

25 YEARS ANNIVERSARY  
SOICT

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



TRƯỜNG ĐẠI HỌC  
BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

# KIẾN TRÚC MÁY TÍNH

## Computer Architecture

Course ID: IT3283

Nguyễn Kim Khánh

# Thông tin tác giả

- Nguyễn Kim Khánh
  - Engineer of HUST, 1985
  - PhD. in Computer Engineering, Belarus State University, 1991
  - Head of Department of Computer Engineering, SoICT
- Office: 802-B1
- e-mail: [khanhmk@soict.hust.edu.vn](mailto:khanhmk@soict.hust.edu.vn)  
[khanh.nguyenkim@hust.edu.vn](mailto:khanh.nguyenkim@hust.edu.vn)

# Mục tiêu học phần

- Sinh viên được trang bị các kiến thức cơ sở về **kiến trúc tập lệnh** và **tổ chức của máy tính**, cũng như những nguyên tắc cơ bản trong thiết kế máy tính.
- Sau khi học xong học phần này, sinh viên có khả năng:
  - Tìm hiểu kiến trúc tập lệnh của các máy tính cụ thể
  - Lập trình hợp ngữ
  - Đánh giá hiệu năng máy tính và cải thiện hiệu năng của chương trình
  - Khai thác và quản trị hiệu quả các hệ thống máy tính
  - Phân tích và thiết kế máy tính

- IT3280 2(0-4-0-4) - Thực hành Kiến trúc máy tính
- Thực hành về lập trình hợp ngữ và kiến trúc máy tính
- Giúp sinh viên hiểu rõ tương tác giữa phần cứng và phần mềm, hoạt động của máy tính thông qua mã hợp ngữ và công cụ mô phỏng

- Bài giảng Kiến trúc máy tính

- Sách tham khảo:

[1] Stallings W., *Computer Organization and Architecture*, 10<sup>th</sup> ed., Pearson 2016

[2] Patterson D., Hennessy J., *Computer Organization and Design*, 5<sup>th</sup> ed., Morgan Kaufmann, 2014

[3] Harris D., Harris S., *Digital Design and Computer Architecture*, 2<sup>nd</sup> ed., Morgan Kaufmann, 2013

[4] Tanenbaum A., *Structured Computer Organization*, 6<sup>th</sup> ed., Pearson 2013

- Phần mềm hỗ trợ học tập:

MARS (MIPS Assembler and Runtime Simulator)

# Nội dung học phần

Chương 1. Giới thiệu chung

Chương 2. Hệ thống máy tính

Chương 3. Kiến trúc tập lệnh

Chương 4. Số học máy tính

Chương 5. Bộ xử lý

Chương 6. Bộ nhớ máy tính

Chương 7. Hệ thống vào-ra

Chương 8. Các kiến trúc song song

## Chương 1

# GIỚI THIỆU CHUNG



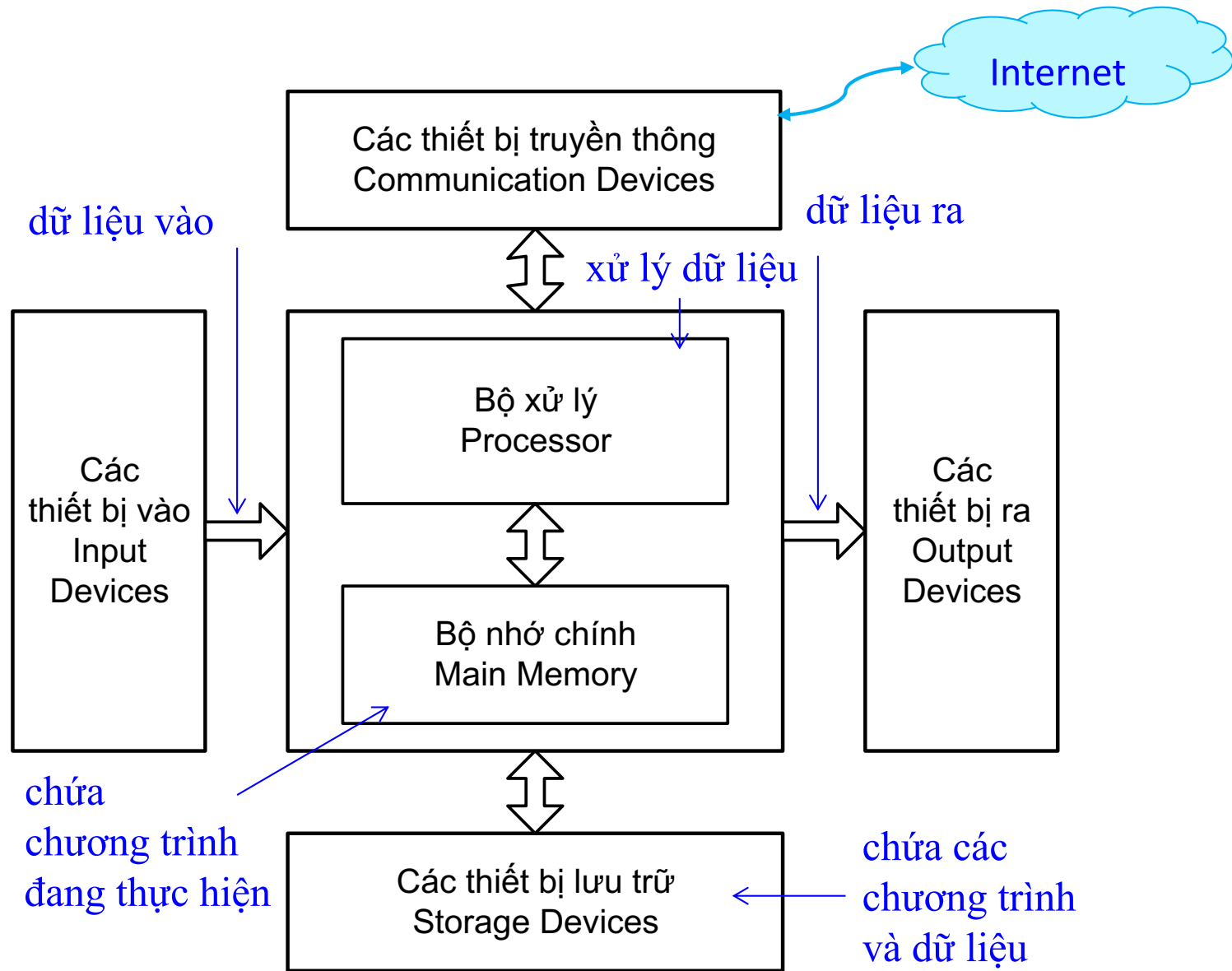
# Nội dung của chương 1

- 1.1. Máy tính và phân loại máy tính
- 1.2. Khái niệm kiến trúc máy tính
- 1.3. Sự tiến hóa của công nghệ máy tính
- 1.4. Hiệu năng máy tính

# 1.1. Máy tính và phân loại máy tính

- **Máy tính (Computer)** là thiết bị điện tử thực hiện các công việc sau:
    - Nhận dữ liệu vào,
    - Xử lý dữ liệu theo dãy các lệnh được nhớ sẵn bên trong,
    - Đưa dữ liệu (thông tin) ra.
  - Dãy các lệnh nằm trong bộ nhớ để yêu cầu máy tính thực hiện công việc cụ thể gọi là **chương trình (program)**.
- Máy tính hoạt động theo chương trình

# Sơ đồ đơn giản của máy tính



# Phân loại máy tính

- Máy tính cá nhân (PC - Personal Computers)
  - Desktop computers, Laptop computers
  - Máy tính đa dụng
- Thiết bị di động (PMD - Personal Mobile Devices)
  - Smartphones, Tablet Computers
  - Kết nối Internet
- Máy tính nhúng/ IoT (Embedded Computers/ Internet of Things)
  - Máy tính được đặt ẩn trong thiết bị khác
  - Được thiết kế chuyên dụng
  - IoT – là máy tính nhúng có kết nối với Internet

# Phân loại máy tính (tiếp)

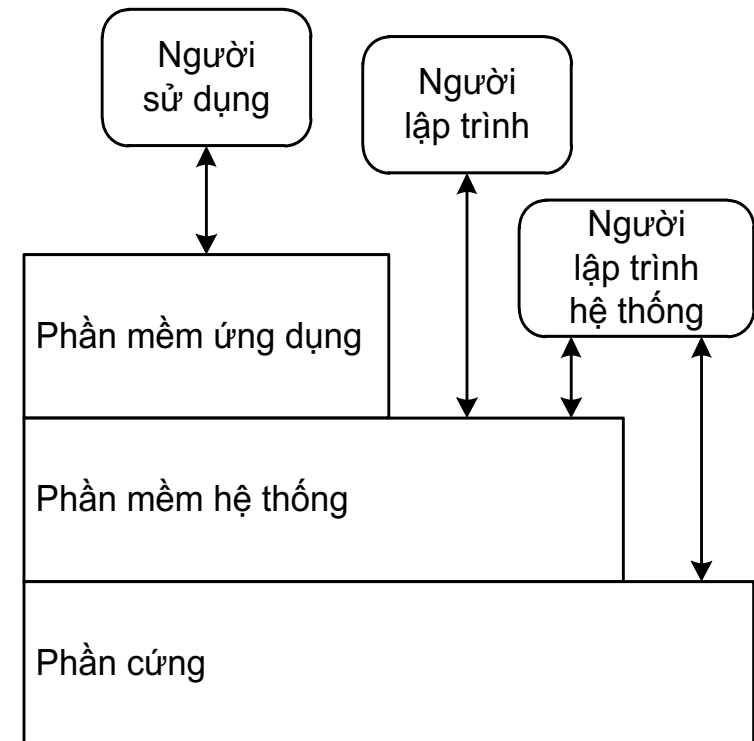
- Máy chủ (Servers) – máy phục vụ
  - Dùng trong mạng để quản lý và cung cấp các dịch vụ
  - Hiệu năng và độ tin cậy cao
  - Hàng nghìn đến hàng triệu USD
- Máy tính lớn/ Siêu máy tính (Warehouse-Scale Computers/ Supercomputers)
  - WSC còn gọi là Clusters: kết nối nhiều PCs hoặc Servers với nhau dùng cho các mục đích khác nhau
  - Supercomputers: Một dạng WSC dùng cho tính toán cao cấp trong khoa học và kỹ thuật
  - Hàng triệu đến hàng trăm triệu USD

## 1.2. Khái niệm kiến trúc máy tính

- Kiến trúc máy tính bao gồm 3 khía cạnh:
  - **Kiến trúc tập lệnh** (Instruction Set Architecture): nghiên cứu máy tính theo cách nhìn của người lập trình
  - **Tổ chức máy tính** (Computer Organization) hay **Vi kiến trúc** (Microarchitecture): nghiên cứu thiết kế máy tính ở mức cao (thiết kế CPU, hệ thống nhớ, cấu trúc bus, ...)
  - **Phần cứng** (Hardware): nghiên cứu thiết kế logic chi tiết và công nghệ đóng gói của máy tính.
- Cùng một kiến trúc tập lệnh có thể có nhiều sản phẩm (tổ chức, phần cứng) khác nhau
- Học phần này tập trung vào **Kiến trúc tập lệnh** và **Tổ chức máy tính**

# Phân lớp máy tính

- Phần mềm ứng dụng
  - Được viết theo ngôn ngữ bậc cao
- Phần mềm hệ thống
  - Chương trình dịch (Compiler): dịch mã ngôn ngữ bậc cao thành ngôn ngữ máy
  - Hệ điều hành (Operating System)
    - Lập lịch cho các nhiệm vụ và chia sẻ tài nguyên
    - Quản lý bộ nhớ và lưu trữ
    - Điều khiển vào-ra
- Phần cứng
  - Bộ xử lý, bộ nhớ, mô-đun vào-ra



# Các mức của mã chương trình

- Ngôn ngữ bậc cao
  - High-level language – HLL
  - Mức trừu tượng gần với vấn đề cần giải quyết
  - Hiệu quả và linh động
- Hợp ngữ
  - Assembly language
  - Mô tả lệnh dưới dạng text
- Ngôn ngữ máy
  - Machine language
  - Mô tả theo phần cứng
  - Các lệnh và dữ liệu được mã hóa theo nhị phân

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;
}
```

Compiler

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
```

Assembler

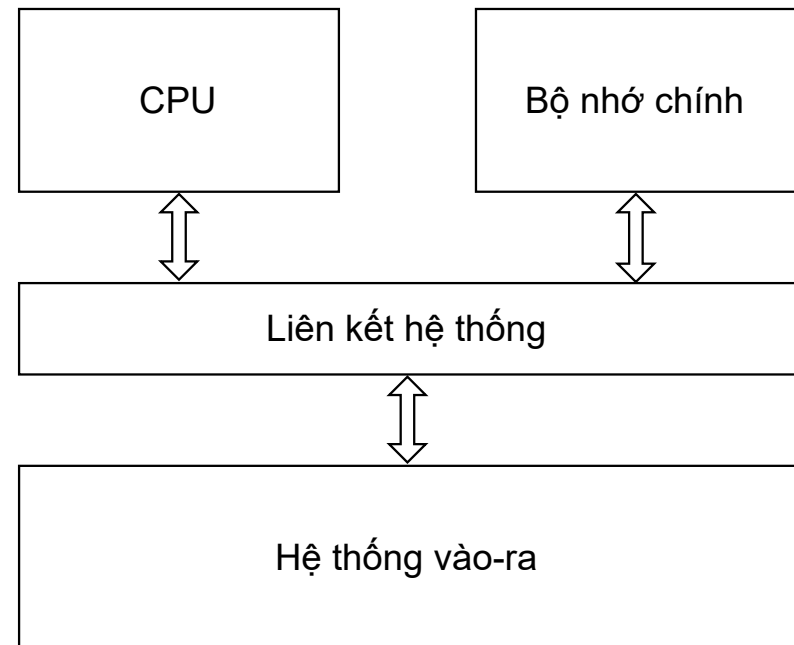
Binary machine  
language  
program  
(for MIPS)

```
000000001010001000000000100011000
000000001000001000010000000100001
100011011110001000000000000000000
100011100001001000000000000000100
101011100001001000000000000000000
101011011110001000000000000000100
00000011111000000000000000001000
```



# Các thành phần cơ bản của máy tính

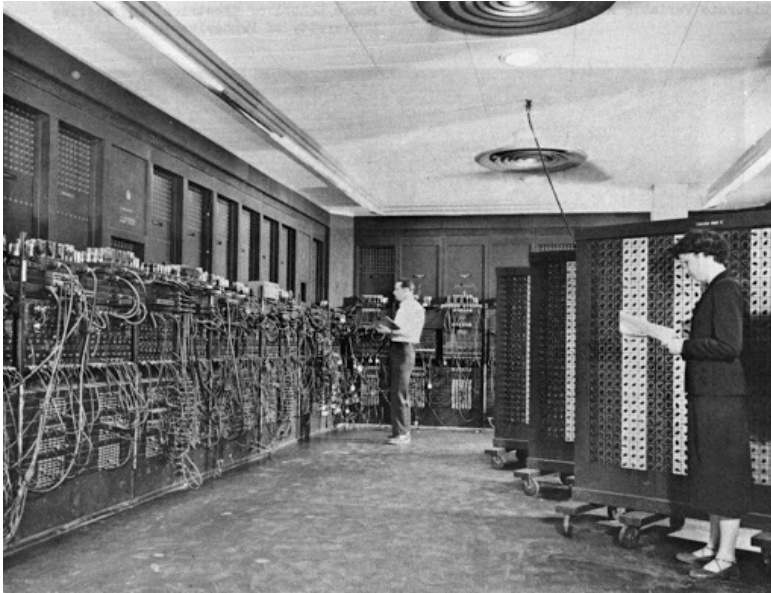
- Giống nhau với tất cả các loại máy tính
- **Bộ xử lý trung tâm** (Central Processing Unit – CPU)
  - Điều khiển hoạt động của máy tính và xử lý dữ liệu
- **Bộ nhớ chính** (Main Memory)
  - Chứa các chương trình đang thực hiện
- **Hệ thống vào-ra** (Input/Output)
  - Trao đổi thông tin giữa máy tính với bên ngoài
- **Liên kết hệ thống** (System interconnection)
  - Kết nối và vận chuyển thông tin



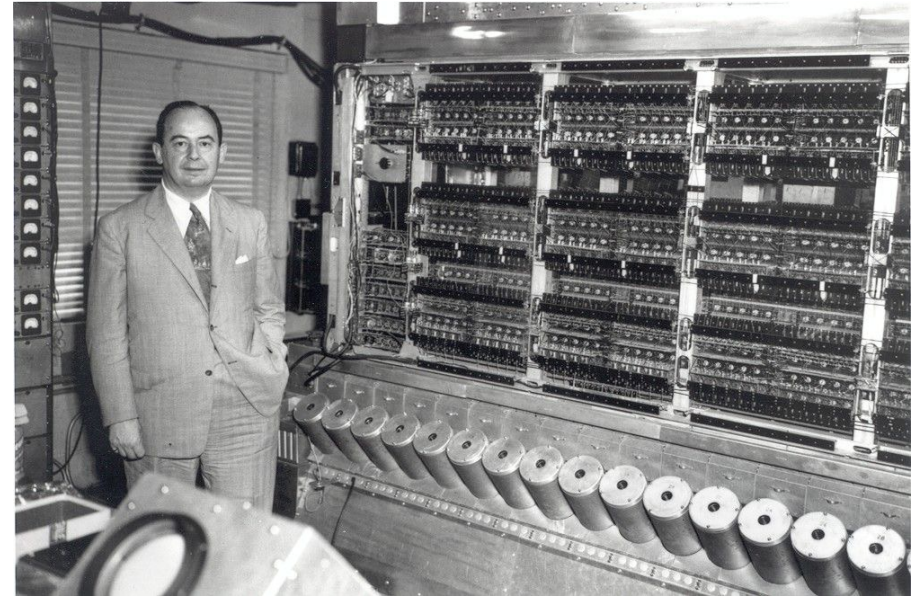
## 1.3. Sự tiến hóa của công nghệ máy tính

- Máy tính dùng đèn điện tử chân không (1950s)
  - Máy tính ENIAC: máy tính đầu tiên (1946)
  - Máy tính IAS: máy tính von Neumann (1952)
- Máy tính dùng transistors (1960s)
- Máy tính dùng vi mạch SSI, MSI và LSI (1970s)
  - SSI - Small Scale Integration
  - MSI - Medium Scale Integration
  - LSI - Large Scale Integration
- Máy tính dùng vi mạch VLSI (1980s)
  - VLSI - Very Large Scale Integration
- Máy tính dùng vi mạch ULSI (1990s-nay)
  - ULSI - Ultra Large Scale Integration

# Máy tính đầu tiên: ENIAC và IAS

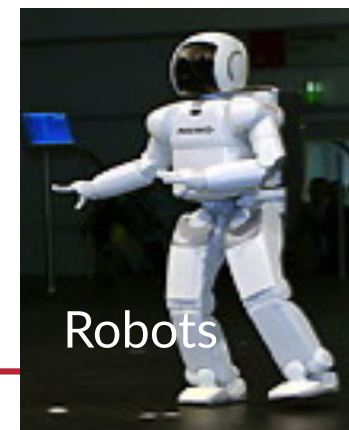
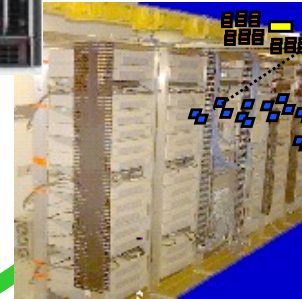
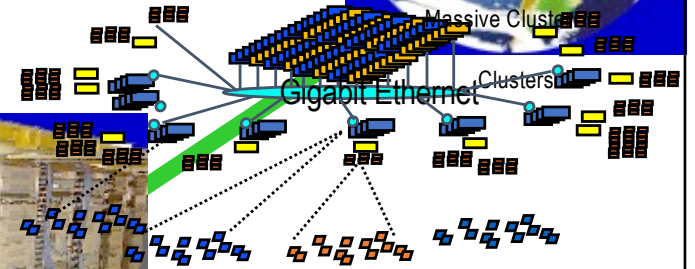
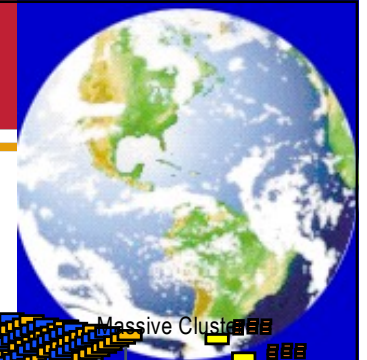


- Electronic Numerical Interpolator and Computer
- Dự án của Bộ Quốc phòng Mỹ
- Do John Mauchly ở đại học Pennsylvania thiết kế
- 30 tấn
- Xử lý theo số thập phân



- Thực hiện tại Princeton Institute for Advanced Studies
- Do John von Neumann thiết kế theo ý tưởng “stored program”
- Xử lý theo số nhị phân
- Trở thành mô hình cơ bản của máy tính

# Máy tính ngày nay



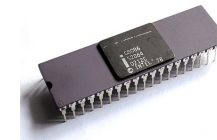
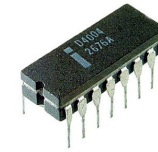
# Một số loại vi mạch số điển hình

- Bộ vi xử lý (Microprocessors)
  - Một hoặc một vài CPU được chế tạo trên một chip
- Vi mạch điều khiển tổng hợp (Chipset)
  - Vi mạch thực hiện các chức năng nối ghép các thành phần của máy tính với nhau
- Bộ nhớ bán dẫn (Semiconductor Memory)
  - ROM, RAM, Flash memory
- Hệ thống trên chip (SoC – System on Chip) hay Bộ vi điều khiển (Microcontrollers)
  - Tích hợp các thành phần chính của máy tính trên một chip vi mạch
  - Được dùng trong smartphone, tablet và máy tính nhúng



# Sự phát triển của bộ vi xử lý

- 1971: bộ vi xử lý 4-bit Intel 4004
- 1972: các bộ xử lý 8-bit
- 1978: các bộ xử lý 16-bit
  - Máy tính cá nhân IBM-PC ra đời năm 1981
- 1985: các bộ xử lý 32-bit
- 2001: các bộ xử lý 64-bit
- 2006: các bộ xử lý đa lõi (multicores)
  - Nhiều CPU trên 1 chip



## 1.4. Hiệu năng máy tính

- Định nghĩa hiệu năng P (Performance):

Hiệu năng =  $1/(\text{thời gian thực hiện})$

hay là:  $P = 1/t$

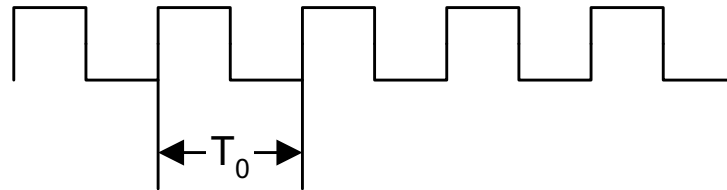
“Máy tính A nhanh hơn máy B k lần”

$$P_A / P_B = t_B / t_A = k$$

- Ví dụ: Thời gian chạy chương trình:
  - 10s trên máy A, 15s trên máy B
  - $t_B / t_A = 15s / 10s = 1.5$
  - Vậy máy A nhanh hơn máy B 1.5 lần

# Tốc độ xung nhịp của CPU

- Về mặt thời gian, CPU hoạt động theo một xung nhịp (clock) có tốc độ xác định



- **Chu kỳ xung nhịp**  $T_0$  (Clock period): thời gian của một chu kỳ
- **Tốc độ xung nhịp**  $f_0$  (Clock rate) hay là **Tần số xung nhịp**: số chu kỳ trong 1s
  - $f_0 = 1/T_0$
- Ví dụ: Bộ xử lý có  $f_0 = 4\text{GHz} = 4 \times 10^9 \text{Hz}$ 
  - $T_0 = 1/(4 \times 10^9) = 0.25 \times 10^{-9} \text{s} = 0.25 \text{ns}$



# Thời gian thực hiện của CPU

- Để đơn giản, ta xét thời gian CPU thực hiện chương trình (CPU time):

Thời gian thực hiện của CPU =

Số chu kỳ xung nhịp x Thời gian một chu kỳ

$$t_{CPU} = n \times T_0 = \frac{n}{f_0}$$

trong đó:  $n$  là số chu kỳ xung nhịp

- Hiệu năng được tăng lên bằng cách:
  - Giảm số chu kỳ xung nhịp  $n$
  - Tăng tốc độ xung nhịp  $f_0$

# Ví dụ

- Hai máy tính A và B cùng chạy một chương trình
- Máy tính A:
  - Tốc độ xung nhịp của CPU:  $f_A = 2\text{GHz}$
  - Thời gian CPU thực hiện chương trình:  $t_A = 10\text{s}$
- Máy tính B:
  - Thời gian CPU thực hiện chương trình:  $t_B = 6\text{s}$
  - Số chu kỳ xung nhịp khi chạy chương trình trên máy B ( $n_B$ ) bằng 1.2 lần số chu kỳ xung nhịp khi chạy chương trình trên máy A ( $n_A$ )
- Hãy xác định tốc độ xung nhịp cần thiết cho máy B ( $f_B$ )?

## Ví dụ (tiếp)

Ta có:  $t = \frac{n}{f}$

Số chu kỳ xung nhịp khi chạy chương trình trên máy A:

$$n_A = t_A \times f_A = 10s \times 2GHz = 20 \times 10^9$$

Số chu kỳ xung nhịp khi chạy chương trình trên máy B:

$$n_B = 1.2 \times n_A = 24 \times 10^9$$

Vậy: tốc độ xung nhịp cần thiết cho máy B:

$$f_B = \frac{n_B}{t_B} = \frac{24 \times 10^9}{6} = 4 \times 10^9 Hz = 4GHz$$

# Số lệnh và số chu kỳ trên một lệnh

Số chu kỳ xung nhịp của chương trình:

**Số chu kỳ = Số lệnh của chương trình x Số chu kỳ trên một lệnh**

$$n = IC \times CPI$$

- $n$  - số chu kỳ xung nhịp
- $IC$  - số lệnh của chương trình (Instruction Count)
- $CPI$  - số chu kỳ trên một lệnh (Cycles per Instruction)

Vậy thời gian thực hiện của CPU:

$$t_{CPU} = IC \times CPI \times T_0 = \frac{IC \times CPI}{f_0}$$

Trong trường hợp các lệnh khác nhau có CPI khác nhau, cần tính CPI trung bình

# Ví dụ

- Hai máy tính A và B có cùng kiến trúc tập lệnh
- Máy tính A có:
  - Chu kỳ xung nhịp:  $T_A = 250\text{ps}$
  - Số chu kỳ/ lệnh trung bình:  $\text{CPI}_A = 2.0$
- Máy tính B:
  - Chu kỳ xung nhịp:  $T_B = 500\text{ps}$
  - Số chu kỳ/ lệnh trung bình:  $\text{CPI}_B = 1.2$
- Hãy xác định máy nào nhanh hơn và nhanh hơn bao nhiêu ?

## Ví dụ (tiếp)

Ta có:  $t_{CPU} = IC \times CPI_{TB} \times T_0$

Hai máy cùng kiến trúc tập lệnh, vì vậy số lệnh của cùng một chương trình trên hai máy là bằng nhau:

$$IC_A = IC_B = IC$$

Thời gian thực hiện chương trình đó trên máy A và máy B:

$$t_A = IC_A \times CPI_A \times T_A = IC \times 2.0 \times 250 ps = IC \times 500 ps$$

$$t_B = IC_B \times CPI_B \times T_B = IC \times 1.2 \times 500 ps = IC \times 600 ps$$

Từ đó ta có:

$$\frac{t_B}{t_A} = \frac{IC \times 600 ps}{IC \times 500 ps} = 1.2$$

Kết luận: máy A nhanh hơn máy B 1.2 lần

# CPI trung bình

Nếu loại lệnh khác nhau có số chu kỳ khác nhau, ta có tổng số chu kỳ:

$$n = \sum_{i=1}^K (CPI_i \times IC_i)$$

Vậy CPI trung bình:

$$CPI_{TB} = \frac{n}{IC} = \frac{1}{IC} \sum_{i=1}^K (CPI_i \times IC_i)$$

# Ví dụ

Cho bảng chỉ ra các dãy lệnh sử dụng các lệnh thuộc các loại A, B, C. Tính CPI trung bình?

Loại lệnh	A	B	C
CPI theo loại lệnh	1	2	3
IC trong dãy lệnh 1	20	10	20
IC trong dãy lệnh 2	40	10	10



# Ví dụ

Cho bảng chỉ ra các dãy lệnh sử dụng các lệnh thuộc các loại A, B, C. Tính CPI trung bình?

Loại lệnh	A cộng	B trừ	C nhân
CPI theo loại lệnh	1	2	3
IC trong dãy lệnh 1	20	10	20
IC trong dãy lệnh 2	40	10	10

■ Dãy lệnh 1: Số lệnh = 50

■ Số chu kỳ =

$$= 1 \times 20 + 2 \times 10 + 3 \times 20 = 100$$

$$\text{CPI}_{\text{TB}} = 100/50 = 2.0$$

■ Dãy lệnh 2: Số lệnh = 60

■ Số chu kỳ =

$$= 1 \times 40 + 2 \times 10 + 3 \times 10 = 90$$

$$\text{CPI}_{\text{TB}} = 90/60 = 1.5$$

Câu hỏi:  $y = x * 4$  hay  $y = x * 3$  hay  $y = x + x + x$  nhanh hơn?



# Tóm tắt về Hiệu năng

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

Thời gian thực hiện chương trình của CPU =  
= Số lệnh của chương trình x Số chu kỳ/lệnh x Số giây của một chu kỳ

$$t_{CPU} = IC \times CPI \times T_0 = \frac{IC \times CPI}{f_0}$$

- Hiệu năng phụ thuộc vào:
  - Thuật giải
  - Ngôn ngữ lập trình
  - Chương trình dịch
  - Kiến trúc tập lệnh
  - Phần cứng

# MIPS như là thước đo hiệu năng

- MIPS: Millions of Instructions Per Second  
Số triệu lệnh trên 1 giây

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}} \times 10^6 = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

$$\text{MIPS} = \frac{f_0}{\text{CPI} \times 10^6}$$

$$\text{CPI} = \frac{f_0}{\text{MIPS} \times 10^6}$$

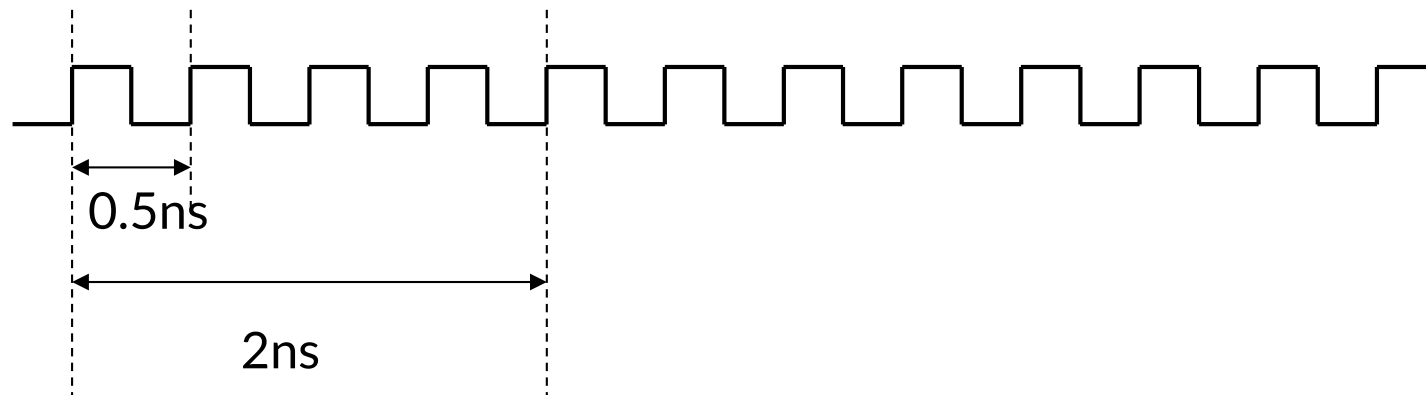
# Ví dụ

Tính MIPS của bộ xử lý với:  
clock rate = 2GHz và CPI = 4

# Ví dụ

Tính MIPS của bộ xử lý với:

clock rate = 2GHz và CPI = 4



- Chu kỳ  $T_0 = 1/(2 \times 10^9) = 0.5\text{ns}$
- $\text{CPI} = 4 \rightarrow$  thời gian thực hiện 1 lệnh =  $4 \times 0.5\text{ns} = 2\text{ns}$
- Số lệnh thực hiện trong 1s =  $(10^9\text{ns})/(2\text{ns}) = 5 \times 10^8$  lệnh
- Vậy bộ xử lý thực hiện được 500 MIPS

# Ví dụ

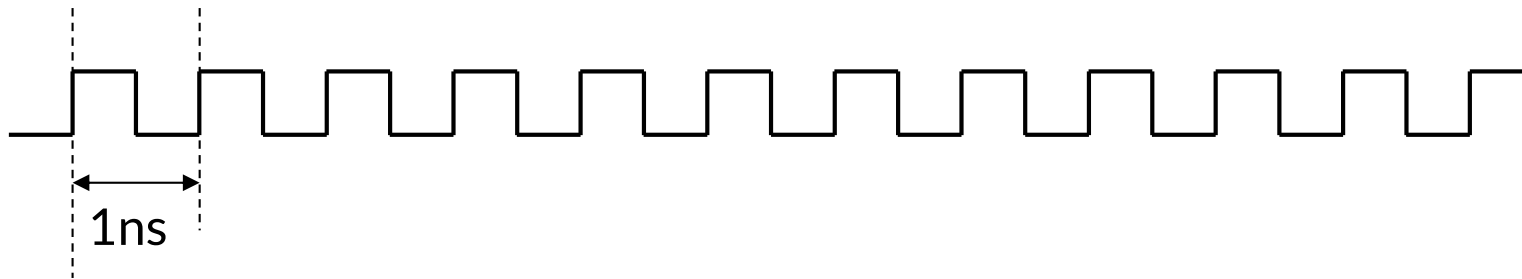
Tính CPI của bộ xử lý với:

clock rate = 1GHz và 400 MIPS

# Ví dụ

Tính CPI của bộ xử lý với:

clock rate = 1GHz và 400 MIPS



- Chu kỳ  $T_0 = 1/10^9 = 1\text{ns}$
- Số lệnh thực hiện trong 1 s là  $400\text{MIPS} = 4 \times 10^8$  lệnh
- Thời gian thực hiện 1 lệnh =  $1/(4 \times 10^8)\text{s} = 2.5\text{ns}$
- Vậy ta có:  $\text{CPI} = 2.5/1 = 2.5$

# MFLOPS

- Sử dụng cho các hệ thống tính toán lớn
- Millions of Floating Point Operations per Second
- Số triệu phép toán số dấu phẩy động trên một giây

$$\text{MFLOPS} = \frac{\text{Executed floating point operations}}{\text{Execution time} \times 10^6}$$

GFLOPS ( $10^9$ )

TFLOPS ( $10^{12}$ )

PFLOPS ( $10^{15}$ )



Hết chương 1