COMP 2825
Computer Architecture/Organization

Instructor: Rahim Oraji

Name : Rahim
Email : roraji@bcit.ca
Office Hours : Online appointment

Evaluation Criteria

Criteria	%	
Quizzes	20	Weekly.
Home reading labs	30	Labs which must be completed by each student individually.
Mid-Term Exam	25	You must achieve a minimum 50% average between the Mid-Term and Final Exam in order to pass this course.
Final Exam	25	You must achieve a minimum 50% average between the Mid-Term and Final Exam in order to pass this course.
Total:	**100%**	**Total Passing Grade: 60%**

Passing Grade: 60%

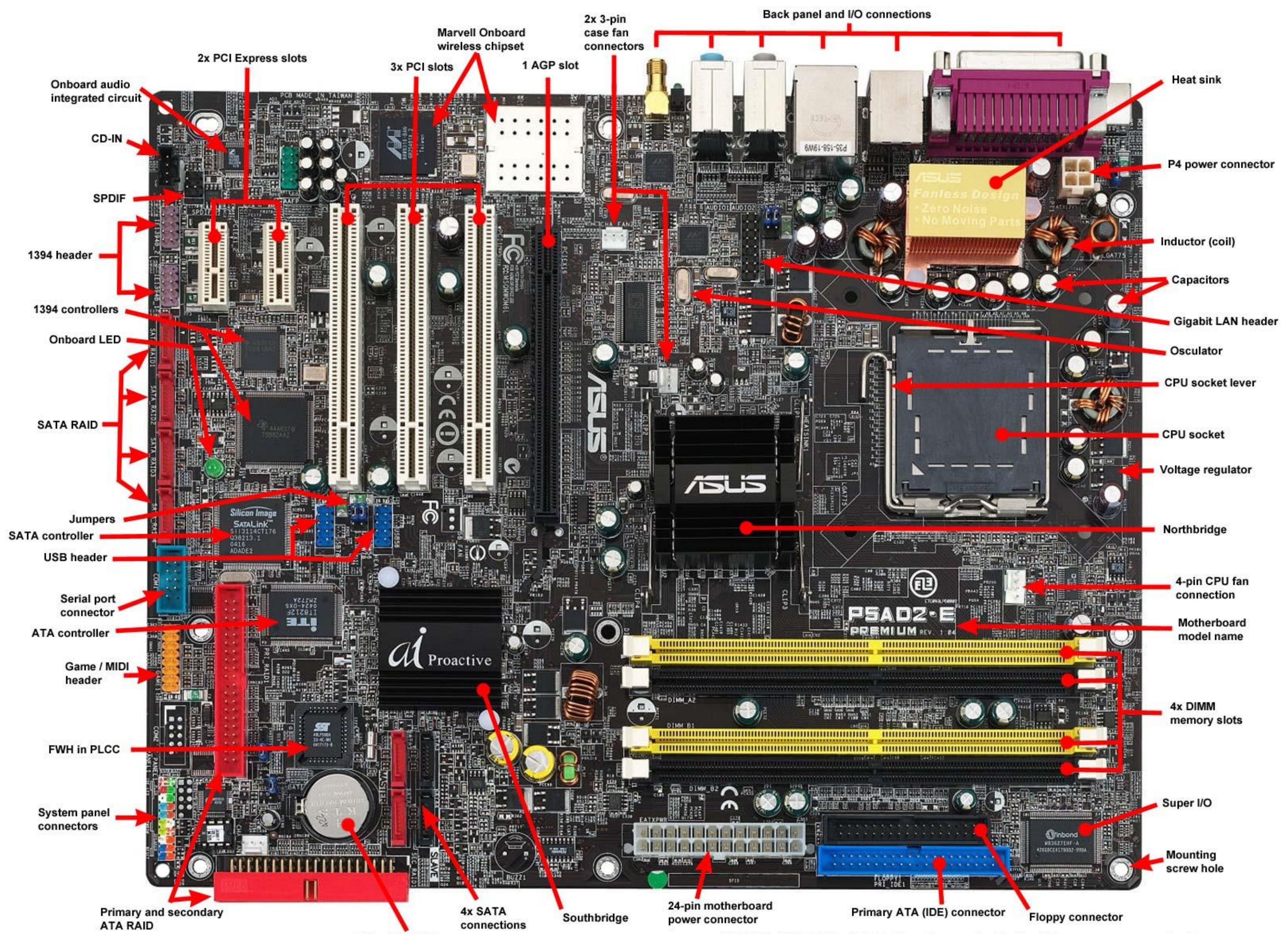
Course outline: https://www.bcit.ca/outlines/comp2825

Structured Computer Organization (Sixth Edition)
ISBN 0-13-148521-0
Andrew S. Tanenbaum
Prentice-Hall

References

- https://tutorialspoint.dev/
- Structured Computer Organization (Sixth Edition)
- Wikipedia
- https://commons.wikimedia.org/

STRUCTURED COMPUTER ORGANIZATION



ASUS P5AD2-E Motherboard - http://www.computerhope.com

Computer architecture

The study of how to design those parts of a computer system that are visible to the programmers is called **computer architecture.** In common practice, however, **computer architecture** and **computer organization** mean essentially the same thing.

Why Computer Architecture?

1. Gaining an understanding of the underlying implementation of code

2. Designing better programs concerning performance, speed, and size

3. Learning important concepts such as Parallelism, caching, and pipelining

4. Understanding time, space, and cost tradeoffs in terms of hardware and software.

What does computer do for us?

Executing given instructions
(e.g. printing, sorting, processing data)

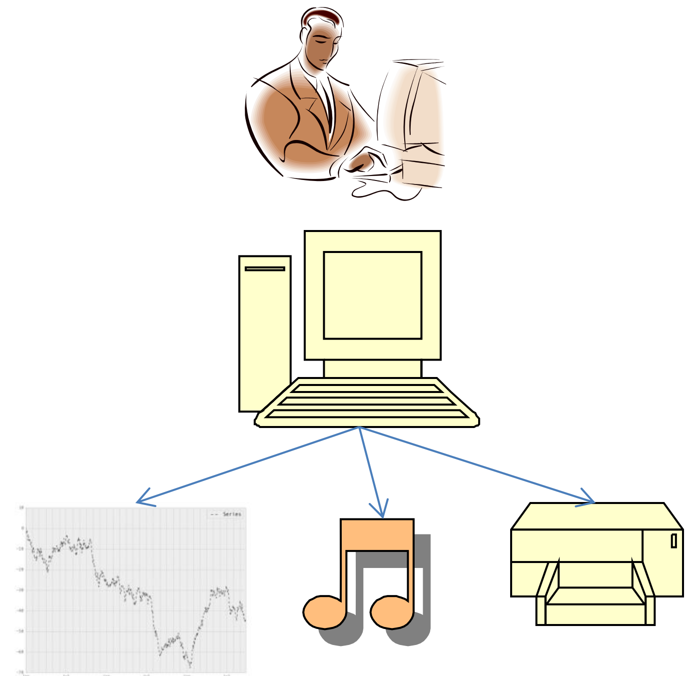
Instructions (program)

↓

Executing

↓

Results



What does computer do for us?

Executing given instructions

(e.g. printing, sorting, processing data)

Program :
a set of instructions written in high level programming language

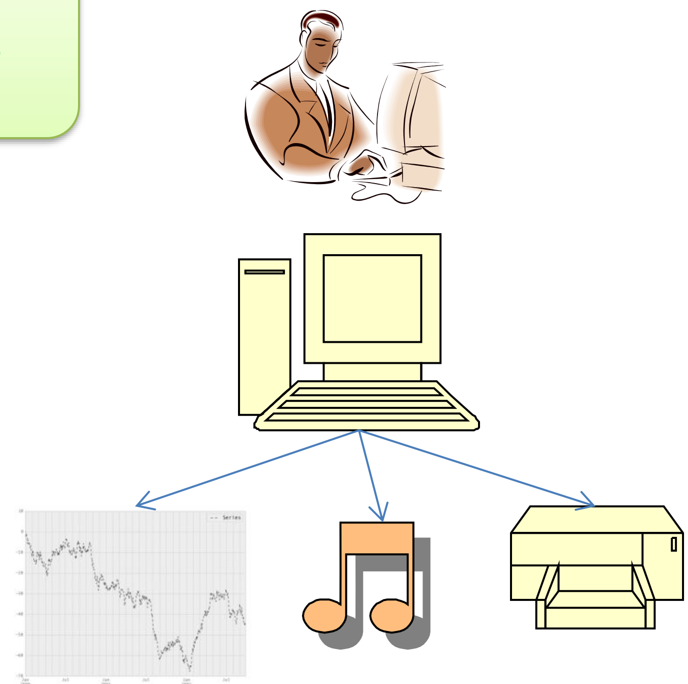
Instructions (program)

↓

Executing

↓

Results



What does computer do for us?

High level programming languages

Python, Java, C++, ...

Why are we using these languages?

```java
/**
Hellow World in Java
**/
class Main {
  public static void main(String[] args) {
    System.out.println("Hello world!");
  }
}
```

```python
# Hello World in Python
print("Hello World")
```

```asm
; Hello World in Assembly
section     .text
global      _start

_start:

    mov     edx,len
    mov     ecx,msg
    mov     ebx,1
    mov     eax,4
    int     0x80

    mov     eax,1
    int     0x80

section     .data

msg     db  'Hello, world!',0xa
len     equ $ - msg
```

What does computer do for us?

Execution

Execution:
Translation vs. Interpretation

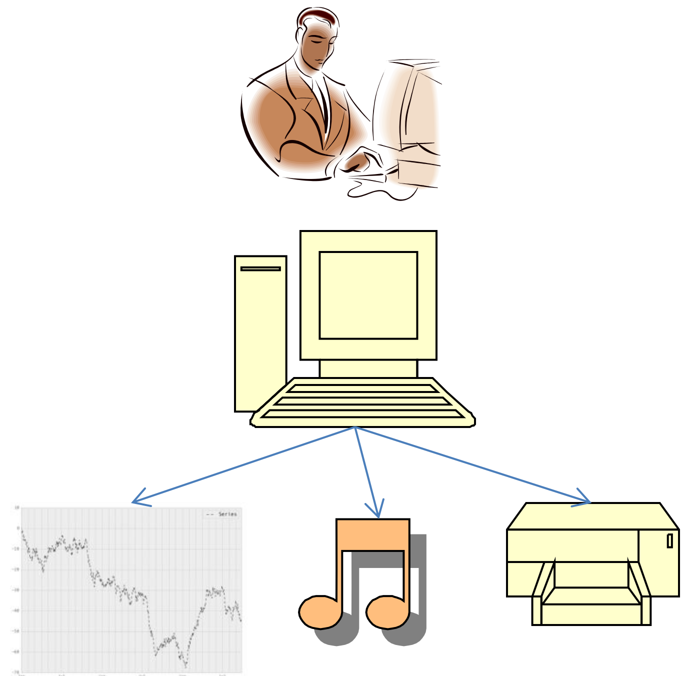
Instructions (program)

↓

executing

↓

results

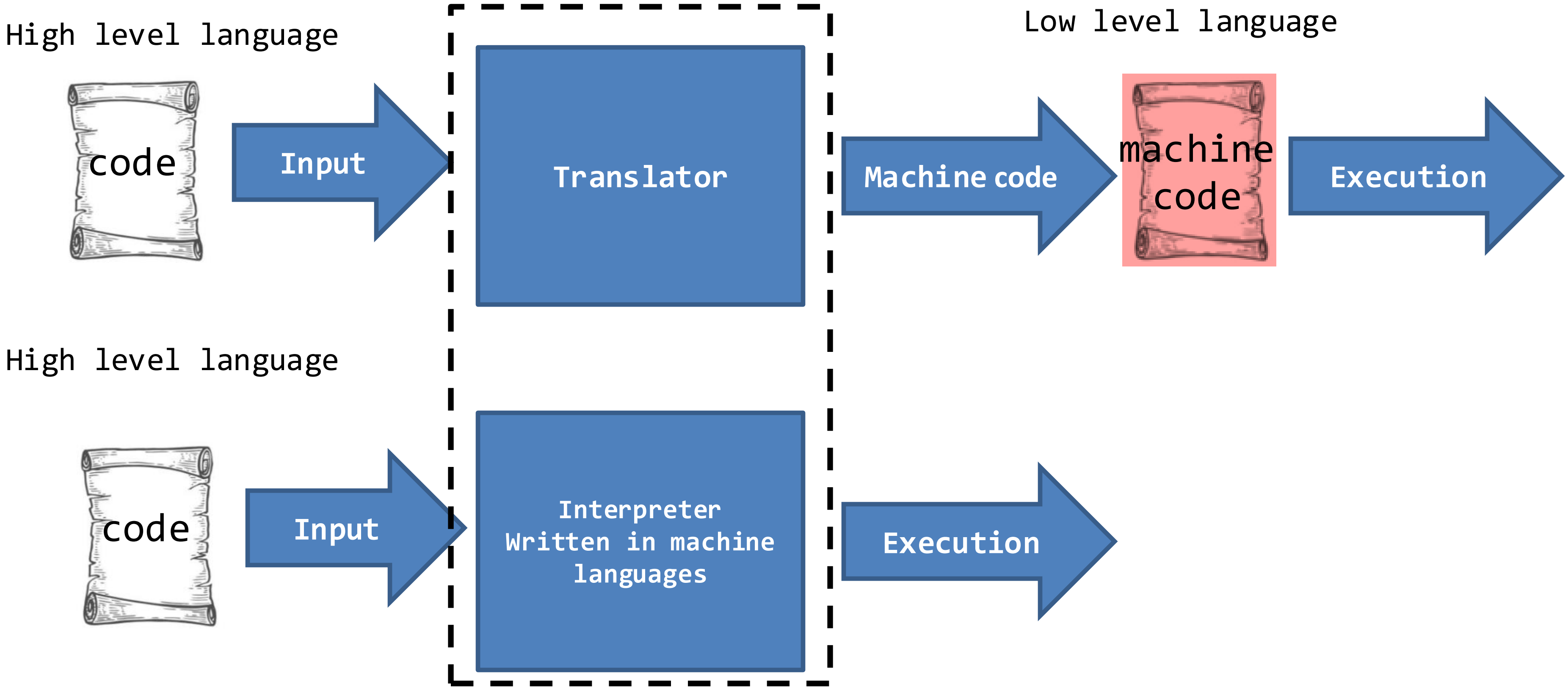


What does computer do for us?

Execution

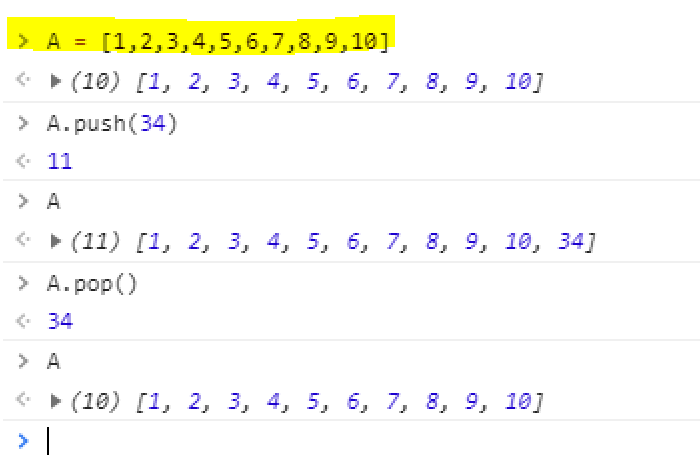
Translation: converting **High level** programs into **Low level** version e.g. Java, C++

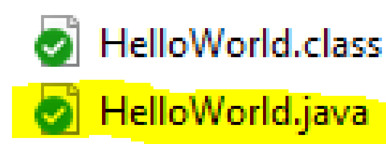
Interpretation: write a program in **Low level** machine that reads **high level** programs and executes it e.g. Python, JavaScript


High level language
code
Input
Translator
Machine code
Low level language
machine code
Execution
High level language
code
Input
Interpreter
Written in machine
languages
Execution

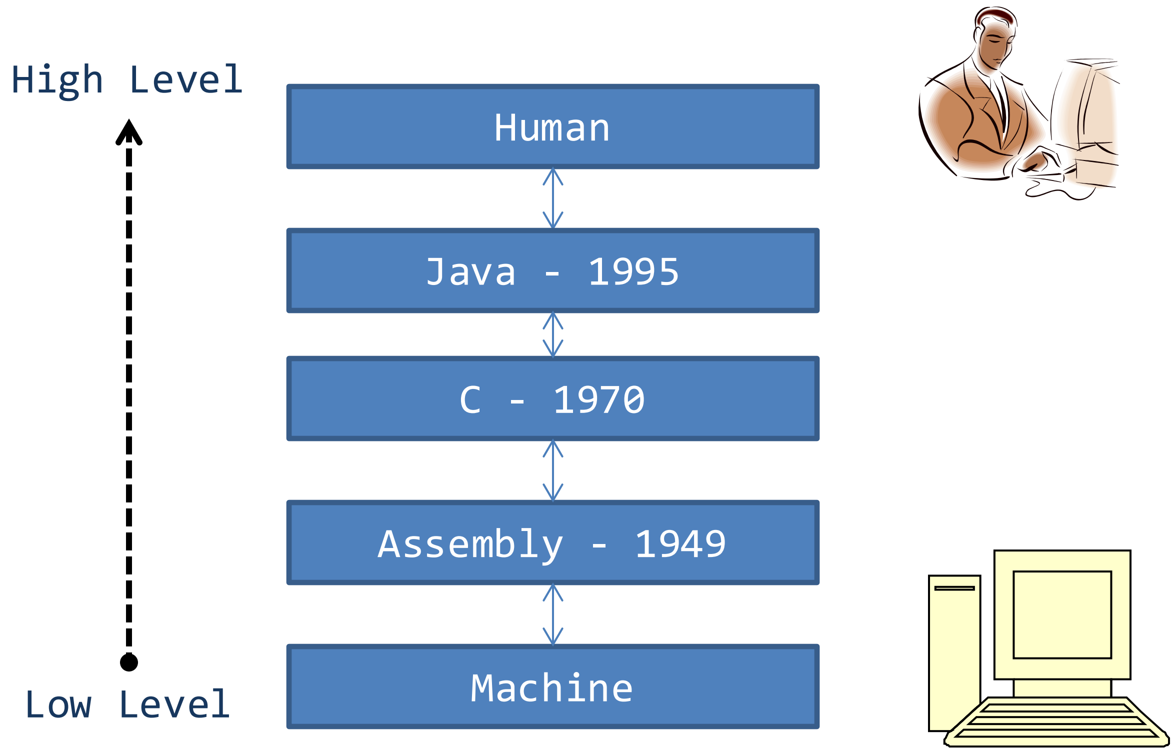
Translation vs. Interpretation Examples

Interpretation	Translation
Python	C++
JavaScript	Java
Perl	C
PHP	Assembly

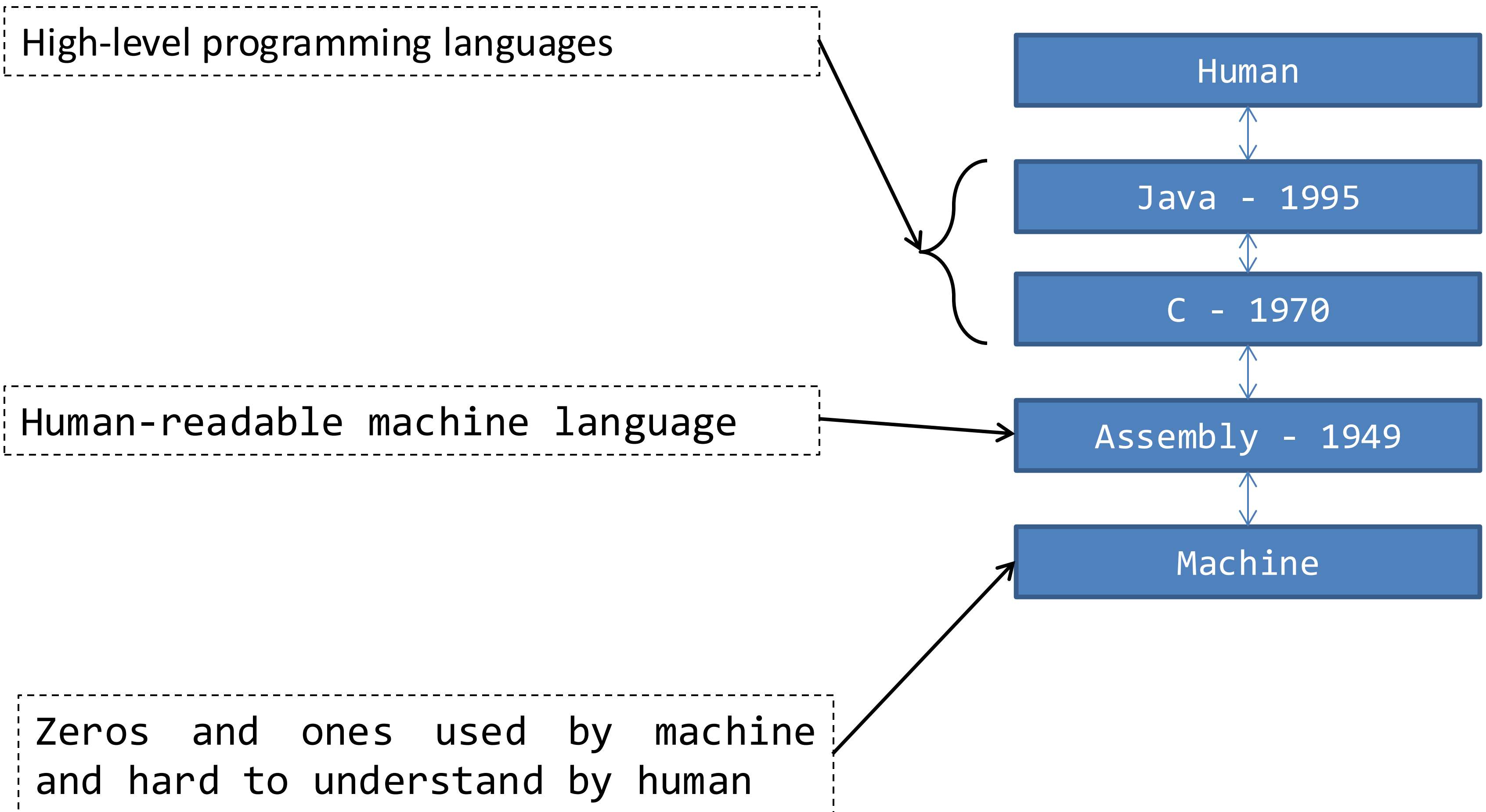




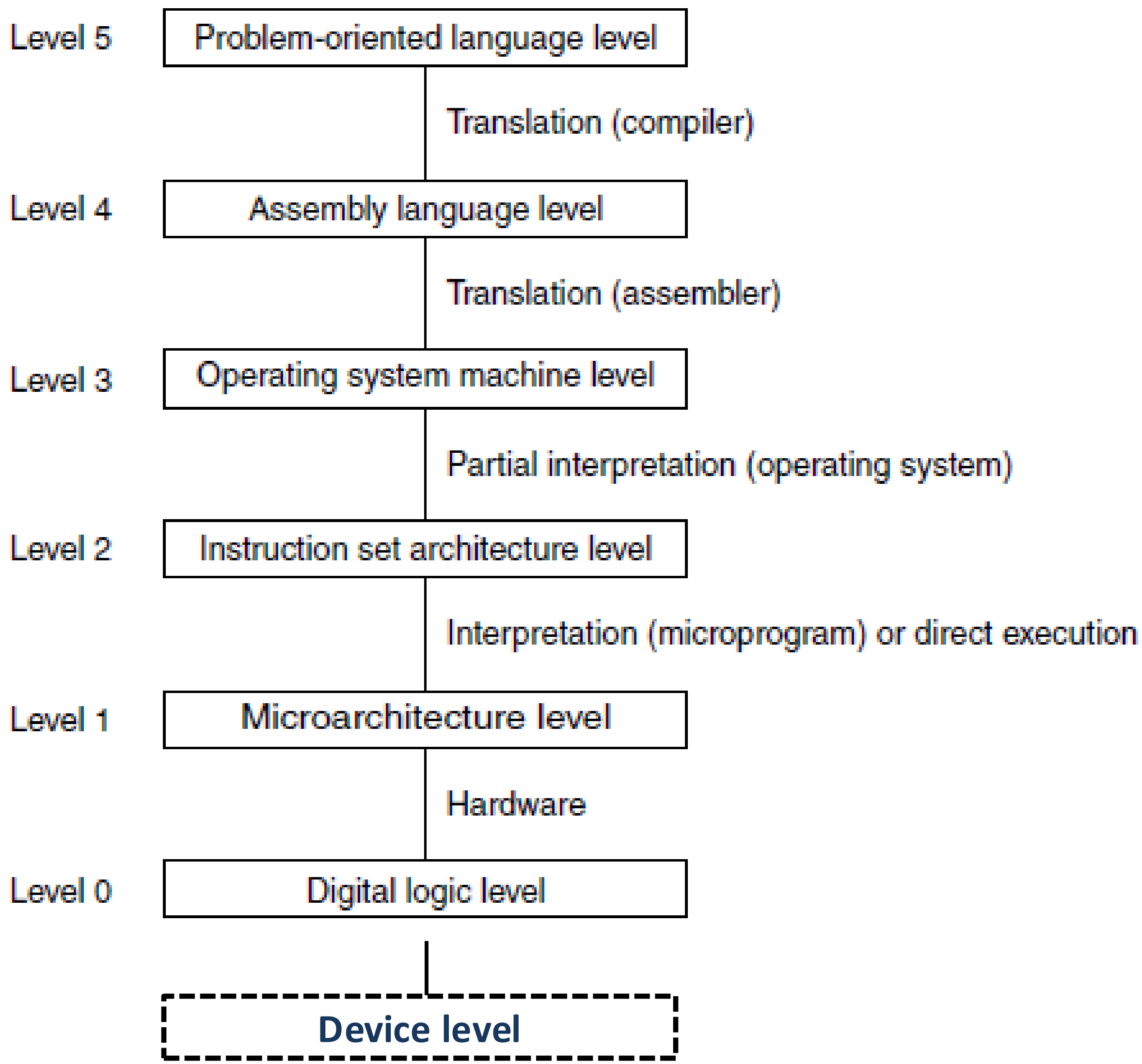
Languages, Levels, and Virtual Machines



Languages, Levels



Current Multilevel Machines



5 - Problem-oriented language level

The level deals with high level programming languages. In this level people usually write programs in languages such as Java, Python, and C++.

From 5 to 4

A translator or interpreter converts source code into assembly code.

4 - Assembly language level

Human readable language, hard to work with, time consuming, and even a simple program contains many lines

From 4 to 3

An assembler will convert the assembly language code into machine language.

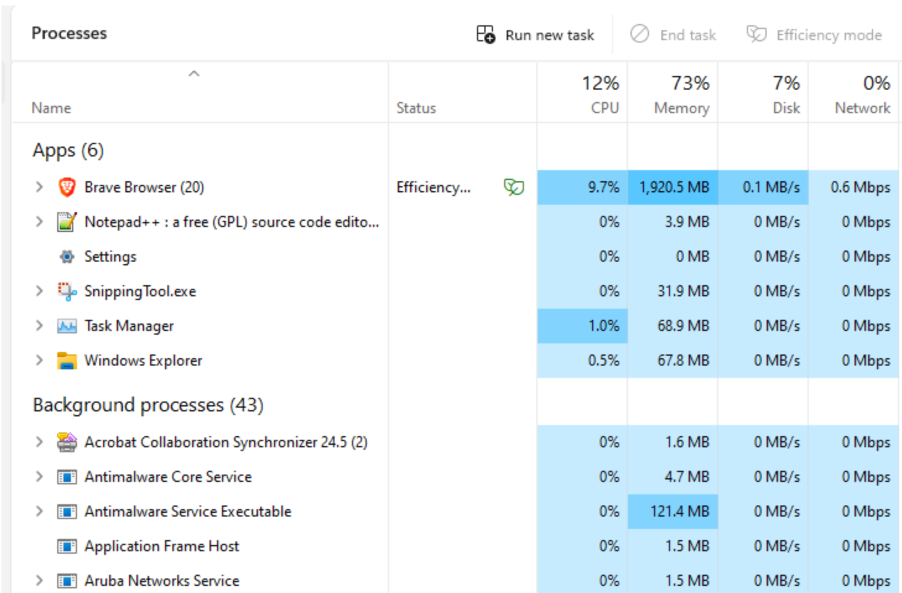
3 - Operating system machine level

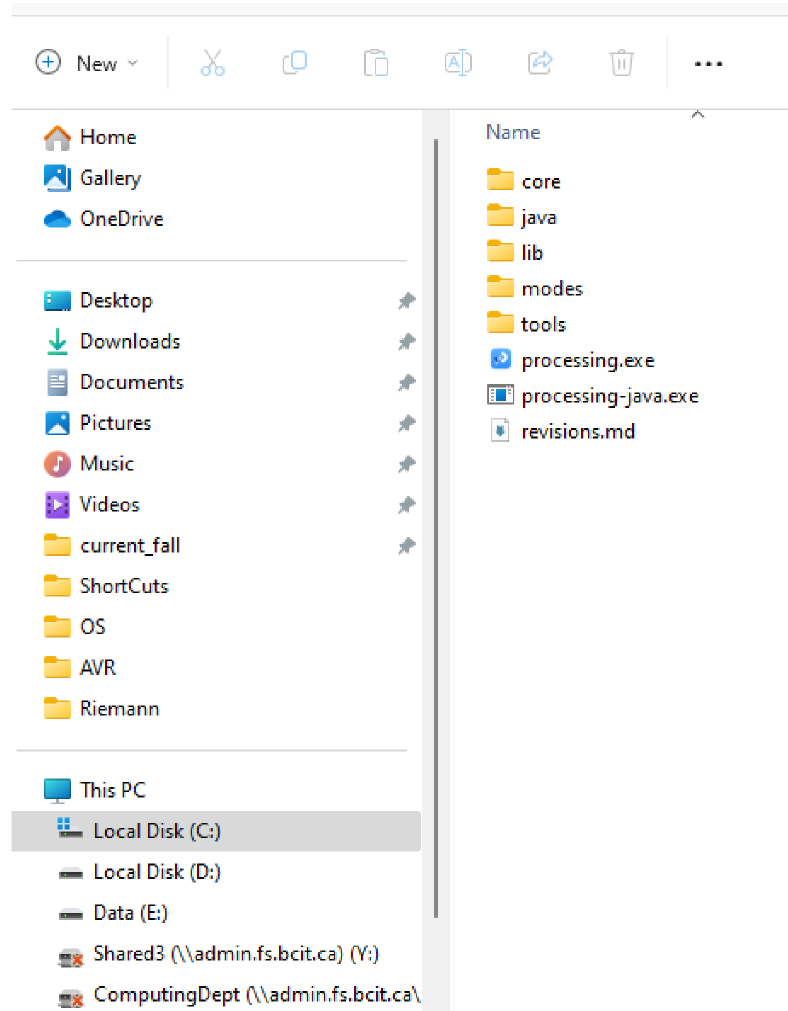
Main duties : Memory management, processes execution, and system resource protection.

From 3 to 2

Most assembly language instructions pass through Level 3 without modification

3 - Operating system machine level





2 - Instruction set architecture level (ISA)

An ISA describes the design of a Computer in terms of the basic operations it must support. The ISA is not concerned with the implementation specific details of a computer. It is only concerned with the set or collection of basic operations the computer must support.

From 2 to 1

<u>All instructions in machine code form are executed in level 1</u>

2 - Instruction set architecture level (ISA)

Instruction Set Summary

ATmega32(L)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF − Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 − Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1

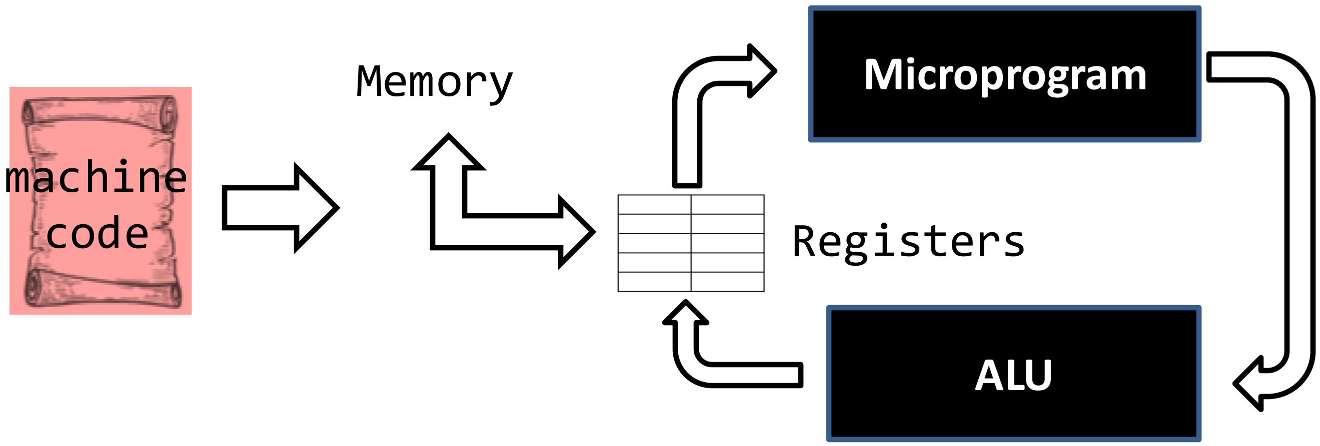
1 - Microarchitecture level

Assume that a program written in a high level language is translated to the low level machine language.

A file is created after compilation containing some instructions in machine codes.

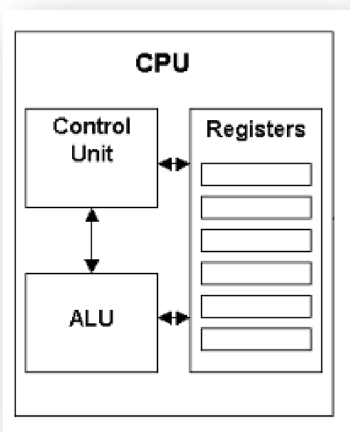
The instructions are executed in this level employing ALU, Registers, and Microprogram.

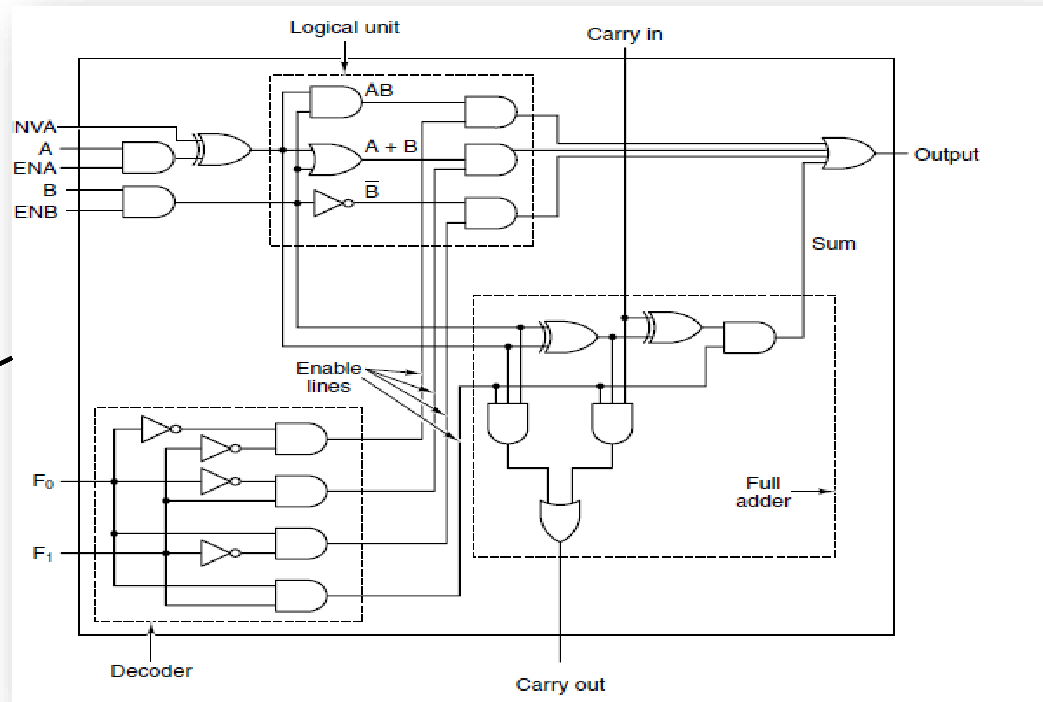
The execution of instructions is controlled by the Microprogram



Digital level

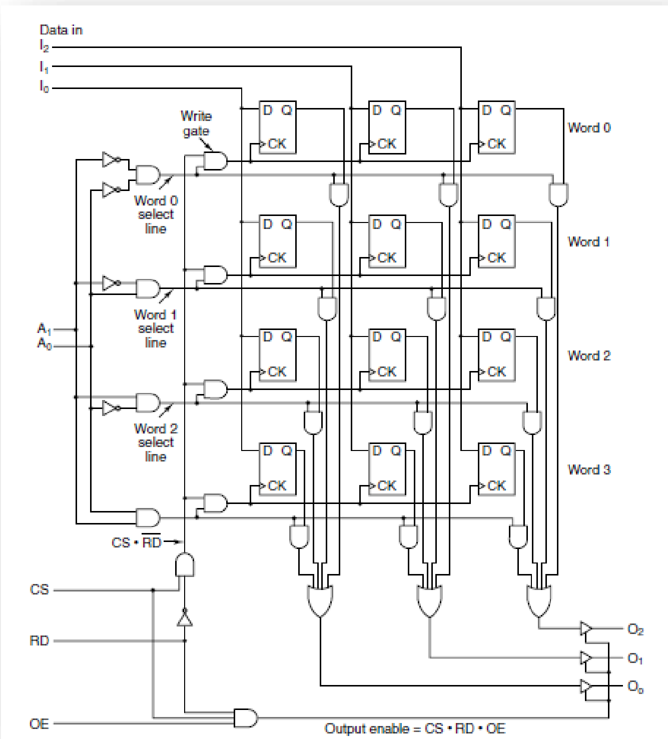
Many components such memory, CPU, and so on are made of basic gates AND, OR, and NOT

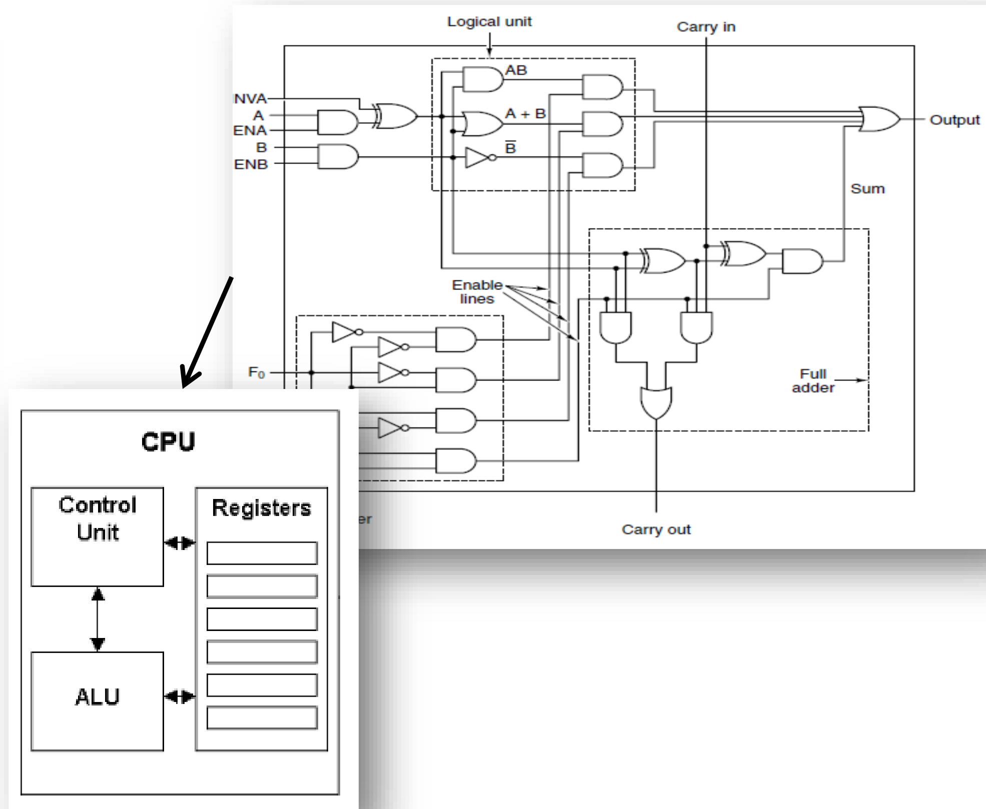




0 - Digital level

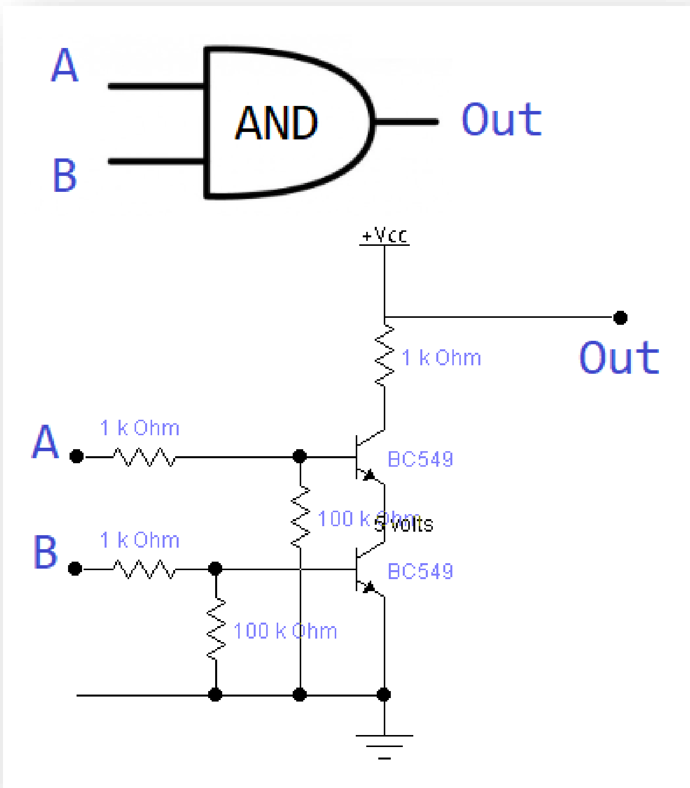
Many components such memory, CPU, and so on are made of basic gates AND, OR, and NOT

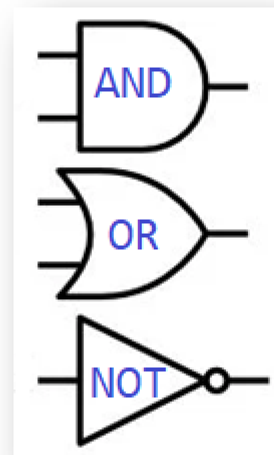




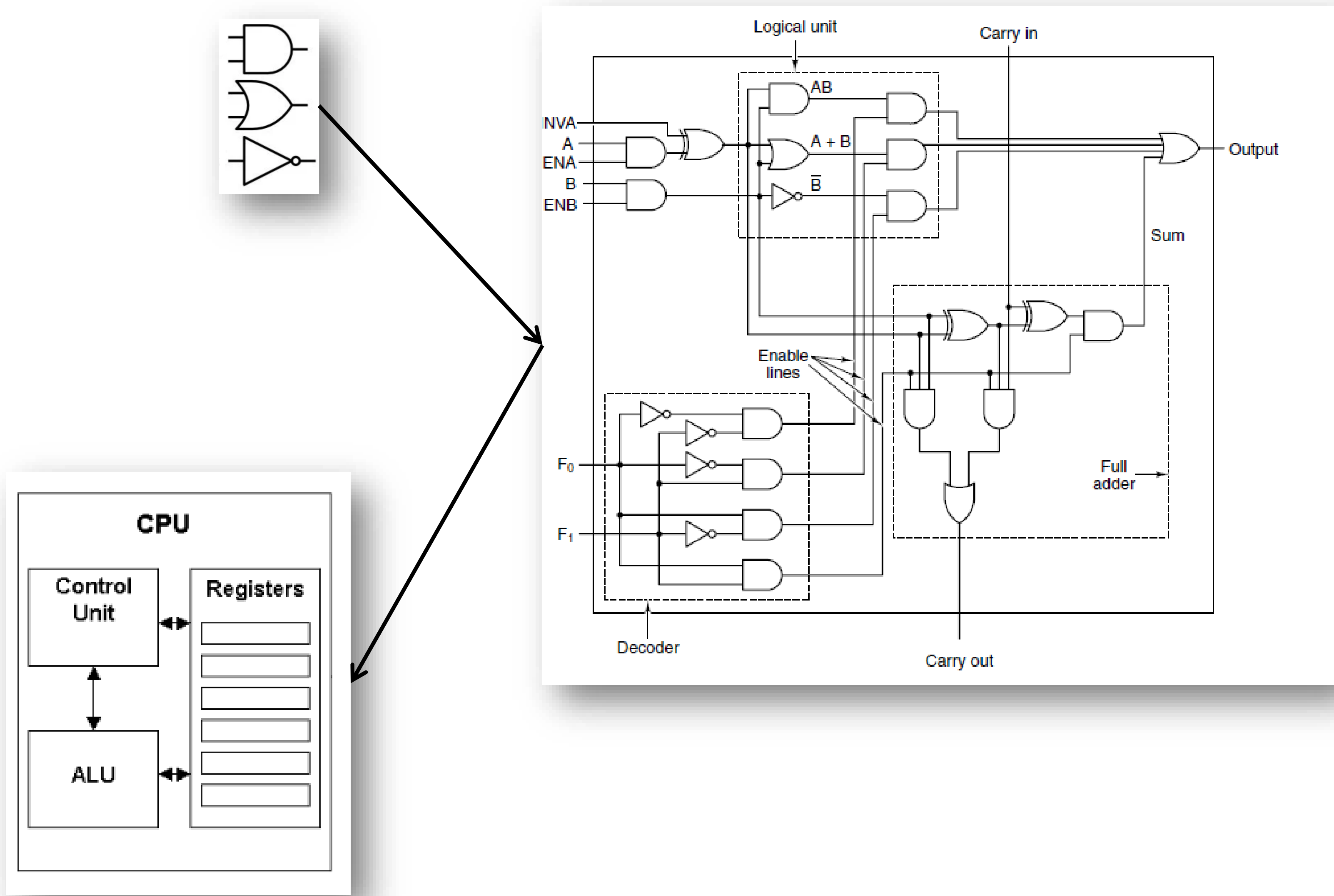
Device level

Device level made of transistors and in turn transistors are made of semiconductors. AND, OR, and NOT are the building blocks of any digital circuits





1 Bit ALU



Logical unit
Carry in
AB
A + B
$\overline{B}$
NVA
A
ENA
B
ENB
Output
Sum
Enable lines
F0
F1
Full adder
Decoder
Carry out
CPU
Control Unit
Registers
ALU

Equivalence of Hardware and Software

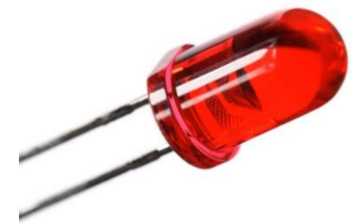
Simply states that anything that can be done with software can also be done with hardware and vice versa (in terms of functionality).

Should consider speed, cost, reliability, frequency tradeoffs. Do not forget hardware is almost always faster and more expensive.

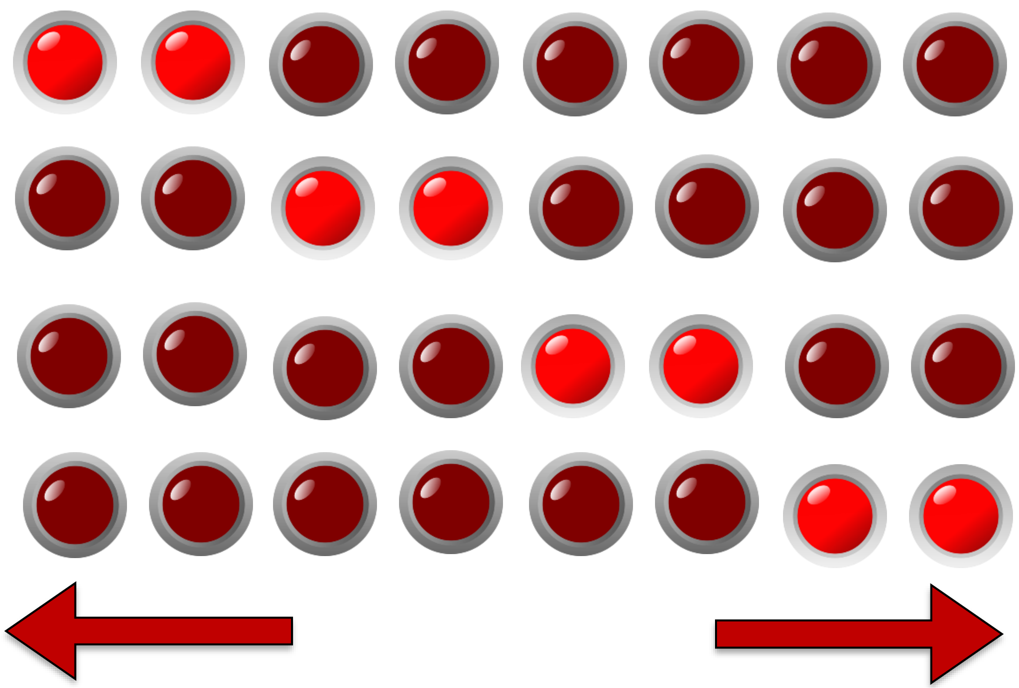
https://www.youtube.com/watch?v=77VG5R4AwT4

Hardwired
vs.
Microprogrammed Approach

Microprogrammed Approach



Array of LEDS

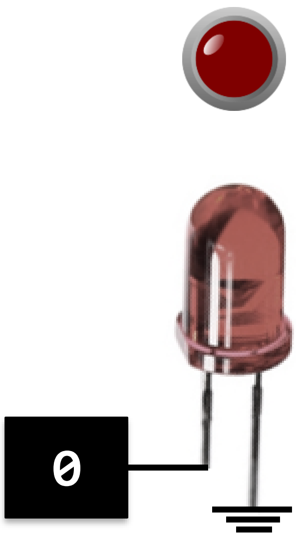


Microprogrammed Approach

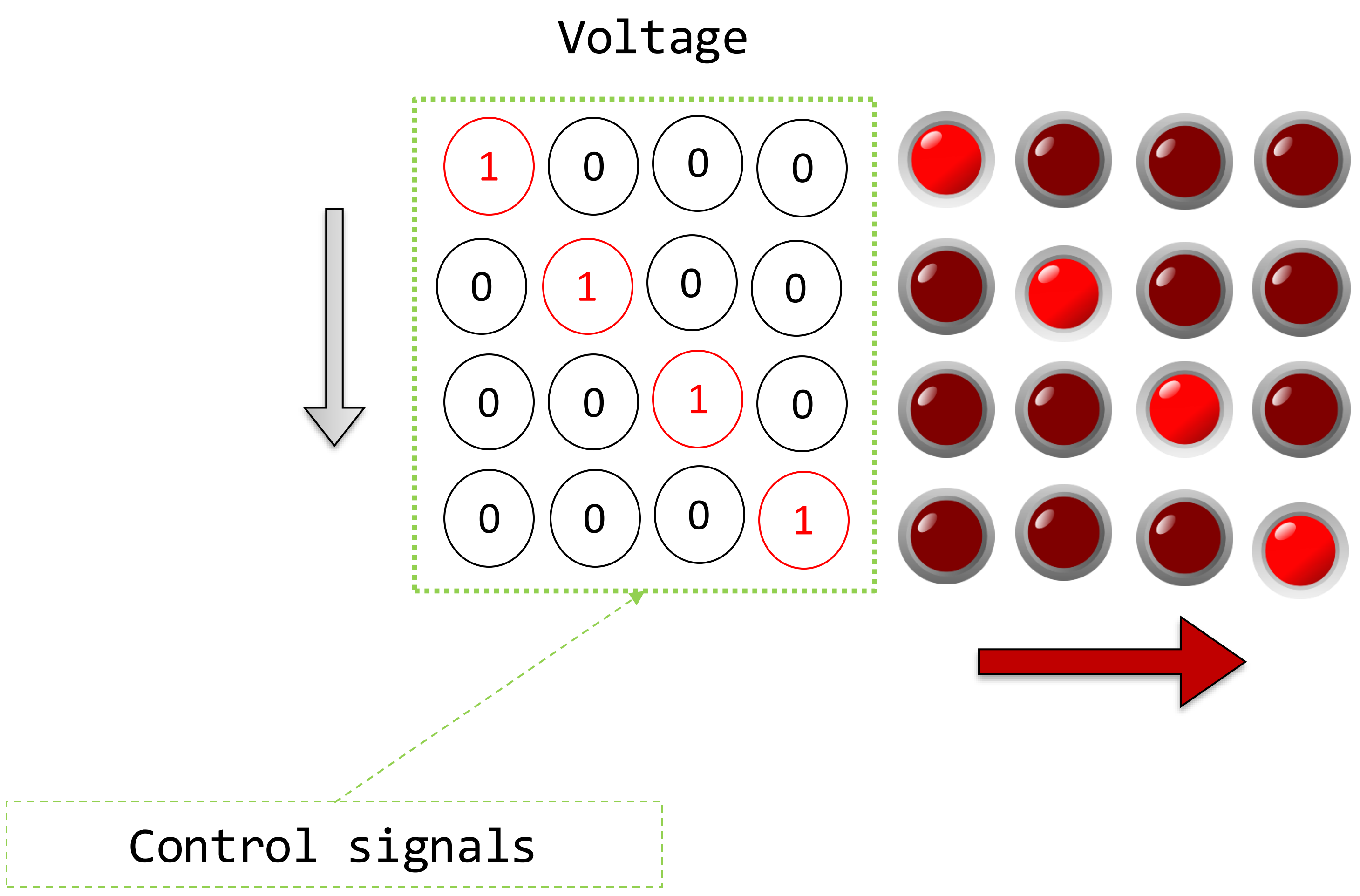
Voltage 1 = on
Voltage 0 = off

Voltage 1
ON

Voltage 0
OFF

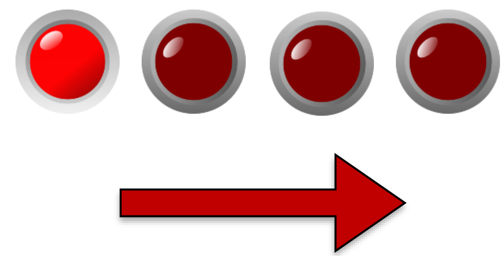


Microprogrammed Approach



Microprogrammed Approach

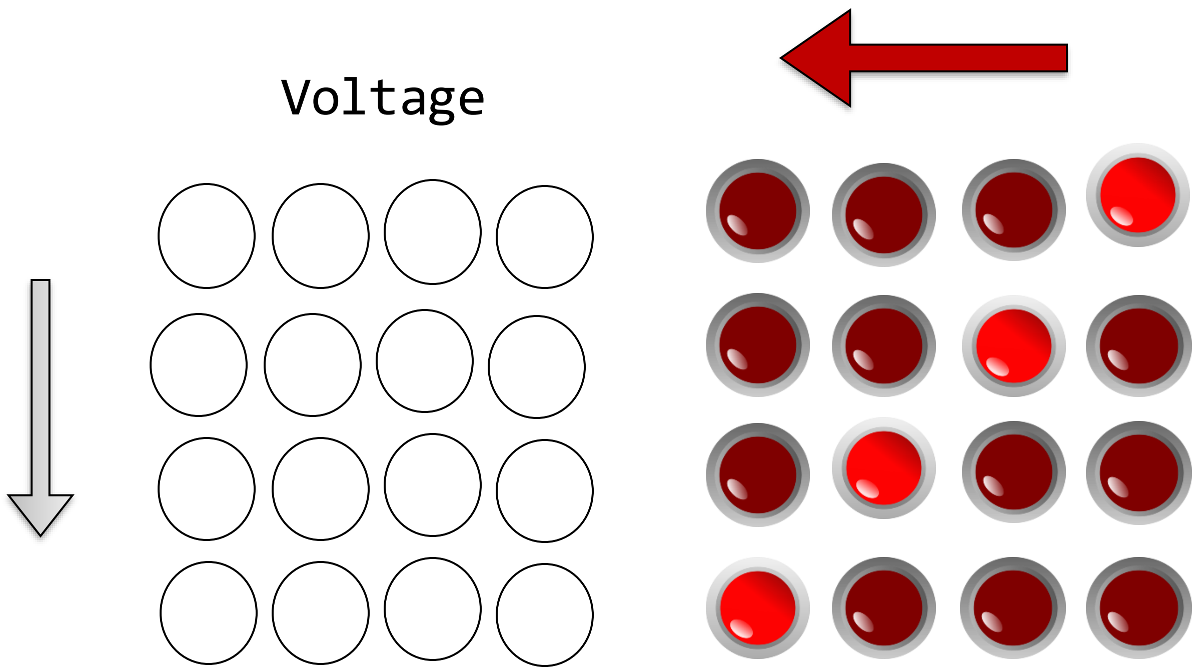
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1



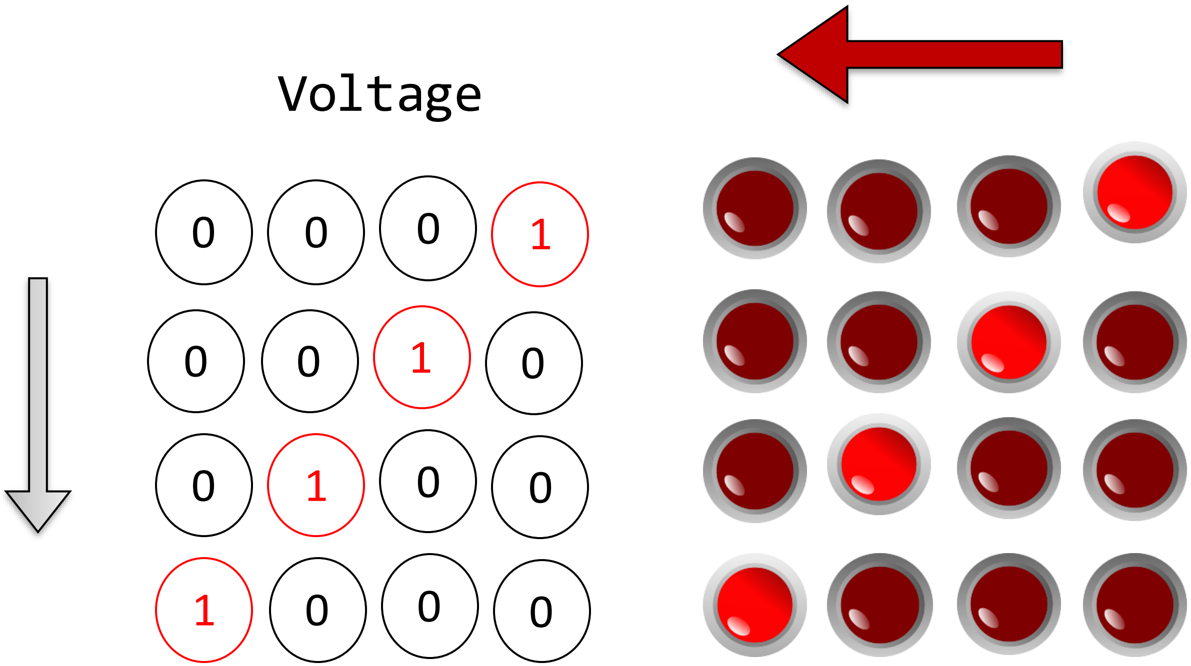
Microprogram (P1)
LED moving to right, stored in ROM

Microprogrammed Approach

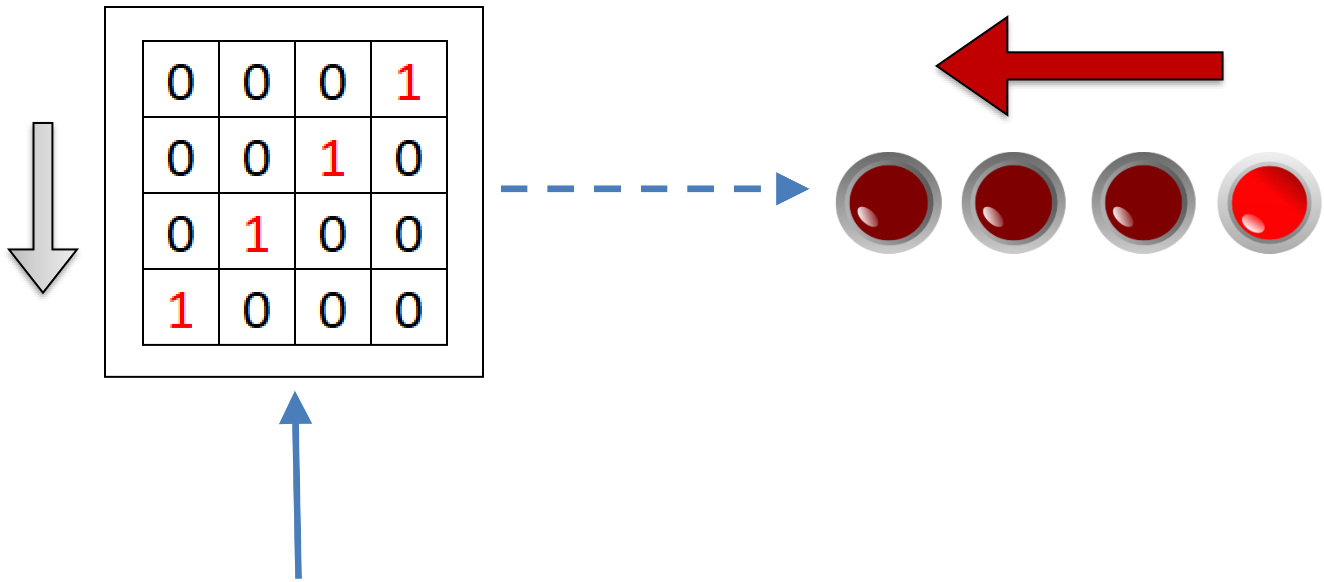
What is the micro program for the following configuration?



Microprogrammed Approach

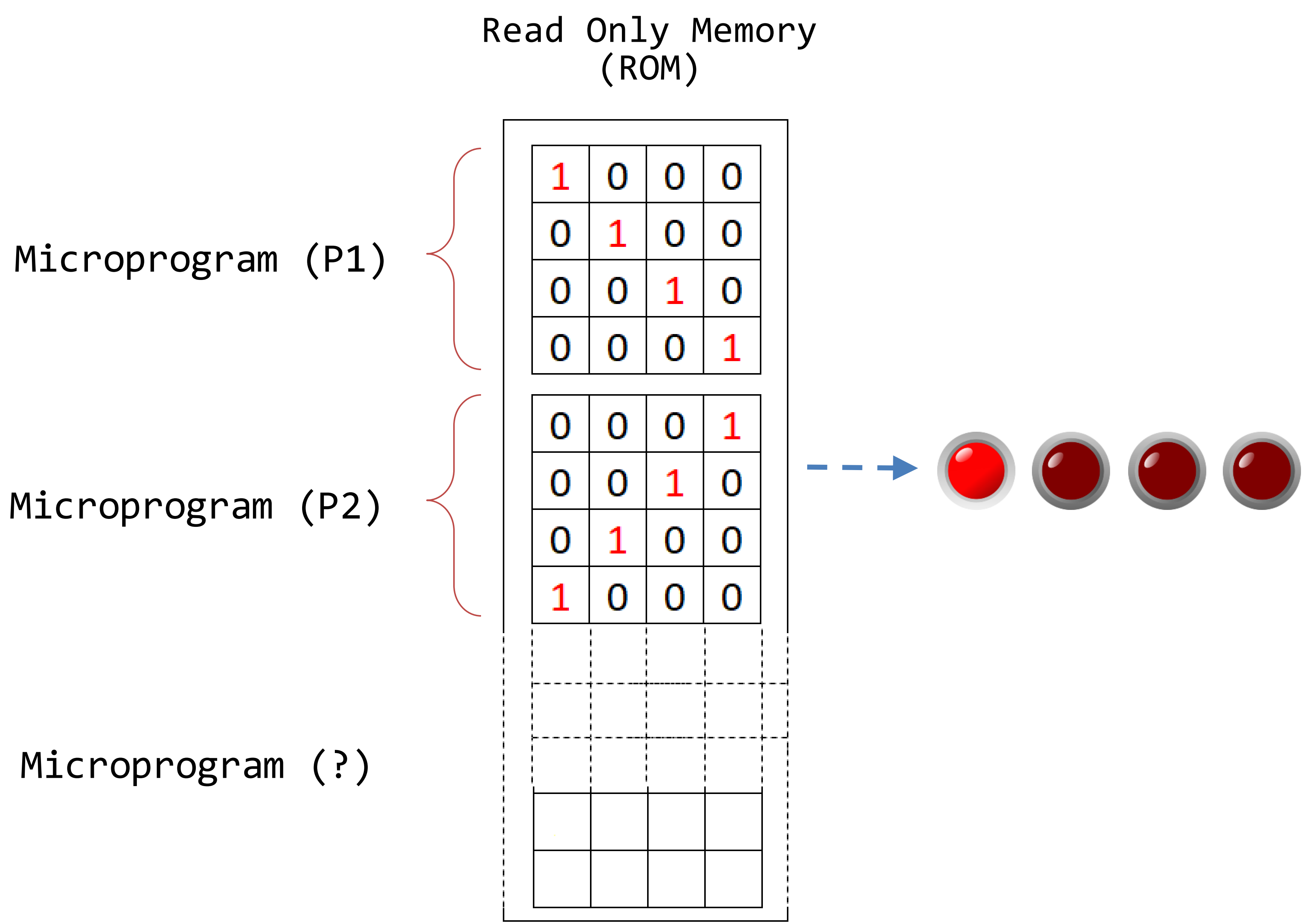


Microprogrammed Approach



Microprogram (P2)
LED moving to left, stored in ROM

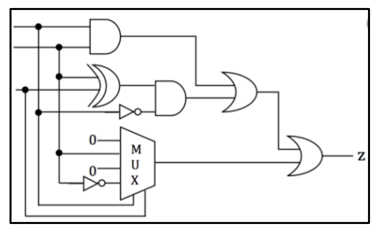
Microprogrammed Approach

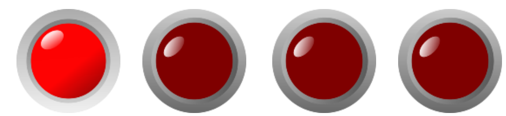


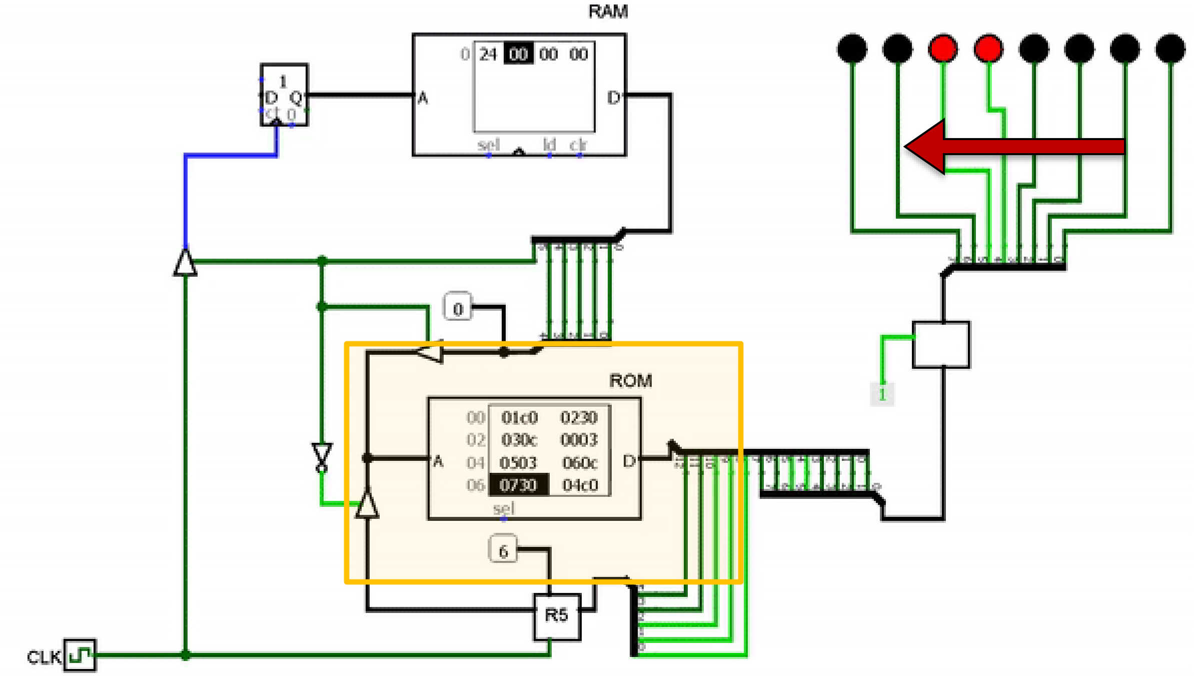
Hardwired Approach

Circuit

P1 or P2

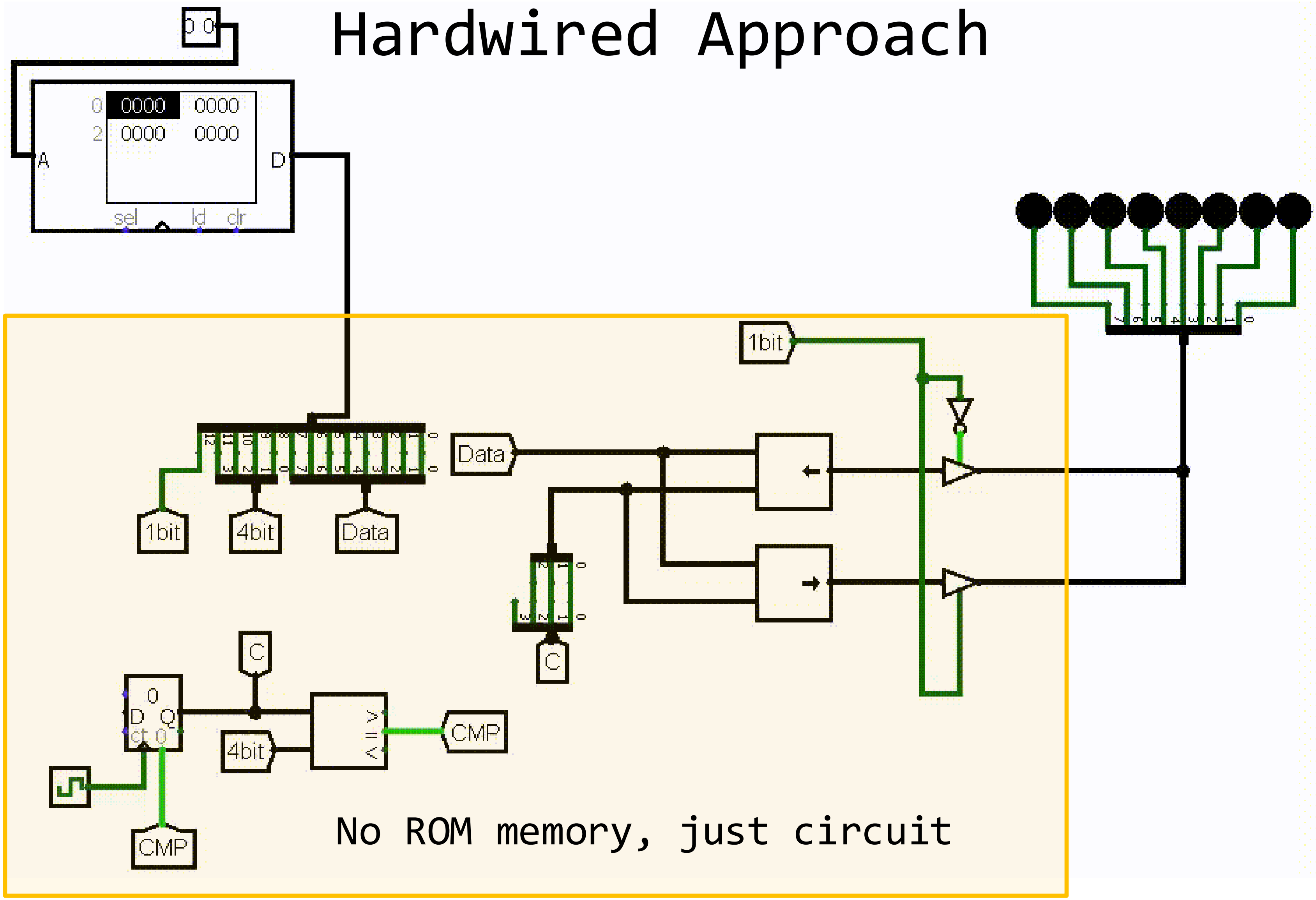






https://vimeo.com/414529620

Hardwired Approach



https://vimeo.com/414529276

Comparison

	Hardwired	**Microprogrammed**
Speed	Fast	Slow
Cost	More	Cheaper
Flexibility	Hard to modify	Flexible
Control memory	Absent	Present

Evolution of Multilevel Machines

- **The Invention of Microprogramming**
  1940s, two levels
  1950-1960s three-level machines were constructed
- **The Invention of the Operating System**
  Early 1960s researchers at Dartmouth College,
  M.I.T., and elsewhere developed operating systems that allowed (multiple) programmers to communicate directly with the computer.
- **The Migration of Functionality to Microcode**
  1970, adding new instructions without using hardware or in other words adding hardware (new machine instructions) by programming

- **The Elimination of Microprogramming**

  - **Microprogramming slows down the computer**: Originally, computers used microprograms to manage complex instructions, but this made the computer slower because each instruction had to be broken down into smaller steps, which took time.
  - **Computer design came full circle**: Early computers directly executed instructions with hardware (without microprogramming). Then, microprogramming was introduced to handle more complex tasks, but it eventually made things too slow. Now, designers are going back to the idea of having the hardware directly execute instructions again.
  - **Eliminating microprogramming**: To speed things up, computer engineers decided to get rid of microprogramming. This means instructions no longer need to be broken down into smaller steps stored in memory.

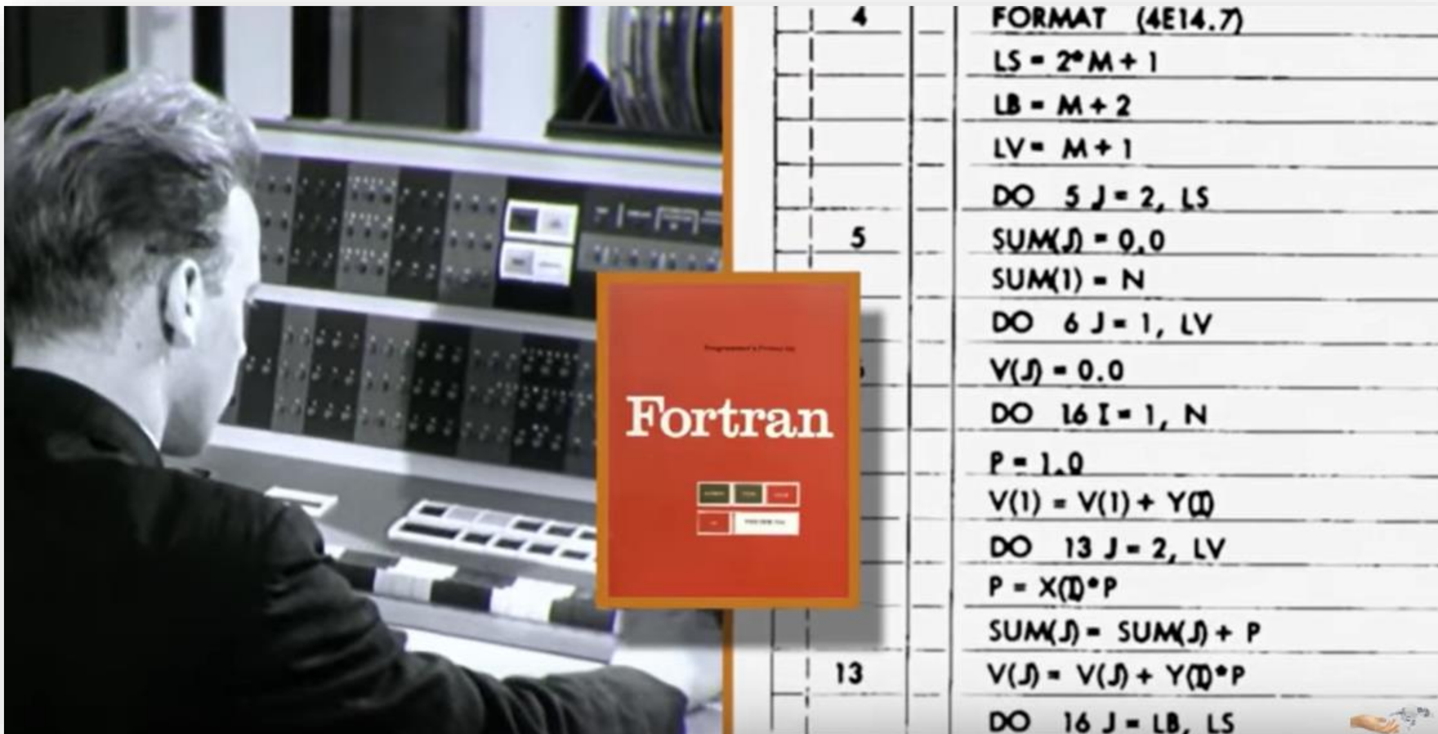
- **The Elimination of Microprogramming**

  - **Simplifying the instruction set**: To make things more efficient, the number of instructions the computer can understand is reduced. This makes the hardware simpler and faster.
  - **Direct execution by hardware**: The remaining, smaller set of instructions is executed directly by hardware, without needing extra layers like microprogramming. This makes the computer faster and more efficient.

MILESTONES IN COMPUTER ARCHITECTURE

The History of Computing

https://www.youtube.com/watch?v=-M6lANfzFsM



Computer Generations

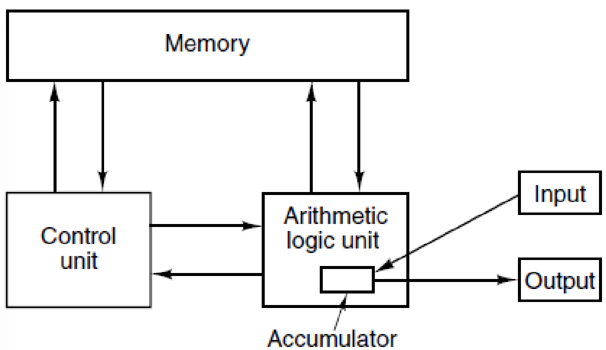
- **Zeroth Generation -** Mechanical Computers (1642 – 1945)

 A tax collector for the French government designed by Pascal (addition and subtraction operations)

Computer Generations

- **First Generation -** Vacuum Tubes (1945 – 1955)
  Electronic Numerical Integrator And Computer(**ENIAC**)
  - 18,000 vacuum tubes and 1500 relays
  - 30 tons
  - consumed 140 kilowatts of power
  - 20 registers, each capable of holding a 10-digit decimal number

  - **Von-Neumann** proposed his computer architecture design in 1945
    - Both data and program stored in memory
    - Still the basis for nearly all digital computers



The original von Neumann machine.

Computer Generations

In the old days, a TV was expected to take 20 to 30 minutes to warm up

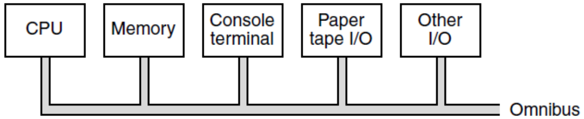
Computer Generations

Computer Generations

- **Second Generation -** Transistors (1955 – 1965)

  e.g., **IBM 7094**

  The **PDP-8** had a major innovation: a single bus, the omnibus,



The PDP-8 omnibus.

- **Third Generation -** Integrated Circuits (IC) (1965 – 1980)

  Dozens of transistors on a single chip. This packaging made it possible to build computers that were smaller, faster, and cheaper than their transistorized predecessors

e.g., **IBM 360**

Property	Model 30	Model 40	Model 50	Model 65
Relative performance	1	3.5	10	21
Cycle time (in billionths of a sec)	1000	625	500	250
Maximum memory (bytes)	65,536	262,144	262,144	524,288
Bytes fetched per cycle	1	2	4	16
Maximum number of data channels	3	3	4	6

The initial offering of the IBM 360 product line.

Computer Generations

- **Fourth Generation** – Very Large Scale Integration (VLSI) (1980 – ?)

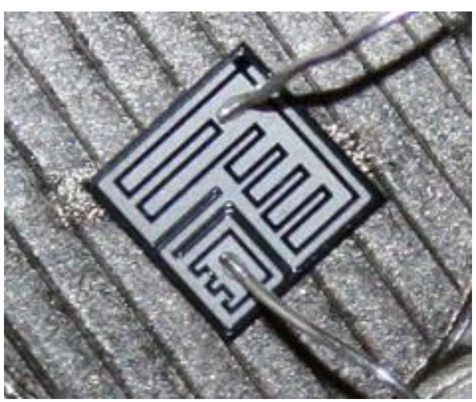
  e.g., the Intel CPUs

  Possible to put first tens of thousands, then hundreds of thousands, and finally millions of transistors on a single chip.

Metal Oxide Semiconductor Field Effect Transistor (MOSFET)

MOSFET

NPN Transistor 2N2222



MOSFET scaling
(process nodes)

10 µm – 1971
6 µm – 1974
3 µm – 1977
1.5 µm – 1981
1 µm – 1984
800 nm – 1987
600 nm – 1990
350 nm – 1993
250 nm – 1996
180 nm – 1999
130 nm – 2001
90 nm – 2003
65 nm – 2005
45 nm – 2007
32 nm – 2009
22 nm – 2012
14 nm – 2014
10 nm – 2016
7 nm – 2018
5 nm – 2019

Future

3 nm – *~2021*
2 nm – *~2024*

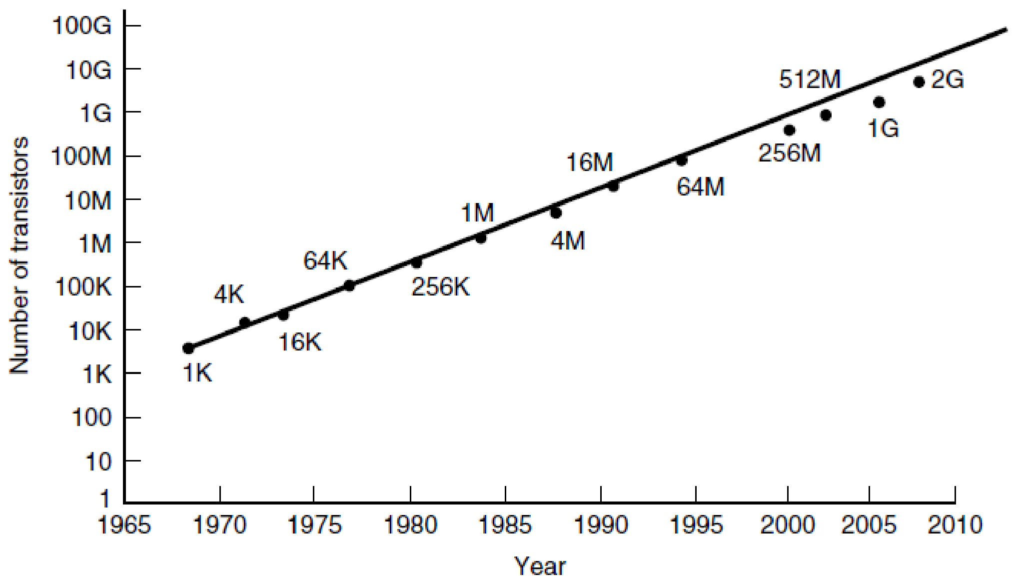
Some milestones in the development of the modern digital computer

Year	Name	Made by	Comments
1834	Analytical Engine	Babbage	First attempt to build a digital computer
1936	Z1	Zuse	First working relay calculating machine
1943	COLOSSUS	British gov't	First electronic computer
1944	Mark I	Aiken	First American general-purpose computer
1946	ENIAC	Eckert/Mauchley	Modern computer history starts here
1949	EDSAC	Wilkes	First stored-program computer
1951	Whirlwind I	M.I.T.	First real-time computer
1952	IAS	Von Neumann	Most current machines use this design
1960	PDP-1	DEC	First minicomputer (50 sold)
1961	1401	IBM	Enormously popular small business machine
1962	7094	IBM	Dominated scientific computing in the early 1960s
1963	B5000	Burroughs	First machine designed for a high-level language
1964	360	IBM	First product line designed as a family
1964	6600	CDC	First scientific supercomputer
1965	PDP-8	DEC	First mass-market minicomputer (50,000 sold)
1970	PDP-11	DEC	Dominated minicomputers in the 1970s
1974	8080	Intel	First general-purpose 8-bit computer on a chip
1974	CRAY-1	Cray	First vector supercomputer
1978	VAX	DEC	First 32-bit superminicomputer
1981	IBM PC	IBM	Started the modern personal computer era
1981	Osborne-1	Osborne	First portable computer
1983	Lisa	Apple	First personal computer with a GUI
1985	386	Intel	First 32-bit ancestor of the Pentium line
1985	MIPS	MIPS	First commercial RISC machine
1985	XC2064	Xilinx	First field-programmable gate array (FPGA)
1987	SPARC	Sun	First SPARC-based RISC workstation
1989	GridPad	Grid Systems	First commercial tablet computer
1990	RS6000	IBM	First superscalar machine
1992	Alpha	DEC	First 64-bit personal computer
1992	Simon	IBM	First smartphone
1993	Newton	Apple	First palmtop computer (PDA)
2001	POWER4	IBM	First dual-core chip multiprocessor

Technological and Economic Forces

Moore’s Law

- **Moore Law**
- – Gordon Moore was co-founder and former chairman of Intel
- – An observation about improvements in hardware
- – Predicts a 60-percent annual increase in the number of transistors that can be put on a chip



Moore’s law predicts a 60 percent annual increase in the number of transistors that can be put on a chip. The data points given above and below the line are memory sizes, in bits.

A. Advances in technology (transistors/chip) lead to better products and lower prices

B. Lower prices lead to new applications

C. New applications lead to new markets and new companies springing up to take advantage of them

D. The existence of all these companies leads to competition, which in turn creates economic demand for better technologies with which to beat the others

Raspberry Pi Zero: the $5 computer

Moore's Law

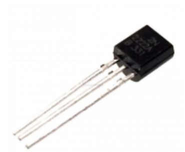
https://www.youtube.com/watch?v=aWLBmapcJRU

The Transistor: a 1953 documentary, anticipating its coming impact on technology

https://www.youtube.com/watch?v=V9xUQWo4vN0

Metal Oxide Semiconductor Field Effect Transistor (MOSFET)

MOSFET



NPN Transistor 2N2222

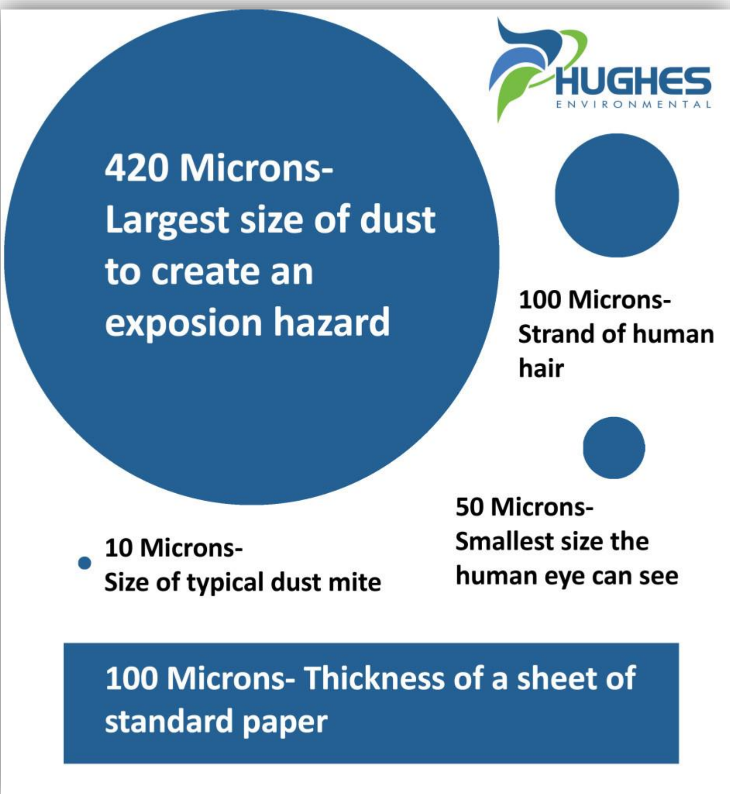
Intel 80486	1,180,235	1989	Intel	1 µm	173 mm²
R4000	1,350,000	1991	MIPS	1.0 µm	213 mm²
Pentium	3,100,000	1993	Intel	0.8 µm	294 mm²
AMD K5	4,300,000	1996	AMD	0.5 µm	251 mm²
Pentium Pro	5,500,000[4]	1995	Intel	0.5 µm	307 mm²
Pentium II	7,500,000	1997	Intel	0.35 µm	195 mm²
AMD K6	8,800,000	1997	AMD	0.35 µm	162 mm²
Pentium III	9,500,000	1999	Intel	0.25 µm	128 mm²

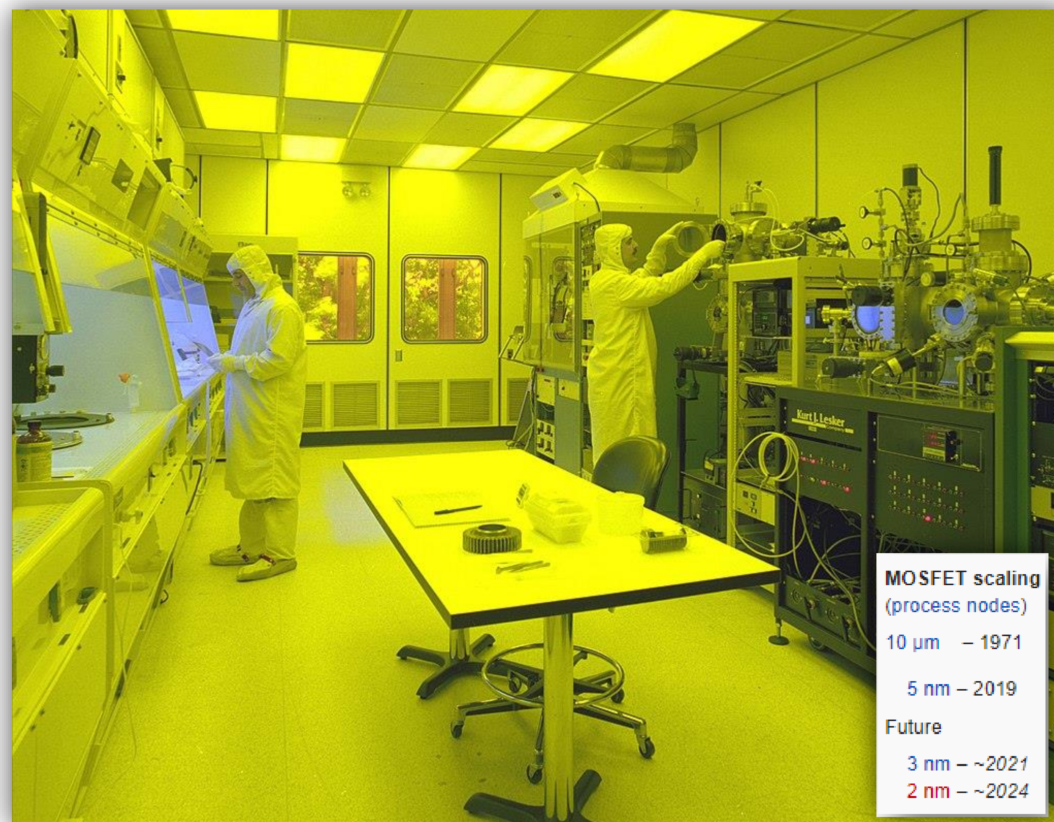
MOSFET scaling
(process nodes)

10 µm	– 1971
6 µm	– 1974
3 µm	– 1977
1.5 µm	– 1981
1 µm	– 1984
800 nm	– 1987
600 nm	– 1990
350 nm	– 1993
250 nm	– 1996
180 nm	– 1999
130 nm	– 2001
90 nm	– 2003
65 nm	– 2005
45 nm	– 2007
32 nm	– 2009
22 nm	– 2012
14 nm	– 2014
10 nm	– 2016
7 nm	– 2018
5 nm	– 2019

Future

3 nm	– ~2021
2 nm	– ~2024





Cleanroom used for the production of **microsystems**. The yellow (red-green) lighting is necessary for photolithography, to prevent unwanted exposure of photoresist to light of shorter wavelengths.

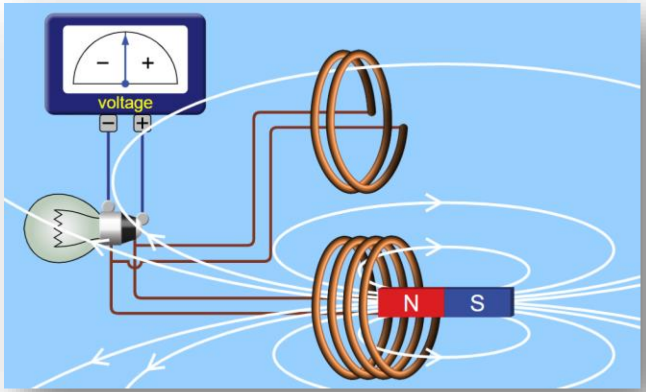
THE COMPUTER ZOO

The Computer Spectrum

Type	Price ($)	Example application
Disposable computer	0.5	Greeting cards
Microcontroller	5	Watches, cars, appliances
Mobile and game computers	50	Home video games and smartphones
Personal computer	500	Desktop or notebook computer
Server	5K	Network server
Mainframe	5M	Batch data processing in a bank

The current spectrum of computers available. The prices should be taken with a grain (or better yet, a metric ton) of salt.

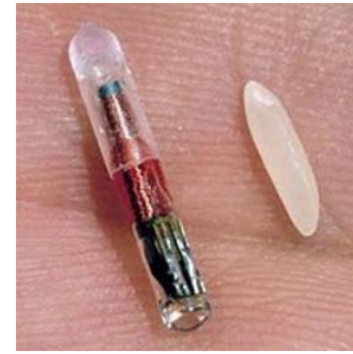
Disposable Computers :
RFID (Radio Frequency IDentification)



https://phet.colorado.edu/sims/html/faradays-law/latest/faradays-law_en.html

A. **RFID** contains a tiny radio transponder and a built-in unique 128-bit number

B. They are powered by the incoming radio signal to transmit Their number back to the antenna



What is RFID?

https://www.youtube.com/watch?v=gEQJxNDSKAE

Transponder : electrical device designed to receive a specific signal and automatically transmit a specific reply

Microcontrollers:

The embedded computers, sometimes called **microcontrollers,** manage the devices and handle the user interface.

1. Appliances (clock radio, washer, dryer, microwave, burglar alarm)
2. Communications gear (cordless phone, cell phone, fax, pager)
3. Computer peripherals (printer, scanner, modem, CD ROM-drive)
4. Entertainment devices (VCR, DVD, stereo, MP3 player, set-top box)
5. Imaging devices (TV, digital camera, camcorder, lens, photocopier)
6. Medical devices (X-ray, MRI, heart monitor, digital thermometer)
7. Military weapon systems (cruise missile, ICBM, torpedo)
8. Shopping devices (vending machine, ATM, cash register)
9. Toys (talking doll, game console, radio-controlled car or boat)

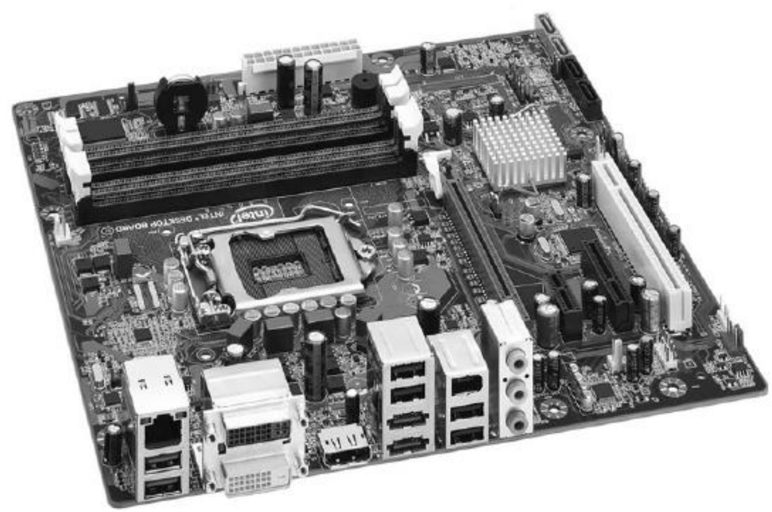
ICBM = Intercontinental Ballistic Missile
ATM = Automated Teller Machine

Mobile and Game Computers:

A. Sony PlayStation 3 - 3.2-GHz multicore proprietary CPU (called the Cell microprocessor), which is based on the IBM PowerPC RISC CPU, and seven 128-bit Synergistic Processing Elements (SPEs).The PlayStation 3 also contains 512 MB of RAM, a 550-MHz custom Nvidia graphics chip, and a Blu-ray player.
B. the Microsoft Xbox 360. It contains a 3.2-GHz IBM triple-core PowerPC CPU with 512 MB of RAM, a 500-MHz custom ATI graphics chip, a DVD player, and a hard disk.
C. Samsung Galaxy Tablet. It contains two 1-GHz ARM cores plus a graphics processing unit (integrated into the Nvidia Tegra 2 system-on-a-chip), 1 GB of RAM, dual cameras, a 3-axis gyroscope, and flash memory storage.

Personal Computers
CPU, a few gigabytes of memory, a hard disk holding up to terabytes of data, a CD-ROM/DVD/Blu-ray drive, sound card, network interface, high-resolution monitor, and other peripherals.

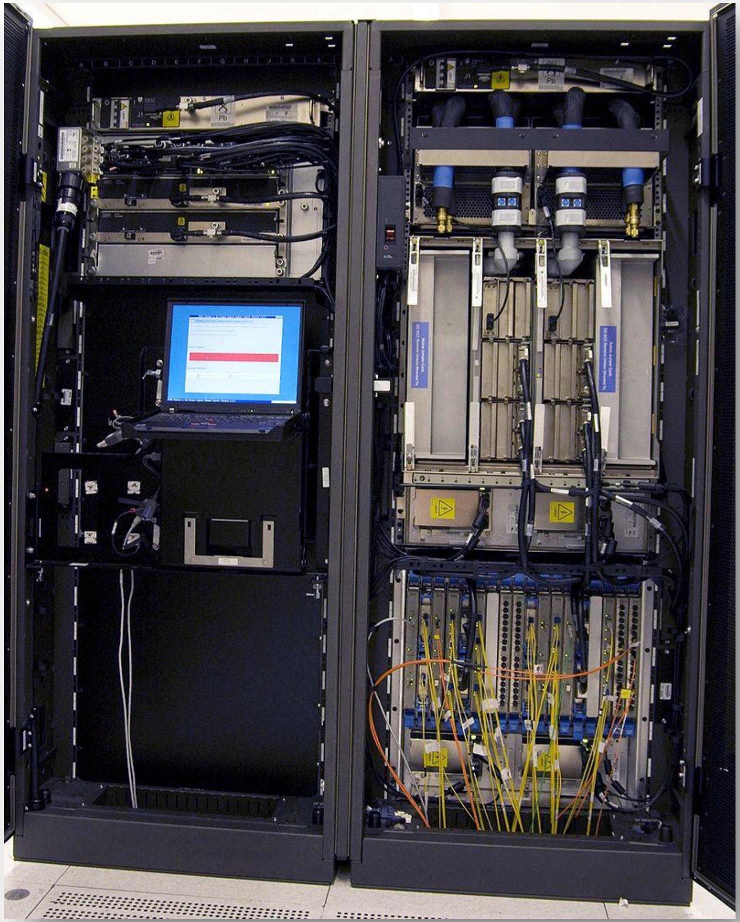
Servers
Beefed-up personal computers or workstations are often used as network servers, both for local area networks (typically within a single company), and for the Internet.



A printed circuit board is at the heart of every personal computer. This one is the Intel DQ67SW board. © 2011 Intel Corporation. Used by permission.

Mainframes

For the most part, they are not much faster than powerful servers, but they always have more I/O capacity and are often equipped with vast disk farms, often holding thousands of gigabytes of data.



Inside an IBM System z9 mainframe

Supercomputers, They had enormously fast CPUs, many gigabytes of main memory, and very fast disks and networks. They were used for massive scientific and engineering calculations such as simulating colliding galaxies, synthesizing new medicines, or modeling the flow of air around an airplane wing

The IBM Blue Gene/P supercomputer "Intrepid" at Argonne National Laboratory runs **164,000 processor cores** using normal data center air conditioning, grouped in **40 racks/cabinets** connected by a high-speed 3-D torus network.

EXAMPLE COMPUTER FAMILIES

a) x86 Architecture
b) ARM Architecture
c) AVR Architecture

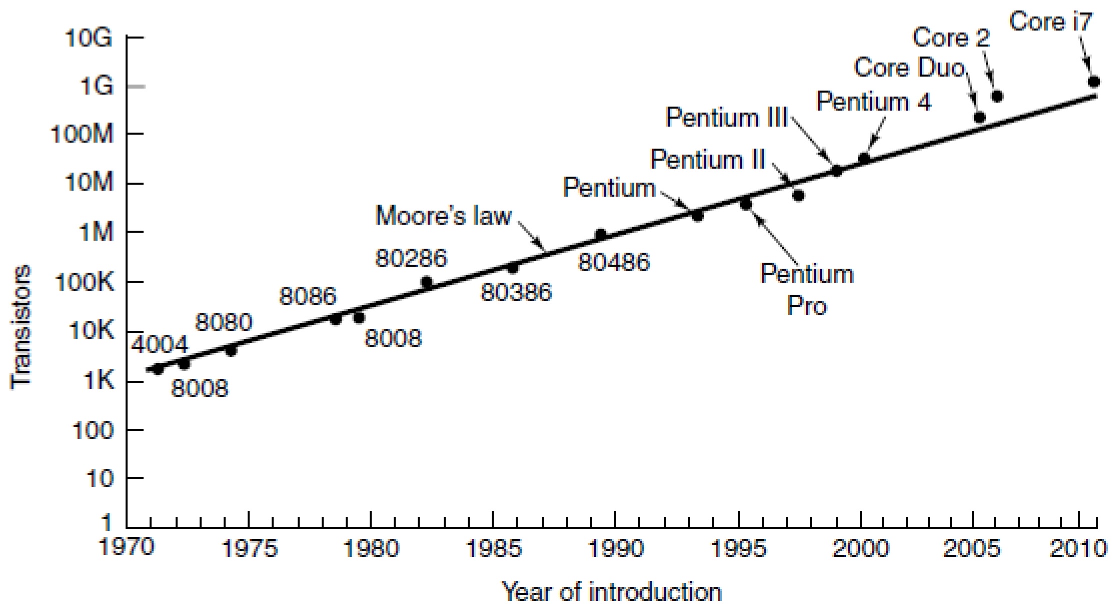
x86 Architecture

(used in personal computers and servers)

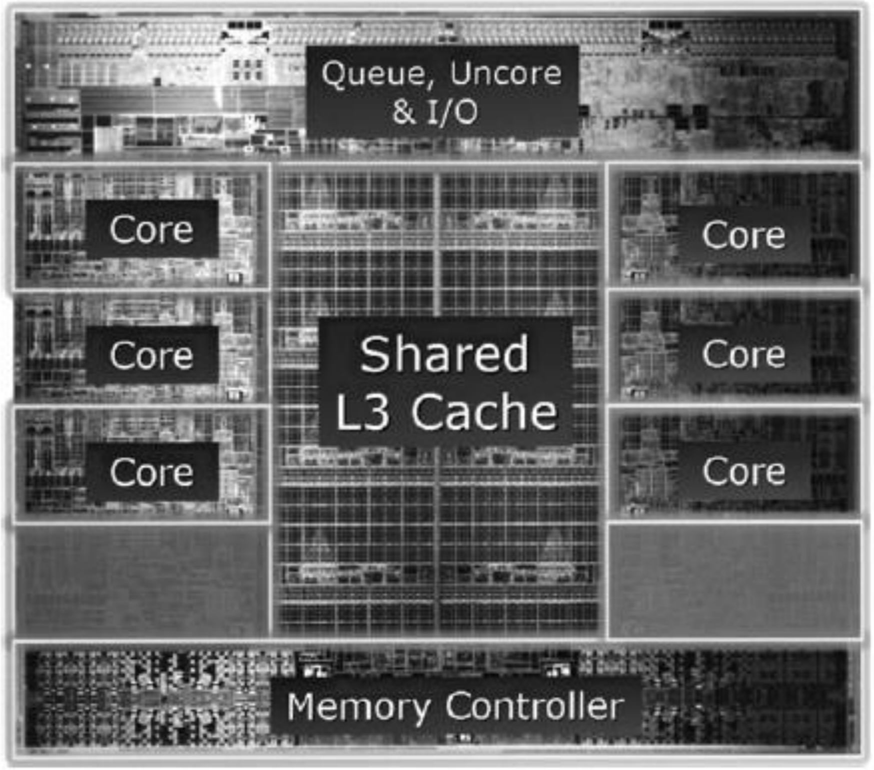
In the late 1960s, calculators were large electromechanical machines the size of a modern laser printer and weighing 20 kg. In Sept. 1969, a Japanese company, **Busicom**, approached Intel with a request that it manufacture 12 custom chips for a proposed electronic calculator. in 1970, the first single-chip CPU (4-bit), the **2300**-transistor **4004**, was born.

Chip	Date	MHz	Trans.	Memory	Notes
4004	4/1971	0.108	2300	640	First microprocessor on a chip
8008	4/1972	0.108	3500	16 KB	First 8-bit microprocessor
8080	4/1974	2	6000	64 KB	First general-purpose CPU on a chip
8086	6/1978	5–10	29,000	1 MB	First 16-bit CPU on a chip
8088	6/1979	5–8	29,000	1 MB	Used in IBM PC
80286	2/1982	8–12	134,000	16 MB	Memory protection present
80386	10/1985	16–33	275,000	4 GB	First 32-bit CPU
80486	4/1989	25–100	1.2M	4 GB	Built-in 8-KB cache memory
Pentium	3/1993	60–233	3.1M	4 GB	Two pipelines; later models had MMX
Pentium Pro	3/1995	150–200	5.5M	4 GB	Two levels of cache built in
Pentium II	5/1997	233–450	7.5M	4 GB	Pentium Pro plus MMX instructions
Pentium III	2/1999	650–1400	9.5M	4 GB	SSE Instructions for 3D graphics
Pentium 4	11/2000	1300–3800	42M	4 GB	Hyperthreading; more SSE instructions
Core Duo	1/2006	1600–3200	152M	2 GB	Dual cores on a single die
Core	7/2006	1200–3200	410M	64 GB	64-bit quad core architecture
Core i7	1/2011	1100–3300	1160M	24 GB	Integrated graphics processor

Key members of the Intel CPU family. Clock speeds are measured in MHz (megahertz), where 1 MHz is 1 million cycles/sec.



Moore's law for (Intel) CPU chips.



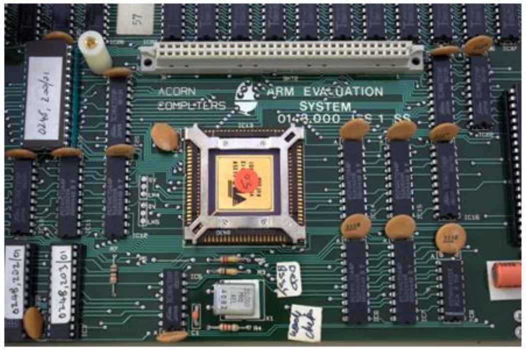
The Intel Core i7-3960X die. The die is 21 by 21 mm and has 2.27 billion transistors. © 2011 Intel Corporation. Used by permission.

ARM Architecture

The main purpose behind designing ARM CPUs was to create efficient, low-power processors that could deliver high performance.

Acorn Computer (U.K.-based company)
The first ARM architecture (called the ARM2) appeared in the Acorn Archimedes personal computer. The Archimedes was a very fast and inexpensive machine for its day, running up to 2 MIPS (millions of instructions per second).

To better focus on the project, the ARM architecture team left Acorn to create a new company called Advanced RISC Machines (ARM). Their new processor was called the ARM 610, which powered the Apple Newton when it was released in 1993.



Nvidia’s Tegra 2 system-on-a-chip
the design contains three ARM processors: two 1.2-GHz ARM Cortex-A9 cores plus an ARM7 core.

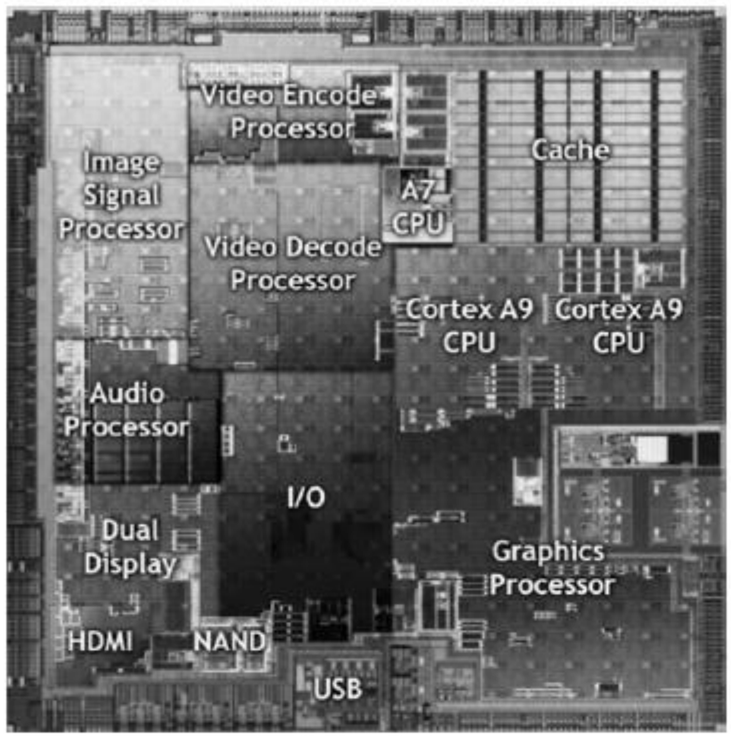


Figure 1-14. The Nvidia Tegra 2 system on a chip. © 2011 Nvidia Corporation. Used by permission.

Tablets, multimedia players, smartphones and other mobile devices, such as wearables.

Raspberry Pi

AVR Architecture very low-end embedded systems

The AVR story starts in 1996 at the Norwegian Institute of Technology.
Alf-Egil Bogen and **V**egard Wollan designed an 8-bit **R**ISC CPU called the **AVR**.

Chip	Flash	EEPROM	RAM	Pins	Features
tinyAVR	0.5–16 KB	0–512 B	32–512 B	6–32	Tiny, digital I/O, analog input
megaAVR	8–256 KB	0.5–4 KB	0.25–8 KB	28-100	Many peripherals, analog out
AVR XMEGA	16–256 KB	1–4 KB	2–16 KB	44–100	Crypto acceleration, USB I/O

Microcontroller classes in the AVR family.

EEPROM = Electrically Erasable Programmable Read-Only Memory

Microcontrollers typically have three types of memory on board:

Flash memory is programmable using an external interface and high voltages, and this is where program code and data are stored. Flash RAM is nonvolatile, so even if the system is powered down, the flash memory will remember what was written to it.

EEPROM is also nonvolatile, but unlike flash RAM, it can be changed by the program while it is running. This is the storage in which an embedded system would keep user configuration information.

RAM is where program variables will be stored as the program runs. This memory is volatile, so any value stored here will be lost once the system loses power.

EEPROM = Electrically Erasable Programmable Read-Only Memory

AVR Applications

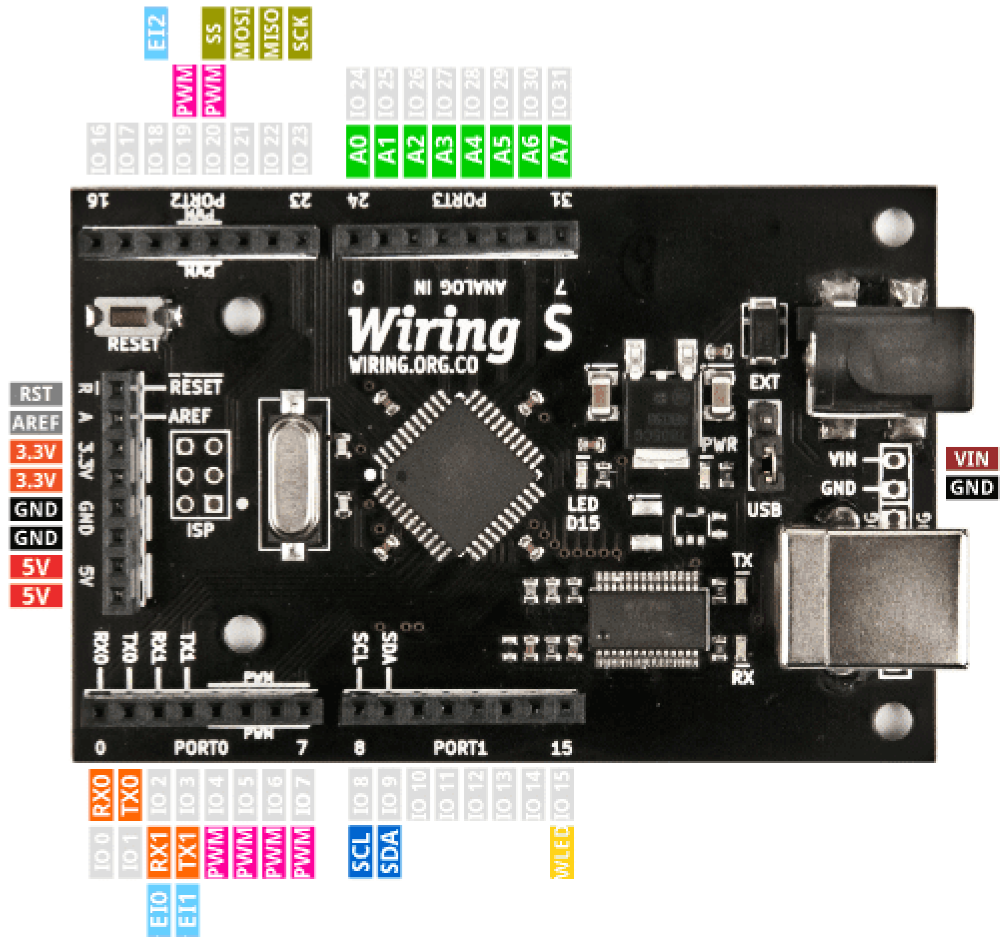
The Arduino environment is based on Atmel Atmega microcontrollers

AVR-Single-Chip-Processors AT90S, ATtiny, ATmega

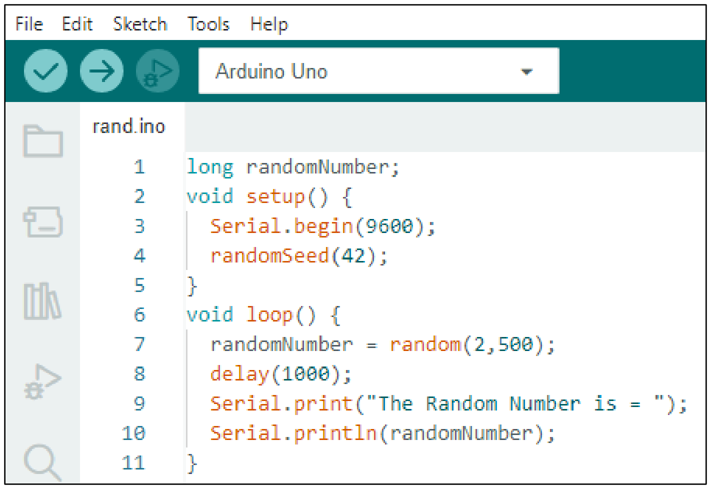
http://www.avr-asm-tutorial.net/avr_en/apps/APPS.html

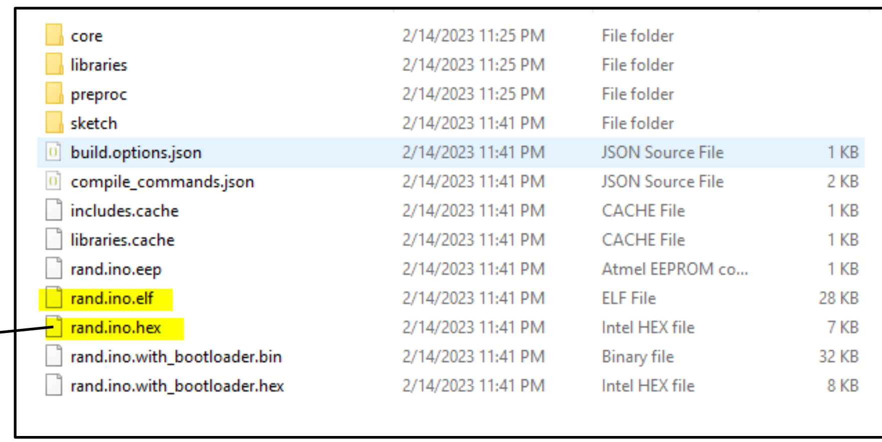
FEATURES

- 1 Amp 7805 5 Volt voltage regulator
- FTDI USB to serial converter
- USB A jack
- 2.1 mm power jack - 7.5 to 12 V DC
- Battery input - 7.0 to 12 V DC
- MCU: Atmel ATmega644P
- 16 MHz crystal oscillator
- Total flash memory: 65536 bytes (64 KiB)
- Available flash memory: 63488 bytes (62 KiB)
- SRAM: 4096 bytes (4 KiB)
- EEPROM: 2048 bytes (2 KiB)
- 2 Hardware serial ports
- 1 Hardware I2C port
- 8 Analog to digital inputs
- 6 Hardware PWM pins



http://wiring.org.co/hardware/





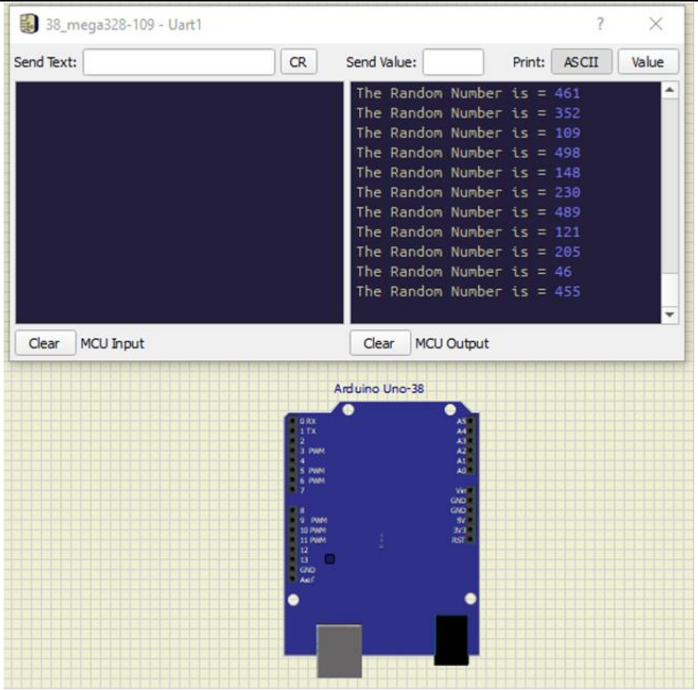
core	2/14/2023 11:25 PM	File folder	
libraries	2/14/2023 11:25 PM	File folder	
preproc	2/14/2023 11:25 PM	File folder	
sketch	2/14/2023 11:41 PM	File folder	
build.options.json	2/14/2023 11:41 PM	JSON Source File	1 KB
compile_commands.json	2/14/2023 11:41 PM	JSON Source File	2 KB
includes.cache	2/14/2023 11:41 PM	CACHE File	1 KB
libraries.cache	2/14/2023 11:41 PM	CACHE File	1 KB
rand.ino.eep	2/14/2023 11:41 PM	Atmel EEPROM co...	1 KB
rand.ino.elf	2/14/2023 11:41 PM	ELF File	28 KB
rand.ino.hex	2/14/2023 11:41 PM	Intel HEX file	7 KB
rand.ino.with_bootloader.bin	2/14/2023 11:41 PM	Binary file	32 KB
rand.ino.with_bootloader.hex	2/14/2023 11:41 PM	Intel HEX file	8 KB

Start code Byte count Address Record type Data Checksum

```
:10010000214601360121470136007EFE09D2190140
:100110002146017E17C20001FF5F16002148011928
:10012000194E79234623965778239EDA3F01B2CAA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
```

Start code	Byte count	Address	Record type	Data	Checksum
:	10	0000	00	0C9434000C9446000C9446000C944600	6A

```
1   :100000000C9435000C945D000C945D000C945D0024
2   :100010000C945D000C945D000C945D000C945D00EC
3   :100020000C945D000C945D000C945D000C945D00DC
4   :100030000C945D000C945D000C945D000C945D00CC
5   :100040000C9479010C945D000C94E9010C94C301AB
6   :100050000C945D000C945D000C945D000C945D00AC
7   :100060000C945D000C945D001C0311241FBECFEFA7
8   :10007000D8E0DEBFCDBF11E0A0E0B1E0E0EBF8E0FA
9   :1000800002C005900D92A233B107D9F721E0A2E397
                        .
                        .
                        .
147 :10092000911DB29F700D811D1124911D0895EE0F30
148 :0E093000FF1F0590F491E02D0994F894FFCF7D
149 :10093E00010000000000000024018400B1007101DC
150 :10094E00E200C000D400546865205261 6E646F6D81
151 :10095E00204E756D62657220697320 3D20000D0A70
152 :02096E00000087
153 :00000001FF
```



Size = 147*16 + 14 + 3*16 + 2 = 2,416 bytes

```
Sketch uses 2416 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 220 bytes (10%) of dynamic memory, leaving 1828 bytes for local variables. Maximum is 2048 bytes.
```

METRIC UNITS

$$1 \quad \text{KiB} = 2^{10} \quad \text{Bytes}$$
$$1 \quad \text{MiB} = 2^{20} \quad \text{Bytes}$$
$$1 \quad \text{GiB} = 2^{30} \quad \text{Bytes}$$
$$1 \quad \text{TiB} = 2^{40} \quad \text{Bytes}$$

$$1 \quad \text{KB} = 10^{3} \quad \text{Bytes}$$
$$1 \quad \text{MB} = 10^{6} \quad \text{Bytes}$$
$$1 \quad \text{GB} = 10^{9} \quad \text{Bytes}$$
$$1 \quad \text{TB} = 10^{12} \quad \text{Bytes}$$

METRIC UNITS

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	yotta

Communication

$$
\begin{aligned}
&\text{1-kbps} &: 10^{3} \quad &\text{bits per second}\\
&\text{1-Mbps} &: 10^{6} \quad &\text{bits per second}\\
&\text{1-Gbps} &: 10^{9} \quad &\text{bits per second}\\
&\text{1-Tbps} &: 10^{12} \quad &\text{bits per second}
\end{aligned}
$$