

Kiến trúc máy tính

+ Chương 7

Thiết bị ngoại vi



+ Chương 7. Thiết bị ngoại vi

7.1 Các thiết bị ngoại vi

7.2 Module vào/ra

7.3 Vào/ra chương trình

7.4 Điều khiển ngắt vào/ra

7.5 Truy xuất bộ nhớ trực tiếp

7.6 Các bộ xử lý và kênh vào/ra

+ 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 I/O)
- Giúp trao đổi dữ liệu với bên ngoài
- Kết nối với máy tính qua module vào/ra (module I/O)
 - 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: các thiết bị ngoại vi dành cho người dùng máy tính như màn hình, máy in,...
 - Máy đọc được: các thiết bị gắn vào máy tính thực hiện một số chức năng giao tiếp (lấy thông tin,...): ổ cứng, cảm biến, băng từ,...
 - Truyền thông: sử dụng truyền thông tin đi xa: modem, card mạng,...

+

Mô hình tổng quát của I/O Module

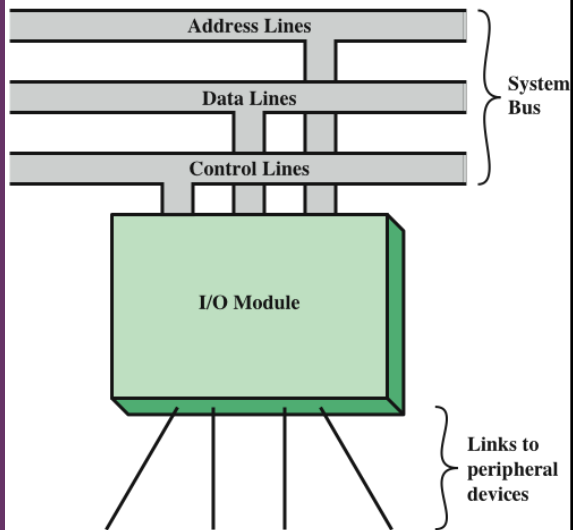


Figure 7.1 Generic Model of an I/O Module

+

a. Sơ đồ khối thiết bị ngoài

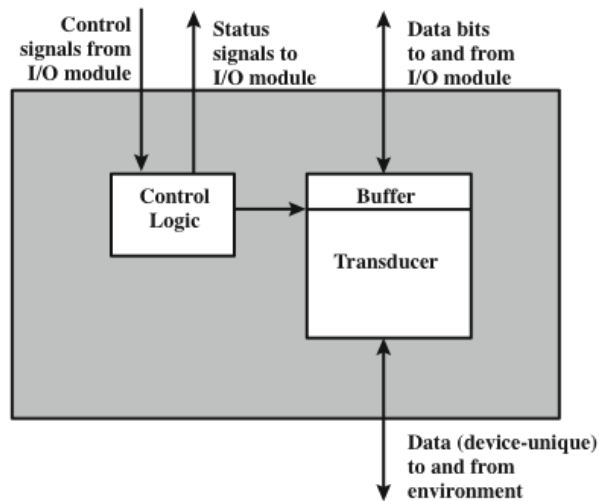


Figure 7.2 Block Diagram of an External Device

+ a. Sơ đồ khối thiết bị ngoài (tiếp)

■ Giao diện (interface) với module I/O:

- *Tín hiệu điều khiển - Control signal*: xác định chức năng mà thiết bị sẽ thực hiện. Ví dụ: gửi dữ liệu đến module I/O (INPUT hoặc READ), nhận dữ liệu từ module I/O (OUTPUT hoặc WRITE), thông tin trạng thái, hoặc thực hiện một số chức năng điều khiển đặc biệt đến thiết bị (ví dụ, đặt head của đĩa cứng lên một vị trí nào đó)
- *Dữ liệu - Data*: một tập các bit được gửi đi hoặc nhận về từ module I/O.
- *Tín hiệu trạng thái - Status signal*: chỉ ra trạng thái của thiết bị. Ví dụ: READY/NOT-READY cho biết rằng khi nào thiết bị sẵn sàng cho việc truyền dữ liệu.

■ Logic điều khiển – Control logic: nhận các tín hiệu điều khiển từ module I/O, điều khiển hoạt động của thiết bị.

■ Bộ chuyển đổi - Transducer: chuyển đổi dữ liệu (đang ở dạng t/h điện) sang các dạng khác (vd: điểm ảnh trên màn hình,...) và ngược lại. Ngoài ra, **bộ đệm (buffer)** để lưu trữ tạm dữ liệu đang được chuyển giao giữa module I/O 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 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
 - VD: 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 biến đổi trong bàn phím và dịch sang mẫu bit của mã IRA tương ứng
- Mẫu bit này được truyền đến mô-đun I/O trong máy tính
- Trên đầu ra, các ký tự mã IRA được truyền đến một thiết bị ngoại vi từ module I/O
- Bộ biến 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

7.2 Module I/O

a. Chức năng Module

Các chức năng chính của một module I/O 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

Ví dụ: chức năng điều khiển truyền dữ liệu giữa VXL và thiết bị I/O:

1. Bộ xử lý yêu cầu module I/O để kiểm tra trạng thái của một thiết bị.
2. Module I/O trả về trạng thái thiết bị.
3. Nếu thiết bị đang hoạt động và sẵn sàng truyền, bộ xử lý yêu cầu truyền dữ liệu, bằng lệnh cho module I/O.
4. Module I/O nhận được một đơn vị dữ liệu (ví dụ, 8 hoặc 16 bit) từ thiết bị bên ngoài.
5. Dữ liệu được chuyển từ module I/O sang bộ xử lý.

+ Chức năng module I/O (tiếp)

- Truyền thông tin với bộ vi xử lý: gồm giải mã lệnh, dữ liệu, báo cáo trạng thái (trạng thái của thiết bị I/O có sẵn sàng hay không), nhận dạng địa chỉ (địa chỉ các port mà TBNV được nối vào)
- Truyền 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 lỗi: phát hiện và báo cáo lỗi truyền
 - Ví dụ: kẹt giấy (máy in), đĩa hỏng (disk driver) hoặc phát hiện lỗi trong quá trình truyền dẫn ký tự IRA: sử dụng bit thứ 8 làm bit parity

b. Cấu trúc Module I/O

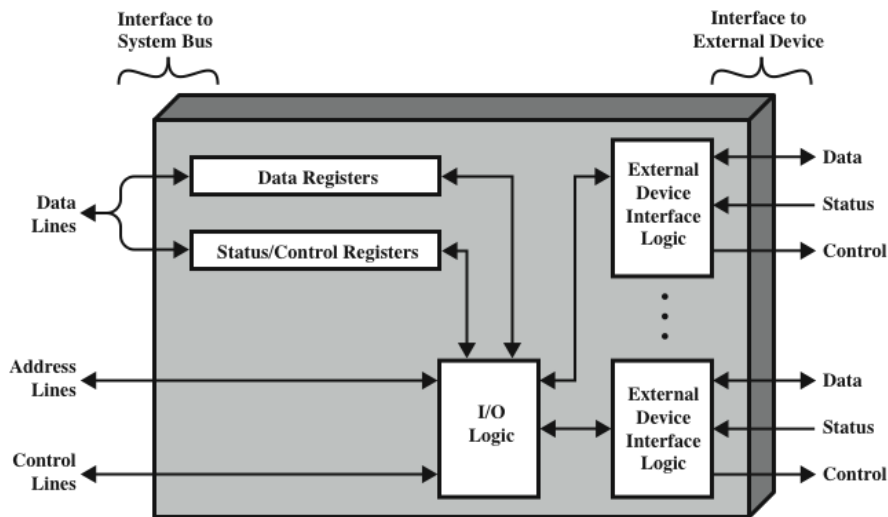


Figure 7.3 Block Diagram of an I/O Module

+ b. Cấu trúc Module I/O (tiếp)

Các module I/O 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 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
- **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 I/O. I/O module 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 kết nối đến các thiết bị được nối đến nó. Mỗi module I/O có một (nếu chỉ nối với một TBNV) hoặc một tập địa chỉ (nếu module nối với nhiều TBNV)
- Module I/O chứa các **logic giao tiếp với thiết bị ngoại vi (External Device Interface Logic)** giao tiếp với thiết bị mà nó điều khiển.

+ 7.3. I/O chương trình

Hoạt động của module I/O theo 3 kỹ thuật sau:

- I/O chương trình (phần 7.3)
 - Dữ liệu được trao đổi giữa bộ vi xử lý và mô đun I/O
 - Bộ vi xử lý thi hành một chương trình cho phép nó trực tiếp điều hành các hoạt động I/O
 - Khi bộ xử lý ra lệnh, nó phải đợi cho đến khi hoạt động I/O hoàn tất
 - Bộ xử lý chạy nhanh hơn module I/O sẽ gây lãng phí thời gian xử lý
- I/O điều khiển ngắt (phần 7.4)
 - Bộ vi xử lý ra lệnh I/O sau đó tiếp tục thi hành các lệnh tiếp theo trong chương trình.
 - Khi module I/O hoàn thành công việc, nó sẽ gửi tín hiệu yêu cầu ngắt đến VXL.
- Truy cập bộ nhớ trực tiếp (DMA) (phần 7.5)
 - Module I/O 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

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

+ 7.3. I/O chương trình

- Khi VXL thực thi một chương trình gồm có hoạt động I/O, nó sẽ gửi lệnh đến module I/O tương ứng
- Module I/O nhận yêu cầu, thiết lập các bit trạng thái trên thanh ghi trạng thái
- Module I/O không gửi thông báo cho VXL.
- VXL phải định kỳ kiểm tra trạng thái của module I/O đến khi hoạt động hoàn tất

+ a. Các lệnh I/O – I/O command (từ VXL đến module I/O)

Để thực thi một hoạt động I/O, VXL đưa ra **địa chỉ**: định ra module I/O và TBNV cụ thể; và **các lệnh I/O**. Có bốn loại lệnh I/O các module cần thực hiện

1) Control

- Dùng để kích hoạt một thiết bị ngoại vi và chỉ định nó phải làm gì

2) Test

- Dùng để 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

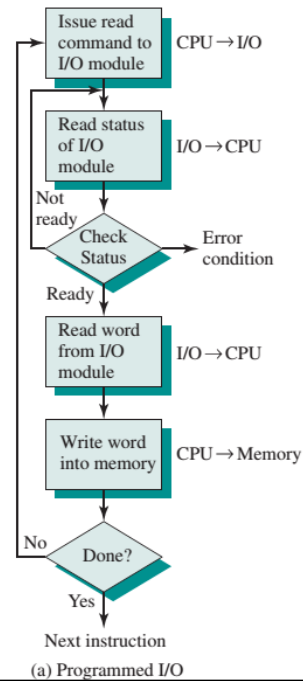
- Cho phép module I/O lấy dữ liệu từ thiết bị ngoại vi và đặt nó vào bộ đệm bên trong

4) Write

- Cho phép 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)



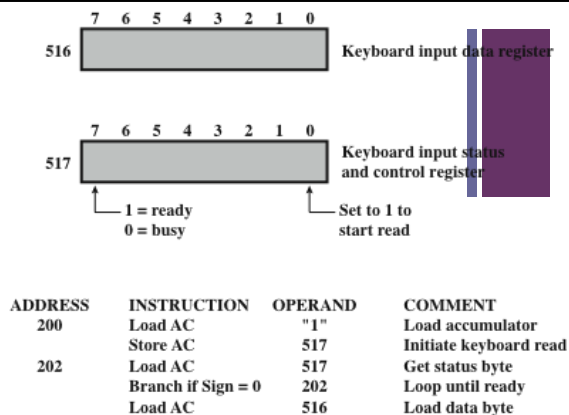
Các chỉ lệnh I/O – I/O instruction (Thực thi bởi vi xử lý)

- Với I/O chương trình, có sự tương ứng chặt chẽ giữa các **chỉ lệnh** liên quan đến I/O mà VXL nạp từ bộ nhớ và các **lệnh I/O** mà bộ xử lý đưa ra cho một module I/O để thi hành
 - Định dạng của chỉ lệnh phụ thuộc vào cách mà các TBNV được định ra
- Mỗi TBNV kết nối qua các module I/O được cấp một mã nhận dạng hoặc địa chỉ duy nhất
 - Khi bộ vi xử lý ra lệnh I/O, lệnh có chứa địa chỉ của thiết bị mong muốn
 - Do đó mỗi module I/O phải dịch các dòng địa chỉ để xem lệnh có phải cho nó hay không

+ Chế độ đánh địa chỉ

- Khi VXL, bộ nhớ và các TBNV chia sẻ hệ thống bus, có hai chế độ đánh địa chỉ thiết bị ngoại vi:
 - **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 chỉ 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 BN hay địa chỉ TBNV
 - Toàn bộ dải địa chỉ dùng cho cả hai. VD: 10 đường địa chỉ cho phép đánh địa chỉ 1024 ô nhớ và 1024 TBNV
 - Tập các chỉ lệnh đến BN và TBNV khác nhau

- + Ví dụ: đọc dữ liệu từ bàn phím với hai trường hợp:
- I/O ánh xạ bộ nhớ
 - I/O riêng biệt



(a) Memory-mapped I/O

ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load I/O	5	Initiate keyboard read
201	Test I/O	5	Check for completion
	Branch Not Ready	201	Loop until complete
	In	5	Load data byte

(b) Isolated I/O

+ 7.4. I/O điều khiển ngắt

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

+ a. Cơ chế làm việc

- Từ phía module I/O:
 - Nhận lệnh READ từ VXL
 - Đọc dữ liệu vào từ TBNV tương ứng
 - Khi dữ liệu được đưa vào thanh ghi dữ liệu, module 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 đặt dữ liệu vào bus và sẵn sàng để thực hiện các hoạt động I/O khác
- Từ phía VXL:
 - VXL đưa ra lệnh READ.
 - Sau đó thực hiện các công việc khác (vd: trong trường hợp có nhiều CT 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

+ b. Quá trình xử lý ngắt

- Khi TBNV hoàn thành một hoạt động I/O, một chuỗi các sự kiện phần cứng xảy ra:
 - Thiết bị gửi yêu cầu ngắt đến VXL
 - Trước khi phục vụ ngắt, VXL hoàn thành nốt lệnh hiện tại
 - VXL kiểm tra yêu cầu ngắt, xác định và gửi tín hiệu ACK đến thiết bị. Sau khi nhận ACK, thiết bị loại bỏ tín hiệu ngắt
 - VXL chuẩn bị cho việc thực hiện ngắt: lưu trữ các thông tin của chương trình hiện tại để khôi phục lại công việc sau khi phục vụ ngắt. Thông tin tối thiểu cần có là: (a) trạng thái của VXL (nằm trong thanh ghi PSW) và (b) địa chỉ lệnh tiếp theo (nằm trong thanh ghi PC).
 - VXL đọc địa chỉ lệnh của chương trình xử lý ngắt vào thanh ghi PC.
 - VXL tiếp tục lưu trữ dữ liệu của các thanh ghi vì có thể các thanh ghi này cũng được sử dụng bởi chương trình xử lý ngắt
 - Xử lý ngắt: kiểm tra các thông tin gây ra ngắt, gửi ACK hoặc các lệnh khác
 - Kết thúc xử lý ngắt, dữ liệu đã được lưu trữ sẽ được khôi phục lại vào các thanh ghi → khôi phục chương trình đang thực hiện



Xử lý ngắt đơn giản

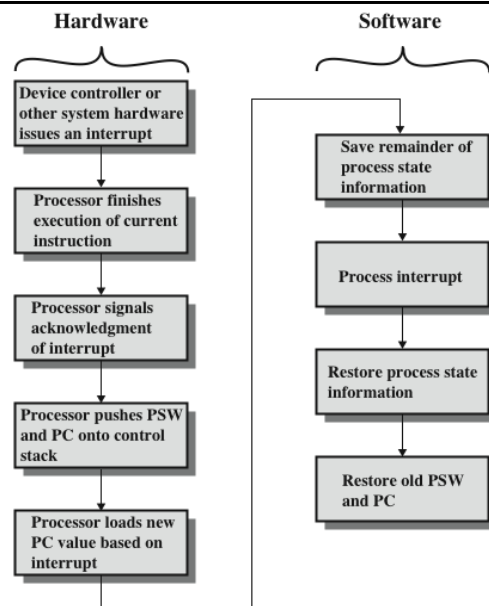


Figure 7.6 Simple Interrupt Processing



Thay đổi trong bộ
nhớ và thanh ghi khi
có một ngắt

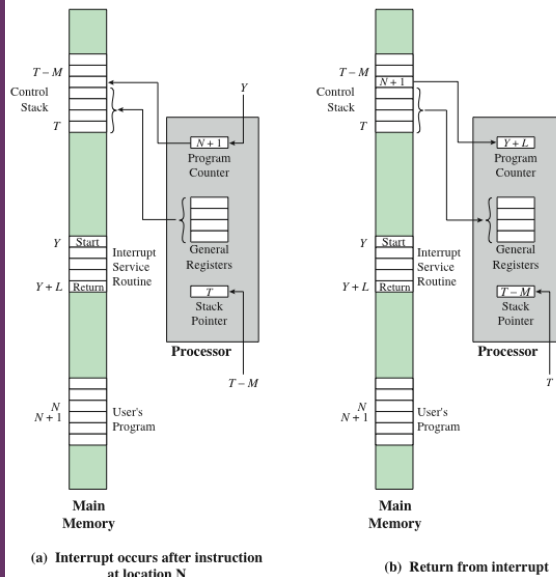


Figure 7.7 Changes in Memory and Registers for an Interrupt

c. Các vấn đề thiết kế

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 I/O, khi có một yêu cầu ngắt gửi tới, bộ vi xử lý sẽ xác định thiết bị nào thực hiện 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 I/O → 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 I/O 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 rẽ nhánh tới một trình phục vụ ngắt chuyên thăm dò từng module I/O để xác định module nào gây ra ngắt.
- Gửi tín hiệu thăm dò (poll) qua đường command và địa chỉ qua các đường đ/c → Module I/O sẽ trả lời nếu nó đưa ra y/c ngắt.
- Hoặc đọc thanh ghi trạng thái của từng module để phát hiện module y/c ngắt
- Nhược điểm: Tốn thời gian

+ 1. Nhận diện thiết bị (tiếp)

■ Chuỗi Daisy (thăm dò phần cứng, vector)

- Tất cả các module I/O sử dụng chung một đường yêu cầu ngắt
- Đường ACK được nối chuỗi qua các module
- Khi VXL nhận được y/c ngắt, nó sẽ gửi lại một ACK (tín hiệu thừa nhận ngắt).
- T/h này truyền qua các module I/O đến khi gặp module y/c ngắt. Module này trả lời bằng cách đặt một vector (địa chỉ của module I/O 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

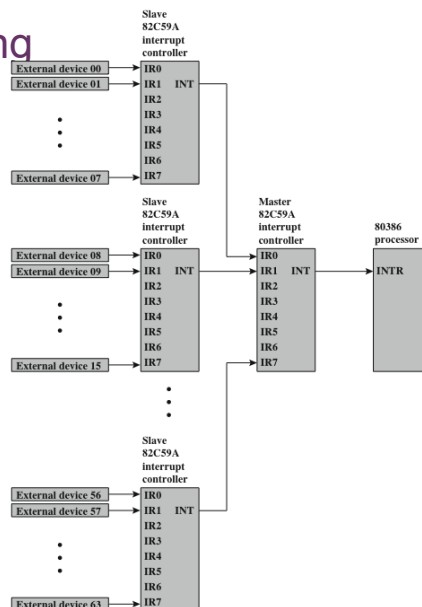
■ Phân xử bus (ngắt vector)

- Đầu tiên module I/O phải điều khiển được bus trước khi có thể đưa ra yêu cầu ngắt. Chỉ một module đưa ra yêu cầu ngắt tại một thời điểm
- Khi bộ xử lý phát hiện ra ngắt, nó trả lời trên đường ACK.
- Module đặt vector của nó lên các đường dữ liệu
- VXL sử dụng vector này trở tới trình phục vụ ngắt tương ứng của TB

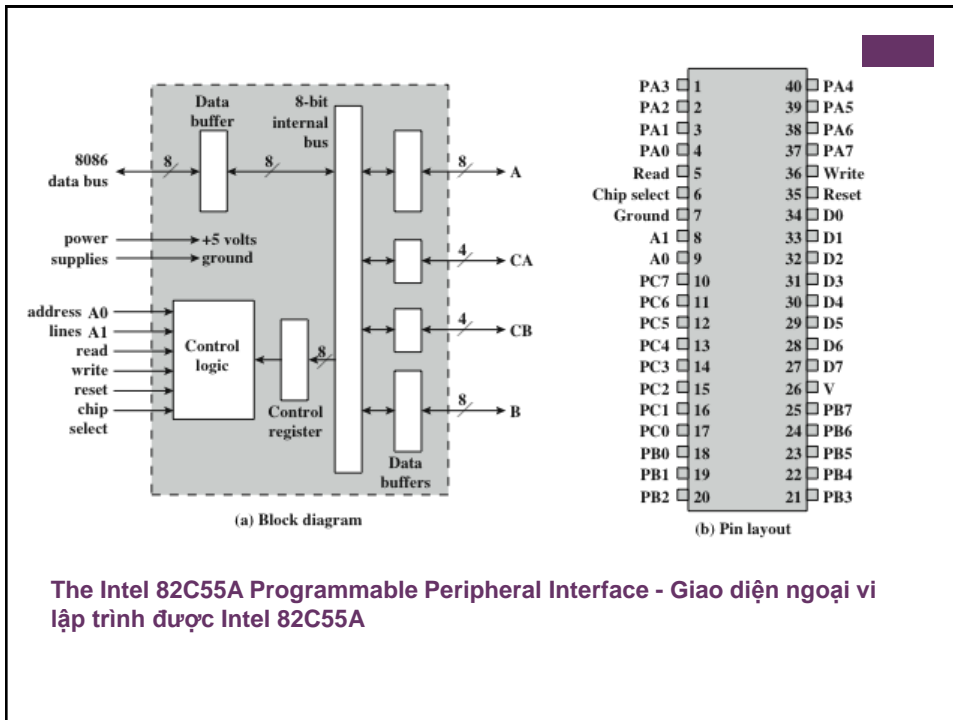
+ 2. Xác định ưu tiên

- Các phương pháp nhận diện thiết bị đồng thời cho phép xác định độ ưu tiên của các TB 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

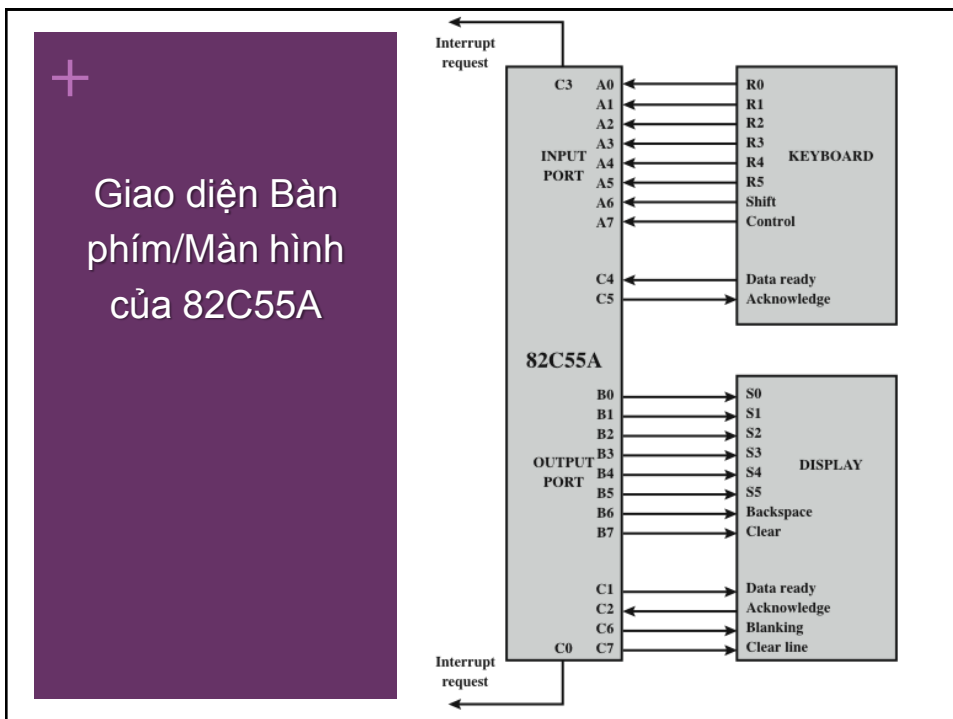
d. Cấu trúc phần cứng



Intel 82C59A Interrupt Controller - Bộ điều khiển ngắt Intel 82C59A



The Intel 82C55A Programmable Peripheral Interface - Giao diện ngoại vi lập trình được Intel 82C55A



+ 5. Truy cập bộ nhớ trực tiếp - DMA

- Nhược điểm của I/O chương trình và I/O điều khiển ngắt: VXL thay gia vào hầu hết chu trình truyền/nhận dữ liệu giữa TBNV và BN
 - 1) Tốc độ truyền I/O 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 I/O; Một số chỉ lệnh phải được thực hiện cho mỗi lần truyền I/O

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

+ a. Chức năng DMA

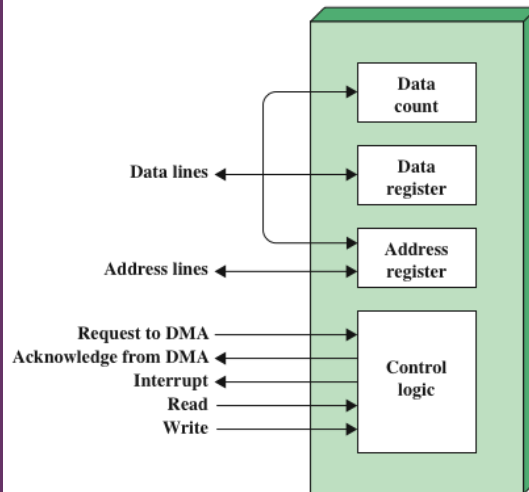
- DMA bao gồm một module bổ sung trên hệ thống bus.
- Module DMA có khả năng thực hiện việc điều khiển việc truyền/nhận dữ liệu thay cho VXL
- Module DMA chỉ sử dụng bus khi VXL không cần đến nó hoặc buộc VXL phải tạm ngừng hoạt động để chiếm bus. Kỹ thuật thứ hai là phổ biến hơn.

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

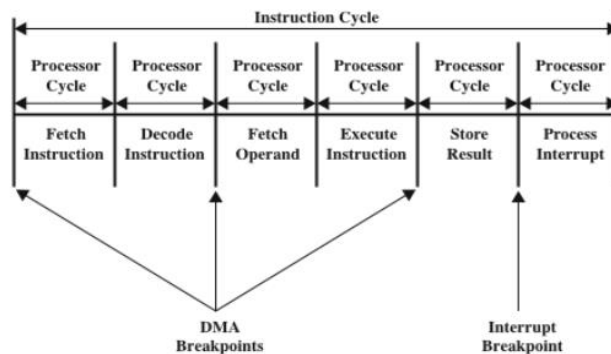


Sơ đồ module DMA điển hình

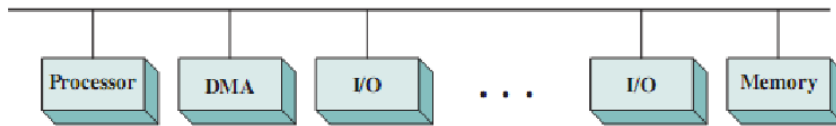


b. Chu kỳ DMA

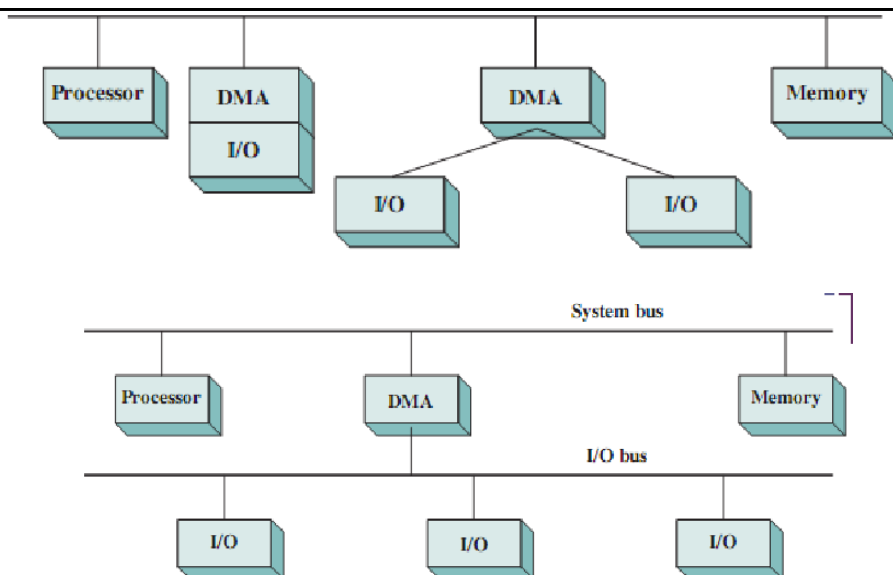
- Khi có yêu cầu bus từ DMA, VXL phải tạm thời treo để nhường bus cho DMA.
- Do không phải là ngắt nên VXL chỉ tạm dừng trong một chu kỳ bus
- DMA cũng làm cho VXL bị chậm hơn, tuy nhiên, khi truyền một lượng lớn dữ liệu thì DMA hiệu quả hơn nhiều so với 2 kỹ thuật còn lại



+ c. Cấu hình DMA



- 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

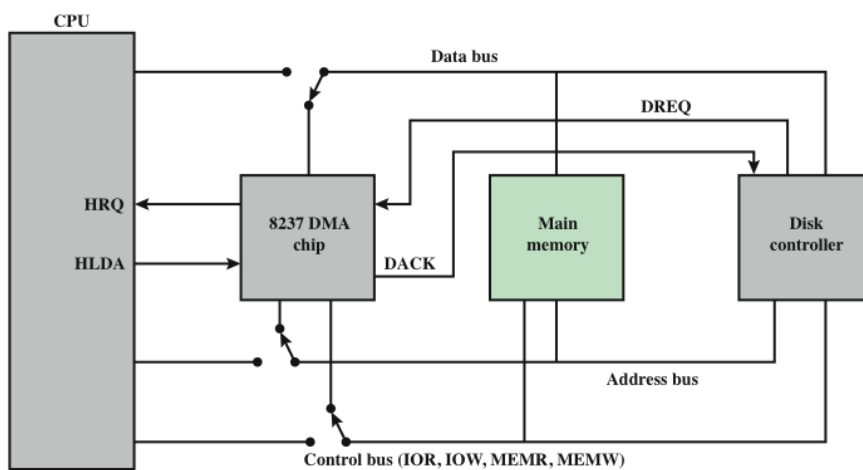


- DMA giao tiếp với I/O không qua bus hệ thống: giảm được chu kỳ bus
- DMA chỉ chiếm bus hệ thống khi cần trao đổi dữ liệu với bộ nhớ chính

+ d. 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 acknowledged) - 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
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request

+ d. 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

Bit	Command	Status	Mode	Single Mask	All Mask
D0	Memory-to-memory E/D	Channel 0 has reached TC	Channel select	Select channel mask bit	Clear/set channel 0 mask bit
D1	Channel 0 address hold E/D	Channel 1 has reached TC			Clear/set channel 1 mask bit
D2	Controller E/D	Channel 2 has reached TC		Clear/set mask bit	Clear/set channel 2 mask bit
D3	Normal/compressed timing	Channel 3 has reached TC	Verify/write/read transfer	Not used	Clear/set channel 3 mask bit
D4	Fixed/rotating priority	Channel 0 request	Auto-initialization E/D		Not used
D5	Late/extended write selection	Channel 0 request	Address increment/decrement select		
D6	DREQ sense active high/low	Channel 0 request			
D7	DACK sense active high/low	Channel 0 request	Demand/single/block/cascade mode select		

E/D = enable/disable

TC = terminal count

Bảng 7.2
Thanh ghi
Intel
8237A

+ 6. VXL và các kênh I/O

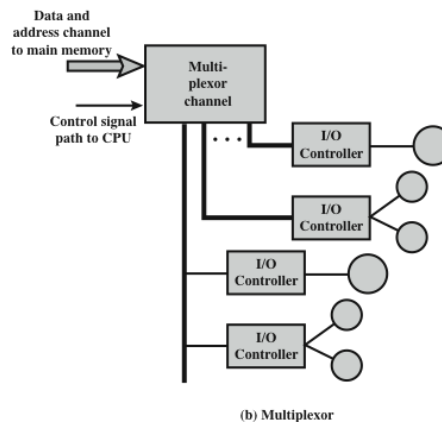
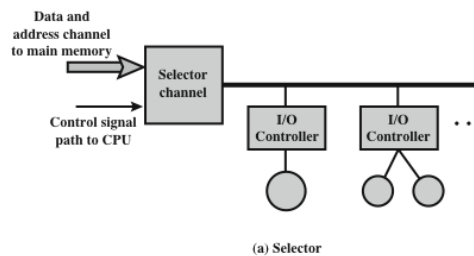
a. Sự phát triển của chức năng I/O qua các thời kỳ

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.

I/O channel

+

Kiến trúc I/O channel



+ Review Questions

1. Liệt kê ba loại thiết bị ngoại vi (thiết bị ngoài).
2. IRA - International Reference Alphabet là gì?
3. Các chức năng chính của module I/O là gì?
4. Trình bày ba kỹ thuật để thực hiện I/O.
5. Sự khác nhau giữa I/O ánh xạ bộ nhớ và I/O riêng biệt là gì?
6. Khi một ngắt được gửi đến VXL, bộ xử lý xác định thiết bị đã yêu cầu ngắt như thế nào?
7. Trong khi một module DMA chiếm quyền điều khiển bus VXL làm gì?