
KIẾN TRÚC MÁY TÍNH

ET4270

TS. Nguyễn Đức Minh

*[Adapted from Computer Organization and Design, 4th Edition, Patterson & Hennessy, © 2008, MK]
[Adapted from Computer Architecture lecture slides, Mary Jane Irwin, © 2008, PennState University]*

Tổ chức lớp

Số tín chỉ 3 (3-1-1-6)

Giảng viên TS. Nguyễn Đức Minh

Văn phòng C9-401

Email minhnd1@gmail.com

Website <https://sites.google.com/site/fethutca/home>

Sách *Computer Org and Design*, 3rd Ed., Patterson & Hennessy, ©2007
 Digital Design and Computer Architecture, David Money Harris

Thí nghiệm 3 bài

Bài tập Theo chương, đề bài và bài giải xem trên trang web

Điểm số

Bài thi cuối kỳ	70%
------------------------	------------

Bài thi giữa kỳ	30%
------------------------	------------

Bài tập	20%
---------	-----

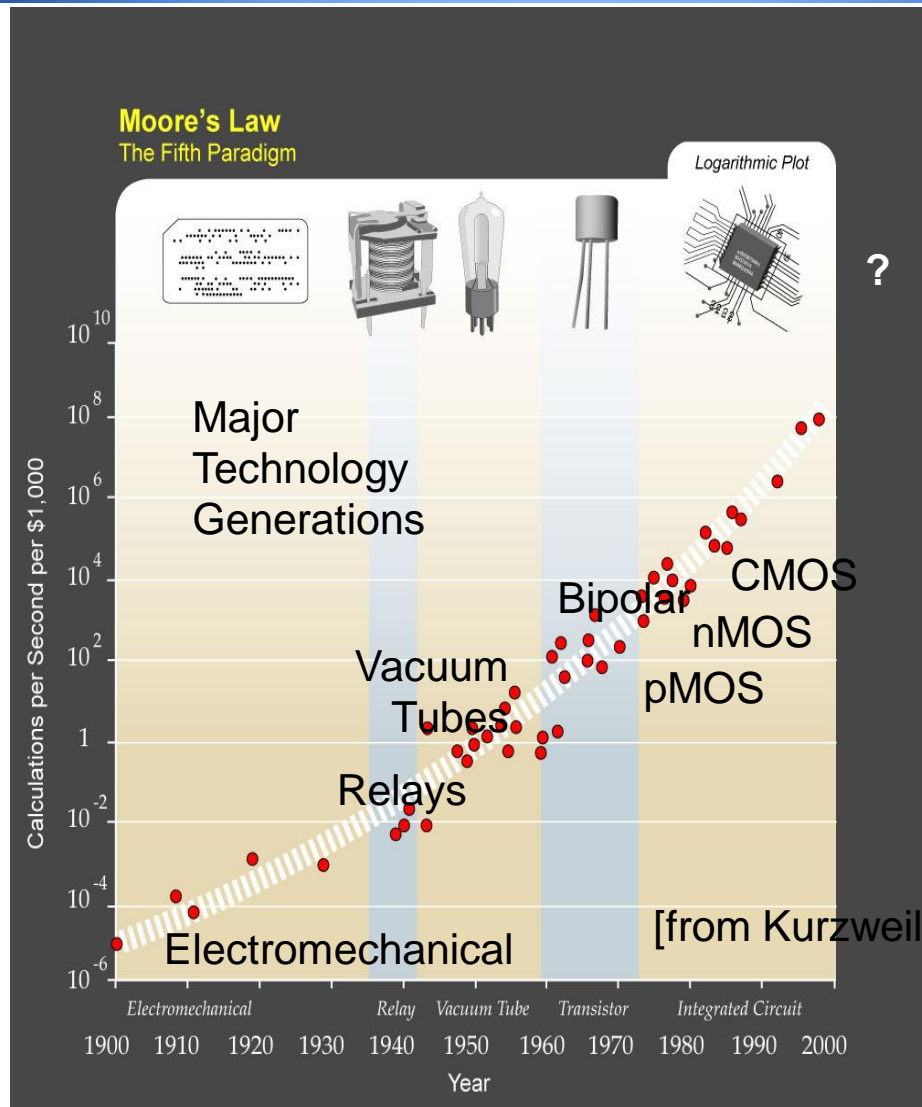
Thí nghiệm	10%
------------	-----

Lịch học

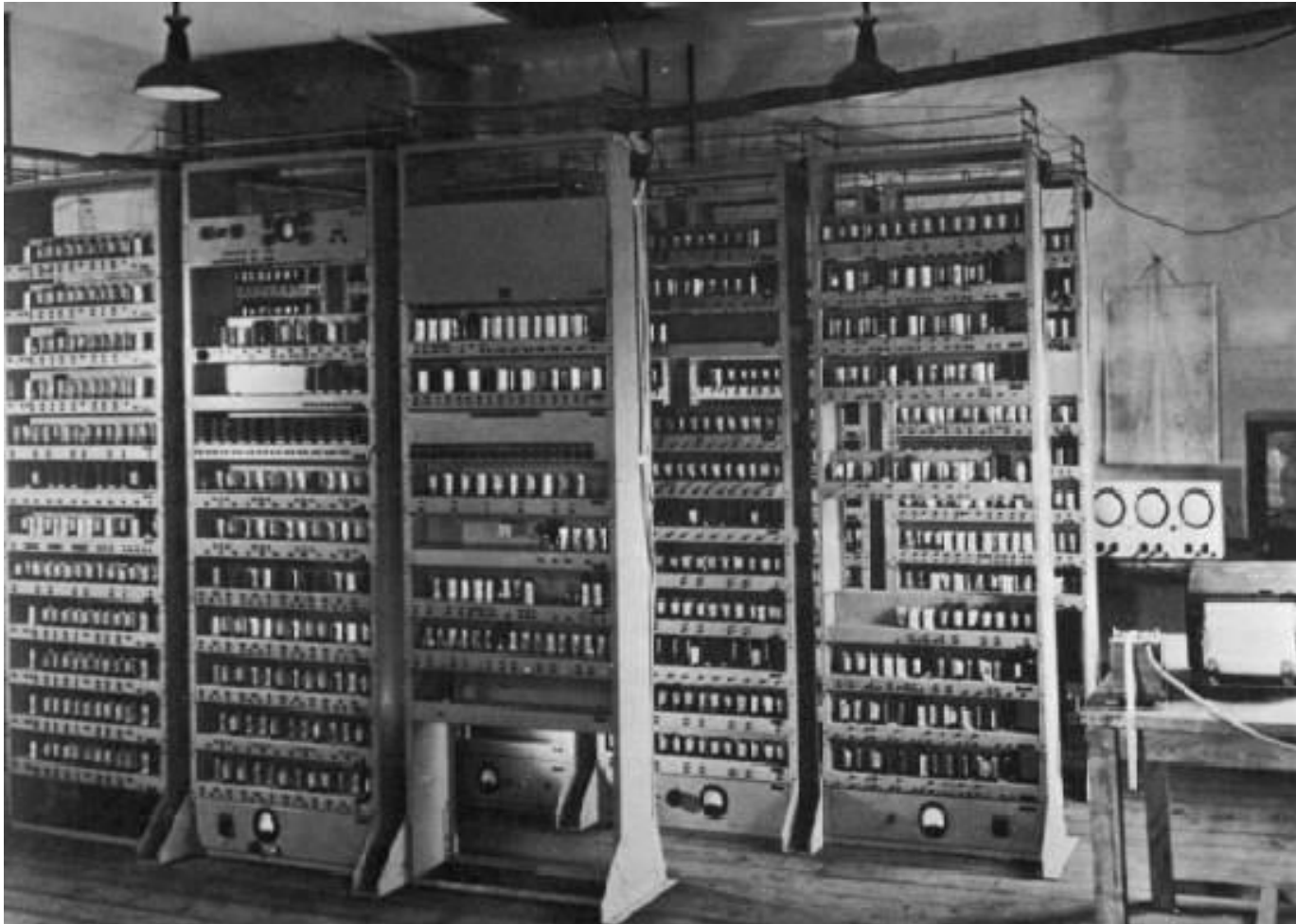
❖ Thời gian:

- ☐ Từ 14h00 đến 17h20
- ☐ 15 buổi x 135 phút / 1 buổi
- ☐ Thay đổi lịch (nghỉ, học bù) sẽ được thông báo trên website trước 2 ngày

Phát triển của công nghệ thông tin

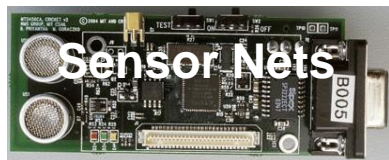
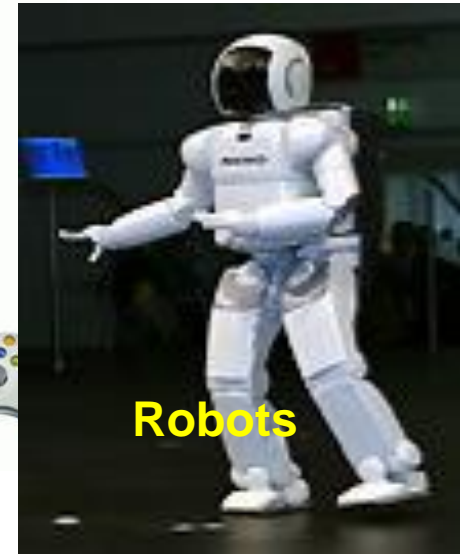


Bắt đầu



EDSAC, University of Cambridge, UK, 1949

Ngày nay



Máy tính có mặt khắp mọi nơi

Ứng dụng

❖ Phương tiện giao thông

- ☐ Khi máy tính trở nên rẻ hơn, nhỏ hơn và có hiệu suất cao hơn, nó được sử dụng trong ô tô, xe máy để tăng hiệu suất sử dụng nhiên liệu, giảm ô nhiễm, tăng độ an toàn.

❖ Điện thoại di động, thiết bị viễn thông

- ☐ Giúp con người giao tiếp dù ở bất kỳ đâu.

❖ Bản đồ gen

- ☐ Máy tính trở nên rẻ và mạnh hơn 10-100 lần so với cách đây 10 năm cho phép trang bị các máy tính để phân tích và ánh xạ bản đồ gen người.

❖ WWW

- ☐ Nhờ sự phổ biến của máy tính, các thiết bị mạng, Internet trở thành môi trường làm thế giới tràn ngập thông tin (thế giới thông tin).

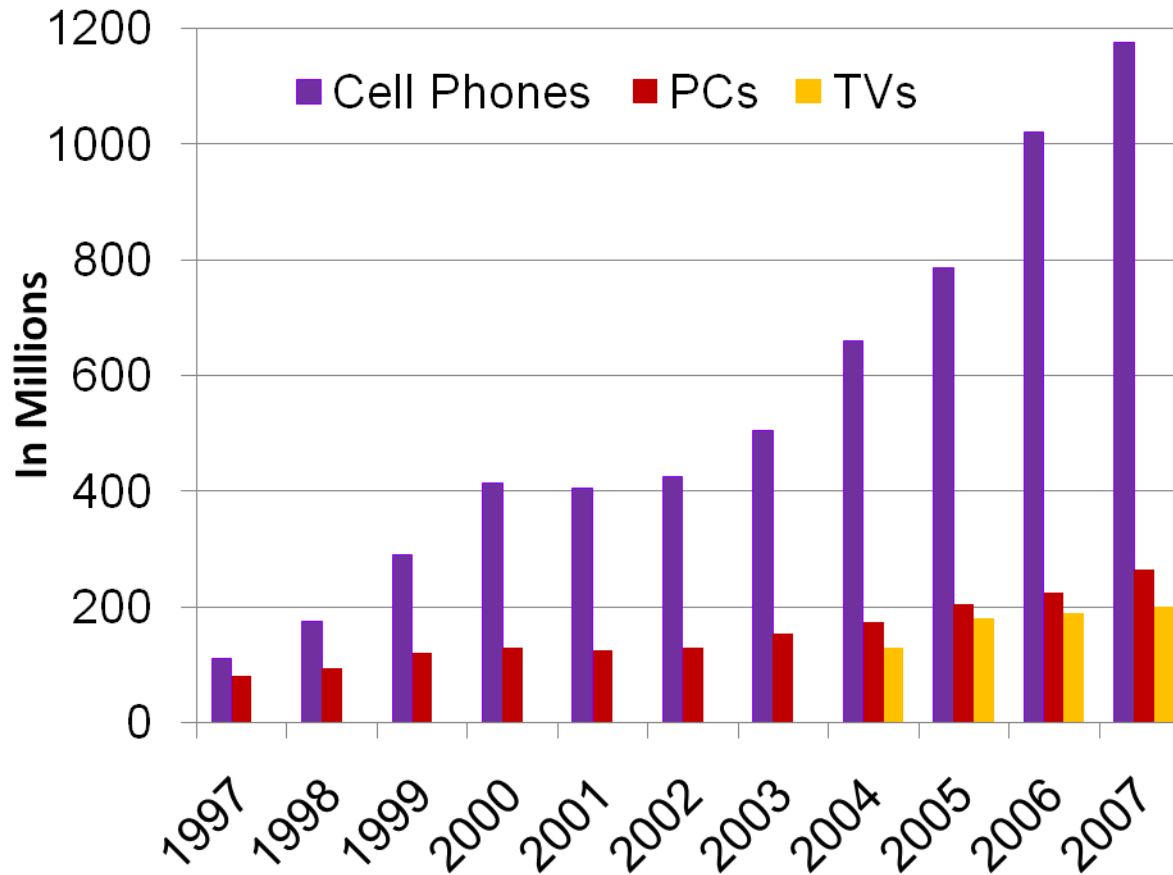
❖ Bộ tìm kiếm

- ☐ Google trở thành 1 động từ và 1 đế chế.

Các loại máy tính

- ❖ Máy tính để bàn (*eng, Desktop computers*)
 - ❑ Một người dùng; Chạy nhiều ứng dụng khác nhau; Đi kèm màn hình, bàn phím và chuột; Yêu cầu giá thành rẻ, hiệu năng cao
- ❖ Máy chủ (*eng, Servers*)
 - ❑ Nhiều người dùng đồng thời; Chạy các ứng dụng lớn; Truy cập qua mạng, Yêu cầu độ ổn định và an toàn cao.
- ❖ Siêu máy tính (*eng, Supercomputers*)
 - ❑ Chạy các ứng dụng khoa học và công nghệ cao cấp; Gồm hàng trăm/ngày bộ xử lý, bộ nhớ và bộ lưu trữ dung lượng lớn; Yêu cầu hiệu năng cao và có giá thành cao.
- ❖ Máy tính nhúng (*eng, Embedded computers (processors)*)
 - ❑ Máy tính nằm bên trong một thiết bị khác, chạy 1 ứng dụng xác định trước.

Tăng trưởng doanh số điện thoại di động



Tăng trưởng điện thoại di động >> Tăng trưởng máy tính để bàn

Đặc điểm của máy tính nhúng

- ❖ Ứng dụng trong nhiều lĩnh vực khác nhau
- ❖ Yêu cầu hiệu năng rất khác nhau
- ❖ Yêu cầu hiệu năng tối thiểu và vừa đủ. Ví dụ?
- ❖ Yêu cầu khắt khe về giá thành. Ví dụ?
- ❖ Yêu cầu khắt khe về năng lượng tiêu thụ. Ví dụ?
- ❖ Ít chấp nhận hỏng hóc. Ví dụ?

Mục tiêu môn học

Kiến thức về hệ thống máy tính:

- ☐ Giao diện giữa phần mềm và phần cứng
- ☐ Quá trình biên dịch chương trình phần mềm
- ☐ Cấu tạo và hoạt động của phần cứng máy tính
- ☐ Phương pháp đánh giá định lượng về hiệu năng máy tính
- ☐ Ảnh hưởng của các thành phần lên hiệu năng máy tính
- ❖ Kỹ sư phần mềm: tận dụng ưu điểm của phần cứng và lựa chọn phần cứng tối ưu
- ❖ Kỹ sư phần cứng: ảnh hưởng của phần cứng lên phần mềm

KIẾN TRÚC MÁY TÍNH

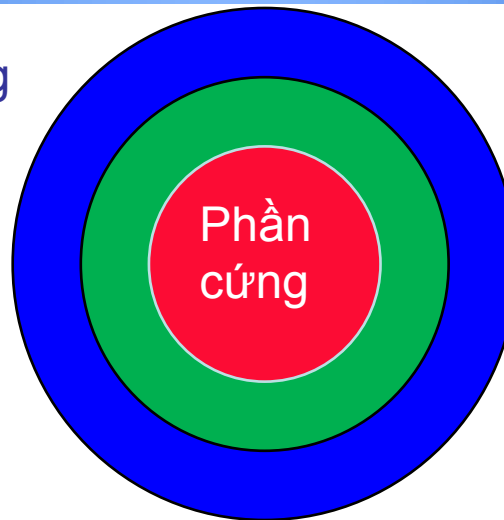
Thành phần cơ bản của máy tính

[Adapted from Computer Organization and Design, 4th Edition, Patterson & Hennessy, © 2008, MK]

[Adapted from Computer Architecture lecture slides, Mary Jane Irwin, © 2008, PennState University]

Phần mềm

Phần mềm ứng dụng



Phần mềm hệ thống

❖ Phần mềm hệ thống

- ❑ Hệ điều hành – giám sát, giao tiếp giữa phần cứng và phần mềm ứng dụng (như Linux, MacOS, Windows)
 - Điều khiển các hoạt động vào ra cơ bản
 - Cấp phát bộ nhớ
 - Cung cấp sự chia sẻ có bảo vệ giữa các ứng dụng
- ❑ Bộ biên dịch – chuyển đổi các chương trình ở ngôn ngữ bậc cao (như C, Java) thành các câu lệnh phần cứng có thể thực hiện

Trình biên dịch

❖ 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;
}
```

one-to-many

C compiler

❖ Assembly language program (for MIPS)

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

one-to-one

assembler

❖ Machine (object, binary) code (for MIPS)

```
000000 00000 00101 0001000010000000
000000 00100 00010 0001000000100000
, , ,
```

Ưu điểm của ngôn ngữ bậc cao

❖ Ngôn ngữ bậc cao

- ❑ Chương trình được viết ở ngôn ngữ tự nhiên và phù hợp với từng ứng dụng (Ví dụ: Fortran, Lisp, Java)
- ❑ Tăng năng suất lập trình viên – mã chương trình dễ hiểu, dễ gỡ lỗi, dễ kiểm tra
- ❑ Tăng khả năng bảo trì chương trình
- ❑ Chương trình độc lập với phần cứng sẽ thực hiện chương trình
- ❑ Chương trình được tối ưu hóa cho từng loại phần cứng nhờ các thuật toán tối ưu trong trình biên dịch

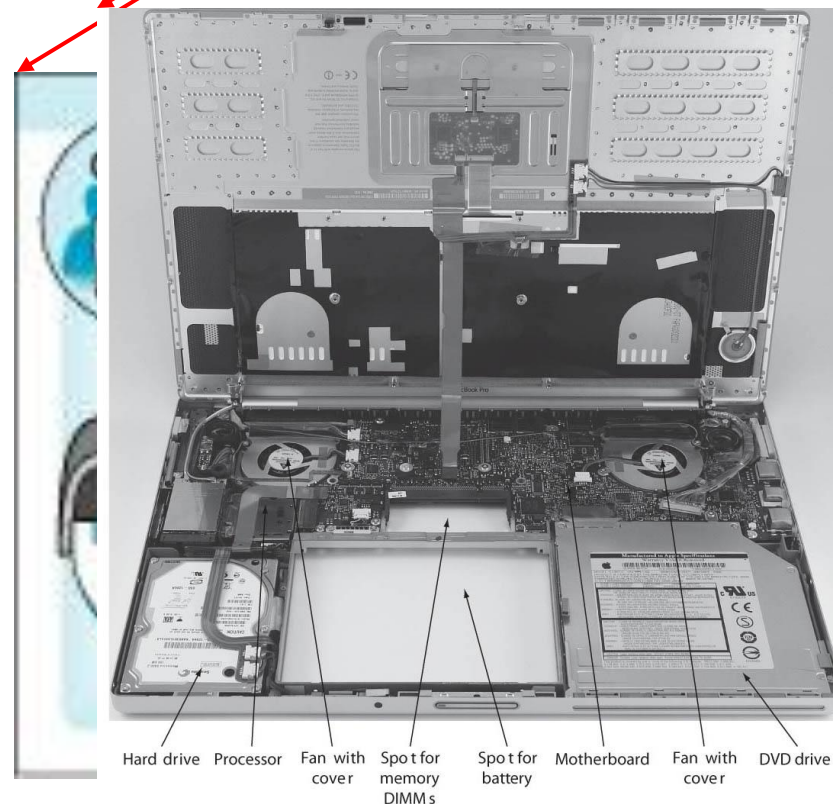
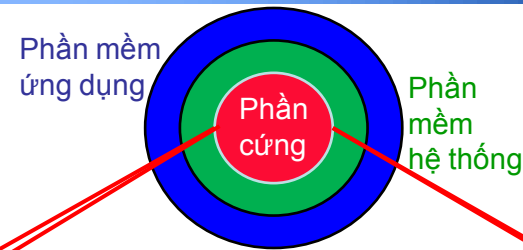
❖ Ít chương trình còn được phát triển bằng hợp ngữ

Phần cứng

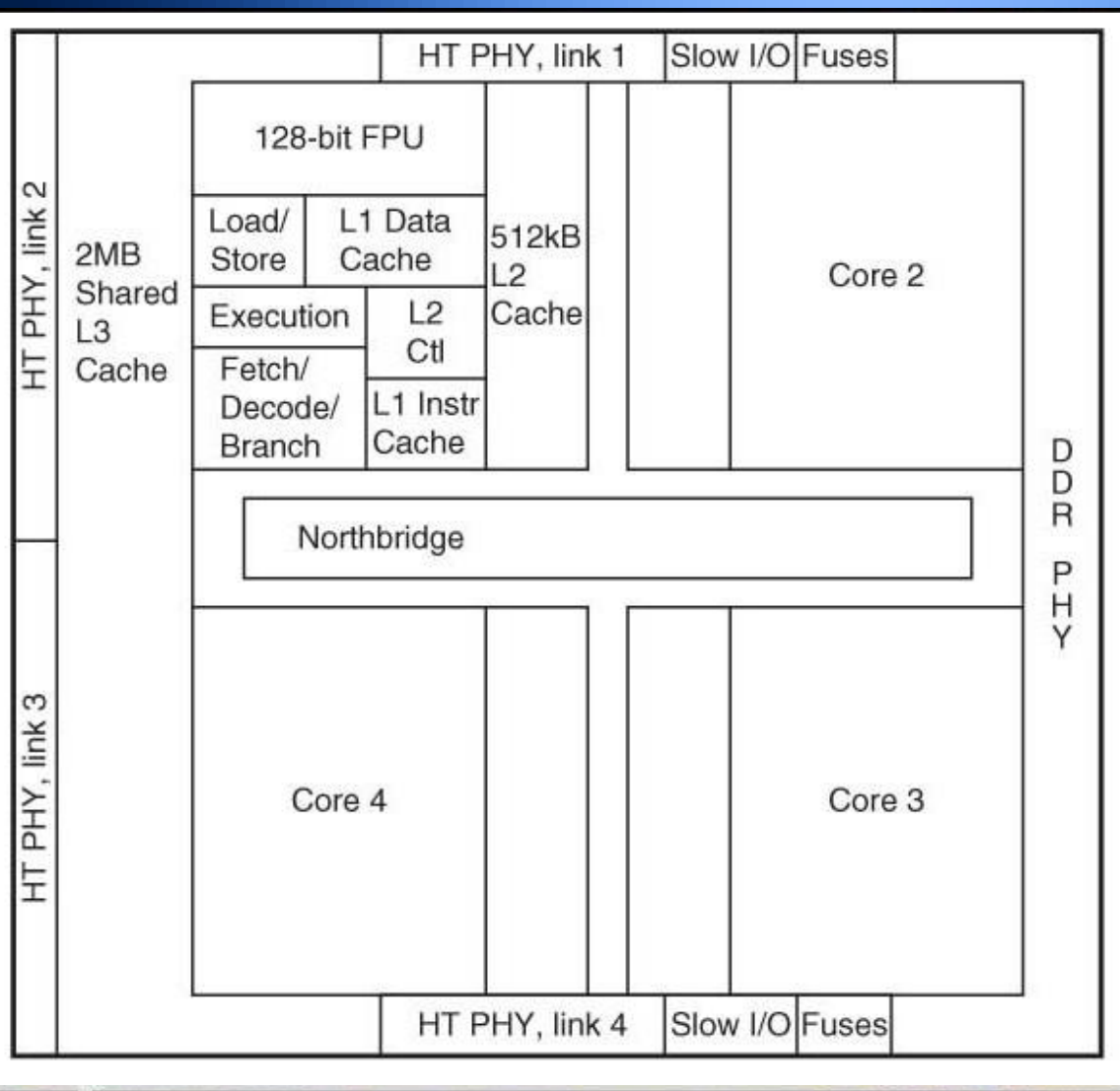
❖ 5 thành phần của hệ thống máy tính:

- ☐ Đường dữ liệu (*eng, datapath*)
- ☐ Khối điều khiển
- ☐ Bộ nhớ
- ☐ Khối vào
- ☐ Khối ra

❖ CPU =
Đường dữ liệu
+ khối điều khiển

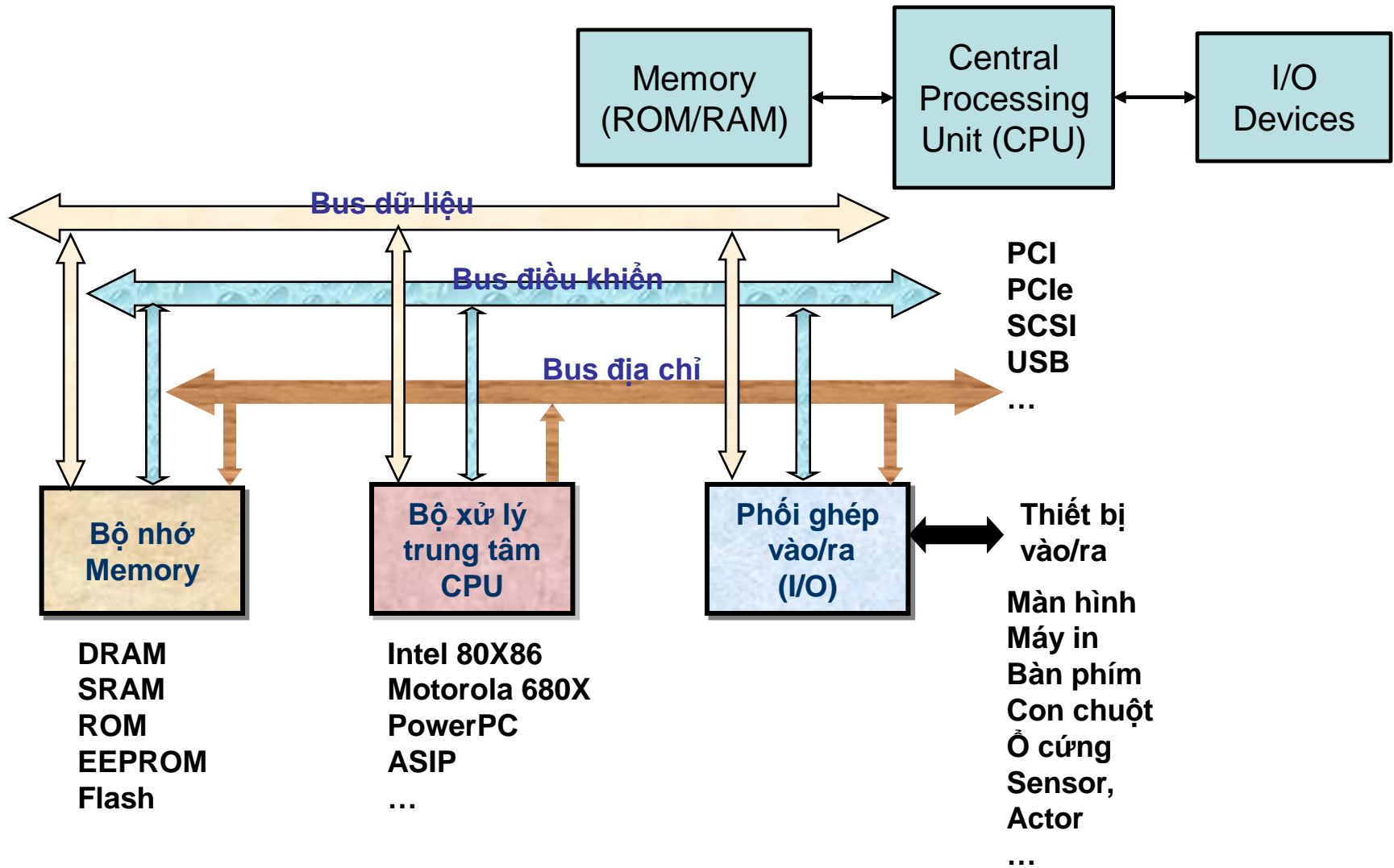


Bộ xử lý đa nhân AMD's Barcelona

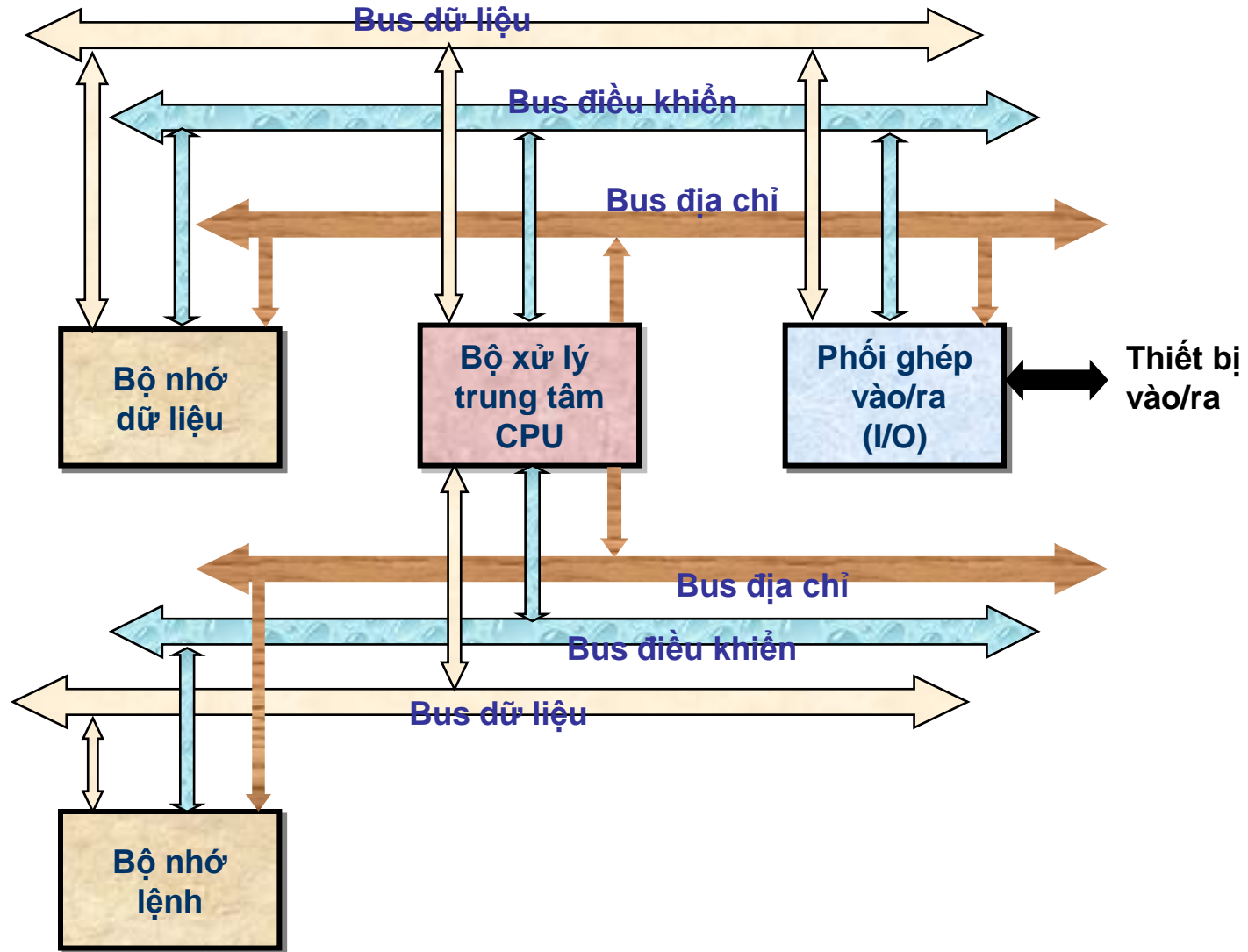


- ❑ 4 nhân, ngoài thứ tự
- ❑ Đồng hồ: 1,9 GHz
- ❑ Công nghệ 65nm
- ❑ 3 mức bộ đệm (L1, L2, L3)
- ❑ Tích hợp bộ điều khiển cầu bắc

Kiến trúc vonNeumann



Kiến trúc Havard



Kiến trúc tập lệnh (ISA)

- ❖ Kiến trúc tập lệnh (*eng, Instruction Set Architecture - ISA*), hay kiến trúc: là giao diện trừu tượng giữa phần cứng và các phần mềm ở lớp thấp nhất, bao gồm tất cả các thông tin cần thiết để viết chương trình ở ngôn ngữ máy như: các chỉ thị máy, thanh ghi, bản đồ bộ nhớ, phương pháp vào ra, ...
- ❖ Giao diện nhị phân ứng dụng (*eng, Application Binary Interface – ABI*) bao gồm các chỉ thị máy cơ bản người dùng có thể sử dụng và các hàm hệ thống cấp thấp do hệ điều hành cung cấp.

Kết luận:

- ❑ Máy tính gồm các lớp phân cấp theo mức độ trừu tượng.
- ❑ Kiến trúc tập lệnh là một lớp then chốt trong hệ thống máy tính.
- ❑ Các triển khai phần cứng khác nhau tuân theo cùng chuẩn về kiến trúc tập lệnh có thể thực hiện cùng một phần mềm giống nhau.

Cấu trúc nội dung môn học

- ❖ Chương 2. Giao diện giữa phần mềm và phần cứng
 - ☐ Kiến trúc tập lệnh
 - ☐ Biểu diễn dữ liệu
 - ☐ Tổ chức và truy cập bộ nhớ
 - ☐ Lệnh vào ra
- ❖ Chương 3. Cấu trúc bộ xử lý
 - ☐ Thiết kế bộ xử lý trung tâm
 - ☐ Kỹ thuật đường ống
- ❖ Chương 4. Bộ nhớ
 - ☐ Phân cấp và thiết kế bộ nhớ
- ❖ Chương 5. Vào ra
 - ☐ Thiết bị, cơ chế vào ra
 - ☐ Cấu trúc bus

Đánh giá và so sánh các máy tính

❖ Quyết định mua máy tính

- ☐ Trong số các máy tính, máy nào có
 - hiệu năng tốt nhất?
 - giá thành rẻ nhất?
 - tỉ lệ giá thành/hiệu năng tốt nhất?

❖ Lựa chọn thiết kế máy tính

- ☐ Trong các lựa chọn thiết kế, thiết kế nào
 - cho cải tiến tốt nhất về hiệu năng?
 - giá thành thấp nhất?
 - tỉ lệ giá thành/hiệu năng tốt nhất?

❖ Yêu cầu:

- ☐ Căn cứ để so sánh
- ☐ Thông số đánh giá

❖ Mục tiêu: nắm rõ sự

- ☐ ảnh hưởng của các nhân tố trong kiến trúc máy tính tới hiệu năng toàn hệ thống;
- ☐ vai trò quan trọng tương đối và giá thành của các nhân tố đó,

Hiệu năng (*eng. Performance*)

- ❖ Thời gian đáp ứng (thời gian thực thi) – là khoảng thời gian giữa thời điểm bắt đầu thực hiện và thời điểm hoàn thành một nhiệm vụ
 - ❑ Quan trọng đối với 1 người sử dụng yêu cầu hệ thống thực hiện 1 nhiệm vụ
- ❖ Thông lượng (dải thông) – là tổng số nhiệm vụ có thể được hoàn thành trong 1 khoảng thời gian
 - ❑ Quan trọng đối với người điều hành trung tâm dữ liệu
- ❖ Cần các hệ đo lường khác nhau cho hiệu năng của máy tính cũng như cần 1 tập hợp các ứng dụng khác nhau để kiểm chuẩn các máy tính nhúng, máy tính để bàn (thường chú trọng đến thời gian đáp ứng) và các máy chủ (thường chú trọng đến thông lượng)

Ví dụ 1.1 – Cải tiến hiệu năng

- ❖ Ảnh hưởng của bộ xử lý lên thời gian đáp ứng và thông lượng
- ❖ Nếu ta thay đổi cấu trúc máy tính như sau thì thời gian đáp ứng và thông lượng của máy tính thay đổi thế nào?
 - ☐ Thay thế bộ xử lý bằng bộ xử lý nhanh hơn
 - ☐ Bổ xung 1 bộ xử lý để thực hiện các nhiệm vụ tách biệt (như trong hệ thống tìm kiếm WWW)

Định nghĩa hiệu năng (tốc độ)

- ❖ Hiệu năng (tốc độ) của máy tính X:

$$\text{Performance}_x = \frac{1}{\text{Execution Time}_x}$$

- ❖ Máy tính X nhanh hơn máy tính Y, n lần:

$$\frac{\text{Performance}_x}{\text{Performance}_y} = \frac{\text{ExecutionTime}_y}{\text{ExecutionTime}_x} = n$$

- ❖ Để tối đa hóa hiệu năng, cần tối thiểu hóa thời gian thực hiện
- ❖ Giảm thời gian đáp ứng thường sẽ tăng thông lượng

Ví dụ 1.2 – So sánh hiệu năng

- ❖ Nếu máy tính A thực hiện 1 chương trình mất 10s và máy tính B chạy cũng chương trình đó mất 15s, máy tính A nhanh hơn máy tính B bao nhiêu lần?

Đo hiệu năng – Đo thời gian thực hiện

Có 3 loại thời gian dùng để tính hiệu năng

- ❖ Thời gian đáp ứng (thời gian đồng hồ, thời gian đã trôi qua):
 - ☐ Tổng thời gian hoàn thành 1 nhiệm vụ
 - ☐ Bao gồm: thời gian truy cập đĩa, bộ nhớ, thời gian vào ra, thời gian cho hệ điều hành
- ❖ Thời gian bộ xử lý (CPU time)
 - ☐ Thời gian CPU người dùng
 - ☐ Thời gian CPU hệ thống

Các yếu tố tính hiệu năng

❖ Thời gian CPU – thời gian bộ xử lý dùng để thực hiện 1 nhiệm vụ

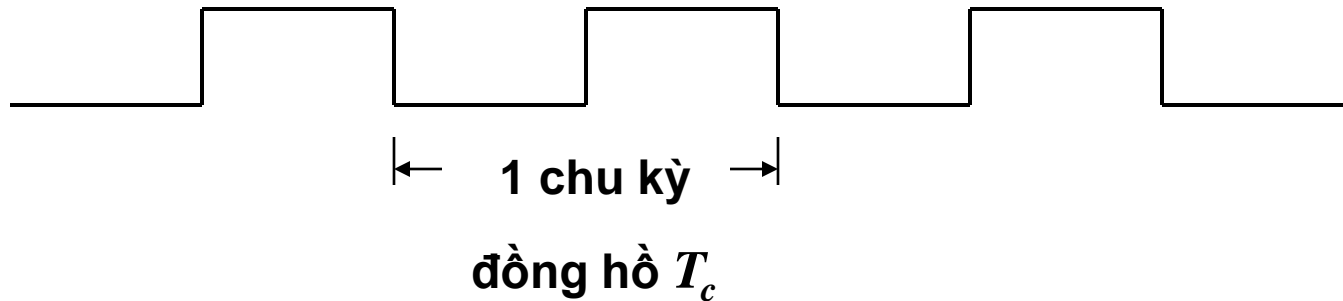
- ❑ Không bao gồm thời gian chờ vào/ra hay thời gian thực hiện các chương trình khác
- ❑ Thời gian CPU cho 1 chương trình, T_{cpu} được tính từ số chu kỳ đồng hồ CPU thực hiện chương trình P và thời gian 1 chu kỳ đồng hồ:

$$T_{cpu} = C \times T_c \text{ or } T_{cpu} = C / f_c$$

❖ Hiệu năng có thể cải thiện bằng cách giảm số chu kỳ 1 xung đồng hồ hoặc giảm số chu kỳ cần thiết để thực hiện chương trình

Xung nhịp đồng hồ

❖ CPU hoạt động đồng bộ theo đồng hồ



10 nsec clock cycle \Rightarrow 100 MHz clock rate

5 nsec clock cycle \Rightarrow 200 MHz clock rate

2 nsec clock cycle \Rightarrow 500 MHz clock rate

1 nsec (10^{-9}) clock cycle \Rightarrow 1 GHz (10^9) clock rate

500 psec clock cycle \Rightarrow 2 GHz clock rate

Ví dụ 1.3 – Cải thiện hiệu năng

❖ Máy tính A với xung đồng hồ 2GHz thực hiện 1 chương trình hết 10 giây. Để thực hiện chương trình đó trong 6 giây bằng máy tính B, ta cần tăng tốc độ xung đồng hồ của máy B. Tuy nhiên, tăng tốc độ xung đồng hồ cũng làm tăng số chu kỳ cần thiết lên 1,2 lần. Xác định tốc độ xung đồng hồ máy tính B.

Số xung đồng hồ

- ❖ Số xung đồng hồ thực hiện 1 chương trình:

$$C = I \times CPI$$

- ❖ Trong đó:

- ☐ I là số chỉ thị máy cần thực hiện trong chương trình
- ☐ CPI (eng. *Clock cycles per Instruction*) là số xung đồng hồ trung bình cần để thực thi 1 chỉ thị máy,

- ❖ CPI có thể dùng để so sánh các máy tính khác nhau cùng triển khai 1 kiến trúc tập lệnh.
- ❖ Ví dụ: có 3 loại lệnh A, B, C khác nhau trong 1 kiến trúc tập lệnh. Mỗi lệnh trong từng loại có CPI tương ứng:

	CPI for this instruction class		
	A	B	C
CPI	1	2	3

Ví dụ 1.4 – So sánh dựa trên CPI

- ❖ Máy tính A và B cùng triển khai 1 kiến trúc tập lệnh. Máy A có chu kỳ đồng hồ là 250ps, và CPI hiệu dụng cho 1 chương trình P là 2,0. Máy B có chu kỳ đồng hồ là 500ps, và CPI hiệu dụng cho cùng 1 chương trình P là 1,2. Máy tính nào nhanh hơn và nhanh hơn bao nhiêu?

CPI hiệu dụng (trung bình)

- ❖ CPI hiệu dụng được tính bằng cách xét tất cả các lớp chỉ thị có trong chương trình và lấy trung bình với trọng số là tỉ lệ xuất hiện của lớp chỉ thị trong chương trình

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

- ❖ Trong đó:

- ☐ IC_i là tỉ lệ (%) số chỉ thị thuộc lớp i được thực thi
- ☐ CPI_i là số chu kỳ (trung bình) cần để thực hiện 1 chỉ thị thuộc thuộc lớp i
- ☐ N là số lớp chỉ thị

- ❖ CPI hiệu dụng phụ thuộc vào tỉ lệ chỉ thị trong một chương trình (tần suất động của các chỉ thị trong 1 hoặc nhiều chương trình)

Công thức hiệu năng

❖ Công thức hiệu năng cơ bản:

$$T_{cpu} = I \times CPI \times T_c \qquad T_{cpu} = \frac{I \times CPI}{f_c}$$

❖ Công thức trên phân tách 3 yếu tố ảnh hưởng đến hiệu năng máy tính

❖ Cho phép đo CPU được đo bằng cách chạy chương trình:

- ☐ Tốc độ đồng hồ được cho trước
- ☐ Số chỉ thị I được đo bằng cách dùng công cụ profilers/mô phỏng sự thực hiện chương trình mà không cần triển khai phần cứng
- ☐ CPI phụ thuộc vào loại chỉ thị, triển khai phần cứng chi tiết

❖ Cho phép so sánh 2 triển khai phần cứng hoặc đánh giá lựa chọn giữa 2 thiết kế

Các yếu tố quyết định hiệu năng CPU

$$T_{cpu} = I \times CPI \times T_c$$

	I	CPI	T_c
Algorithm			
Programming language			
Compiler			
ISA			
Core organization			
Technology			

Ví dụ 1.5 – So sánh đoạn mã chương trình

- ❖ Người thiết kế một máy tính triển khai kiến trúc tập lệnh gồm 3 loại chỉ thị A, B, C được CPI như sau:

	A	B	C
CPI	1	2	3

- ❖ Với 1 câu lệnh ở ngôn ngữ bậc cao, người viết trình biên dịch có thể lựa chọn 2 đoạn chỉ thị máy gồm có tần suất các loại chỉ thị như sau:

Đoạn mã	A	B	C
1	2	1	2
2	4	1	1

- ❖ Đoạn mã nào gồm nhiều chỉ thị hơn? Đoạn mã nào nhanh hơn? Tính CPI của từng đoạn mã.

Ví dụ 1.6 – Cải tiến hiệu năng

- ❖ Cho một máy tính thực hiện 1 chương trình gồm 4 loại chỉ thị máy có các thông số về tần suất và CPI như sau:

Op	Tần suất (IC_i)	CPI_i	$IC_i \times CPI_i$
ALU	50%	1	
Load	20%	5	
Store	10%	3	
Branch	20%	2	
			$\Sigma =$

- ❖ Nếu ta có bộ đệm dữ liệu làm giảm thời gian nạp (Load) xuống 2 chu kỳ, máy tính sẽ nhanh lên bao nhiêu lần?
- ❖ Nếu ta có khối dự báo rẽ nhánh cho phép tiết kiệm 1 chu kỳ khi rẽ nhánh, hiệu năng sẽ thế nào?
- ❖ Nếu ta có 2 khối ALU thực hiện 2 chỉ thị ALU đồng thời?

Chương trình đo kiểm chuẩn

- ❖ Benchmarks – là tập hợp các chương trình tạo nên 1 “workload” được chọn để đo hiệu năng
- ❖ SPEC (System Performance Evaluation Cooperative) tạo ra 1 tập các benchmark chuẩn bắt đầu từ SPEC89. Chuẩn mới nhất là SPEC CPU2006 gồm 12 chương trình số nguyên (CINT2006) và 17 chương trình số thực (CFP2006)

www.spec.org

- ❖ Ngoài ra, còn có các tập các workload để đo kiểm năng lượng (SPECpower_ssj2008), mail (SPECmail2008), hay đa phương tiện (mediabench), ...

Các chương trình đo kiểm

Integer benchmarks		FP benchmarks	
gzip	compression	wupwise	Quantum chromodynamics
vpr	FPGA place & route	swim	Shallow water model
gcc	GNU C compiler	mgrid	Multigrid solver in 3D fields
mcf	Combinatorial optimization	applu	Parabolic/elliptic pde
crafty	Chess program	mesa	3D graphics library
parser	Word processing program	galgel	Computational fluid dynamics
eon	Computer visualization	art	Image recognition (NN)
perlbmk	perl application	equake	Seismic wave propagation simulation
gap	Group theory interpreter	facerec	Facial image recognition
vortex	Object oriented database	ammp	Computational chemistry
bzip2	compression	lucas	Primality testing
twolf	Circuit place & route	fma3d	Crash simulation fem
		sixtrack	Nuclear physics accel
		apsi	Pollutant distribution

SPEC CINT2006 on Barcelona ($T_c = 0,4 \times 10^9$)

Name	1×10^9	CPI	ExTime	RefTime	SPEC ratio
perl	2 118	0,75	637	9 770	15,3
bzip2	2 389	0,85	817	9 650	11,8
gcc	1 050	1,72	724	8 050	11,1
mcf	336	10,00	1 345	9 120	6,8
go	1 658	1,09	721	10 490	14,6
hmmer	2 783	0,80	890	9 330	10,5
sjeng	2 176	0,96	837	12 100	14,5
libquantum	1 623	1,61	1 047	20 720	19,8
h264avc	3 102	0,80	993	22 130	22,3
omnetpp	587	2,94	690	6 250	9,1
astar	1 082	1,79	773	7 020	9,1
xalancbm	1 058	2,70	1 143	6 900	6,0
Geometric Mean					11,7

So sánh và tổng kết hiệu năng

❖ Tổng kết hiệu năng cho 1 tập tiêu chuẩn thành một số duy nhất:

- ❑ Thời gian thực hiện được chuẩn hóa thành SPEC Ratio (tỉ số lớn có nghĩa là nhanh hơn)
- ❑ Lấy trung bình nhân của các SPEC Ratio:

$$GM = \sqrt[n]{\prod_{i=1}^n \text{SPEC ratio}_i}$$

Các thông số hiệu năng khác

❖ Năng lượng tiêu thụ là một thông số quan trọng, đặc biệt với thị trường hệ nhúng (thời lượng pin là quan trọng với hệ thống)

Target Load %	Performance (ssj_ops)	Average Power (Watts)
100 %	231,86 7	295
90%	211 ,282	286
80%	185,80 3	275
70%	163,42 7	265
60%	140,16 0	256
50%	118,32 4	246
40%	92,03 5	233
30%	70,50 0	222
20%	47,12 6	206
10%	23,06 6	180
0%	0	141
Overall Sum	1,283, 590	2,605
$\Sigma \text{ssj_ops} / \Sigma \text{power} =$		493

Luật Amdahl

- ❖ Khi thực hiện cải tiến một đặc điểm của hệ thống, tác dụng của việc cải tiến bị giới hạn bởi đặc điểm được cải tiến

$$T_{cpu} \text{ sau khi cải tiến} = \frac{T_{cpu} \text{ ảnh hưởng bởi sự cải tiến}}{Tỉ lệ cải tiến} + T_{cpu} \text{ không ảnh hưởng}$$

- ❖ Luật cơ bản để tính toán định lượng sự cải tiến
- ❖ Khi cải tiến, cần chú trọng đến các trường hợp thông dụng
- ❖ Đặt ra giới hạn số lượng bộ xử lý hoạt động song song

Ví dụ 1.7 - Luật Amdahl

- ❖ 1 chương trình thực hiện bởi 1 máy tính trong 100 giây. Trong đó, 80 giây được dùng để thực hiện phép nhân. Vậy cần phải tăng tốc độ phép nhân lên mấy lần để có thể tăng tốc độ thực hiện chương trình lên 5 lần?

$$T_{cpu} \text{ sau khi cải tiến} = \frac{T_{cpu} \text{ ảnh hưởng bởi sự cải tiến}}{\text{Tỉ lệ cải tiến}} + T_{cpu} \text{ không ảnh hưởng}$$

- ❖ Không thể cải tiến để tăng tốc được 5 lần

Kết luận chương

- ❖ Hệ thống máy tính được xây dựng **từ phân cấp các lớp trừu tượng**. Các chi tiết triển khai lớp dưới bị che khuất khỏi lớp trên.
- ❖ **Kiến trúc tập lệnh** – lớp giao tiếp giữa phần cứng và phần mềm mức thấp – là lớp trừu tượng quan trọng trong hệ thống máy tính.
- ❖ Phần cứng máy tính gồm **5 thành phần: đường dữ liệu, khối điều khiển, bộ nhớ, khối vào, và khối ra**. 5 thành phần đó kết nối với nhau bằng hệ thống bus theo mô hình **vonNeumann** hoặc mô hình **Havard**.
- ❖ Phương pháp đánh giá hiệu năng một hệ thống máy tính là dùng **thời gian thực hiện 1 chương trình**. Thời gian thực hiện chương trình được tính bằng công thức:

$$T_{cpu} = I \times CPI \times T_c$$