Info Representation

This is not accurate but will do for now.

⇨ Where to begin?

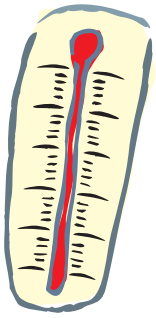
Computers store & process info

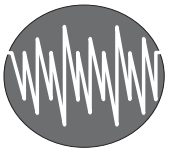
⇨ Info representation

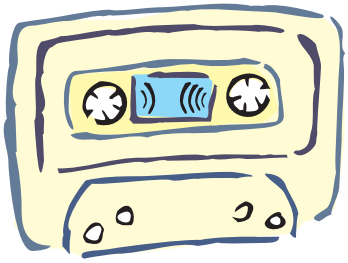
In what form?

Analog Representation

Analog devices store and process information *encoded* as continuous physical signals.



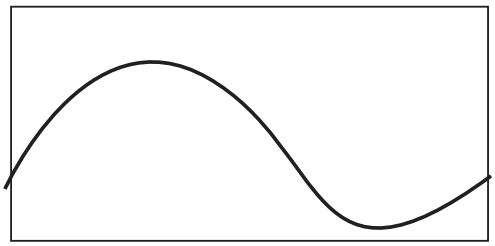


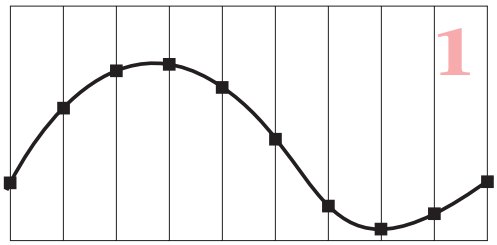


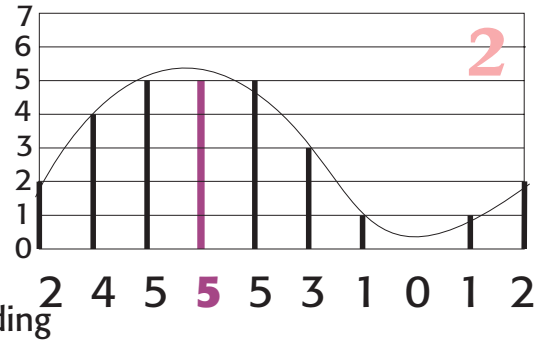
Quiz
What is the desired information in shown devices? What is the continuos *analog signal* in each case?

Digital Representation

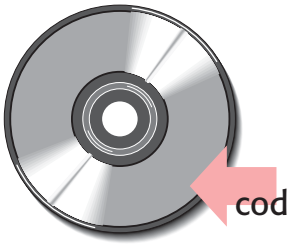
Analog to digital (A-D) conversion







Digital devices store and process information encoded as numbers.

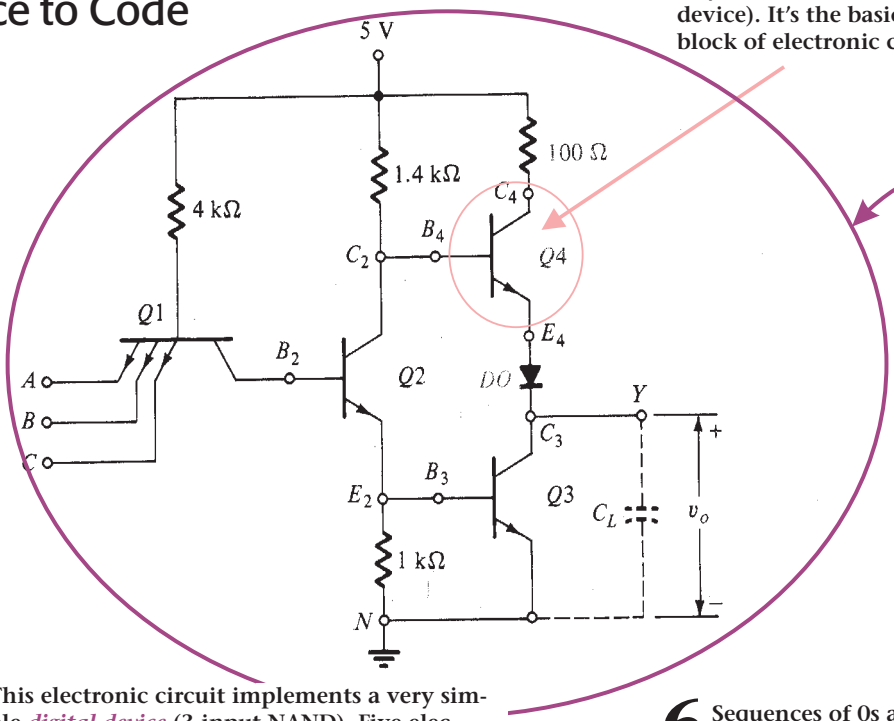


Electronic Circuits

From Physical Device to Code

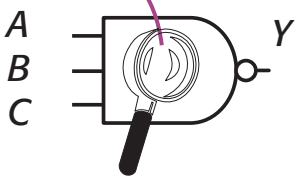
1 Electronic circuits are used to *implement* digital devices.

2 The transistor is an elementary *electronic switch* (2-state device). It's the basic building block of electronic circuits.



3 This electronic circuit implements a very simple *digital device* (3-input NAND). Five electronic switches are used: 4 transistors, and a diode (simpler electric current switch). **Source:** Microelectronics, Millman & Grabel, 2nd ed (my electronics textbook when I was a student).

4 The NAND gate is a *logical switch.* It switches based on some logic (always ON except when all inputs are 1). It is a basic building block of computing digital devices.



bit (binary digit)

5 Any two symbols can *encode* the two states of logic switches (we like 0 and 1; you will see why later).

0101010101

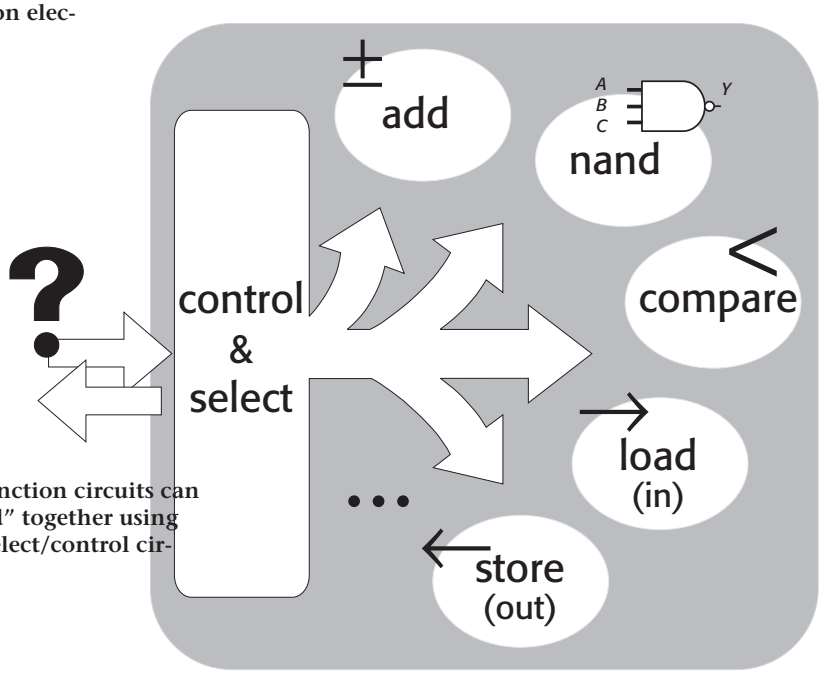
Byte = 8 bits

6 Sequences of 0s and 1s encode the states of switches used in digital devices (in other words, encode the state of the device).

7 The resulting *binary code* can be interpreted as a number expressed in the binary number system.

Multifunction Devices

A multifunction digital device can be constructed from many single-function electronic circuits.

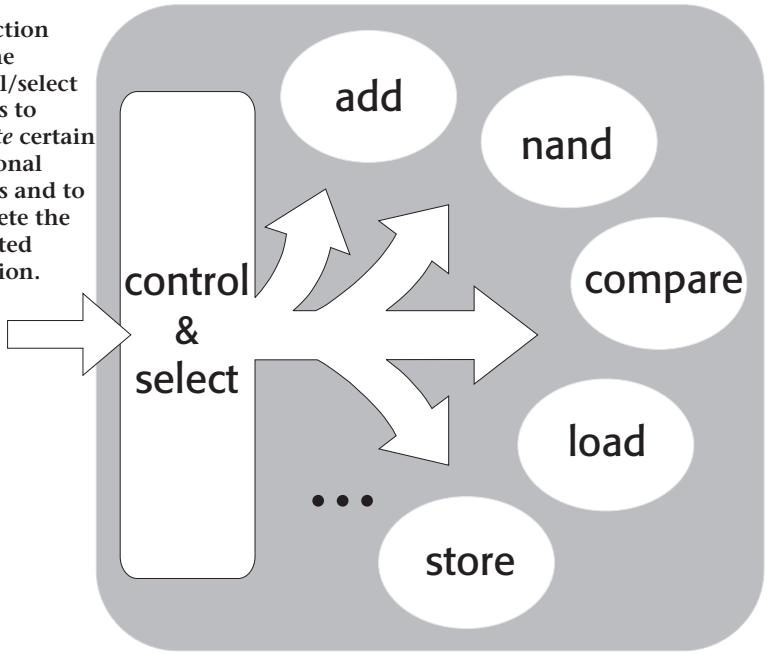


Single-function circuits can be "glued" together using special select/control circuitry.

How to communicate with the multifunction device?

Encoding, Instruction Sets & Machine Language

3 Each instruction tells the control/select circuits to *activate* certain functional circuits and to complete the requested operation.

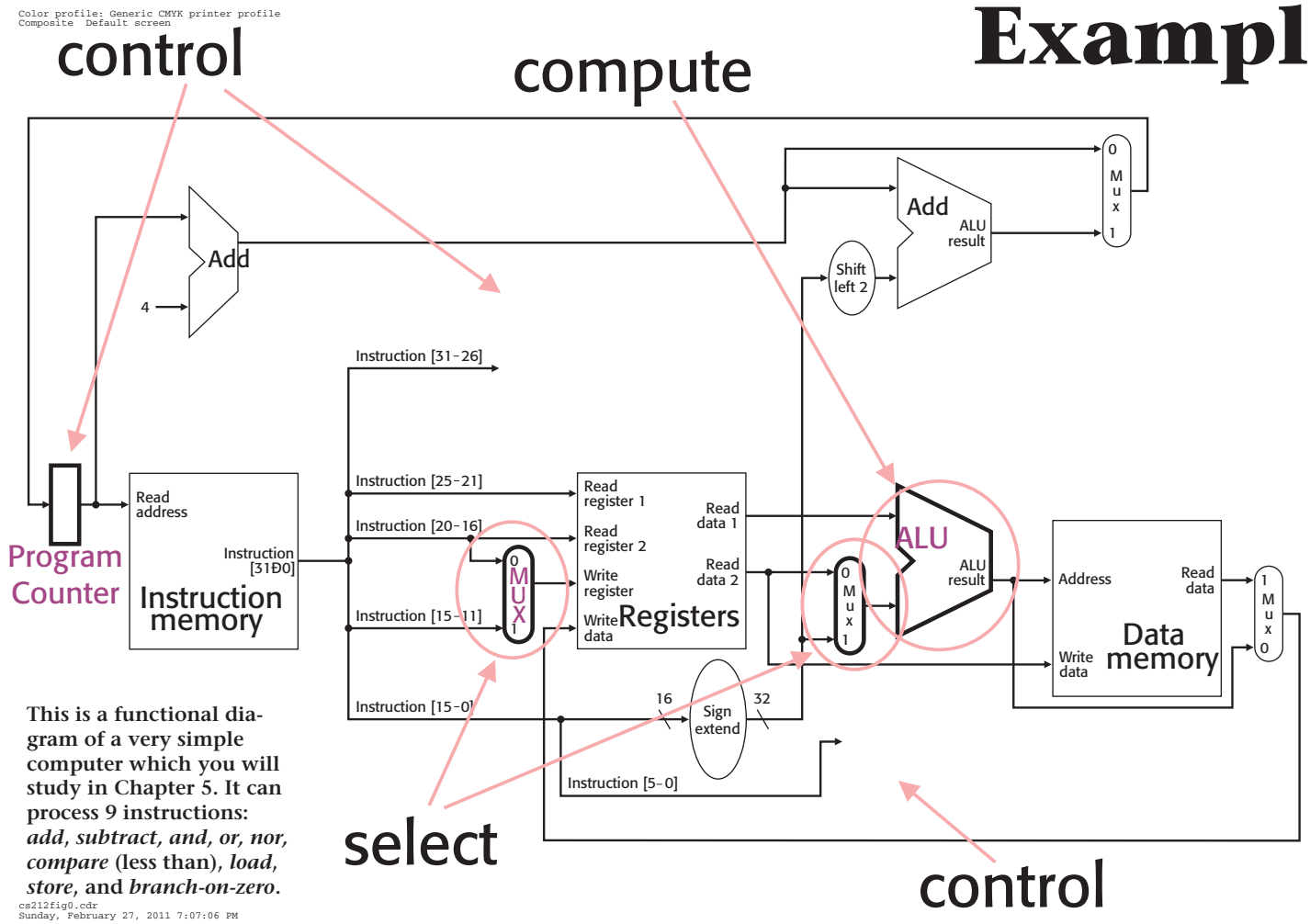


1 Numbers can *encode* (represent) operations (ops) performed in circuits.

add	0	000
nand	1	001
compare	2	010
load	3	011
store	4	100
. . .		

2 An *instruction* uses the operation's binary code (or *opcode)* to encode the operation together with any info needed to complete that operation.

Example



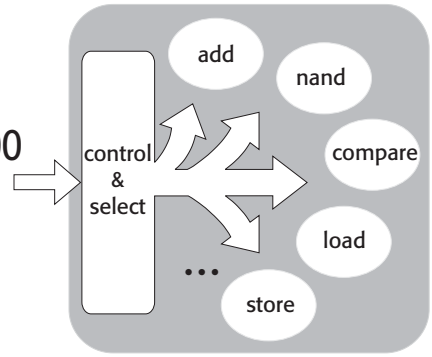
This is a functional diagram of a very simple computer which you will study in Chapter 5. It can process 9 instructions: *add*, *subtract*, *and*, *or*, *nor*, *compare* (less than), *load*, *store*, and *branch-on-zero*.

Hard vs. Soft Ops

Hardware operation

Some circuit handles the operation directly. That circuit is activated by an instruction.

00000000101000010000000000011000



Software operation

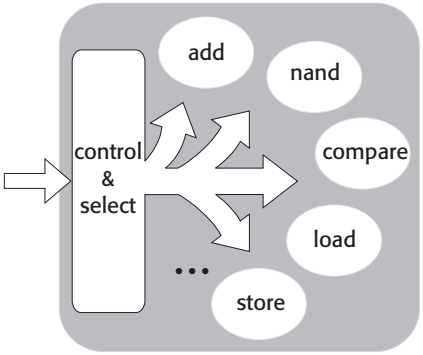
No particular circuit is able to do the operation. The operation is logically defined by a *sequence* of simpler hardware operations, each encoded by an instruction that activates some functional circuit.

10101100111100100000000000000000

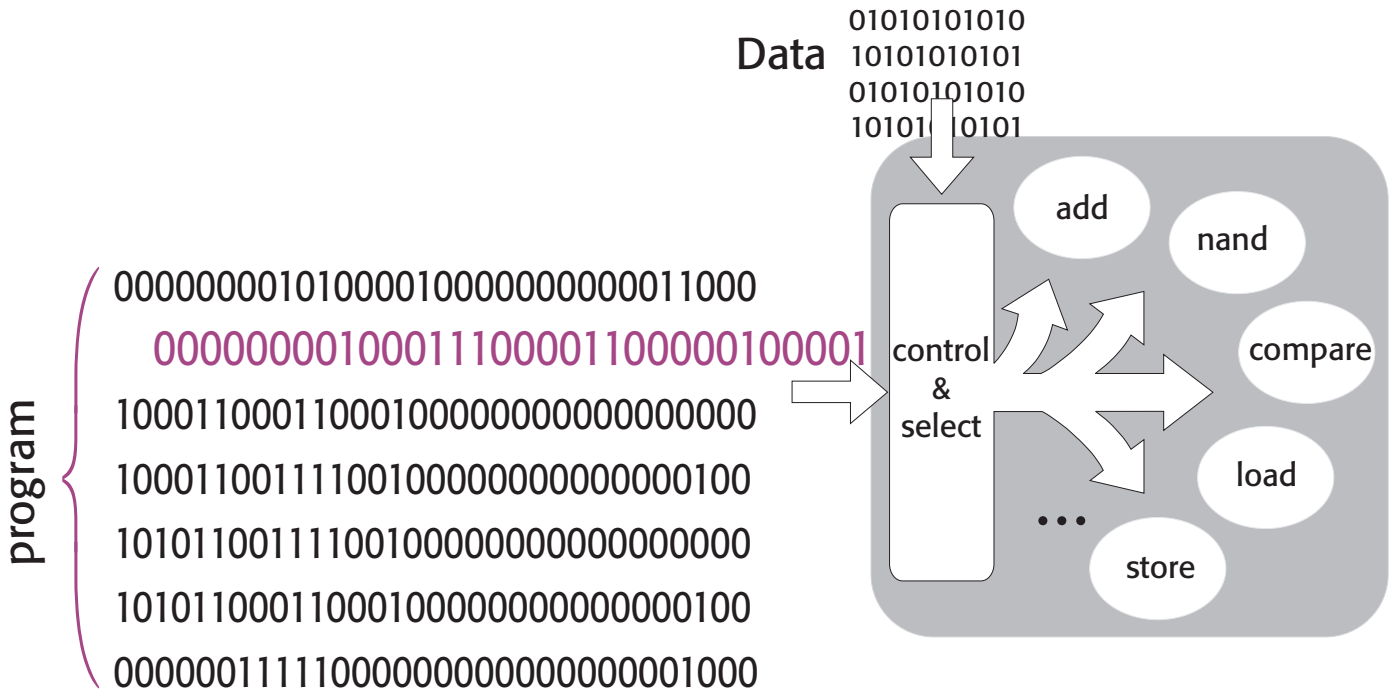
10101100011000100000000000000100

00000011111000000000000000001000

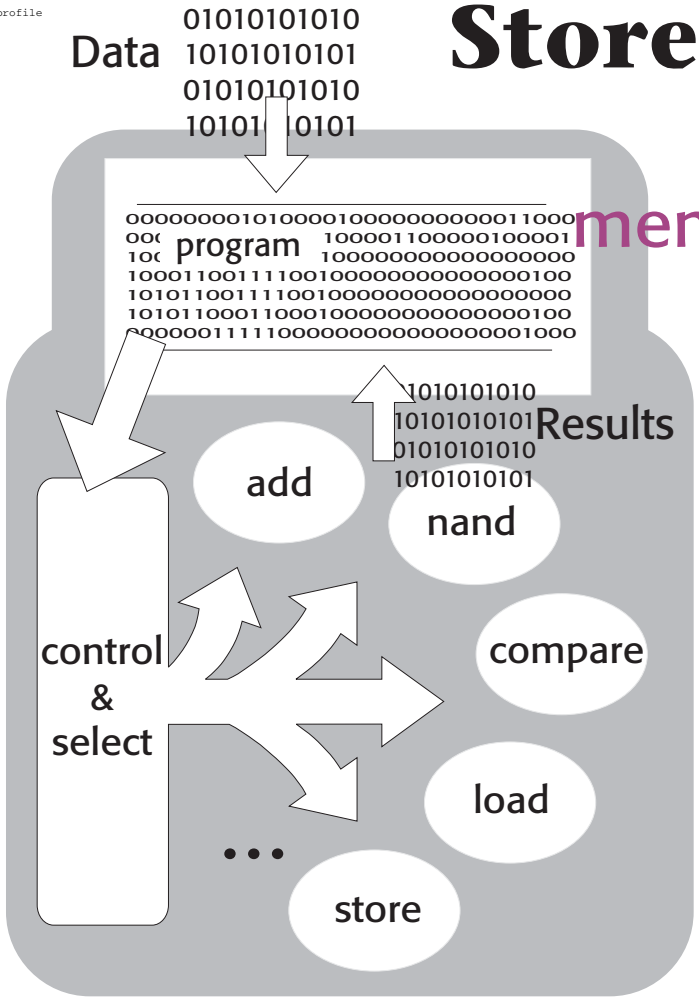
Example: repeat add number 3 times to create multiply-by-3 function



Programming



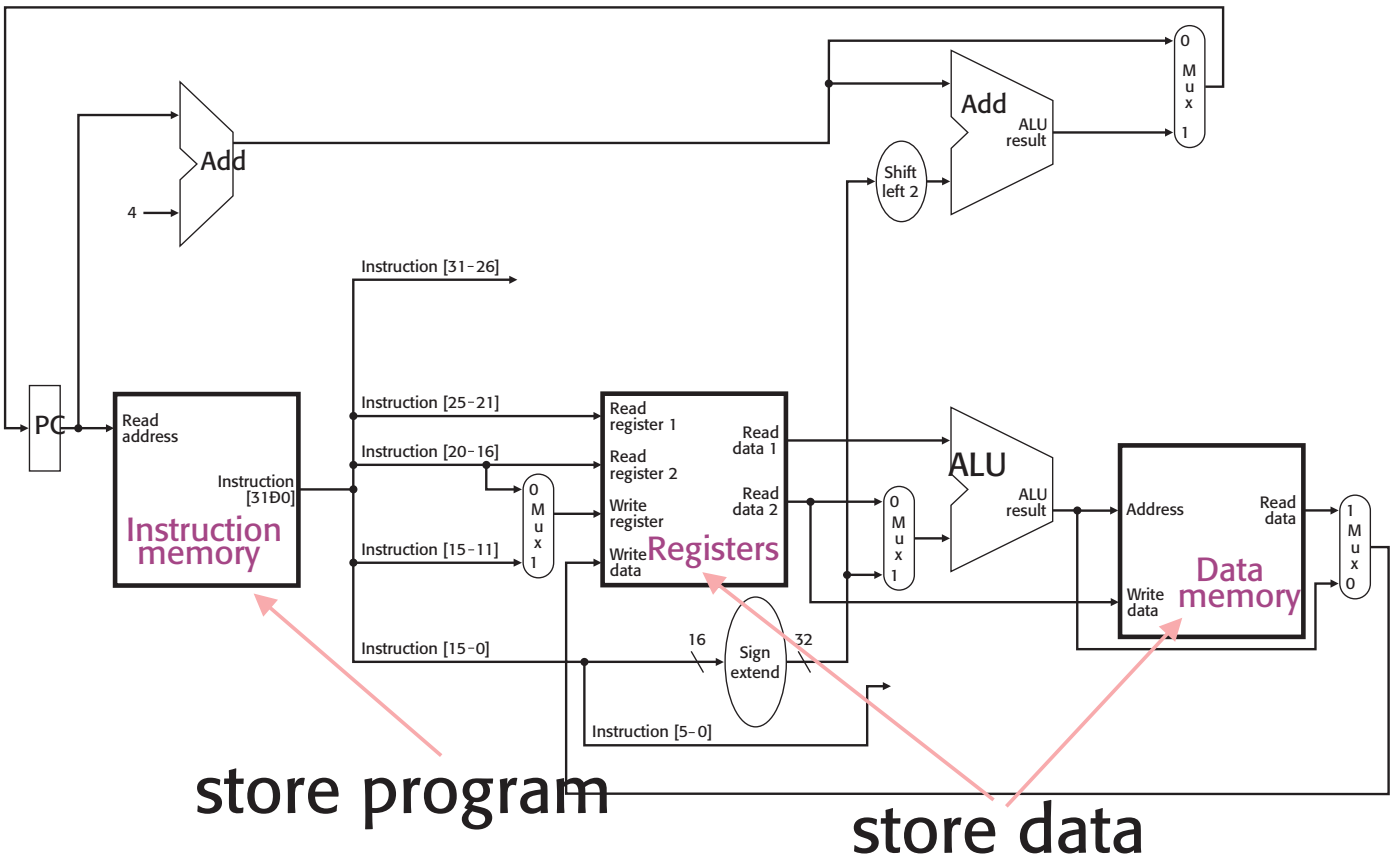
Stored Program



Memory is an electronic circuit which stores program instructions in the same form as data.

Quiz
Do you see now why a computer is much more that a device to store and process info?

Example

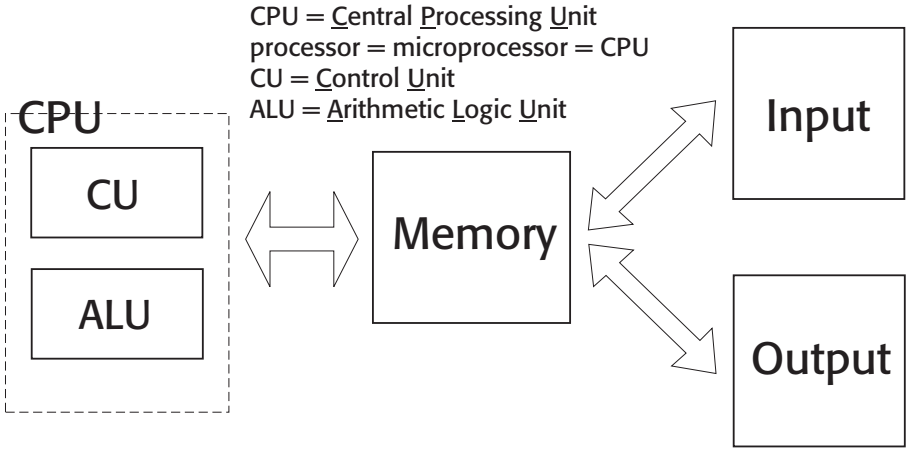


von Neumann Computer

Check your textbook about opinion on naming the first modern computer model after von Neumann. (Use the index to lookup "von Neumann".)

The main feature of the von Neumann computer is that instructions are binary-coded, the same as data, and stored in the same memory.

CPU = Central Processing Unit
processor = microprocessor = CPU
CU = Control Unit
ALU = Arithmetic Logic Unit



Modern Computers

- Programmable: stored program
- Digital
- Electronic (so far!)
- Multifunction

Quiz
Why is binary code so essential to modern computers?

Early Modern Computers

ENIAC

First programmable electronic computer
US WW2-1946, Eckert & Mauchly

EDSAC

First stored program
UK 1949, Wilkes

1st Week Assignment

- ✓ 1-Page syllabus
- ✓ FAQ (in Arabic)
- ✓ 📖 1.1 (pp. 7-9 What You Can Learn...)