HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kiến Trúc Máy Tính

CHƯƠNG 5 – PHỐI GHÉP VỚI BỘ NHỚ VÀ THIẾT BỊ VÀO RA

Giảng viên: TS. Nguyễn Văn Thuỷ

Điện thoại/E-mail: thuynv@ptit.edu.vn

Bộ môn: Khoa học máy tính - Khoa CNTT1

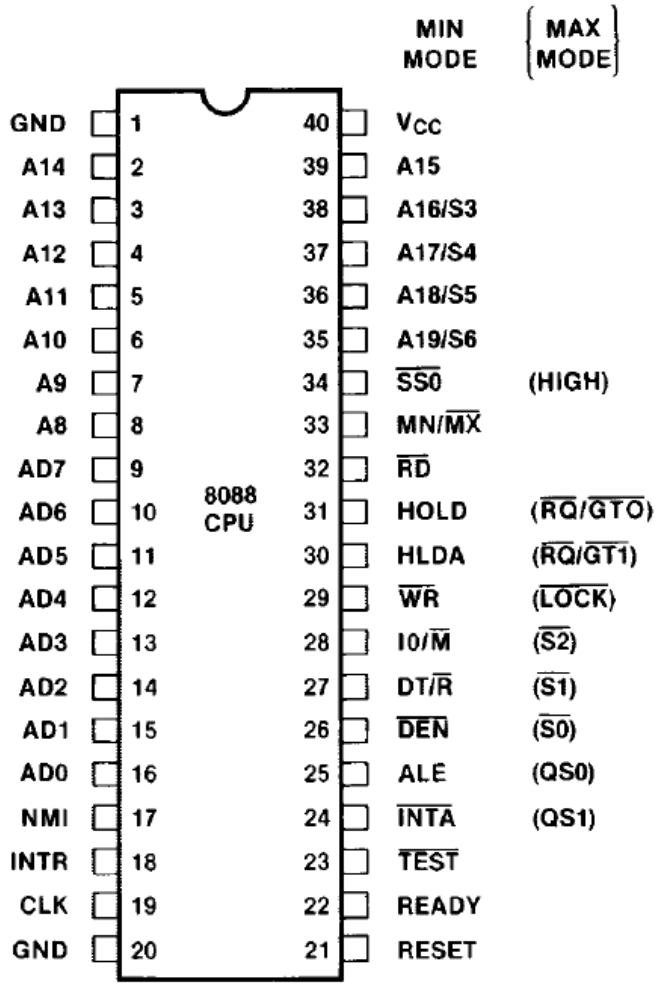
Học kỳ/Năm biên soạn: Học kỳ 2 năm học 2021-2022

NỘI DUNG

1. Giới thiệu về CPU 8088
2. Phối ghép CPU với bộ nhớ
3. Phối ghép CPU với thiết bị ngoại vi
4. Bài tập lớn (ví dụ của emu 8086)
    1. Lập trình điều khiển bàn phím
    2. Lập trình điều khiển hệ thống đèn led
    3. Lập trình điều khiển nhiệt độ
    4. Lập trình điều khiển đèn giao thông
    5. Lập trình điều khiển robot đơn giản
    6. Các phương pháp vào ra dữ liệu

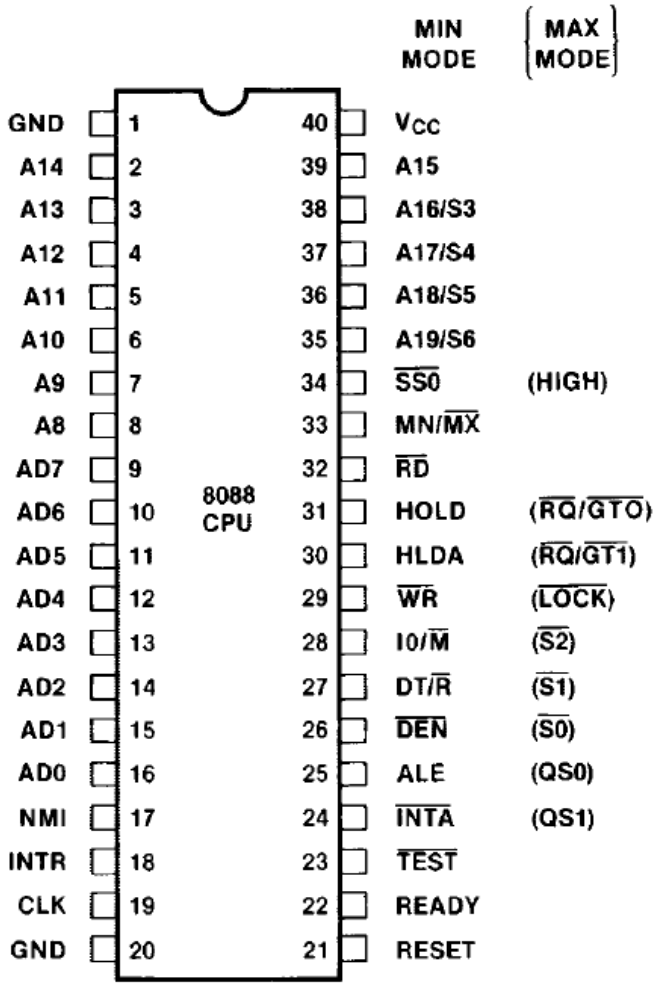
5.1. Các tín hiệu của 8088

- VXL 8088 có có 40 chân tín hiệu, gồm các nhóm:
  - Nhóm tín hiệu địa chỉ:
    - AD_0-AD_7: 8 chân dồn kênh cho phần thấp bus A và bus D ;
    - A_8-A_{15}: 8 chân tín hiệu phân cao bus A
    - A_{16}/S_3-A_{19}/S_6: 4 chân dồn kênh cho phần cao bus A và bus C;
  - Nhóm tín hiệu dữ liệu
    - AD_0-AD_7: 8 chân dồn 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. Các tín hiệu của 8088

- Nhóm tín hiệu điều khiển hệ thống:
  - IO/$\overline{M}$: tín hiệu CPU chọn làm việc với thiết bị vào ra hay bộ nhớ. IO/$\overline{M}$ =1 → CPU chọn làm việc với thiết bị vào ra; IO/$\overline{M}$ =0 → CPU chọn làm việc với bộ nhớ. Địa chỉ tương ứng của bộ phận được lựa chọn xuất hiện trên bus địa chỉ.
  - DT/$\overline{R}$: Tín hiệu xác định chiều vận chuyển dữ liệu trên bus dữ liệu. DT/$\overline{R}$=1 → dữ liệu đi ra từ CPU; DT/$\overline{R}$=0 → dữ liệu đi đến CPU.
  - $\overline{\text{RD}}$: Xung cho phép đọc (đảo). Khi $\overline{RD}$ = 0 bus dữ liệu sẵn sàng nhận dữ liệu từ bộ nhớ hoặc thiết bị ngoại vi.
  - $\overline{\text{WR}}$: Tín hiệu cho phép ghi. Khi $\overline{\text{WR}}$ = 0, dữ liệu đã ổn định trên bus dữ liệu và được ghi vào bộ nhớ hoặc thiết bị vào ra khi $\overline{\text{WR}}$ = 1.
  - $\overline{\text{DEN}}$: Tín hiệu báo cho mạch ngoài biết dữ liệu đã ổn định trên bus dữ liệu.



5.1 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)

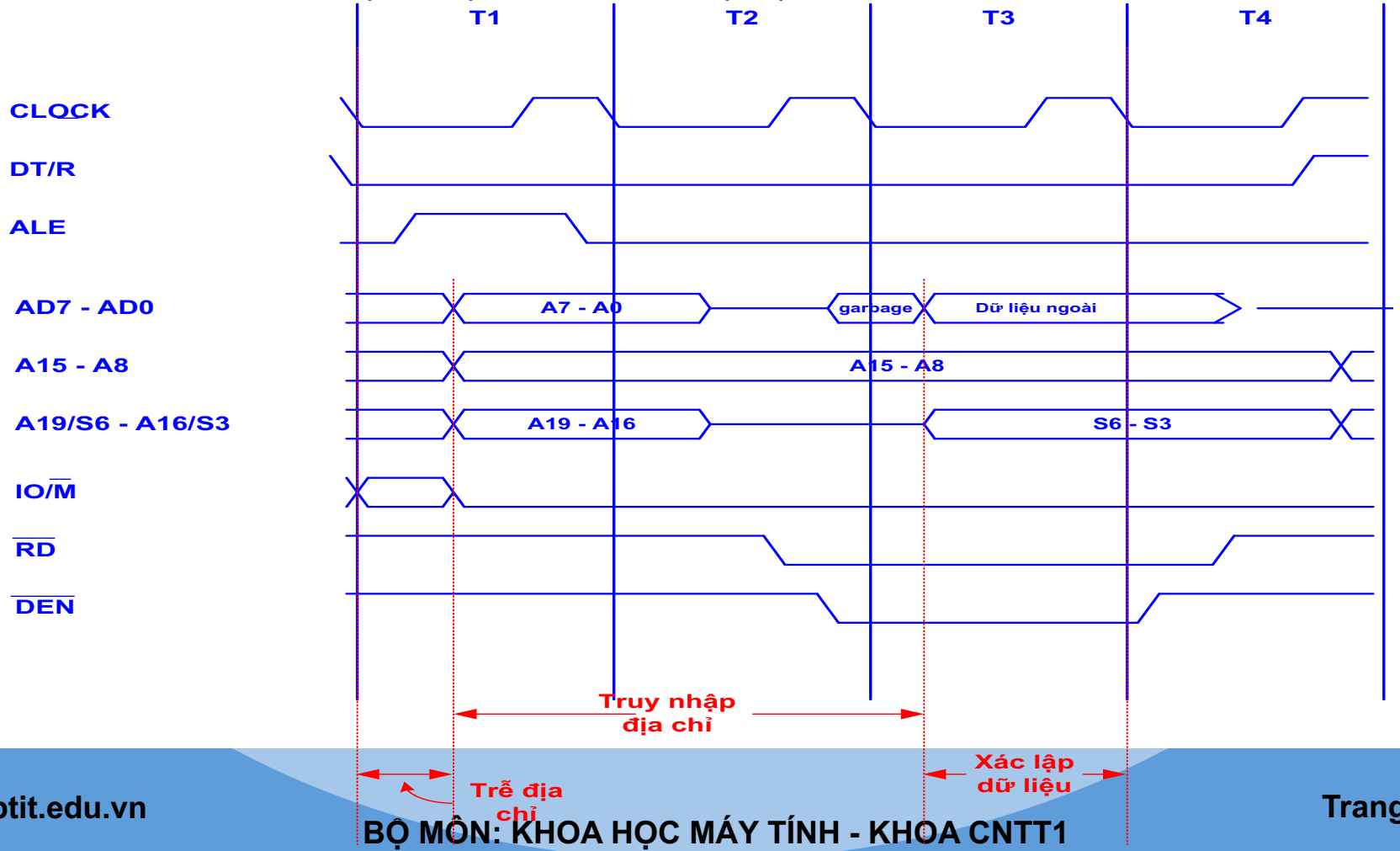
5.1 Định thời và chu trình đọc ghi bus

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

5.1 Chu trình đọc bus

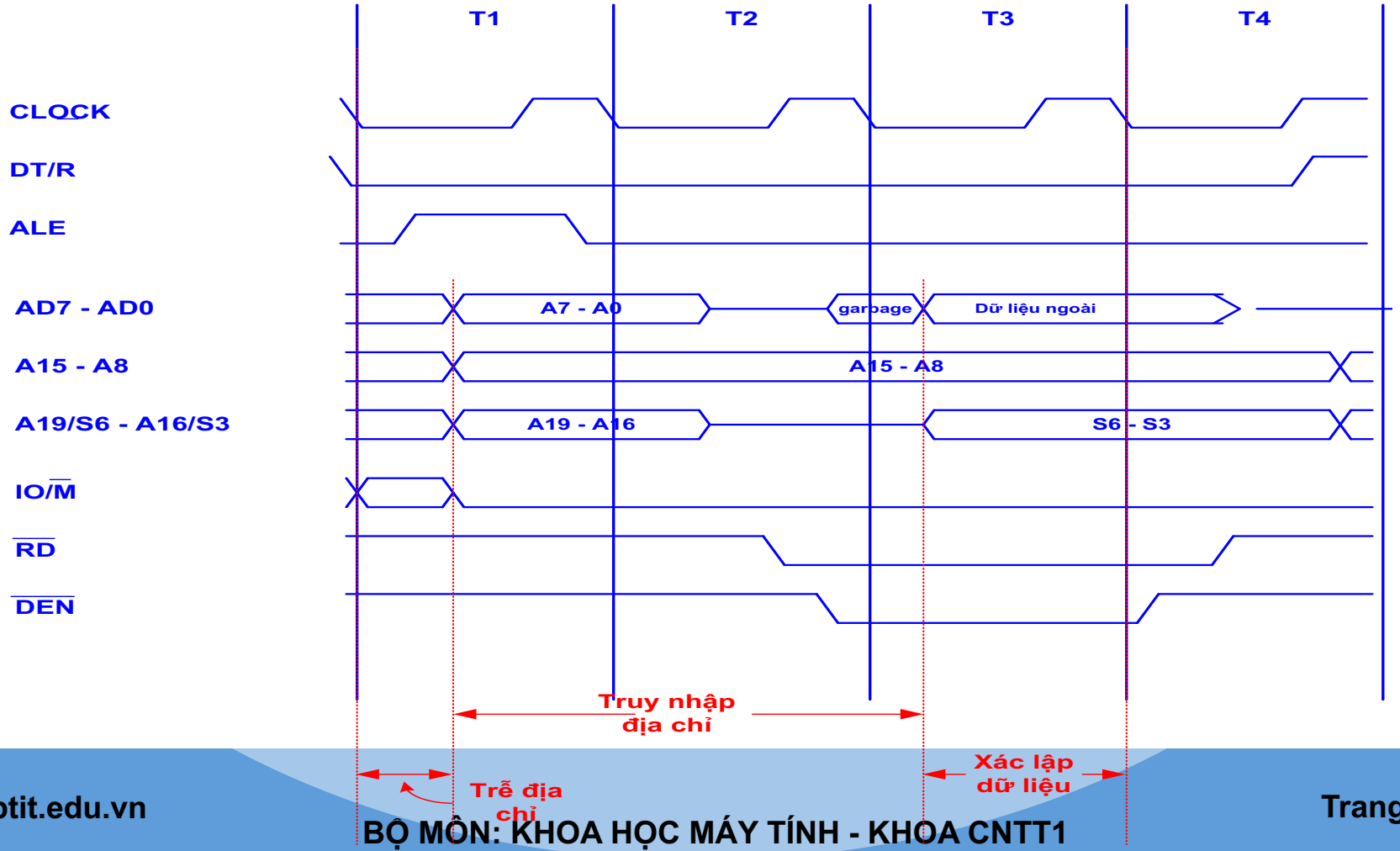
DT/$\overline{R}$: Tín hiệu xác định chiều vận chuyển dữ liệu trên bus dữ liệu. DT/$\overline{R}$=1 → dữ liệu đi ra từ CPU; DT/$\overline{R}$=0 → dữ liệu đi đến CPU.

Khi chân chốt ALE=0 → tín hiệu dữ liệu, ALE=1 → tín hiệu địa chỉ.



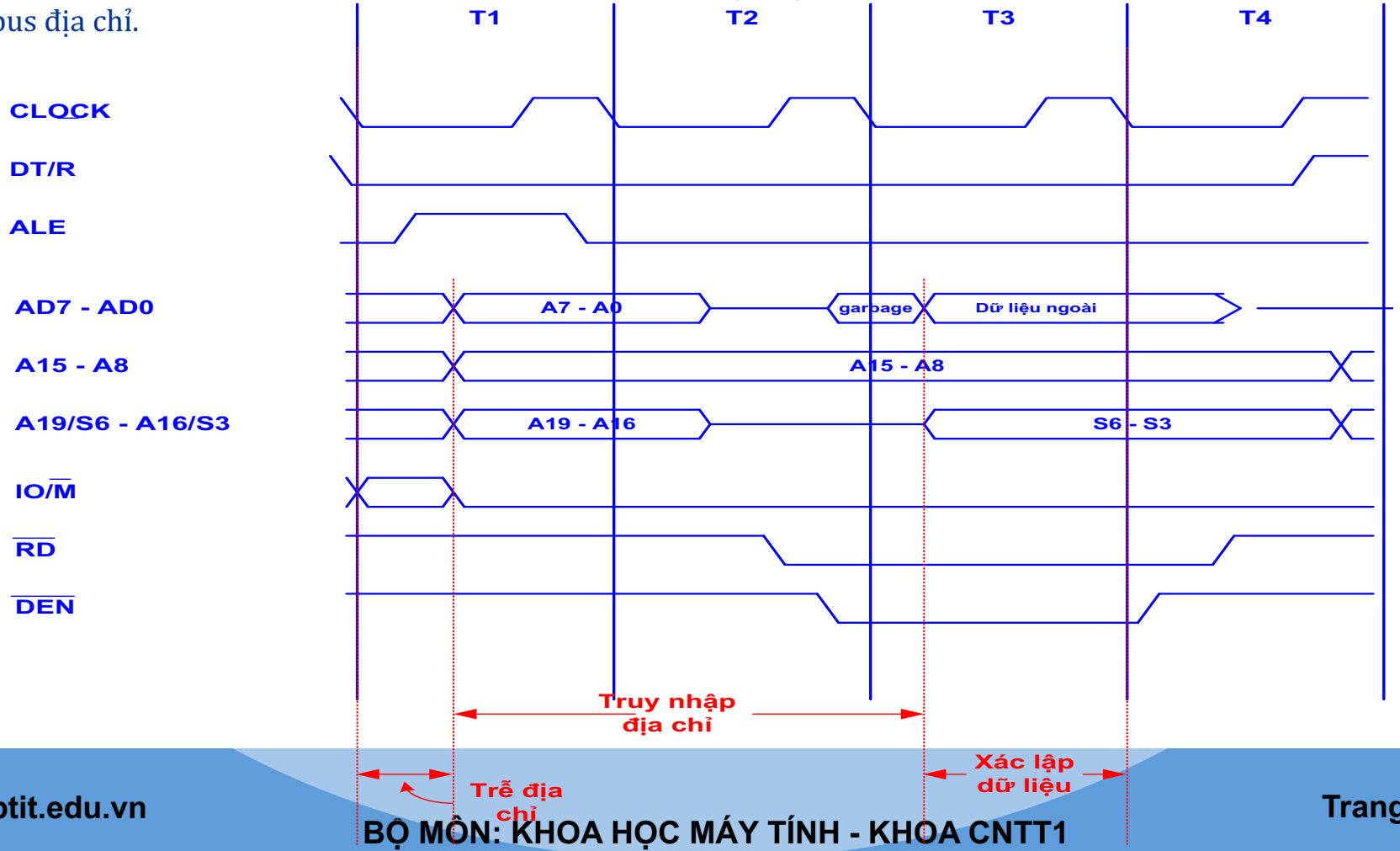
5.1 Chu trình đọc bus

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



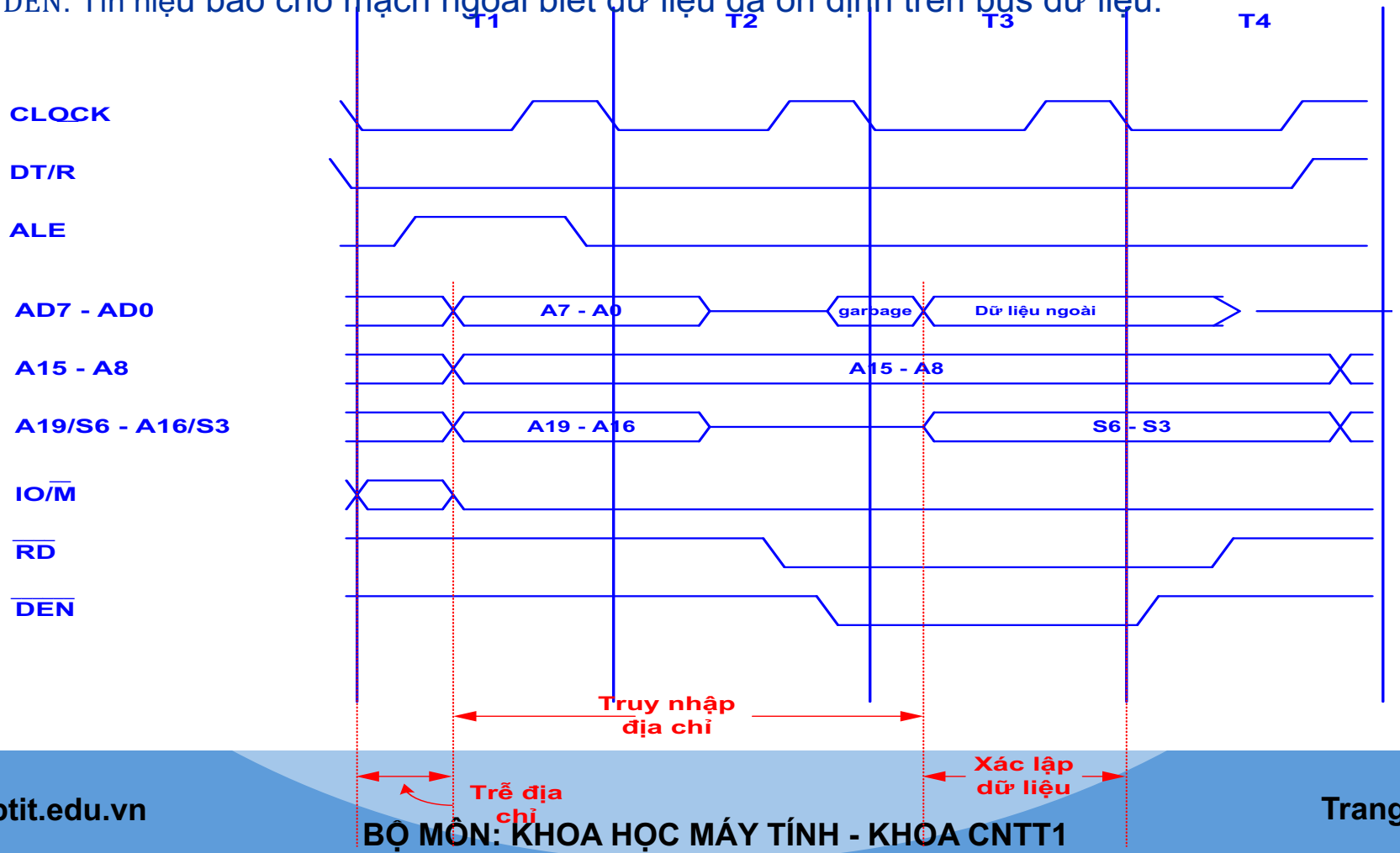
5.1 Chu trình đọc bus

- IO/$\overline{M}$: tín hiệu CPU chọn làm việc với thiết bị vào ra hay bộ nhớ.
- IO/$\overline{M}$ =1 → CPU chọn làm việc với thiết bị vào ra;
- IO/$\overline{M}$ =0 → CPU chọn làm việc với bộ nhớ. Địa chỉ tương ứng của bộ phận được lựa chọn xuất hiện trên bus địa chỉ.

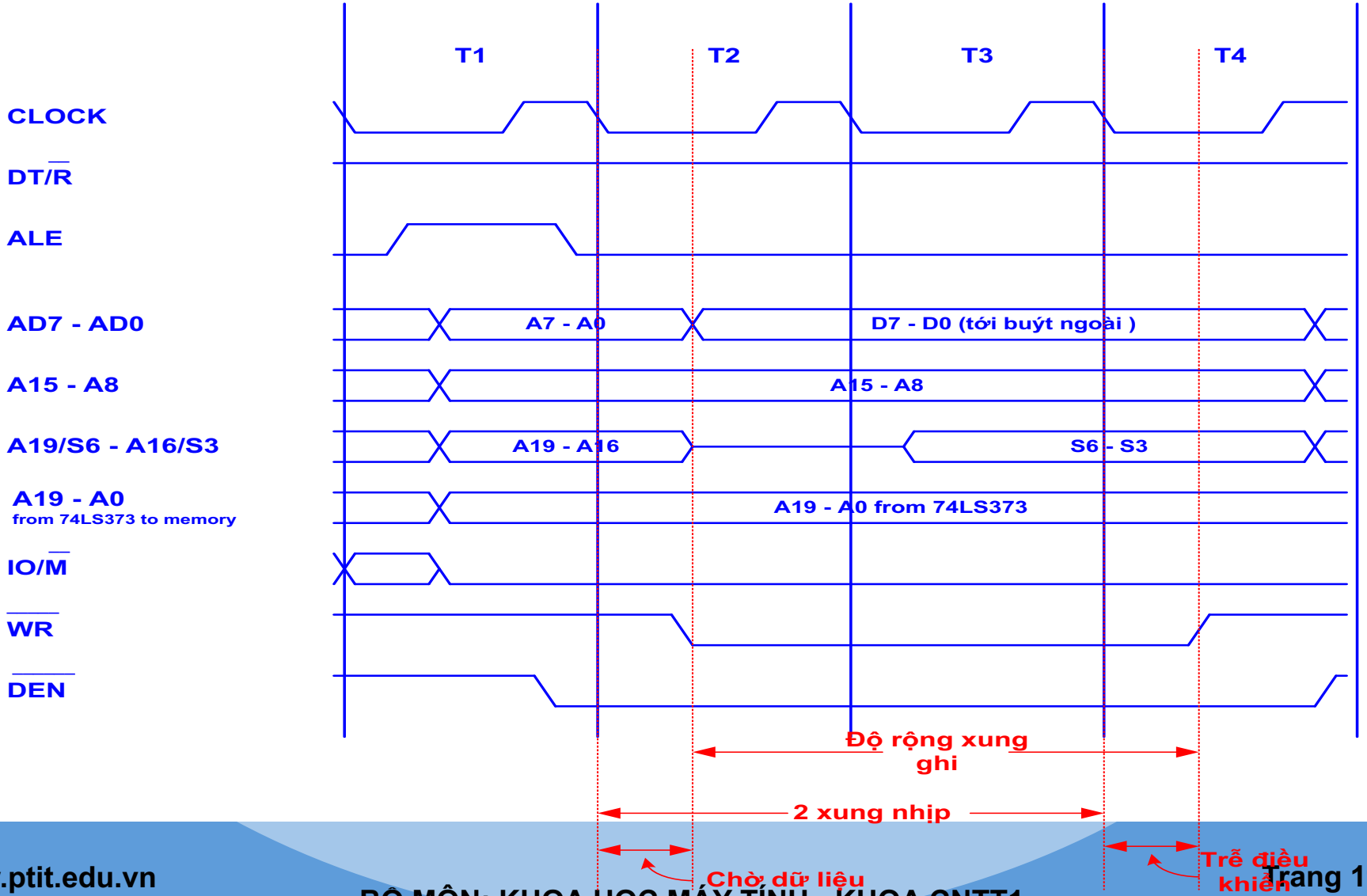


5.1 Chu trình đọc bus

- $\overline{RD}$: Xung cho phép đọc (đảo). Khi $\overline{RD}$ = 0 bus dữ liệu sẵn sàng nhận dữ liệu từ bộ nhớ hoặc thiết bị ngoại vi.
- $\overline{DEN}$: Tín hiệu báo cho mạch ngoài biết dữ liệu đã ổn định trên bus dữ liệu.



5.1 Chu trình ghi bus

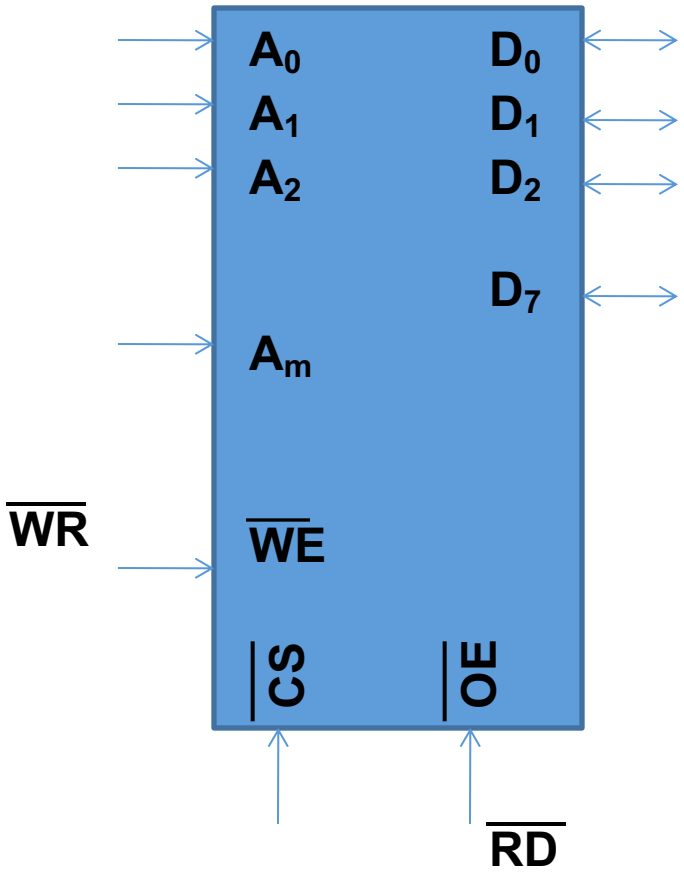


5.2 Phối ghép CPU với bộ nhớ

- Vai trò:
  - Chọn mạch nhớ cần đọc ghi
  - Chọn ô nhớ cần đọc ghi
- Đầu vào:
  - 20 bit địa chỉ vật lý
  - Các tín hiệu IO/M và RD (đọc) hoặc WR (ghi)
- Các loại mạch nhớ:
  - ROM/EPROM
  - SRAM
  - DRAM
- Mạch phối ghép: NAND, 74LS134, EPROM

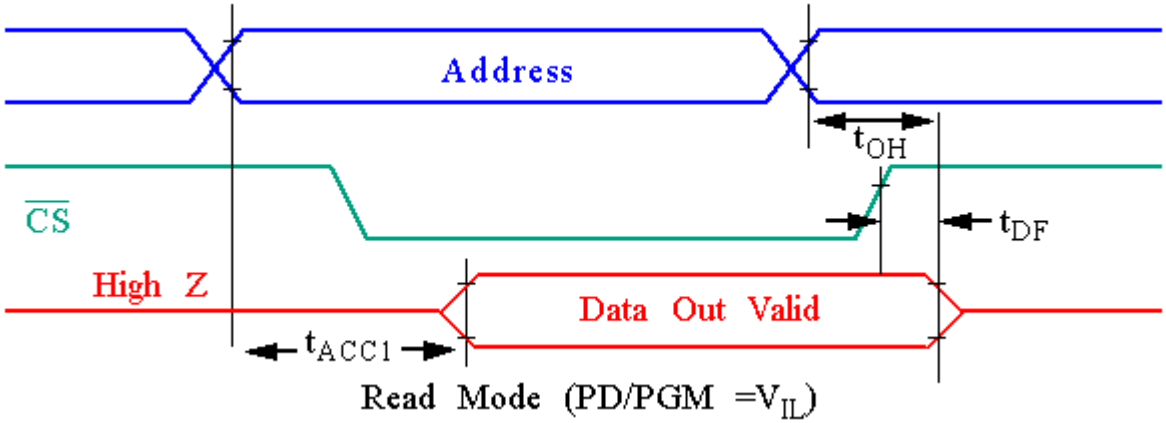
5.2.1 Cấu trúc mạch nhớ - tổng quát

- A_1-A_m: Địa chỉ
- D_0-D_7: Dữ liệu
- WE: Cho phép ghi
- OE: Cho phép ra
- CS: Kích hoạt

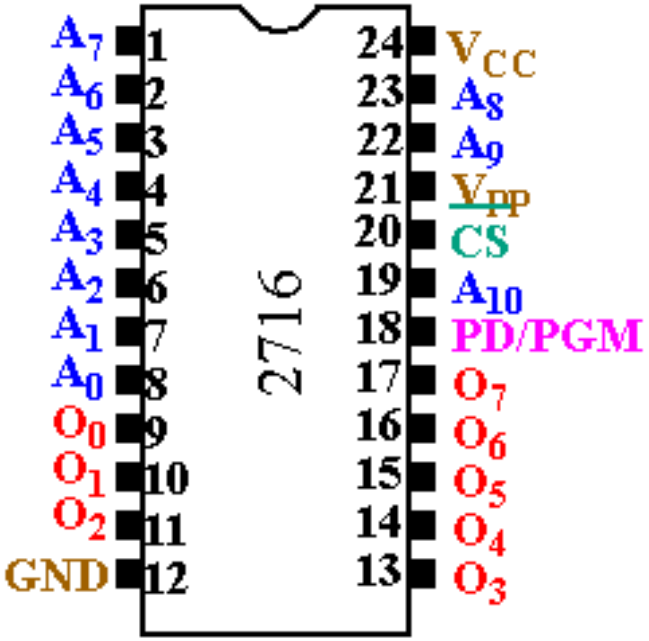


4.3.1 Cấu trúc mạch nhớ - EFROM Intel 2176(2Kx8)

- A_0-A_{10}: Tín hiệu địa chỉ
- O_0-O_7: Tín hiệu dữ liệu
- CS: chọn chíp
  (0-đọc,1-ghi)
- PD/PGM: Duy trì/Lập trình
  V_{pp} = 25V



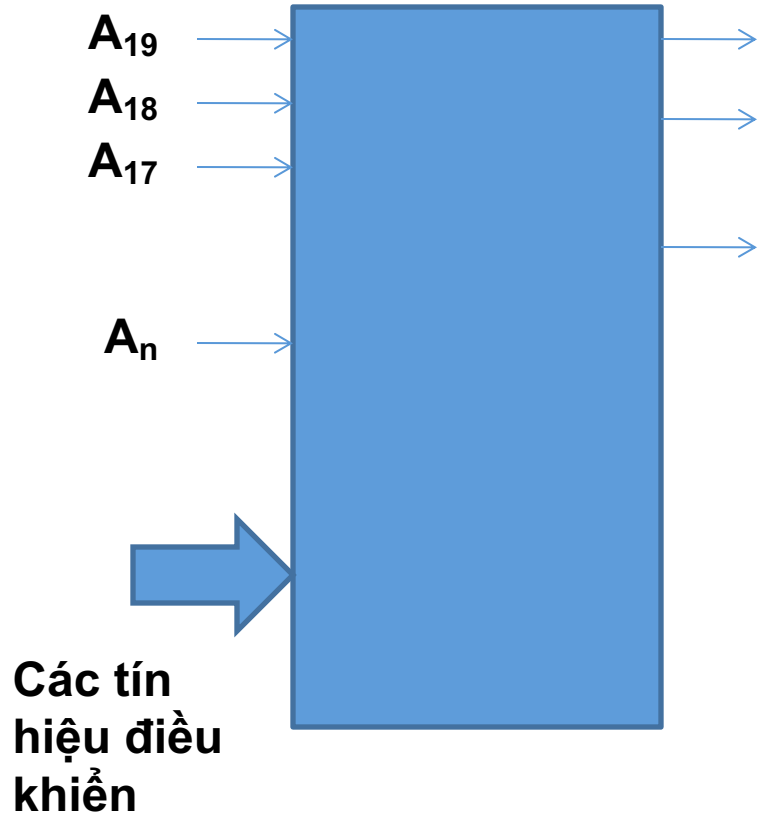
Read Mode (PD/PGM =V_{IL})



2K x 8 EPROM

5.2.2 Giải mã địa chỉ bộ nhớ

- Ánh xạ các tín hiệu địa chỉ thành tín hiệu chọn (kích hoạt) chíp nhớ
  - $A_{19}A_{18}..A_n \rightarrow CS_0,CS_1,\ldots,CS_n$
- Giải mã đầy đủ
  - Sử dụng $A_{19}A_{18}..A_n$
  - Tín hiệu đầu ra chọn duy nhất 1 mạch nhớ.
- Giải mã rút gọn
  - Sử dụng $A_{19}A_{18}..A_m;m>n$
  - Tín hiệu đầu ra có thể chọn nhiều hơn 1 mạch nhớ.



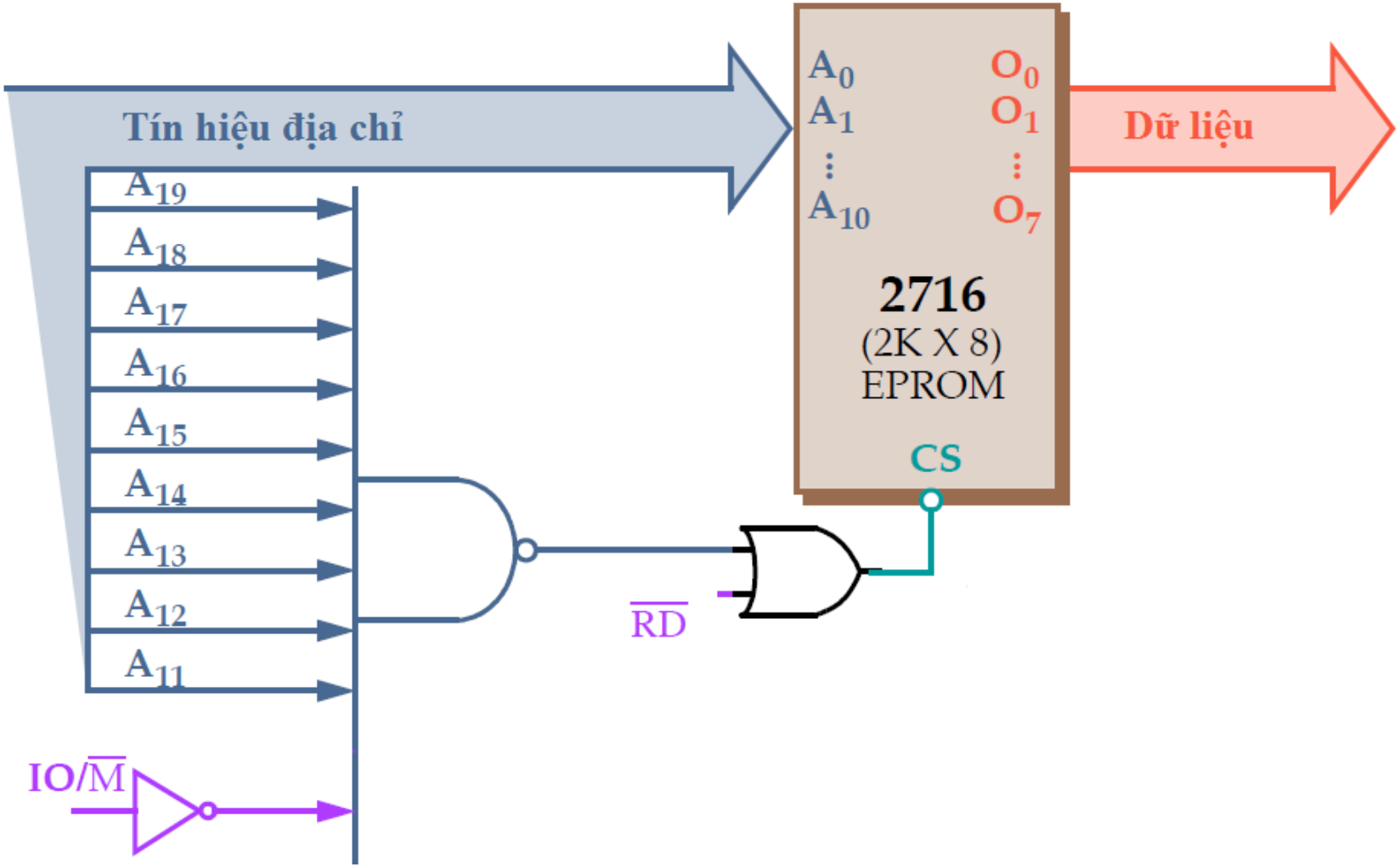
5.2.2 Giải mã đ.c b.nhớ sử dụng mạch lôgic cơ bản

- Chíp nhớ ROM 2K×8
- Khoảng địa chỉ cấp: FF800–FFFFF
- Tín hiệu địa chỉ dùng để địa chỉ hóa các ô nhớ trong chip ROM 2K: 11bit (A_0-A_{10}).
- Tín hiệu địa chỉ dùng để chọn chíp
  - $A_{19}…A_{16}A_{15}A_{12}A_{11}$
  - 1111 1111 1000 0000 0000 – 1111 1111 1111 1111 1111

NOT AND

- CS = RD OR NOT(($A_{19}…A_{16}A_{15}A_{12}A_{11}$)AND(IO/M))

5.3.2 Giải mã đ.c b.nhớ sử dụng mạch lôgic cơ bản

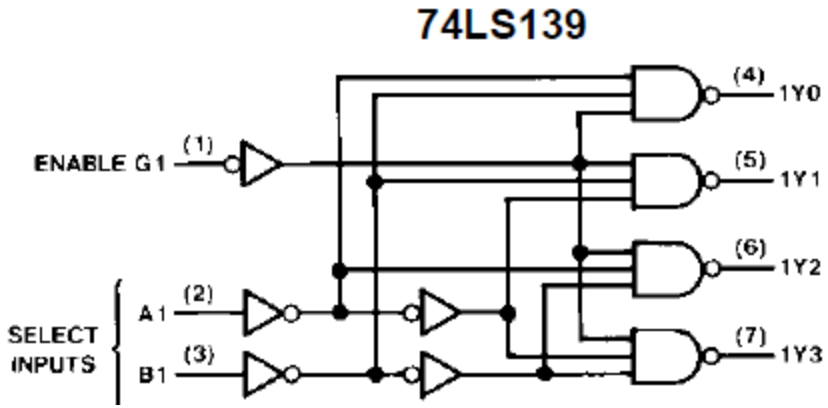


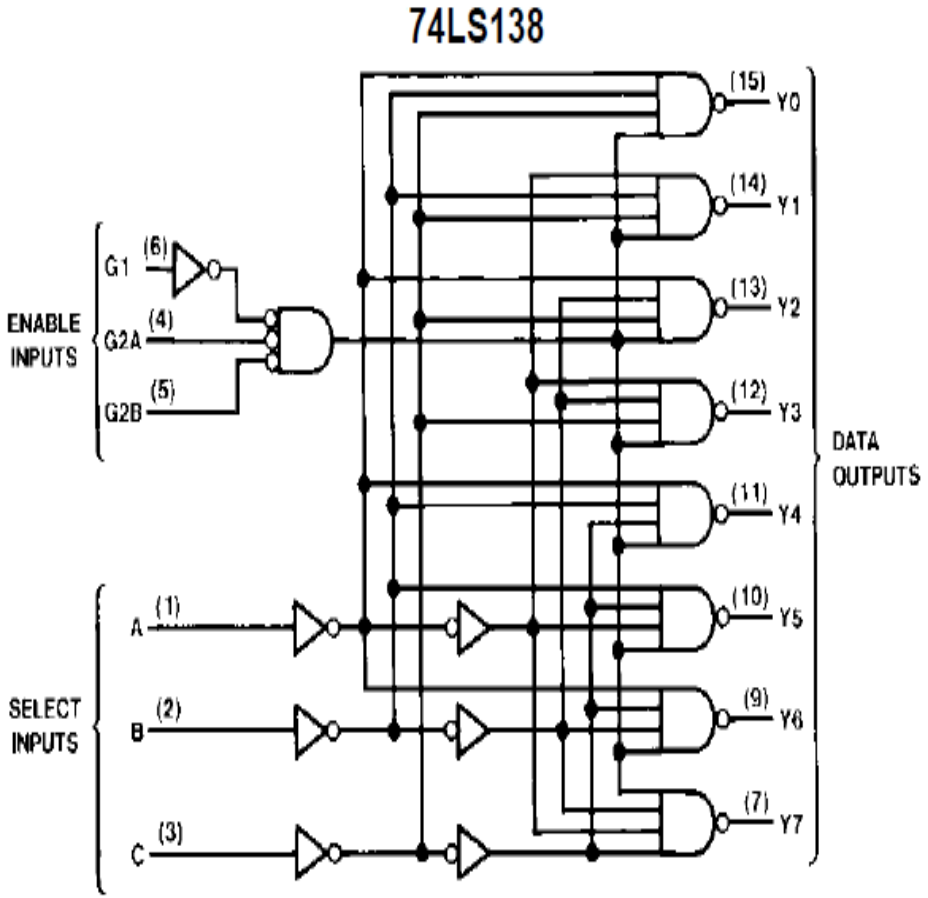
5.2.2 Giải mã đ.c b.nhớ sử dụng mạch lôgic cơ bản

- Ưu điểm
  - Cho phép tạo mạch giải mã đầy đủ
  - Tương đối đơn giản rẻ tiền khi chỉ cần 1 hoặc ít đầu ra.
- Nhược điểm:
  - Cồng kềnh khi cần giải mã cho nhiều đầu ra do số mạch tăng nhanh.

5.2.3 Giải mã đ.c b.nhớ sử dụng mạch tích hợp

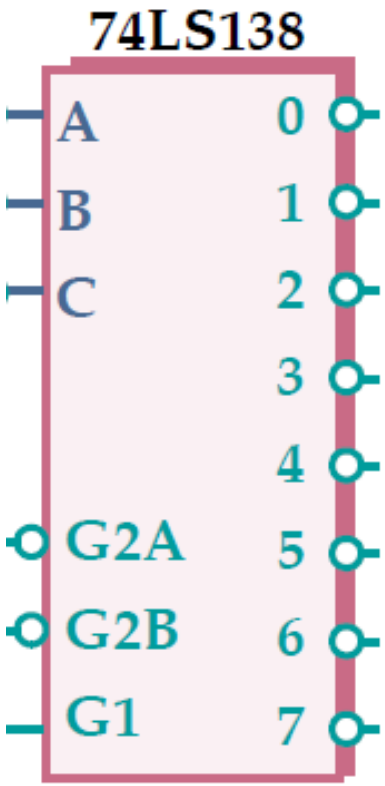
- 74-138 mạch giải mã 3→8
- 74-139 mạch giải mã 2→4





5.2.3 Giải mã đ.c b.nhớ sử dụng mạch tích hợp

- Bảng dữ liệu mạch giải mã 74LS138



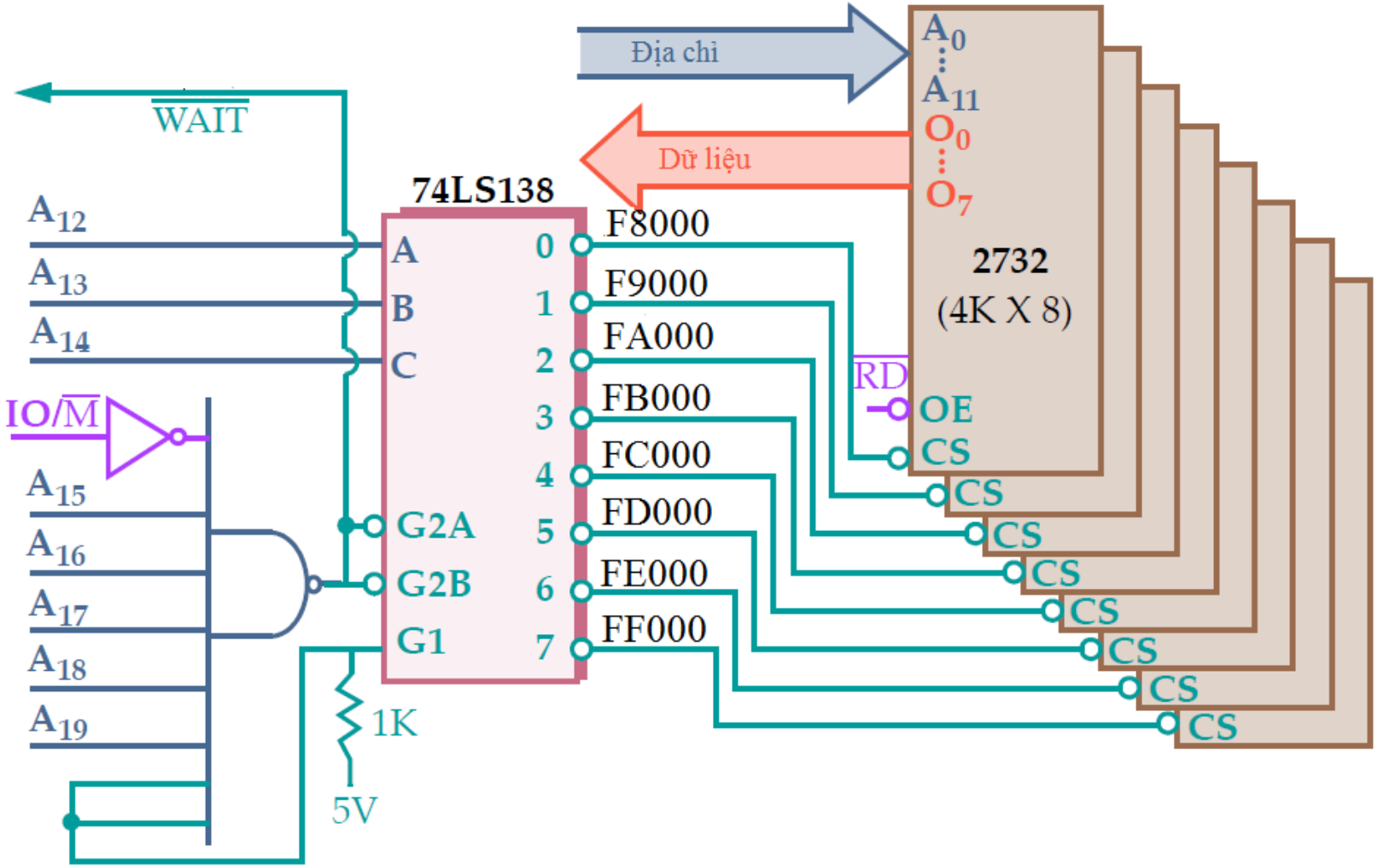
Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
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

5.2.3 Giải mã đ.c b.nhớ sử dụng mạch tích hợp

- Chíp nhớ EPROM 4K×8
- Khoảng địa chỉ cấp: F8000–FFFFF (32KB)
- Tín hiệu địa chỉ dùng để địa chỉ hóa các ô nhớ trong chip EPROM 4K: 12bit (A_0-A_{11}).
- Tín hiệu địa chỉ dùng để chọn chíp: A_{19}…$A_{16}A_{15}$…..A_{12}
  - Các tín hiệu địa chỉ $A_{12}A_{13}A_{14}$ thay đổi, còn các tín hiệu A_{15}-A_{19} không thay đổi và luôn bằng 1.

➔ $A_{12}A_{13}A_{14}$ đưa vào các đầu vào A, B, C của mạch giải mã, còn các tín hiệu địa chỉ còn lại A_{15}-A_{19} và tín hiệu điều khiển IO/M được nối vào tín hiệu điều khiển của 74LS138 (G2A, G2B). Tín hiệu G1 luôn ở mức lô-gíc 1. Các đầu ra của 74LS138 được nối lần lượt với các mạch nhớ ứng với dải địa chỉ gán trước.

5.2.3 Giải mã đ.c b.nhớ sử dụng mạch tích hợp



5.2.3 Giải mã đ.c b.nhớ sử dụng mạch tích hợp

- Ưu điểm
  - Cho phép tạo mạch giải mã đầy đủ
  - Cho phép tạo mạch giải mã chấp nhận một số hạn chế đầu vào và tạo ra một số hạn chế tín hiệu chọn mạch đầu ra.
- Nhược điểm:
  - Không thích hợp với mạch giải mã cần chấp nhận một số lượng lớn tín hiệu đầu vào và sinh ra nhiều tín hiệu đầu ra.

→ Cần sử dụng bổ sung mạch logic phụ thì mạch tích hợp mới có thể cho phép giải mã đầy đủ.

5.2.4 Giải mã đ.c b.nhớ sử dụng PROM

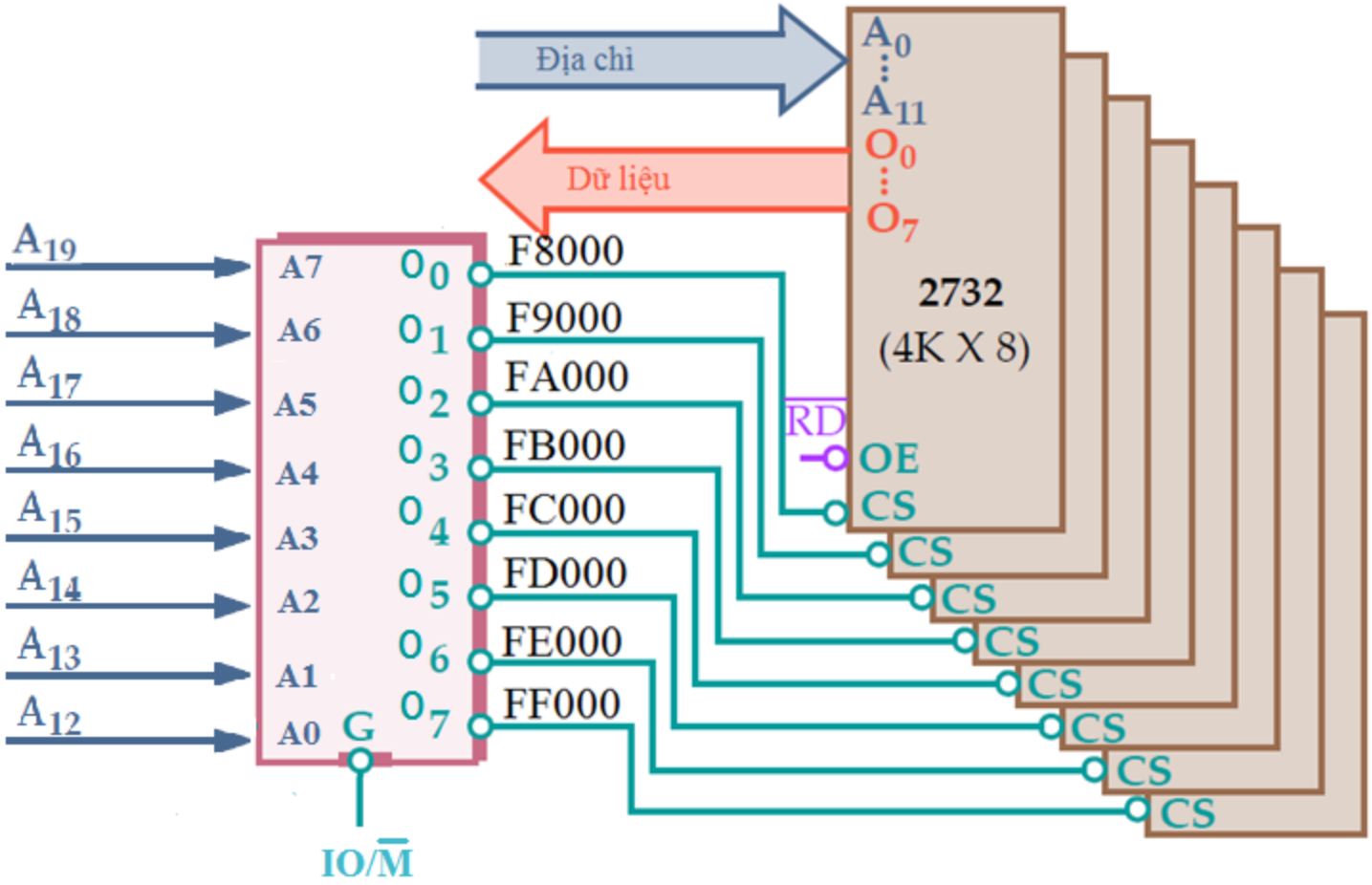
- Bộ nhớ ROM/PROM có thể được sử dụng làm bộ giải mã do:
  - Chấp nhận một nhóm tín hiệu địa chỉ và điều khiển đầu vào
  - Sinh ra một nhóm các tín hiệu dữ liệu đầu ra; Trạng thái của các tín hiệu dữ liệu này tùy thuộc vào giá trị được lưu vào trong ROM trước đó.
  - Nếu các tín hiệu dữ liệu đầu ra loại trừ lẫn nhau thì chúng có thể được dùng làm các tín hiệu chọn vi mạch nhớ.
  - Ví dụ: sử dụng PROM 256 byte để làm bộ giải mã cho các chíp nhớ 2732 4Kx8 vào không gian địa chỉ F8000-FFFFF.

5.2.4 Giải mã đ.c b.nhớ sử dụng PROM

- Mẫu dữ liệu ghi vào PROM 256 bytes:
  - Chỉ 8 ô nhớ (nằm trên đường chéo) lưu giá trị ở mức thấp (00)
  - Còn tất cả các ô nhớ khác giá trị ở mức cao (FF)

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	$\_O_0$	$\_O_1$	$\_O_2$	$\_O_3$	$\_O_4$	$\_O_5$	$\_O_6$	$\_O_7$
1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1
1	1	1	1	1	0	0	1	1	0	1	1	1	1	1	1
1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1
1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

5.2.4 Giải mã đ.c b.nhớ sử dụng PROM

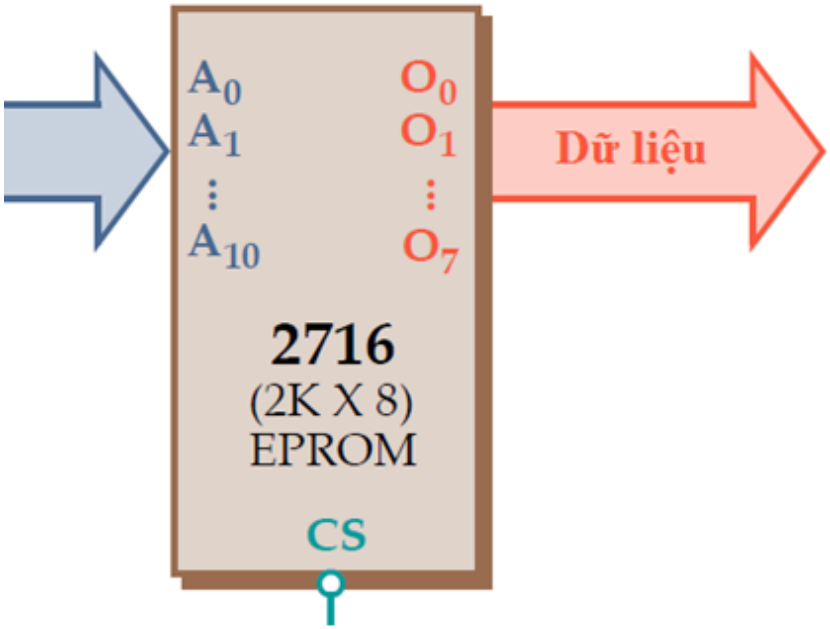


5.2.4 Giải mã đ.c b.nhớ sử dụng PROM

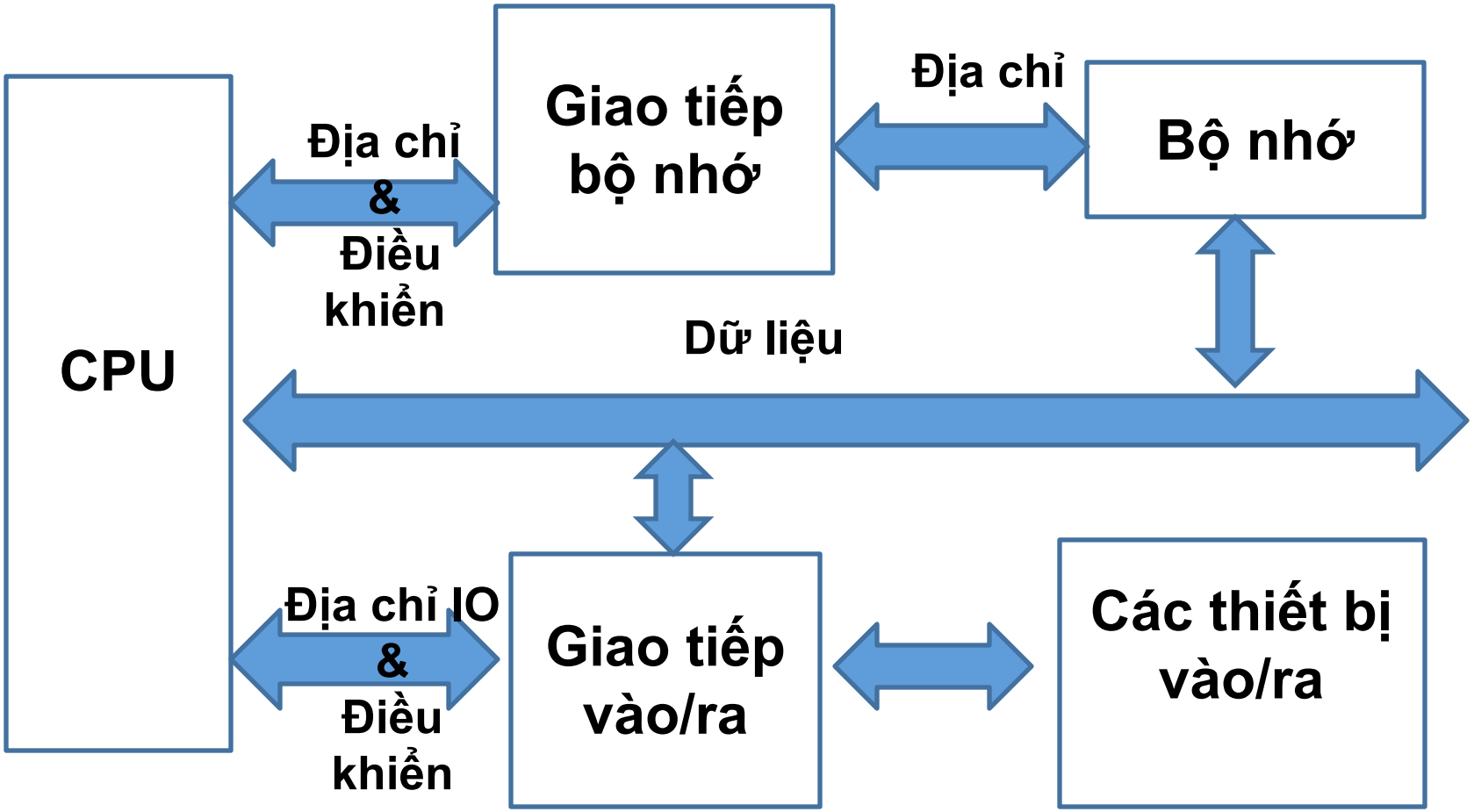
- Ưu điểm
  - Cho phép tạo mạch giải mã đầy đủ mà không cần phải sử dụng mạch phụ trợ → giảm kích thước bộ giải mã.
  - Cho phép tạo mạch giải mã chấp nhận nhiều tín hiệu đầu vào và tạo ra một lớn tín hiệu chọn mạch đầu ra.
  - Dễ dàng thay đổi địa chỉ của các mạch nhớ bằng cách thay đổi vị trí và giá trị dữ liệu trong mạch nhớ giải mã ROM.
- Nhược điểm:

Bài tập bổ sung – Xây dựng mạch giải mã địa chỉ

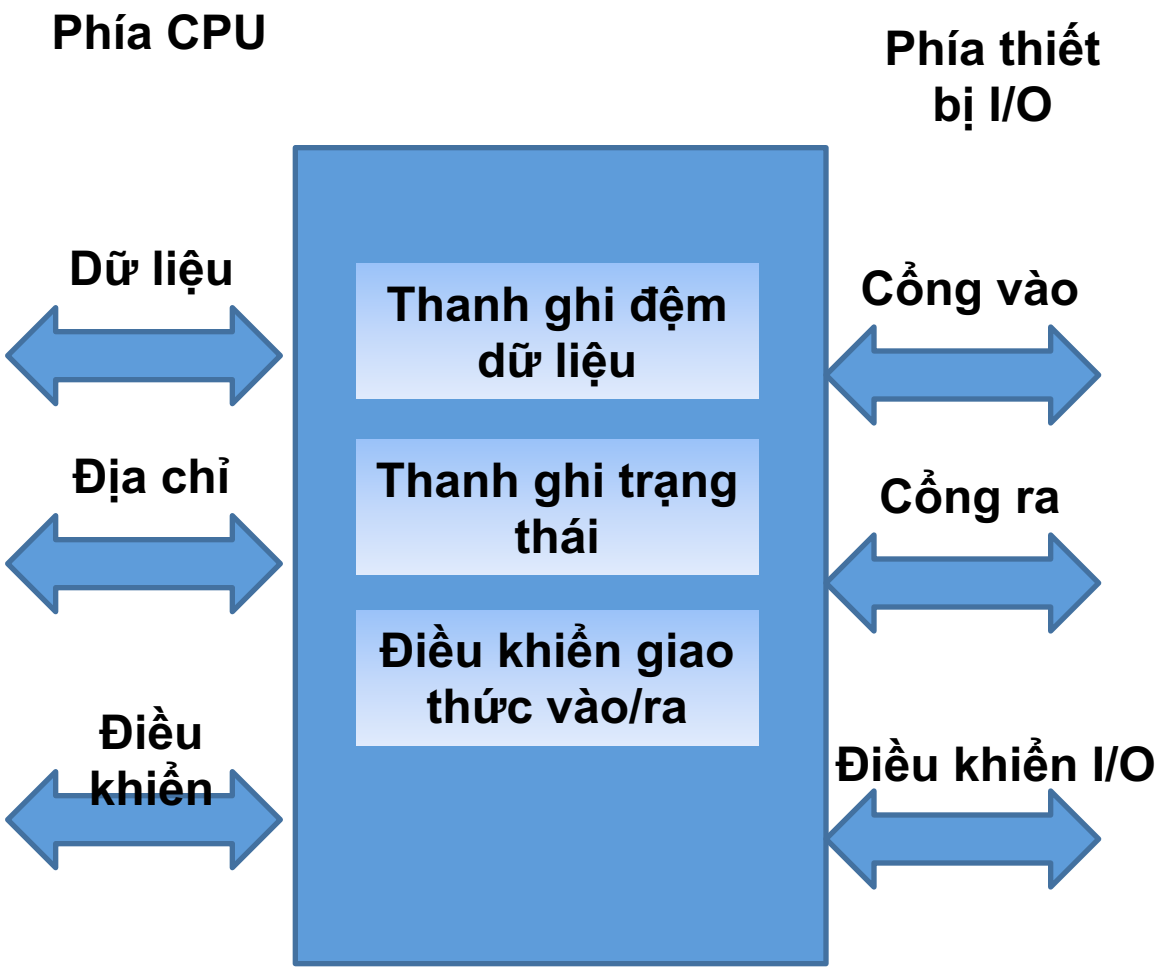
1. Xây dựng mạch giải mã địa chỉ dùng các mạch lô-gíc cơ bản cho bộ nhớ ROM dung lượng 8KB có địa chỉ cơ sở 05800H dùng vi mạch nhớ 2Kx8, và mạch 74LS138.



5.3. Phối ghép CPU với thiết bị vào ra

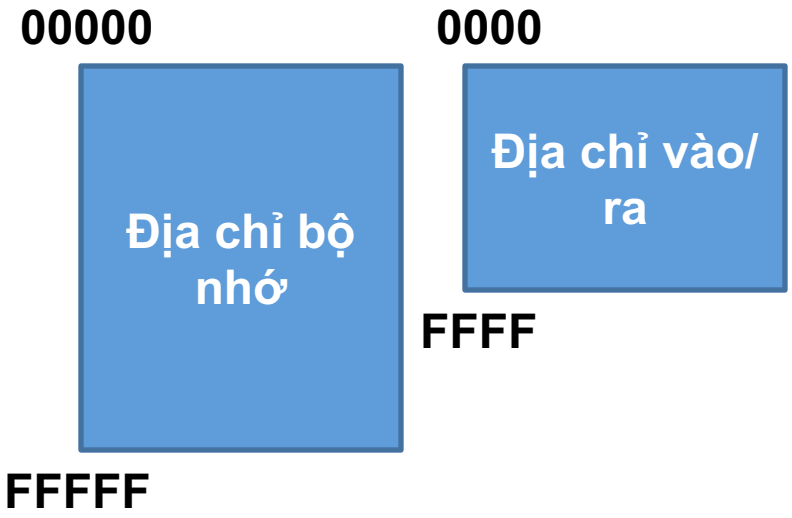


5.3. Phối ghép CPU với thiết bị vào ra



5.3.1 Phân loại thiết bị vào ra theo không gian địa chỉ

- Thiết bị vào/ra có không gian địa chỉ tách biệt
- Thiết bị vào/ra dùng chung không gian địa chỉ với bộ nhớ

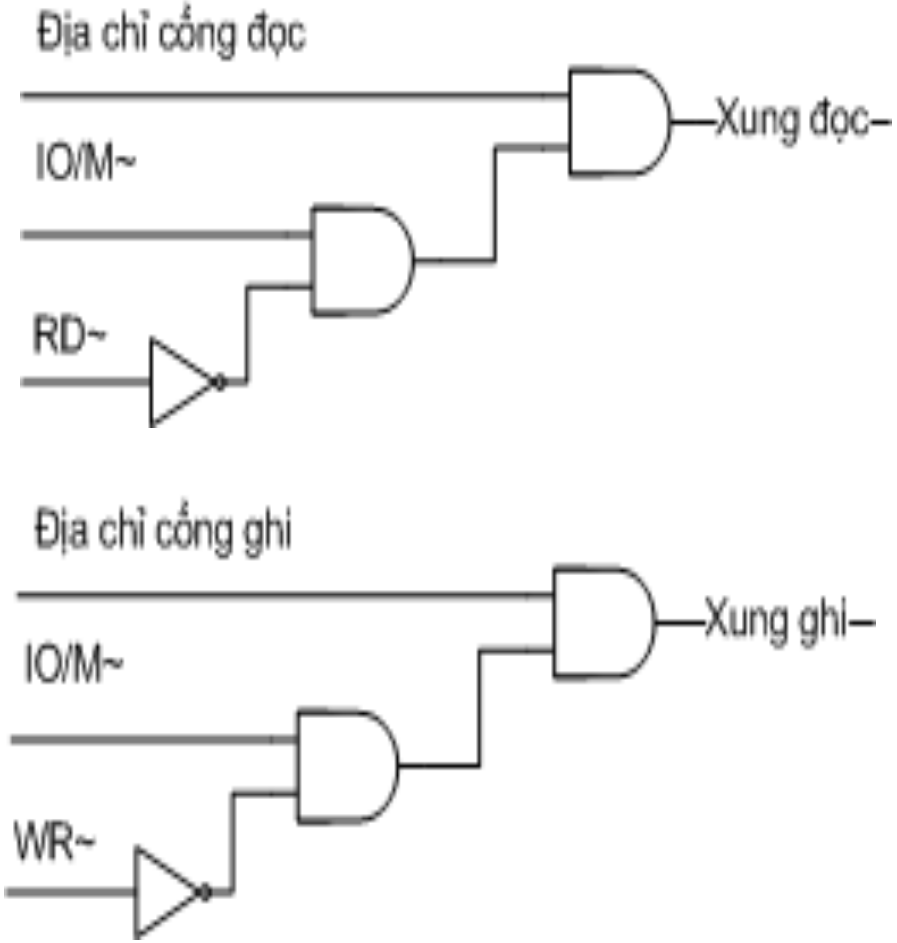


5.3.1 Phân loại thiết bị vào ra theo không gian địa chỉ

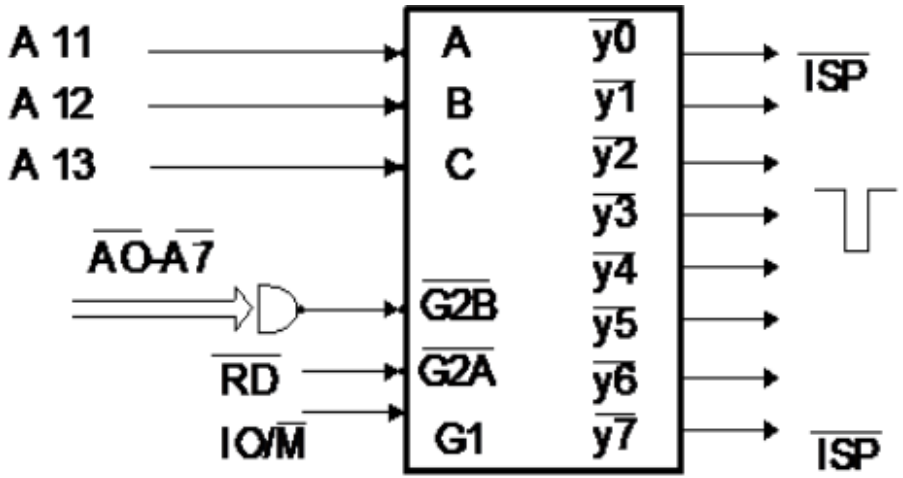
- Thao tác đọc/ghi dữ liệu với không gian địa chỉ tách biệt:
  - IN AX, [Địa chỉ cổng]
  - OUT [Địa chỉ cổng], AX
  - Địa chỉ cổng vào/ra
    - 0000-FFFF: Lưu trong DX
    - 00-FF: địa chỉ trực tiếp
- Thao tác đọc/ghi dữ liệu với không gian địa chỉ dùng chung:
  - MOV [Địa chỉ cổng], AX
  - Đọc: MOV AX, [Địa chỉ cổng]
  - Địa chỉ cổng vào/ra
    - 00000-FFFFF

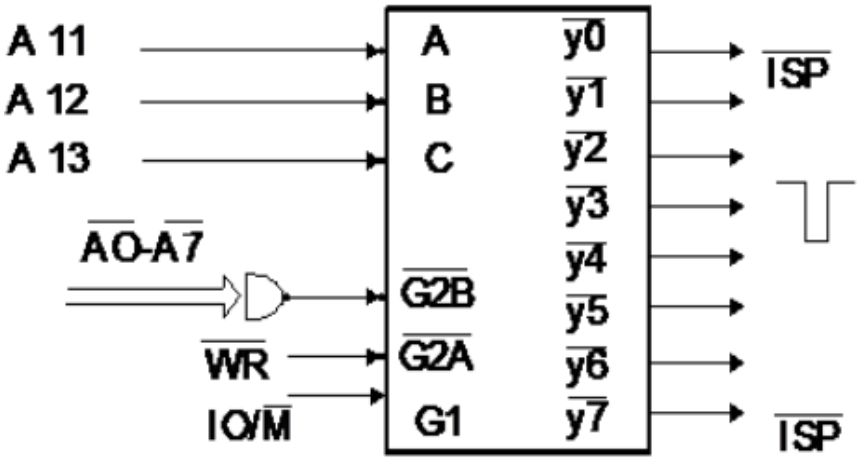
5.3.2 Giải mã đ.chỉ t.b vào ra sử dụng cổng logic

- Tổ hợp các tín hiệu địa chỉ và điều khiển thành xung đọc/ghi
  - Địa chỉ riêng
    - IO + RD~ + A_i…A_j = IN
    - IO + WR~ + A_i…A_j = OUT
  - Địa chỉ chung với bộ nhớ
    - M~ + RD~ + A_i…A_j = IN
    - M~ + WR~ + A_i…A_j = OUT



5.3.2 Giải mã đ.chỉ tb vào ra sử dụng mạch tích hợp



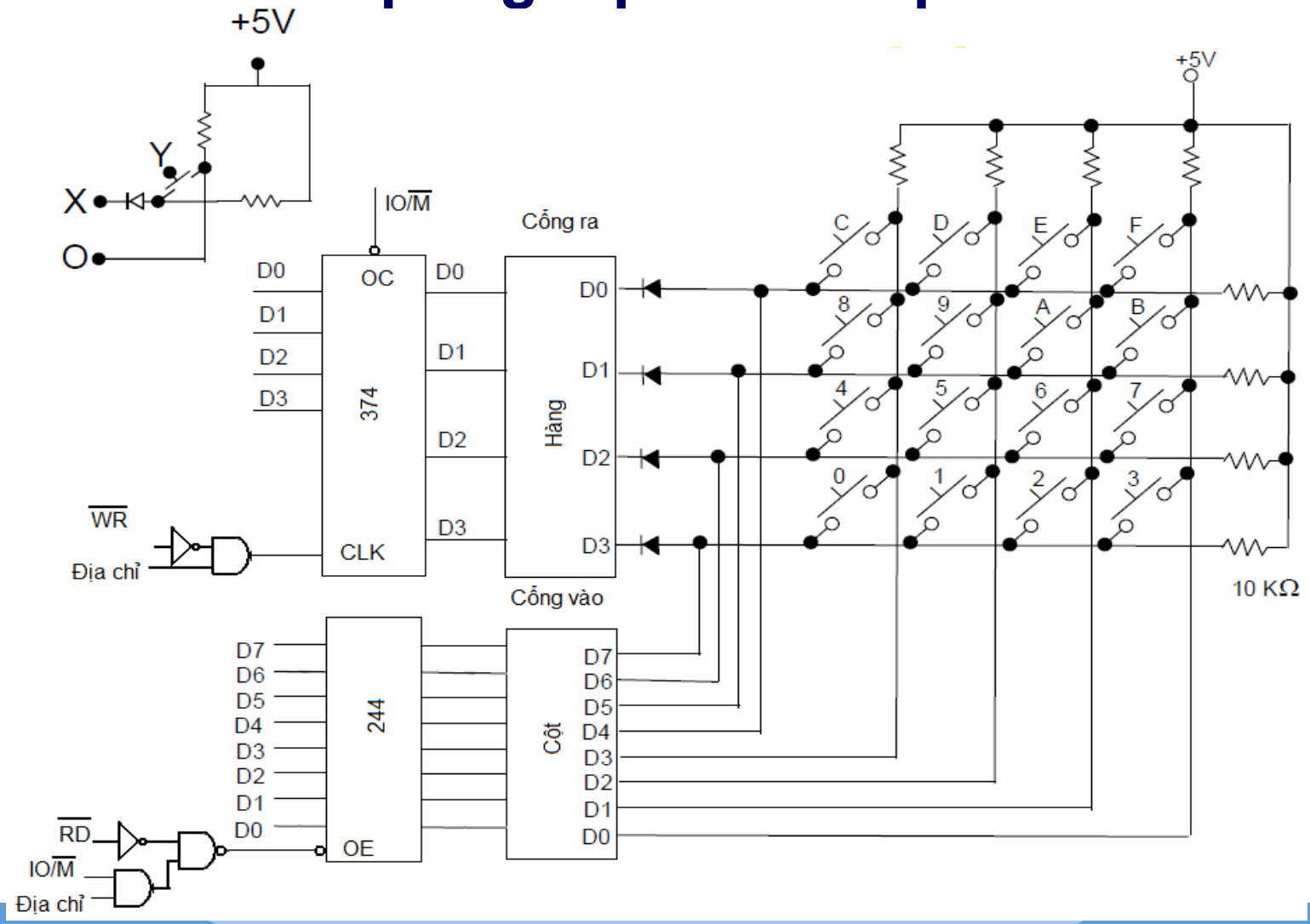


Giải mã địa chỉ cổng dùng 74LS138

5.4 Lập trình điều khiển bàn phím

- Có thể sử dụng các mạch tích hợp cỡ vừa để làm cổng phối ghép với vi xử lý để vào/ra dữ liệu. Các mạch này thường được cấu tạo từ:
  - Các mạch chốt 8 bít có đầu ra 3 trạng thái (74LS373,74LS374)
  - Các mạch khuếch đại đệm 2 chiều 8 bít đầu ra 3 trạng thái (74LS245)

5.4 Mạch ghép nối bàn phím



5.4 Mạch ghép nối bàn phím

- Cổng ghép nối bàn phím 16 số dạng tiếp điểm sử dụng các mạch tích hợp:
  - Vi mạch 74LS374 được dùng để điều khiển các tín hiệu hàng và mạch 74LS244 dùng để điều khiển các tín hiệu cột;
- Nguyên tắc hoạt động:
  - Nếu tín hiệu X ở mức cao (lô-gíc 1) thì đi-ốt sẽ khóa lại, vậy nên tiếp điểm Y có đóng xuống hay không thì tại đầu O ta luôn thu được điện áp 5V (không có dòng điện).
  - Nếu tín hiệu X ở mức thấp (lô-gíc 0), thì đi-ốt mở và khi tiếp điểm Y đóng xuống tại đầu O ta thu được điện áp 0V.
- Xác định phím được ấn: quét tuần các dòng và đọc các cột. Nếu một phím được ấn thì bit tương ứng ở cổng ra Di = 0.

5.4 Chương trình kiểm tra một phím

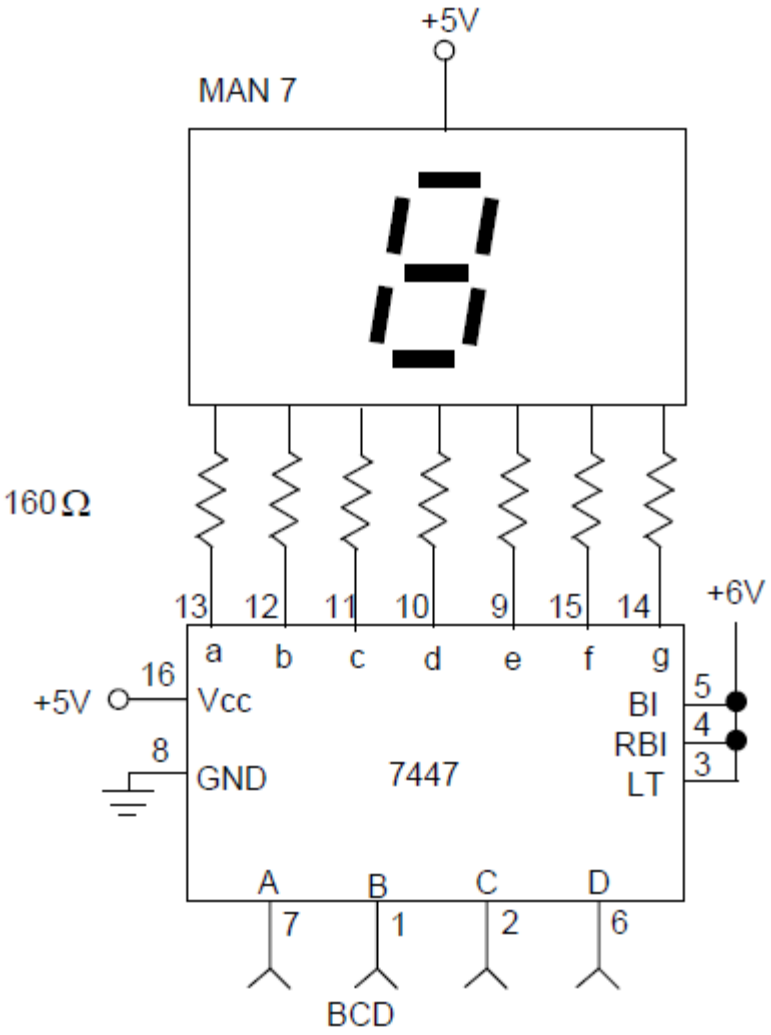
- Đoạn chương trình kiểm tra xem phím C (hàng D0, cột D3) có được bấm hay không. Biết địa chỉ cổng hàng là 0Ah và địa chỉ cổng cột là 0Bh.

```
Hang    EQU 0AH                 ; Địa chỉ cổng hàng
Cot     EQU 0BH                 ; Địa chỉ cổng cột
        MOV AL,11111110b        ; Chỉ có D0=0 – hàng thứ nhất
        OUT Hang, AL
Ktra:   IN AL, Cot              ; Đọc tín hiệu cột
        AND AL,00001000b        ; Giữ lại bít D3 ứng với phím C
        JNZ Ktra                ; Không bấm
        ...                     ; Phím C được bấm
```

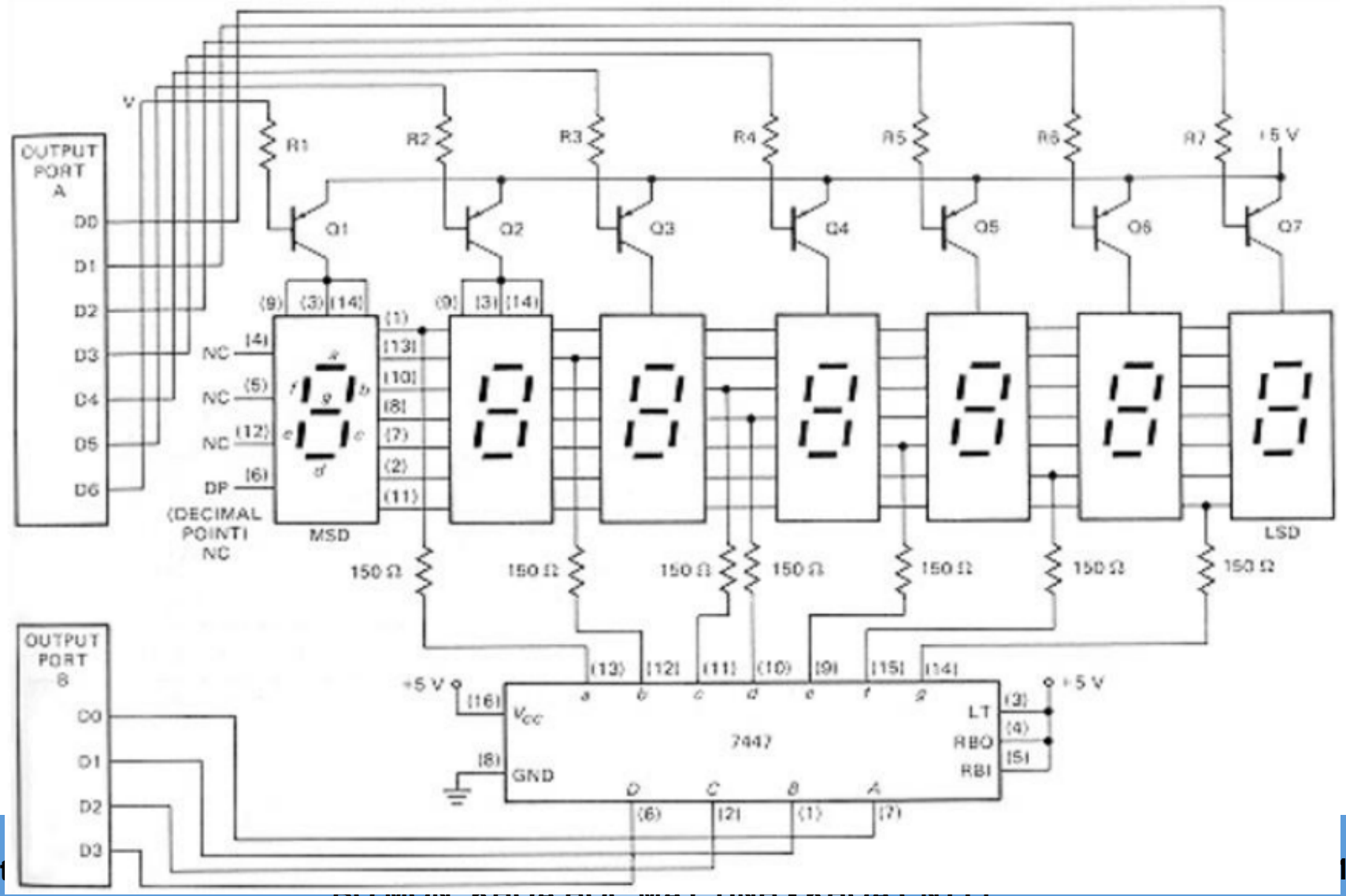
5.5 Lập trình điều khiển hệ thống đèn LED

- Ghép nối hiển thị số sử dụng mạch tích hợp 7447 và LED bảy đoạn:
  - Cổng A nhận thông tin điều khiển (bật/tắt) các thanh led thông qua 7 transitor Q1-Q7. Địa chỉ cổng A là 0Ah.
  - Cổng B nhận dữ liệu số hiển thị - thông qua mạch 7447 giải mã số đầu vào dạng BCD ở cổng B (A-D) sinh ra các tín hiệu kích hoạt (a-g) các thanh led của LED bảy đoạn. Địa chỉ cổng B là 0Bh.
  - Bật đèn led thứ i: gửi bit $D_i = 0$ ra cổng A
  - Tắt đèn led thứ i: gửi bit $D_i = 1$ ra cổng A
  - Hiển thị số: Gửi số cần hiện thị ra cổng B, bật đèn led i bằng cách gửi bit $D_i = 0$ ra cổng A.

5.5.1 Mạch ghép nối hiển thị số



5.5.1 Mạch ghép nối hiển thị số



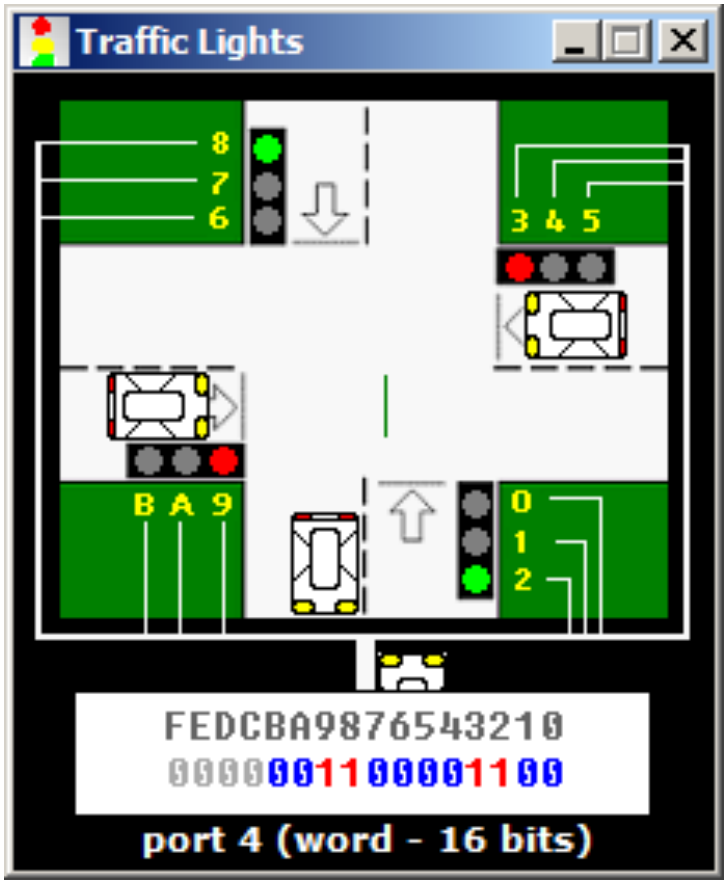
5.5.2 Chương trình hiển thị số trên LED

- Đoạn chương trình kiểm tra hệ thống LED bằng cách hiển thị trên cả 7 LED số 8. Biết địa chỉ cổng điều khiển LED là 0Ah và địa chỉ cổng dữ liệu hiển thị là 0Bh.

```
DK_LED EQU 0AH          ; Cổng điều khiển LED
DL_LED EQU 0BH          ; Cổng dữ liệu hiển thị
       MOV AL,FFH       ; Tắt tất cả các LED
       OUT DK_LED, AL
       MOV CX,64        ; Trễ bằng 64 lệnh NOP
Tre:   NOP
       LOOP Tre
       MOV AL,8         ; Đưa số 8 ra 7447
       OUT DL_LED,AL
       XOR AL,AL        ; Đặt AL=0
       OUT DK_LED,AL ; Bật tất cả các LED
```

5.6 Lập trình điều khiển đèn giao thông

- Thiết bị ảo hệ thống đèn giao thông sử dụng cổng số 4 – cổng 16 bít để nhận thông tin điều khiển;
- Sử dụng 12 bít (0-11) cho 4 cụm đèn:
  - Mỗi cụm gồm 3 đèn Green, Yellow và Red;
  - Bít 0 – tắt đèn, bít 1 – bật đèn
- 4 bít (12-15) không sử dụng – nên đặt là 0.



5.6.1 Giới thiệu thiết bị ảo – Đèn giao thông

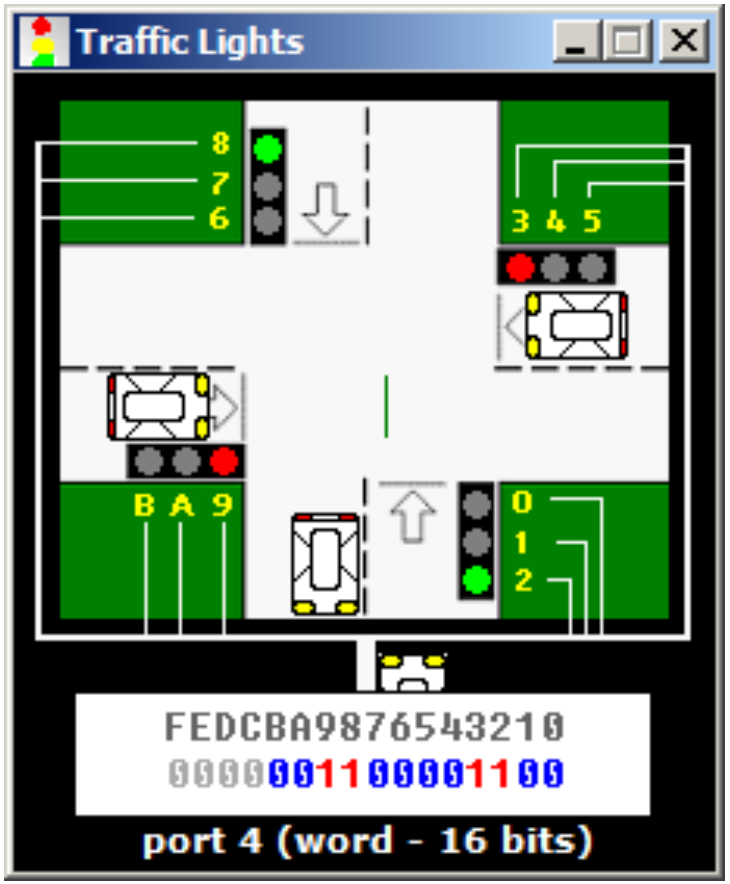
- Điều khiển đèn giao thông:
  - Gửi từ điều khiển (2 bytes) ra cổng số 4;
  - Các bít của từ điều khiển được đặt sao cho phù hợp với ý đồ điều khiển đèn (Bít 0 – tắt đèn, bít 1 – bật đèn)

VD: từ điều khiển:

0000 001 100 001 100

GYR GYR GYR GYR

  - Dùng hàm 86h của ngắt BIOS 15h để tạo thời gian đợi – thời gian giữ trạng thái vừa thiết lập của cụm đèn. Số micro giây được đặt vào CX:DX trước khi gọi ngắt.



5.6.2 Lập trình điều khiển đèn giao thông

```
mov ax, all_red
out 4, ax
mov si, offset situation
next:
mov ax, [si]
out 4, ax

; wait 5 seconds (5 million microseconds)
mov     cx, 4Ch    ;    004C4B40h = 5,000,000
mov     dx, 4B40h
mov     ah, 86h
int     15h
add si, 2 ; next situation
cmp si, sit_end
jb  next
mov si, offset situation
jmp next
```

5.6.2 Lập trình điều khiển đèn giao thông

```
;                    FEDC_BA98_7654_3210
situation       dw      0000_0011_0000_1100b
s1              dw      0000_0110_1001_1010b
s2              dw      0000_1000_0110_0001b
s3              dw      0000_1000_0110_0001b
s4              dw      0000_0100_1101_0011b
sit_end = $


all_red         equ     0000_0010_0100_1001b
```

5.6.2 Lập trình điều khiển đèn giao thông

```
.Model small
.Stack 100H
.Data
  ;                          GYR GYR GYR GYR
  R1 DW                    0000_0011_0000_1100b
  R2 DW                    0000_0010_1000_1010b
  R3 DW                    0000_1000_0110_0001b
  R4 DW                    0000_0100_0101_0001b
  ;                         FEDC_BA9 876 543 210
all_red         equ     0000_0010_0100_1001b
PORT EQU  4     ; output port

  ; time constants (in secs)
  WAIT_3_SEC_CX   EQU   2Dh
  WAIT_3_SEC_DX   EQU   0C6C0h

  WAIT_10_SEC_CX   EQU   98h
  WAIT_10_SEC_DX   EQU   9680h
```

5.6.2 Lập trình điều khiển đèn giao thông

```
.code
; define a macro
waitMacro macro t1, t2
  mov cx, t1
  mov dx, t2
  mov     ah, 86h
  int     15h
waitMacro endm

main proc
  ; initilize the ds and es registers
  mov ax, @Data
  mov ds,ax

  ; set lights to Red for all direction
  mov ax, all_red
  out PORT, ax
  waitMacro WAIT_3_SEC_CX, WAIT_3_SEC_DX
```

5.6.2 Lập trình điều khiển đèn giao thông

```
Start:
  lea si, R1
  mov ax, [si]
  out PORT, ax

  waitMacro WAIT_10_SEC_CX, WAIT_10_SEC_DX

  lea si, R2
  mov ax, [si]
  out PORT, ax

  waitMacro WAIT_3_SEC_CX, WAIT_3_SEC_DX

  lea si, R3
  mov ax, [si]
  out PORT, ax
```

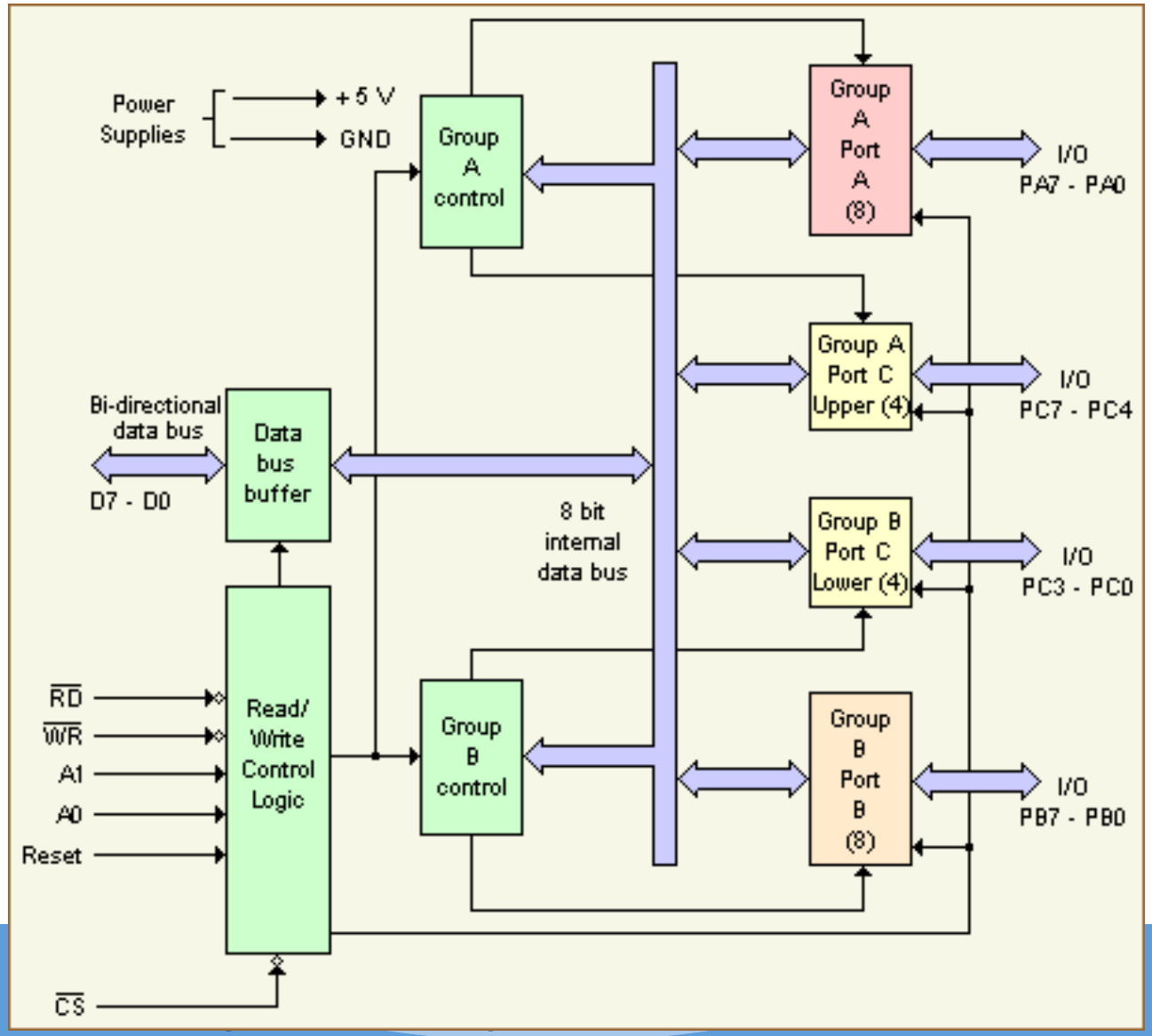
5.6.2 Lập trình điều khiển đèn giao thông

```
waitMacro WAIT_10_SEC_CX, WAIT_10_SEC_DX

  lea si, R4
  mov ax, [si]
  out PORT, ax

  waitMacro WAIT_3_SEC_CX, WAIT_3_SEC_DX

  jmp Start

  ; end program
  mov ah, 4CH
  int 21H
main endp

end main
```

5.7 Các phương pháp vào ra dữ liệu

- Vi xử lý sử dụng một số mạch chuyên dụng phục vụ trao đổi dữ liệu với các thiết bị ngoại vi theo 2 phương pháp chính:
  - Trao đổi dữ liệu kiểu song song: sử dụng mạch ghép nối 8255A.
    - Cho phép trao đổi dữ liệu nhiều bit dữ liệu đồng thời → tốc độ cao
    - Không yêu cầu phải biến đổi dữ liệu
    - Hạn chế về khoảng cách do cần nhiều dây tín hiệu
  - Trao đổi dữ liệu kiểu nối tiếp: sử dụng mạch ghép nối 8250 hoặc 8251.
    - Có thể tăng khoảng cách truyền dữ liệu do cần ít dây tín hiệu
    - Tốc độ chậm
    - Cần biến đổi dữ liệu khi gửi đi (song song → nối tiếp) và khi nhận về (nối tiếp → song song).

5.7.1 Mạch vào ra song song 8255A



5.7.2 Mạch vào ra nối tiếp 8251A