



**TÍNH TOÁN SONG SONG**

# GIÁO TRÌNH

**CÔNG BỐ CÔNG TRÌNH:**

hập, Giáo trình kỹ  
trình, Tập 1, NXB  
tự nhiên và Công  
ng 10-2008.

áp, An application of  
ontrol problem for  
n of investments,  
các bài báo Khoa  
Bách Khoa Hà Nội,  
1906

áp, Phân mảnh trên  
của các thuộc tính  
quan hệ, Tập chỉ  
điều khiển tối ưu,  
1 (2007) 86 - 98.

áp, Mô hình kinh tế vĩ  
n bộ kỹ thuật công  
án để điều khiển tối  
chỉ Tin học và Điều  
ưu, Tập 1 số 3  
-26.

áp, Thi trắc nghiệm  
của hệ chuyên gia,  
l thảo Quốc gia "Một  
chọn lọc của CNTT  
thông", Huế, 12-  
., NXB Khoa học và  
t Hà Nội, 422-430.

áp, Loại bỏ các mẫu  
ẩn thừa trong cơ sở  
an hệ, Kỹ yếu hội  
c gia, Đại Lải, 14-  
., NXB Khoa học Tự  
Công nghệ, Hà Nội  
-227

hội nghị, hội thảo

**NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG**

Trụ sở chính: 18 Nguyễn Du, Hà Nội  
ĐT Biên tập: 04.35772143, 04.35772145      ĐT Phát hành: 04.35772138  
E-mail: nxb.tttt@mic.gov.vn      Fax: 04.35772037  
Website: www.nxbthongtintruyenthong.vn

**Chi nhánh TP. Hồ Chí Minh:** 8A đường D2, phường 25, quận Bình Thạnh, TP. Hồ Chí Minh  
Điện thoại: 08.35127750, 08.35127751      Fax: 08.35127751  
E-mail: cnsq.nxbtttt@mic.gov.vn

**Chi nhánh TP. Đà Nẵng:** 42 Trần Quốc Toản, quận Hải Châu, TP. Đà Nẵng  
Điện thoại: 0511.3897467      Fax: 0511.3843359  
E-mail: cndn.nxbtttt@mic.gov.vn

**MỜI CÁC BẠN TÌM ĐỌC**

1. GIÁO TRÌNH MẠNG MÁY TÍNH
2. GIÁO TRÌNH KỸ THUẬT VI XỬ LÝ (02 TẬP)
3. GIÁO TRÌNH LÝ THUYẾT THÔNG TIN
4. GIÁO TRÌNH KỸ THUẬT LẬP TRÌNH
5. GIÁO TRÌNH CƠ SỞ DỮ LIỆU: LÝ THUYẾT VÀ THỰC HÀNH

**SÁCH CỦA NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG CÓ BÁN TẠI:**

1. **Nhà sách Tiến phong**  
175 Nguyễn Thái Học, Hà Nội
2. **Nhà sách Nguyễn Văn Cừ**  
36 Xuân Thủy, Cầu Giấy, Hà Nội
3. **Nhà sách Minh Châu**  
Số 10 và 14/40 Tạ Quang Bửu, Hai Bà Trưng, Hà Nội
4. **Nhà sách PTIT**  
Km 10, Nguyễn Trãi, Hà Nội
5. **Nhà sách Bách Khoa**  
Số 1 đường Giải Phóng, Hà Nội  
86 - 107 Tô Hiến Thành, quận 10, TP. HCM
6. **Nhà sách Thăng Long**  
2 Bis Nguyễn Thị Minh Khai, quận 1, TP. HCM
7. **Nhà sách Minh Khai**  
249 Nguyễn Thị Minh Khai, quận 1, TP. HCM

Giá: ....000đ


TS. LÊ HUY THẬP

**CƠ SỞ  
LÝ THUYẾT  
SONG SONG**

**NHÀ XUẤT BẢN  
THÔNG TIN VÀ TRUYỀN THÔNG**

**NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG**

**TS. LÊ HUY THẬP**



**NGVC. TS. LÊ HUY THẬP**  
Sinh năm: 1949  
Quê quán: Hương Khê - Hà Tĩnh

Tốt nghiệp Đại học chuyên  
toán tại Khoa toán lý Đại học  
Bách khoa Hà Nội năm 1973, bác  
vệ luận văn Tiến sĩ toán tại Khoa  
toán, Trường đại học Tổng hợp  
Komenský, Tiệp Khắc 1989.  
Trưởng phòng nghiên cứu và ủy  
viên Hội đồng khoa học Viện  
CNTT, kiêm các chức như Chủ  
tịch công đoàn viện, Bí thư chi bộ  
nhiều khóa.....

Đã tham gia giảng dạy  
Toán, CNTT từ những năm 1974  
cho đến nay tại các trường: Học  
viện Công nghệ BCVT, Đại học  
Bách Khoa Hà Nội, Đồng Đố  
Phương Đông, Sư phạm Hà Nội,  
Genetic-Singapore, Đại học Thái  
Nguyên (giảng dạy cao học),  
Viện Cơ học Nhà nước (giảng dạy  
cao học).

Chủ nhiệm nhiều đề tài  
Khoa học,...Đã hướng dẫn tốt  
nghiệp hàng trăm sinh viên, hàng  
chục thạc sĩ và tiến sĩ.

# 1. 1. Tổng quan

Một công việc, một vấn đề hay một bài toán nào đó,... (như xây dựng nhà ở, cầu cống, giải một bài toán,... gọi chung là công việc) nếu được phân chia thành các công việc nhỏ hơn, các công việc nhỏ này nếu có thể thực hiện đồng thời, khi tất cả các công việc con được thực hiện xong, kết quả của chúng sẽ được kết nối lại để có kết quả chung cho cả công việc thì ta nói công việc đó được thực hiện song song..

Chẳng hạn việc đào móng công trình và chuẩn bị sắt làm móng có thể làm đồng thời. Sau khi hai cả hai phần việc này xong, chúng ta có thể đổ móng cho công trình. Hoặc khi cộng hai véc tơ  $n$  phần tử, chúng ta có thể phân nhóm các phần tử và chia cho các bộ xử lý cùng thực hiện phép cộng,...

Tính toán song song là sự phát triển của tính toán tuần tự, tính toán song song rất phức tạp, các sự kiện liên quan với nhau xảy ra tại cùng thời điểm, cùng trong phạm vi một chuỗi hành động. Chẳng hạn:

- Hệ hô hấp và hệ tuần hoàn của các loài động vật
- Việc hình thành dãy Ngân Hà
- Vận động của hành tinh
- Đại dương và thời tiết
- Trôi dạt, tích tụ và kiến tạo địa tầng.
- Giao thông giờ cao điểm
- Tuyến đường Ô tô trong thành phố
- Xây dựng các tuyến giao thông trong thành phố

# 1.1.1. Tính toán tuần tự, chương trình tuần tự và song song

## ***Tính toán tuần tự***

Một công việc được chia thành một dãy rời rạc các chỉ thị mà máy tính có thể hiểu để thực hiện.

Máy tính thực hiện các chỉ thị theo thứ tự mà chúng xuất hiện.

Chỉ có duy nhất một chỉ thị được thực hiện tại một thời điểm.

## ***Chương trình tuần tự***

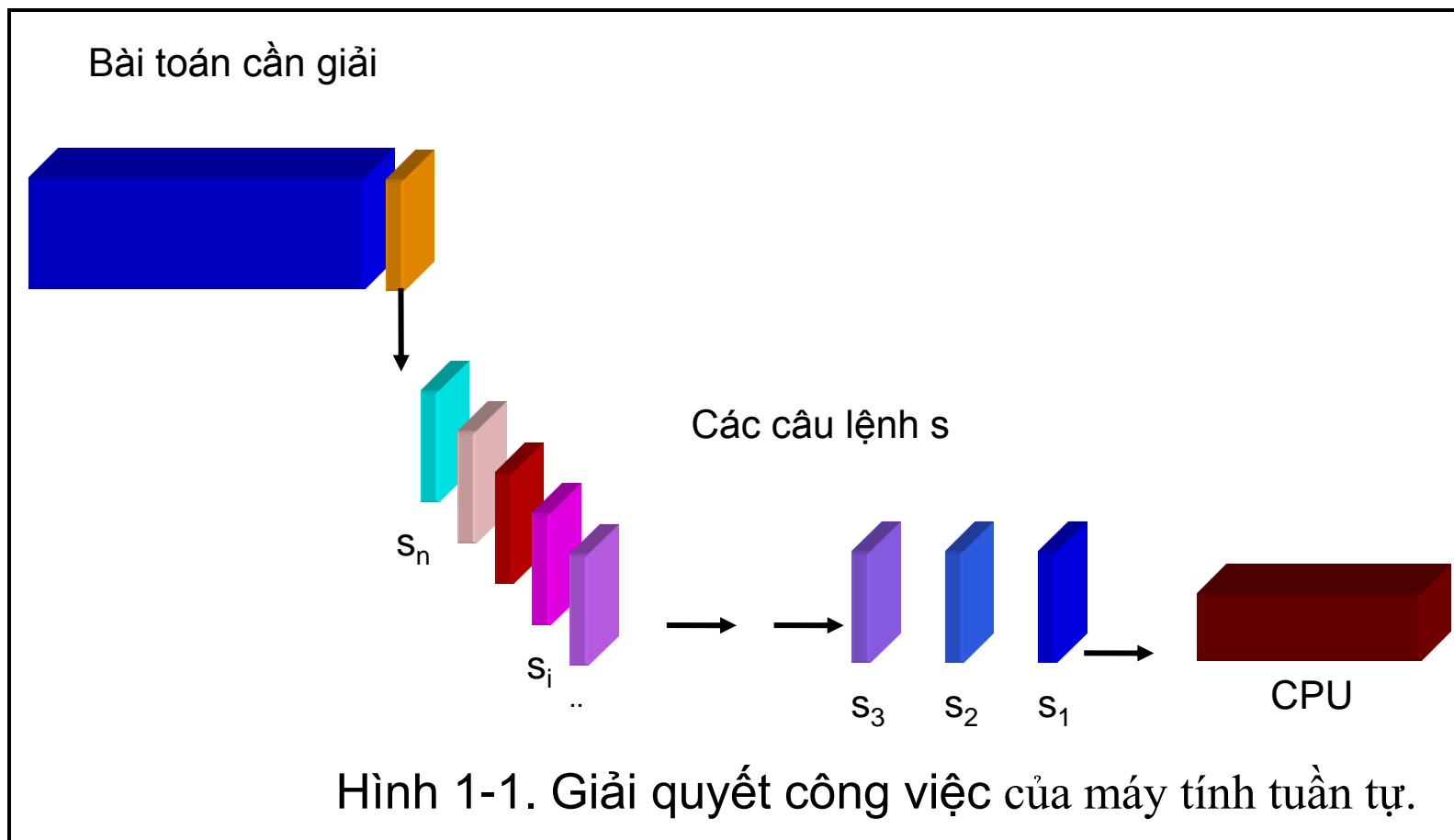
Chương trình được viết trên một ngôn ngữ nào đó để thể hiện việc tính toán tuần tự được gọi là chương trình tuần tự.

# Máy tính tuần tự

---

Loại máy tính thực hiện các lệnh một cách tuần tự và tại mỗi thời điểm chỉ một chỉ thị duy nhất được thực hiện – máy tính chỉ có một CPU. Vì chỉ có một CPU nên còn gọi là máy tính đơn nguyên.

# MINH HỌA TÍNH TOÁN TUẦN TỰ





## Ví dụ 1-1

---

Chương trình HELLO.CPP (trong C++) đưa dòng “Hello World !” ra màn hình là chương trình tuần tự sau đây:

t1 : *#include <iostream. h>*

t2 : *int main ( )*

t3: {

t4 :     *cout << "Hello World ! \n " ,*

t5 :     *return 0 ;*

t6: }

Kết quả thực hiện chương trình cho:

Hello World !

Rõ ràng là các lệnh của chương trình được thực hiện thứ tự từ lệnh t1 đến lệnh t2, ..., cho đến lệnh cuối cùng là lệnh t6.



## Tính toán song song

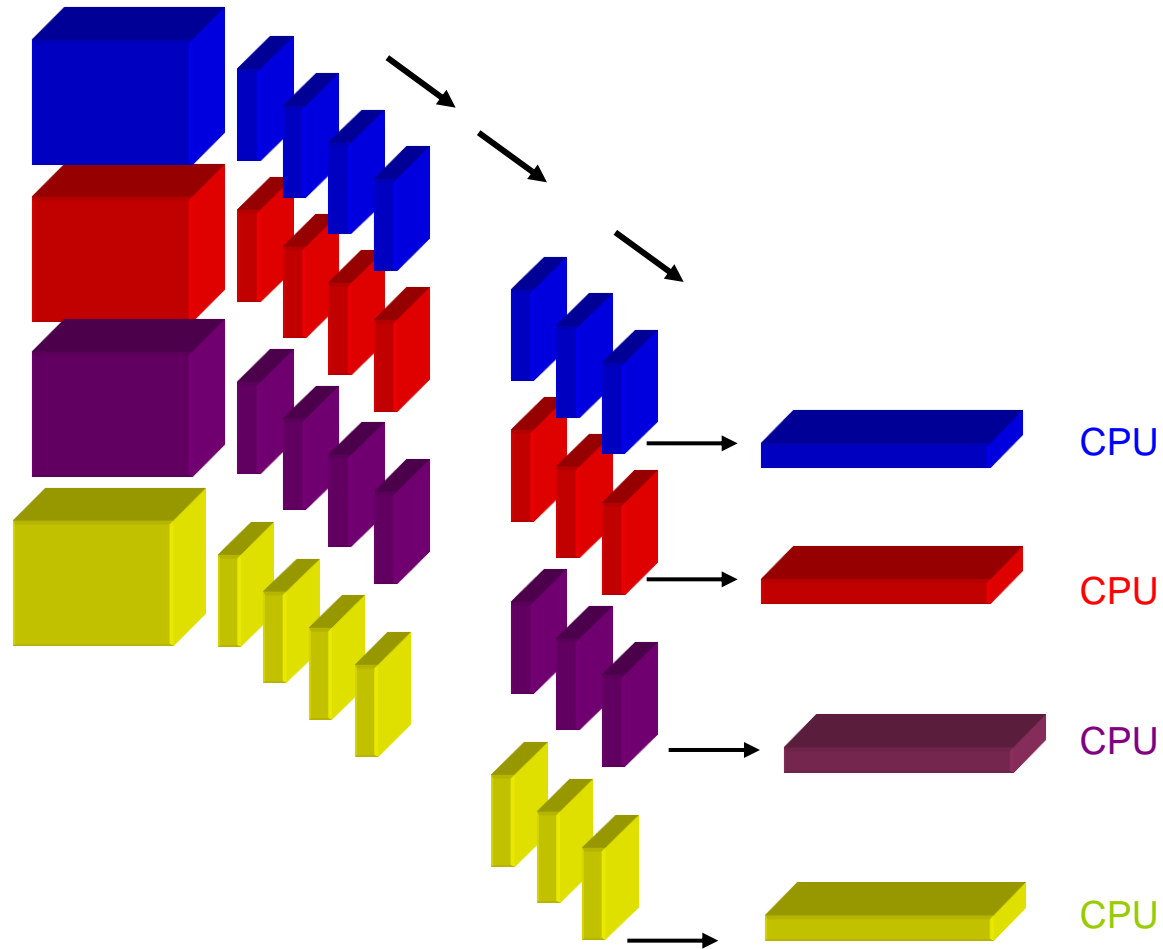
*Tính toán song song là quá trình tính toán gồm nhiều tiến trình đồng thời và cùng tham gia tính toán giải quyết một công việc, và nói chung phải thực hiện trên các hệ thống đa bộ xử lý.*

Công việc cần tính toán được phân hoạch thành các phần tính toán rời rạc, các phần này được tính toán song song với nhau.

Các chỉ thị của từng phần được thực hiện trên các CPU khác nhau.

Quan niệm đơn giản nhất, tính toán song song là tính toán đồng thời các phần của công việc trên nhiều máy tính được kết nối lại hoặc trên một máy tính có nhiều hơn một bộ xử lý.

# MINH HỌA TÍNH TOÁN SONG SONG



Hình 1-2. Cách thực hiện các phần song song trên máy tính song song

# Các tài nguyên cần cho TTSS

## ***Thiết bị cần cho tính toán song song:***

- Hoặc một máy tính nhưng có nhiều bộ xử lý,
- Hoặc một số tùy ý các máy tính được kết nối thành mạng.
- Hoặc tổ hợp cả hai loại trên.

## ***Tính chất công việc có thể tính toán song song được:***

- Có thể chia thành các phần rời rạc, các phần đó được thực hiện đồng thời;
- Có thể thực hiện nhiều câu lệnh tại cùng một thời điểm;
- Hiệu năng - theo nghĩa sử dụng ít thời gian (hay tiền bạc) hơn tính toán tuần tự trên máy đơn nguyên.

# Chương trình song song

---

Chương trình được viết trên một ngôn ngữ lập trình nào đó để thực hiện việc tính toán song song được gọi là chương trình song song.



## Ví dụ 1-2

---

Tính tổng hai véc tơ:  $C = A + B$

Chương trình tuần tự trên máy tính đơn nguyên có thể như sau (giả mã)

```
for( $i = 0 ; i < n ; i++$ )  
{  
     $C[i] = A[i] + B[i];$   
}
```

Máy tính có  $p$  bộ xử lý. Có thể phân hoạch  $N$  phần tử của hai mảng  $A$  và  $B$  thành  $p$  phần, bộ xử lý thứ  $k$  nhận  $n_k$  phần tử.

| Bộ xử lý                  | 1             | 2                 | ... | p                     |
|---------------------------|---------------|-------------------|-----|-----------------------|
| Các chỉ số phần tử mảng A | 0 ... $n_1-1$ | $n_1$ ... $n_2-1$ | ... | $n_{p-1}$ ... $n_p-1$ |
| Các chỉ số phần tử mảng B | 0 ... $n_1-1$ | $n_1$ ... $n_2-1$ | ... | $n_{p-1}$ ... $n_p-1$ |
| Các chỉ số phần tử mảng C | 0 ... $n_1-1$ | $n_1$ ... $n_2-1$ | ... | $n_{p-1}$ ... $n_p-1$ |

```

for (j =  $n_{k-1}$  ; j < ( $n_k - 1$ ) ; j++)
{
    C[j] = A[j] + B[j]
}

```

Trong đó  $k = 1, 2, 3, \dots, p$

p bộ xử lý cùng làm việc đồng thời

# Mục đích của tính toán song song

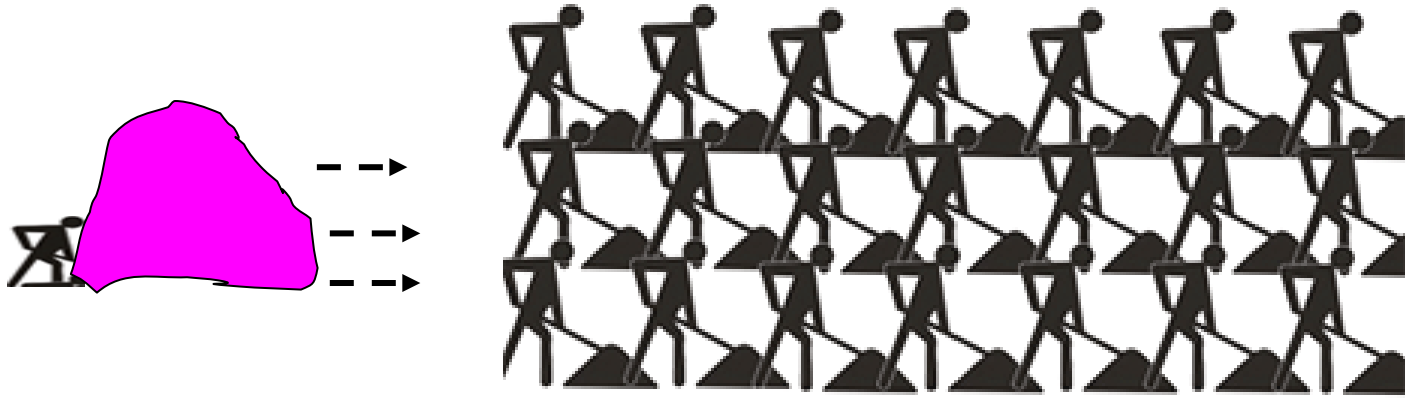
1. Nâng cao hiệu năng tính toán
2. Tính toán các vấn đề khoa học kỹ thuật rất phức tạp trong thế giới thực....

Chẳng hạn:

- Khí quyển, Trái đất, Môi trường
- Vật lý ứng dụng, Hạt nhân, Hạt, Áp suất cao, Sự nóng chảy, Lượng tử ánh sáng
- Sinh học, Công nghệ sinh học, Di truyền học
- Hóa học, Khoa học phân tử
- Địa chất, Địa chấn học
- Công nghệ vũ trụ
- Kỹ thuật điện, Thiết kế mạch, Vi điện tử
- Khoa học máy tính, Toán học
- .....

## 1.1.2. Tại sao sử dụng tính toán song song

Tiết kiệm thời gian và / hoặc tài chính



Khi chi phí nhiều tài nguyên cho một công việc nào đó sẽ rút ngắn được thời gian hoàn thành công việc đó, và tiết kiệm các chi phí tiềm năng.

Phần cứng song song có thể được xây dựng với giá thành thấp và dễ mua.



## **Giải quyết được các vấn đề lớn hơn, phức tạp hơn**

- Nhiều vấn đề quá lớn, quá phức tạp đến nỗi nó không thực tế hoặc không có khả năng giải quyết chúng trên máy tính đơn, đặc biệt là bộ nhớ máy tính lại bị hạn chế. Chẳng hạn các vấn đề:
- Dự báo thời tiết, bão, động đất, sóng thần, mô hình sinh thái, ...
- Máy dò tìm web/CSDL để xử lý hàng triệu triệu giao dịch trên một giây (công cụ tìm kiếm web/CSDL xử lý hàng triệu giao dịch/giây)
- .....

- ***Cung cấp tính đồng thời***

Tài nguyên của một máy tính đơn nguyên chỉ có khả năng thực hiện một lệnh tại một thời điểm. Tài nguyên của máy tính đa bộ xử lý có thể thực hiện nhiều lệnh đồng thời.

- ***Sử dụng tài nguyên phi cục bộ***

Sử dụng tài nguyên tính toán trên một mạng diện rộng, hoặc thậm chí Internet khi tài nguyên tại nơi tính toán tài nguyên đang khan hiếm, không đủ.

# Những hạn chế khi tính toán tuần tự

- Tốc độ truyền - tốc độ của một máy tính tuần tự phụ thuộc trực tiếp vào tốc độ chuyển dữ liệu qua phần cứng. (Giới hạn tốc độ tuyệt đối là tốc độ của ánh sáng (30 cm/nanosecond) và hạn chế sự lan truyền (bằng thông) của dây đồng (9 cm/nanosecond))
- Các hạn chế về kinh tế - làm một bộ xử lý đơn càng nhanh thì càng đắt tiền.
- .....
- Các kiến trúc máy tính hiện tại đang ngày càng dựa vào khả năng song song hóa phần cứng để cải thiện hiệu suất như:
  - Có nhiều đơn vị xử lý
  - Dùng các lệnh đường ống (Pipelined instructions)
  - Đa nhân (Multi-core)

# Ưu - Nhược điểm cơ bản của vấn đề song song

## Ưu điểm:

- Có thể kết thúc công việc sớm hơn,
- Chi phí có thể ít hơn nhiều,
- ...

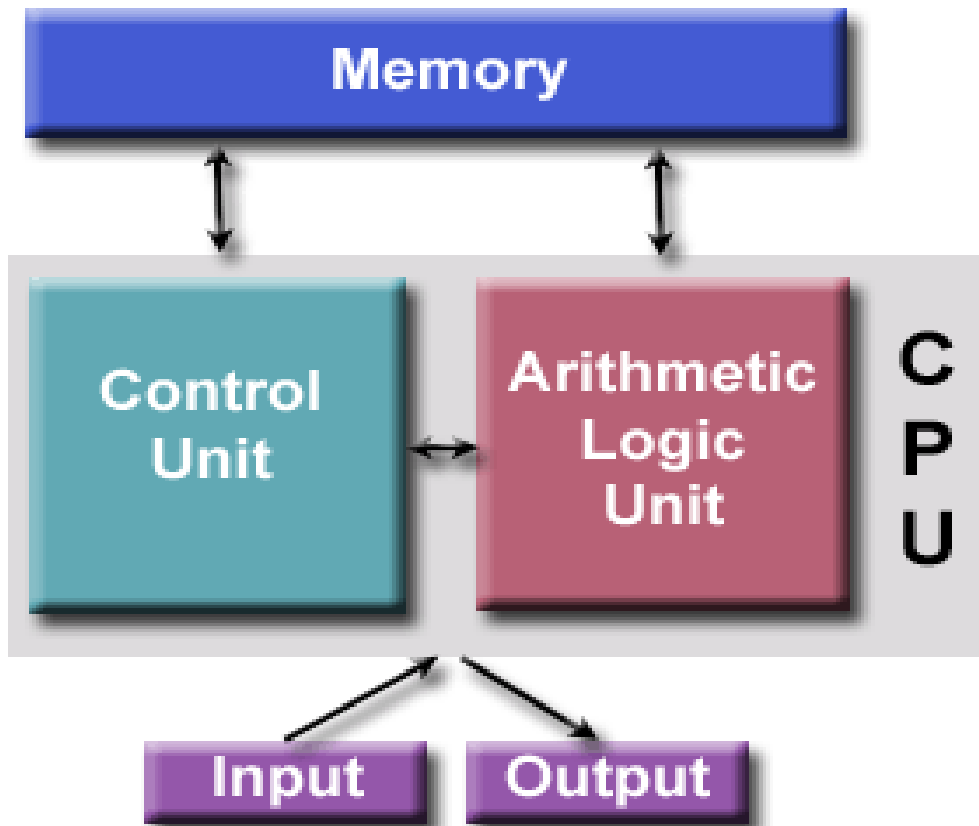
## • Nhược điểm:

- Tăng tính phức tạp cho hệ thống.
- Tăng chi phí cho nhân lực vì phải tăng nhân lực
- Tăng chi phí mua các thiết bị.
- Tăng chi phí lắp đặt hệ thống.
- .....

# 1.3. Kiến trúc máy tính và thuật ngữ

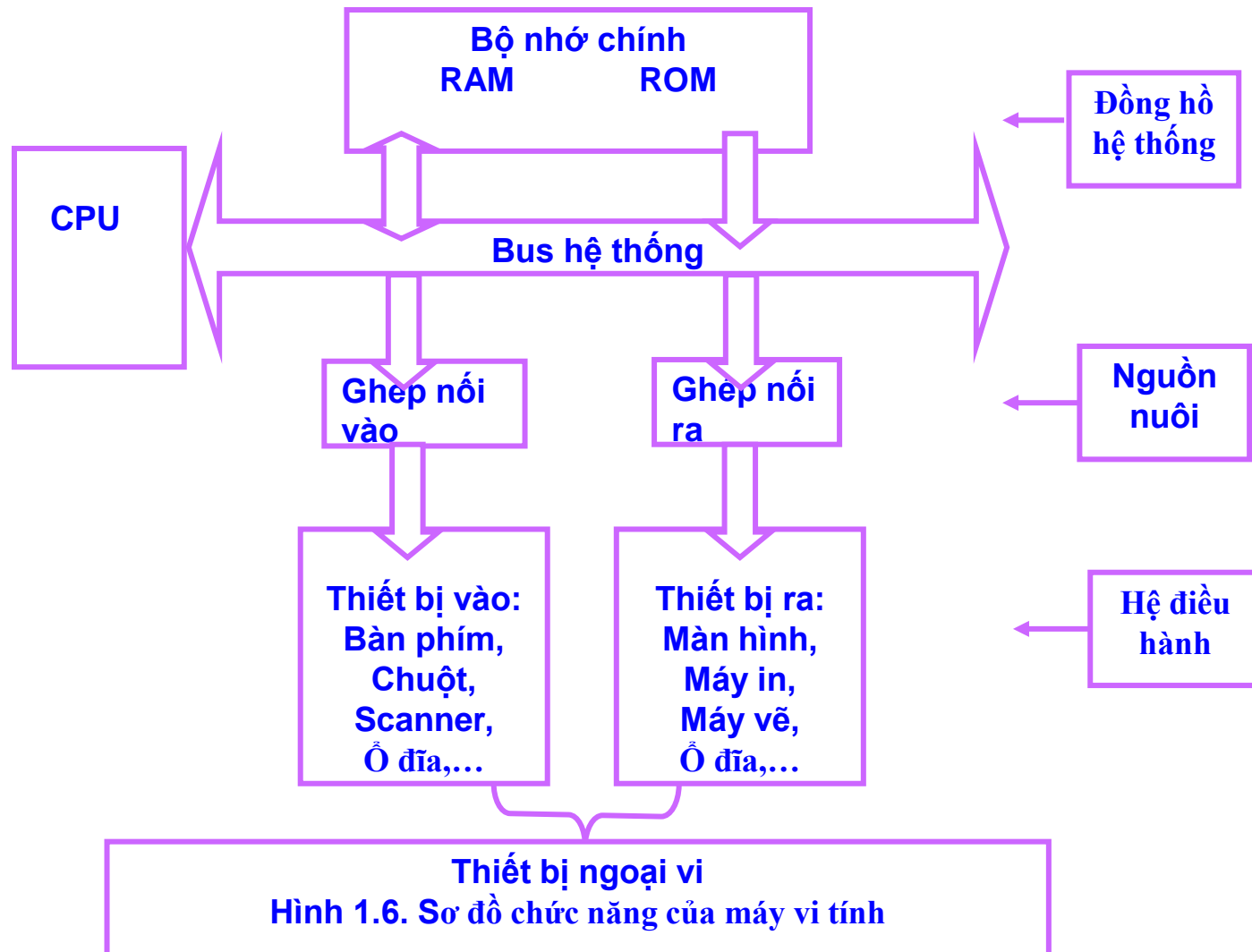
## 1.3.1. Kiến trúc máy tính đơn nguyên-Tuần tự

### Kiến trúc máy tính Von Neumann



1. *Memory* - Bộ nhớ
2. *Control Unit* – Bộ điều khiển
3. *Arithmetic Logic Unit* – Bộ logic và số học
4. *Input/Output* – Các thiết bị vào ra

# Sơ đồ của máy tính đơn nguyên, có thể thể hiện như hình 1.6



## 1.3.2. Máy tính song song và cách phân loại

### ***Máy tính song song***

- Tính toán song song liên quan chặt chẽ đến kiến trúc máy tính, đến phần mềm hệ thống, thuật toán và cả ngôn ngữ lập trình, v,v...
- Máy tính song song bao gồm một tập hợp các bộ xử lý, nói chung là cùng loại (đồng nhất) chúng được kết nối với nhau theo một kiến trúc nào đó cùng với các nghi thức trao đổi dữ liệu và các quy tắc hoạt động giữa các bộ xử lý này.
- Các hệ điều hành hiện nay đều được hỗ trợ đa đa nhiệm nên có thể dùng phương pháp lập trình song song trên đó. Vấn đề là làm thế nào để các bộ xử lý độc lập cùng hoạt động, cùng tham gia giải quyết một vấn đề, tức là các tiến trình được thực hiện trên các bộ xử lý một cách đồng bộ, phải trao đổi với nhau, và phải cùng giải quyết một vấn đề cho trước.

## Phân loại máy tính song song - Kiểu phân loại kinh điển của Flynn

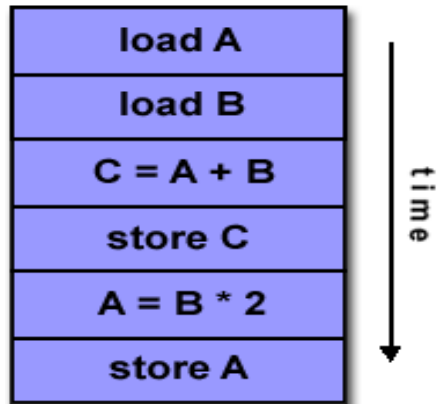
- Phân loại kiến trúc máy tính song song (hay đa bộ xử lý) của Flynn là dựa vào cách thức mà máy đó tải các dòng chỉ thị và các luồng dữ liệu. Mỗi một trong các cách đó là chỉ ra trạng thái đơn hay đa chỉ thị hoặc /và dữ liệu .
- Bốn cách phân loại theo Flynn được trình bày tóm tắt trên hình 1.7:



|  |  |
|--|--|
| <b>S I S D</b><br>Single Instruction,<br>Single Data   | <b>S I M D</b><br>Single Instruction,<br>Multiple Data   |
| <b>M I S D</b><br>Multiple Instruction,<br>Single Data | <b>M I M D</b><br>Multiple Instruction,<br>Multiple Data |

Hình 1.7. Bốn loại máy tính song song theo Flynn

# 1. Cách phân loại thứ nhất SISD (Single Instruction, Single Data). Dùng cho máy tính đơn nguyên tuần tự



Hình 1.8. Đơn chỉ thị, đơn luồng dữ liệu

- **Đơn chỉ thị:** mỗi thời điểm chỉ có một chỉ thị đang được xử lý bởi CPU trong một chu kỳ đồng hồ nào đó.

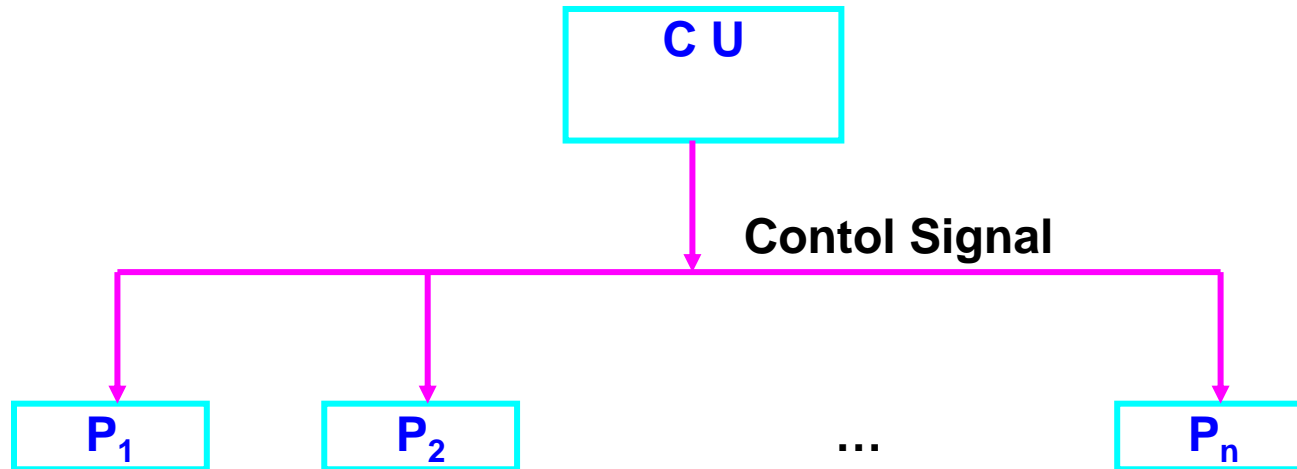
Chẳng hạn, chu kỳ đồng hồ thứ nhất *load*, chu kỳ đồng hồ thứ hai *load*, ... , chu kỳ đồng hồ cuối cùng *Store*

**Đơn luồng dữ liệu:** chỉ có một dữ liệu đang được sử dụng như đầu vào trong một chu kỳ đồng hồ nào đó. Chẳng hạn , hình 1.8, chu kỳ đồng hồ thứ nhất A, chu kỳ đồng hồ thứ hai B, ... , chu kỳ đồng hồ cuối A

## Nhận xét:

- Thực hiện theo tiền định, nghĩa là công việc được hoạch định rõ ràng khi lập trình (tĩnh), trong quá trình thực thi các câu lệnh và dữ liệu không bị thay đổi.
- Đây là loại máy tính lâu đời nhất và thậm chí cho đến ngày nay vẫn là loại máy tính phổ biến nhất

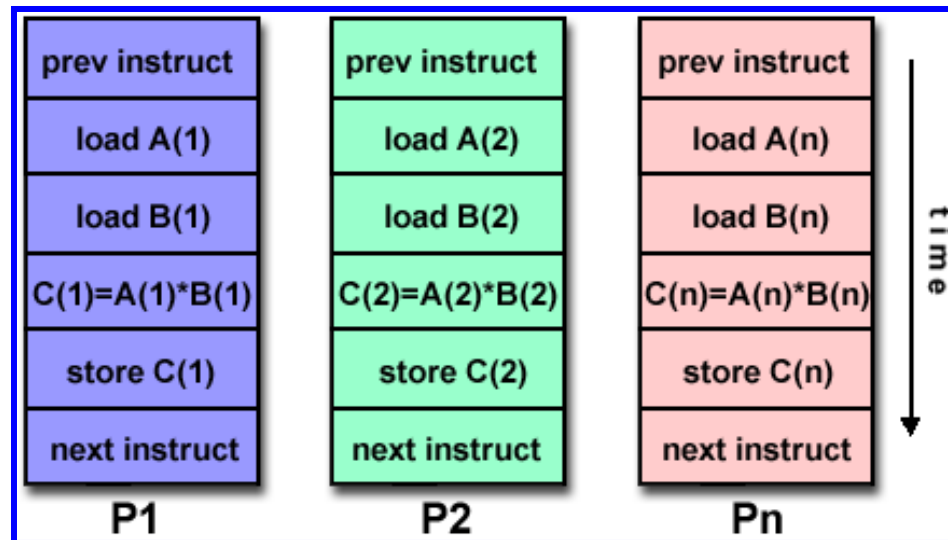
## 2. Cách phân loại thứ hai: SIMD



Hình 1-9. Kiến trúc kiến trúc SIMD

- **Đơn chỉ thị:** Các bộ vi xử lý  $p_1, p_2, \dots, p_n$  cùng thực thi một chỉ thị tại cùng một chu kỳ đồng hồ đã cho. Chẳng hạn, hình 1.8, chu kỳ đồng hồ thứ nhất,  $n$  bộ xử lý này cùng có một chỉ thị là *load*, chu kỳ đồng hồ thứ hai,  $n$  bộ xử lý cùng có một chỉ thị là *load*, ... , chu kỳ đồng hồ thứ 4,  $n$  bộ xử lý cùng có một chỉ thị là *Store*,...
- **Đa luồng dữ liệu:** Mỗi bộ xử lý có thể hoạt động trên một phần tử dữ liệu khác với dữ liệu của bộ xử lý khác - có nhiều dữ liệu đang được sử dụng như đầu vào trong một chu kỳ đồng hồ nào đó.

# Hình minh họa cách truy cập SIMD

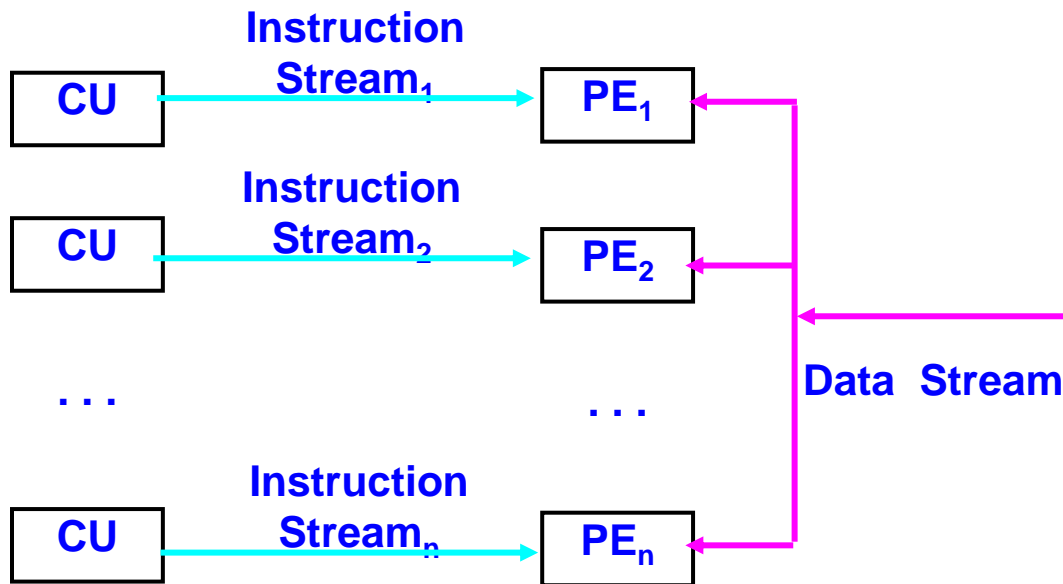


Chẳng hạn, chu kỳ đồng hồ thứ nhất nhập  $A(1), A(2), \dots, A(n)$ ; chu kỳ đồng hồ thứ hai nhập  $B(1), B(2), \dots, B(n)$ ; ... , chu kỳ đồng hồ thứ 3 là tính  $C(1), C(2), \dots, C(n)$ ; ...

# Nhận xét

- Loại máy tính này rất phù hợp cho các vấn đề chuyên ngành được đặc trưng bởi mức đồng bộ cao, chẳng hạn như xử lý đồ họa hoặc hình ảnh,....
- Đồng bộ và thực hiện tiên định.
- Hai kiến trúc bộ xử lý hay được sử dụng nhất trong trường hợp này là: Bộ xử lý mảng và xử lý kiểu đường ống vectơ (Processor Arrays and Vector Pipelines)

### 3. Cách phân loại thứ ba: MISD



Hình 1-11. Kiến trúc kiến trúc MISD

- Một dòng dữ liệu duy nhất được đưa vào các đơn vị đa xử lý.
- Mỗi đơn vị xử lý hoạt động trên các dữ liệu một cách độc lập thông qua các luồng chỉ thị độc lập khác nhau

## 4. Cách phân loại thứ tư: MIMD

**Hiện nay, các loại phổ biến nhất của máy tính song song hiện đại đều thuộc vào loại này.**

**Đa chỉ thị:**

**Mỗi bộ xử lý có thể được thực hiện một dòng chỉ thị khác với các dòng chỉ thị của các bộ xử lý khác.**

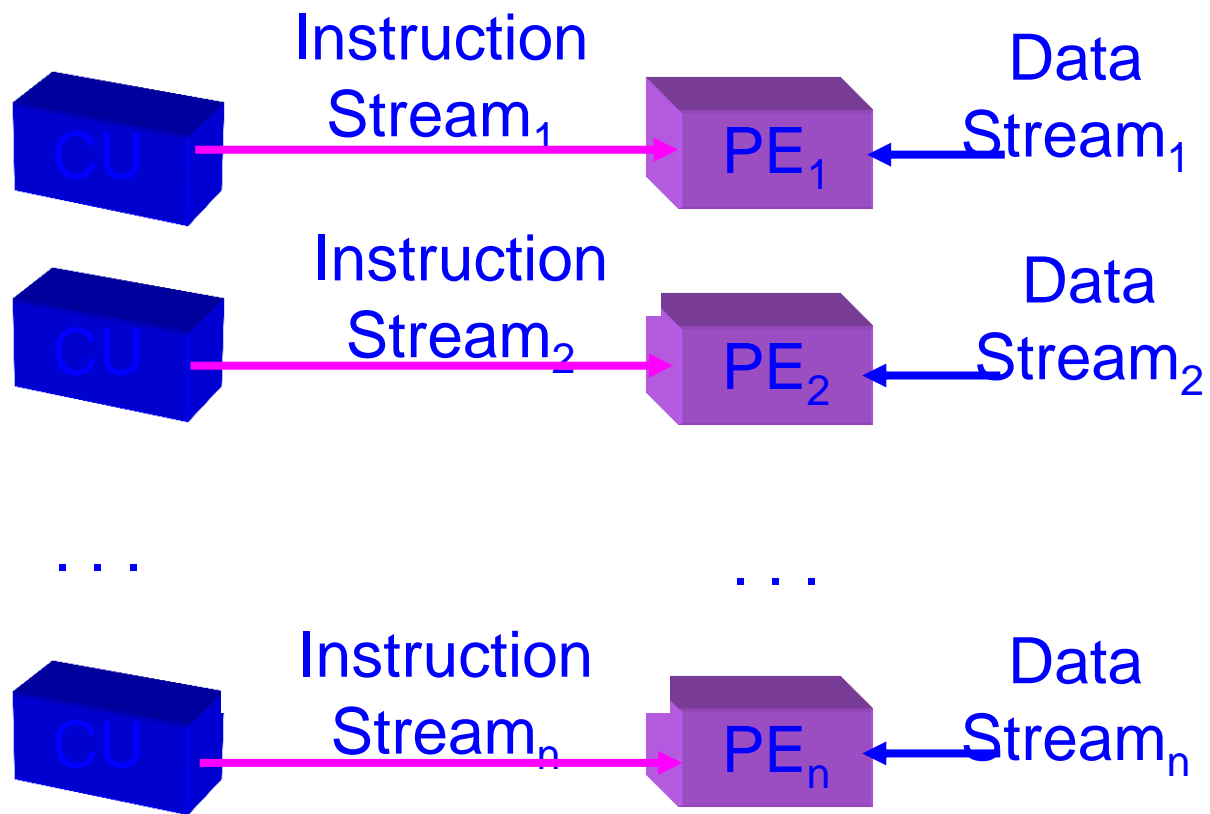
**Đa dữ liệu:**

**Mỗi bộ xử lý có thể làm việc với một dòng dữ liệu khác với các dòng dữ liệu của các bộ xử lý khác.**

**Việc thực hiện các chỉ thị có thể được đồng bộ hoặc không đồng bộ, tiền định hoặc không tiền định.**

**Lưu ý:**

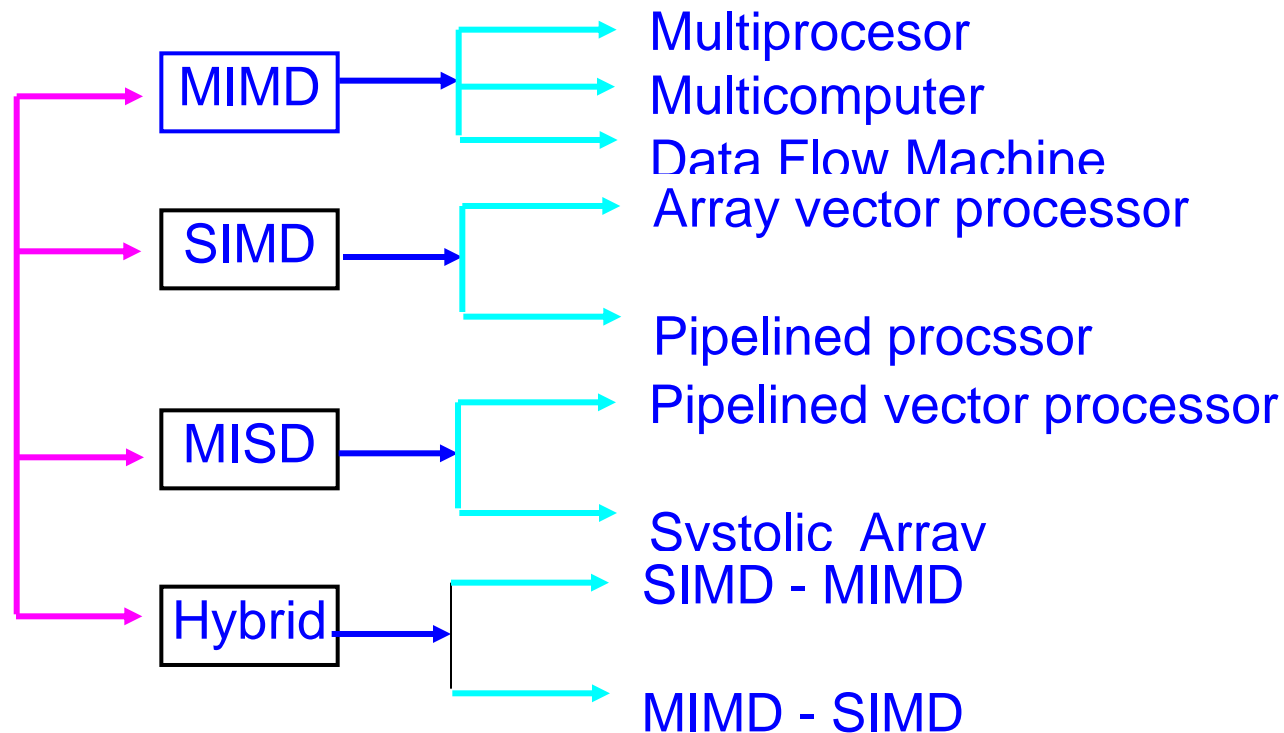
- Kiến trúc MIMD cũng có thể bao gồm các SIMD thực hiện các thành phần con.**
- Kiến trúc MIMD còn được gọi là máy đa bộ xử lý, mỗi bộ xử lý có thể thực hiện bởi luồng lệnh và luồng dữ liệu riêng**
- MIMD có bộ nhớ chung, các bộ xử lý của MIMD đều có bộ nhớ riêng, nhưng được phép truy cập vào bộ nhớ chung khi cần, do vậy giảm được các thao tác trao đổi giữa các bộ xử lý trong hệ thống, xem hình 1-12.**



Hình 1-12. Kiến trúc kiến trúc MIMD



### 1.3.3. Phân nhóm kiến trúc MT song song



**Chú ý** Các kiến trúc khác nhau sẽ tạo ra các khả năng xử lý song song khác nhau. Trong kiến trúc máy tuần tự cũng có thể tận dụng tốc độ cực nhanh của các CPU để thực hiện song song theo nguyên lý Shared time hoặc Shared Resource. Tất nhiên đối với các kiến trúc song song thì mục đích hàng đầu là khả năng chạy các chương trình song song trên đó.

# 1- Song song hoá trong máy tính tuần tự.

***a) Xử lý song song trong máy tuần tự nhờ vào đa đơn vị chức năng trong một CPU.***

Hiện nay, các máy tính có thể có nhiều đơn vị chức năng đặc biệt là các đơn vị chức năng chuyên biệt (ví dụ nhiều ALU, nhiều CU), các đơn vị chức năng này có thể thực hiện xử lý song song.

Điều cơ bản đối với máy tính đa đơn vị chức năng xử lý tuần tự là phải có bộ lập lịch tối ưu để phân chia các câu lệnh được thực hiện sao cho tận dụng được tối đa hiệu năng của các đơn vị chức năng, các tài nguyên của máy tính và phải tránh được xung đột.

# 1- Song song hoá trong máy tính tuần tự.

## ***b) Xử lý song song nhờ vào kiểu xử lý đường ống.***

Xử lý kiểu đường ống trong một CPU như sau:

Tiến trình thực hiện câu lệnh gồm các pha khác nhau, các tiến trình được tiến hành xử lý theo kiểu đường ống.

***Ví dụ 1.3.3-1.*** Giả sử cho tiến trình thực hiện câu lệnh gồm các pha sau đây:

**Nạp câu lệnh → giải mã lệnh → xác định toán hạng → thực hiện phép toán (số học/ logic) → lưu kết quả, v.v...**

Như vậy, khi một pha nào đó của tiến trình câu lệnh một kết thúc, thì pha sớm hơn của câu lệnh hai đã có thể thực hiện. Chẳng hạn:

**Chẳng hạn:**

**Câu lệnh 1 xác định toán hạng,**

**Câu lệnh 2 có thể đang giải mã lệnh**

**Câu lệnh 3 lại có thể đang nạp câu lệnh,v.v...**

**Cách xử lý kiểu đường ống là gói đầu các câu lệnh, nên bộ xử lý có thể thực hiện được nhiều câu lệnh gói đầu nhau trong cùng một thời điểm.**

***c) Xử lý song song nhờ sự gối đầu CPU và các thao tác I/O, nhờ vào phân cấp bộ nhớ và dựa vào đa chương trình và chia sẻ thời gian.***

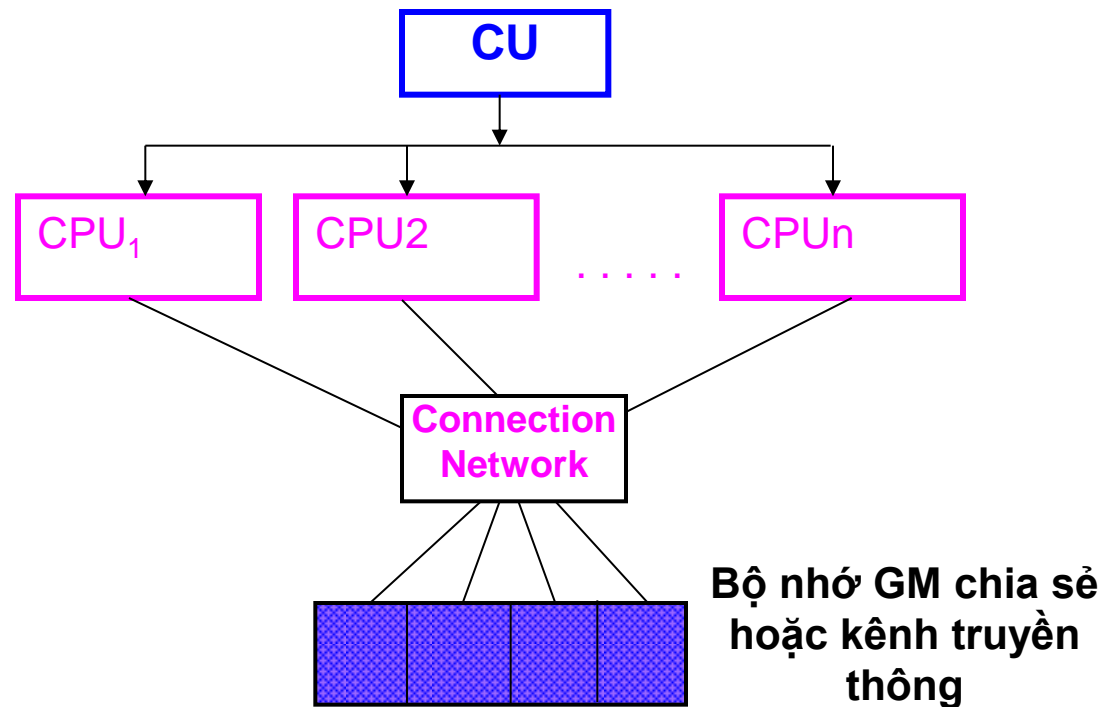
**Các phép vào/ra có thể thực hiện đồng thời đối với nhiều nhiệm vụ tính toán khác nhau bằng cách sử dụng những bộ điều khiển vào/ra, các kênh hay những bộ xử lý vào/ra khác nhau.**

**Nhiều máy tính hiện nay có nhiều bộ điều khiển thiết bị vào/ra, cho phép đa xử lý vào/ra làm tăng được tốc độ trao đổi dữ liệu giữa các thiết bị ngoài với CPU**

## 2- Kiến trúc của máy tính song song.

### Kiến trúc P-RAM (Parallel Random Access Machine)

Máy tính truy cập ngẫu nhiên song song *P-RAM* chứa một *CU*, bộ nhớ chung *GM* (General Memory) và một tập các *CPU*; Mỗi *CPU* lại có một bộ nhớ riêng *OM* (Own ). Hình 7.1-8



Hình 7.1-8. Kiến trúc P-RAM

# Một số yêu cầu của P-RAM

Mỗi CPU được gán một chỉ số duy nhất giúp cho quá trình trao đổi các tín hiệu và quản lý ngắt.

Tất cả các CPU đều chia sẻ GM với giả thiết: không hạn chế về dung lượng, không hạn chế về thời điểm và không hạn chế về khoảng thời gian truy cập.

Các câu lệnh được phép bắt đầu thực hiện ở bất kỳ thời điểm nào, bất kỳ vị trí nào của OM và GM.

Đây là kiến trúc tổng quát của máy tính song song kiểu MIMD.

# Một số yêu cầu khi phát triển thuật toán cho các máy tính song song P-RAM

- i. Không bị giới hạn về số lượng CPU;
- ii. Mọi vị trí của bộ nhớ đều truy cập được bởi bất kỳ bộ xử lý nào;
- iii. Không giới hạn về dung lượng bộ nhớ chia sẻ trong hệ thống;
- iv. Các CPU có thể đọc bất kỳ một vị trí nào của bộ nhớ (kể cả OM và GM) mà không cần đợi để các bộ xử lý khác kết thúc công việc truy cập vào bộ nhớ.

Khi chuyển các thuật toán trên máy tính song song tổng quát sang một máy tính cụ thể để lập trình song song thì cần thêm một số ràng buộc thích hợp để chương trình chạy được trên máy cụ thể đó.



## *Một số ràng buộc thường gặp*

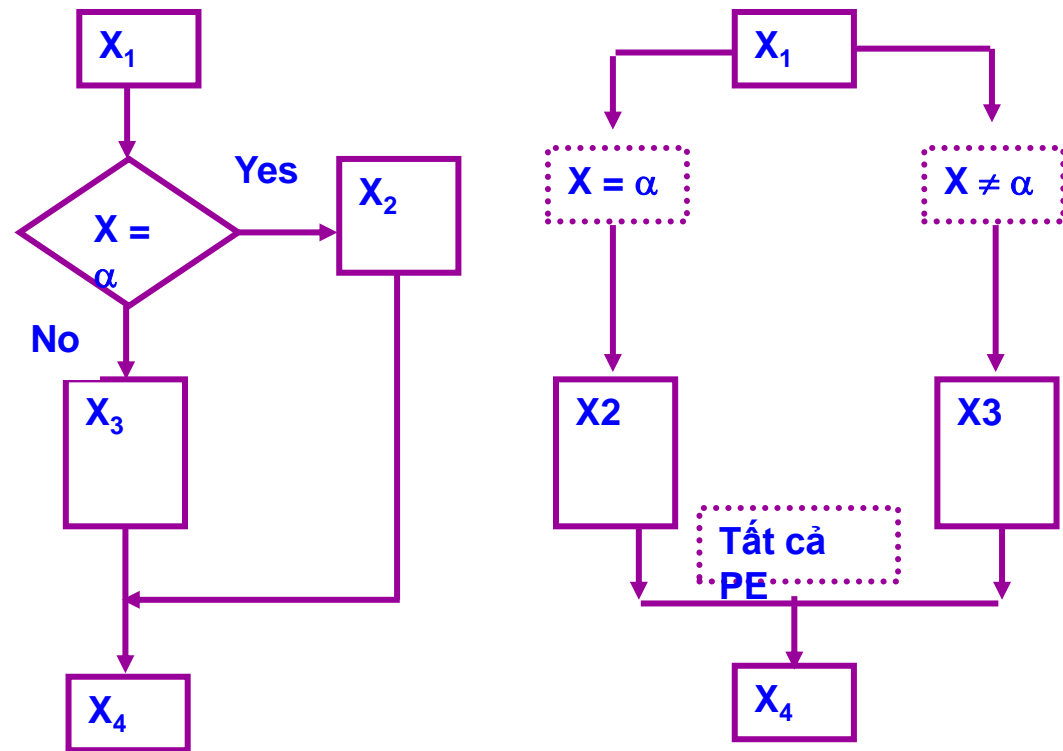
- a. EREW (Exclusive Read, Exclusive Write):**  
Tránh xung đột khi đọc và khi ghi.
- b. CREW (Concurrent Read, Exclusive Write):**  
Tránh đọc đồng thời, tránh xung đột khi ghi.
- c. CRCW (Concurrent Read, Concurrent Write):**  
Đọc đồng thời, ghi đồng thời (kiến trúc kiểu này ít dùng, không có giá trị thực tiễn).

# Kiến trúc SIMD

Chỉ có một CU điều hành mọi PE . Các PE nhận cùng một chỉ thị từ CU nhưng thực hiện trên các tập dữ liệu khác nhau, hình 7.1-8.

*Ví dụ 1.3.3-2*

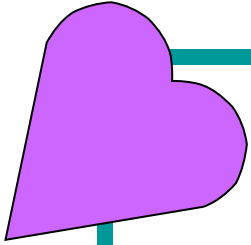
*Xét đoạn chương trình có sơ đồ khối sau, với  $x_1, x_2, x_3, x_4$  là các khối lệnh,  $X$ -data. trên hình 7.1-9*



(a) Thực hiện theo SISD

(b) Thực hiện theo SIMD

Hình 7.1-9. Xử lý của SISD và SIMD



## Giải thích các trường hợp trên Hình 7.1.9

### *Trường hợp (a): SISD*

Việc thực hiện các khối lệnh là tuần tự, sau khi thực hiện xong  $x_1$ , phải kiểm tra  $x = \alpha$  để thực hiện  $x_2$  hoặc  $x_3$  và cuối cùng là  $x_4$ .

### *Trường hợp (b): SIMD*

Có thể có một số luồng dữ liệu thoả mãn  $x = \alpha$ , còn các luồng khác lại thoả mãn  $x \neq \alpha$ , vì vậy một số PE thực hiện  $x_3$ , còn các PE khác thực hiện  $x_2$  và cuối cùng tất cả đều thực hiện  $x_4$ . Chú ý rằng không phải tất cả PE đều làm việc khi hệ thống thực hiện các khối  $x_2$  và  $x_3$ .

# Kiến trúc MISD.

Bộ xử lý MISD làm việc theo kiểu đường ống. Một tiến trình tính toán được phân đoạn thành các tiến trình nhỏ hơn gọi là pha để thực hiện liên tiếp các pha của tiến trình đó. Khi thực hiện xong các pha của tiến trình thì thực hiện tiến trình tiếp theo. Kết quả của pha trước sẽ là đầu vào của pha tiếp theo và khi một pha của tiến trình này đang thực hiện công việc của nó thì pha khác của tiến trình thứ hai có thể đang nạp dữ liệu ...

# Ví dụ

Cho một tiến trình được chia làm 4 pha

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| Pha <sub>1</sub> | Pha <sub>2</sub> | Pha <sub>3</sub> | Pha <sub>4</sub> |
|------------------|------------------|------------------|------------------|

Để thực hiện 3 tiến trình cần đến 12 pha

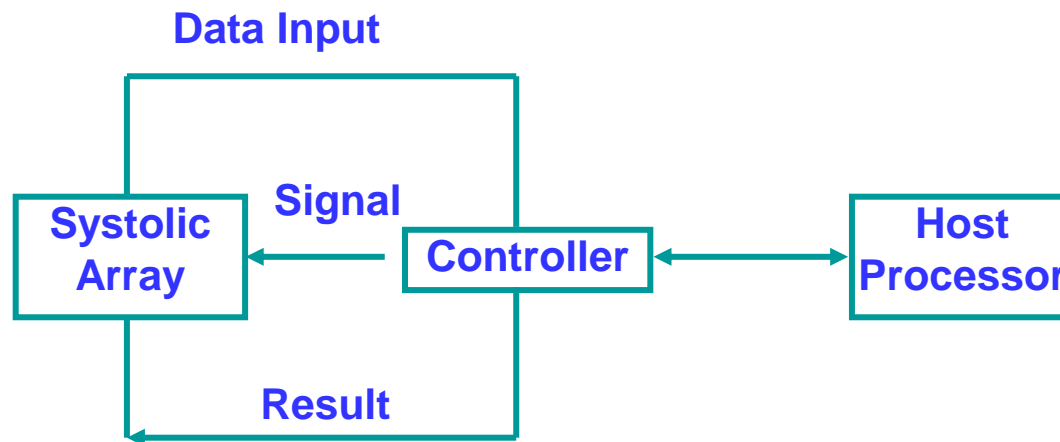
|                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Pha <sub>1</sub> | Pha <sub>2</sub> | Pha <sub>3</sub> | Pha <sub>4</sub> | Pha <sub>1</sub> | Pha <sub>2</sub> | Pha <sub>3</sub> | Pha <sub>4</sub> | Pha <sub>1</sub> | Pha <sub>2</sub> | Pha <sub>3</sub> | Pha <sub>4</sub> |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|

Khi thực hiện theo kiểu đường ống, chỉ cần 6 pha

|                  |                  |                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|------------------|------------------|
| Pha <sub>1</sub> | Pha <sub>2</sub> | Pha <sub>3</sub> | Pha <sub>4</sub> |                  |                  |
|                  | Pha <sub>1</sub> | Pha <sub>2</sub> | Pha <sub>3</sub> | Pha <sub>4</sub> |                  |
|                  |                  | Pha <sub>1</sub> | Pha <sub>2</sub> | Pha <sub>3</sub> | Pha <sub>4</sub> |

Hình 1-16. Đường ống 3 tiến trình 4 pha

## Kiến trúc SAP (Systolic Array Processor) xử lý mảng Tâm Thu.



Hình 1-17. Kiến trúc SAP

Kiến trúc của SAP có thể mô tả trên hình 1-17. Mảng gồm các đơn vị xử lý được kết nối cục bộ với nhau.

Trong SA, gồm các PE mỗi PE được gọi là là một tế bào, nó bao gồm.

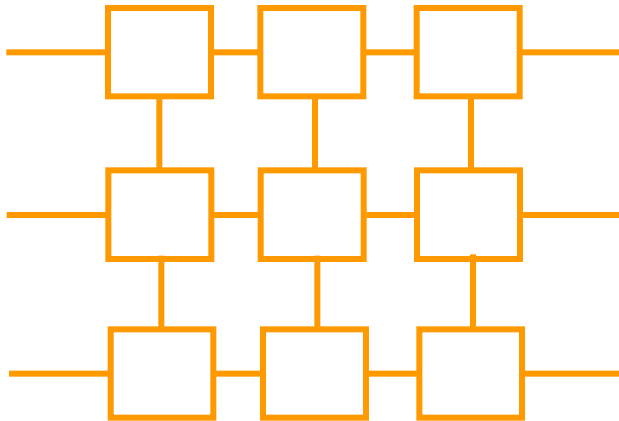
- + Một số thanh ghi (Registers);
- + Một bộ cộng (Adder)
- + Bộ ALU

Bộ điều khiển – Controller làm nhiệm vụ giao diện cho bộ xử lý chính (Host processor) và gửi tín hiệu điều khiển quá trình vào/ra dữ liệu cho SA. Hoạt động của hệ thống là theo nhịp và lặp lại một cách đều đặn để tận dụng được khả năng song song của tất cả các PE.

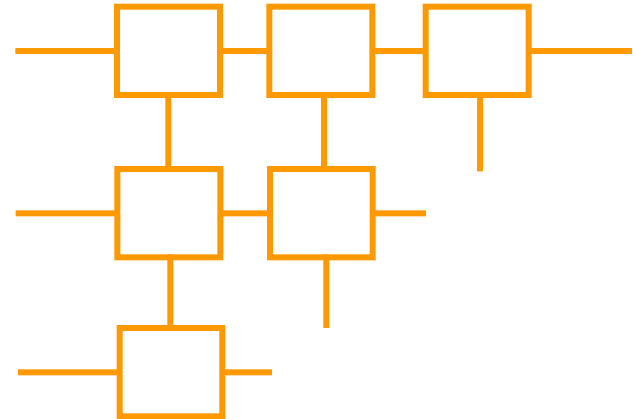
# Cấu hình phổ biến của SA



(a). Cấu hình tuyến tính



(c). Cấu hình hình vuông



(b). Cấu hình tam giác

Hình 1-18. Cấu hình phổ biến của SA

# Kiến trúc máy tính MIMD

Máy tính kiểu MIMD là loại đa bộ xử lý hoặc còn gọi là hệ thống đa máy tính, trong mỗi bộ xử lý có CU riêng và thực hiện chương trình riêng của mình MIMD có những đặc trưng sau:

- + Xử lý phân tán trên một số bộ xử lý độc lập;
- + Chia sẻ với nhau một số tài nguyên, trong đó có hệ thống bộ nhớ;
- + Mỗi bộ xử lý thao tác độc lập và có thể thực hiện đồng thời với nhau;
- + Mỗi bộ xử lý chạy một chương trình riêng.



## 1.3.4. Một số thuật ngữ song song hay dùng

**Task** - Tác vụ

**Parallel task** - Tác vụ song song

**Serial Execution** - Thực thi tuần tự

**Parallel Execution** - Thực thi song song

**Pipelining Execution** - Thực thi kiểu đường ống

**Shared Memory** - Bộ nhớ chia sẻ

**Symmetric Multi-Processor** - Bộ đa xử lý đối xứng - SMP

**Distributed Memory** - Bộ nhớ phân tán

**Communication** - Truyền thông hay giao tiếp

**Synchronzation** - Đồng bộ hóa

**Granularity** - Hạt

**Observed Speedup** - Nhận dạng sự tăng tốc

**Parallel Overhead** Chi phí ngoài

**Massively Parallel** - Đại song song (ồ ạt)

**Embarrassingly Parallel** Song song không tương minh - nhập nhằng

**Scalability** - Khả năng mở rộng

**Multi-core Processors** - Các bộ xử lý đa nhân

**Cluster Computing** - Cụm máy tính

**Supercomputing / High Performance Computing** - Siêu tính toán / Tính toán hiệu quả

**Symmetric Multi-Processor (SMP)** - Bộ đa xử lý đối xứng