

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN TOÁN ỨNG DỤNG & TIN HỌC



BÀI GIẢNG  
TÍNH TOÁN SONG SONG

Giảng viên: Đoàn Duy Trung  
Bộ môn : Toán Tin



# BÀI 2. KIẾN TRÚC HỆ THỐNG



# NỘI DUNG BÀI HỌC

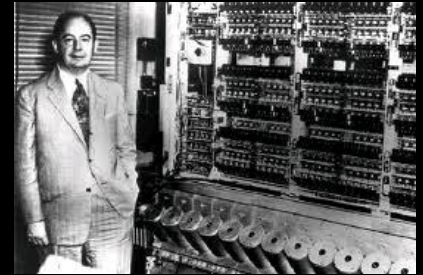
- Kiến trúc Von-Neumann
- Kiến trúc song song
  - Phân loại Flyn
  - Phân loại theo bộ nhớ
- Mô hình RAM
- Mô hình PRAM



# 1. KIẾN TRÚC VON-NEUMANN

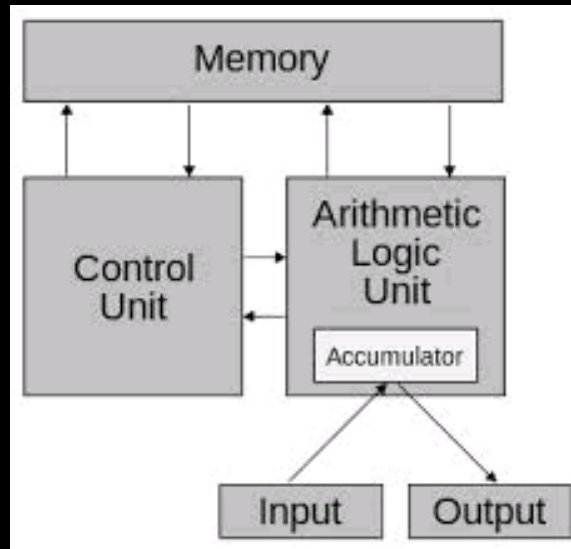


# 1.1.Kiến trúc máy tính Von-Neumann



- Đặc điểm:
  - Được đặt theo tên của nhà toán học người Hungary - John Von Neuman – người đầu tiên đưa ra yêu cầu về kiến trúc của một máy tính điện tử
  - Hầu hết các máy tính đều tuân theo cấu trúc cơ bản này
  - Sự khác nhau giữa các máy chỉ là việc xếp đặt các thiết bị ở bên trong máy tính

# 1.1. Kiến trúc máy tính Von-Neumann



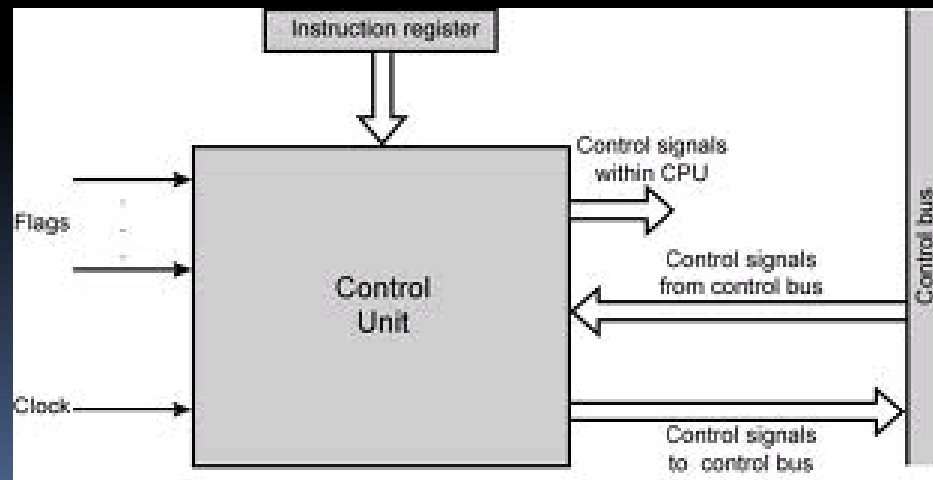
- Các thành phần chính:
  - Memory: Bộ nhớ (M)
  - Control Unit: Bộ điều khiển (CU)
  - Arithmetic Logic Unit: Bộ Logic và số học (ALU)
  - Input/Output: Thiết bị vào ra (I/O)

# 1.1.Kiến trúc máy tính Von-Neumann

- Bộ nhớ truy cập ngẫu nhiên RAM lưu trữ các lệnh chương trình và dữ liệu khi thực thi:
  - Lệnh chương trình: được mã hóa dữ liệu để yêu cầu máy tính thực hiện
  - Dữ liệu: Thông tin được sử dụng trong chương trình

# 1.1. Kiến trúc Von-Neumann

- Control Unit:
  - Lấy các lệnh chương trình và dữ liệu từ bộ nhớ, giải mã câu lệnh rồi phối hợp một cách tuần tự các phép toán, các thao tác để thực hiện công việc được lập trình sẵn





## 1.1.Kiến trúc Von-Neumann

- Arithmetic Logic Unit: Thực hiện các phép toán logic và số học cơ bản
- Input/Output: Giao tiếp với người sử dụng hoặc các thiết bị ngoại vi

# 1.1.Kiến trúc Von-Neumann

- Đặc điểm:
  - Phần lớn các máy tính cá nhân đều được thiết kế theo mô hình này.
  - Nguyên tắc làm việc:
    - Chương trình, dữ liệu vào, ra được nạp vào trong bộ nhớ chính.
    - Bộ xử lý trung tâm tuần tự lấy các mã lệnh, dữ liệu để thực hiện.
- Hạn chế tốc độ xử lý do đó nhu cầu đặt ra cần thiết kế các kiến trúc máy tính khác.



## 2. KIẾN TRÚC SONG SONG

## 2. Kiến trúc song song

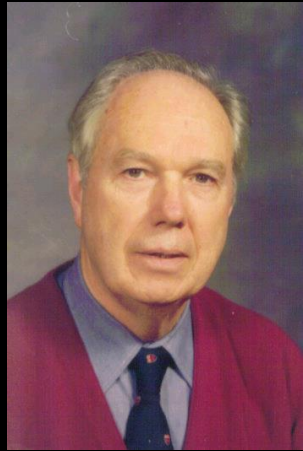
- Đặc điểm:
  - Giữ nguyên thiết kế cơ bản của Von-Neumann
  - Tập các bộ xử lý (cùng loại) kết nối với nhau theo 1 kiến trúc cùng với giao thức trao đổi dữ liệu và các quy tắc hoạt động.
  - Về cơ bản kiến trúc còn lại giống tuần tự



## 2.1. PHÂN LOẠI THEO FLYN



## 2.1. Phân loại theo Flynn



- Đặc điểm:
  - Có nhiều cách phân loại máy tính song song, phân loại sử dụng rộng rãi từ năm 1966 được gọi là Flynn's Taxonomy
  - Theo Flynn phân loại các kiến trúc máy tính bởi cách thực hiện các phép toán (instruction) trên các tập dữ liệu (data)

## 2.1. Phân loại theo Flynn

Flynn's taxonomy		
	Single instruction	Multiple instruction
Single data	SISD	MISD
Multiple data	SIMD	MIMD

- Có 4 loại chính:
  - SISD: Single Instruction Single Data
  - SIMD: Single Instruction Multiple Data.
  - MISD: Multiple Instruction Single Data
  - MIMD: Multiple Instruction Multiple Data

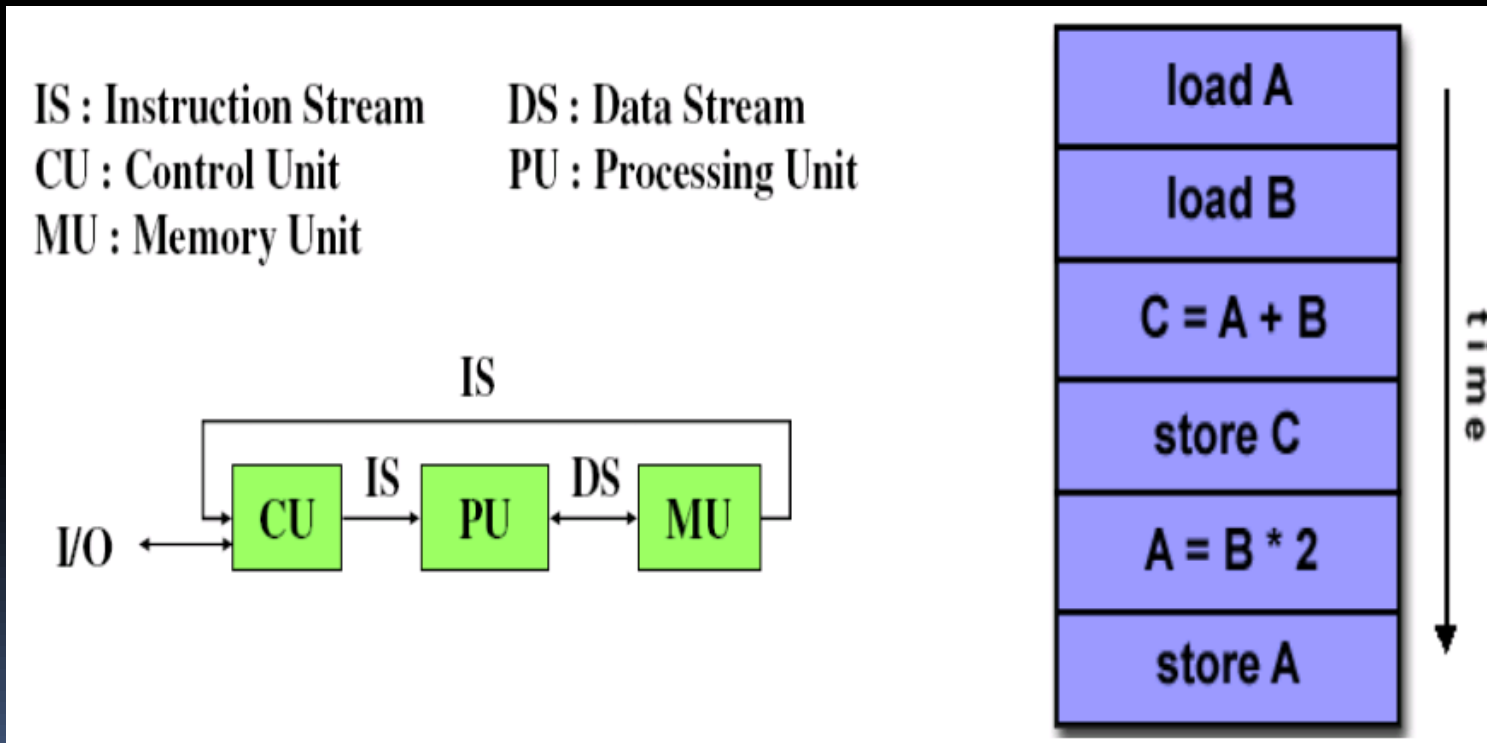
## 2.1. Phân loại theo Flynn

- Single Instruction Single Data
  - Mô hình máy tính tuần tự (Von-Neuman)
  - Single Instruction: Tại một thời điểm nào đó chỉ có một câu lệnh được thực hiện.
  - Single Data: Tại một thời điểm chỉ có một dữ liệu được thực hiện. Đơn vị dữ liệu do CPU xác định (CPU word)
  - Ví dụ:
    - Máy tính cá nhân (PC), máy trạm (workstation),...  
Đều thuộc dạng này



## 2.1. Phân loại theo Flynn

- Single Instruction Single Data

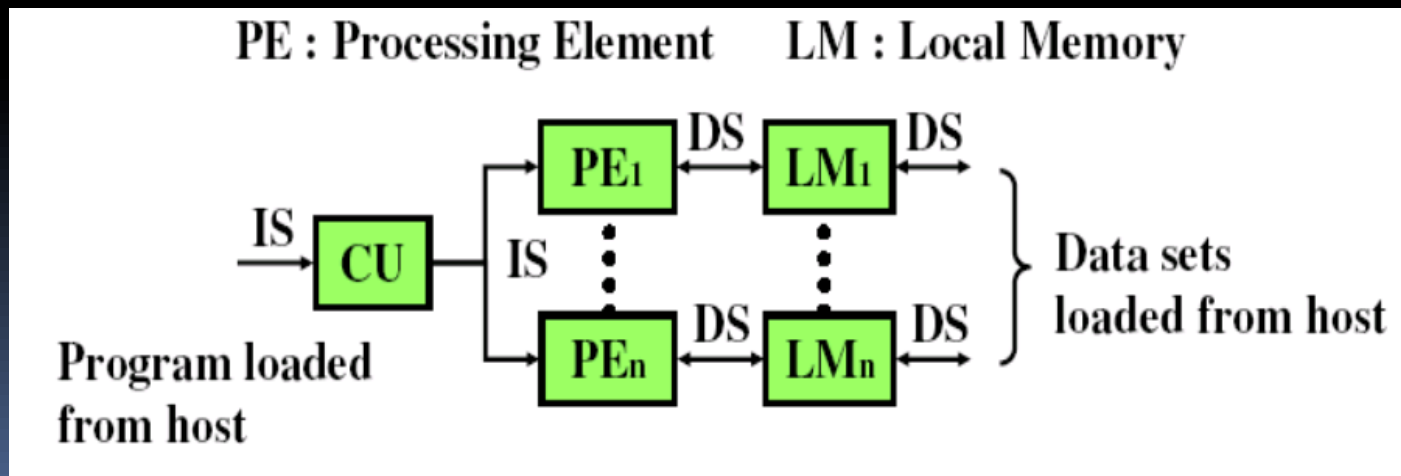


## 2.1. Phân loại theo Flynn

- Single Instruction Multiple Data:
  - Mô hình máy tính song song.
  - Single Instruction: Tất cả các bộ xử lý đều thực hiện cùng một câu chỉ thị tại cùng một chu kỳ đồng hồ đã cho
  - Multiple Data: Mỗi bộ xử lý thao tác trên một dữ liệu riêng
  - Sử dụng rộng rãi:
    - Trong các lĩnh vực mà dữ liệu có thể song song hóa
    - Xử lý đồ thị, xử lý hình ảnh

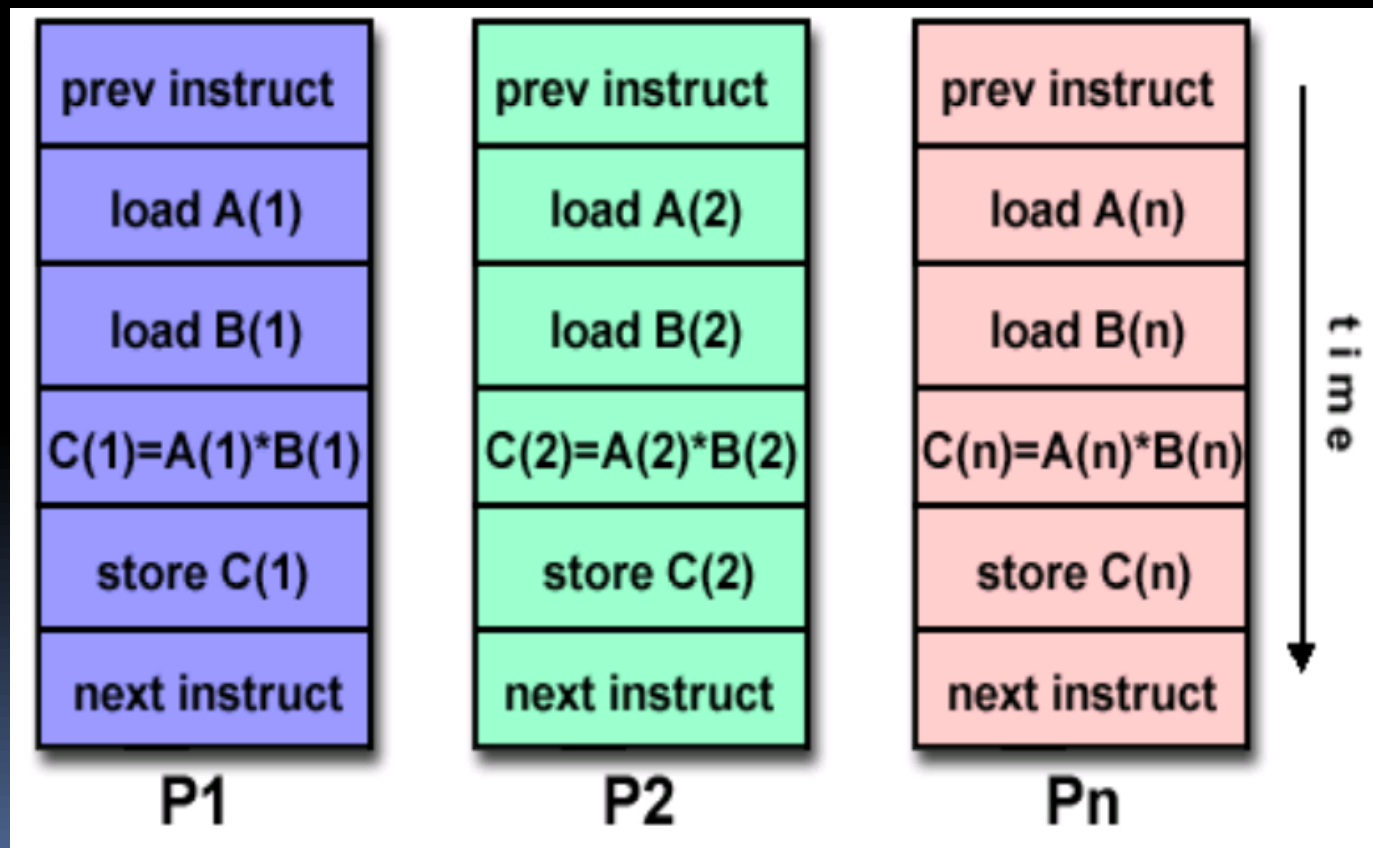
## 2.1. Phân loại theo Flynn

- Kiến trúc cơ bản:
  - 1 CU (Control Unit) điều khiển nhiều PE (Processing Elements) theo 1 luồng lệnh.
  - Cho phép các PE khác nhau (ALU khác nhau) cùng làm một phép tính trên các toán hạng khác nhau



## 2.1. Phân loại theo Flynn

- Single Instruction Multiple Data
  - ▣ Ví dụ: Câu lệnh  $C = A + B = A[i] + B[i]$



## 2.1. Phân loại theo Flynn

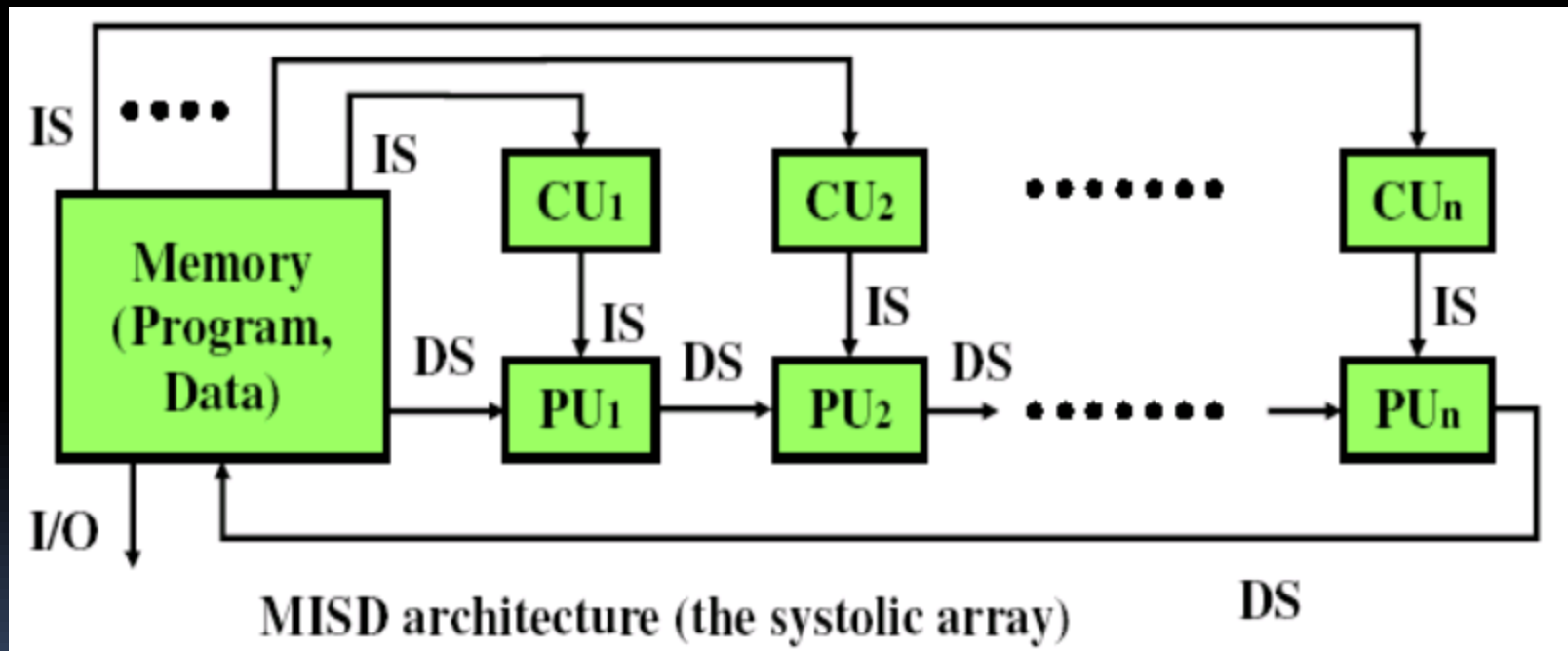
- 2 kiến trúc phổ biến của SIMD:
  - Processor Arrays: Connection Machine CM-2, MasPar MP-1 & MP-2, ILLIAC IV
  - Vector Pipelines: IBM 9000, Cray X-MP, Y-MP & C90, Fujitsu VP, NEC SX-2, Hitachi S820, ETA10
- Máy tính hiện đại, máy xử lý đồ thị (GPU) sử dụng kiểu máy SIMD

## 2.1. Phân loại theo Flynn

- Multiple Instruction Single Data
  - Là một loại máy tính song song
  - Multiple Instruction: Mỗi đơn vị xử lý hoạt động trên dữ liệu độc lập thông qua dòng câu lệnh riêng biệt
  - Single Data: Duy nhất dòng dữ liệu được đưa vào nhiều đơn vị xử lý.
  - Nguyên tắc: Một dữ liệu sẽ được thực hiện với nhiều câu lệnh trên các bộ xử lý khác nhau.
  - Ít ví dụ thực tế của loại này
  - Thử nghiệm Carnegie-Mellon C.mmp (1971)

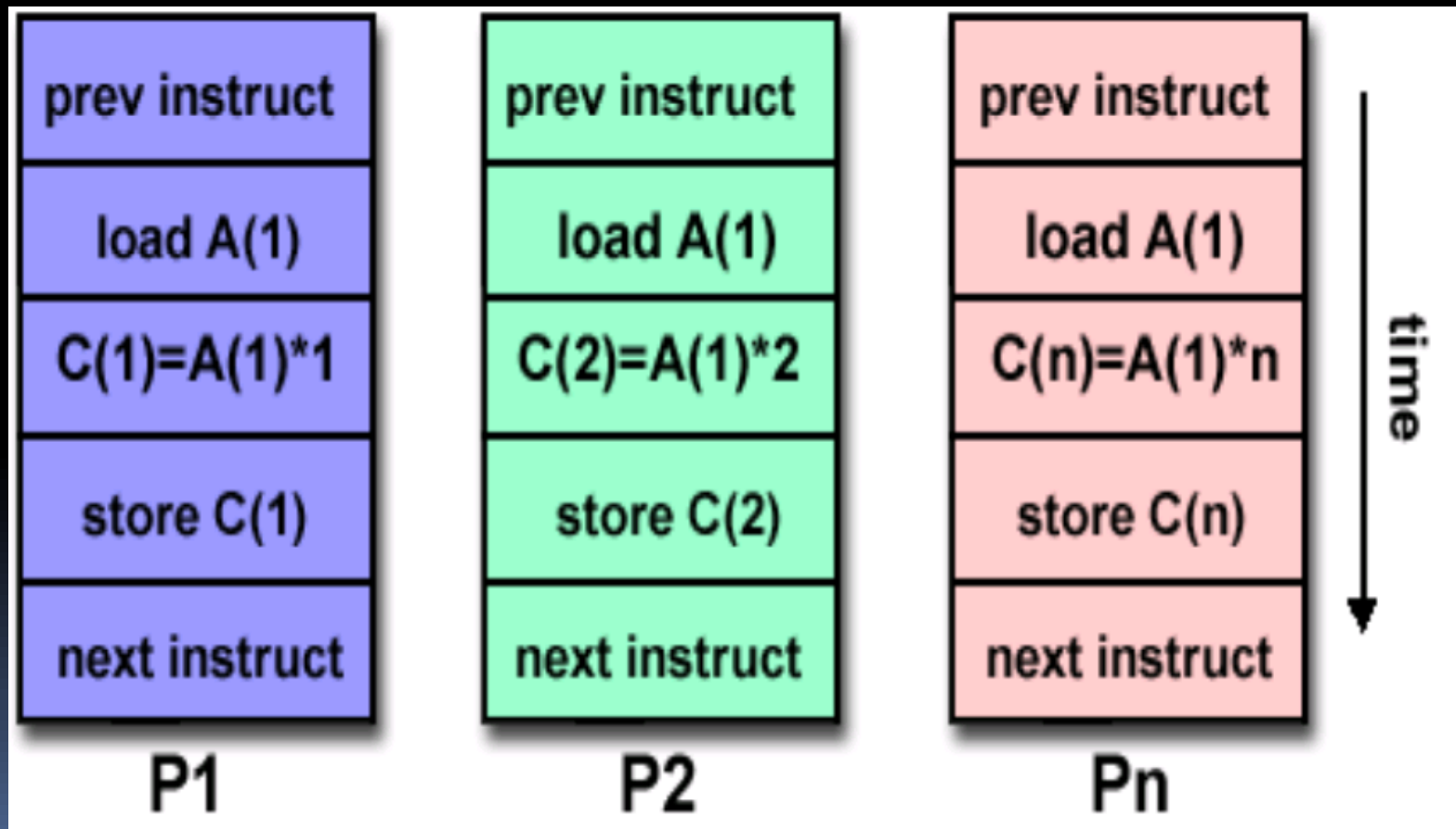
## 2.1. Phân loại theo Flynn

### Mô hình kiến trúc MISD



## 2.1. Phân loại theo Flynn

- Ví dụ sử dụng kiến trúc MISD





## 2.1. Phân loại theo Flynn

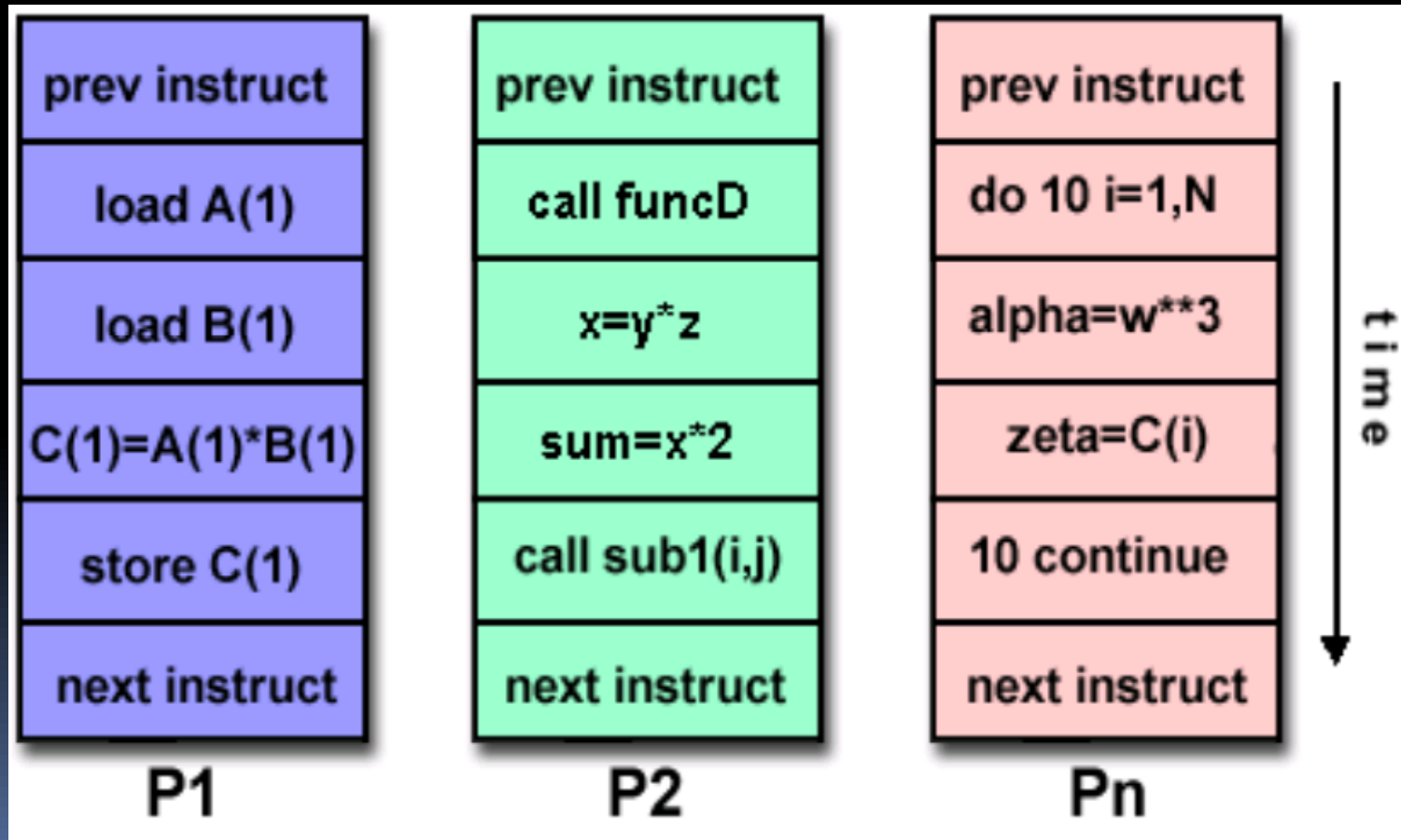
- Một vài ứng dụng sử dụng MISD:
  - Bộ lọc (Filter) nhiều tần số hoạt động trên 1 tín hiệu duy nhất
  - Nhiều thuật toán giải mã cùng cố gắng giải mã 1 tín hiệu duy nhất

## 2.1. Phân loại theo Flynn

- Multiple Instruction Multiple Data - MIMD
  - Đây là mô hình song song hiện đại nhất và phổ biến nhất hiện nay.
  - Multiple Instruction: Mỗi bộ xử lý thực hiện một câu lệnh của mình
  - Multiple Data: Mỗi bộ xử lý thao tác với dữ liệu riêng của mình.
  - Thực hiện các câu lệnh có thể:
    - Synchronous or asynchronous
    - Deterministic or non-deterministic

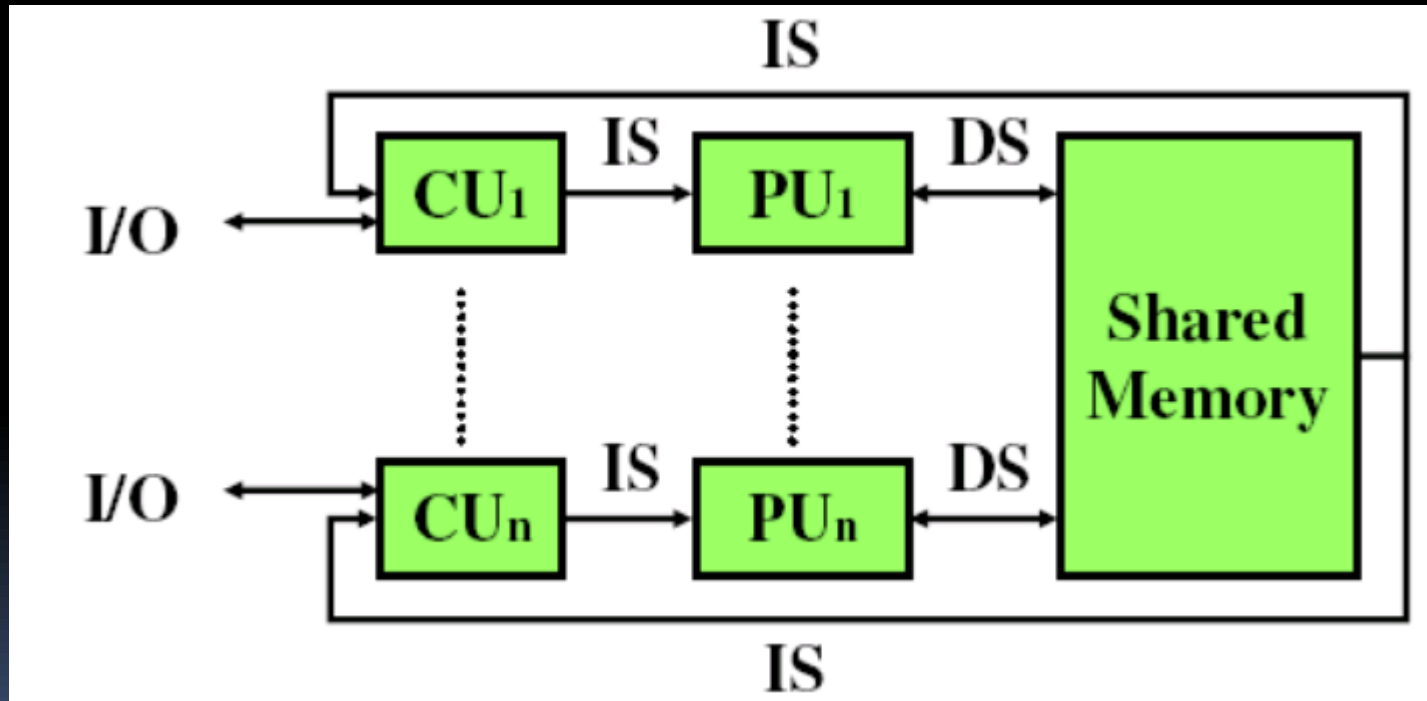
## 2.1. Phân loại theo Flynn

- Ví dụ sử dụng MIMD



## 2.1. Phân loại theo Flynn

- Mô hình kiến trúc MIMD



## 2.1. Phân loại theo Flynn

- MIMD:

- Ví dụ:

- Các Super Computer hiện nay
    - Các cụm máy tính nối mạng song song, lưới
    - Máy SMP đa bộ xử lý đối xứng
    - Máy đa lõi PCS – Personal Communication Services

- Lưu ý

- MIMD = SIMD ghép lại
    - MIMD có bộ nhớ chung bên cạnh các bộ nhớ riêng của của MIMD



## 2.2. PHÂN LOẠI THEO BỘ NHỚ

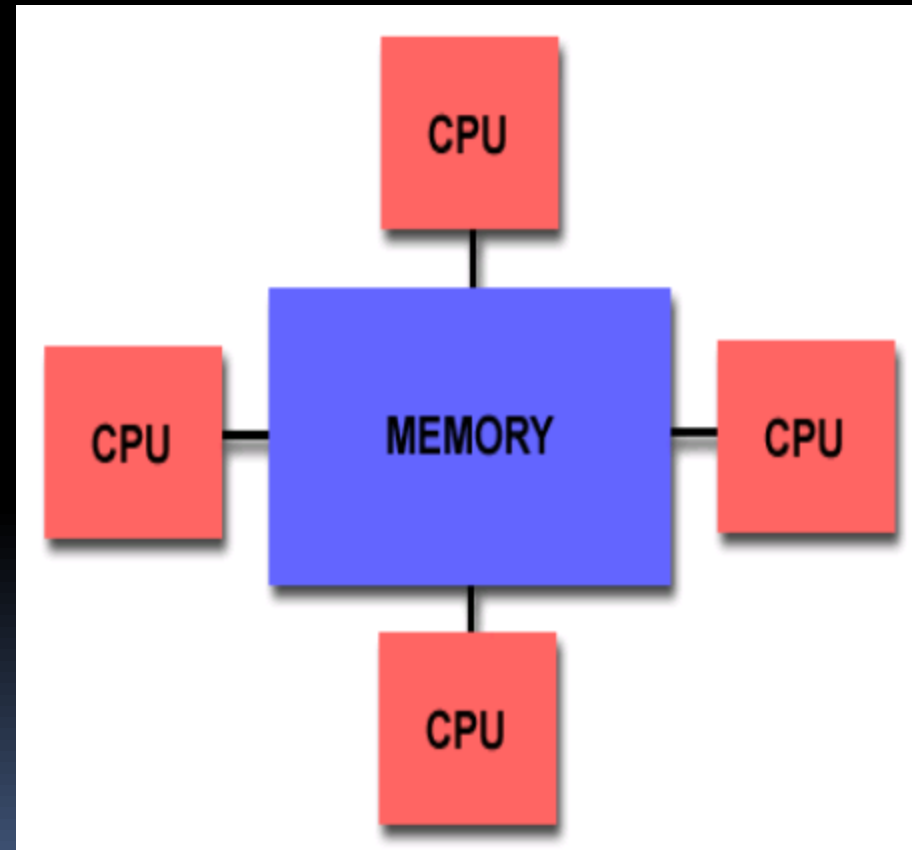


## 2.2. Phân loại theo bộ nhớ

- Đặc điểm:
  - Bộ xử lý truy cập vào bộ nhớ để lấy ra mã lệnh và dữ liệu thực hiện.
  - Phân loại theo bộ nhớ phụ thuộc vào cách truy cập vào bộ nhớ:
    - Shared memory: Bộ nhớ dùng chung.
    - Distributed memory: Bộ nhớ phân tán
    - Hybrid Distributed-Shared Memory: Mô hình bộ nhớ lai

## 2.2. Phân loại theo bộ nhớ

- Shared Memory
  - Kiến trúc bộ nhớ chia sẻ - Global Memory
  - Các bộ xử lý cùng truy cập đến bộ nhớ dùng chung thông qua BUS hệ thống.
  - Một CPU thay đổi nội dung một ô nhớ do đó các CPU khác đều nhìn thấy sự thay đổi.
  - 2 loại chính:
    - UMA
    - NUMA

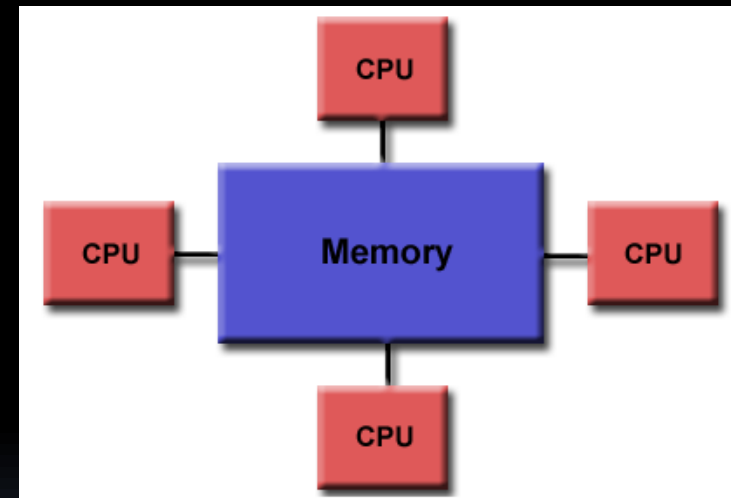




## 2.2. Phân loại theo bộ nhớ

- Shared Memory UMA (Uniform Memory Access) – Bộ nhớ chia sẻ truy cập ngang quyền:

- Các CPU giống nhau do đó mô hình này cũng có thể gọi là mô hình Symmetric MultiProcessor (SMP)
- Thời gian truy cập đến bộ nhớ là như nhau.
- Các bộ vi xử lý truy cập vào bộ nhớ chia sẻ là đồng nhất

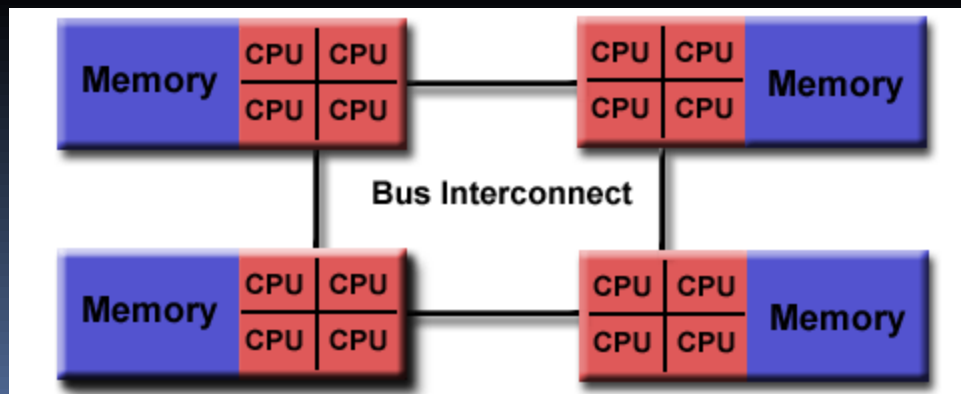


## 2.2. Phân loại theo bộ nhớ

- Shared Memory UMA:
  - Gọi là CC-UMA – Cache Coherent UMA:
    - Nếu một bộ xử lý thông tin cập nhật một vị trí trong bộ nhớ chia sẻ, thì tất cả các bộ xử lý khác đều biết đến sự cập nhật này.
    - Thực hiện ở cấp phần cứng
  - Cơ chế chuyển mạch tập trung (central switching):
    - Sử dụng chung đường dẫn (Common Bus)
    - Dùng cách chuyển mạch chéo (Crossbar Switch)
    - Dùng mạng đa giai đoạn (MultiStage Network)

## 2.2. Phân loại theo bộ nhớ

- Shared Memory NUMA – Non Uniform Memory Access – Bộ nhớ chia sẻ không ngang quyền:
  - Bộ nhớ chia thành các đơn thể độc lập, đơn thể này được phân chia cho các bộ xử lý => bộ nhớ cục bộ
  - Thể hiện bởi liên kết vật lý hai hoặc nhiều đa bộ xử lý SMPs



## 2.2. Phân loại theo bộ nhớ

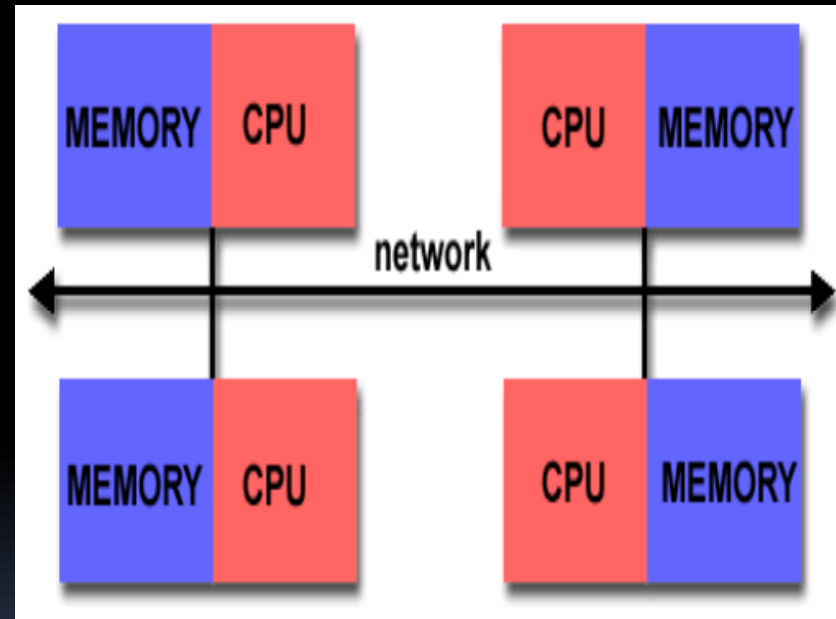
- Shared Memory NUMA:
  - Một SMP có thể truy cập trực tiếp bộ nhớ của tất cả SMPs
  - Thời gian truy cập vào tất cả bộ nhớ không như nhau
  - Truy cập bộ nhớ kết nối chéo sẽ chậm hơn
  - Ưu điểm:
    - Chia sẻ dữ liệu giữa các tác vụ nhanh và thống nhất cho các CPU gần gũi
  - Nhược điểm:

## 2.2. Phân loại theo bộ nhớ

- Shared Memory NUMA:
  - Nhược điểm:
    - Thiếu khả năng mở rộng giữa bộ nhớ và CPU
    - Chịu trách nhiệm về việc xây dựng đồng bộ hóa và đảm bảo truy cập
    - Gia tăng bộ nhớ sẽ tăng chi phí

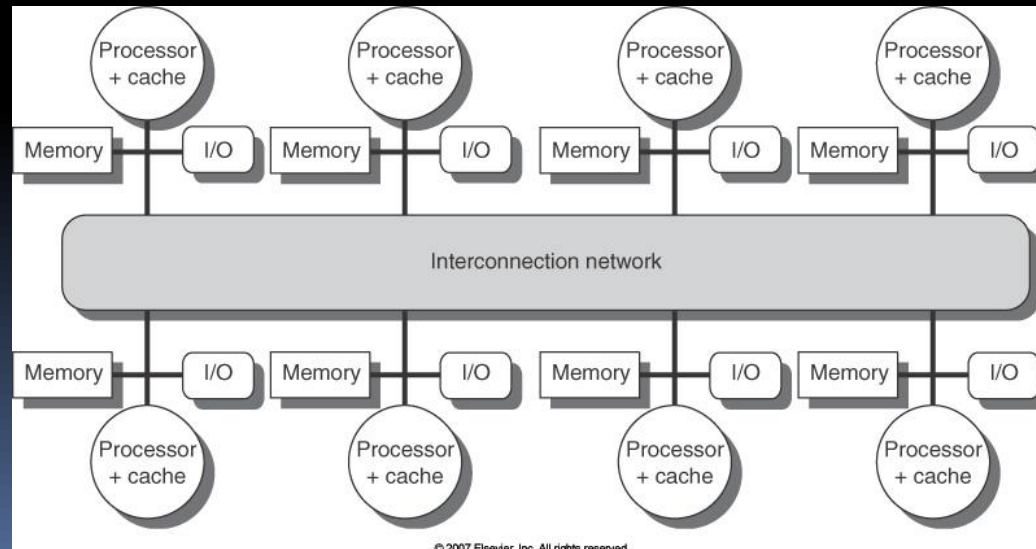
## 2.2. Phân loại theo bộ nhớ

- Distributed Memory:
  - Có mạng truyền thông để kết nối các bộ nhớ trong liên bộ xử lý
  - Liên kết được thực hiện bằng kỹ thuật truyền thông điệp
  - Thay đổi trong bộ xử lý pi không tác động vào pj



## 2.2. Phân loại theo bộ nhớ

- Distributed Memory:
  - Mỗi CPU có một bộ nhớ riêng và không cho phép các CPU khác truy cập đến:
    - Địa chỉ bộ nhớ trong một bộ xử lý không ảnh xạ đến bộ xử lý khác.
    - Không có khái niệm không gian địa chỉ toàn cục



## 2.2. Phân loại theo bộ nhớ

- Distributed Memory:
  - Mỗi bộ xử lý có bộ nhớ riêng, hoạt động độc lập:
    - Việc thay đổi bộ nhớ cục bộ không ảnh hưởng đến bộ nhớ khác.
    - Khái niệm Cache Coherence không được áp dụng
  - Khi bộ xử lý cần truy cập đến dữ liệu trong bộ xử lý khác => do lập trình viên:
    - Xác định cách nào, dữ liệu nào được truyền
    - Đồng bộ hóa giữa các task do lập trình



## 2.2. Phân loại theo bộ nhớ

- Distributed Memory:

- Ưu điểm:

- Bộ nhớ phân tán cho khả năng mở rộng tùy ý số lượng các bộ vi xử lý.
      - Bộ nhớ tỷ lệ với bộ xử lý
    - Mỗi bộ xử lý có thể truy cập nhanh chóng bộ nhớ riêng của mình mà không cần sự can thiệp và không tốn chi phí phát sinh.
    - Sử dụng thiết bị sẵn có, rẻ tiền

## 2.2. Phân loại theo bộ nhớ

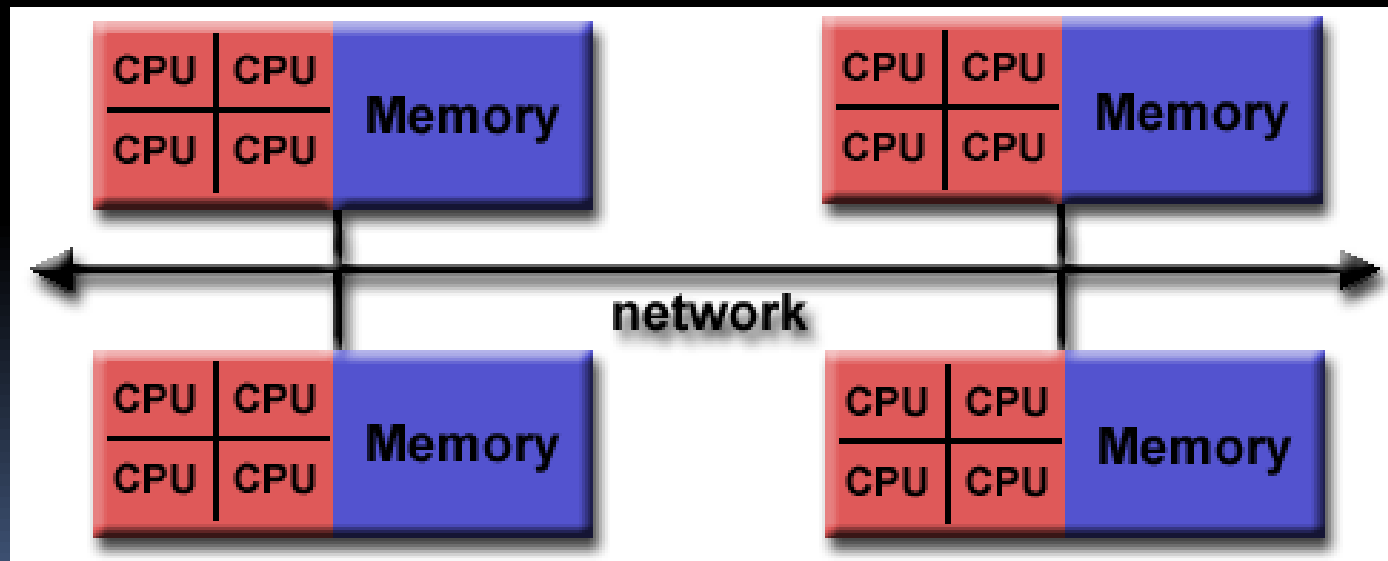
- Distributed Memory:

- Nhược điểm:

- Lập trình viên chịu trách nhiệm chi tiết liên quan đến việc truyền dữ liệu giữa các bộ xử lý.
    - Khó khăn khi cần ánh xạ đến cấu trúc dữ liệu hiện có, khi dựa vào bộ nhớ toàn cục
    - Thời gian truy cập bộ nhớ không đồng đều

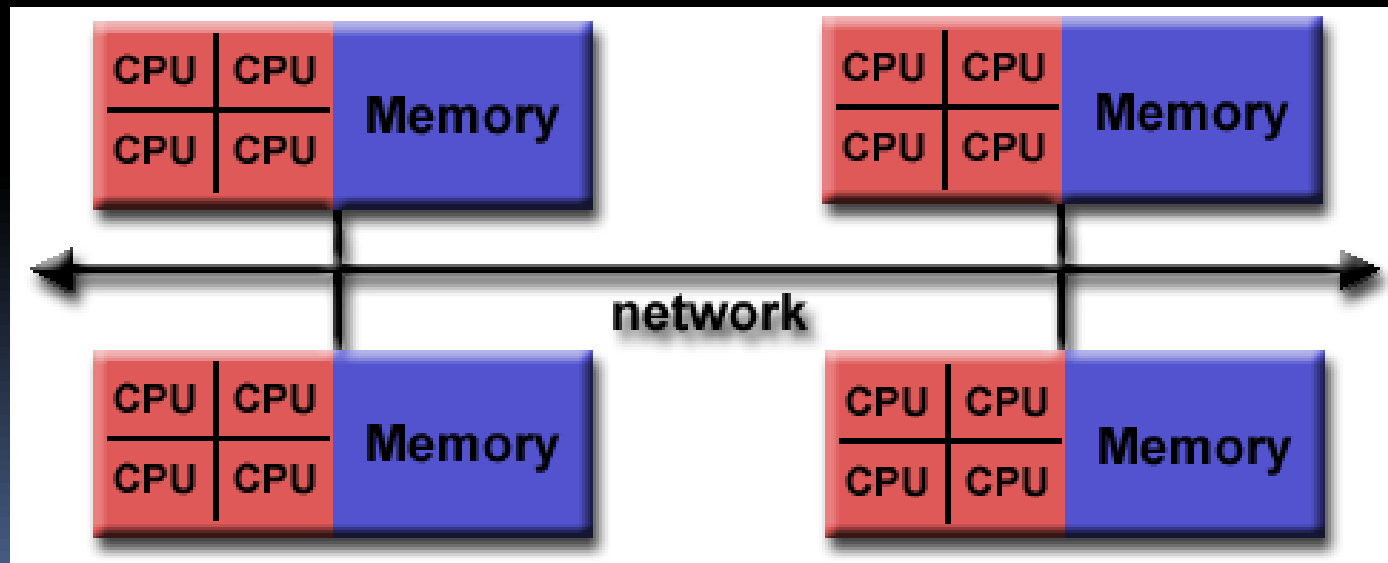
## 2.2. Phân loại theo bộ nhớ

- Hybrid Distributed - Shared Memory:
  - Các máy tính song song trên thế giới = Distributed Memory + Shared Memory



## 2.2. Phân loại theo bộ nhớ

- Hybrid Distributed – Shared Memory:
  - Các thành phần của bộ nhớ chia sẻ có thể là một máy SMP với cache coherence hoặc (và) là GPU (Graphic Processing Unit)



## 2.2. Phân loại theo bộ nhớ

- Hybrid Distributed – Shared Memory:
  - Các thành phần bộ nhớ phân tán là mạng cyar các máy SMP / GPU:
    - Do đó chỉ biết bộ nhớ của chính nó mà không biết bộ nhớ của máy khác.
    - Giao tiếp mạng là cần thiết để truyền dữ liệu
  - Kiến trúc này tiếp tục phát triển trong tương lai
  - Ưu / nhược điểm: Đầy đủ tính chất 2 loại trên



### 3. CÁC PHƯƠNG THỨC TRUY CẬP BỘ NHỚ

### 3. Các phương thức truy cập bộ nhớ

- Bao gồm 4 loại
  - CR (Concurrent Read – Đọc đồng thời)
  - ER (Exclusive Read – Đọc loại trừ)
  - CW (Concurrent Write – Ghi đồng thời)
  - EW (Exclusive Write – Ghi loại trừ)

### 3. Phương thức truy cập bộ nhớ

- Concurrent Write:
  - Priority CW – Ghi đồng thời có ưu tiên
  - Common CW – Ghi đồng thời chung
  - Arbitrary CW – Ghi đồng thời tự do
  - Random CW – Ghi ngẫu nhiên
  - Combining CW – Ghi đồng thời hỗn hợp





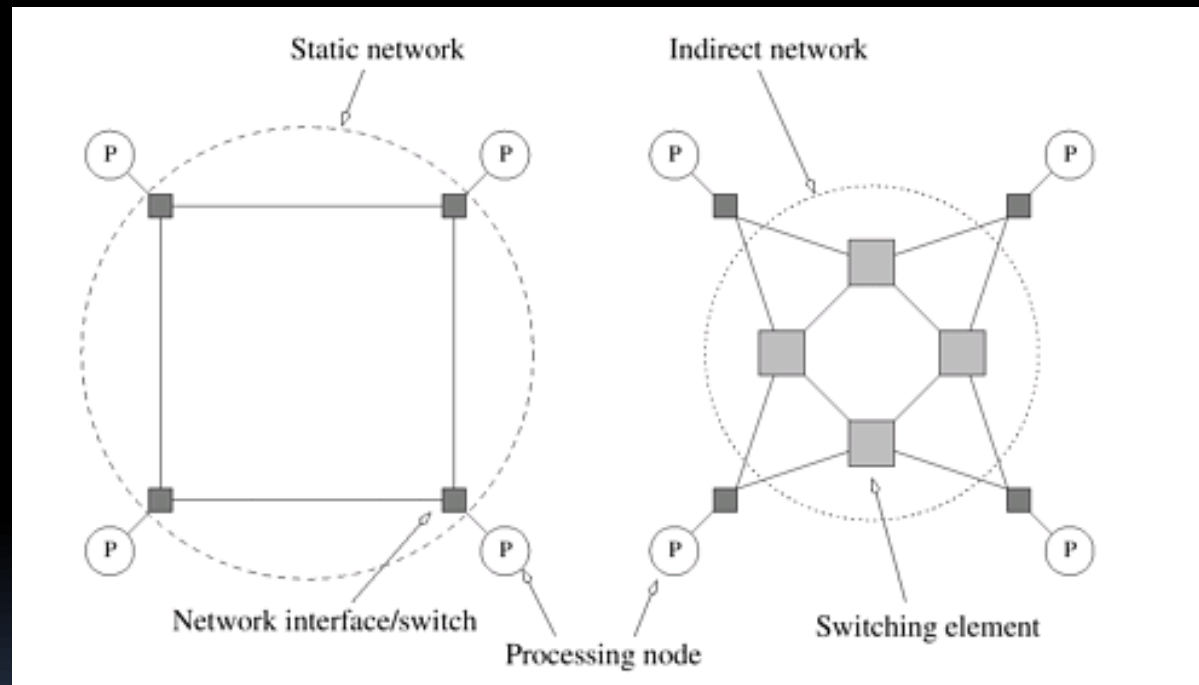
## 4. CÁC KIỂU KẾT NỐI CÁC BỘ XỬ LÝ CỦA HỆ THỐNG ĐA BỘ XỬ LÝ



## 4. Các kiểu kết nối

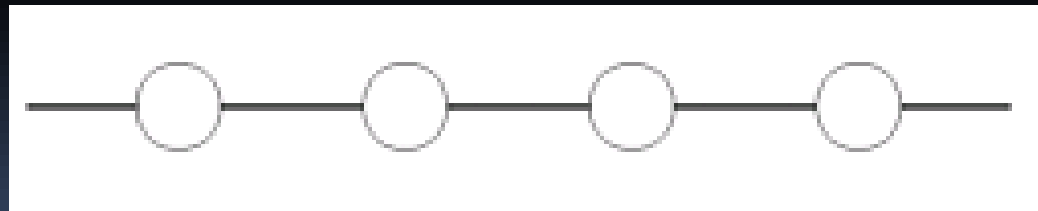
- 2 loại kết nối cơ bản:
  - Kết nối tĩnh: bộ xử lý, bộ nhớ được liên kết với nhau một cách cố định
  - Kết nối động: bộ xử lý, bộ nhớ được liên kết với nhau không cố định, có thể thay đổi được
  - Thể hiện bởi đồ thị  $G(V,E)$ 
    - $V$ : tập các đỉnh
    - $E$ : tập các cạnh

## 4. Các kiểu kết nối



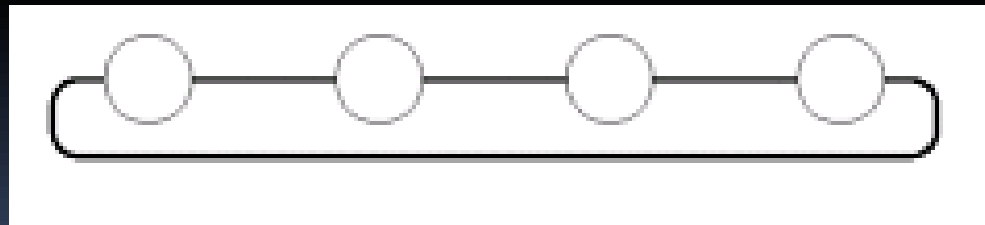
## 4.1. Đồ thị kết nối tuyến tính

- Đặc điểm:
  - Liên kết với nhau theo dãy và được đánh số theo thứ tự tăng dần
  - Dạng liên kết đơn giản, nhưng dữ liệu lại phải chuyển qua nhiều bộ xử lý => chậm chễ khi truyền dữ liệu



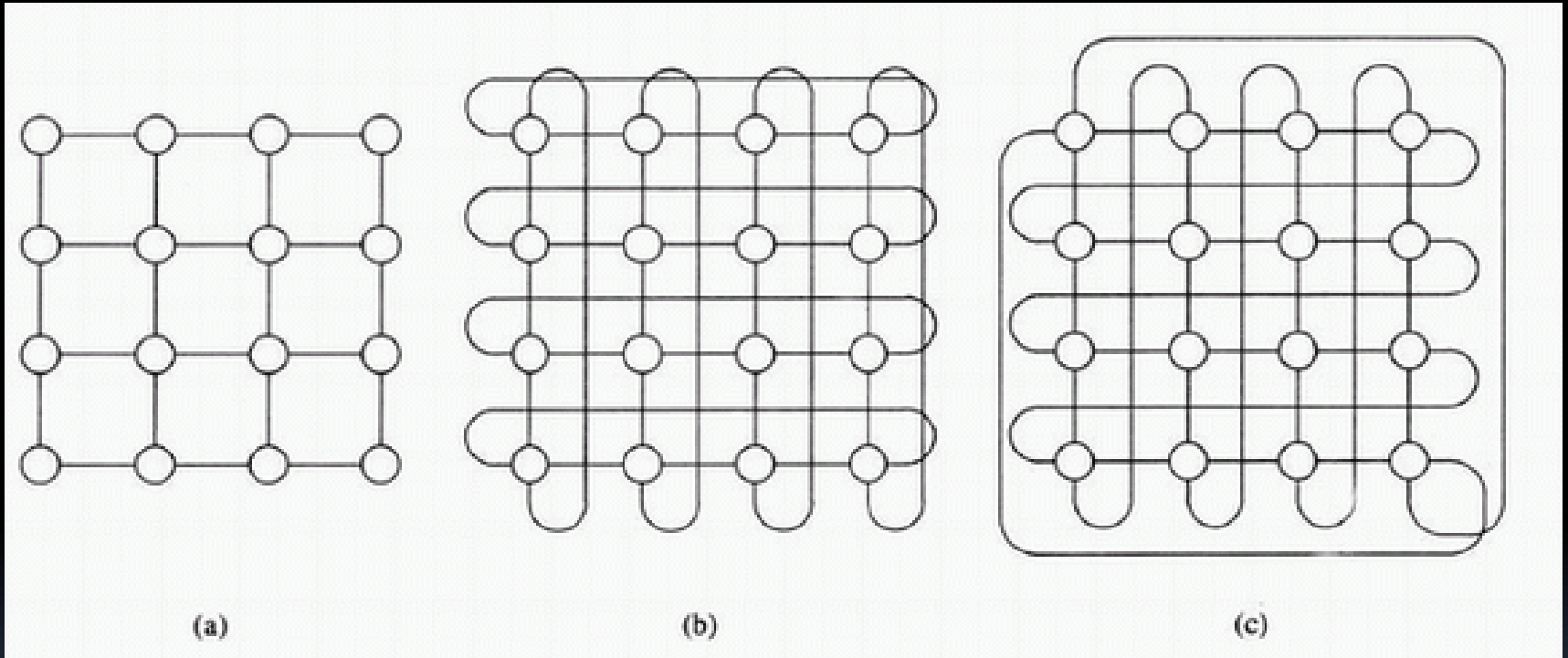
## 4.2. Đồ thị kết nối vòng

- Đặc điểm:
  - Là cách kết nối tuyến tính nhưng bộ xử lý đầu và cuối được nối lại với nhau
  - Trao đổi dữ liệu theo 1 chiều => mạng đơn
  - Trao đổi dữ liệu theo 2 chiều => mạng kép



## 4.3. Đồ thị kết nối Mesh

- Đặc điểm



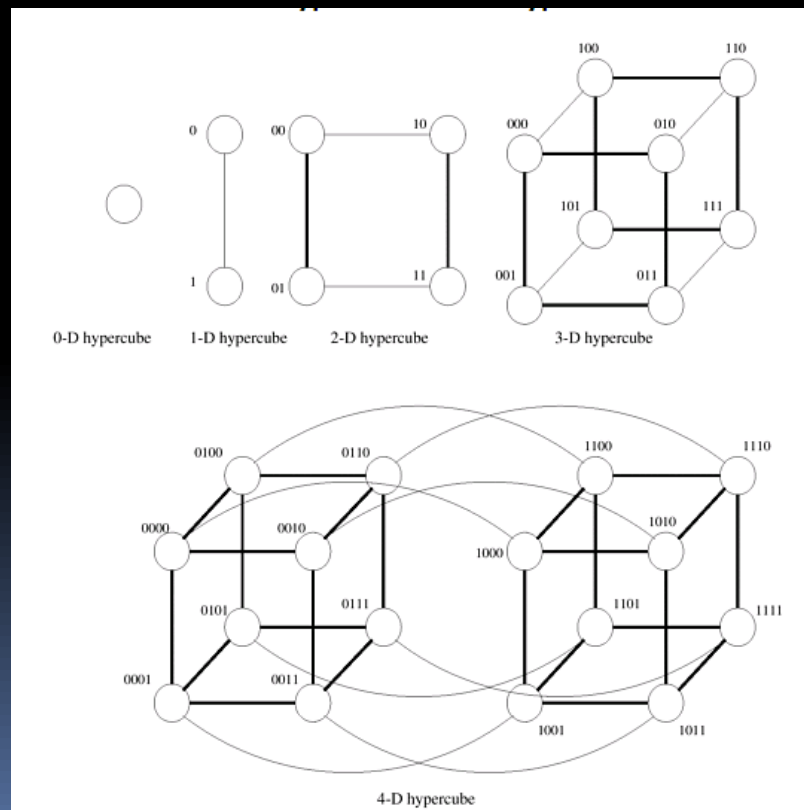
Không có liên kết bọc xung quanh

Có liên kết bọc xung quanh giữa các bộ xử lý trên cùng dòng hoặc cùng cột

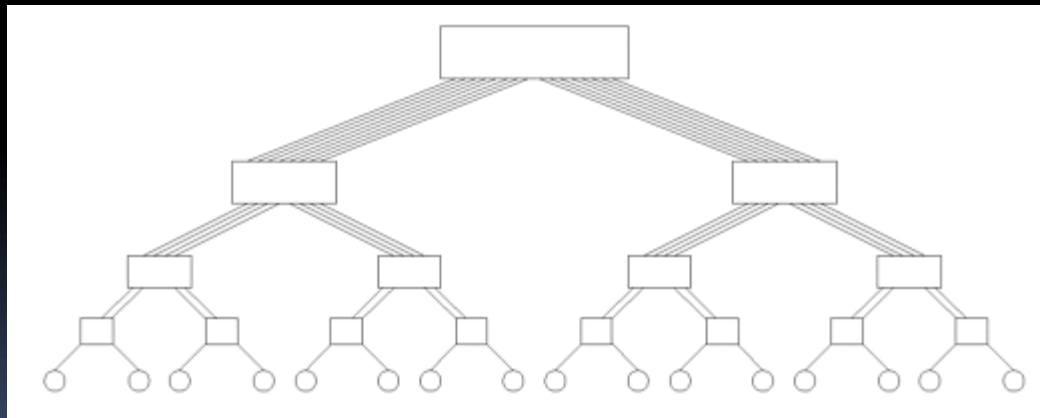
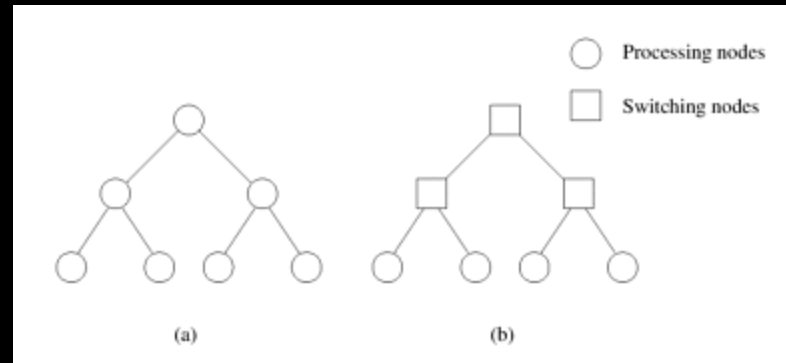
Liên kết bọc xung quanh giữa các bộ xử lý trên dòng hoặc cột kề cận

## 4.4. Đồ thị kết nối n-chiều (khối)

- Đặc điểm:
  - n BXL ( $n=2^q$ ), mỗi BXL liên kết với q BXL lân cận  
 $\Rightarrow$  mạng liên kết siêu khối n chiều



## 4.5. Đồ thị kết nối Tree Based Network





## 4.6. Đánh giá đối với mạng tĩnh có $p$ nút

Network	Diameter	Bisection Width
Completely-connected	1	$p^2/4$
Star	2	1
Complete binary tree	$2 \log((p + 1)/2)$	1
Linear array	$p - 1$	1
2-D mesh, no wraparound	$2(\sqrt{p} - 1)$	$\sqrt{p}$
2-D wraparound mesh	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$
Hypercube	$\log p$	$p/2$
Wraparound $k$ -ary $d$ -cube	$d\lfloor k/2\rfloor$	$2k^{d-1}$



# 5. CHƯƠNG TRÌNH DỊCH & HỆ ĐIỀU HÀNH



## 5.1. Chương trình dịch

- 3 cách tiếp cận chương trình dịch cho máy song song:
  - Phân hoạch khi thực thi và lập lịch khi thực thi
  - Phân hoạch khi dịch chương trình và lập lịch khi thực thi chương trình
  - Phân hoạch khi dịch chương trình và lập lịch khi dịch chương trình

## 5.2. Hệ điều hành

- Thực hiện các chức năng chính:
  - Khởi động hệ thống
  - Phân đoạn chương trình và lập lịch cho các tiến trình
  - Trao đổi và đồng bộ hóa các tiến trình
  - Quản lý và điều hành hệ thống

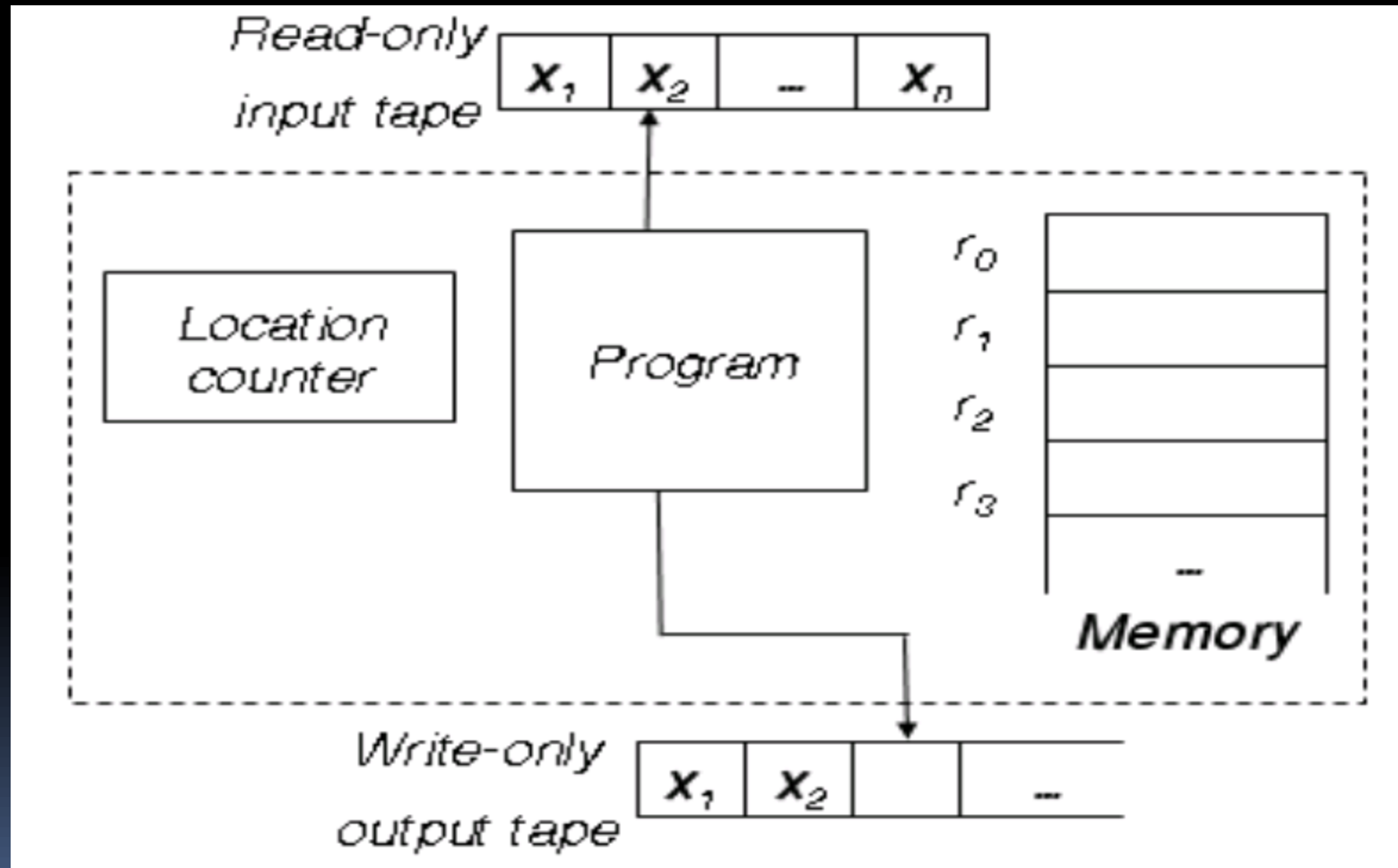
## 5.2. Hệ điều hành

- Hệ điều hành đa bộ xử lý
  - Quản lý tiến trình
  - Quản lý bộ nhớ
  - Quản lý tài nguyên
  - Quản lý tệp
  - Chia làm 3 loại:
    - Hệ điều hành mở rộng phát triển từ hệ đơn: VMS, VNIIX
    - Hệ điều hành thiết kế riêng cho kiến trúc song song: Hydra, Medusa...
    - Hệ điều hành tổng hợp thiết kế cài đặt trên các kiến trúc song song: MACH Multi processor



## **6. RANDOM ACCESS MACHINE & PARALLEL RANDOM ACCESS MACHINE**

## 6.1. Random Access Machine



## 6.1. Random Access Machine

- Đặc điểm:
  - Cấu trúc của máy RAM được xác định thông qua các thành phần như sau:
    - Khối xử lý chương trình thực hiện các câu lệnh
    - Con trỏ lệnh đánh dấu thứ tự thực hiện các lệnh
    - Bộ nhớ được xác định bằng chuỗi vô hạn các thanh ghi.
    - 1 đầu đọc các dữ liệu đầu vào.
    - 1 đầu ghi dữ liệu đầu ra.
    - Các dữ liệu được chuyển vào các thanh ghi để xử lý



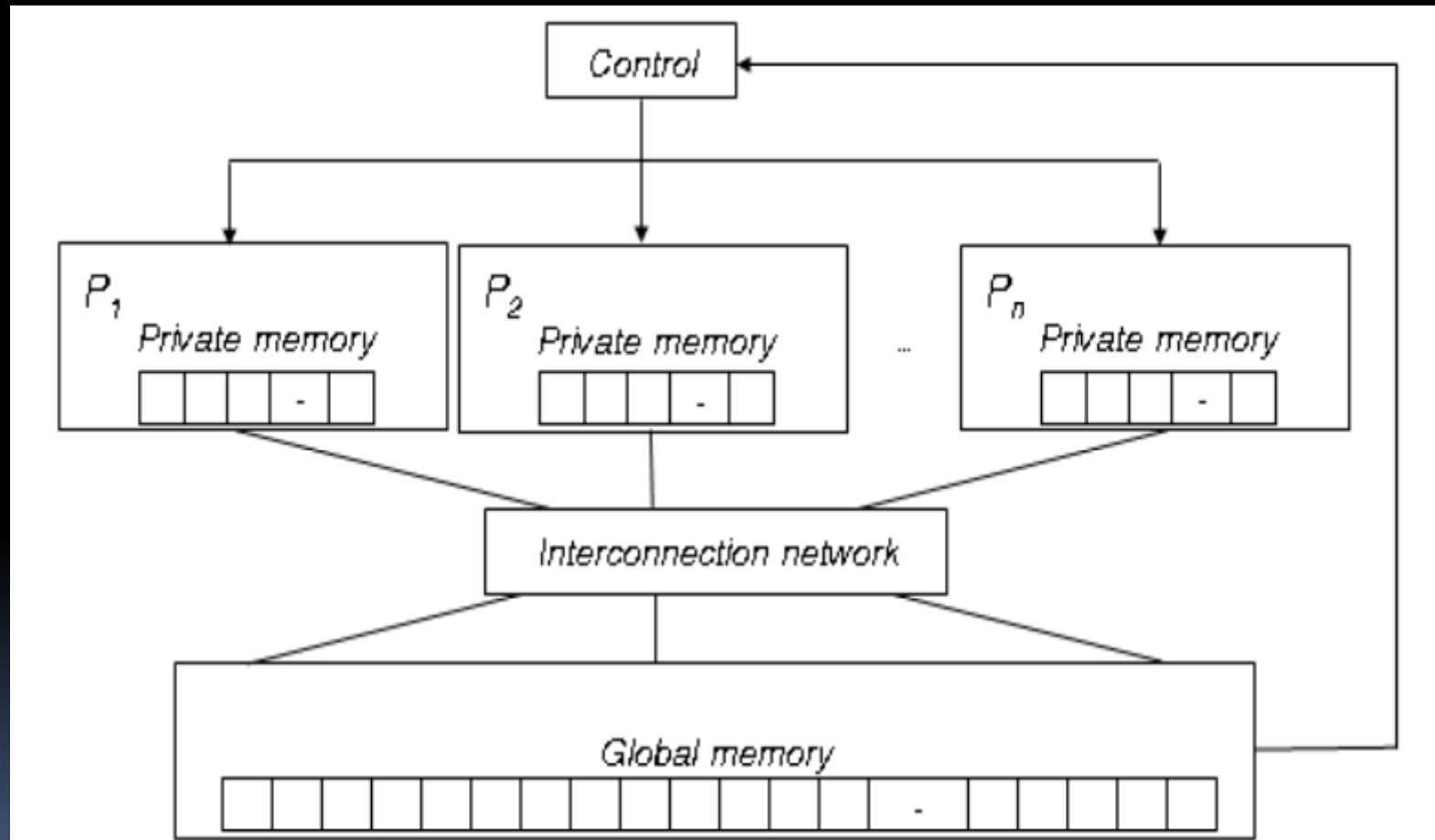
## 6.1. Random Access Machine

- Đặc điểm:
  - Các phép toán trong RAM cũng không quá quan trọng như các phép toán trong thực tế. Nó chỉ mô phỏng một số phép toán cơ bản:
    - Lấy dữ liệu từ bộ nhớ, ghi dữ liệu vào bộ nhớ.
    - Các phép toán số học, logic cơ bản.
    - Các phép toán liên quan đến xử lý câu lệnh chi nhánh (các phép toán nhảy, lặp...)
    - Phép toán đánh dấu kết thúc chương trình.

## 6.1. Random Access Machine

- Đặc điểm:
  - Thời gian thực hiện các phép toán được coi như bằng 1 đơn vị thời gian bất kể độ dài khác nhau của các toán hạng.
  - Thời gian tính sẽ được xác định bằng số phép toán cần phải thực hiện

## 6.2. Parallel Random Access Machine



## 6.2. Parallel Random Access Machine

- Đặc điểm:

- Không giới hạn số bộ xử lý RAM:  $P_1, P_2 \dots P_n$  và có 1 khối điều khiển các bộ xử lý này.
- Sở hữu bộ nhớ dùng chung không giới hạn kích thước  $M_1, M_2, \dots$
- Mỗi bộ xử lý RAM đều được đánh dấu bởi 1 số thứ tự riêng và trong nó có 1 vùng nhớ cục bộ riêng.
- Quản lý hoạt động của bộ xử lý thông qua chỉ số của nó

## 6.2. Parallel Random Access Machine

- Đặc điểm:
  - Mỗi bộ xử lý có thể truy cập đến 1 số lượng ô nhớ tùy ý, trừ khi vùng bộ nhớ đó đang bị busy bởi các bộ xử lý khác
  - Dữ liệu đầu vào / ra được lưu trữ trong bộ nhớ dùng chung.
  - Các phép toán cơ bản trong PRAM:
    - Lấy dữ liệu từ bộ nhớ dùng chung
    - Thực hiện phép toán với dữ liệu đó trong máy RAM.
    - Ghi kết quả ra bộ nhớ dùng chung

## 6.2. Parallel Random Access Machine

- Đặc điểm:
  - Các bộ xử lý sẽ thực hiện các câu lệnh này theo nguyên tắc đồng bộ hóa.
  - Chỉ có 1 cách duy nhất thực hiện việc trao đổi thông tin giữa các bộ xử lý đó là ghi và đọc thông tin đó từ các ô nhớ trong bộ nhớ dùng chung.

## 6.2. Parallel Random Access Machine

- Phân loại Parallel Random Access Machine
  - Do các máy RAM cùng dùng chung bộ nhớ để đọc/ghi dữ liệu -> vấn đề tranh chấp khi cùng thao tác trên 1 ô nhớ.
  - Phân loại theo cơ chế đọc / ghi dữ liệu:
    - EREW: Exclusive Read Exclusive Write.
    - CREW: Concurrent Read Exclusive Write
    - CRCW: Concurrent Read Concurrent Write

## 6.2. Parallel Random Access Machine

- CRCW: Concurrent Read Concurrent Write
  - CONSISTENT – CRCW – PRAM
  - ARBITRARY – CRCW - PRAM
  - PRIORITY – CRCW – PRAM
  - FUSION – CRCW – PRAM
    - Toán tử  $\circ$  sẽ áp dụng với các giá trị  $v_1 \dots v_k$  ghi vào L điều đó dẫn đến  $v_1 \circ v_2 \circ \dots \circ v_k$  của toán tử  $\circ$  sẽ được ghi vào L.

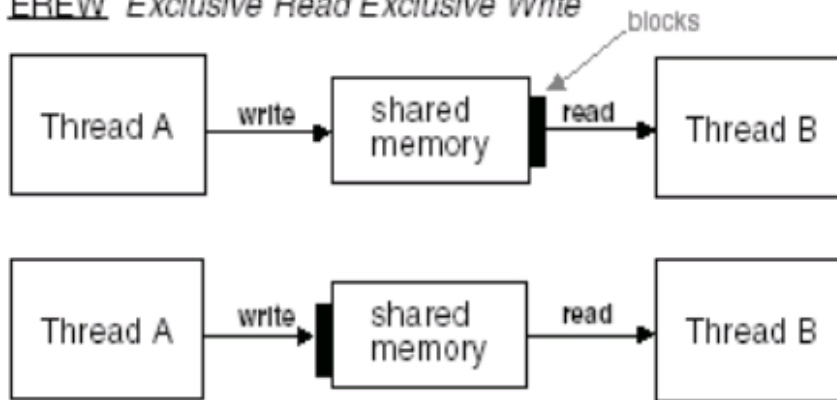


## 6.2. Parallel Random Access Machine

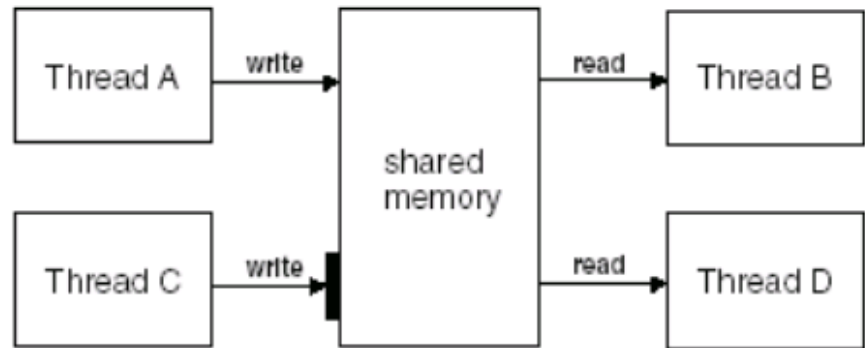
- Định lý: Mọi thuật toán giải quyết một vấn đề tính toán  $\pi$  trên CRCW-PRAM( $p$ ) nhiều nhất  $O(\log p)$  thời gian nhanh hơn thuật toán nhanh nhất để giải quyết vấn đề  $\pi$  trên EREW-PRAM( $p$ )

# PHÂN LOẠI PRAM

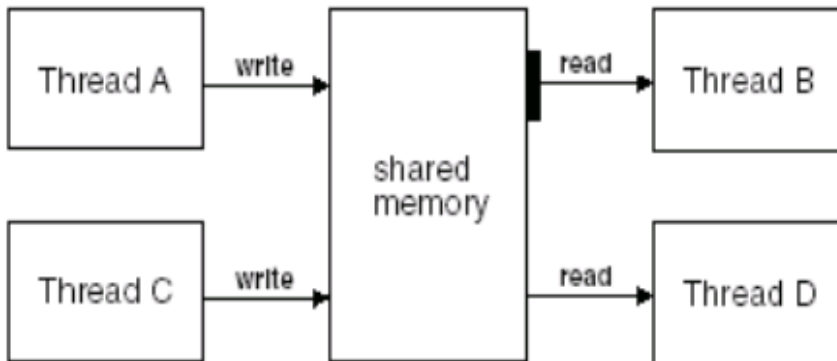
EREW Exclusive Read Exclusive Write



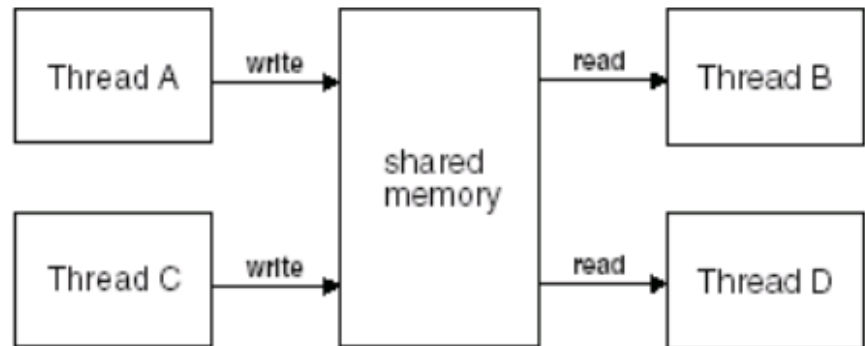
CREW Concurrent Read Exclusive Write



ERCW Exclusive Read Concurrent Write



CRCW Concurrent Read Concurrent Write



## 6.2. Parallel Random Access Machine

- Lý do nghiên cứu Parallel Random Access Machine:
  - Các bộ xử lý có thể thực hiện việc đọc hay ghi lên bất kỳ ô nhớ nào trong bộ nhớ dùng chung cũng chỉ trong 1 đơn vị thời gian.
  - Người thiết kế giải thuật bỏ qua các vấn đề truyền thông, chỉ quan tâm đến cách xử lý song song.
  - Hỗ trợ các kỹ thuật về truyền dữ liệu hay xử lý đồng bộ hóa.
  - Nếu không tồn tại thuật toán giải trên mô hình PRAM thì sẽ không tồn tại thuật toán trên các mô hình thực tế.



Hết bài!!!!