

Chương 2

Cấp logic số

Máy tính được xây dựng từ những chip mạch tích hợp (Integrated Circuit - IC) hay còn gọi là mạch logic. Các mạch logic được xây dựng từ các phần tử chuyên mạch rất nhỏ gọi là cổng (gate). Các cổng thông dụng nhất là AND, OR, NAND, NOR, và NOT.

Tuỳ theo mức độ tích hợp, các mạch logic có thể được phân loại như sau:

1. Mạch SSI (Small Scale Integrated): ít hơn 10 cổng.
2. Mạch MSI (Medium Scale Integrated): từ 10 đến 100 cổng.
3. Mạch LSI (Large Scale Integrated): từ hơn 100 đến 100000 cổng.
4. Mạch VLSI (Very Large Scale Integrated): lớn hơn 100000 cổng.

Kiến trúc cơ bản về các mạch logic được sử dụng ở trong máy tính được trình bày trong môn học kiến trúc máy tính, ở đây không đề cập lại. Trong chương này, tập trung vào nghiên cứu kiến trúc của chip vi xử lý và nguyên lý hoạt động của nó với bộ nhớ và các thiết bị ngoại vi.

1. Các chip vi xử lý và các bus

Trong mục này, chúng ta sẽ xem xét các bộ vi xử lý từ góc nhìn cấp logic số. Vì bộ vi xử lý giao tiếp với các thành phần khác thông qua các bus, cho nên thiết kế bus có ảnh hưởng trực tiếp đến sự hoạt động của nó. Tiếp theo đó chúng ta sẽ tìm hiểu về các loại bus của máy tính cũng ở góc nhìn cấp logic số. Cuối cùng sẽ đưa ra một ví dụ về bộ vi xử lý và bus của máy tính.

1.1. Bộ vi xử lý

Bộ vi xử lý là một chip có chức năng như một CPU (Central Processing Unit), chip này chứa cả 3 đơn vị chức năng: Số học - Logic, Điều khiển và Bộ nhớ. Vì vậy, đôi khi người ta cũng gọi chip này là CPU. CPU là vi mạch có chức năng hoạt động rất linh hoạt và công dụng nhất trong các loại vi mạch số, nó thực hiện các phép tính logic, số học và điều khiển toàn bộ hoạt động của hệ thống máy tính. Nó nhận các lệnh (dữ liệu vào) từ bộ nhớ, giải mã và thực hiện lệnh.

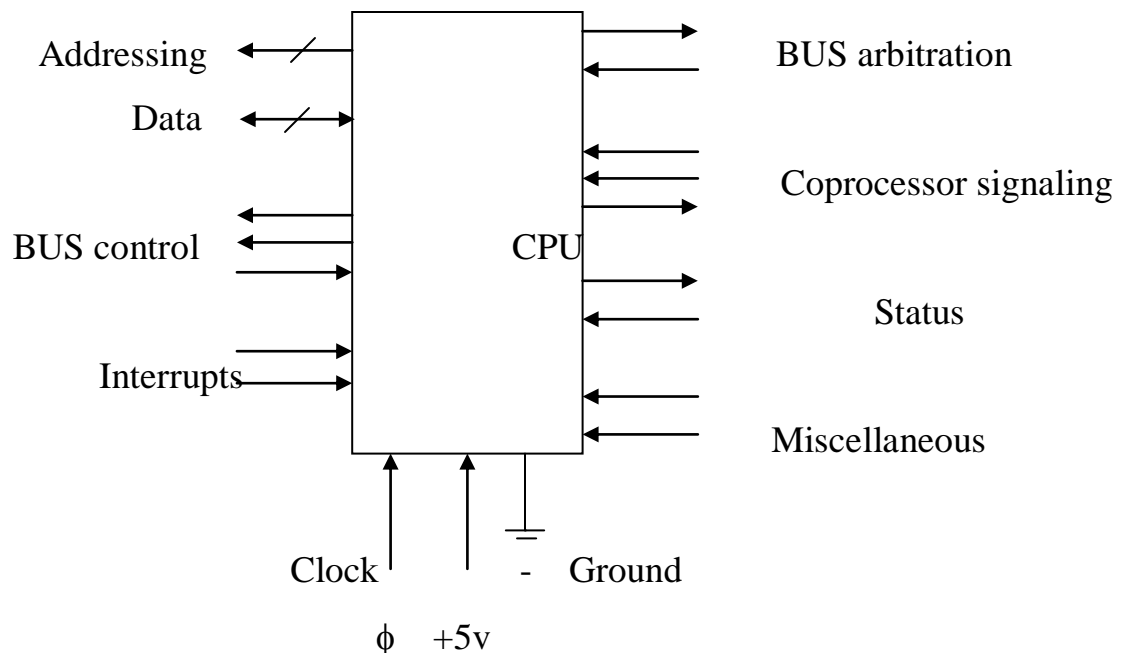
Chip CPU thường có hàng trăm chân, mọi sự liên hệ với thế giới bên ngoài đều phải thông qua các chân này. Một số chân để gửi thông tin từ CPU ra ngoài, một số chân để nhận thông tin từ ngoài vào bên trong CPU, cũng có một số chân thực hiện cả

hai. Thông qua việc tìm hiểu chức năng của các chân CPU, chúng ta sẽ nắm được cách CPU giao tiếp với các thành phần khác của hệ thống máy tính ở cấp logic số như thế nào.

Các chân của CPU có thể được chia thành 3 nhóm: địa chỉ, dữ liệu và điều khiển. Các chân này được nối với các chân tương ứng của các chip nhớ và các chip I/O thông qua các đường dây chạy song song được gọi là bus.

Để nhận một lệnh từ bộ nhớ đệm vào, trước tiên CPU đặt địa chỉ của lệnh đó lên các chân địa chỉ, sau đó nó truyền tín hiệu đọc một từ lên các chân điều khiển. Bộ nhớ đáp lại bằng cách đưa từ mà CPU cần lên đường dữ liệu nối tới các chân dữ liệu của CPU, sau đó phát tín hiệu báo cho CPU là công việc đã được thực hiện. Khi CPU nhận được tín hiệu này nó sẽ nhận từ được yêu cầu và thực hiện lệnh. Lệnh này lại có thể đòi hỏi đọc hoặc ghi dữ liệu, quá trình lại tiếp tục tương tự như vậy.

Cần lưu ý là trên một số chân của CPU, tín hiệu ứng với +5v kích hoạt một công việc nào đó, nhưng một số chân khác lại là -5v, để tránh nhầm lẫn ta dùng thuật ngữ tích cực / không tích cực chứ không nói tín hiệu cao hay thấp. Người ta qui ước với các tín hiệu tích cực ở mức điện áp thấp thì tên của tín hiệu sẽ thêm một gạch ngang ở phía trên.



Hình 2.1: Các chân ra của một CPU điển hình

- **Các chân địa chỉ và chân dữ liệu của CPU**

Hai tham số quan trọng của CPU là số chân địa chỉ và số chân dữ liệu. Một chip có m chân địa chỉ, có thể đánh địa chỉ được 2^m vị trí nhớ. Trong thực tế m thường là 16, 20, 24, 32 và 36. Một chip có n chân dữ liệu có thể đọc hoặc ghi một từ n bit trong một thao tác đọc hoặc ghi. Trong thực tế n thường có các giá trị 8, 16, 32 và 64. Một CPU có 8 chân dữ liệu sẽ cần 4 thao tác để đọc một từ 32 bit, khi đó CPU có 32 chân dữ liệu chỉ cần một thao tác. Tức là chip có 32 chân dữ liệu sẽ xử lý nhanh hơn và tất nhiên giá thành sẽ đắt hơn.

- **Các chân điều khiển của CPU**

Ngoài các chân địa chỉ và dữ liệu, CPU còn có các chân điều khiển, chúng điều khiển dòng dữ liệu và phân định thời gian cho các tín hiệu vào/ra CPU. Các chân điều khiển của CPU có thể chia thành các nhóm sau:

1. Điều khiển bus (bus Control): là nhóm các chân truyền các tín hiệu đi ra từ CPU tới các bus, báo cho các đơn vị khác (bộ nhớ, thiết bị vào ra) biết CPU muốn đọc/ghi dữ liệu hay làm công việc gì khác.

2. Ngắt (Interrupts): là nhóm các chân truyền các tín hiệu từ ngoài đi vào CPU yêu cầu CPU phục vụ. Trong hầu hết các hệ thống, chính các CPU có thể ra lệnh cho các thiết bị vào/ra bắt đầu thực hiện một thao tác nào đó.

3. Trọng tài BUS (bus arbitration): là nhóm các chân truyền các tín hiệu điều hoà sự lưu thông trên các bus để tránh gây xung đột.

4. Tín hiệu đồng xử lý (Coprocesor signaling): có nhiều trường hợp chip CPU được thiết kế để có thể cùng làm việc với bộ đồng xử lý. Các chip Coprocessor thường là các chip tính toán số dấu phẩy động hay các chip đồ hoạ, ... Để việc trao đổi dữ liệu giữa CPU với bộ đồng xử lý được thuận lợi CPU thường có các chân dành cho việc trao đổi tín hiệu với Coprocessor.

5. Trạng thái (Status): nhóm các chân này thông báo trạng thái của CPU hoặc nhận thông tin trạng thái từ các thiết bị khác.

6. Các tín hiệu khác (Miscellaneous): các CPU khác nhau có thể có các tín hiệu khác nhau, thí dụ tín hiệu reset CPU,...

CPU ngày càng được sử dụng nhiều trong thực tế: trong đồng hồ, lò vi sóng, máy giặt, các rô bốt, các thiết bị kiểm soát nhiệt độ trong các cao ốc, bộ không chế nhiên liệu trong ô tô, trong các động cơ máy bay và đặc biệt là trong máy tính điện tử.

- **Bộ nhớ trong của CPU**

Bộ nhớ trong của CPU là bộ nhớ nhỏ, tốc độ cao được sử dụng để chức tạm thời các lệnh, dữ liệu, kết quả trung gian và những thông tin điều khiển nhất định. Bộ nhớ này bao gồm các thanh ghi và có thể cả bộ nhớ ROM, RAM bên trong CPU.

Mỗi thanh ghi của CPU thường có những chức năng nhất định, thí dụ: thanh ghi PC (Program Counter), trở đến lệnh tiếp theo sẽ được thực hiện; thanh ghi IR (Instruction Register) chứa lệnh đang được thực hiện,...

Bộ nhớ ROM bên trong CPU thường chứa các vi chương trình, đó là tập các vi lệnh, nhằm thực hiện các lệnh của CPU.

Bộ nhớ RAM bên trong CPU (cache) thường chứa các lệnh sắp được thực hiện hoặc các số liệu được sử dụng thường xuyên.

1.2. Các bus của máy tính

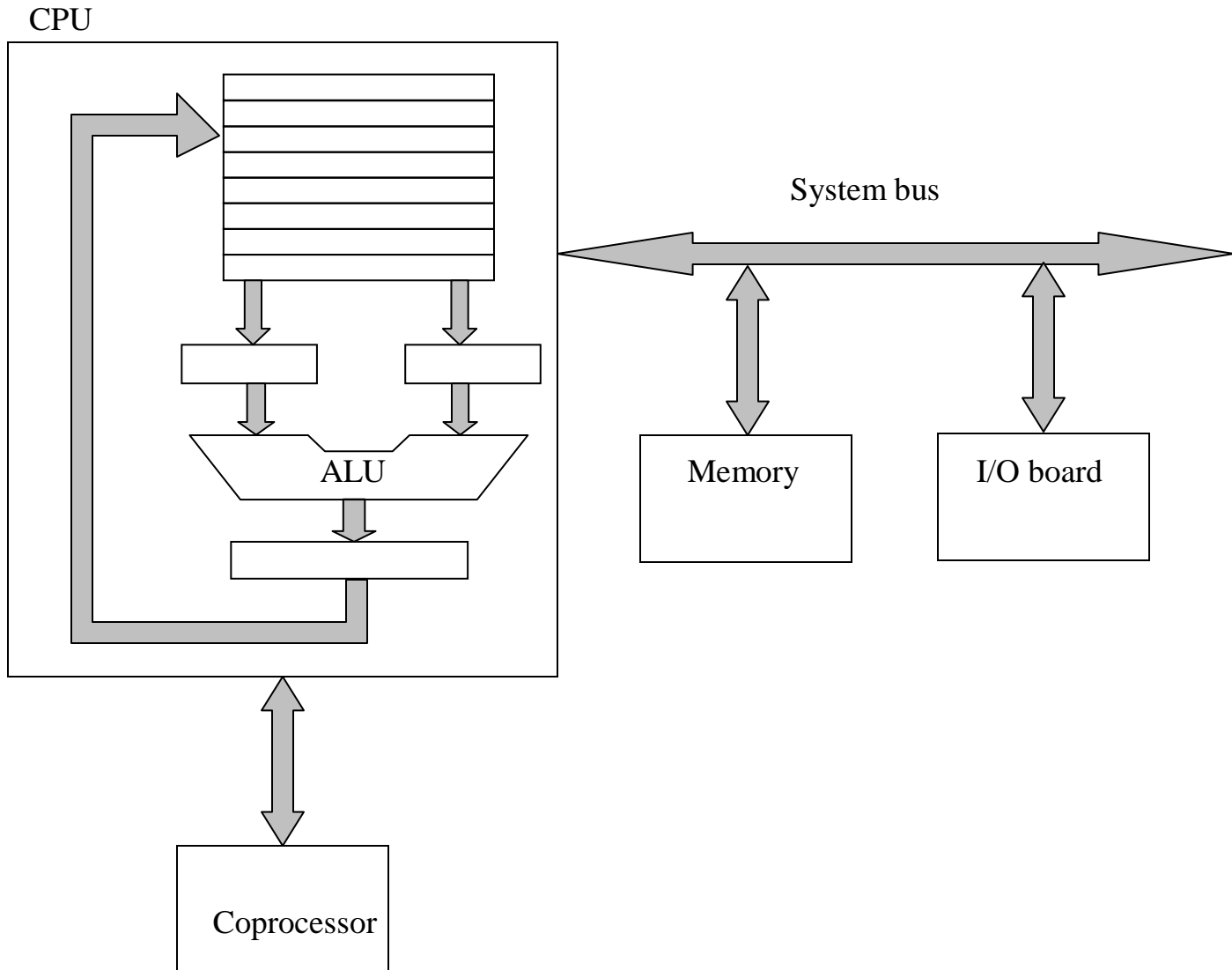
Bus là đường truyền tín hiệu điện chung nối các thiết bị khác nhau trong một hệ thống máy tính. Một ví dụ điển hình về bus đó là system bus gồm từ 50 đến 100 đường truyền dẫn được gắn chặt với Mainboard với các đầu nối để có thể cắm vào đó các bảng mạch mở rộng hoặc các mô đun nhớ. Ngoài ra cũng có những loại bus được danhf cho mục đích đặc biệt như là: kết nối CPU với bộ đồng xử lí, hay kết nối CPU với bộ nhớ cục bộ, ... Trong chính CPU cũng có một số bus để kết nối các thành phần bên trong CPU với nhau.

Người thiết kế CPU có thể tùy ý lựa chọn loại bus bên trong CPU, còn với các bus liên hệ với bên ngoài CPU, cần phải xác định rõ các quy tắc làm việc mà tất cả các thiết bị kết nối với bus đều phải tuân theo. Các qui tắc này còn gọi là nghi thức bus (bus protocol). Ngoài ra, còn có các chỉ tiêu kỹ thuật về điện và cơ khí để các bảng mạch mở rộng có thể cắm vào các khe cắm trên mainboard.

• Sự làm việc của các bus

Thường có nhiều thiết bị nối với bus, một số thiết bị là tích cực và có thể đòi hỏi truyền thông tin trên bus, trong khi đó lại có các thiết bị thụ động chờ các yêu cầu từ các thiết bị khác, các thiết bị tích cực được gọi là chủ bus (bus master), các thiết bị thụ động gọi là tớ bus (bus slave).

Ví dụ: Khi CPU ra lệnh cho bộ điều khiển đĩa đọc/ghi một khối dữ liệu thì CPU là chủ bus, còn bộ điều khiển là tớ bus. Tuy nhiên, khi bộ điều khiển đĩa ra lệnh cho bộ nhớ nhận dữ liệu mà nó đọc từ đĩa thì nó lại là chủ bus.



Hình 2.2: Các bus của máy tính

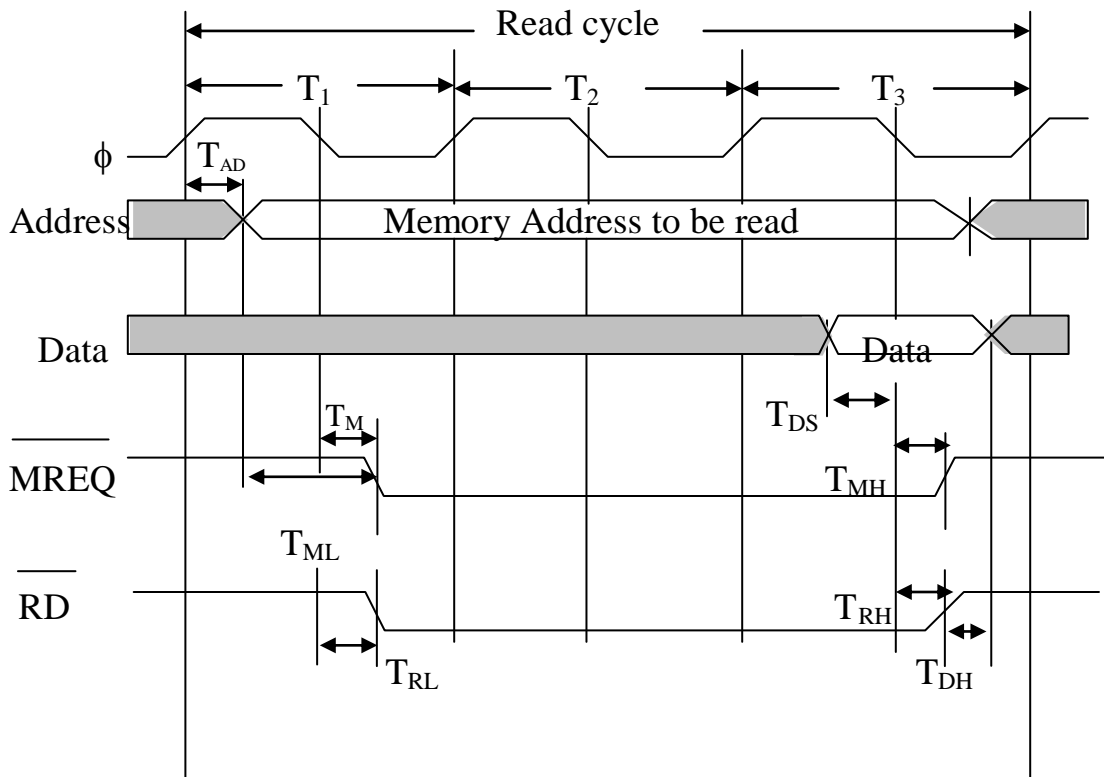
- **Bus Driver và bus Receiver**

Tín hiệu điện mà các thiết bị trong máy tính phát ra thường không đủ mạnh để điều khiển được bus, nhất là khi bus khá dài và có nhiều thiết bị nối với nó. Chính vì vậy mà hầu hết các bus master được nối với bus thông qua một chip gọi là bus driver, đó là một bộ khuếch đại tín hiệu số. Tương tự như vậy, hầu hết các bus slave được nối với bus thông qua bus receiver. Đối với các thiết bị khi thì đóng vai trò master khi thì đóng vai trò slave người ta sử dụng một chip kết hợp gọi là bus transceiver. Các chip này đóng vai trò ghép nối và thường là các thiết bị 3 trạng thái, cho phép nó có thể ở trạng thái thứ 3 - hở mạch.

Giống như CPU, bus có các đường địa chỉ, số liệu và điều khiển. Tuy nhiên không nhất thiết phải có ánh xạ 1-1 giữa các tín hiệu ở các chân ra của CPU và các đường dây của bus. Ngoài ra, còn các vấn đề khác liên quan đến thiết kế bus như là đồng bộ bus, trọng tải bus, xử lý ngắt. Chúng ta sẽ lần lượt xem xét chúng ở các mục tiếp theo.

1.3. Bus đồng bộ (Synchronous bus)

Bus đồng bộ có một đường dây được điều khiển bởi một bộ dao động thạch anh. Tín hiệu trên đường dây này có dạng sóng vuông, với tần số hàng trăm Mhz (1 Mhz = 1000000 chu kỳ / giây) mọi hoạt động của bus đều xảy ra trong một số nguyên lần của chu kỳ này và được gọi là chu kỳ bus.



Hình 2.3: Hoạt động của một bus đồng bộ

Hình 2.3 là giản đồ thời gian của một bus đồng bộ với tần số đồng hồ là 4 Mhz (4000.000 chu kỳ/giây), 1 giây = 1000.000.000 ns ---> 1 chu kỳ mất 250 ns).

Giả sử việc đọc 1 byte từ bộ nhớ chiếm 3 chu kỳ dao động (750 ns), tương ứng với T_1 , T_2 , T_3 trên hình vẽ. Vì tất cả các tín hiệu điện thay đổi mức không phải là tức thời, nên trên hình vẽ có các sườn xung, ta giả sử các sườn xung kéo dài 10 ns.

+ T_1 bắt đầu bằng sườn lên của tín hiệu đồng hồ, trong một phần thời gian của T_1 , CPU hoàn thành việc đặt địa chỉ của byte cần đọc lên đường địa chỉ (bus địa chỉ). Sau khi đường tín hiệu địa chỉ được thiết lập giá trị mới, các tín hiệu $\overline{\text{MREQ}}$ (Memory REQuest) và RD (Read) được đặt ở mức tích cực (tích cực ở mức điện áp thấp).

+ T_2 là khoảng thời gian cần thiết để Memory giải mã địa chỉ và đưa dữ liệu lên bus dữ liệu.

+ Tại sườn xuống của T_3 , CPU nhận dữ liệu trên bus dữ liệu, chứa vào thanh ghi bên trong CPU. Sau đó CPU đảo các tín hiệu $\overline{\text{MREQ}}$ và RD thành không tích cực.

Như vậy thao tác đọc đã hoàn thành và tại chu kỳ bus tiếp theo CPU có thể thực hiện một thao tác khác.

• Giải thích chi tiết:

$T_{AD} \leq 110$ ns, nghĩa là nhà sản xuất chip phải đảm bảo rằng trong một chu kỳ đọc toán hạng từ bộ nhớ, CPU phải đưa được địa chỉ của toán hạng lên bus địa chỉ không chậm hơn 110 ns tính từ thời điểm giữa sườn lên của T_1 .

$T_{DS} \geq 50$ ns, có nghĩa là dữ liệu đã ổn định trên bus dữ liệu ít nhất là 50 ns trước thời điểm giữa sườn xuống của T_3 . Thời gian này đảm bảo cho CPU đọc dữ liệu tin cậy.

Khoảng thời gian bắt buộc đối với T_{AD} và T_{DS} nói lên rằng, trong trường hợp tồi nhất bộ nhớ chỉ còn có $250+250+125-110-50 = 465$ ns từ thời điểm có tín hiệu địa chỉ cho tới khi nó sinh ra dữ liệu trên bus dữ liệu. Nếu bộ nhớ không có khả năng đáp ứng đủ nhanh, nó cần phát tín hiệu đòi chờ $\overline{\text{WAIT}}$ trước sườn xuống của T_2 . Thao tác này đưa thêm các trạng thái chờ (wait states) vào chu kỳ đọc bộ nhớ cho tới khi bộ nhớ hoàn tất công việc, sau đó nó đảo tín hiệu $\overline{\text{WAIT}}$ thành không tích cực.

$T_{ML} \geq 60$ ns có nghĩa rằng tín hiệu địa chỉ phải được thiết lập trước tín hiệu $\overline{\text{MREQ}}$ ít nhất là 60 ns. Khoảng thời gian này là quan trọng nếu tín hiệu $\overline{\text{MREQ}}$ điều khiển sự tạo ra tín hiệu chọn chip, bởi vì một số chip nhớ yêu cầu thời gian thiết lập địa chỉ phải hoàn tất trước tín hiệu chọn chip. Như vậy, với bus này, người thiết kế máy tính sẽ không thể chọn chip nhớ với thời gian thiết lập là 75 ns.

$T_M \leq 85$ ns, $T_{RL} \leq 85$ ns, các giá trị bắt buộc đối với 2 đại lượng này có nghĩa là cả 2 tín hiệu $\overline{\text{MREQ}}$ và $\overline{\text{RD}}$ sẽ là tích cực trong khoảng thời gian 85 ns tính từ thời điểm xuống của xung đồng hồ T_1 . Trong trường hợp tồi nhất, chip nhớ chỉ có $250 + 250 - 85 - 50 = 365$ ns sau khi 2 tín hiệu trên là tích cực để đưa dữ liệu lên bus dữ liệu.

$T_{MH} \leq 85 \text{ ns}$, $T_{RH} \leq 85 \text{ ns}$, hai đại lượng này cho biết cần bao nhiêu thời gian để các tín hiệu \overline{MREQ} và \overline{RD} được đảo về mức không tích cực sau khi dữ liệu đã được CPU nhận.

$T_{DH} \geq 0 \text{ ns}$, đại lượng này cho biết bộ nhớ cần phải giữ data trên bus bao lâu sau tín hiệu \overline{RD} đã đảo về mức không tích cực. Như vậy, bộ nhớ có thể loại bỏ dữ liệu trên bus dữ liệu ngay sau khi \overline{RD} được đảo.

Giản đồ trên được đơn giản rất nhiều so với thực tế.

Ký hiệu	Tham số	Min (nsec)	Max (nsec)
T_{AD}	Thời gian trễ của tín hiệu địa chỉ		110
T_{ML}	Thời gian địa chỉ ổn định trước tín hiệu \overline{MREQ}	60	
T_M	Thời gian trễ của \overline{MREQ} so với sườn xuống của T1		85
T_{RL}	Thời gian trễ của \overline{RD} so với sườn xuống của T1		85
T_{DS}	Thời gian thiết lập dữ liệu trước sườn xuống của T3	50	
T_{MH}	Thời gian trễ của \overline{MREQ} so với sườn xuống của T3		85
T_{RH}	Thời gian trễ của \overline{RD} so với sườn xuống của T3		85
T_{DH}	Thời gian lưu trữ dữ liệu từ lúc đảo tín hiệu \overline{RD}	0	

• Block Transfer

Ngoài các chu kỳ đọc / ghi, một số bus đồng bộ còn hỗ trợ việc truyền dữ liệu theo khối. Khi một thao tác đọc khối bắt đầu, bus master phải báo cho bus slave biết có bao nhiêu byte cần được truyền đi bằng cách đưa con số đó lên đường dữ liệu trong thời gian của chu kỳ T_1 . Sau đó, thay vì truyền đi 1 byte, bus slave đưa ra từng byte trong thời gian mỗi chu kỳ đồng hồ cho tới khi đủ số byte được thông báo. Như vậy, khi đọc dữ liệu theo khối, n byte dữ liệu cần n+2 chu kỳ đồng hồ chứ không phải 3n chu kỳ.

Một cách khác làm cho tốc độ truyền dữ liệu nhanh hơn là làm cho các chu kỳ ngắn lại. Tuy nhiên, việc này gặp khó khăn về mặt kỹ thuật, các tín hiệu truyền trên các đường dây khác nhau không phải luôn có cùng vận tốc, dẫn đến một hiệu ứng là bus skew (lệch bus). Vì vậy, thời gian chu kỳ phải đủ dài để tránh việc những khoảng thời gian được số hoá (Digital) trở thành liên tục (Analog).

Khi nghiên cứu về bus cũng cần quan tâm đến vấn đề tín hiệu tích cực là mức thấp hay mức cao, điều này tùy thuộc vào người thiết kế xác định tín hiệu nào là thuận lợi hơn.

1.4. Bus không đồng bộ (Asynchronous bus)

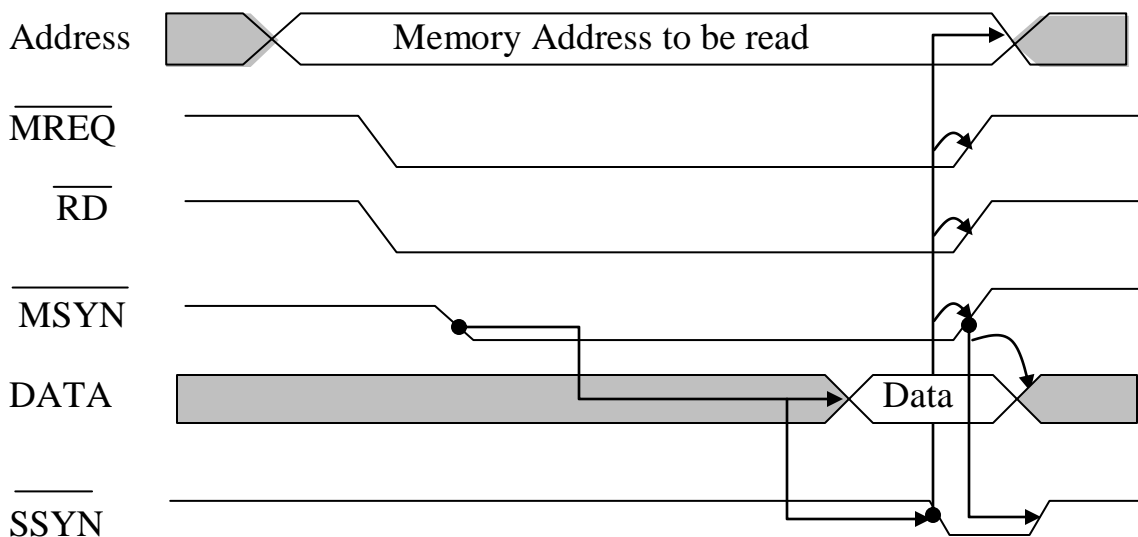
Làm việc với các bus đồng bộ sẽ dễ dàng hơn do nó được định thời một cách cố định, tuy nhiên nó có nhược điểm sau đây:

- + Mọi công việc đều được tiến hành trong khoảng thời gian là bội số của chu kỳ xung đồng hồ. Như vậy, nếu một thao tác nào đó của CPU và bộ nhớ có thể hoàn thành trong 3.1 chu kỳ, thì nó sẽ phải kéo dài thành 4 chu kỳ.

- + Khi đã chọn chu kỳ bus, bộ nhớ và các thiết bị I/O đã được thiết kế cho chu kỳ bus này thì khó có tận dụng được những tiến bộ của công nghệ. Ví dụ, công nghệ mới đưa ra CPU và chip nhớ hoạt động với thời gian chu kỳ bus là 100 ns chứ không phải 250 ns, thì chúng vẫn phải chạy với tốc độ như các CPU và các chip nhớ loại cũ.

- + Nếu có nhiều thiết bị khác nhau nối với một bus thì cần phải đặt bus hoạt động phù hợp với thiết bị có tốc độ thấp nhất.

Bus không đồng bộ ra đời chính là để khắc phục những nhược điểm nói trên. Bus không đồng bộ không sử dụng xung đồng hồ chủ, chu kỳ của nó có thể kéo dài tùy ý và có thể là khác nhau đối với các cặp thiết bị khác nhau. Hình 2.4 là sơ đồ minh họa sự hoạt động của bus không đồng bộ.



Hình 2.4: Hoạt động của bus không đồng bộ

Hoạt động đọc dữ liệu từ bộ nhớ của CPU có thể được thực hiện như sau:

+ Sau khi CPU đặt địa chỉ của byte cần đọc lên bus địa chỉ, đảo các tín hiệu $\overline{\text{MREQ}}$ và $\overline{\text{RD}}$ thành tích cực, nó đảo tín hiệu đồng bộ chủ $\overline{\text{MSYN}}$ (Master SYNchronization) thành tích cực.

+ Khi nhận được tín hiệu $\overline{\text{MSYN}}$, memory giải mã địa chỉ và đưa dữ liệu lên bus dữ liệu với tốc độ nhanh nhất có thể được, khi hoàn thành nó đảo tín hiệu đồng bộ phụ thuộc $\overline{\text{SSYN}}$ (Slave SYNchronization) thành tích cực.

+ Khi CPU nhận được tín hiệu $\overline{\text{SSYN}}$ này, nó biết dữ liệu đã sẵn sàng (trên bus dữ liệu), nó thực hiện việc chốt dữ liệu lại, sau đó nó xoá địa chỉ trên bus địa chỉ, đảo các tín hiệu $\overline{\text{MREQ}}$, $\overline{\text{RD}}$, $\overline{\text{MSYN}}$ về mức không tích cực.

+ Khi memory nhận được sự đảo tín hiệu $\overline{\text{MSYN}}$, nó biết rằng CPU đã nhận được dữ liệu và đảo tín hiệu $\overline{\text{SSYN}}$ thành không tích cực. Một chu kỳ bus đã kết thúc, hệ thống trở lại trạng thái ban đầu, mọi tín hiệu đều là không tích cực, chờ đợi một chu kỳ bus mới.

Trên giản đồ thời gian của bus không đồng bộ, chúng ta thấy các đường mũi tên để thể hiện nguyên nhân và kết quả. Việc đưa $\overline{\text{MSYN}}$ lên mức tích cực dẫn đến việc truyền dữ liệu ra bus dữ liệu và đồng thời cũng dẫn đến memory đặt tín hiệu $\overline{\text{SSYN}}$ lên mức tích cực. Tín hiệu $\overline{\text{SSYN}}$ tích cực lại gây ra sự đảo mức của các đường địa chỉ, $\overline{\text{MREQ}}$, $\overline{\text{RD}}$, $\overline{\text{MSYN}}$. Cuối cùng sự đảo mức của $\overline{\text{MSYN}}$ lại gây ra sự đảo mức tín hiệu $\overline{\text{SSYN}}$ và kết thúc một chu kỳ đọc.

- **Full Handshake**

Các tín hiệu phối hợp với nhau như vậy được gọi là sự bắt tay hoàn toàn. **Full Handshake** gồm 4 sự kiện chủ yếu sau đây:

1. $\overline{\text{MSYN}}$ được đặt tích cực.
2. $\overline{\text{SSYN}}$ được đặt tích cực để đáp lại
3. $\overline{\text{MSYN}}$ được đảo để đáp lại.
4. $\overline{\text{SSYN}}$ được đảo để đáp lại sự đảo tín hiệu $\overline{\text{MSYN}}$

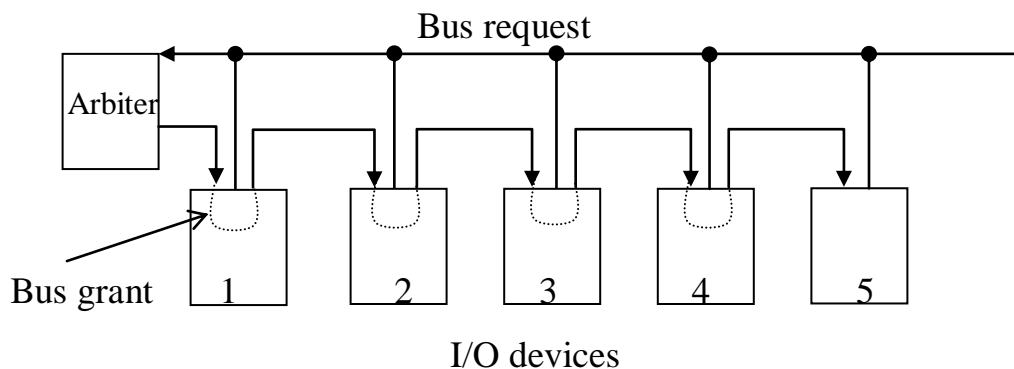
Chúng ta nhận thấy **Full Handshake** là độc lập thời gian, mỗi sự kiện được gây ra bởi một sự kiện trước đó chứ không phải bởi một xung đồng hồ. Nếu một cặp Master - Slave nào hoạt động chậm thì cặp kế tiếp không hề bị ảnh hưởng.

Chúng ta thấy ưu điểm của bus không đồng bộ là rõ ràng, song trong thực tế phần lớn các bus đang được sử dụng là đồng bộ. Lý do chủ yếu là các hệ thống sử dụng bus đồng bộ dễ thiết kế hơn. CPU chỉ cần nâng các mức tín hiệu cần thiết lên tích cực là các chip nhớ đáp lại ngay, không cần tín hiệu phản hồi. Chỉ cần các chip được chọn phù hợp là mọi hoạt động đều trôi chảy, không cần phải bắt tay.

1.5. Trọng tài bus (bus arbitration)

Trong hệ thống máy tính không phải chỉ có CPU làm chủ bus, các chip I/O cũng có thể làm chủ bus khi đọc hoặc ghi bộ nhớ và để gọi ngắt. Ngoài ra, các bộ đồng xử lý cũng có thể làm chủ bus. Như vậy, nảy sinh vấn đề cùng một thời gian có nhiều thiết bị muốn trở thành chủ bus. Do vậy, cần phải có một cơ chế trọng tài để tránh sự tranh chấp này. Cơ chế trọng tài bus có thể là tập trung hoặc không tập trung, chúng ta sẽ lần lượt xem xét hai loại trọng tài bus này.

1.5.1 Trọng tài bus tập trung



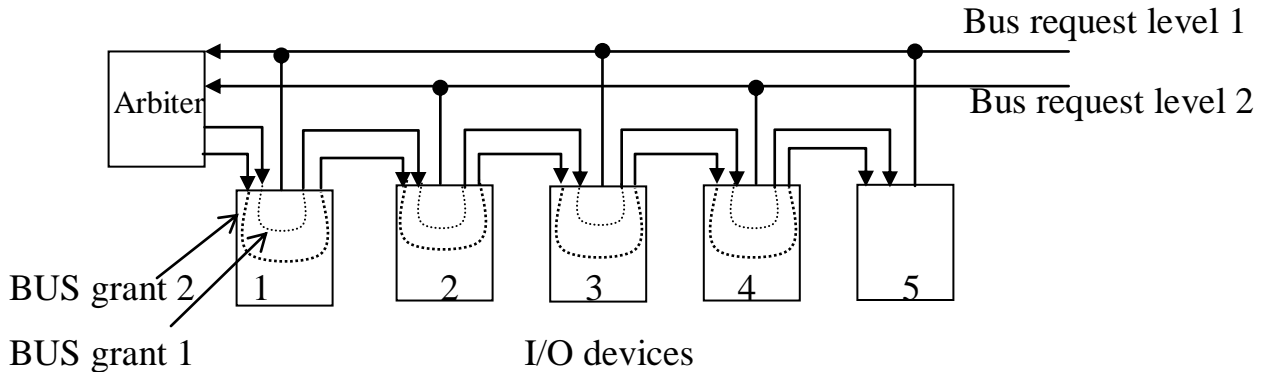
Hình 2.5: Trọng tài bus tập trung một mức

Nhiều bộ vi xử lý có đơn vị trọng tài nằm ngay bên trong CPU, nhưng trong các máy tính mini, đơn vị trọng tài nằm ngoài chip CPU.

Theo cơ chế tập trung, trọng tài bus chỉ biết là có yêu cầu chiếm dụng bus hay không, chứ không biết có bao nhiêu đơn vị muốn chiếm dụng bus. Khi nhận được yêu cầu bus, trọng tài bus sẽ phát ra một tín hiệu cho phép trên đường dây bus grant (chuyển tín hiệu này thành mức tích cực). Đường dây này nối qua tất cả các thiết bị I/O theo kiểu nối tiếp. Khi thiết bị nằm gần trọng tài nhất nhận tín hiệu cho phép nó sẽ kiểm tra xem có phải chính nó đã phát ra yêu cầu bus hay không. Nếu đúng, nó chiếm dụng bus và không truyền tín hiệu cho phép sang thiết bị kế tiếp, nếu không nó truyền tiếp tín hiệu cho phép trên đường dây tới thiết bị kế tiếp. Sự việc lại xảy ra giống như

với thiết bị trước nó, quá trình cứ tiếp diễn như vậy cho tới khi có một thiết bị chiếm dụng bus. Trong trường hợp này, thứ tự ưu tiên phụ thuộc vào thiết bị nào ở gần đơn vị trọng tài bus hơn.

Một số loại bus có nhiều mức độ ưu tiên khác nhau, mỗi mức ưu tiên có một đường dây yêu cầu bus và một đường dây cho chiếm dụng bus (Bus grant).



Hình 2.6: Trọng tài bus tập trung nhiều mức

Khi có nhiều mức ưu tiên cùng yêu cầu, trọng tài bus sẽ phát tín hiệu cho phép cho mức ưu tiên cao nhất. Trong các thiết bị có cùng mức ưu tiên, thiết bị gần trọng tài hơn sẽ có mức ưu tiên cao hơn.

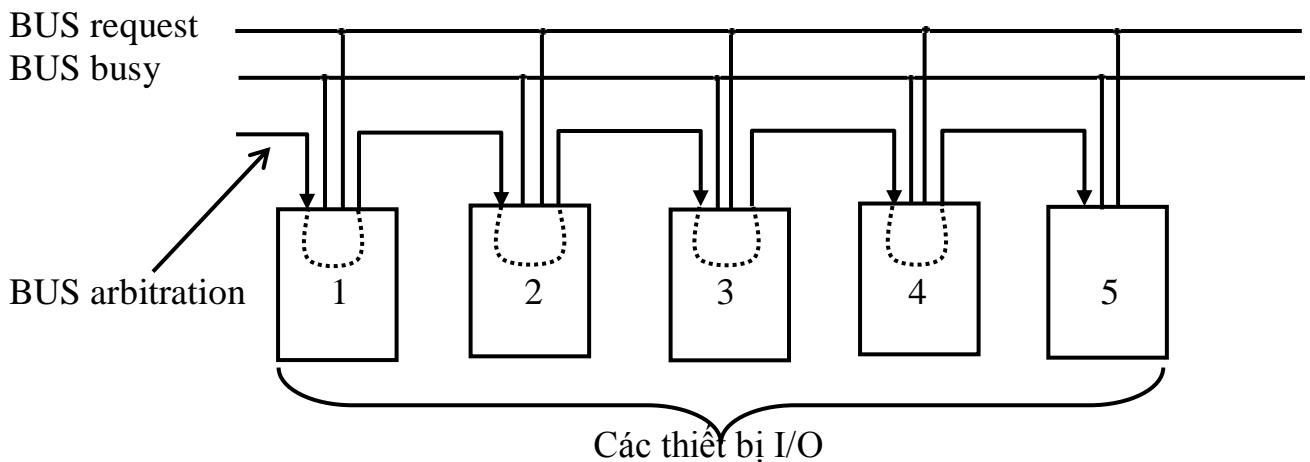
Một số trọng tài bus có đường dây thứ 3 nối tới các thiết bị gọi là đường dây ACK (acknowledgement), khi một thiết bị chiếm dụng bus, nó phát tín hiệu tích cực trên đường dây ACK, khi đó các tín hiệu trên các đường dây request và grant được đảo xuống mức không tích cực. Như vậy các thiết bị khác có thể yêu cầu chiếm dụng bus trong khi một thiết bị đang chiếm dụng bus. Khi thao tác hoàn thành, thiết bị vừa chiếm dụng bus đảo tín hiệu ACK thành không tích cực. Thiết bị yêu cầu bus có thể bắt đầu ngay khi nhận được tín hiệu này. Cách làm như vậy sẽ làm tăng hiệu quả sử dụng của bus, nhưng cấu trúc của các thiết bị cũng phải phức tạp hơn.

1.5.2 Trọng tài bus không tập trung

Trọng tài bus không tập trung không sử dụng đơn vị trọng tài riêng, như vậy làm giảm giá thành phần cứng. Hình 2.7 là một ví dụ về trọng tài bus không tập trung. Dù cho số thiết bị I/O nhiều hay ít, cũng chỉ có 3 đường dây:

1. Đường dây yêu cầu bus.
2. Đường dây báo bận.
3. Đường dây trọng tài.

Khi không có thiết bị nào yêu cầu bus, đường dây trọng tài được đặt ở mức tích cực. Khi một thiết bị muốn chiếm dụng bus, đầu tiên nó kiểm tra xem bus có rỗi không (đường dây báo bận không tích cực), tín hiệu trọng tài bus có ở mức tích cực không. Nếu bus rỗi và tín hiệu trọng tài tích cực, nó đảo đường trọng tài bus thành không tích cực và trở thành chủ bus, đồng thời nó đặt đường báo bận lên mức tích cực và bắt đầu truyền dữ liệu.



Hình 2.7: Trong tài BUS không tập trung

2. Ví dụ về các chip vi xử lý

Trong phần này chúng ta sẽ khảo sát các chip vi xử lý của Intel (vi chip của Intel chiếm tới trên 90% thị trường chip của thế giới).

2.1. 8088 của Intel

- **8088 có một số đặc trưng kỹ thuật sau:**

- + Độ rộng thanh ghi 16 bit.
- + Độ rộng bus dữ liệu 8 bit.
- + Độ rộng bus địa chỉ 20 bit.
- + Bộ đồng xử lý 8087 là chip độc lập.
- + 8088 có 2 chế độ hoạt động:

1. Chế độ tối thiểu: dùng trong các hệ thống nhỏ với vài thiết bị ngoại vi, chỉ có một bộ xử lý. Ví dụ 8088 sử dụng trong bộ điều khiển máy giặt.
2. Chế độ tối đa: dùng trong các hệ máy tính cá nhân, là chế độ có nhiều hơn một bộ xử lý. Trong tài liệu này chúng ta chỉ đề cập đến chế độ tối đa.

GND	<input type="checkbox"/>	1	40	<input type="checkbox"/>	VCC
A14	<input type="checkbox"/>	2	39	<input type="checkbox"/>	A15
A13	<input type="checkbox"/>	3	38	<input type="checkbox"/>	A16/S3
A12	<input type="checkbox"/>	4	37	<input type="checkbox"/>	A17/S4
A11	<input type="checkbox"/>	5	36	<input type="checkbox"/>	A18/S5
A10	<input type="checkbox"/>	6	35	<input type="checkbox"/>	A19/S6
A9	<input type="checkbox"/>	7	34	<input type="checkbox"/>	
A8	<input type="checkbox"/>	8	33	<input type="checkbox"/>	MN/ $\overline{\text{MX}}$
AD7	<input type="checkbox"/>	9	32	<input type="checkbox"/>	$\overline{\text{RD}}$
AD6	<input type="checkbox"/>	10	31	<input type="checkbox"/>	$\overline{\text{RQ}}/\overline{\text{GT0}}$
AD5	<input type="checkbox"/>	11	30	<input type="checkbox"/>	$\overline{\text{RQ}}/\overline{\text{GT1}}$
AD4	<input type="checkbox"/>	12	29	<input type="checkbox"/>	$\overline{\text{LOCK}}$
AD3	<input type="checkbox"/>	13	28	<input type="checkbox"/>	$\overline{\text{S2}}$
AD2	<input type="checkbox"/>	14	27	<input type="checkbox"/>	$\overline{\text{S1}}$
AD1	<input type="checkbox"/>	15	26	<input type="checkbox"/>	$\overline{\text{S0}}$
AD0	<input type="checkbox"/>	16	25	<input type="checkbox"/>	QS0
NMI	<input type="checkbox"/>	17	24	<input type="checkbox"/>	QS1
INTR	<input type="checkbox"/>	18	23	<input type="checkbox"/>	$\overline{\text{TEST}}$
CLK	<input type="checkbox"/>	19	22	<input type="checkbox"/>	READY
GND	<input type="checkbox"/>	20	21	<input type="checkbox"/>	RESET

Hình 2.8: Các chân ra vật lí của 8088

- **Các chân ra của 8088:**

8088 có 40 chân ra:

1. 20 chân địa chỉ A0 - A19. Vì vậy 8088 có thể địa chỉ hoá được 1 MB bộ nhớ chính.
2. 8 chân dữ liệu D0 - D7 được ghép chung với các chân A0 - A7. Vì vậy, 8088 chỉ truyền đồng thời được 8 bit dữ liệu vào / ra.
3. Các chân ngắt NMI và INTR được dùng để ngắt CPU:

+ NMI (None-Maskable Interrupt): ngắt không che, được dùng để chỉ ra các lỗi kiểm tra chẩn lẻ của bộ nhớ hoặc một số vấn đề quan trọng khác không thể trì hoãn. Với loại ngắt này phần mềm không thể tạm thời vô hiệu hoá (không che được).

+ INTR (maskable INTeRrupt): ngắt có thể che, loại ngắt này có thể tạm thời bị che (vô hiệu hoá) bởi phần mềm cho tới khi được phần mềm cho phép ngắt trở lại. Thông thường, INTR được các thiết bị I/O sử dụng.

4. CLK = CLoCK, GND = GrouND, Vcc : nguồn điện +5v.
5. Các chân S3 - S6 và SQ0 - SQ1 (Status): chứa thông tin về trạng thái bên trong CPU, trong thực tế các chân này thường không được sử dụng.
6. MN / $\overline{\text{MX}}$ = MiNimum / MaXimum: chân này ở mức cao sẽ đưa CPU vào chế độ tối thiểu, ở mức thấp đưa CPU vào chế độ tối đa.
7. $\overline{\text{RD}}$ = ReaD: chân này tích cực ở mức thấp cho biết CPU đang đọc bộ nhớ, khi chân này ở mức cao tức là CPU đang ghi bộ nhớ.
8. RQ / GTx = ReQuest / GranT: các chân này dùng cho trọng tài bus. Ví dụ, 8087 có thể sử dụng 2 chân này để yêu cầu 8088 thả nổi bus, nếu được phép nó trở thành chủ bus truy xuất bộ nhớ.
9. LOCK: chân này được sử dụng để báo cho các thiết bị khác không được sử dụng bus.
10. $\overline{\text{TEST}}$: chân này cho phép 8088 kiểm tra trạng thái của 8087. Ví dụ, khi gặp chỉ thị dấu chấm động, 8088 sẽ khởi động 8087, sau đó làm việc khác. Khi cần kết quả, 8088 kiểm tra 8087 xem đã thực hiện chưa, nếu chưa thì phải chờ.
11. READY: khi CPU đọc bộ nhớ, nếu bộ nhớ đưa dữ liệu lên bus dữ liệu đúng theo yêu cầu thì đặt tín hiệu này ở mức tích cực để báo cho CPU biết.
12. RESET: chân này được sử dụng để thiết lập lại trạng thái cho CPU. Ví dụ, khi người sử dụng máy tính nhấn phím reset.
13. Các chân $\overline{\text{S0}}$ - $\overline{\text{S2}}$ (Status):

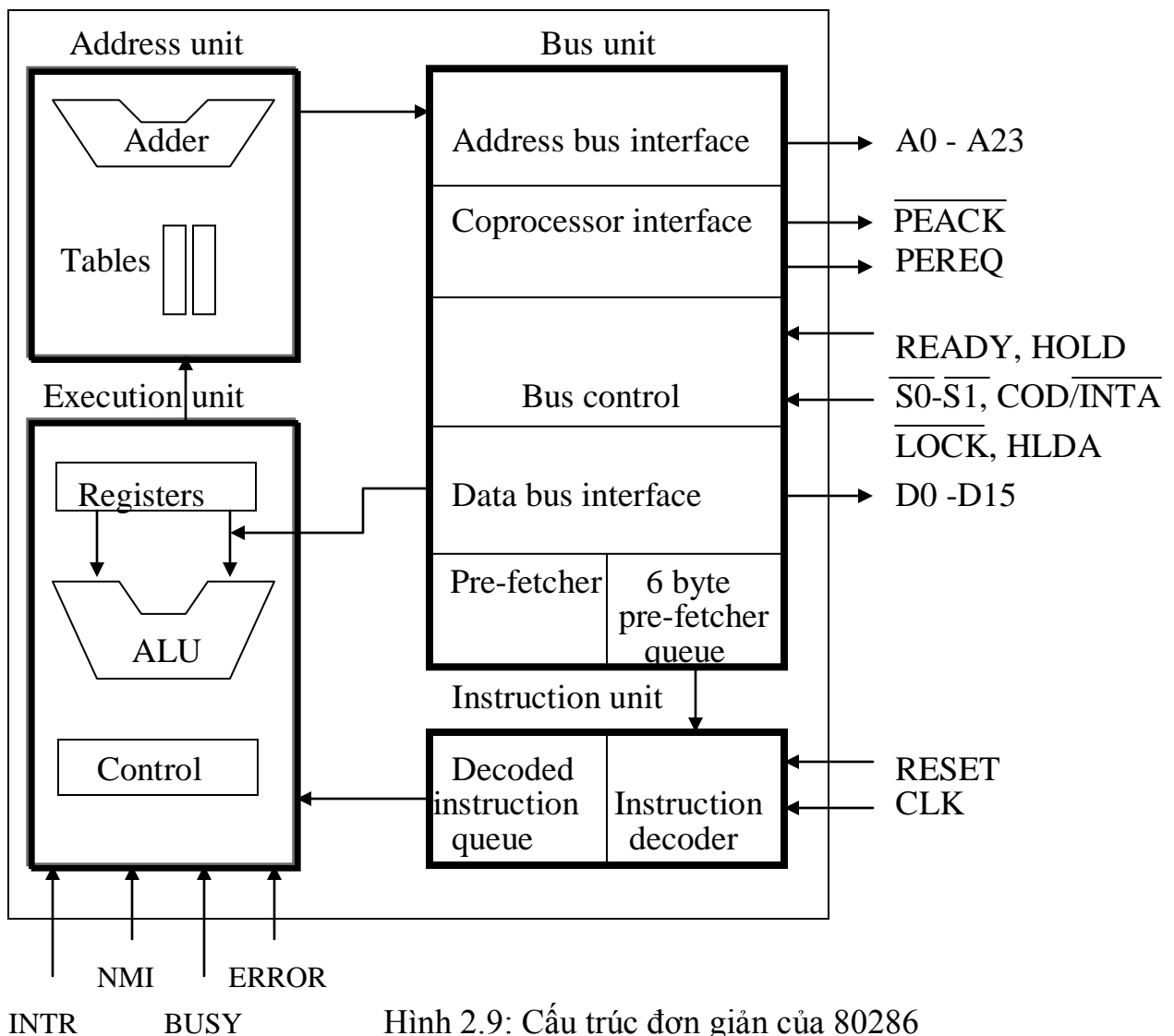
$\overline{\text{S2}}$	$\overline{\text{S1}}$	$\overline{\text{S0}}$	Loại chu kỳ bus
0	0	0	Chấp nhận ngắt
0	0	1	Đọc cổng vào ra
0	1	0	Ghi cổng vào ra
0	1	1	Dừng
1	0	0	Truy xuất mã(tìm nạp chỉ thị)
1	0	1	Đọc bộ nhớ
1	1	0	Ghi bộ nhớ
1	1	1	Thả nổi bus

2.2. 80286 của Intel

Bộ xử lý 286 được Intel giới thiệu năm 1981, nó lắp trong các máy PC AT (Personal Computer Advanced Technology) của IBM và các máy tương thích IBM. Các máy dựa trên bộ xử lý 286 có tốc độ nhanh hơn nhiều các máy thế hệ 1 vì nó có thể thực hiện một lệnh chỉ trong 4,5 chu kỳ và có thể truyền đồng thời 16 bit dữ liệu. 80286 có thể hoạt động được ở cả chế độ thực và chế độ bảo vệ.

- **80286 có các đặc trưng kỹ thuật chủ yếu sau:**

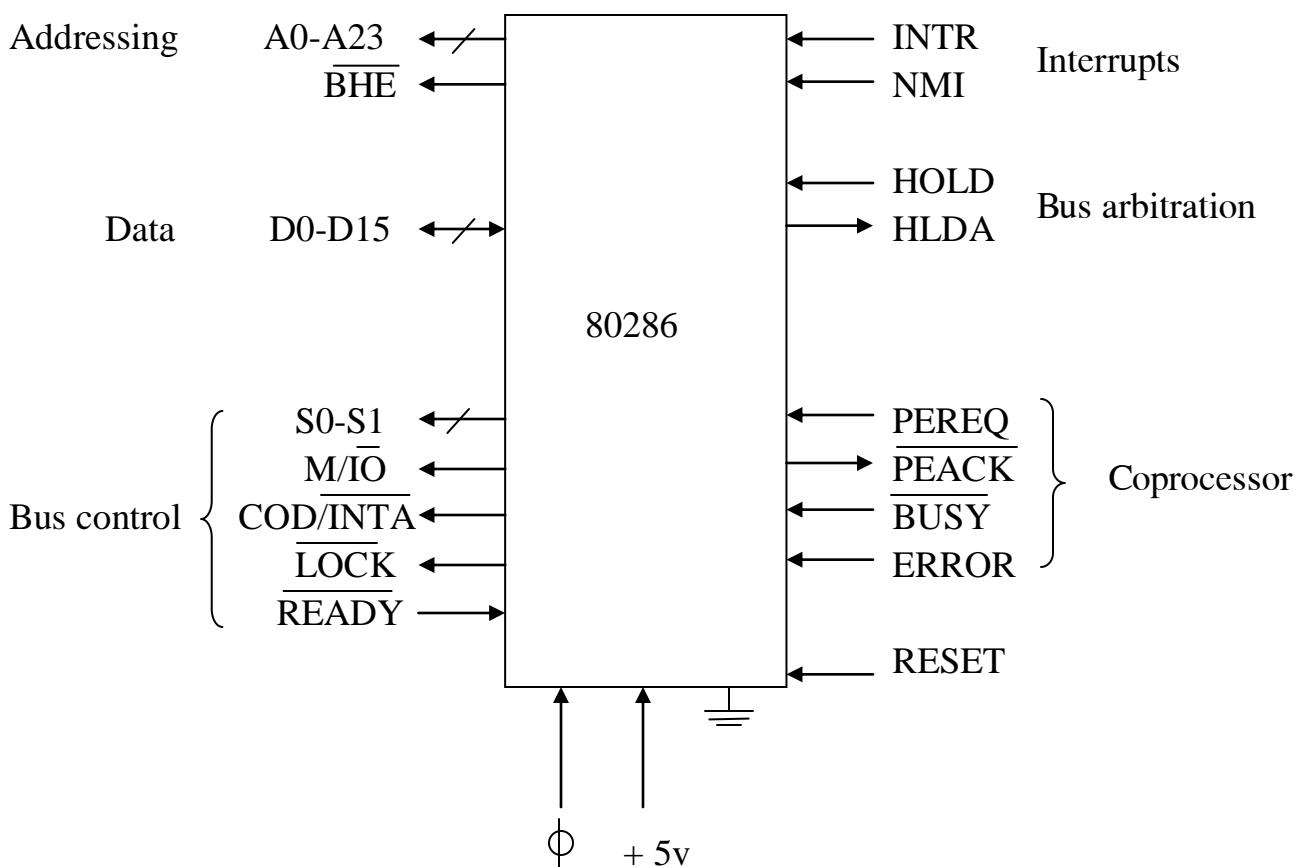
1. Độ rộng thanh ghi: 16 bit.
2. Độ rộng bus dữ liệu: 16 bit.
3. Độ rộng bus địa chỉ: 24 bit (không gian địa chỉ nhớ 16 MB).
4. Bộ đồng xử lý 80287 là một chip riêng biệt.



Hình 2.9: Cấu trúc đơn giản của 80286

- **80286 có 4 đơn vị chức năng độc lập:**

1. Đơn vị bus: thực hiện tất cả các thao tác về bus cho CPU, tìm nạp và lưu giữ các chỉ thị và dữ liệu khi cần.
2. Đơn vị chỉ thị: lấy các byte đã được tìm nạp bởi đơn vị bus, giải mã chúng thành các chỉ thị cho việc thực thi tuần tự. Đơn vị chỉ thị có thể lưu giữ 3 chỉ thị đã giải mã cùng một lúc. Vì vậy, rất hiếm khi CPU phải chờ đợi khi lấy chỉ thị kế tiếp.
3. Đơn vị thi hành: thực thi các chỉ thị đã giải mã từ đơn vị chỉ thị đưa tới. Một số chỉ thị có chứa địa chỉ bộ nhớ, các địa chỉ này được đưa tới đơn vị địa chỉ để được xử lý tiếp.
4. Đơn vị địa chỉ: thực hiện tất cả các công việc tính toán địa chỉ và quản lý bộ nhớ ảo (bộ nhớ cho phép cho phép chương trình sử dụng dung lượng bộ nhớ lớn hơn dung lượng RAM của máy tính. Bộ nhớ ảo được lấy từ đĩa cứng) .



Hình 2.10: Các chân ra logic của 80286

- **Các chân ra của 80286:**

80286 được đặt trong một vỏ hình vuông, có 68 chân được bố trí đều trên 4 cạnh. 80286 không phân biệt chế độ tối thiểu / tối đa và chỉ có một chế độ chân ra. Hình 6.9 là sơ đồ các chân ra logic của 80286:

1. 24 chân địa chỉ A0 - A23.
2. 16 chân dữ liệu D0 - D15.
3. $\overline{\text{BHE}}$ = Byte hight Enable: vì 80286 có thể đọc / ghi từ 2 byte như là một đơn vị. Tuy nhiên các chỉ thị của CPU còn có thể đọc và ghi 1 byte. Vấn đề đọc 1 byte không khó, vì CPU đọc một từ rồi sau đó lấy ra byte cần thiết. Vấn đề ghi sẽ khó khăn hơn, vì nếu chỉ cần ghi 1 byte, mà ta ghi cả từ thì nửa từ còn lại trong bộ nhớ đáng lẽ phải được giữ nguyên lại bị ghi đè. BHE không tích cực sẽ không cho phép byte cao được truyền đi.
4. Các chân điều khiển bus: xác định loại chu kỳ bus như: đọc / ghi bộ nhớ, đọc / ghi thiết bị vào ra, hay một chu kỳ bus nào khác. $\overline{\text{S0}} - \overline{\text{S1}}$ (Status), $\text{M} / \overline{\text{IO}}$ = Memory / IO, COD / INTA = CODE / INTerrupt Acknowledge).
5. Các chân $\overline{\text{LOCK}}$, $\overline{\text{READY}}$, INTR và NMI có chức năng giống như trong 8088.
6. HOLD , HLDA = HoLD Acknowledge: khi một thiết bị muốn làm chủ bus, tín hiệu HOLD được đặt ở mức tích cực. Khi thấy tín hiệu này, nếu chấp nhận, 80286 sẽ thả nổi bus và đặt tín hiệu HLDA ở mức tích cực. Thiết bị này nhận được tín hiệu HLDA sẽ chiếm giữ bus. Sau khi thực hiện xong công việc, HOLD đảo thành không tích cực.
7. + PEREQ = Processor Extension REQuest: tín hiệu yêu cầu bộ đồng xử lý mở rộng, thông qua tín hiệu này, co-CPU có thể yêu cầu CPU tìm nạp nội dung của một địa chỉ trong bộ nhớ ảo. Khi dữ liệu đã xuất hiện trên bus, đặt tín hiệu $\overline{\text{PEACK}}$ = Processor Extension ACKnowledge thành mức tích cực cho phép co-CPU lấy dữ liệu.
+ $\overline{\text{BUSY}}$ và $\overline{\text{ERROR}}$ được sử dụng để co-processor thông báo trạng thái khi chưa hoàn thành công việc. $\overline{\text{BUSY}}$ được dùng cho các báo hiệu bình thường, $\overline{\text{ERROR}}$ được dùng để ngắt CPU khi có lỗi nào đó xảy ra như tràn số chẳng hạn.
8. RESET có chức năng giống 8088.

2.3. 80386 của Intel

Bộ xử lý 386 là bộ xử lý 32 bit được giới thiệu 1985 và được đưa vào trong các máy tính cuối 1986. Bộ xử lý thế hệ ba có một số đặc trưng kỹ thuật chung sau đây:

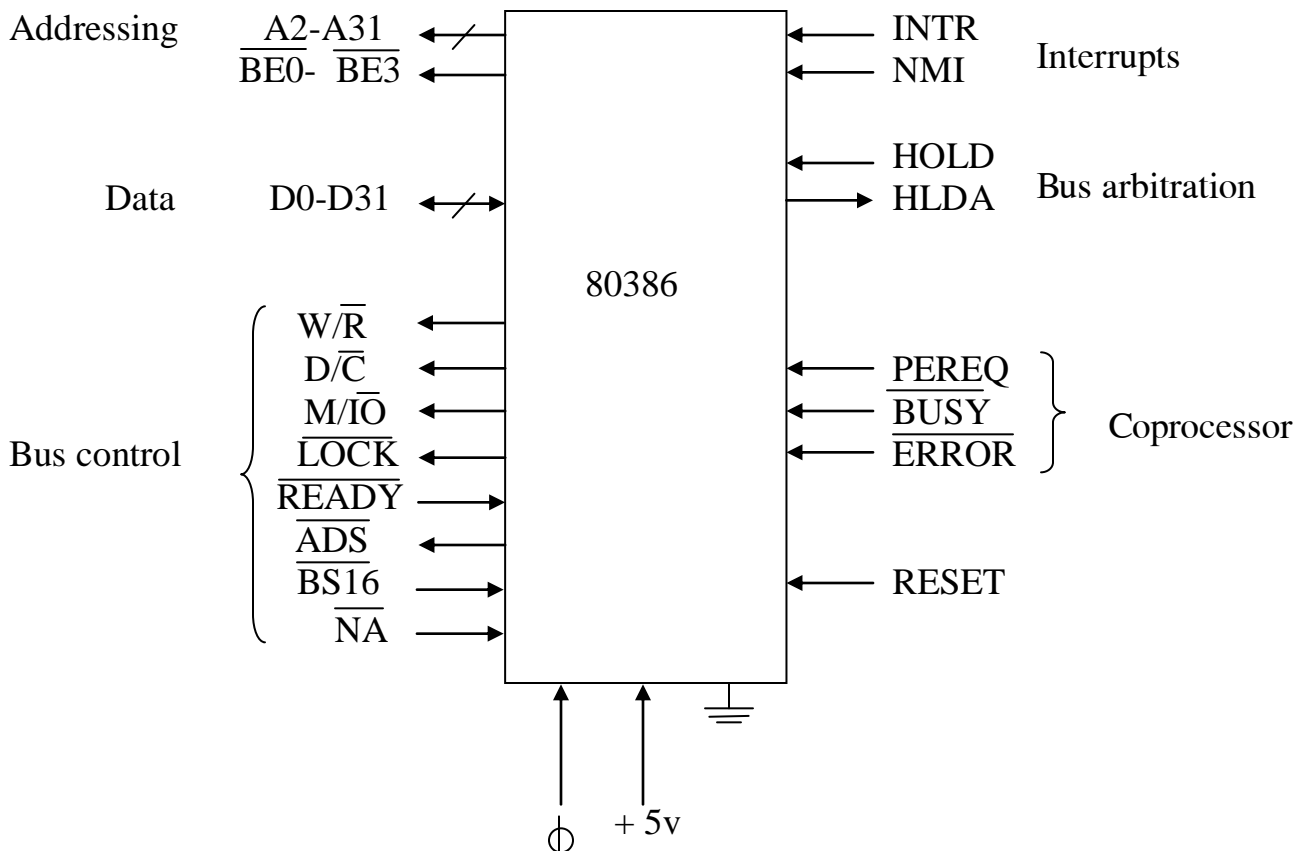
1. Bộ đồng xử lý 80387 là một chip riêng biệt, là chip đồng xử lý toán học có hiệu năng cao được thiết kế đặc biệt để làm việc với CPU 386. Có 2 loại bộ đồng xử lý đó là: 80387 DX làm việc với 386 DX, và 80387 SX làm việc với 386 SX và 386 SL.

2. Bộ xử lý 386 hỗ trợ cả hai chế độ làm việc: chế độ thực và chế độ bảo vệ, và nó có thể chuyển từ chế độ này sang chế độ kia mà không cần khởi động lại máy. Ngoài ra 386 còn hỗ trợ chế độ thực ảo cho phép chạy chế độ thực trong môi trường chế độ bảo vệ.

386 của Intel có nhiều phiên bản, phiên bản mạnh nhất có cấu hình sau:

1. Độ rộng thanh ghi: 32 bit.
2. Độ rộng bus dữ liệu: 32 bit.
3. Độ rộng bus địa chỉ: 32 bit (không gian địa chỉ nhớ 4 GB).

80386 có 8 đơn vị chức năng độc lập vì đơn vị địa chỉ và đơn vị thi hành lại được chia làm nhiều đơn vị nhỏ. Vì vậy, hiệu năng của 386 mạnh hơn 286 nhiều.



Hình 2.11: Các chân ra logic của 80386

- **Các chân ra của 80386:**

1. Ba mươi chân địa chỉ A2 - A31: vì 386 có thể làm việc với các từ 32 bit, do vậy CPU có thể tìm nạp các địa chỉ 0, 4, 8,... Tức là 2 bit cuối của các địa chỉ luôn là 00. Cho nên không cần A0, A1. Tuy nhiên, khi truy cập các địa từ nhớ 8 bit và 16 bit trong bộ nhớ phải dùng đến 4 tín hiệu BE0 - BE3, để chỉ ra các byte nào trong 4 byte của từ.

2. Ba hai chân dữ liệu D0 - D31.
3. Các chân điều khiển bus:
 - + Các chân xác định loại chu kỳ bus như: $\overline{W/R}$ (Write/Read), $\overline{D/C}$ (Data/Code), M / \overline{IO} (Memory / IO), xác định các chu kỳ bus: đọc mã, đọc dữ liệu, ghi dữ liệu, đọc I/O, ghi I/O,...
 - + Các chân \overline{LOCK} , \overline{READY} có chức năng giống như trong 8088.
 - + \overline{ADS} (Address Status): chân này báo hiệu địa chỉ cần truy cập đã ở trên bus.
 - + $\overline{BS16}$ (Bus Size 16): tín hiệu này tích cực, 386 chỉ truyền đồng thời 16 bit.
 - + \overline{NA} (Next Address): tín hiệu này tích cực, bộ nhớ báo cho 386 biết địa chỉ kế tiếp đã sẵn sàng cho chu kỳ bus tiếp theo mặc dù chu kỳ hiện tại chưa kết thúc. Đây chính là giải pháp đường ống nhằm tăng tốc hệ thống.
4. Các chân còn lại có chức năng giống như chip 286.

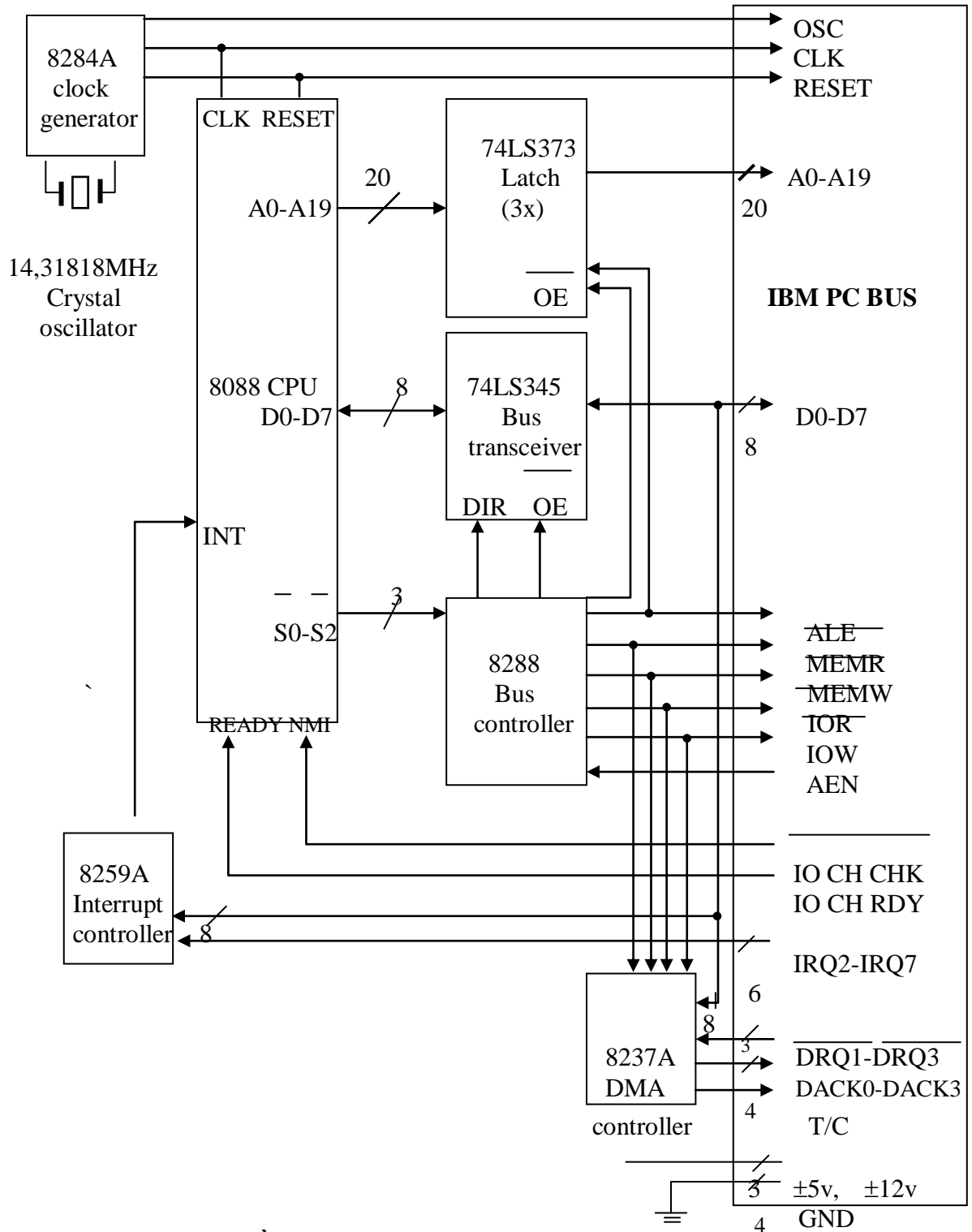
3. Thí dụ về BUS

3.1. IBM PC BUS

Signal	Lines	In	Out	Description
OSC	1		×	70 ns Clock signal (14,31818 MHz)
CLK	1		×	210 ns Clock signal (4,17 MHz)
RESET	1		×	Used to reset the CPU and I/O devices
A ₀ -A ₁₉	20		×	20 Address lines
D ₀ -D ₇	8		×	8 Data lines
ALE	1		×	Address latch enable (cho phép chốt địa chỉ)
\overline{MEMR}	1		×	Memory read
\overline{MEMW}	1		×	Memory write
\overline{IOR}	1		×	I/O read
\overline{IOW}	1		×	I/O write
AEN	1	×		Address Enable (CPU float the bus)
$\overline{IO\ CH\ CHK}$	1	×		I/O channel check
IO CH RDY	1	×		I/O channel ready
IRQ ₂ -IRQ ₇	6	×		Interrupt request lines
DRQ ₁ -DRQ ₃	3	×		DMA request lines
DACK ₀ -DACK ₃	4		×	DMA ACKnowledge lines
T/C	1		×	Terminal/Count (indicates DMA completed)
Pwoer	5		×	±5v, ±12v
GND	3			Ground
Reserved	1			Card select on XT

Hình 2.12: Các đường tín hiệu trong IBM PC bus

Cột nhãn In, chỉ các tín hiệu đi từ bus vào motherboard. Cột nhãn Out, chỉ các tín hiệu đi từ motherboard ra bus.



Hình 2.13: Sơ đồ máy PC dựa trên 8088, các chip hỗ trợ và bus hệ thống của nó

IBM PC bus được sử dụng trong các hệ thống giá thành thấp. Nó được sử dụng rộng rãi trong các hệ thống dựa trên chip 8088, hầu hết họ IBM PC và các máy tương thích sử dụng bus này để cho các bảng mạch I/O kết nối vào hệ thống. Chính IBM PC bus là cơ sở cho IBM PC/AT bus và nhiều bus khác. IBM PC bus có 62 đường dây, trong đó có 20 đường địa chỉ, 8 đường dữ liệu và các đường tín hiệu khác được liệt kê trong hình 2.12.

Hình 2.13 là sơ đồ đơn giản một PC dựa trên chip 8088, các chip hỗ trợ và bus hệ thống của nó.

Về mặt vật lý, IBM PC bus được khắc trên bảng mạch chính, thường có khoảng 6 đầu nối (slot) để cắm các bảng mạch mở rộng. Ngoài 20 dây địa chỉ, 8 dây số liệu, IBM PC bus còn nhiều đường dây khác, chúng có liên quan tới các chip hỗ trợ cho chip 8088. Trên hình vẽ trình bày 7 chip chính (trái tim của máy PC dựa trên 8088):

1. 8284A: bộ tạo xung đồng hồ.
2. 8259A: bộ điều khiển ngắt.
3. 74LS373 (3x): bộ chốt (3 bộ).
4. 74LS245: bộ thu phát bus.
5. 8088: chip vi xử lý.
6. 8288: bộ điều khiển bus.
7. 8237A DMA: bộ điều khiển truy cập bộ nhớ trực tiếp.

- **Các đường tín hiệu OSC, CLK, RESET:**

+ Các máy IBM PC đầu tiên sử dụng bộ tạo dao động thạch anh (Crystal oscillator) tạo ra xung nhịp đồng hồ tần số 14,31818 MHz. Tần số này nhằm tạo ra tín hiệu đồng bộ với tần số sử dụng trong các hệ thống truyền hình màu NTSC được sử dụng ở Bắc Mỹ và Nhật Bản thời bấy giờ. Vì IBM cho rằng khách hàng sẽ dùng máy thu hình làm thiết bị hiển thị để tiết kiệm tiền. Tuy nhiên không phải là như vậy, nhưng tần số này đã được chọn và IBM không thay đổi nữa. Đường tín hiệu OSC có tần số này.

+ Tần số OSC (OSCillator) là cao đối với 8088 (tần số hoạt động tối đa của 8088 là 5 MHz), do đó OSC được chia 3 để có tín hiệu với tần số 4,77 MHz (việc chia 3 nhờ chip 8284A). Tần số 4,77 MHz được sử dụng như một xung đồng hồ chủ để xác định chu kỳ bus. Tín hiệu tần số 4,77 MHz cũng có trên IBM PC bus và ký hiệu là CLK (CLOCK). Tín hiệu này không cân xứng như tín hiệu đồng hồ thạch anh

(14,31818 MHz), trong một chu kỳ 2/3 có mức thấp, 1/3 có mức cao. Sau này, có một số trong họ PC sử dụng chip có tốc độ cao hơn 8 MHz, trong các hệ này tín hiệu CLK cũng tăng lên một cách tỷ lệ.

+ Chip 8284A cũng sinh ra tín hiệu RESET trên BUS. Để thiết lập lại trạng thái ban đầu cho CPU, các mạch điện bên ngoài gửi tín hiệu tới 8284A, chip này đặt tín hiệu RESET lên mức tích cực, buộc CPU và các thiết bị I/O khởi tạo lại chính chúng.

- **Các đường địa chỉ và dữ liệu - Address, Data:**

+ 8088 không nối trực tiếp với các đường địa chỉ và số liệu của bus mà thông qua các chip khác. Các đường địa chỉ được chốt bằng cách dùng chip 3 bộ chốt 74LS373 - mỗi bộ 8 bit, tuy vậy chỉ sử dụng 20/24 đường của chúng. Sở dĩ cần phải chốt tín hiệu địa chỉ vì một số tín hiệu địa chỉ và dữ liệu được truyền thông qua một số chân dùng chung của 8088 ($AD_0 - AD_7$). Đầu mỗi chu kỳ bus của CPU, 8088 đưa ra các tín hiệu địa chỉ, các thanh ghi chốt địa chỉ sẽ chốt các giá trị địa chỉ lại (giữ cho nó không bị mất) trong khoảng thời gian còn lại của chu kỳ bus, khi mà các tín hiệu địa chỉ đó trên các chân của 8088 đã mất đi.

+ Các đường dữ liệu đưa dữ liệu ra hoặc nhận dữ liệu vào CPU trong những khoảng thời gian xác định vì vậy không cần chốt. Các đường dữ liệu của bus được điều khiển bởi chip 74LS245. Chân DIR (DIRection) xác định hướng của tín hiệu đi vào hay đi ra khỏi CPU.

Lý do chính của việc nối các chân của 8088 với bên ngoài thông qua các bộ đệm là vì nó được chế tạo theo công nghệ MOS (Metal Oxide Semiconductor). Nó không có khả năng cung cấp đủ dòng điện để điều khiển một bus có nhiều bảng mạch nối vào. các chip đệm thuộc loại TTL (?) có khả năng cung cấp đủ dòng điện cho bus .

Một lý do khác là: khi có một thiết bị nào đó không phải là CPU muốn trở thành bus master thì nó phải yêu cầu CPU thả nổi các bus. Phương pháp đơn giản nhất được áp dụng là thiết bị đó phát tín hiệu AEN (Address ENable) để đảo tín hiệu cho phép xuất " \overline{OE} - Output Enable " trên các bộ chốt và bộ thu phát, làm cho bus được thả nổi.

- **Tín hiệu ALE (Address Latch Enable):**

Tín hiệu ALE được đặt mức tích cực khi CPU đang điều khiển các tín hiệu địa chỉ, tín hiệu này cho phép các chip 74LS373 biết khi nào cần chốt địa chỉ lại, và để bộ nhớ và các chip I/O biết khi nào các tín hiệu địa chỉ trên bus là hợp lệ. Trước khi ALE ở mức tích cực, các đường địa chỉ sẽ không có giá trị sử dụng.

- **Các đường tín hiệu \overline{MEMR} , \overline{MEMW} , \overline{IOR} , \overline{IOW} :**

+ Các tín hiệu này được sử dụng để đọc / ghi bộ nhớ, đọc / ghi các thiết bị vào / ra. Trong thực tế, bus cung cấp 2 không gian địa chỉ riêng biệt, một cho bộ nhớ (địa chỉ RAM) và một cho vào / ra (địa chỉ I/O). Bộ nhớ sẽ không phản ứng khi các tín hiệu $\overline{\text{IOR}}$ và $\overline{\text{IOW}}$ ở mức tích cực và ngược lại.

+ Như ở phần trên chúng ta đã biết. CPU sử dụng các tín hiệu S_0 - S_2 (Status) đưa vào chip điều khiển bus 8288 để tạo ra các tín hiệu này cùng với tín hiệu ALE. Chip 8288 cũng nhận tín hiệu AEN từ bus, tín hiệu này do một thiết bị muốn trở thành chủ bus phát ra. Khi nhận được tín hiệu AEN, 8288 sẽ phát tín hiệu điều khiển các chip 74LS373 và 74LS245 thả nổi bus.

- **Tín hiệu $\overline{\text{IOCHCHK}}$ (I/O CHannel Check):**

Tín hiệu này sẽ tích cực khi có lỗi parity bị phát hiện trên bus. Tín hiệu này sẽ kích một ngắt không thể che (NMI).

- **Tín hiệu IOCHRDY (I/O CHannel Ready):**

Tín hiệu này do bộ nhớ phát ra khi tốc độ hoạt động của nó chậm, tín hiệu này kích hoạt chân READY yêu cầu CPU chèn wait states vào các chu kỳ đọc/ghi bộ nhớ.

- **Các tín hiệu IRQ2 - IRQ7 (Interrupt ReQuest):**

Là các tín hiệu do các thiết bị ngoại vi sinh ra, đưa vào chip điều khiển ngắt 8259A. Khi có tín hiệu gửi đến, 8259A sẽ phát ra một tín hiệu yêu cầu ngắt tới CPU và đặt số hiệu vector ngắt lên đường dữ liệu khi CPU yêu cầu. IRQ_0 thường được dùng bởi chip clock, IRQ_1 thường được sử dụng bởi keyboard.

- **Các tín hiệu liên quan đến DMA:**

Các tín hiệu còn lại nói chung liên quan tới hoạt động của DMA (Direct Memory Access), chip 8237A được Intel thiết kế nhằm quản lý các nghi thức bus và thực hiện DMA trong đó có việc tăng địa chỉ ô nhớ và giảm con đếm sau khi truyền mỗi byte. Việc này nó làm thay cho các thiết bị I/O và giúp giảm giá thành của chúng.

Về cơ bản chip 8237A là một CPU nhỏ có các vi chương trình được ghi sẵn bên trong. Khi 8088 muốn bắt đầu hoạt động DMA đối với một thiết bị ngoại vi nào đó, nó nạp địa chỉ ô nhớ, số byte cần truyền, số hiệu thiết bị I/O, hướng truyền và các thông tin khác vào các thanh ghi bên trong chip 8237A. Khi bộ điều khiển đĩa cứng đã sẵn sàng đọc hoặc ghi byte đầu tiên, nó đặt mức tích cực lên một trong các đường DRQ_x (DMA request) của bus để đưa vào chip 8237A. Khi nhận được tín hiệu này, 8237A đòi chiếm dụng bus để sẵn sàng truyền 1 byte. Sau đó 8237A phát ra tín hiệu

DACKx (DMA ACKnowledge) tới bộ điều khiển đĩa báo cho nó biết hãy ghi byte dữ liệu (hoặc đọc). Trong khoảng thời gian của chu kỳ này, bộ điều khiển đĩa là bus master còn bộ nhớ là bus slave. Chip 8237A có 4 kênh độc lập và có thể quản lý đồng thời 4 đường truyền.

- **Tín hiệu T/C:**

Đường T/C (Terminal/Count) được chip 8237A đặt mức tích cực khi byte count = 0, báo cho bộ điều khiển I/O biết rằng công việc yêu cầu đã hoàn thành, đã đến lúc báo hiệu cho 8259A gọi ngắt.

- **Các đường còn lại:**

8 đường dây còn lại của bus là các đường nguồn nuôi (2 đường +5v, 1 đường -5v, 1 đường +12v, một đường -12v, 3 đường nối mát).

Các máy 286, 386 cũng dùng các chip tương tự và cũng sắp xếp theo sơ đồ hình 2.13.

3.2. IBM PC/AT bus

Khi hãng IBM đưa ra loại máy IBM PC/AT, họ phải giải quyết một số vấn đề quan trọng, là làm sao vừa kế thừa được IBM PC vừa phát huy được những khả năng hơn hẳn của bộ xử lý 80286: có độ rộng bus địa chỉ 24 bit, do đó đánh địa chỉ được 16 MB bộ nhớ và có độ rộng bus dữ liệu 16 bit do đó có thể đọc/ghi đồng thời 16 bit.

IBM chọn giải pháp mở rộng PC bus, bổ sung thêm các khe cắm mở rộng cũ một đoạn slot, đoạn bổ sung có 36 đường tín hiệu: 4 đường địa chỉ, 8 đường dữ liệu, các đường gọi ngắt, kênh DMA, dây nguồn nuôi và dây mát. Nhờ cải tiến này các bảng mạch mở rộng trước đây dùng cho IBM PC vẫn có thể cắm vào và hoạt động trong máy IBM PC/AT.

Tần số tín hiệu đồng hồ IBM PC/AT bus cũng tăng lên 8 MHz, nhờ đó tốc độ truyền thông trên bus cũng tăng lên nhiều.

Năm 1991, IEEE (Institute of Electrical and Electronic Engineers - viện nghiên cứu về điện và các công nghệ điện tử) đã đưa ra tiêu chuẩn quốc tế cho bus của máy PC gọi là ISA (Industrial Standard Architecture- cấu trúc chuẩn công nghiệp). Loại 8 bit dành cho IBM PC bus, loại 16 bit dành cho IBM PC/AT bus.

3.3. Các bus 32 bit

Với sự ra đời của các chip CPU 32 bit dữ liệu, bus ISA không còn thích nghi nữa. Khi IBM thiết kế họ PS/2, đó là thế hệ sau của IBM PC và IBM PC/AT, họ đã

xây dựng một bus hoàn toàn mới là bus MCA (Micro Channel Architecture) và đăng ký bản quyền. MCA có kiến trúc khác hoàn toàn ISA và có khả năng vượt trội hơn về kỹ thuật. Tuy nhiên MCA không tương thích ngược với ISA, các card thiết kế cho ISA không cắm được trên máy bus MCA. Vì lí do này mà một loại bus 32 bit kế thừa và mở rộng bus ISA ra đời có tên là bus EISA (Extended Industrial Standard Architecture).

Về căn bản bus này là sự mở rộng PC/AT bus thành 32 bit, nó giữ nguyên tính tương thích với các máy tính và các card mở rộng đã có. Ngoài ra nó cung cấp các khe cắm mở rộng có 32 đường dữ liệu.