



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KIẾN TRÚC MÁY TÍNH

TS. Nguyễn Đức Minh
minhnd@ptit.edu.vn



Chương 5. Phối ghép và lập trình điều khiển thiết bị

5.1 Phối ghép CPU với bộ nhớ

5.2 Phối ghép CPU với thiết bị ngoại vi

5.3 Lập trình phối ghép bàn phím

5.4 Lập trình điều khiển hệ thống đèn LED

5.5 Lập trình điều khiển nhiệt độ

5.6 Lập trình điều khiển hệ thống đèn giao thông

5.7 Lập trình điều khiển rô bốt đơn giản và một số ví dụ

5.8 Các phương pháp vào/ra dữ liệu



5.1 Phối ghép CPU với bộ nhớ

5.1.1. Các tín hiệu của CPU

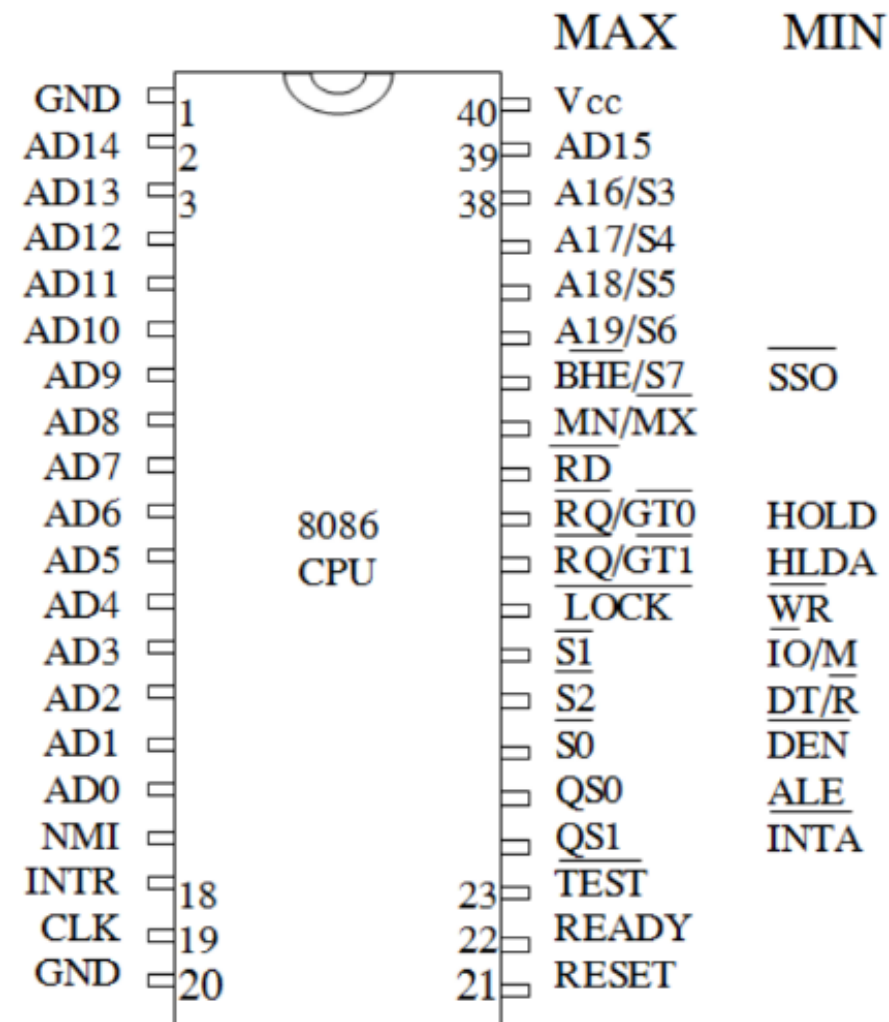
Vi xử lý Intel 8086/8088 có 40 chân tín hiệu, bao gồm

✓ Nhóm tín hiệu địa chỉ:

- **AD0-AD7**: 8 chân ghép kênh cho phần thấp bus A và bus D ;
- **A8-A15**: 8 chân tín hiệu phần cao bus A
- **A16/S3-A19/S6**: 4 chân ghép kênh cho phần cao bus A và bus C

✓ Nhóm tín hiệu dữ liệu

- AD0-AD7: 8 chân ghép kênh cho phần thấp bus A và bus D;
- Khi chân chốt ALE=0 → tín hiệu dữ liệu, ALE=1 → tín hiệu địa chỉ.





5.1 Phối ghép CPU với bộ nhớ

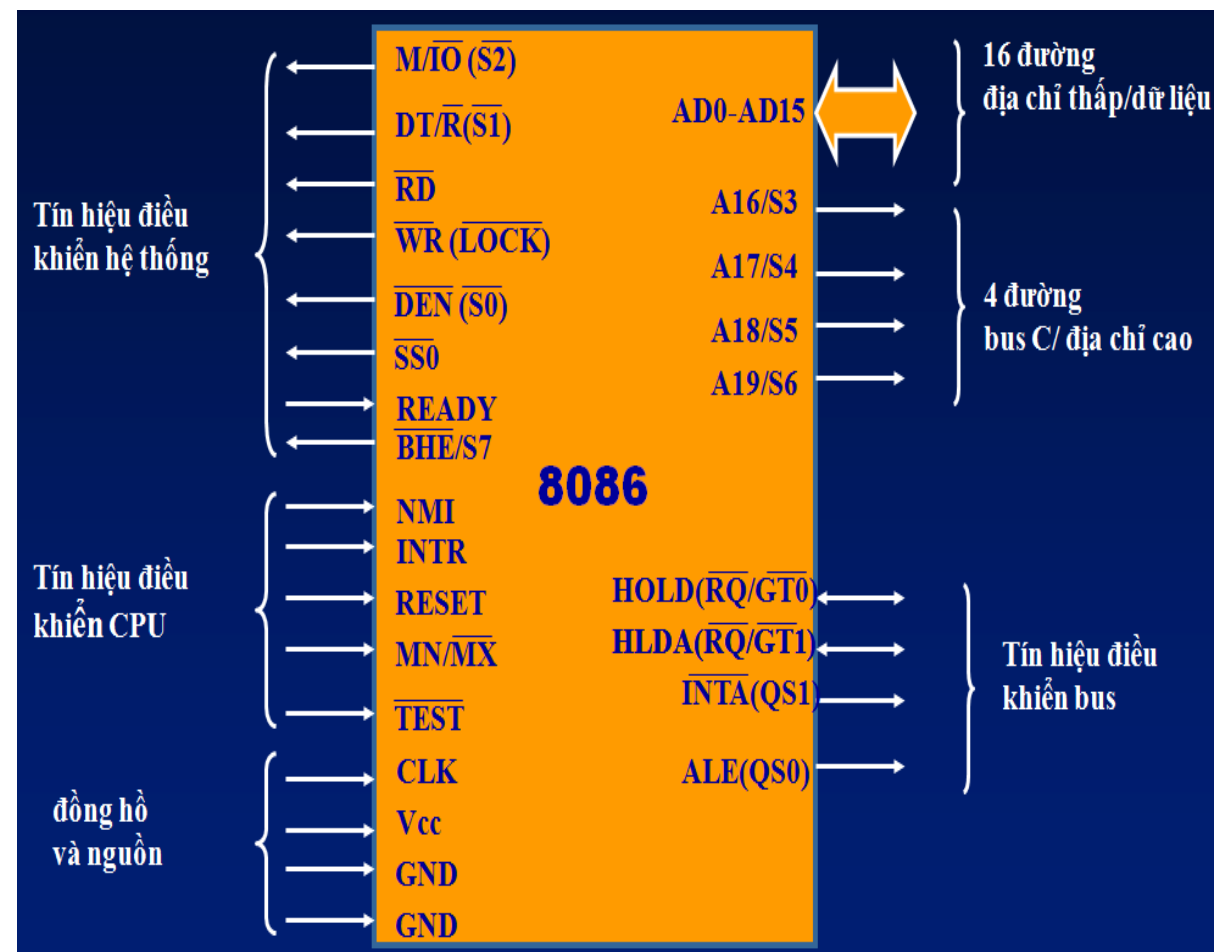
5.1.1. Các tín hiệu của CPU

✓ Nhóm tín hiệu điều khiển hệ thống:

- M/\overline{IO}
- DT/\overline{R}
- \overline{RD}
- \overline{WR}
- \overline{DEN}
- $\overline{SS0}$
- $READY$

✓ Nhóm tín hiệu điều khiển bus:

- $HOLD$
- $HLDA$
- \overline{INTA}
- ALE





5.1 Phối ghép CPU với bộ nhớ

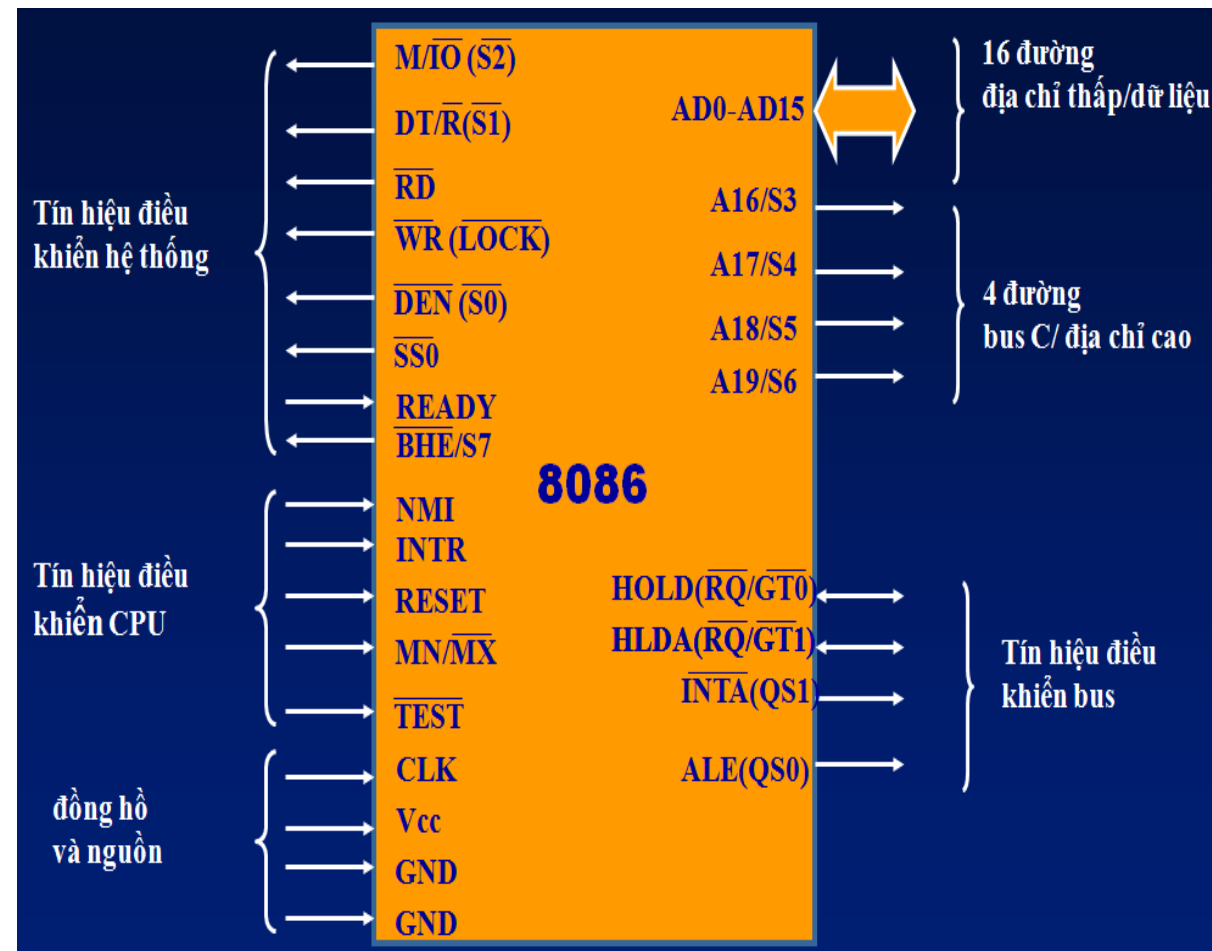
5.1.1. Các tín hiệu của CPU

✓ Nhóm tín hiệu điều khiển CPU:

- NMI
- INTR
- RESET
- MN/ $\overline{\text{MX}}$
- $\overline{\text{TEST}}$

✓ Nhóm tín hiệu đồng hồ và nguồn:

- CLK
- Vcc
- GND
- GND



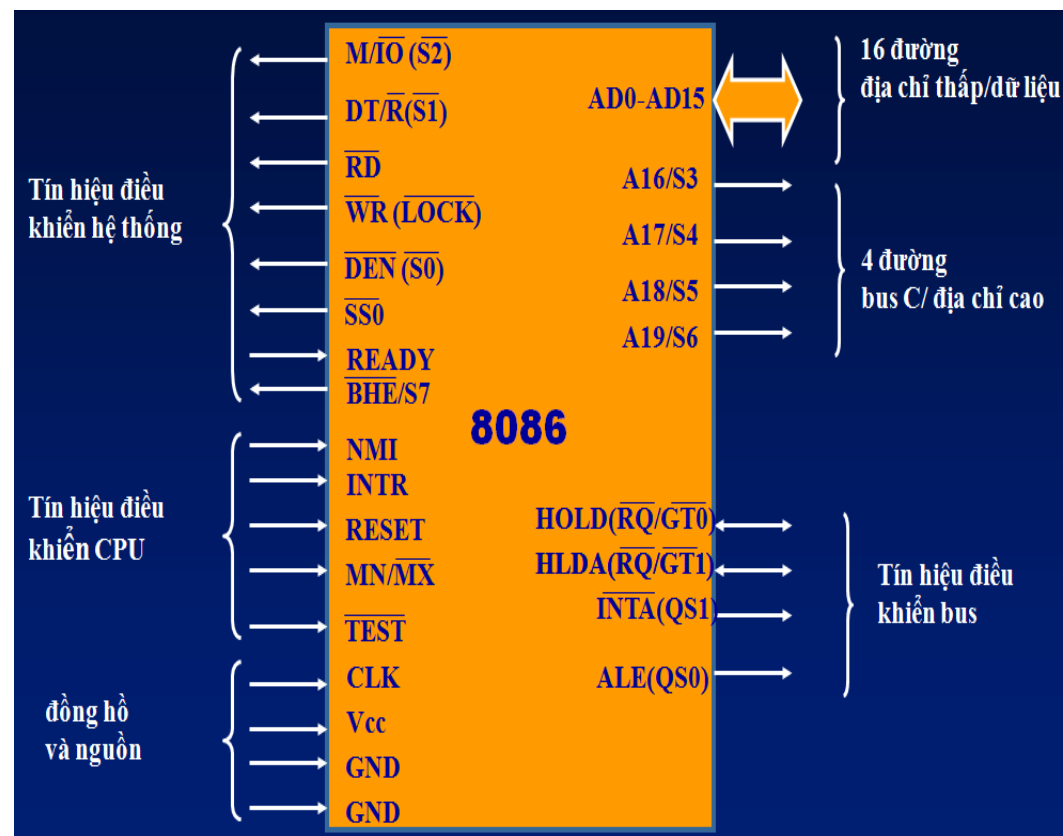


5.1 Phối ghép CPU với bộ nhớ

5.1.1. Các tín hiệu của CPU

✓ Nhóm các tín hiệu trạng thái:

- S3, S4: phối hợp cho biết trạng thái truy nhập các thanh ghi đoạn
 - 00: CPU truy nhập đoạn dữ liệu phụ ES
 - 01: CPU truy nhập đoạn ngăn xếp SS
 - 10: CPU truy nhập đoạn mã hoặc không đoạn nào
 - 11: CPU truy nhập đoạn dữ liệu
 - S5: S5 phản ánh giá trị cờ IF
 - S6: S6 luôn bằng 0





5.1 Phối ghép CPU với bộ nhớ

5.1.2. Chu kỳ bus

IO/M	DT/R	SS0	
0	0	0	Đọc mã lệnh
0	0	1	Đọc bộ nhớ
0	1	0	Ghi bộ nhớ
0	1	1	Buýt rồi
1	0	0	Chấp nhận yêu cầu ngắt
1	0	1	Đọc thiết bị ngoại vi
1	1	0	Ghi thiết bị ngoại vi
1	1	1	Dừng



CPU 8086/8088 có thể làm việc ở 2 chế độ: **Min** và **Max**

- Chân MN/MX nối nguồn 5v
- CPU tự sinh các tín hiệu điều khiển bộ nhớ và các thiết bị ngoại vi truyền thống
- Các tín hiệu: IO/M, WR, INTA, ALE, HOLD, HLDA, DT/R, DEN

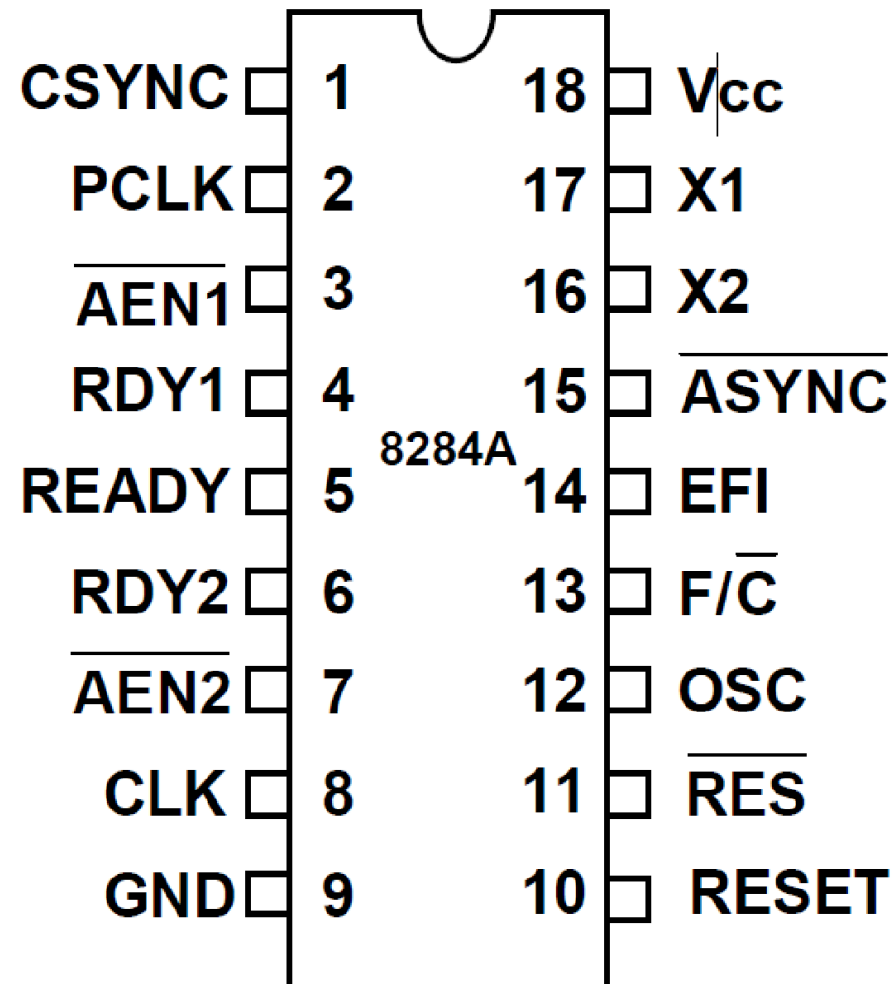
- Chân MN/MX nổi đất
- CPU gửi các tín hiệu trạng thái đến mạch phụ trợ và các mạch này sinh các tín hiệu điều khiển bộ nhớ và các thiết bị ngoại vi.
- Các tín hiệu: RQ/GT0, RQ/GT1, LOCK, S2, S1, S0, QS0, QS1



5.1 Phối ghép CPU với bộ nhớ

5.1.3. Mạch tạo xung nhịp 8284

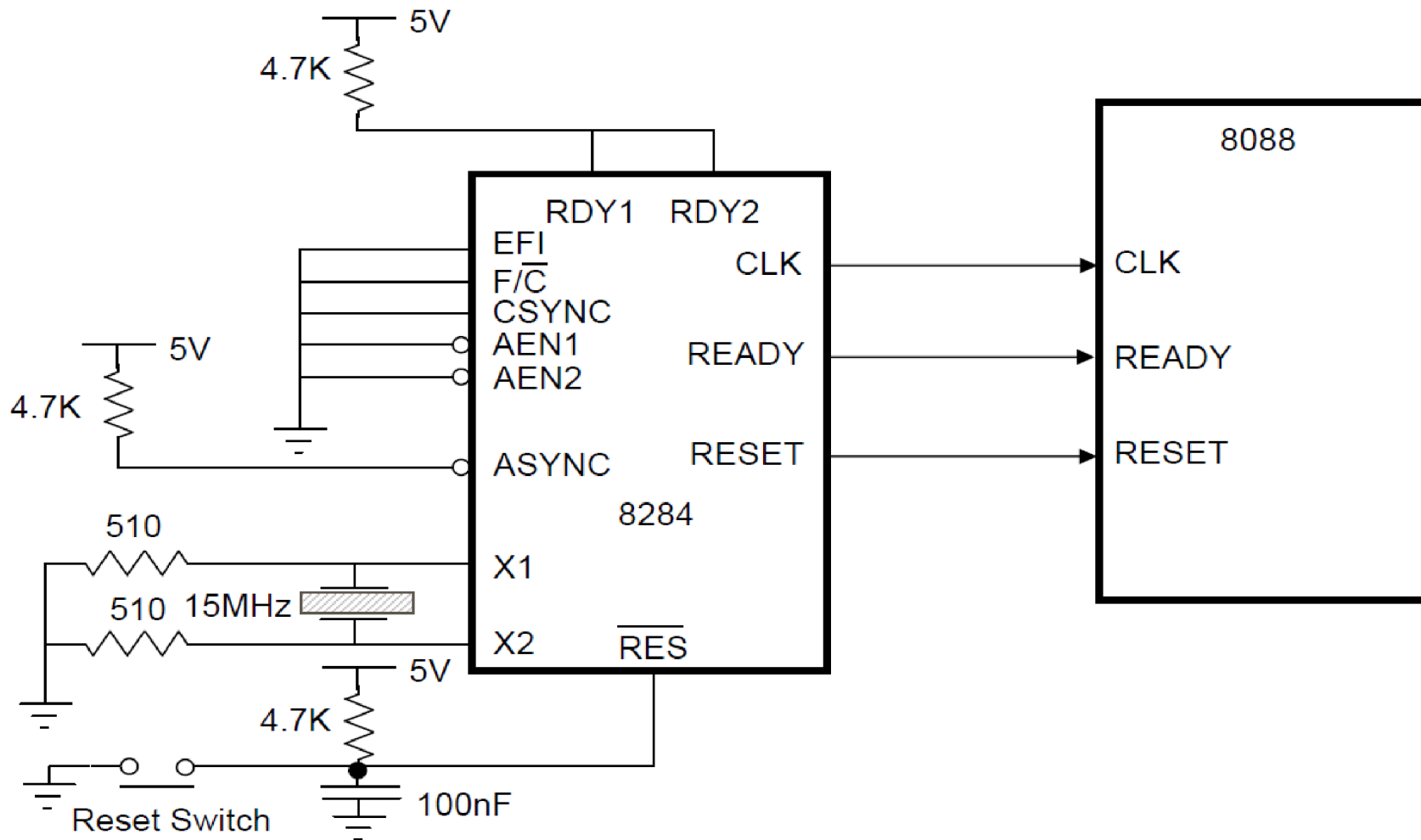
- ❖ Cung cấp tín hiệu CLOCK, READY và RESET
- ❖ OSC: Tín hiệu xung nhịp vào (thạch anh) f_x
- ❖ EFI: Xung nhịp ngoài
- ❖ CLK: Xung nhịp ($f_{CLK} = f_x/3$)
- ❖ PCLK: Xung nhịp ngoại vi ($f_{PCLK} = f_x/6$)





5.1 Phối ghép CPU với bộ nhớ

5.1.3. Mạch tạo xung nhịp 8284





5.1 Phối ghép CPU với bộ nhớ

5.1.4. Định thời và chu trình đọc ghi bus

- ❖ Truy nhập bộ nhớ, vào/ra tính theo chu trình bus. Chu trình bus tiêu biểu gồm 4 xung nhịp đồng hồ (T)
 - Sinh tín hiệu địa chỉ trên bus địa chỉ (T_1)
 - Sinh tín hiệu đọc/ghi trong xung (T_2 - T_3)
 - Đọc/Lưu dữ liệu trên bus dữ liệu (T_3)
- ❖ Để truyền dữ liệu không lỗi, các tín hiệu trên bus cần được tạo và duy trì trong chu trình bus
 - Biến dạng do trở kháng (tự cảm, điện dung)
 - Trễ tín hiệu khi lan truyền trên bus
 - Hình dạng xung (sườn lên, xuống, độ rộng)



5.1 Phối ghép CPU với bộ nhớ

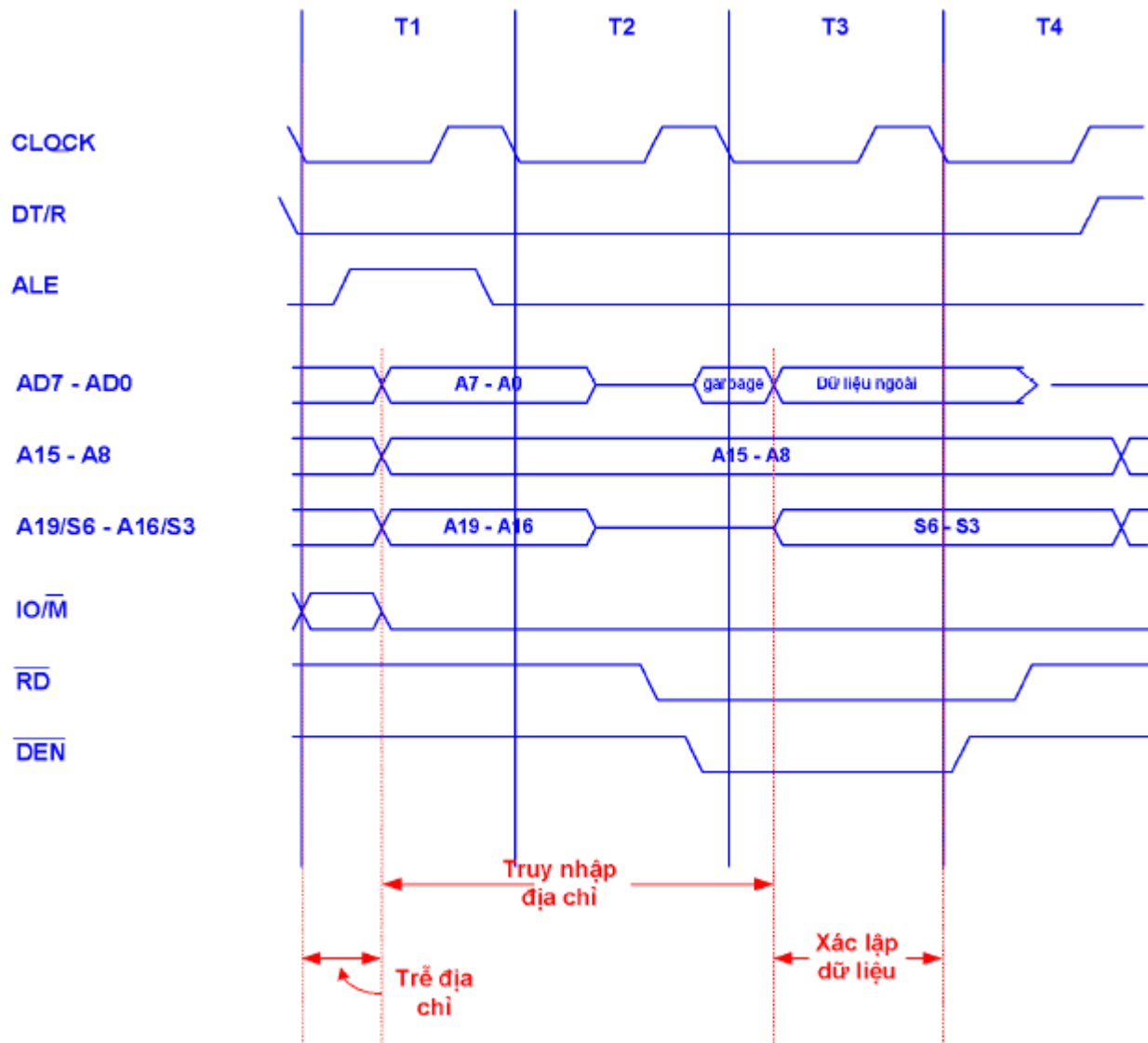
5.1.4. Định thời và chu trình đọc ghi bus

- ❖ T_1 : khởi đầu chu trình. Sinh các tín hiệu điều khiển chốt, kiểu thao tác, hướng dữ liệu và địa chỉ
- ❖ T_2 : sinh tín hiệu điều khiển đọc/ghi. Chân DEN báo dữ liệu ra sẵn sàng. Chân READY báo dữ liệu vào sẵn sàng.
- ❖ T_3 : Đọc/Ghi dữ liệu
- ❖ T_4 : Kết thúc các tín hiệu điều khiển



5.1 Phối ghép CPU với bộ nhớ

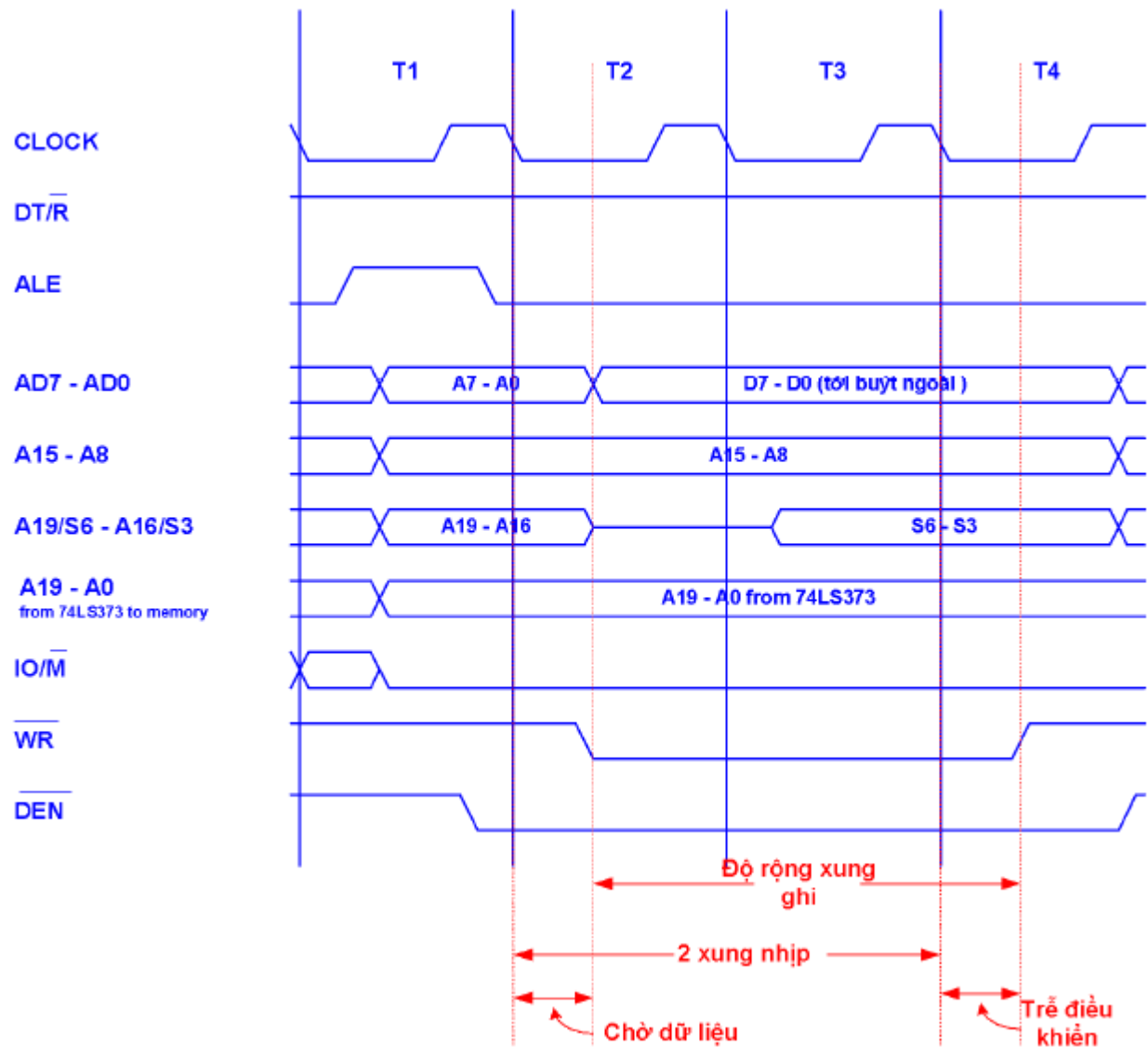
5.1.4. Chu trình đọc bus





5.1 Phối ghép CPU với bộ nhớ

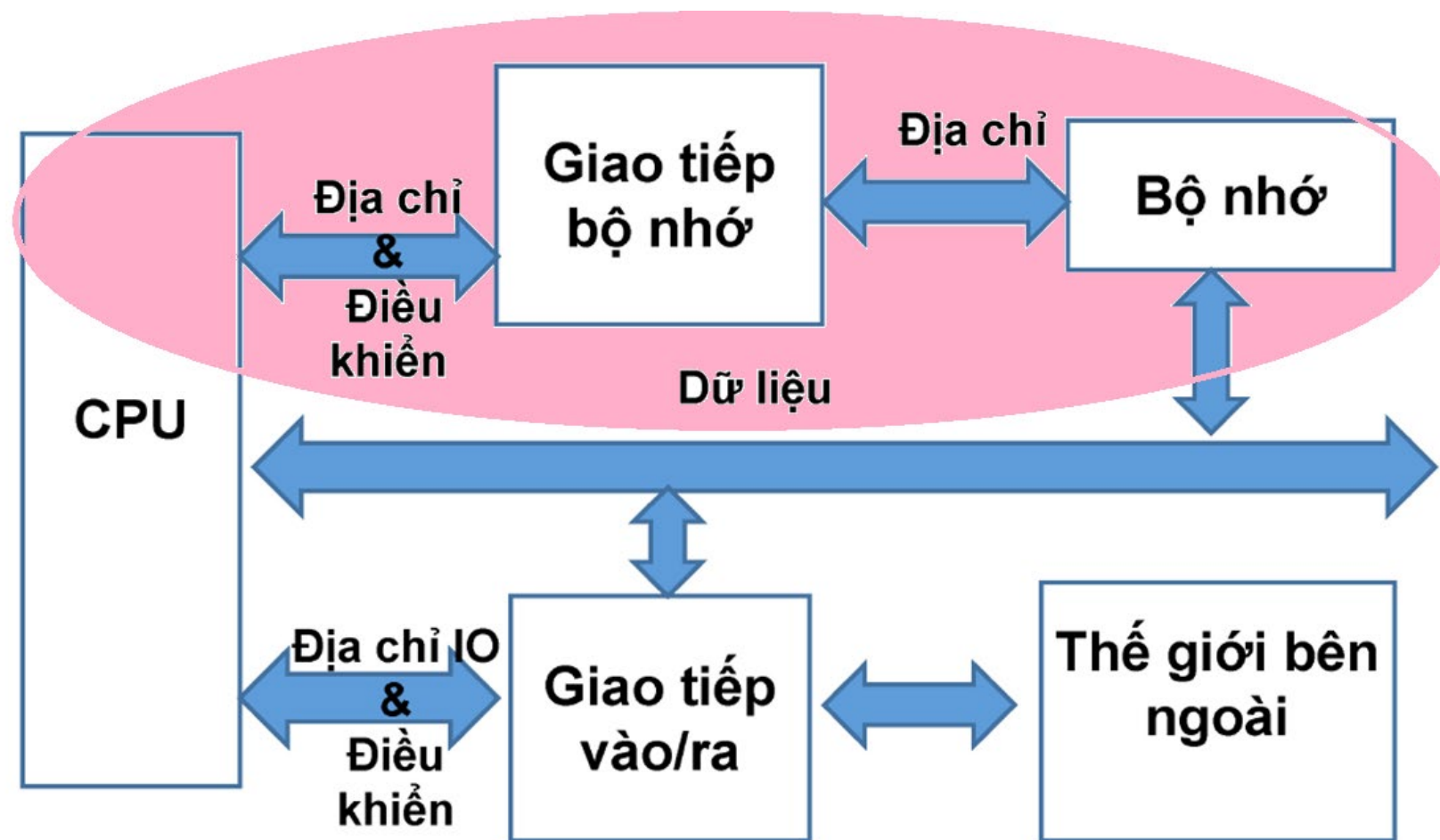
5.1.4. Chu trình ghi bus





5.1 Phối ghép CPU với bộ nhớ

5.1.5. Phối ghép CPU với bộ nhớ



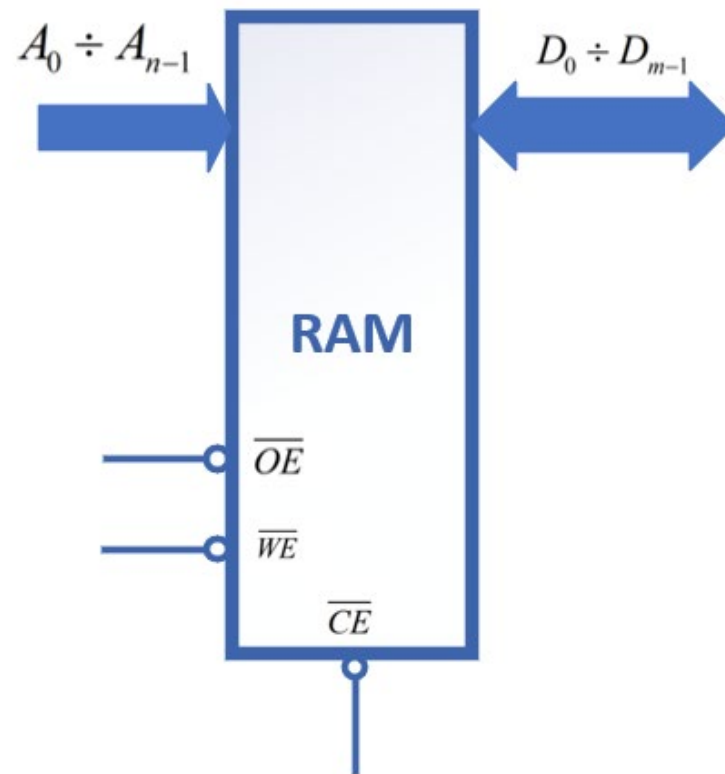


5.1 Phối ghép CPU với bộ nhớ

Cấu trúc chung của một vi mạch nhớ

Một chip nhớ được chế tạo bao gồm:

- Ma trận nhớ
- Các mạch logic giải mã địa chỉ cho ô nhớ
- Các mạch logic cho phép đọc/ghi vào ô nhớ
- Các mạch vào ra, mở rộng địa chỉ



Sơ đồ cấu trúc chung của một mạch nhớ RAM



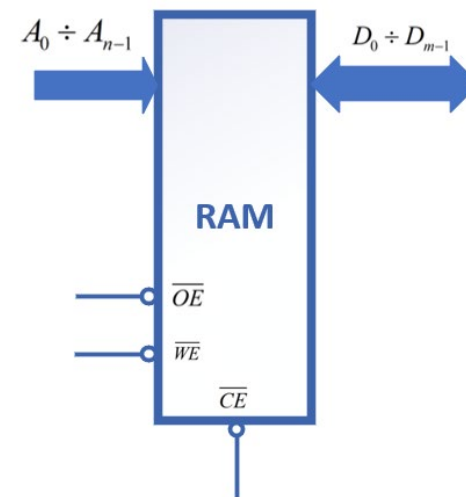
5.1 Phối ghép CPU với bộ nhớ

Cấu trúc chung của một vi mạch nhớ

- Dung lượng vi mạch nhớ là số bit thông tin tối đa mà bộ nhớ có thể lưu trữ nó.
- Đối với bộ nhớ máy tính, đơn vị nhớ được tính theo **byte** còn với chip nhớ, đơn vị nhớ được tính theo **bit**.

Các nhóm tín hiệu của một vi mạch nhớ:

- **Nhóm tín hiệu địa chỉ:** Bao gồm n đường địa chỉ, số lượng đường địa chỉ liên quan tới dung lượng của bộ nhớ theo công thức: **Dung lượng** (tính theo byte) = 2^n .
- Các ô nhớ liên kề nhau sẽ có địa chỉ kế tiếp nhau.
 - Nếu có n bit để biểu diễn cho một địa chỉ thì số địa chỉ tối đa có thể có là 2^n (từ địa chỉ 0 tới địa chỉ $2^n - 1$)
 - Tín hiệu địa chỉ được biểu diễn dưới dạng nhị phân là: $A_{n-1} A_{n-2} \dots A_2 A_1 A_0$



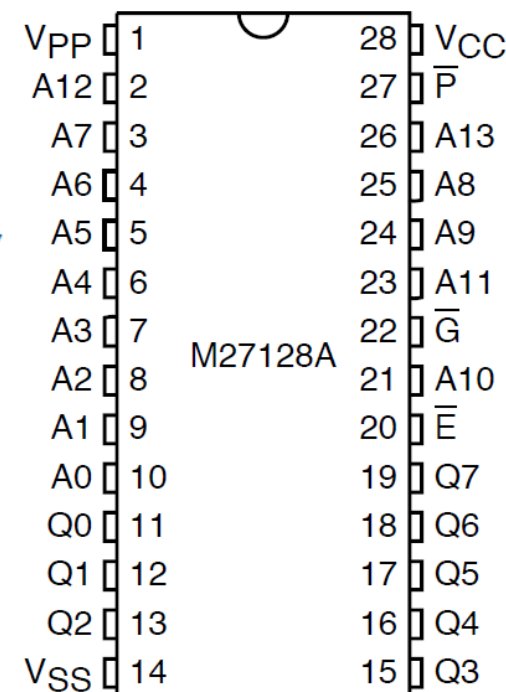
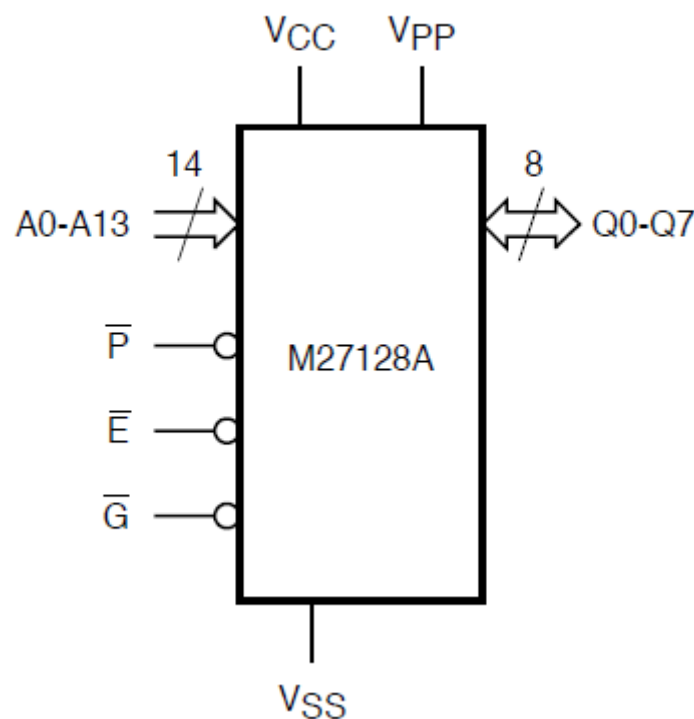
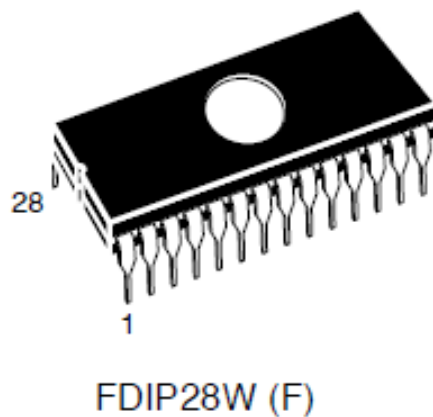
Sơ đồ cấu trúc chung của một mạch nhớ RAM



5.1 Phối ghép CPU với bộ nhớ

Nhóm tín hiệu dữ liệu: Bao gồm m đường dữ liệu, số lượng đường dữ liệu liên quan tới độ dài của ô nhớ (chính bằng số bit trong một ô nhớ được biểu diễn dưới dạng nhị phân là $D_{m-1} D_{m-2} \dots D_2 D_1 D_0$).

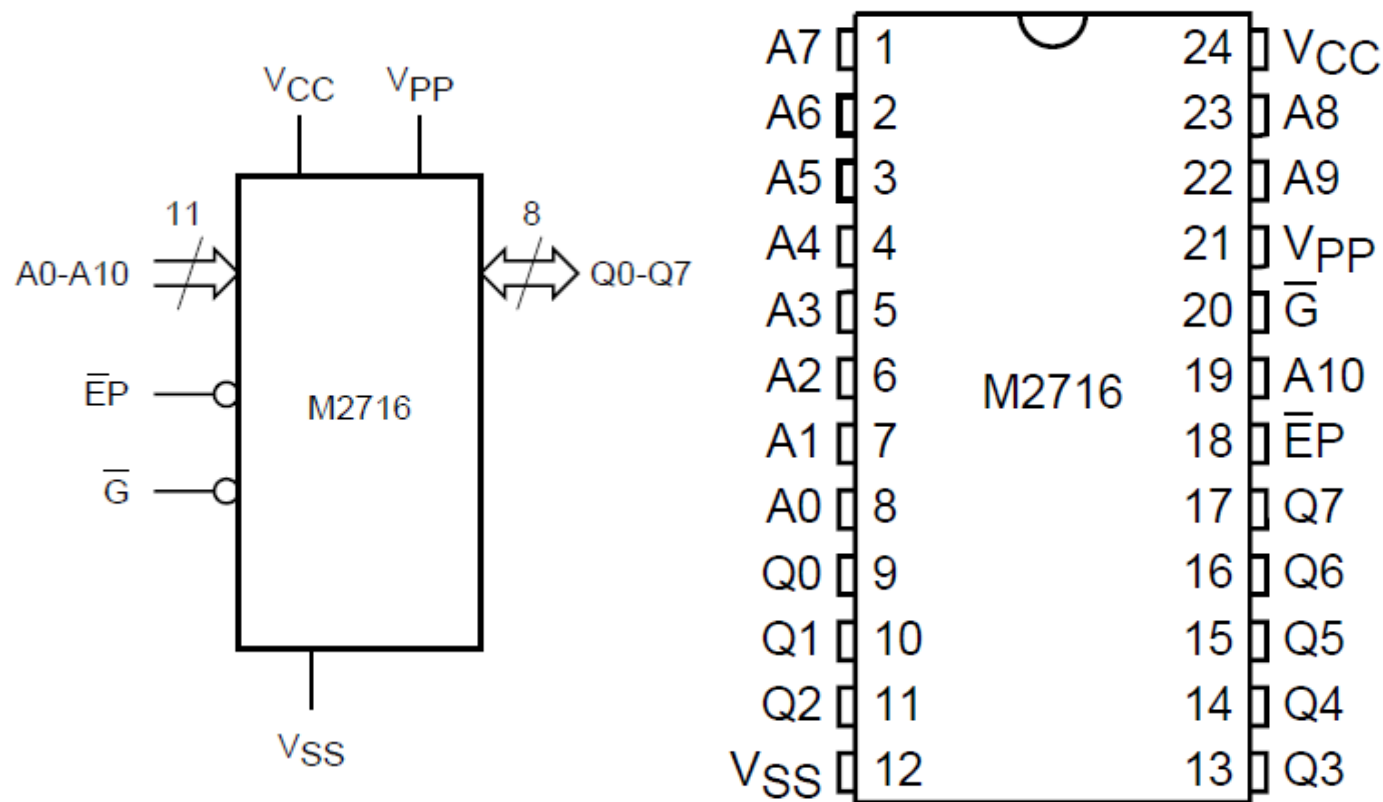
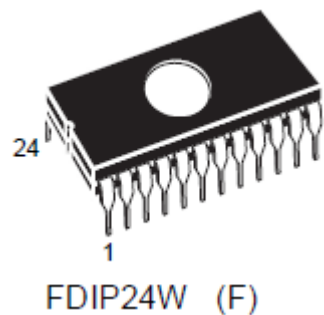
Ví dụ: Chip nhớ UV EPROM M27128A có dung lượng 128Kbit ($16Kb \cdot 8 = 16K$ ô nhớ, mỗi ô nhớ 8 bit \rightarrow 16Kbyte). Bộ nhớ EPROM này có số đường địa chỉ $n=14$ và số đường dữ liệu $m=8$





5.1 Phối ghép CPU với bộ nhớ

Ví dụ: Chip nhớ UV EPROM M2716 có dung lượng 16Kbit ($2K \times 8 = 2K$ ô nhớ, mỗi ô nhớ 8bít \rightarrow 2Kbyte). Bộ nhớ EPROM này có số đường địa chỉ $n=11$ và số đường dữ liệu $m=8$.





5.1 Phối ghép CPU với bộ nhớ

Nhóm tín hiệu điều khiển: Gồm có tín hiệu cho phép đọc \overline{OE} (Output Enable) để cho phép dữ liệu được đưa ra. Đối với vi mạch nhớ RAM còn có tín hiệu cho phép ghi \overline{WE} (Write Enable) để cho phép dữ liệu được ghi vào chip nhớ.

- Ngoài ra còn có thể có các tín hiệu chốt địa chỉ hàng \overline{RAS} (Row Access Strobe), tín hiệu chốt địa chỉ cột \overline{CAS} (Column Access Strobe) đối với DRAM.
- Các chân tín hiệu này đều được tác động tích cực ở mức thấp.

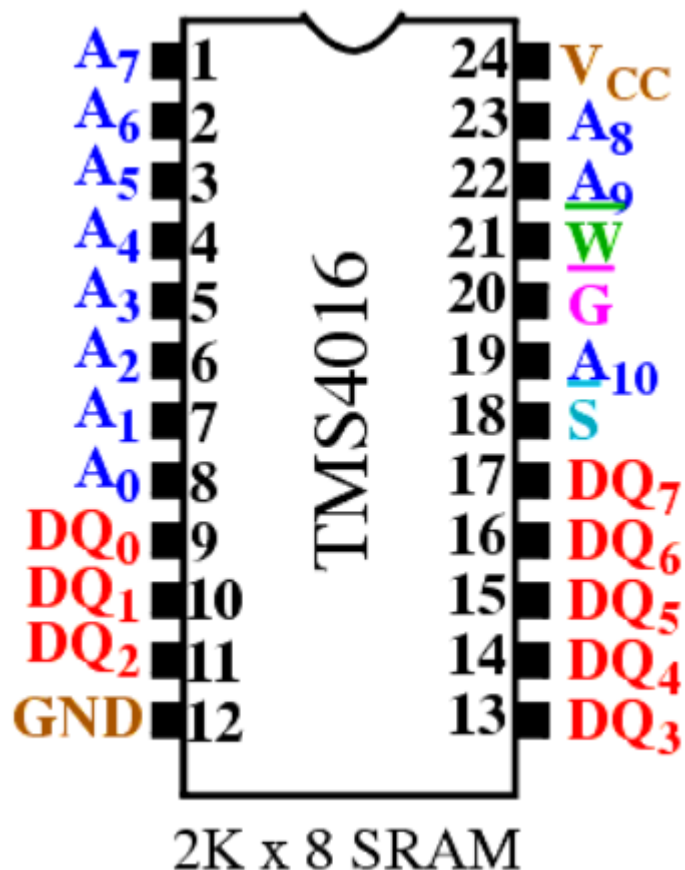
Nhóm tín hiệu chọn vi mạch: Gồm có tín hiệu chọn chip \overline{CS} (Chip Select) hay cho phép vi mạch hoạt động \overline{CE} (Chip Enable).

- Các tín hiệu chọn vi mạch thường được nối với đầu ra của mạch giải mã địa chỉ.
- Khi một vi mạch nhớ không được chọn thì bus dữ liệu của nó bị treo.



5.1 Phối ghép CPU với bộ nhớ

Ví dụ: Chip nhớ SRAM TMS4016 có dung lượng $2K \times 8$. Bộ nhớ SRAM này có số đường địa chỉ $n=11$ và số đường dữ liệu $m=8$.

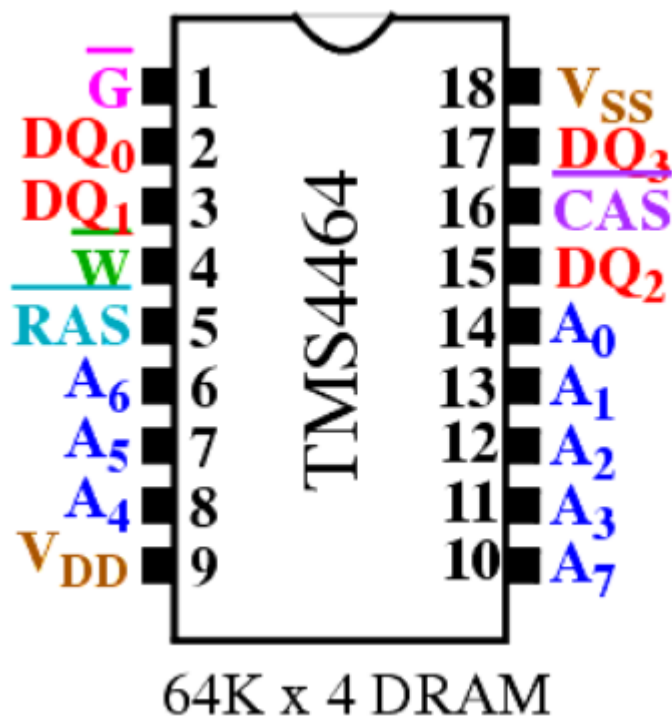


Pin(s)	Function
A_0-A_{10}	Address
DQ_0-DQ_7	Data In/Data Out
\overline{S} (\overline{CS})	Chip Select
\overline{G} (\overline{OE})	Read Enable
\overline{W} (\overline{WE})	Write Enable



5.1 Phối ghép CPU với bộ nhớ

Ví dụ: Chip nhớ DRAM TMS4464 có dung lượng 64K*4. Bộ nhớ SRAM này có số đường địa chỉ $n=8$ và số đường dữ liệu $m=4$.



Pin(s)	Function
A_0-A_7	Address
DQ_0-DQ_3	Data In/Data Out
\overline{RAS}	Row Address Strobe
\overline{CAS}	Column Address Strobe
\overline{G}	Output Enable
\overline{W}	Write Enable



5.1 Phối ghép CPU với bộ nhớ

Giải mã địa chỉ bộ nhớ:

- Một bộ vi xử lý thường có thể quản lý không gian bộ nhớ lớn hơn rất nhiều so với dung lượng của một vi mạch nhớ ROM, RAM. Để có thể gắn các vi mạch nhớ này vào trong không gian địa chỉ của bộ vi xử lý thì chúng ta cần các mạch giải mã địa chỉ.
- *CPU sử dụng một Bus dữ liệu duy nhất để làm việc với tất cả các bộ nhớ và các cổng vào/ra*, vì thế các IC nhớ cần được giải mã địa chỉ sao cho chỉ có một bộ nhớ được chọn tương ứng với một địa chỉ bộ nhớ mà CPU cung cấp. **Ví dụ:** Vi xử lý Intel 8086/8088 sử dụng bus địa chỉ **20 bit** như vậy tổng không gian bộ nhớ mà nó có thể quản lý được lên tới **1 Mega byte**. Tuy nhiên, **BIOS** của máy tính dùng vi mạch **EPROM 2716** chỉ có dung lượng nhớ 2Kbyte với 11 đường dây địa chỉ.

Làm sao có thể ghép nối chúng với nhau để vi xử lý có thể truy cập được không gian nhớ của BIOS ?

→ *Sử dụng các mạch giải mã để thêm 9 chân địa chỉ và cho phép EPROM được đặt ở bất kỳ đoạn 2KB nào trong không gian nhớ 1MB.*



5.1 Phối ghép CPU với bộ nhớ

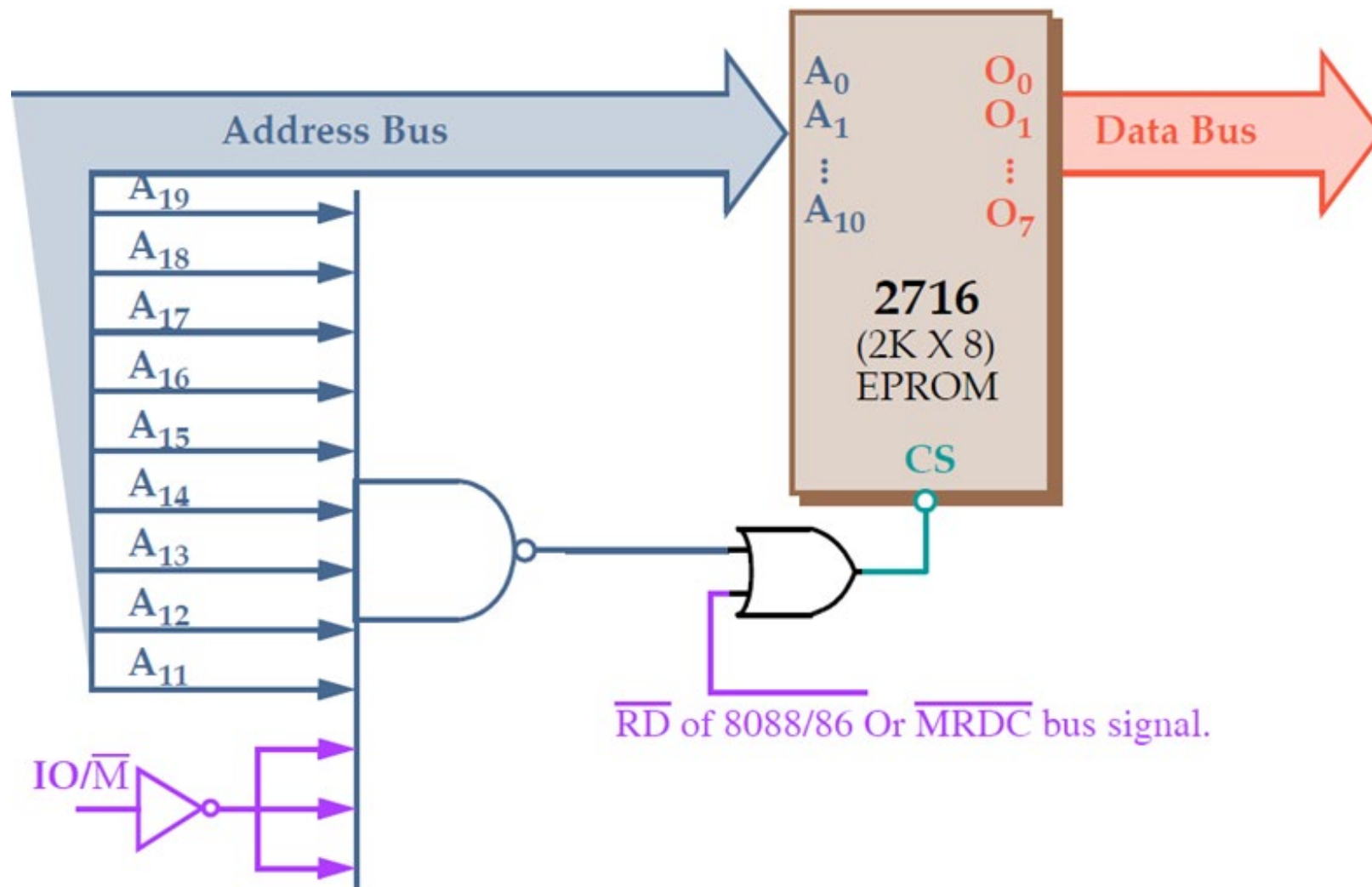
Các yêu cầu đối với mạch giải mã địa chỉ bộ nhớ

- Mỗi ô nhớ của một vi mạch nhớ khi ghép nối với CPU đều được gán một địa chỉ xác định bằng mạch giải mã bên trong vi mạch nhớ. Với mỗi một vi mạch nhớ phải được gán với một vùng địa chỉ riêng, xác định nằm trong không gian tổng thể của bộ nhớ. Việc gán địa chỉ cho vi mạch nhớ được thực hiện nhờ một xung lấy từ *mạch giải mã địa chỉ*.
- Mạch giải mã địa chỉ luôn phải đảm bảo đồng thời cả hai điều kiện:
 - ✓ Nếu vi xử lý đưa ra địa chỉ của một ô nhớ nằm trong vi mạch nhớ nào đó đã được giải mã thì phải truy cập đọc/ghi được ô nhớ đó (chân tín hiệu $\overline{\text{CE}}$ phải ở mức tích cực).
 - ✓ Nếu vi xử lý đưa ra địa chỉ của ô nhớ nằm ngoài vi mạch nhớ nào đó đã được giải mã thì sẽ không truy cập tới được ô nhớ đó, chân tín hiệu $\overline{\text{CE}}$ của vi mạch phải ở mức **không tích cực** (nhằm tránh xung đột bộ nhớ).



5.1 Phối ghép CPU với bộ nhớ

Mạch giải mã địa chỉ cho 2KB EPROM BIOS





5.1 Phối ghép CPU với bộ nhớ

Thiết kế mạch giải mã địa chỉ bộ nhớ

- Việc thiết kế mạch giải mã địa chỉ bộ nhớ chính là việc ánh xạ các tín hiệu địa chỉ thành tín hiệu chọn chip

(kích hoạt chip nhớ): $A_{19}, A_{18}, A_{17}, \dots, A_n \rightarrow CS_0, CS_1, CS_2, \dots, CS_n$

- Có hai kiểu giải mã: Giải mã đầy đủ và Giải mã rút gọn (giải mã thiếu)

- Giải mã đầy đủ: Sử dụng tất cả các đường địa chỉ: $A_{19}, A_{18}, A_{17}, \dots, A_0$
- Giải mã rút gọn: không sử dụng hết 20 đường địa chỉ: $A_{19}, A_{18}, A_{17}, \dots, A_n, n > 0$

- Mạch giải mã địa chỉ có thể được xây dựng từ các cổng logic cơ sở (các cổng NAND, OR,..) hoặc sử dụng các vi mạch giải mã chuyên dụng như 74LS138, 74LS139.... Cũng có thể dùng các PLD (Programmable Logic Device). Các PLD (1970s) có thể có 3 dạng

- ✓ PLA (Programmable Logic Array)
- ✓ PAL (Programmable Array Logic)
- ✓ GAL (Gated Array Logic)



5.1 Phối ghép CPU với bộ nhớ

Các bước thiết kế mạch giải mã địa chỉ bộ nhớ

Trường hợp dùng các cổng logic cơ sở:

Bước 1: Phân tích yêu cầu bài toán: số chip nhớ, địa chỉ bắt đầu, kết thúc, khả năng mở rộng...vv

Bước 2: Lập bảng phân giải địa chỉ cụ thể

Bước 3: Vẽ sơ đồ logic mạch điện ghép nối CPU và các chip nhớ

.



5.1 Phối ghép CPU với bộ nhớ

Các bước thiết kế mạch giải mã địa chỉ bộ nhớ

Ví dụ: Máy tính dùng chip vi xử lý Intel8086. Hãy xây dựng mạch giải mã địa chỉ cho vi mạch nhớ EPROM 2716 có dung lượng nhớ $2K \times 8$. Vi mạch nhớ được gán địa chỉ bắt đầu từ FF800H.

Chú ý: Các vi mạch nhớ ROM trong hệ thống máy tính thường là chứa các chương trình BIOS, mỗi loại CPU sẽ quy định cụ thể địa chỉ lệnh đầu tiên mà CPU thực hiện (ngay sau khi cấp nguồn hoặc Reset). Vì vậy, bộ nhớ ROM phải được thiết kế nằm trong vùng nhớ có địa chỉ này. Các bộ nhớ RAM có thể nằm tại các vị trí bất kỳ trên các vùng nhớ còn lại, nhưng chúng cần được xác định rõ vùng địa chỉ để không sử dụng sai khi thực hiện phần mềm



5.1 Phối ghép CPU với bộ nhớ

Ví dụ: Máy tính dùng chip vi xử lý Intel8086. Hãy xây dựng mạch giải mã địa chỉ cho vi mạch nhớ EPROM 2716 có dung lượng nhớ $2K \times 8$. Vi mạch nhớ được gán địa chỉ bắt đầu từ FF800H.

Giải

Bước 1: Phân tích yêu cầu bài toán

- Bộ vi xử lý Intel 8086 sử dụng 20 bit địa chỉ nên có thể quản lý bộ nhớ có dung lượng tối đa bằng $2^{20} = 1\text{Mbyte}$.
- Vi mạch 2716 là vi mạch nhớ ROM có dung lượng $16\text{Kbit} = 2K \times 8\text{bit} = 2\text{Kbyte} = 2^{11}$ byte với 11 đường địa chỉ và 8 đường dữ liệu nên ta cần 11 đường dây địa chỉ từ A_0 đến A_{10} để quản lý từng ô nhớ (mỗi ô nhớ 8 bit). Số đường dây địa chỉ còn lại (9 đường từ A_{11} đến A_{19}) được tổ hợp với nhau để đưa vào chân chọn chip \overline{CS} .



5.1 Phối ghép CPU với bộ nhớ

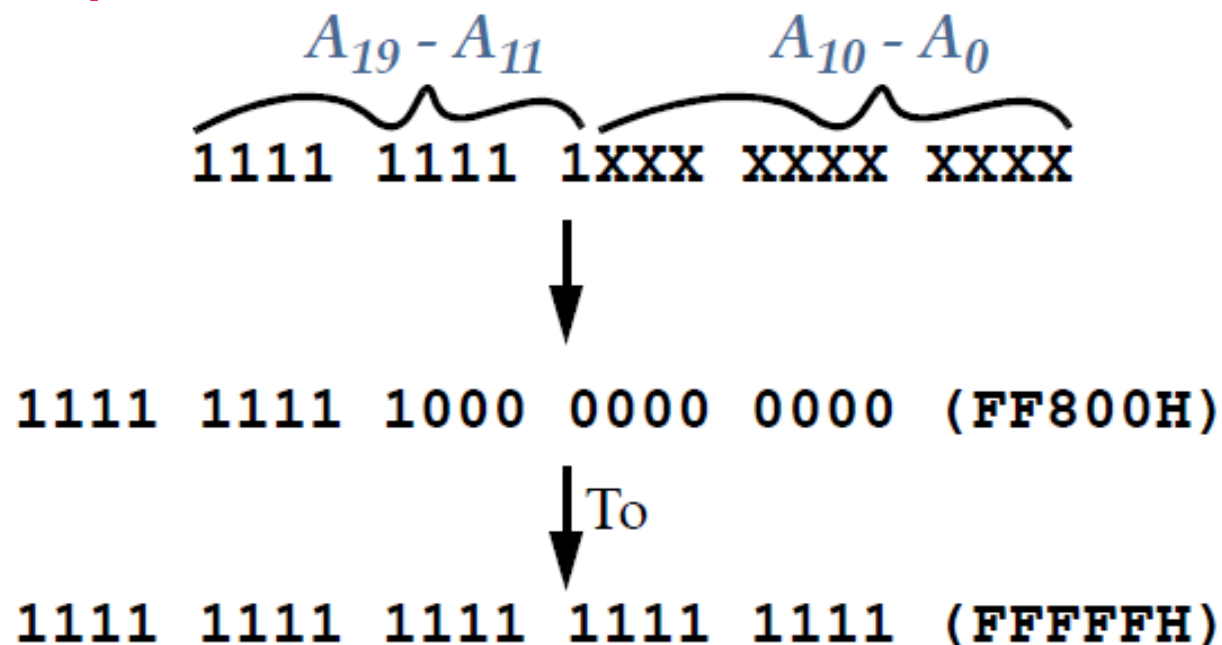
Bước 2: Lập bảng phân giải địa chỉ

A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	FF800h
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	
...	1	1	
...	
...	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFFh



5.1 Phối ghép CPU với bộ nhớ

Bước 2: Lập bảng phân giải địa chỉ

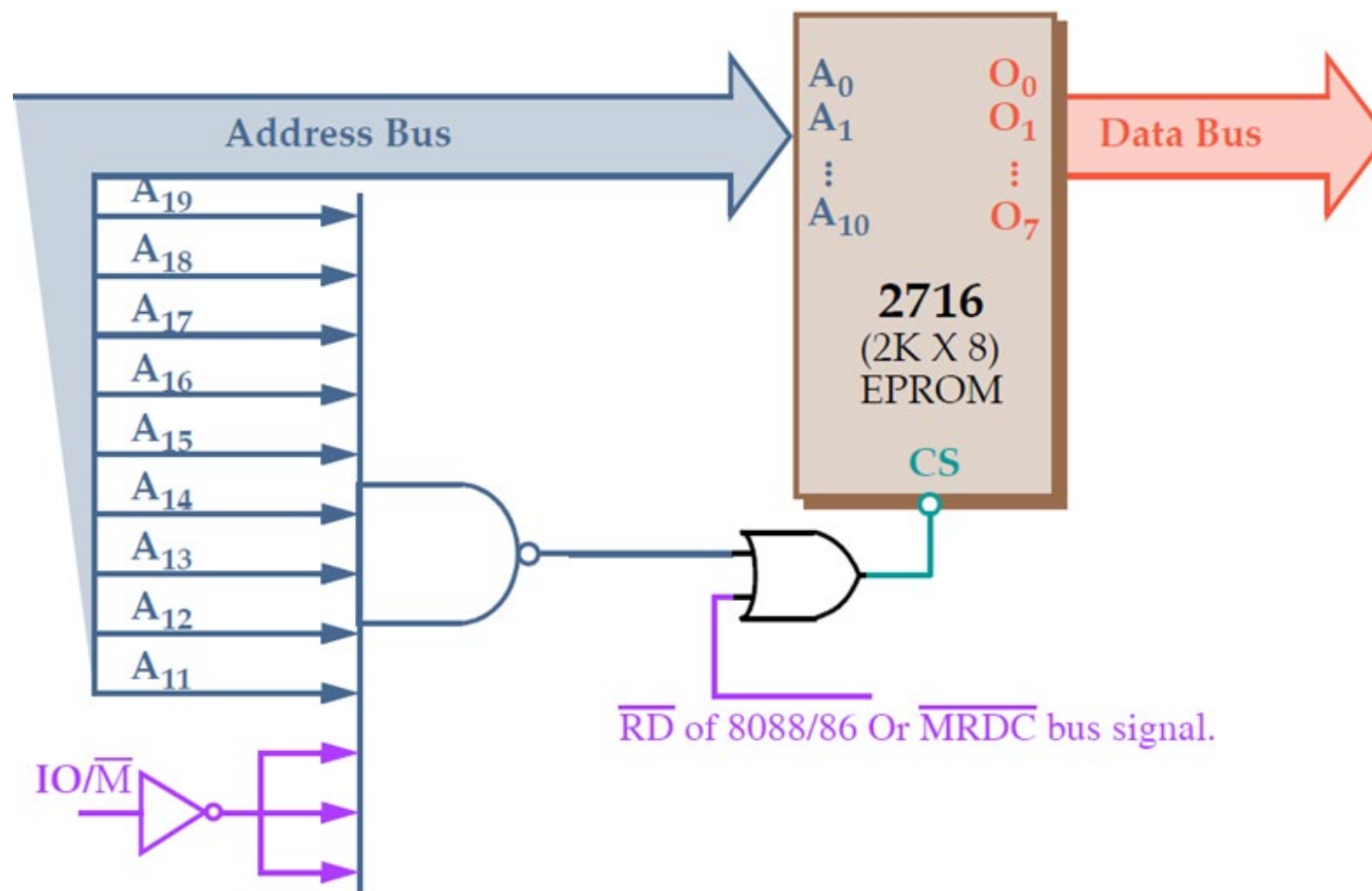


Nhìn vào bảng phân giải địa chỉ của chip nhớ ta thấy bit địa chỉ từ A_0 tới A_{10} thay đổi theo từng ô nhớ \rightarrow đưa vào bus địa chỉ. Còn các bit A_{11} tới A_{19} cố định bằng 1, các bit này sẽ được tổ hợp để tạo ra tín hiệu mức logic thấp đưa tới chân chọn chip \overline{CS}



5.1 Phối ghép CPU với bộ nhớ

Bước 3: Vẽ sơ đồ logic mạch điện ghép nối CPU và chip nhớ





Ví dụ: Máy tính dùng chip vi xử lý Intel8086. Hãy xây dựng mạch giải mã địa chỉ cho vi mạch nhớ SRAM có dung lượng nhớ 4K*8 để thành bộ nhớ 4K*16. Các vi mạch nhớ được gán địa chỉ bắt đầu từ 7C000H.

Giải

Bước 1: Phân tích yêu cầu bài toán

- Bộ vi xử lý Intel 8086 sử dụng 20 bit địa chỉ nên có thể quản lý tối đa bộ nhớ có dung lượng $2^{20} = 1\text{Mbyte}$.
- Bus dữ liệu của 1 chip nhớ SRAM này là 8 bit mà ta cần ghép với bus dữ liệu 16 bit, do vậy cần ghép *song song* 2 chip nhớ có độ lớn dữ liệu 8 bit để tạo ra dữ liệu 16 bit.
- Số chip nhớ cần mắc *song song*: $m = \frac{\text{Độ rộng bus dữ liệu mong muốn}}{\text{Độ rộng bus dữ liệu của 1 chip}} = \frac{16}{8} = 2$
- Dải địa chỉ bắt đầu là 7C000h và có độ lớn 4KB do vậy sẽ kết thúc ở 7CFFFh (chính là 7C000h + 4KB).
 - Địa chỉ bắt đầu ở dạng nhị phân là: 7C000h \rightarrow 0111 1100 0000 0000 0000b
 - Địa chỉ kết thúc ở dạng nhị phân là: 7CFFFh \rightarrow 0111 1100 1111 1111 1111b



5.1 Phối ghép CPU với bộ nhớ

Bước 2: Lập bảng phân giải địa chỉ

A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7C000h
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
...	1	1	
...	
...	
0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	7CFFFh



5.1 Phối ghép CPU với bộ nhớ

Bước 2: Vẽ sơ đồ logic mạch điện ghép nối CPU và chip nhớ

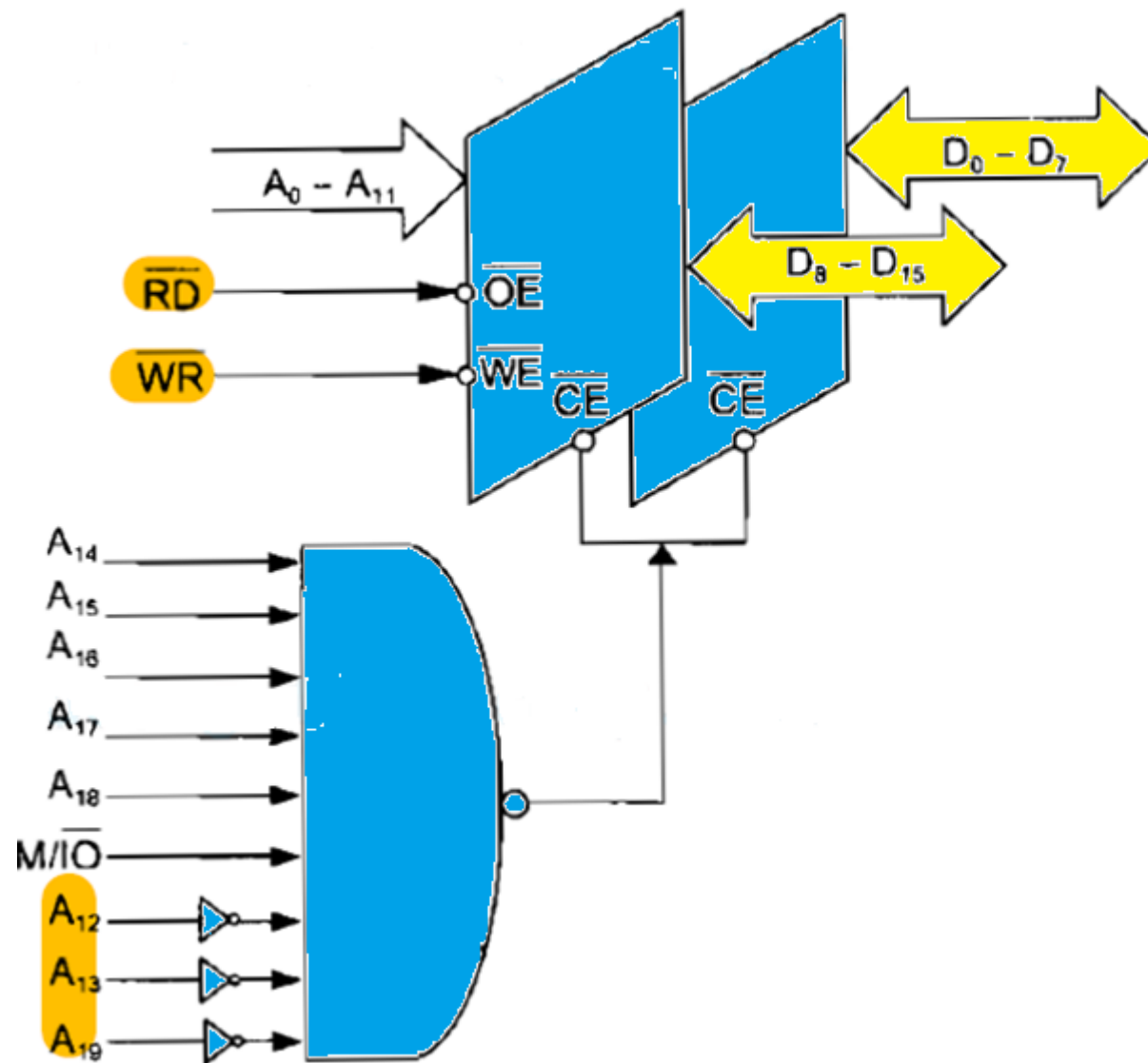
➤ **Xác định các tín hiệu địa chỉ trực tiếp vào chip nhớ:**

- ☐ Phần thay đổi trong dải địa chỉ sẽ đưa trực tiếp vào các bus địa chỉ của IC nhớ SRAM từ A_0 đến A_{11}
- ☐ Phần địa chỉ không thay đổi sẽ đưa vào mạch giải mã để tạo ra tín hiệu chọn vi mạch SRAM từ A_{12} đến A_{19}
- ☐ Xây dựng mạch giải mã dùng cổng logic cơ sở NAND. Tín hiệu đưa trực tiếp vào mạch NAND là các tín hiệu $A_{14}, A_{15}, A_{16}, A_{17}$ và A_{18} cùng với tín hiệu M/\overline{IO} của 8086.
 - ☐ Tín hiệu đưa vào cổng NAND thông qua mạch đảo (NOT) là A_{12}, A_{13} và A_{19} .

➤ **Vẽ sơ đồ logic mạch điện**



5.1 Phối hợp CPU với bộ nhớ



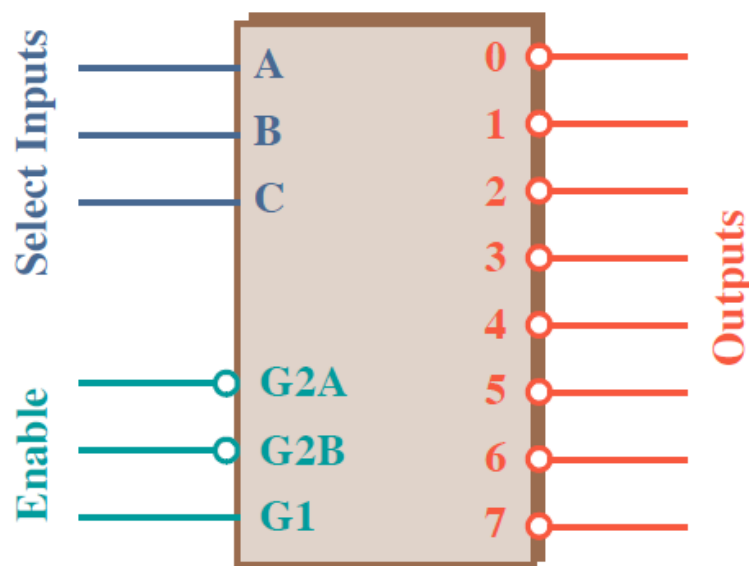


5.1 Phối ghép CPU với bộ nhớ

Các bước thiết kế mạch giải mã địa chỉ bộ nhớ

Mạch giải mã địa chỉ dùng các cổng logic cơ sở sẽ trở nên cồng kềnh khi cần giải mã cho nhiều vi mạch nhớ. Vì vậy, người ta ít dùng cổng logic mà thường dùng các vi mạch giải mã có sẵn.

Vi mạch giải mã 74LS138

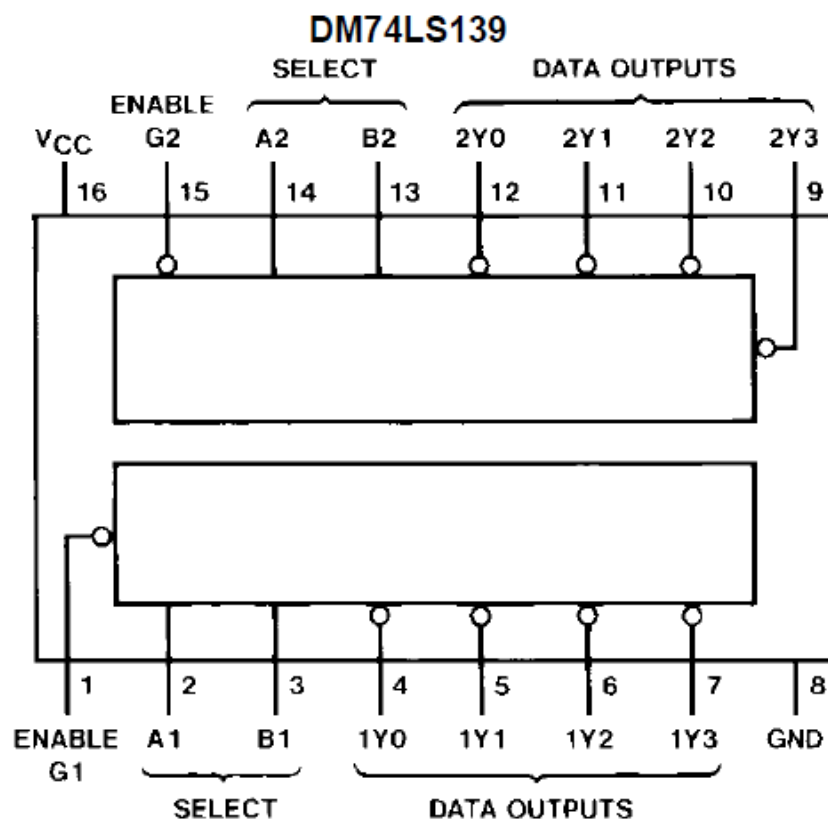


Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0



5.1 Phối ghép CPU với bộ nhớ

Vì mạch giải mã **74LS139**



DM74LS139

Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L



Ví dụ: Xây dựng mạch giải mã địa chỉ cho **4** vi mạch nhớ RAM (**4K*8bit**) liên kề nhau. Địa chỉ đầu của vi mạch nhớ thứ nhất là **E0000h**

Giải

Bước 1: Phân tích yêu cầu bài toán

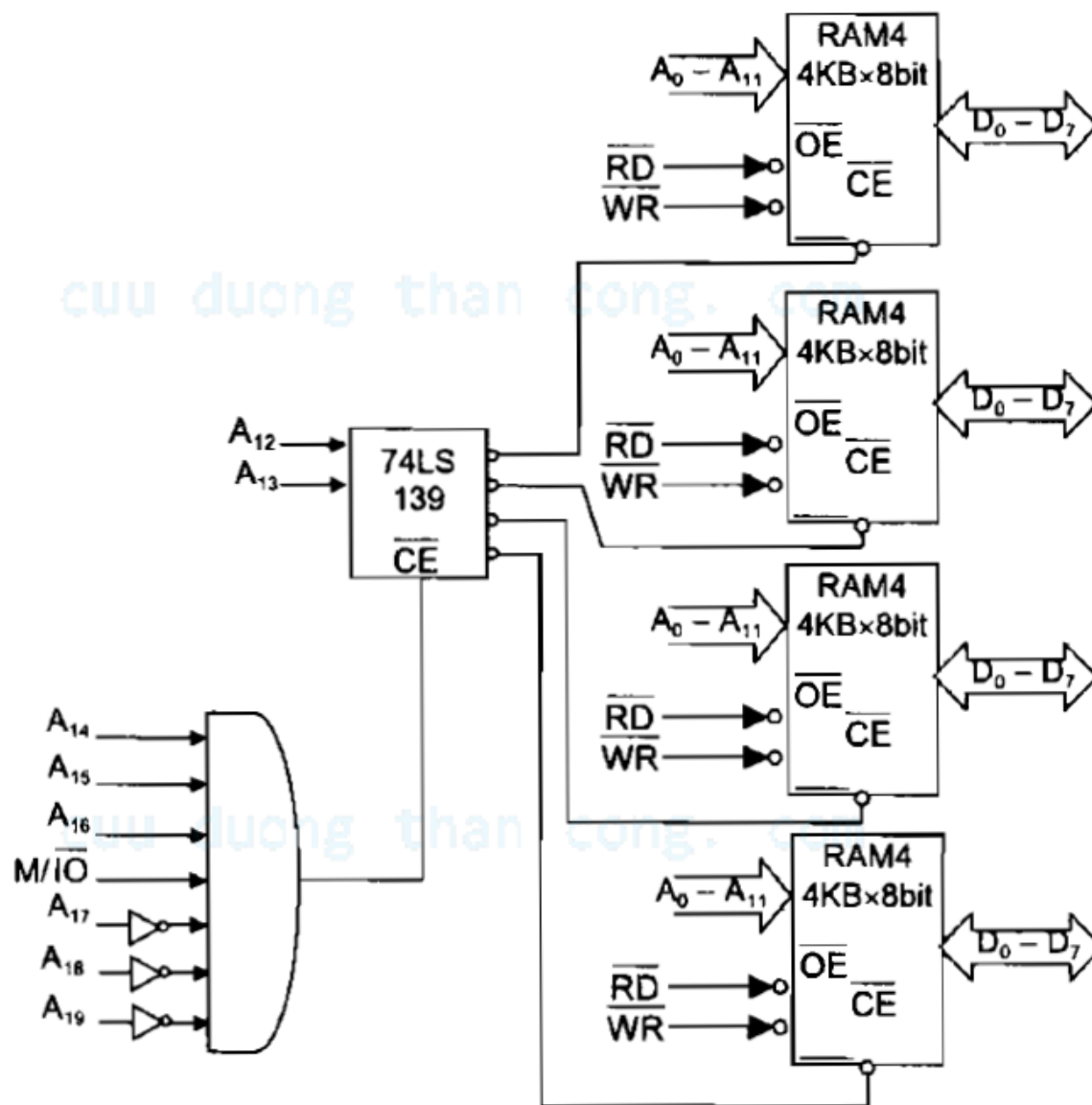
- Có 4 chip nhớ, số lượng này trùng khớp với số đầu ra của vi mạch giải mã **74LS139** (vào 2 ra 4), do vậy ta sử dụng vi mạch này để làm mạch giải mã.

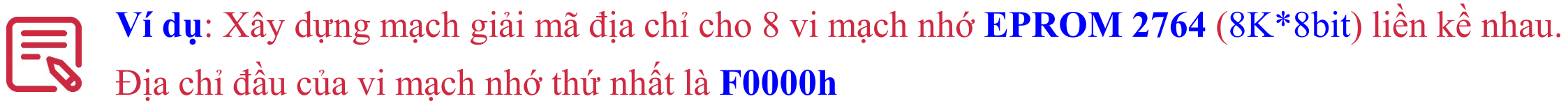
Bước 2: Lập bảng phân giải địa chỉ: Giống như các ví dụ trước (sinh viên tự làm)

Bước 3: Vẽ sơ đồ logic mạch điện ghép nối CPU và chip nhớ



5.1 Phối ghép CPU với bộ nhớ





Ví dụ: Xây dựng mạch giải mã địa chỉ cho 8 vi mạch nhớ **EPROM 2764** (8K*8bit) liên kề nhau.
Địa chỉ đầu của vi mạch nhớ thứ nhất là **F0000h**

Giải

Bước 1: Phân tích yêu cầu bài toán

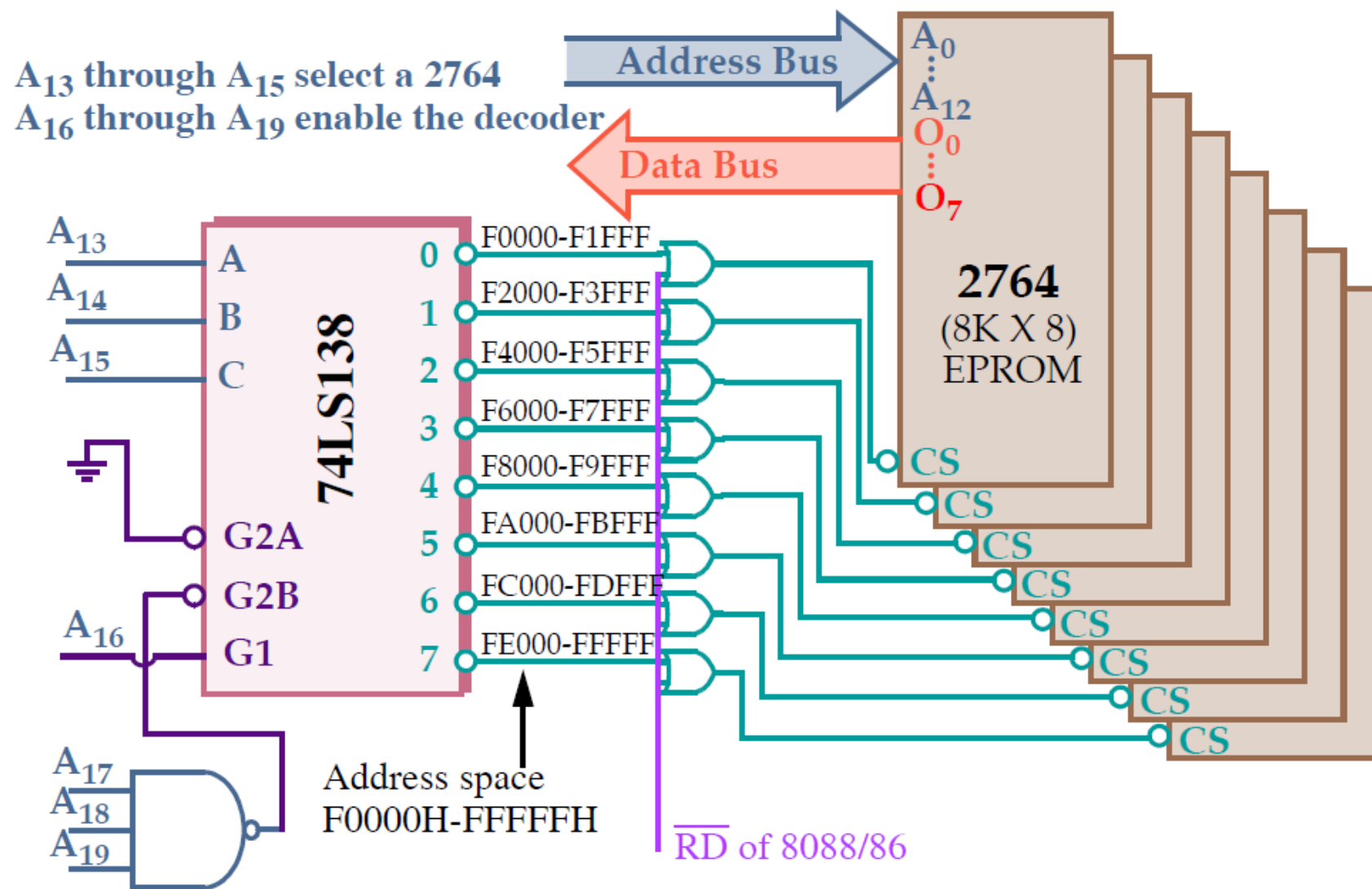
- Có 4 chip nhớ, số lượng này trùng khớp với số đầu ra của vi mạch giải mã 74LS138, do vậy ta sử dụng vi mạch này để làm mạch giải mã.

Bước 2: Lập bảng phân giải địa chỉ: Giống như các ví dụ trước (sinh viên tự làm)

Bước 3: Vẽ sơ đồ logic mạch điện ghép nối CPU và chip nhớ



5.1 Phối ghép CPU với bộ nhớ

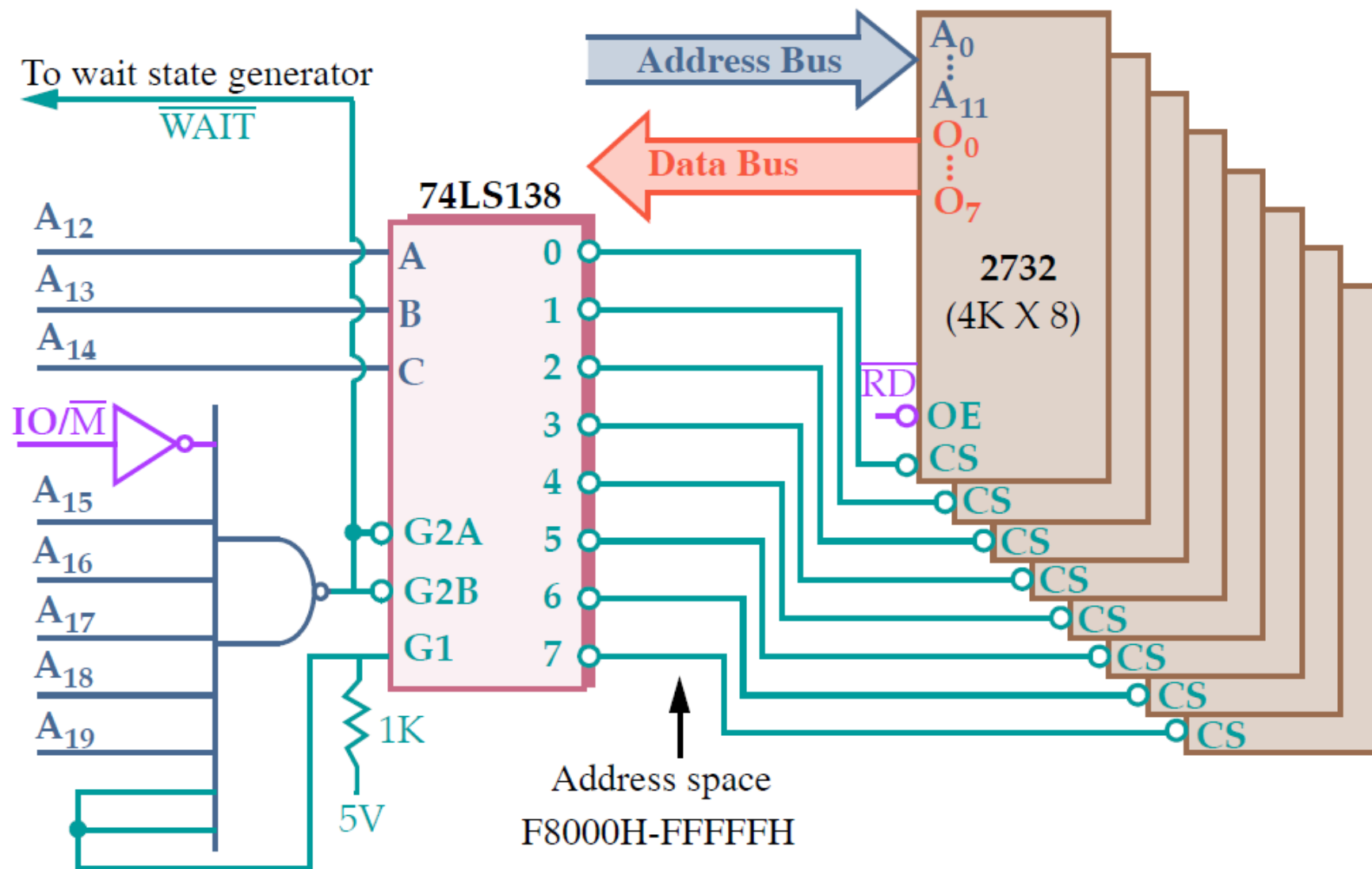


The EPROMs cover a 64KB section of memory.



5.1 Phối ghép CPU với bộ nhớ

Ví dụ: Mạch giải mã dùng 74LS138 cho 32K*8bit EPROM bắt đầu từ địa chỉ **F8000h** tới **FFFFFFh** dùng 8 vi mạch 4K*8bit EPROM **2732**



The 8088 cold starts execution at **FFFF0H**. JMP to F8000H occurs here.



5.1 Phối ghép CPU với bộ nhớ

Các bước thiết kế mạch giải mã địa chỉ bộ nhớ

Trường hợp dùng các PLD (Programmable Logic Device)

PLD: Xuất hiện từ những năm 1970s nhưng gần đây mới có mặt trong các hệ thống mạch điện bộ nhớ (các PAL đã thay thế cho các bộ giải mã dùng PROM)

PAL và PLA: Lập trình bằng phương pháp đốt cầu chì (giống PROM). Một số PAL, PAL có thể xóa được giống EPROM.

PAL 16L8 được sử dụng làm mạch giải mã cho bộ nhớ của vi xử lý 32 bit của Intel 80386DX và các thế hệ sau.

.

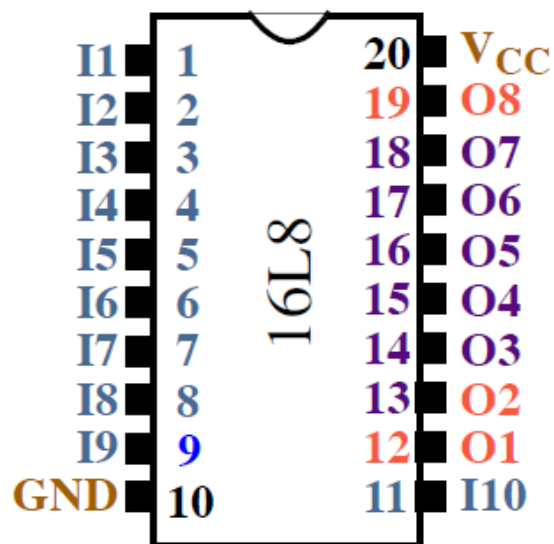


5.1 Phối ghép CPU với bộ nhớ

AMD 16L8 PAL decoder.

It has 10 fixed inputs (Pins 1-9, 11), two fixed outputs (Pins 12 and 19) and 6 pins that can be either (Pins 13-18).

Programmed to decode address lines $A_{19} - A_{13}$ onto 8 outputs.



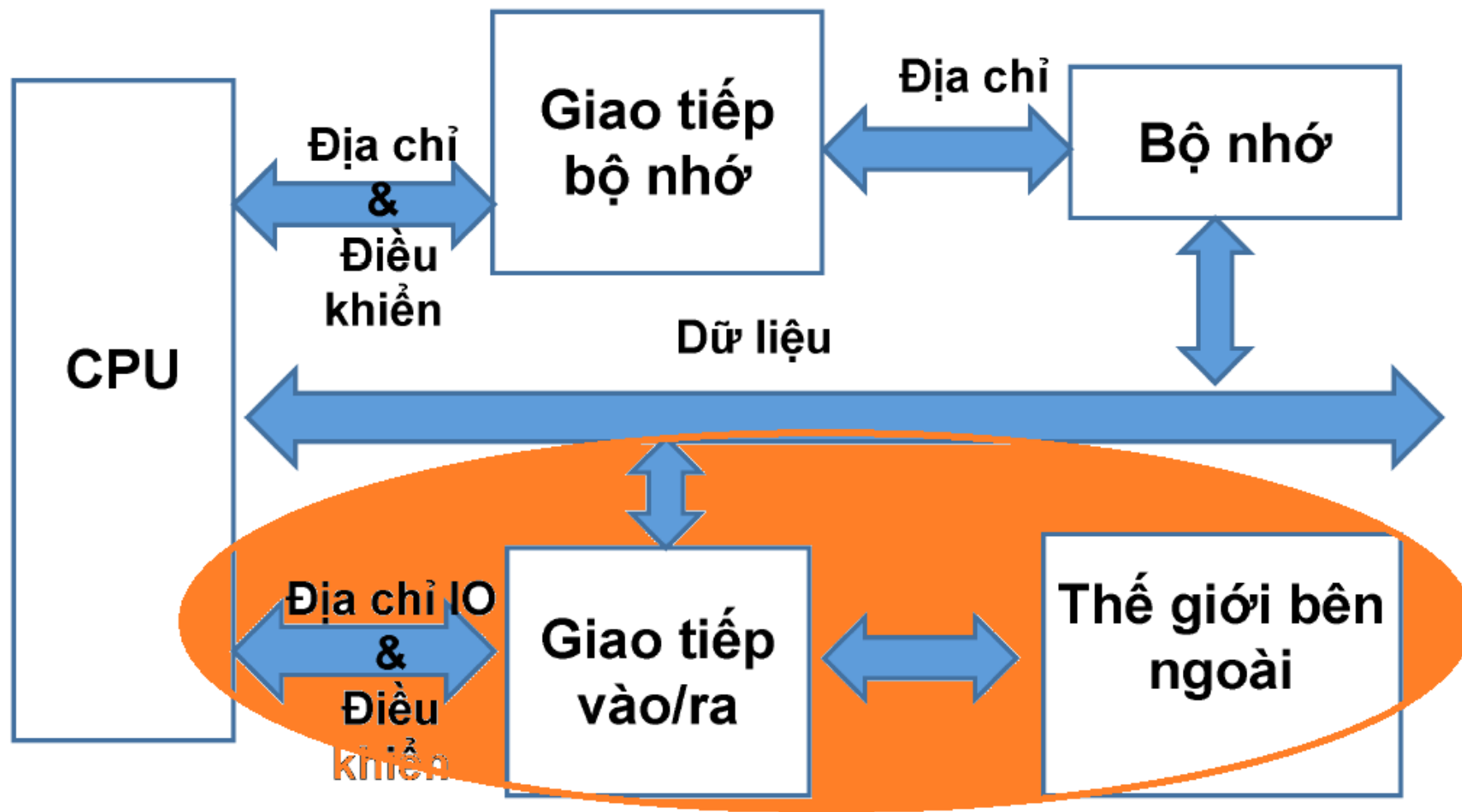
<i>;pins</i>	1	2	3	4	5	6	7	8	9	10
	A19	A18	A17	A16	A15	A14	A13	NC	NC	GND
<i>;pins</i>	11	12	13	14	15	16	17	18	19	20
	NC	O8	O7	O6	O5	O4	O3	O2	O1	VCC

Equations:

$$\begin{aligned}
 /O1 &= A19 * A18 * A17 * A16 * /A15 * /A14 * /A13 \\
 /O2 &= A19 * A18 * A17 * A16 * /A15 * /A14 * A13 \\
 /O3 &= A19 * A18 * A17 * A16 * /A15 * A14 * /A13 \\
 /O4 &= A19 * A18 * A17 * A16 * /A15 * A14 * A13 \\
 /O5 &= A19 * A18 * A17 * A16 * A15 * /A14 * /A13 \\
 /O6 &= A19 * A18 * A17 * A16 * A15 * /A14 * A13 \\
 /O7 &= A19 * A18 * A17 * A16 * A15 * A14 * /A13 \\
 /O8 &= A19 * A18 * A17 * A16 * A15 * A14 * A13
 \end{aligned}$$



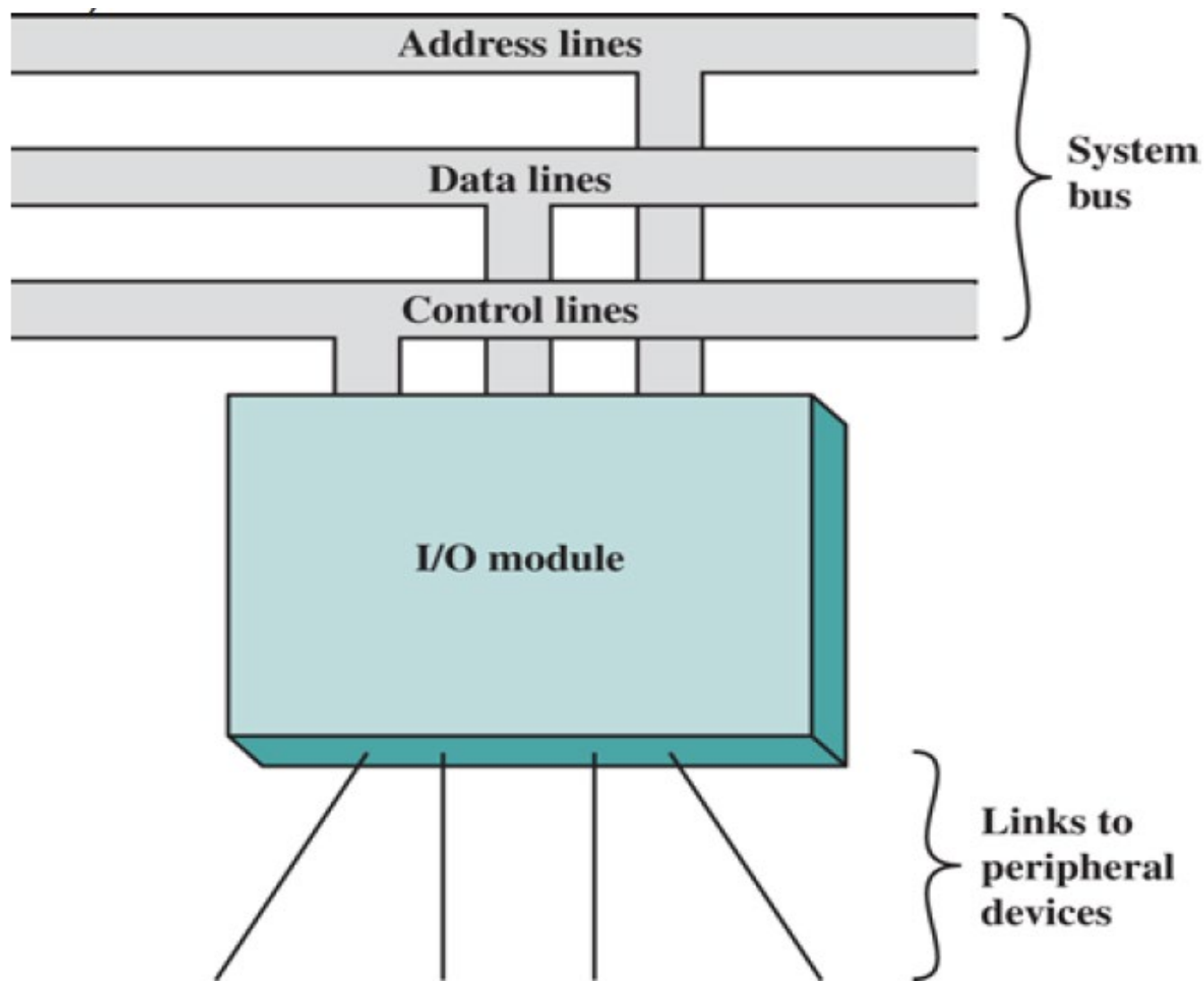
5.2 Phối hợp CPU với thiết bị ngoại vi





5.2 Phối ghép CPU với thiết bị ngoại vi

Sơ đồ khối vị trí một module I/O ghép nối với bus hệ thống và thiết bị ngoại vi →





5.2 Phối ghép CPU với thiết bị ngoại vi

- Các thiết bị ngoại vi làm việc với CPU thông qua các Cổng (Port). Mỗi cổng có địa chỉ riêng.
- Để ghép nối CPU với các thiết bị ngoại vi (thế giới bên ngoài) cần sử dụng các Module I/O làm nhiệm vụ phối hợp hoạt động giữa CPU, bộ nhớ RAM và 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 mạch điện tử thực hiện chức năng trao đổi thông tin giữa thiết bị ngoại vi và bus.
- Module I/O có 2 chức năng chính:
 - ✓ Giao tiếp với bộ xử lý và bộ nhớ thông qua bus hệ thống hoặc một chuyên mạch trung tâm.
 - ✓ Giao tiếp với một hoặc nhiều thiết bị ngoại vi bằng các liên kết dữ liệu phù hợp

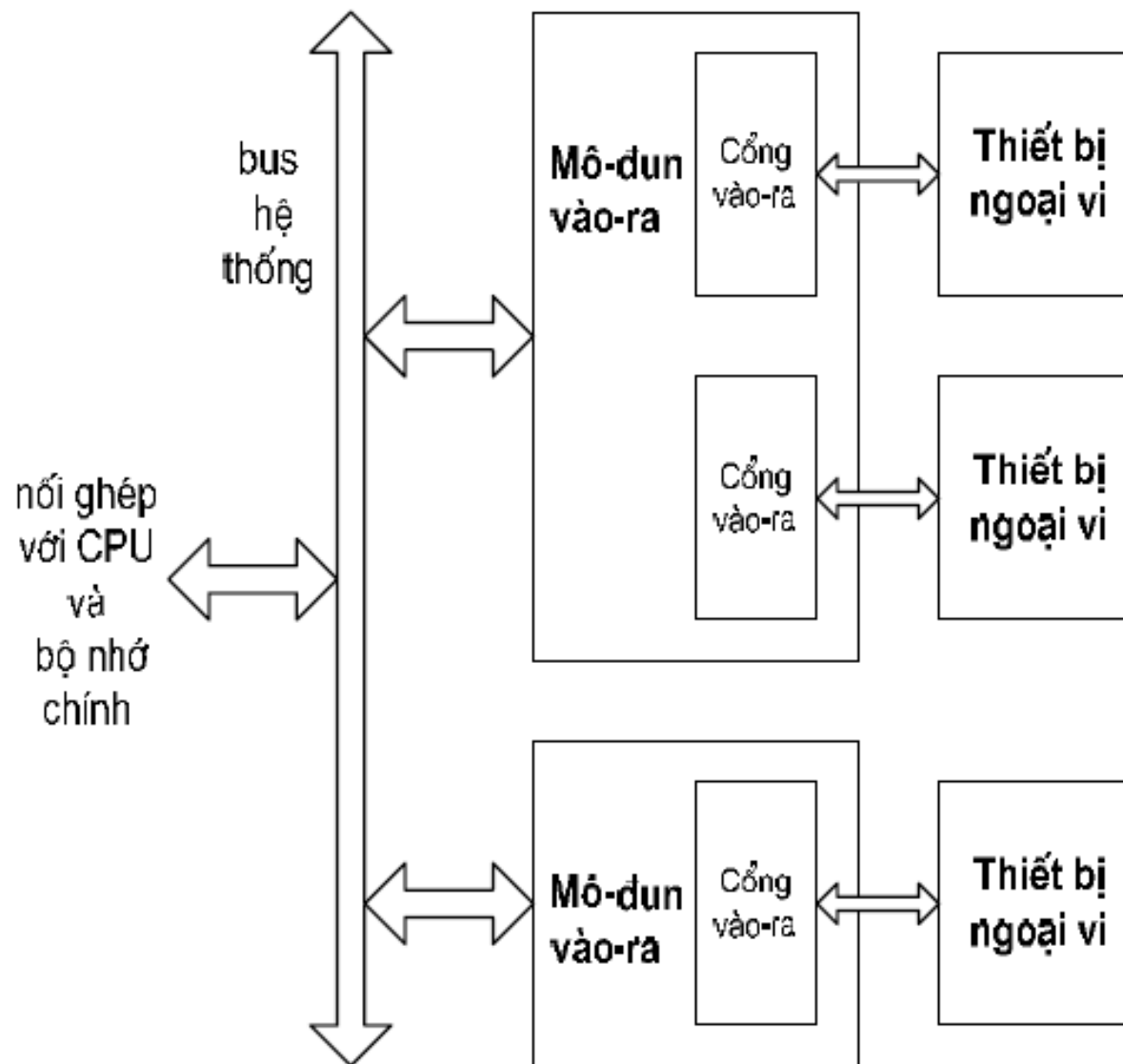
Tại sao không kết nối trực tiếp thiết bị ngoại vi với bus hệ thống ?

- *Phương thức hoạt động khác nhau*
- *Tốc độ truyền dữ liệu khác nhau*
- *Định dạng dữ liệu của thiết bị ngoại vi khác với bus hệ thống*



5.2 Phối ghép CPU với thiết bị ngoại vi

Sơ đồ khối vị trí một module I/O ghép nối với bus hệ thống và thiết bị ngoại vi →





5.2 Phối ghép CPU với thiết bị ngoại vi

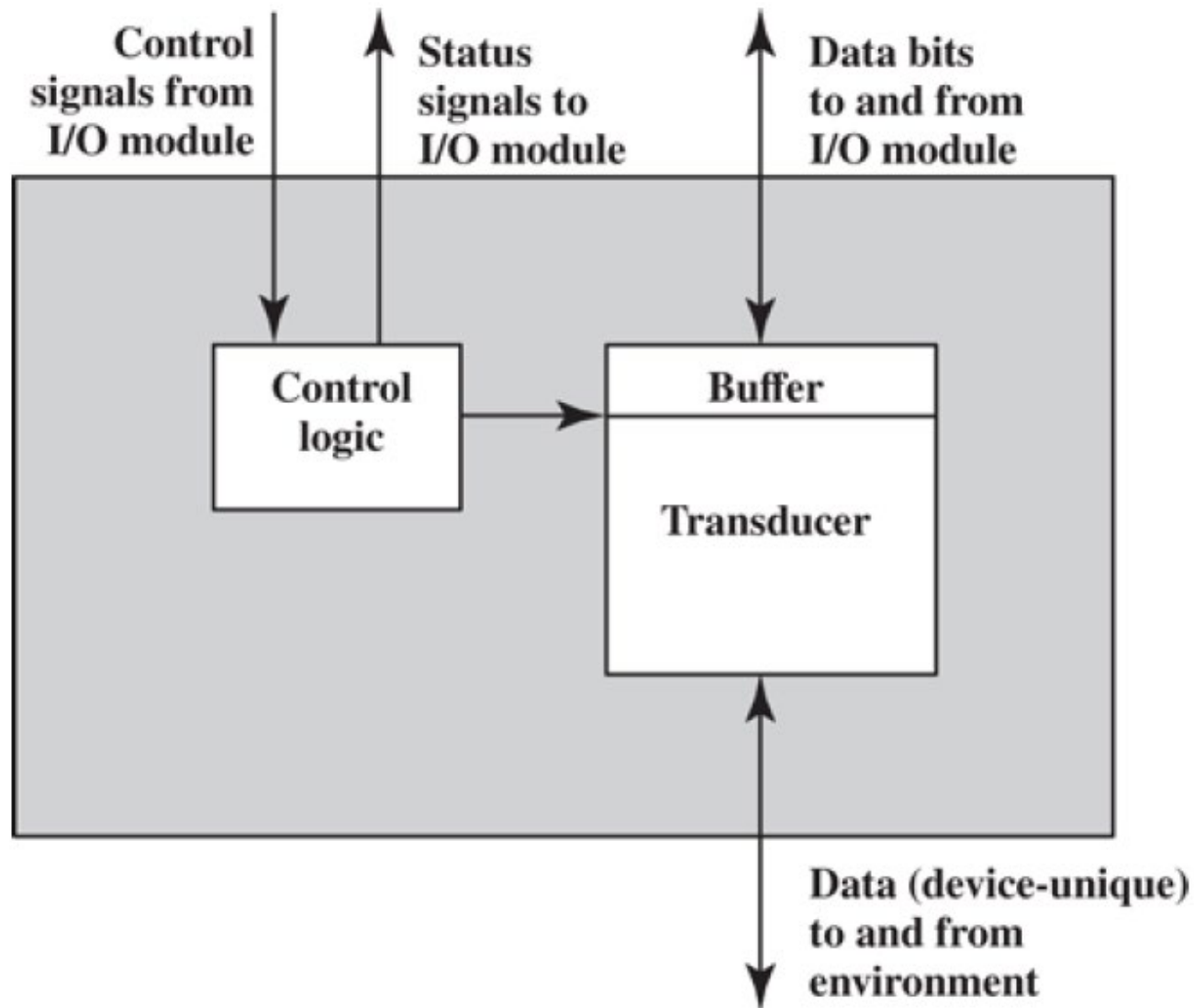
5.2.1 Các thiết bị ngoại vi

- Các hoạt động I/O được thực hiện thông qua nhiều loại thiết bị bên ngoài cung cấp phương tiện trao đổi dữ liệu giữa môi trường bên ngoài và máy tính.
- Một thiết bị bên ngoài gắn vào máy tính bằng liên kết đến mô-đun I/O. Liên kết được sử dụng để trao đổi các tín hiệu điều khiển, trạng thái và dữ liệu giữa mô-đun I/O và thiết bị bên ngoài.
- Một thiết bị bên ngoài được kết nối với mô-đun I/O thường được gọi là thiết bị ngoại vi.
- Có 3 loại thiết bị ngoại vi:
 - ✓ Thiết bị ngoại vi giao tiếp **Người-Máy** (*Human readable*): màn hình, máy in, bàn phím,...
 - ✓ Thiết bị ngoại vi giao tiếp **Máy-Máy** (*Machine readable*): camera, máy đọc thẻ nhớ, vân tay,..
 - ✓ Thiết bị ngoại vi cho mục đích **Truyền thông tin** (*Communication*): modem, NIC, ...vv



5.2 Phối ghép CPU với thiết bị ngoại vi

Sơ đồ khối chức năng của
một thiết bị ngoại vi →





5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.1 Các thiết bị ngoại vi

- 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, tiếp nhận dữ liệu từ module I/O, 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 chuỗi các bit đến hoặc đi từ module I/O
- Các tín hiệu trạng thái cho biết trạng thái của thiết bị. Ví dụ: READY/NOT-READY
- Logic điều khiển nhận tín hiệu điều khiển từ module I/O và 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 (Buffer) để lưu trữ dữ liệu tạm thời trong quá trình trao đổi dữ liệu giữa module I/O và thiết bị vào/ra;
 - Một bộ đệm có kích thước phổ biến từ 8 đến 16 bit.



5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.1 Các thiết bị ngoại vi

Một số ví dụ về các thiết bị vào/ra

Input type	Prime examples	Other examples	Data rate (b/s)	Main uses
Symbol	Keyboard, keypad	Music note, OCR	10s	Ubiquitous
Position	Mouse, touchpad	Stick, wheel, glove	100s	Ubiquitous
Identity	Barcode reader	Badge, fingerprint	100s	Sales, security
Sensory	Touch, motion, light	Scent, brain signal	100s	Control, security
Audio	Microphone	Phone, radio, tape	1000s	Ubiquitous
Image	Scanner, camera	Graphic tablet	1000s-10 ⁶ s	Photos, publishing
Video	Camcorder, DVD	VCR, TV cable	1000s-10 ⁹ s	Entertainment
Output type	Prime examples	Other examples	Data rate (b/s)	Main uses
Symbol	LCD line segments	LED, status light	10s	Ubiquitous
Position	Stepper motor	Robotic motion	100s	Ubiquitous
Warning	Buzzer, bell, siren	Flashing light	A few	Safety, security
Sensory	Braille text	Scent, brain stimulus	100s	Personal assistance
Audio	Speaker, audiotape	Voice synthesizer	1000s	Ubiquitous
Image	Monitor, printer	Plotter, microfilm	1000s	Ubiquitous
Video	Monitor, TV screen	Film/video recorder	1000s-10 ⁹ s	Entertainment
Two-way I/O	Prime examples	Other examples	Data rate (b/s)	Main uses
Mass storage	Hard/floppy disk	CD, tape, archive	10 ⁶ s	Ubiquitous
Network	Modem, fax, LAN	Cable, DSL, ATM	1000s-10 ⁹ s	Ubiquitous



5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.2 Các module vào/ra (I/O)

- Chức năng của các Module I/O

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



5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.2 Các module vào/ra (I/O)

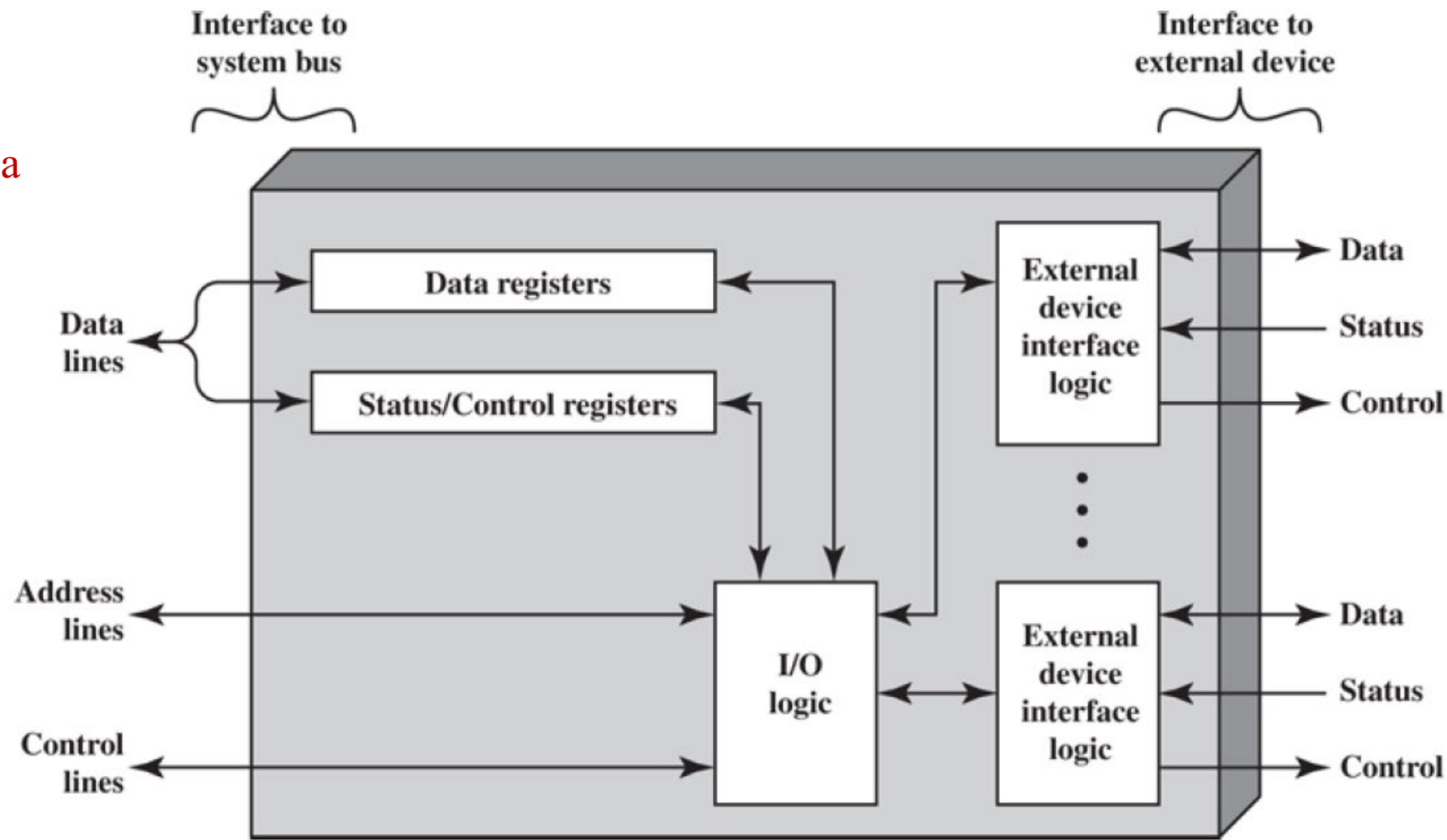
Ví dụ: quá trình điều khiển truyền dữ liệu từ thiết bị ngoại vi đến bộ xử lý gồm các bước sau:

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 vi xử lý.



5.2 Phối ghép CPU với thiết bị ngoại vi

Cấu trúc chung của một module I/O





5.2 Phối ghép CPU với thiết bị ngoại vi

Cấu trúc module I/O

- Module I/O 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 đi từ module I/O 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).
 - Module I/O chứa các logic giao tiếp với từng thiết bị nối vào nó.



5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.3 Các phương pháp địa chỉ hóa cổng vào/ra (I/O)

CPU giao tiếp với Bộ nhớ và Các module I/O đều sử dụng chung bus hệ thống !

Một vấn đề khi sử dụng bus chung đó là: *làm thế nào để CPU có thể biết được đâu là các phần tử vào/ra I/O, đâu là bộ nhớ để có thể truy cập đọc/ghi ?*

→ Có **4 cách** và tùy theo từng bộ vi xử lý khác nhau sẽ lựa chọn cách thức khác nhau.

- Không gian địa chỉ bộ nhớ vật lý mà CPU có thể quản lý sẽ được chia thành hai phần riêng biệt: Một phần cho bộ nhớ chính và một phần cho các phần tử vào/ra.
- CPU hỗ trợ những tín hiệu đặc trưng cho phép truy cập các phần tử vào/ra (tín hiệu M/\overline{IO} ở 8086).
- Trong tập lệnh có những lệnh dành riêng cho hoạt động vào/ra.
- Sử dụng các **Bộ xử lý vào/ra chuyên dụng** - Các kênh (trên các máy tính lớn)



5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.3 Các phương pháp địa chỉ hóa cổng vào/ra (I/O)

- Một số bộ xử lý chỉ quản lý **duy nhất** một không gian địa chỉ: 2^N địa chỉ

Ví dụ:

- Các bộ xử lý 680x0 (Motorola)
- Các bộ xử lý theo kiến trúc RISC: MIPS, ...

→ **Memory Mapped I/O**

(Vào/ra ánh xạ bộ nhớ)



- Một số bộ xử lý quản lý **hai** không gian địa chỉ tách biệt:

Không gian địa chỉ vào-ra: 2^{N_1} địa chỉ

- Tập lệnh có các lệnh vào-ra chuyên dụng

- Không gian địa chỉ bộ nhớ = 2^{32} byte = 4GB

- Không gian địa chỉ vào-ra = 2^{16} byte = 64KB

- Tín hiệu điều khiển

- Lệnh vào-ra chuyên dụng: IN, OUT, INS, OUTS

→ Isolated I/O hay I/O Mapped I/O

(Vào/ra riêng biệt)

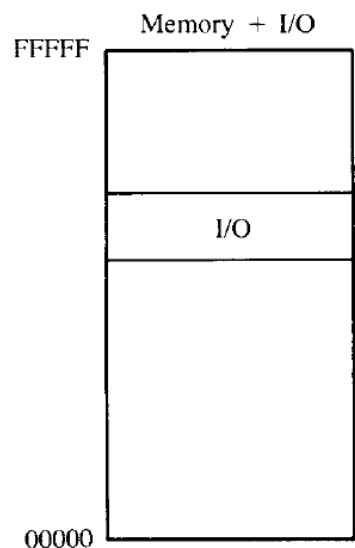




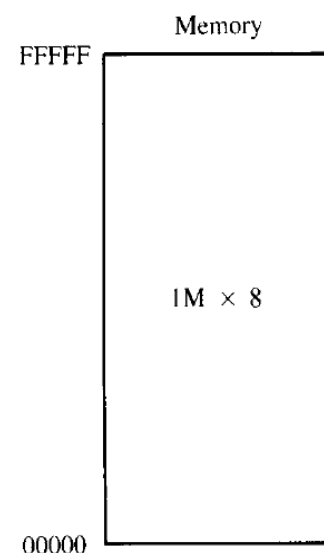
5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.3 Các phương pháp địa chỉ hóa cổng vào/ra (I/O)

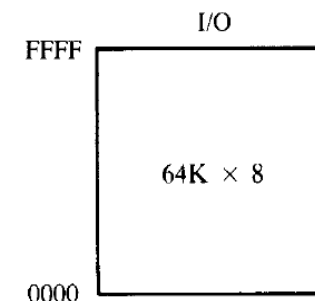
- Một số hệ thống máy vi tính sử dụng cả hai loại I/O— nghĩa là, một số thiết bị ngoại vi được coi là thiết bị I/O riêng biệt và một số khác là thiết bị I/O được ánh xạ bộ nhớ.
- Các phương thức I/O này khác nhau ở cách tham chiếu các cổng I/O trong không gian địa chỉ của vi xử lý.
- Máy tính PC thường dùng phương thức địa chỉ riêng biệt **Isolated I/O**



Memory mapped I/O cho 8086



Isolated I/O cho 8086





5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phương pháp Vào/Ra ánh xạ bộ nhớ (Memory mapped I/O)

- Máy tính sử dụng cùng một không gian địa chỉ bộ nhớ để đánh địa chỉ cho cả bộ nhớ chính và các thiết bị I/O. Vì xử lý coi các cổng I/O như thể nó là một vị trí lưu trữ trong bộ nhớ !
- Bộ nhớ và các thanh ghi của các thiết bị I/O được ánh xạ (liên kết) tới các giá trị địa chỉ bộ nhớ chính, do đó 1 địa chỉ bộ nhớ có thể cho phép truy cập đến ROM, RAM hoặc các thanh ghi của thiết bị I/O.
- Các lệnh của CPU dùng để truy cập bộ nhớ (*Ví dụ: MOV...*) cũng có thể được sử dụng để truy cập thiết bị vào ra I/O.
- Mỗi thiết bị I/O giám sát bus địa chỉ của CPU và đáp ứng bất kỳ yêu cầu truy cập CPU nào của địa chỉ được gán cho thiết bị đó.
 - Mỗi thiết bị I/O kết nối bus hệ thống với thanh ghi của thiết bị đó hoặc sử dụng một vi mạch chuyên dụng để thực hiện điều này.
 - Để chứa các địa chỉ của thiết bị I/O, một số vùng của bus địa chỉ được CPU dành riêng cho I/O và không được dùng cho bộ nhớ vật lý thông thường.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phương pháp Vào/Ra ánh xạ bộ nhớ (Memory mapped I/O)

- Việc đặt trước các địa chỉ riêng này có thể được xác định bằng phần cứng.
- Việc đặt trước có thể là cố định hoặc tạm thời (được thực hiện thông qua việc chuyển đổi các bank nhớ).

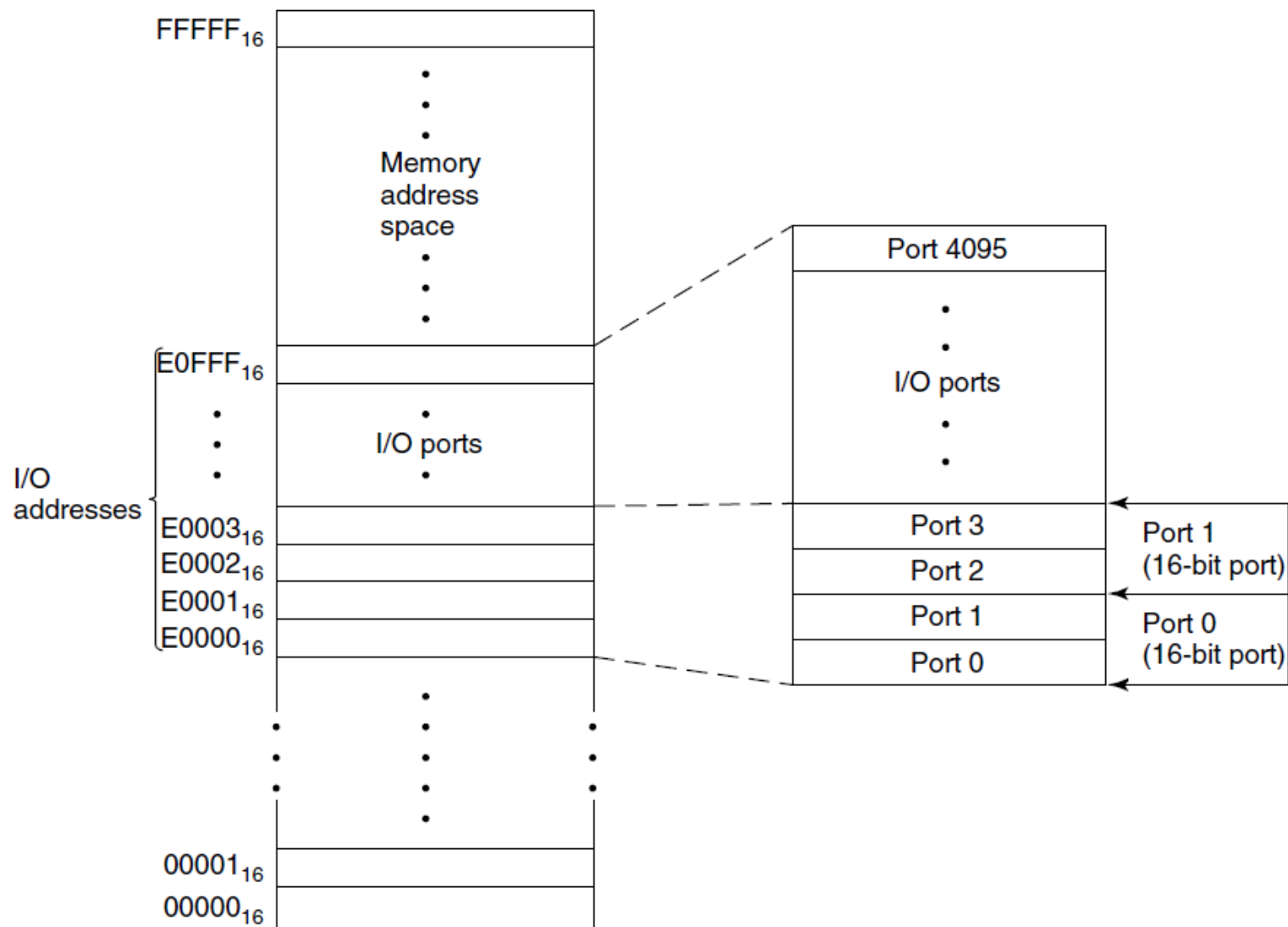
Ví dụ: Máy tính Comodo 64 sử dụng ánh xạ bộ nhớ để khiến cho RAM hoặc phần cứng của thiết bị I/O xuất hiện địa chỉ nằm trong dải địa chỉ 0xD000 tới 0xDFFF.



5.2 Phối ghép CPU với thiết bị ngoại vi

Ví dụ: Phương pháp Vào/Ra ánh xạ bộ nhớ với vi xử lý **Intel 8086/8088**

- 4096 địa chỉ bộ nhớ trong dải từ E0000h tới E0FFFh được gán cho các địa chỉ I/O.
- Nội dung của địa chỉ E0000h tương ứng với cổng I/O số 0 có độ rộng 8 bit.
- Nội dung của địa chỉ ô nhớ E0000h và E0001h tương ứng với cổng I/O số 0 có độ rộng 16 bit.





5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Ưu khuyết điểm của phương pháp Vào/Ra ánh xạ bộ nhớ (Memory mapped I/O)

Ưu điểm:

- Lập trình đơn giản, linh hoạt, sử dụng hiệu quả không gian nhớ. Có thể thực thi trên mọi hệ thống.
- Khi I/O được cấu hình theo cách này, các lệnh ảnh hưởng đến dữ liệu trong bộ nhớ (MOV, XCHG,..) sẽ được sử dụng thay cho các lệnh vào/ra đặc biệt (IN, OUT). Đây là một lợi thế ở chỗ có nhiều lệnh và chế độ địa chỉ hơn để thực hiện các thao tác I/O.
- Không cần các mạch phụ trợ và các tín hiệu đọc/ghi I/O riêng biệt

Ví dụ: nội dung của cổng I/O được ánh xạ bộ nhớ có thể được AND trực tiếp với một giá trị trong thanh ghi nội bộ. Ngoài ra, việc di chuyển dữ liệu với các cổng I/O có thể diễn ra giữa cổng I/O và các thanh ghi nội bộ khác không chỉ giới hạn ở AL hoặc AX.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Ưu khuyết điểm của phương pháp Vào/Ra ánh xạ bộ nhớ (Memory mapped I/O)

Khuyết điểm:

- Các lệnh bộ nhớ có xu hướng thực thi **chậm hơn** so với các lệnh được thiết kế riêng cho I/O riêng biệt. Do đó, quy trình I/O được ánh xạ bộ nhớ có thể mất nhiều thời gian hơn để thực hiện so với một chương trình tương đương sử dụng các lệnh IN, OUT.
- Việc **khó quản lý bộ nhớ đệm** khi đọc ghi dẫn đến các hiệu ứng đọc/ghi I/O ngoài ý muốn.
- Gặp **rào cản về dung lượng bộ nhớ** mà bộ vi xử lý quản lý được (giảm), chậm nếu I/O phản hồi chậm.

Ví dụ: Rào cản 640 KB trong dòng máy tính PC IBM là do dành riêng vùng từ 640KB đến 1024KB cho **vùng bộ nhớ cao**. Điều này đã giới hạn tổng dung lượng RAM có sẵn trong không gian địa chỉ khả dụng 20 bit.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ **Ưu khuyết điểm của phương pháp Vào/Ra ánh xạ bộ nhớ (Memory mapped I/O)**

Ví dụ: Bản đồ bộ nhớ của một hệ thống máy tính với CPU 8 bit

Một hệ thống đơn giản được xây dựng từ bộ vi xử lý 8 bit có thể cung cấp các không gian địa chỉ 16 bit, cho phép nó quản lý bộ nhớ lên tới 64 KB. Trên hệ thống như vậy, 32 KB đầu tiên của không gian địa chỉ có thể được phân bổ cho RAM, 16 KB khác cho ROM và phần còn lại cho nhiều thiết bị khác như bộ hẹn giờ, bộ đếm, video, chip hiển thị, thiết bị tạo âm thanh, v.v.

Address range (hexadecimal)	Size	Device
0000–7FFF	32 KiB	RAM
8000–80FF	256 bytes	General-purpose I/O
9000–90FF	256 bytes	Sound controller
A000–A7FF	2 KiB	Video controller/text-mapped display RAM
C000–FFFF	16 KiB	ROM



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phương pháp Vào/Ra riêng biệt (Isolated I/O hoặc IO mapped I/O)

- Cổng vào/ra được đánh địa chỉ theo không gian địa chỉ vào/ra riêng biệt
- CPU trao đổi dữ liệu với cổng vào/ra thông qua các lệnh vào/ra chuyên dụng (IN, OUT)
- Chỉ có thể thực hiện trên các hệ thống có quản lý không gian địa chỉ vào/ra riêng biệt.

Ví dụ: Với các máy tính dựa trên dòng VXL Intel thì kỹ thuật Vào/Ra riêng biệt được sử dụng phổ biến hơn kỹ thuật ánh xạ bộ nhớ !

Trong máy tính cá nhân (PC) các cổng I/O riêng biệt được sử dụng để điều khiển các thiết bị ngoại vi.

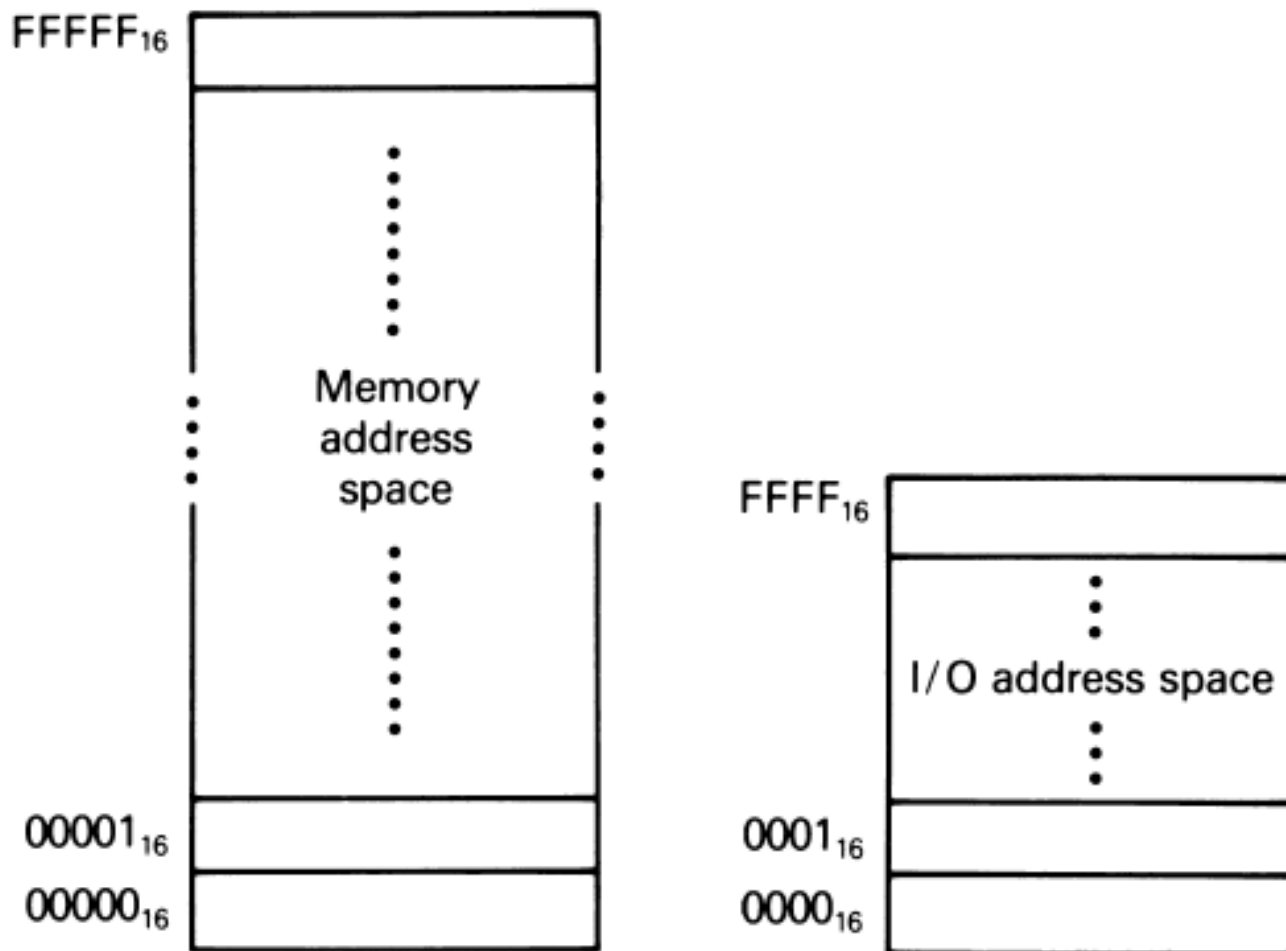
Địa chỉ cổng 8 bit được dùng để truy cập các thiết bị nằm trên Mainboard như bộ hẹn giờ, giao diện bàn phím, trong khi cổng 16 bit được sử dụng để truy cập các cổng song song và hệ thống video, ổ đĩa.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phương pháp Vào/Ra riêng biệt (Isolated I/O hoặc IO mapped I/O)

Không gian địa chỉ cổng I/O và không gian bộ nhớ của máy tính PC với vi xử lý Intel 8086/8088 →

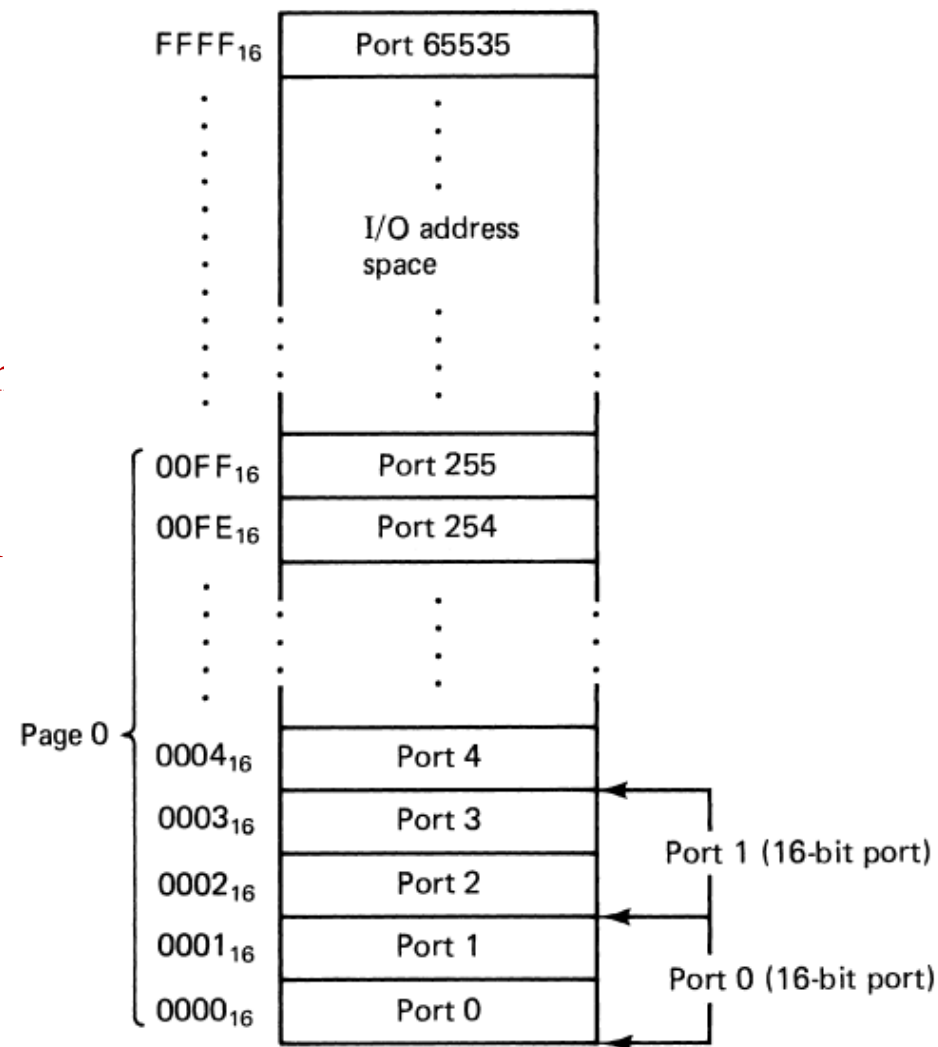




5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phương pháp Vào/Ra riêng biệt (Isolated I/O hoặc mapped I/O)

- Lưu ý rằng phần không gian địa chỉ I/O trong hình vẽ từ địa chỉ 0000h đến 00FFh được gọi là **trang 0 (page 0)**.
- Một số lệnh I/O nhất định (**fixed I/O or direct I/O**) chỉ có thể thực hiện các thao tác với các cổng trong phần này của dải địa chỉ.
- Các lệnh I/O khác (**variable I/O**) có thể nhập hoặc xuất dữ liệu cho các cổng ở bất kỳ đâu trong không gian địa chỉ I/O.





5.2 Phối ghép CPU với thiết bị ngoại vi

➤ **Ưu khuyết điểm của phương pháp Vào/Ra riêng biệt (Isolated I/O hoặc IO mapped I/O)**

Ưu điểm:

- Không gian địa chỉ bộ nhớ (1Mbyte với 8086/8088) hoàn toàn để sử dụng cho bộ nhớ.
- Các lệnh đặc biệt đã được cung cấp trong tập lệnh của vi xử lý để thực hiện các hoạt động đầu vào và đầu ra I/O riêng biệt. Các lệnh này đã được điều chỉnh để tối đa hóa hiệu suất I/O.

Khuyết điểm:

- Tất cả việc truyền dữ liệu đầu vào và đầu ra phải diễn ra giữa thanh ghi AL (các cổng 8 bit) hoặc AX (các cổng 16 bit) và cổng I/O → kém linh hoạt khi lập trình.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ **Bản đồ Vào/Ra của máy tính cá nhân (PC)**

Máy tính cá nhân sử dụng một phần bản đồ I/O cho các chức năng chuyên dụng.

- Không gian I/O giữa các cổng 0000H và 03FFH thường được dành riêng cho hệ thống máy tính và bus ISA.

Các cổng I/O nằm ở 0400H–FFFFH thường có sẵn cho các ứng dụng người dùng, các chức năng của bo mạch chính và bus PCI.

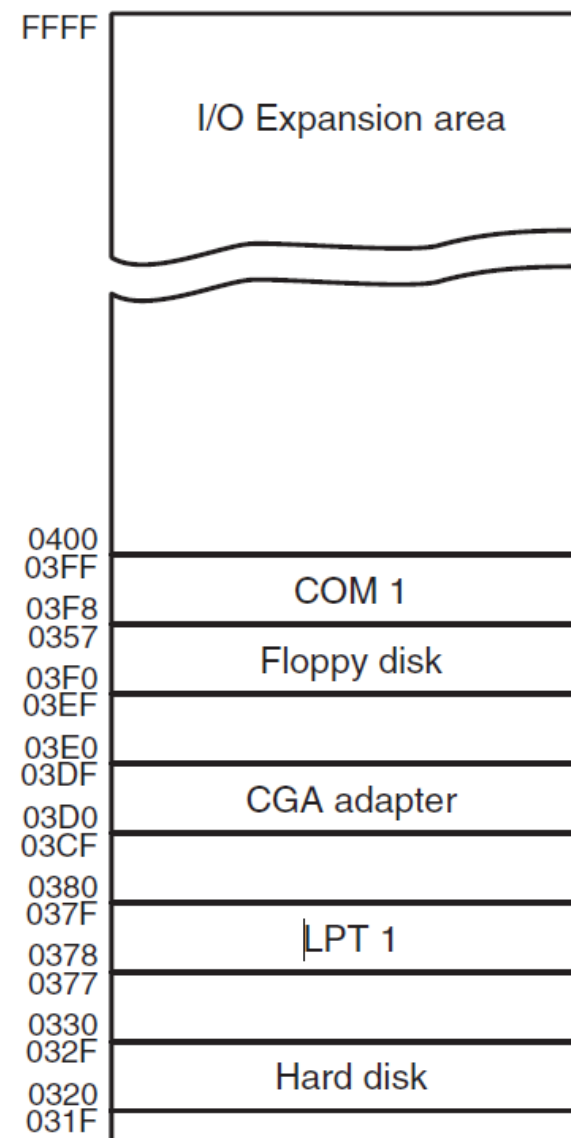
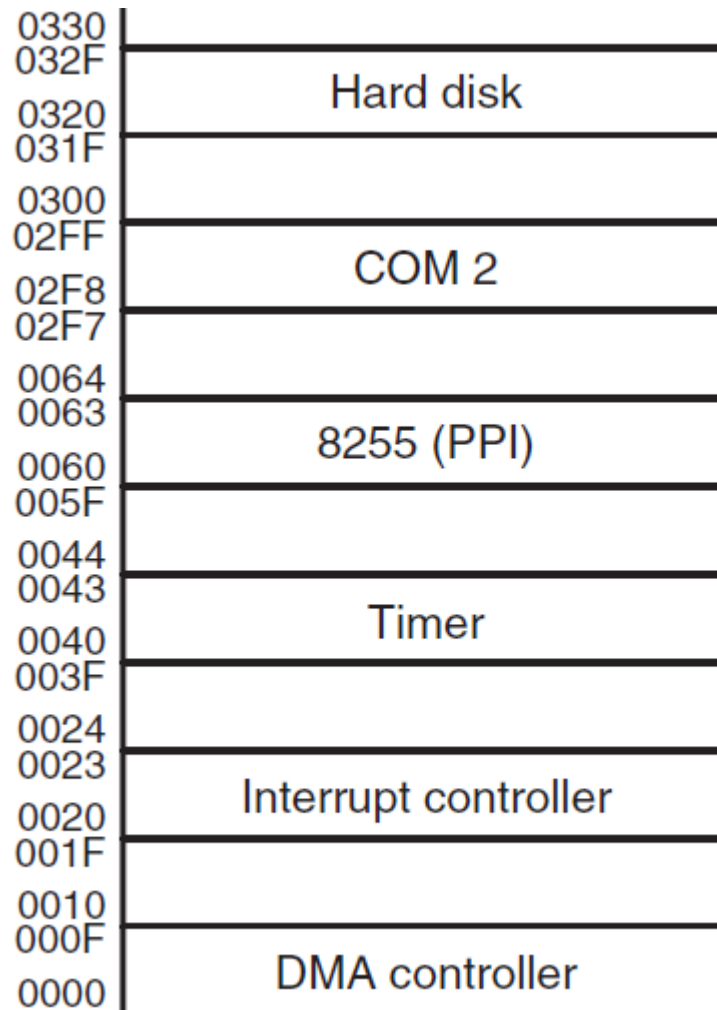
- Bộ xử lý số học 80287 sử dụng địa chỉ I/O 00F8H–00FFH để liên lạc. Vì lý do này, Intel dành riêng các cổng I/O 00F0H–00FFH. 80386–Core2 sử dụng cổng I/O 800000F8–800000FFH để liên lạc với bộ đồng xử lý của chúng.

Các cổng I/O nằm giữa 0000H và 00FFH được truy cập thông qua các lệnh I/O trực tiếp (Directed I/O); các cổng nằm trên 00FFH được truy cập thông qua các lệnh cổng I/O gián tiếp (Variable I/O).



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Bản đồ Vào/Ra của máy tính cá nhân (PC)

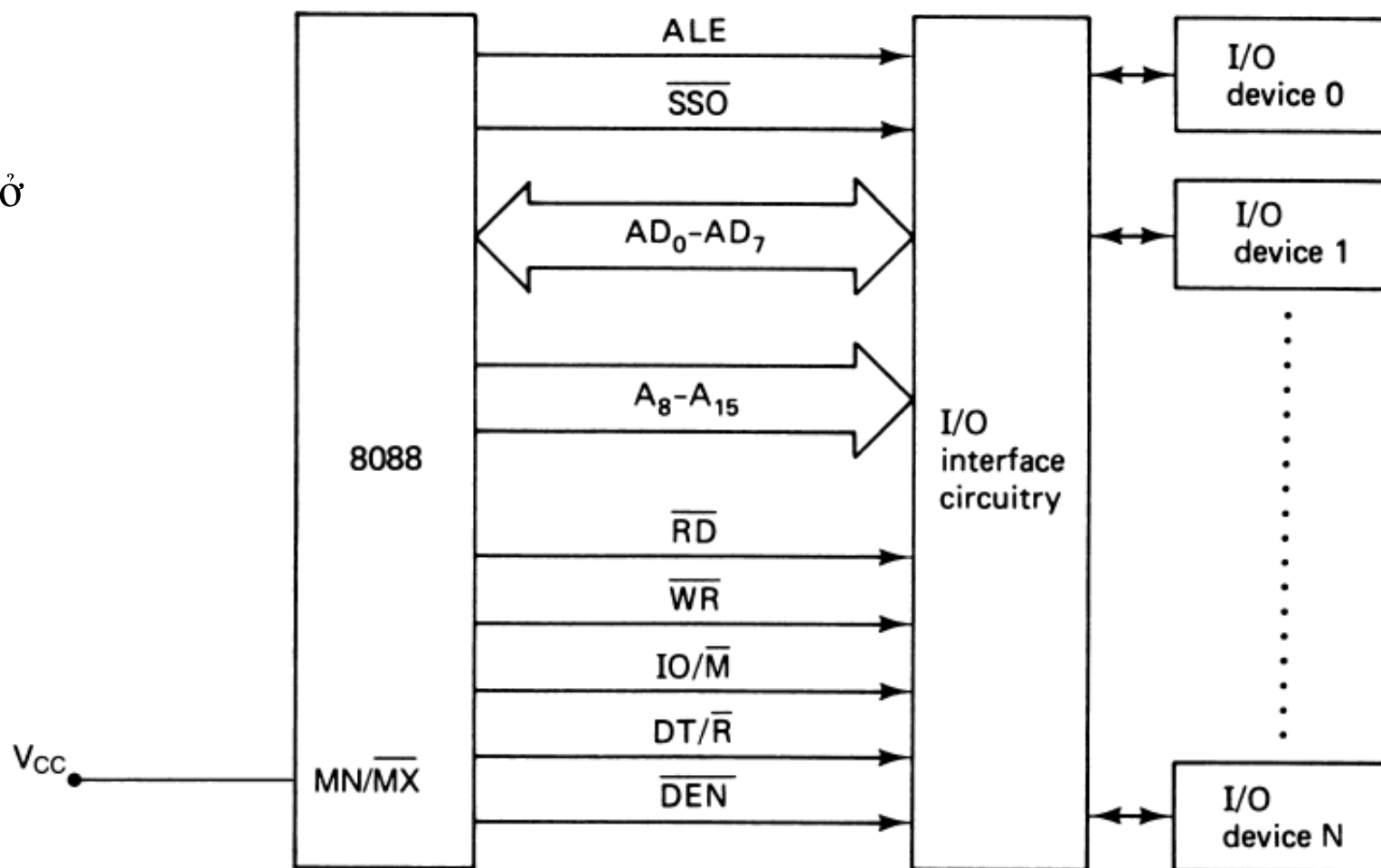




5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phối ghép giữa 8086/8088 và I/O

Phối ghép **8088** và I/O ở chế độ **MIN** mode

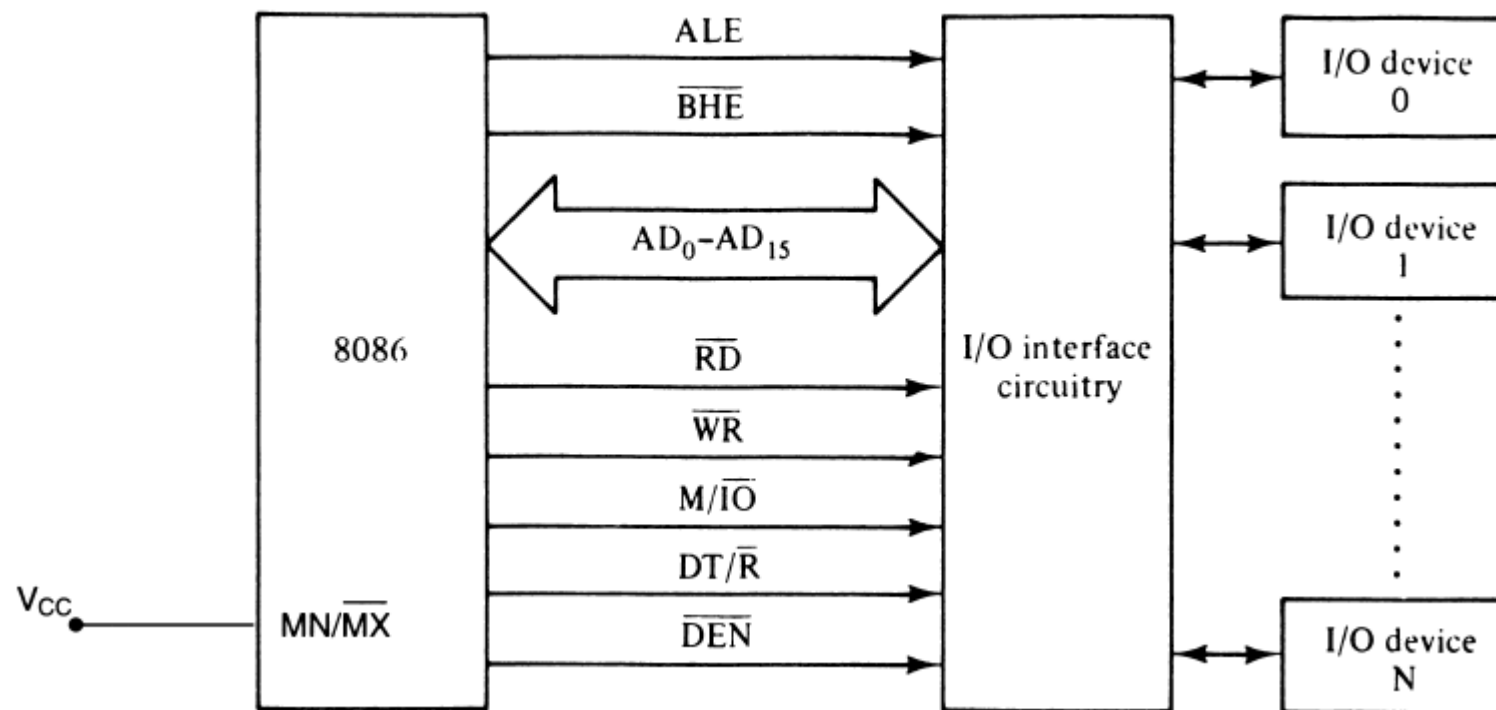




5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phối ghép giữa 8086/8088 và I/O

Phối ghép **8086** và I/O
ở chế độ **MIN** mode

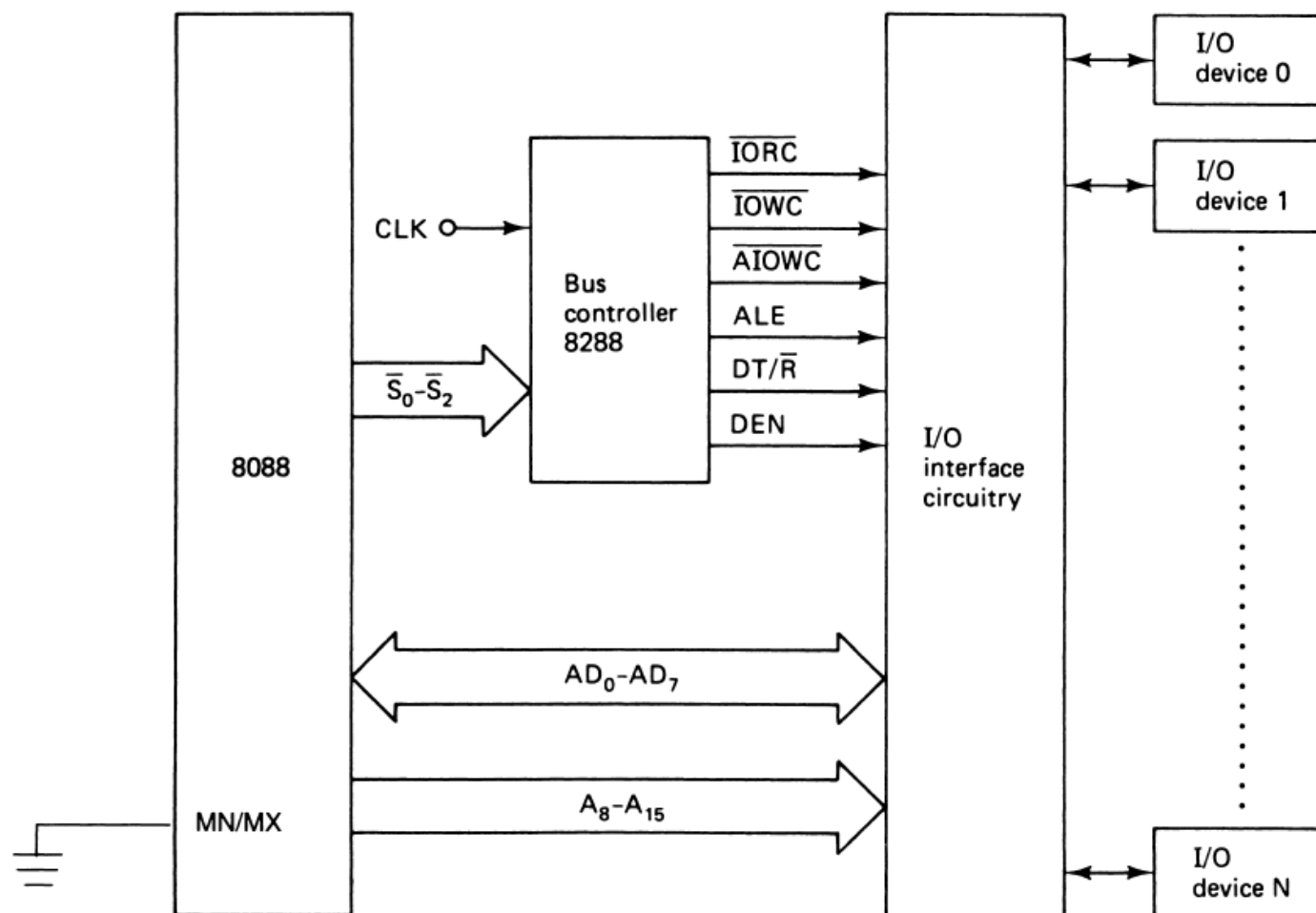




5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phối ghép giữa 8086/8088 và I/O

Phối ghép **8088** và I/O
ở chế độ **MAX** mode

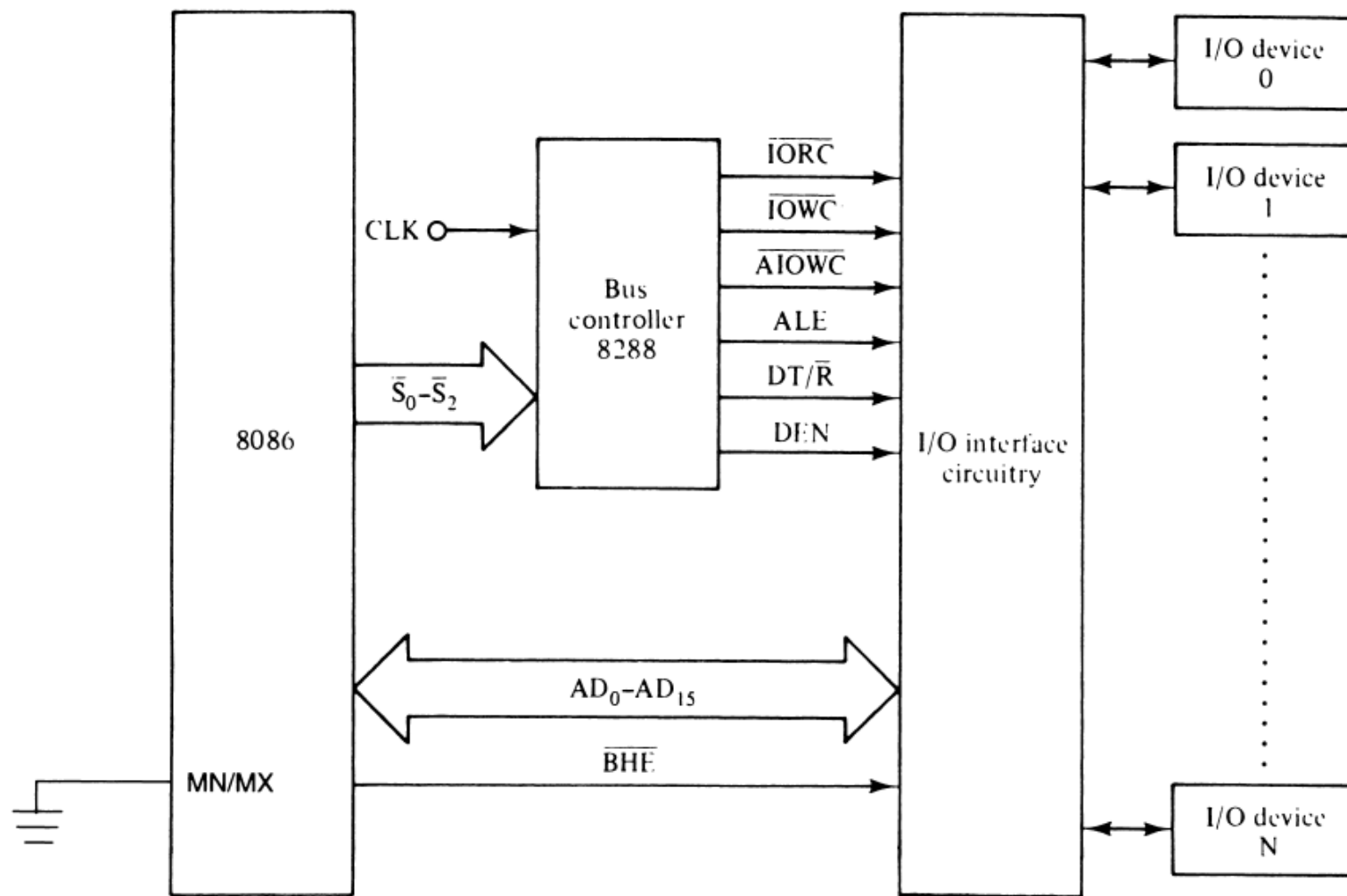




5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Phối ghép giữa 8086/8088 và I/O

Phối ghép **8086** và I/O
ở chế độ **MAX** mode





5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Truyền dữ liệu giữa CPU và I/O

- Việc truyền dữ liệu vào/ra trong máy vi tính 8088 và 8086 có thể theo byte (8 bit) hoặc word (16 bit)
- Cổng được truy cập để nhập hoặc xuất dữ liệu được chọn theo địa chỉ I/O. Địa chỉ này được chỉ ra như một phần của lệnh thực hiện thao tác I/O.

Với Intel 8088

- Địa chỉ I/O có độ dài 16 bit và được 8088 xuất ra giao diện I/O qua các tuyến bus AD_0 đến AD_7 và A_8 đến A_{15} . Các dòng địa chỉ quan trọng nhất, từ A_{16} đến A_{19} , được giữ ở mức logic 0 trong khoảng thời gian tồn tại tín hiệu địa chỉ của tất cả các chu kỳ bus I/O.
 - Do 16 đường địa chỉ được sử dụng để đánh địa chỉ các cổng I/O nên **không gian địa chỉ I/O của 8088 bao gồm các cổng I/O có độ lớn 64K byte.**
 - 8088 báo hiệu cho mạch ngoài rằng địa chỉ trên bus là dành cho cổng I/O thay vì vị trí bộ nhớ bằng cách chuyển đường dây điều khiển IO/\overline{M} sang mức logic 1. Tín hiệu này được giữ ở mức 1 trong suốt chu kỳ bus đầu vào hoặc đầu ra hoàn chỉnh.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Truyền dữ liệu giữa CPU và I/O

Với Intel 8088

- Việc truyền dữ liệu giữa 8088 và các thiết bị I/O được thực hiện qua bus dữ liệu.
- Việc truyền dữ liệu đến các cổng I/O có độ rộng dữ liệu 1 byte luôn yêu cầu một chu kỳ bus.
- Việc truyền dữ liệu byte (8 bit) đến một cổng được thực hiện qua các bus D_0 đến D_7 .
- Việc truyền word (16 bit) cũng diễn ra trên bus dữ liệu, từ D_0 đến D_7 . Tuy nhiên, loại hoạt động này được thực hiện dưới dạng hai lần truyền dữ liệu có độ lớn 1 byte liên tiếp và mất hai chu kỳ bus.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Truyền dữ liệu giữa CPU và I/O

Với Intel 8086

- Địa chỉ I/O được xuất ra trên các đường bus địa chỉ/dữ liệu từ AD_0 đến AD_{15} .
- Các mức logic của tín hiệu A_0 và \overline{BHE} xác định xem dữ liệu là đầu vào/đầu ra cho cổng rộng 1 byte có địa chỉ lẻ, cổng rộng 1 byte có địa chỉ chẵn hay cổng rộng 1 từ (word).

Ví dụ: nếu $A_0, \overline{BHE} = 1, 0$, một cổng I/O rộng 1 byte có địa chỉ lẻ được truy cập.

- Việc truyền dữ liệu byte đến một cổng ở địa chỉ chẵn được thực hiện qua các bus D_0 đến D_7 và các dữ liệu đến cổng có địa chỉ lẻ được thực hiện qua D_8 đến D_{15} .
 - Việc truyền dữ liệu tới các cổng I/O có dung lượng byte luôn diễn ra trong một chu kỳ bus. Việc truyền dữ liệu từ giữa 8086 và các thiết bị I/O được kèm theo tín hiệu $A_0 \overline{BHE} = 00$ và được thực hiện trên bus dữ liệu đầy đủ, D_0 đến D_{15} .
 - Việc truyền 1 word có thể mất một hoặc hai chu kỳ bus. Để đảm bảo rằng chỉ cần một chu kỳ bus để truyền dữ liệu word, các cổng I/O trên toàn word phải được căn chỉnh ở ranh giới địa chỉ chẵn.



5.2 Phối ghép CPU với thiết bị ngoại vi

➤ Các lệnh vào/ra dữ liệu

- Các hoạt động vào/ ra được thực hiện bởi bộ vi xử lý 8088/8086 dùng I/O riêng biệt bằng cách sử dụng các lệnh vào (**IN**) và ra (**OUT**) đặc biệt cùng với các chế độ đánh địa chỉ cổng I/O.

Mnemonic	Meaning	Format	Operation
IN	Input direct	IN Acc,Port	(Acc) ← (Port) Acc = AL or AX
	Input indirect (variable)	IN Acc,DX	(Acc) ← ((DX))
OUT	Output direct	OUT Port,Acc	(Port) ← (Acc)
	Output indirect (variable)	OUT DX,Acc	((DX)) ← (Acc)

Có hai dạng lệnh IN và OUT khác nhau:

- Lệnh I/O trực tiếp (I/O direct)
- Lệnh I/O gián tiếp (I/O indirect or variable)

Việc truyền dữ liệu diễn ra giữa các thiết bị I/O và thanh ghi tích lũy AL, hoặc AX của vi xử lý 8086/8088.



5.2 Phối ghép CPU với thiết bị ngoại vi

Lệnh I/O trực tiếp (I/O direct):

- Địa chỉ cổng được chỉ ra như một phần của lệnh.
- Địa chỉ cổng có dải từ 0h đến FFh (256 cổng), tương ứng với page 0 trong không gian địa chỉ I/O

Ví dụ: Lệnh **IN AL, 0FEh** → Đưa 1 byte từ cổng có địa chỉ 0FEh vào thanh ghi AL.
Thực hiện lệnh này mất 1 chu kỳ bus.

Ví dụ: *Viết lệnh xuất dữ liệu FFh ra cổng rộng 1 byte có địa chỉ ABh trong không gian địa chỉ I/O ?*

MOV AL, FFh ; nạp cho AL giá trị FFh

OUT 0ABh, AL ; xuất dữ liệu ra cổng có địa chỉ 0ABh



5.2 Phối ghép CPU với thiết bị ngoại vi

Lệnh I/O gián tiếp (Indirect I/O or Variable I/O)

- Địa chỉ có độ dài 16 bit của các cổng được đặt vào thanh ghi DX chứ không được là 1 giá trị tức thì (1 số). Các lệnh I/O gián tiếp có thể truy xuất đến bất kỳ cổng nào trong không gian địa chỉ I/O có độ lớn 64 Kbyte.
- Giá trị của thanh ghi DX không phải là địa chỉ offset mà nó thực sự là địa chỉ được đặt lên bus địa chỉ từ AD_0 tới AD_{15} trong chu kỳ bus I/O.

Ví dụ: *Viết lệnh đọc dữ liệu từ 2 cổng rộng 1 word có địa chỉ AAh và A9h sau đó xuất hai byte này ra cổng có độ rộng 16 bit tại địa chỉ B000h ?*

IN AL, 0AAh	; đọc cổng 1 byte có địa chỉ 0AAh
MOV AH, AL	; cất dữ liệu đã đọc được vào AH
IN AL, 0A9h	; đọc cổng tiếp theo
MOV DX, 0B00h	; nạp địa chỉ cổng 16 bit vào DX
OUT DX, AX	; xuất dữ liệu ra cổng 16 bit



5.2 Phối ghép CPU với thiết bị ngoại vi

5.2.4 Các phương pháp vào/ra dữ liệu

❖ Vai trò của vào ra dữ liệu:

- ☐ Là phương tiện giúp CPU giao tiếp với thế giới bên ngoài
- ☐ Cung cấp dữ liệu đầu vào cho CPU xử lý
- ☐ Cung cấp phương tiện để CPU kết xuất dữ liệu đầu ra

❖ Các phương pháp vào ra chính:

- ☐ Vào/ra theo chương trình (**Programmed I/O**)
- ☐ Vào/ra bằng Ngắt (**Interrupt**)
- ☐ Truy nhập trực tiếp bộ nhớ (**DMA-Direct Memory Access**)



5.2 Phối ghép CPU với thiết bị ngoại vi

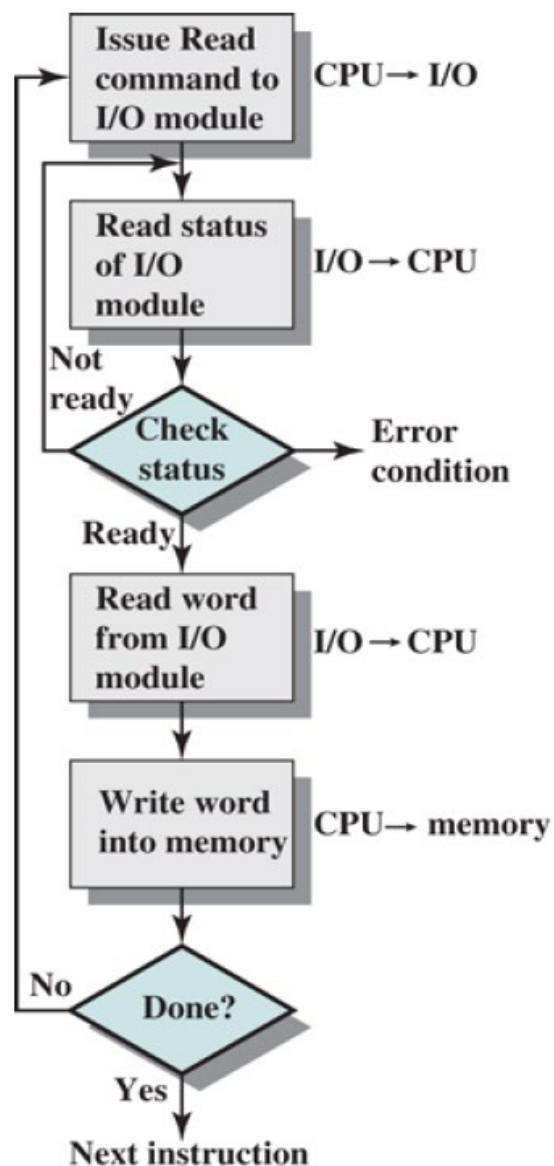
❖ Các cổng vào ra của máy tính

- ☐ PS/2: cổng ghép nối với bàn phím và chuột
- ☐ COM: các cổng ghép nối nối tiếp
- ☐ LPT: các cổng ghép nối song song
- ☐ IDE, SATA, SCSI: các cổng ghép nối ổ đĩa
- ☐ LAN: cổng ghép nối mạng cục bộ
- ☐ Audio: cổng ghép nối âm thanh (speaker, mic và line-in)
- ☐ Video: Cổng ghép nối với màn hình (tương tự)
 - ☐ DVI : Cổng ghép nối với màn hình (số)
 - ☐ USB: Cổng ghép nối theo chuẩn USB
 - USB 1.0: 12Mb/s
 - USB 2.0: 480Mb/s
 - USB 3.0: 1.5Gb/s





5.2 Phối hợp CPU với thiết bị ngoại vi



❖ Vào ra theo chương trình (Programmed I/O)

- 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 phát hiện trạng thái thiết bị ngoại vi, 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.



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Vào ra theo chương trình (Programmed 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 ngoại vi 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.
- ✓ Kiểm tra: Được sử dụng để kiểm tra điều kiện trạng thái của module I/O và thiết bị ngoại vi
- ✓ Đọc: Yêu cầu module I/O lấy dữ liệu từ thiết bị ngoại vi và đặt nó vào bộ đệm
- ✓ Ghi: Yêu cầu module I/O chuyển dữ liệu (1 byte hoặc 1 từ) từ bus dữ liệu đến thiết bị ngoại vi.



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Vào ra theo chương trình (Programmed I/O)

Khi bộ xử lý, bộ nhớ và module I/O chia sẻ một bus chung, hai chế độ định địa chỉ cho các thiết bị ngoại vi 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).

Ví dụ:

Ánh xạ bộ nhớ

(memory mapped I/O) →

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

I/O riêng biệt

(Isolated I/O) →

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 tb sẵn sàng
	In	5	Đọc dữ liệu



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Vào ra theo chương trình (Programmed I/O)

Vào ra theo chương trình có thể được chia thành 2 loại sau:

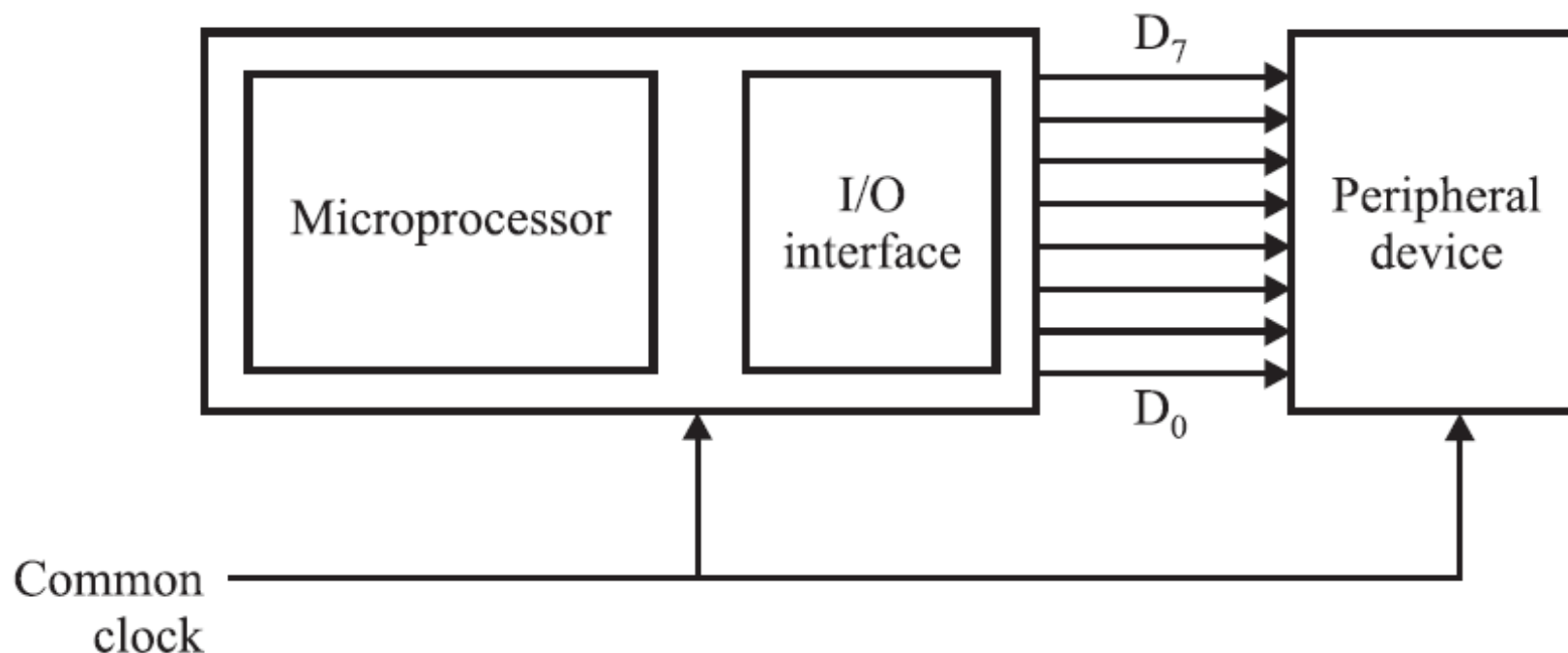
- Truyền dữ liệu đồng bộ (Synchronous data transfer)
- Chế độ đồng bộ có trễ hoặc truyền dữ liệu với tín hiệu **READY**
- Truyền dữ liệu không đồng bộ (Asynchronous mode) hoặc truyền dữ liệu với tín hiệu **HANDSHAKE**



5.2 Phối ghép CPU với thiết bị ngoại vi

Truyền dữ liệu đồng bộ (Synchronous data transfer)

- Truyền dữ liệu song song, đồng bộ được sử dụng khi bộ vi xử lý và thiết bị ngoại vi có cùng tốc độ hoạt động và do vậy sử dụng chung tín hiệu đồng hồ (Clock).
- Các thiết bị ngoại vi làm việc theo phương pháp này thường là các thiết bị điện tử chứ không phải là các thiết bị cơ khí. Ví dụ như máy in, các máy điều khiển quá trình..

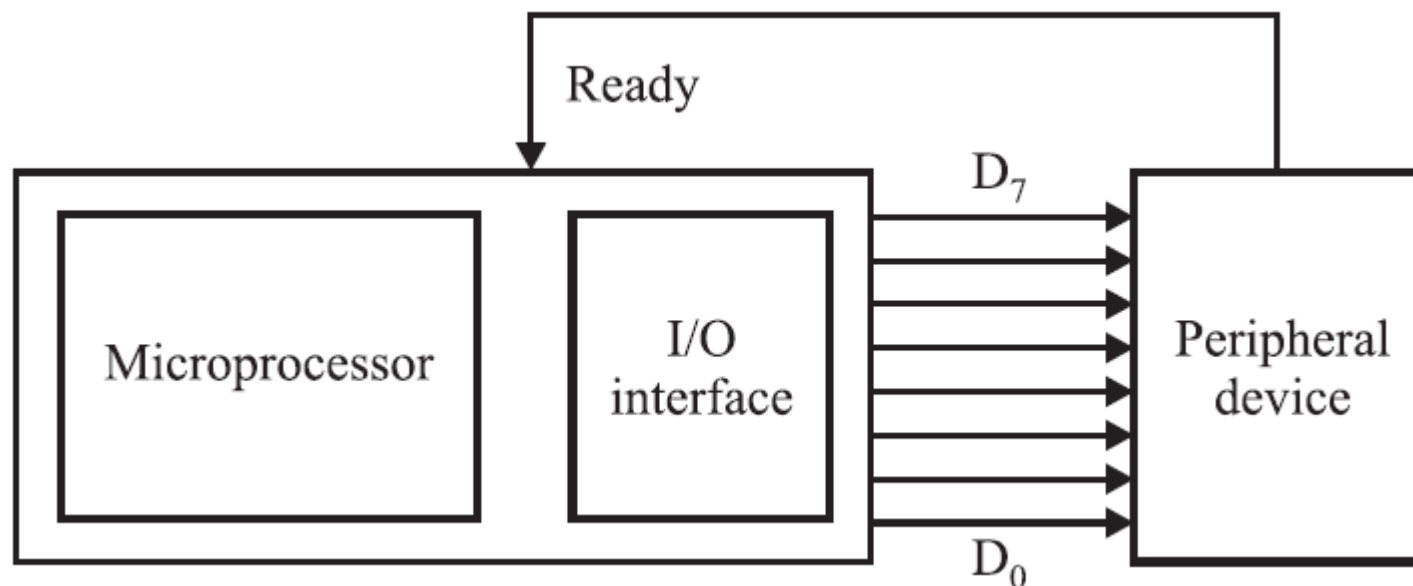




5.2 Phối ghép CPU với thiết bị ngoại vi

Chế độ đồng bộ có trễ hoặc truyền dữ liệu với tín hiệu **READY**

- Chế độ này được sử dụng khi thiết bị ngoại vi có tốc độ hoạt động chậm hơn vi xử lý
- Trong chế độ hoạt động này bộ vi xử lý dùng chân READY để đồng bộ với tốc độ hoạt động của ngoại vi.
- Khi ngoại vi sẵn sàng trao đổi dữ liệu với vi xử lý, nó sẽ gửi tín hiệu tới chân READY của VXL.
- Khi VXL nhận được tín hiệu READY sẽ gửi dữ liệu và chuyển vào trạng thái chờ cho tới lần kế tiếp nhận được tín hiệu READY lần nữa.

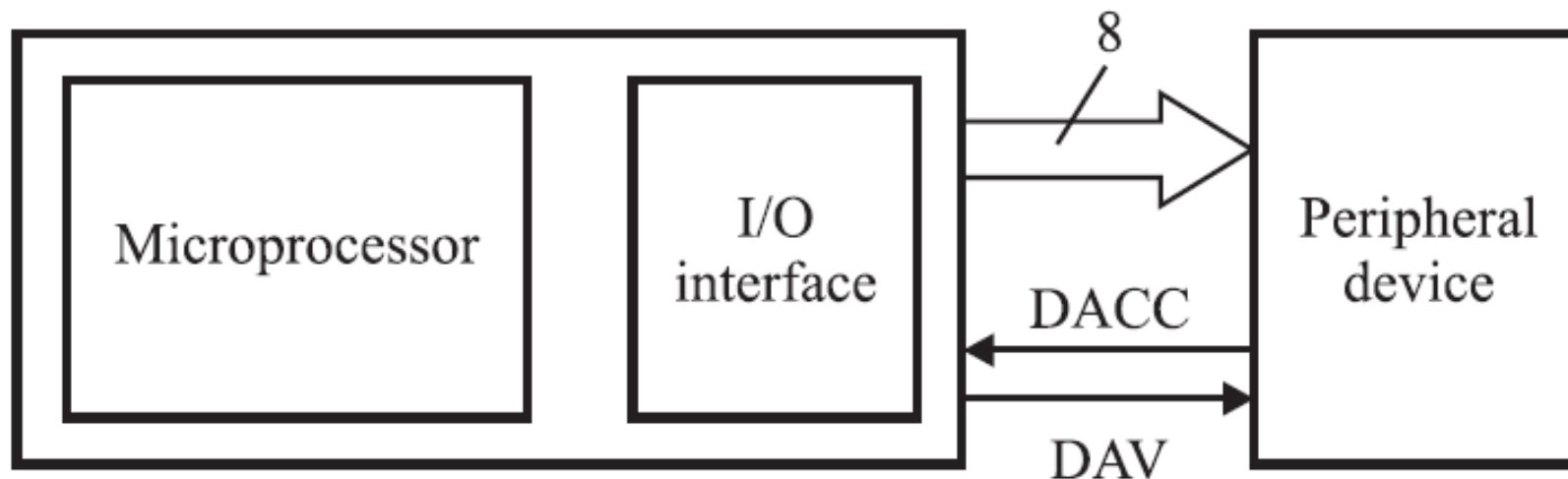




5.2 Phối ghép CPU với thiết bị ngoại vi

Truyền dữ liệu không đồng bộ (Asynchronous mode) hoặc truyền dữ liệu với tín hiệu **HANDSHAKE**

- Nhìn chung các thiết bị ngoại vi thường có tốc độ chậm hơn VXL, để có thể trao đổi dữ liệu thành công với ngoại vi thì chế độ truyền không đồng bộ thường được sử dụng.
- Chế độ truyền không đồng bộ sử dụng các tín hiệu điều khiển **Handshake** (Data available - DAV và Data Accepted - DACC).





Truyền dữ liệu không đồng bộ (Asynchronous mode) hoặc truyền dữ liệu với tín hiệu Handshake

Quá trình truyền dữ liệu diễn ra như sau:

- VXL đặt 1 byte dữ liệu lên cổng vào/ra
- Sau đó kích hoạt tín hiệu DAV=1 để thông báo với ngoại vi rằng dữ liệu đã sẵn sàng.
- Bộ vi xử lý sau đó sẽ giám sát tín hiệu DACC để xác định dữ liệu đã được đọc bởi thiết bị ngoại vi hay chưa ?
- Thiết bị ngoại vi giám sát đường tín hiệu DAV. Nếu thấy $DAV = 1$ nó sẽ đọc dữ liệu.
 - Sau đó tín hiệu DACC được kích hoạt để thông báo với VXL rằng ngoại vi đã sẵn sàng nhận dữ liệu mới.
 - Quá trình này được lặp lại cho tới khi truyền hết dữ liệu.



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Vào ra điều khiển bằng ngắt (Interrupted I/O)

Phương pháp vào/ra theo chương trình (Programmed I/O) có nhược điểm:

- Tốc độ chậm
- Không tận dụng hết năng lực xử lý của VXL
- Không dùng được trong các ứng dụng thời gian thực
- Không thể điều khiển đồng bộ các sự kiện xảy ra bất ngờ một cách hiệu quả

→ Phương pháp vào/ra điều khiển bằng Ngắt sẽ khắc phục những nhược điểm này



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Vào ra điều khiển bằng ngắt (Interrupted I/O)

Bộ VXL gửi lệnh I/O đến module I/O và sau đó thực hiện các hoạt động khác của riêng nó. 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 VXL thực hiện việc truyền dữ liệu rồi quay trở lại công việc riêng của nó.

Việc đọc dữ liệu (nhìn từ phía Module I/O):

- Module I/O nhận lệnh READ từ VXL. 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 VXL qua một đường dây điều khiển.
 - Module chờ đợi cho đến khi VXL 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.



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Vào ra điều khiển bằng ngắt (Interrupted I/O)

Việc đọc dữ liệu (nhìn từ phía bộ VXL):

- Bộ VXL 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 xem có yêu cầu ngắt nào xuất hiện hay không ?
- Khi có ngắt gửi từ module I/O, bộ VXL 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.



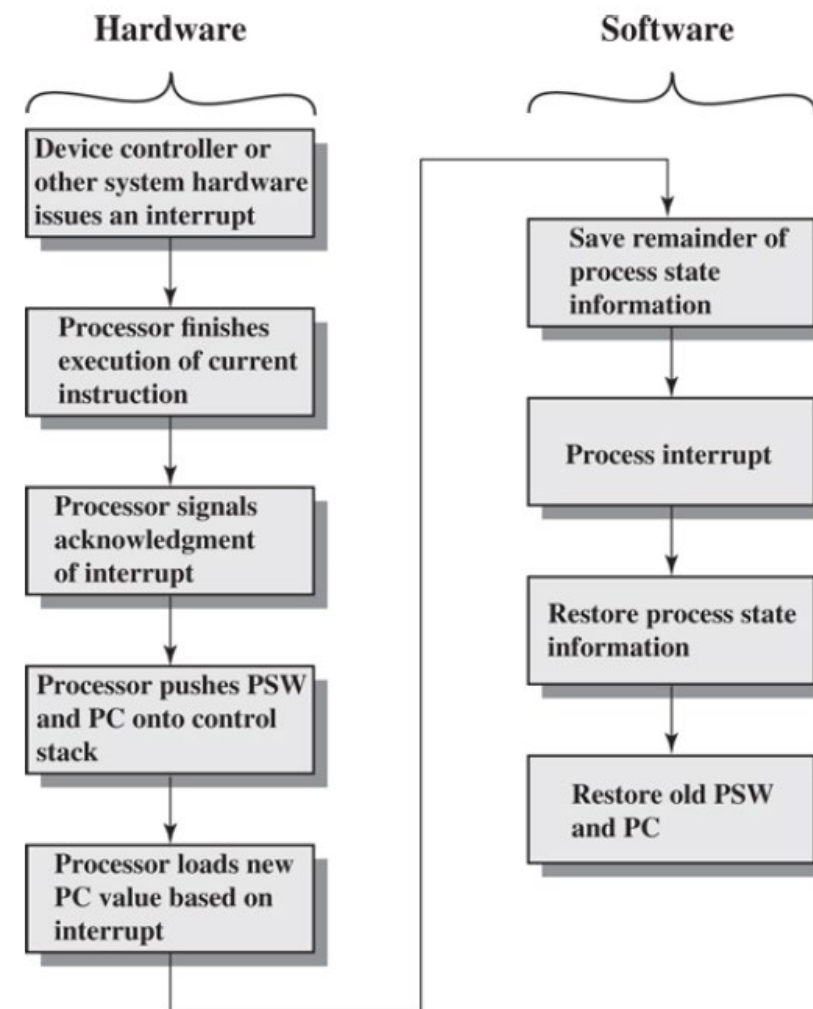
5.2 Phối hợp CPU với thiết bị ngoại vi

❖ Vào ra điều khiển bằng ngắt (Interrupted I/O)

Xử lý ngắt khi thực hiện vào/ra bằng ngắt

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:

- 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 yêu cầu ngắt. Khi nhận được ACK, thiết bị loại bỏ tín hiệu ngắt.



Quá trình xử lý ngắt đơn giản



5.2 Phối ghép CPU với thiết bị ngoại vi

Xử lý ngắt khi thực hiện vào/ra bằng ngắt

- 4 - Bộ VXL chuyển điều khiển sang chế độ ngắt. VXL 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.
- 5- Bộ VXL sẽ tải vị trí đầu tiên của chương trình xử lý ngắt vào thanh ghi PC. Sau khi thanh ghi PC đã được nạp, bộ VXL sẽ thực thi các lệnh trong chương trình xử lý ngắt.
- 6- 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.



5.2 Phối ghép CPU với thiết bị ngoại vi

Một số vấn đề nảy sinh khi thiết kế hệ thống vào/ra bằng ngắt

- Làm sao để xác định được module vào/ra nào phát ra tín hiệu ngắt ?
- CPU xử lý ra sao khi có nhiều ngắt cùng xảy ra ?

Các phương pháp ghép nối ngắt

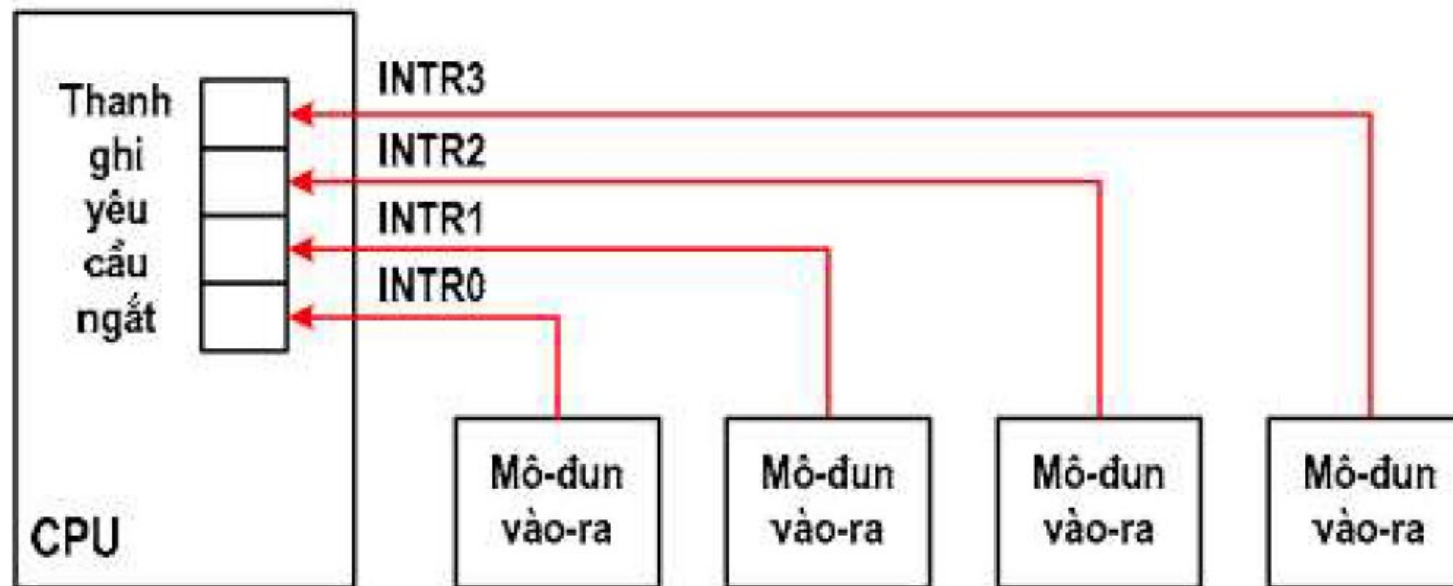
- Sử dụng nhiều đường dây yêu cầu ngắt
- Sử dụng phương pháp hỏi vòng Polling bằng phần mềm (Software Polling)
- Sử dụng phương pháp hỏi vòng bằng phần cứng (Daisy Chain)
- Sử dụng bộ điều khiển ngắt (PIC)



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Sử dụng nhiều đường yêu cầu ngắt

- ✓ Mỗi một module vào/ra được nối với một đường yêu cầu ngắt
- ✓ CPU phải có nhiều đường tín hiệu yêu cầu ngắt
- ✓ Hạn chế số lượng module vào/ra
- ✓ Các đường ngắt được quy định mức ưu tiên

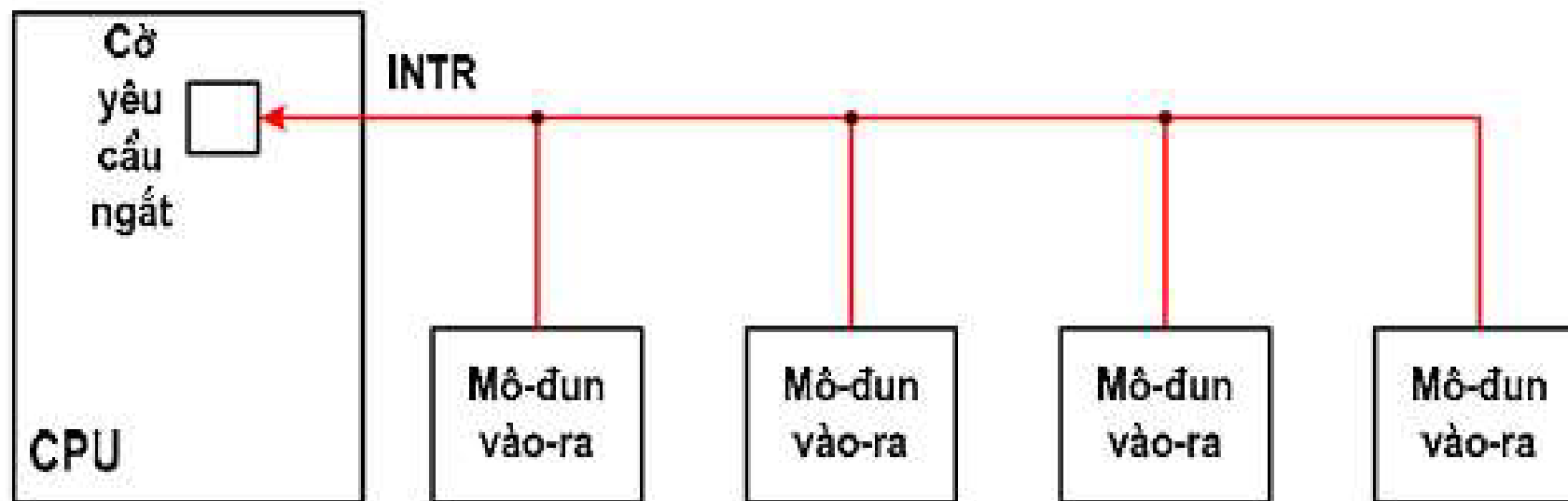




5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Hỏi vòng Polling bằng phần mềm

- ✓ CPU thực hiện phần mềm hỏi lần lượt từng mô-đun vào-ra
- ✓ Chậm
- ✓ Thứ tự các mô-đun được hỏi vòng chính là thứ tự ưu tiên

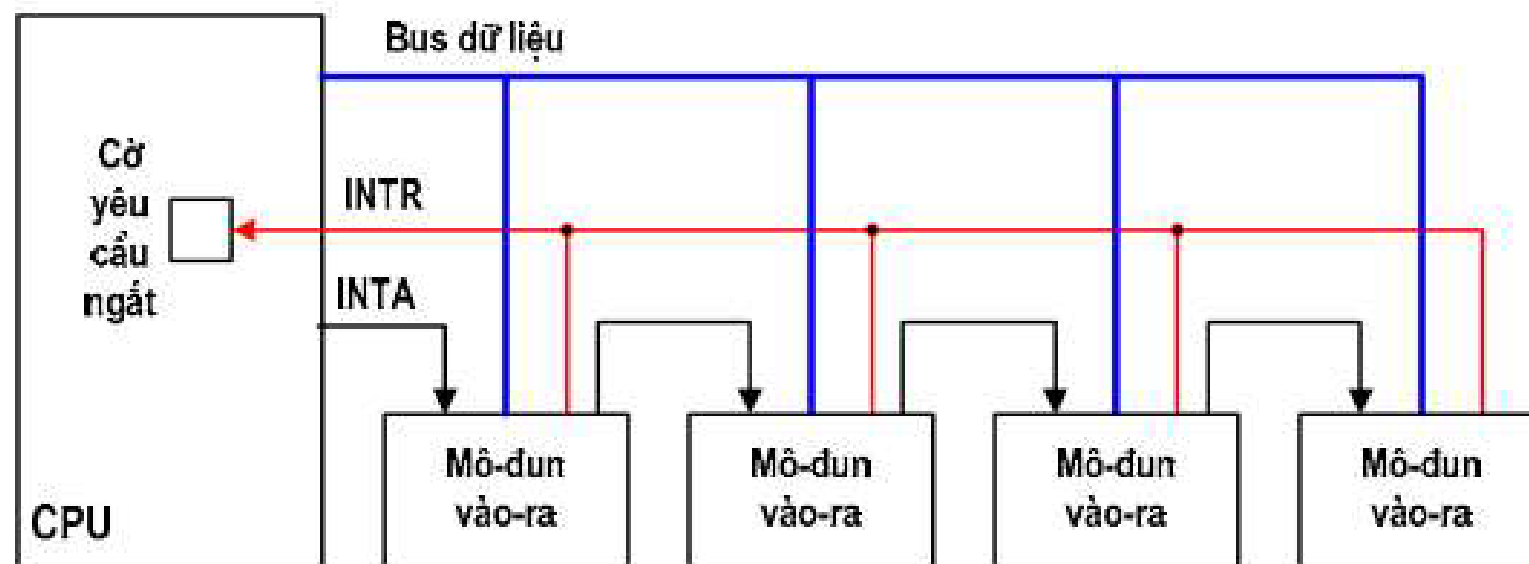




5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Hở vòng bằng phần cứng (Daisy chain)

- CPU phát tín hiệu chấp nhận ngắt (INTA) đến module vào/ra đầu tiên
- Nếu module vào/ra đó không gây ra ngắt thì nó gửi tín hiệu đến module kế tiếp cho đến khi xác định được module gây ngắt
- Thứ tự các module vào/ra kết nối trong chuỗi xác định thứ tự ưu tiên

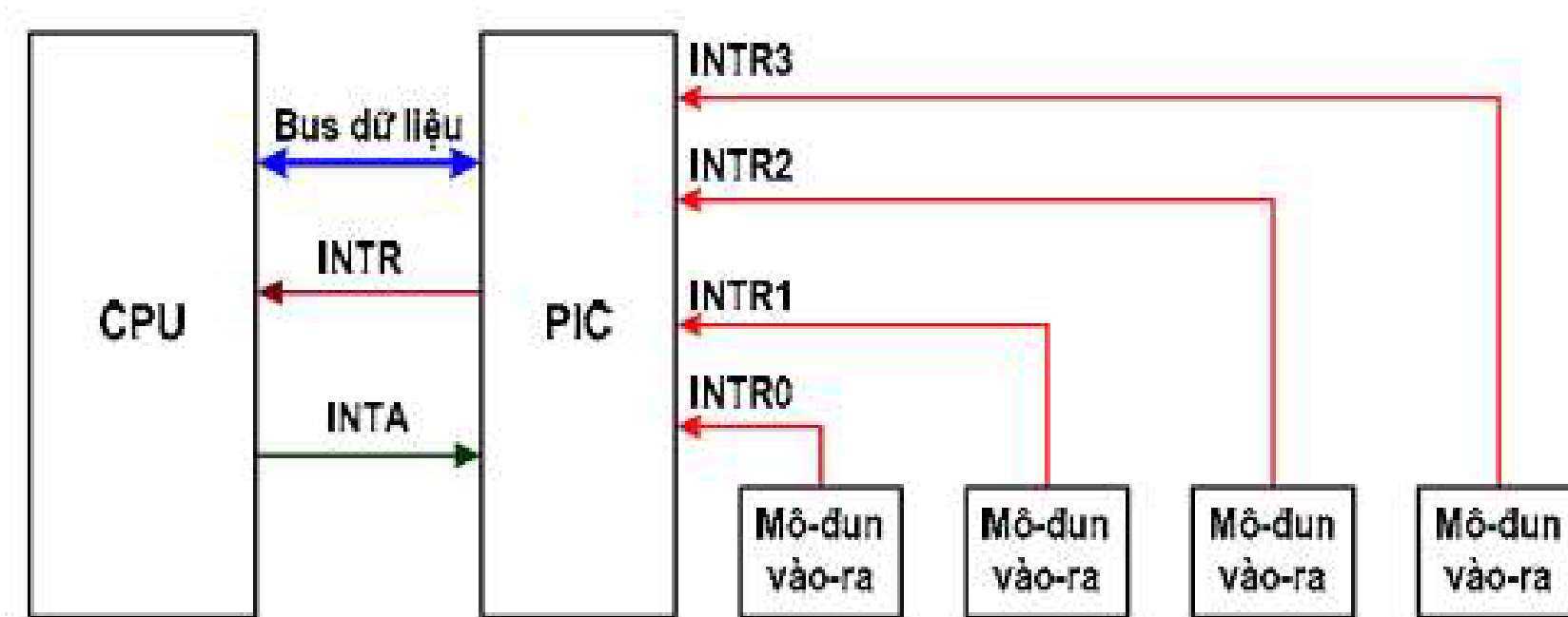




5.2 Phối hợp CPU với thiết bị ngoại vi

❖ Sử dụng bộ điều khiển ngắt lập trình được PIC

- PIC – Programmable Interrupt Controller
- PIC có nhiều đường vào yêu cầu ngắt có qui định mức ưu tiên
- PIC chọn một yêu cầu ngắt không bị cấm có mức ưu tiên cao nhất gửi tới CPU





5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Vào ra điều khiển bằng DMA (Direct Memory Accesss)

Các phương pháp vào/ra bằng chương trình hoặc bằng ngắt CPU luôn tham gia vào quá trình trao đổi dữ liệu. Dữ liệu đi từ ngoại vi qua CPU tới bộ nhớ và ngược lại. Điều này làm tiêu tốn thời gian và năng lực của CPU dẫn đến giảm hiệu năng hệ thống đặc biệt khi trao đổi từng khối dữ liệu lớn !

Kỹ thuật DMA sẽ khắc phục những khuyết điểm này !

Đặc điểm của DMA:

- Sử dụng module điều khiển vào/ra chuyên dụng DMAC (DMA Controller)
- CPU không tham gia trong quá trình trao đổi dữ liệu
- DMAC điều khiển trao đổi dữ liệu giữa bộ nhớ chính với module vào/ra (hoàn toàn bằng phần cứng) nên tốc độ nhanh.
- Phù hợp với các yêu cầu trao đổi mảng dữ liệu có kích thước lớn



5.2 Phối hợp CPU với thiết bị ngoại vi

❖ Vào ra điều khiển bằng DMA (Direct Memory Accesss)

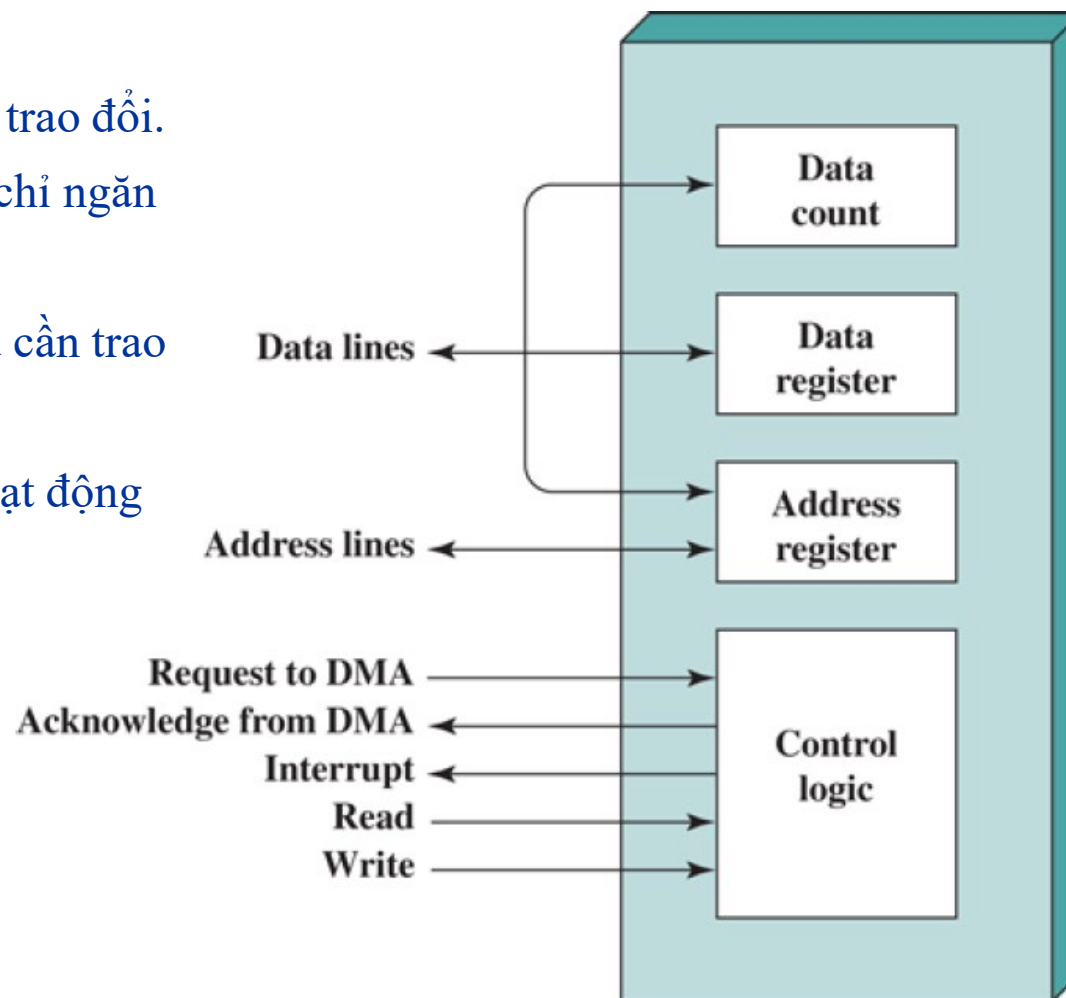
- Sơ đồ khối cấu trúc DMA

Thanh ghi dữ liệu (Data Register): chứa dữ liệu trao đổi.

Thanh ghi địa chỉ (Address Register): chứa địa chỉ ngăn nhớ dữ liệu.

Bộ đếm dữ liệu (Data Count): chứa số từ dữ liệu cần trao đổi.

Logic điều khiển (Control Logic): điều khiển hoạt động của DMAC





5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Cơ chế hoạt động

➤ CPU thông báo cho DMA

- Vào hay Ra dữ liệu ?
- Địa chỉ thiết bị vào/ra (cổng vào/ra tương ứng)
- Địa chỉ đầu vào của mảng nhớ chứa dữ liệu và nạp vào thanh ghi địa chỉ
- Số từ cần chuyển và nạp vào bộ đếm dữ liệu

➤ CPU thực hiện công việc khác

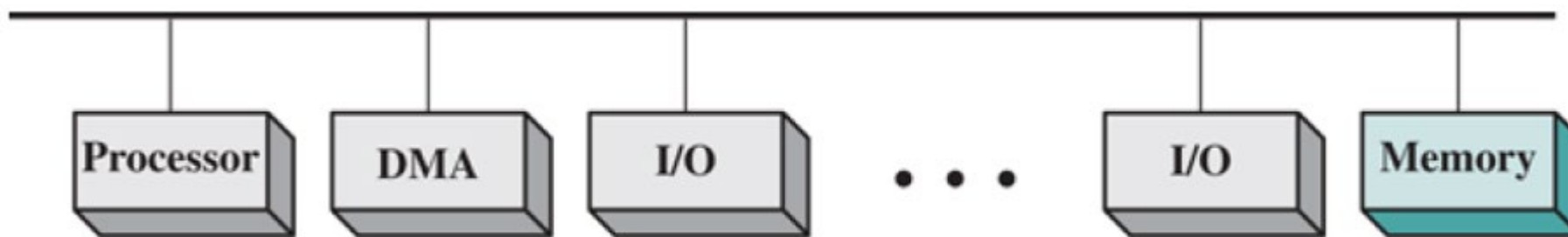
- DMA điều khiển trao đổi dữ liệu
- Sau khi truyền được 1 từ dữ liệu thì: tăng nội dung thanh ghi địa chỉ, nội dung bộ đếm dữ liệu giảm.
 - Khi bộ đếm dữ liệu bằng 0, DMAC gửi tín hiệu ngắt CPU để báo kết thúc trao đổi.



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Các cấu hình DMA

- Single bus (Detached DMA) → DMA tách biệt



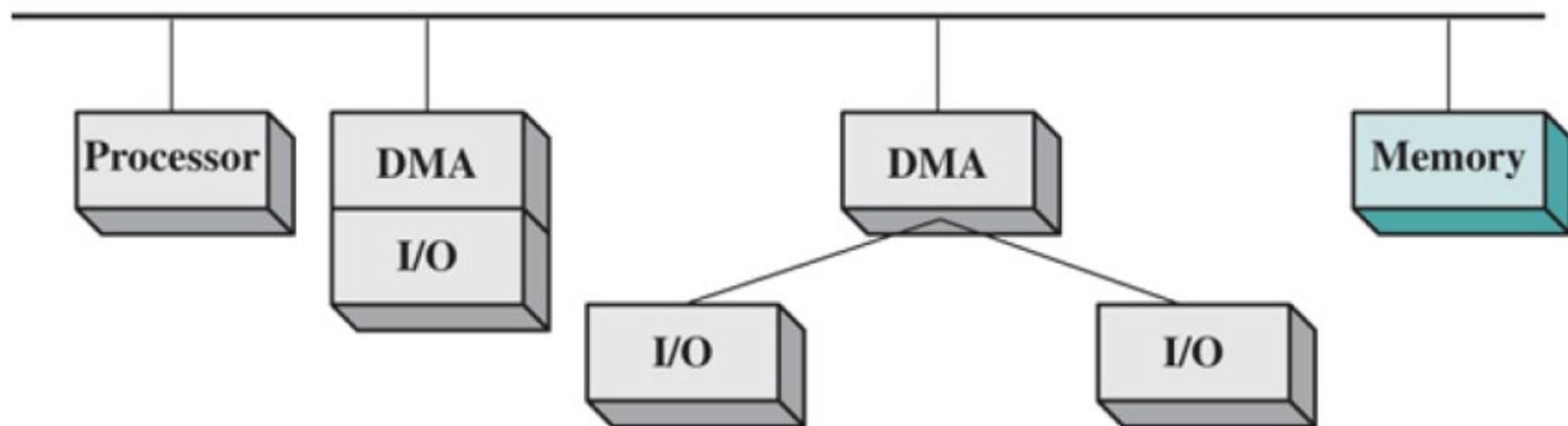
- Tất cả các module chia sẻ cùng một bus hệ thống.
- Module DMA, hoạt động như một bộ xử lý thay thế, sử dụng I/O được lập trình để trao đổi dữ liệu giữa bộ nhớ và module I/O thông qua module DMA.
- Cấu hình này, mặc dù có thể không tốn kém, nhưng rõ ràng là không hiệu quả.
- Giống như I/O được lập trình do bộ xử lý điều khiển, mỗi lần truyền một từ sẽ tiêu tốn hai chu kỳ bus



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Các cấu hình DMA

- Single bus (Integrated DMA-I/O) → DMA tích hợp I/O



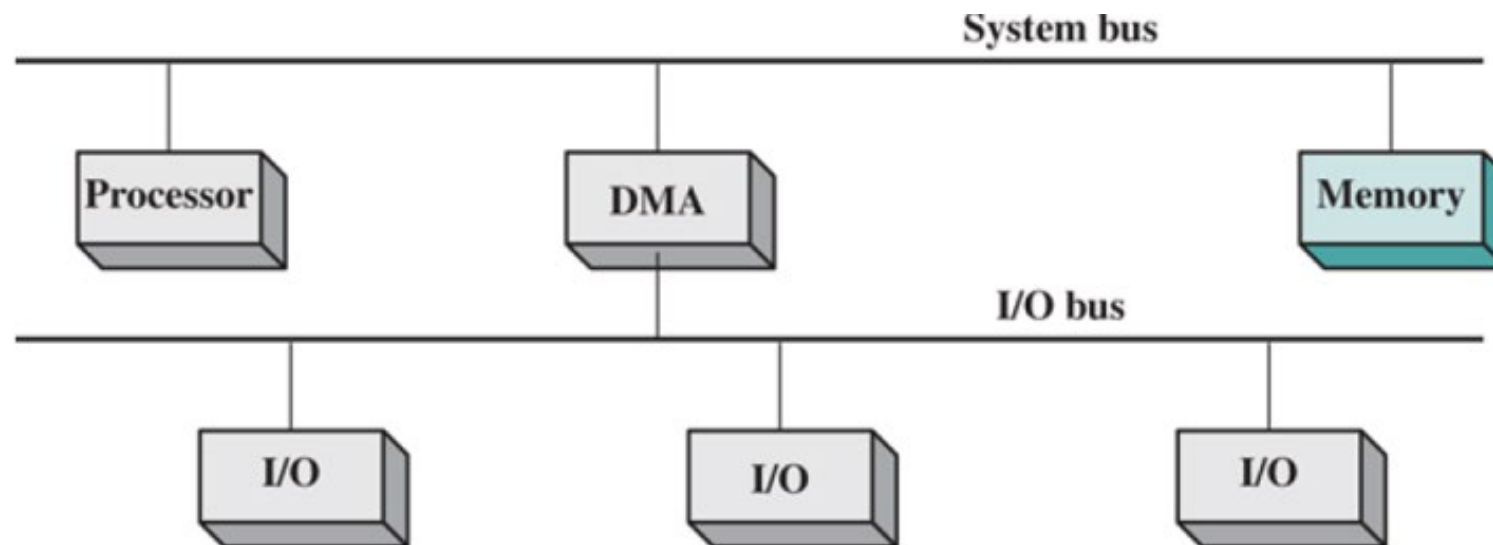
- Tất cả các module chia sẻ cùng một bus hệ thống.
- Có một đường dẫn giữa module DMA và một hoặc nhiều module I/O không bao gồm bus hệ thống.
- Logic DMA thực sự có thể là một phần của mô-đun I/O hoặc có thể là một mô-đun riêng biệt điều khiển một hoặc nhiều mô-đun I/O.



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Các cấu hình DMA

- I/O bus → DMA sử dụng bus I/O



- Tất cả các module I/O được quản lý bởi một DMA thông qua bus I/O
- Việc trao đổi dữ liệu giữa các module DMA và I/O diễn ra ngoài bus hệ thống.
- Cấu hình này làm giảm giao diện I/O trong module DMA và dễ mở rộng



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Các phương pháp trao đổi dữ liệu dùng DMA

- Trao đổi cả mảng dữ liệu (**Block-Transfer DMA**)

CPU tự treo và trao quyền điều khiển cho DMAC cho tới khi truyền xong khối dữ liệu.

- Lấy lén chu kỳ bus (**Cycle Stealing DMA**)

Thỉnh thoảng CPU lại bị treo để DMAC thực hiện trao đổi 1 byte dữ liệu (tốc độ CPU giảm đi chứ không bị dừng lại)

- Tận dụng thời gian CPU không dùng bus để trao đổi dữ liệu (**Transparent DMA**)

Phải có các logic phụ để phát hiện các chu kỳ bus rỗi dùng cho trao đổi DMA

DMAC và CPU luân phiên sử dụng bus

Không ảnh hưởng gì tới hoạt động của CPU



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Bộ điều khiển DMA Intel 8237

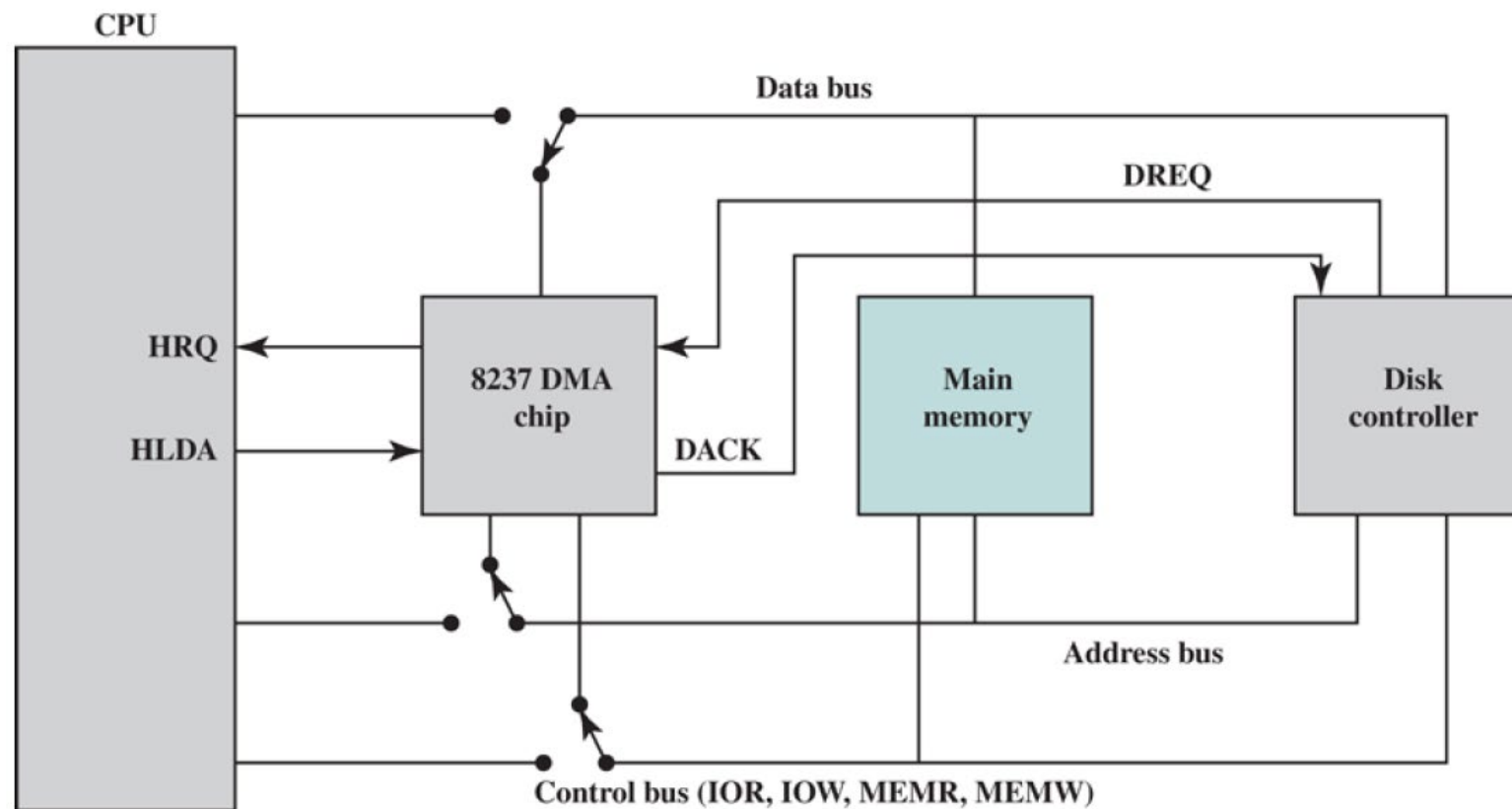
- Bộ điều khiển DMA Intel 8237 được dùng để giao tiếp giữa các bộ vi xử lý 80x86 và bộ nhớ DRAM nhằm cung cấp khả năng trao đổi theo phương pháp DMA.
- Khi module DMA cần sử dụng bus hệ thống (bus địa chỉ, dữ liệu, điều khiển) nó sẽ gửi một tín hiệu HOLD tới bộ vi xử lý.
- Bộ vi xử lý sẽ trả lời bằng tín hiệu HLDA (hold acknowledge) để báo hiệu rằng module DMA có thể sử dụng bus.



5.2 Phối ghép CPU với thiết bị ngoại vi

❖ Bộ điều khiển DMA Intel 8237

- Vị trí của DMAC Intel 8237 trong hệ thống →



DACK = DMA acknowledge
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request



5.3 Lập trình giao tiếp bàn phím

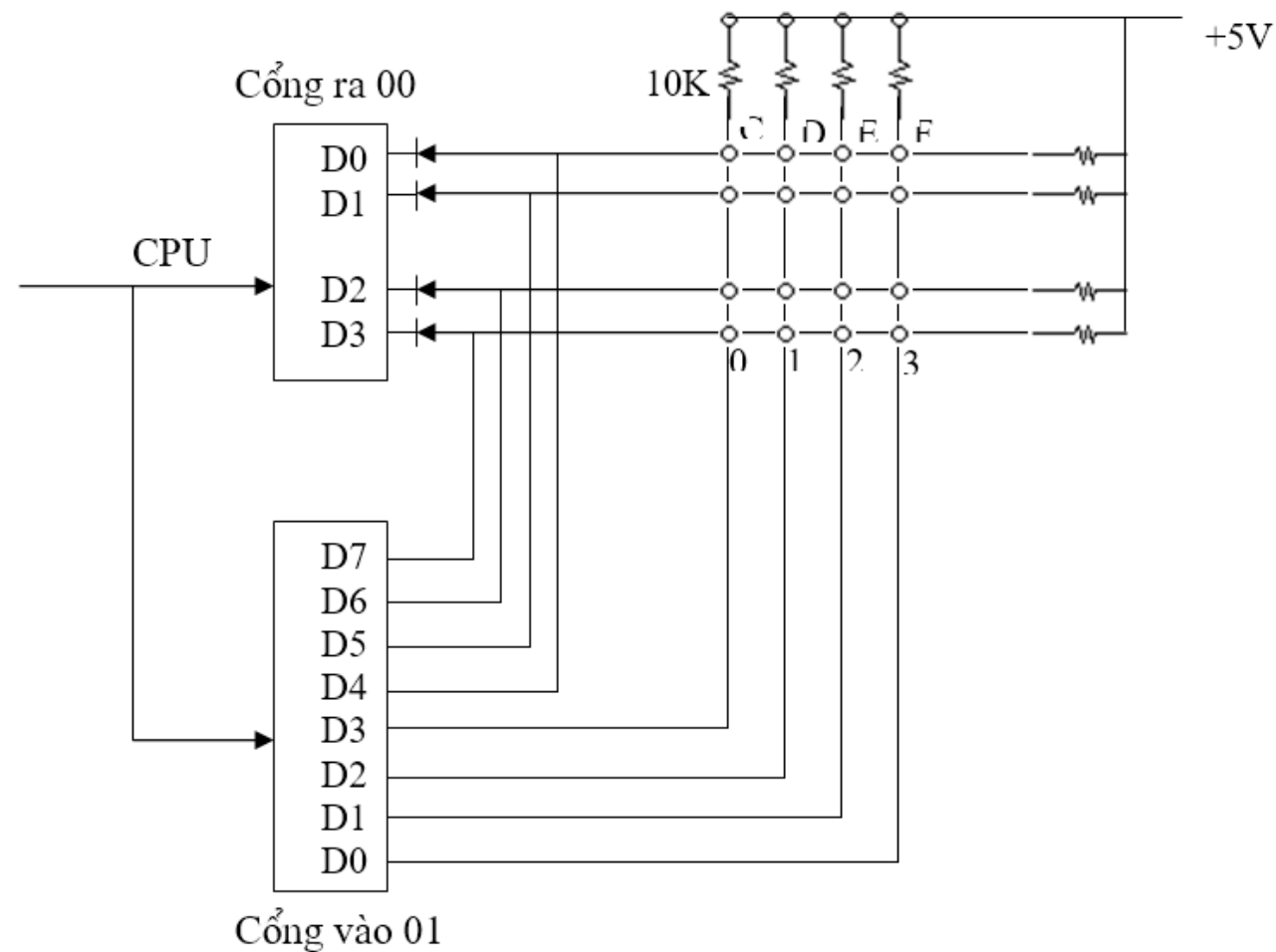
- Có nhiều kiểu bàn phím, phím nhấn cơ khí, phím nhấn kiểu điện dung (chạm)
- Bàn phím thường được tổ chức theo kiểu ma trận
- Có thể gắn trực tiếp vào các cổng của VXL hoặc qua các vi mạch chuyên dụng

Chúng ta xét trường hợp đơn giản: bàn phím được ghép với các cổng của VXL và bàn phím có cấu tạo ma trận với các hàng và các cột, sử dụng phím nhấn cơ khí → CPU trực tiếp quản lý bàn phím

- Giả thiết bàn phím 4x4 (16 phím) được đánh số từ 0-9, từ A tới F.
- Để xác nhận trạng thái của phím được ấn thường phải làm 3 việc sau:
 - ✓ Phát hiện có phím ấn
 - ✓ Chờ công tắc ổn định (chống rung)
 - ✓ Tìm phím được ấn và tạo mã cho nó



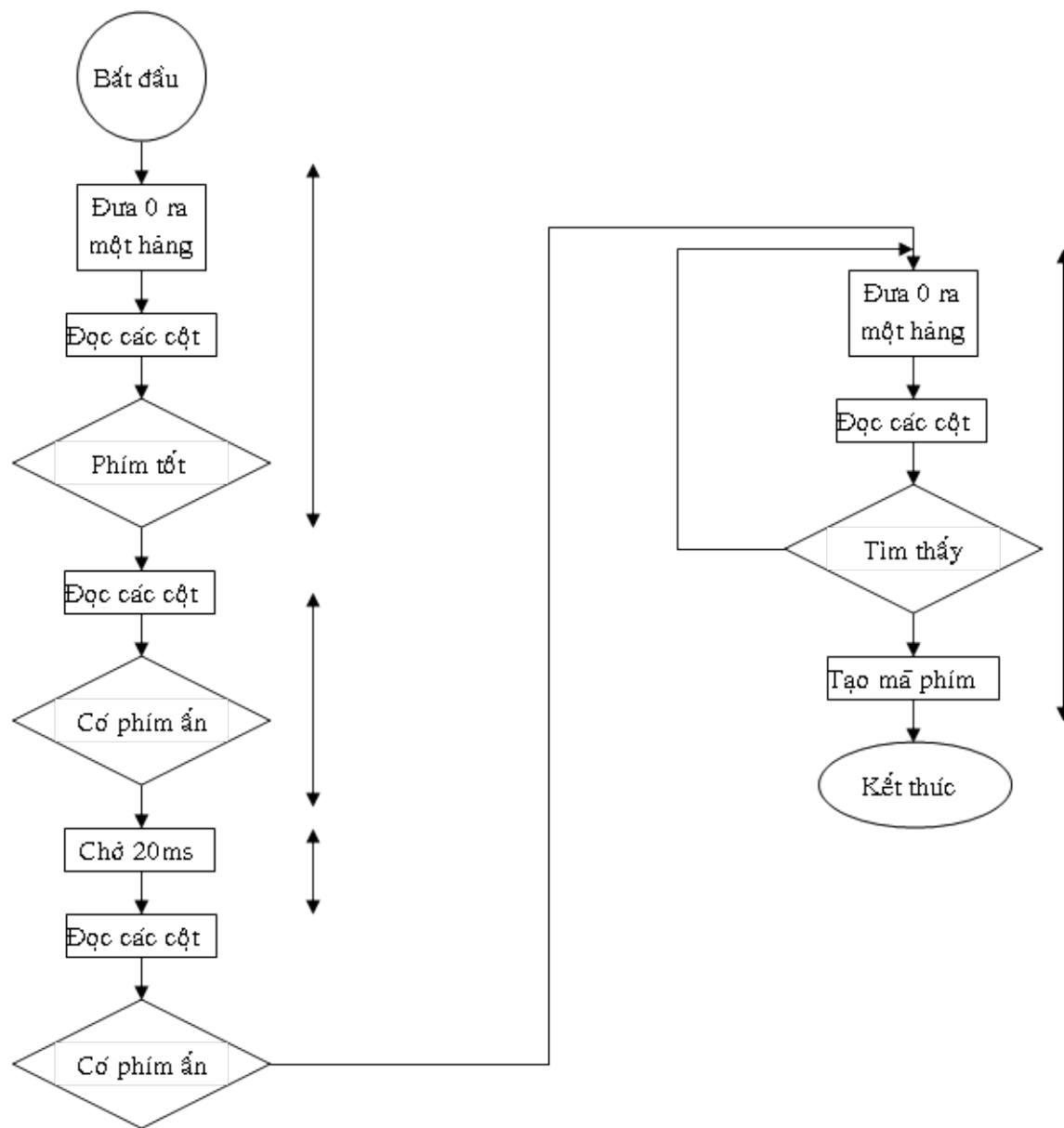
5.3 Lập trình giao tiếp bàn phím





5.3 Lập trình giao tiếp bàn phím

Lưu đồ thuật toán giao tiếp bàn phím →





5.3 Lập trình điều khiển hệ thống đèn LED

Bài tập 1.

Cho mạch điều khiển 8 đèn LED (tương ứng với các bit D_0 đến D_7) được nối với hệ vi xử lý 8086 tại cổng có địa chỉ 2C0h. Biết rằng đèn được bật sáng nếu bit điều khiển tương ứng nhận giá trị 0. Ngược lại khi bit điều khiển bằng 1 thì đèn sẽ tắt. Vẽ lưu đồ thuật toán tạo các hiệu ứng sau:

- Tất cả các đèn đều cùng bật, tắt liên tục với khoảng trễ giữa hai lần bật là 250 chu kỳ lệnh NOP
- Tạo hiệu ứng một đèn sáng chạy từng bước từ trái sang phải, bắt đầu từ đèn D_5 , mỗi bước dịch chuyển tương ứng với 1 đèn, khoảng thời gian nghỉ của một bước dịch chuyển là 550 chu kỳ lệnh NOP.



5.3 Lập trình điều khiển nhiệt độ

Bài tập 2: Viết chương trình điều khiển ổn định nhiệt độ phòng làm việc, yêu cầu nhiệt độ ổn định trong khoảng từ 22°C tới 25°C.

Nếu nhiệt độ cao quá 25°C thì bật quạt để làm mát.

Nếu nhiệt độ thấp dưới 22°C thì bật đèn sưởi để tăng nhiệt độ.

Cổng 8 bit gắn cảm biến nhiệt có địa chỉ **3FDh**.

Cổng 8 bit gắn đèn có địa chỉ **3FEh**

Cổng 8 bit gắn quạt có địa chỉ **3FFh**

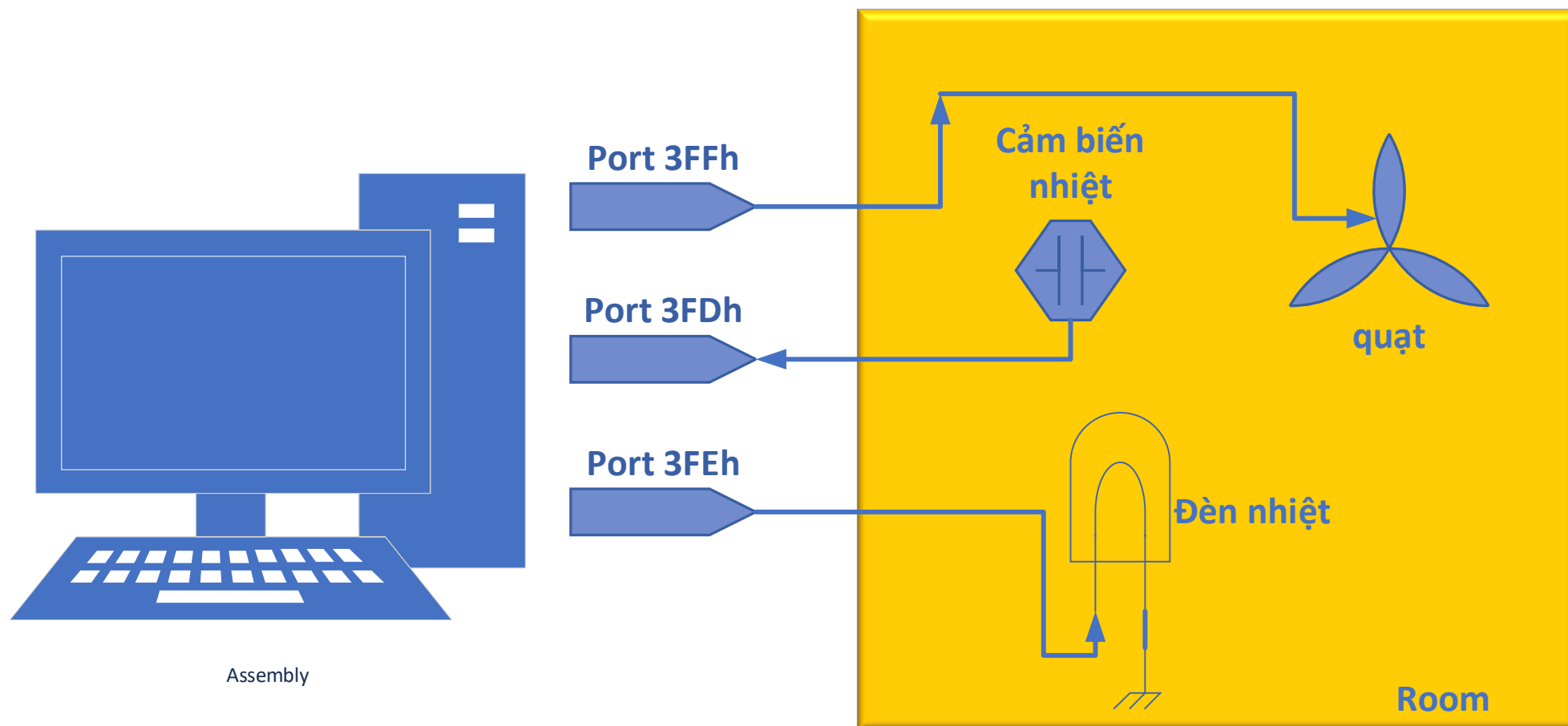
Giả sử đèn nhiệt hoặc quạt mát được bật khi cổng có các bit D_0 tới D_7 toàn bằng 1, tắt khi toàn bằng 0.

Mỗi lần nhiệt độ được kiểm tra cách nhau khoảng thời gian 1 phút.

Giả sử 1 phút = 600 lệnh NOP.



5.3 Lập trình điều khiển nhiệt độ





5.3 Lập trình điều khiển hệ Robot đơn giản và một số ví dụ

- Tìm hiểu về các ví dụ trong EMU80806 (bài điều khiển robot)
- Các ví dụ lập trình hợp ngữ trên Internet