

If you don’t want the execution to proceed to the next instruction, what must be changed?

1) Instruction Register (IR)

2) Memory Address Register (MDR)

3) Memory Data Register (MDR)

4) Program Counter (PC) <-----

Instruction Set Architecture (ISA):

* Used by Machine Language programmers, Assembly programmers, Compiler writers
* Also used by high level language programmers (Java, etc) in performance and debugging
  + Elegant code may not be most efficient, ISA can help establish why
* Contains memory organization, registers, instruction set (opcodes, data types, addressing modes)

LC-3 Instruction Set

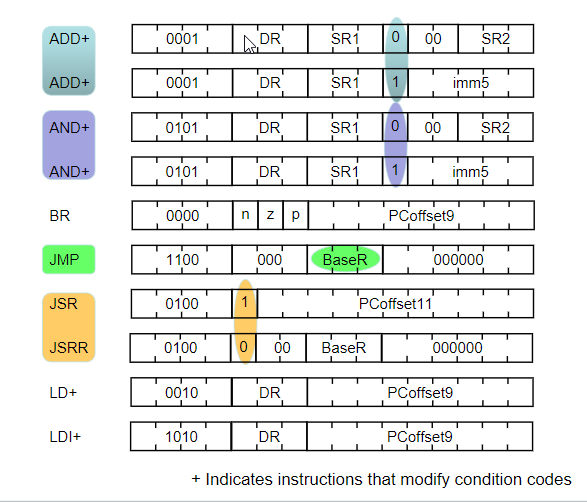
* Instruction
  + Op Code
  + Operands
* Instruction Set
  + Op codes
    - Operate
    - Data Movement
    - Control
  + Data types
  + Addressing Modes

Lot of instructions -> CISC (Complex Instruction Set Computer)

* X86, DEC VX, IBM Z-Series
* Do as much as you can in a single instruction
* Relatively easy to write efficient machine code by hand

Few instructions -> RISC (Reduced Instruction Set Computer)

* ARM, PowerPC, MIPS, LC-3 (what we are looking at in this class)
* Expose as much as you can to the compiler so that I can optimize
* Hard to write efficient machine code by hand



JMP has same Opcode