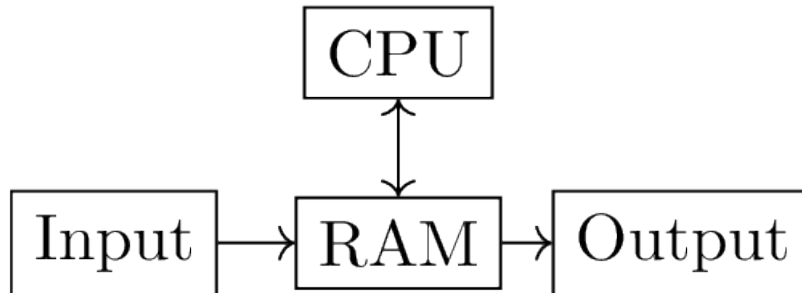


Contents

1 A Hardware Block Diagram

1

1 A Hardware Block Diagram



This is a diagram of the hardware of a computer. The boxes are components and the arrows show the movement of data.

CPU The *central processing unit* is where instructions execute. Instructions and data in the RAM are loaded into the CPU (it has **extremely** limited memory called *registers*) and manipulated; if the value is to be saved, registers can be stored back to RAM. One special register is the *instruction pointer* (IP). The IP always has the address of the next machine instruction the CPU is to execute.

RAM The *random-access memory* is where data (think: variables) and instructions (think: code) are stored while a computer is running. The RAM and CPU are connected with a bidirectional connection. When you were told computer memory is an "array of bytes", this was the RAM.

Input The collection of input devices such as hard drives, network connections, and keyboards.

Output The collection of output devices such as hard drives, network connections, and screens. Notice that in the block diagram a hard drive, for example, looks like two items, depending on how it is being used. This separation is a simplification.

A *program* is a collection of machine code and variables. When you type the name of the executable in a shell or double click on an icon in

the GUI, the operating system *loads* the program. This means it copies the instructions and variables into the RAM from the hard drive (technically it could come from any input device). Then the *instruction pointer* (IP; also known as the *program counter* or PC) is set to the address of the first instruction. On any given platform, every program starts at the same address.

The computer, itself, runs a very simple loop at its heart:

```
while (!machine halted)
  fetch RAM[IP]
  decode instruction
  execute instruction
```

This is typically abbreviated as the *fetch-decode-execute* cycle. Modern computers do not often actually halt, instead loading and running program after program at the direction of the *operating system* (which is, itself, just a program with instructions that are fetched, decoded, and executed).