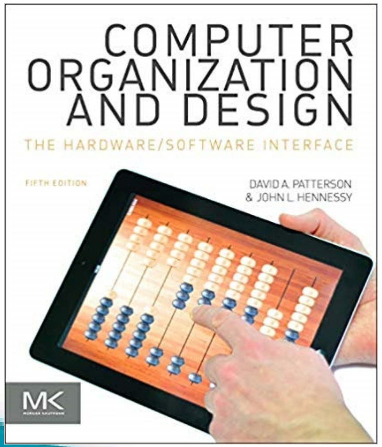
Computer Architecture and Organization

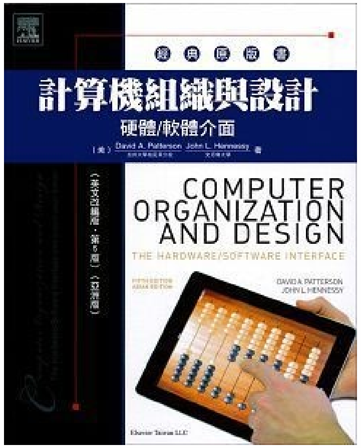
INSTRUCTOR: YAN-TSUNG PENG

DEPT. OF COMPUTER SCIENCE, NCCU

Logistics

- Time: Wed. D56
- Location: Room 200101, Da Ren Buliding
- Textbook: Computer Organization and Design MIPS Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) 5th Edition by David A. Patterson (Author), John L. Hennessy


COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
FIFTH EDITION
DAVID A. PATTERSON & JOHN L. HENNESSY
MK


經典原版書
計算機組織與設計
硬體/軟體介面
(美) David A. Patterson John L. Hennessy 著
COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
DAVID A. PATTERSON
JOHN L. HENNESSY
Elsevier Taiwan LLC

NCCU Moodle Platform

- Website: https://moodle.nccu.edu.tw/course/view.php?id=35025
  - 1121_703019001_計算機結構與組織
  - Course slides
  - Grading
  - Turn in HW

Teaching Assistant & Office Hours

- 蕭紘軒 111753132@nccu.edu.tw
- 鄭昱亭 108304034@nccu.edu.tw
- Office hours: Thurs. 16:00-17:00, 大仁樓308 or scheduled via email

- What TAs can help with
  - Answering your questions
  - Review sections
  - Grading

Grading

- Participation - up to 15%
  - Class Quizzes
- Homework - 30%
  - Generally, you will have two weeks to finish it
  - Handwritten HW needs to be turned in in class (Department, Grade, Name, and ID)
  - Programming HW needs to be submitted to moodle
  - Any late submission only receives 50% credit (no later than the answer is released).
  - No plagiarism
- Midterm exam - 30%
- Final exam - 30%

Misc.

- Please check moodle or email for new notifications and announcements.
- Recommended prerequisite: Logic Design

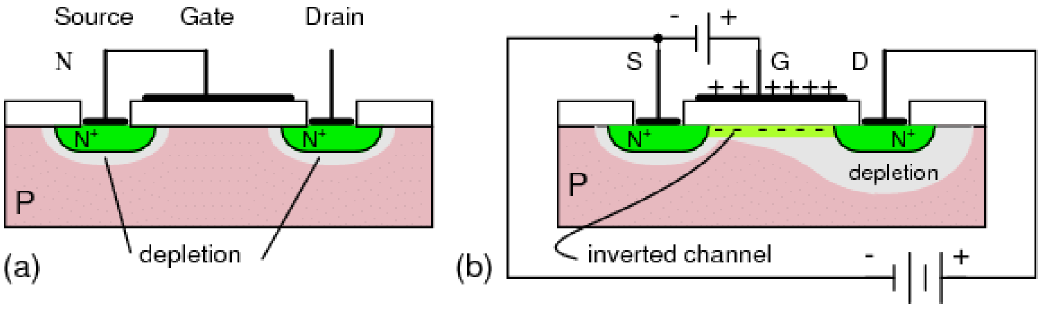
Course Introduction

- This course aims to introduce fundamental concepts and implementation of computer architecture. We will be offering students many chances to practice what is taught in class with using various examples.
- Students are expected to learn and understand how a computer (especially CPUs and memory systems) works between the digital logic level and operating system level after this class.

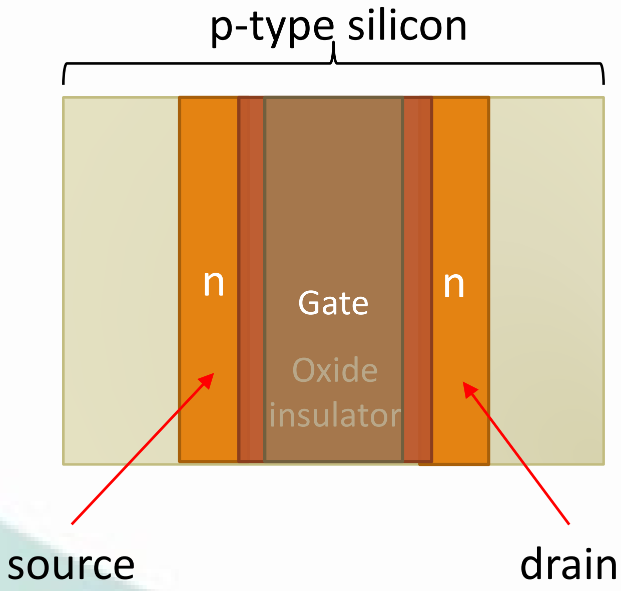
What you are going to learn?

- What determines program performance how it can be improved
- How programs are translated into the machine language and how the hardware executes them
- The hardware/software interface
- Arithmetic for computers
- How to design a CPU
- Cache and memory
- Storage and I/O
- Multicores, multiprocessors, and clusters

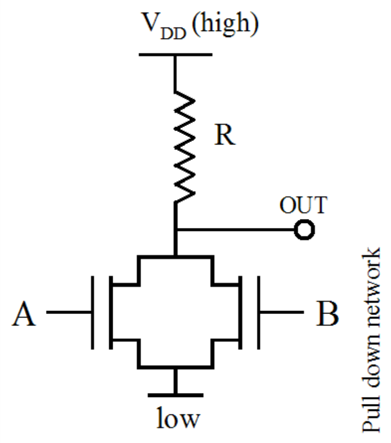
Signal : 0/1



- N-type metal-oxide-semiconductor (NMOS) transistor
  - Can be used to implement logic gates



N-type Transitor



https://en.wikipedia.org/wiki/NMOS_logic

NOR Gate

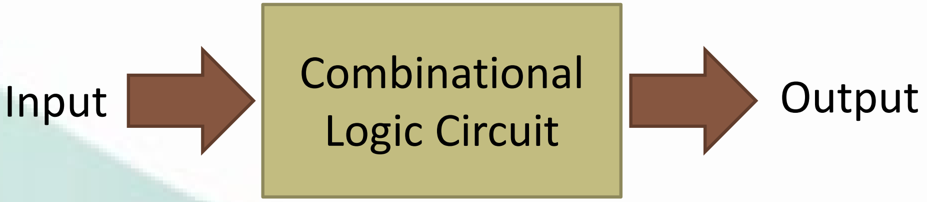
A	B	O
0	0	1
0	1	0
1	0	0
1	1	0

Applying positive voltage to the gate can form a channel between the source and drain

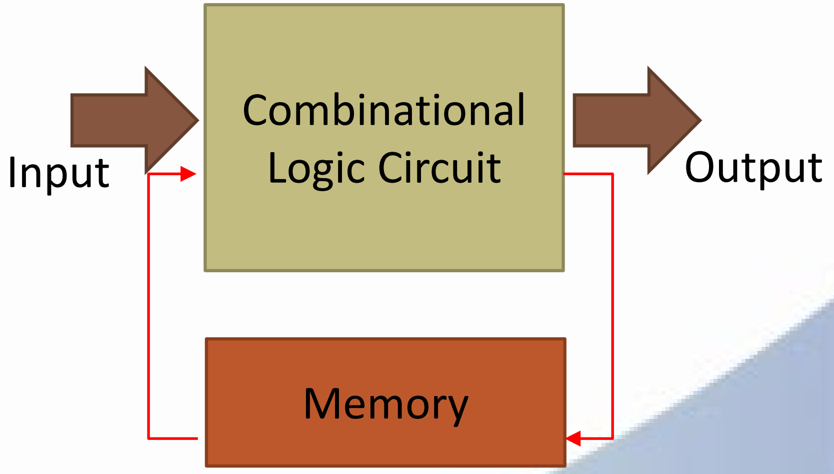
Figure on the top is taken from Kuphaldt, T. R. Insulated-gate Field-effect Transistors (MOSFET) <https : / / www . allaboutcircuits.com/textbook/semiconductors/chpt-2/insulated-gate-fieldeffect-transistors-mosfet/>.

Logic Gates and Circuits

- Using a functionally complete gate (NAND and NOR), we can build all types of logic gates, such as AND, OR, NOT.
- With logic gates, we can build different circuits
  - Combinational logic circuits
  - Sequential logic circuits

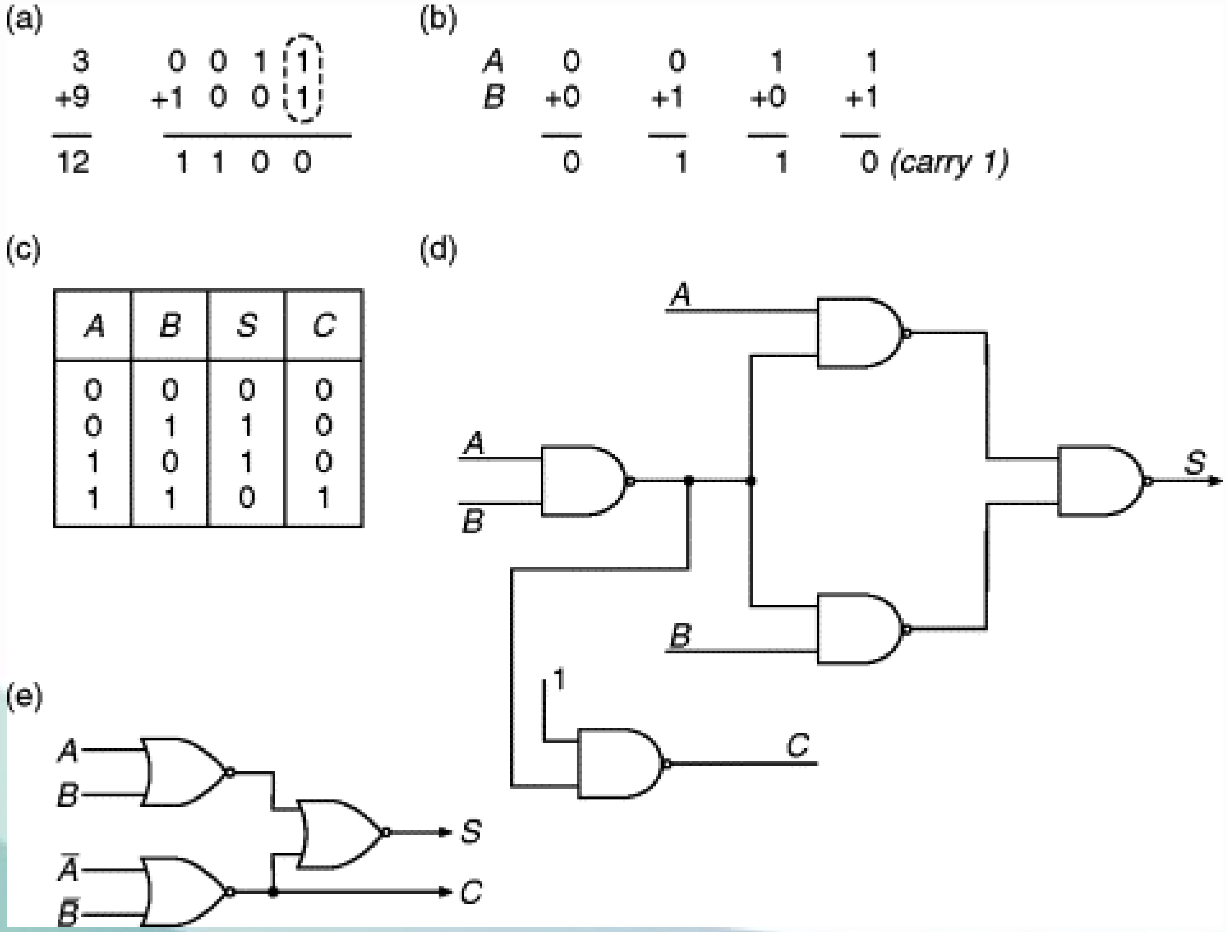


ex: Adder



ex: Multiplier

1-bit Adder



(a)

```
  3      0 0 1 1
 +9     +1 0 0 1
 ──     ────────
 12      1 1 0 0
```

(b)

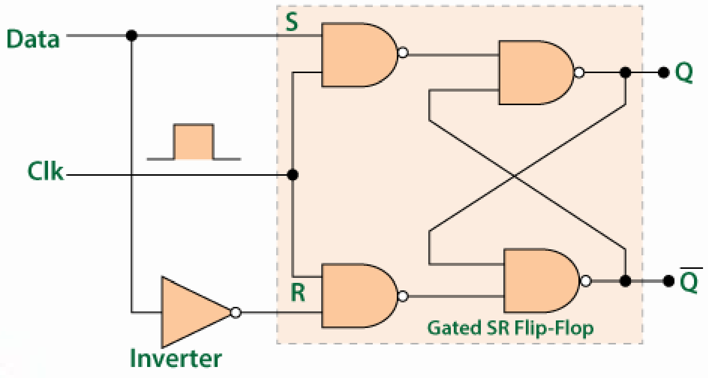
```
A   0     0     1     1
B  +0    +1    +0    +1
   ──    ──    ──    ──
    0     1     1     0 (carry 1)
```

(c)

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

https://www.sciencedirect.com/topics/computer-science/half-adder

D Flip-flop



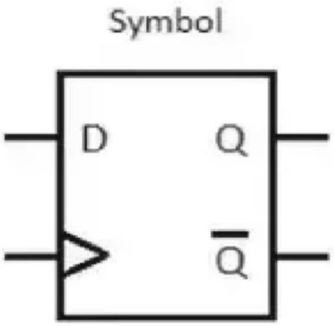


Table of truth:

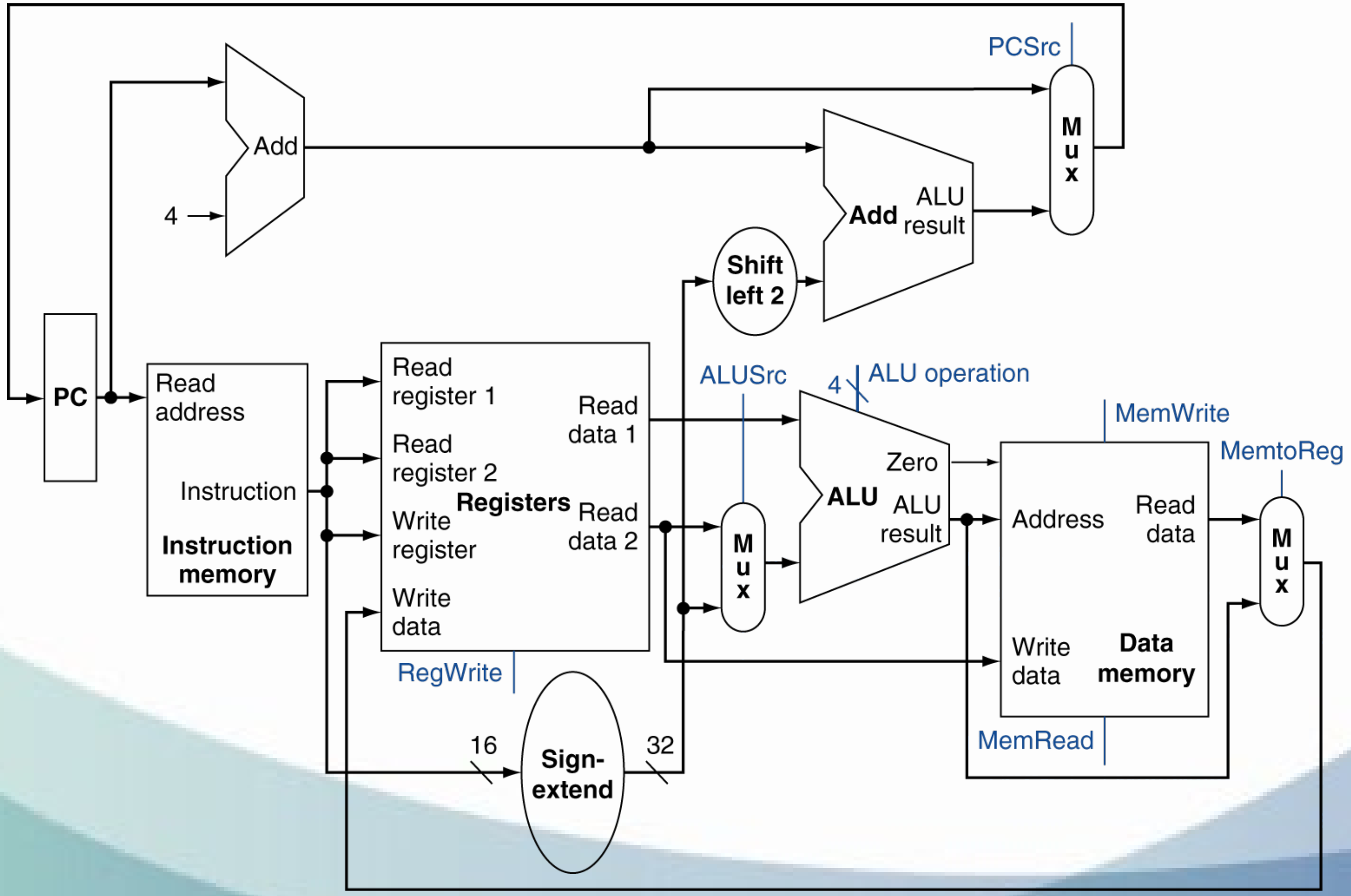
clk	D	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	Q	$\overline{Q}$
1	0	0	1
1	1	1	0

$$\text{clk} = \begin{cases} 0, \text{not changing} \\ 1, \text{low to high, rising} \end{cases}$$

https://www.javatpoint.com/d-flip-flop-in-digital-electronics

Damodaran *et al.*, All optical clocked D flip flop for 1.72 Tb/s optical computing, Microelectronics Journal, 2020

Processor


PCSrc
Add
4
M
u
x
Add
ALU
result
Shift
left 2
PC
Read
address
Instruction
Instruction
memory
Read
register 1
Read
register 2
Write
register
Write
data
Registers
Read
data 1
Read
data 2
RegWrite
ALUSrc
M
u
x
4
ALU operation
Zero
ALU
ALU
result
MemWrite
MemtoReg
Address
Read
data
Write
data
Data
memory
MemRead
M
u
x
16
Sign-
extend
32

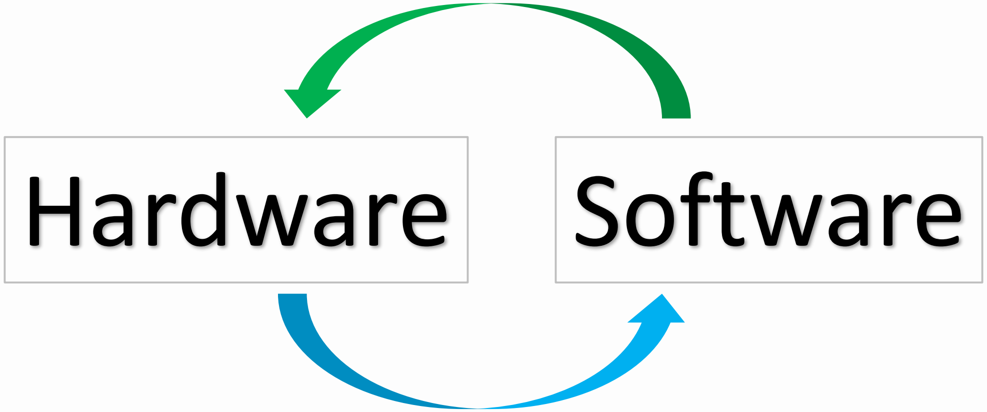
Chapter 1 – Computer Abstractions and Technology

- Moore's Law - Moore's perception
  - For every two years, the number of transistors on a microchip doubles, while the cost of computers reduces by half

- Advanced technology in computer development
  - *Computers in automobiles*
  - *Cell phones*
  - *Human genome project*
  - *World Wide Web*
  - *Search engines*

- Ubiquitous Computing

Computer Development



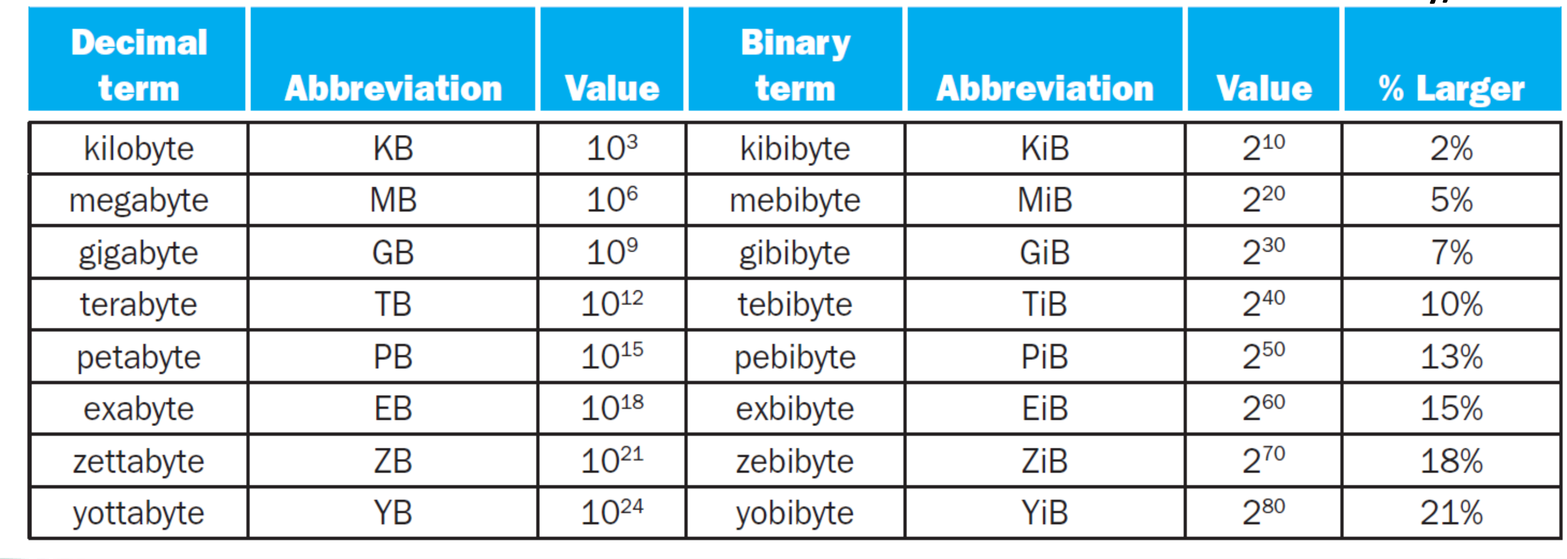
Hardware advances push software to move forward, and vice versa

Types of Computers

- Desktop
  - For general purpose
  - Low cost, decent performance
- Server
  - For a purpose of serving multiple users
  - High cost, high-end computing capability and reliability
- Supercomputer
  - Highest computing capability
  - Terabytes of memory
- Embedded Computer
  - For specific purposes, such as low power, small size
  - Limited hardware resources compared to a desktop or server
  - Run one application or a set of related applications

Common Size Terms

Binary/Decimal

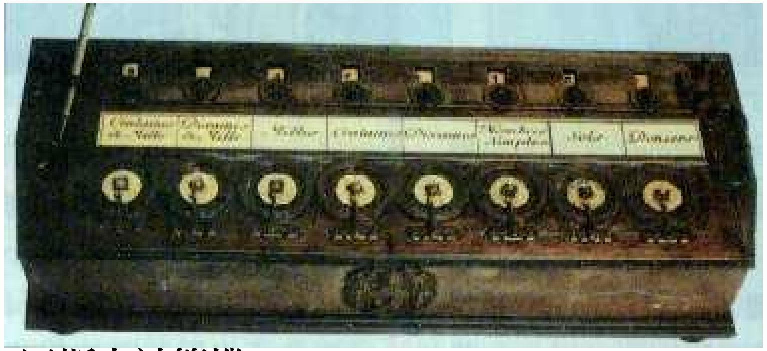


Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

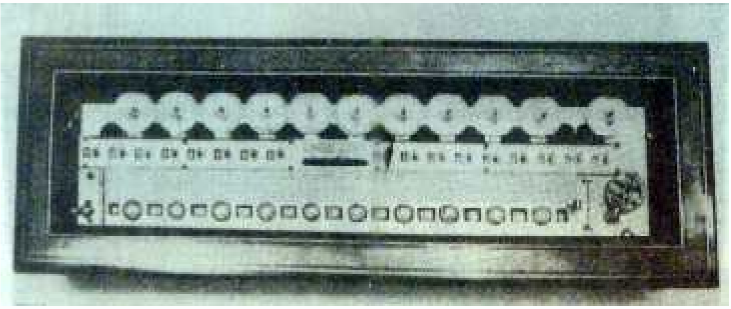
Figure is from “Computer Organization and Design MIPS Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design)” 5th Edition by David A. Patterson (Author), John L. Hennessy

History of Computer Development

- 巴斯卡計算機
- 1642年由法國數學家巴斯卡發明



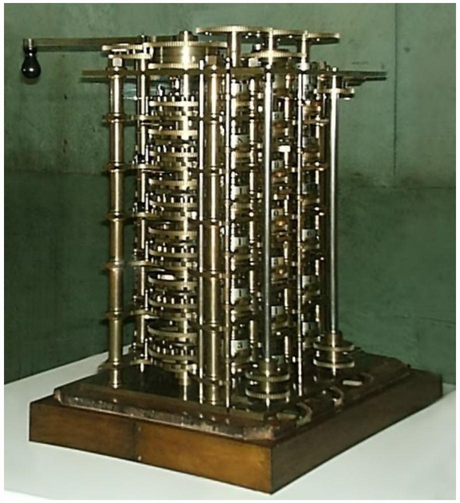
巴斯卡計算機



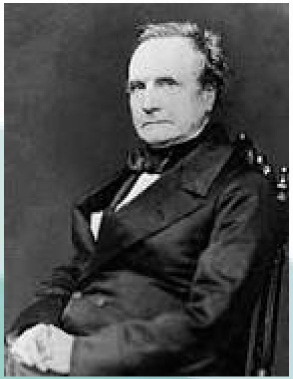
莫南計算機 (1666, 能做除法)

- 包含一組相連的齒輪，只要在上面撥出數字，齒輪之間互相轉動算出累加後的值，並顯示在上方的小窗口上
- 原理和算盤接近，低位齒輪轉10周，高位就會多轉一周，只能做加減法運算，但不需要記憶特定的口訣
- 清朝末年有一台巴斯卡計算機被送給慈禧太后，目前還保留在故宮
- 為記念巴斯卡，1971年Niklaus Wirth發明的程式語言便以Pascal命名

「機械式」計算機



巴倍奇計算機模組圖/Wikipedia



巴倍奇(1792-1871)

巴倍奇計算機 (1830)

- 巴倍奇24歲就昇等為英國劍橋大學第三位路卡斯講座教授
  - 前二位是牛頓的老師巴羅及牛頓
- 以當時技術，組合如此多細小機械式齒輪零件幾乎不太可能，因此到最後都只停留在設計階段，最後花了英國政府近二萬英磅仍以失敗收場

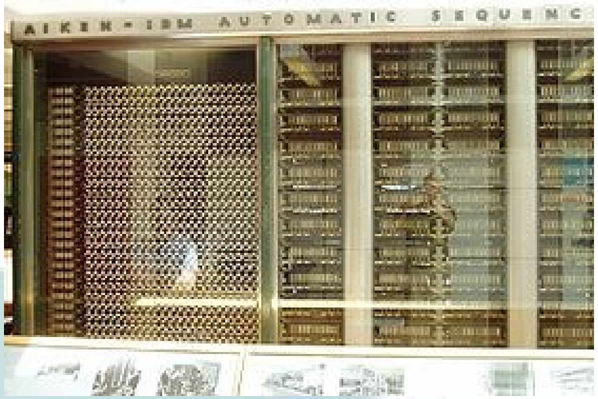
當時一台蒸氣火車頭值178英磅

- 其設計已具備現代電腦的重要架構: 輸出入、時序、儲存、運算、控制，但當時被視為癡人的臆想
  - 100年後第一台真正的電腦才誕生

IBM Mark I (1944)

- 發明人
  - 哈佛大學 Howard Aiken博士

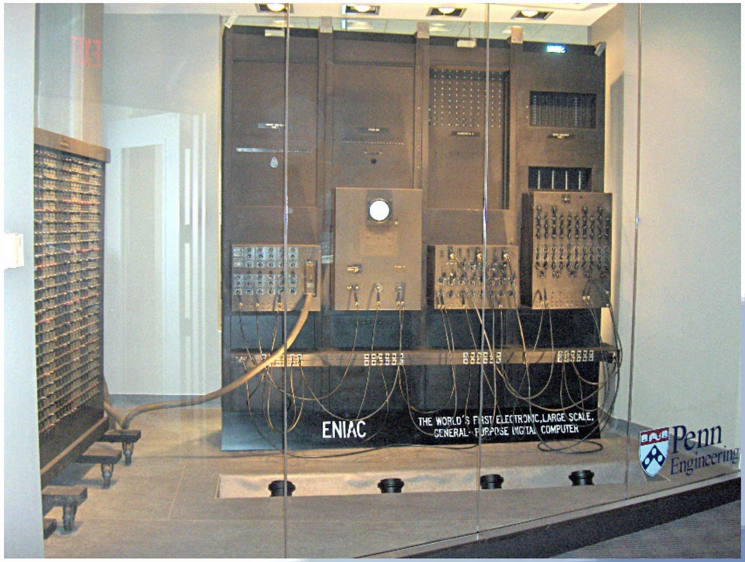
- 機械電子式計算器
  - 約略根據**巴倍奇原始設計改進而成**主要採用繼電器及大量齒輪組合而成
  - 每0.33秒1次加法，每6秒可計算1次乘法，每12秒1次除法
  - 沒有控制流程功能，據說寫迴圈時要真的將紙帶捲成圈圈



德國在**1941**年就發展出使用**2**進位的繼電器通用電腦

ENIAC (1946公開)

- Electronic Numerical Integrator And Computer
  - 人類史上第一部真正的「電」腦
    - 計算時未用到任何機械裝置
    - 人: John Mauchly (莫克利) 與 J. Presper Eckert (艾克特)
    - 地: 賓州大學莫爾學院
    - 構造: 18000個真空管、7200電晶體、10000電容
    - 性能: 每秒1900個加/減法、提供條件式跳躍
      - 具有累加器、可存儲十位十進位數
      - 計算速度比機電機器快1,000x
    - 邏輯和資料分開
      - 邏輯: 實體接線、切換器
      - 資料: 紙卡
  - 問題:
    - 可靠度:幾乎每天都必須更換新的真空管
    - 設計複雜: 20個十進位暫存器 (佔61cm)


ENIAC
THE WORLD'S FIRST ELECTRONIC,LARGE SCALE, GENERAL-PURPOSE DIGITAL COMPUTER
Penn Engineering

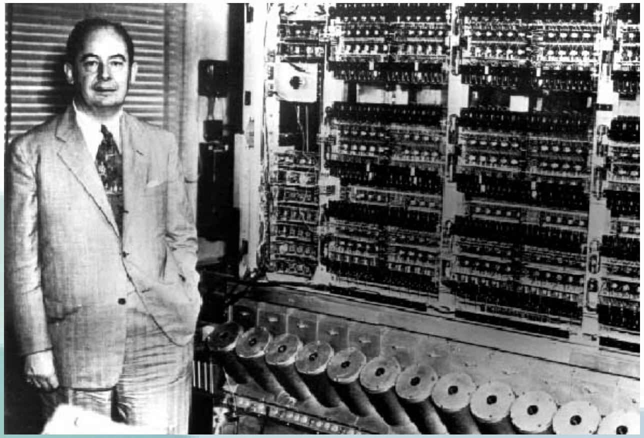
圖/Wikipedia

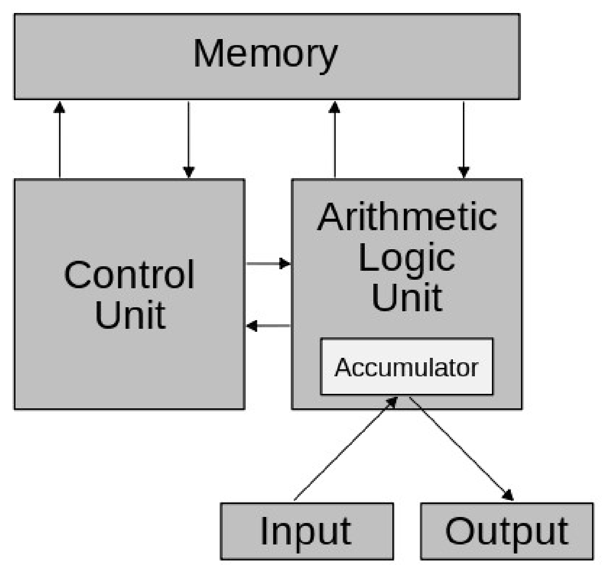
EDVAC設計草案

- Electronic Discrete Variable Automatic Computer
  - 以２進位為基礎，解決ENIAC的效能問題
  - 採用指令式可存式程式概念
    - ENIAC輸入程式非常麻煩，幾分鐘的計算要花幾小時佈線，每次都要重來
- EDVAC技術顧問馮・諾曼(John von Neumann)所提出的EDVAC設計草案成為現今所有電腦的基礎架構
  - Von Neumann architecture
  - 在實作上，John Mauchly 與 J. Presper Eckert 也有相當貢獻

Von Neumann Architecture

- 傳統硬體裝置的運作過程都是死的(實體管線)，因此更改設計代表著整個機器要重新設計
- 在馮・諾曼架構下，所有運算過程被視為「一連串指令」(也就是現在說的「程式」)
  - 程式可存在儲存元件中，被即時修改或移動 (重新定址)
  - 間接造就了「軟體」的概念





莫克利

艾克特

創業

馮．諾曼

普林斯頓大學

1945年底，三人為文件署名問題鬧翻

間接造成賓州大學莫爾學院失去在電腦發展史的重要地位

莫克利

艾克特

1950 → Remington Rand, Inc. → 1955 → Sperry Corporation → 1986 → UNISYS

UNIVAC1
(美國第一台商用電腦)
正確預測1952年選舉結果

艾克特 -
莫克利公司

馮・諾曼

1945 → 普林斯頓大學 → IBM

1947:電晶體發明

設計並公開IAS架構，
效能是ENIAC的240倍

1952 IBM發表701型電腦是IBM第一部商用電腦

電晶體/積體電路電腦的時代

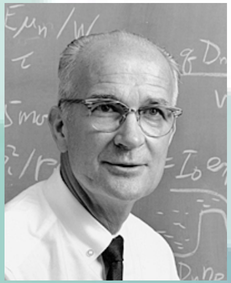
1960年, IBM推出第一款由電晶體製作的電腦 IBM 7090

包括著名的摩爾（R.Moore）博士

「博士生產線」

FAIRCHILD SEMICONDUCTOR®

1957: 快捷半導體成立



1955蕭克利半導體實驗室

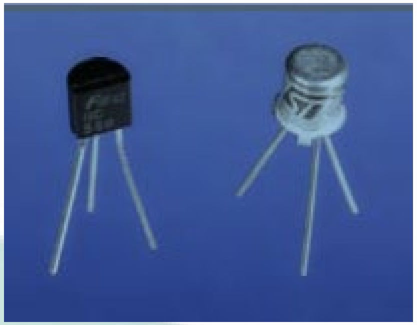
諾依斯

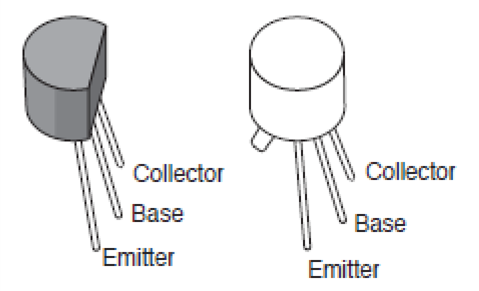
1968: Intel成立

Integrated Electronics

電晶體

- 發明人
  - 1948年由Bardeen, Shockley和Brattain共同發明
  - 1956年
    - 三人共得諾貝爾物理學獎





SEPTEMBER · 1948
electronics
A McGRAW-HILL PUBLICATION
Revolutionary Amplifier
The CRYSTAL TRIODE

Apple

- 創立於1975-76，以個人電腦起家
  - 在1970年代電腦是只有大公司及政府機構才負擔得起的昂貴商品
  - 個人電腦的概念是近乎不切實際的於奇特概念

IBM加入個人電腦戰局

- 從1975-1980，Apple靠銷售個人電腦進入全美500大企業
- 1981年，IBM以開放式架構正式加入個人電腦戰局
  - Intel 8088, 開放式架構，將所有技術文件公開
  - 採用微軟MS-DOS為作業系統 (原名QD-DOS)
  - 1983年IBM PC市佔率首度超越Apple

Apple反擊

- Apple反擊失敗
  - 1984年，Jobs推出研發許久的麥金塔(Mac)並重金聘請了百事可樂的John Sculley，推出有名的廣告: 1984 (耗資160萬美元)
    - John Sculley於1985將Jobs趕出Apple
    - 但John Sculley管理Apple不當，最終於1993年離開蘋果
  - 1997年，Jobs回到Apple
    - 被迫接受Microsoft 150億美元的投資
    - 10年後以iPhone讓Apple在行動市場上重創微軟

1984

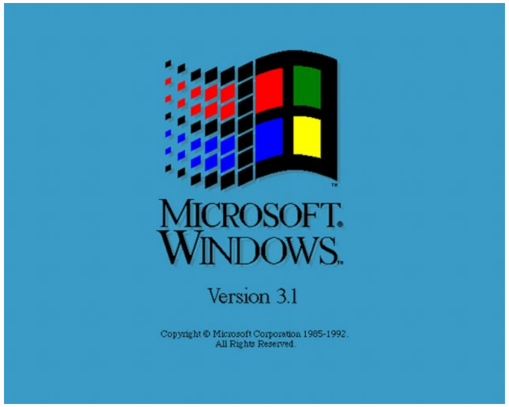
1984 廣告概念:
一無名主角會從Big Brother手上拯救全人類
而該無名主角代表Mac電腦

Intel的年代

- 80286 (1990)
  - 成功擊敗Apple的代表作
- 80386(1985)
  - IBM開始向相容商收取高額授權金
  - 80386被IBM拒絕，但被COMPAQ採用為個人電腦主處理器
  - 1994年起，COMPAQ超越IBM成為個人電腦市佔率最高的廠商
    - IBM個人電腦部門最後因績效不佳，在2004年賣給聯想

Wintel的年代

- Intel 80486(1994)
  - 由華人虞有澄領軍
  - 效能直逼工作站電腦的CPU
- 重要事件
  - Intel Pentium系列I~III (1995-2000)
    - 到PIII時遇到時脈瓶頸
  - Windows 95/98/NT/2000: 1996-2000
    - Windows 2000開始，完全採用新的NT系統核心，大幅提升了微軟作業系統的穩定度
  - Java: 1996
  - 瀏覽器大戰 IE vs. Netscape : 1996
  - Google成立: 1998
  - Code Red: 第一個普遍傳染的惡意網路病毒: 2000

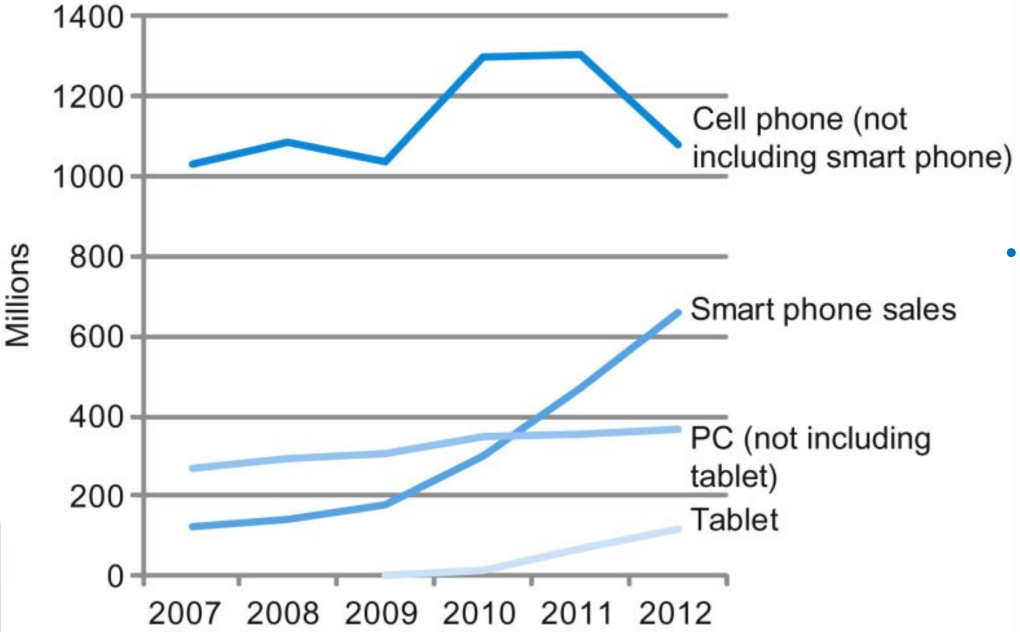


MICROSOFT.
WINDOWS.
Version 3.1
Copyright © Microsoft Corporation 1985-1992.
All Rights Reserved.

2000年以後: Google的年代

- PC市場已成紅海，戰場轉向網路: 2000
  - Google成立: 1998
    - 上市: 2004
  - Facebook: 2004
    - 上市: 2012
- 智慧手機普及化: 2009
  - 微軟、Nokia在智慧手機競爭中失利
  - Google 收購Android: 2007
  - iPhone第一代: 2007

PostPC Era

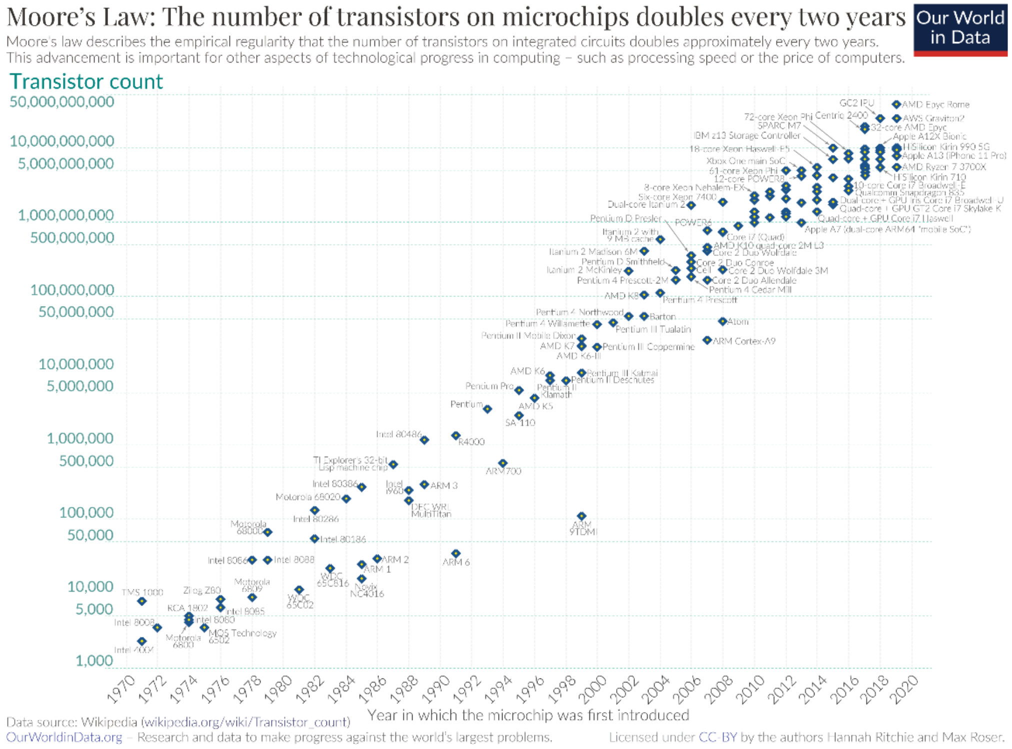


- Servers are gradually replaced by cloud computing farms, like AWS from Amazon and Google Cloud

Figure is from “Computer Organization and Design MIPS Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design)” 5th Edition by David A. Patterson (Author), John L. Hennessy

Moore’s Law

- In 1965, Gordon Moore observed that the number of transistors on a microchip doubles every two years.
- Based on the past data, the linear correlation between time and the number of transistor has been corrected to doubling every 18 months



Moore’s Law: The number of transistors on microchips doubles every two years

Moore’s law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

https://en.wikipedia.org/wiki/Moore%27s_law

Below Your Program

- Applications
  - "If I have seen further it is by standing on the shoulders of Giants." by Isaac Newton in 1675
  - Written in high-level languages, Abstraction
- System Software
  - Compiler, operating system
- Hardware
  - CPU, memory, I/O controllers, storage, etc.

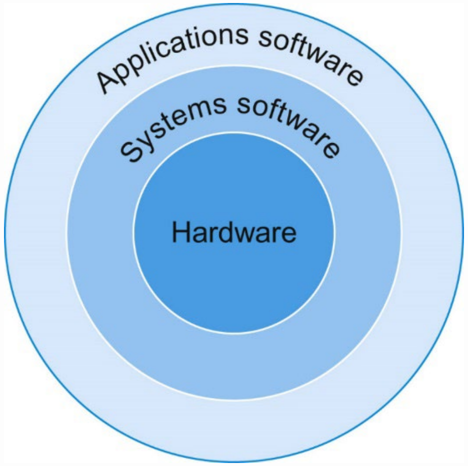
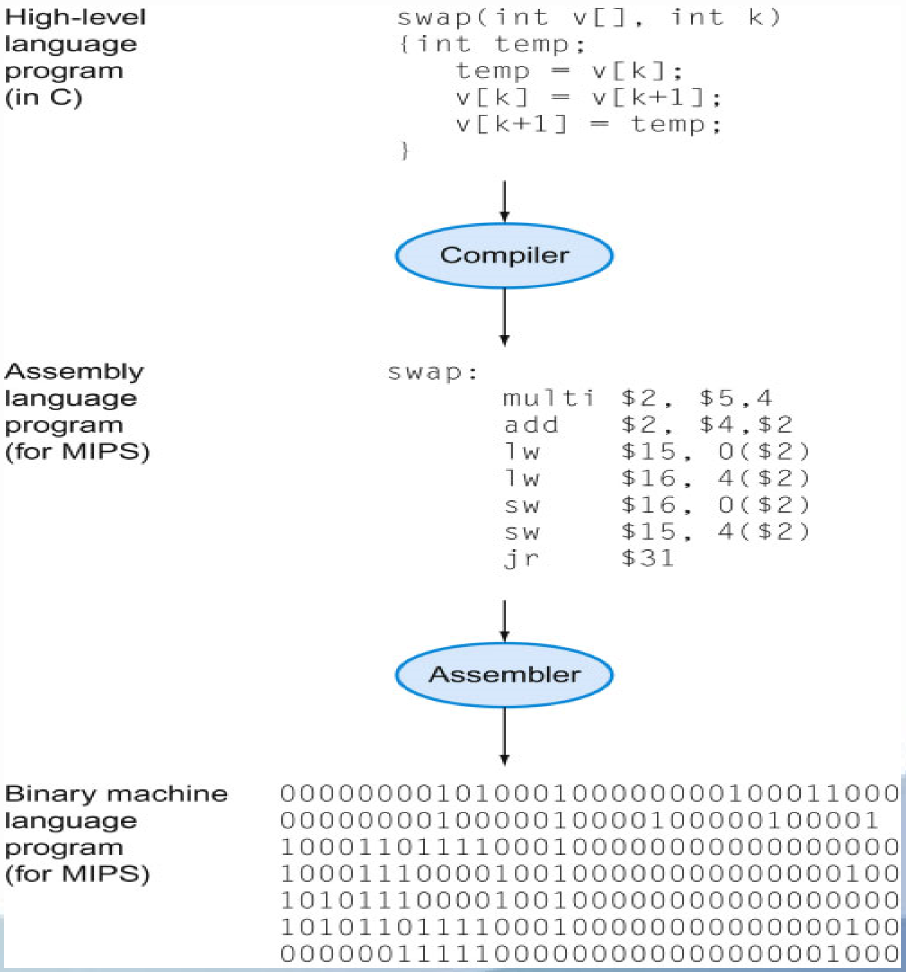


Figure is from "Computer Organization and Design MIPS Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design)" 5th Edition by David A. Patterson (Author), John L. Hennessy

Levels of Programming Languages

- High-level language
- Assembly language
- Hardware representation



High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Assembly language program (for MIPS)

```
swap:
      multi $2, $5,4
      add   $2, $4,$2
      lw    $15, 0($2)
      lw    $16, 4($2)
      sw    $16, 0($2)
      sw    $15, 4($2)
      jr    $31
```

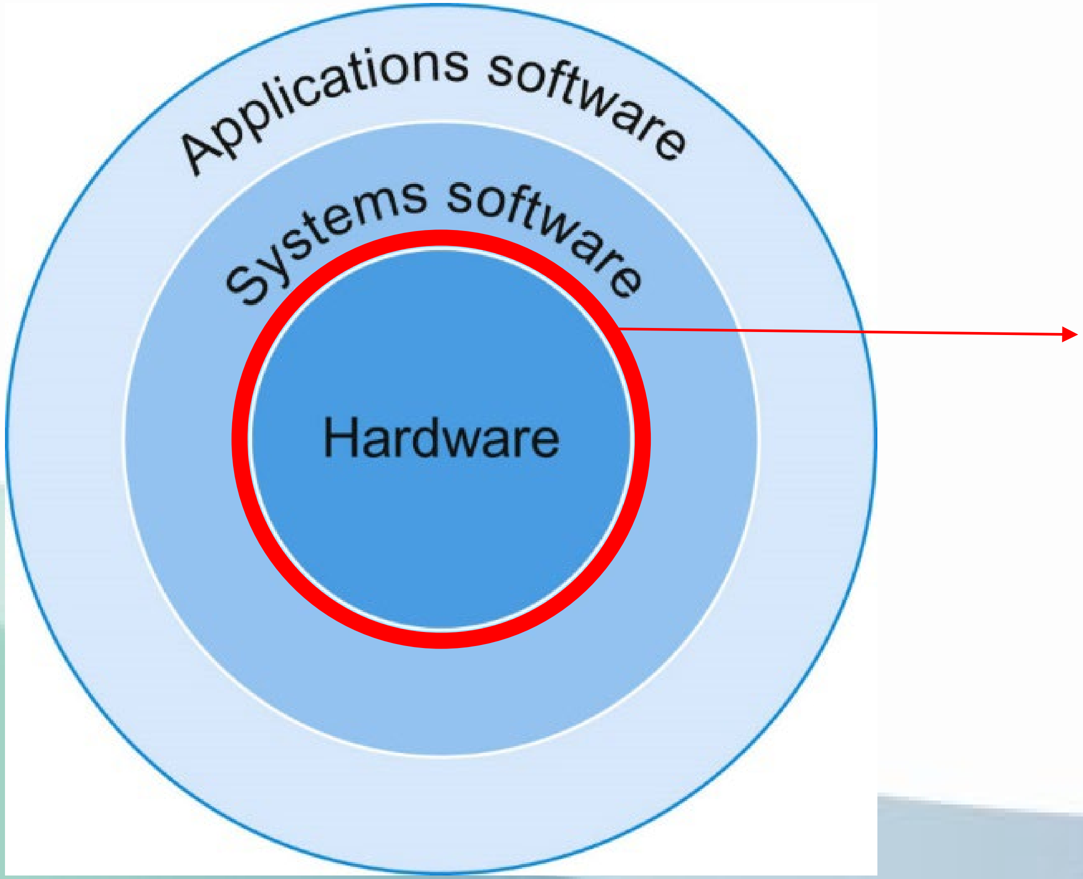
Binary machine language program (for MIPS)

```
00000000101000100000000100011000
0000000010000010000100000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
00000011111000000000000000001000
```

Figure is from "Computer Organization and Design MIPS Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design)" 5th Edition by David A. Patterson (Author), John L. Hennessy

Computer Architecture

Computer Architecture = Instruction Set Architecture + Hardware Organization



Instruction Set Architecture

- Language you use to command a computer's hardware
- Its vocabulary is called an **instruction set**
- Different computers may have different instruction sets
- Registers, Data Representations, Format, Addressing Mode, etc.

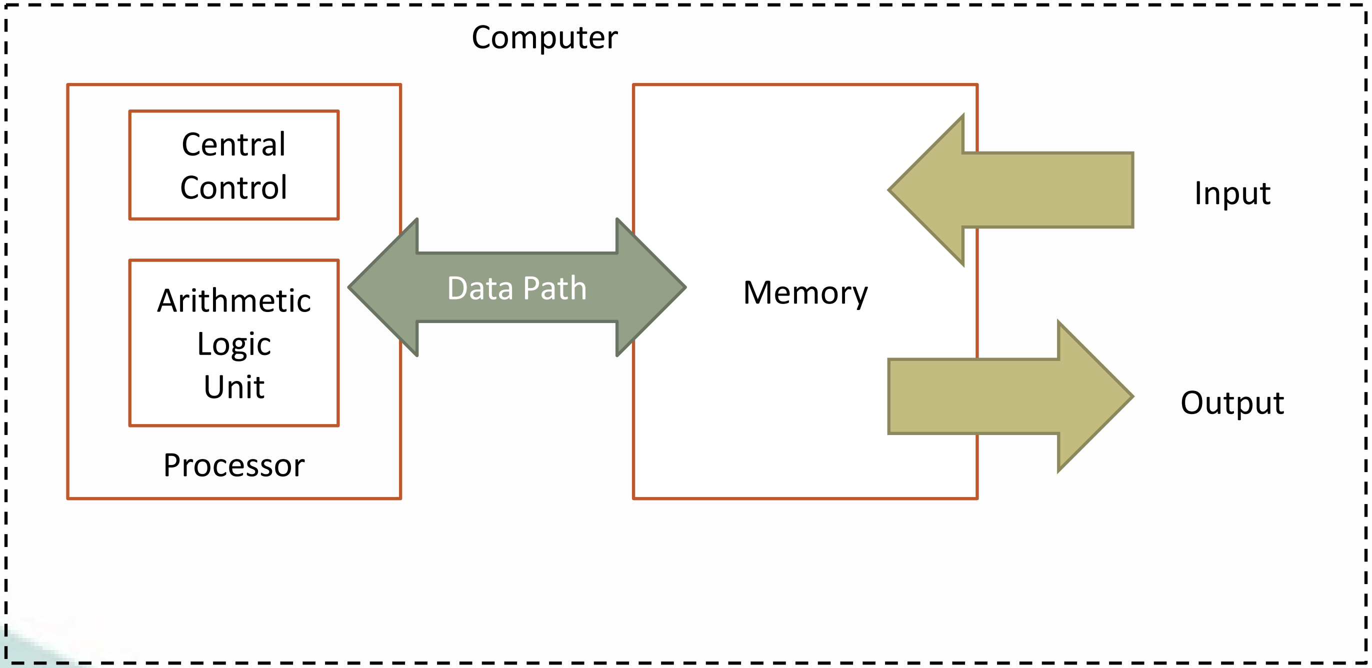
ISA

- An instruction is the most basic operation for a CPU, called a primitive operation.
- Instruction set is an abstraction interface between the low-level software and hardware, including a set of instructions that can be executed by a CPU
- Instruction Set Architecture includes
  - Instruction format
  - Registers, memory
  - Addressing modes

Types of ISA

Reduced Instruction Set Computer (RISC)	Complex Instruction Set Computer (CISC)
A small number of simple instructions	A large number of complex instructions
Each instruction does a little work and runs faster	Each instruction does a lot of work and runs slower
Easy for pipelining	Hard for pipelining
A program needs more instructions to implement and requires more memory to run	A program needs fewer instructions to implement and require less memory to run

Under the Covers

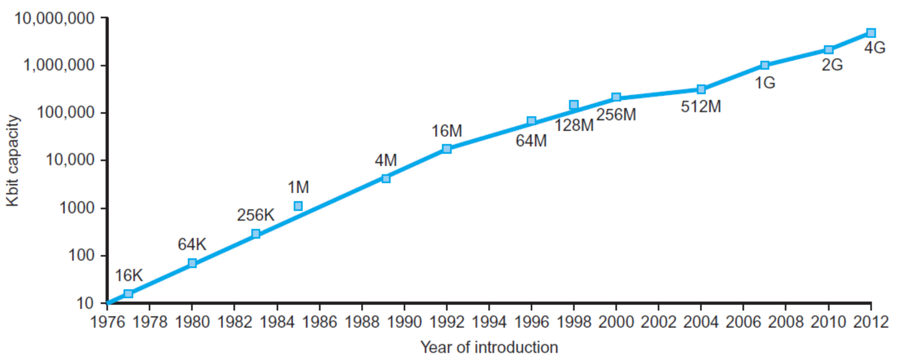


I/O devices:

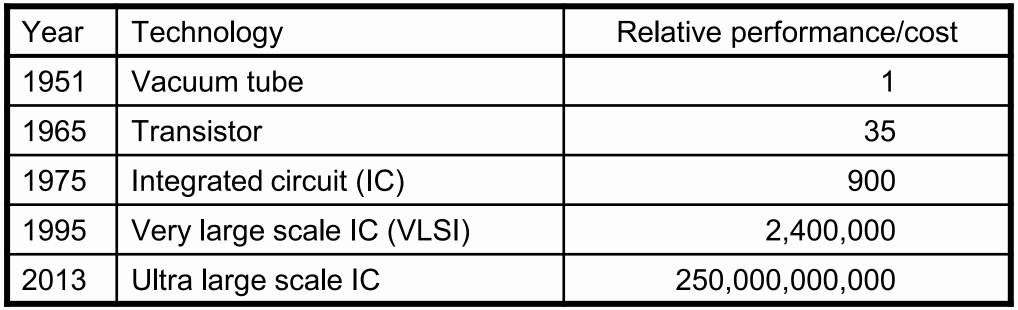
- Display
- Keyboard
- Mouse
- HD, flash
- Network devices

Technology Trends

- For Memory



- For Processors



Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Figure is from “Computer Organization and Design MIPS Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design)” 5th Edition by David A. Patterson (Author), John L. Hennessy