

CHƯƠNG 6. HỆ THỐNG VÀO RA

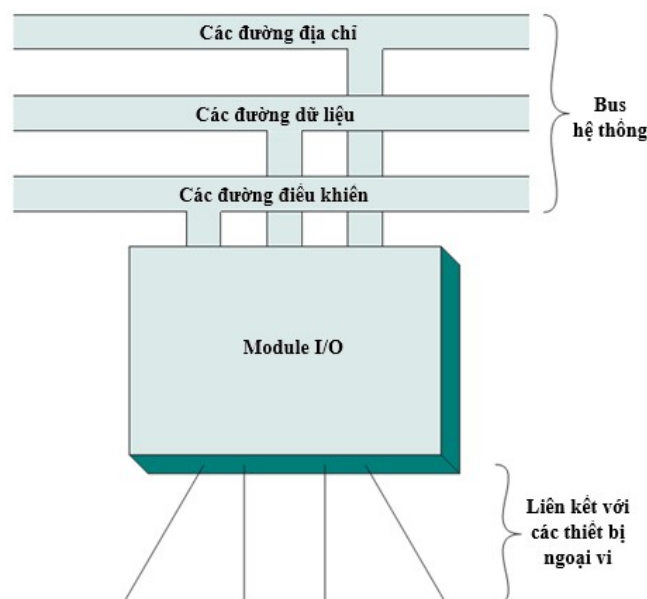
Ngoài bộ vi xử lý và bộ nhớ, thành phần quan trọng thứ ba của một hệ thống máy tính là các module I/O. Mỗi module giao tiếp với bus hệ thống hoặc bộ chuyển mạch trung tâm và điều khiển một hoặc nhiều thiết bị ngoại vi. Một module I/O không đơn giản chỉ kết nối cơ học các thiết bị với bus hệ thống mà còn chứa các logic thực hiện chức năng truyền thông tin giữa thiết bị ngoại vi và bus.

Vậy tại sao ta không kết nối trực tiếp thiết bị ngoại vi bus hệ thống. Có một số lý do như sau:

- Có rất nhiều thiết bị ngoại vi với nhiều phương thức hoạt động khác nhau. Nếu bộ xử lý kết nối trực tiếp với thiết bị ngoại vi nó sẽ cần phải được trang bị các logic để điều khiển các thiết bị đó, điều này không cần thiết và gây lãng phí tài nguyên của bộ xử lý.
- Tốc độ truyền dữ liệu của thiết bị ngoại vi thường chậm hơn nhiều so với bộ nhớ hoặc bộ vi xử lý. Do đó, sẽ là không thực tế nếu sử dụng bus tốc độ cao để giao tiếp trực tiếp với thiết bị ngoại vi.
- Mặt khác, tốc độ truyền dữ liệu của một số thiết bị ngoại vi nhanh hơn bộ nhớ hoặc bộ vi xử lý. Một lần nữa, sự chênh lệch này sẽ dẫn đến sự không hiệu quả trong hiệu năng hệ thống nếu ta không có cơ chế quản lý phù hợp.
- Thiết bị ngoại vi thường sử dụng các định dạng dữ liệu và kích thước từ khác so với máy tính mà chúng được gắn vào.

Do đó, việc sử dụng một module I/O là cần thiết. Module này có hai chức năng chính

- Giao tiếp với bộ xử lý và bộ nhớ thông qua bus hệ thống hoặc một chuyển mạch trung tâm
- Giao tiếp với một hoặc nhiều thiết bị ngoại vi bằng các liên kết dữ liệu phù hợp



Hình 6.1 Sơ đồ kết nối module vào/ra

6.1. THIẾT BỊ NGOẠI VI

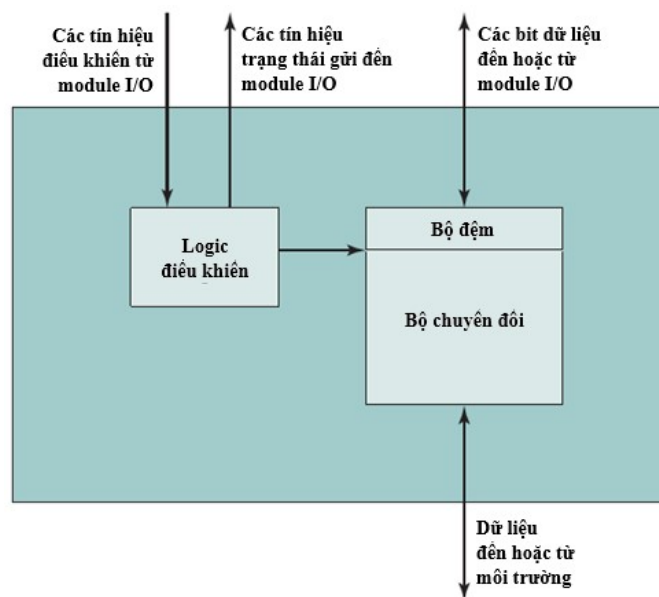
Các hoạt động I/O được thực hiện thông qua một loạt các thiết bị ngoài cung cấp một phương tiện trao đổi dữ liệu giữa môi trường bên ngoài và máy tính. Thiết bị ngoài kết nối với máy tính thông qua một liên kết tới module I/O (Hình 7.1). Liên kết này được sử dụng để chuyển tiếp điều khiển, trạng thái và dữ liệu giữa module I/O và thiết bị ngoại vi. Chúng ta có thể phân loại các thiết bị ngoại vi thành ba loại như sau:

- Con người có thể đọc được: Thích hợp để giao tiếp với người sử dụng máy tính
- Máy có thể đọc được: Thích hợp để giao tiếp với thiết bị
- Truyền thông tin: Thích hợp để giao tiếp với các thiết bị từ xa

Ví dụ về thiết bị con người có thể đọc được đó là màn hình và máy in. Các thiết bị máy có thể đọc được là các hệ thống đĩa và băng từ, các cảm biến và bộ truyền động được sử dụng trong ứng dụng robot. Có một điểm cần chú ý là ta đang xem các hệ thống đĩa và băng từ là các thiết bị I/O trong khi ở chương 6, ta xem chúng như các thiết bị nhớ. Về mặt chức năng, các thiết bị này là một phần của hệ thống phân cấp bộ nhớ, tuy nhiên về mặt cấu trúc, các thiết bị này được điều khiển bởi các module I/O và do đó phải được xét đến trong chương này. Thiết bị truyền thông tin cho phép máy tính trao đổi dữ liệu với thiết bị từ xa.

Hình 6.2 là sơ đồ khối chung của các thiết bị ngoại vi. Giao diện với module I/O thông qua các tín hiệu điều khiển, dữ liệu và trạng thái. Các tín hiệu điều khiển xác định chức năng mà thiết bị sẽ thực hiện, chẳng hạn như gửi dữ liệu đến module I/O (INPUT hoặc ĐỌC), tiếp nhận dữ liệu từ module I/O (OUTPUT hoặc GHI), báo cáo trạng thái hoặc thực hiện một số chức năng điều khiển đặc biệt đối với thiết bị (ví dụ: đặt đầu đọc/ghi của đĩa từ vào một vị trí nào đó). *Dữ liệu* ở dạng một tập bit được gửi tới hoặc nhận từ module I/O. *Các tín hiệu trạng thái* cho biết trạng thái của thiết bị. Ví dụ là READY/NOT-READY (sẵn sàng hay chưa sẵn sàng) để thông báo thiết bị có sẵn sàng cho việc truyền dữ liệu hay không.

Logic điều khiển tiếp nhận tín hiệu điều khiển từ module I/O và thực hiện việc điều khiển hoạt động thiết bị. Bộ chuyển đổi thực hiện việc chuyển đổi dữ liệu từ dạng tín hiệu điện sang các dạng biểu diễn khác đối với các thiết bị ra và từ các dạng tín hiệu khác nhau thành dữ liệu dạng điện với các thiết bị vào. Bộ chuyển đổi thường đi kèm với một bộ nhớ đệm để lưu trữ dữ liệu tạm thời trong quá trình trao đổi dữ liệu giữa module I/O và môi trường bên ngoài; một kích thước bộ đệm phổ biến từ 8 đến 16 bit.



Hình 6.2 Sơ đồ khối của các thiết bị ngoài

Giao diện giữa module I/O và thiết bị bên ngoài sẽ được trình bày trong phần sau. Giao diện giữa thiết bị ngoài và môi trường không nằm trong phạm vi kiến thức của cuốn giáo trình này nhưng chúng tôi vẫn trình bày tóm tắt một số thiết bị.

Bàn phím/Màn hình

Các thiết bị ngoại vi phổ biến nhất tương tác giữa con người và máy tính là bàn phím và màn hình. Người dùng đưa thông tin vào thông qua bàn phím. Dữ liệu vào này sau đó được truyền đến máy tính và cũng có thể được hiển thị trên màn hình. Ngoài ra, màn hình hiển thị dữ liệu được máy tính đưa ra.

Đơn vị dữ liệu cơ bản được trao đổi là các ký tự. Mỗi ký tự được liên kết với một mã, thường có chiều dài 7 đến 8 bit. Bảng mã ký tự được sử dụng phổ biến nhất là bảng mã International Reference Alphabet (IRA). Mỗi ký tự trong bảng mã này được biểu diễn bởi một mã nhị phân 7-bit; do đó có 128 ký tự khác nhau có thể được mã hóa. Ký tự có hai loại: ký tự có thể in được và ký tự điều khiển. Các ký tự in được là chữ cái, chữ số và một số ký tự đặc biệt có thể được in trên giấy hoặc hiển thị trên màn hình. Các ký tự điều khiển liên quan đến điều khiển việc in hoặc hiển thị ký tự. Một số ký tự điều khiển khác liên quan đến các thủ tục truyền thông.

Với bàn phím, khi người dùng nhấn một phím, bàn phím sẽ phát ra một tín hiệu điện, tín hiệu này được bộ chuyển đổi dịch sang mẫu bit nhị phân dưới dạng mã IRA tương ứng. Mẫu bit này sau đó được truyền đến module I/O trong máy tính. Với đầu ra, các ký tự biểu diễn bằng mã IRA được truyền đến thiết bị ngoại vi từ module I/O. Bộ chuyển đổi của thiết bị này giải mã và gửi thông tin đến thiết bị đầu ra để hiển thị ký tự hoặc thực hiện chức năng điều khiển.

Ổ cứng

Một ổ đĩa chứa một số bộ phận điện tử để trao đổi các tín hiệu dữ liệu, điều khiển và trạng thái với module I/O cộng với các bộ phận khác để điều khiển cơ chế đọc/ghi đĩa. Với đĩa đầu cố định, bộ chuyển đổi sẽ thực hiện việc chuyển đổi giữa các mẫu từ tính trên bề mặt đĩa thành các bit ghi vào bộ đệm và ngược lại (Hình 7.2). Với đĩa cứng đầu di chuyển, cũng cần phải có một bộ phận thực hiện chức năng điều khiển việc di chuyển đầu đọc/ghi đến vị trí nhất định trên bề mặt đĩa.

6.2. CÁC MODULE I/O

6.2.1. Chức năng của module I/O

Các chức năng chính module I/O như sau:

- Điều khiển và định thời
- Giao tiếp với bộ vi xử lý
- Giao tiếp với thiết bị
- Đệm dữ liệu
- Phát hiện lỗi

Trong bất kỳ thời điểm nào, bộ xử lý có thể giao tiếp với một hoặc nhiều thiết bị ngoài theo nhiều cách khác nhau tùy thuộc vào yêu cầu vào/ra dữ liệu của chương trình. Các tài nguyên bên trong, ví dụ như bộ nhớ chính và hệ thống bus sẽ bị chia sẻ với nhiều hoạt động, trong đó gồm cả các hoạt động I/O. Vì vậy, chức năng I/O bao gồm cả các yêu cầu **điều khiển và định thời** để phối hợp luồng lưu lượng giữa các tài nguyên bên trong và các thiết bị ngoài. Ví dụ quá trình điều khiển truyền dữ liệu từ thiết bị ngoại vi đến bộ xử lý gồm các bước sau:

1. Bộ xử lý yêu cầu module I/O kiểm tra trạng thái của 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 cách gửi lệnh cho module I/O.
4. Module I/O nhận một đơn vị dữ liệu (ví dụ, 8 hoặc 16 bit) từ thiết bị ngoài.
5. Dữ liệu được module I/O chuyển sang cho bộ xử lý.

Nếu hệ thống chỉ sử dụng một bus thì tương tác giữa bộ vi xử lý và module I/O sẽ phải sử dụng các cơ chế phân xử bus.

Trong đó, module I/O phải thực hiện việc giao tiếp với bộ vi xử lý và thiết bị ngoại vi. Giao tiếp với bộ xử lý như sau:

- **Giải mã lệnh:** Module I/O nhận lệnh từ bộ xử lý (thường được gửi dưới dạng các tín hiệu trên bus điều khiển). Ví dụ, một module I/O cho ổ đĩa có thể nhận các lệnh sau: READ SECTOR (đọc một sector), WRITE SECTOR (ghi một sector), SEEK track number (tìm kiếm một track), và SCAN record ID (thăm dò một bản ghi). Hai lệnh sau được gửi kèm với một tham số qua bus dữ liệu
- **Dữ liệu:** Dữ liệu được trao đổi giữa bộ xử lý và module I/O qua bus dữ liệu.

- **Báo cáo trạng thái:** Vì các thiết bị ngoại vi thường rất chậm nên bộ xử lý thường phải biết tình trạng của module I/O để yêu cầu thực hiện các hoạt động tiếp theo. Ví dụ, nếu một module I/O được yêu cầu gửi dữ liệu đến bộ xử lý (READ), có thể module này chưa sẵn sàng để thực hiện yêu cầu vì nó vẫn đang thực hiện yêu cầu I/O trước. Điều này cần được báo cáo với bộ xử lý thông qua một tín hiệu trạng thái. Tín hiệu trạng thái thường sử dụng BUSY và READY. Ngoài ra, cũng có thể có một số tín hiệu báo cáo khác với các điều kiện lỗi khác nhau.

- **Nhận dạng địa chỉ:** Giống như mỗi từ nhớ có một địa chỉ, các thiết bị I/O cũng vậy. Một module I/O phải nhận ra một địa chỉ duy nhất cho mỗi thiết bị ngoại vi mà nó điều khiển.

Mặt khác, module I/O phải thực hiện việc giao tiếp với các thiết bị. Giao tiếp này gồm có các lệnh, thông tin trạng thái và dữ liệu (Hình 6.2).

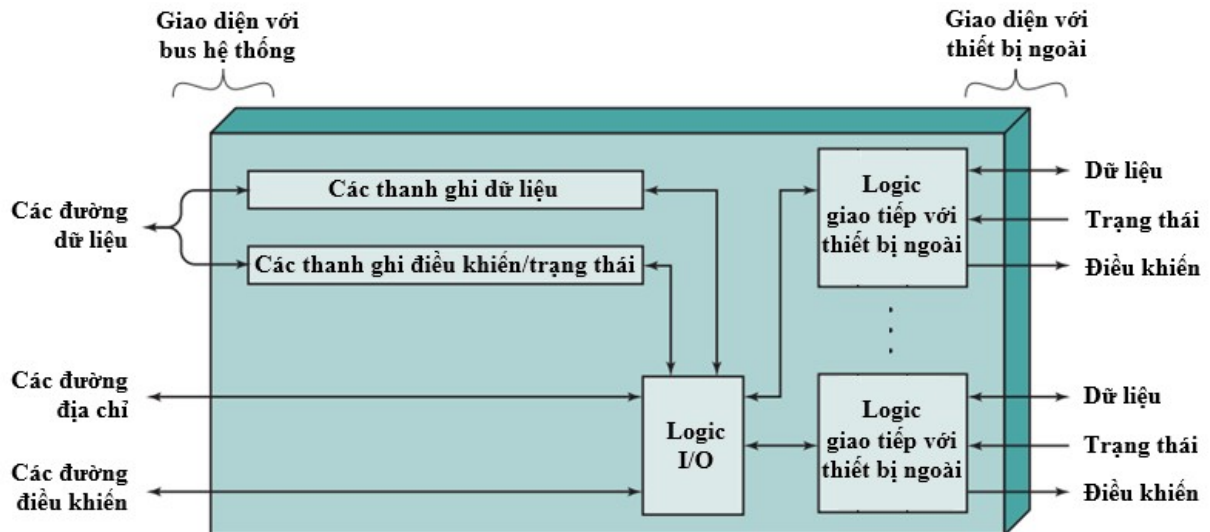
Một nhiệm vụ thiết yếu nữa của module I/O là **đệm dữ liệu**. Chức năng này hết sức cần thiết do tốc độ trao đổi dữ liệu của bộ nhớ chính và bộ xử lý khá cao so với hầu hết các thiết bị ngoại vi. Dữ liệu từ bộ nhớ chính được gửi đến một module I/O với tốc độ nhanh và cần được đệm trong module I/O sau đó gửi đến thiết bị ngoại vi với tốc độ của thiết bị. Ngược lại, dữ liệu được đệm lại để không làm mất thời gian của bộ nhớ nhận truyền chậm. Do đó, module I/O phải hoạt động ở cả tốc độ của bộ nhớ và thiết bị ngoại vi. Tương tự trong trường hợp nếu thiết bị I/O có tốc độ cao hơn tốc độ truy cập bộ nhớ, module I/O sẽ phải thực hiện hoạt động đệm dữ liệu cần thiết.

Cuối cùng, một module I/O thường chịu trách nhiệm **phát hiện lỗi** và sau đó báo lỗi cho bộ vi xử lý. Một số loại bao gồm: hỏng về điện hoặc cơ học (ví dụ: kẹt giấy, một track trong ổ cứng). Một loại lỗi khác có thể do mẫu dữ liệu truyền từ thiết bị đến module I/O vô tình bị thay đổi. Người ta sử dụng một loại mã phát hiện lỗi để phát hiện lỗi truyền tải. Ví dụ như việc truyền mã ký tự IRA, mã này chiếm 7 bit của một byte, vậy bit thứ tám được thiết lập để tổng các bit 1 trong byte là chẵn (parity chẵn) hoặc lẻ (parity lẻ). Khi module I/O nhận được byte, nó sẽ kiểm tra bit chẵn lẻ này để xác định xem có lỗi xảy ra hay không.

6.2.2.Cấu trúc Module I/O

Các module I/O khác nhau đáng kể về độ phức tạp và số lượng thiết bị bên ngoài mà chúng điều khiển. Tại đây ta sẽ cố gắng chỉ mô tả một cấu trúc chung nhất của các module này. (Một thiết bị cụ thể là Intel 82C55A sẽ được mô tả chi tiết hơn trong Phần 7.4). Sơ đồ khối chung của module I/O được minh họa trong Hình 7.3. Module này kết nối với phần còn lại của máy tính thông qua một tập hợp các đường tín hiệu (ví dụ: bus hệ thống). Dữ liệu được chuyển đến hoặc từ module được lưu đệm trong các thanh ghi dữ liệu. Các thanh ghi trạng thái cung cấp thông tin về trạng thái hiện tại của thiết bị. Thanh ghi trạng thái cũng có chức năng như một thanh ghi điều khiển: nhận các thông tin điều khiển từ bộ vi xử lý. Logic I/O tương tác với bộ vi xử lý thông qua một tập các đường điều khiển. Bộ vi xử

lý sử dụng các đường điều khiển để chuyển các lệnh cho module I/O. Một số đường điều khiển có thể được sử dụng bởi module I/O (ví dụ các tín hiệu phân xử và trạng thái bus). Module này cũng phải có khả năng nhận dạng và sinh ra các địa chỉ liên kết với các thiết bị mà nó điều khiển. Mỗi module I/O có một (nếu chỉ nối với một TBNV) hoặc một tập địa chỉ (nếu module nối với nhiều TBNV). Cuối cùng, module I/O chứa các logic giao tiếp với từng thiết bị nối vào nó.



Hình 6.3 Sơ đồ khối Module I/O

Với hoạt động của module I/O, bộ xử lý có thể nhìn nhận một loạt các thiết bị theo một cách đơn giản. Module I/O có thể ẩn các thông tin về định thời, định dạng, các vấn đề về điện-cơ của thiết bị ngoại vi, bộ xử lý chỉ cần điều khiển hoạt động thông qua các lệnh đọc và ghi đơn giản hoặc có thể là các lệnh đóng, mở tập tin. Tuy nhiên, cũng có những module I/O hoạt động khá đơn giản và vì vậy để lại phần lớn công việc điều khiển thiết bị cho bộ xử lý.

Những module I/O có khả năng xử lý cao, hỗ trợ nhiều cho bộ xử lý thường được gọi là *kênh I/O (I/O channel)* hoặc *bộ xử lý I/O (I/O processor)*. Những module I/O đơn giản hơn và cần có sự điều khiển chi tiết hơn từ bộ xử lý thường được gọi là *bộ điều khiển I/O (I/O controller)* hoặc *bộ điều khiển thiết bị (device controller)*. Bộ điều khiển I/O thường được sử dụng trong các máy vi tính, còn kênh I/O thường được sử dụng trên các dòng máy tính lớn (mainframe).

Phần dưới đây, chúng tôi sẽ sử dụng khái niệm module I/O chung cho cả hai loại trên, nếu có điểm khác biệt thì chúng tôi sẽ giải thích cụ thể trong từng trường hợp.

6.3. CÁC KỸ THUẬT VÀO/RA

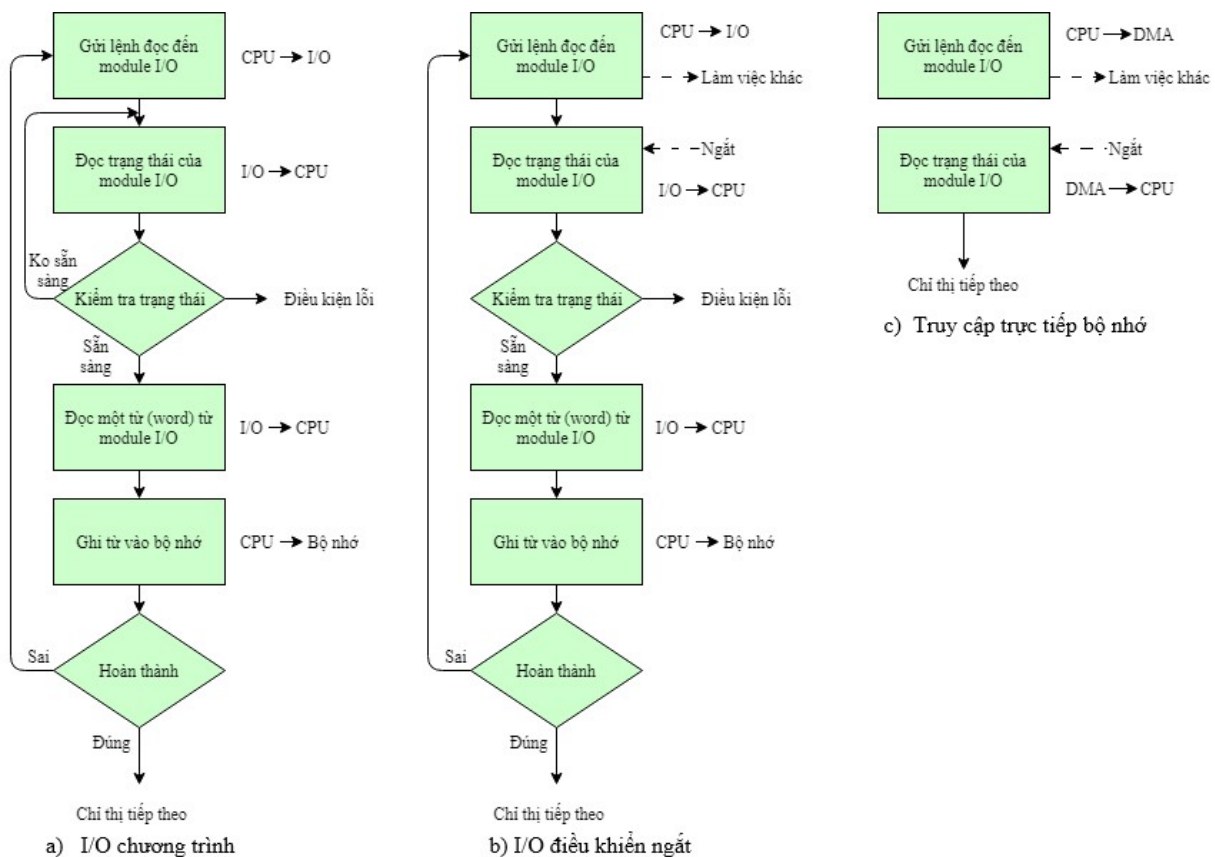
Có ba kỹ thuật để thực hiện các hoạt động vào/ra (I/O).

- **I/O chương trình:** Dữ liệu được trao đổi giữa bộ vi xử lý và module I/O. Bộ xử lý thực thi một chương trình cho phép nó trực tiếp điều khiển hoạt động vào/ra, bao gồm cảm nhận tình

trạng thiết bị, gửi lệnh đọc hoặc ghi, truyền dữ liệu. Khi bộ vi xử lý ra lệnh cho module I/O, nó phải đợi cho đến khi hoạt động I/O hoàn thành. Nếu bộ xử lý nhanh hơn module I/O thì việc chờ đợi này gây lãng phí thời gian của bộ xử lý.

- **I/O điều khiển ngắt:** bộ xử lý đưa ra một *lệnh I/O* (I/O command) sau đó tiếp tục thực hiện các lệnh (instruction) khác trong chương trình, khi nào module I/O hoàn thành công việc của mình nó sẽ gửi yêu cầu ngắt tới bộ xử lý để bộ xử lý tiếp tục điều khiển hoạt động vào/ra.

- Với cả I/O chương trình và I/O điều khiển ngắt, bộ vi xử lý có trách nhiệm lấy dữ liệu từ bộ nhớ chính rồi chuyển cho thiết bị ngoại vi (hoạt động Ghi) hoặc lưu dữ liệu từ thiết bị ngoại vi vào bộ nhớ chính (hoạt động Đọc). Vì vậy, một cơ chế thay thế được đưa ra là **cơ chế DMA (direct memory access – truy cập bộ nhớ trực tiếp)** để giảm thiểu sự tham gia của bộ xử lý vào quá trình trao đổi dữ liệu giữa bộ nhớ và thiết bị ngoại vi, tăng hiệu năng của hệ thống.



Hình 6.4. Ba kỹ thuật Đọc một khối dữ liệu vào

6.3.1. I/O chương trình

Khi bộ vi xử lý đang thực hiện một chương trình và gặp một chỉ thị (instruction) liên quan đến vào/ra dữ liệu, nó thực hiện chỉ thị đó bằng cách phát ra *lệnh (command)* cho module I/O thích

hợp. Với I/O chương trình, module I/O thực hiện hành động yêu cầu và sau đó thiết lập các bit thích hợp trong thanh ghi trạng thái I/O (Hình 7.3). Module I/O không có tác vụ nào khác để báo cáo về bộ vi xử lý. Do đó, để biết được công việc đã hoàn thành hay chưa, bộ xử lý phải định kỳ để kiểm tra trạng thái của module I/O cho đến khi hoạt động đã hoàn thành.

Để giải thích cho kỹ thuật I/O chương trình, chúng ta xem xét nó từ cách nhìn của các *lệnh I/O* do bộ vi xử lý gửi đến module I/O và sau đó từ cách nhìn của các chỉ thị được thực hiện bởi bộ vi xử lý.

❖ Các lệnh I/O

Để thực hiện chỉ thị liên quan đến vào/ra, bộ xử lý sẽ đưa ra một địa chỉ xác định module I/O và thiết bị bên ngoài cụ thể và đưa ra một lệnh I/O. Có bốn loại lệnh I/O như sau:

- **Điều khiển:** Được sử dụng để kích hoạt một thiết bị ngoại vi và ra lệnh cho nó phải làm gì. Ví dụ, một ổ đĩa được điều khiển để quay lại hoặc tới trước một bản ghi. Các lệnh điều khiển được thiết kế riêng cho từng loại thiết bị ngoại vi.
- **Kiểm tra:** Được sử dụng để kiểm tra các điều kiện trạng thái của module I/O và thiết bị ngoại vi. Bộ xử lý cần phải biết liệu các thiết bị ngoại vi có đang bật nguồn và sẵn sàng sử dụng. Nó cũng cần phải biết hoạt động I/O gần nhất đã hoàn thành chưa và có lỗi nào xảy ra không.
- **Đọc:** Yêu cầu module I/O lấy dữ liệu từ thiết bị ngoại vi và đặt nó vào bộ đệm (thanh ghi dữ liệu trong Hình 6.3). Bộ xử lý sau đó có thể lấy dữ liệu bằng cách yêu cầu module I/O đặt dữ liệu lên bus dữ liệu.
- **Ghi:** Yêu cầu module I/O chuyển dữ liệu (1 byte hoặc 1 từ) từ bus dữ liệu đến thiết bị ngoại vi.

Hình 6.4a là một ví dụ về việc sử dụng I/O chương trình để đọc một khối dữ liệu từ thiết bị ngoại vi (như một bản ghi từ băng từ) vào bộ nhớ. Dữ liệu được đọc mỗi lần một từ (16 bit). Đối với mỗi từ được đọc, bộ xử lý phải kiểm tra trạng thái cho đến khi nó xác định được từ đó đã được đọc vào thanh ghi dữ liệu của module I/O. Từ sơ đồ này ta thấy nhược điểm chính của kỹ thuật này là một quá trình lặp đi lặp lại việc kiểm tra trạng thái làm cho bộ xử lý thêm bận rộn.

❖ Chỉ thị vào/ra

Với I/O chương trình, có sự tương ứng chặt chẽ giữa các chỉ thị (lệnh) liên quan đến vào/ra mà bộ vi xử lý truy xuất từ bộ nhớ và các lệnh I/O (command) mà bộ xử lý gửi đến module I/O để thực hiện chỉ thị đó. Do đó, các chỉ thị cần được ánh xạ thành các lệnh I/O và chúng thường có quan hệ một-một đơn giản. Dạng của chỉ thị phụ thuộc vào cách các thiết bị ngoại vi được định địa chỉ.

Thông thường, có nhiều thiết bị I/O được kết nối với hệ thống thông qua các module I/O. Mỗi thiết bị có một số nhận dạng hoặc địa chỉ duy nhất. Khi bộ xử lý đưa lệnh I/O đến thiết bị thì

trong lệnh phải chứa thông tin địa chỉ của thiết bị. Sau đó, các module I/O phải dịch các dòng địa chỉ để xác định xem liệu lệnh này có phải gửi cho nó.

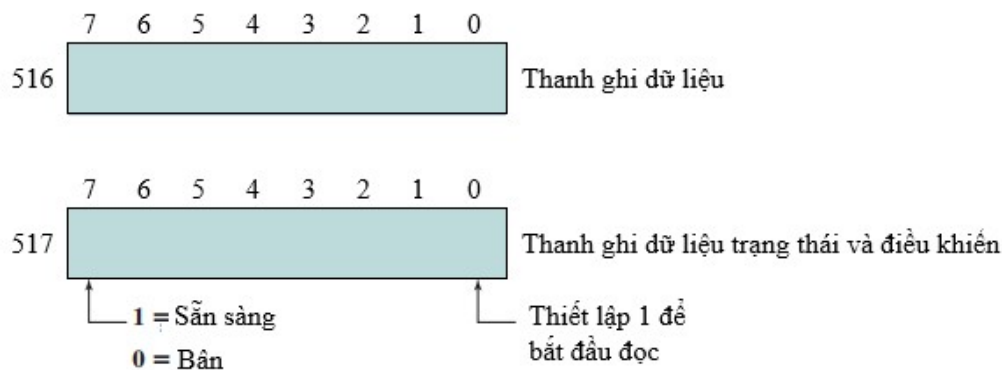
Khi bộ xử lý, bộ nhớ và module I/O chia sẻ một bus chung, hai chế độ định địa chỉ có thể được thực hiện như sau: **I/O ánh xạ bộ nhớ (memory-mapped I/O)** và **I/O riêng biệt (isolated I/O)**. I/O ánh xạ bộ nhớ sử dụng một không gian địa chỉ chung cho bộ nhớ và các thiết bị ngoại vi. Bộ xử lý coi thanh ghi trạng thái và dữ liệu của các module I/O giống như vị trí bộ nhớ và sử dụng cùng các lệnh máy để truy cập cả bộ nhớ và các thiết bị ngoại vi.

Ví dụ, với 10 đường địa chỉ ta có một không gian địa chỉ gồm 1024 địa chỉ vị trí bộ nhớ. Với kỹ thuật I/O riêng biệt, bus được trang bị thêm một đường command line, đường này sẽ cho biết địa chỉ trên bus là địa chỉ của vị trí bộ nhớ hay thiết bị I/O. Như vậy, ta có thể sử dụng toàn bộ không gian địa chỉ cho cả bộ nhớ và thiết bị ngoại vi. Như vậy, với 10 đường địa chỉ, hệ thống có thể hỗ trợ 1024 vị trí bộ nhớ và 1024 địa chỉ I/O.

Hình 6.5 so sánh hai kỹ thuật I/O chương trình. Với Hình 6.5a, bàn phím được kết nối với hệ thống theo kỹ thuật I/O ánh xạ bộ nhớ. Giả sử hệ thống có 10-bit địa chỉ, trong đó 512 giá trị dành cho bộ nhớ (từ 0-511) và 512 dành cho I/O (từ 512-1023). Hai địa chỉ được dành riêng cho việc giao tiếp với bàn phím. Địa chỉ 516 được gán cho thanh ghi dữ liệu và 517 được gán cho thanh ghi trạng thái. Thanh ghi trạng thái hoạt động như một thanh ghi điều khiển để nhận các lệnh của bộ vi xử lý. Chương trình trong hình có mục đích đọc 1 byte dữ liệu từ bàn phím vào bộ thanh ghi AC trong bộ xử lý.

Với I/O riêng biệt (Hình 6.5b), các cổng vào/ra chỉ có thể truy cập bằng các lệnh I/O đặc biệt. Các lệnh này sẽ kích hoạt các đường command line trên bus.

Hầu hết các bộ xử lý đều có một tập khá lớn các chỉ thị khác nhau để tham chiếu bộ nhớ. Nếu sử dụng I/O riêng biệt, số lượng chỉ thị vào/ra sẽ khá ít. Do đó, ưu điểm của I/O ánh xạ bộ nhớ là số lượng các chỉ thị tham chiếu bộ nhớ cũng có thể được sử dụng trong tham chiếu thiết bị ngoại vi, vì vậy cho phép việc lập trình hiệu quả hơn. Tuy nhiên, nhược điểm của nó là không gian địa chỉ phải chia sẻ với thiết bị ngoại vi. Cả hai kỹ thuật I/O này đều được sử dụng phổ biến.



ĐỊA CHỈ	CHỈ THỊ	TOÁN HẠNG	CHÚ THÍCH
200	Load AC	“1”	Đọc “1” vào AC
201	Store AC	517	Ghi AC vào 517
			Khởi tạo đọc bàn phím
202	Load AC	517	Đọc 517 vào AC (đọc trạng thái thiết bị)
	Branch if Sign = 0	202	Lặp lại 202 đến khi sẵn sàng
	Load AC	516	Đọc 516 vào AC

a) I/O ánh xạ bộ nhớ

ĐỊA CHỈ	CHỈ THỊ	TOÁN HẠNG	CHÚ THÍCH
200	Load I/O	5	Khởi tạo việc đọc bàn phím
201	Test I/O	5	Kiểm tra sự sẵn sàng
	Branch If Not Ready	201	Lặp lại 201 đến khi thiết bị sẵn sàng
	In	5	Đọc dữ liệu

b) I/O riêng biệt

Hình 6.5 So sánh I/O ánh xạ bộ nhớ và I/O riêng biệt

6.3.2. I/O điều khiển ngắt

Một vấn đề của I/O chương trình là bộ xử lý phải đợi một thời gian dài cho đến khi module I/O sẵn sàng tiếp nhận hoặc truyền dữ liệu. Trong khoảng thời gian đó, bộ xử lý phải liên tục kiểm tra trạng thái của module I/O. Kết quả, hiệu năng của toàn bộ hệ thống bị giảm đi.

Để giải quyết vấn đề trên, người ta đưa ra một giải pháp như sau: bộ vi xử lý gửi lệnh I/O đến module và sau đó thực hiện các hoạt động khác. Khi nào module I/O đã sẵn sàng để trao đổi dữ liệu nó sẽ ngắt bộ xử lý để yêu cầu phục vụ. Bộ xử lý thực thi việc truyền dữ liệu rồi quay trở lại công việc của nó.

Chúng ta sẽ xem xét cách hoạt động này từ hai phía: bộ xử lý và module I/O.

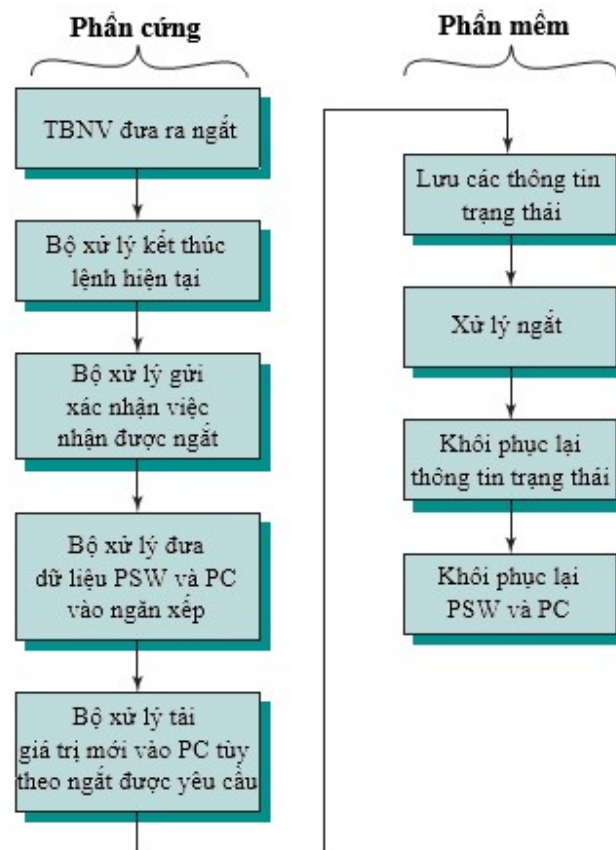
Từ phía module I/O, việc đọc dữ liệu vào như sau: module I/O nhận lệnh READ từ bộ xử lý. Sau đó, nó tiến hành đọc dữ liệu từ một thiết bị ngoại vi được yêu cầu. Khi dữ liệu đã được đọc vào thanh ghi dữ liệu của module, module gửi một báo hiệu ngắt tới bộ xử lý qua một đường điều khiển. Module chờ đợi cho đến khi bộ xử lý gửi yêu cầu dữ liệu. Khi có yêu cầu, module đặt dữ liệu vào bus dữ liệu và được giải phóng, sẵn sàng cho một hoạt động vào/ra khác.

Từ phía bộ xử lý, hoạt động Đọc dữ liệu vào như sau: bộ xử lý ra lệnh READ, sau đó nó sẽ thực hiện một công việc khác (ví dụ: bộ xử lý có thể đang chạy trên nhiều chương trình cùng một lúc). Vào cuối mỗi chu kỳ lệnh, bộ xử lý sẽ kiểm tra các ngắt. Khi có ngắt gửi từ module I/O, bộ xử lý lưu lại ngữ cảnh (ví dụ: thanh ghi PC và các thanh ghi khác của bộ xử lý) của chương trình hiện tại và thực hiện việc xử lý ngắt. Khi xử lý ngắt, bộ xử lý đọc dữ liệu từ module I/O và lưu trữ nó vào bộ nhớ. Sau đó nó khôi phục lại ngữ cảnh trước và tiếp tục thực thi công việc.

Hình 6.4b minh họa việc sử dụng I/O điều khiển ngắt để đọc trong một khối dữ liệu. So với hình 6.4a, rõ ràng việc sử dụng ngắt hiệu quả hơn vì nó giúp loại bỏ khoảng thời gian chờ không cần thiết của bộ xử lý. Tuy nhiên, việc truyền dữ liệu giữa bộ nhớ và module I/O vẫn phải có sự tham gia của bộ xử lý.

Xử lý ngắt

Ta hãy xem xét chi tiết hơn vai trò của bộ xử lý trong I/O điều khiển ngắt. Sự xuất hiện của một ngắt gây ra một số sự kiện của cả phần cứng và phần mềm (Hình 6.6). Khi một thiết bị vào/ra thực hiện một hoạt động vào/ra, một chuỗi các hoạt động phần cứng sau đây sẽ xảy ra



Hình 6.6. Quá trình xử lý ngắt đơn giản

1. Thiết bị phát tín hiệu ngắt cho bộ xử lý.
2. Bộ xử lý hoàn thành lệnh hiện tại trước khi trả lời ngắt.
3. Bộ xử lý kiểm tra xem có ngắt hay không, nếu có một ngắt, và gửi một tín hiệu báo đã nhận (tín hiệu ACK) đến thiết bị đã gửi ngắt. Khi nhận được ACK, thiết bị loại bỏ tín hiệu ngắt.
4. Bộ xử lý cần phải chuyển điều khiển sang chế độ ngắt. Đầu tiên, nó cần phải lưu lại các thông tin của chương trình hiện tại để có thể khôi phục lại công việc sau khi hoàn thành xong việc xử lý ngắt. Các thông tin tối thiểu bắt buộc là (a) trạng thái của bộ xử lý được lưu trữ trong một thanh ghi PSW (thanh ghi trạng thái chương trình), và (b) địa chỉ lệnh tiếp theo sẽ được thực hiện (nội dung thanh ghi PC). Chúng được đẩy lên vùng bộ nhớ ngăn xếp của hệ thống.

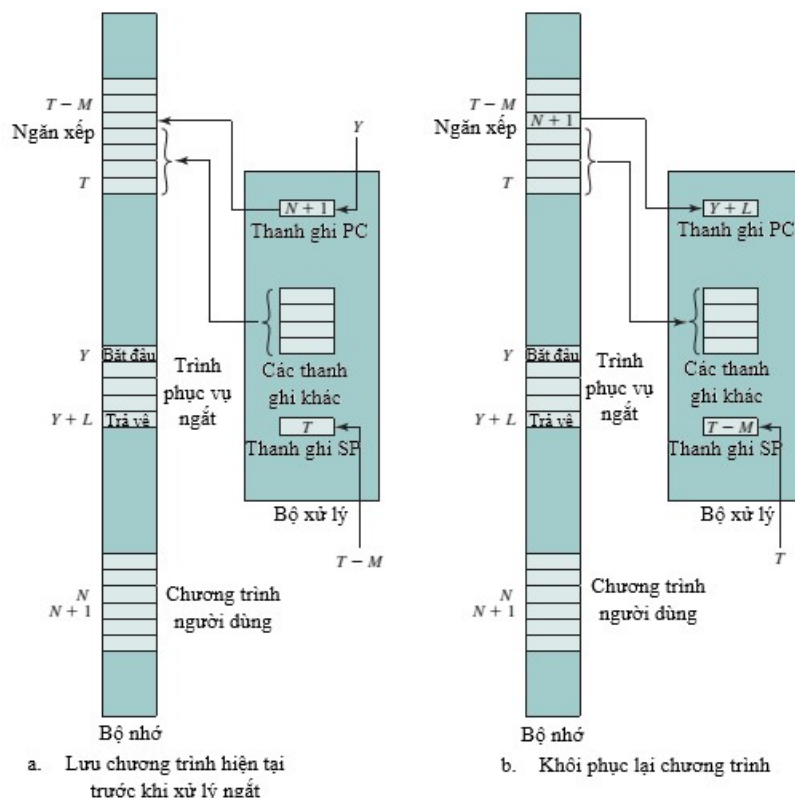
5. Sau đó, bộ xử lý sẽ tải vị trí đầu tiên của chương trình xử lý ngắt vào thanh ghi PC. Tùy thuộc vào kiến trúc hệ thống và thiết kế hệ điều hành, trình xử lý ngắt có thể là một chương trình; một chương trình cho mỗi loại ngắt hoặc một chương trình cho mỗi thiết bị và mỗi loại ngắt. Nếu có nhiều hơn một trình xử lý ngắt, bộ xử lý phải xác định xem ngắt được gửi đến từ đâu. Thông tin đó có thể đã được chứa trong tín hiệu ngắt gửi đến hoặc bộ xử lý phải gửi đến thiết bị gửi ngắt để yêu cầu phản hồi các thông tin cần thiết. Sau khi thanh ghi PC đã được nạp, bộ xử lý bắt đầu thực thi các lệnh trong chương trình xử lý ngắt. Việc thực hiện chương trình này dẫn đến các hoạt động sau:

6. Tại thời điểm này, nội dung các thanh ghi PC và PSW của chương trình bị ngắt đã được lưu trên vùng nhớ ngăn xếp của hệ thống. Tuy nhiên, vẫn còn một số thông tin khác liên quan đến trạng thái chương trình cũng cần phải được lưu lại, đặc biệt là nội dung các thanh ghi trong bộ xử lý vì các thanh ghi này có thể được sử dụng bởi trình xử lý ngắt. Do đó, thông thường, trình xử lý ngắt sẽ bắt đầu bằng việc lưu nội dung của tất cả các thanh ghi vào ngăn xếp. Một ví dụ đơn giản được minh họa trong Hình 6.7a. Trong trường hợp này, một chương trình bị ngắt tại vị trí N . Nội dung của tất cả các thanh ghi cộng với địa chỉ của lệnh tiếp theo ($N + 1$) được đẩy lên ngăn xếp. Thanh ghi SP (con trỏ ngăn xếp) (đọc thêm chương 12) trở đến đỉnh mới của ngăn xếp, và thanh ghi PC trở đến lệnh đầu tiên của trình phục vụ ngắt.

7. Lúc này, ngắt được xử lý bao gồm việc kiểm tra thông tin trạng thái liên quan đến hoạt động I/O hoặc các sự kiện khác gây ra ngắt. Nó cũng có thể thực hiện việc gửi thêm lệnh tiếp theo của một hoạt động I/O đang được thực hiện hoặc gửi tín hiệu ACK đến thiết bị ngoại vi.

8. Khi quá trình xử lý ngắt hoàn tất, giá trị các thanh ghi đã lưu trên ngăn xếp sẽ được lấy ra và khôi phục lại vào các thanh ghi (ví dụ Hình 6.7b).

9. Hoạt động cuối cùng là khôi phục giá trị PSW và PC từ ngăn xếp. Nhờ đó, chương trình bị ngắt sẽ được khôi phục lại tiếp tục được thực thi.



Hình 7.7 Sự thay đổi Bộ nhớ và Thanh ghi khi có Ngắt

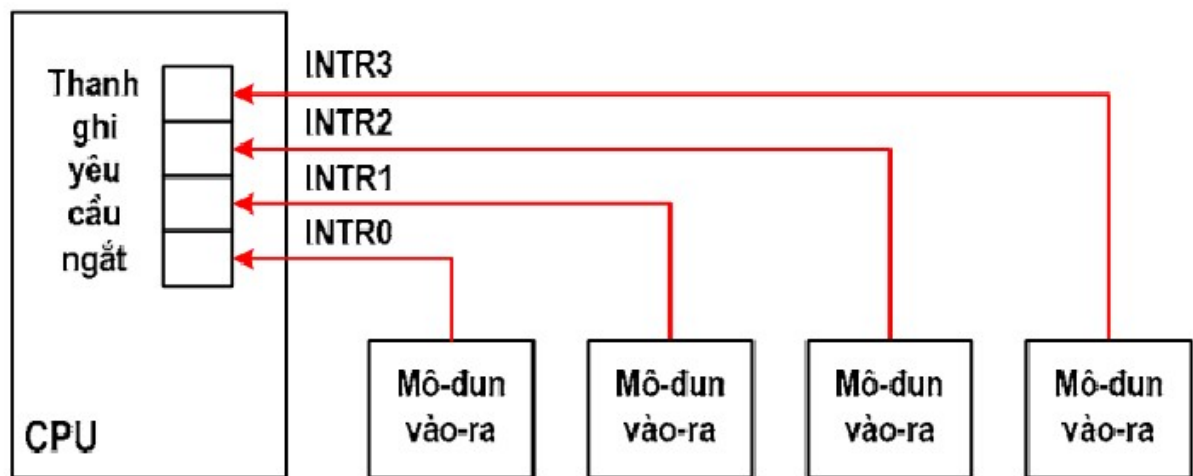
Vấn đề thiết kế

Thực hiện ngắt I/O phát sinh hai vấn đề. Thứ nhất, bởi vì trong hệ thống thường có nhiều module I/O, vậy bộ xử lý xác định thiết bị gửi tín hiệu ngắt như thế nào? Thứ hai, nếu nhiều yêu cầu ngắt được cùng gửi đến, bộ xử lý sẽ quyết định phục vụ ngắt nào trước ngắt nào sau?

Vấn đề đầu tiên, có bốn kỹ thuật thường được sử dụng để xác định thiết bị gửi ngắt như sau:

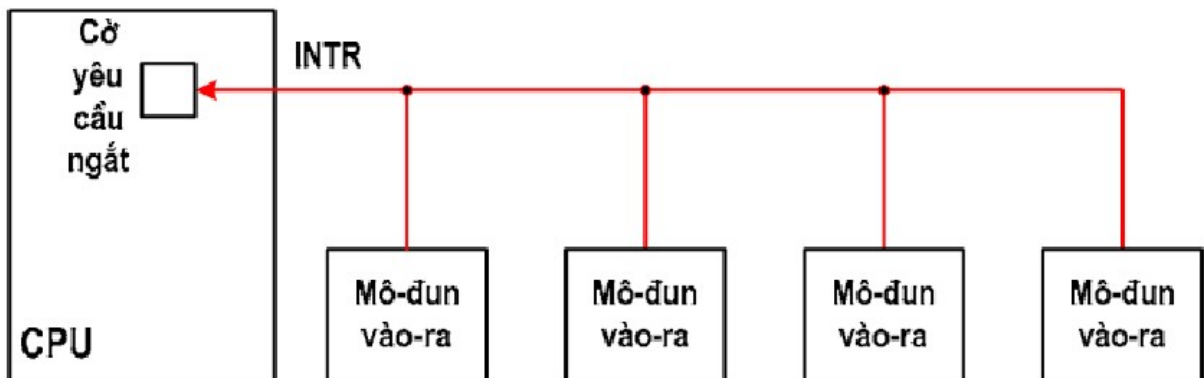
- Sử dụng nhiều đường ngắt
- Thăm dò phần mềm
- Chuỗi Daisy (thăm dò phần cứng, vector)
- Phân xử bus (vector)

Phương pháp đơn giản nhất để giải quyết vấn đề này là ta có thể sử dụng **nhiều đường truyền yêu cầu ngắt** giữa bộ xử lý và các module I/O. Tuy nhiên, việc sử dụng các đường bus và các chân của bộ xử lý để làm các đường ngắt là không thực tế. Do đó, ngay cả khi sử dụng kỹ thuật này, có thể mỗi đường ngắt vẫn sẽ nối với nhiều module I/O và vẫn phải sử dụng một trong ba kỹ thuật còn lại cho mỗi đường.



Hình 6.8 Kỹ thuật nhiều đường ngắt

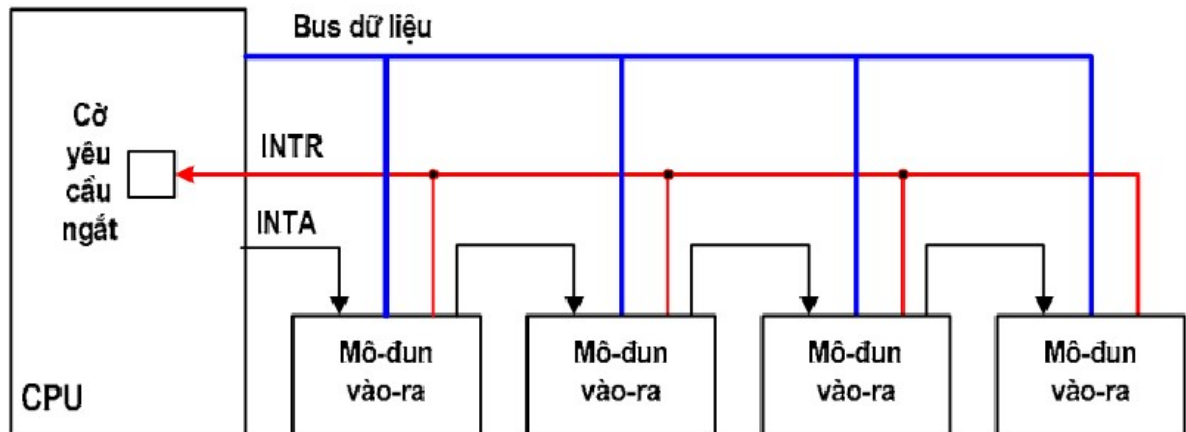
Với kỹ thuật **thăm dò phần mềm**: khi bộ xử lý phát hiện ra một ngắt, nó rẽ nhánh sang một *trình phục vụ ngắt chung* có nhiệm vụ thăm dò từng module I/O để xác định module nào phát ra ngắt. Tín hiệu thăm dò có thể dưới dạng một đường lệnh riêng (ví dụ: TEST I/O). Bộ xử lý phát ra tín hiệu TEST I/O và đặt địa chỉ của từng module vào các đường địa chỉ. Module I/O gửi yêu cầu ngắt sẽ trả lời khi nhận được thăm dò. Một cách thăm dò khác là bộ xử lý sẽ đọc các thanh ghi trạng thái của từng module để xác định module nào gửi ngắt. Sau khi xác định được module ngắt, bộ xử lý sẽ chuyển điều khiển sang trình phục vụ ngắt dành riêng cho thiết bị đó. Tuy nhiên, nhược điểm của phương pháp này là tốn thời gian.



Hình 6.9 Kỹ thuật thăm dò phần mềm

Kỹ thuật chuỗi Daisy thực hiện thăm dò bằng phần cứng. Kỹ thuật này sử dụng một đường ngắt chung (INTR) cho tất cả các module. Khi nhận được yêu cầu ngắt, bộ xử lý gửi một tín hiệu ACK thừa nhận ngắt, tín hiệu này được truyền trên một đường INTA đi qua tất cả các module (nối chuỗi) (Hình 6.9) cho đến khi gặp được module gửi ngắt, module này sẽ gửi lại một tín hiệu trả lời bằng cách đặt một từ lên bus dữ liệu, từ này là có thể là địa chỉ hoặc thông tin nhận diện

của module đó, được gọi là *vector*. Khi đó, bộ xử lý sẽ sử dụng vector để trở tới trình phục vụ ngắt của module đó. Với kỹ thuật này, bộ xử lý không cần phải rẽ nhánh đến trình phục vụ ngắt chung như trường hợp trên.



Hình 7.10 Kỹ thuật chuỗi Daisy

Một kỹ thuật khác cũng sử dụng vector ngắt là **kỹ thuật phân xử bus**. Một module I/O muốn gửi yêu cầu ngắt thì cần phải có quyền chiếm bus trước. Vì vậy, tại một thời điểm chỉ có một module có thể sử dụng bus. Khi bộ xử lý phát hiện ra ngắt, nó trả lời bằng đường ACK. Khi nhận được ACK, module yêu cầu đặt các vector của nó vào các đường dữ liệu và bộ xử lý cũng thực hiện việc xử lý ngắt tương tự như với kỹ thuật chuỗi Daisy. Với các kỹ thuật trên, ta cũng có thể giải quyết luôn vấn đề xác định thứ tự ưu tiên trong trường hợp có nhiều ngắt gửi đến bộ xử lý cùng một thời điểm. Với kỹ thuật nhiều đường ngắt, mỗi đường ngắt sẽ được gán một độ ưu tiên nhất định, khi đó, bộ xử lý chỉ cần chọn đường ngắt có độ ưu tiên cao nhất. Với thăm dò phần mềm, bộ xử lý sẽ thực hiện việc thăm dò các module lần lượt theo thứ tự ưu tiên nó. Tương tự, thứ tự các module được nối chuỗi daisy sẽ được xác định theo độ ưu tiên của chúng. Cuối cùng, trong kỹ thuật phân xử bus, module có độ ưu tiên cao hơn sẽ được cho phép chiếm bus trước, vì vậy cũng đã giải quyết được vấn đề này.

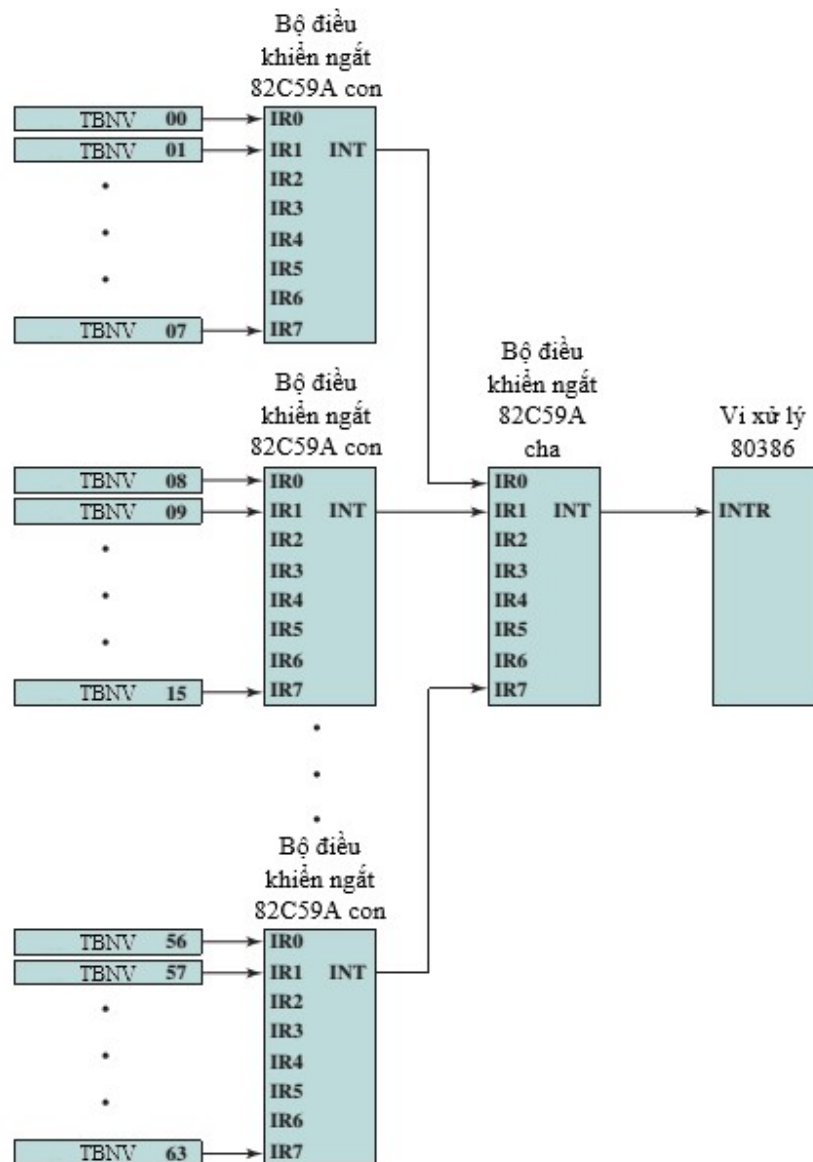
Chúng tôi giới thiệu hai cấu trúc ngắt trong thực tế là: Bộ điều khiển ngắt Intel 82C59A và giao diện thiết bị ngoại vi lập trình được Intel 82C55A

Bộ điều khiển ngắt Intel 82C59A

Chip Intel 80386 có một đường INTR (Yêu cầu Ngắt) và một đường INTA (Chấp nhận Ngắt). Để 80386 có thể giao tiếp với nhiều loại thiết bị ngoại vi và phân xử ưu tiên, nó thường được cấu hình với *bộ điều khiển ngắt 82C59A*. Nghĩa là, 80386 giao tiếp với các thiết bị ngoại vi thông qua 82C59A (Hình 6.11)

Một 82C59A có thể xử lý tối đa tám module. Nếu số lượng module cần điều khiển lớn hơn tám, người ta có thể thực hiện việc ghép nối tầng và có thể xử lý tới 64 module (như trong Hình

6.11). Nhiệm vụ duy nhất của 82C59A là quản lý các yêu cầu ngắt. Nó nhận các yêu cầu ngắt từ các module I/O, xác định ngắt nào có mức ưu tiên cao nhất, và sau đó báo hiệu bộ xử lý bằng cách thiết lập đường INTR. Bộ xử lý gửi tín hiệu thừa nhận ACK qua đường dây INTA. Khi nhận được ACK, 82C59A đặt thông tin vector thích hợp lên bus dữ liệu. Bộ xử lý tiến hành xử lý ngắt và liên lạc trực tiếp với module I/O để đọc hoặc ghi dữ liệu.



Hình 6.11 Bộ điều khiển Ngắt 82C59A

Chip 82C59A cho phép lập trình. 80386 xác định sơ đồ ưu tiên rồi thiết lập nó vào từ điều khiển trong 82C59A. Sơ đồ ưu tiên này có thể thiết lập theo một trong ba chế độ ngắt như sau:

- **Lồng nhau hoàn toàn:** Các yêu cầu ngắt được sắp xếp theo thứ tự ưu tiên từ 0 (IR0) đến 7 (IR7).

- **Xoay vòng:** Một số thiết bị ngắt có mức ưu tiên ngang nhau. Sau khi một thiết bị được phục vụ ngắt nó sẽ nhận mức ưu tiên thấp nhất trong nhóm.

- **Mặt nạ đặc biệt:** Cho phép bộ xử lý chặn các ngắt từ một số thiết bị nhất định

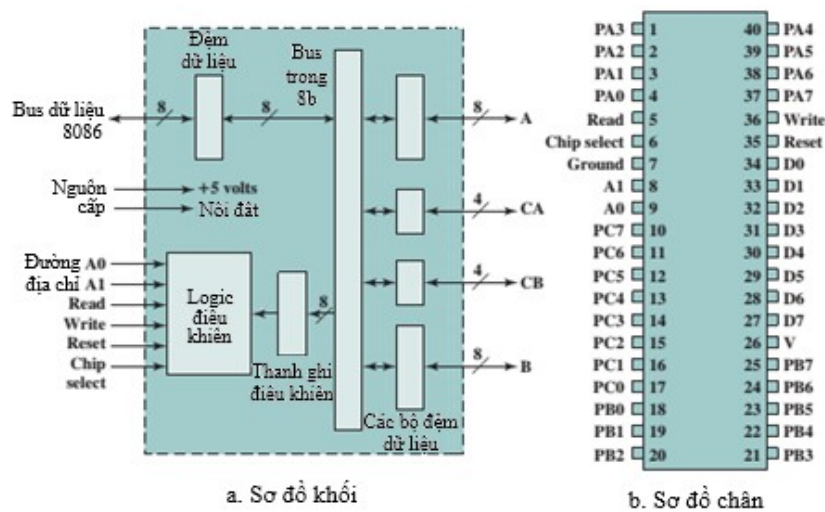
Giao diện ngoại vi có thể lập trình của Intel 82C55A

Chip Intel 82C55A sử dụng cho kỹ thuật I/O chương trình và I/O điều khiển ngắt. 82C55A là một module I/O đa nhiệm được thiết kế để sử dụng với vi xử lý Intel 80386. Hình 7.12 minh họa cấu trúc tổng quát và thông tin 40 chân của chip.

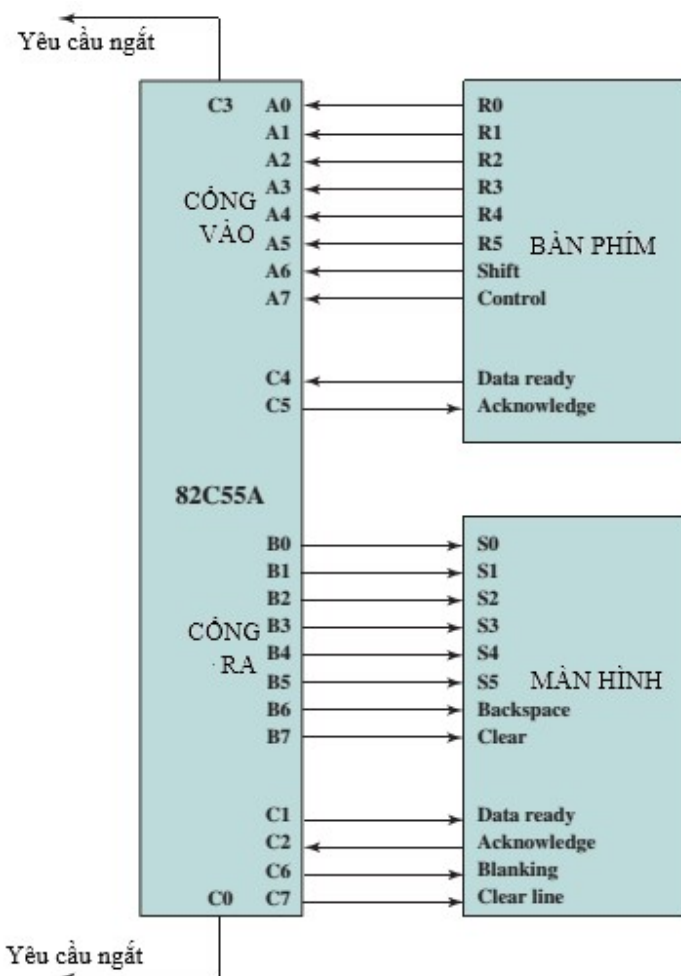
Phía bên phải của sơ đồ khối chip 82C55A gồm có: 24 đường vào/ra có thể lập trình được bằng cách 80386 thiết lập thanh ghi điều khiển để chỉ định đến một loạt các chế độ và cấu hình hoạt động. 24 đường được chia thành ba nhóm 8-bit (A, B, C). Mỗi nhóm có thể hoạt động như một cổng vào/ra 8-bit. Ngoài ra, nhóm C được chia thành hai nhóm 4-bit (C_A và C_B), hai nhóm này có thể được sử dụng kết hợp với các nhóm A và B. Với cách cấu hình này, các đường của nhóm C mang tín hiệu điều khiển và trạng thái.

Phía bên trái của sơ đồ khối là giao diện với bus 80386, gồm có: một bus dữ liệu hai chiều 8-bit (D0 đến D7), được sử dụng để truyền dữ liệu đến và đi tới các cổng vào/ra (A, B, C) hoặc để truyền thông tin điều khiển tới thanh ghi điều khiển; hai đường địa chỉ giúp xác định một trong ba cổng vào/ra hoặc thanh ghi điều khiển. Việc trao đổi dữ liệu sẽ xảy ra khi đường CHIP SELECT được thiết lập ở mức cao cùng với đường READ (Đọc) hoặc WRITE (Ghi). Đường RESET được sử dụng để khởi tạo lại module.

Như ta đã biết, bộ xử lý nạp các thông tin điều khiển vào thanh ghi điều khiển để định ra chế độ hoạt động hoặc định ra tín hiệu, nếu có. Trong chế độ 0, ba nhóm A, B, C hoạt động như ba cổng vào/ra 8-bit. Mỗi cổng có thể được chỉ định là đầu vào hoặc đầu ra. Ở một chế độ khác, các nhóm A và B được thiết lập là các cổng vào/ra còn các đường của nhóm C là các đường điều khiển cho A và B. Các tín hiệu điều khiển phục vụ cho hai mục đích chính: cơ chế "bắt tay" và yêu cầu ngắt. Cơ chế "bắt tay" là một cơ chế định thời đơn giản. Bên gửi sử dụng một đường điều khiển để gửi tín hiệu DATA READY báo cho bên nhận là dữ liệu đã có mặt trên đường dữ liệu. Bên nhận sử dụng một đường khác để gửi tín hiệu ACKNOWLEDGE (ACK) báo cho bên gửi biết rằng dữ liệu đã được đọc vào, bên gửi có thể loại bỏ các đường dữ liệu. Một đường được sử dụng làm đường Yêu cầu Ngắt.



Hình 6.12 Chip Intel 82C55A



Hình 6.13 Giao diện Bàn phím/Màn hình với 82C55A

Bởi vì 82C55A có thể lập trình được thông qua việc thiết lập thanh ghi điều khiển, nó còn được sử dụng để điều khiển một loạt các thiết bị ngoại vi đơn giản. Hình 7.10 minh họa việc sử dụng 82C55A để điều khiển Bàn phím/Màn hình. Bàn phím cung cấp 8 bit đầu vào. Hai trong số các bit này là SHIFT và CONTROL, tương ứng với phím Shift và Ctrl trên bàn phím. Việc tiếp nhận dữ liệu từ bàn phím rồi chuyển đến bus hệ thống của 82C55A trong trường hợp này là hoàn toàn trong suốt, tức là dữ liệu được nhận vào rồi chuyển vào bus, 82C55A không phải thực hiện thao tác gì khác. Hai đường điều khiển được sử dụng để thực hiện cơ chế “bắt tay”.

Màn hình cũng được kết nối với một cổng dữ liệu 8-bit. Tương tự như với bàn phím, hai bit có ý nghĩa đặc biệt và trong suốt đối với 82C55A. Ngoài hai đường điều khiển thực hiện “bắt tay”, hai đường điều khiển được sử dụng để thực hiện các chức năng điều khiển bổ sung.

6.3.3. Cơ chế DMA – Truy cập bộ nhớ trực tiếp

Nhược điểm của các phương pháp I/O chương trình và I/O điều khiển ngắt

I/O điều khiển ngắt mặc dù đã hiệu quả hơn rất nhiều so với I/O chương trình, tuy nhiên nó vẫn cần có sự tham gia của bộ xử lý vào việc trao đổi dữ liệu giữa bộ nhớ và các module I/O, bất cứ dữ liệu nào cũng phải đi qua bộ xử lý. Do đó các phương pháp này có hai nhược điểm sau:

1. Tốc độ trao đổi dữ liệu vào/ra bị hạn chế bởi tốc độ bộ xử lý thực hiện các thao tác kiểm tra và phục vụ thiết bị.
2. Bộ xử lý phải quản lý việc truyền I/O; với mỗi thao tác vào/ra dữ liệu, bộ xử lý phải thực thi một số chỉ thị (ví dụ như hình 6.5).

Có một sự cân bằng giữa hai nhược điểm này. Khi xét đến việc trao đổi khối dữ liệu. Nếu sử dụng I/O chương trình, bộ xử lý được dành hoàn toàn cho việc vào/ra dữ liệu nên việc trao đổi này diễn ra với tốc độ khá cao, tuy nhiên bộ xử lý lại không làm được việc gì khác. Với I/O điều khiển ngắt, bộ xử lý được giải phóng nhưng thời gian thực hiện một hoạt động truyền dữ liệu lại nhiều hơn. Cả hai phương pháp đều có ảnh hưởng nhất định đến hoạt động của bộ xử lý và tốc độ truyền I/O.

Khi cần trao đổi một khối lượng lớn dữ liệu, người ta đưa ra một kỹ thuật hiệu quả hơn, đó là: truy cập bộ nhớ trực tiếp (DMA).

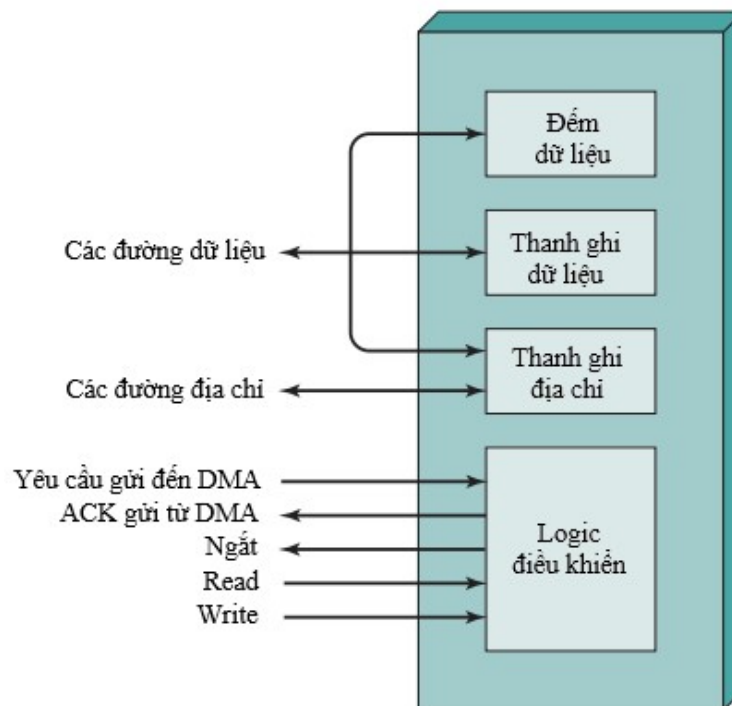
Chức năng DMA

DMA thường là một module bổ sung trên hệ thống bus. Module DMA (Hình 6.14) có khả năng bắt chước bộ xử lý, tức là tiếp nhận việc điều khiển hệ thống từ bộ xử lý. Nó cần phải thực hiện điều này để trao đổi dữ liệu với bộ nhớ qua bus hệ thống. Với mục đích này, module DMA chỉ sử dụng bus khi bộ xử lý không cần đến nó hoặc DMA phải buộc bộ xử lý phải tạm ngừng hoạt động (gọi là *trộm chu kỳ*). Kỹ thuật trộm chu kỳ thường phổ biến hơn.

Khi bộ xử lý muốn đọc hoặc ghi một khối (block) dữ liệu, nó sẽ đưa ra một lệnh cho module DMA bằng cách gửi tới module DMA các thông tin sau:

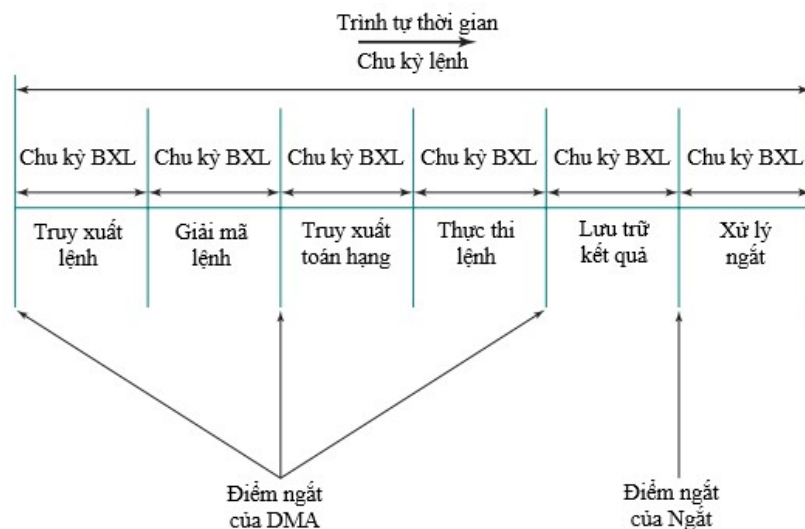
- Yêu cầu Đọc hoặc Ghi bằng cách sử dụng các đường điều khiển READ hoặc WRITE bộ xử lý và module DMA
- Địa chỉ của thiết bị I/O liên quan, gửi qua các đường dữ liệu
- Địa chỉ vị trí đầu tiên trong bộ nhớ để đọc hoặc ghi lên, truyền qua các đường dữ liệu và được lưu trữ lại bởi module DMA trong Thanh ghi địa chỉ (xem hình)
- Số lượng từ cần đọc hoặc ghi, truyền qua các đường dữ liệu và được lưu trữ trong thanh ghi Đếm dữ liệu

Sau đó, bộ xử lý tiếp tục thực hiện các công việc khác. Hoạt động vào/ra được ủy quyền hoàn toàn cho module DMA. Module DMA chuyển toàn bộ khối dữ liệu giữa bộ nhớ và thiết bị ngoại vi mà không cần sự tham gia của bộ xử lý. Khi quá trình trao đổi hoàn tất, module DMA gửi một tín hiệu ngắt tới bộ xử lý. Do đó, bộ xử lý chỉ tham gia vào thời điểm bắt đầu và kết thúc quá trình trao đổi (Hình 6.4c).



Hình 6.14 Sơ đồ khối DMA

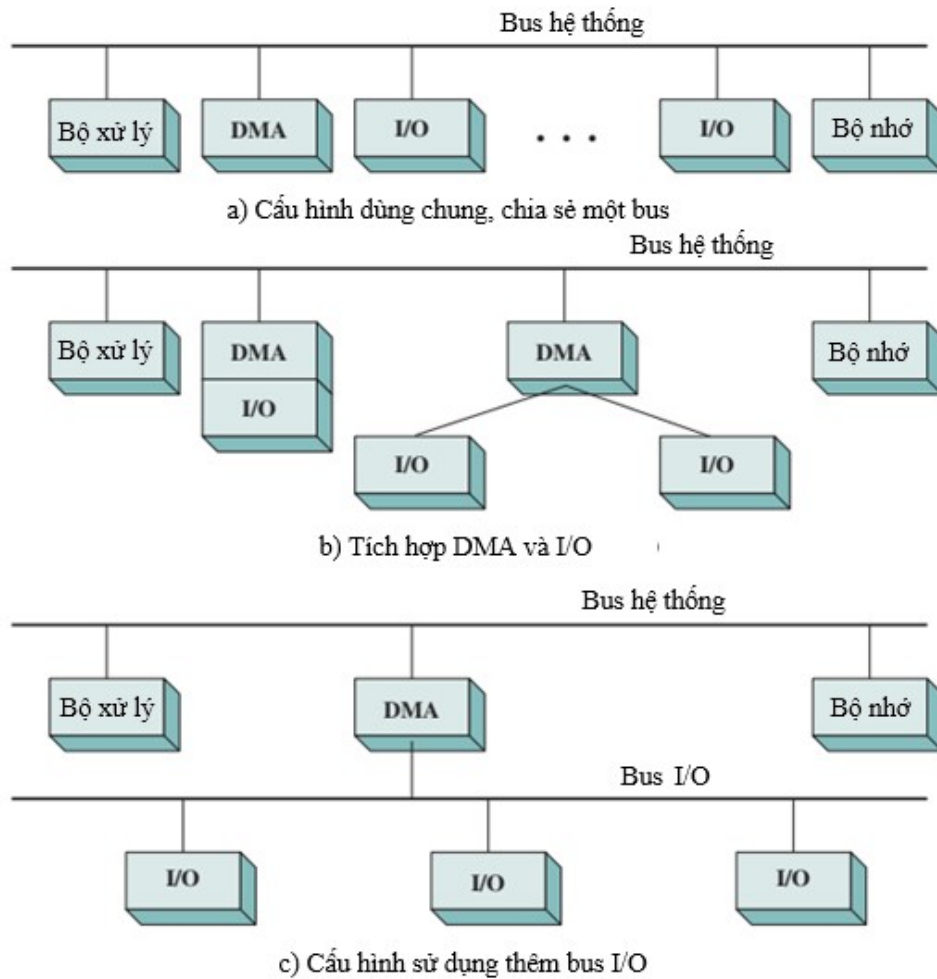
Như đã nói ở trên, khi DMA cần chiếm bus để truyền, nó sẽ trộm chu kỳ bus, Hình 6.15 chỉ ra các thời điểm trong chu kỳ lệnh mà bộ xử lý bị treo (trong lúc DMA sử dụng bus). Trong mỗi trường hợp, bộ xử lý bị treo ngay trước khi nó cần sử dụng bus. Module DMA chuyển một từ sau đó trả lại điều khiển cho bộ xử lý. Lưu ý rằng đây không phải là ngắt; bộ xử lý không cần lưu trữ ngữ cảnh hoặc bất cứ thông tin gì. Đơn giản là bộ xử lý chỉ dừng lại một chu kỳ bus. Về mặt hiệu năng của bộ xử lý thì nó sẽ hoạt động chậm hơn. Tuy nhiên, với việc trao đổi vào/ra nhiều từ, cơ chế DMA hiệu quả hơn rất nhiều so với I/O điều khiển ngắt hoặc I/O chương trình.



Hình 6.15 Điểm ngắt của DMA và của Ngắt trong chu kỳ lệnh

Cơ chế DMA có thể được cấu hình theo nhiều cách khác nhau. Hình 6.16 là sơ đồ một số cấu hình DMA. Hình 6.16a, tất cả các module chia sẻ chung hệ thống bus. Module DMA đại diện cho bộ xử lý sử dụng I/O chương trình để trao đổi dữ liệu giữa bộ nhớ và module I/O. Cấu hình này mặc dù không tốn kém nhưng rõ ràng là không hiệu quả. Giống như I/O chương trình do bộ xử lý điều khiển, mỗi lần truyền một từ sẽ tiêu tốn hai chu kỳ bus.

Số lượng chu kỳ bus có thể được giảm bớt bằng cách tích hợp DMA và các module I/O. Như trong Hình 6.16b, module DMA kết nối trực tiếp với một hoặc nhiều module I/O mà không đi qua bus hệ thống. Logic DMA có thể là một phần của module I/O hoặc có thể là một module riêng biệt điều khiển một hoặc nhiều module I/O. Phương thức này có thể được cải tiến thêm một bước nữa bằng cách kết nối các module I/O với module DMA qua một bus I/O (Hình 6.16c). Cấu hình này giảm số giao diện ngoại vi của module DMA thành một và cho phép khả năng mở rộng (tăng thêm số lượng module I/O thì chỉ cần kết nối với bus I/O). Trong cả hai trường hợp hình 6.16b và c, module DMA chỉ chiếm bus hệ thống khi cần trao đổi dữ liệu với bộ nhớ, như vậy chỉ mất một chu kỳ bus cho mỗi từ được truyền.



Hình 6.16 Các cấu hình DMA

Bộ điều khiển Intel 8237A DMA

Bộ điều khiển Intel 8237A DMA giao tiếp với họ vi xử lý 80x86 và bộ nhớ DRAM, cung cấp cơ chế DMA (Hình 7.17). Khi module DMA cần sử dụng bus hệ thống (bus dữ liệu, địa chỉ và điều khiển) để truyền dữ liệu, nó sẽ gửi một tín hiệu HOLD cho bộ xử lý. Bộ xử lý trả lời bằng tín hiệu HLDA (chấp nhận Hold) để cho module DMA biết rằng nó có thể sử dụng bus. Xét ví dụ: nếu module DMA là truyền một khối dữ liệu từ bộ nhớ sang đĩa từ, nó sẽ làm như sau:

1. Thiết bị ngoại vi (bộ điều khiển đĩa từ) sẽ yêu cầu dịch vụ của DMA bằng cách thiết lập đường DREQ (Yêu cầu DMA) ở mức cao.
2. DMA sẽ đặt tín hiệu HOLD qua đường HRQ vào bộ xử lý để báo cho CPU biết rằng nó cần sử dụng bus.
3. CPU sẽ kết thúc chu kỳ bus hiện tại (không nhất thiết là phải kết thúc chu kỳ lệnh hiện tại) và đáp ứng yêu cầu của DMA bằng cách thiết lập đường HDLA lên mức cao, khi đó DMA

8237 biết rằng nó có thể sử dụng bus để thực hiện nhiệm vụ. Tín hiệu HOLD sẽ luôn giữ ở mức cao trong cả quá trình DMA trao đổi dữ liệu.

4. DMA thiết lập tín hiệu DACK (Chấp nhận DMA) để báo cho thiết bị ngoại vi rằng nó sẽ bắt đầu truyền dữ liệu.

5. DMA bắt đầu điều khiển việc truyền dữ liệu từ bộ nhớ sang thiết bị ngoại vi bằng cách đặt địa chỉ byte đầu tiên của khối dữ liệu lên bus địa chỉ và kích hoạt tín hiệu MEMR (tín hiệu điều khiển Đọc bộ nhớ), để một byte từ bộ nhớ vào bus dữ liệu; sau đó nó kích hoạt IOW (Ghi I/O) để ghi vào thiết bị ngoại vi. Tiếp đó, DMA giảm thanh ghi đếm dữ liệu và tăng thanh ghi địa chỉ để tiếp tục thao tác truyền các từ tiếp theo. Quá trình này lặp lại cho đến khi Thanh ghi đếm dữ liệu về không.

6. Sau khi DMA kết thúc công việc, nó sẽ huỷ bỏ HRQ để báo cho CPU biết rằng nó có thể lấy lại quyền điều khiển bus.

Trong khi DMA sử dụng bus để truyền dữ liệu, bộ xử lý không hoạt động (bị treo tạm thời). Tương tự như vậy, khi bộ xử lý đang sử dụng bus, DMA không hoạt động. 8237 DMA được biết đến như là một *bộ điều khiển fly-by DMA*, có nghĩa là dữ liệu được di chuyển từ vị trí này đến vị trí khác không đi qua chip DMA và không được lưu trữ trong chip DMA. Do đó, DMA chỉ có thể điều khiển việc truyền dữ liệu giữa cổng I/O và bộ nhớ, chứ không phải giữa hai cổng I/O hoặc hai vị trí bộ nhớ. Tuy nhiên, như được giải thích sau đây, chip DMA có thể thực hiện truyền dữ liệu từ bộ nhớ đến bộ nhớ thông qua thanh ghi.

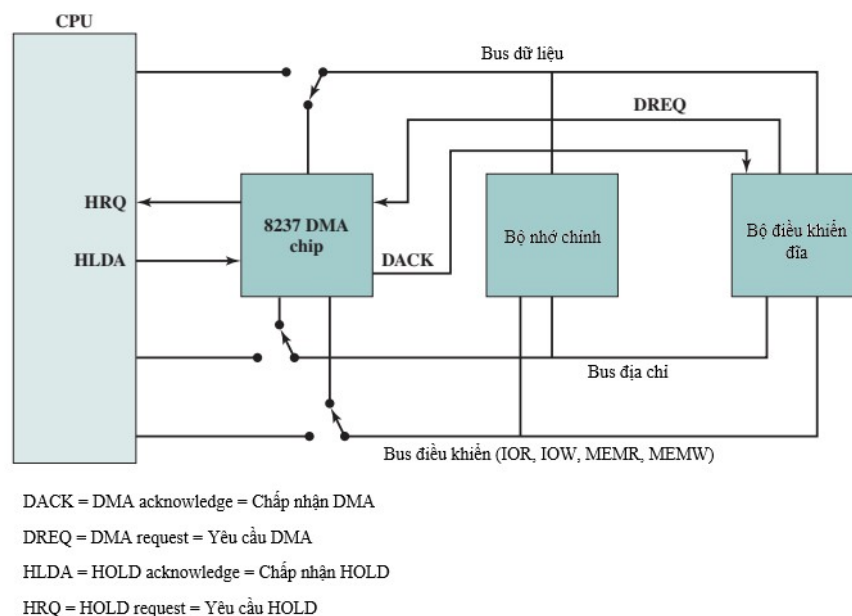
8237 có bốn kênh DMA có thể được lập trình độc lập và một kênh có thể hoạt động bất kỳ lúc nào. Các kênh này được đánh số 0, 1, 2 và 3.

8237 có 5 thanh ghi điều khiển/lệnh để lập trình và điều khiển hoạt động DMA qua một trong các kênh của nó như sau:

- **Lệnh:** Bộ xử lý nạp thanh ghi này để điều khiển hoạt động của DMA. D0 cho phép hoặc không cho phép việc chuyển dữ liệu giữa hai vị trí bộ nhớ, trong đó kênh 0 được sử dụng để truyền một byte vào thanh ghi dữ liệu tạm thời của 8237, kênh 1 được sử dụng để truyền byte từ thanh ghi vào vị trí bộ nhớ (bên nhận). Khi bộ nhớ được kích hoạt, D1 có thể được thiết lập để vô hiệu hóa việc tăng/giảm trên kênh 0 để ghi một giá trị cố định vào một khối bộ nhớ. D2 có tác dụng bật hoặc tắt DMA.
- **Trạng thái:** Bộ xử lý đọc thanh ghi này để xác định trạng thái DMA. Các bit D0-D3 được sử dụng để cho biết liệu các kênh 0-3 đã đạt được TC (số cuối cùng). Các bit D4-D7 được bộ xử lý sử dụng để xác định xem liệu có kênh nào có yêu cầu DMA đang chờ được xử lý.
- **Chế độ:** Bộ xử lý thiết lập thanh ghi này để xác định chế độ hoạt động của DMA. Các bit D0 và D1 được sử dụng để chọn một kênh. Các bit khác chọn các chế độ hoạt động khác nhau

cho kênh đó. Các bit D2 và D3 quy định việc truyền dữ liệu từ thiết bị ngoài sang bộ nhớ (Ghi) hoặc từ bộ nhớ sang I/O (Đọc) hoặc hoạt động kiểm tra. Nếu D4 được thiết lập, thanh ghi địa chỉ bộ nhớ và thanh ghi đếm được nạp lại với giá trị ban đầu của chúng sau khi kết thúc một hoạt động DMA. Các bit D6 và D7 quy định các chế độ (mode) của 8237. Trong chế độ đơn mode, duy nhất một byte dữ liệu được truyền. Chế độ truyền nhóm được sử dụng cho hoạt động truyền một khối dữ liệu, chế độ yêu cầu cho phép kết thúc sớm một hoạt động truyền. Chế độ nối tầng cho phép nhiều 8237 được nối tầng với nhau để mở rộng số kênh

- Single Mask: Bộ vi xử lý thiết lập thanh ghi này. Các bit D0 và D1 để chọn kênh. Bit D2 quy định việc thiết lập hoặc không thiết lập bit mặt nạ cho kênh đó. Thanh ghi Lệnh có thể được sử dụng để vô hiệu hóa toàn bộ chip DMA, còn thanh ghi Single Mask có thể được lập trình để cho phép hoặc cho không phép một kênh cụ thể.
- All Mask: thanh ghi này tương tự như thanh ghi Single Mask như nó thực hiện việc thiết lập hoặc không thiết lập mặt nạ với tất cả bốn kênh trong những hoạt động Ghi.



Hình 6.17 DMA 8237

Ngoài ra, 8237A có tám thanh ghi dữ liệu: một thanh ghi địa chỉ bộ nhớ và một thanh ghi đếm dữ liệu cho mỗi kênh. Bộ vi xử lý thiết lập những thanh ghi này để chỉ định đến một vị trí của bộ nhớ chính bị ảnh hưởng khi thực hiện truyền dữ liệu.

6.4. KÊNH VÀO/RA VÀ BỘ XỬ LÝ VÀO/RA

6.4.1. Quá trình phát triển của các chức năng I/O

Khi hệ thống máy tính đã trở nên phát triển, mô hình máy tính ngày càng phức tạp và tinh vi.

Điều này thể hiện rõ nhất ở chức năng của các module vào/ra. Các bước phát triển của chức năng vào/ra có thể tóm tắt như sau:

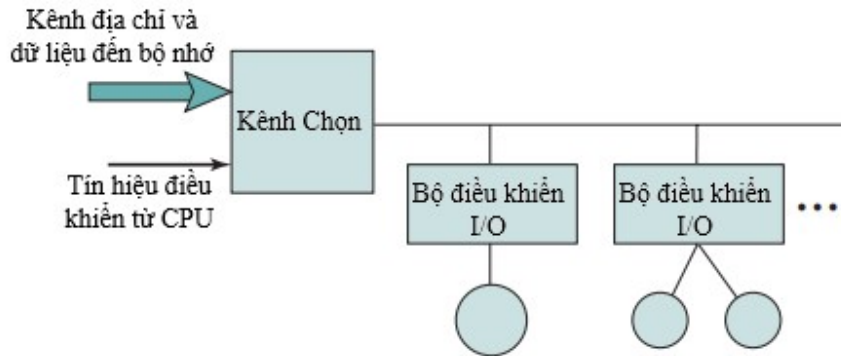
1. Ban đầu, CPU trực tiếp điều khiển một thiết bị ngoại vi. Điều này được thấy trong các vi điều khiển đơn giản.
2. Sử dụng các bộ điều khiển hoặc module I/O. CPU sử dụng cơ chế I/O chương trình. Từ thời điểm này, CPU bắt đầu tách ra khỏi việc điều khiển chi tiết thiết bị ngoại vi.
3. Cấu trúc tương tự như trong giai đoạn 2, tuy nhiên sử dụng cơ chế I/O điều khiển ngắt. CPU không phải tốn nhiều thời gian chờ đợi một hoạt động I/O được thực hiện, do đó tăng hiệu năng của hệ thống.
4. Module I/O được truy cập trực tiếp tới bộ nhớ thông qua cơ chế DMA. Nó có thể thực hiện việc trao đổi một khối dữ liệu đến hoặc đi từ bộ nhớ mà không cần có sự tham gia của CPU ngoại trừ thời điểm bắt đầu và kết thúc quá trình truyền.
5. Module I/O được cải tiến để trở thành một bộ xử lý I/O chuyên thực thi các hoạt động I/O. CPU khi này chỉ cần đưa ra chỉ thị cho bộ xử lý I/O thực hiện một chương trình I/O trong bộ nhớ. Bộ xử lý I/O sẽ nạp và thực hiện các chỉ thị này mà không cần sự can thiệp của CPU. Điều này cho phép CPU có thể chỉ thị một loạt các hoạt động vào/ra và chỉ bị ngắt khi nào toàn bộ công việc đã được thực hiện.
6. Module I/O có bộ nhớ cục bộ của riêng nó, thực chất nó có thể được coi là một máy tính với khả năng nhất định. Với kiến trúc này, một loạt các thiết bị ngoại vi có thể được điều khiển với sự tham gia tối thiểu của CPU. Bộ xử lý I/O sẽ thực hiện hầu hết các nhiệm vụ liên quan đến việc điều khiển các thiết bị đầu cuối.

Với quá trình phát triển này, điều dễ nhận thấy là sự tham gia của CPU vào các tác vụ vào/ra ngày càng ít, do đó giúp cải thiện hiệu suất CPU. Ở giai đoạn 5,6, một bước thay đổi lớn đối với khái niệm module I/O, module I/O từ đây có khả năng thực hiện các chương trình. Ở giai đoạn 5, module I/O sẽ được gọi kênh I/O. Còn với bước 6, người ta sử dụng thuật ngữ bộ xử lý I/O. Tuy nhiên, thông thường cả hai thuật ngữ này đều được sử dụng ngang nhau trong cả hai giai đoạn. Ở phần sau chúng tôi sẽ dùng chung thuật ngữ Kênh I/O.

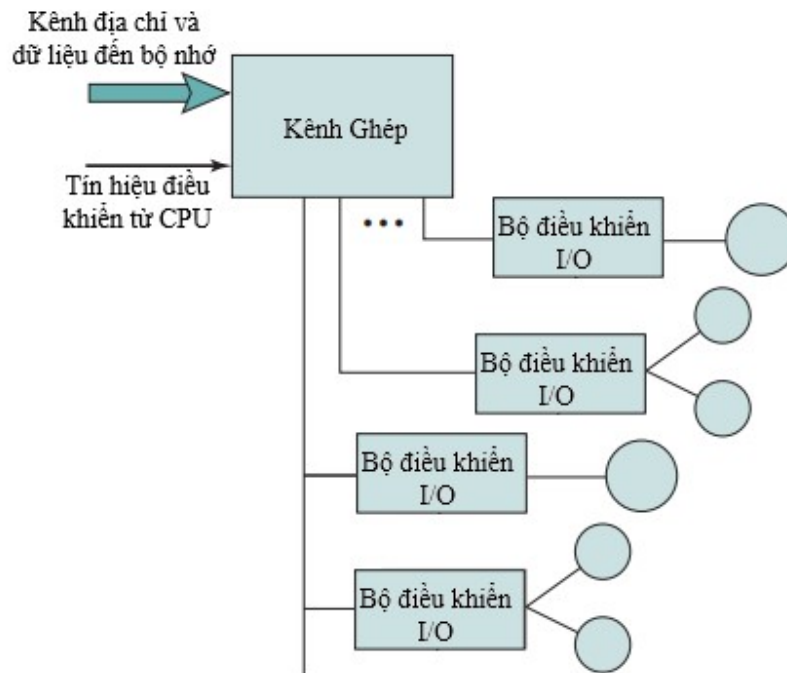
6.4.2. Đặc điểm của các Kênh I/O

Khái niệm Kênh I/O là mở rộng của khái niệm DMA. Kênh I/O có khả năng thực hiện các lệnh vào/ra, cho phép điều khiển hoàn toàn các hoạt động vào/ra. Trong một hệ thống máy tính với các thiết bị như vậy, CPU không thực hiện các lệnh vào/ra. Những chỉ thị này được lưu trữ trong bộ nhớ chính và được bộ xử lý của Kênh I/O thực thi. Do đó, khi có một hoạt động vào/ra, CPU chỉ thị cho Kênh I/O để thực thi một chương trình trong bộ nhớ. Chương trình này sẽ định ra một hoặc nhiều thiết bị, một hoặc nhiều vùng bộ nhớ để lưu trữ, các mức ưu tiên và hoạt động

sẽ được thực hiện trong trường hợp có các điều kiện lỗi nhất định. Kênh I/O sẽ theo các chỉ thị này và điều khiển việc truyền dữ liệu.



a. Kênh chọn



b. Kênh ghép

Hình 6.18 Kiến trúc Kênh I/O

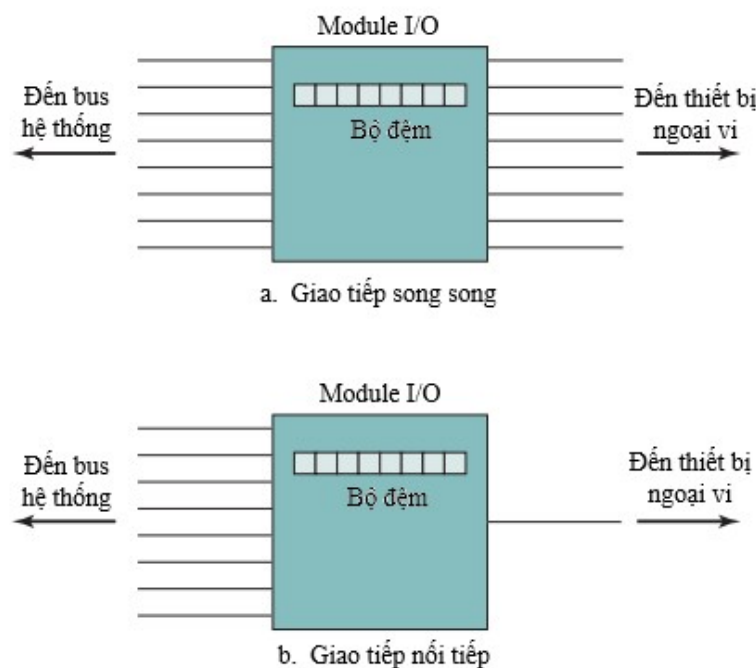
Có hai loại Kênh I/O phổ biến như trong Hình 7.18. **Kênh Chọn** điều khiển nhiều thiết bị tốc độ cao và được dành riêng cho việc truyền dữ liệu với một trong những thiết bị đó tại bất cứ thời điểm nào. Do đó, Kênh I/O lựa chọn một thiết bị và tác động đến việc truyền dữ liệu. Một thiết bị hoặc một nhóm thiết bị được điều khiển bởi một bộ điều khiển hoặc module I/O. Do đó, các Kênh I/O sẽ thay cho CPU trong việc kiểm soát, điều khiển các module I/O. **Kênh Ghép kênh**

có thể xử lý vào/ra nhiều thiết bị cùng một lúc. Với các thiết bị tốc độ thấp, *Bộ ghép kênh* theo byte sẽ chấp nhận hoặc truyền các ký tự nhanh nhất có thể cho nhiều thiết bị. Ví dụ: ba luồng ký tự từ ba thiết bị có tốc độ khác nhau như sau: $A_1A_2A_3A_4 \dots$, $B_1B_2B_3B_4 \dots$ và $C_1C_2C_3C_4 \dots$ có thể được truyền như sau $A_1B_1C_1A_2C_2A_3B_2C_3A_4 \dots$. Với các thiết bị tốc độ cao, một khối dữ liệu của nó sẽ được ghép nối xen kẽ với một khối dữ liệu từ các thiết bị khác.

6.5. GIAO TIẾP NGOÀI

6.5.1. Các loại giao tiếp

Giao tiếp giữa một thiết bị ngoại vi và một module I/O phải được điều chỉnh phù hợp với tính chất và hoạt động của thiết bị ngoại vi. Đặc điểm chính của giao tiếp này là nó có thể là giao tiếp nối tiếp hoặc giao tiếp song song (Hình 7.19). Trong giao tiếp song song, có nhiều đường kết nối giữa module I/O và thiết bị ngoại vi, các bit được truyền đồng thời giống như việc truyền một từ trên bus dữ liệu. Trong một giao tiếp nối tiếp, chỉ có một đường truyền dữ liệu, mỗi bit được truyền một thời điểm. Giao tiếp song song thường được sử dụng cho các thiết bị ngoại vi tốc độ cao như băng từ, đĩa từ, còn giao tiếp nối tiếp thường được sử dụng cho máy in và các thiết bị đầu cuối khác. Ngày nay, với sự phát triển của các giao tiếp nối tiếp tốc độ cao, giao tiếp song song đang trở nên ít phổ biến hơn.



Hình 6.19 Giao tiếp song song và nối tiếp

Trong cả hai trường hợp, để thực hiện việc trao đổi dữ liệu, module I/O phải đối thoại với thiết bị ngoại vi. Ví dụ với trường hợp thực hiện hoạt động ghi như sau:

1. Module I/O gửi tín hiệu điều khiển yêu cầu cho phép gửi dữ liệu.
2. Thiết bị ngoại vi chấp nhận yêu cầu.
3. Module I/O truyền dữ liệu (một từ hoặc một khối dữ liệu tùy thuộc vào thiết bị ngoại vi).
4. Thiết bị ngoại vi nhận dữ liệu và gửi xác nhận việc đó.

Hoạt động đọc cũng được thực hiện với các thao tác tương tự.

Một thành phần quan trọng của module I/O là bộ đệm, nó có tác dụng lưu trữ dữ liệu tạm thời để cân bằng sự chênh lệch tốc độ giữa bus hệ thống và các đường kết nối ngoài.

7.5.2. Cấu hình Điểm – Điểm và Đa điểm

Kết nối giữa một module I/O và các thiết bị ngoài có thể là kết nối điểm-điểm hoặc kết nối đa điểm. Với giao tiếp điểm-điểm, module I/O được nối với mỗi thiết bị ngoại vi qua một đường riêng. Trên các hệ thống nhỏ (các máy tính cá nhân hoặc máy trạm), các kết nối điểm - điểm điển hình gồm có: kết nối với bàn phím, máy in và modem ngoài.

Giao tiếp đa điểm ngày càng phổ biến và trở nên quan trọng hơn. Giao tiếp này thường được sử dụng để hỗ trợ các thiết bị lưu trữ ngoài (như ổ đĩa và băng từ) và các thiết bị đa phương tiện (CD-ROM, video, audio). Các giao tiếp đa điểm này là các bus mở rộng và có cùng logic hoạt động giống bus.

CÂU HỎI CỦA CHƯƠNG

1. Liệt kê ba loại thiết bị ngoại vi hoặc thiết bị ngoài.
2. Bảng IRA là bảng gì?
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?
7. Bộ xử lý sẽ làm gì trong khi một module DMA chiếm và duy trì quyền kiểm soát bus?