



TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN



KIẾN TRÚC MÁY TÍNH

Giảng viên: TS. Đoàn Thị Quế

Bộ môn: Mạng và an toàn thông tin

Chương 7. Thiết bị ngoại vi

7.1 Các thiết bị ngoại vi

7.2 Module vào/ra

7.3 Các kỹ thuật I/O

- a. I/O chương trình

- b. I/O điều khiển ngắt

- c. Truy cập bộ nhớ trực tiếp

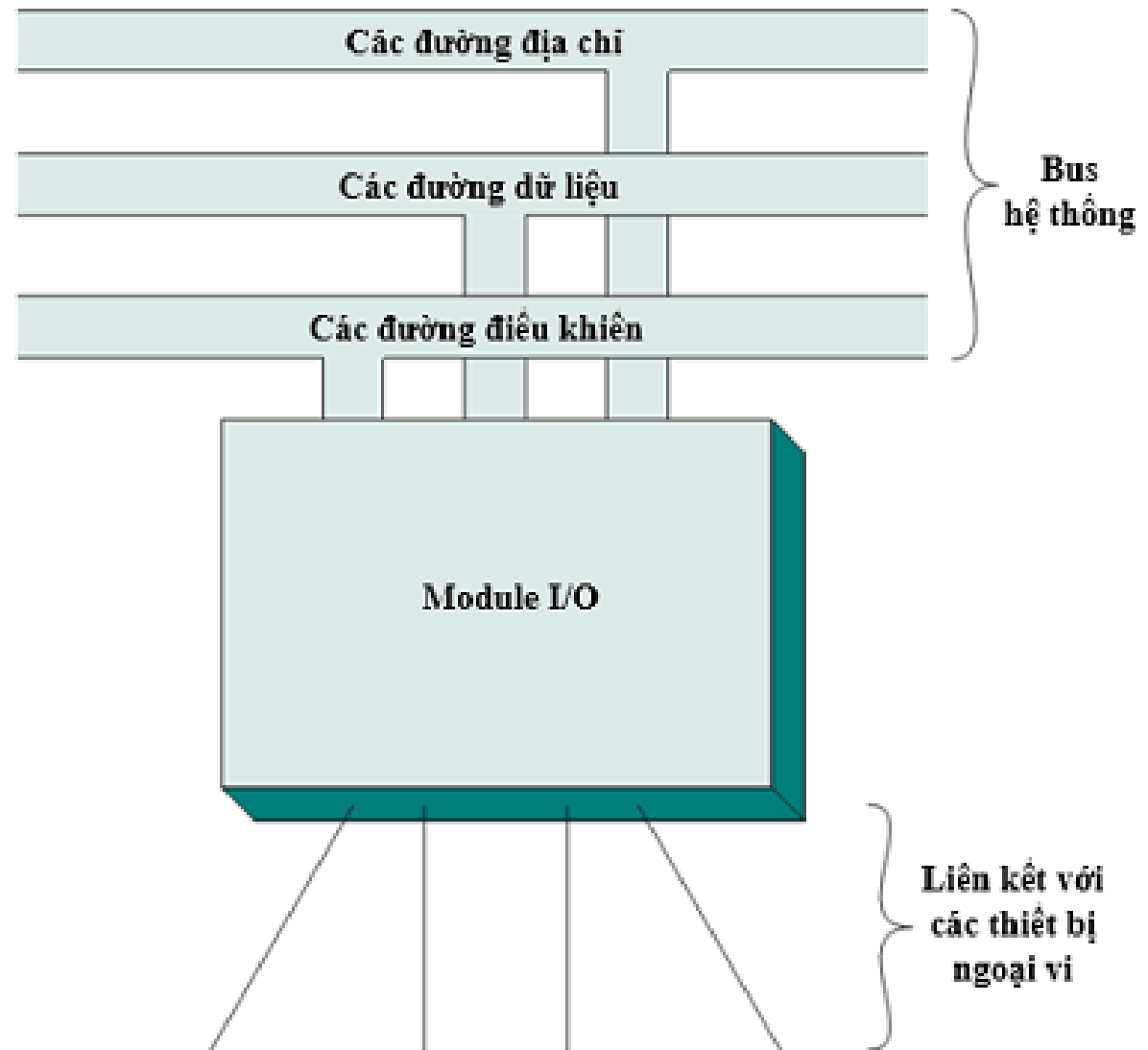
7.4 Các bộ xử lý và kênh vào/ra

7.5 Giao diện ngoài

7.1 Thiết bị ngoại vi

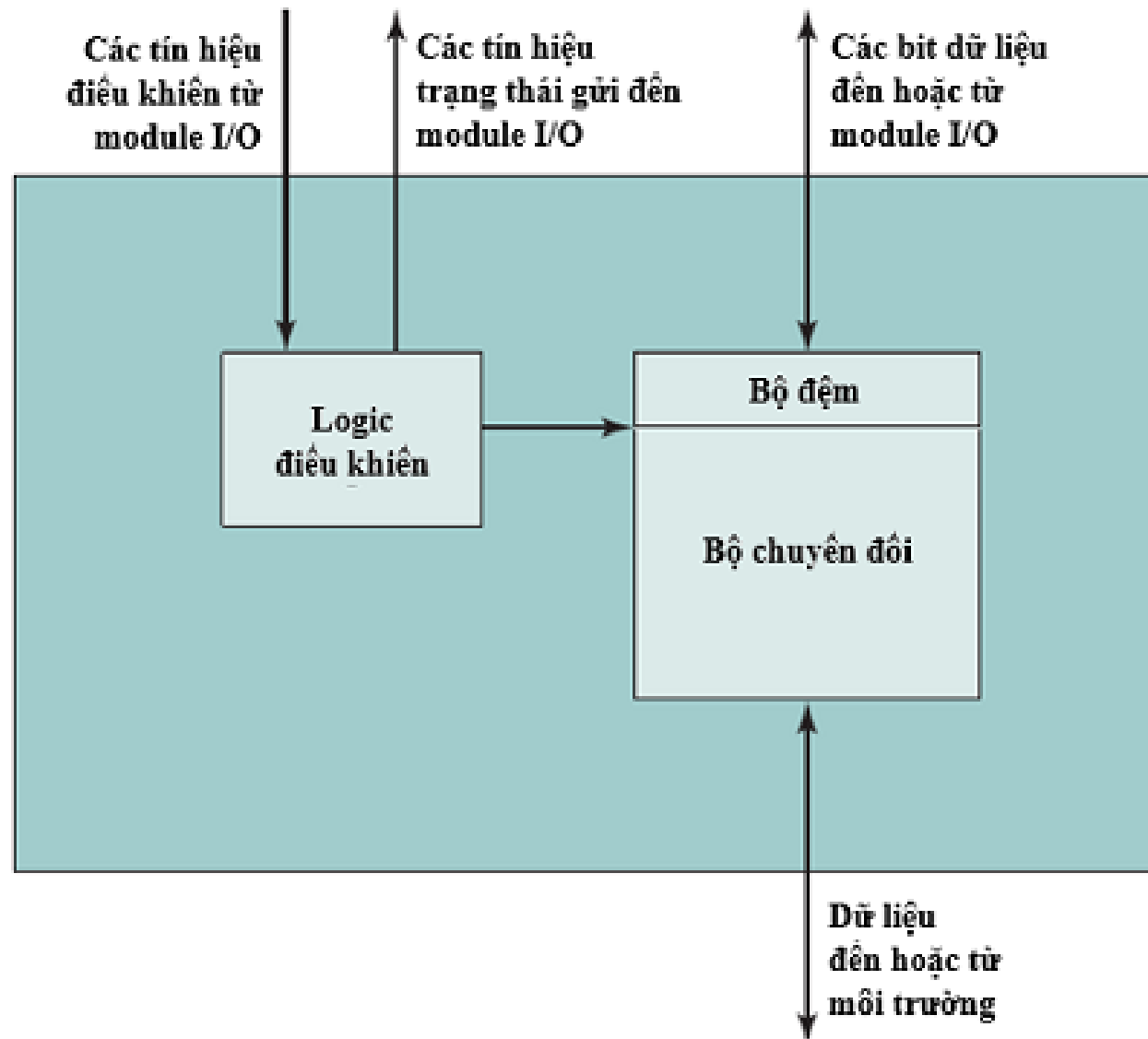
- Một trong ba thành phần cơ bản của hệ thống máy tính: CPU, bộ nhớ và thiết bị ngoại vi (thông qua module vào/ra)
- Chức năng: Cung cấp phương tiện trao đổi dữ liệu giữa môi trường bên ngoài và máy tính
- Kết nối với máy tính qua module vào/ra
 - Module vào/ra: Truyền các thông tin điều khiển, dữ liệu và địa chỉ giữa CPU và thiết bị ngoại vi
- Có ba loại
 - Con người đọc được: màn hình, máy in,...
 - Máy đọc được: ổ cứng, cảm biến, băng từ,...
 - Truyền thông: modem, card mạng,...

Mô hình tổng quát của Module vào/ra



Hình 7.1 Mô hình cơ bản của các module vào/ra

a. Sơ đồ khối thiết bị ngoại vi



Hình 7.2 Sơ đồ khối của các thiết bị ngoại vi

a. Sơ đồ khối thiết bị ngoại vi (tiếp)

- **Giao diện với module vào/ra:**
 - *Tín hiệu điều khiển - Control signal:* xác định chức năng mà thiết bị sẽ thực hiện:
 - READ: yêu cầu thiết bị gửi dữ liệu vào module vào/ra (INPUT)
 - WRITE: yêu cầu thiết bị nhận dữ liệu từ module vào/ra (OUTPUT)
 - Các tín hiệu điều khiển đặc biệt
 - *Dữ liệu – Data:* một tập các bit được gửi đến hoặc đi từ module vào/ra.
 - *Tín hiệu trạng thái - Status signal:* cho biết trạng thái của thiết bị. Ví dụ:
 - READY: thiết bị sẵn sàng cho việc truyền dữ liệu.
 - NOT-READY: không sẵn sàng truyền dữ liệu
- **Logic điều khiển – Control logic:** nhận các tín hiệu điều khiển từ module vào/ra, điều khiển hoạt động của thiết bị.
- **Bộ chuyển đổi - Transducer:** chuyển đổi dữ liệu (đang ở dạng tín hiệu điện) sang các dạng khác (ví dụ: điểm ảnh trên màn hình,...) và ngược lại.
- **Bộ đệm (buffer)** để lưu trữ tạm dữ liệu đang được chuyển giao giữa module vào/ra và môi trường bên ngoài; kích thước bộ đệm thường từ 8 đến 16 bit.

b. Bàn phím/Màn hình

Bảng chữ cái tham khảo quốc tế (IRA)

- Ký tự
 - Gắn với mỗi ký tự là một mã
 - Mỗi ký tự được biểu diễn bởi một mã nhị phân, ví dụ 7-bit : biểu diễn 128 ký tự
- Hai loại ký tự:
 - In được
 - Các ký tự chữ cái, số và ký tự đặc biệt có thể được in trên giấy hoặc hiển thị trên màn hình
 - Điều khiển
 - Điều khiển việc in/hiển thị các ký tự
 - Ví dụ: carriage return
 - Các ký tự điều khiển khác liên quan đến các thủ tục truyền tin

Công cụ tương tác máy tính/ người dùng phổ biến nhất
Người dùng cung cấp đầu vào thông qua bàn phím
Màn hình hiển thị dữ liệu được cung cấp bởi máy tính
Đơn vị chuyển đổi cơ bản là ký tự

Mã bàn phím

- Khi người dùng bấm một phím, một tín hiệu điện tử được tạo ra bởi một bộ chuyển đổi trong bàn phím và dịch sang mẫu bit của mã IRA tương ứng
- Mẫu bit này được truyền đến module vào/ra trong máy tính
- Trên đầu ra, các ký tự mã IRA được truyền đến một thiết bị ngoại vi từ module vào/ra
- Bộ chuyển đổi giải mã và gửi các tín hiệu điện tử yêu cầu đến thiết bị đầu ra để hiển thị ký tự được chỉ định hoặc thực hiện chức năng điều khiển yêu cầu

Chương 7. Thiết bị ngoại vi

7.1 Các thiết bị ngoại vi

7.2 Module vào/ra

7.3 Các kỹ thuật I/O

- a. I/O chương trình
- b. I/O điều khiển ngắt
- c. Truy xuất bộ nhớ trực tiếp

7.4 Các bộ xử lý và kênh vào/ra

7.5 Giao diện ngoài

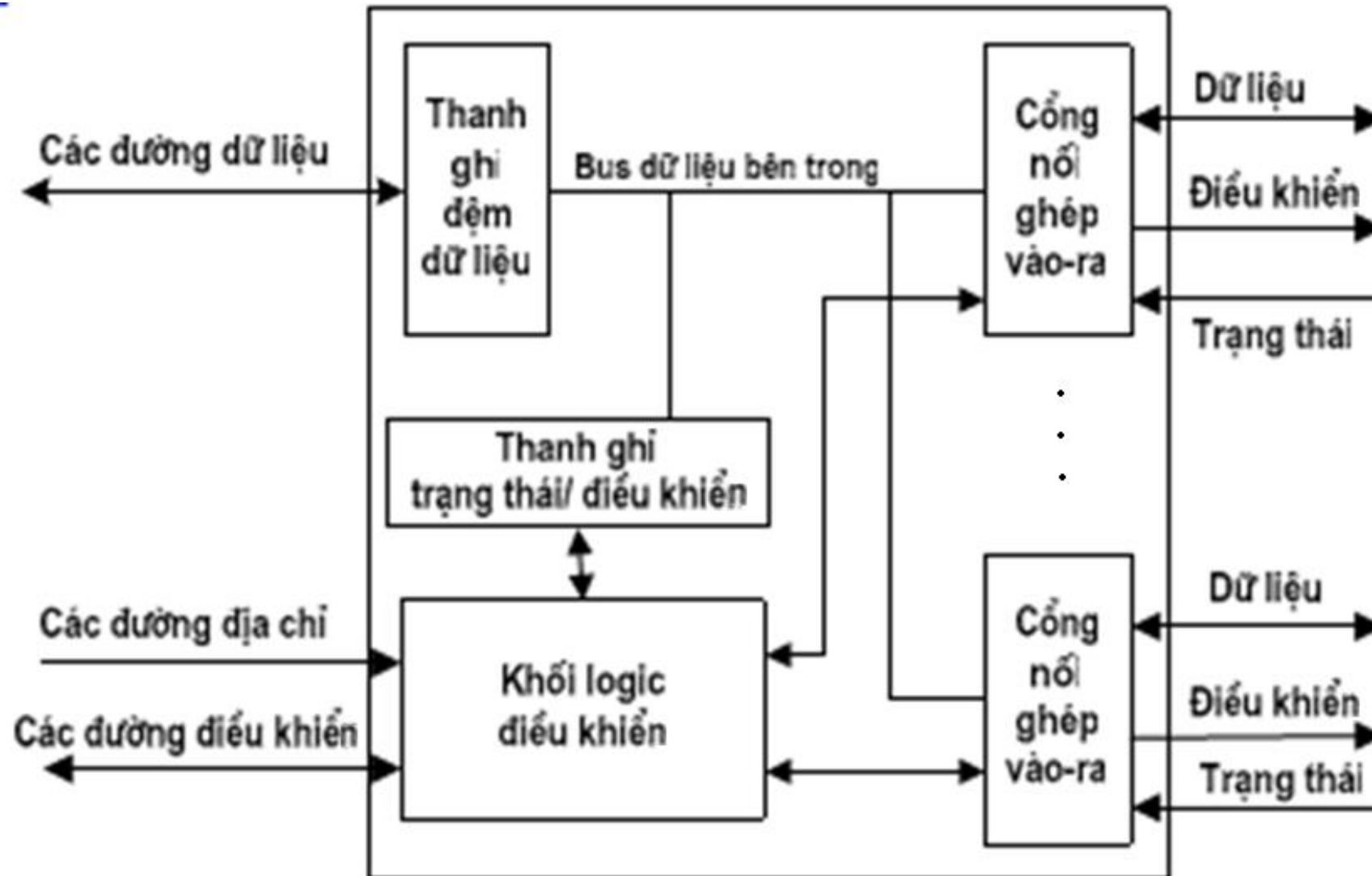
7.2 Module vào/ra

a. Chức năng

Các chức năng chính của một module vào/ra gồm:

- **Điều khiển và định thời:** phối hợp luồng lưu lượng truy cập giữa thành phần thiết bị bên trong (main memory, bus) và thiết bị ngoại vi
- **Trao đổi thông tin với VXL:** gồm giải mã lệnh, dữ liệu, báo cáo trạng thái (trạng thái của thiết bị vào/ra có sẵn sàng hay không), nhận dạng địa chỉ (địa chỉ các cổng mà TBNV được nối vào)
- **Trao đổi thông tin với TBNV:** gồm các lệnh, thông tin trạng thái và dữ liệu
- **Đệm dữ liệu:** thực hiện các hoạt động đệm cần thiết để cân bằng tốc độ TBNV và bộ nhớ
- **Phát hiện và báo cáo lỗi**

b. Cấu trúc Module vào/ra



b. Cấu trúc Module vào/ra (tiếp)

Các module vào/ra thay đổi khác nhau theo sự phức tạp và số lượng các thiết bị ngoài mà nó điều khiển. Cấu trúc chung nhất:

- Dữ liệu được truyền đến và đi từ module vào/ra được đệm qua một hoặc nhiều **thanh ghi dữ liệu (data register)**.
- **Thanh ghi trạng thái/ điều khiển (status/control register)**: lưu trữ thông tin trạng thái của thiết bị hoặc thông tin điều khiển của bộ VXL
- **Khối logic điều khiển - I/O logic**: tương tác với VXL qua một tập các **đường điều khiển (control line)**. VXL sử dụng các đường điều khiển để ra lệnh cho module vào/ra. Module vào/ra cũng có thể sử dụng một số đường điều khiển để gửi các tín hiệu phản xử bus hoặc tín hiệu trạng thái.
- Module cũng có khả năng nhận diện và sinh ra các địa chỉ với mỗi thiết bị được nối đến nó (địa chỉ cổng). Mỗi module vào/ra có một (nếu chỉ nối với một TBNV) hoặc một tập địa chỉ (nếu module nối với nhiều TBNV)
- **Cổng nối ghép vào/ra (External Device Interface Logic)**: giao tiếp với thiết bị ngoại vi

c. Địa chỉ cổng vào/ra

- Cũng giống như bộ nhớ, các TBNV được gắn vào module vào/ra qua các cổng. Để CPU giao tiếp được với các TBNV, các cổng này phải được gán một địa chỉ

Không gian địa chỉ vào/ra

- Có hai phương thức thực hiện không gian địa chỉ cho các TBNV:
 - ***I/O ánh xạ bộ nhớ (memory-mapped I/O):***
 - Bộ nhớ và TBNV chia sẻ chung không gian địa chỉ. VXL coi các thanh ghi dữ liệu và trạng thái như các ô nhớ và sử dụng cùng các lệnh để truy cập cả bộ nhớ và thiết bị ngoại vi
 - Chỉ sử dụng một đường đọc và ghi, do đó bus phải sắp xếp giữa việc đọc/ghi bộ nhớ và vào/ra TBNV
 - ***I/O riêng biệt (isolated I/O):***
 - Sử dụng một đường command line để xác định: địa chỉ bộ nhớ hay địa chỉ TBNV
 - Toàn bộ dải địa chỉ dùng cho cả hai. Ví dụ: 10 đường địa chỉ cho phép đánh địa chỉ 1024 ô nhớ và 1024 TBNV
 - Tập các chỉ lệnh đến bộ nhớ và TBNV khác nhau

Ví dụ

- Bộ xử lý 680x0 của Motorola : quản lý một không gian địa chỉ chung cho cả bộ nhớ và I/O.
- Bộ xử lý Intel Pentium:
 - Không gian địa chỉ bộ nhớ = 2^{32} địa chỉ
 - Không gian địa chỉ vào-ra = 2^{16} địa chỉ
 - Tín hiệu điều khiển: M/IO
 - Lệnh vào-ra chuyên dụng: IN, OUT

Chương 7. Thiết bị ngoại vi

7.1 Các thiết bị ngoại vi

7.2 Module vào/ra

7.3 Các kỹ thuật I/O

- a. I/O chương trình
- b. I/O điều khiển ngắt
- c. Truy xuất bộ nhớ trực tiếp

7.4 Các bộ xử lý và kênh vào/ra

7.5 Giao diện ngoài

7.3. Các kỹ thuật I/O

Hoạt động của module vào/ra theo 3 kỹ thuật sau:

■ I/O chương trình

- CPU thực thi một chương trình trực tiếp điều khiển các hoạt động vào/ra
- Khi bộ xử lý ra lệnh, nó phải đợi cho đến khi hoạt động vào/ra hoàn tất
- Bộ xử lý chạy nhanh hơn module vào/ra sẽ gây lãng phí thời gian xử lý

■ I/O điều khiển ngắt

- Bộ vi xử lý ra lệnh I/O sau đó tiếp tục thi hành các lệnh tiếp theo trong chương trình.
- Khi module vào/ra hoàn thành công việc, nó sẽ gửi tín hiệu yêu cầu ngắt đến VXL.

• Truy cập bộ nhớ trực tiếp (DMA)

- Module vào/ra và bộ nhớ chính trực tiếp trao đổi dữ liệu mà không có sự tham gia của bộ vi xử lý

Các kỹ thuật I/O

Bảng 7.1 Các kỹ thuật vào/ra

| | Không có ngắt | Sử dụng ngắt |
|---|------------------|---------------------------------|
| Truyền dữ liệu giữa bộ nhớ và thiết bị ngoại vi thông qua bộ xử lý | I/O chương trình | I/O điều khiển ngắt |
| Truyền dữ liệu trực tiếp giữa bộ nhớ và thiết bị ngoại vi | | Truy cập bộ nhớ trực tiếp (DMA) |

a. Kỹ thuật I/O chương trình

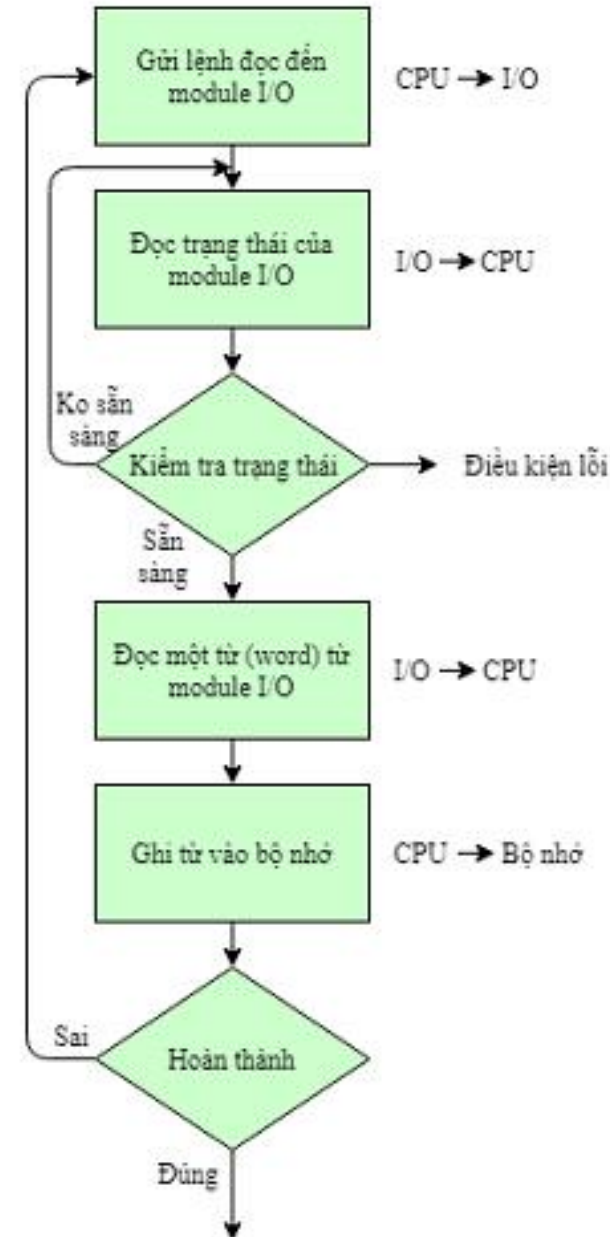
- Khi cần thực hiện một tác vụ vào/ra:
 - VXL thực thi một chương trình và gửi lệnh đến module vào/ra tương ứng
 - Module vào/ra nhận yêu cầu, thiết lập các bit trạng thái trên thanh ghi trạng thái
 - CPU định kỳ kiểm tra trạng thái của module vào/ra
 - Chưa sẵn sàng thì tiếp tục định kỳ kiểm tra
 - Đã sẵn sàng, thiết lập việc truyền dữ liệu đến module vào/ra

a. Các lệnh vào/ra – I/O command

Để thực thi một lệnh vào/ra, VXL thực hiện công việc sau:

- **Đặt địa chỉ lên bus địa chỉ:** định ra module vào/ra và TBNV cụ thể
- **Đưa các mệnh lệnh vào/ra:** Thiết lập các đường điều khiển trong bus điều khiển. Có 4 loại mệnh lệnh vào/ra:
 - 1) **Control:** kích hoạt một thiết bị ngoại vi và chỉ định nó phải làm gì
 - 2) **Test:** kiểm tra các điều kiện trạng thái liên quan đến một module I/O và các thiết bị ngoại vi: *TBNV bật hay tắt, hoạt động I/O đang thực hiện đã xong chưa, có lỗi gì*
 - 3) **Read:** yêu cầu đọc dữ liệu từ TBNV vào VXL
 - Module vào/ra lấy dữ liệu từ thiết bị ngoại vi và đặt nó vào bộ đệm bên trong → đặt dữ liệu vào bus cho CPU
 - 4) **Write:** yêu cầu ghi dữ liệu ra TBNV
 - Module I/O lấy dữ liệu từ bus dữ liệu rồi chuyển dữ liệu đó đến thiết bị ngoại vi

Ví dụ: Hoạt động đọc (READ) từ thiết bị ngoại vi vào RAM



b. I/O điều khiển ngắt

- Vấn đề với **I/O chương trình** là bộ xử lý phải đợi một thời gian dài để module vào/ra sẵn sàng cho việc nhận hoặc truyền dữ liệu
- Giải pháp thay thế là VXL ra lệnh vào/ra cho module vào/ra, sau đó nó thực hiện các việc khác.
- Khi module vào/ra sẵn sàng trao đổi dữ liệu với VXL, nó sẽ gửi tín hiệu ngắt đến VXL
- VXL thực hiện việc truyền dữ liệu và tiếp tục quá trình xử lý trước đó

Cơ chế làm việc

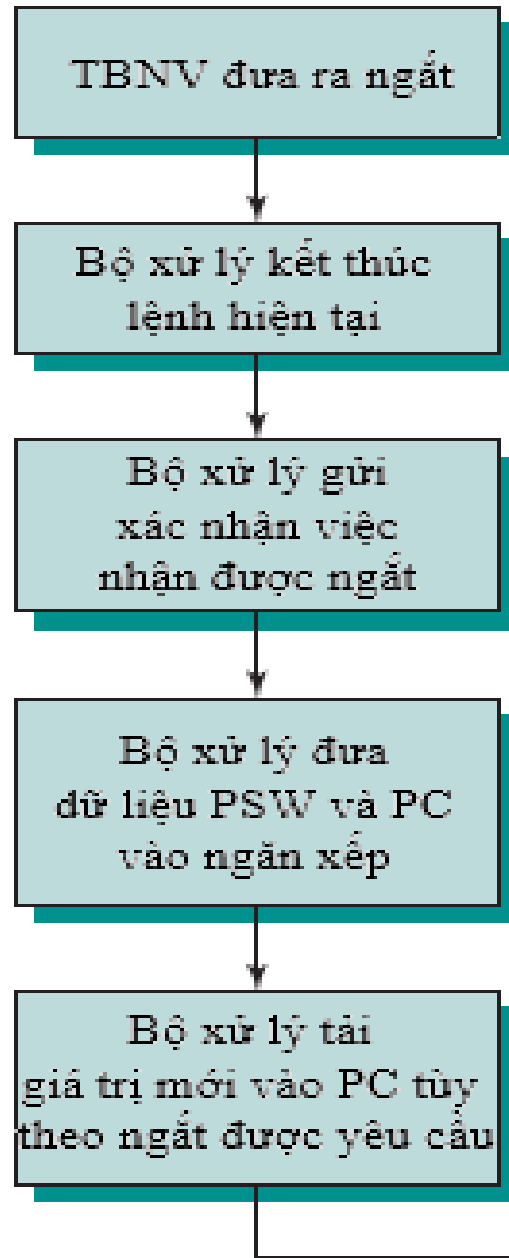
Từ phía VXL

- VXL đưa ra lệnh READ.
- Sau đó thực hiện các công việc khác (ví dụ: trong trường hợp có nhiều chương trình đang chạy tại một thời điểm)
- Sau mỗi chu kỳ lệnh, VXL sẽ kiểm tra xem có tín hiệu yêu cầu ngắt được gửi tới
- Nếu có, VXL lưu trữ nội dung đang thực hiện và xử lý ngắt
- VXL nhận dữ liệu từ bus lưu trữ vào bộ nhớ và tiếp tục chương trình

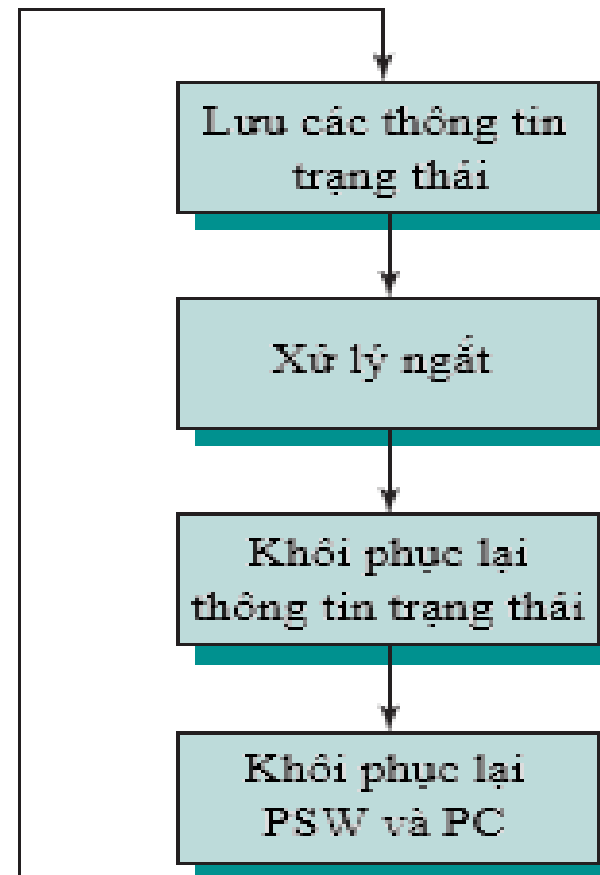
Từ phía module vào/ra

- Nhận lệnh READ từ VXL
- Đọc dữ liệu vào từ TBNV tương ứng
- Khi dữ liệu được đưa vào thanh ghi dữ liệu, module vào/ra gửi tín hiệu yêu cầu ngắt đến VXL và chờ đợi tín hiệu yêu cầu dữ liệu từ VXL.
- Khi có tín hiệu đó, module vào/ra đặt dữ liệu vào bus và sẵn sàng để thực hiện các hoạt động I/O khác

Phần cứng



Phần mềm



Cơ chế xử lý ngắt

VXL nhận được yêu cầu ngắt

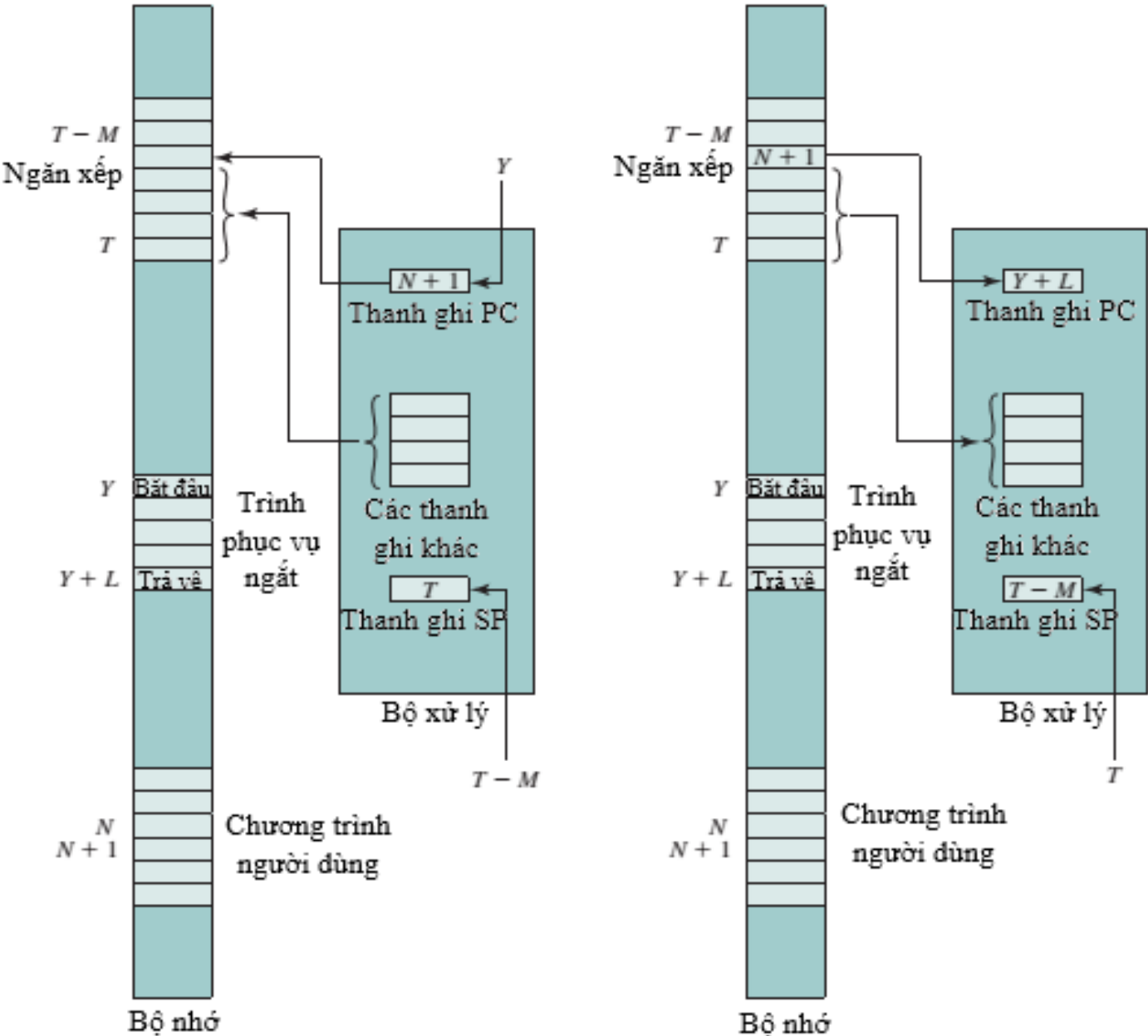
- Thiết bị phát tín hiệu ngắt cho bộ xử lý
- Bộ xử lý hoàn thành lệnh hiện tại → Kiểm tra thấy có yêu cầu ngắt → gửi ACK báo đã nhận ngắt.
- Chuyển sang chế độ phục vụ ngắt: lưu trữ nội dung các thanh ghi vào vùng ngăn xếp của RAM (hình trang sau)
- Tải trình điều khiển ngắt: đặt địa chỉ đầu tiên của trình này vào thanh ghi PC → thực hiện hoạt động vào/ra

VXL thực hiện xong yêu cầu ngắt

- Sau khi thực hiện xong hoạt động vào/ra, CPU khôi phục lại công việc đang thực hiện
- Nạp lại nội dung từ vùng ngăn xếp vào các thanh ghi
- Khôi phục lại luồng điều khiển

a. Quá trình chuyển sang chế độ phục vụ ngắt

b. Khôi phục lại luồng điều khiển sau khi thực hiện yêu cầu ngắt xong



a. Lưu chương trình hiện tại trước khi xử lý ngắt

b. Khôi phục lại chương trình

Hai vấn đề phát sinh

Hai vấn đề thiết kế phát sinh khi thực hiện I/O điều khiển ngắt:

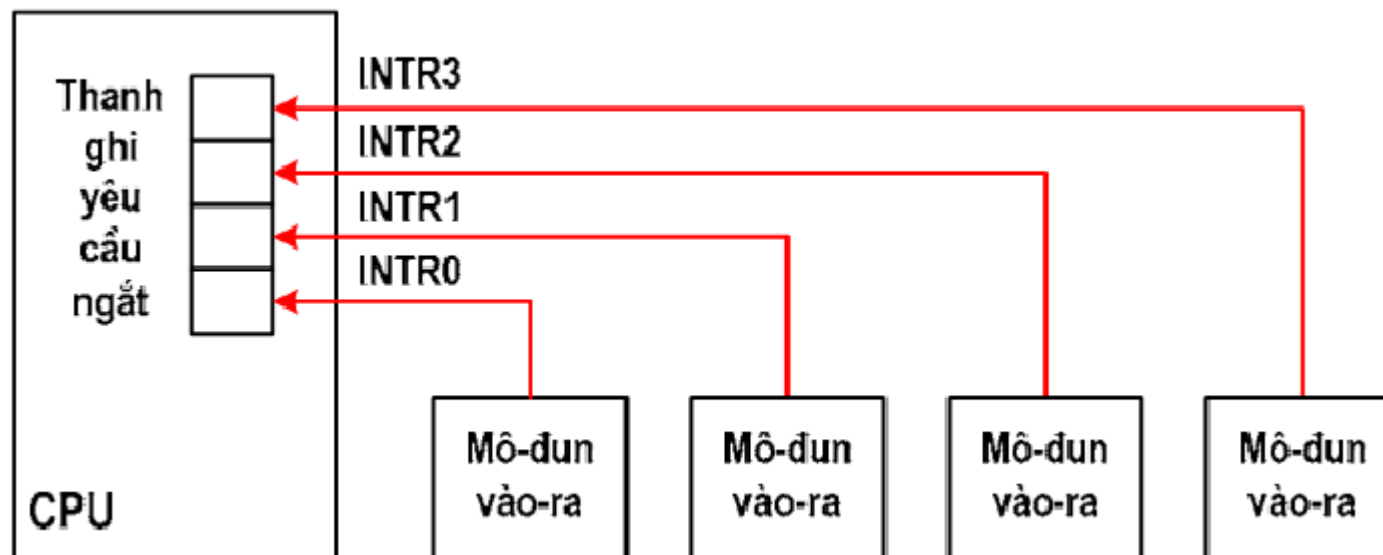
- 1) **Nhận diện thiết bị:** Bởi vì sẽ có nhiều module vào/ra, khi có một yêu cầu ngắt gửi tới, bộ vi xử lý sẽ xác định thiết bị đưa ra yêu cầu ngắt bằng cách nào?
- 2) **Xác định ưu tiên.** Nếu xảy ra nhiều ngắt cùng một thời điểm, VXL lựa chọn ngắt nào để xử lý?

1) Nhận diện thiết bị

Bốn loại kỹ thuật chung được sử dụng phổ biến:

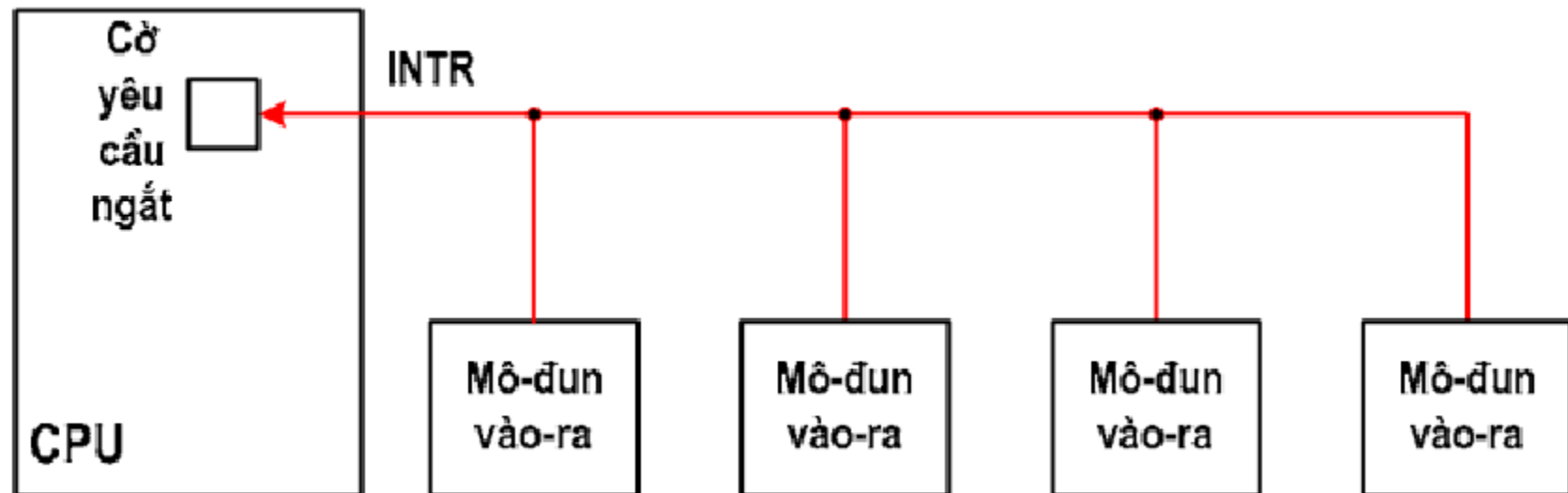
- **Nhiều đường ngắt**

- Sử dụng nhiều đường ngắt giữa VXL và các module vào/ra → dễ dàng xác định thiết bị
- Không thực tế do kỹ thuật này làm tăng số đường bus và các chân của VXL. Thêm vào đó, vẫn phải có nhiều module vào/ra nối với một đường → vẫn cần một trong ba kỹ thuật còn lại



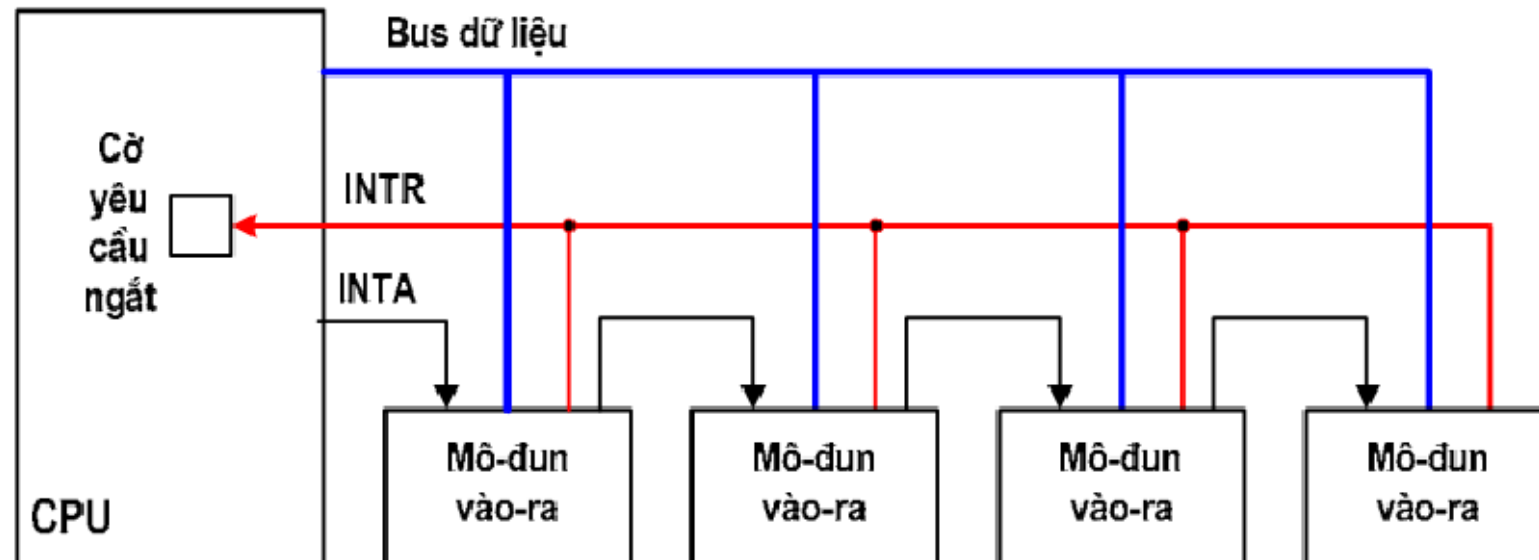
• Thăm dò phần mềm

- Khi bộ xử lý phát hiện ra một ngắt, nó thực thi một phần mềm thăm dò:
 - Thực hiện lệnh thăm dò (ví dụ: lệnh TEST I/O) tới từng module vào/ra (thông qua địa chỉ) → Module vào/ra sẽ trả lời nếu nó đưa ra yêu cầu ngắt.
 - Hoặc thực hiện lệnh đọc thanh ghi trạng thái của từng module vào/ra để phát hiện module vào/ra nào có yêu cầu ngắt
- Nhược điểm: Tốn thời gian



▪ Chuỗi Daisy (thăm dò phần cứng, vector)

- Tất cả các module vào/ra sử dụng chung một đường yêu cầu ngắt (INTR)
- Đường nhận biết ngắt (INTA) được nối chuỗi qua các module vào/ra
- Khi VXL nhận được yêu cầu ngắt, nó sẽ gửi lại một tín hiệu ACK qua đường INTA
- Tín hiệu này truyền qua các module vào/ra đến khi gặp module vào/ra yêu cầu ngắt. Module này trả lời bằng cách đặt một word lên bus dữ liệu: được gọi là **vector ngắt** (chứa thông tin địa chỉ của module vào/ra hoặc mã nhận dạng thiết bị khác).
- VXL sử dụng vector này trở tới trình phục vụ ngắt tương ứng của thiết bị → ngắt vector



■ **Phân xử bus (cũng sử dụng vector ngắt)**

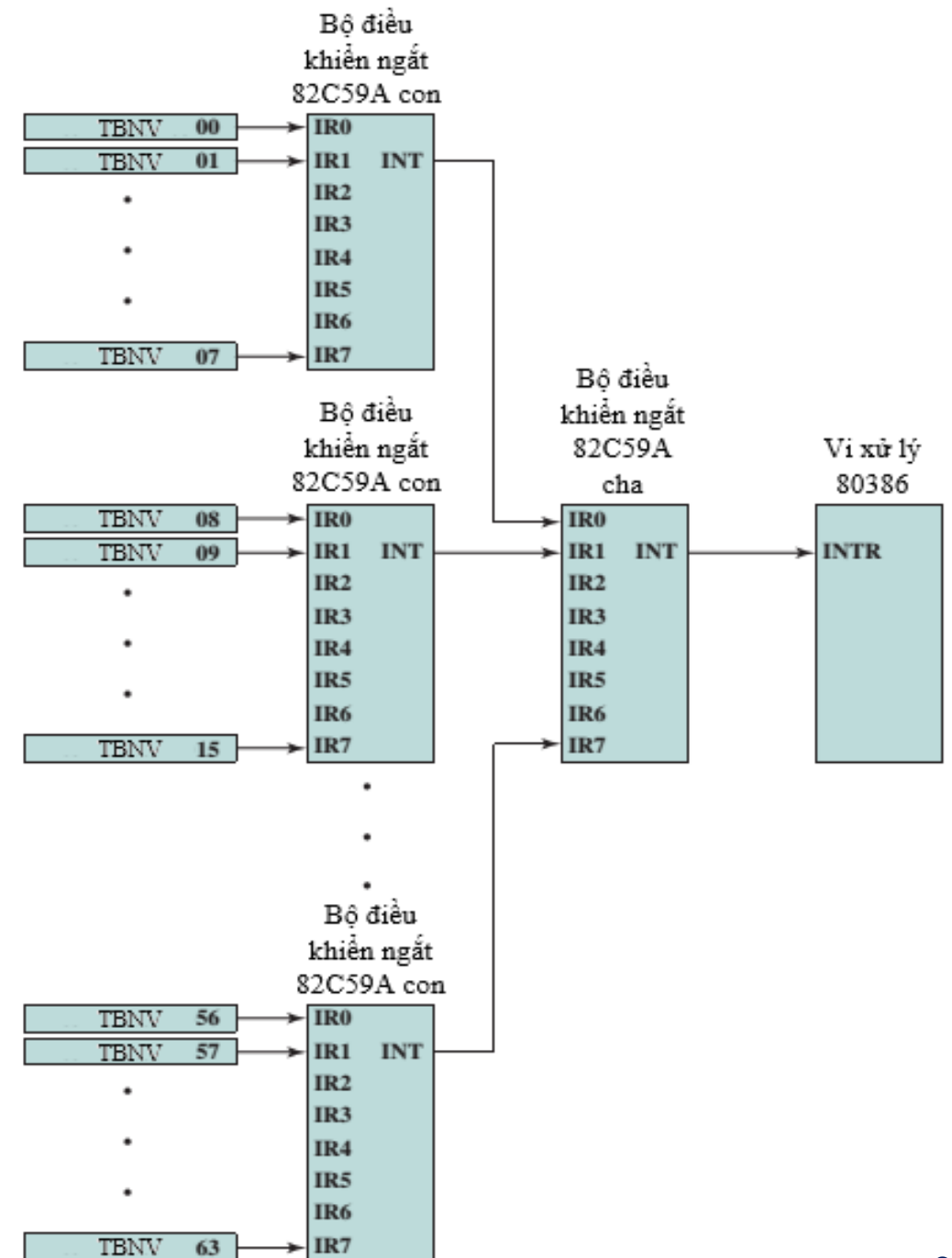
- Sử dụng cơ chế cho phép một module vào/ra chiếm quyền sử dụng bus rồi mới gửi yêu cầu ngắt
- Khi bộ xử lý phát hiện ra ngắt, nó trả lời trên đường ACK.
- Module vào/ra đặt vector ngắt của nó lên các đường dữ liệu
- VXL sử dụng vector này trở tới trình phục vụ ngắt tương ứng của thiết bị

2) Xác định ưu tiên

- Các phương pháp nhận diện thiết bị đồng thời cho phép xác định độ ưu tiên của các thiết bị khi có nhiều yêu cầu ngắt cũng một thời điểm:
 - Nhiều đường ngắt: VXL sẽ chọn đường có độ ưu tiên cao hơn để xử lý trước
 - Thăm dò phần mềm: thứ tự thăm dò các thiết bị được sắp xếp theo độ ưu tiên
 - Chuỗi daisy: tương tự thăm dò phần mềm
 - Phân xử bus: cơ chế phân xử bus đã có phân xử theo độ ưu tiên

Ví dụ

- Kiến trúc dòng máy sử dụng chip Intel 80386: thực hiện việc điều khiển ngắt thông qua vi điều khiển Intel 82C59A

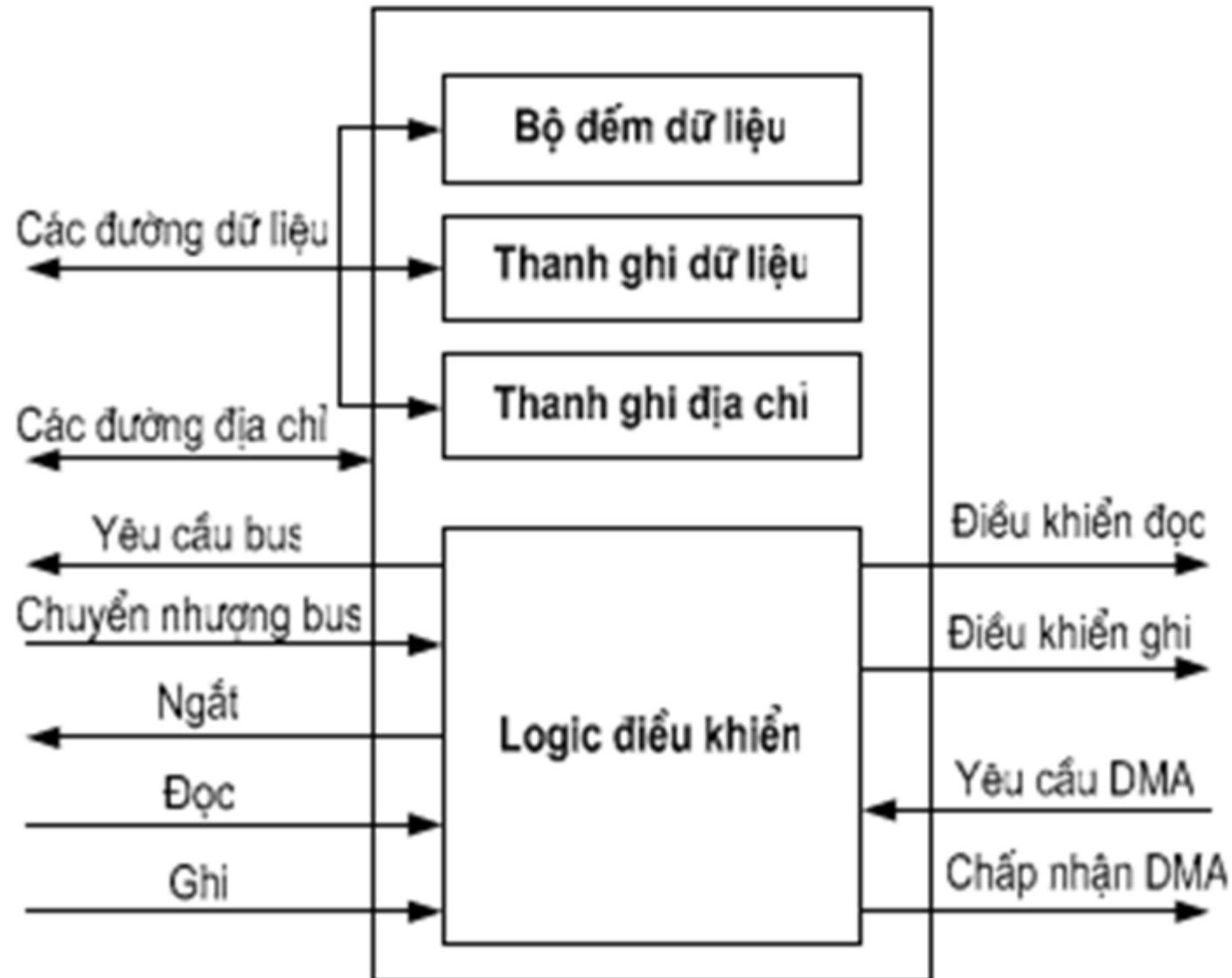


c. Truy cập bộ nhớ trực tiếp - DMA

- Nhược điểm của I/O chương trình và I/O điều khiển ngắt: VXL tham gia vào hầu hết chu trình truyền/nhận dữ liệu giữa TBNV và bộ nhớ
 - 1) Tốc độ truyền vào/ra bị giới hạn bởi tốc độ kiểm tra và phục vụ thiết bị của bộ xử lý
 - 2) Bộ xử lý gắn với việc quản lý truyền vào/ra; Một số chỉ lệnh phải được thực hiện cho mỗi lần truyền vào/ra

Khi khối lượng dữ liệu lớn được di chuyển, một kỹ thuật hiệu quả hơn là truy cập bộ nhớ trực tiếp (DMA - Direct memory access)

Sơ đồ cấu trúc của DMAC



Các thành phần của DMAC

- Thanh ghi dữ liệu: chứa dữ liệu trao đổi
- Thanh ghi địa chỉ: chứa địa chỉ ngăn nhớ dữ liệu
- Bộ đếm dữ liệu: chứa số từ dữ liệu cần trao đổi
- Logic điều khiển: điều khiển hoạt động của DMAC

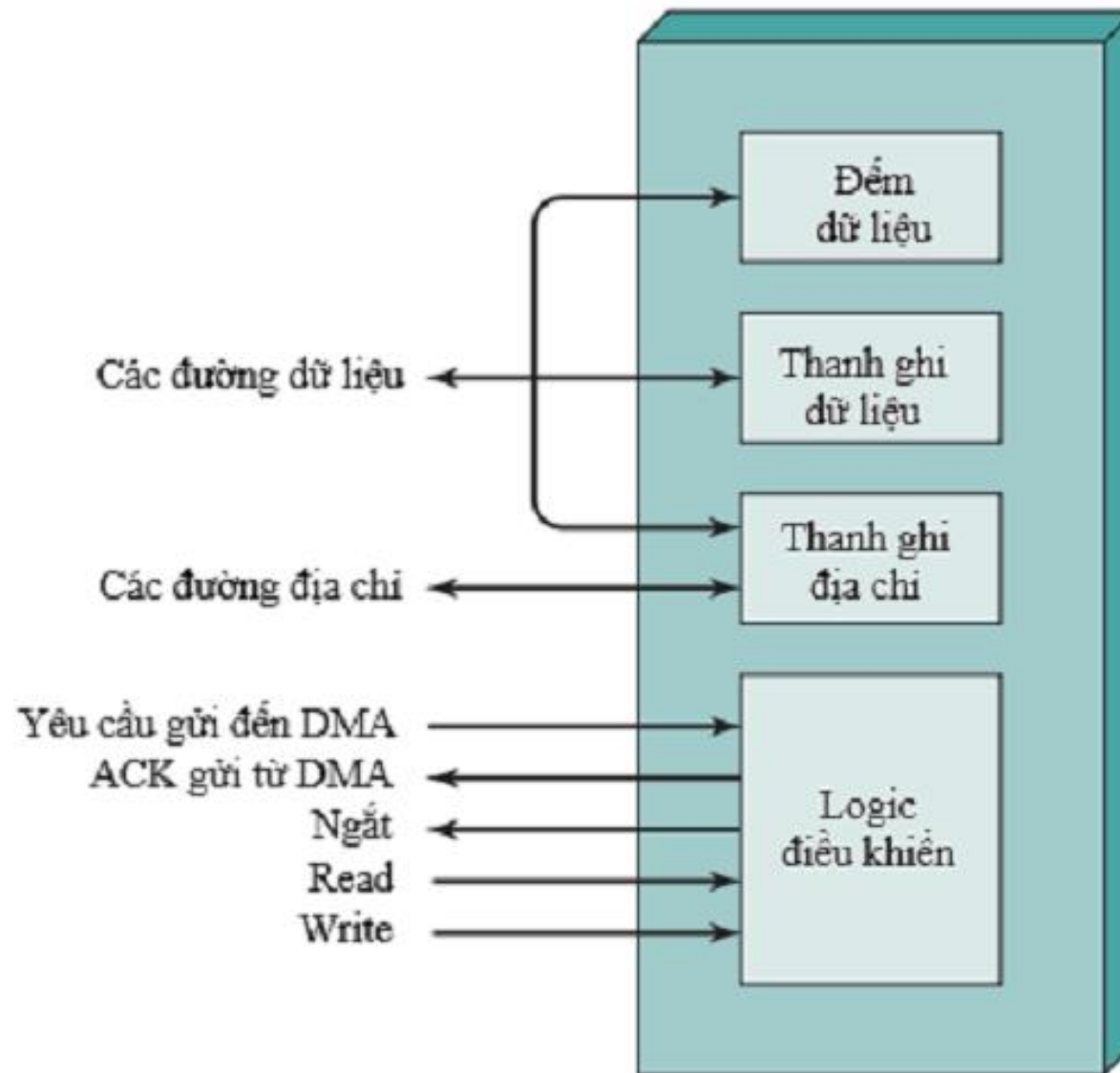
Chức năng DMA

- DMA bao gồm một module bổ sung trên hệ thống bus.
- Module DMA có khả năng thực hiện việc điều khiển việc truyền/nhận dữ liệu thay cho VXL
- Module DMA chỉ sử dụng bus khi VXL không cần đến nó hoặc buộc VXL phải tạm ngừng hoạt động để chiếm bus. Kỹ thuật thứ hai là phổ biến hơn.

Quá trình đọc/ghi dữ liệu sử dụng DMA

- Khi cần đọc/ghi dữ liệu, VXL gửi đến module DMA các thông tin sau:
 - Yêu cầu đọc/ghi: đường điều khiển
 - Địa chỉ TBNV: đường dữ liệu
 - Vị trí bộ nhớ bắt đầu để lưu trữ dữ liệu: đường dữ liệu và lưu trữ trên thanh ghi địa chỉ
 - Số lượng word được đọc/ghi: đường dữ liệu, lưu trữ trên thanh ghi data count
 - VXL thực hiện công việc khác, DMA thực hiện việc truyền giữa BN và TBNV
 - Sau khi hoàn thành, DMA gửi tín hiệu ngắt cho VXL
- VXL chỉ tham gia vào thời điểm bắt đầu và kết thúc việc truyền tin

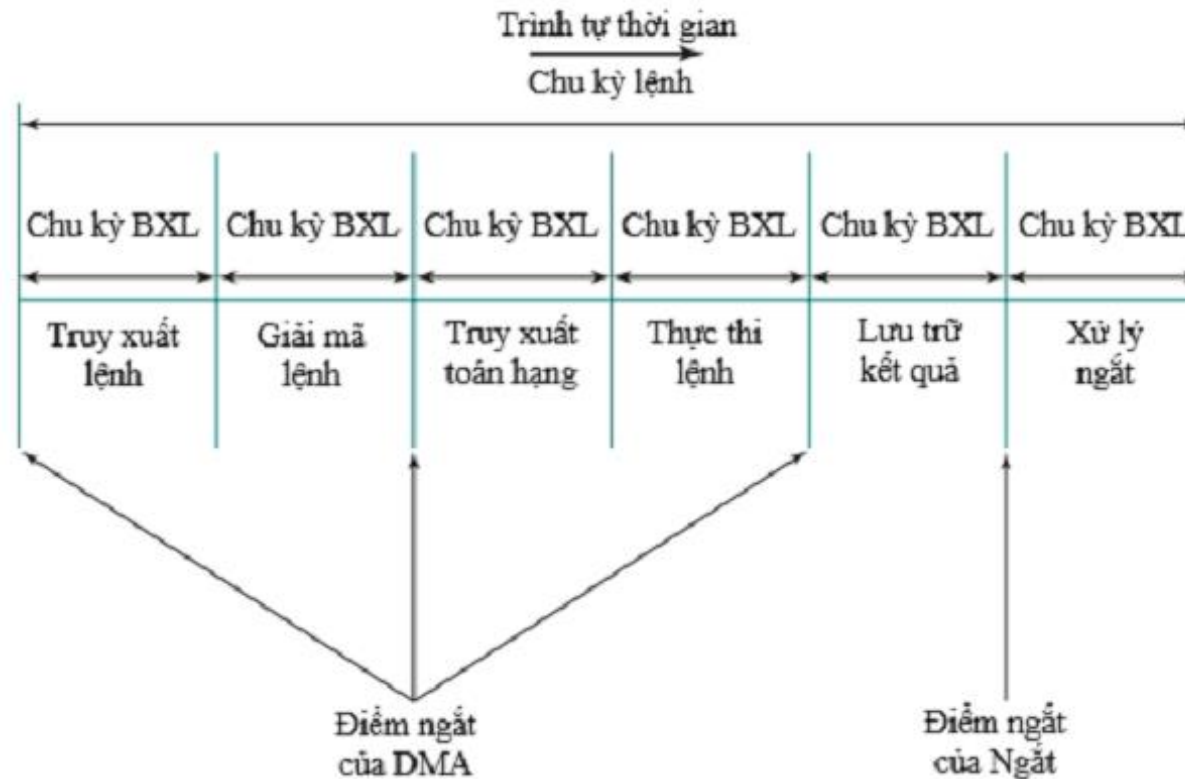
Sơ đồ module DMA điển hình



Hình 7.14 Sơ đồ khối DMA

Chu kỳ DMA

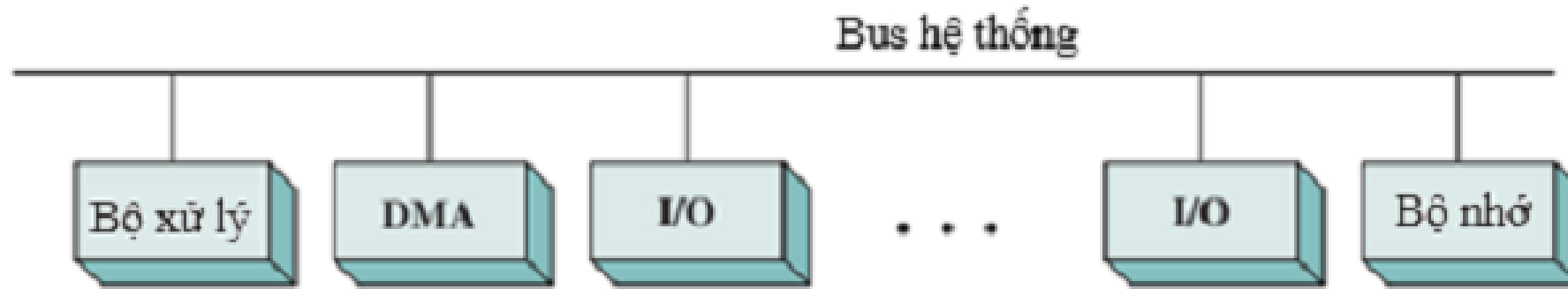
- Khi có yêu cầu bus từ DMA, VXL phải tạm thời “treo” để nhường bus cho DMA.
- Do không phải là ngắt nên VXL chỉ tạm dừng trong một chu kỳ bus
- DMA cũng làm cho VXL bị chậm hơn, tuy nhiên, khi truyền một lượng lớn dữ liệu thì DMA hiệu quả hơn nhiều so với 2 kỹ thuật còn lại



DMA

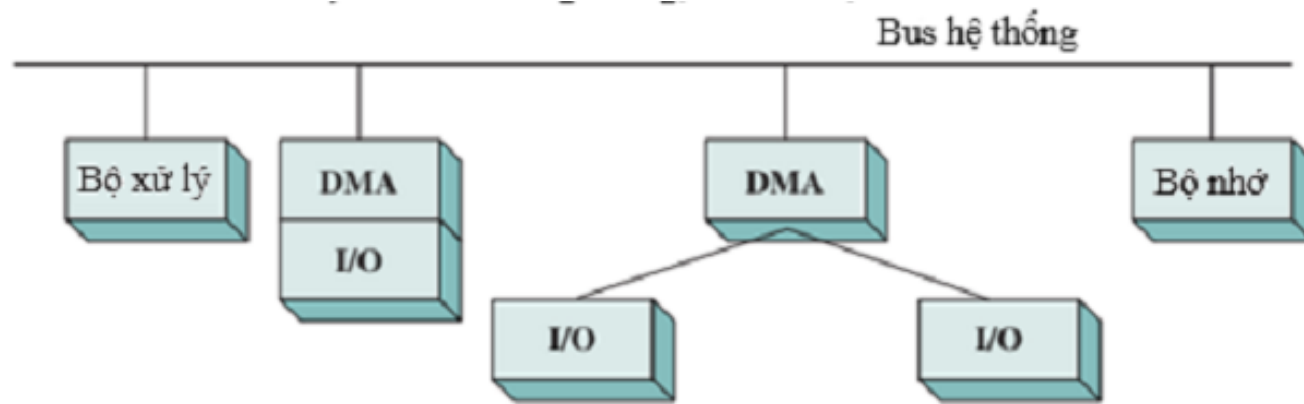
Hình 7.15 Điểm ngắt của DMA và của Ngắt trong chu kỳ lệnh

Cấu hình DMA

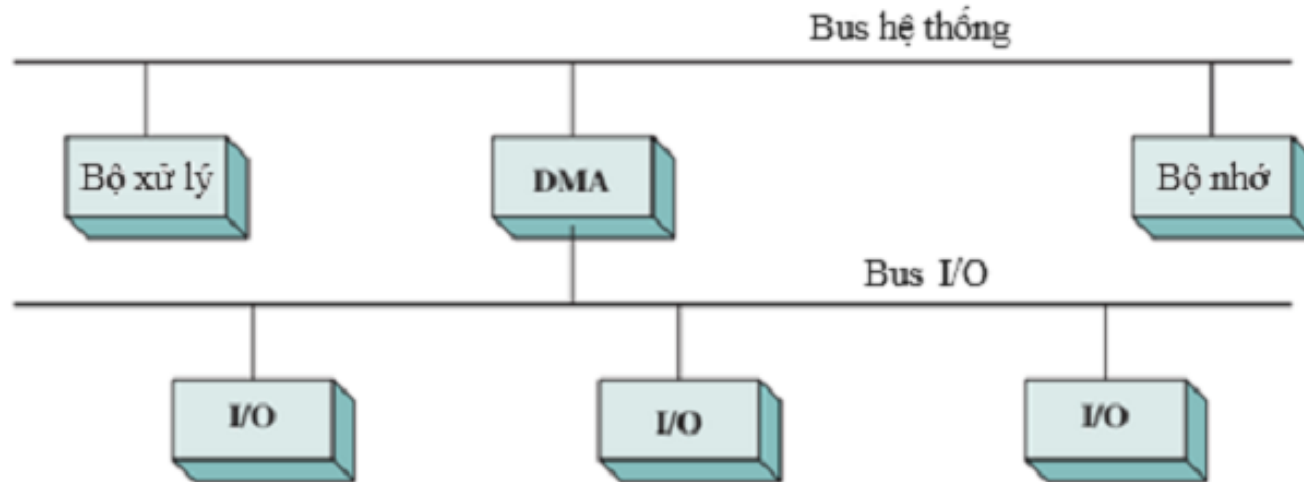


a) Cấu hình dùng chung, chia sẻ một bus

- Cấu hình DMA: tất cả các module chia sẻ chung bus hệ thống
- Module DMA điều khiển việc truyền dữ liệu giữa I/O và memory trong hai chu kỳ bus
- Cấu hình có chi phí thấp nhưng rõ ràng không hiệu quả



b) Tích hợp DMA và I/O



c) Cấu hình sử dụng thêm bus I/O

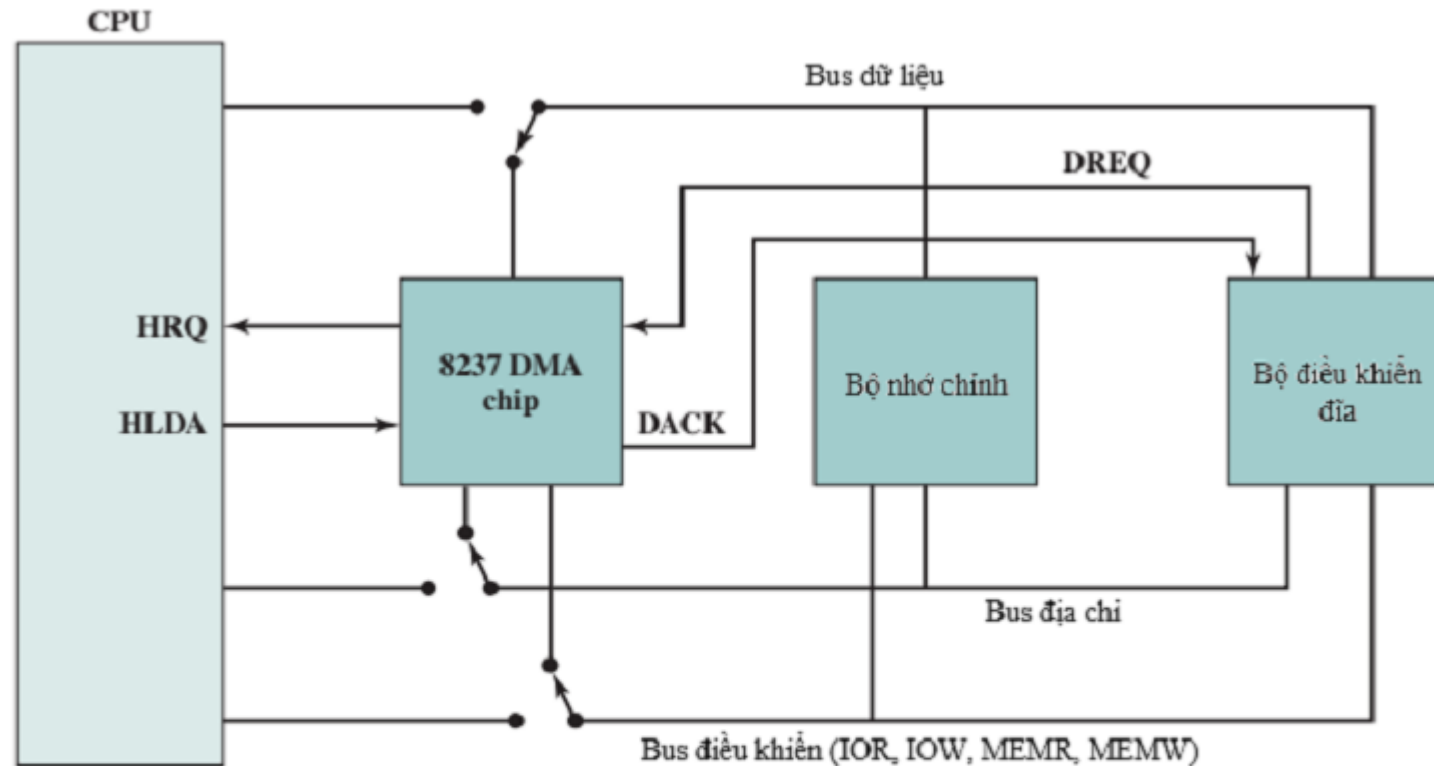
- DMA giao tiếp với I/O không qua bus hệ thống: giảm được chu kỳ bus
- DMA chỉ chiếm bus hệ thống khi cần trao đổi dữ liệu với bộ nhớ chính

Bộ điều khiển DMA Intel 8237

- Giao tiếp với họ 80x86 và DRAM
- Khi module DMA cần bus nó sẽ gửi tín hiệu HOLD tới bộ vi xử lý
- CPU phản hồi HLDA (hold acknowledge) - Mô-đun DMA có thể sử dụng bus
- Ví dụ: truyền dữ liệu từ bộ nhớ tới đĩa
 1. Thiết bị yêu cầu DMA bằng cách nâng DREQ (yêu cầu DMA)
 2. DMA đặt HRQ (yêu cầu giữ) lên cao
 3. CPU kết thúc chu kỳ bus hiện tại và đặt HDLA lên cao. HOLD duy trì trong suốt thời gian DMA
 4. DMA kích hoạt DACK (DMA ack), bảo thiết bị bắt đầu truyền
 5. DMA bắt đầu truyền bằng cách đặt byte đầu tiên của địa chỉ lên bus địa chỉ và kích hoạt MEMR; sau đó kích hoạt IOW để ghi vào ngoại vi. DMA giảm bộ đếm và tăng con trỏ địa chỉ. Lặp lại cho đến khi đếm về 0
 6. DMA hủy HRQ, trả bus trở lại CPU

8237 DMA

Cách sử dụng Bus Hệ thống



DACK = DMA acknowledge = Chấp nhận DMA

DREQ = DMA request = Yêu cầu DMA

HLDA = HOLD acknowledge = Chấp nhận HOLD

HRQ = HOLD request = Yêu cầu HOLD

Hình 7.17 DMA 8237

Bộ điều khiển DMA Fly-By

- Khi DMA sử dụng bus, bộ xử lý nghỉ. Khi bộ xử lý sử dụng bus, DMA nghỉ. DMA 8237 được gọi là bộ điều khiển DMA fly-by
- Dữ liệu không đi qua, không lưu trữ trong chip DMA
 - DMA giữa cổng I/O và bộ nhớ
 - Không đặt giữa hai cổng I/O hoặc hai vị trí bộ nhớ
- Thực hiện giao tiếp bộ nhớ - bộ nhớ qua thanh ghi
- 8237 có bốn kênh DMA
 - Lập trình độc lập
 - Kênh nào cũng có thể hoạt động tại 1 thời điểm
 - Được đánh số 0, 1, 2, và 3

Chương 7. Thiết bị ngoại vi

7.1 Các thiết bị ngoại vi

7.2 Module vào/ra

7.3 Các kỹ thuật I/O

- a. I/O chương trình
- b. I/O điều khiển ngắt
- c. Truy xuất bộ nhớ trực tiếp

7.4 Các bộ xử lý và kênh vào/ra

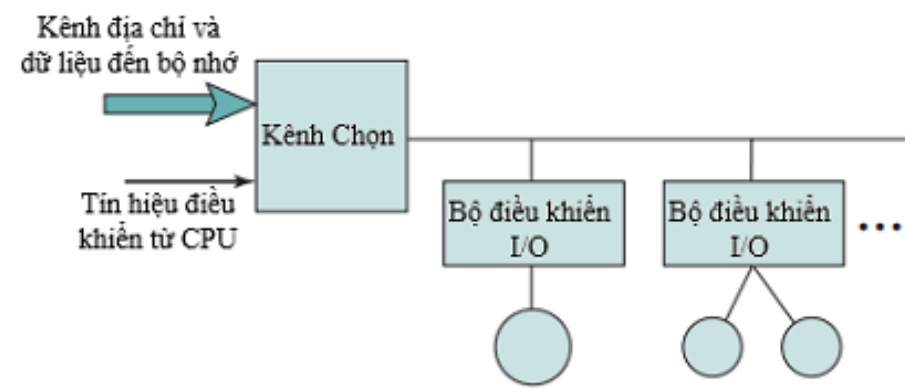
7.5 Giao diện ngoài

7.4 Các bộ xử lý và kênh vào/ra

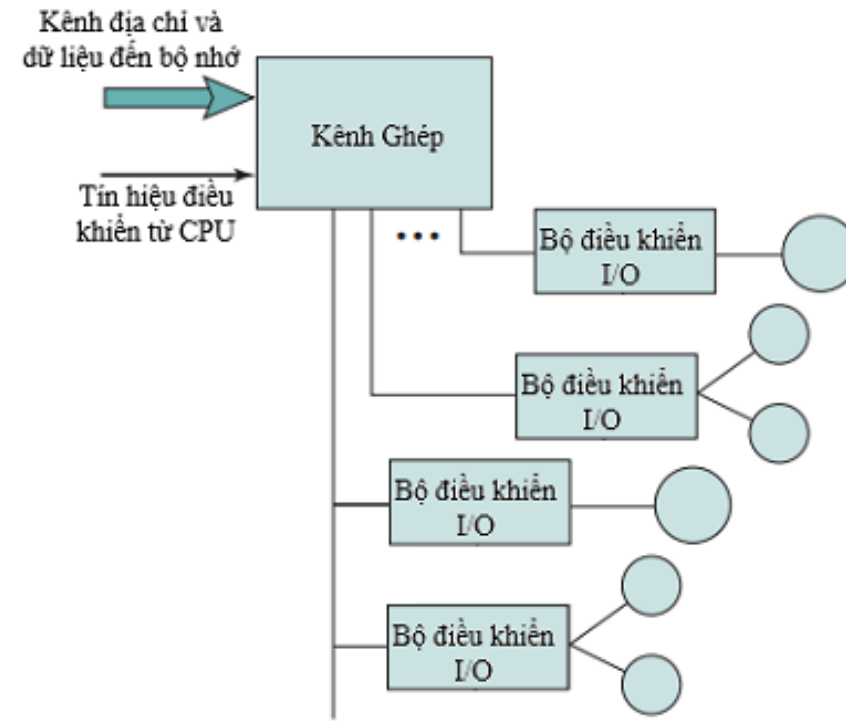
a. Sự phát triển của chức năng I/O qua các thời kỳ

1. CPU trực tiếp điều khiển một thiết bị ngoại vi.
2. Thêm vào Một bộ điều khiển hoặc module I/O. CPU sử dụng I/O lập trình, không có ngắt.
3. Cấu hình tương tự như bước 2, nhưng có sử dụng ngắt. CPU không phải tốn nhiều thời gian chờ đợi một hoạt động I/O được thực hiện, do đó tăng hiệu quả.
4. Module I/O được truy cập trực tiếp tới bộ nhớ qua DMA. Nó có thể di chuyển một khối dữ liệu đến/từ bộ nhớ mà không liên quan đến CPU, ngoại trừ khi bắt đầu và kết thúc quá trình truyền.
5. Module I/O được tăng cường để trở thành một bộ xử lý theo quyền riêng của nó, với một tập hợp chỉ lệnh dành riêng cho I/O
6. Module I/O có bộ nhớ cục bộ riêng và trên thực tế là một máy tính theo quyền riêng của nó. Với kiến trúc này, có thể kiểm soát một tập hợp lớn các thiết bị I/O với sự tham gia tối thiểu của CPU.

Kiến trúc I/O channel



a. Kênh chọn



b. Kênh ghép

Hình 7.18 Kiến trúc Kênh I/O

Chương 7. Thiết bị ngoại vi

7.1 Các thiết bị ngoại vi

7.2 Module vào/ra

7.3 Các kỹ thuật I/O

- a. I/O chương trình
- b. I/O điều khiển ngắt
- c. Truy xuất bộ nhớ trực tiếp

7.4 Các bộ xử lý và kênh vào/ra

7.5 Giao diện ngoài

I/O song song và nối tiếp

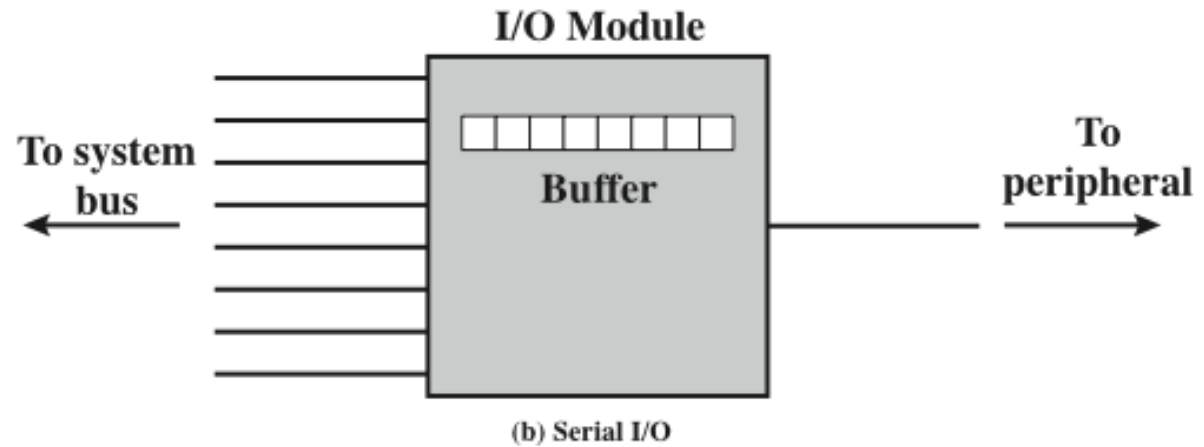
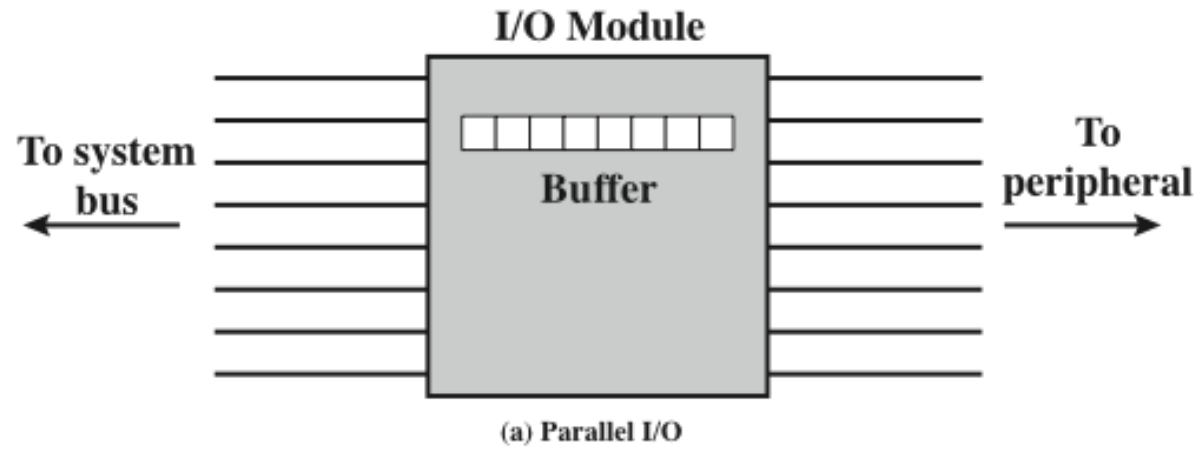


Figure 7.16 Parallel and Serial I/O

Cấu hình Điểm-Điểm (Point-to-Point) và Đa điểm (Multipoint)

Kết nối giữa 1 mô-đun I/O trong 1 hệ thống máy tính và các thiết bị ngoại vi có thể là:

point-to-point

multipoint

Giao diện point-to-point cung cấp 1 liên kết riêng giữa mô-đun I/O và thiết bị ngoại vi

Ở các hệ thống nhỏ (PC, workstation) các liên kết điểm-điểm điển hình bao gồm các liên kết tới bàn phím, máy in, và modem ngoài

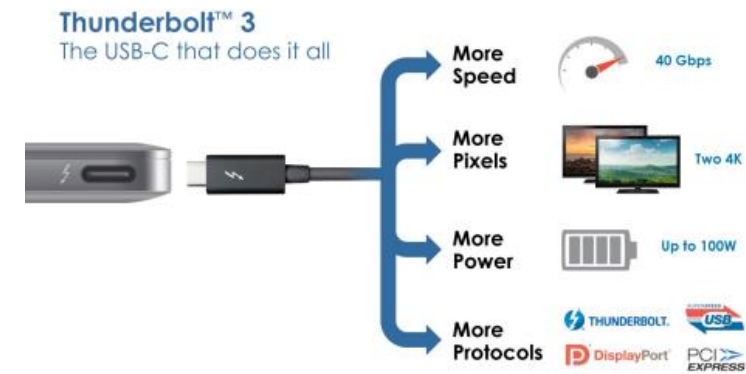
Ví dụ: EIA-232

Các giao diện multipoint bên ngoài hỗ trợ các thiết bị lưu trữ ngoài (ổ đĩa và băng) và các thiết bị đa phương tiện (CD-ROM, video, âm thanh)

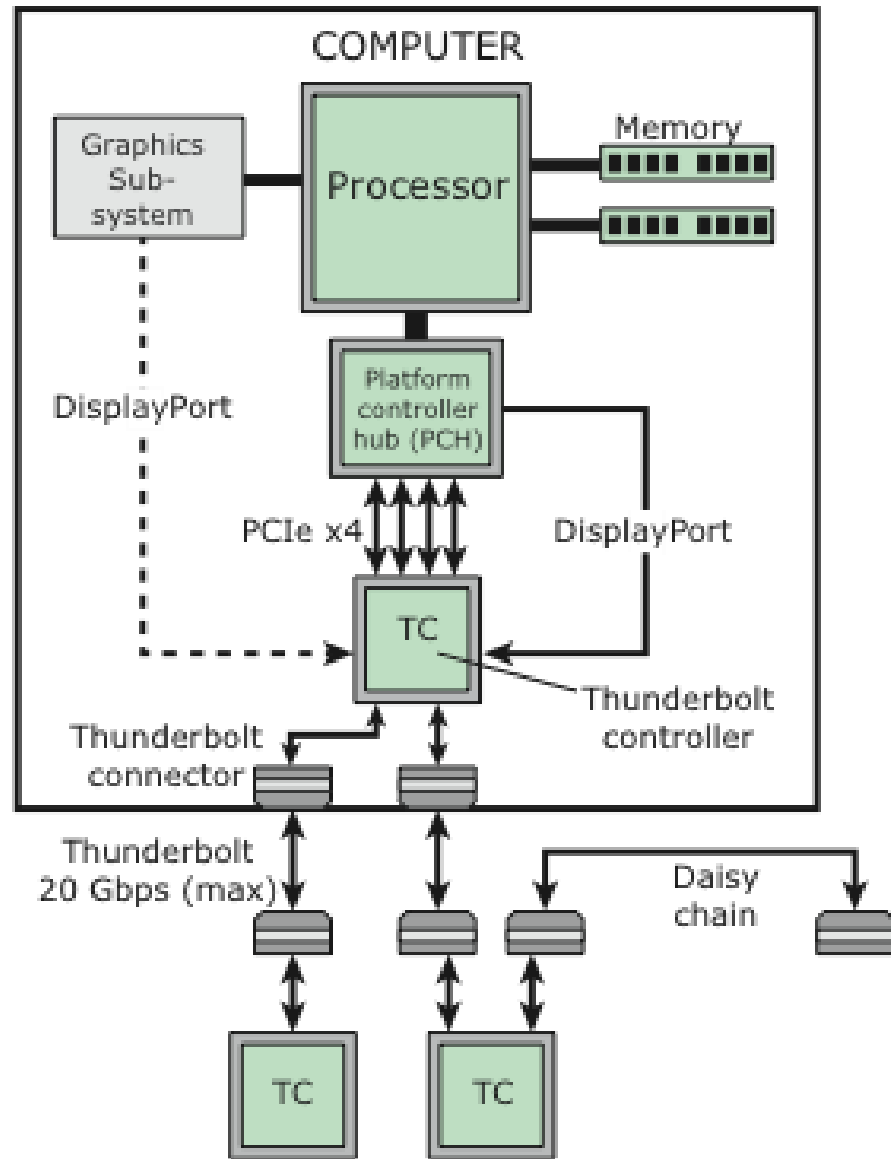
Trong các bus ngoại vi

Thunderbolt

- Công nghệ kết nối ngoại vi nhanh nhất và mới nhất sẵn sàng cho mục đích sử dụng chung
- Phát triển bởi Intel với sự hợp tác của Apple
- Kết hợp dữ liệu, video, âm thanh và năng lượng vào một kết nối tốc độ cao cho nhiều thiết bị ngoại vi như ổ cứng, mảng RAID, các hộp thu video và giao diện mạng
- Cung cấp thông lượng 10 Gbps theo từng hướng, công suất tối đa 10 W cho các thiết bị ngoại vi đã kết nối



- 1 giao diện ngoại vi tương thích Thunderbolt thì phức tạp hơn nhiều so với 1 thiết bị USB đơn giản
- Các sản phẩm thế hệ đầu tiên chủ yếu nhằm vào thị trường tiêu dùng chuyên nghiệp, VD như biên tập viên nghe nhìn cần di chuyển nhanh khối lượng lớn dữ liệu giữa các thiết bị lưu trữ và máy tính xách tay
- Thunderbolt là một tính năng chuẩn của máy tính xách tay Apple MacBook Pro và máy tính bàn iMac

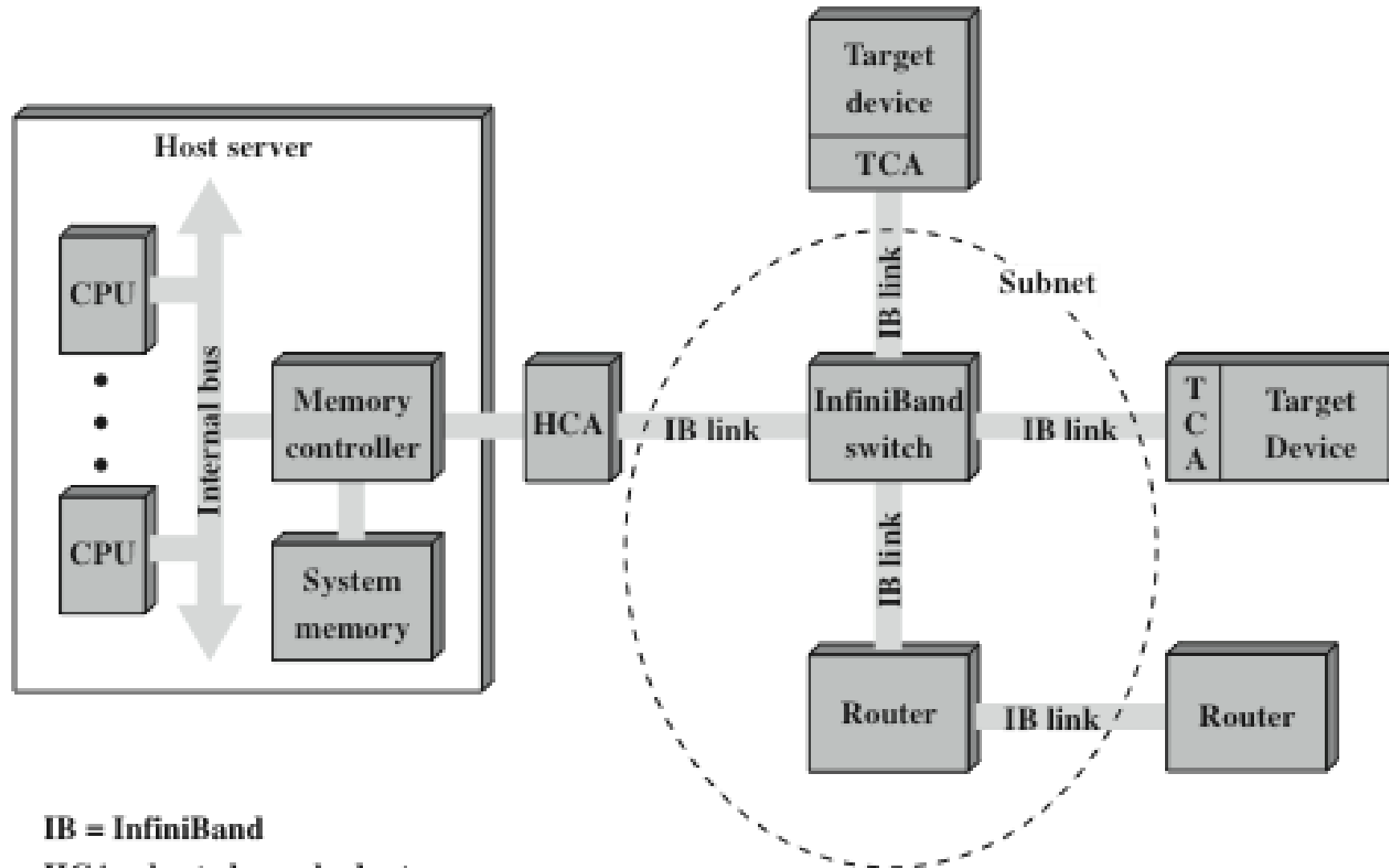


Cấu hình máy tính với Thunderbolt

InfiniBand

- Công nghệ I/O gần đây nhằm vào thị trường máy chủ cao cấp
- Phiên bản đầu tiên được phát hành vào đầu năm 2001
- Chuẩn này mô tả 1 kiến trúc và thông số kỹ thuật cho luồng dữ liệu giữa các bộ xử lý và thiết bị I/O thông minh
- Trở thành giao diện phổ biến cho các khu vực lưu trữ mạng và các cấu hình lưu trữ lớn khác
- Cho phép các máy chủ, bộ lưu trữ từ xa và các thiết bị mạng khác được gắn trên một lưới tập trung các switch và link
- Kiến trúc switch có thể kết nối tới 64.000 máy chủ, hệ thống lưu trữ và thiết bị mạng

Lưới Switch InfiniBand



IB = InfiniBand

HCA = host channel adapter

TCA = target channel adapter

Hoạt động của InfiniBand

- Mỗi liên kết vật lý giữa một switch và một giao diện đính kèm có thể hỗ trợ tối đa 16 kênh logic, được gọi là làn đường ảo (virtual lane)
 - Một đường dành cho việc quản lý lưới và các đường khác để vận chuyển dữ liệu
- Một làn ảo tạm thời được dành riêng cho việc truyền dữ liệu từ nút này sang nút khác trên lưới InfiniBand
- Switch InfiniBand ánh xạ lưu lượng từ làn đường đến sang làn đường đi để định tuyến dữ liệu giữa các điểm đến mong muốn
- Sử dụng kiến trúc giao thức phân lớp, gồm 4 lớp:
 - Physical - Vật lý
 - Link - Liên kết
 - Network - Mạng
 - Transport - Vận chuyển

Bảng 7.3

Liên kết InfiniBand và Tốc độ truyền dữ liệu

| Link | Signal rate (unidirectional) | Usable capacity (80% of signal rate) | Effective data throughput (send + receive) |
|---------|---------------------------------|---|---|
| 1- wide | 2.5 Gbps | 2 Gbps (250 MBps) | (250 + 250) MBps |
| 4-wide | 10 Gbps | 8 Gbps (1 GBps) | (1 + 1) GBps |
| 12-wide | 30 Gbps | 24 Gbps (3 GBps) | (3 + 3) Gbps |

Tổng kết

Chương 7

- Thiết bị ngoại vi
 - Bàn phím/ màn hình
 - Ổ đĩa
- Module vào/ra
 - Chức năng
 - Cấu trúc
- I/O chương trình
 - Tổng quan về I/O lập trình
 - Lệnh I/O
 - Chỉ thị I/O
- I/O điều khiển ngắt
 - Xử lý ngắt
 - Các vấn đề thiết kế
 - Bộ điều khiển ngắt Intel 82C59A
 - Giao diện ngoại vi lập trình được Intel 82C55A

Input/Output

- Truy cập bộ nhớ trực tiếp
 - Nhược điểm của I/O lập trình và I/O định hướng gián đoạn
 - Chức năng DMA
 - Bộ điều khiển Intel 8237A DMA
- Các bộ xử lý và kênh vào/ra
 - Sự phát triển của chức năng I/O
 - Đặc điểm của các kênh I/O
- Giao diện ngoài
 - Các loại giao diện
 - Cấu hình điểm-điểm và đa điểm
 - Thunderbolt
 - InfiniBand

Câu hỏi ôn tập

1. Chức năng của thiết bị ngoại vi là gì?
2. Liệt kê ba loại thiết bị ngoại vi.
3. Các chức năng chính của module vào/ra là gì?
4. Liệt kê và nêu ngắn gọn ba kỹ thuật để thực hiện vào/ra.
5. Sự khác biệt giữa I/O ánh xạ bộ nhớ và I/O riêng biệt là gì?
6. Khi xảy ra ngắt thiết bị, bộ xử lý xác định thiết bị đưa ra yêu cầu ngắt như thế nào?

Hình ảnh và nội dung trong bài giảng này tham khảo từ cuốn sách và slide bài giảng “Computer Organization and Architecture”, 10th Edition, của tác giả William Stallings.