

CHƯƠNG 2

PHẦN CỨNG CHIP VI ĐIỀU KHIỂN 8051

I. TỔNG QUÁT:

1. Giới thiệu chung:

MCS-51 là họ vi điều khiển của hãng Intel. Vi mạch tổng quát của họ MCS-51 là chip 8051. Chip 8051 có một số đặc trưng cơ bản sau:

- Bộ nhớ chương trình bên trong: 4 KB (*ROM*).
- Bộ nhớ dữ liệu bên trong: 128 byte (*RAM*).
- Bộ nhớ chương trình bên ngoài: 64 KB (*ROM*).
- Bộ nhớ dữ liệu bên ngoài: 64 KB (*RAM*).
- 4 port xuất nhập (*I/O port*) 8 bit.
- 2 bộ định thời 16 bit.
- Mạch giao tiếp nối tiếp.
- Bộ xử lý bit (*thao tác trên các bit riêng lẻ*).
- 210 vị trí nhớ được định địa chỉ, mỗi vị trí 1 bit.
- Nhân / Chia trong 4 μ s.

Ngoài ra, trong họ MCS-51 còn có một số chip vi điều khiển khác có cấu trúc tương đương như:

Chip	ROM trong	RAM trong	Bộ định thời
8031	0 KB	128 byte	2
8032	0 KB	256 byte	3
8051	4 KB PROM	128 byte	2
8052	8 KB PROM	256 byte	3
8751	4 KB UV-EPROM	128 byte	2
8752	8 KB UV-EPROM	256 byte	3
8951	4 KB FLASH ROM	128 byte	2
8952	8 KB FLASH ROM	256 byte	3

2. Các phiên bản của chip vi điều khiển 8051:

2.1 Bộ vi điều khiển 8031:

8031 là một phiên bản khác của họ 8051. Chip này thường được coi là 8051 không có ROM trên chip. Để có thể dùng được chip này cần phải bổ sung thêm ROM ngoài chứa chương trình cần thiết cho 8031. 8051 có chương trình được chứa ở ROM trên chip bị giới hạn đến 4KB, còn ROM ngoài của 8031 thì có thể lên đến 64KB. Tuy nhiên, để có thể truy cập hết bộ nhớ ROM ngoài thì cần dùng thêm hai cổng (*Port 0* và *Port 2*), do vậy chỉ còn lại có hai cổng (*Port 1* và *Port 3*) để sử dụng. Nhằm khắc phục vấn đề này, chúng ta có thể bổ sung thêm cổng vào/ra cho 8031.

2.2 Bộ vi điều khiển 8052:

8052 là một phiên bản của họ 8051. 8052 có tất cả các thông số kỹ thuật của 8051, ngoài ra còn có thêm 128 byte RAM, 4KB ROM và một bộ định thời nữa. Như vậy, 8052 có tổng cộng 256 byte RAM, 8KB ROM và ba bộ định thời.

Đặc tính kỹ thuật	8031	8051	8052
ROM trên chip (KB)	0	4	8
RAM trên chip (byte)	128	128	256
Bộ định thời	2	2	3
Chân vào/ra	32	32	32
Cổng nối tiếp	1	1	1
Nguồn ngắt	5	5	6

Như bảng thông số trên ta thấy 8051 là một trường hợp riêng của 8052. Mọi chương trình viết cho 8051 đều có thể chạy được trên 8052 nhưng điều ngược lại có thể là không đúng.

2.3 Bộ vi điều khiển 8751:

Chip 8751 chỉ có 4KB bộ nhớ UV-EPROM trên chip. Để sử dụng chip này cần phải có thiết bị lập trình PROM và thiết bị xóa UV-EPROM. Do ROM trên chip của 8751 là UV-EPROM, nên cần phải mất khoảng 20 phút để xóa 8751 trước khi được lập trình. Vì đây là quá trình mất nhiều thời gian nên nhiều nhà sản xuất đã cho ra phiên bản Flash ROM và UV-RAM.

2.4 Bộ vi điều khiển AT8951 của Atmel Corporation:

AT8951 là phiên bản 8051 có ROM trên chip là bộ nhớ Flash. Phiên bản này rất thích hợp cho các ứng dụng nhanh vì bộ nhớ Flash có thể được xóa trong vài giây. Dĩ nhiên là để dùng AT8951 cần phải có thiết bị lập trình PROM hỗ trợ bộ nhớ Flash nhưng không cần đến thiết bị xóa ROM vì bộ nhớ Flash được xóa bằng thiết bị lập trình PROM. Để tiện sử dụng, hiện nay hãng Atmel đang nghiên cứu một phiên bản của AT8951 có thể được lập trình qua cổng COM của máy tính PC và như vậy sẽ không cần đến thiết bị lập trình PROM.

Ký hiệu	ROM	RAM	I/O	Timer	Ngắt	Vcc	Số chân IC
AT89C51	4KB	128	32	2	5	5V	40
AT89LV51	4KB	128	32	2	5	3V	40
AT89C1051	1KB	64	15	1	3	3V	20
AT89C2051	2KB	128	15	2	5	3V	20
AT89C52	8KB	256	32	3	6	5V	40
AT89LV52	8KB	256	32	3	6	3V	40

2.5 Bộ vi điều khiển DS5000 của Dallas Semiconductor:

Một phiên bản phổ biến khác nữa của 8051 là DS5000 của hãng Dallas Semiconductor. Bộ nhớ ROM trên chip của DS5000 là NV-RAM. DS5000 có khả năng nạp chương trình vào ROM trên chip trong khi nó vẫn ở trong hệ thống mà không cần phải lấy ra. Cách thực hiện là dùng qua cổng COM

của máy tính PC. Đây là một điểm mạnh rất được ưa chuộng. Ngoài ra, NV-RAM còn có ưu việt là cho phép thay đổi nội dung RAM theo từng byte mà không cần phải xóa hết trước khi lập trình như bộ nhớ EPROM.

Ký hiệu	ROM	RAM	I/O	Timer	Ngắt	Vcc	Số chân IC
DS5000-8	8KB	128	32	2	6	5V	40
DS5000-32	32KB	128	32	2	6	5V	40
DS5000T-8	8KB	128	32	2	6	5V	40
DS5000T-32	32KB	128	32	2	6	5V	40

Điểm đặc biệt là các chip có chữ “T” theo sau ký hiệu “5000” có nghĩa là chip đó có thiết kế thêm một đồng hồ thời gian thực (RTC: Real Time Clock) bên trong. Lưu ý đồng hồ thời gian thực RTC hoàn toàn khác với bộ định thời Timer. RTC tạo và lưu giữ thời gian của ngày (giờ/phút/giây) và ngày tháng (ngày/tháng/năm) trên thực tế ngay cả khi không có nguồn cung cấp.

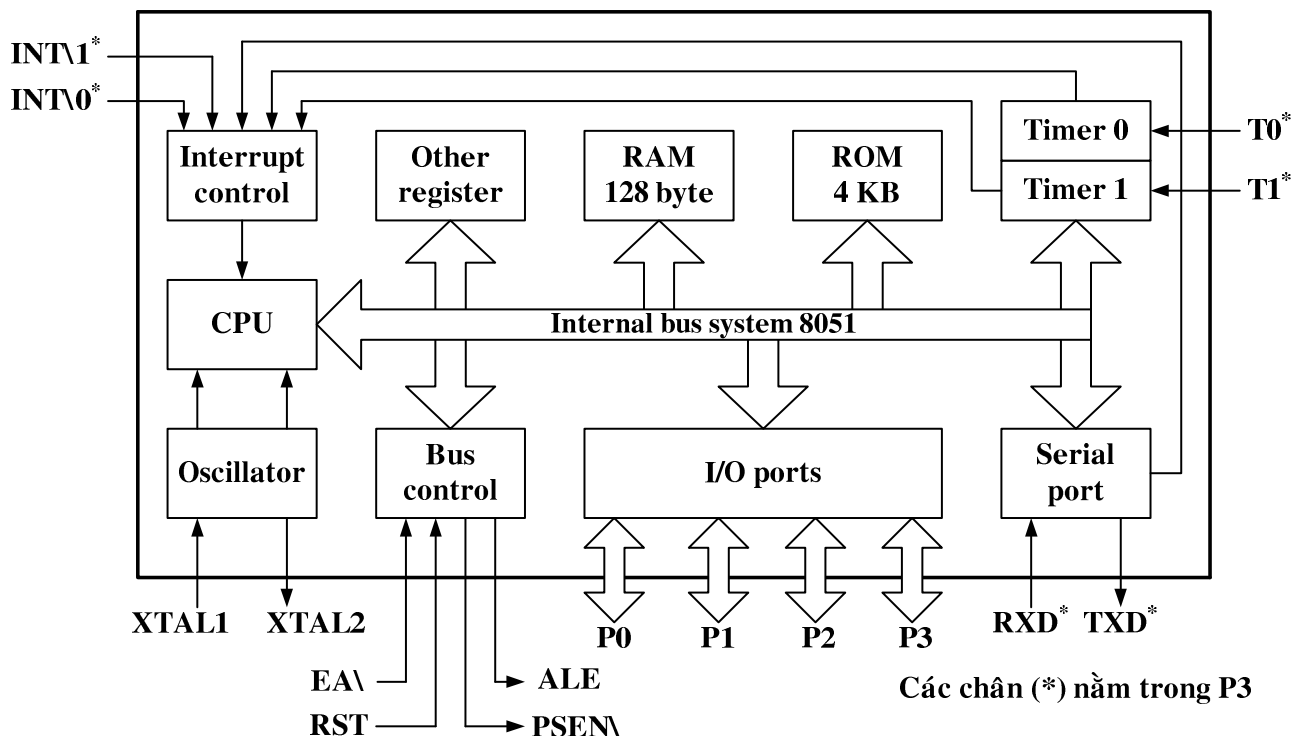
2.6 Bộ vi điều khiển P89V51xx của Philips Corporation:

Đây là một phiên bản cải tiến sử dụng CPU là bộ vi điều khiển 80C51 với nhiều tính năng vượt trội: dung lượng ROM/RAM trên chip rất lớn, 3 Timer 16 bit + 1 Watch-dog Timer, 2 thanh ghi DPTR, 8 nguồn ngắt, PWM (Pulse Width Modulator), SPI (Serial Peripheral Interface) và đặc biệt là bộ nhớ chương trình trên chip có tính năng ISP (In-System Programming) và IAP (In-Application Programming),...

II. CÁC CHÂN CỦA CHIP 8051:

1. Sơ đồ khối và chức năng các khối của chip 8051:

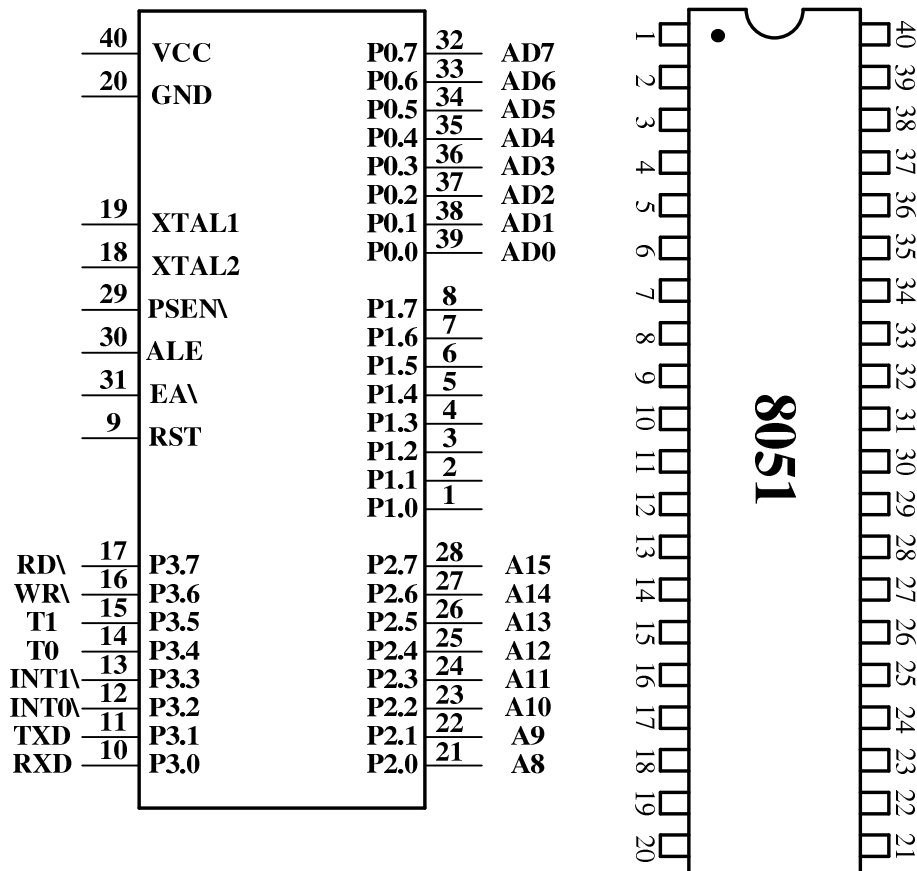
SƠ ĐỒ KHỐI CỦA CHIP 8051



- CPU (**Central Processing Unit**): Đơn vị xử lý trung tâm → tính toán và điều khiển quá trình hoạt động của hệ thống.
- OSC (**Oscillator**): Mạch dao động → tạo tín hiệu xung clock cung cấp cho các khối trong chip hoạt động.
- Interrupt control: Điều khiển ngắt → nhận tín hiệu ngắt từ bên ngoài ($INT0$, $INT1$), từ bộ định thời ($Timer\ 0$, $Timer\ 1$) và từ cổng nối tiếp ($Serial\ port$), lần lượt đưa các tín hiệu ngắt này đến CPU để xử lý.
- Other registers: Các thanh ghi khác → lưu trữ dữ liệu của các port xuất/nhập, trạng thái làm việc của các khối trong chip trong suốt quá trình hoạt động của hệ thống.
- RAM (**Random Access Memory**): Bộ nhớ dữ liệu trong chip → lưu trữ các dữ liệu.
- ROM (**Read Only Memory**): Bộ nhớ chương trình trong chip → lưu trữ chương trình hoạt động của chip.
- I/O ports (**In/Out ports**): Các port xuất/nhập → điều khiển việc xuất nhập dữ liệu dưới dạng song song giữa trong và ngoài chip thông qua các port P0, P1, P2, P3.
- Serial port: Port nối tiếp → điều khiển việc xuất nhập dữ liệu dưới dạng nối tiếp giữa trong và ngoài chip thông qua các chân TxD, RxD.
- Timer 0, Timer 1: Bộ định thời 0, 1 → dùng để định thời gian hoặc đếm sự kiện (*đếm xung*) thông qua các chân T0, T1.
- Bus control: Điều khiển bus → điều khiển hoạt động của hệ thống bus và việc di chuyển thông tin trên hệ thống bus.
- Bus system: Hệ thống bus → liên kết các khối trong chip lại với nhau.

2. Sơ đồ chân và chức năng các chân của chip 8051:

SƠ ĐỒ CHÂN CỦA CHIP 8051



2.1. Port 0:

- Port 0 ($P0.0 - P0.7$) có số chân từ 32 – 39.
- Port 0 có hai chức năng:
 - Port xuất nhập dữ liệu ($P0.0 - P0.7$) → không sử dụng bộ nhớ ngoài.
 - Bus địa chỉ byte thấp và bus dữ liệu đa hợp ($AD0 - AD7$) → có sử dụng bộ nhớ ngoài.

✓ **Lưu ý:** Khi Port 0 đóng vai trò là port xuất nhập dữ liệu thì phải sử dụng các điện trở kéo lên bên ngoài.

- Ở chế độ mặc định (khi reset) thì các chân Port 0 ($P0.0 - P0.7$) được cấu hình là **port xuất** dữ liệu. Muốn các chân Port 0 làm **port nhập** dữ liệu thì cần phải lập trình lại, bằng cách ghi mức logic cao (mức 1) đến tất cả các bit của port trước khi bắt đầu nhập dữ liệu từ port (vấn đề này được trình bày ở phần kế tiếp).

- Khi lập trình cho ROM trong chip thì Port 0 đóng vai trò là ngõ vào của dữ liệu ($D0 - D7$) (xem sách “Họ vi điều khiển 8051” trang 333-352).

2.2. Port 1:

- Port 1 ($P1.0 - P1.7$) có số chân từ 1 – 8.
- Port 1 có một chức năng:
 - Port xuất nhập dữ liệu ($P1.0 - P1.7$) → sử dụng hoặc không sử dụng bộ nhớ ngoài.

- Ở chế độ mặc định (khi reset) thì các chân Port 1 ($P1.0 - P1.7$) được cấu hình là **port xuất** dữ liệu. Muốn các chân Port 1 làm **port nhập** dữ liệu thì cần phải lập trình lại, bằng cách ghi mức logic cao (mức 1) đến tất cả các bit của port trước khi bắt đầu nhập dữ liệu từ port (vấn đề này được trình bày ở phần kế tiếp).

- Khi lập trình cho ROM trong chip thì Port 1 đóng vai trò là ngõ vào của địa chỉ byte thấp ($A0 - A7$) (xem sách “Họ vi điều khiển 8051” trang 333-352).

2.3. Port 2:

- Port 2 ($P2.0 - P2.7$) có số chân từ 21 – 28.
- Port 2 có hai chức năng:
 - Port xuất nhập dữ liệu ($P2.0 - P2.7$) → không sử dụng bộ nhớ ngoài.
 - Bus địa chỉ byte cao ($A8 - A15$) → có sử dụng bộ nhớ ngoài.

- Ở chế độ mặc định (khi reset) thì các chân Port 2 ($P2.0 - P2.7$) được cấu hình là **port xuất** dữ liệu. Muốn các chân Port 2 làm **port nhập** dữ liệu thì cần phải lập trình lại, bằng cách ghi mức logic cao (mức 1) đến tất cả các bit của port trước khi bắt đầu nhập dữ liệu từ port (vấn đề này được trình bày ở phần kế tiếp).

- Khi lập trình cho ROM trong chip thì Port 2 đóng vai trò là ngõ vào của địa chỉ byte cao ($A8 - A11$) và các tín hiệu điều khiển (xem sách “Họ vi điều khiển 8051” trang 333-352).

2.4. Port 3:

- Port 3 ($P3.0 - P3.7$) có số chân từ 10 – 17.
- Port 3 có hai chức năng:
 - Port xuất nhập dữ liệu ($P3.0 - P3.7$) → không sử dụng bộ nhớ ngoài hoặc các chức năng đặc biệt.
 - Các tín hiệu điều khiển → có sử dụng bộ nhớ ngoài hoặc các chức năng đặc biệt.

- Ở chế độ mặc định (khi reset) thì các chân Port 3 ($P3.0 - P3.7$) được cấu hình là **port xuất** dữ liệu. Muốn các chân Port 3 làm **port nhập** dữ liệu thì cần phải lập trình lại, bằng cách ghi mức logic cao (mức 1) đến tất cả các bit của port trước khi bắt đầu nhập dữ liệu từ port (vấn đề này được trình bày ở phần kế tiếp).

- Khi lập trình cho ROM trong chip thì Port 3 đóng vai trò là ngõ vào của các tín hiệu điều khiển (xem sách “Họ vi điều khiển 8051” trang 333-352).

- Chức năng của các chân Port 3:

Bit	Tên	Địa chỉ bit	Chức năng
P3.0	RxD	B0H	Chân nhận dữ liệu của port nối tiếp.
P3.1	TxD	B1H	Chân phát dữ liệu của port nối tiếp.
P3.2	INT0\	B2H	Ngõ vào ngắt ngoài 0.
P3.3	INT1\	B3H	Ngõ vào ngắt ngoài 1.
P3.4	T0	B4H	Ngõ vào của bộ định thời/đếm 0.
P3.5	T1	B5H	Ngõ vào của bộ định thời/đếm 1.
P3.6	WR\	B6H	Điều khiển ghi vào RAM ngoài.
P3.7	RD\	B7H	Điều khiển đọc từ RAM ngoài.

2.5. Chân PSEN\:

- PSEN (**Program Store Enable**): cho phép bộ nhớ chương trình, chân số 29.

- Chức năng:

- Là tín hiệu cho phép truy xuất (**đọc**) bộ nhớ chương trình (ROM) ngoài.
- Là tín hiệu xuất, tích cực mức thấp.

PSEN\ = 0 → trong thời gian CPU tìm - nạp lệnh từ ROM ngoài.

PSEN\ = 1 → CPU sử dụng ROM trong (*không sử dụng ROM ngoài*).

- Khi sử dụng bộ nhớ chương trình bên ngoài, chân PSEN\ thường được nối với chân OE\ của ROM ngoài để cho phép CPU đọc mã lệnh từ ROM ngoài.

2.6. Chân ALE:

- ALE (**Address Latch Enable**): cho phép chốt địa chỉ, chân số 30.

- Chức năng:

- Là tín hiệu cho phép chốt địa chỉ để thực hiện việc giải đa hợp cho bus địa chỉ byte thấp và bus dữ liệu đa hợp (AD0 – AD7).
- Là tín hiệu xuất, tích cực mức cao.

ALE = 0 → trong thời gian bus AD0 - AD7 đóng vai trò là bus D0 - D7.

ALE = 1 → trong thời gian bus AD0 - AD7 đóng vai trò là bus A0 - A7.

- Khi lập trình cho ROM trong chip thì chân ALE đóng vai trò là ngõ vào của xung lập trình (PGM\) (xem sách “Họ vi điều khiển 8051” trang 333-352).

✓ **Lưu ý:** $f_{ALE} = \frac{f_{OSC}}{6}$ → có thể dùng làm xung clock cho các mạch khác.

f_{ALE} (MHz): tần số xung tại chân ALE.

f_{OSC} (MHz): tần số dao động trên chip (tần số thạch anh).

- Khi lệnh lấy dữ liệu từ RAM ngoài (MOVX) được thực hiện thì một xung ALE bị bỏ qua (xem giản đồ trang 38-39 sách “Họ vi điều khiển 8051”).

2.7. Chân EA\:

- EA (**External Access**): truy xuất ngoài, chân số 31.

- Chức năng:

- Là tín hiệu cho phép truy xuất (**sử dụng**) bộ nhớ chương trình (ROM) ngoài.

- Là tín hiệu nhập, tích cực mức thấp.

EA\ = 0 → Chip 8051 sử dụng chương trình của ROM ngoài.

EA\ = 1 → Chip 8051 sử dụng chương trình của ROM trong.

- Khi lập trình cho ROM trong chip thì chân EA đóng vai trò là ngõ vào của điện áp lập trình ($V_{pp} = 12V - 12,5V$ cho họ 89xx; $21V$ cho họ 80xx, 87xx) (xem sách “Họ vi điều khiển 8051” trang 333-352).

✓ **Lưu ý:** Chân EA\ phải được nối lên Vcc (nếu sử dụng chương trình của ROM trong) hoặc nối xuống GND (nếu sử dụng chương trình của ROM ngoài), không bao giờ được phép bỏ trống chân này.

2.8. Chân XTAL1, XTAL2:

- XTAL (Crystal): tinh thể thạch anh, chân số 18-19.

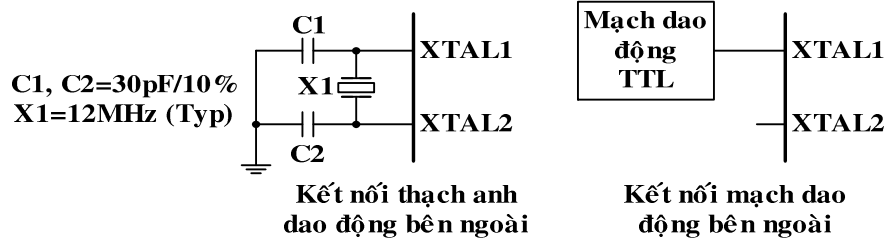
- Chức năng:

- Dùng để nối với thạch anh hoặc mạch dao động tạo xung clock bên ngoài, cung cấp tín hiệu xung clock cho chip hoạt động.
- XTAL1 → ngõ vào mạch tạo xung clock trong chip.
- XTAL2 → ngõ ra mạch tạo xung clock trong chip.

✓ **Lưu ý:**

$$f_{TYP} = 12MHz$$

f_{TYP} (MHz): tần số danh định.



2.9. Chân RST:

- RST (Reset): thiết lập lại, chân số 9.

- Chức năng:

- Là tín hiệu cho phép thiết lập (đặt) lại trạng thái ban đầu cho hệ thống.
- Là tín hiệu nhập, tích cực mức cao.

RST = 0 → Chip 8051 hoạt động bình thường.

RST = 1 → Chip 8051 được thiết lập lại trạng thái ban đầu.

✓ **Lưu ý:**

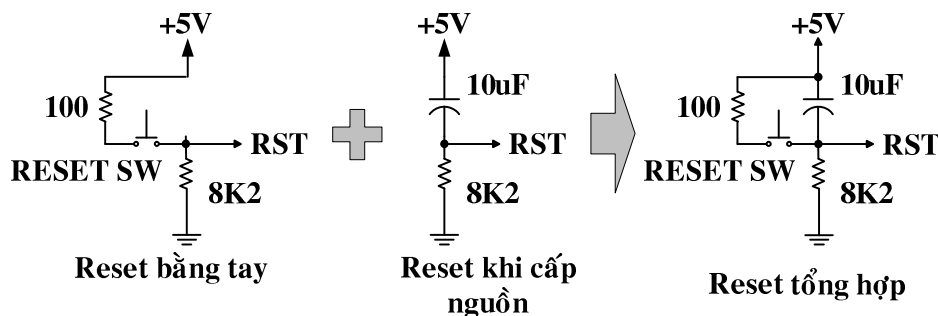
$$t_{Reset} \geq 2 \times T_{Machine}$$

$$T_{Machine} = \frac{12}{f_{OSC}}$$

t_{RESET} (μs): thời gian reset.

f_{OSC} (MHz): tần số thạch anh.

$T_{MACHINE}$ (μs): chu kỳ máy.



Ví dụ: Xác định chu kỳ máy và thời gian reset tương ứng cho từng trường hợp $f_{OSC} = 11,0592\text{MHz}$, $f_{OSC} = 12\text{MHz}$ và $f_{OSC} = 16\text{MHz}$.

Giải

- $f_{OSC} = 11,0592\text{MHz} \rightarrow T_{\text{MACHINE}} = 1,085\mu\text{s}$ và $t_{\text{RESET}} \geq 2,17\mu\text{s}$.
- $f_{OSC} = 12\text{MHz} \rightarrow T_{\text{MACHINE}} = 1\mu\text{s}$ và $t_{\text{RESET}} \geq 2\mu\text{s}$.
- $f_{OSC} = 16\text{MHz} \rightarrow T_{\text{MACHINE}} = 0,75\mu\text{s}$ và $t_{\text{RESET}} \geq 1,5\mu\text{s}$

2.10. Chân Vcc, GND:

- Vcc, GND: nguồn cấp điện, chân số 40 và 20.
- Chức năng:
 - Cung cấp nguồn điện cho chip 8051 hoạt động.
 - $V_{CC} = +5V \pm 10\%$ và $GND = 0V$.

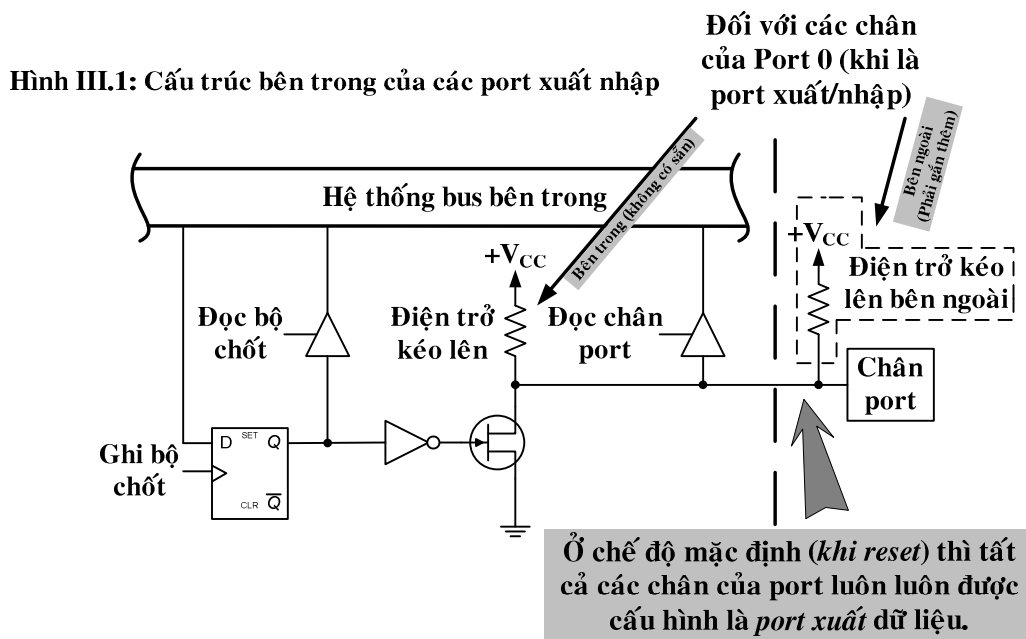
III. CẤU TRÚC CÁC PORT XUẤT NHẬP CHIP 8051:

Khả năng fanout (số lượng tải đầu ra) của các từng chân port chip 8051 là:

- Port 0: 8 tải TTL.
- Port 1: 4 tải TTL.
- Port 2: 4 tải TTL.
- Port 3: 4 tải TTL.

✓ **Lưu ý:**

• Khi **Port 0 đóng vai trò là port xuất nhập** thì sẽ không có điện trở kéo lên bên trong $\rightarrow$ do đó người sử dụng cần thêm vào điện trở kéo lên bên ngoài (xem Hình III.1).

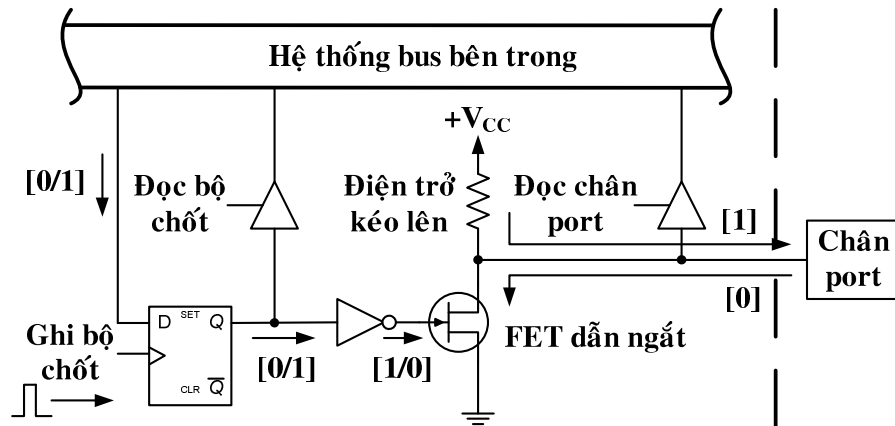


• Ở chế độ mặc định (khi reset) thì tất cả các chân của các port ($P0 - P3$) được cấu hình là **port xuất** dữ liệu.

• Muốn các chân port của chip 8051 làm **port nhập** dữ liệu thì ta cần phải được lập trình lại, bằng cách ghi mức logic cao (mức 1) đến tất cả các bit (các chân) của port trước khi bắt đầu nhập dữ liệu từ port (vấn đề này được trình bày ở phần kế tiếp).

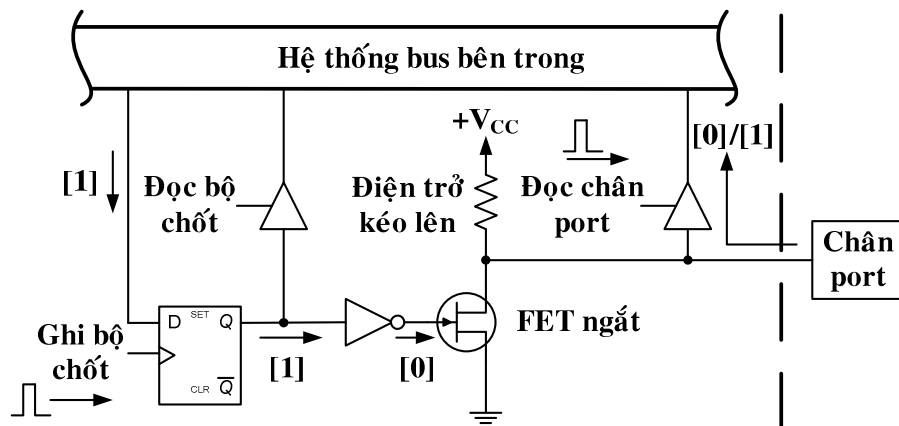
- Các chân trong cùng một port không nhất thiết phải có cùng kiểu cấu hình (*port xuất hoặc port nhập*). Nghĩa là trong cùng một port có thể có chân dùng để nhập dữ liệu, có thể có chân dùng để xuất dữ liệu. Điều này là tùy thuộc vào nhu cầu và mục đích của người lập trình.

Quá trình ghi chân port (*xuất dữ liệu ra chân port*).



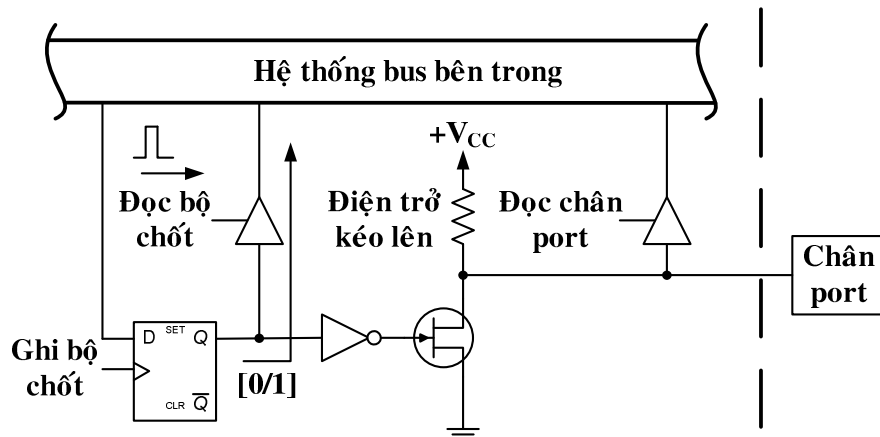
Hình III.3: Thao tác ghi chân port

Quá trình đọc chân port (*nhập dữ liệu từ chân port*).



Hình III.2: Thao tác đọc chân port

Quá trình đọc bộ chốt (*kiểm tra dữ liệu tại chân port*).



Hình III.4: Thao tác đọc bộ chốt

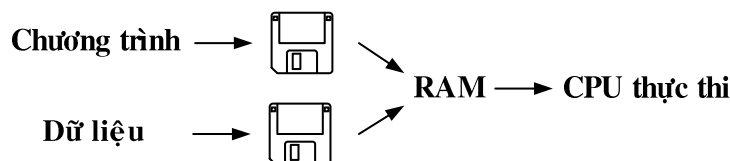
✓ **Lưu ý:** Việc đọc dữ liệu của bất kỳ một port nào có thể cho ta hai giá trị khác nhau tùy thuộc vào lệnh mà ta sử dụng để đọc dữ liệu từ port (*xem thêm trong phần tập lệnh*). Xảy ra hiện tượng không mong muốn này là do quá trình đọc dữ liệu của chip 8051 gồm hai quá trình khác nhau: *quá trình đọc chân port* và *quá trình đọc bộ chốt*.

○ **Quá trình đọc chân port:** Khi ta sử dụng các lệnh **MOV, ADD,...** Dữ liệu nhận được sau khi thực hiện quá trình đọc là **dữ liệu hiện tại ở các chân port**.

○ **Quá trình đọc bộ chốt:** Khi ta sử dụng các lệnh **ANL, ORL, XRL, CPL, INC, DEC, DJNZ, JBC, CLR bit, SETB bit, MOV bit**. Dữ liệu nhận được sau khi thực hiện quá trình đọc là **dữ liệu hiện tại ở các bộ chốt** (là các dữ liệu đã được ghi ra port tại thời điểm trước đó bởi quá trình ghi chân port), chứ không phải là dữ liệu hiện tại ở các chân port. Cho nên, nếu tại thời điểm thực hiện quá trình đọc mà dữ liệu tại các chân port có bị thay đổi đi chăng nữa thì dữ liệu đọc về cũng không được cập nhật.

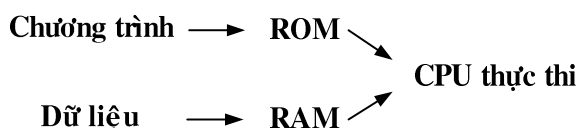
IV. TỔ CHỨC BỘ NHỚ CỦA CHIP 8051:

- Bộ vi xử lý → có không gian bộ nhớ chung cho dữ liệu và chương trình.



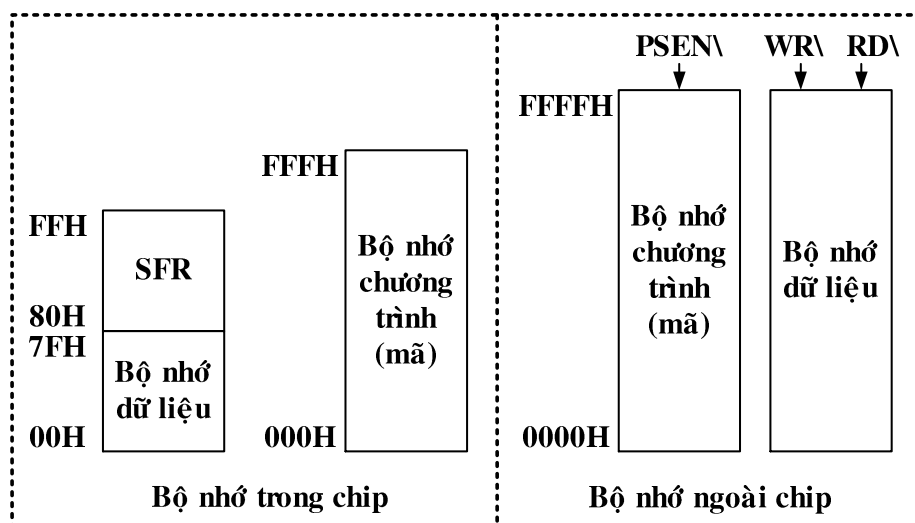
→ chương trình và dữ liệu nằm chung trên RAM trước khi đưa vào CPU để thực thi.

- Bộ vi điều khiển → có không gian bộ nhớ riêng cho dữ liệu và chương trình.

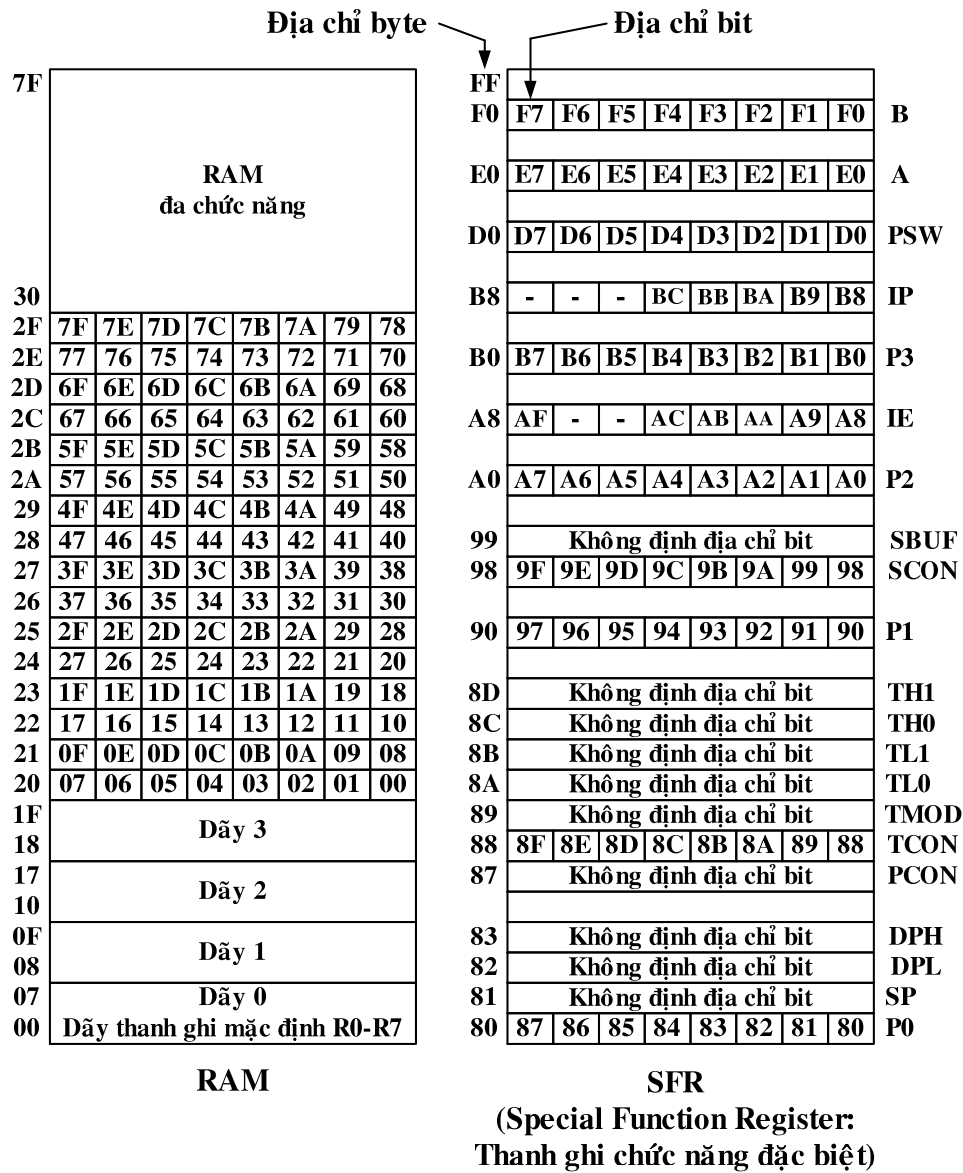


→ chương trình và dữ liệu nằm riêng trên ROM và RAM trước khi đưa vào CPU để thực thi.

- Tổ chức bộ nhớ của chip 8051:

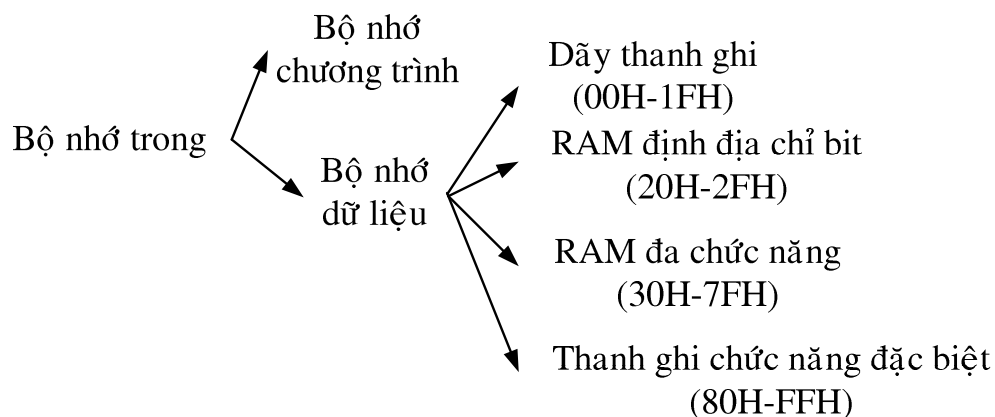


Không gian bộ nhớ của chip 8051



Bộ nhớ dữ liệu trên chip 8051

1. Bộ nhớ trong:



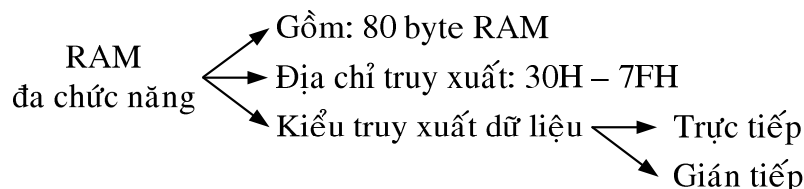
1.1. Bộ nhớ chương trình (ROM):

- Dùng để lưu trữ chương trình điều khiển cho chip 8051 hoạt động.
- Chip 8051 có 4 KB ROM trong, địa chỉ truy xuất: 000H – FFFH.

1.2. Bộ nhớ dữ liệu (RAM):

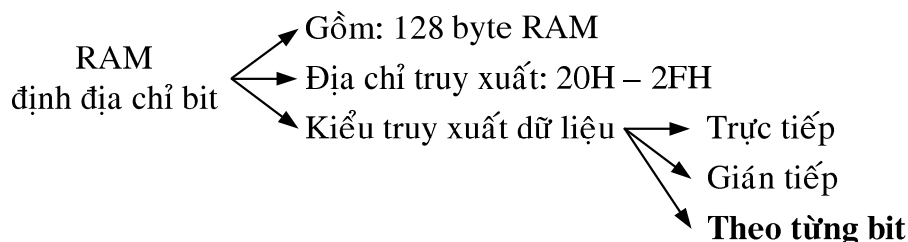
- Dùng để lưu trữ các dữ liệu và tham số.
- Chip 8051 có 128 byte RAM trong, địa chỉ truy xuất: 00H – 7FH.
- RAM trong của chip 8051 được chia ra:

- RAM đa chức năng:



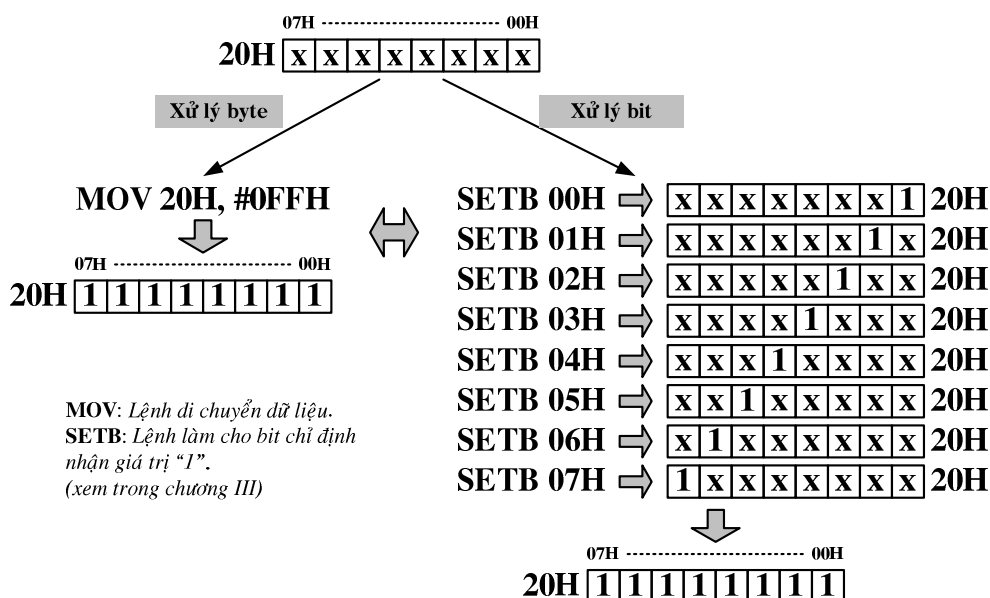
- RAM định địa chỉ bit:

→ cho phép xử lý từng bit dữ liệu riêng lẻ mà không ảnh hưởng đến các bit khác trong cả byte.



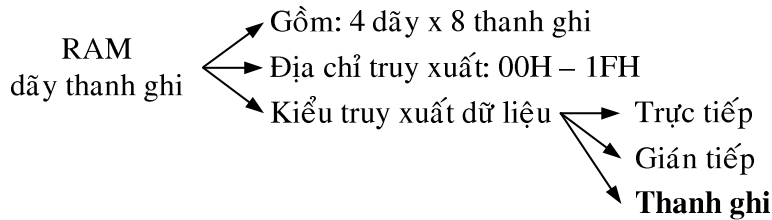
✓ **Lưu ý:** Nếu trong chương trình không sử dụng các bit trong vùng RAM định địa chỉ bit này, ta có thể sử dụng vùng nhớ 20H – 2FH cho các mục đích khác của ta. Ngược lại, ta phải viết chương trình cẩn thận khi sử dụng vùng nhớ 20H – 2FH vì nếu sơ suất ta có thể ghi dữ liệu đè lên các bit đã được sử dụng.

✓ **Ví dụ:** Viết lệnh làm cho 8 bit trong ô nhớ có địa chỉ 20H thuộc RAM nội có giá trị là 1 (xét trường hợp địa chỉ byte và địa chỉ bit).



- Các dây thanh ghi:

→ cho phép truy xuất dữ liệu nhanh, lệnh truy xuất đơn giản và ngắn gọn.



Bảng số liệu dưới đây minh họa địa chỉ của các ô nhớ trong một dây và các ký hiệu thanh ghi R0 – R7 được gán cho từng ô nhớ trong dây tích cực.

	Dây 0	Dây 1	Dây 2	Dây 3
R0	00H	08H	10H	18H
R1	01H	09H	11H	19H
R2	02H	0AH	12H	1AH
R3	03H	0BH	13H	1BH
R4	04H	0CH	14H	1CH
R5	05H	0DH	15H	1DH
R6	06H	0EH	16H	1EH
R7	07H	0FH	17H	1FH

Địa chỉ của các thanh ghi (R0 – R7) tương ứng với dây thanh ghi tích cực.

✓ Lưu ý:

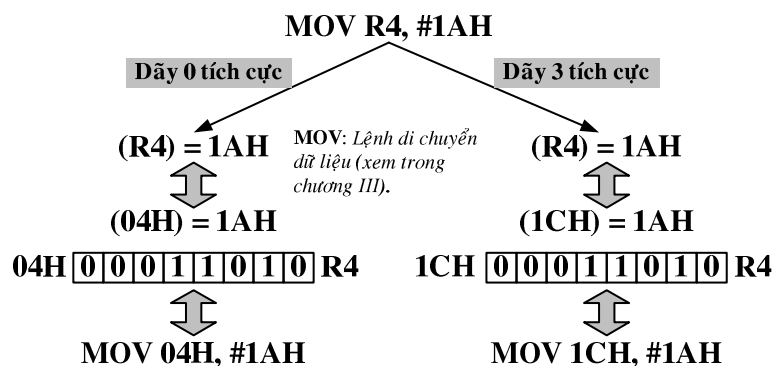
○ Ở chế độ mặc định thì dây thanh ghi tích cực (đang được sử dụng) là **dây 0** và các thanh ghi trong dây lần lượt có tên là **R0 - R7**. Có thể thay đổi dây tích cực bằng cách thay đổi các bit chọn dây thanh ghi RS1 và RS0 trong thanh ghi PSW (xem phần thanh ghi PSW).

○ Nếu chương trình của ta chỉ sử dụng dây thanh ghi đầu tiên (dây 0) thì ta có thể sử dụng vùng nhớ 08H – 1FH cho các mục đích khác của ta. Nhưng nếu trong chương trình có sử dụng các dây thanh ghi (dây 1, 2 hoặc 3) thì phải rất cẩn thận khi sử dụng vùng nhớ từ 1FH trở xuống vì nếu sơ suất ta có thể ghi dữ liệu đè lên các thanh ghi R0 – R7 của ta.

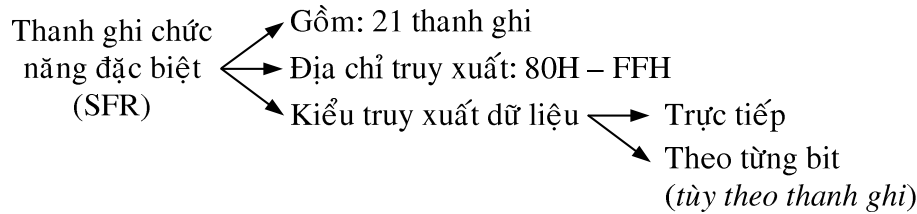
✓ Ví dụ 1: Quan hệ giữa ký hiệu thanh ghi R4 với các ô nhớ có địa chỉ tương ứng trong dây thanh ghi tích cực?

- Nếu dây 0 tích cực: Thanh ghi **R4** ⇔ Ô nhớ **04H** RAM nội.
- Nếu dây 1 tích cực: Thanh ghi **R4** ⇔ Ô nhớ **0CH** RAM nội.
- Nếu dây 2 tích cực: Thanh ghi **R4** ⇔ Ô nhớ **14H** RAM nội.
- Nếu dây 3 tích cực: Thanh ghi **R4** ⇔ Ô nhớ **1CH** RAM nội.

✓ Ví dụ 2: Khi chip 8051 thực hiện lệnh **MOV R4, #1AH** thì giá trị “1AH” sẽ được nạp vào trong ô nhớ có địa chỉ là bao nhiêu thuộc RAM nội. Xét tương ứng cho từng trường hợp dây thanh ghi tích cực là Dây 0 và Dây 3?



1.3. Thanh ghi chức năng đặc biệt (SFR):



✓ Lưu ý:

○ Không được phép đọc hay ghi dữ liệu vào các địa chỉ SFR mà nó chưa được đăng ký (nghĩa là các địa chỉ SFR chưa được đặt tên). Vì việc đọc hay ghi dữ liệu vào các nơi này có thể làm phát sinh những hoạt động không mong muốn và đó có thể là nguyên nhân làm cho chương trình của ta không tương thích với các phiên bản sau của chip MCS-51 (có thể ở các phiên bản đó các địa chỉ SFR này được sử dụng cho một vài mục đích khác).

○ Chỉ được truy xuất các SFR bằng kiểu định địa chỉ trực tiếp (tuyệt đối không sử dụng kiểu định địa chỉ gián tiếp trong trường hợp này).

✓ Ví dụ: Cho biết trước (R0)=90H. Viết lệnh dùng để xuất (ghi) giá trị 5AH ra Port1 như sau (xem giải thích lệnh trong “Chương 3: Tập lệnh của 8051.”):

- Sử dụng kiểu định địa chỉ trực tiếp:

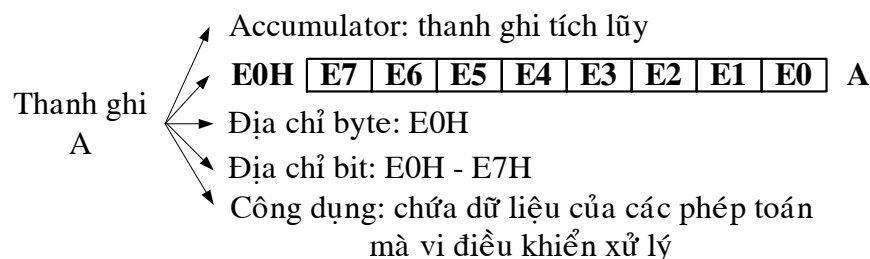
MOV P1, #5AH hoặc **MOV 90H, #5AH**

- Sử dụng kiểu định địa chỉ gián tiếp:

MOV @R0, #5AH ⇒ **SAI**

⇒ Điều này không hợp lệ đối với chip 8051 vì phương pháp định địa chỉ gián tiếp như trên chỉ sử dụng cho vùng nhớ RAM nội. Trong khi đó RAM nội của chip 8051 chỉ có 128 byte (00H – 7FH), cho nên khi thực hiện lệnh này nó sẽ trả về kết quả không xác định. (Lưu ý: nếu ta dùng phiên bản chip 8052 thì sẽ tránh được điều này).

1.3.1. Thanh ghi A:



- Thương số → chứa vào thanh ghi A.
- Số dư → chứa vào thanh ghi B.

✓ Ví dụ: Thực hiện phép tính $12H \times 2AH$. Hỏi (A)=?, (B)=?

$$\begin{array}{r}
 \begin{array}{c} 1 \ 2 \ H \\ \times 2 \ A \ H \\ \hline B \ 4 \\ 2 \ 4 \\ \hline 0 \ 2 \ F \ 4 \ H \end{array} \\
 \begin{array}{c} \xrightarrow{(B)} \quad \quad \quad \xrightarrow{(A)} \end{array}
 \end{array}
 \Rightarrow \begin{array}{l} (A) = F4H \\ (B) = 02H \end{array}$$

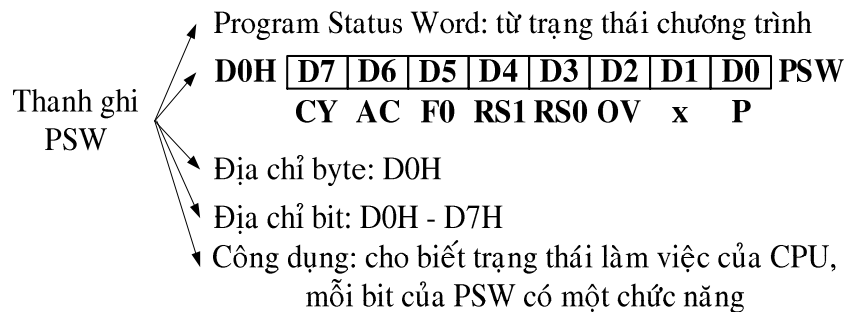
✓ Ví dụ: Thực hiện phép tính $A6H : 21H$. Hỏi (A)=?, (B)=?

$$\begin{array}{r}
 \begin{array}{c} A \ 6 \ H \\ A \ 5 \\ \hline 0 \ 1 \ H \end{array} \Bigg| \begin{array}{c} 2 \ 1 \ H \\ 0 \ 5 \ H \end{array} \\
 \begin{array}{c} \xrightarrow{(B)} \quad \quad \quad \xrightarrow{(A)} \end{array}
 \end{array}
 \Rightarrow \begin{array}{l} (A) = 05H \\ (B) = 01H \end{array}$$

✓ Ví dụ: Thực hiện phép tính $FDH : 0CH$. Hỏi (A)=?, (B)=?

$$\begin{array}{r}
 \begin{array}{c} F \ D \ H \\ C \ \downarrow \\ 3 \ D \\ 3 \ C \\ \hline 0 \ 1 \ H \end{array} \Bigg| \begin{array}{c} 0 \ C \ H \\ 1 \ 5 \ H \end{array} \\
 \begin{array}{c} \xrightarrow{(B)} \quad \quad \quad \xrightarrow{(A)} \end{array}
 \end{array}
 \Rightarrow \begin{array}{l} (A) = 15H \\ (B) = 01H \end{array}$$

1.3.3. Thanh ghi từ PSW:



- Cờ CY (*Carry Flag*): cờ nhớ → báo có nhớ/mượn tại bit 7.
 - CY = 0: nếu không có nhớ từ bit 7 hoặc không có mượn cho bit 7.
 - CY = 1: nếu có nhớ từ bit 7 hoặc có mượn cho bit 7.
- Cờ AC (*Auxiliary Carry*): cờ nhớ phụ → báo có nhớ/mượn tại bit 3.
 - AC = 0: nếu không có nhớ từ bit 3 hoặc không có mượn cho bit 3.
 - AC = 1: nếu có nhớ từ bit 3 hoặc có mượn cho bit 3.
- Cờ F0 (*Flag 0*): cờ zero → có nhiều mục đích dành cho các ứng dụng khác nhau của người lập trình (*dự trữ cho các phiên bản chip trong tương lai*).

- Bit RS0, RS1 (*Register Select*): bit chọn dãy thanh ghi → cho phép xác định dãy thanh ghi tích cực (*hay dãy thanh ghi mà các thanh ghi có tên là R0-R7*).

RS1	RS0	Dãy thanh ghi	R0 → R7
0	0	Dãy 0	00H → 07H
0	1	Dãy 1	08H → 0FH
1	0	Dãy 2	10H → 17H
1	1	Dãy 3	18H → 1FH

- Cờ OV (*Overflow*): cờ tràn → báo kết quả tính toán của phép toán số học (*phép toán có dấu*) có nằm trong khoảng từ -128 đến +127 hay không.

- OV = 0: nếu $-128 \leq \text{kết quả} \leq +127$.
- OV = 1: nếu kết quả < -128 hoặc kết quả $> +127$. Nói cách khác là: Đối với *phép cộng* thì OV=1 nếu có nhớ từ bit 7 nhưng không có nhớ từ bit 6 hoặc nếu có nhớ từ bit 6 nhưng không có nhớ từ bit 7. Đối với *phép trừ* thì OV=1 nếu có mượn cho bit 7 nhưng không có mượn cho bit 6 hoặc nếu có mượn bit 6 nhưng không có mượn bit 7.

DEC	BIN	HEX
-128	1000 0000	80
-127	1000 0001	81
-126	1000 0010	82
.....		
-2	1111 1110	FE
-1	1111 1111	FF
0	0000 0000	00
1	0000 0001	01
2	0000 0010	02
.....		
+125	0111 1101	7D
+126	0111 1110	7E
+127	0111 1111	7F

Biểu diễn toán hạng 8 bit có dấu.

Bảng liệt kê các số 8 bit có dấu.

- Cờ P (*Parity*): cờ chẵn lẻ → báo số chữ số 1 trong thanh ghi A là số chẵn hay số lẻ (*trong chip 8051 sử dụng chế độ parity chẵn*).

- P = 0: nếu số chữ số 1 trong thanh ghi A là số chẵn (*parity chẵn*).
- P = 1: nếu số chữ số 1 trong thanh ghi A là số lẻ (*parity chẵn*).

✓ Ví dụ: Minh họa cách 8051 biểu diễn số -5.

Giải

Các bước thực hiện:

B1: 0000 0101

Biểu diễn số 5 dạng nhị phân 8 bit.

B2: 1111 1010

Lấy bù 1.

B3: 1111 1011

Lấy bù 2.

Vậy số FBH là biểu diễn số có dấu dạng bù 2 của số -5.

✓ Ví dụ: Minh họa cách 8051 biểu diễn số -34H.

Các bước thực hiện:

B1: 0011 0100

Biểu diễn số 34H dạng nhị phân 8 bit.

B2: 1100 1011

Lấy bù 1.

B3: 1100 1100

Lấy bù 2.

Vậy số CCH là biểu diễn số có dấu dạng bù 2 của số -34H.

✓ Ví dụ: Minh họa cách 8051 biểu diễn số -128.

Các bước thực hiện:

B1: 1000 0000 Biểu diễn số -128 dạng nhị phân 8 bit.

B2: 0111 1111 Lấy bù 1.

B3: 1000 0000 Lấy bù 2.

Vậy số 80H là biểu diễn số có dấu dạng bù 2 của số -128.

✓ Ví dụ: Minh họa trạng thái hoạt động của các cờ CY, AC, OV và P khi thực hiện phép cộng/trừ số học hai giá trị với nhau.

Các ký hiệu:

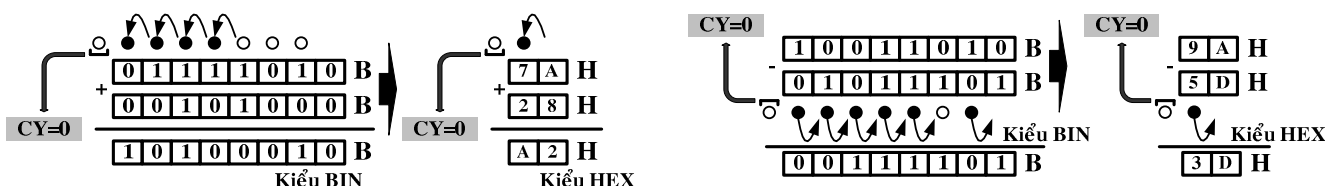
- Có nhớ / có mượn.
- Không nhớ / không mượn.
- ? Tùy thuộc trạng thái trước đó.

■ Cờ nhớ (CY):

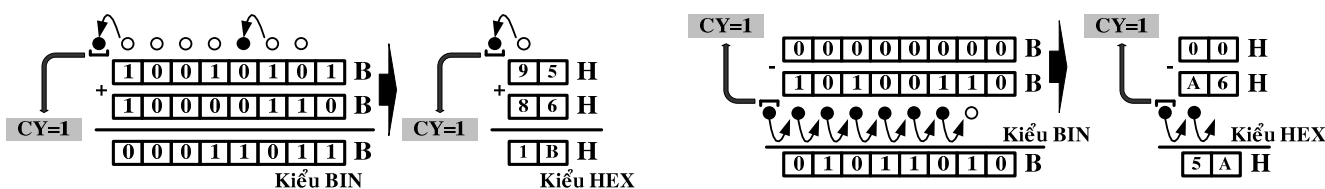
Minh họa hoạt động của cờ CY trong trường hợp CY = 1:



Xét cờ CY trong hai trường hợp “7AH+28H” và “9AH-5DH”:

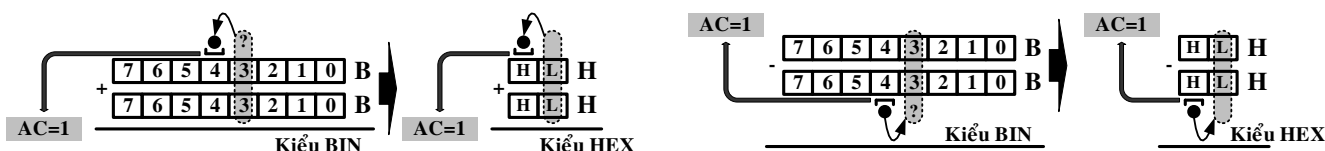


Xét cờ CY trong hai trường hợp “95H+86H” và “00H-A6H”:

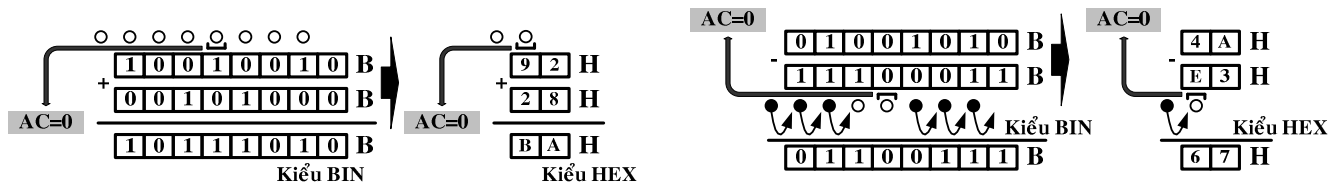


■ Cờ nhớ phụ (AC):

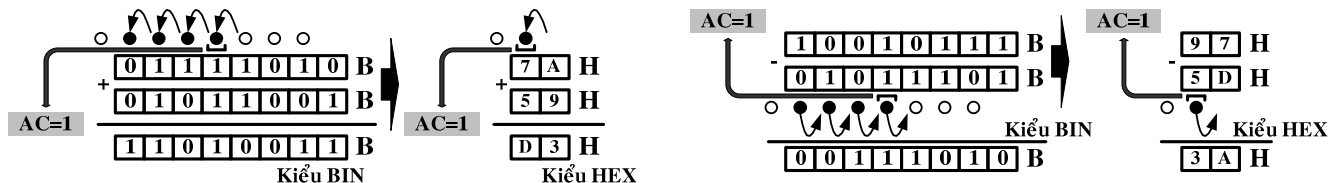
Minh họa hoạt động của cờ AC trong trường hợp AC = 1



Xét cờ AC trong hai trường hợp “92H+28H” và “4AH-E3H”:

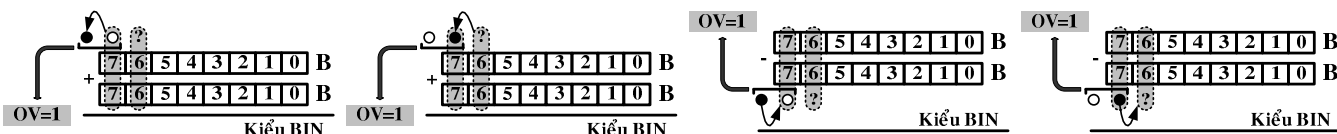


Xét cờ AC trong hai trường hợp “7AH+59H” và “97H-5DH”:

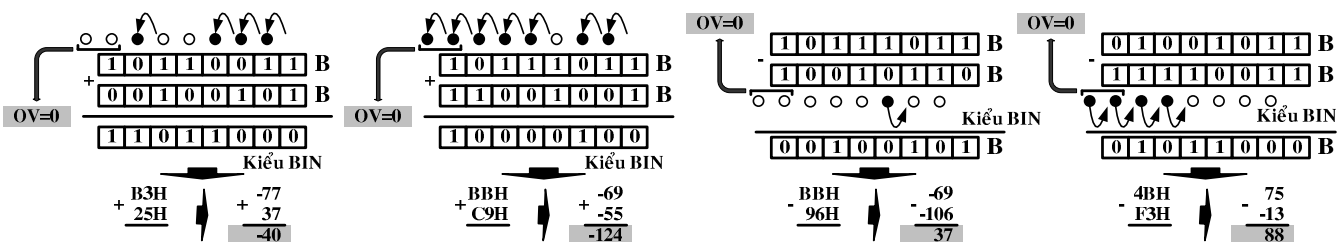


■ Cờ tràn (OV):

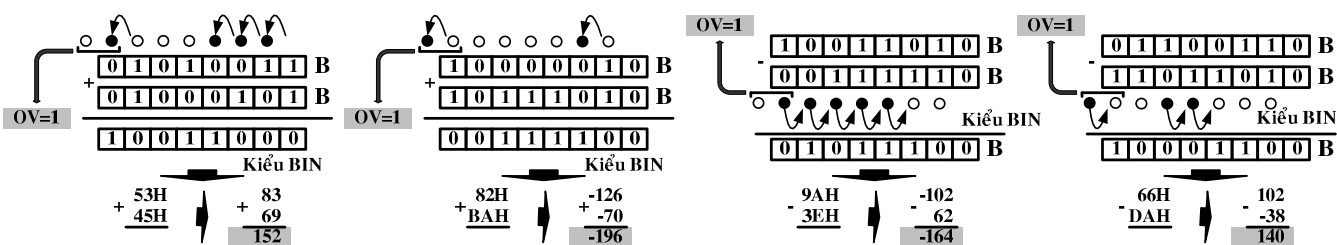
Minh họa hoạt động của cờ OV trong trường hợp OV = 1



Xét cờ OV trong các trường hợp “B3H+25H”, “BBH+C9H”, “BBH-96H” và “4BH-F3H”:

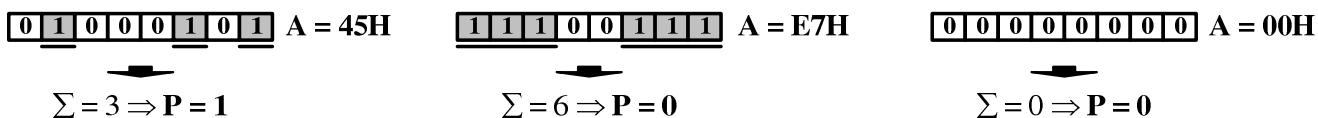


Xét cờ OV trong các trường hợp “53H+45H”, “82H+BAH”, “9AH-3EH” và “66H-DAH”:



■ Cờ Parity (P):

Xét cờ P trong các trường hợp “(A)=45H”, “(A)=E7H”, “(A)=00H”:



✓ Ví dụ: Xác định nội dung các ô nhớ thuộc RAM nội của đoạn chương trình sau:

SETB PSW.3 ;Chọn dây thanh ghi 1, RS0=1
CLR PSW.4 ;Chọn dây thanh ghi 1, RS1=0
MOV R0, #10H ;Nạp thanh ghi R0 giá trị 10H
MOV R1, #2AH ;Nạp thanh ghi R1 giá trị 2AH
MOV R3, #3FH ;Nạp thanh ghi R3 giá trị 3FH
MOV R5, #57H ;Nạp thanh ghi R5 giá trị 57H
MOV R7, #0A9H ;Nạp thanh ghi R7 giá trị A9H

MOV: Lệnh di chuyển dữ liệu.

SETB: Lệnh làm cho bit chỉ định nhận giá trị "1".

CLR: Lệnh làm cho bit chỉ định nhận giá trị "0".
 (xem trong chương III)

Vì dây tích cực là **Dây 1** nên ta có:

Ô nhớ có địa chỉ 08H nhận giá trị 10H.
 Ô nhớ có địa chỉ 09H nhận giá trị 2AH.
 Ô nhớ có địa chỉ 0BH nhận giá trị 3FH.
 Ô nhớ có địa chỉ 0DH nhận giá trị 57H.
 Ô nhớ có địa chỉ 0FH nhận giá trị A9H.

✓ Ví dụ: Xác định nội dung các ô nhớ thuộc RAM nội của đoạn chương trình sau:

CLR PSW.3 ;Chọn dây thanh ghi 2, RS0=0
SETB PSW.4 ;Chọn dây thanh ghi 2, RS1=1
MOV R0, #10H ;Nạp thanh ghi R0 giá trị 10H
MOV R1, #2AH ;Nạp thanh ghi R1 giá trị 2AH
MOV R3, #3FH ;Nạp thanh ghi R3 giá trị 3FH
MOV R5, #57H ;Nạp thanh ghi R5 giá trị 57H
MOV R7, #0A9H ;Nạp thanh ghi R7 giá trị A9H

MOV: Lệnh di chuyển dữ liệu.

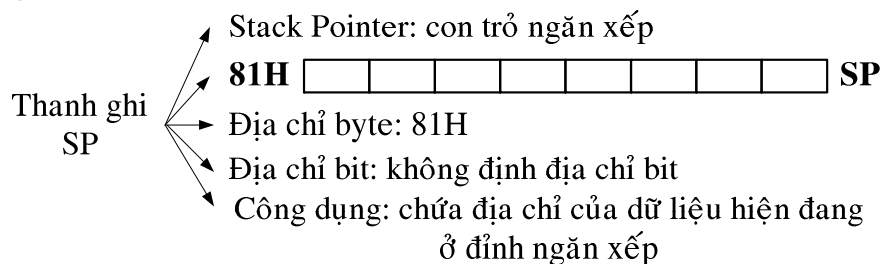
SETB: Lệnh làm cho bit chỉ định nhận giá trị "1".

CLR: Lệnh làm cho bit chỉ định nhận giá trị "0".
 (xem trong chương III)

Vì dây tích cực là **Dây 2** nên ta có:

Ô nhớ có địa chỉ 10H nhận giá trị 10H.
 Ô nhớ có địa chỉ 11H nhận giá trị 2AH.
 Ô nhớ có địa chỉ 13H nhận giá trị 3FH.
 Ô nhớ có địa chỉ 15H nhận giá trị 57H.
 Ô nhớ có địa chỉ 17H nhận giá trị A9H.

1.3.4. Thanh ghi SP:



- Ngăn xếp là vùng nhớ dùng để lưu trữ tạm thời các dữ liệu.
- Đối với chip 8051 thì vùng nhớ được dùng để làm ngăn xếp được giữ trong RAM nội.
- Để sử dụng ngăn xếp thì ta phải khởi động thanh ghi SP (*nghĩa là nạp giá trị cho thanh ghi SP*) → vùng nhớ của ngăn xếp có **địa chỉ bắt đầu: (SP)+1** và **địa chỉ kết thúc: 7FH**.
- Nếu không khởi động SP → vùng nhớ của ngăn xếp có **địa chỉ bắt đầu: 08H** và **địa chỉ kết thúc: 7FH** (chế độ mặc định).

✓ **Lưu ý:** Trong trường hợp không khởi động SP (*chế độ mặc định*) thì dây thanh ghi 1 (và có thể là dây 2 và dây 3) sẽ không còn hợp lệ vì khi đó vùng nhớ này đã được sử dụng để làm ngăn xếp. Điều này có nghĩa là nếu ta sử dụng các dây thanh ghi này và lưu trữ dữ liệu vào đó thì có khả năng sẽ bị mất do tác động cất dữ liệu vào ngăn xếp của các lệnh (*PUSH, ACALL, LCALL, ...*).

✓ Ví dụ: Hãy cho biết tầm địa chỉ của vùng nhớ ngăn xếp trong hai trường hợp sau: (SP)=5FH và (SP)=49H.

Theo qui định thì vùng nhớ của ngăn xếp có địa chỉ bắt đầu: (SP)+1 và địa chỉ kết thúc: 7FH.

- Trường hợp (SP)=5FH: tầm địa chỉ của vùng nhớ ngăn xếp là 60H - 7FH.
- Trường hợp (SP)=49H: tầm địa chỉ của vùng nhớ ngăn xếp là 4AH - 7FH.

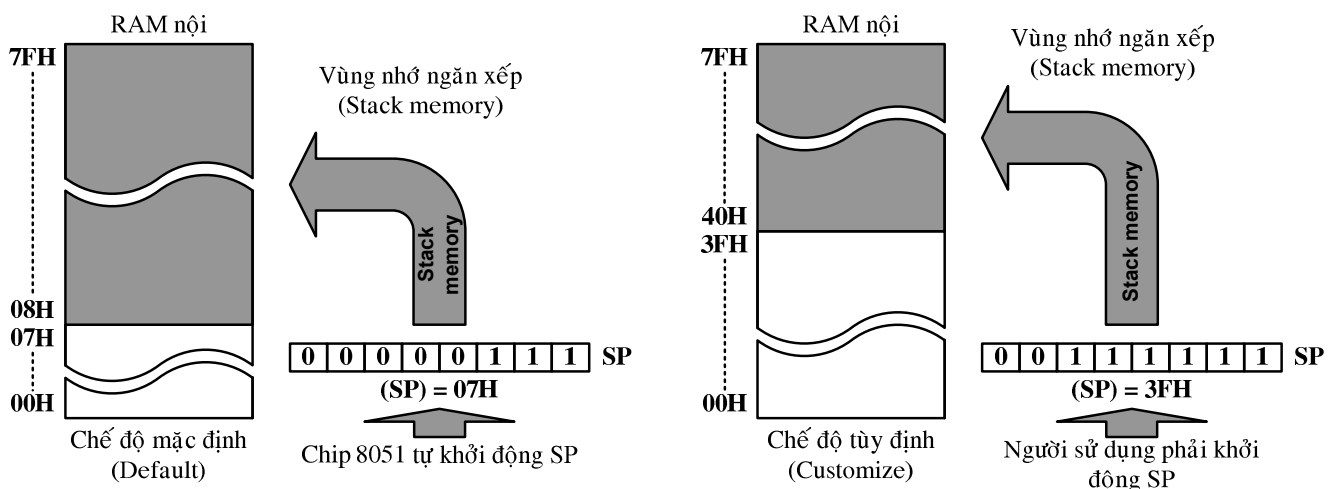
✓ Ví dụ: Hãy cho biết giá trị cần phải nạp cho thanh ghi SP để vùng nhớ ngăn xếp có tầm địa chỉ trong hai trường hợp sau: 62H - 7FH và 50H - 7FH.

Theo qui định thì vùng nhớ của ngăn xếp có địa chỉ bắt đầu: (SP)+1 và địa chỉ kết thúc: 7FH.

- Trường hợp 62H - 7FH: giá trị cần nạp cho thanh ghi SP là 61H.
- Trường hợp 50H - 7FH: giá trị cần nạp cho thanh ghi SP là 4FH.

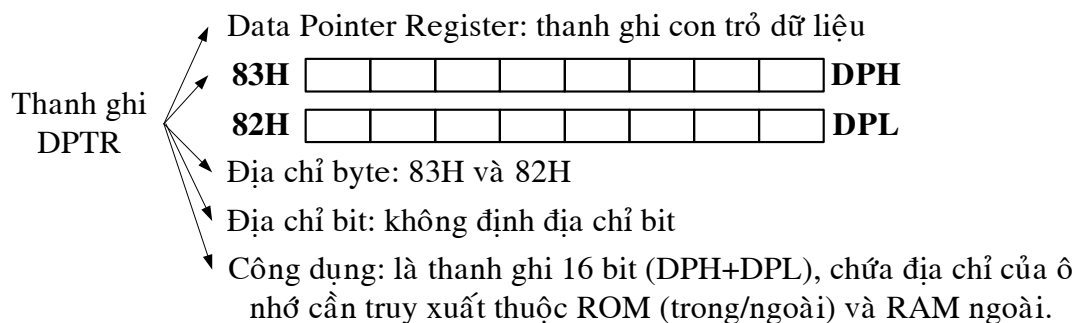
✓ Ví dụ: Minh họa vùng nhớ ngăn xếp trong trường hợp không khởi động SP (chế độ mặc định) và có khởi động SP (với (SP) = 3FH).

■ Nếu người sử dụng không khởi động thanh ghi SP (chế độ mặc định) thì: (xem hình bên dưới, phía trái)



■ Nếu người sử dụng muốn vùng nhớ ngăn xếp (chế độ tùy định) có tầm địa chỉ là 40H-7FH thì: (xem hình bên trên, phía phải)

1.3.5. Thanh ghi DPTR:



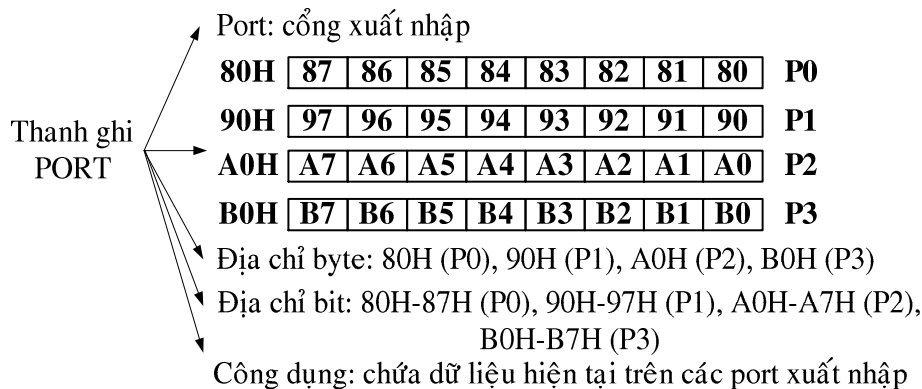
✓ Ví dụ: Khi ta muốn truy xuất (ghi/đọc) dữ liệu từ một ô nhớ thuộc RAM ngoài có địa chỉ là 0123H thì ta phải làm sao nạp được giá trị 0123H vào thanh ghi DPTR và sau đó thực hiện lệnh truy xuất **MOVX** (xem giải thích lệnh trong “Chương 3: Tập lệnh của 8051.”).

$$(DPTR) = 0123H \Leftrightarrow (DPH) = 01H \text{ và } (DPL) = 23H$$

✓ Ví dụ: Khi ta muốn truy xuất (đọc) byte mã từ một ô nhớ thuộc ROM trong có địa chỉ là 0ABCH thì ta phải làm sao nạp được giá trị 0ABCH vào thanh ghi DPTR và sau đó thực hiện lệnh truy xuất **MOVC** (xem giải thích lệnh trong “Chương 3: Tập lệnh của 8051.”).

$$(DPTR) = 0ABCH \Leftrightarrow (DPH) = 0AH \text{ và } (DPL) = BCH$$

1.3.6. Thanh ghi port xuất nhập:



✓ Lưu ý:

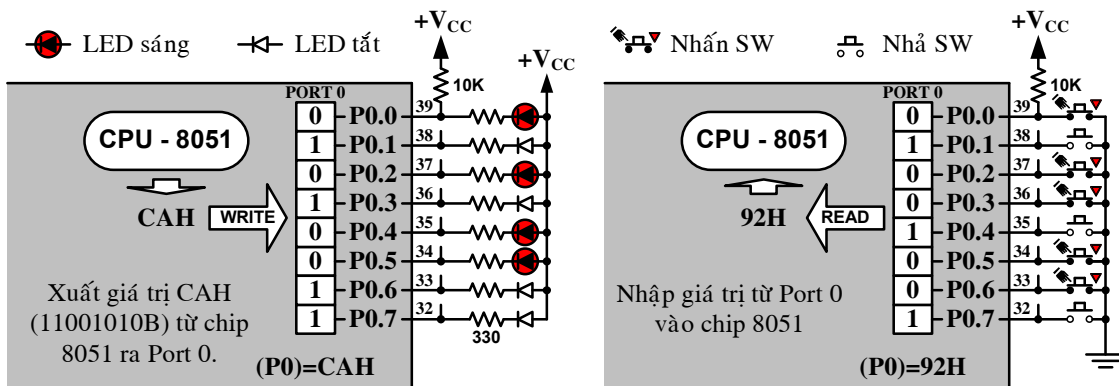
■ Trong trường hợp phần cứng có sử dụng ROM hoặc RAM bên ngoài thì ta không thể sử dụng Port 0 và Port 2 để xuất nhập dữ liệu. Vì khi đó chip 8051 sẽ sử dụng hai port này để xác định địa chỉ và dữ liệu cho bộ nhớ ngoài. Khi đó, ta chỉ có thể sử dụng Port 1 và Port 3 để xuất nhập dữ liệu.

■ Ở chế độ mặc định (khi reset) thì tất cả các chân của các port (P0 – P3) được cấu hình là **port xuất** dữ liệu. Muốn các chân port của chip 8015 làm **port nhập** dữ liệu thì ta cần phải được lập trình lại, bằng cách ghi mức logic cao (mức 1) đến tất cả các bit (các chân) của port trước khi bắt đầu nhập dữ liệu từ port.

✓ Ví dụ 1: Hoạt động xuất (ghi) và nhập (đọc) dữ liệu tại các chân port (Port 0) của chip 8051 (xem hình minh họa bên dưới).

• Hình phía trái: Minh họa trạng thái hoạt động của port khi thực hiện lệnh xuất (ghi) dữ liệu ra Port 0 của chip 8051.

• Hình phía phải: Minh họa trạng thái hoạt động của port khi thực hiện lệnh nhập (đọc) dữ liệu từ Port 0 của chip 8051.



✓ *Ví dụ 2:* Đoạn chương trình dưới đây sẽ cấu hình cho Port 0 làm port nhập (đọc) dữ liệu. Sau đó liên tục đọc dữ liệu từ port này và gửi dữ liệu đó đến Port 1 (xem giải thích lệnh trong “Chương 3: Tập lệnh của 8051.”):

MOV P0, #0FFH ;Cấu hình P0 làm port nhập bằng
;cách ghi “1” vào tất cả các bit.

BACK:

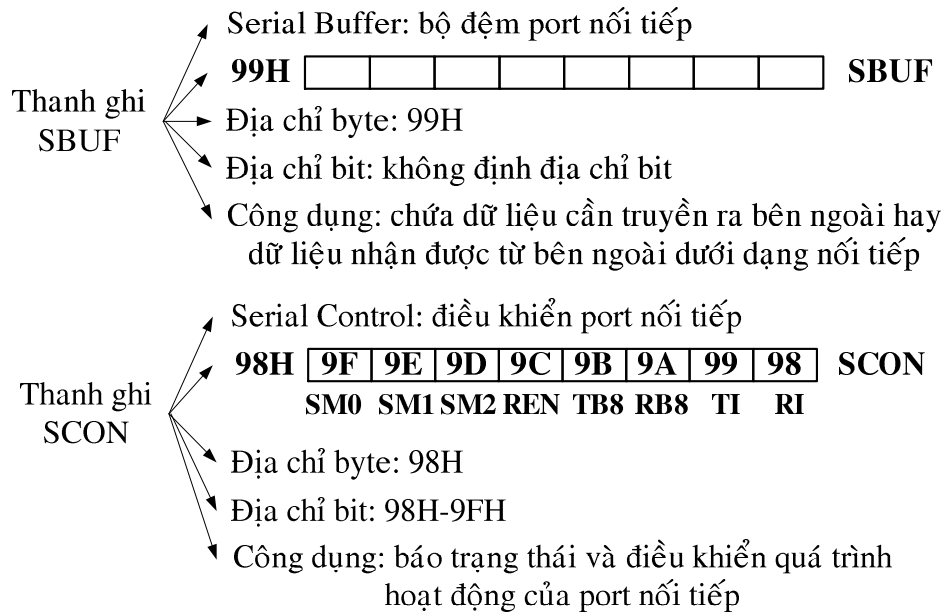
MOV A, P0 ;Đọc dữ liệu từ P0.
MOV P1, A ;Gửi dữ liệu đó ra P1.
SJMP BACK ;Lặp lại.

✓ *Ví dụ 3:* Đoạn chương trình dưới đây sẽ thực hiện các thao tác sau (xem giải thích lệnh trong “Chương 3: Tập lệnh của 8051.”):

- Liên tục kiểm tra bit P1.2 cho đến khi bit này bằng 1.
- Khi P1.2 = 1, hãy xuất (ghi) giá trị 45H ra P0.
- Gửi một xung mức cao tới P1.3.

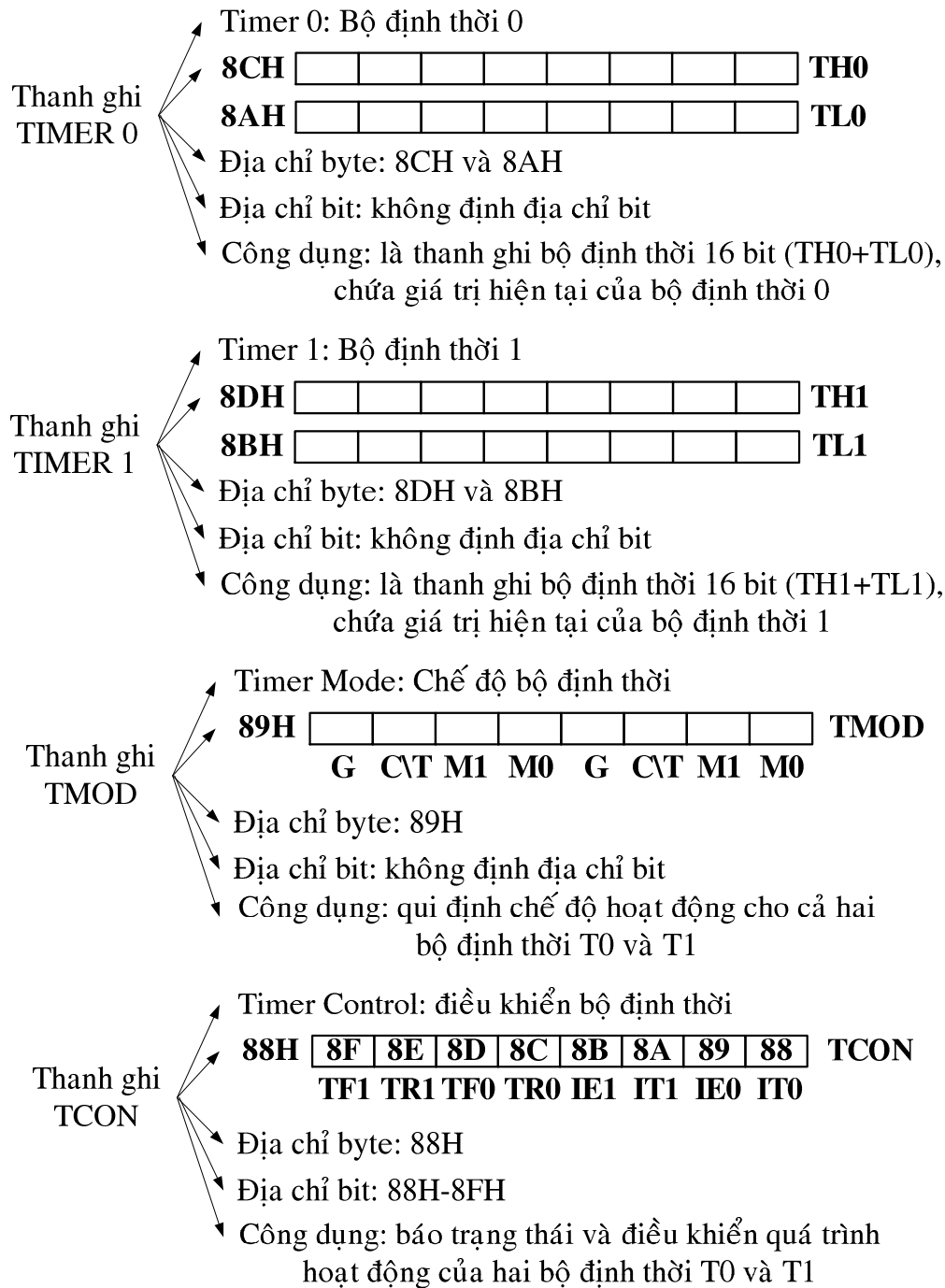
SETB P1.2 ;Cấu hình P1.2 làm ngõ vào.
JNB P1.2, \$;Kiểm tra liên tục nếu P1.2 = 0.
MOV P0, #45H ; Xuất giá trị 45H ra P0.
SETB P1.3 ;Đưa P1.3 lên cao rồi đưa P1.3
CLR P1.3 ;xuống thấp để tạo xung.

1.3.7. Thanh ghi port nối tiếp:



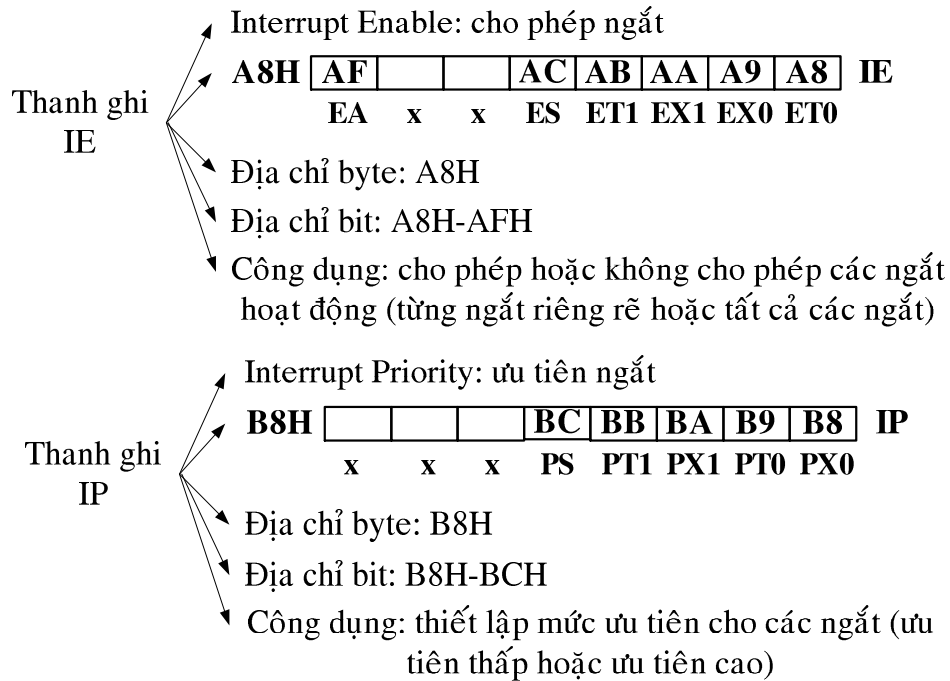
(xem thêm trong “Chương 5: Hoạt động của port nối tiếp.”)

1.3.8. Thanh ghi định thời:



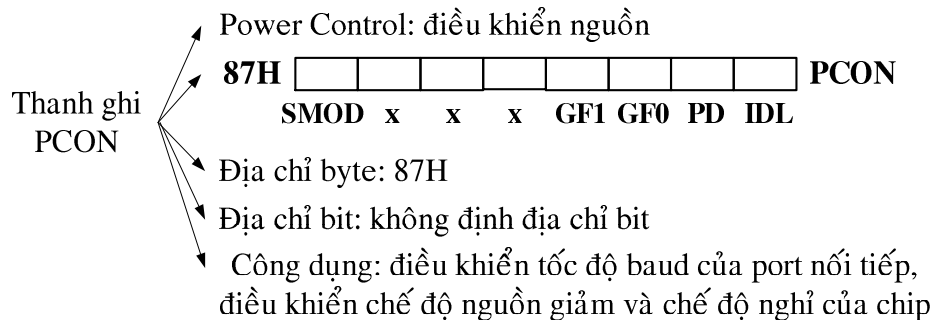
(xem thêm trong “Chương 4: Hoạt động của bộ định thời.”)

1.3.9. Thanh ghi ngắt:



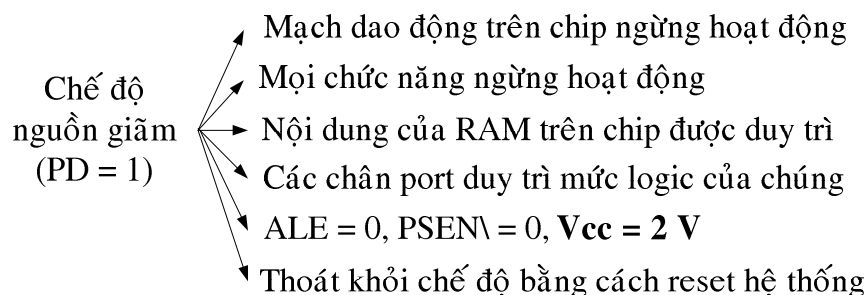
(xem thêm trong “Chương 6: Hoạt động ngắt.”)

1.3.10. Thanh ghi điều khiển nguồn:

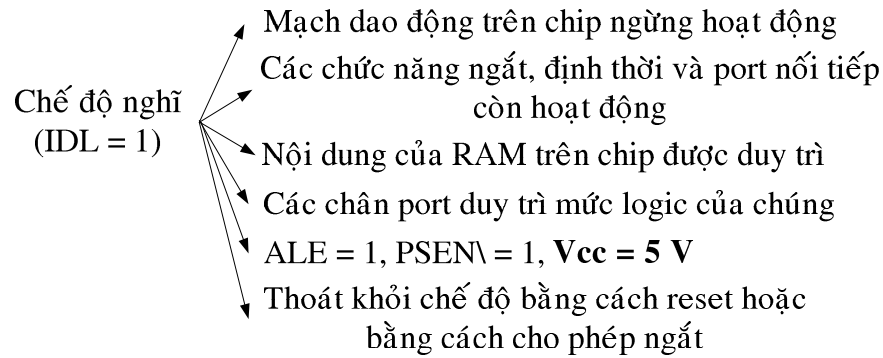


(xem thêm trong “Chương 5: Hoạt động của port nối tiếp.”)

- Bit SMOD (Serial Mode) → cho phép tăng gấp đôi tốc độ truyền dữ liệu nối tiếp (tốc độ baud) khi SMOD = 1.
- Bit GF1, GF0 (General Function) → cho phép người lập trình dùng với mục đích riêng (dự trữ cho các phiên bản chip trong tương lai).
- Bit PD (Power Down) → dùng để qui định chế độ nguồn giảm.
- Bit IDL (Idle) → dùng để qui định chế độ nghỉ.

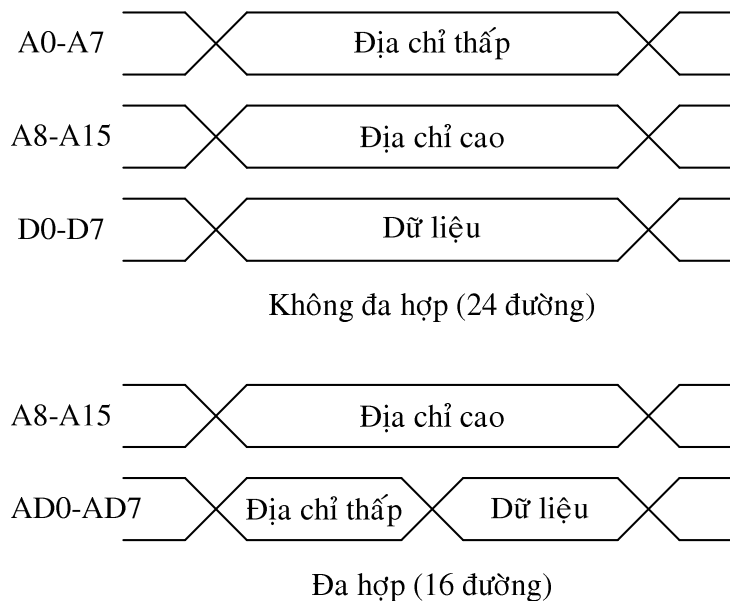


✓ **Lưu ý:** Hệ thống phải phục hồi $V_{cc} = 5V$ trước khi thoát khỏi chế độ nguồn giảm.



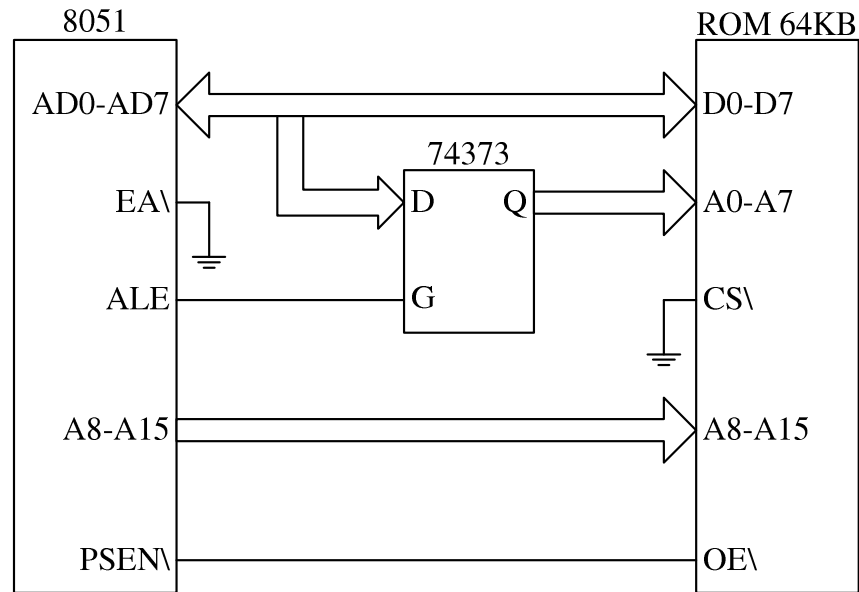
2. Bộ nhớ ngoài:

- Chip 8051 cho ta khả năng mở rộng:
 - Không gian bộ nhớ chương trình lên đến 64 KB.
 - Không gian bộ nhớ dữ liệu lên đến 64 KB.
- Khi sử dụng bộ nhớ ngoài:
 - Port 0 → bus địa chỉ byte thấp và bus dữ liệu đa hợp (A0-A7).
 - Port 2 → bus địa chỉ byte cao (A8-A15).
 - Port 3 → các tín hiệu điều khiển (WR , RD).
- Sự khác nhau giữa đa hợp và không đa hợp bus địa chỉ và bus dữ liệu:

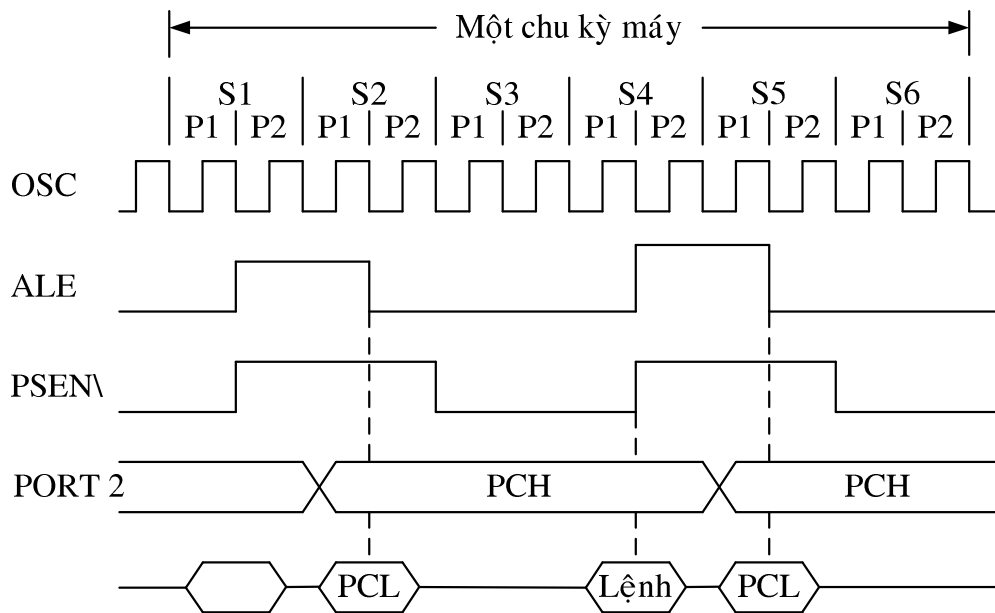


→ nhằm làm giảm số lượng chân đưa ra ngoài chip → giảm kích thước của chip.

2.1. Kết nối và truy xuất bộ nhớ chương trình ngoài:

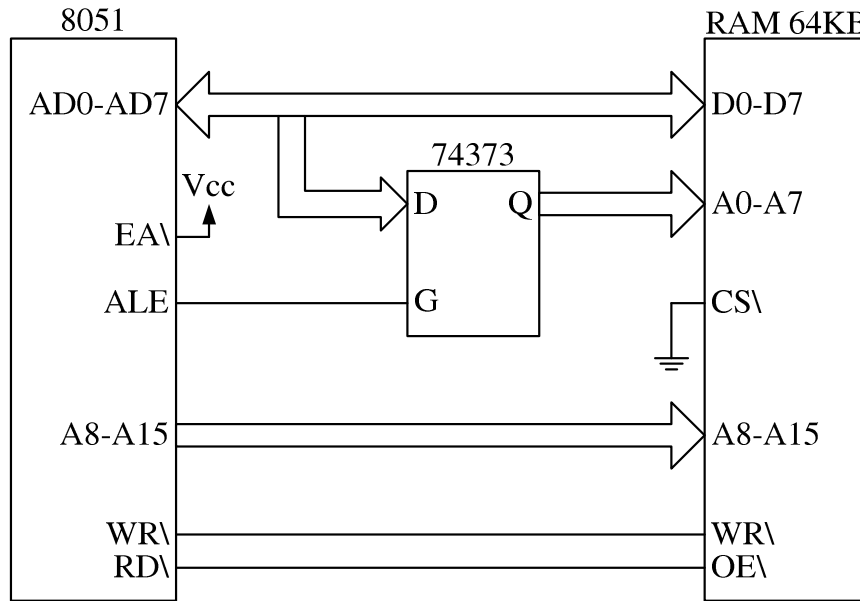


Truy xuất bộ nhớ chương trình bên ngoài

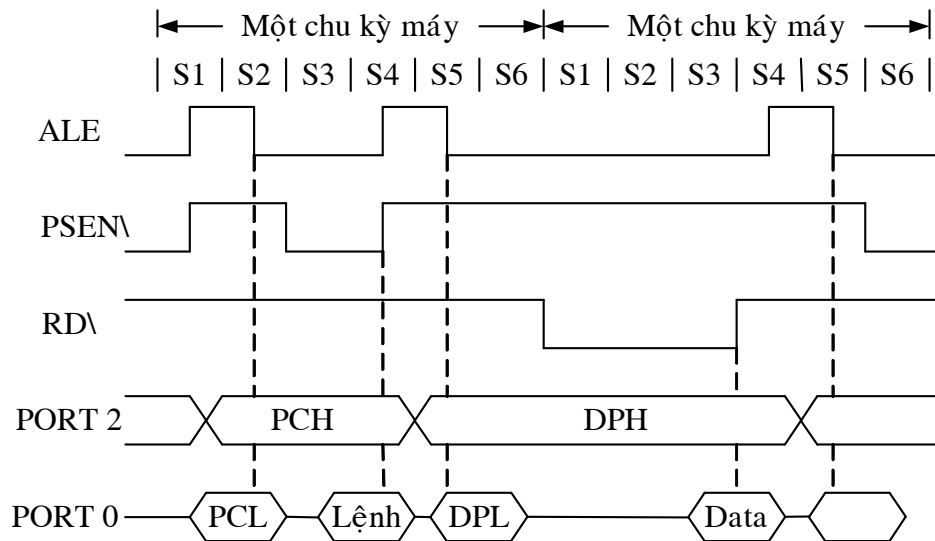


Giản đồ thời gian của chu kỳ tìm nạp lệnh ở bộ nhớ ngoài

2.2. Kết nối và truy xuất bộ nhớ dữ liệu ngoài:



Truy xuất bộ nhớ dữ liệu bên ngoài

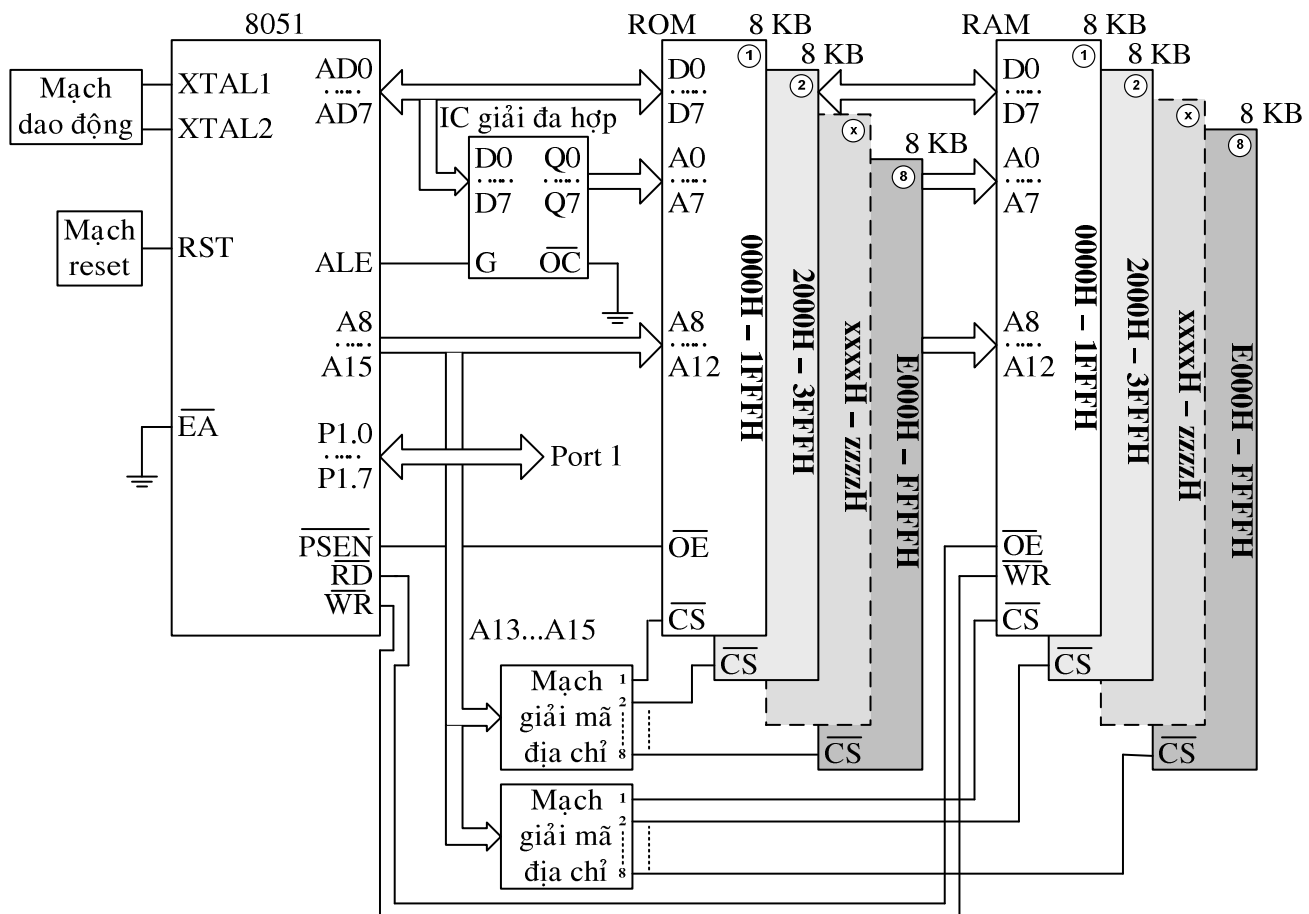


2.3. Giải mã địa chỉ:

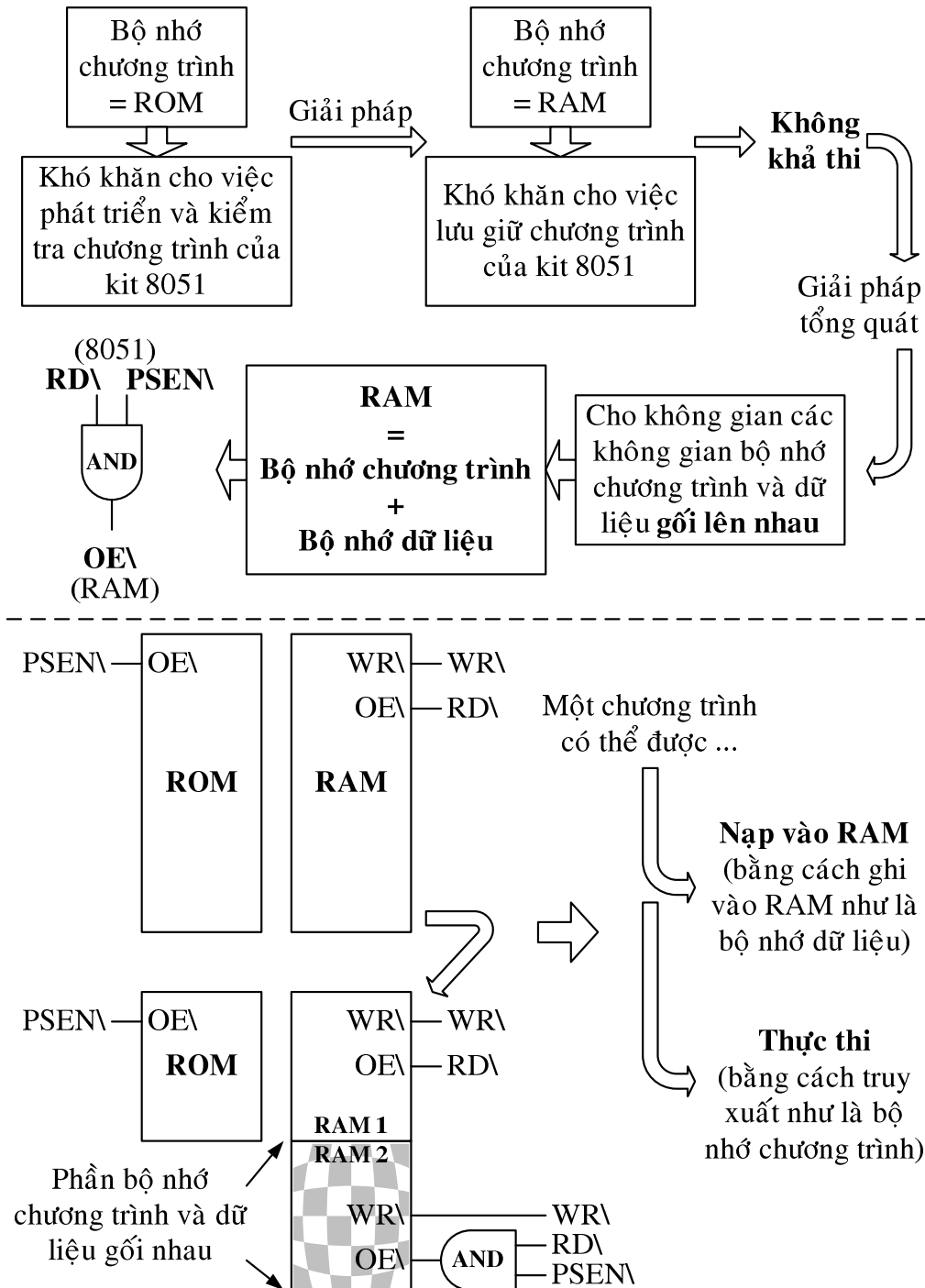
Nếu ta trường hợp ROM và RAM được kết hợp từ nhiều bộ nhớ có dung lượng nhỏ hoặc cả hai giao tiếp với chip 8051 thì ta cần phải giải mã địa chỉ. Việc giải mã địa chỉ này cũng cần cho hầu hết các bộ vi xử lý (*xem thêm trong “Chương phụ lục 1: Giải mã địa chỉ.”*).

Ví dụ nếu các ROM và RAM có dung lượng 8KB được sử dụng thì tầm địa chỉ mà chip 8051 quản lý ($0000H - FFFFH$) cần phải được giải mã thành từng đoạn 8 KB để chip có thể chọn từng IC nhớ trên các giới hạn 8KB tương ứng: IC1: $0000H - 1FFFH$, IC2: $2000H - 3FFFH$, ...

IC chuyên dùng cho việc tạo tín hiệu giải mã là 74HC138, các ngõ ra của IC này lần lượt được nối với các ngõ vào chọn chip CS\ tương ứng của các IC nhớ để cho phép các IC nhớ hoạt động (*tại một thời điểm chỉ có một IC nhớ được phép hoạt động*). Cần lưu ý là do các đường cho phép IC nhớ hoạt động riêng lẻ cho từng loại ($PSEN\$ cho bộ nhớ chương trình, $RD\$ và $WR\$ cho bộ nhớ dữ liệu) nên 8051 có thể quản lý không gian nhớ lên đến 64KB cho ROM và 64KB cho RAM.



2.4. Các không gian nhớ chương trình và dữ liệu gối nhau:



RAM 1: đóng vai trò là bộ nhớ dữ liệu.

RAM 2: đóng vai trò là bộ nhớ chương trình + bộ nhớ dữ liệu.

V. HOẠT ĐỘNG RESET:

Chân RST = 0 → Chip 8051 hoạt động bình thường.

Chân RST = 1 → Chip 8051 được reset.

✓ **Lưu ý:**

○ Để hoàn tất quá trình reset thì chân RST phải ở mức cao tối thiểu là 2 chu kỳ máy và sau đó chuyển xuống mức thấp.

○ Nội dung của RAM trong chip không bị ảnh hưởng bởi hoạt động reset.

○ Sau khi reset, việc thực thi chương trình luôn luôn bắt đầu ở vị trí đầu tiên trong bộ nhớ chương trình: **Địa chỉ 0000H**.

○ Trạng thái của các thanh ghi sau khi reset hệ thống:

■ Bộ đếm chương trình (PC)	0000H
■ Thanh ghi A	00H
■ Thanh ghi B	00H
■ Thanh ghi PSW	00H
■ Thanh ghi SP	07H
■ Thanh ghi DPTR	0000H
■ Port 0 – Port 3	FFH
■ Thanh ghi IP	xxx00000B
■ Thanh ghi IE	0xx00000B
■ Các thanh ghi định thời	00H
■ Thanh ghi SCON	00H
■ Thanh ghi SBUF	00H
■ Thanh ghi PCON (HMOS)	0xxxxxxxB
■ Thanh ghi PCON (CMOS)	0xxx0000B

VI. PHẦN BÀI TẬP:

Bài 1: Sử dụng một vi mạch 74138 và các cổng cần thiết để thiết kế mạch giải mã địa chỉ tạo ra các tín hiệu chọn chip tương ứng các vùng địa chỉ sau:

Tín hiệu chọn chip	Vùng địa chỉ	Đặc tính truy xuất
$\overline{CS0}$	0000H - 3FFFH	$\overline{PSEN}$
$\overline{CS1}$	4000H - 7FFFH	$\overline{PSEN}$
$\overline{CS2}$	6000H - 7FFFH	$\overline{RD}, \overline{WR}$
$\overline{CS3}$	8000H - 87FFFH	$\overline{RD}$
$\overline{CS4}$	8800H - 8FFFH	$\overline{WR}$

Bài 2: Sử dụng một vi mạch 74138 và các cổng cần thiết để thiết kế mạch giải mã địa chỉ tạo ra các tín hiệu chọn chip tương ứng các vùng địa chỉ sau:

Tín hiệu chọn chip	Vùng địa chỉ	Đặc tính truy xuất
$\overline{CS0}$	9800H - 9BFFH	$\overline{PSEN}$
$\overline{CS1}$	9800H - 9BFFH	$\overline{RD}, \overline{WR}$
$\overline{CS2}$	9C00H - 9DFFH	$\overline{RD}, \overline{WR}$
$\overline{CS3}$	9E00H - 9EFFH	$\overline{RD}, \overline{WR}$

Bài 3: Chỉ dùng một vi mạch 74138 (không dùng thêm cổng), thiết kế mạch giải mã địa chỉ tạo ra một tín hiệu chọn chip /CS tương ứng tầm địa chỉ F000H-F3FFH.