What is Computing?

Scientific Computing

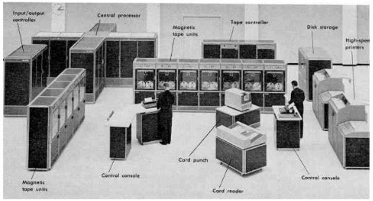
Stefan Abi-Karam

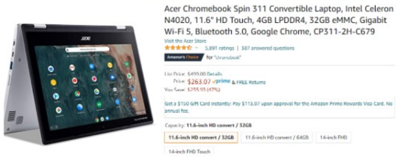
Summer 2023

Table of Contents

Table of Contents

What is a Computer?





Acer Chromebook Spin 311 Convertible Laptop, Intel Celeron N4020, 11.6" HD Touch, 4GB LPDDR4, 32GB eMMC, Gigabit Wi-Fi 5, Bluetooth 5.0, Google Chrome, CP311-2H-C679

Visit the Acer Store

5,891 ratings | 387 answered questions

Amazon's Choice for "chromebook"

List Price: $499.00 Details

Price: $263.07 prime & FREE Returns

You Save: $235.93 (47%)

Get a $150 Gift Card instantly: Pay $113.07 upon approval for the Amazon Prime Rewards Visa Card. No annual fee.

Capacity: 11.6-inch HD convert / 32GB

11.6-inch HD convert / 32GB | 11.6-inch HD convert / 64GB | 14-inch FHD | 14-inch FHD Touch

A "computer" is anything that executes a set of defined instructions.

- We will mainly talk about computers that use electrical circuits.
- We can assume instructions are executed one at a time in order.
- In the common sense of the word, we will be talking about computers like your laptop or desktop computer (or also your phone).

Operating Systems

Most computers we interact with have a special set of instructions call an operating system that manage a lot of the tedious task of running the computer.

- Manages your internet connection
- Manages your screen display
- Manages your files and storage

Operating systems also provide an easy way to run our own instructions to do cool stuff along with other features.

- I can run software other people wrote, like a web browser or games.
- I can write my own code to run on the computer to do my research stuff.

Operating Systems

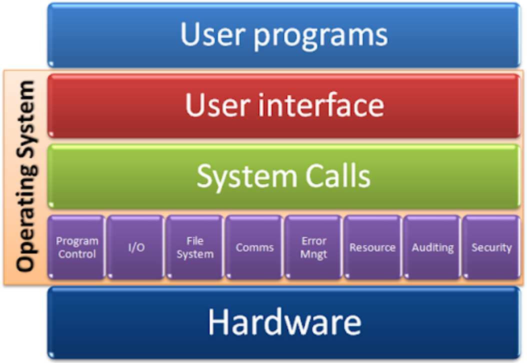


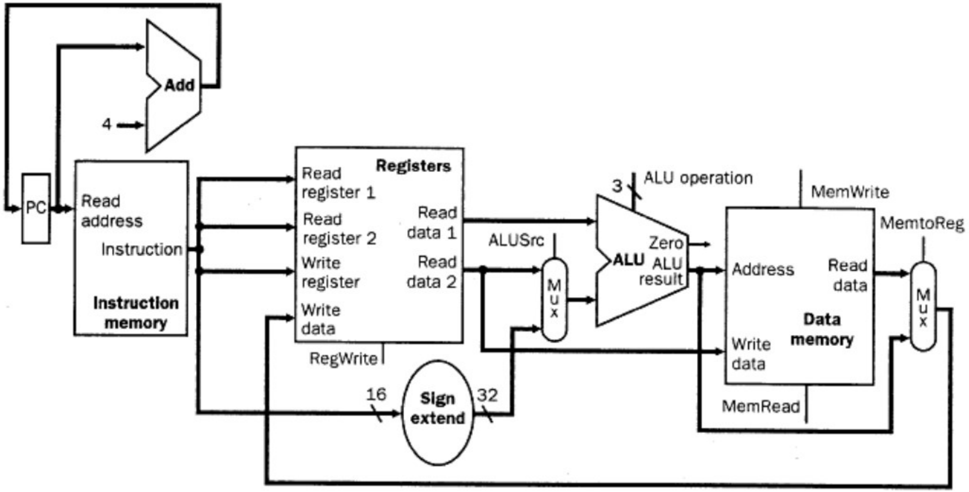
Table of Contents

Lowest Level of a Computer

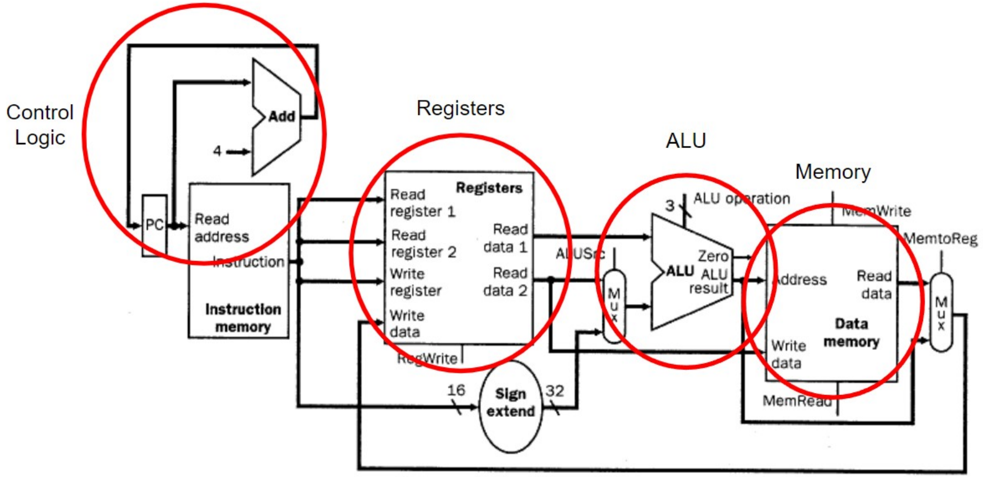
In the lowest level of hardware, there are several components:

- **Clock**: Keep a constant time, one tick every x seconds
- **Registers**: Where data is stored while working with it
- **Arithmetic Logic Unit (ALU)**: Does the actual computations
  - Add / subtract two numbers, compare two numbers, execute logic functions (`AND`, `OR`, `INVERSE`, `XOR`)
  - Modern computers have cool features built-in like multiply, divide, multiply-add, decimal math, etc...
- **Control Logic**: Tells the computer what to do next
  - Keeps track of current instruction and next instruction
  - Can jump to different instructions and save current location to jump back to later
- **Memory**: Where data is stored while it's waiting to be worked on
  - Temporary - gets erased when you turn off the computer
- **Storage (Optional)**: Where data is stored in the long-term
  - Permanent - doesn't get erased when you turn off the computer

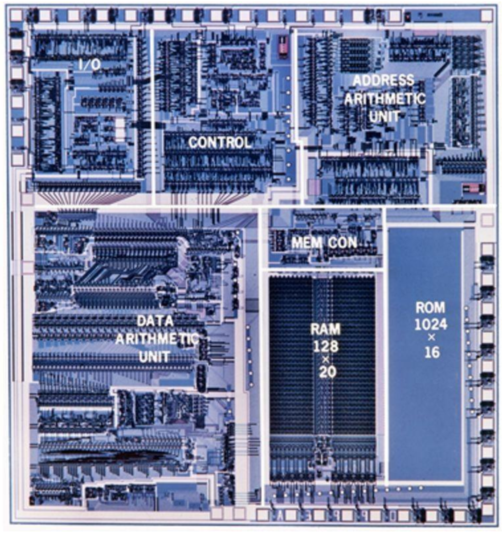
Lowest Level of a Computer



Lowest Level of a Computer



Lowest Level of a Computer

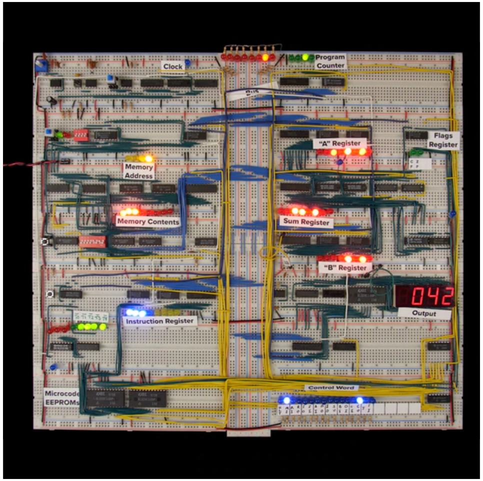


These elements usually make up your computer's central processing unit, or CPU. In reality, your computer can have

more than one copy of a CPU "core" to do different things simultaneously, making it faster to compute overall. Your computer

also has a bunch of extra hardware to handle power, graphics, storage, and interfacing with other things like your mouse, keyboard, screen, camera, USB port, wifi chip, ...

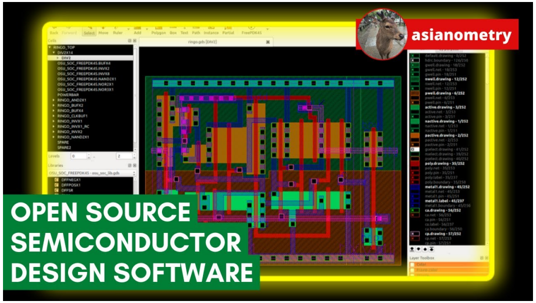
Making a Computer



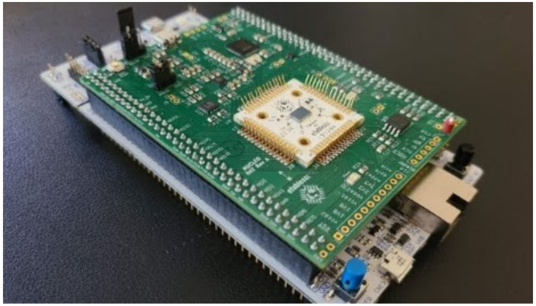
"Build an 8-bit computer from scratch"

You (yes you!) can make your own computer very easily from scratch. In fact RISC-V and other open-source hardware tools are commnly used by both students and professionals to do reseach and develop new hardware.

Making a Computer



"The Promise of Open Source Semiconductor Design Tools"



Example of a fabricated chip using open-source tools and fabrication technology

Table of Contents

Computer Instructions

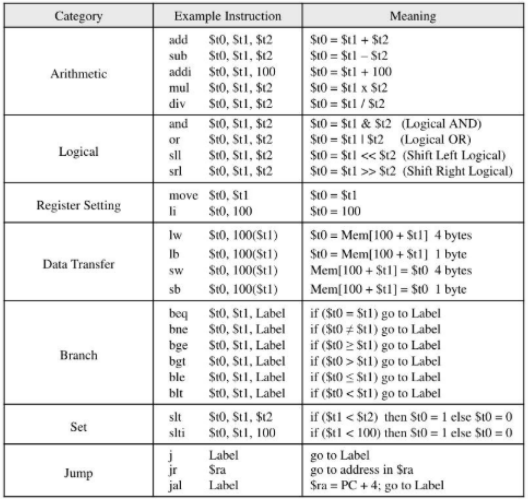
At the lowest level, each CPU hardware design has a set of basic instructions called an Instruction Set Architecture or **ISA**.

- Intel has an ISA called `x86`
- Many phones and Apple computers use an ISA called `ARM`
- Arduinos use an ISA called `AVR`

An ISA is an outline of the set of instructions that a CPU can perform.

- Add two numbers
- Move this number from location A to location B
- Compare two numbers or compare a number to zero
- Jump to this location in the code
- Run this block of code and come back when done
- Load and store this number in this location of memory

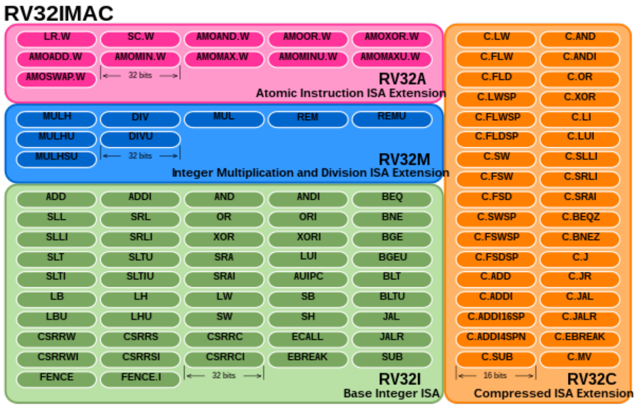
Computer Instructions



Category	Example Instruction	Meaning
Arithmetic	add $t0, $t1, $t2	$t0 = $t1 + $t2
	sub $t0, $t1, $t2	$t0 = $t1 – $t2
	addi $t0, $t1, 100	$t0 = $t1 + 100
	mul $t0, $t1, $t2	$t0 = $t1 x $t2
	div $t0, $t1, $t2	$t0 = $t1 / $t2
Logical	and $t0, $t1, $t2	$t0 = $t1 & $t2 (Logical AND)
	or $t0, $t1, $t2	$t0 = $t1 | $t2 (Logical OR)
	sll $t0, $t1, $t2	$t0 = $t1 << $t2 (Shift Left Logical)
	srl $t0, $t1, $t2	$t0 = $t1 >> $t2 (Shift Right Logical)
Register Setting	move $t0, $t1	$t0 = $t1
	li $t0, 100	$t0 = 100
Data Transfer	lw $t0, 100($t1)	$t0 = Mem[100 + $t1] 4 bytes
	lb $t0, 100($t1)	$t0 = Mem[100 + $t1] 1 byte
	sw $t0, 100($t1)	Mem[100 + $t1] = $t0 4 bytes
	sb $t0, 100($t1)	Mem[100 + $t1] = $t0 1 byte
Branch	beq $t0, $t1, Label	if ($t0 = $t1) go to Label
	bne $t0, $t1, Label	if ($t0 ≠ $t1) go to Label
	bge $t0, $t1, Label	if ($t0 ≥ $t1) go to Label
	bgt $t0, $t1, Label	if ($t0 > $t1) go to Label
	ble $t0, $t1, Label	if ($t0 ≤ $t1) go to Label
	blt $t0, $t1, Label	if ($t0 < $t1) go to Label
Set	slt $t0, $t1, $t2	if ($t1 < $t2) then $t0 = 1 else $t0 = 0
	slti $t0, $t1, 100	if ($t1 < 100) then $t0 = 1 else $t0 = 0
Jump	j Label	go to Label
	jr $ra	go to address in $ra
	jal Label	$ra = PC + 4; go to Label

An example of a simple ISA

Computer Instructions



An example of a RISC-V ISA

Programming Langauges

Programmers want a more expressive way to write programs for their computers with more features rather than just the assembly language

They designed higher level languages that can be translated, aka compiled, into lower level code (ex. an ISA)

These higher level languages have more useful features and better language and syntax

- Standard libraries for things like math, file reading, networking, graphics, user interfaces
- Memory management
- Concurrency and parallelism

Programming Langauges

You can now build higher level functionality like a file system, a desktop user interface, user management, networking (wifi and ethernet), a way to run other applications, and a web browser

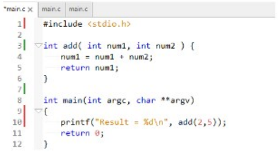
All of a sudden you have an operating system

- That previous point kinda sounds like Chrome OS on your Chromebook
- Windows, Mac, Linux

Programming Langauges

Programs and Applications

Higher Level Language (ex. C / C++)

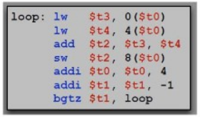


```c
#include <stdio.h>

int add( int num1, int num2 ) {
    num1 = num1 + num2;
    return num1;
}

int main(int argc, char **argv)
{
    printf("Result = %d\n", add(2,5));
    return 0;
}
```

Assembly Instructions



```
loop: lw   $t3, 0($t0)
      lw   $t4, 4($t0)
      add  $t2, $t3, $t4
      sw   $t2, 8($t0)
      addi $t0, $t0, 4
      addi $t1, $t1, -1
      bgtz $t1, loop
```

Running On Hardware

Example of abstaction for software and hardware layers