



Kiến trúc máy tính (Computer Architecture)

Chương 3

Tổ chức bộ xử lý



Mục đích, yêu cầu

- **Mục đích:** Giới thiệu tổ chức, nguyên lý hoạt động của các bộ phận bên trong bộ xử lý nhằm thực hiện nhiệm vụ mà kiến trúc phần mềm đã đề ra. Giới thiệu các kỹ thuật nâng cao hiệu quả hoạt động của các bộ xử lý hiện đại.
- **Yêu cầu:** Sinh viên nắm vững cấu trúc của bộ xử lý trung tâm, diễn tiến thi hành một lệnh mã máy và các kỹ thuật nâng cao hiệu quả hoạt động của các bộ xử lý như: kỹ thuật ống dẫn, siêu ống dẫn, siêu vô hướng, máy tính có lệnh thật dài, máy tính véc-tơ, xử lý song song.

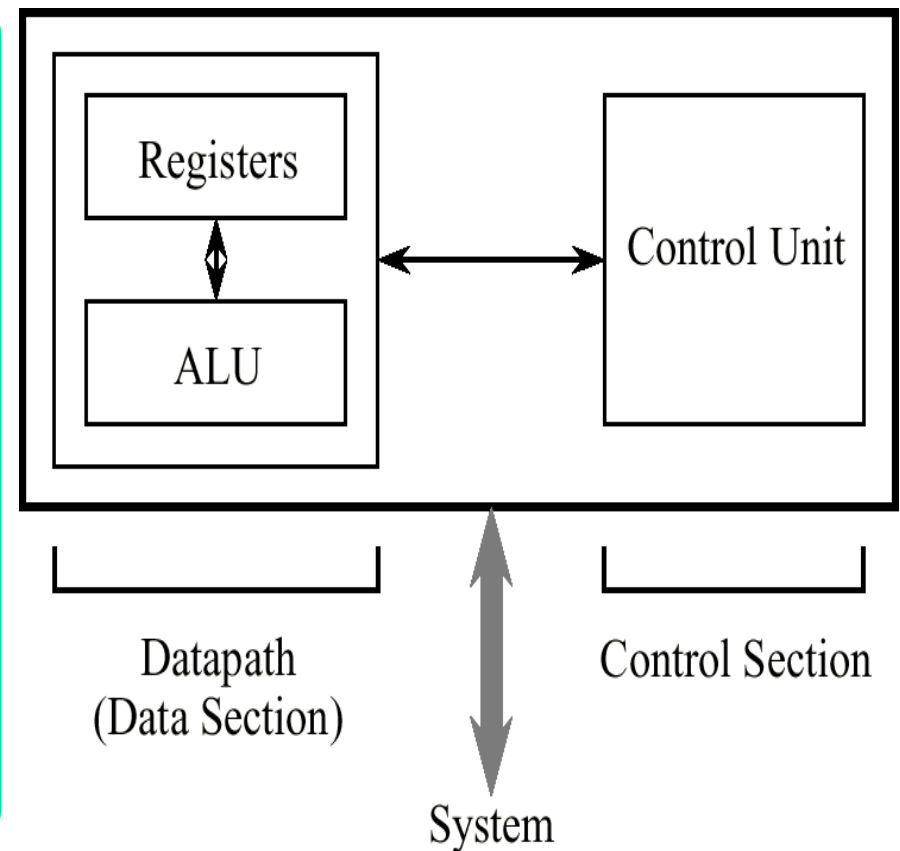


Nội dung

1. Đường đi của dữ liệu
2. Bộ điều khiển
3. Diễn tiến thi hành lệnh mã máy
4. Ngắt quãng
5. Kỹ thuật ống dẫn
6. Khó khăn trong kỹ thuật ống dẫn
7. Siêu ống dẫn
8. Siêu vô hướng
9. Máy tính có lệnh thật dài
10. Kiến trúc IA64
11. Máy tính vector
12. Máy tính song song

1. Đường đi của dữ liệu

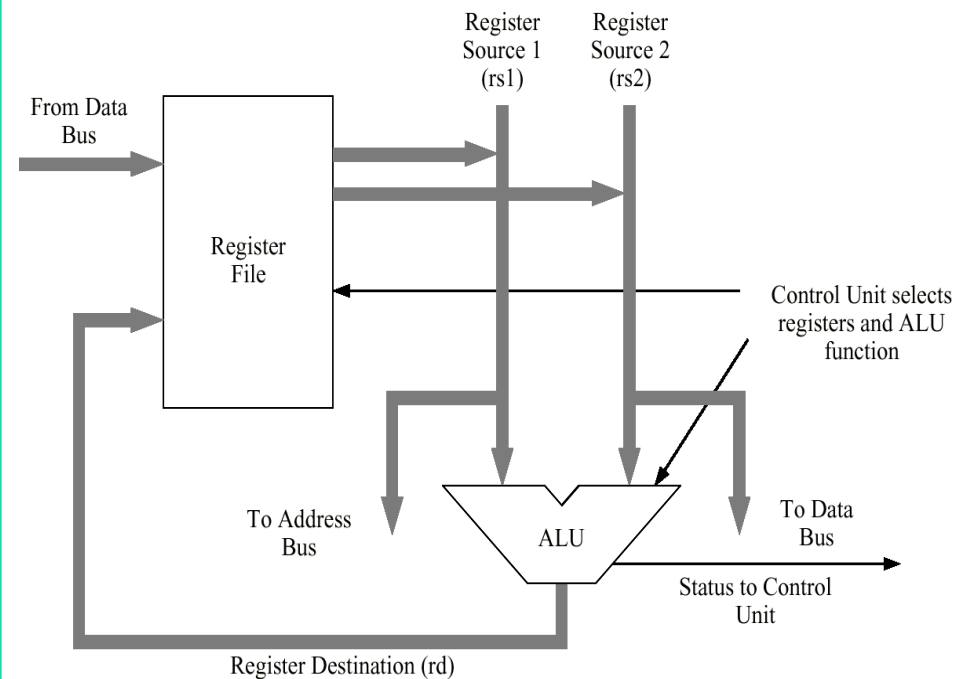
- Bộ xử lý trung tâm gồm:
 - Phần thi hành lệnh (EU: Execution Unit)
Gồm: ALU & Registers
 - Bộ điều khiển (CU: Control Unit).
- Phần thi hành lệnh còn được gọi là đường đi dữ liệu (Datapath).



1. Đường đi của dữ liệu

■ Nhiệm vụ:

- Đọc toán hạng từ các thanh ghi.
- Thực hiện các phép tính trong ALU.
- Lưu kết quả trong các thanh ghi





1. Đường đi của dữ liệu

Các thành phần:

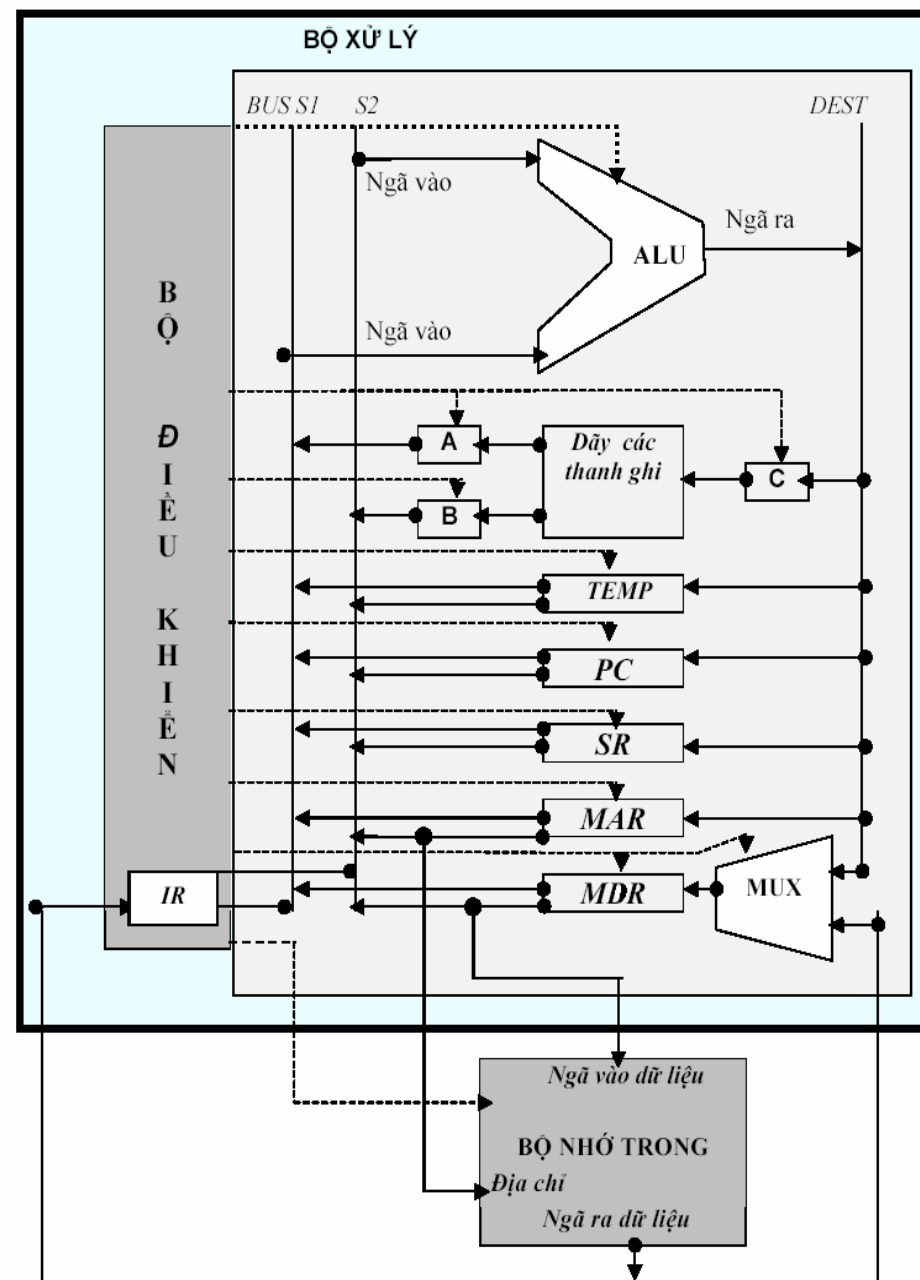
■ Bộ phận số học và luận lý (ALU)

■ Các thanh ghi

- Tổng quát
- Bộ đếm chương trình (PC)
- Trạng thái (SR)
- Đệm (TEMP)
- Địa chỉ bộ nhớ (MAR)
- Dữ liệu bộ nhớ (MDR)

■ Các đường nối kết

- Dữ liệu nguồn 1 (S1)
- Dữ liệu nguồn 2 (S2)
- Dữ liệu đích (Dest)



Hình III.1: Tổ chức của một xử lý điển hình
(Các đường không liên tục là các đường điều khiển)



2. Bộ điều khiển

■ Nhiệm vụ

- Tạo các tín hiệu điều khiển di chuyển số liệu.
- Điều khiển thực hiện các tác vụ của các bộ phận chức năng (ALU, R/W bộ nhớ).
- Tạo các tín hiệu điều khiển thực hiện các lệnh một cách tuần tự.

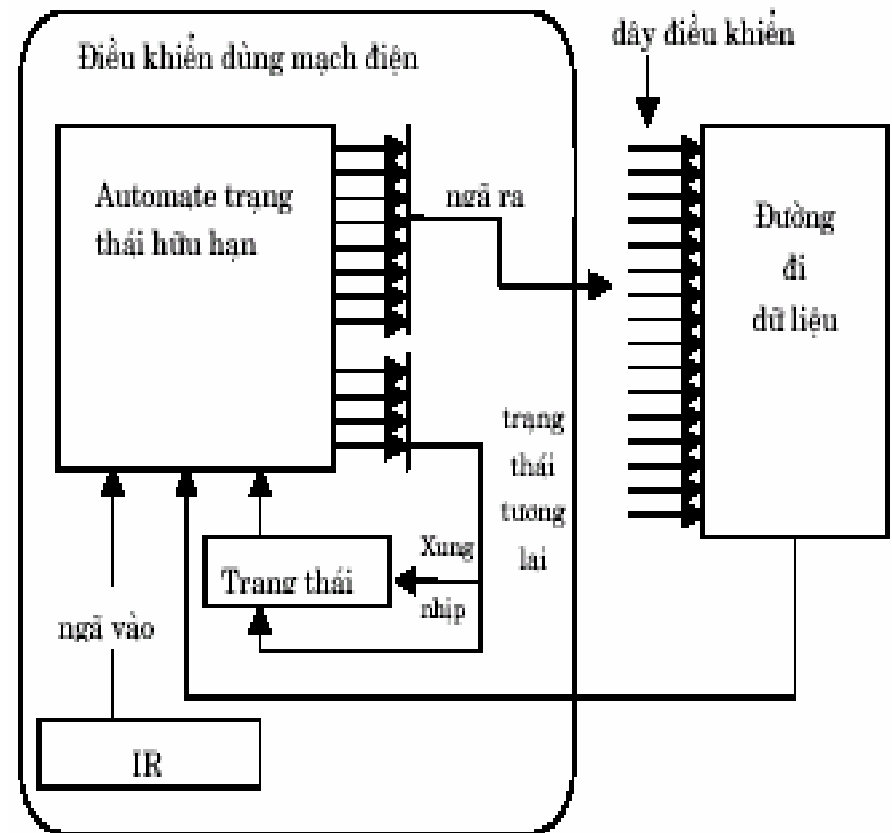
■ Phân loại:

- Bộ điều khiển dùng mạch điện tử.
- Bộ điều khiển dùng vi chương trình.

2. Bộ điều khiển

■ Bộ điều khiển mạch điện tử:

- Các đường điều khiển là các ngã ra của một hoặc nhiều automat trạng thái hữu hạn.
- Các ngã vào của automat gồm có thanh ghi lệnh, trạng thái hiện tại và những thông tin từ bộ đường đi số liệu.
- Ứng với các ngã vào, automat sẽ cho các đường điều khiển phần đường đi số liệu (ngã ra) và trạng thái tương lai.



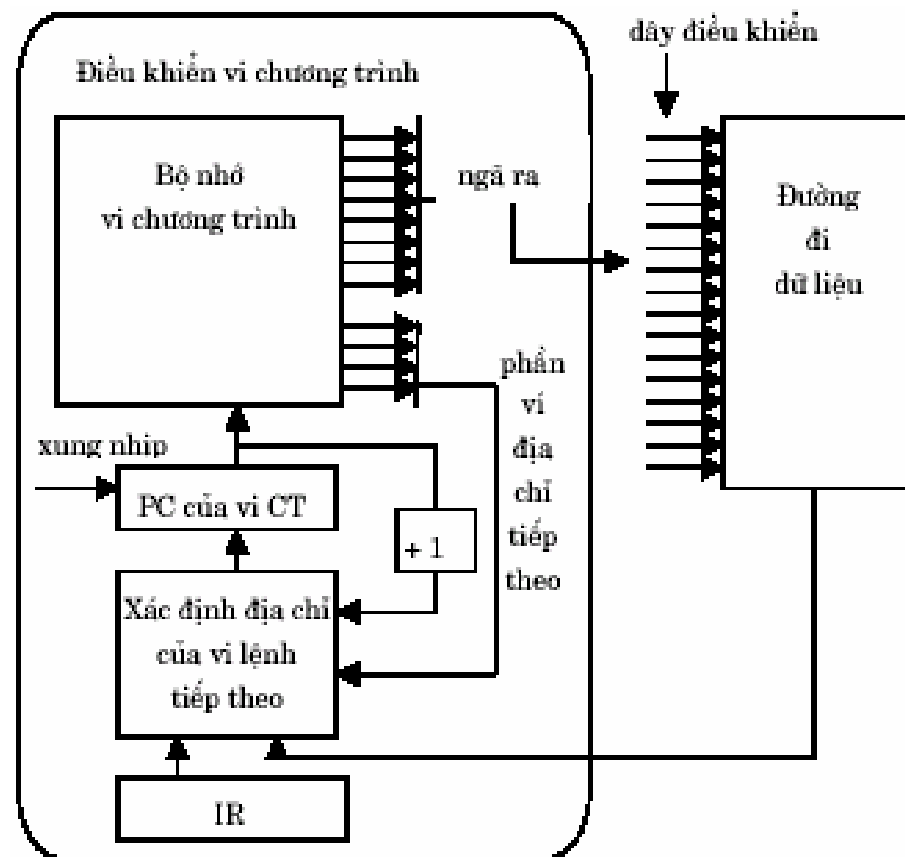
- Nhận xét: Kỹ thuật điều khiển dùng mạch điện tử đơn giản và hữu hiệu khi các lệnh có chiều dài cố định, có dạng thức đơn giản. Nó được dùng nhiều trong các bộ xử lý RISC.



2. Bộ điều khiển

Bộ điều khiển vi chương trình:

- Các đường điều khiển bộ đường đi dữ liệu ứng với các ngõ ra của một vi lệnh trong bộ nhớ vi chương trình.
- Điều khiển các tác vụ của lệnh mã máy bằng cách thực hiện một chuỗi các vi lệnh.
- Một vi máy tính bên trong bộ điều khiển thực hiện từng lệnh của vi chương trình.
- Các tác vụ của lệnh mã máy tùy thuộc vào trạng thái của phần đường đi số liệu.
- Nhận xét: Bộ điều khiển vi chương trình được dùng trong các CPU CISC, người ta cài đặt một lệnh mã máy bằng một vi chương trình.





2. Giải mã lệnh và đọc thanh ghi nguồn:

A	←	Rs1
B	←	Rs2
PC	←	PC + 4

3. Thi hành lệnh:

- **Thâm nhập bộ nhớ:** MAR \leftarrow Địa chỉ (Rs2)
- Ghi: MDR \leftarrow Rs1
- **Lệnh của ALU:**
Ngõ ra ALU \leftarrow Kết quả phép tính
- **Lệnh nhảy:**
Ngõ ra ALU \leftarrow Địa chỉ (ALU tính).

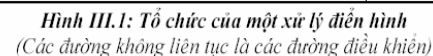
4. Thâm nhập bộ nhớ / nhảy lần cuối:

- **Thâm nhập bộ nhớ:**
 - **Đọc:** $MDR \leftarrow M[MAR]$
 - **Ghi:** $M[MAR] \leftarrow MDR$
- **Nhảy:** If (điều kiện), $PC \leftarrow$ ngã ra ALU

5. Lưu trữ kết quả:

- Thâm nhập bộ nhớ: - Đọc: $Rd \leftarrow MDR$
- Lệnh của ALU: $Rd \leftarrow \text{Ngã ra ALU}$

Tổ chức bộ xử





4. Ngắt quãng

- **Khái niệm:** Ngắt quãng là một sự kiện xảy ra một cách ngẫu nhiên trong máy tính và làm ngưng tính tuần tự của chương trình (nghĩa là tạo ra một lệnh nhảy).
- **Nhận xét:** Bộ điều khiển của CPU là bộ phận khó thực hiện nhất và ngắt quãng là phần khó thực hiện nhất của bộ điều khiển.
- **Ngắt quãng được dùng cho các công việc sau đây:**
 - Ngoại vi đòi hỏi nhập hoặc xuất số liệu.
 - Người lập trình muốn dùng dịch vụ của hệ điều hành.
 - Cho một chương trình chạy từng lệnh.
 - Làm điểm dừng của một chương trình.
 - Báo tràn số liệu trong tính toán số học.
 - Trang bộ nhớ thực sự không có trong bộ nhớ.
 - Báo vi phạm vùng cấm của bộ nhớ.
 - Báo dùng một lệnh không có trong tập lệnh.
 - Báo phần cứng máy tính bị hư.
 - Báo điện bị cắt.



4. Ngắt quãng

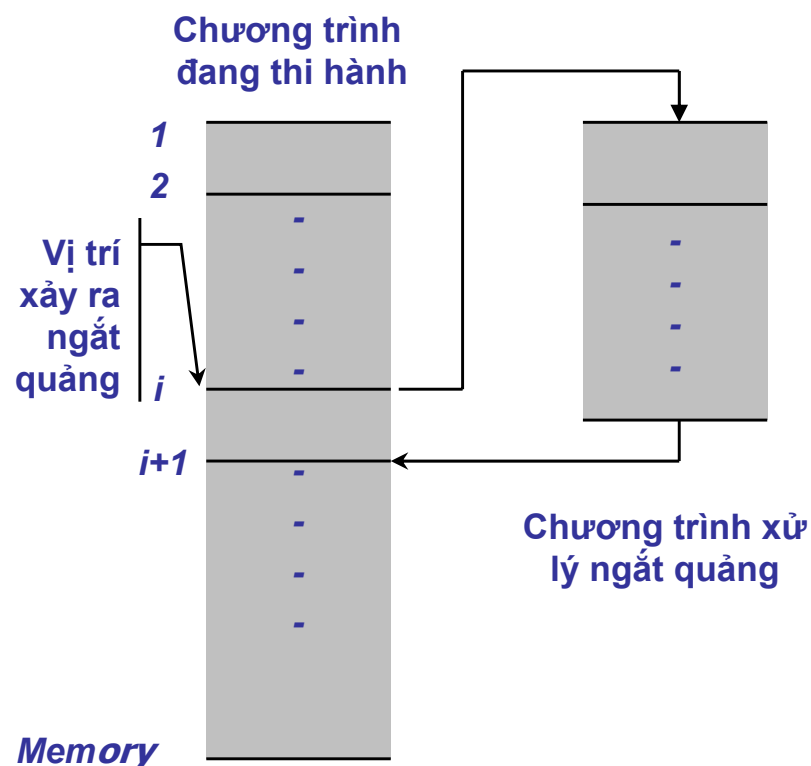
Các giai đoạn thực hiện:

■ Các vấn đề cần lưu ý:

- CPU lưu giữ trạng thái trước khi phục vụ ngắt quãng.
- Sau khi thực hiện phục vụ ngắt, CPU khôi phục trạng thái để tiếp tục công việc.
- CPU chấp nhận ngắt sau khi thực hiện xong lệnh đang làm.

■ Các giai đoạn thực hiện:

1. Thực hiện xong lệnh đang làm.
2. Lưu trữ trạng thái hiện tại.
3. Nhảy đến chương trình phục vụ ngắt quãng
4. CPU khôi phục trạng thái và tiếp tục thực hiện chương trình trước khi bị ngắt.



5. Kỹ thuật ống dẫn

- **Khái niệm:** Kỹ thuật ống dẫn là một kỹ thuật làm cho các giai đoạn khác nhau của nhiều lệnh được thi hành cùng một lúc.
- **Các giai đoạn:**
 - Lấy lệnh (FO: Fetch Operation),
 - Giải mã (DE: Decode),
 - Thi hành (EX : Execute),
 - Thăm nhập bộ nhớ (MEM : Memory access),
 - Lưu trữ kết quả (RS : Result Storing).

Lệnh	Chu kỳ xung nhịp								
	1	2	3	4	5	6	7	8	9
Lệnh i	FO	DE	EX	MEM	RS				
Lệnh i+1		FO	DE	EX	MEM	RS			
Lệnh i+2			FO	DE	EX	MEM	RS		
Lệnh i+3				FO	DE	EX	MEM	RS	
Lệnh i+4					FO	DE	EX	MEM	RS

Hình III.4: Các giai đoạn khác nhau của nhiều lệnh được thi hành cùng một lúc.



5. Kỹ thuật ống dẫn

- **Các ràng buộc:**

- Phải có một mạch điện để thi hành một giai đoạn của lệnh.
- Phải có nhiều thanh ghi.
- Giải mã các lệnh phải đơn giản
- Cần phải có các bộ làm tính ALU hữu hiệu.
- Cần phải có nhiều thanh ghi lệnh.
- Có nhiều thanh ghi bộ đếm chương trình



6. Khó khăn trong kỹ thuật ống dẫn

- Khi thi hành lệnh trong một máy tính dùng kỹ thuật ống dẫn, có nhiều trường hợp làm cho kỹ thuật ống dẫn không thực hiện được hoặc thực hiện không hiệu quả:
 - Thiếu các mạch chức năng,
 - Một lệnh dùng kết quả của lệnh trước,
 - Một lệnh nhảy.

1. Khó khăn do cấu trúc

- Nguyên nhân: Thiếu bộ phận chức năng (ALU, PC, IR, ...) thoả mãn các ràng buộc của kỹ thuật ống dẫn.
- Khắc phục: Trang bị thêm các bộ phận chức năng cần thiết & hiệu quả

6. Khó khăn trong kỹ thuật ống dẫn

2. Khó khăn do số liệu:

- Nguyên nhân: Lệnh kế sau có sử dụng kết quả của lệnh kế trước.

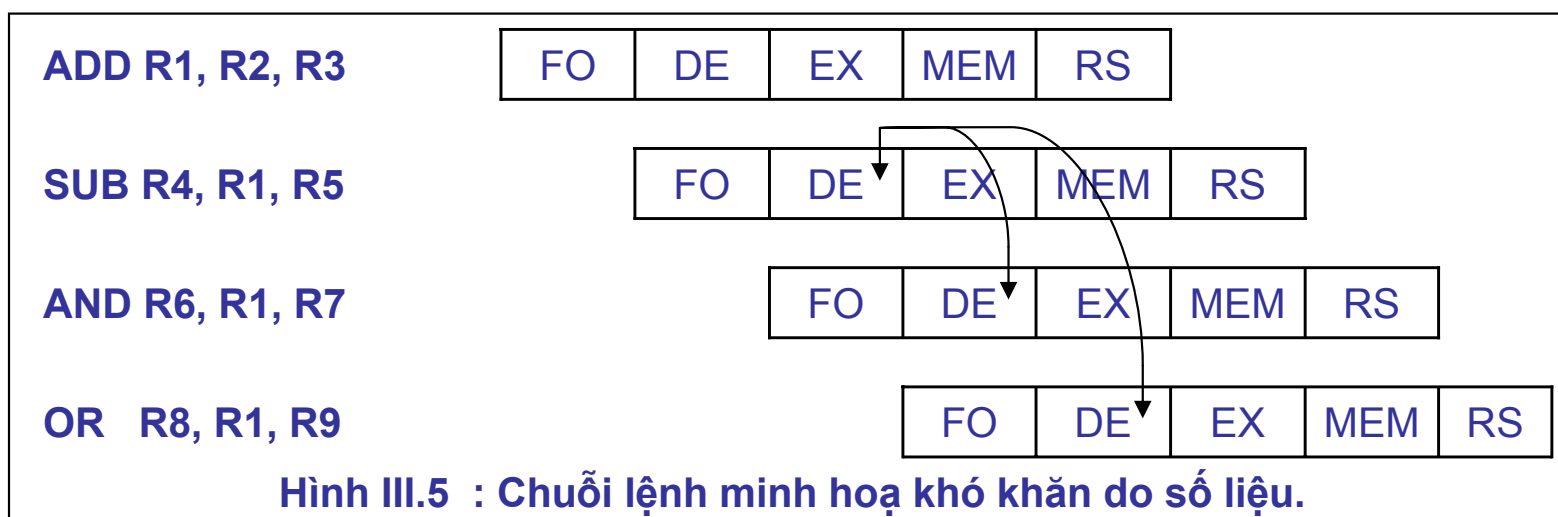
- Ví dụ:

1- ADD R1, R2, R3

2- SUB R4, R1, R5

3- AND R6, R1, R7

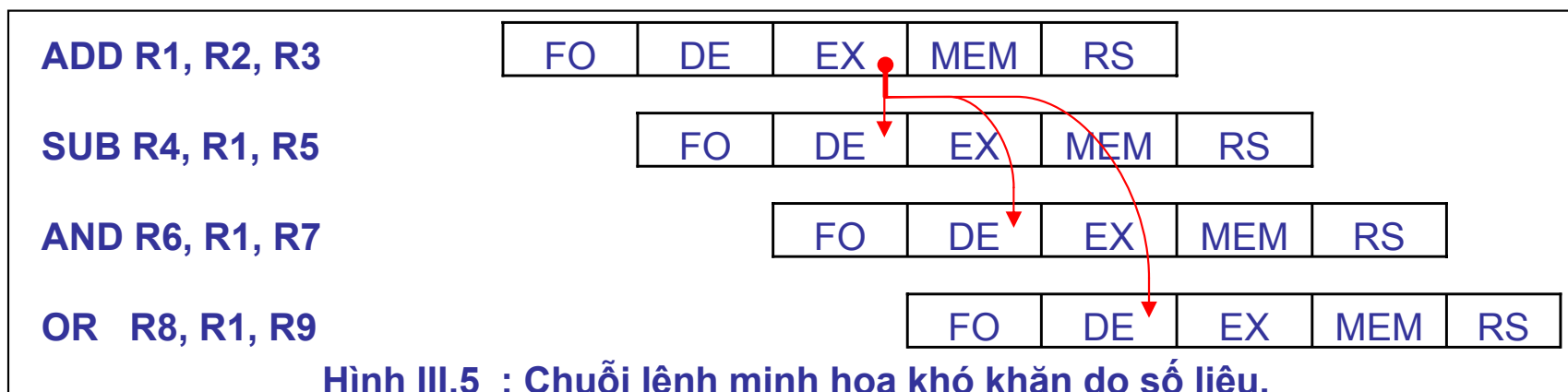
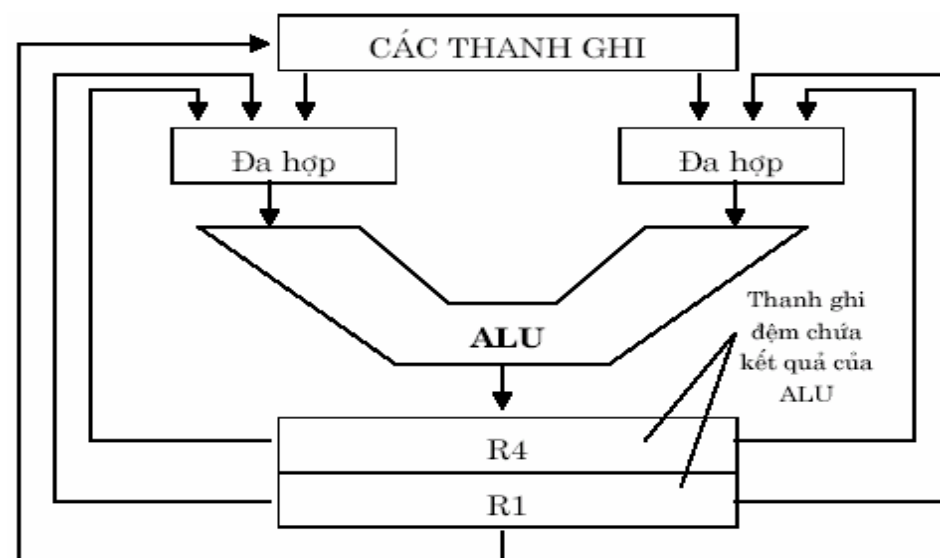
4- OR R8, R1, R9



6. Khó khăn trong kỹ thuật ống dẫn

2. Khó khăn do số liệu:

- **Khắc phục:** Trang bị thêm bộ phận phần cứng đặc biệt để đưa kết quả ở ngõ ra ALU, trực tiếp vào một trong các thanh ghi ngõ vào.



Hình III.5 : Chuỗi lệnh minh họa khó khăn do số liệu.

6. Khó khăn trong kỹ thuật ống dẫn

3. Khó khăn do điều khiển

- **Nguyên nhân:** Các lệnh nhảy làm thay đổi tính tuần tự khi thi hành các lệnh

Đối với lệnh nhảy không điều kiện:

- Tác vụ nhảy không thể biết trước giai đoạn giải mã. Nếu lệnh nhảy bắt đầu ở chu kỳ c thì lệnh nhảy tới bắt đầu ở chu kỳ $c+2$.
- Trong lệnh nhảy tương đối, ta phải cộng độ dời chứa trong thanh ghi IR vào thanh ghi PC. Việc tính địa chỉ được thực hiện vào giai đoạn DE với điều kiện phải có một mạch cộng riêng biệt.

Lệnh	Chu kỳ xung nhịp								
	1	2	3	4	5	6	7	8	9
Lệnh i	FO	*DE	EX	MEM	RS				
Lệnh $i+1$		FO	DE	EX	MEM	RS			
Lệnh $i+2$			*FO	DE	EX	MEM	RS		
Lệnh $i+3$				FO	DE	EX	MEM	RS	
Lệnh $i+4$					FO	DE	EX	MEM	RS



6. Khó khăn trong kỹ thuật ống dẫn

3. Khó khăn do điều khiển

- **Nguyên nhân:** Các lệnh nhảy làm thay đổi tính tuần tự khi thi hành các lệnh

Đối với các lệnh nhảy có điều kiện:

- Phải tính toán điều kiện. Trong kiến trúc RISC, kết quả so sánh đặt vào thanh ghi trạng thái / tổng quát. Trong cả 2 trường hợp, đọc điều kiện tương đương với đọc thanh ghi.
- Trường hợp khó hơn có thể xảy ra trong lệnh nhảy có điều kiện:
 - Điều kiện có được khi so sánh 2 thanh ghi và chỉ thực hiện lệnh nhảy khi kết quả so sánh là đúng.
 - Tính toán trên các đại lượng logic không thể thực hiện trong phân nửa chu kỳ và như thế phải kéo dài thời gian thực hiện lệnh nhảy.

⇒ **Người ta thường tránh các trường hợp này để không làm giảm mức hữu hiệu của máy tính.**



6. Khó khăn trong kỹ thuật ống dẫn

- **Khắc phục:**

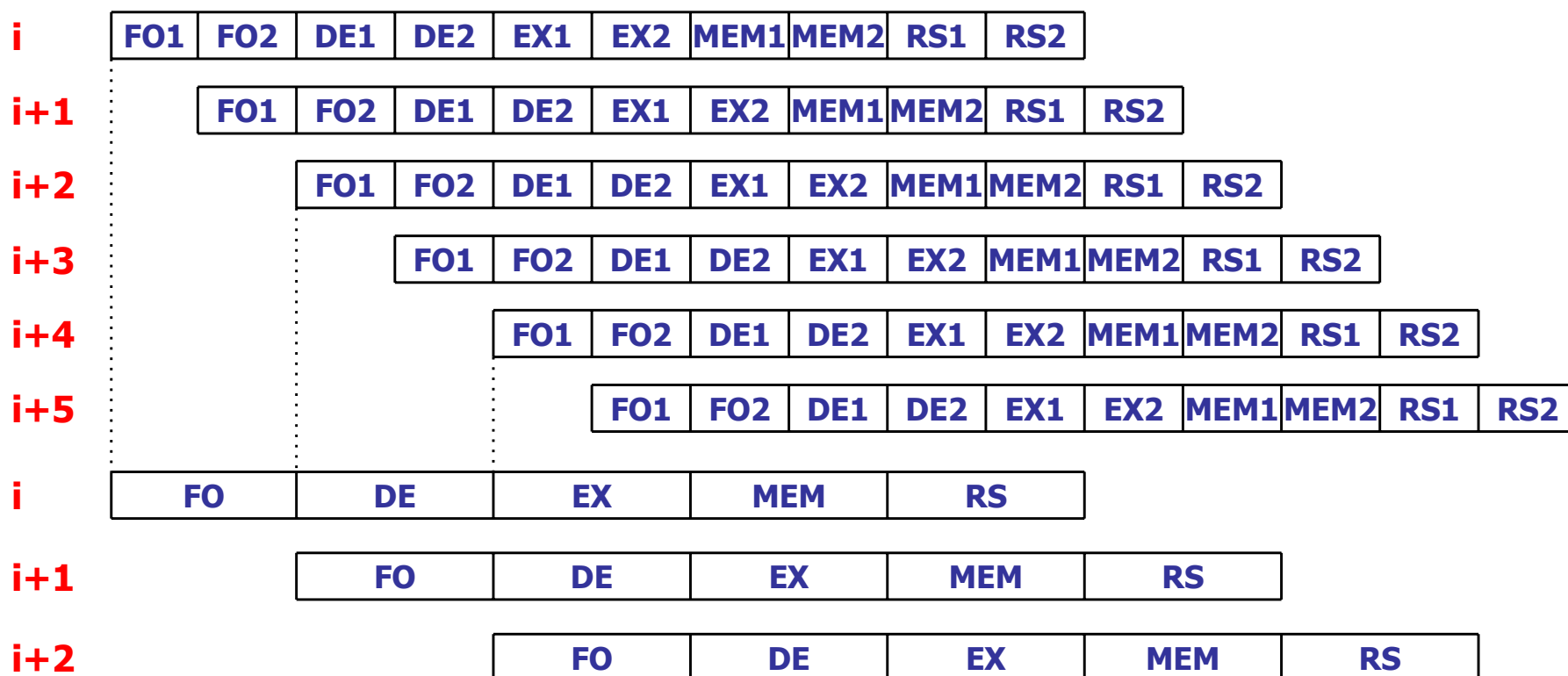
- **Cách thứ nhất:** Đóng băng kỹ thuật ống dẫn trong một chu kỳ, nghĩa là ngưng thi hành lệnh thứ $i+1$ đang làm nếu lệnh thứ i là lệnh nhảy. Ta mất trắng một chu kỳ cho mỗi lệnh nhảy.
- **Cách thứ hai:** Thi hành lệnh sau lệnh nhảy (lưu ý rằng hiệu quả của một lệnh nhảy bị chậm mất một lệnh). Vậy lệnh theo sau lệnh nhảy được thực hiện trước khi lệnh mà chương trình phải nhảy tới được thực hiện. Chương trình dịch hay người lập trình có nhiệm vụ xen vào một lệnh hữu ích sau lệnh nhảy.

Lưu ý: Trong trường hợp nhảy có điều kiện, việc nhảy có thể được thực hiện hay không. Lệnh hữu ích đặt sau lệnh nhảy không làm sai chương trình dù điều kiện nhảy đúng hay sai.

Ví dụ: Bộ xử lý SPARC có những lệnh nhảy với hủy bỏ. Các lệnh này cho phép thi hành lệnh sau lệnh nhảy nếu điều kiện nhảy đúng và hủy bỏ thực hiện lệnh đó nếu điều kiện nhảy sai.

7. Siêu ống dẫn

- Khái niệm:** Người ta có kỹ thuật siêu ống dẫn bậc n bằng cách chia các giai đoạn của kỹ thuật ống dẫn đơn giản (mỗi giai đoạn được thực hiện trong thời gian T_c), thành n giai đoạn con thực hiện trong thời gian T_c/n .





7. Siêu ống dẫn

■ Lợi điểm:

- Độ hữu hiệu của kỹ thuật siêu ống dẫn tương đương với việc thi hành n lệnh trong mỗi chu kỳ T_c .
- Ta thấy trong một chu kỳ T_c , máy dùng kỹ thuật siêu ống dẫn làm 2 lệnh thay vì làm 1 lệnh trong máy dùng kỹ thuật ống dẫn bình thường.
- Trong máy tính siêu ống dẫn, tốc độ thực hiện lệnh tương đương với việc thực hiện một lệnh trong khoảng thời gian T_c/n .

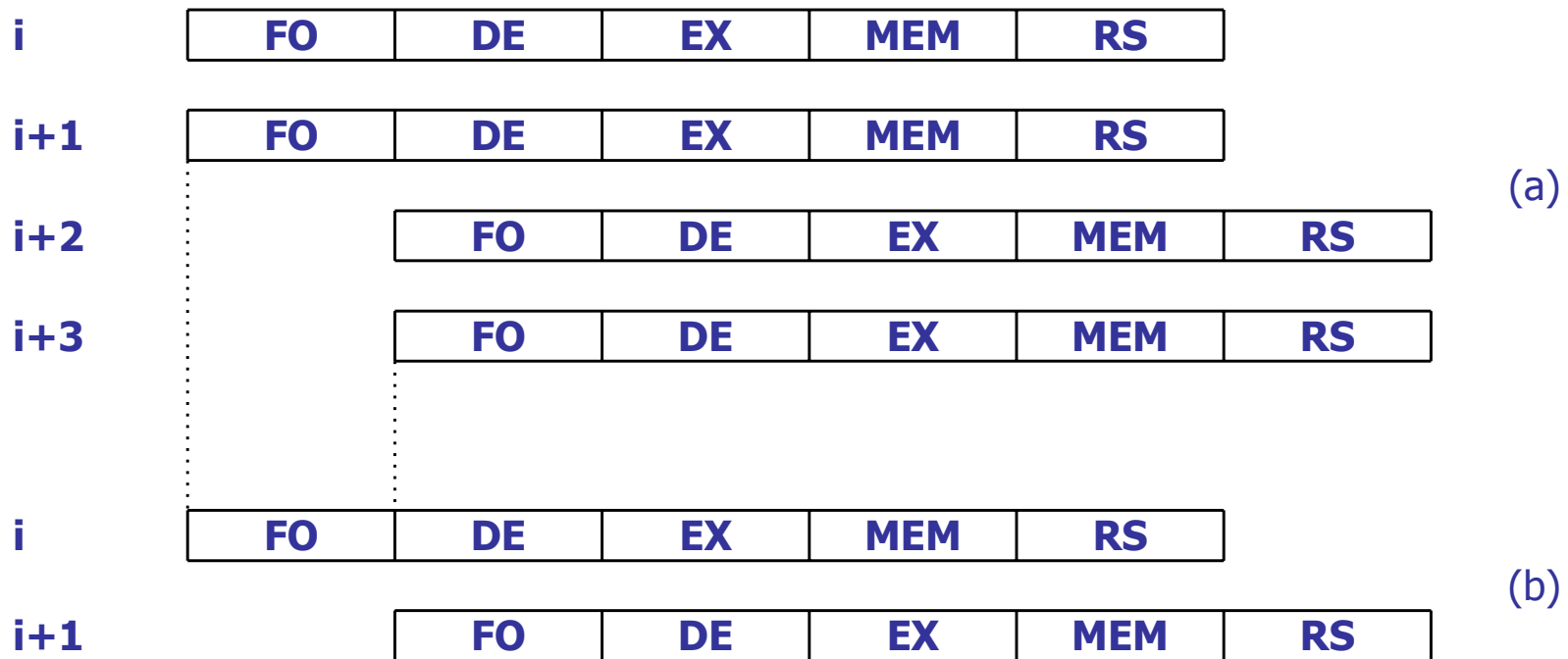
■ Bất lợi:

- Bất lợi của siêu ống dẫn là thời gian thực hiện một giai đoạn con ngắn T_c/n và việc trì hoãn trong thi hành lệnh nhảy lớn.
- Nếu lệnh thứ i là một lệnh nhảy tương đối thì lệnh này được giải mã trong giai đoạn DE, địa chỉ nhảy đến được tính vào giai đoạn EX, lệnh phải nhảy tới là lệnh thứ $i+4$, vậy có trì trệ 3 lệnh thay vì 1 lệnh trong kỹ thuật ống dẫn bình thường.

8. Siêu vô hướng

- **Khái niệm:** Máy tính siêu vô hướng bậc n có thể thực hiện đồng thời n lệnh trong một chu kỳ xung nhịp T_c .

Ví dụ: Sự vận hành của một máy tính siêu vô hướng bậc 2 so với một máy tính dùng kỹ thuật ống dẫn.



Hình III.8 : Siêu vô hướng (a) so với kỹ thuật ống dẫn (b).



8. Siêu vô hướng

■ Nhận xét:

- Trong một máy tính siêu vô hướng phần cứng phải quản lý việc đọc và thi hành đồng thời nhiều lệnh. Vậy nó phải có khả năng quản lý các quan hệ giữa số liệu với nhau.
- Cần chọn các lệnh có khả năng được thi hành cùng một lúc.

■ Ví dụ:

- Những bộ xử lý đầu tiên đưa ra thị trường dùng kỹ thuật này là Intel i860 và IBM RS/6000. Các bộ xử lý này có khả năng thực hiện song song nhiều tác vụ trên số nguyên và trên số lẻ.
- Năm 1992, người ta thấy xuất hiện các bộ xử lý có nhiều bộ thực hiện tác vụ độc lập với nhau (nhiều ALU, bộ tính toán số lẻ, nạp dữ liệu, lưu dữ liệu, nhảy), có thể thực hiện song song nhiều lệnh. Số lệnh có thể được thi hành song song càng nhiều thì phần cứng thực hiện việc này càng phức tạp.



8. Siêu vô hướng

- **Kỹ thuật siêu luồng (HT: Hyper Threading):**
 - Người ta tìm cách tận dụng các tài nguyên dư thừa của máy tính sử dụng kỹ thuật siêu vô hướng.
 - Giả lập một bộ xử lý vật lý dùng kỹ thuật siêu vô hướng thành 2 bộ xử lý logic để tăng hiệu quả thực thi các chương trình.
 - Tăng hiệu quả thực hiện các chương trình:
 - Đa chương trình mức độ tăng gần 30%
 - Đơn chương trình mức độ tăng đến 18%



Minh họa kỹ thuật
Hyper Threading



Tổ chức bộ xử lý



Hiệu quả kỹ thuật
Hyper Threading



9. Máy tính có lệnh thật dài

■ Nhận xét:

- Máy tính siêu vô hướng có thể thực hiện 2 hoặc 3 lệnh trong mỗi chu kỳ xung nhịp.
- Do kỹ thuật ống dẫn đòi hỏi các lệnh phải phụ thuộc vào nhau nên rất khó thực hiện nhiều lệnh trong một chu kỳ.
- Như vậy, thay vì thực hiện nhiều lệnh trong một chu kỳ, người ta tìm cách đưa vào nhiều lệnh (tác vụ) trong một từ lệnh dài.

■ Đặc điểm:

- Một lệnh thật dài (VLIW: Very Long Instruction Word) có thể chứa 2 tác vụ tính toán số nguyên, 2 tác vụ tính toán số lẻ, 2 tác vụ thâm nhập bộ nhớ và một lệnh nhảy.
- Một lệnh như vậy được chia thành trường, mỗi trường có từ 16 đến 24 bit, chiều dài của lệnh VLIW từ 112 đến 168 bit.
- Có nhiều kỹ thuật tạo ra một lệnh VLIW trong đó tất cả các trường đều được dùng.
- Giá thành và độ phức tạp của một máy tính có lệnh thật dài tăng lên rất nhiều nếu tăng số trường trong một lệnh VLIW.



10. Kiến trúc IA64

- Kiến trúc IA-64 là một kiến trúc mới được giới thiệu trong những năm gần đây. Kiến trúc này là sản phẩm của sự kết hợp nghiên cứu giữa hai công ty máy tính hàng đầu thế giới là Intel, HP (Hewlett Packard) và một số trường đại học.
- Kiến trúc IA-64 giới thiệu một sự khởi đầu mới quan trọng của kỹ thuật siêu vô hướng - kỹ thuật xử lý lệnh song song (EPIC: Explicitly Parallel Instruction Computing).
- Sản phẩm đầu tiên thuộc kiến trúc này là bộ xử lý ***Itanium***.



10. Kiến trúc IA64

Đặc trưng của kiến trúc IA-64:

- Cơ chế xử lý song song các lệnh mã máy (EPIC) thay vì các bộ xử lý song song như hệ thống đa bộ xử lý.
- Các lệnh dài hay rất dài (LIW hay VLIW).
- Các lệnh rẽ nhánh xác định (thay vì đoán các lệnh rẽ nhánh như các kiến trúc trước).
- Nạp trước các lệnh (theo sự suy đoán).

Số thanh ghi các bộ xử lý kiến trúc IA-64 là 256 thanh ghi.

Trong đó:

- 128 thanh ghi tổng quát 64 bit cho các tính toán số nguyên, luận lý
- 128 thanh ghi 82 bit cho các phép tính dấu chấm động và dữ liệu đồ họa
- 64 thanh ghi 1 bit thuộc tính để chỉ ra các thuộc tính lệnh đang thi hành.



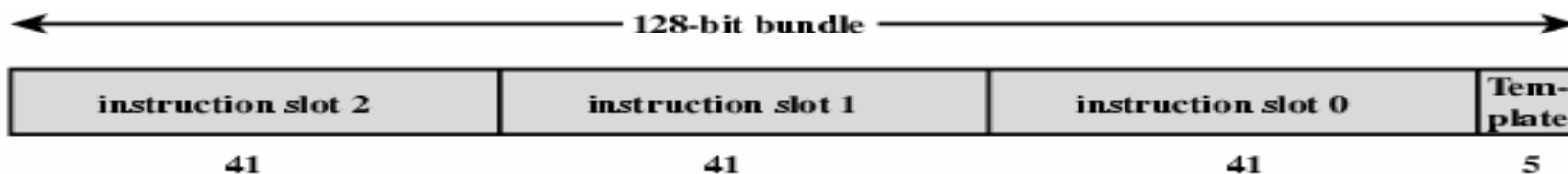
10. Kiến trúc IA64

- Nhiều bộ thi hành lệnh: một máy tính có thể có tám hay nhiều hơn các bộ thi hành lệnh song song. Các bộ thi hành lệnh này được chia thành bốn kiểu:
 - Kiểu I (I-Unit): dùng xử lý các lệnh tính toán số nguyên, dịch, luận lý, so sánh, đa phương tiện.
 - Kiểu M (M-Unit): Nạp và lưu trữ giữa thanh ghi và bộ nhớ thêm vào một vài tác vụ ALU.
 - Kiểu B (B-Unit): Thực hiện các lệnh rẽ nhánh.
 - Kiểu F (F-Unit): Các lệnh tính toán số dấu chấm động

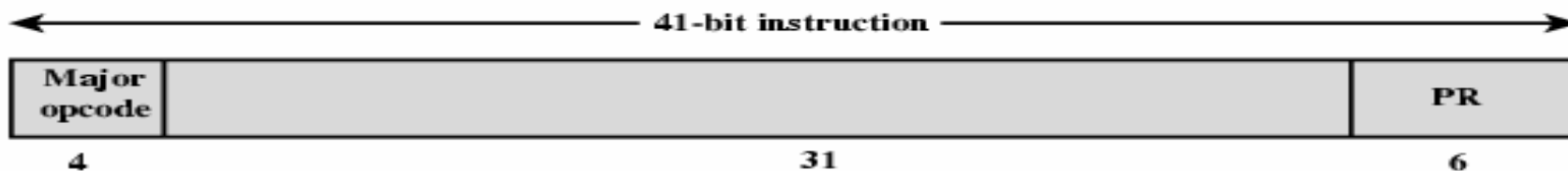


10. Kiến trúc IA64

Các định dạng lệnh:



(a) IA-64 bundle



(b) General IA-64 instruction format



(c) Typical IA-64 instruction format

PR = Predicate register

GR = General or floating-point register

Hình III.12: Định dạng lệnh trong kiến trúc IA-64



11. Máy tính vector

■ **Khái niệm:** Một máy tính vector bao gồm:

- Một bộ tính toán vô hướng bình thường dùng kỹ thuật ống dẫn
- Một bộ làm tính vector thực hiện các phép tính vector.
- Đa số các máy tính vector cho phép làm các phép tính trên:
 - Vector số nguyên,
 - Vector số lẻ,
 - Vector số logic (số Bool).

■ **Các kiểu kiến trúc máy tính vector:**

- **Kiểu vector ô nhớ - ô nhớ:** Trong máy tính loại vector bộ nhớ - bộ nhớ, các phép tính vector được thực hiện trong bộ nhớ
Ví dụ: Các máy tính CDC thuộc loại này
- **Kiểu thanh ghi vector:** Có một bộ nhiều thanh ghi vector và những tác vụ vector được thực hiện trên các thanh ghi này.
Ví dụ: Các siêu máy tính: CRAY-1, CRAY-2, X-MP, Y-MP, NEC SX/2, Hitachi S820. Máy CRAY-2 (1995) có 8 thanh ghi vector, mỗi thanh ghi chứa 64 vector, mỗi vector có chiều dài 64 bit.



12. Máy tính song song

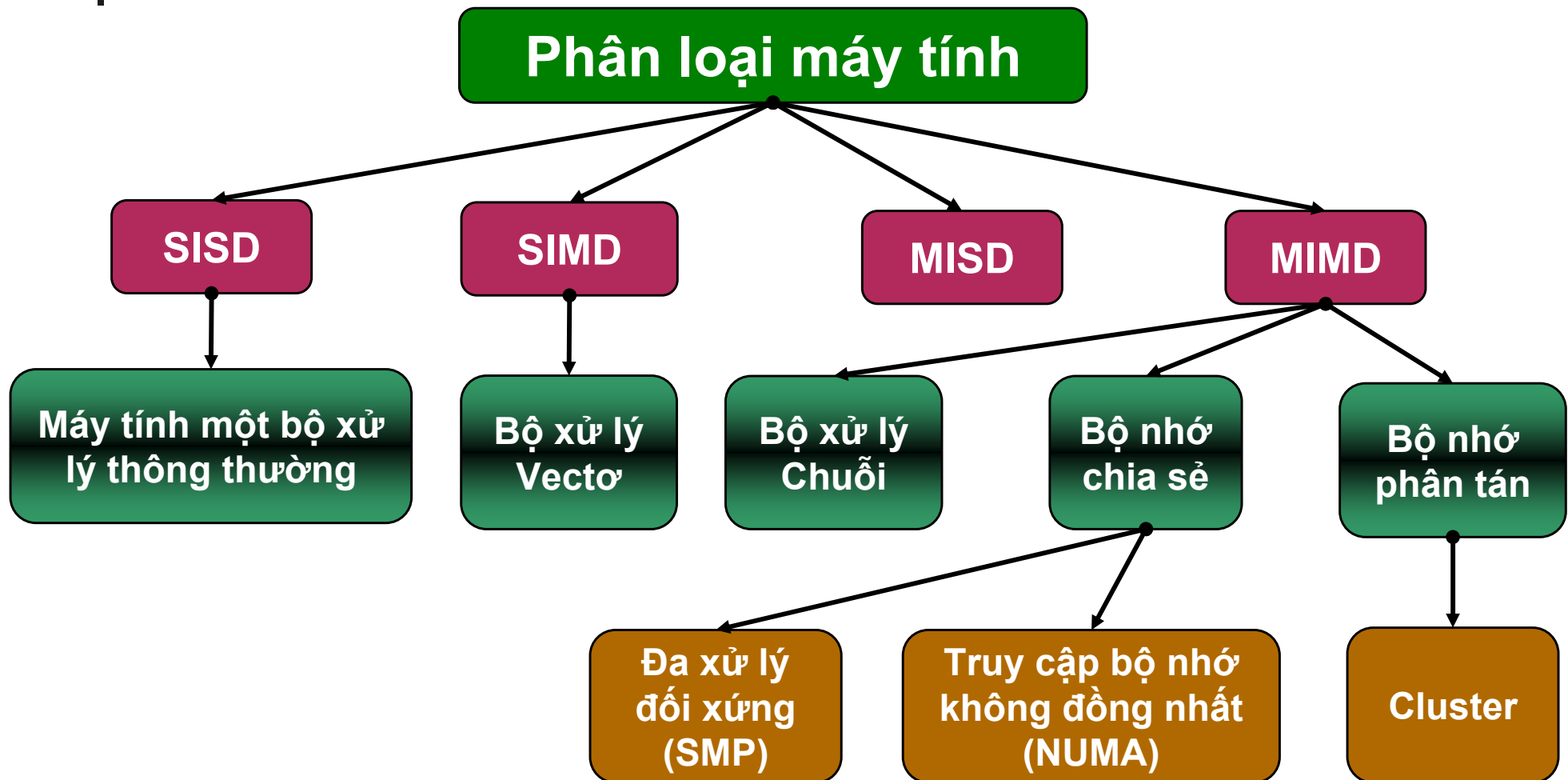
■ Nhận xét:

- Trong các máy tính siêu ống dẫn, siêu vô hướng, máy tính vectơ, máy tính VLIW, người ta đã dùng tính thực hiện song song các lệnh ở các mức độ khác nhau để làm tăng hiệu quả.
- Giới hạn về khả năng tính toán của loại máy trên khiến người ta nghĩ tới giải pháp song song theo đó người ta tăng cường hiệu quả của máy tính bằng cách tăng số lượng bộ xử lý.

■ Phân loại: Năm 1966, Flynn đã xếp các máy tính vào 4 loại :

- **SISD** (Single Instructions stream, Single Data stream): Máy tính một dòng lệnh, một dòng dữ liệu
- **SIMD** (Single Instructions stream, Multiple Data stream): Máy tính một dòng lệnh, nhiều dòng dữ liệu
- **MISD** (Multiple Instructions stream, Single Data stream): Máy tính nhiều dòng lệnh, một dòng dữ liệu
- **MIMD** (Multiple Instruction stream, Multiple Data stream): Máy tính nhiều dòng lệnh, nhiều dòng dữ liệu

12. Máy tính song song



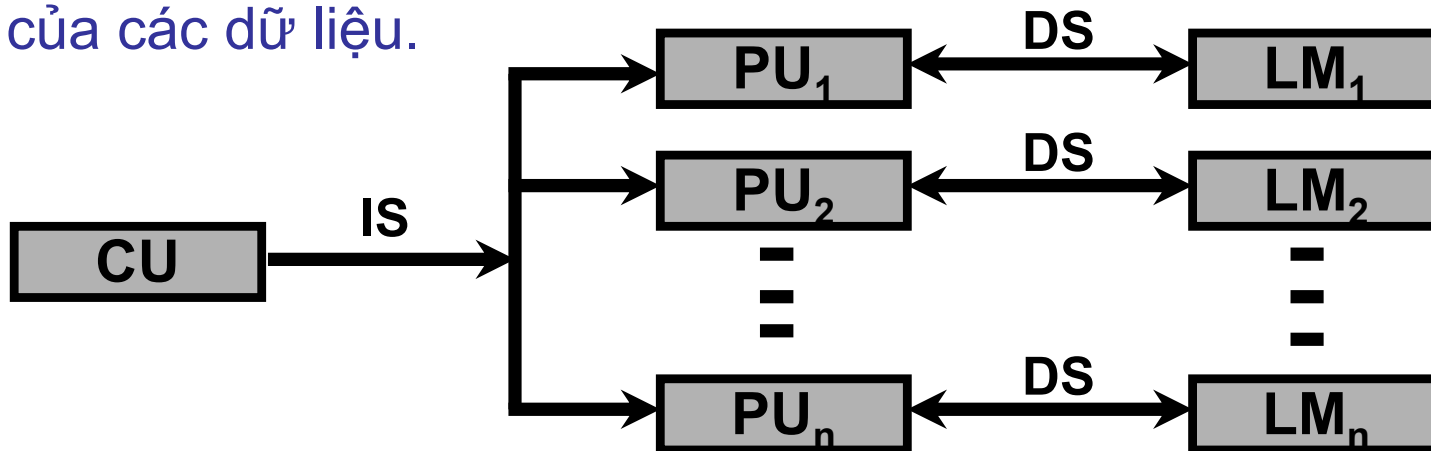
12. Máy tính song song

SISD: Máy tính một dòng lệnh, một dòng dữ liệu là máy tính với 1 bộ xử lý mà ta đã nghiên cứu.



SIMD: Có một số lớn các bộ xử lý giống nhau, cùng thực hiện một lệnh giống nhau để xử lý nhiều dòng dữ liệu khác nhau.

- Mỗi bộ xử lý có bộ nhớ dữ liệu riêng, nhưng chỉ có một bộ nhớ lệnh và một bộ điều khiển, bộ này đọc và thi hành các lệnh.
- Tính song song dùng trong các máy SIMD là tính song song của các dữ liệu.



Tổ chức bộ xử lý



12. Máy tính song song

MISD: Máy tính nhiều dòng lệnh, một dòng dữ liệu chưa chế tạo.

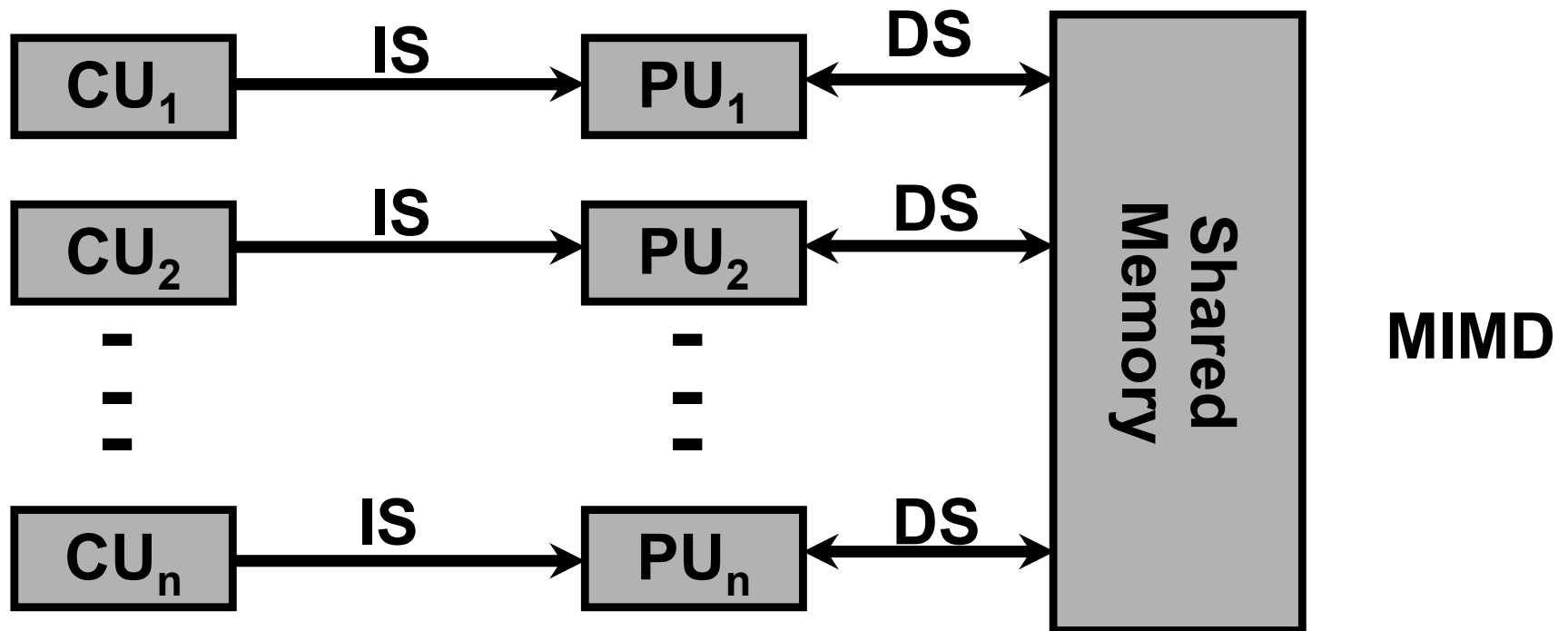
MIMD: Máy tính nhiều dòng lệnh, nhiều dòng dữ liệu là một kiến trúc đương nhiên cho các máy nhiều bộ xử lý dùng trong các ứng dụng thông thường.

Người ta thường phân loại các máy MISD theo tổ chức bộ nhớ:

- SMP (Symmetric Multiprocessors): Hệ thống đa xử lý bộ nhớ chia sẻ, thời gian truy cập bộ nhớ giống nhau cho các bộ xử lý.
- NUMA (NonUniform Memory Access): Hệ thống đa xử lý bộ nhớ chia sẻ, thời gian truy cập các vùng nhớ khác nhau thì khác nhau.
- CC-NUMA (Cache-Coherent NUMA): Hệ thống NUMA với sự liên kết và duy trì hệ thống cache trên các bộ xử lý riêng lẻ.
- Cluster: Toàn bộ các máy tính riêng lẻ được kết nối với nhau thực hiện các công việc như một máy tính hợp nhất có chung các tài nguyên. Mỗi máy tính được xem như là một bộ phận của Cluster.

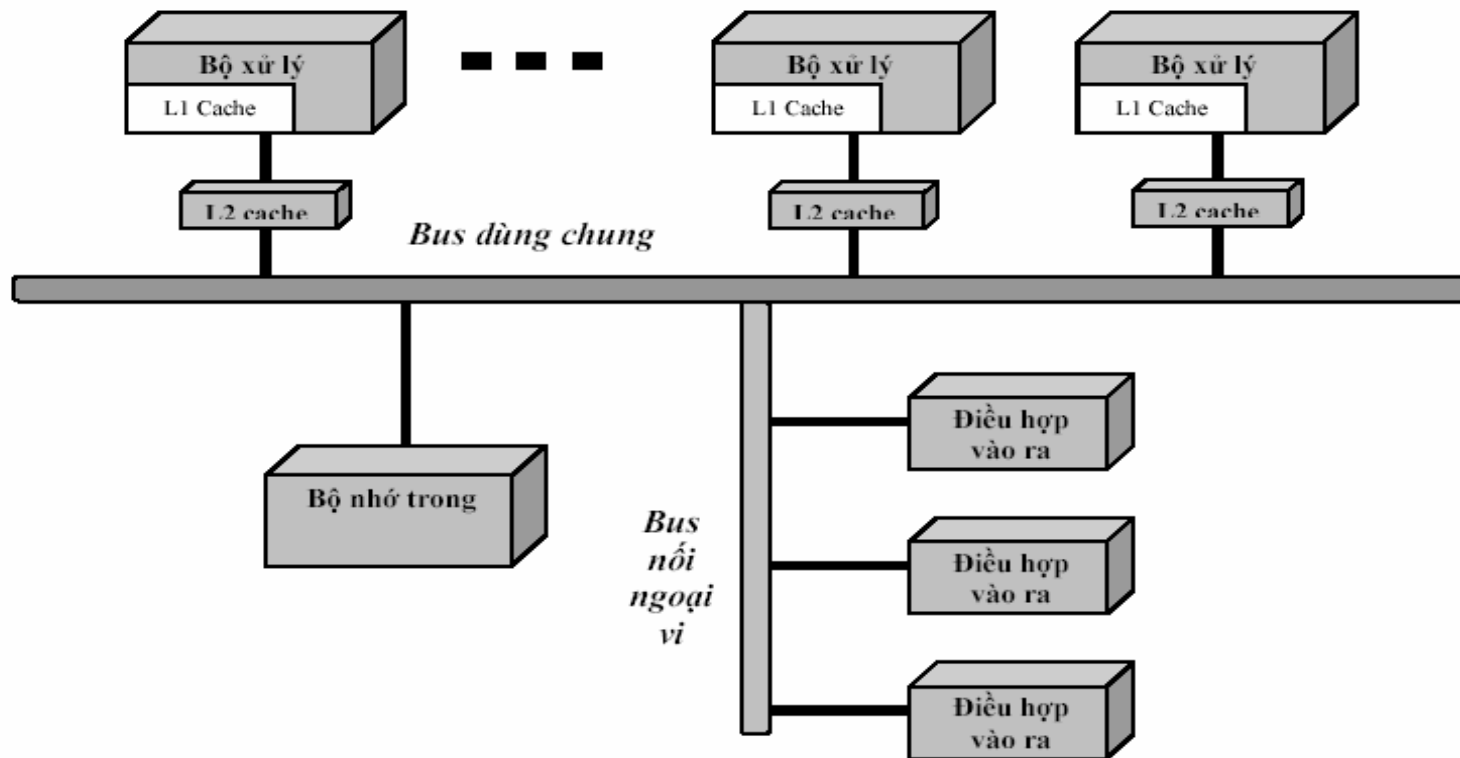
12. Máy tính song song

- SMP (Symmetric Multiprocessors):
 - Bộ nhớ chia sẻ, các trao đổi thông qua bộ nhớ chia sẻ.
 - Dùng chung hệ thống bus bộ nhớ.
 - Thời gian truy cập bộ nhớ giống nhau cho các bộ xử lý



12. Máy tính song song

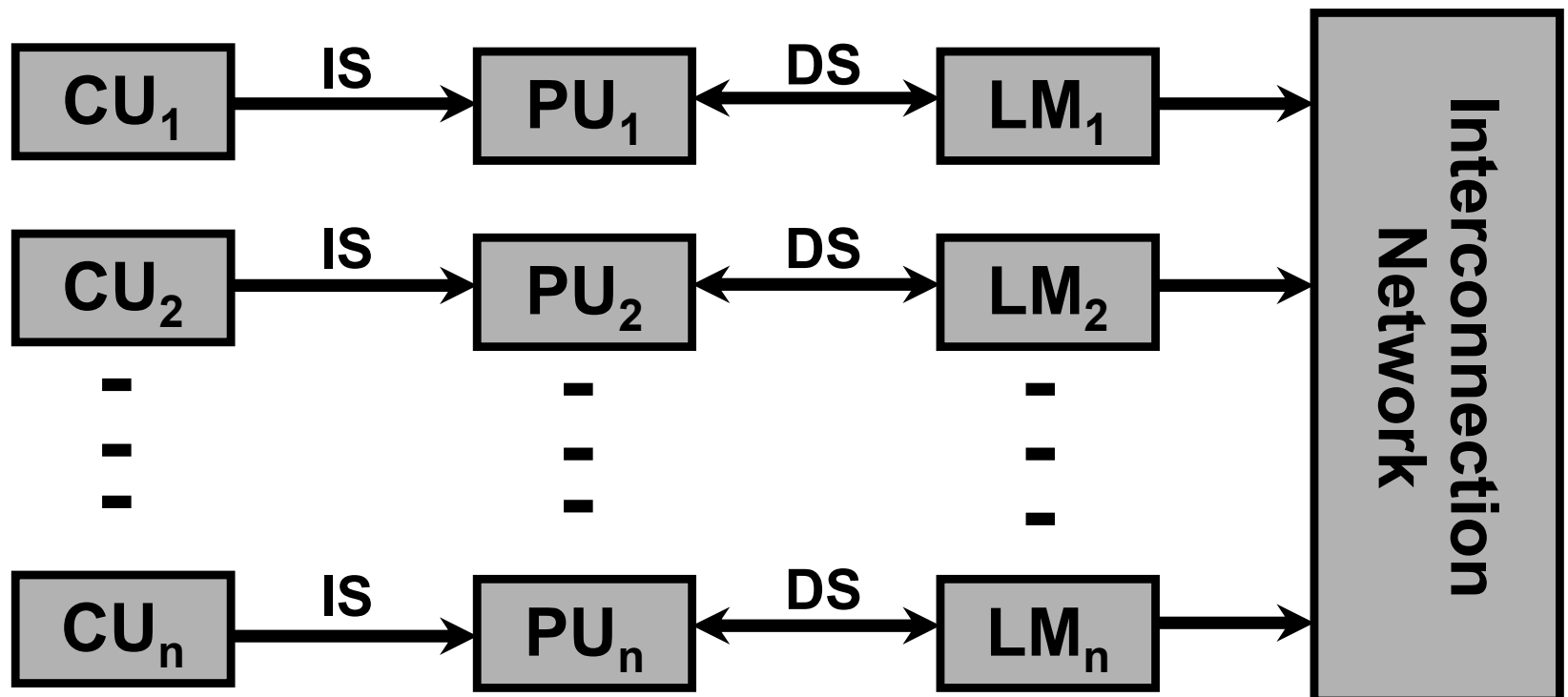
- NUMA (NonUniform Memory Access)
- CC-NUMA (Cache-Coherent NUMA):



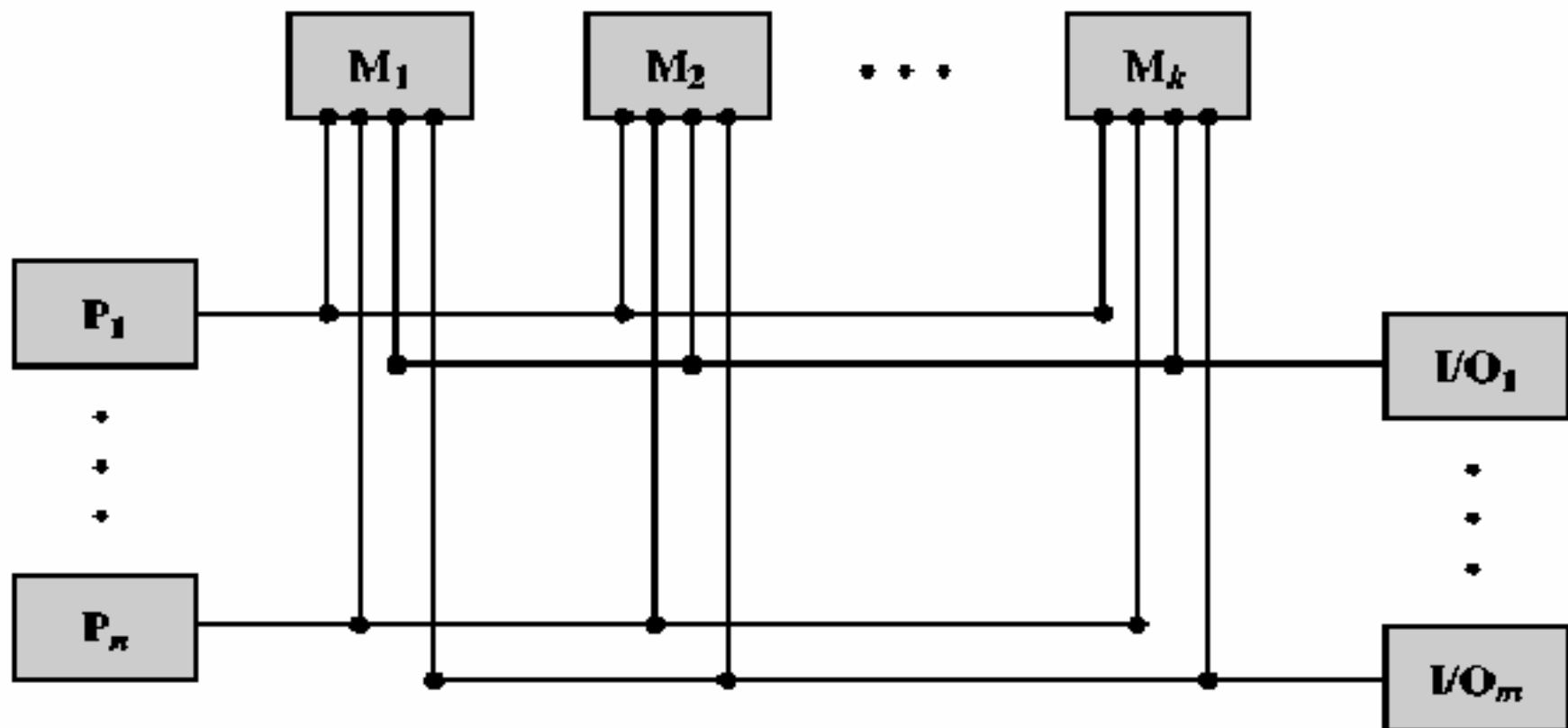
Hình III.9: Máy tính song song với bộ nhớ dùng chung, hệ thống bus dùng chung

12. Máy tính song song

- Máy tính MIMD với bộ nhớ phân tán
 - Mỗi máy gồm có các nút,
 - Mỗi nút chứa một bộ xử lý, bộ nhớ, một vài ngõ vào ra và một giao diện với hệ thống kết nối giữa các nút.

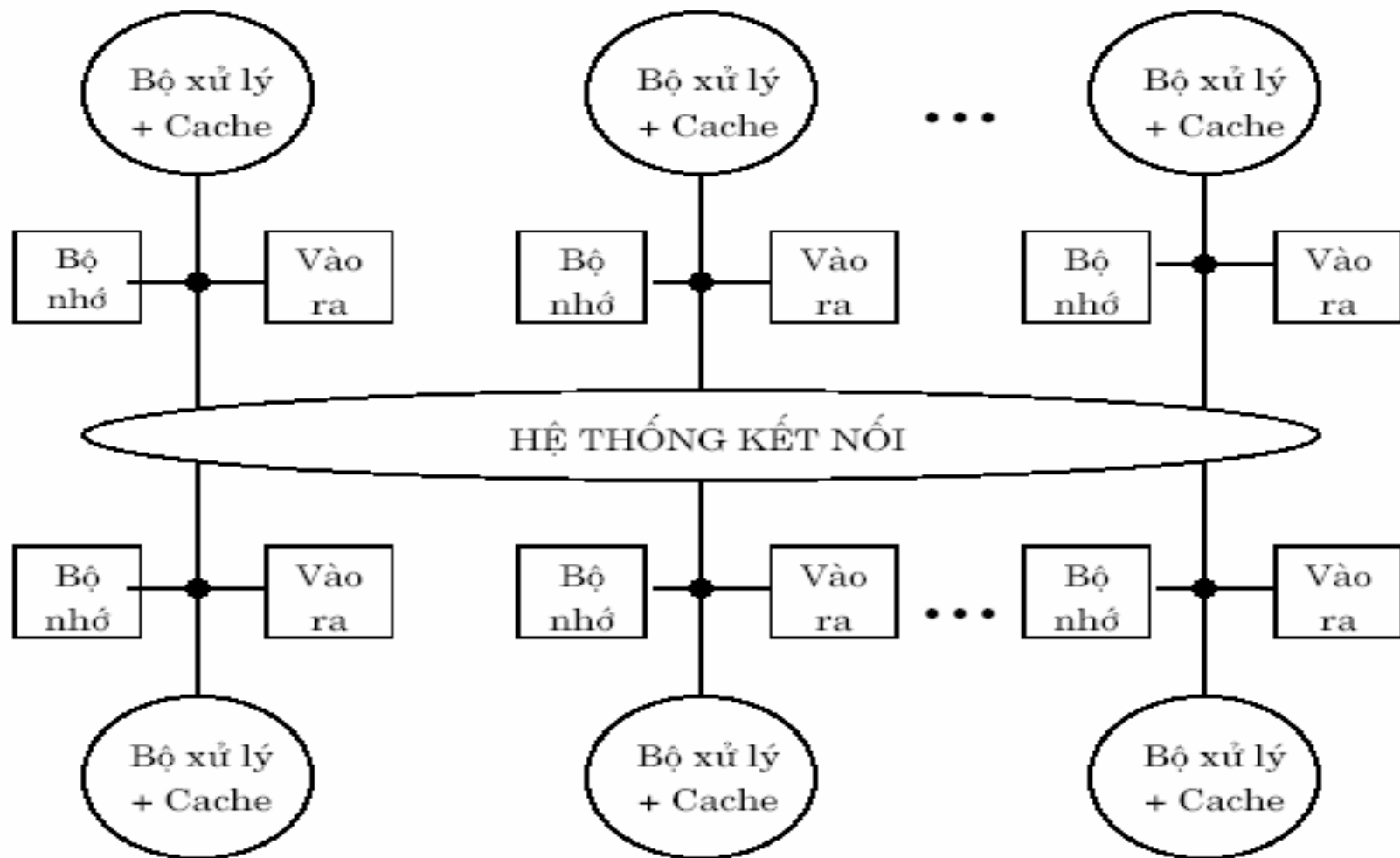


12. Máy tính song song



Hình III.11: Tổ chức kết nối của máy tính song song có bộ nhớ phân tán

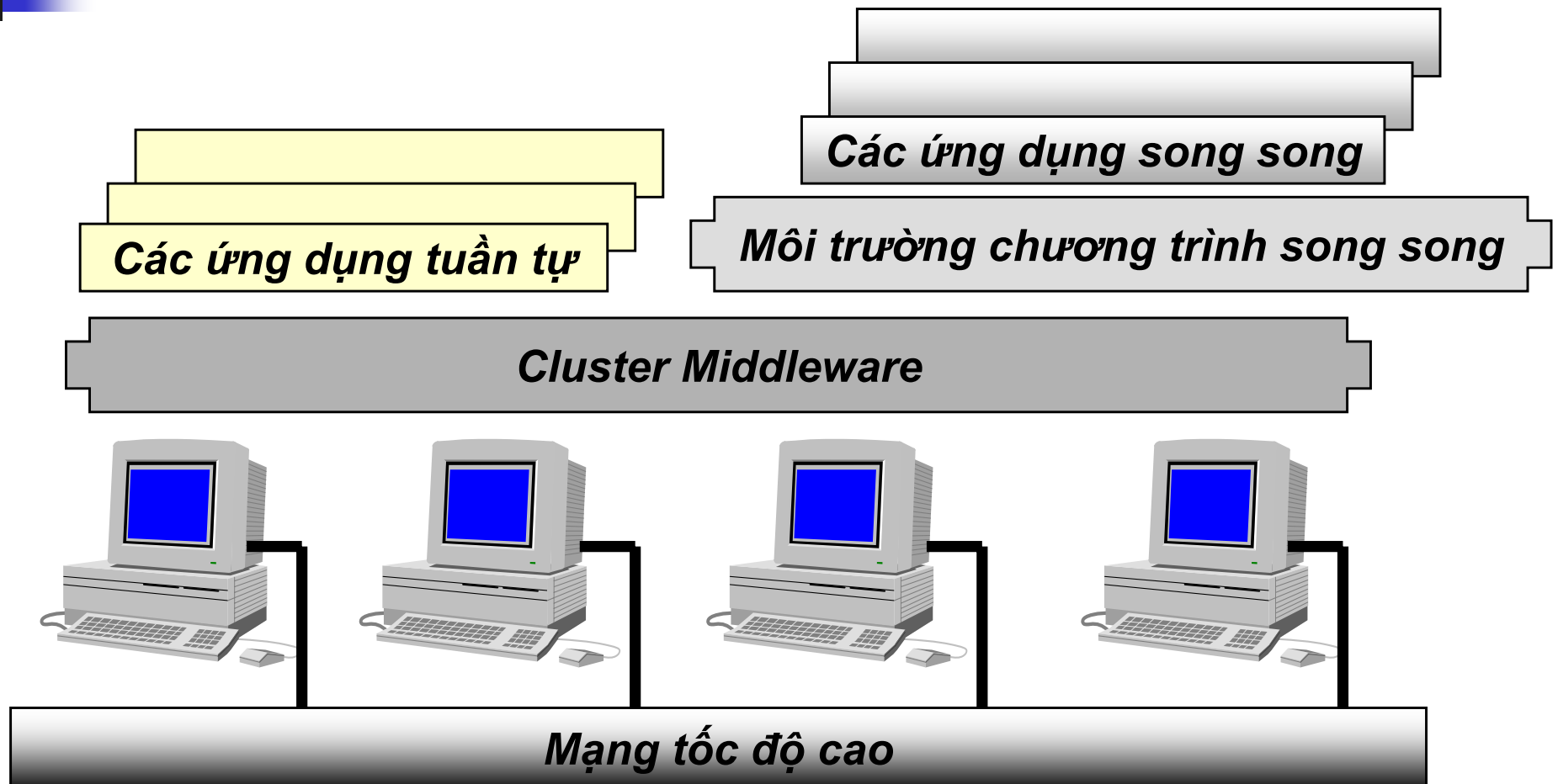
12 Máy tính song song



Cấu trúc nền của một máy đa xử lý với bộ nhớ phân tán

Tổ chức bộ xử lý

12 Máy tính song song



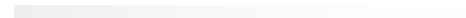
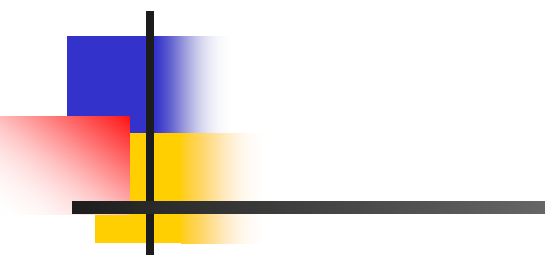
Kiến trúc máy tính Cluster

Tổ chức bộ xử lý



12 Máy tính song song

- **Nhận xét:** Kiến trúc song song đang tiến triển chậm:
 - Do vấn đề phần mềm gặp nhiều khó khăn
 - Do các máy tính đa xử lý còn tiến chậm trong việc cải tiến giao diện với người sử dụng và tăng cường hiệu quả của chúng.
- **Hướng phát triển:** Có nhiều lý do để ta lạc quan vào tương lai của việc xử lý song song và của các máy đa xử lý:
 - Việc dùng xử lý song song bắt đầu được hiểu biết trong vài lĩnh vực, đặc biệt trong lĩnh vực tính toán khoa học và công nghệ. Trong các lĩnh vực này người ta luôn cần đến máy tính có tính năng cao hơn.
 - Hiện tại, người ta đã chấp nhận rằng một trong những cách hiệu quả nhất để chế tạo máy tính có tính năng cao hơn các máy đơn xử lý là chế tạo các máy tính đa xử lý.
 - Máy tính đa xử lý rất hiệu quả khi dùng cho đa chương trình. Đa chương trình được dùng chủ yếu cho các máy tính lớn và cho các máy phục vụ lớn.





Câu hỏi ?

1. Nhiệm vụ của phần đường đi dữ liệu ?
2. Nhiệm vụ của bộ điều khiển ?
3. Bộ điều khiển bằng mạch điện tử ?
4. Bộ điều khiển dùng vi chương trình ?
5. Ngắt quãng là gì ?
6. Lợi điểm của kỹ thuật ngắt quãng ?
7. Tại sao CPU chỉ chấp nhận ngắt sau khi thực hiện xong lệnh đang làm ?
8. Các giai đoạn thực hiện ngắt quãng ?
9. Kỹ thuật ống dẫn là gì ?
10. Trên lý thuyết kỹ thuật ống dẫn làm tăng tốc độ của CPU bao nhiêu lần ?
11. Tại sao thực tế, kỹ thuật ống dẫn tốc độ không tăng như trên lý thuyết ?
12. Trong kỹ thuật ống dẫn, nguyên nhân nào tạo ra khó khăn do điều khiển ?
13. Nguyên nhân nào sau đây gây ra khó khăn do số liệu ?
14. Khắc phục khó khăn do số liệu người ta dùng biện pháp nào ?
15. Biện pháp đóng băng kỹ thuật ống dẫn 1 chu kỳ có nghĩa là gì ?



Câu hỏi ?

16. Biện pháp thực hiện lệnh kế tiếp của lệnh nhảy có lợi ích gì ?
17. Kỹ thuật siêu ống dẫn là gì ?
18. Lợi điểm và bất lợi của kỹ thuật siêu ống dẫn ?
19. Kỹ thuật siêu vô hướng là gì ?
20. Trong kỹ thuật siêu vô hướng có những khó khăn ?
21. Đặc điểm của máy tính siêu vô hướng?
22. Máy tính có lệnh thật dài là gì ?
23. Đặc điểm của máy tính có lệnh thật dài ?
24. Máy tính VLIW giải quyết khó khăn gì trong kỹ thuật siêu vô hướng?
25. Đặc điểm của máy tính vector?
26. Trong kiểu kiến trúc thanh ghi vector, tác vụ nào có toán hạng trong bộ nhớ?
27. Thể hiện mức độ song song trong các máy SISD, SIMD, MISD, MIMD?
28. Tại sao máy tính MISD chưa được chế tạo ?
29. Tại sao MIMD là một kiến trúc đương nhiên cho các máy nhiều bộ xử lý ?
30. Các phương thức tổ chức bộ nhớ của máy tính MIMD ?



Bài tập

Mô tả việc vận chuyển dữ liệu khi thực hiện của các lệnh sau:

1. Load R5, (R20)
2. Store R7, 100(R21)
3. Add R5, R4, R20
4. Sub R10, R12, R24
5. And R5, R2, R15
6. Or R15, R3, R7
7. JMP R7
8. BRA +5
9. BGT R4, +2

Load R5, (R20)

1. Đọc lệnh:

MAR \leftarrow PC

IR \leftarrow M[MAR]

2. Giải mã lệnh và đọc thanh ghi nguồn:

A \leftarrow 0

B \leftarrow R₂₀

PC \leftarrow PC + 4

3. Thi hành lệnh:

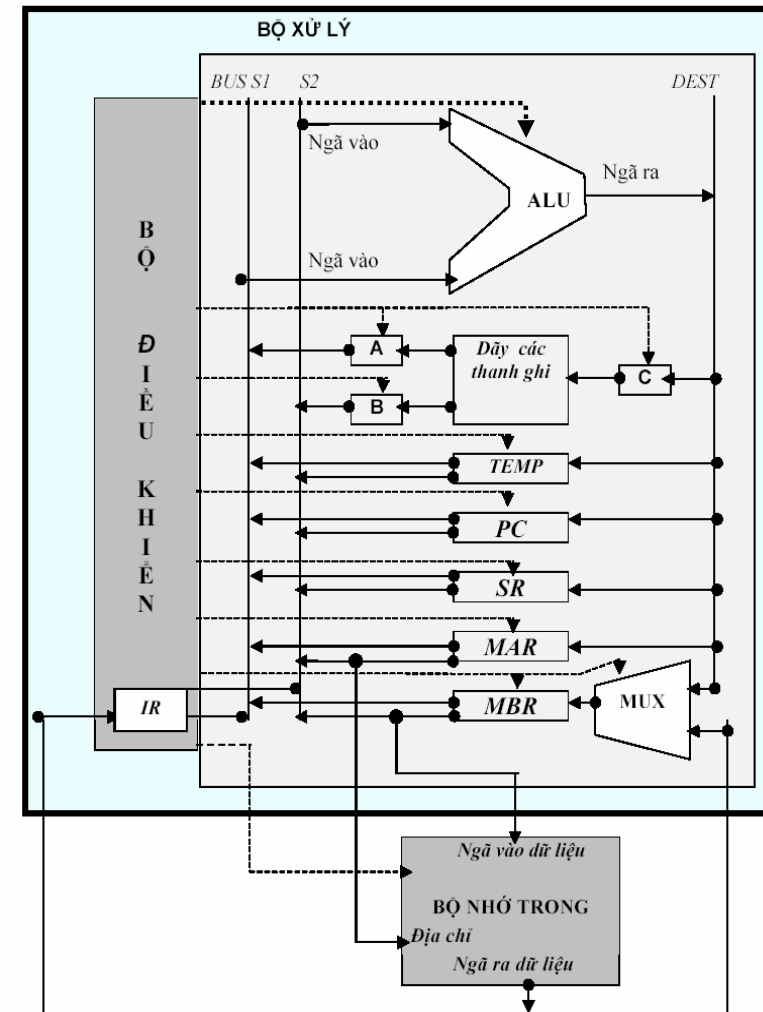
MAR \leftarrow A+B

4. Thâm nhập bộ nhớ / nhảy lần cuối:

MDR \leftarrow M[MAR]

5. Lưu trữ kết quả:

R₅ \leftarrow MDR



Hình III.1: Tổ chức của một xử lý điển hình
(Các đường không liên tục là các đường điều khiển)