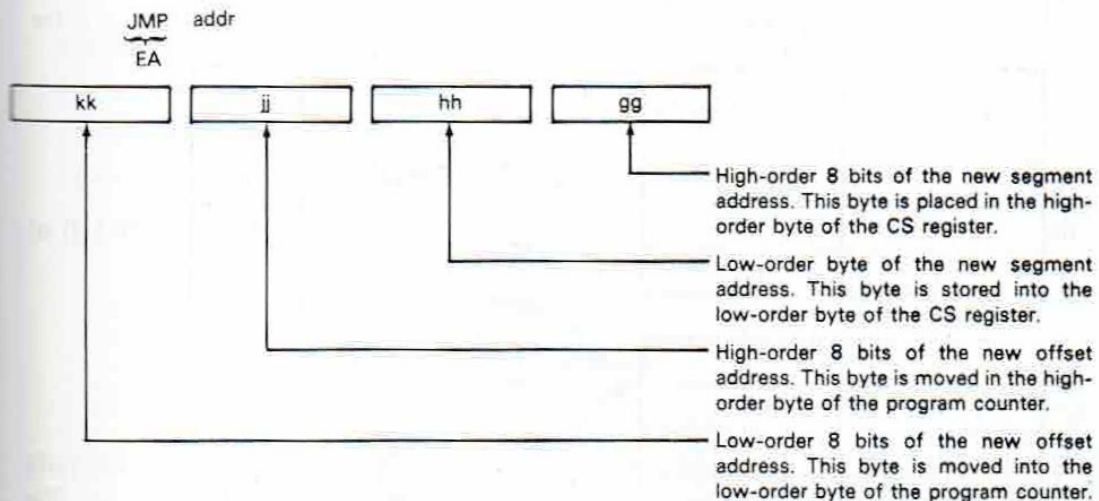
A pure, raw jump instruction just replaces the contents of the PC register, thus effecting the address from which to source the next instruction. (The PC register is used as an offset relative to the CS or "code segment" register's address, so it gets overwritten with new content too. Together, they pinpoint where the next instruction to be executed is.)

JMP addr

Jump to the Instruction Identified in the Operand

Move the contents of the next two program memory bytes into the PC register. Move the contents of the succeeding two program memory bytes (bytes 4 and 5 of the instruction) into the CS register. Continue execution from this point. The previous program counter and Code Segment register contents are lost.

The encoding for this instruction is:



CALL addr**CALL the Subroutine Specified in the Operand (Intersegment)**

Store the contents of the CS and PC registers on the top of the stack, i.e., push the address of the instruction following the CALL onto the top of the stack. Place the contents of the succeeding four memory bytes into the PC and CS registers. Place the bytes in the following manner:

1. Store the second and third bytes of this instruction into the PC register.
2. Store the fourth and fifth bytes of this instruction into the CS register.

The encoding for this instruction is:

