

Notes: The LC-3 Computer

A. Why?

We'll be writing machine and assembler programs for the LC-3.

B. Outcomes

By the end of the class you should

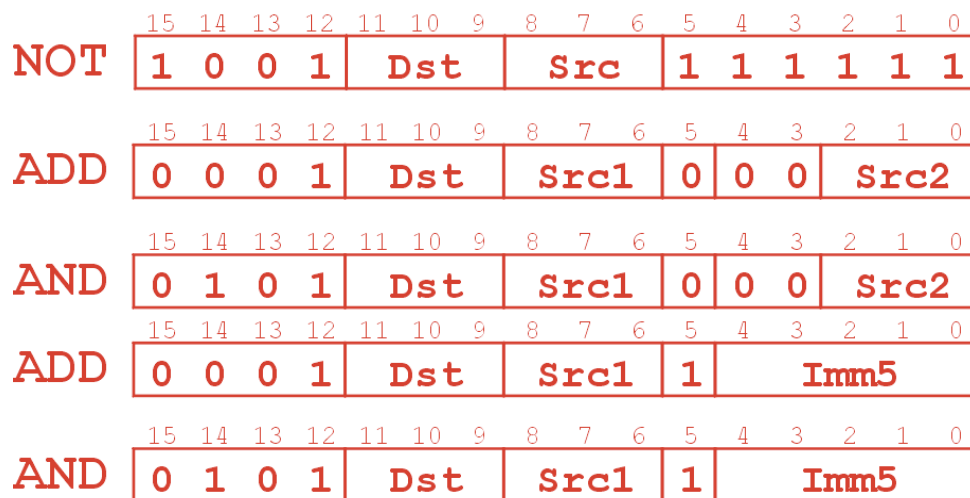
- Know the basic architecture of the LC-3: word size, number of registers, data types supported.
- Know the different addressing modes of the LC-3 [not covered]

C. The LC-3 Computer

- Text uses the Little Computer version 3
- 16-bit addresses (2^{16} memory locations), 16-bit word at each location
- 2's complement integers
- 8 data registers (named R0 – R7; 3 bits to name a register)
 - Temporary storage, accessed in 1 machine cycle
 - Accessing memory generally takes > 1 cycle
- 3 condition code bits for tests (we'll see this later).
- 4-bit opcodes (16 instructions)
 - Calculation/data manipulation: addition, logical and, logical not
 - Data movement: load to register, store to memory (multiple variations for both)
 - Control: branch/jump [go to]; subroutine call and return; call operating system
- Four kinds of **addressing modes** (= ways to specify operands). (Not every instruction has every kind of operand.)
 - Immediate (contained in instruction)
 - Register (number 000, 001, ..., 110, 111)
 - Memory address
 - Base offset: Requires a register number and a binary number: Add contents of register to binary number; result is a memory address.
 - PC offset: Requires a binary number: Add contents of Program Counter and binary number; result is a memory address.
 - Program counter points to next instruction, not current instruction, so offset=0 specifies next instruction, not current instruction.
 - Indirect: Starts like PC offset but uses the value *stored* at the memory address, not the memory address itself. (Pointer!)

D. Calculation/Data Operation Instructions

- Source and destination operands are registers.



- These instructions do not reference memory.
- ADD and AND can use “immediate” mode, where one operand is hard-wired into the instruction.
- Dataflow Diagrams
 - Dataflow diagrams illustrate when and where data moves for an operation.
 - For NOT instruction:
 - Perform bitwise NOT on source, store result in destination.
 - Note: Src and Dst can be the same register.
 - For AND and ADD instructions (normal mode)
 - Destination register gets set to Source 1 register plus Source 2 register.
 - Bit 5 = 0 in the instruction indicates that Source 2 is a register
 - For immediate mode of AND and ADD:
 - Destination register gets set to Source 1 register plus Source 2 data.
 - Bit 5 = 1 in the instruction indicates that Source 2 is the last 5 bits of the instruction.
- Don't have separate instructions for
 - Subtraction
 - Logical OR
 - Set a register to zero.
 - Copy from one register to another.

