Week 4 - Machine Language

Computer theory - what is a universal computer?

Theory → Alan Turing → Universal Turing Machine

Practice → John von Neumann → von Neumann Architecture

The program is in memory in a von Neumann machine (a "stored program" computer)

A program is a sequence of instructions which are written in a machine language.

Elements of an instruction:

1) An operation (opcode)
2) Which instruction to run next? (jump)
3) What data is the instruction operating on, and where is the result stored?

A compiler translates a high-level programming language into machine language.

Can write "machine language" by using mnemonics for various binary codes and write assembly language instead.

Translating the assembly language to machine language is very easy and done by a program called an assembler.

The assembler can also keep track of and manage symbols.

↳ allows the programmer to use symbols like "index" rather than Mem[129]

The Machine Language defines the interface between software and hardware.

Most important interface in the whole computer system!

- Usually closely corresponds to the actual hardware architecture.
- Always a cost/performance tradeoff in machine language/ISA design!

↳ Instruction Set Architecture

Possible operations:

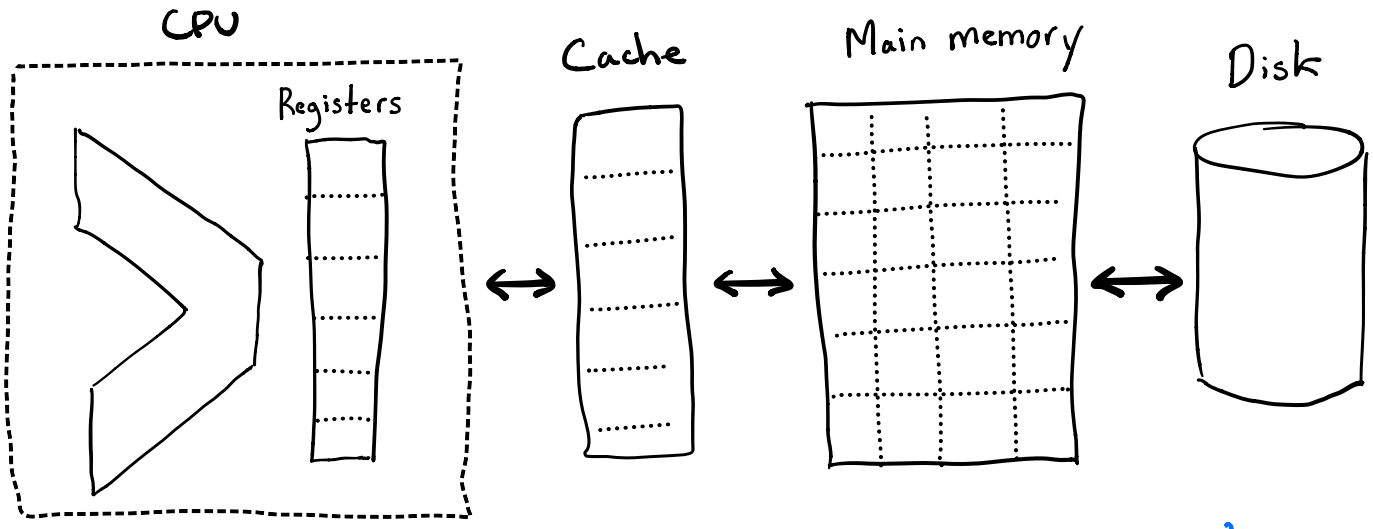
- Arithmetic operations (addition, subtraction, ...)
- Logical operations (and, or, not, ...)
- Flow control (conditional and unconditional jumps)

Differences between machine languages:

- Richness of operations
- Supported data types (e.g. long int, floating point)
- Supported addressing modes

Memory Hierarchy

Modern computers use a hierarchy of memory with different cost/performance tradeoffs.



fast
small
$$$$

slow
large
$

What is "performance"?

1) Bandwidth
2) Latency

Number and function of registers are a central part of the machine language.

The Hack ISA (machine language)

2 main instructions:

- 16-bit A-instructions
- 16-bit C-instructions

3 "registers":

- 16-bit D register
- 16-bit A register
- M represents the 16-bit RAM register addressed by A

A-instruction

Symbolic: @value

Binary: 0 v v v v v v v v v v v v v v v

15-bit value

C-instruction

Symbolic: dest = comp ; jump

Binary: 1 1 1 a c1 c2 c3 c4 c5 c6 d1 d2 d3 j1 j2 j3

unused

I/O devices

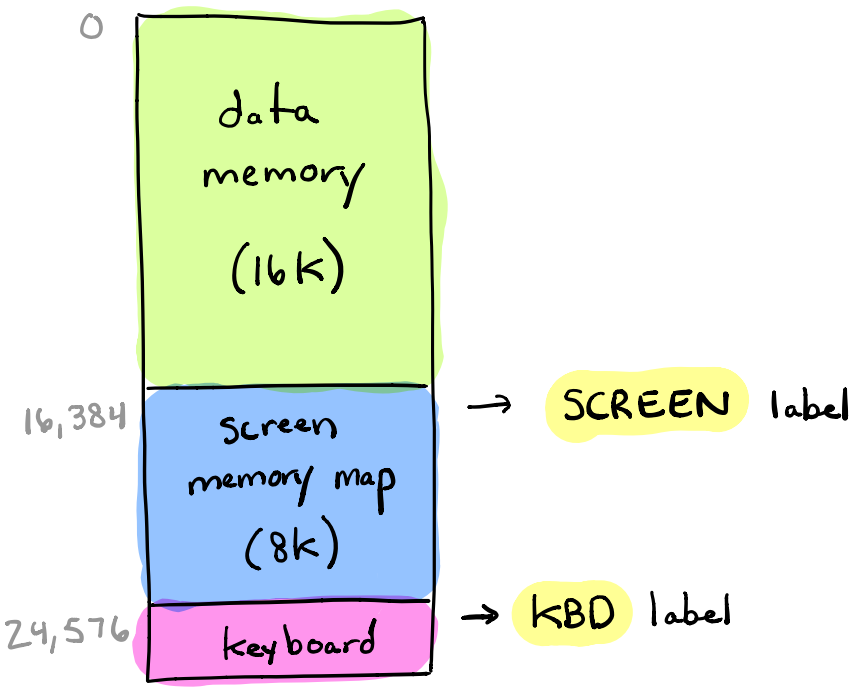
Hack computer platform has two I/O devices:

- Display - output
- keyboard - input

Hack uses memory-mapped I/O!

- Screen contents are mapped to RAM address range
  - 1-bit display, 256 rows x 512 columns
  - 32 16-bit words per row (32 · 16 = 512)
- keyboard is associated with a keyboard memory map (single 16-bit register)

Hack memory space



Perspective

- Hack ISA is very simple, but it can still compute anything that can be computed (at a performance cost)