

Instruction in 8085

In 8085 microprocessor, an instruction is a set of binary codes that represents a specific operation to be performed by the CPU. Each instruction has a unique opcode and may also include one or more operands that specify the data to be operated on. The instruction execution process consists of fetching, decoding, and executing the instruction.

The 8085 microprocessor has a total of 246 instructions, which are grouped into the following five categories:

1. **Data transfer instructions**: These instructions are used to transfer data between registers, memory locations, and input/output devices. Examples include MOV, MVI, LXI, and LDA instructions.
2. **Arithmetic instructions**: These instructions are used to perform arithmetic operations such as addition, subtraction, increment, and decrement. Examples include ADD, ADC, SUB, INR, and DCR instructions.
3. **Logical instructions**: These instructions are used to perform logical operations such as AND, OR, XOR, and complement. Examples include ANA, ORA, XRA, and CMA instructions.
4. **Branching instructions**: These instructions are used to change the flow of program execution based on a specified condition. Examples include JMP, JNZ, JC, and CALL instructions.
5. **System instructions**: These instructions are used to perform system-level tasks such as interrupt handling, I/O operations, and stack operations. Examples include EI, DI, IN, OUT, and PUSH instructions.

**Arithmetic instructions**

* Arithmetic instructions are used to perform arithmetic operations such as addition, subtraction, increment, and decrement.
* These operations can be performed on data stored in registers or memory locations.
* The most commonly used arithmetic instructions in the 8085 microprocessor are ADD, ADC, SUB, SBB, INR, and DCR.
* ADD instruction adds two numbers and stores the result in the destination register or memory location.
* ADC instruction adds two numbers and the carry flag and stores the result in the destination register or memory location.
* SUB instruction subtracts two numbers and stores the result in the destination register or memory location.
* SBB instruction subtracts two numbers and the carry flag and stores the result in the destination register or memory location.
* INR instruction increments the value of the specified register or memory location by 1.
* DCR instruction decrements the value of the specified register or memory location by 1.
* Arithmetic instructions are executed by fetching the instruction, decoding the opcode, and performing the specified operation on the operands.
* The carry flag and the zero flag are set based on the result of the operation and can be used to control program flow using branching instructions.

Here are the 5 main uses of arithmetic instructions in computer programming:

1. Data manipulation: Arithmetic instructions are used to manipulate numerical data such as addition, subtraction, multiplication, and division. These instructions are essential for performing mathematical calculations in programs.

1. Control flow: Arithmetic instructions can be used to control program flow by modifying program counters, branching, and looping based on the result of arithmetic operations. This is often done using conditional branching instructions such as JZ (Jump if Zero), JC (Jump if Carry), and JNC (Jump if No Carry).
2. Counters and timers: Arithmetic instructions can be used to implement counters and timers that count the number of times a program loop has been executed or the amount of time that has elapsed.
3. Logical operations: Arithmetic instructions can also be used for logical operations such as bitwise AND, OR, XOR, and NOT. These operations are commonly used in programming for data encoding, cryptography, and error correction.
4. Floating-point arithmetic: Arithmetic instructions can be used to perform floating-point arithmetic, which is necessary for scientific and engineering applications that require high-precision calculations. Specialized arithmetic instructions or libraries are often used for floating-point arithmetic to improve performance and accuracy.