

CS 42—Stored-program computers, part 1

Tuesday, September 25, 2018

Summary

This week, we'll be talking about **stored-program computers**—a sequential-logic design that allows people to write software (i.e., programs that can be stored *in* a computer and executed *by* that same computer).

Terminology

Here is some key terms we'll use when talking about stored-program computers

Central Processing Unit (CPU): the part of the computer that runs the program, also known as the processor

Register: a small amount of storage, located near the CPU so that its contents can be accessed quickly. Think of it like a variable and / or like one of the “words” of memory you're building in the current assignment.

Random-access Memory (RAM): larger and slower (but still pretty fast) memory, used to store the program, among other things.

Address: a way to refer to a particular location in RAM.

Program counter (PC): A special register whose value is the RAM address of the next instruction to be executed.

Instruction register (IR): A special register whose value is the next instruction to be executed.

Fetch / execute cycle: The main loop that a stored-program computer uses to run a program. The loop follows these steps:

1. Fetch the next instruction from RAM. In other words, copy the contents in the RAM address at the program counter into the instruction register.
2. Decode the current instruction. In other words, look at the value in the instruction register and figure which operation should be performed and with what data.
3. Execute the current instruction. Perform the operation identified in the previous step, including any reading or writing that should happen as part of the operation.
4. Update the program counter so that its value is the location of the next instruction.
5. Goto step 1.

The Harvey Mudd Miniature Machine (HMMM)

Today, we'll look at a simple stored-program computer and part of its programming language. You don't need to memorize the syntax, but you might want to take notes about important features.

Next time: Functions in Hmmm