Cấu trúc bộ VXL

Giáo viên: Vũ Mạnh Khánh

Mục đích – Yêu cầu

- Mục đích
  - *Giới thiệu cấu trúc của bộ xử lý trung tâm: tổ chức, chức năng và nguyên lý hoạt động*
  - *Mô tả diễn tiến thi hành một lệnh mã máy*
  - *Một số kỹ thuật xử lý thông tin*
- Yêu cầu
  - *Nắm vững cấu trúc của bộ xử lý trung tâm*
  - *Diễn tiến thi hành một lệnh mã máy*

Cấu trúc bộ VXL

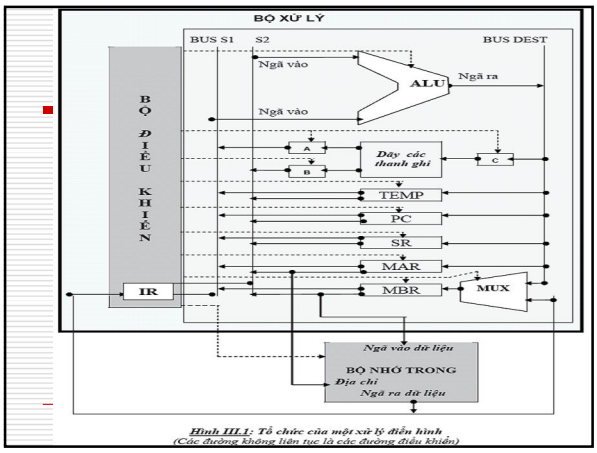
- Bộ xử lý được chia chủ yếu thành hai bộ phận
  - Phần điều khiển
  - Phần đường đi của dữ liệu (data path)

ĐƯỜNG ĐI CỦA DỮ LIỆU

- Bộ phận làm tính và luận lý (ALU: Arithmetic and Logic Unit)
- Các mạch dịch, các thanh ghi và các đường nối kết
- Thanh ghi đếm chương trình (PC: Program Counter)
- Thanh ghi trạng thái (SR: Status Register)
- Thanh ghi đệm TEMP (Temporary)
- Thanh ghi địa chỉ bộ nhớ (MAR: Memory Address Register)
- Thanh ghi số liệu bộ nhớ (MBR: Memory Buffer Register)
- Bộ đa hợp (MUX: Multiplexor), đây là điểm cuối của các kênh dữ liệu - CPU và bộ nhớ, với nhiệm vụ lập thời biểu truy cập bộ nhớ từ CPU và các kênh dữ liệu
- Hệ thống bus nguồn (S1, S2) và bus kết quả (Dest).

ĐƯỜNG ĐI CỦA DỮ LIỆU

- Nhiệm vụ:
  - Đọc các toán hạng từ các thanh ghi tổng quát
  - Thực hiện các phép tính trên toán hạng này trong bộ làm tính và luận lý ALU
  - Lưu trữ kết quả trong các thanh ghi tổng quát.



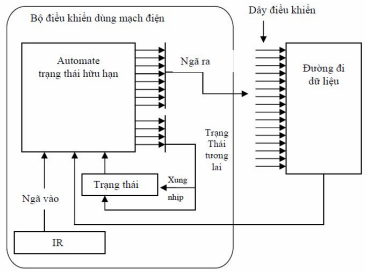
Hình III.1: Tổ chức của một xử lý điển hình (Các đường không liên tục là các đường điều khiển)

BỘ ĐIỀU KHIỂN

- Nhiệm vụ:
  - Tạo các tín hiệu điều khiển di chuyển số liệu (tín hiệu di chuyển số liệu từ các thanh ghi đến bus hoặc tín hiệu viết vào các thanh ghi)
  - Điều khiển các tác vụ mà các bộ phận chức năng phải làm (điều khiển ALU, điều khiển đọc và viết vào bộ nhớ trong...).
  - Bộ điều khiển cũng tạo các tín hiệu giúp các lệnh được thực hiện một cách tuần tự.
- Cài đặt bộ điều khiển có thể dùng một trong hai cách sau:
  - dùng mạch điện tử
  - dùng vi chương trình (microprogram).

Bộ điều khiển mạch điện tử

- Sử dụng mô hình automate trạng thái hữu hạn
- mỗi thời điểm xem xét đều có một trạng thái (state)

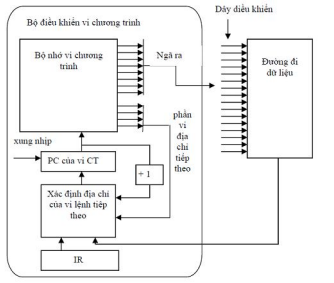


Hình III.2: Nguyên tắc vận hành của bộ điều khiển dùng mạch điện

C¸ch thøc ho¹t ®éng

- Các đường điều khiển của phần đường đi số liệu là các ngã ra của một hoặc nhiều Automate trạng thái hữu hạn.
- Các ngã vào của Automate gồm có thanh ghi lệnh, thanh ghi này chứa lệnh phải thi hành và những thông tin từ bộ đường đi số liệu.
- Ứng với cấu hình các đường vào và trạng thái hiện tại, Automate sẽ cho trạng thái tương lai và các đường ra tương ứng với trạng thái hiện tại.
- Automate được cài đặt dưới dạng là một hay nhiều mạch mảng logic lập trình được (PLA: Programmable Logic Array) hoặc các mạch logic ngẫu nhiên.

Bộ điều khiển vi chương trình



Hình III.3: Nguyên tắc vận hành của bộ điều khiển vi chương trình

C¸ch thøc ho¹t ®éng

- Các đường dây điều khiển của bộ đường đi dữ liệu ứng với các ngã ra của một vi lệnh nằm trong bộ nhớ vi chương trình.
- Việc điều khiển các tác vụ của một lệnh mã máy được thực hiện bằng một chuỗi các vi lệnh.
- Một vi máy tính nằm bên trong bộ điều khiển thực hiện từng lệnh của vi chương trình này.
- Các tác vụ của lệnh mã máy cũng tuỳ thuộc vào trạng thái của phần đường đi dữ liệu.

DIỄN TIẾN THI HÀNH LỆNH MÃ MÁY

- Việc thi hành một lệnh mã máy có thể chia thành 5 giai đoạn:
  - *Đọc lệnh* (IF: Instruction Fetch)
  - *Giải mã lệnh* (ID: Instruction Decode)
  - *Thi hành lệnh* (EX: Execute)
  - *Thâm nhập bộ nhớ trong hoặc nhảy* (MEM: Memory access)
  - *Lưu trữ kết quả* (RS: Result Storing).
- Mỗi giai đoạn được thi hành trong một hoặc nhiều chu kỳ xung nhịp

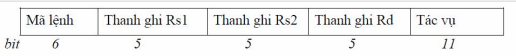
Đọc lệnh

- MAR ← PC
- IR ← M[MAR]
  - Bộ đếm chương trình PC được đưa vào MAR . Lệnh được đọc từ bộ nhớ trong, tại các ô nhớ có địa chỉ nằm trong MAR và được đưa vào thanh ghi lệnh IR.

Giải mã lệnh và đọc các thanh ghi nguồn

- A ← Rs1
- B ← Rs2
- PC ← PC + 4
  - Lệnh được giải mã. Kế đó các thanh ghi Rs1 và Rs2 được đưa vào A và B. Thanh ghi PC được tăng lên để chỉ tới lệnh kế đó.

Giải mã lệnh và đọc các thanh ghi nguồn



1. Các thanh ghi nguồn Rs1 và Rs2 được sử dụng tuỳ theo tác vụ, kết quả được đặt trong thanh ghi đích Rd.
2. Ta thấy việc giải mã được thực hiện cùng lúc với việc đọc các thanh ghi Rs1 và Rs2 vì các thanh ghi này luôn nằm tại cùng vị trí ở trong lệnh

Thi hành lệnh

- Tuỳ theo loại lệnh mà một trong ba nhiệm vụ sau đây được thực hiện:
  - Liên hệ tới bộ nhớ
    - MAR ← Địa chỉ do ALU tính tuỳ theo kiểu định vị (Rs2).
    - MBR ← Rs1
    - Địa chỉ hiệu dụng do ALU tính được đưa vào MAR và thanh ghi nguồn Rs1 được đưa vào MBR để được lưu vào bộ nhớ trong.

Thi hành lệnh

- Một lệnh của ALU
  - Ngã ra ALU ← Kết quả của phép tính
  - ALU thực hiện phép tính xác định trong mã lệnh, đưa kết quả ra ngã ra.
- Một phép nhảy
  - Ngã ra ALU ← Địa chỉ lệnh tiếp theo do ALU tính.
  - ALU cộng địa chỉ của PC với độ dời để làm thành địa chỉ đích và đưa địa chỉ này ra ngã ra. Nếu là một phép nhảy có điều kiện thì thanh ghi trạng thái được đọc quyết định có cộng độ dời vào PC hay không.

Thâm nhập bộ nhớ trong hoặc nhảy lần cuối

- Giai đoạn này thường chỉ được dùng cho các lệnh nạp dữ liệu, lưu giữ dữ liệu và lệnh nhảy.
  - Tham khảo đến bộ nhớ:
    - MBR ← M[MAR] hoặc M[MAR] ← MBR
    - Số liệu được nạp vào MBR hoặc lưu vào địa chỉ mà MAR trỏ đến.
  - Nhảy:
    - If (điều kiện), PC ← ngã ra ALU
    - Nếu điều kiện đúng, ngã ra ALU được nạp vào PC. Đối với lệnh nhảy không điều kiện, ngả ra ALU luôn được nạp vào thanh ghi PC.

Lưu trữ kết quả

- Rd ← Ngã ra ALU hoặc Rd ← MBR
- Lưu trữ kết quả trong thanh ghi đích.

NGẮT QUÃNG (INTERRUPT)

- Ngắt quãng là một sự kiện xảy ra một cách ngẫu nhiên trong máy tính và làm ngưng tính tuần tự của chương trình
- Một số tên gọi khác: Ngoại lệ, lỗi, bẫy

NGẮT QUÃNG (INTERRUPT)

- Bộ điều khiển của CPU là bộ phận khó thực hiện nhất và ngắt quãng là phần khó thực hiện nhất trong bộ điều khiển.
- Để nhận biết được một ngắt quãng lúc đang thi hành một lệnh, ta phải biết điều chỉnh chu kỳ xung nhịp và điều này có thể ảnh hưởng đến hiệu quả của máy tính.

NGẮT QUÃNG (INTERRUPT)

- Ứng dụng của ngắt
  - Trước đây:nhận biết các sai sót trong tính toán số học, và ứng dụng cho những hiện tượng thời gian thực

NGẮT QUÃNG (INTERRUPT)

- Ứng dụng của ngắt
- Ngày nay:
  - 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.

NGẮT QUÃNG (INTERRUPT)

- Chú ý:
  - Ngắt quãng không xảy ra thường xuyên nhưng bộ xử lý phải được thiết kế sao cho có thể lưu giữ trạng thái của nó trước khi nhảy đi phục vụ ngắt quãng. Sau khi thực hiện xong chương trình phục vụ ngắt, bộ xử lý phải khôi phục trạng thái của nó để có thể tiếp tục công việc.

NGẮT QUÃNG (INTERRUPT)

- Quy trình thi hành ngắt
  - *1. Thực hiện xong lệnh đang làm.*
  - *2. Lưu trữ trạng thái hiện tại.*
  - *3. Nhảy đến chương trình phục vụ ngắt*
  - *4. Khi chương trình phục vụ chấm dứt, bộ xử lý khôi phục lại trạng thái cũ của nó và tiếp tục thực hiện chương trình mà nó đang thực hiện khi bị ngắt.*

KỸ THUẬT ỐNG DẪN (PIPELINE)

- Đây là một kỹ thuật làm cho các giai đoạn khác nhau của nhiều lệnh được thi hành cùng một lúc

KỸ THUẬT ỐNG DẪN (PIPELINE)

- Ví dụ
  - Chúng ta có những lệnh đều đặn, mỗi lệnh được thực hiện trong cùng một khoản thời gian.
  - Mỗi lệnh được thực hiện trong 5 giai đoạn và mỗi giai đoạn được thực hiện trong 1 chu kỳ xung nhịp.
  - Các giai đoạn thực hiện một lệnh là: lấy lệnh (IF: Instruction Fetch), giải mã (ID: Instruction Decode), thi hành (EX: Execute), thâm nhập bộ nhớ (MEM: Memory Access), lưu trữ kết quả (RS: Result Storing).

KỸ THUẬT ỐNG DẪN (PIPELINE)

Chuỗi lệnh	Chu kỳ xung nhịp								
	1	2	3	4	5	6	7	8	9
Lệnh thứ i	IF	ID	EX	MEM	RS				
Lệnh thứ i+1		IF	ID	EX	MEM	RS			
Lệnh thứ i+2			IF	ID	EX	MEM	RS		
Lệnh thứ i+3				IF	ID	EX	MEM	RS	
Lệnh thứ i+4					IF	ID	EX	MEM	RS

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

KỸ THUẬT ỐNG DẪN (PIPELINE)

- So sánh với kiểu thông thường
  - 5 lệnh được thực hiện trong 25 chu kỳ xung nhịp,
  - xử lý lệnh theo kỹ thuật ống dẫn thực hiện 5 lệnh chỉ trong 9 chu kỳ xung nhịp.

KỸ THUẬT ỐNG DẪN (PIPELINE)

- Các ràng buộc:
  - Cần phải có một mạch điện để thi hành mỗi giai đoạn của lệnh vì tất cả các giai đoạn của lệnh được thi hành cùng lúc
  - Phải có nhiều thanh ghi khác nhau dùng cho các tác vụ đọc và viết.
  - Trong một máy có kỹ thuật ống dẫn, có khi kết quả của một tác vụ trước đó, là toán hạng nguồn của một tác vụ khác.
  - Cần phải giải mã các lệnh một cách đơn giản để có thể giải mã và đọc các toán hạng trong một chu kỳ duy nhất của xung nhịp.

KỸ THUẬT ỐNG DẪN (PIPELINE)

- Các ràng buộc:
  - Cần phải có các bộ làm tính ALU hữu hiệu để có thể thi hành lệnh số học dài nhất, có số giữ, trong một khoảng thời gian ít hơn một chu kỳ của xung nhịp.
  - Cần phải có nhiều thanh ghi lệnh để lưu giữ lệnh mà chúng ta phải xem xét cho mỗi giai đoạn thi hành lệnh.
  - Cuối cùng phải có nhiều thanh ghi bộ đếm chương trình PC để có thể tái tục các lệnh trong trường hợp có ngắt quãng.

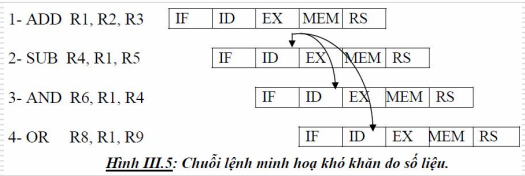
CÁC VẤN ĐỀ TRONG KỸ THUẬT ỐNG DẪN

- **Khó khăn do cấu trúc**
  - Đây là khó khăn do thiếu bộ phận chức năng, ví dụ trong một máy tính dùng kỹ thuật ống dẫn phải có nhiều ALU, nhiều PC, nhiều thanh ghi lệnh IR
- Cách giải quyết:
  - Thêm các bộ phận chức năng cần thiết và hữu hiệu

CÁC VẤN ĐỀ TRONG KỸ THUẬT ỐNG DẪN

- **Khó khăn do số liệu**
- Ví dụ
  - Lệnh 1: ***ADD R1, R2, R3***
  - Lệnh 2: ***SUB R4, R1, R5***
  - Lệnh 3: ***AND R6, R1, R7***
  - Lệnh 4: ***OR R8, R1, R9***

CÁC VẤN ĐỀ TRONG KỸ THUẬT ỐNG DẪN

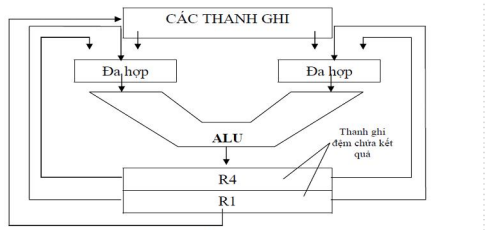


Hình III.5: Chuỗi lệnh minh hoạ khó khăn do số liệu.

R1, kết quả của lệnh 1 chỉ có thể được dùng cho lệnh 2 sau giai đoạn MEM của lệnh 1, nhưng R1 được dùng cho lệnh 2 vào giai đoạn EX của lệnh 1. Chúng ta cũng thấy R1 được dùng cho các lệnh 3 và 4.

CÁC VẤN ĐỀ TRONG KỸ THUẬT ỐNG DẪN

- Cách khắc phục



Hình III.6: ALU với bộ phận phần cứng đưa kết quả tính toán trở lại ngã vào

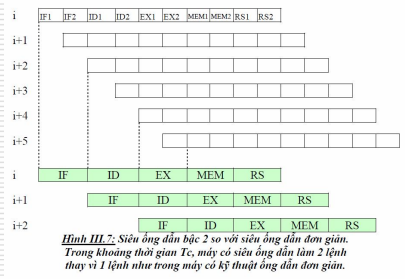
CÁC VẤN ĐỀ TRONG KỸ THUẬT ỐNG DẪN

- **Khó khăn do điều khiển**
  - Các lệnh làm thay đổi tính thi hành các lệnh một cách tuần tự (nghĩa là PC tăng đều đặn sau mỗi lệnh), gây khó khăn về điều khiển.
- Cách giải quyết
  - Cách thứ nhất là đóng băng kỹ thuật ống dẫn trong một chu kỳ, nghĩa là ngưng thi hành lệnh thứ i+1 đang làm nếu lệnh thư i là lệnh nhảy. Ta mất trắng một chu kỳ cho mỗi lệnh nhảy.
  - Cách thứ hai là thi hành lệnh sau lệnh nhảy nhưng lưu ý rằng hiệu quả của một lệnh nhảy bị chậm mất một lệnh.

SIÊU ỐNG DẪN

- Máy tính 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 khoản thời gian Tc, thành n giai đoạn con thực hiện trong khoản thời gian Tc/n.

SIÊU ỐNG DẪN

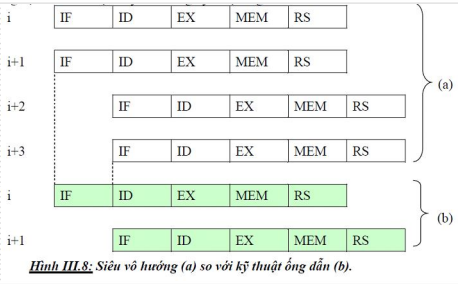


Hình III.7: Siêu ống dẫn bậc 2 so với siêu ống dẫn đơn giản. Trong khoảng thời gian Tc, máy có siêu ống dẫn làm 2 lệnh thay vì 1 lệnh như trong máy có kỹ thuật ống dẫn đơn giản.

SIÊU VÔ HƯỚNG (SUPERSCALAR)

- Máy tính siêu vô hướng bậc n có thể thực hiện đồng thời n lệnh trong một chu kỳ xung nhịp Tc.

SIÊU VÔ HƯỚNG (SUPERSCALAR)



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

MÁY TÍNH CÓ LỆNH THẬT DÀI VLIW (VERY LONG INSTRUCTION WORD)

- 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 cố thực hiện nhiều lệnh trong một chu kỳ, người ta tìm cách đưa vào nhiều lệnh trong một từ lệnh dài.

MÁY TÍNH VECTƠ

- Một máy tính vectơ bao gồm một bộ tính toán vô hướng bình thường dùng kỹ thuật ống dẫn và một bộ làm tính vectơ. Bộ tính toán vô hướng, giống như bộ xử lý dùng kỹ thuật ống dẫn, thực hiện các phép tính vô hướng, còn bộ làm tính vectơ thực hiện các phép tính vectơ.
- Có 2 kiểu kiến trúc máy tính vectơ
  - kiểu *vectơ ô nhớ - ô nhớ* :các phép tính vectơ được thực hiện trong bộ nhớ.
  - *kiểu thanh ghi vectơ*.

MÁY TÍNH SONG SONG

- 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 của chúng.
- Với sự phát triển của công nghệ máy tính 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.

MÁY TÍNH SONG SONG

- Các máy tính có thể sắp xếp vào 4 loại sau:
  - **SISD** (Single Instructions Stream, Single Data Stream): Máy tính một dòng lệnh, một dòng số liệu.
  - **SIMD** (Single Instructions Stream, Multiple Data Stream): Máy tính một dòng lệnh, nhiều dòng số liệu.
  - **MISD** (Multiple Instructions Stream, Single Data Stream):Máy tính nhiều dòng lệnh, một dòng số liệu.
  - **MIMD** (Multiple Instruction Stream, Multiple Data Stream): Máy tính nhiều dòng lệnh, nhiều dòng số liệu.
    - Các máy tính SISD tương ứng với các máy một bộ xử lý mà chúng ta đã nghiên cứu.
    - Các máy MISD kiểu máy tính này không sản xuất thương mại.

MÁY TÍNH SONG SONG

- Các máy MIMD hiện tại có thể được xếp vào ba loại hệ thống sẽ được giới thiệu trong phần tiếp theo của chương trình là:
  - *SMP (Symmetric Multiprocesors)*
  - *Cluster*
  - *NUMA (Nonunifrom Memory Access)*

MÁY TÍNH SONG SONG

- Một hệ thống SMP bao gồm nhiều bộ xử lý giống nhau được lắp đặt bên trong một máy tính, các bộ xử lý này kết nối với nhau bởi một hệ thống bus bên trong hay một vài sự sắp xếp chuyển mạch thích hợp.
- Vấn đề lớn nhất trong hệ thống SMP là sự kết hợp các hệ thống cache riêng lẻ. Vì mỗi bộ xử lý trong SMP có một cache riêng của nó, do đó, một khối dữ liệu trong bộ nhớ trong có thể tồn tại trong một hay nhiều cache khác nhau.

MÁY TÍNH SONG SONG

- Trong hệ thống cluster, các máy tính độc lập được kết nối với nhau thông qua một hệ thống kết nối tốc độ cao (mạng tốc độ cao Fast Ethernet hay Gigabit) và hoạt động như một máy tính thống nhất.

MÁY TÍNH SONG SONG

- Ưu điểm của hệ thống Cluster
  - Tốc độ cao: Có thể tạo ra một hệ thống cluster có khả năng xử lý mạnh hơn bất cứ một máy tính đơn lẻ nào. Mỗi cluster có thể bao gồm hàng tá máy tính, mỗi máy có nhiều bộ xử lý.
  - Khả năng mở rộng cao: có thể nâng cấp, mở rộng một cluster đã được cấu hình và hoạt động ổn định.
  - Độ tin cậy cao: Hệ thống vẫn hoạt động ổn định khi có một nút (node) trong hệ thống bị hư hỏng. Trong nhiều hệ thống, khả năng chịu lỗi (fault tolerance) được xử lý tự động bằng phần mềm.
  - Chi phí đầu tư thấp: hệ thống cluster có khả năng mạnh hơn một máy tính đơn lẻ mạnh nhất với chi phí thấp hơn.

MÁY TÍNH SONG SONG

- Một hệ thống NUMA (Nonunifrom Memory Access) là hệ thống đa xử lý được giới thiệu trong thời gian gần đây, đây là hệ thống với bộ nhớ chia sẻ, thời gian truy cập các vùng nhớ dành riêng cho các bộ xử lý thì khác nhau.

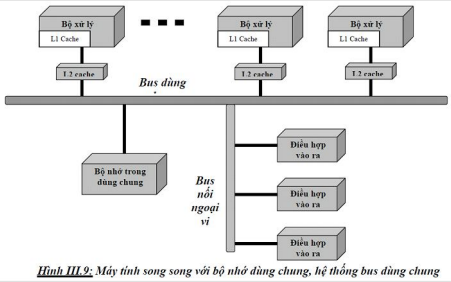
MÁY TÍNH SONG SONG

- *Thuận lợi:*
  - Thực hiện hiệu quả hơn so với hệ thống SMP trong các xử lý song song.
  - Không thay đổi phần mềm chính.
  - Bộ nhớ có khả năng bị nghẽn nếu có nhiều truy cập đồng thời, nhưng điều này có thể được khắc phục bằng cách:
  - Cache L1&L2 được thiết kế để giảm tối thiểu tất cả các thâm nhập bộ nhớ.
  - Cần các phần mềm cục bộ được quản lý tốt để việc các ứng dụng hoạt động hiệu quả.
  - Quản trị bộ nhớ ảo sẽ chuyển các trang tới các nút cần dùng.

MÁY TÍNH SONG SONG

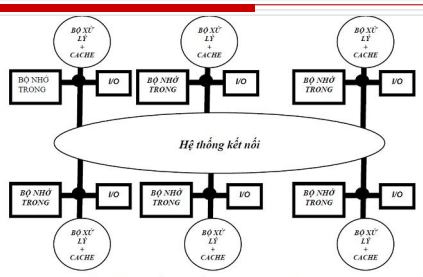
- *Bất lợi:*
  - Hệ thống hoạt động không trong suốt như SMP: việc cấp phát các trang, các quá trình có thể được thay đổi bởi các phần mềm hệ thống nếu cần.
  - Hệ thống phức tạp.

MÁY TÍNH SONG SONG



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

MÁY TÍNH SONG SONG



Hình III.10: Cấu trúc nền của một bộ nhớ phân tán