

---

# AT89C51

## 1. Giới thiệu khái quát về họ IC MCS-51™

**MCS-51™** là một họ IC vi điều khiển do Intel phát triển và sản xuất. Một số nhà sản xuất được phép cung cấp các IC tương thích với các sản phẩm MCS-51™ của Intel là Siemens, Advanced Micro Devices, Fujitsu, Philips, Atmel...

Các IC của họ MCS-51™ có các đặc trưng chung như sau:

- 4 port I/O 8 bit
- Giao tiếp nối tiếp
- 64K không gian bộ nhớ chương trình mở rộng
- 64K không gian bộ nhớ dữ liệu mở rộng
- Một bộ xử lý luận lý (thao tác trên các bit đơn)
- 210 bit được địa chỉ hóa
- Bộ nhân/chia 4  $\mu$ s.

Ngoài ra, tùy theo số hiệu sản xuất mà chúng có những khác biệt về bộ nhớ và bộ định thời/bộ đếm như trong bảng so sánh dưới đây:

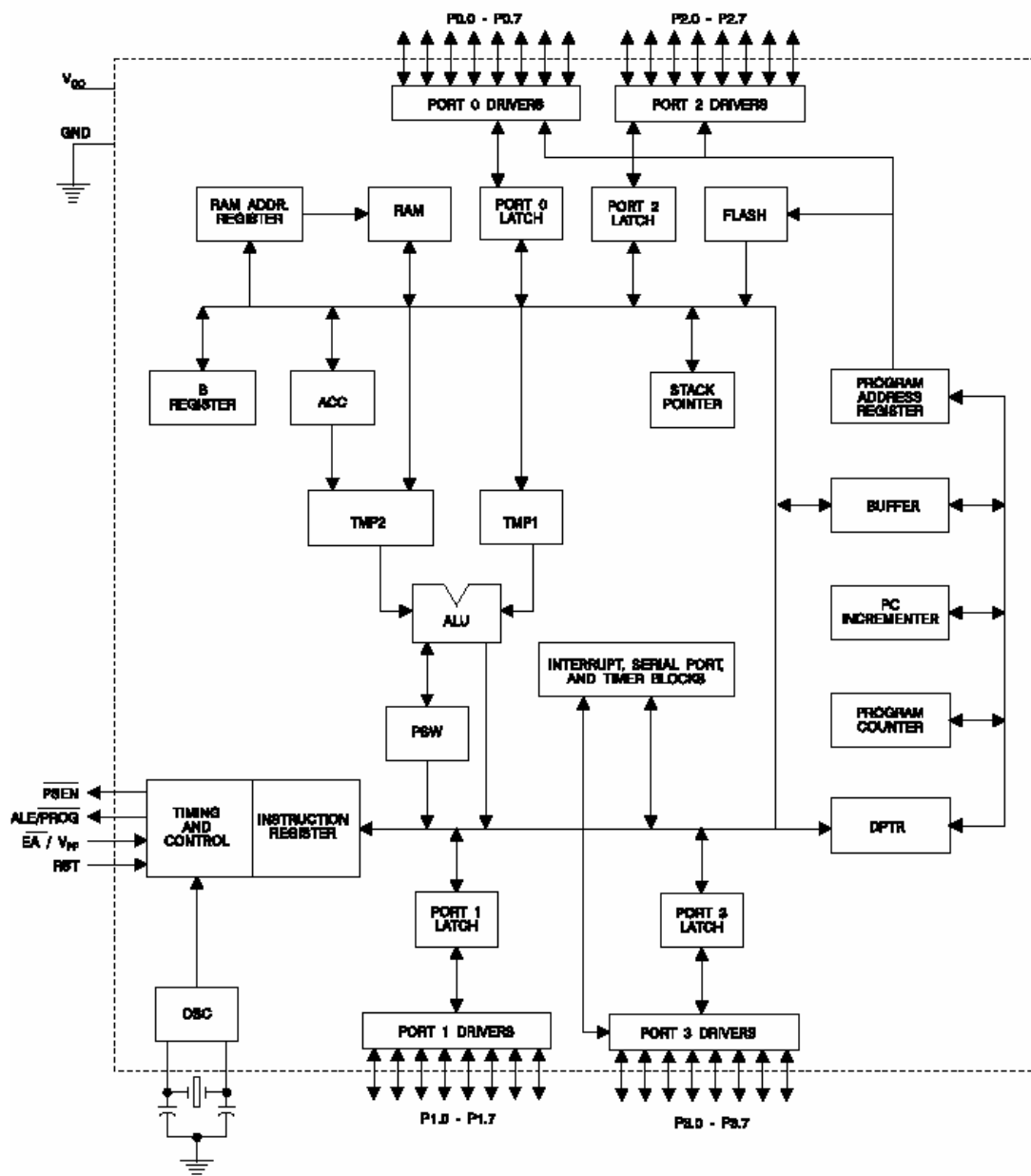
sản xuất	ó chương trình trên	ữ liệu trên chip	định thời (m)
	M ROM ASH	byte byte byte byte	
	M ROM ASH	byte byte byte byte	

## 2 Giới thiệu AT89C51

---

AT89C51 là một Microcomputer 8 bit, loại CMOS, có tốc độ cao và công suất thấp với bộ nhớ Flash có thể lập trình được. Nó được sản xuất với công nghệ bộ nhớ không bay hơi mật độ cao của hãng Atmel, và tương thích với chuẩn công nghiệp của 80C51 và 80C52 về chân ra và bộ lệnh. Vì lý do đó, kể từ đây về sau ta sẽ dùng thuật ngữ "80C51" (hoặc "8051")

*Sơ đồ khối của AT89C51*

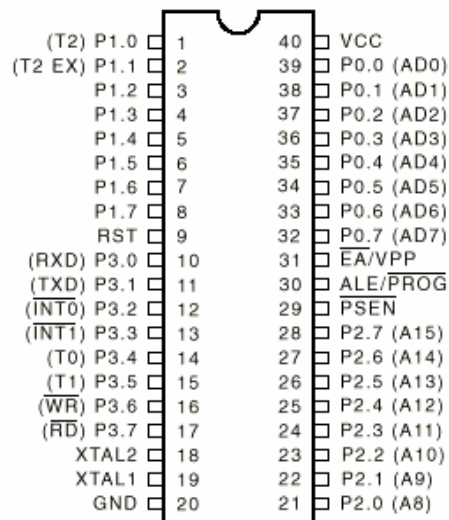


## 2.1 Những đặc trưng của AT89C51.

+ Tương thích với các sản phẩm MCS-51

- + 4KByte bộ nhớ Flash có thể lập trình lại với 1000 chu kỳ đọc/xoá
- + Hoạt động tĩnh đầy đủ: 0Hz đến 24MHz
- + Khoá bộ nhớ chương trình ba cấp
- + 128 x 8 bit RAM nội
- + 32 đường xuất/nhập lập trình được (tương ứng 4 port)
- + Hai timer/counter 16 bit
- + Một port nối tiếp song công lập trình được
- + Mạch đồng hồ và bộ dao động trên chip

*Cấu hình chân của AT89C51 như sau:*



Như vậy AT89C51 có tất cả 40 chân. Mỗi chân có chức năng như các đường I/O (xuất/nhập), trong đó 24 chân có công dụng kép: mỗi đường có thể hoạt động như một đường I/O hoặc như một đường điều khiển hoặc như thành phần của bus địa chỉ và bus dữ liệu.

#### *Mô tả chân*

- **VCC** (chân 40)

Chân cấp nguồn.

- **GND** (chân 20)

Chân nối đất.

- **Port 0**

Port 0 là một port xuất/nhập song hướng cực máng hồ 8 bit. Nếu được sử dụng như là một ngõ xuất thì mỗi chân có thể kéo 8 ngõ vào TTL. Khi mức 1 được viết vào các chân của port 0, các chân này có thể được dùng như là các ngõ nhập tổng trở cao.

Port 0 có thể được định cấu hình để hợp kênh giữa bus địa chỉ và bus dữ liệu (phần byte thấp) khi truy cập đến bộ nhớ dữ liệu và bộ nhớ chương trình ngoài. Ở chế độ này, PO có các điện trở pullup bên trong.

Port 0 cũng nhận các byte code (byte mã chương trình) khi lập trình Flash, và xuất ra các byte code khi kiểm tra chương trình. Cần có các điện trở pullup bên ngoài khi thực hiện việc kiểm tra chương trình.

- **Port 1**

Port 1 là một port xuất/nhập song hướng 8 bit có các điện trở pullup bên trong. Các bộ đệm ngõ ra của port 1 có thể kéo hoặc cung cấp 4 ngõ nhập TTL. Khi mức 1 được viết vào các chân của port 1, chúng được kéo lên cao bởi các điện trở pullup nội và có thể được dùng như là các ngõ nhập. Nếu đóng vai trò là các ngõ nhập, các chân của port 1 (được kéo xuống thấp qua các điện trở bên ngoài) sẽ cấp dòng  $I_{IL}$  do các điện trở pullup bên trong.

- **Port 2**

Port 2 là một port xuất/nhập song hướng 8 bit có các điện trở pullup bên trong.

Các bộ đệm ngõ ra của port 2 có thể kéo hoặc cung cấp 4 ngõ vào TTL. Khi các mức 1 được viết vào các chân của port 2 thì chúng được kéo lên cao bởi các điện trở pullup nội và có thể được dùng như các ngõ vào. Khi được dùng như các ngõ vào, các chân của port 2 (được kéo xuống qua các điện trở bên ngoài) sẽ cấp dòng  $I_{IL}$  do có các điện trở pullup bên trong.

Port 2 phát ra byte cao của địa chỉ khi đọc từ bộ nhớ chương trình ngoài và khi truy cập bộ nhớ dữ liệu ngoài dùng các địa chỉ 16 bit (MOVX @DPTR). Trong ứng dụng này, nó dùng các điện trở pullup nội "mạnh" khi phát ra các mức 1. Khi truy cập bộ nhớ dữ liệu ngoài dùng các địa chỉ 8 bit (MOVX @RI), port 2 phát ra các nội dung của thanh ghi chức năng đặc biệt P2.

Port 2 cũng nhận các bit cao của địa chỉ và một vài tín hiệu điều khiển khi lập trình và kiểm tra Flash.

- **Port 3**

Port 3 là một port xuất/nhập song hướng 8 bit có điện trở pullup nội bên trong.

Các bộ đệm ngõ ra của port 3 có thể kéo hoặc cung cấp 4 ngõ vào TTL. Khi các mức 1 được viết vào các chân của port 3 thì chúng được kéo lên cao bởi các điện trở pullup nội và có thể được dùng như các ngõ vào. Khi được dùng như các ngõ vào, các chân của port 3 (được kéo xuống qua các điện trở bên ngoài) sẽ cấp dòng  $I_{IL}$  do có các điện trở pullup bên trong.

Port 3 cũng cung cấp các chức năng của các đặc trưng đặc biệt như được liệt kê dưới đây:

		ức năng chuyển đổi
		nhập nối tiếp xuất nối tiếp bên ngoài bên ngoài Timer/Counter 0 Timer/Counter 1 bộ nhớ dữ liệu ngoài bộ nhớ dữ liệu ngoài

### *Các chức năng chuyển đổi trên Port 3*

- **RST** (chân 9)

Ngõ vào reset. Một mức cao trên chân này khoảng hai chu kỳ máy trong khi bộ dao động đang chạy sẽ reset thiết bị.

- **ALE/PROG**

ALE là một xung ngõ ra để chốt byte thấp của địa chỉ trong khi truy cập bộ nhớ ngoài. Chân này cũng là ngõ nhập xung lập trình (**PROG**) khi lập trình Flash.

Khi hoạt động bình thường, ALE được phát với một tỷ lệ không đổi là 1/6 tần số bộ dao động và có thể được dùng cho các mục đích timing và clocking bên ngoài. Tuy nhiên, lưu ý rằng một xung ALE sẽ bị bỏ qua mỗi khi truy cập bộ nhớ dữ liệu ngoài.

Nếu muốn, hoạt động ALE có thể cấm được bằng cách set bit 0 của SFR tại địa chỉ 8Eh. Nếu bit này được set, ALE chỉ được hoạt động khi có một lệnh MOVX hoặc MOVC. Ngược lại, chân này được kéo lên cao bởi các điện trở pullup "nhẹ". Việc set bit cấm-ALE không có tác dụng khi bộ vi điều khiển đang ở chế độ thi hành ngoài.

- **PSEN**

PSEN (Program Store Enable) là xung đọc bộ nhớ chương trình ngoài. Khi AT89C52 đang thi hành mã (code) từ bộ nhớ chương trình ngoài, **PSEN** được kích hoạt hai lần mỗi chu kỳ máy, nhưng hai hoạt động **PSEN** sẽ bị bỏ qua mỗi khi truy cập bộ nhớ dữ liệu ngoài.

- **EA/Vpp**

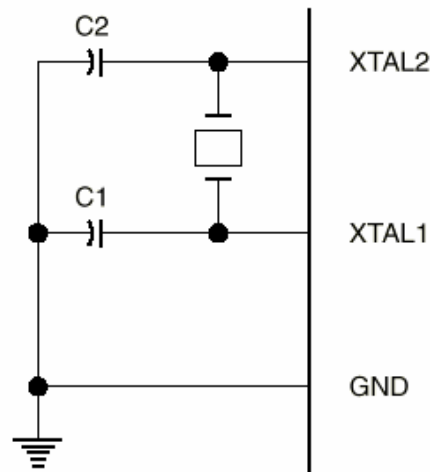
**EA** (External Access Enable) phải được nối với GND để cho phép thiết bị đọc code từ bộ nhớ chương trình ngoài có địa chỉ từ 0000H đến FFFFH. Tuy nhiên, lưu ý rằng nếu bit khoá 1 (lock-bit 1) được lập trình, **EA** sẽ được chốt bên trong khi reset.

**EA** phải được nối với Vcc khi thi hành chương trình bên trong. Chân này cũng nhận điện áp cho phép lập trình Vpp=1.2V khi lập trình Flash (khi đó áp lập trình 1.2V được chọn).

- ***XTAL1 và XTAL2***

XTAL1 và XTAL2 là hai ngõ vào và ra của một bộ khuếch đại dao động nghịch được cấu hình để dùng như một bộ dao động trên chip.

**Figure 7. Oscillator Connections**



Notes: C1, C2 = 30 pF  $\pm$  10 pF for Crystals  
= 40 pF  $\pm$  10 pF for Ceramic Resonators

*Hình. Các kết nối của bộ dao động*

Không có yêu cầu nào về duty cycle của tín hiệu xung ngoài, vì ngõ nhập nối với mạch tạo xung nội là một flip-flop chia đôi, nhưng các chỉ định về thời gian high và low, các mức áp tối đa và tối thiểu phải được tuân theo.

Các đặc trưng khác sẽ được trình bày một cách chi tiết hơn ở những phần tiếp theo sau đây.

### **3 Tổ chức bộ nhớ**

8051/8031 có bộ nhớ theo cấu trúc Harvard: có những vùng bộ nhớ riêng biệt cho chương trình và dữ liệu. Như đã nói ở trên, cả chương trình và dữ liệu có thể ở bên trong; dù vậy chúng có thể được mở rộng bằng các thành phần ngoài lên đến tối đa 64 Kbytes bộ nhớ chương trình và 64 Kbytes bộ nhớ dữ liệu.

Bộ nhớ bên trong bao gồm ROM và RAM trên chip, RAM trên chip bao gồm nhiều phần: phần lưu trữ đa dụng, phần lưu trữ địa chỉ hóa từng bit, các bank thanh ghi và các thanh ghi chức năng đặc biệt.

Hai đặc tính cần lưu ý là :

- Các thanh ghi và các port xuất nhập đã được xếp trong bộ nhớ và có thể được truy xuất trực tiếp giống như các địa chỉ bộ nhớ khác.
- Ngăn xếp bên trong RAM nội nhô hơn so với RAM ngoài như trong các bộ vi xử lý khác.

#### *Chi tiết về bộ nhớ RAM trên chip:*

Như ta thấy trên hình ( ), RAM bên trong 8051/8031 được phân chia giữa các bank thanh ghi (00H-1FH), RAM địa chỉ hóa từng bit (20H-2FH), RAM đa dụng (30H-7FH) và các thanh ghi chức năng đặc biệt (80H-FFH).

### 3.1 RAM đa dụng

Mặc dù trên hình cho thấy 80 byte RAM đa dụng chiếm các địa chỉ từ 30H-7FH, 32 byte dưới cùng từ 00H đến 1FH cũng có thể được dùng với mục đích tương tự (mặc dù các địa chỉ này đã có mục đích khác).

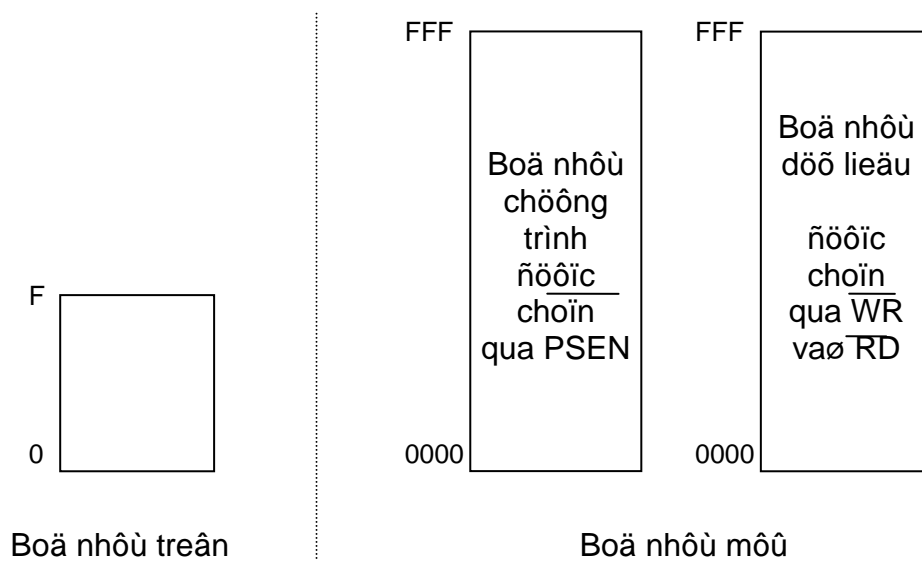
Mọi địa chỉ trong vùng RAM đa dụng đều có thể được truy xuất tự do dùng cách đánh địa chỉ trực tiếp hoặc gián tiếp. Ví dụ, để đọc nội dung ở địa chỉ 5FH của RAM nội vào thanh ghi tích lũy, lệnh sau sẽ được dùng :

`MOV A, 5FH`

Lệnh này di chuyển 1 byte dữ liệu dùng cách đánh địa chỉ trực tiếp để xác định "địa chỉ nguồn" (5FH). Đích nhận dữ liệu được ngầm xác định trong mã lệnh là thanh ghi tích lũy A.

RAM bên trong cũng có thể được truy xuất dùng cách đánh địa chỉ gián tiếp qua R0 hay R1. Ví dụ, hai lệnh sau thì hành cùng nhiệm vụ như lệnh đơn ở trên :

*Tóm tắt các vùng bộ nhớ của 8031/8051.*



MOV R0, #5FH

MOV A, @R0

Lệnh đầu dùng địa chỉ tức thời để di chuyển giá trị 5FH vào thanh ghi R0, và lệnh thứ hai dùng địa chỉ trực tiếp để di chuyển dữ liệu "được trỏ bởi R0" vào thanh ghi tích lũy.

Nhà chế	Nhà chế								Nhà chế	Nhà chế																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
7	RAM ã dùng								F																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
									F	F	F	F	F	F	F	F	B																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
									E	E	E	E	E	E	E	E	ACC																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						</

Tóm tắt bộ nhớ dữ liệu trên chip.

RAM

CÁC THANH GHI CHỖC NAËNG

### 3.2 RAM địa chỉ hóa từng bit

8051/8031 chứa 210 bit được địa chỉ hóa, trong đó 128 bit là ở các địa chỉ byte 20H đến 2FH, và phần còn lại là trong các thanh ghi chức năng đặc biệt.



Ý tưởng truy xuất từng bit riêng rẽ bằng phần mềm là một đặc tính tiện lợi của vi điều khiển nói chung. Các bit có thể được đặt, xóa, AND, OR, ... với một lệnh đơn. Đa số các vi xử lý đòi hỏi một chuỗi lệnh đọc-sửa-ghi để đạt được hiệu quả tương tự. Hơn nữa, các port I/O cũng được địa chỉ hóa từng bit làm đơn giản phần mềm xuất nhập từng bit.

Có 128 bit được địa chỉ hóa đa dụng ở các byte 20H đến 2FH. Các địa chỉ này được truy xuất như các byte hoặc như các bit phụ thuộc vào lệnh được dùng. Ví dụ, để đặt bit 67H, ta dùng lệnh sau :

```
SETB 67H
```

Chú ý rằng "địa chỉ bit 67H" là bit có trọng số lớn nhất (MSB) ở "địa chỉ byte 2CH". Lệnh trên sẽ không tác động đến các bit khác ở địa chỉ này. Các vi xử lý sẽ phải thi hành nhiệm vụ tương tự như sau :

```
MOV A, 2CH ; đọc cả byte
```

```
ORL A, #10000000B ; set MSB
```

```
MOV 2CH, A ; ghi lại cả byte
```

### 3.3 Các bank thanh ghi :

32 byte thấp nhất của bộ nhớ nội là dành cho các bank thanh ghi. Bộ lệnh của 8051/8031 hỗ trợ 8 thanh ghi (R0 đến R7) và theo mặc định (sau khi reset hệ thống) các thanh ghi này ở các địa chỉ 00H-07H. Lệnh sau đây sẽ đọc nội dung ở địa chỉ 05H vào thanh ghi tích lũy :

```
MOV A, R5
```

Đây là lệnh 1 byte dùng địa chỉ thanh ghi. Tất nhiên, thao tác tương tự có thể được thi hành bằng lệnh 2 byte dùng địa chỉ trực tiếp nằm trong byte thứ hai :

```
MOV A, 05H
```

Các lệnh dùng các thanh ghi R0 đến R7 thì sẽ ngắn hơn và nhanh hơn các lệnh tương ứng nhưng dùng địa chỉ trực tiếp. Các giá trị dữ liệu được dùng thường xuyên nên dùng một trong các thanh ghi này.

Bank thanh ghi tích cực có thể chuyển đổi bằng cách thay đổi các bit chọn bank thanh ghi trong từ trạng thái chương trình (PSW). Giả sử rằng bank thanh ghi 3 được tích cực, lệnh sau sẽ ghi nội dung của thanh ghi tích lũy vào địa chỉ 18H :

```
MOV R0, A
```

Ý tưởng dùng "các bank thanh ghi" cho phép "chuyển hướng" chương trình nhanh và hiệu quả (từng phần riêng rẽ của phần mềm sẽ có một bộ thanh ghi riêng không phụ thuộc vào các phần khác).

---

#### 4 Các thanh ghi chức năng đặc biệt

Một bản đồ vùng bộ nhớ trên chip được gọi là không gian thanh ghi chức năng đặc biệt (SFR) như được trình bày trong bảng.

Lưu ý rằng không phải tất cả các địa chỉ đều được sử dụng, và các địa chỉ không được sử dụng có thể không được cung cấp trên con chip. Các hành động đọc đến các địa chỉ này nói chung sẽ trả về các dữ liệu ngẫu nhiên, và các hành động viết sẽ có một hiệu ứng không xác định.

Các phần mềm người dùng không nên viết các mức 1 đến những vị trí không được liệt kê này, vì chúng có thể được dùng trong các sản phẩm tương lai khi thêm vào các đặc trưng mới. Trong trường hợp này, các giá trị reset hoặc không tích cực của các bit mới sẽ luôn là 0.

Các thanh ghi nội của 8051/8031 được truy xuất ngầm định bởi bộ lệnh. Ví dụ lệnh "INC A" sẽ tăng nội dung của thanh ghi tích lũy A lên 1. Tác động này được ngầm định trong mã lệnh.

Các thanh ghi trong 8051/8031 được định dạng như một phần của RAM trên chip. Vì vậy mỗi thanh ghi sẽ có một địa chỉ (ngoại trừ thanh ghi đếm chương trình và thanh ghi lệnh vì các thanh ghi này hiếm khi bị tác động trực tiếp, nên không lợi lộc gì khi đặt chúng vào trong RAM trên chip).

Đó là lý do để 8051/8031 có nhiều thanh ghi như vậy. Cũng như R0 đến R7, có 21 thanh ghi chức năng đặc biệt (SFR : Special Function Register) ở vùng trên của RAM nội, từ địa chỉ 80H đến FFH. Chú ý rằng hầu hết 128 địa chỉ từ 80H đến FFH không được định nghĩa. Chỉ có 21 địa chỉ SFR là được định nghĩa.

Ngoại trừ tích lũy (A) có thể được truy xuất ngầm như đã nói, đa số các SFR được truy xuất dùng địa chỉ trực tiếp. Chú ý rằng một vài SFR có thể được địa chỉ hóa bit hoặc byte. Người thiết kế phải thận trọng khi truy xuất bit và byte. Ví dụ lệnh sau : SETB 0EOH sẽ set bit 0 trong thanh ghi tích lũy, các bit khác không đổi. Ta thấy rằng EOH đồng thời là địa chỉ byte của cả thanh ghi tích lũy và là địa chỉ bit của bit có trọng số nhỏ nhất trong thanh ghi tích lũy. Vì lệnh SETB chỉ tác động trên bit, nên chỉ có địa chỉ bit là có hiệu quả.

##### 4.1 Từ trạng thái chương trình :

Từ trạng thái chương trình (PSW : Program Status Word) ở địa chỉ D0H chứa các bit trạng thái như bảng tóm tắt sau :

Bit	Kỳ hiệu	Nhà chæ	Ý nghĩa
PSW.7	CY	D7H	Cờ nhôu
PSW.6	AC	D6H	Cờ nhôu phui
PSW.5	F0	D5H	Cờ 0
PSW.4	RS1	D4H	Bit 1 chọn bank thanh ghi
PSW.3	RS0	D3H	Bit 0 chọn bank thanh ghi
			00 = bank 0 : nhà chæ 00H–07H
			01 = bank 1 : nhà chæ 08H–0FH
			10 = bank 2 : nhà chæ 10H–17H
			11 = bank 3 : nhà chæ 18H–1FH
PSW.2	OV	D2H	Cờ traøn
PSW.1	–	D1H	Đôi trôõ
PSW.0	P	D0H	Cờ parity chæun

### \* Cờ nhớ

Cờ nhớ (CY) có công dụng kép. Thông thường nó được dùng cho các lệnh toán học : nó sẽ được set nếu có một số nhớ sinh ra bởi phép cộng hoặc có một số mượn bởi phép trừ. Ví dụ, nếu thanh ghi tích lũy chứa FFH, thì lệnh `ADD A, #1` sẽ trả về thanh ghi tích lũy kết quả 00H và set cờ nhớ trong PSW.

Cờ nhớ cũng có thể xem như một thanh ghi 1 bit cho các lệnh luận lý thi hành trên bit. Ví dụ, lệnh sau sẽ AND bit 25H với cờ nhớ và đặt kết quả trở vào cờ nhớ : `ANL C, 25H`

### \* Cờ nhớ phụ

Khi cộng các số BCD, cờ nhớ phụ (AC) được set nếu kết quả của 4 bit thấp trong khoảng 0AH đến 0FH. Nếu các giá trị được cộng là số BCD, thì sau lệnh cộng cần có `DA A` (hiệu chỉnh thập phân thanh ghi tích lũy) để mang kết quả lớn hơn 9 vào nibble cao.

### \* Cờ 0

Cờ 0 (FO) là 1 bit cờ đa dụng dành cho các ứng dụng của người dùng.

### \* Các bit chọn bank thanh ghi

Các bit chọn bank thanh ghi (RS0 và RS1) xác định bank thanh ghi được tích cực. Chúng được xóa sau khi reset hệ thống và được thay đổi bằng phần mềm nếu cần. Ví dụ, ba lệnh sau cho phép bank thanh ghi 3 và di chuyển nội dung của thanh ghi R7 (địa chỉ byte 1FH) đến thanh ghi tích lũy :

```
SETB RS1
```

```
SETB RS0
```

```
MOV A, R7
```

Khi chương trình được hợp dịch, các đại chỉ bit đúng được thay thế cho các ký hiệu "RS1" và "RS0". Vậy, lệnh `SETB RS1` sẽ giống như lệnh `SETB 0D4H`.

### \* Cờ tràn

Cờ tràn (OV) được set sau một lệnh cộng hoặc trừ nếu có một phép toán bị tràn. Khi các số có dấu được cộng hoặc trừ với nhau, phần mềm có thể kiểm tra bit này để xác định xem kết quả có nằm trong tầm xác định không. Khi các số không dấu được cộng, bit OV có thể được bỏ qua. Các kết quả lớn hơn +127 hoặc nhỏ hơn -128 sẽ set bit OV.

---

Kết quả là một số có dấu 8EH được xem như -116, không phải là kết quả đúng (142), vì vậy, bit OV được set.

#### 4.2 Thanh ghi B.

Thanh ghi B ở địa chỉ FOH được dùng cùng với thanh ghi tích lũy A cho các phép toán nhân và chia. Lệnh MUL AB sẽ nhân các giá trị không dấu 8 bit trong A và B rồi trả về kết quả 16 bit trong A (byte thấp) và B (byte cao). Lệnh DIV AB sẽ chia A cho B rồi trả về kết quả nguyên trong A và phần dư trong B. Thanh ghi B cũng có thể được xem như thanh ghi đệm đa dụng. Nó được địa chỉ hóa từng bit bằng các địa chỉ bit FOH đến F7H.

#### 4.3 Con trỏ ngăn xếp .

Con trỏ ngăn xếp (SP) là một thanh ghi 8 bit ở địa chỉ 81H. Nó chứa địa chỉ của byte dữ liệu hiện hành trên đỉnh của ngăn xếp. Các lệnh trên ngăn xếp bao gồm các thao tác cất dữ liệu vào ngăn xếp và lấy dữ liệu ra khỏi ngăn xếp. Lệnh cất dữ liệu vào ngăn xếp sẽ làm tăng SP trước khi ghi dữ liệu, và lệnh lấy dữ liệu ra khỏi ngăn xếp sẽ đọc dữ liệu và giảm SP. Ngăn xếp của 8051/8031 được giữ trong RAM nội và được giới hạn các địa chỉ có thể truy xuất bằng địa chỉ gián tiếp. Chúng là 128 byte đầu của 8051/8031.

Để khởi động lại SP với ngăn xếp bắt đầu tại 60H, các lệnh sau đây được dùng:

```
MOV SP, #5FH
```

Trên 8051/8031 ngăn xếp bị giới hạn 32 byte vì địa chỉ cao nhất của RAM trên chip là 7FH. Sở dĩ dùng giá trị 5FH vì SP sẽ tăng lên 60H trước khi cất byte dữ liệu đầu tiên.

Người thiết kế có thể chọn không phải khởi động lại con trỏ ngăn xếp mà để nó lấy giá trị mặc định khi reset hệ thống. Giá trị mặc định đó là 07H và kết quả là ngăn đầu tiên để cất dữ liệu có địa chỉ là 08H. Nếu phần mềm ứng dụng không khởi động lại SP, bank thanh ghi 1 (có thể cả 2 và 3) sẽ không dùng được vì vùng RAM này đã được dùng làm ngăn xếp.

Ngăn xếp được truy xuất trực tiếp bằng các lệnh PUSH và POP để lưu trữ tạm thời và lấy lại dữ liệu, hoặc được truy xuất ngầm bằng các lệnh gọi chương trình con (ACALL, LCALL) và các lệnh trở về (RET, RETI) để cất và lấy lại bộ đếm chương trình.

#### 4.4 Con trỏ dữ liệu

Con trỏ dữ liệu (DPTR) được dùng để truy xuất bộ nhớ ngoài là một thanh ghi 16 bit ở địa chỉ 82H (DPL : byte thấp) và 83H (DPH : byte cao). Ba lệnh sau sẽ ghi 55H vào RAM ngoài ở địa chỉ 1000H :

```
MOV A, #55H
```

```
MOV DPTR, #1000H
```

```
MOVX @DPTR, A
```

---

Lệnh đầu tiên dùng địa chỉ tức thời để tải dữ liệu 55H vào thanh ghi tích lũy. Lệnh thứ hai cũng dùng địa chỉ tức thời, lần này để tải dữ liệu 16 bit 1000H vào con trỏ dữ liệu. Lệnh thứ ba dùng địa chỉ gián tiếp để di chuyển dữ liệu trong A (55H) đến RAM ngoài ở địa chỉ được chứa trong DPTR (1000H).

#### 4.5 Các thanh ghi port xuất nhập .

Các port của 8051/8031 bao gồm Port 0 ở địa chỉ 80H, Port 1 ở địa chỉ 90H, Port 2 ở địa chỉ A0H và Port 3 ở địa chỉ B0H. Tất cả các port đều được địa chỉ hóa từng bit. Điều đó cung cấp một khả năng giao tiếp thuận lợi. Ví dụ nếu một motor được nối qua một cuộn dây có transistor lái đến bit 7 của Port 1, nó có thể được bật và tắt bằng một lệnh đơn :

SETB            P1.7    ; bật motor

CLR            P1.7    ; tắt motor

Các lệnh trên dùng dấu chấm để xác định một bit trong một byte. Trình hợp dịch sẽ thi hành sự chuyển đổi cần thiết, vì vậy hai lệnh sau đây là như nhau :

CLR            P1.7

CLR            97H

Trong một ví dụ khác, xem xét giao tiếp đến một thiết bị với một bit trạng thái gọi là BUSY, được set khi thiết bị đang bận và được xóa khi thiết bị đã sẵn sàng. Nếu BUSY được nối tới P1.5, vòng lặp sau sẽ được dùng để chờ thiết bị trở lại trạng thái sẵn sàng :

WAIT :            JB            P1.5, WAIT

Lệnh này có nghĩa là "nếu bit P1.5 được set thì nhảy tới nhãn WAIT". Nói cách khác "nhảy trở lại và kiểm tra lần nữa".

#### 4.6 Các thanh ghi timer.

8051/8031 chứa hai bộ định thời / đếm 16 bit được dùng cho việc định thời hoặc đếm sự kiện. Timer 0 ở địa chỉ 8AH (TLO : byte thấp) và 8CH (TH0 : byte cao). Timer 1 ở địa chỉ 8BH (TL1 : byte thấp) và 8DH (TH1 : byte cao). Việc vận hành timer được set bởi thanh ghi Timer Mode (TMOD) ở địa chỉ 89H và thanh ghi điều khiển timer (TCON) ở địa chỉ 88H. Chỉ có TCON được địa chỉ hóa từng bit.

#### 4.7 Các thanh ghi port nối tiếp.

8051/8031 chứa một port nối tiếp trên chip dành cho việc trao đổi thông tin với các thiết bị nối tiếp như máy tính, modem hoặc cho việc giao tiếp với các IC khác có giao tiếp nối tiếp (các bộ chuyển đổi A/D, các thanh ghi dịch...). Một thanh ghi gọi là bộ đệm dữ liệu nối tiếp (SBUF) ở địa chỉ 99H sẽ giữ cả hai dữ liệu truyền và nhận. Khi truyền dữ liệu thì ghi lên SBUF, khi nhận dữ liệu thì đọc SBUF.

---

Các mode vận hành khác nhau được lập trình qua thanh ghi điều khiển port nối tiếp (SCON) (được địa chỉ hóa từng bit) ở địa chỉ 98H.

#### 4.8 Các thanh ghi ngắt.

8051/8031 có cấu trúc 5 nguồn ngắt, 2 mức ưu tiên. Các ngắt bị cấm sau khi reset hệ thống và sẽ được cho phép bằng việc ghi thanh ghi cho phép ngắt (IE) ở địa chỉ A8H. Cả hai thanh ghi được địa chỉ hóa từng bit.

#### 4.9 Thanh ghi điều khiển công suất.

Thanh ghi điều khiển công suất (PCON) ở địa chỉ 87H chứa nhiều bit điều khiển. Chúng được tóm tắt trong bảng sau :

Bit nghe	Ký hiệu	YÚ
7 set	SMOD thì	Bit gáp ñôi toác ñôi baud, neáu ñôi toác ñôi baud sẽ taêng gáp ñôi trong cauc mode
6	=	1,2 và 3 của port nối tiếp.
5	GF1	Khoảng ñình nghe.
2	GF0	Khoảng ñình nghe.
1	PD	Bit cô ña ñuông 0.
reset		Giaùm công suát, ñôi set ñe kích hoạt mode giaùm công suát, chæ thoát khi 0 IDL Mode chø, set ñe kích hoạt
mode chø, chæ		thoát khi cò ngát hoặ reset he

#### 5 Bảo vệ bộ nhớ.

Các bit khoá bộ nhớ chương trình. Vì điều khiển AT89C51 có 3 bit khoá có thể bỏ không lập trình (U) hoặc được lập trình (P) để nhận các đặc trưng thêm vào được liệt kê trong bảng dưới đây (với LB1, LB2, LB3 là các bit khóa tương ứng).

				ảo vệ
				khóa chương trình
				nh MOV C được thi hành từ bộ nhớ chương trình ngoài bị hi lấy các byte mã từ bộ nhớ nội, /EA được lấy mẫu và hốt lại khi reset và hơn nữa, việc lập trình bộ nhớ Flash là
				ế độ 2 nhưng việc kiểm tra cũng bị cấm
				ế độ 3 nhưng việc thi hành ngoài cũng bị cấm.



Khi bit khoá 1 được lập trình, mức logic tại chân  $\overline{EA}$  được lấy mẫu và chốt lại khi reset. Nếu thiết bị được bật nguồn mà không có reset, việc chốt sẽ được khởi tạo với một giá trị ngẫu nhiên cho đến khi được reset. Giá trị được chốt của  $\overline{EA}$  phải bằng với mức logic hiện tại ở chân đó để cho thiết bị làm việc một cách chính xác.

## 6 Hoạt động của port nối tiếp.

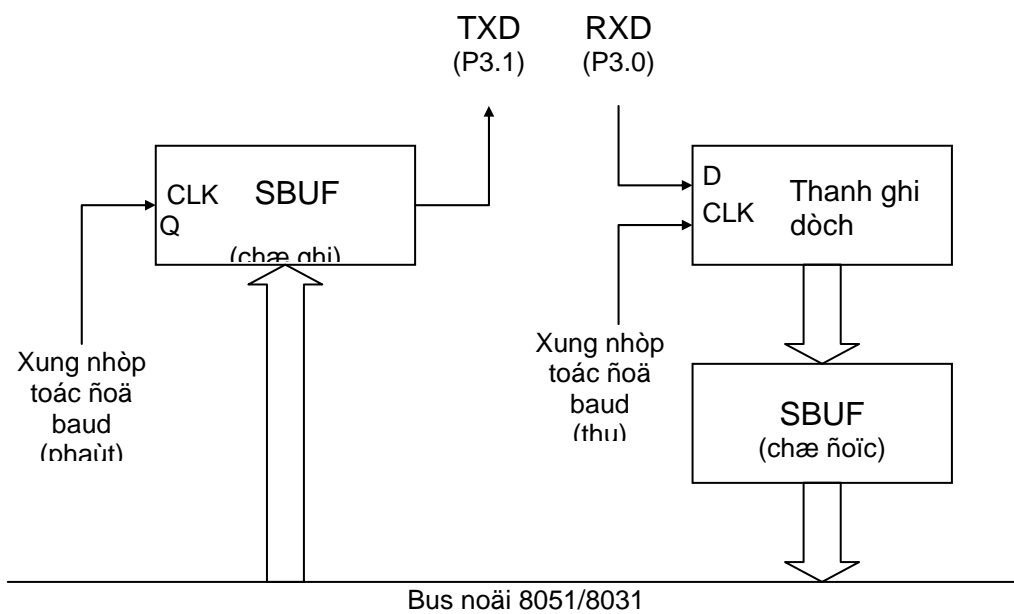
### 6.1 Giới thiệu.

8051/8031 có một port nối tiếp trong chip có thể hoạt động ở nhiều chế độ trên một dải tần số rộng. Chức năng chủ yếu của port nối tiếp là thực hiện chuyển đổi song song sang nối tiếp đối với dữ liệu xuất, và chuyển đổi nối tiếp sang song song với dữ liệu nhập.

Truy xuất phần cứng đến port nối tiếp qua các chân TXD và RXD. Các chân này có các chức năng khác với hai bit của Port 3, P3.1 ở chân 11 (TXD) và P3.0 ở chân 10 (RXD).

Port nối tiếp cho hoạt động song công (full duplex : thu và phát đồng thời), và đệm lúc thu (receiver buffering) cho phép một ký tự sẽ được thu và được giữ trong khi ký tự thứ hai được nhận. Nếu CPU đọc ký tự thứ nhất trước khi ký tự thứ hai được thu được thu đầy đủ thì dữ liệu sẽ không bị mất.

Hai thanh ghi chức năng đặc biệt cho phép phần mềm truy xuất đến port nối tiếp là : SBUF và SCON. Bộ đệm port nối tiếp (SBUF) ở địa chỉ 99H thật sự là hai bộ đệm. Viết vào SBUF để nạp dữ liệu sẽ được phát, và đọc SBUF để truy xuất dữ liệu thu được. Đây là hai thanh ghi riêng biệt : thanh ghi chỉ ghi để phát và thanh ghi chỉ đọc để thu.



Thanh ghi điều khiển port nối tiếp (SCON) ở địa chỉ 98H là thanh ghi có địa chỉ bit chứa các bit trạng thái và các bit điều khiển. Các bit điều khiển đặt chế độ hoạt động cho port nối tiếp, và các bit trạng

thái báo kết thúc việc phát hoặc thu ký tự. Các bit trạng thái có thể được kiểm tra bằng phần mềm hoặc có thể được lập trình để tạo ngắt.

Tần số làm việc của port nối tiếp, còn gọi là tốc độ baud có thể cố định (lấy từ bộ dao động trên chip). Nếu sử dụng tốc độ baud thay đổi, Timer 1 sẽ cung cấp xung nhịp tốc độ baud và phải được lập trình.

## 6.2 Thanh ghi điều khiển port nối tiếp

Chế độ hoạt động của port nối tiếp được đặt bằng cách ghi vào thanh ghi chế độ port nối tiếp (SCON) ở địa chỉ 98H. Sau đây là các bảng tóm tắt thanh ghi SCON và các chế độ của port nối tiếp :

Bit	Ký hiệu Nhà chế			Mo tả
SCON.7	SM0	9FH	Bit 0 của chế độ nối tiếp.	
SCON.6	SM1	9EH	Bit 1 của chế độ nối tiếp.	
SCON.5	SM2	9DH	Bit 2 của chế độ nối tiếp. Cho phép truyền thông nối tiếp trong các chế độ nối 2 và 3; RI sẽ không bỏ qua nếu	
SCON.4	REN	9CH	Cho phép bỏ thu phát nối tiếp lên 1 lần thu (nhận)	
SCON.3	TB8	9BH	Bit 8 phát, bit 9 thu nối tiếp trong các chế độ 2 và 3; nối tiếp sẽ chờ đợi phần mềm.	
SCON.2	RB8	9AH	Bit 8 thu, bit 9 thu nối tiếp.	
SCON.1	TI	99H	Cờ phát. Sẽ lên 1 khi kết thúc phát ký tự; nối tiếp chờ đợi phần mềm.	
SCON.0	RI	98H	Cờ thu. Sẽ lên 1 khi kết thúc thu ký tự; nối tiếp chờ đợi phần mềm.	

*Tóm tắt thanh ghi chế độ port nối tiếp SCON.*

SM0	SM1	Chế độ	Mo tả	Tốc độ baud
0	0	0	Thanh ghi dòch	Cố định ( $F_{OSC}/12$ ).
0	1	1	UART 8 bit	Thay đổi (sẽ bằng timer).
1	0	2	UART 9 bit	Cố định ( $F_{OSC}$ chia cho 12 hoặc 64).
1	1	3	UART 9 bit	Thay đổi (sẽ bằng timer).



### *Các chế độ port nối tiếp.*

Trước khi sử dụng port nối tiếp, phải khởi động SCON cho đúng chế độ. Ví dụ, lệnh `MOV SCON, #01010010B` khởi động port nối tiếp cho chế độ 1 ( $SM0/SM1 = 0/1$ ), cho phép bộ thu ( $REN = 1$ ) và đặt cờ ngắt phát ( $TI = 1$ ) để chỉ bộ phát sẵn sàng hoạt động.

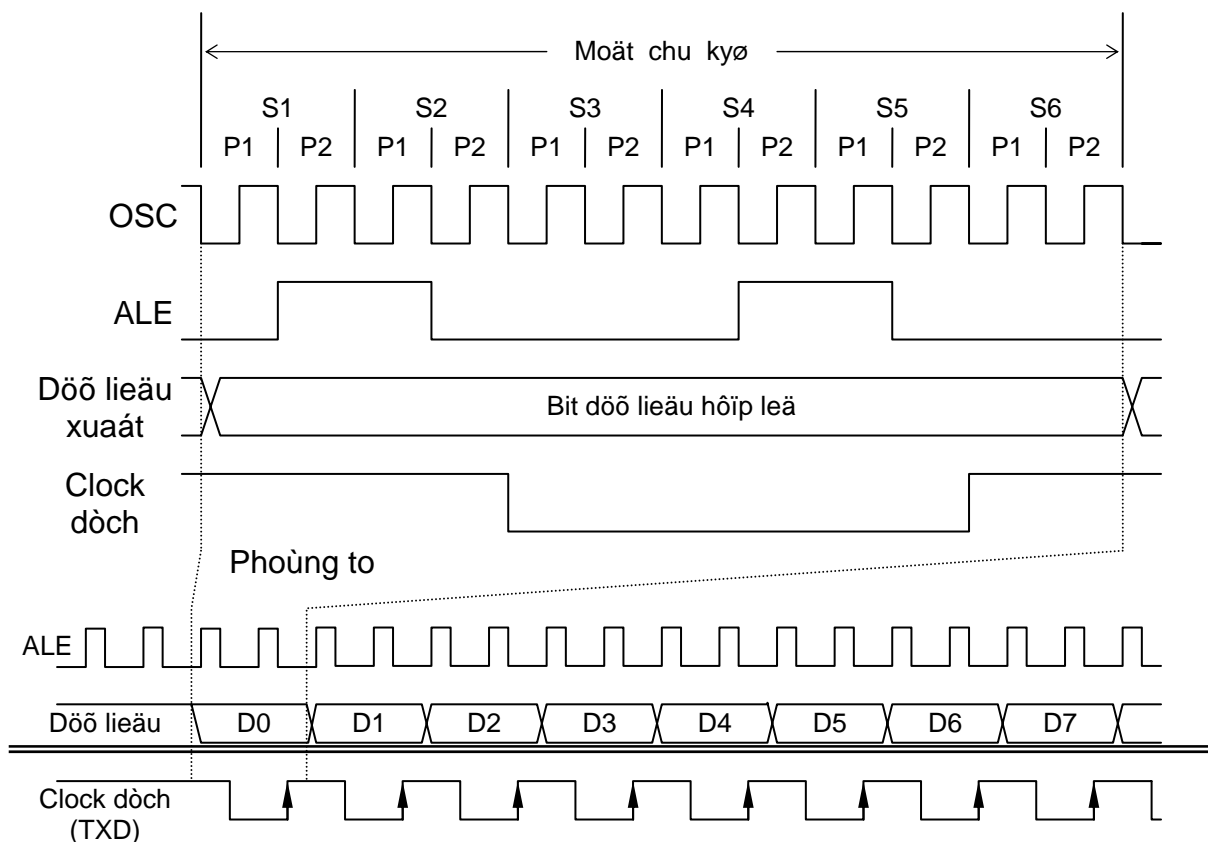
### 6.3 Các chế độ hoạt động.

Port nối tiếp có 4 chế độ hoạt động, có thể chọn được bằng cách viết các số 1 hay 0 vào các bit  $SM0$  và  $SM1$  trong  $SCON$ . Có ba chế độ cho phép truyền thông bất đồng bộ, với mỗi ký tự được thu (nhận) hoặc phát đều được đóng khung bằng một bit start và 1 bit stop. Ở chế độ thứ tư, port nối tiếp hoạt động như một thanh ghi dịch đơn giản.

#### 6.3.1 Thanh ghi dịch 8 bit (chế độ 0).

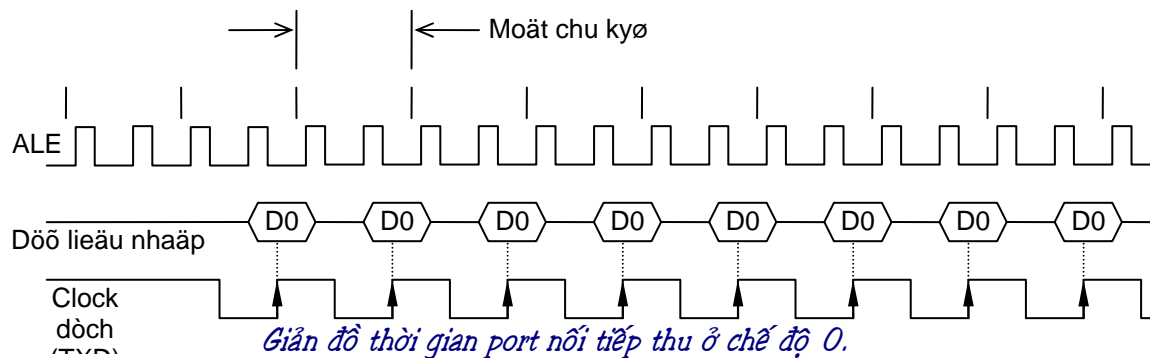
Chế độ 0 được chọn bằng cách ghi các bit 0 vào  $SM1$  và  $SM0$  của  $SCON$ , đưa port nối tiếp vào chế độ thanh ghi dịch 8 bit. Dữ liệu nối tiếp vào và ra qua  $RXD$  và  $TXD$  xuất xung nhịp dịch. 8 bit được phát hoặc thu với bit đầu tiên là LSB. Tốc độ baud cố định ở 1/12 tần số dao động trên chip.

Việc phát đi được khởi động bằng bất cứ lệnh nào ghi dữ liệu vào  $SBUF$ . Dữ liệu được dịch ra ngoài trên đường  $RXD$  ( $P3.0$ ) với các xung nhịp được gửi ra đường  $TXD$  ( $P3.1$ ). Mỗi bit phát đi hợp lệ (trên  $RXD$ ) trong một chu kỳ máy. Trong mỗi chu kỳ máy, tín hiệu xung nhập xuống thấp ở  $S3P1$  và trở về mức cao ở  $S6P1$ .

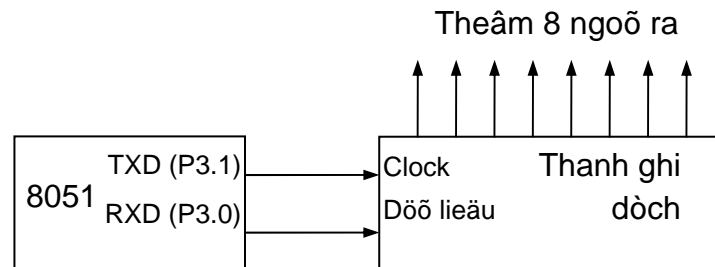


*Giãn đồ thời gian port nối tiếp phát ở chế độ 0.*

Việc thu được khởi động khi bit cho phép bộ thu (REN) là 1 và bit ngắt thu (RI) là 0. Qui tắc tổng quát là đặt REN khi bắt đầu chương trình để khởi động port nối tiếp, rồi xóa RI để bắt đầu hoạt động nhập dữ liệu. Khi RI bị xóa, các xung nhịp được đưa ra đường TXD, bắt đầu chu kỳ máy kế tiếp, và dữ liệu theo xung nhịp ở đường RXD. Lấy xung nhịp cho dữ liệu vào port nối tiếp xảy ra ở cạnh dương của TXD.



Một ứng dụng của chế độ thanh ghi dịch là mở rộng khả năng xuất của 8051/8031. IC thanh ghi dịch nối tiếp ra song song có thể được nối vào các đường TXD và RXD của 8051/8031 để cung cấp thêm 8 đường ra. Có thể nối xâu chuỗi thêm các thanh ghi dịch để mở rộng thêm.



*Chế độ thanh ghi dịch của port nối tiếp.*

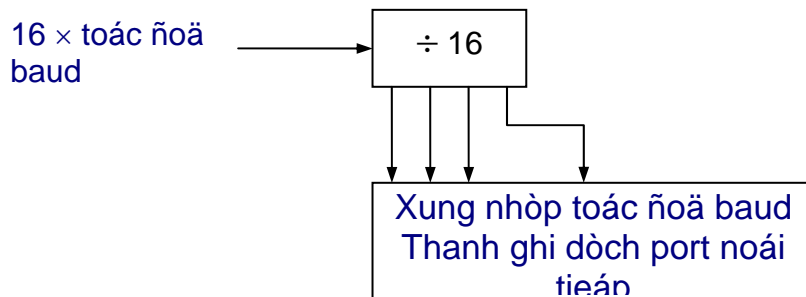
**6.3.2 UART 8 bit với tốc độ baud thay đổi được (chế độ 1).**

Ở chế độ 1, port nối tiếp của 8051/8031 làm việc như một UART 8 bit với tốc độ baud thay đổi được. Một UART (Universal Asynchronous Receiver/Transmitter : Bộ thu/phát bất đồng bộ vạn năng) là một dụng cụ thu và phát dữ liệu nối tiếp với mỗi ký tự dữ liệu đi trước là bit start ở mức thấp và theo sau là bit stop ở mức cao. Đôi khi xen thêm bit kiểm tra chẵn lẻ giữa bit dữ liệu cuối cùng và bit stop. Hoạt động chủ yếu của UART là chuyển đổi song song sang nối tiếp với dữ liệu xuất và chuyển đổi nối tiếp sang song song với dữ liệu nhập.

Ở chế độ 1, 10 bit được phát trên TXD hoặc thu trên RXD. Những bit đó là : 1 bit start (luôn luôn

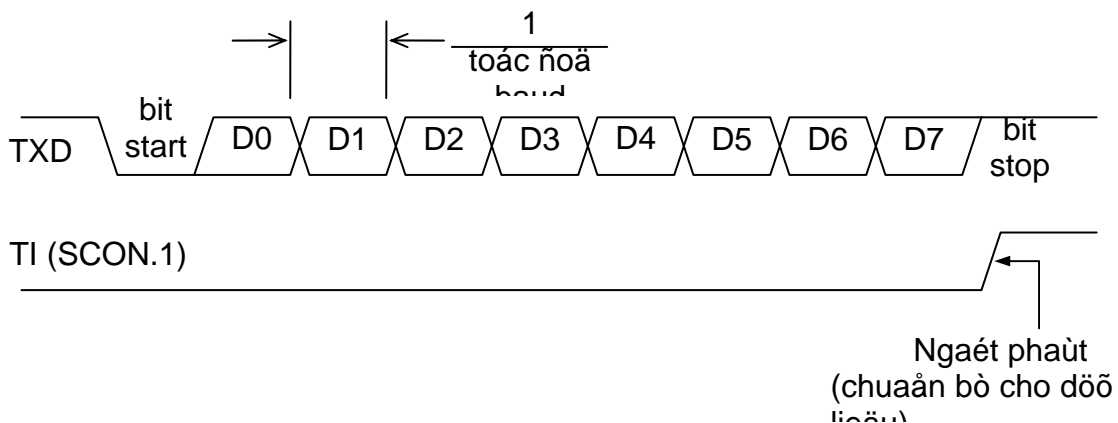
là 0), 8 bit dữ liệu (LSB đầu tiên) và 1 bit stop (luôn luôn là 1). Với hoạt động thu, bit stop được đưa vào RB8 trong SCON. Trong 8051/8031 chế độ baud được đặt bằng tốc độ báo tràn của Timer 1.

Tạo xung nhịp và đồng bộ hóa các thanh ghi dịch của port nối tiếp trong các chế độ 1, 2 và 3 được thiết lập bằng bộ đếm 4 bit chia cho 16, ngõ ra là xung nhịp tốc độ baud. Ngõ vào của bộ đếm này được chọn qua phần mềm



*Tạo xung nhịp port nối tiếp.*

Truyền dữ liệu (phát) được khởi động bằng cách ghi vào SBUF, nhưng vẫn chưa thật sự bắt đầu chạy cho đến khi sự thay thế kế tiếp của bộ đếm chia cho 16 cung cấp tốc độ baud cổng nối tiếp. Dữ liệu được dịch ra ngoài trên đường TXD bắt đầu bằng bit start, theo sau là 8 bit dữ liệu và sau cùng là bit stop. Độ rộng (theo thời gian của mỗi bit) là nghịch đảo của tốc độ baud được lập trình trong timer. Cờ ngắt phát (TI) được đặt lên 1 khi xuất hiện bit stop trên TXD.



*Đặt cờ TI port nối tiếp.*

Việc thu dữ liệu được khởi động bằng một chuyển trạng thái từ 1 xuống 0 trên RXD. Bộ đếm 16 tức thời được xóa để đồng bộ số đếm với luồng bit đến. Luồng bit đến được lấy mẫu giữa 16 lần đếm.

Bộ thu sẽ phát hiện được bit start sai bằng cách yêu cầu trạng thái 0 ở (bit start) ở lần đếm thứ 8 sau khi có chuyển trạng thái từ 1 xuống 0 đầu tiên. Nếu điều này không xảy ra, người ta giả sử là bộ thu được kích bởi nhiễu chứ không phải do một ký tự hợp lệ. Bộ thu được reset và quay về trạng thái nghỉ (idle), tìm kiếm (đợi) chuyển trạng thái từ 1 xuống 0 kế.

Giả sử đã phát hiện được bit start hợp lệ, thì tiếp tục thu ký tự. Bit start được bỏ qua và 8 bit dữ liệu được đưa vào thanh ghi dịch cổng nối tiếp theo xung nhịp. Khi đã có được tất cả 8 bit, điều sau đây xảy ra :

1. Bit thứ 9 (bit stop) được chốt vào RB8 trong SCON.
2. SBUF được nạp với 8 bit dữ liệu.
3. Cờ ngắt bộ thu (RI) được đặt lên 1.

Tuy nhiên, những điều này chỉ xảy ra nếu đã có những điều kiện sau :

1. RI = 0
2. SM2 = 1 và bit stop thu được là 1, hoặc SM2 = 0.

Đòi hỏi RI = 0 để bảo đảm là phần mềm đã đọc ký tự trước (và RI được xóa). Điều kiện thứ hai hơi phức tạp nhưng chỉ áp dụng trong chế độ truyền thông đa xử lý. Điều đó hàm ý là "không đặt RI lên 1 trong chế độ truyền thông đa xử lý khi bit dữ liệu thứ 9 là 0).

### 6.3.3 UART 9 bit với tốc độ baud cố định (chế độ 2).

Khi SM1 = 1 và SM0 = 0, cổng nối tiếp làm việc ở chế độ 2, như một UART 9 bit có tốc độ baud cố định. 11 bit sẽ được phát hoặc thu : 1 bit start, 8 bit dữ liệu, bit dữ liệu thứ 9 có thể lập trình được và 1 bit stop. Khi phát, bit thứ 9 là bất cứ gì đã được đưa vào TB8 trong SCON (có thể là bit parity). Khi thu, bit thứ 9 thu được sẽ ở trong RB8. Tốc độ baud ở chế độ 2 là 1/32 hoặc 1/16 tần số dao động trên chip.

### 6.3.4 UART 9 bit với tốc độ baud thay đổi được.

Chế độ này giống như chế độ 2 ngoại trừ tốc độ baud có thể lập trình được và được cung cấp bởi timer. Thật ra, các chế độ 1, 2 và 3 rất giống nhau. Các khác biệt là ở tốc độ baud (cố định trong chế độ 2, thay đổi trong các chế độ 1 và 3) và ở số bit dữ liệu (8 trong chế độ 1, 9 trong các chế độ 2 và 3).

## 6.4 Khởi động và truy xuất các thanh ghi cổng nối tiếp.

### 6.4.1 Cho phép thu.

Bit cho phép bộ thu (REN = Receiver Enable) trong SCON phải được đặt lên 1 bằng phần mềm để cho phép thu các ký tự. Thông thường thực hiện việc này ở đầu chương trình khi khởi động cổng nối tiếp, timer, ... Có thể thực hiện việc này theo hai cách. Lệnh SETB REN sẽ đặt REN lên 1, hoặc lệnh MOV SCON, #xxh1xxhB sẽ đặt REN lên 1 và đặt hoặc xóa các bit khác trong SCON khi cần (Các x phải là 0 hoặc 2 để đặt chế độ làm việc).

### 6.4.2 Bit dữ liệu thứ 9.

---

Bit dữ liệu thứ 9 cần phát trong các chế độ 2 và 3 phải được nạp vào trong TB8 bằng phần mềm. Bit dữ liệu thứ 9 thu được đặt ở RB8. Phần mềm có thể cần hoặc không cần bit dữ liệu thứ 9, phụ thuộc vào đặc tính kỹ thuật của thiết bị nối tiếp sử dụng. (Bit dữ liệu thứ 9 cũng đóng một vai trò quan trọng trong truyền thông đa xử lý).

#### 6.4.3 Thêm 1 bit parity.

Thường sử dụng bit dữ liệu thứ 9 để thêm parity vào ký tự. Như đã xét ở các chương trước, bit P trong từ trạng thái chương trình (PSW) được đặt lên 1 hoặc bị xóa mỗi chu kỳ máy để thiết lập kiểm tra chẵn với 8 bit trong thanh ghi tích lũy. Ví dụ, nếu truyền thông cần 8 bit dữ liệu cộng thêm kiểm tra chẵn, có thể sử dụng các lệnh sau để phát 8 bit trong thanh ghi tích lũy với kiểm tra chẵn thêm vào bit thứ 9 :

MOV C, P ; Đặt bit parity chẵn vào TB8

MOV TB8, C ; nó trở thành bit dữ liệu thứ 9

MOV SBUF, A ; Chuyển 8 bit từ ACC vào SBUF.

Nếu cần parity lẻ thì sửa các lệnh lại như sau :

MOV C, P ; Đặt bit parity chẵn vào cờ C

CPL C ; Đổi sang parity lẻ

MOV TB8, C

MOV SBUF, A

Dĩ nhiên, việc sử dụng parity không bị giới hạn ở các chế độ 2 và 3. Ở chế độ 1, 8 bit dữ liệu được truyền đi có thể bao gồm 7 bit dữ liệu cộng thêm bit parity. Để truyền mã ASCII 7 bit với parity chẵn ở bit 8, có thể sử dụng các lệnh sau :

CLR ACC.7 ; bảo đảm MSB được xóa

PMOV C, P ; parity chẵn ở trong

MOV ACC.7, C ; Đặt parity chẵn vào MSB

MOV SBUF, A ; Gửi ký tự đi 7 bit dữ liệu cộng parity chẵn.

#### 6.4.4 Các cờ ngắt.

Hai cờ ngắt thu và phát (RI và TI) trong SCON đóng một vai trò quan trọng trong truyền thông nối tiếp dùng 8051/8031. Cả hai bit được đặt lên 1 bằng phần cứng, nhưng phải được xóa bằng phần mềm.

Ví dụ, thường RI được đặt lên 1 khi kết thúc việc thu ký tự và báo "bộ đệm thu tràn". Điều kiện này có thể được kiểm tra trong phần mềm hoặc có thể được lập trình để gây ra một ngắt. Nếu phần mềm muốn nhập một ký tự từ thiết bị được nối vào cổng nối tiếp (có thể là thiết bị đầu cuối hiển thị video), nó phải đợi cho đến khi RI được đặt lên 1, rồi xóa RI và đọc ký tự từ SBUF. Chương trình như

---

sau :

WAIT : JNB RI, WAIT ; Kiểm tra RI cho đến khi nó = 1

CLR RI ; Xóa RI

MOV A, SBUF ; Đọc ký tự

TI được đặt lên 1 ở cuối lúc phát ký tự và báo " bộ đếm phát trống". Nếu phần mềm muốn gửi một ký tự đến một thiết bị được nối vào cổng nối tiếp, trước hết nó phải kiểm tra xem cổng nối tiếp sẵn sàng chưa. Nói cách khác, nếu ký tự trước đã được gửi đi, đợi cho đến khi việc truyền dữ liệu hoàn tất trước khi gửi ký tự kế. Các lệnh sau sẽ truyền ký tự trong thanh ghi tích lũy:

WAIT : JNB TI, WAIT ; Kiểm tra TI cho đến khi nó bằng 1

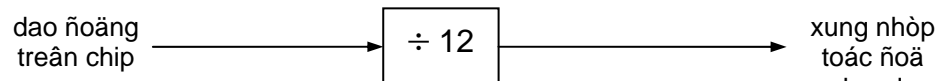
CLR TI ; Xóa TI

MOV SBUF, A ; Gửi ký tự đi.

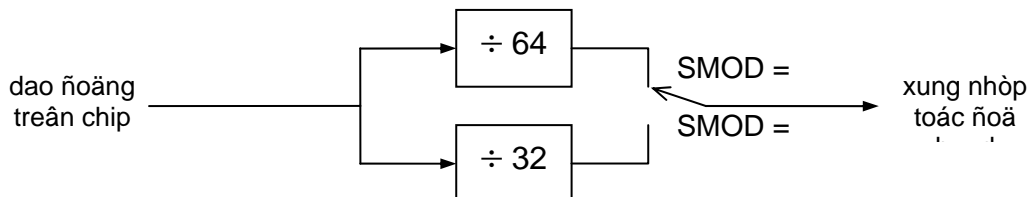
Các đoạn chương trình trên là một phần của các hàm nhập và xuất ký tự chuẩn.

## 6.5 Tốc độ baud port nối tiếp.

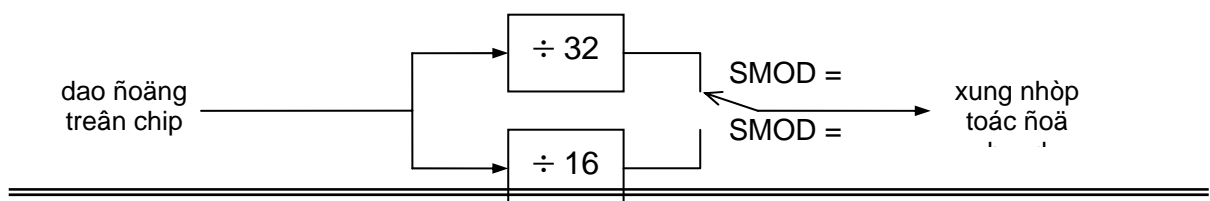
Như đã nói, tốc độ baud cố định ở các chế độ 0 và 2. Trong chế độ 0, nó luôn luôn là tần số dao động trên chip được chia cho 12. Thông thường thạch anh ấn định tần số dao động trên chip của 8051/8031, nhưng cũng có thể sử dụng nguồn xung nhịp khác. Giả sử với tần số dao động danh định là 12 MHz, thì tốc độ baud chế độ 0 là 1 MHz.



a) chế độ 0



b) chế độ 2



c) chế độ 1 và 3.

*Các nguồn tạo xung nhịp cho port nối tiếp.*

Mặc nhiên sau khi reset hệ thống, tốc độ baud chế độ 2 là tần số bộ dao động chia cho 64. Tốc độ baud cũng bị ảnh hưởng bởi một bit trong thanh ghi điều khiển nguồn cung cấp (PCON). Bit 7 của PCON là bit SMOD. Đặt bit SMOD lên 1 làm gấp đôi tốc độ baud trong các chế độ 1, 2 và 3. Trong chế độ 2, tốc độ baud có thể bị gấp đôi từ giá trị mặc nhiên của 1/64 tần số dao động (SMOD = 0) đến 1/32 tần số dao động (SMOD = 1).

Vì PCON không được định địa chỉ theo bit, nên để đặt bit SMOD lên 1 cần phải theo các lệnh sau :

MOV A, PCON ; Lấy giá trị hiện thời của PCON

SETB ACC.7 ; Đặt bit 7 (SMOD) lên 1

MOV PCON, A ; Ghi giá trị ngược về PCON.

Các tốc độ baud trong các chế độ 1 và 3 được xác định bằng tốc độ tràn của Timer 1. Vì timer hoạt động ở tần số tương đối cao, tràn timer được chia thêm cho 32 (16 nếu SMOD = 1) trước khi cung cấp xung nhịp tốc độ baud cho port nối tiếp.

**\* Sử dụng Timer 1 làm xung nhịp tốc độ baud**

Xét 8051, cách thông dụng để tạo tốc độ baud là khởi động TMOD cho chế độ 8 bit tự động nạp lại (chế độ 2) và đặt giá trị nạp lại đúng vào TH1 để cho tốc độ tràn đúng với tốc độ baud. TMOD được khởi động như sau :

MOV TMOD, #0010xxxxB

Các x là các bit 1 hoặc 0 cần cho timer.

Cũng có thể đạt được các tốc độ baud thấp bằng cách sử dụng timer chế độ 1 với TMOD = 0001xxxxB. Tuy nhiên, tốn thêm phần mềm vì các thanh ghi TH1/TL1 phải được khởi động lại sau mỗi lần tràn. Việc này sẽ được thực hiện trong chương trình phục vụ ngắt. Một chọn lựa khác là cấp xung nhịp cho Timer 1 từ ngoài dùng T1(P3.5). Và luôn luôn tốc độ baud là tốc độ tràn của Timer 1 được chia cho 32 (hoặc cho 16, nếu SMOD = 1).

Công thức tổng quát để xác định tốc độ baud trong các chế độ 1 và 3 là :

Tốc độ baud = Tốc độ tràn của Timer 1 ÷ 32.

Ví dụ, muốn làm việc với tốc độ baud là 1200 baud, thì tốc độ tràn của Timer 1 phải là :

$$1200 \times 32 = 38.4 \text{ KHz.}$$

Nếu dùng thạch anh 12 MHz, Timer 1 được cấp xung nhịp 1 MHz hay 1000 KHz. Vì tốc độ tràn của Timer 1 là 38.4 KHz và timer được cấp xung nhịp 1000 KHz, thì cần tràn sau  $1000 \div 38.4$

---



= 26.04 xung nhịp (làm tròn là 26). Vì timer đếm lên và tràn xảy ra khi có sự thay đổi từ FFH xuống 00H ở số đếm. Như vậy giá trị đúng cần nạp vào TH1 là -26. Cách dễ nhất để đặt giá trị nạp lại vào TH1 là :

```
MOV TH1, #-26
```

Trình hợp dịch sẽ thực hiện chuyển đổi cần thiết. Trong trường hợp này -26 được chuyển thành 0E6H. Như vậy, lệnh trên hoàn toàn giống với lệnh :

```
MOV TH1, # 0E6H
```

Do việc làm tròn nên có sai số nhỏ trong tốc độ baud. Tổng quát thì cho phép dung sai 5% trong truyền thông bất đồng bộ (start/stop). Có thể có được tốc độ baud chính xác nếu dùng thạch anh 11.059 MHz. Bảng sau đây tóm tắt các giá trị nạp lại cho các tốc độ baud thông dụng nhất, dùng thạch anh 12 MHz hoặc 11.059 MHz :

Toác ñoä	Taàn soá thaích anh	SMOD	Giaù trò naïp laïi vaøo TH1	Toác ñoä baud thaät	Sai soá
9600	12.000 MHz	1	-7 (F9H)	8923	7%
2400	12.000 MHz	0	-13 (F3H)	2404	0.16%
1200	12.000 MHz	0	-26 (E6H)	1202	0.16%
19200	11.059 MHz	1	-3 (FDH)	19200	0
9600	11.059 MHz	0	-3 (FDH)	9600	0
2400	11.059 MHz	0	-12 (F4H)	2400	0
1200	11.059 MHz	0	-24 (E8H)	1200	0

*Bảng tóm tắt tốc độ baud.*