



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

KIẾN TRÚC MÁY TÍNH

Computer Architecture

Course ID: IT3030, IT3034, IT3283

Version: CA.RISCV.2025



© Nguyễn Kim Khánh

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

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

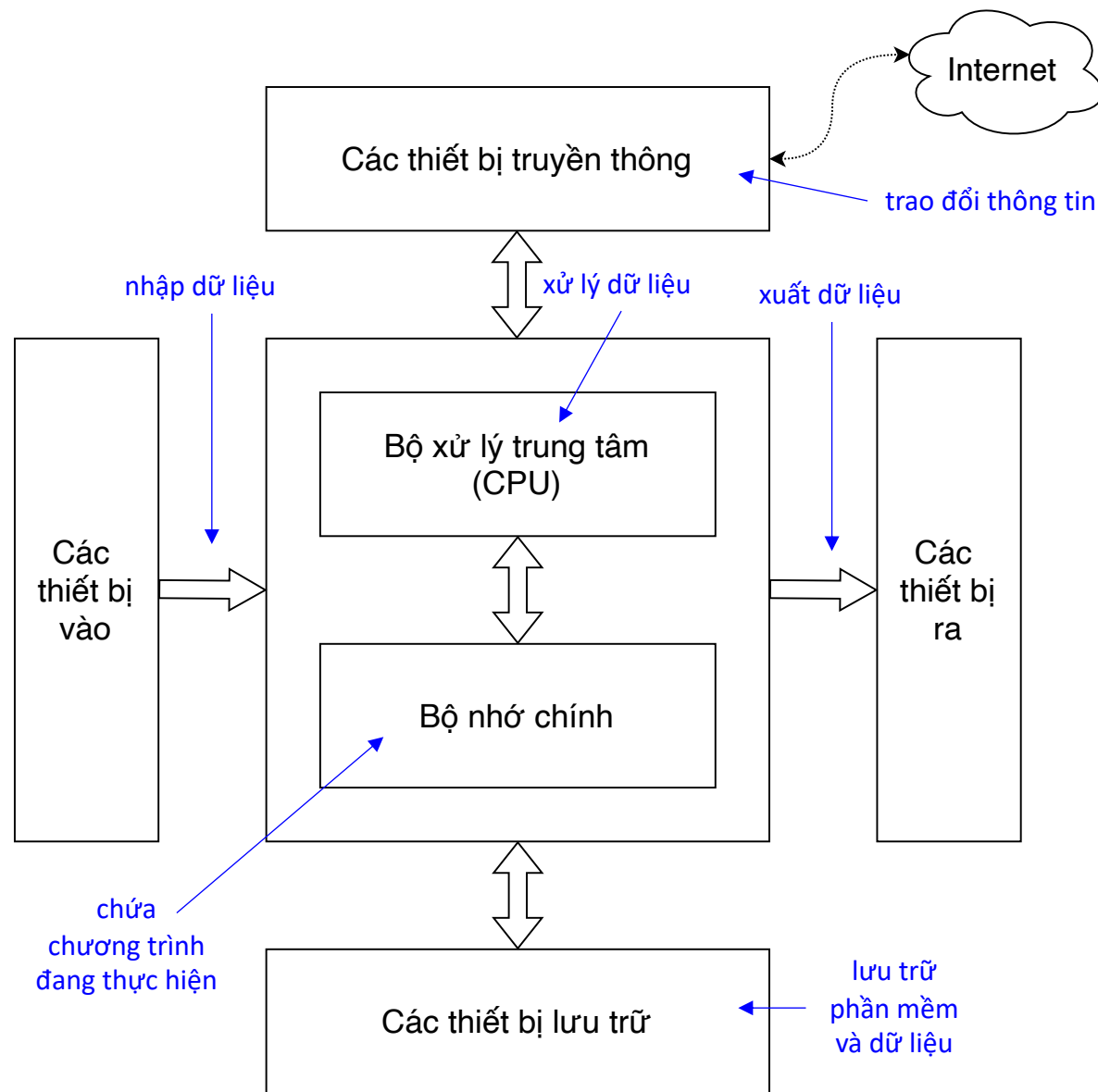


1.1. Máy tính và phân loại máy tính

- **Máy tính (Computer)** là thiết bị điện tử hoạt động theo chương trình, có thể nhập dữ liệu (data), lưu trữ dữ liệu, xử lý dữ liệu, và xuất ra thông tin (information).
- **Chương trình (Program)** là dãy các lệnh được lập trình sẵn và được lưu trong bộ nhớ để điều khiển máy tính thực hiện theo.

→ 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
- 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)
 - Thực chất là "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 các máy chủ với nhau dùng cho các mục đích khác nhau
 - Supercomputers: Một dạng WSC dùng cho siêu tính toán trong khoa học và kỹ thuật
 - Hàng triệu đến hàng trăm triệu USD



1.2. Kiến trúc máy tính

- Nghiên cứu máy tính theo ba khía cạnh:
 - **Kiến trúc tập lệnh** (Instruction Set Architecture): nghiên cứu máy tính theo cách nhìn của người lập trình, giao diện giữa phần cứng và phần mềm
 - **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 ở mức 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 tập trung vào **Kiến trúc tập lệnh** và **Tổ chức máy tính**



Nội dung chính sẽ được học

- Cách các chương trình được dịch sang ngôn ngữ máy
 - Và cách phần cứng thực thi chúng
- Giao diện giữa phần cứng và phần mềm (tương tác)
- Những yếu tố quyết định hiệu năng của chương trình
 - Và cách có thể cải thiện hiệu năng
- Cách các nhà thiết kế phần cứng nâng cao hiệu năng
- Xử lý song song là gì



Các yếu tố quyết định hiệu năng

- **Thuật toán**

→ Quyết định số phép toán được thực hiện.

- **Ngôn ngữ lập trình, trình biên dịch, kiến trúc**

→ Quyết định số lệnh máy được thực hiện cho mỗi phép toán.

- **Bộ xử lý và hệ thống bộ nhớ**

→ Quyết định tốc độ thực hiện các lệnh.

- **Hệ thống vào/ra (bao gồm cả hệ điều hành)**

→ Quyết định tốc độ thực hiện các thao tác vào/ra.

7 ý tưởng lớn trong kiến trúc máy tính

- Sử dụng trừu tượng để đơn giản hóa thiết kế (*Use abstraction to simplify design*)
- Tối ưu hóa cho trường hợp thường gặp (*Make the common case fast*)
- Hiệu năng nhờ song song (*Performance via parallelism*)
- Hiệu năng nhờ ống dẫn (*Performance via pipelining*)
- Hiệu năng nhờ dự đoán (*Performance via prediction*)
- Phân cấp bộ nhớ (*Hierarchy of memories*)
- Độ tin cậy nhờ dự phòng (*Dependability via redundancy*)



ABSTRACTION



COMMON CASE FAST



PARALLELISM



PIPELINING



PREDICTION



HIERARCHY



DEPENDABILITY

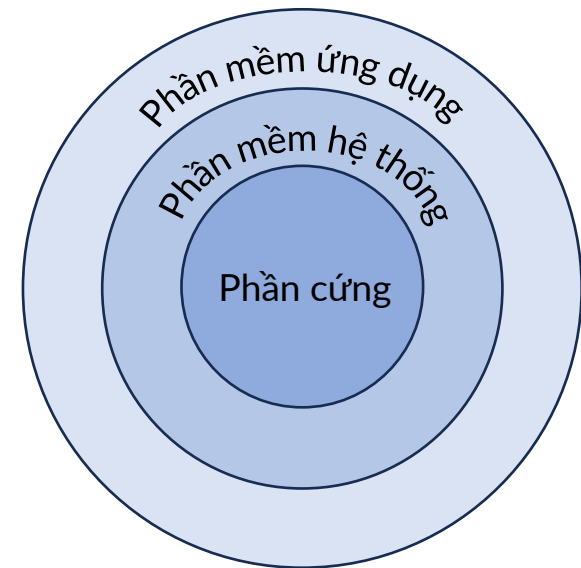
Các kiến trúc tập lệnh thông dụng

- Kiến trúc Intel x86
 - Từ 1978
 - Sử dụng phổ biến trong Laptop, Desktop, Servers
- Kiến trúc ARM
 - Từ 1985
 - Sử dụng trong Smartphones, Tablets, ...
- Kiến trúc RISC-V : kiến trúc tập lệnh mở
 - Từ 2010 tại University of California, Berkeley
 - RISC-V Foundation (2015)
 - RISC-V International Association (2020) hiện tại đặt ở Thụy Sĩ
 - ~ 4500 thành viên từ 70 quốc gia
 - Miễn phí bản quyền
 - Kiến trúc cho tương lai ?



Phân lớp máy tính

- Phần mềm ứng dụng
 - Được viết theo ngôn ngữ bậc cao
- Phần mềm hệ thống
 - Hệ điều hành (Operating Systems)
 - Lập lịch cho các nhiệm vụ và chia sẻ tài nguyên
 - Quản lý bộ nhớ và lưu trữ
 - Điều khiển vào-ra
 - Chương trình dịch (Compilers): dịch mã ngôn ngữ bậc cao thành mã máy
- 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
- Phụ thuộc kiến trúc tập lệnh cụ thể
- Mô tả lệnh dưới dạng text

▪ Ngôn ngữ máy

- Machine language
- Được mã hóa bằng số nhị phân

Ngôn ngữ
bậc cao
(C)

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

Compiler

Hợp ngữ
(RISC-V)

```
swap:
    slli t0, a1, 2
    add  t0, a0, t0
    lw   t1, 0(t0)
    lw   t2, 4(t0)
    sw   t2, 0(t0)
    sw   t1, 4(t0)
    jalr zero, 0(ra)
```

Assembler

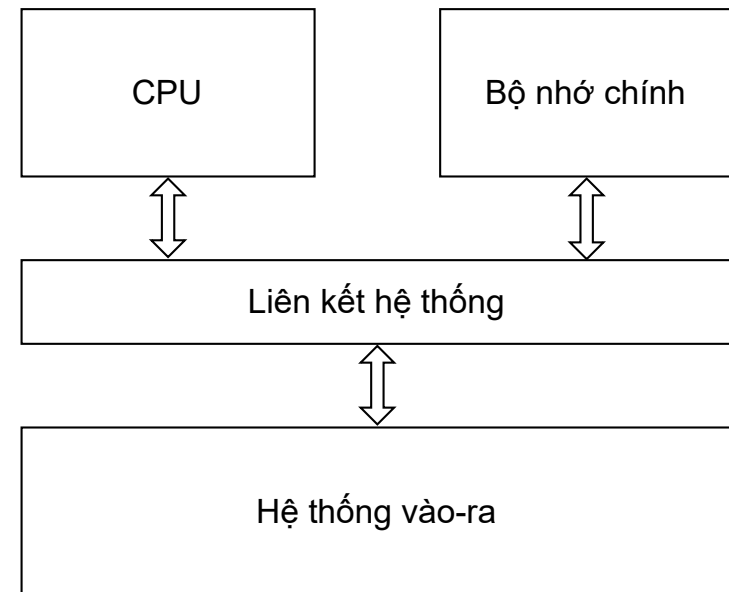
Mã máy
(RISC-V)

```
0000 0000 0010 0101 1001 0010 1001 0011
0000 0000 0101 0101 0000 0010 1011 0011
0000 0000 0000 0010 1010 0011 0000 0011
0000 0000 0100 0010 1010 0011 1000 0011
0000 0000 0111 0010 1010 0000 0010 0011
0000 0000 0110 0010 1010 0010 0010 0011
0000 0000 0000 0000 1000 0000 0110 0111
```



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

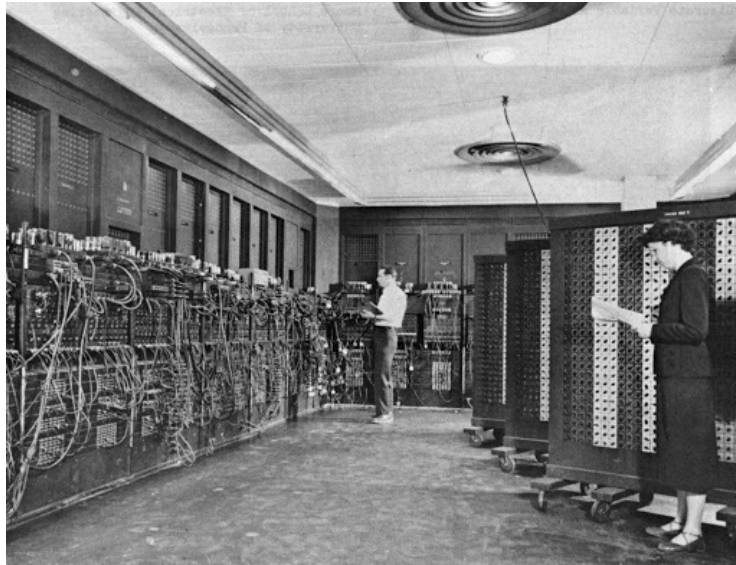


1.3. Sự tiến hóa của công nghệ máy tính

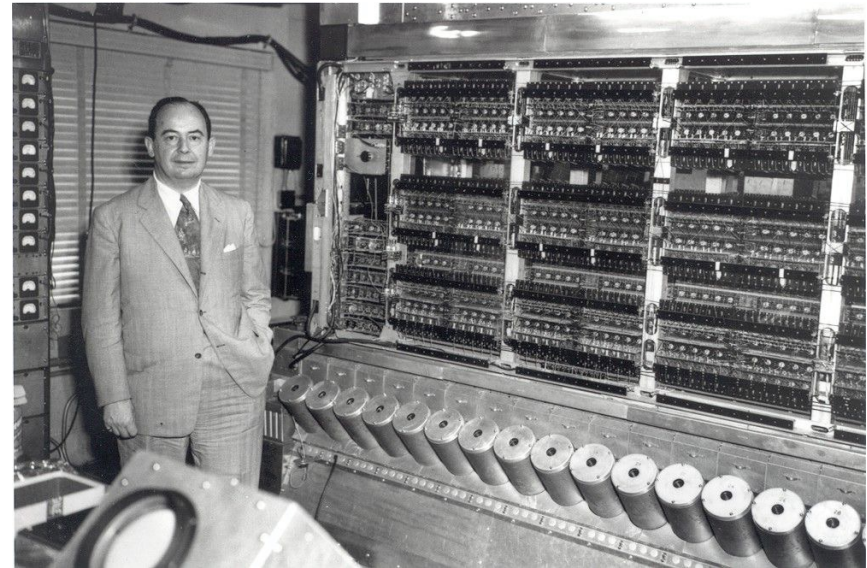
- Máy tính dùng đèn điện tử chân không (1950s)
 - Máy tính ENIAC: máy tính đầu tiên (1946)
 - Máy tính IAS: máy tính von Neumann (1952)
- Máy tính dùng transistors (1960s)
- Máy tính dùng IC (vi mạch) SSI, MSI và LSI (1970s)
 - SSI - Small Scale Integration
 - MSI - Medium Scale Integration
 - LSI - 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 GS 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

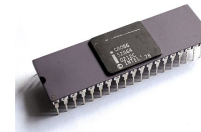
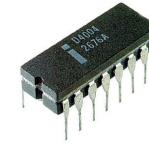
Một số loại vi mạch số điển hình (IC- Integrated Circuits)

- Bộ vi xử lý (Microprocessor)
 - Một hoặc nhiều 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
- Bộ xử lý đồ họa (GPU - Graphic Processing Unit)

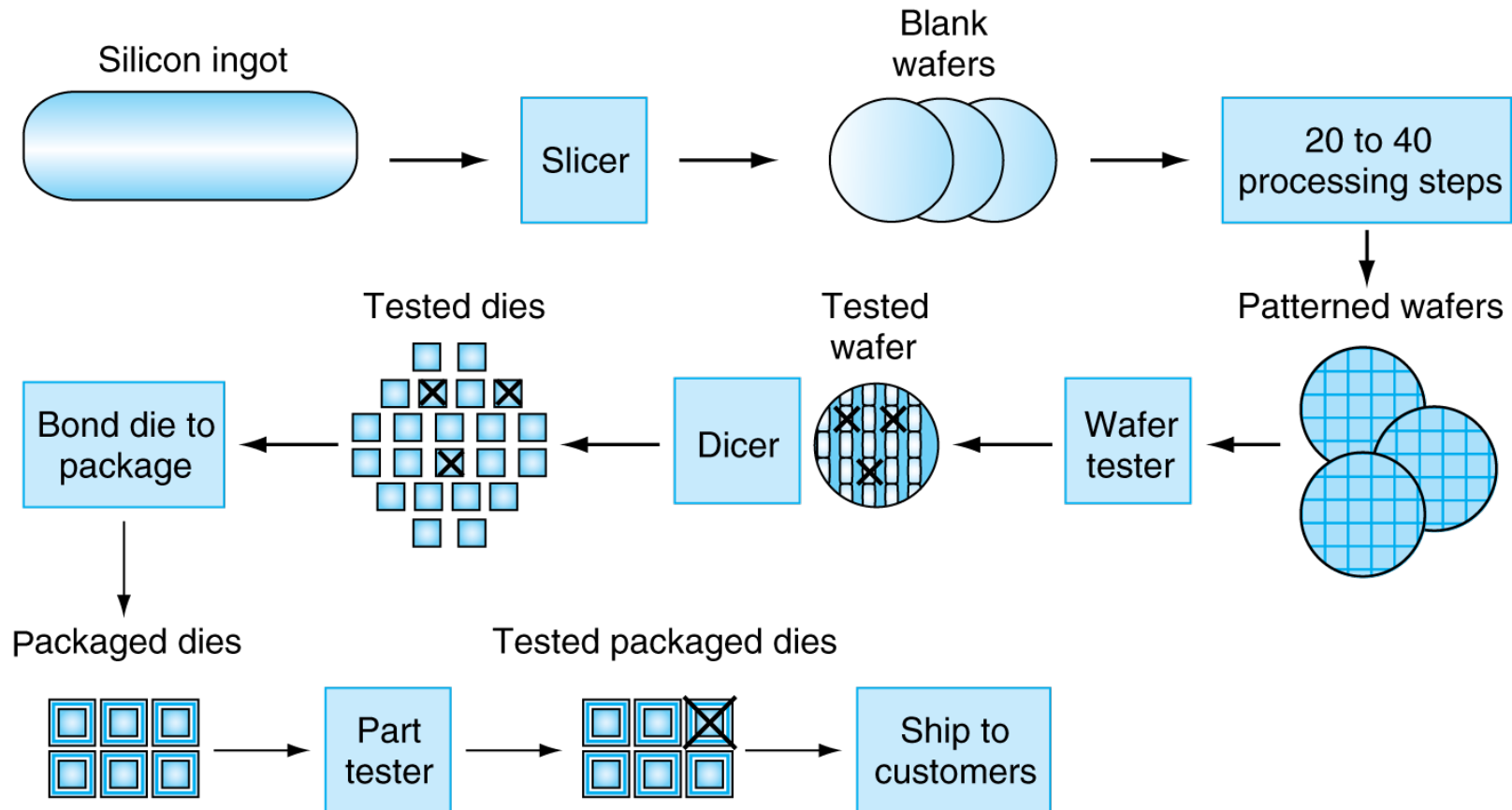


Sự phát triển của bộ vi xử lý

- 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

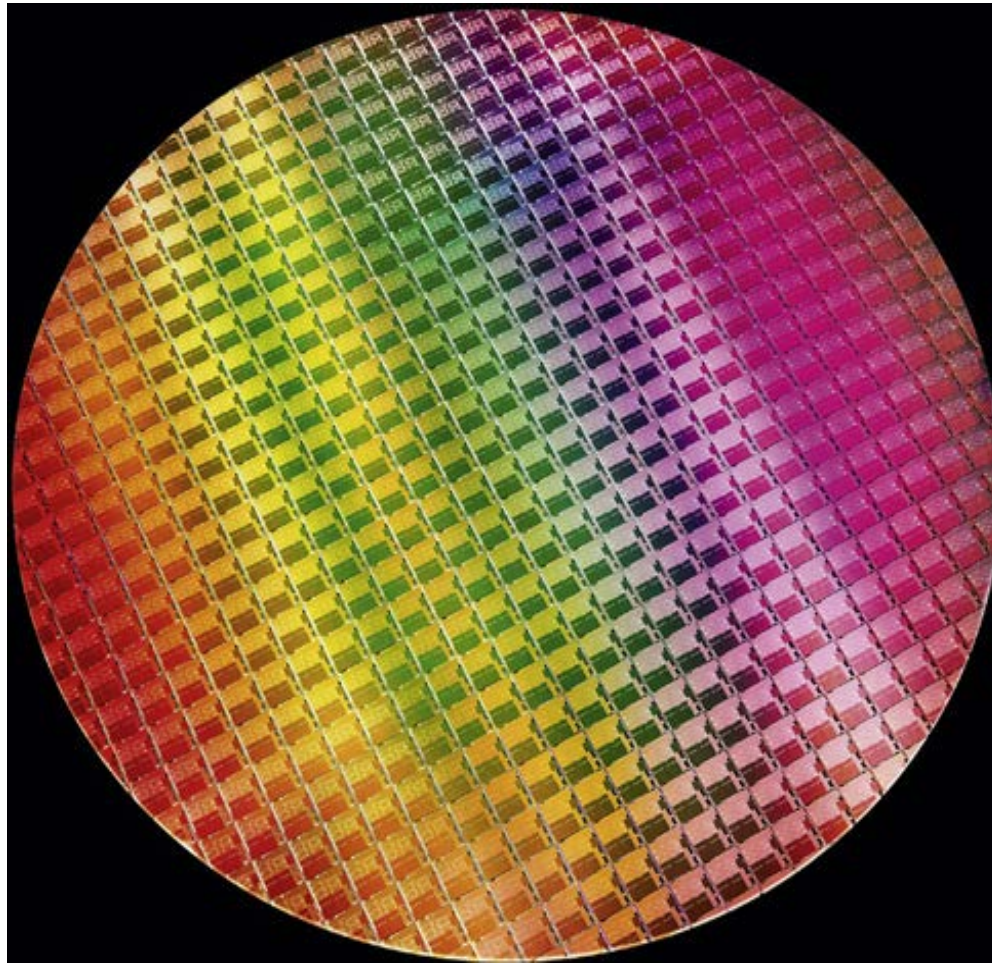


Qui trình sản xuất chip



Từ cát thành chip

10th Gen Intel Core Wafer



- 300mm wafer, 506 chips, 10nm technology
- Each chip is 11.4 x 10.7 mm

1.4. Hiệu năng máy tính

- Định nghĩa hiệu năng P (Performance):

$$\text{Hiệu năng} = \frac{1}{\text{Thời gian thực hiện}}$$

hay là: $P = \frac{1}{t}$

“Máy tính A nhanh hơn máy B k lần”

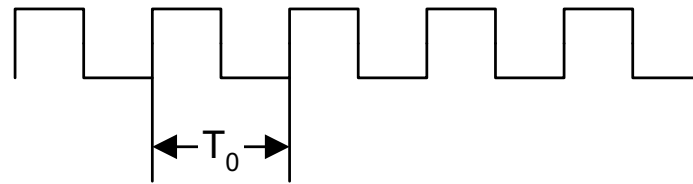
$$\frac{P_A}{P_B} = \frac{t_B}{t_A} = k$$

- 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



Tần số xung nhịp của CPU

- Mỗi bộ xử lý hoạt động theo một xung nhịp (clock) có tần số xác định



- Chu kỳ xung nhịp T_0 (Clock period): thời gian của một chu kỳ
- Tần số xung nhịp f_0 (Clock rate)
 - Còn được gọi là **Tốc độ xung nhịp** (Clock speed)
 - Số xung trong 1s, đo bằng đơn vị Hz
 - $f_0 = 1/T_0$
- 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 tần số xung nhịp f_0



Ví dụ

- Hai máy tính A và B cùng chạy một chương trình
- Máy tính A:
 - Tần số 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 tần số xung nhịp cần thiết cho máy B (f_B)?

Ví dụ

- Ta có: $t = \frac{n}{f_0}$
- 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:

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

- Tần số xung nhịp cần thiết cho máy B:

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



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:

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

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

- Từ đó ta có:

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

- Kết luận: máy A nhanh hơn máy B 1.2 lần



CPI trung bình

- Nếu các 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

- Dãy lệnh 1: Số lệnh = 50
 - Số chu kỳ =
 $= 1 \times 20 + 2 \times 10 + 3 \times 20 = 100$
 - $CPI_{TB} = 100/50 = 2.0$
- Dãy lệnh 2: Số lệnh = 60
 - Số chu kỳ =
 $= 1 \times 40 + 2 \times 10 + 3 \times 10 = 90$
 - $CPI_{TB} = 90/60 = 1.5$



Tóm tắt về Hiệu năng

$$CPUtime = \frac{Instructions}{Program} \times \frac{Clock\ cycles}{Instruction} \times \frac{Seconds}{Clock\ Cycle}$$

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 (Millions of Instructions Per Second)

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

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

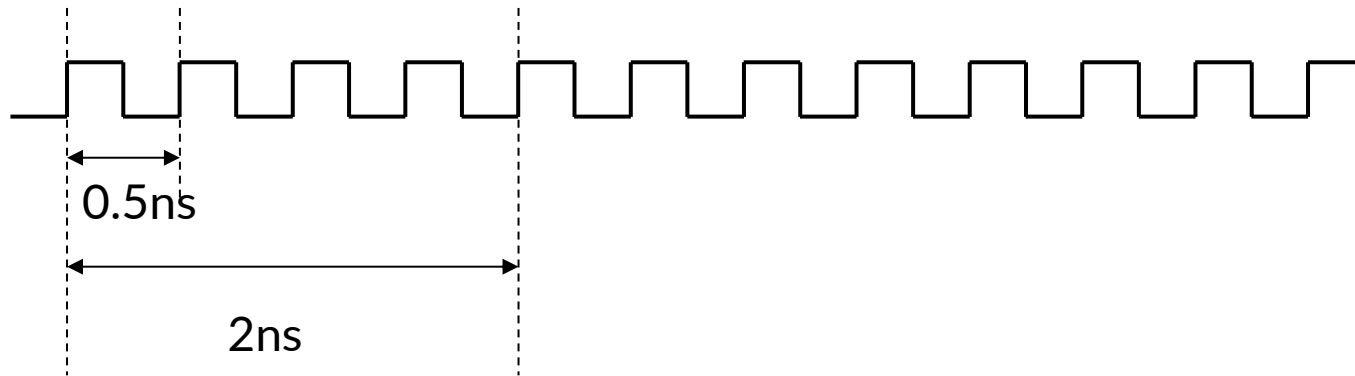
$$\text{CPI} = \frac{f_0}{\text{MIPS} \times 10^6}$$



Ví dụ

Tính MIPS của bộ xử lý với:

$$f_0 = 2\text{GHz và CPI} = 4$$

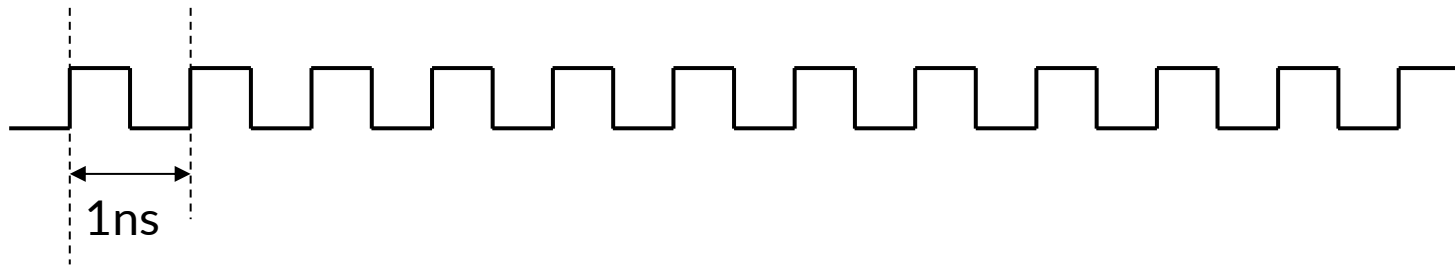


- Chu kỳ $T_0 = 1/(2 \times 10^9) = 0.5\text{ns}$
- $\text{CPI} = 4 \rightarrow$ thời gian thực hiện 1 lệnh $= 4 \times 0.5\text{ns} = 2\text{ns}$
- Số lệnh thực hiện trong 1s $= (10^9\text{ns})/(2\text{ns}) = 5 \times 10^8$ lệnh
- Vậy bộ xử lý thực hiện được 500 MIPS

Ví dụ

Tính CPI của bộ xử lý với:

$f_0 = 1\text{GHz}$ và 400 MIPS



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

Hạn chế của tham số MIPS

- **Chỉ đo tốc độ thực thi lệnh**, không phản ánh khả năng của tập lệnh.
- **Không so sánh được giữa các ISA khác nhau**, vì số lệnh thực thi sẽ khác.
- **Thay đổi theo từng chương trình** trên cùng một máy tính → không có chỉ số MIPS duy nhất.
- **Phụ thuộc vào CPI và xung nhịp (clock rate)**.
- **Có thể độc lập với hiệu năng thực sự**: một chương trình có nhiều lệnh hơn nhưng lệnh nhanh hơn → MIPS tăng/giảm không phản ánh đúng hiệu năng.

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