Machine Language Programming

Tom Kelliher, CS 220

Chapter 4, Textbook

- Section 4.2: Name and describe the two Hack instruction formats. At the machine language level (binary instructions), how are they distinguished?

  Name and describe the three fields of the C-instruction.

Chapter 4, Slides

- Slide 3: What is the difference between an assembly language program and a machine language program?

  What are the elements of a machine language? (Instruction set, addressing modes, the number of registers, the instruction formats, and word size, among others. Note that these same elements describe a machine's architecture.)

- Slide 5: The Hack architecture has three addressing modes: register, immediate, and direct.

  For comparison purposes, the ARM Cortex 32-bit architecture has 15 general purpose registers and can have up to 32 64-bit floating-point registers. The ARM 64-bit architecture has 31 general purpose registers and up to 32 128-bit floating-point registers.

  The X86 64-bit architecture has 16 general purpose registers and eight 80-bit floating-point registers, in addition to special purpose registers for MMX, SSE, etc. However, the current microarchitecture (implementation/organization) of the X86-64 architecture, Skylake, features 180 physical general purpose integer registers and 168 physical floating-point registers. At the microarchitectural level, a technique known as *register renaming* is used to map the architectural registers to the physical registers at run-time.

- Slide 7: Value is 15 bits. When loaded into the A register, the msb is set to 0. How do we load a negative value?

- Slides 8, 9, and 11: Be prepared to implement any of the given high level language statements in Hack.

  Slide 9: The symbol table is maintained by the Assembler. It maps a symbolic name to the memory location to which the symbolic name has been bound.

- Slides 12 and 13: Remember — labels must be unique.

- Slide 15: Note that the condition `i <= 100` is equivalent to `i - 100 <= 0`.