

CHƯƠNG 5

HOẠT ĐỘNG CỦA PORT NỐI TIẾP (SERIAL PORT)

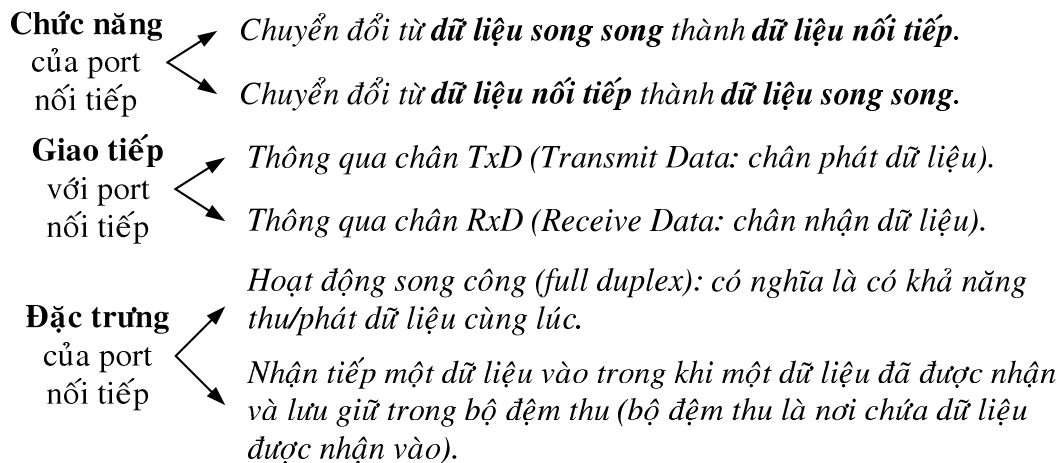
I. MỞ ĐẦU:

Máy tính truyền dữ liệu theo hai phương pháp: truyền dữ liệu song song và truyền dữ liệu nối tiếp.

- **Truyền song song:** Sử dụng nhiều dây dẫn để truyền dữ liệu giữa các thiết bị có khoảng cách gần nhau (*khoảng vài mét*). Phương pháp này cho phép truyền dữ liệu với tốc độ cao nhờ sử dụng nhiều dây dẫn để truyền dữ liệu đồng thời nên tại một thời điểm có thể truyền được nhiều bit thông tin nhưng khoảng cách truyền thì có nhiều hạn chế.

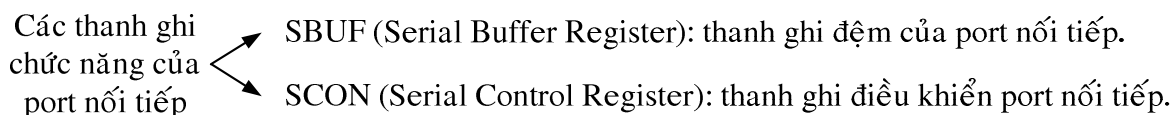
- **Truyền nối tiếp:** Sử dụng một dây dẫn để truyền dữ liệu (*một dây phát đi và một dây thu về*) giữa các thiết bị có khoảng cách xa nhau (*khoảng vài trăm mét trở lên*). Phương pháp này sẽ truyền dữ liệu với tốc độ chậm hơn (*so với phương pháp truyền song song*) vì chỉ sử dụng một dây dẫn để truyền dữ liệu nên tại một thời điểm chỉ có thể truyền được một bit thông tin nhưng khoảng cách truyền thì không bị hạn chế như ở phương pháp song song.

Chip 8051 có một port nối tiếp (serial port) với các tính năng như sau:



- **Lưu ý:** Ở trường hợp đặc trưng thứ hai thì dữ liệu thứ nhất sẽ không bị mất nếu CPU đọc xong dữ liệu thứ nhất trước khi dữ liệu thứ hai được nhận đầy đủ.

Các thanh ghi chức năng đặc biệt của port nối tiếp:



Đại lượng đặc trưng cho tốc độ truyền dữ liệu nhanh hay chậm là **tốc độ baud (baud rate)** hay còn gọi là tần số hoạt động của port nối tiếp có thể là giá trị cố định hay thay đổi tùy theo yêu cầu của người lập trình. Khi chế độ tốc độ baud thay đổi được sử dụng, bộ định thời 1 cung cấp xung clock tốc độ baud và ta phải lập trình sao cho phù hợp. Ở phiên bản chip 8031/8052, bộ định thời 2 cũng có thể được lập trình để cung cấp xung clock tốc độ baud.

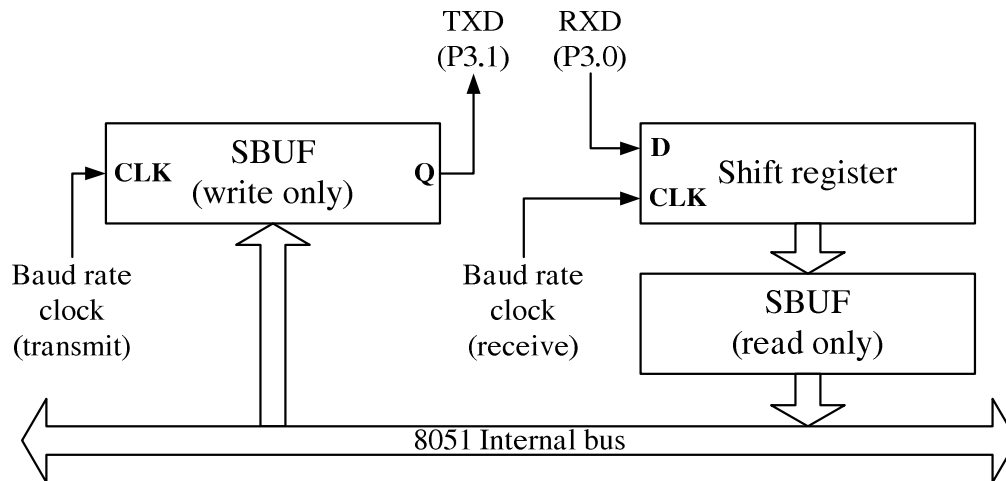
II. THANH GHI ĐỆM PORT NỐI TIẾP (SBUF):

Thanh ghi SBUF (Serial Buffer Register): được dùng để lưu giữ dữ liệu cần phát đi và dữ liệu đã nhận được. Việc ghi dữ liệu vào thanh ghi SBUF sẽ nạp dữ liệu để phát đi và việc đọc dữ liệu từ thanh ghi SBUF sẽ truy xuất dữ liệu đã thu được.

Thanh ghi SBUF bao gồm 2 thanh ghi:

- ⇒ **Thanh ghi phát (bộ đệm phát)**: dùng để lưu giữ dữ liệu cần phát đi.
- ⇒ **Thanh ghi thu (bộ đệm thu)**: dùng để lưu giữ dữ liệu đã nhận được.

Cấu trúc của thanh ghi SBUF:



Write only: chỉ ghi.

Read only: chỉ đọc.

Shift register: thanh ghi dịch bit.

Baud rate clock (transmit): xung clock tốc độ baud (phát).

Baud rate clock (receive): xung clock tốc độ baud (thu).

8051 internal bus: bus nội của 8051.

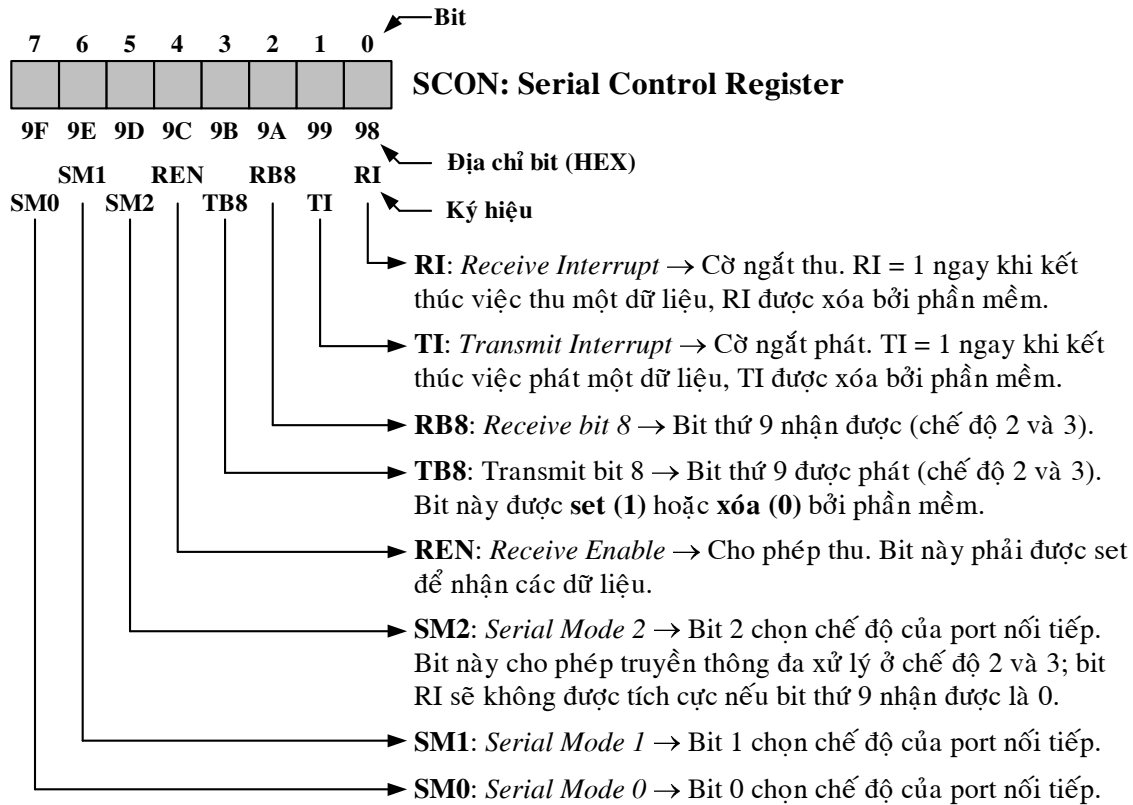
- Ví dụ: Các lệnh ghi dữ liệu vào SBUF và đọc dữ liệu từ SBUF.

MOV	SBUF, #45H	;Phát giá trị 45H qua port nối tiếp.
MOV	SBUF, #"D"	;Phát giá trị 44H qua port nối tiếp.
MOV	SBUF, A	;Phát nội dung của A qua port nối tiếp.
MOV	A, SBUF	;Đọc dữ liệu thu được từ port nối tiếp.

III. THANH GHI ĐIỀU KHIỂN PORT NỐI TIẾP (SCON):

Thanh ghi SCON (Serial Control Register): chứa các bit dùng để điều khiển chế độ hoạt động và báo trạng thái của port nối tiếp.

Cấu trúc của thanh ghi SCON:



Các chế độ của port nối tiếp:

SM0	SM1	Chế độ	Mô tả	Tốc độ baud
0	0	→ 0	→ Thanh ghi dịch.	→ Cố định ($f_{OSC} / 12$).
0	1	→ 1	→ UART 8 bit.	→ Thay đổi (thiết lập bởi Timer 1).
1	0	→ 2	→ UART 9 bit.	→ Cố định ($f_{OSC} / 32$ hoặc $f_{OSC} / 64$).
1	1	→ 3	→ UART 9 bit.	→ Thay đổi (thiết lập bởi Timer 1).

Trước khi sử dụng port nối tiếp cần phải:

Qui định chế độ của port nối tiếp $\Rightarrow$ PHỤ THUỘC Thanh ghi SCON

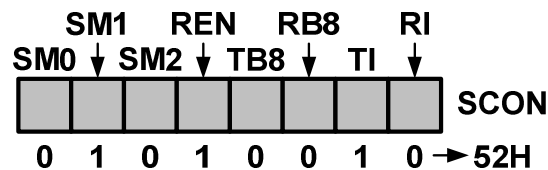
Ví dụ: Khởi động port nối tiếp ở chế độ 1, cho phép port thu dữ liệu từ chân RxD và sẵn sàng phát dữ liệu từ chân TxD.

Giải

Ta dùng lệnh:

MOV SCON, #52H

Giải thích:



SM0 = 0, SM1 = 1 → cho phép port hoạt động ở chế độ 1.

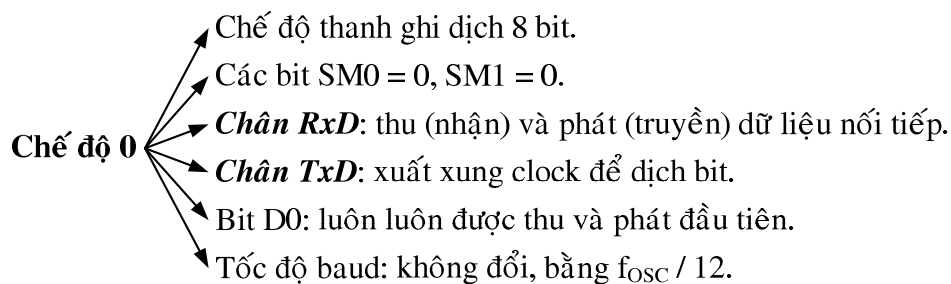
REN = 1 → cho phép port nối tiếp được phép thu dữ liệu.

TI = 1 → chuẩn bị port nối tiếp sẵn sàng phát dữ liệu qua chân TxD.

RI = 0 → chuẩn bị port nối tiếp sẵn sàng thu dữ liệu qua chân RxD.

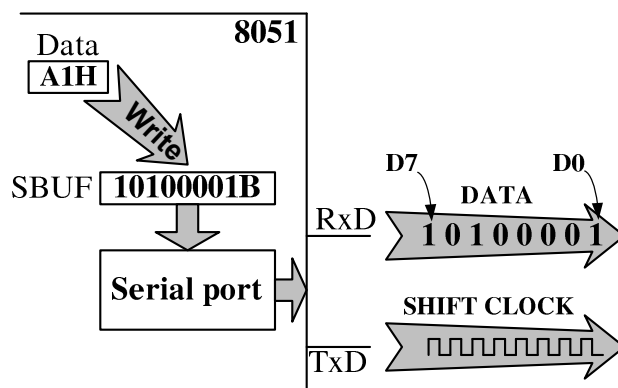
IV. CÁC CHẾ ĐỘ HOẠT ĐỘNG CỦA PORT NỐI TIẾP:

1. Chế độ 0 – Thanh ghi dịch 8 bit:

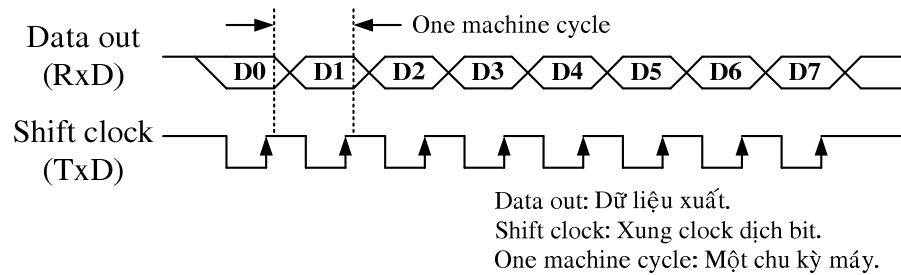


Quá trình phát dữ liệu:

- Quá trình khởi động:** Ghi dữ liệu cần phát vào SBUF ⇒ **Việc phát dữ liệu bắt đầu:** Dữ liệu từ SBUF được dịch ra chân RxD đồng thời với các xung clock dịch bit được gửi ra chân TxD (mỗi bit được truyền đi trên chân RxD trong 1 chu kỳ máy).

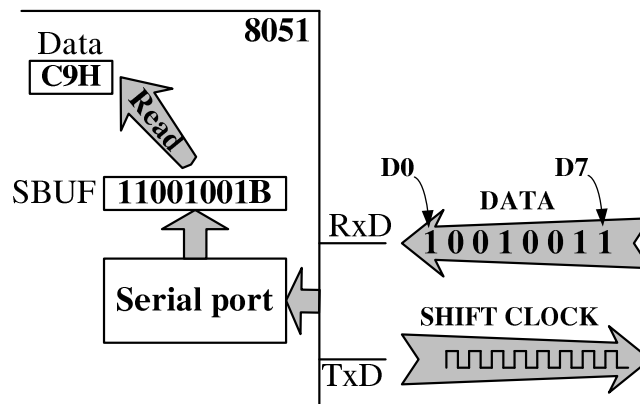


- Giản đồ thời gian phát dữ liệu:

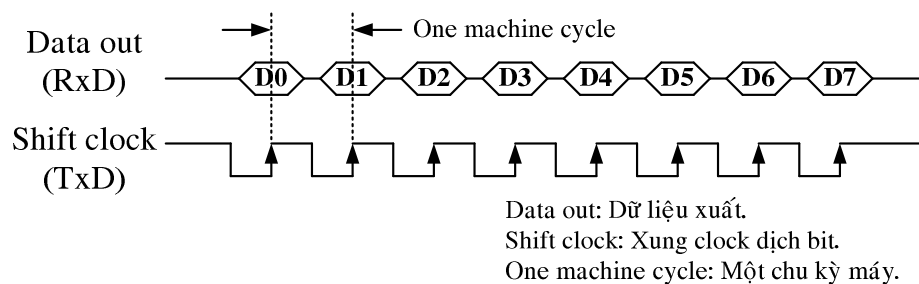


Quá trình thu dữ liệu:

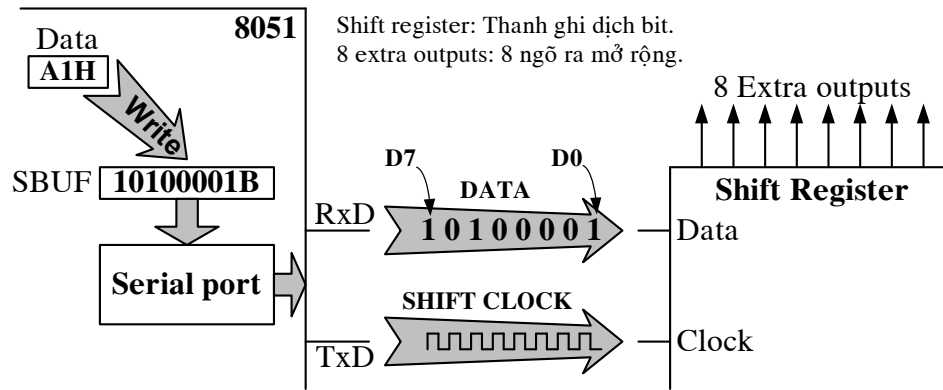
- Quá trình khởi động:** Set bit cho phép thu ($REN=1$) → Xóa cờ ngắt thu ($RI=0$) ⇒ **Việc thu dữ liệu bắt đầu:** Các xung clock dịch bit được gửi ra chân TxD và dữ liệu từ thiết bị bên ngoài được dịch vào chân RxD bởi các xung clock dịch bit này (việc dịch dữ liệu vào chân RxD xảy ra ở cạnh lên của xung clock dịch bit).



- Giản đồ thời gian thu dữ liệu:



Ứng dụng: Một ứng dụng khả thi của chế độ 0 (chế độ thanh ghi dịch bit) là mở rộng thêm các ngõ ra cho chip 8051. Một vi mạch thanh ghi dịch nối tiếp – song song có thể được nối với các chân TxD và RxD của chip 8051 để cung cấp thêm 8 đường xuất (xem hình vẽ bên dưới). Các thanh ghi dịch bit khác có thể ghép *cascade* với thanh ghi dịch bit đầu tiên để mở rộng thêm nữa.



Chế độ thanh ghi dịch bit của port nối tiếp.

2. Chế độ 1 – UART 8 bit có tốc độ baud thay đổi:

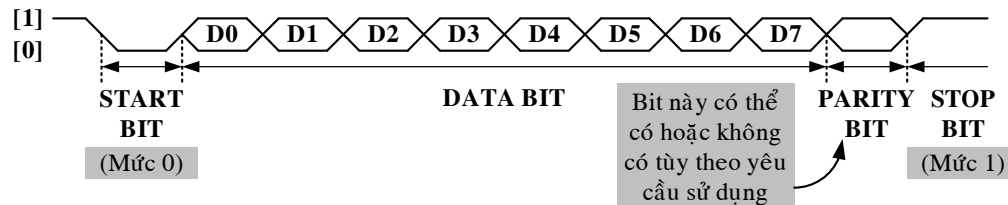
Trong chế độ 1, port nối tiếp của 8051 hoạt động như một bộ thu phát không đồng bộ 8 bit có tốc độ baud thay đổi (UART - Universal Asynchronous Receiver Transmitter).

Nhiệm vụ UART

- Biến đổi dữ liệu phát từ song song thành nối tiếp.
- Biến đổi dữ liệu thu từ nối tiếp thành song song.

UART là một bộ thu phát dữ liệu nối tiếp với mỗi ký tự dữ liệu được đứng trước bởi một bit START (logic 0) và được đứng sau bởi một bit STOP (logic 1). Thành thạo, một bit chẵn lẻ (Parity bit) được chèn vào giữa bit dữ liệu sau cùng và bit stop. Hoạt động chủ yếu của UART là biến đổi dữ liệu phát từ song song thành nối tiếp và biến đổi dữ liệu thu từ nối tiếp thành song song.

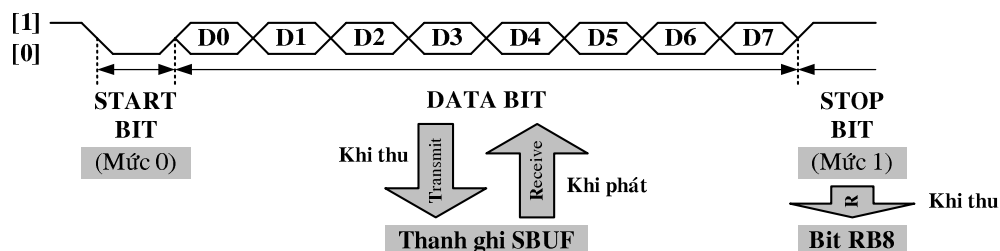
Hình vẽ khung dạng dữ liệu khi được sử dụng ở chế độ UART:



Chế độ 1

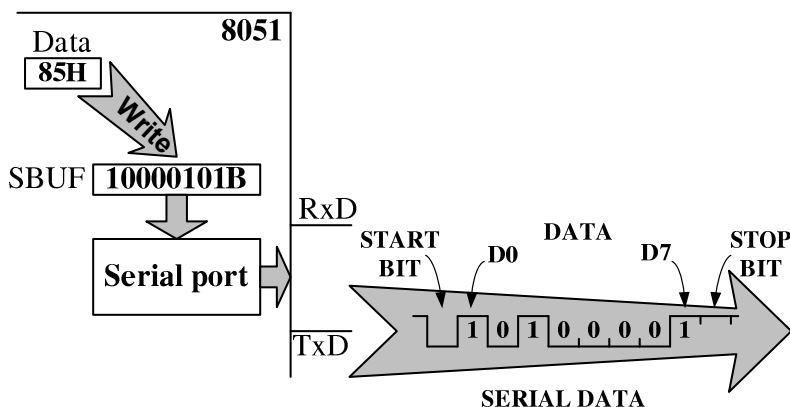
- Chế độ UART 8 bit.
- Các bit SM0 = 0, SM1 = 1.
- Chân RxD: thu (nhận) dữ liệu nối tiếp.
- Chân TxD: phát (truyền) dữ liệu nối tiếp.
- Dữ liệu được truyền kèm theo Start bit và Stop bit.
- Bit D0: luôn luôn được thu và phát đầu tiên.
- Tốc độ baud: thay đổi, được thiết lập bởi Timer 1.

Khuông dạng của một dữ liệu khi sử dụng chế độ UART 8 bit:



Quá trình phát dữ liệu:

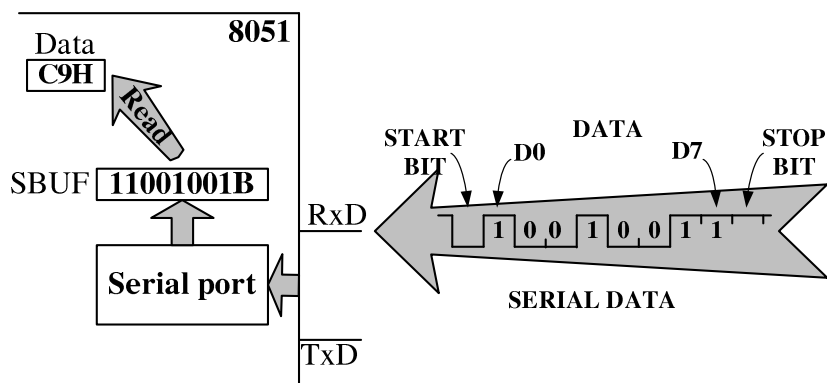
- **Quá trình khởi động:** Ghi dữ liệu cần phát vào SBUF $\Rightarrow$ Việc phát dữ liệu bắt đầu: Dữ liệu từ SBUF được dịch ra chân TxD (theo thứ tự: **Start bit** $\rightarrow$ 8 bit data (D0 .. D7) $\rightarrow$ **Stop bit**) $\rightarrow$ cờ TI=1.



- Tốc độ baud: do người lập trình thiết lập và được qui định bởi tốc độ tràn của Timer 1.
- Thời gian của 1 bit trên đường truyền: bằng nghịch đảo của tốc độ baud ($1 / \text{Baud rate}$).
- Cờ ngắt phát TI = 1: khi bit stop được xuất hiện trên chân TxD.

Quá trình thu dữ liệu:

- **Quá trình khởi động:** Một sự chuyển trạng thái từ mức 1 xuống mức 0 tại chân RxD (tức xuất hiện bit Start) $\Rightarrow$ Việc thu dữ liệu bắt đầu: 8 bit dữ liệu được dịch vào trong SBUF (theo thứ tự: D0 $\rightarrow$ D1 $\rightarrow$... $\rightarrow$ D7) $\rightarrow$ Stop bit (bit thứ 9) được đưa vào bit RB8 (thuộc thanh ghi SCON) $\rightarrow$ cờ RI=1.



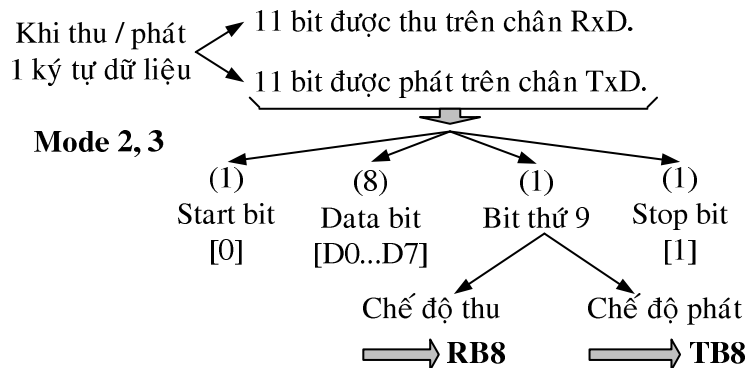
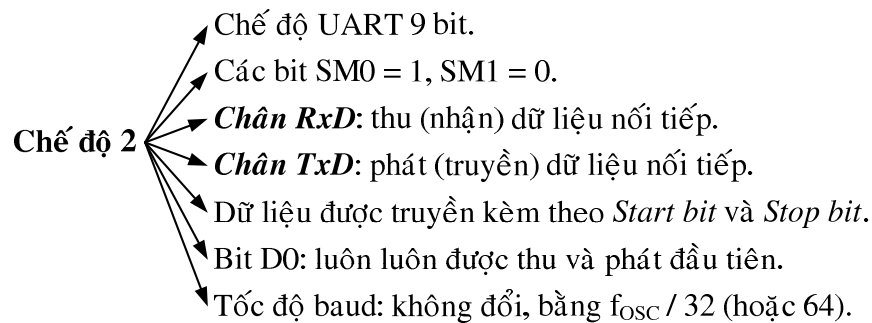
- Tốc độ baud: do người lập trình thiết lập và được qui định bởi tốc độ tràn của Timer 1.
- Hai điều kiện bắt buộc để thực hiện quá trình thu dữ liệu như trên:
 - **RI = 0** $\rightarrow$ Yêu cầu này có nghĩa là chip 8051 đã đọc xong dữ liệu trước đó và xóa cờ RI..
 - **(SM2 = 1 và Stop bit = 1) hoặc SM2 = 0** $\rightarrow$ chỉ áp dụng trong chế độ truyền thông đa xử lý. Yêu cầu này có nghĩa là không set cờ RI bằng 1 trong chế độ truyền thông đa xử lý khi bit dữ liệu thứ 9 là 0.
- Cờ ngắt thu RI = 1: khi 8 bit dữ liệu đã được nạp vào SBUF.

Lưu ý: Trường hợp các tín hiệu nhiễu xuất hiện trên đường truyền (làm cho đường truyền xuất hiện mức thấp) dẫn đến làm cho bộ thu nhận dạng sai, cho đó là sự xuất hiện của START bit (logic 0) và tiến hành thực hiện quá trình thu dữ liệu, từ đó dẫn đến kết quả nhận vào sẽ không đúng. Để tránh điều này xảy ra thì khi đường truyền có sự chuyển trạng thái từ 1 xuống 0, bộ thu yêu cầu mức 0 này phải

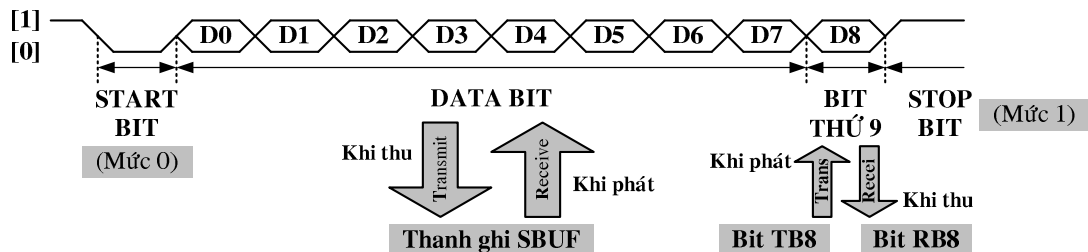
được duy trì trên đường truyền trong một khoảng thời gian xác định. Nếu không đảm bảo được như thế, bộ thu được giả sử rằng đã nhận được nhiều thay vì nhận được START bit hợp lệ. Lúc đó bộ thu sẽ được thiết lập lại, quay về trạng thái nghỉ và chờ sự chuyển trạng thái từ 1 xuống 0 kế tiếp trên đường truyền.

3. Chế độ 2 – UART 9 bit có tốc độ baud cố định:

(Tương tự như UART 8 bit, chỉ khác ở số bit dữ liệu là 9 bit)

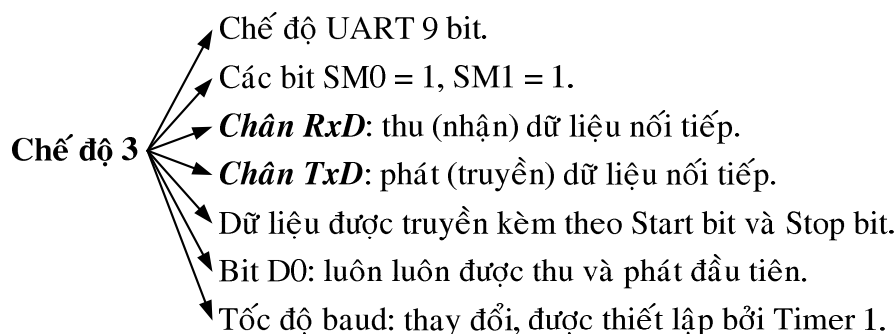


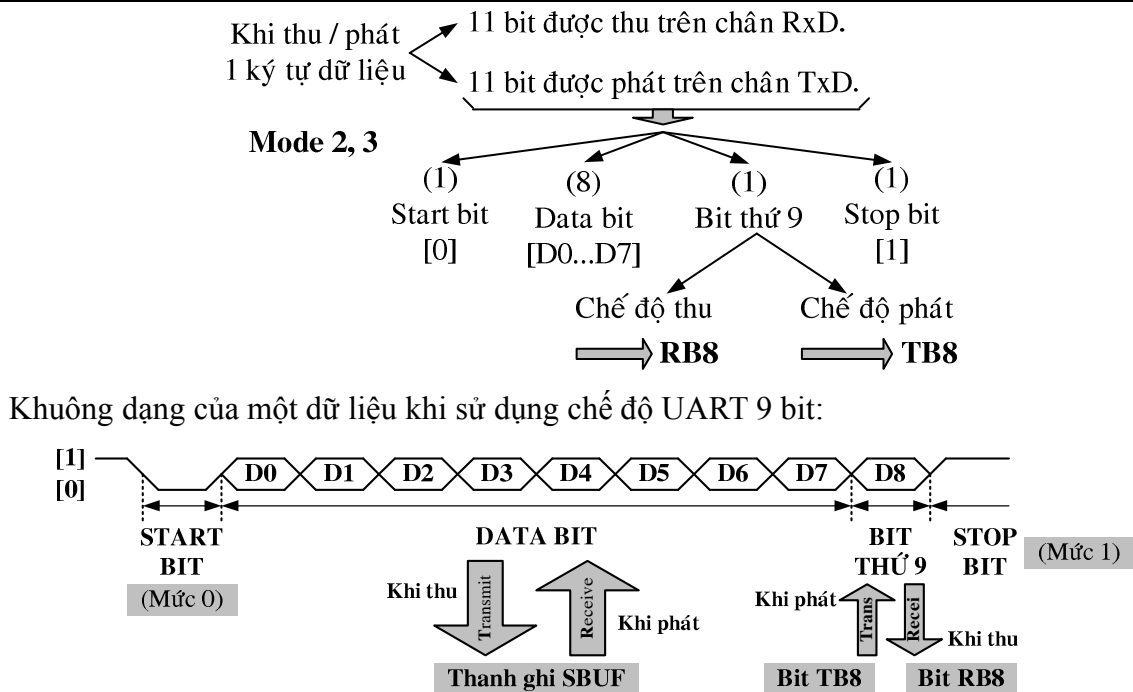
Khuông dạng của một dữ liệu khi sử dụng chế độ UART 9 bit:



4. Chế độ 3 – UART 9 bit có tốc độ baud thay đổi:

(Tương tự như UART 9 bit, chỉ khác ở tốc độ baud có thể thay đổi)





V. KHỞI ĐỘNG VÀ TRUY XUẤT CÁC THANH GHI:

1. Bit cho phép thu (nhận) dữ liệu (REN: Receive Enable):

- Công dụng:** dùng để cho phép (hoặc không cho phép) nhận các ký tự dữ liệu.

REN = 1: Cho phép nhận dữ liệu $\Rightarrow$ lệnh thực hiện: **SETB REN**

REN = 0: Không cho phép dữ liệu $\Rightarrow$ lệnh thực hiện: **CLR REN**

2. Bit dữ liệu thứ 9:

- Công dụng:** tùy thuộc vào đặc tính kỹ thuật của thiết bị nối tiếp mà có thể yêu cầu hoặc không yêu cầu bit dữ liệu thứ 9.

Khi phát dữ liệu: bit dữ liệu thứ 9 *phải được nạp vào bit TB8* của SCON trước khi phát đi.

Khi thu dữ liệu: bit dữ liệu thứ 9 *sẽ được nạp vào bit RB8* của SCON sau khi thu xong.

3. Bit kiểm tra chẵn / lẻ (P: Parity):

- Công dụng:** Trong chip 8051 thì bit Parity được dùng để thiết lập việc **kiểm tra chẵn** cho 8 bit dữ liệu chứa trong thanh ghi A (thường dùng để kiểm tra lỗi khi truyền dữ liệu).

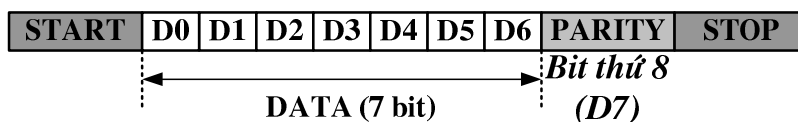
P = 1 $\Rightarrow$ Số lượng bit 1 trong thanh ghi A là số lẻ.

P = 0 $\Rightarrow$ Số lượng bit 1 trong thanh ghi A là số chẵn.

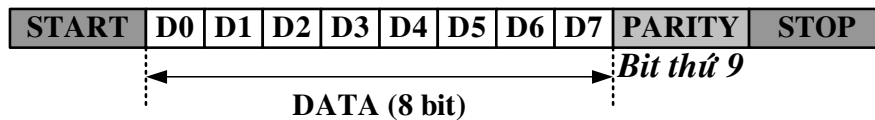
hoặc

Số lượng bit 1 trong thanh ghi A và bit P là một **số chẵn**.

Ở chế độ 1 (UART 8 bit) thì bit chẵn/lẻ do chip 8051 tạo ra có thể được thêm vào tại bit thứ 8 (vị trí D7) và khi đó ta chỉ có thể truyền dữ liệu chỉ có 7 bit.



Ở chế độ 2, 3 (UART 9 bit) thì bit chẵn/lẻ do chip 8051 tạo ra có thể được thêm vào tại bit thứ 9 (nghĩa là thêm vào bit TB8 của SCON) và khi đó ta có thể truyền dữ liệu có 8 bit.



Ví dụ: Truyền dữ liệu (chế độ 2, 3 – UART 9 bit) chứa trong thanh ghi A thông qua port nối tiếp với yêu cầu truyền 8 bit dữ liệu + 1 bit kiểm tra chẵn (bit P).

Chuỗi lệnh thực hiện:

```
MOV    C, P           ;Chuyển bit kiểm tra chẵn (bit P) vào TB8 và
MOV    TB8, C          ;bit này trở thành bit thứ 9.
MOV    SBUF, A         ;Truyền 8 bit dữ liệu trong A thông qua port.
```

Ví dụ: Truyền dữ liệu (chế độ 2, 3 – UART 9 bit) chứa trong thanh ghi A thông qua port nối tiếp với yêu cầu truyền 8 bit dữ liệu + 1 bit kiểm tra lẻ (lấy bù bit P).

Chuỗi lệnh thực hiện:

```
MOV    C, P           ;Biến đổi bit kiểm tra chẵn (bit P) thành bit
CPL    C              ;kiểm tra lẻ, chuyển bit kiểm tra lẻ vào TB8 và
MOV    TB8, C          ;bit này trở thành bit thứ 9.
MOV    SBUF, A         ;Truyền 8 bit dữ liệu trong A thông qua port.
```

Ví dụ: Truyền dữ liệu (chế độ 1 – UART 8 bit) chứa trong thanh ghi A thông qua port nối tiếp với yêu cầu truyền 7 bit dữ liệu + 1 bit kiểm tra chẵn (bit P).

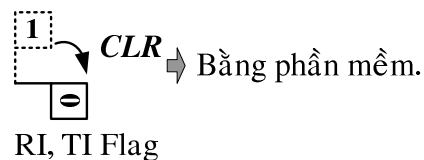
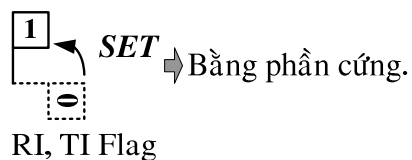
Chuỗi lệnh thực hiện:

```
CLR    ACC.7          ;Xoá bit thứ 8 (D7) trong thanh ghi A.
MOV    C, P           ;Sao chép bit kiểm tra chẵn vào C.
MOV    ACC.7, C        ;Đặt bit kiểm tra chẵn vào bit thứ 8 trong A.
MOV    SBUF, A         ;Truyền 7 bit dữ liệu cộng bit kiểm tra chẵn.
```

4. Các cờ ngắt của port nối tiếp:

Các cờ ngắt của port nối tiếp

- RI: cờ ngắt thu. **RI=1** khi kết thúc việc thu dữ liệu và chỉ ra rằng bộ đệm thu đã đầy.
- TI: cờ ngắt phát. **TI=1** khi kết thúc việc phát dữ liệu và chỉ ra rằng bộ đệm phát đã rỗng.

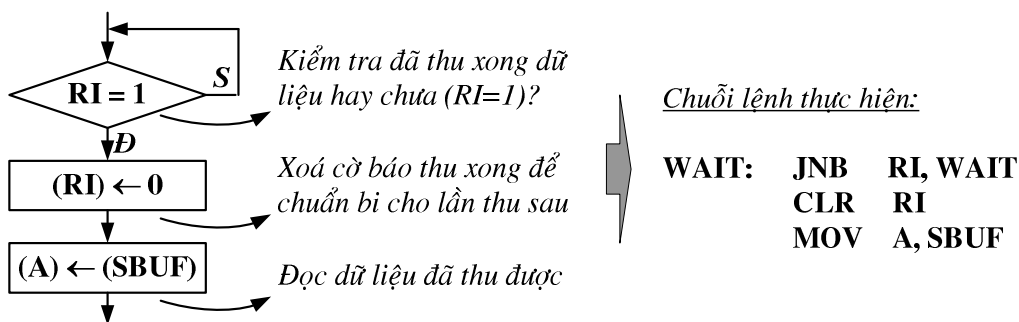


Từ phần trình bày trên đây, ta có thể thấy rằng:

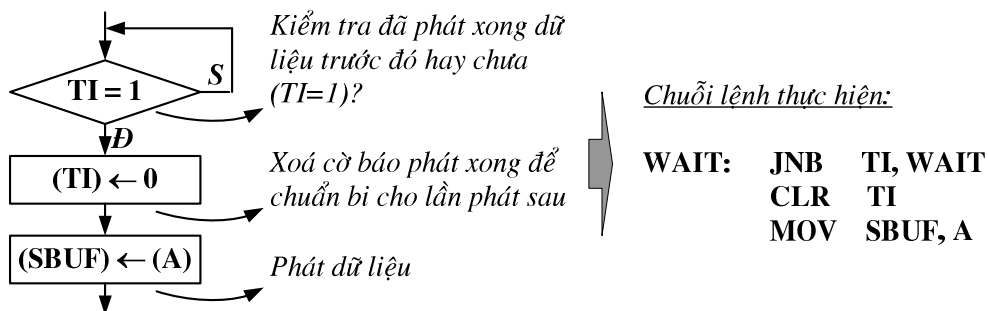
- Thông qua việc kiểm tra **cờ ngắt TI** có thể biết được chip 8051 đã sẵn sàng để truyền một byte dữ liệu hay chưa. Cần chú ý rằng, ở đây cờ TI được đặt ($TI = 1$) khi 8051 đã hoàn tất việc truyền một byte dữ liệu, còn được xoá ($TI = 0$) thì phải do người lập trình thực hiện bằng lệnh (**CLR TI**). Cũng nên nhớ rằng, nếu ghi một byte vào thanh ghi SBUF trước khi cờ TI được đặt ($TI = 1$) thì sẽ có nguy cơ bị mất phần dữ liệu trước đó do chưa kịp truyền đi. Cờ TI có thể được kiểm tra bằng lệnh (**JNB TI,...**) hoặc sử dụng phương pháp ngắt (sẽ được trình bày trong chương tiếp theo).

• Thông qua việc kiểm tra **cờ ngắt RI** có thể biết được chip 8051 đã nhận xong một byte dữ liệu hay chưa. Cần chú ý rằng, ở đây cờ RI được đặt ($RI = 1$) khi 8051 đã hoàn tất việc nhận một byte dữ liệu, còn được xoá ($RI=0$) thì phải do người lập trình thực hiện bằng lệnh (**CLR RI**). Cũng nên nhớ rằng, nếu không tiến hành cất nội dung của thanh ghi SBUF vào nơi an toàn thì sẽ có nguy cơ bị mất dữ liệu vừa nhận được do dữ liệu tiếp theo được chuyển vào. Cờ RI có thể được kiểm tra bằng lệnh (**JNB RI,...**) hoặc sử dụng phương pháp ngắt (sẽ được trình bày trong chương tiếp theo).

Lưu đồ và đoạn lệnh để kiểm tra và thu một dữ liệu nối tiếp từ thiết bị bên ngoài vào chip 8051 (chứa vào A):

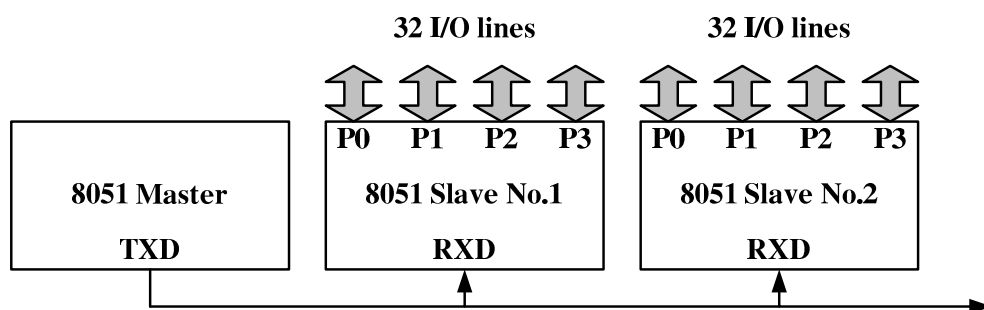


Lưu đồ và đoạn lệnh để kiểm tra và phát một dữ liệu nối tiếp từ chip 8051 (chứa trong A) ra thiết bị bên ngoài:



VI. TRUYỀN THÔNG ĐA XỬ LÝ:

Các chế độ 2 và 3 là các chế độ dự phòng cho việc truyền thông đa xử lý. Trong các chế độ này, 9 bit dữ liệu được thu và bit thứ 9 đưa đến RB8. Port có thể được lập trình sao cho khi bit stop được nhận, ngắt do port nối tiếp chỉ được tích cực nếu RB8=1. Đặc trưng này có thể được bằng cách set bit SM2 trong thanh ghi SCON bằng 1. Một ứng dụng cho điều này là một môi trường mạng sử dụng nhiều 8051 được sắp xếp theo mô hình chủ/tớ (master/slave) như hình dưới đây.

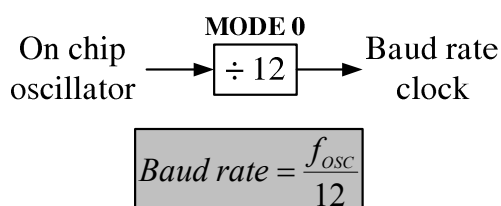


Khi bộ vi xử lý chủ (*master*) muốn truyền một khối dữ liệu đến một trong nhiều bộ vi xử lý tớ (*slave*), trước tiên bộ vi xử lý chủ phát đi một byte địa chỉ nhận dạng bộ vi xử lý tớ đích. Một byte địa chỉ khác với một byte dữ liệu ở chỗ bit thứ 9 là **1** (đối với *byte địa chỉ*) và là **0** (đối với *byte dữ liệu*). Một byte địa chỉ ngắt tất cả các bộ vi xử lý tớ để cho mỗi một bộ vi xử lý tớ có thể khảo sát byte nhận được để kiểm tra xem có phải là bộ vi xử lý tớ đang được định địa chỉ không. Bộ vi xử lý tớ được định địa chỉ sẽ xóa bit SM2 của mình và chuẩn bị nhận các byte dữ liệu theo sau. Các bộ vi xử lý tớ không được định địa chỉ có các bit SM2 của chúng được set bằng 1 và thực thi các công việc của riêng chúng, bỏ qua không nhận các byte dữ liệu. Các bộ vi xử lý này sẽ được ngắt lần nữa khi bộ vi xử lý chủ phát tiếp byte địa chỉ kế. Các sơ đồ cụ thể có thể được nêu ra sao cho một khi liên kết chủ tớ đã được thiết lập, bộ vi xử lý tớ cũng có thể phát đến bộ vi xử lý chủ. Mưu mẹo ở đây là không sử dụng bit dữ liệu thứ 9 sau khi liên kết vừa được thiết lập (*ngược lại các bộ vi xử lý tớ khác có thể được chọn một cách không cố ý*).

SM2 không ảnh hưởng đến chế độ 0, và trong chế độ 1 thì bit này có thể được dùng để kiểm tra sự hợp lệ của bit stop. Ở chế độ 1 thu, nếu SM2 = 1, ngắt thu sẽ không được tích cực trừ khi bit stop thu được là hợp lệ.

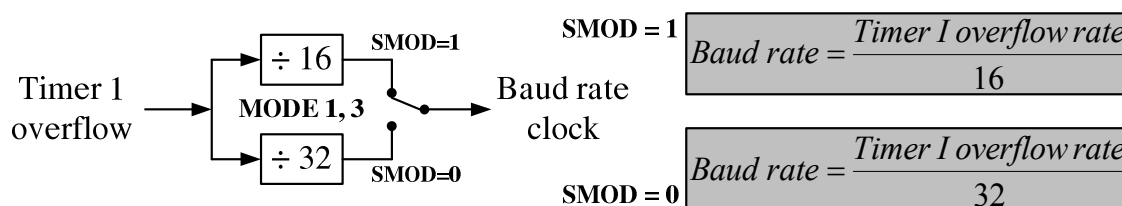
VII. TỐC ĐỘ BAUD CỦA PORT NỐI TIẾP:

1. Tốc độ baud cho chế độ 0:

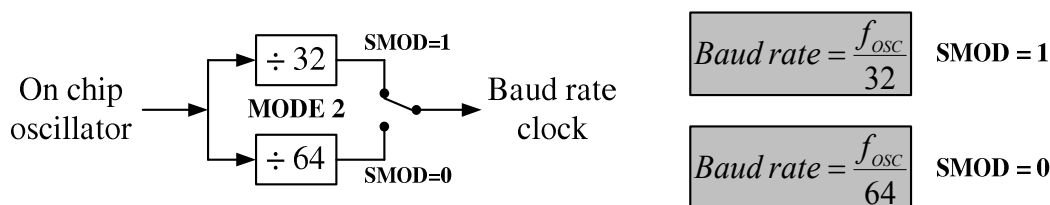


On chip oscillator: mạch dao động trên chip.
Baud rate clock: xung clock tốc độ baud.
Baud rate: tốc độ baud.
 f_{osc} : tần số mạch dao động trên chip.
Timer 1 overflow rate: tốc độ tràn Timer 1.

2. Tốc độ baud cho chế độ 1, 3:



3. Tốc độ baud cho chế độ 2:



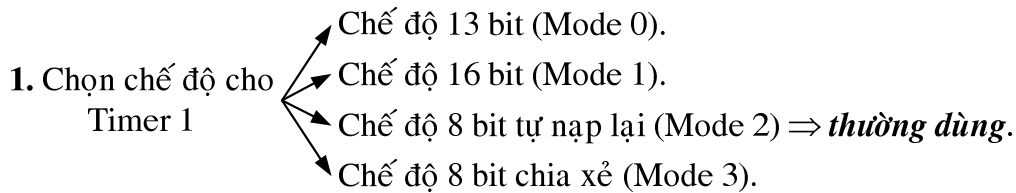
- Lưu ý:**

- Sau khi hệ thống *reset* thì bit SMOD = 0 (*chế độ mặc định*).
- Vì thanh ghi PCON không được định địa chỉ từng bit, nên để tăng gấp đôi tốc độ baud (*tức làm cho SMOD=1*) ta phải thực hiện bằng những dòng lệnh sau:

```
MOV A, PCON      ;Lấy giá trị từ thanh ghi PCON.
SETB ACC.7       ;SMOD = 1.
MOV PCON, A      ;Chuyển giá trị mới vào PCON.
```

4. Sử dụng Timer 1 làm xung clock tốc độ baud cho port nối tiếp:

Kỹ thuật tạo xung clock tốc độ baud bằng Timer 1:



2. Nạp giá trị thích hợp vào thanh ghi TH1 để có tốc độ tràn đúng
⇒ tạo ra tốc độ baud cho port nối tiếp.

Về chọn chế độ: thường dùng Timer 1 ở chế độ 8 bit tự động nạp lại (Mode 2). Các tốc độ baud rất chậm có thể nhận được bằng cách sử dụng chế độ 16 bit (Mode 1).

• **Ví dụ:** Khởi động thanh ghi TMOD để dùng T1 làm bộ tạo xung tốc độ baud (cho T1 hoạt động ở chế độ 2):

MOV TMOD, #2xH x: dành cho Timer 0

Về chọn tốc độ baud:

Gọi M là giá trị cần nạp cho thanh ghi TH1 để có tốc độ baud như yêu cầu, ta có:

$$M = - \frac{f_{Timer}}{Timer\ 1\ overflow\ rate}$$

$$\text{Mà: } f_{Timer} = \frac{f_{Timer}}{12} \quad \text{và} \quad Baud\ rate = \frac{Timer\ 1\ overflow\ rate}{16\ (hoặc\ 32)}$$

$$\Rightarrow M = - \frac{f_{Osc}}{12} \times \frac{1}{Baud\ rate \times 16\ (hoặc\ 32)}$$

Vậy ta có:

$$M = - \frac{f_{Osc}}{192 \times Baud\ rate}, (SMOD = 1) \quad \text{hoặc} \quad M = - \frac{f_{Osc}}{384 \times Baud\ rate}, (SMOD = 0)$$

Trong đó: f_{Osc} (Hz): tần số thạch anh.

Baud rate (bps): tốc độ baud của port nối tiếp.

• **Ví dụ:** Tạo tốc độ baud là 1200 với trường hợp $SMOD = 0$ và chip 8051 dùng thạch anh 12 MHz.

Gọi M là giá trị cần nạp cho thanh ghi TH1 để có tốc độ baud như yêu cầu, ta có:

$$M = - \frac{f_{Osc}}{384 \times Baud\ rate}, (SMOD = 0)$$

$$\Rightarrow M = - \frac{12 \cdot 10^6}{384 \times 1200} = -26,0416 \cong -26 \quad (\text{làm tròn số}).$$

Chuyển giá trị này vào thanh ghi TH1:

MOV TH1, #(-26) hoặc **MOV TH1, #E6H**

Chú ý: Do việc làm tròn cho nên sẽ có một sai số nhỏ trong việc xác định chính xác tốc độ baud. Cho nên để có tốc độ baud chính xác trong việc truyền dữ liệu thông qua port nối tiếp thì người ta thường dùng thạch anh dao động có tần số **11,0592 MHz** (thay vì là 12 MHz).

Ví dụ:

$$\Rightarrow M = -\frac{11,0592 \cdot 10^6}{384 \times 1200} = -24 \text{ (không cần phải làm tròn số)}$$

- Bảng tính tốc độ baud cho port nối tiếp:

Tốc độ baud	Tần số thạch anh	Giá trị nạp cho SMOD	Giá trị nạp cho TH1 (làm tròn)	Giá trị nạp cho TH1 (chưa làm tròn)	Tốc độ baud thực tế	Sai số
19200	12,000 MHz	1	-3 (FDH)	-3,25	20833	8,5%
9600	12,000 MHz	1	-7 (F9H)	-6,51	8928	7%
2400	12,000 MHz	0	-13 (F3H)	-13,02	2404	0,16%
1200	12,000 MHz	0	-26 (E6H)	-26,4	1202	0,16%
19200	11,0592 MHz	0	-15 (F1H)	-15	19200	0%
9600	11,0592 MHz	0	-3 (FDH)	-3	9600	0%
2400	11,0592 MHz	0	-12 (F4H)	-12	2400	0%
1200	11,0592 MHz	0	-24 (E8H)	-24	1200	0%

VIII. CÁC BƯỚC CƠ BẢN LẬP TRÌNH PORT NỐI TIẾP:

Trong các chế độ truyền dữ liệu nối tiếp của 8051 đã nêu trên thì trên thực tế sử dụng, để thực hiện việc thu và phát dữ liệu nối tiếp giữa chip 8051 với các thiết bị khác (8051, máy tính, các thiết bị SPI,...) thường người lập trình chỉ sử dụng hai chế độ sau: Mode 1 (UART 8 bit có tốc độ baud thay đổi) hoặc Mode 3 (UART 9 bit có tốc độ baud thay đổi). Còn hai chế độ còn lại thì rất ít sử dụng khi cần truyền dữ liệu nối tiếp. Cho nên ở đây chúng ta chỉ xem xét đến trình tự thực hiện việc lập trình (bao gồm thao tác khởi động và điều khiển thu/phát dữ liệu) để 8051 có thể truyền (phát) và nhận (thu) dữ liệu thông qua port nối tiếp theo hai chế độ UART nêu trên.

1. Lập trình 8051 truyền (phát) dữ liệu nối tiếp:

- Chọn chế độ hoạt động cho port nối tiếp:

MOV SCON, #...(1)...

- Chọn chế độ hoạt động Timer 1, cho bit GATE=0 và C/T=0:

MOV TMOD, #...(2)...

- Chọn giá trị thích hợp (căn cứ vào tốc độ baud) cho Timer 1:

MOV TH1, #...(3)...

- Cho Timer 1 chạy:

SETB TR1

- Kiểm tra xem đã phát xong toàn bộ dữ liệu trước đó hay chưa?

JNB TI, \$

hoặc

WAIT: JNB TI, WAIT

- Xoá cờ ngắt phát TI (chuẩn bị cho lần phát dữ liệu tiếp theo):

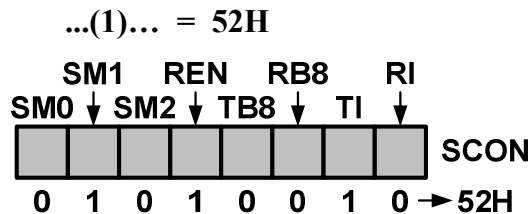
CLR TI

- Ghi dữ liệu cần phát vào port nối tiếp để phát đi:
MOV SBUF, ...(4)...
- Quay trở lại bước 5 để phát một dữ liệu tiếp theo.

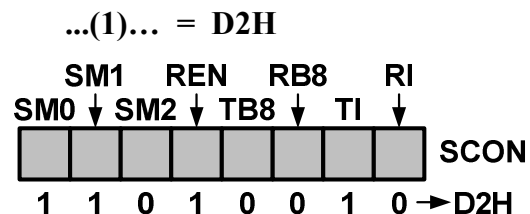
Lưu ý:

(1): Giá trị dùng để qui định chế độ hoạt động của port nối tiếp. Để đơn giản trong việc lập trình, ta có thể khởi động thanh ghi SCON theo như trình bày dưới đây:

Mode 1:

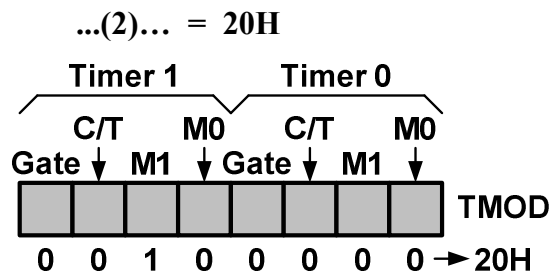


Mode 3:



(2): Giá trị dùng để qui định chế độ hoạt động của Timer 1 (dùng để tạo tốc độ baud cho việc truyền dữ liệu nối tiếp). Để đơn giản trong việc lập trình, ta có thể khởi động thanh ghi TMOD theo như trình bày dưới đây (chủ yếu ở đây ta chỉ cần sử dụng Timer 1 ở Mode 2 – Chế độ 8 bit tự động nạp lại):

Mode 2:



(3): Giá trị dùng để qui định tốc độ baud cho port nối tiếp. Giá trị này phụ thuộc vào tần số thạch anh, bit SMOD và tốc độ baud mà người lập trình mong muốn (xem thêm “**Bảng tính tốc độ baud cho port nối tiếp**” như trên đã trình bày).

...(3)... = M

$$M = \frac{f_{Osc}}{192 \times \text{Baud rate}}, (SMOD = 1) \text{ hoặc } M = \frac{f_{Osc}}{384 \times \text{Baud rate}}, (SMOD = 0)$$

Trong đó: f_{Osc} (Hz): tần số thạch anh.

Baud rate (bps): tốc độ baud của port nối tiếp.

(4): Dữ liệu cần phát đi thông qua port nối tiếp. Dữ liệu này có thể là nội dung của một ô nhớ, thanh ghi hoặc một giá trị tức thời.

Nên nhớ rằng, nếu có yêu cầu thì bit gửi kèm theo (ví dụ như bit Parity) cần phải thêm vào trước khi tiến hành quá trình phát dữ liệu (Mode 1: kèm thêm vào vị trí của bit D7, Mode 3: kèm thêm vào vị trí của bit TB8).

2. Lập trình 8051 nhận (thu) dữ liệu nối tiếp:

- Chọn chế độ hoạt động cho port nối tiếp:
MOV SCON, #...(1)...
- Chọn chế độ hoạt động Timer 1, cho bit GATE=0 và C/T=0:
MOV TMOD, #...(2)...
- Chọn giá trị thích hợp (căn cứ vào tốc độ baud) cho Timer 1:
MOV TH1, #...(3)...
- Cho Timer 1 chạy:
SETB TR1
- Kiểm tra xem đã thu toàn bộ dữ liệu hay chưa?
JNB RI, \$
hoặc
WAIT: JNB RI, WAIT
- Xóa cờ ngắt thu RI (chuẩn bị cho lần thu dữ liệu tiếp theo):
CLR RI
- Cất dữ liệu vừa thu được vào nơi an toàn (tránh bị mất dữ liệu):
MOV ...(4)...., SBUF
- Quay trở lại bước 5 để nhận một dữ liệu tiếp theo.

Lưu ý:

- ✚ (1): Xem thêm “*Lập trình 8051 truyền (phát) dữ liệu nối tiếp*”.
- ✚ (2): Xem thêm “*Lập trình 8051 truyền (phát) dữ liệu nối tiếp*”.
- ✚ (3): Xem thêm “*Lập trình 8051 truyền (phát) dữ liệu nối tiếp*”.
- ✚ (4): Địa chỉ của một ô nhớ, thanh ghi mà dữ liệu thu được từ port nối tiếp sẽ lưu giữ vào trong đó.

Nên nhớ rằng, nếu có yêu cầu thì bit gửi kèm theo (ví dụ như bit Parity) cần phải được xử lý trước khi tiến hành việc cất dữ liệu thu được (Mode 1: nằm tại vị trí của bit D7, Mode 3: nằm tại vị trí của bit RB8).

IX. CÁC VÍ DỤ MINH HỌA:

1. Ví dụ 1: (Chọn tốc độ baud)

Chip 8051 sử dụng thạch anh 11,0592MHz. Hãy xác định giá trị cần nạp cho thanh ghi TH1 để có được các tốc độ baud: 9600, 2400, 1200 (nếu SMOD=0) và 19200 (nếu SMOD=1).

Giải

Xét trường hợp SMOD=0:

Gọi M là giá trị cần nạp cho thanh ghi TH1 để có tốc độ baud như yêu cầu, ta có:

$$M = - \frac{f_{Osc}}{384 \times \text{Baud rate}}$$

Baud rate = 9600

$$\Rightarrow M = - \frac{11,0592 \times 10^6}{384 \times 9600} = -3 \Rightarrow (\text{TH1}) = -3 \text{ hay } (\text{TH1}) = \text{FDH.}$$

Baud rate = 2400

$$\Rightarrow M = -\frac{11,0592 \times 10^6}{384 \times 2400} = -12 \Rightarrow (\text{TH1}) = -12 \text{ hay } (\text{TH1}) = \text{F4H}.$$

Baud rate = 1200

$$\Rightarrow M = -\frac{11,0592 \times 10^6}{384 \times 1200} = -24 \Rightarrow (\text{TH1}) = -24 \text{ hay } (\text{TH1}) = \text{F8H}.$$

Xét trường hợp SMOD=1:

Gọi M là giá trị cần nạp cho thanh ghi TH1 để có tốc độ baud như yêu cầu, ta có:

$$M = -\frac{f_{Osc}}{192 \times \text{Baud rate}}$$

Baud rate = 19200

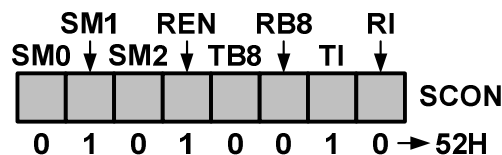
$$\Rightarrow M = -\frac{11,0592 \times 10^6}{192 \times 19200} = -3 \Rightarrow (\text{TH1}) = -3 \text{ hay } (\text{TH1}) = \text{FDH}.$$

2. Ví dụ 2: (Khởi động port nối tiếp)

Viết một chuỗi lệnh để khởi động port nối tiếp sao cho port này hoạt động như một UART 8 bit với tốc độ baud là 2400, sử dụng Timer1 để cung cấp xung clock tốc độ baud. Chip 8051 sử dụng thạch anh 12MHz.

Giải

Để khởi động port nối tiếp có cấu hình như trên ta cần tác động đến các thanh ghi: SCON, TMOD, TCON và TH1.

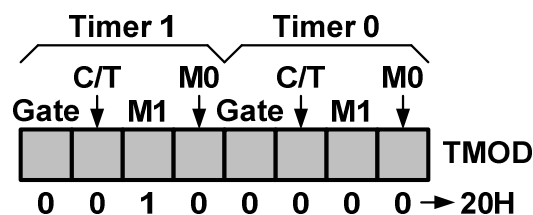


$SM0 = 0, SM1 = 1, SM2 = 0 \rightarrow$ chế độ UART 8 bit.

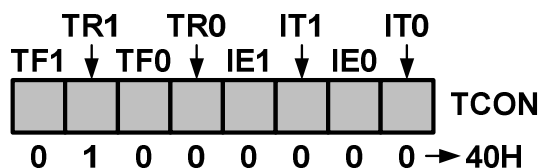
$REN = 1 \rightarrow$ cho phép port nối tiếp thu dữ liệu.

$TI = 1 \rightarrow$ cho phép port sẵn sàng phát dữ liệu (bộ đệm phát rỗng).

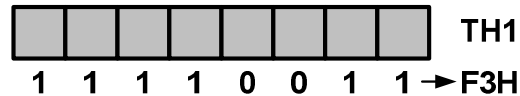
$RI = 0 \rightarrow$ cho phép port sẵn sàng thu dữ liệu (bộ đệm thu rỗng).



$GATE = 0, C/T = 0, M1 = 1, M0 = 0 \rightarrow$ Timer 1 ở chế độ định thời 8 bit tự nạp lại.



$TR1 = 1 \rightarrow$ cho phép Timer 1 hoạt động.



Gọi M là giá trị cần nạp cho thanh ghi TH1 để có tốc độ baud như yêu cầu, ta có:

$$M = -\frac{f_{Osc}}{384 \times \text{Baud rate}} \quad (SMOD = 0)$$

$$\Rightarrow M = -\frac{12.10^6}{384 \times 2400} = -13,02 \cong -13 \text{ (làm tròn số).}$$

Chuyển giá trị này vào thanh ghi TH1:

MOV TH1, #(-13) hoặc **MOV TH1, #F3H**

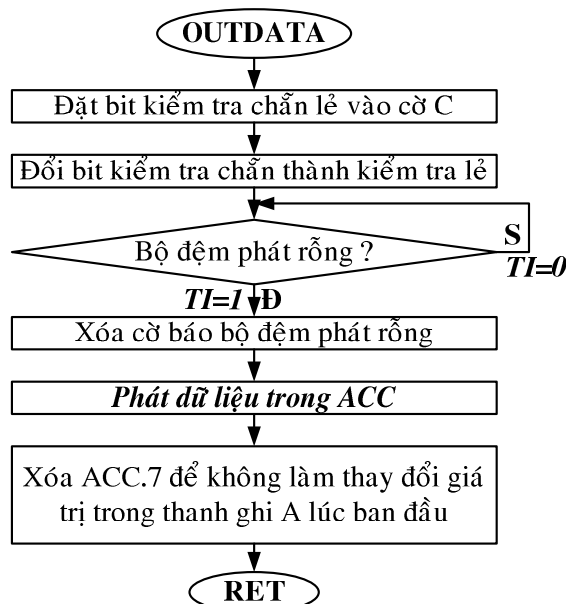
Chuỗi lệnh khởi động port nối tiếp được viết như sau:

MOV SCON, #52H
MOV TMOD, #20H
MOV TH1, #-13
SETB TR1

3. Ví dụ 3: (Chương trình con phát (xuất) dữ liệu)

Giả sử port nối tiếp đã được khởi động (như ở ví dụ 1). Hãy viết một chương trình con để phát dữ liệu (dạng 7 bit) chứa trong thanh ghi A ra port nối tiếp với bit kiểm tra lẻ là bit thứ 8. Chú ý rằng, việc trở về từ chương trình con này không được làm thay đổi nội dung thanh ghi A.

Giải



Chương trình được viết như sau:

OUTDATA:

MOV C, P
CPL C
MOV ACC.7, C
JNB TI, \$
CLR TI
MOV SBUF, A
CLR ACC.7
RET

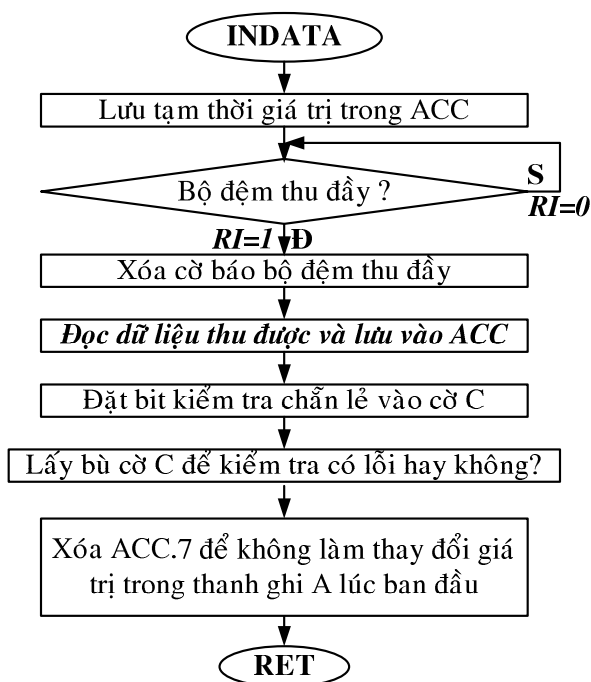
Ba lệnh đầu tiên đặt bit kiểm tra lẻ vào bit 7 của thanh ghi A (ACC.7). Do bit P trong thanh ghi PSW được thiết lập để kiểm tra chẵn cho giá trị trong thanh ghi A, cho nên bit này phải được lấy bù để trở thành bit kiểm tra lẻ trước khi đặt vào ACC.7. Lệnh JNB tạo ra một vòng lặp chờ để kiểm tra cờ ngắt phát TI cho đến khi cờ này được set bằng 1. Khi TI=1 (do việc phát ký tự trước đó vừa kết thúc), bit này sẽ được xóa và sau đó ký tự trong thanh ghi A được ghi vào bộ đệm của port nối tiếp SBUF và

việc phát ký tự được bắt đầu ở lần tràn kế của bộ đếm tạo xung clock cho port nối tiếp. Sau cùng bit ACC.7 sẽ được xóa để giá trị trả về giống như khi mã 7 bit được chuyển đến chương trình con.

4. Ví dụ 4: (Chương trình con thu (nhận) dữ liệu)

Giả sử port nối tiếp đã được khởi động (như ở ví dụ 1). Hãy viết một chương trình con để thu dữ liệu từ port nối tiếp và nạp giá trị thu được vào thanh ghi A (dữ liệu thu được có dạng 7 bit dữ liệu + 1 bit kiểm tra lỗi, tương tự dữ liệu từ ví dụ 2). Sử dụng bit thứ 8 thu được làm bit kiểm tra lỗi và set cờ nhớ C nếu có lỗi chẵn lẻ.

Giải



Chương trình được viết như sau:

INDATA:

```

JNB RI, $
CLR RI
MOV A, SBUF
MOV C, P
CPL C
CLR ACC.7
RET
    
```

Chương trình con này bắt đầu bằng việc chờ cờ ngắt thu RI được set bằng 1 để chỉ ra rằng ký tự đã sẵn sàng trong bộ đếm thu SBUF (để được đọc). Khi RI=1, lệnh JNB chuyển điều khiển đến lệnh tiếp theo sau lệnh này. Cờ RI được xóa và mã trong SBUF được chứa vào thanh ghi A. Do bit P trong thanh ghi PSW được thiết lập để kiểm tra chẵn cho giá trị trong thanh ghi A. Cho nên bit này sẽ được set bằng 1 nếu nội dung thanh ghi A (tức dữ liệu vừa thu được) có chứa bit kiểm tra lỗi ở bit thứ 7 của thanh ghi này và ngược lại thì bit này sẽ được xóa bằng 0 thông qua lệnh CPL. Việc di chuyển bit P vào trong cờ nhớ CY sẽ làm cho CY=0 nếu không có lỗi hoặc CY=1 nếu có một lỗi chẵn lẻ. Sau cùng bit ACC.7 sẽ được xóa để đảm bảo rằng chỉ có mã 7 bit được trả về cho chương trình đã gọi.

5. Ví dụ 5: (Truyền dữ liệu)

Viết chương trình cho 8051 ($f_{osc}=11,0592\text{MHz}$) để truyền liên tục một ký tự “A” thông qua port nối tiếp với tốc độ 4800 baud (Mode 1).

Giải

- **Tính toán:** Dựa vào những công thức đã học, ta có:

Port nối tiếp (Mode 1)

⇒ (SCON) = 52H

Timer 1 (Mode 2)

⇒ (TMOD) = 20H

Baud rate = 4800 và $f_{osc}=11,0592\text{MHz}$

⇒ (TH1) = -6 với (SMOD) = 0

- **Chương trình:** Dựa vào những tính toán trên, ta có:

```

MOV     SCON, #52H      ;UART 8 bit, tốc độ baud thay đổi.
MOV     TMOD, #20H      ;Chế độ 8 bit tự nạp lại.
MOV     TH1, #(-6)      ;Baud rate = 4800.
SETB    TR1             ;Khởi động Timer 1.
LOOP:
JNB     TI, $           ;Kiểm tra phát dữ liệu trước hoàn tất?
CLR     TI              ;Xoá cờ TI, chuẩn bị cho lần phát kế tiếp.
MOV     SBUF, #'A'      ;Phát ký tự "A".
SJMP    LOOP            ;Lặp lại quá trình phát.
END      ;Kết thúc chương trình.
    
```

6. Ví dụ 6: (Truyền dữ liệu)

Viết chương trình cho 8051 ($f_{Osc}=11,0592MHz$) để truyền liên tục chuỗi ký tự "TTCNDT" thông qua port nối tiếp với tốc độ 19200 baud (Mode 1).

Giải

- **Tính toán:**

Port nối tiếp (Mode 1) $\Rightarrow (SCON) = 52H$
 Timer 1 (Mode 2) $\Rightarrow (TMOD) = 20H$
 Baud rate = 19200 và $f_{Osc}=11,0592MHz$ $\Rightarrow (TH1) = -3$ với $(SMOD) = 1$

- **Chương trình:** Dựa vào những tính toán trên, ta có:

```

MOV     SCON, #52H      ;UART 8 bit, tốc độ baud thay đổi.
MOV     TMOD, #20H      ;Chế độ 8 bit tự nạp lại.
MOV     TH1, #(-3)      ;Baud rate = 19200.
MOV     A, PCON          ;Lấy giá trị từ thanh ghi PCON.
SETB    ACC.7           ;SMOD = 1.
MOV     PCON, A          ;Chuyển giá trị mới vào PCON.
SETB    TR1             ;Khởi động Timer 1.
LOOP:
MOV     DPTR, #MYDATA    ;Nạp con trỏ vùng dữ liệu.
NEXT:
CLR     A                ;Xoá ACC, A = 0
MOVC    A, @A+DPTR       ;Lấy dữ liệu tại ô nhớ ROM do
                        ;(A+DPTR) trỏ đến đưa vào A.
JZ      EXIT            ;Thoát nếu là ký tự Null.
JNB     TI, $           ;Kiểm tra phát dữ liệu trước hoàn tất?
CLR     TI              ;Xoá cờ TI, chuẩn bị phát tiếp.
MOV     SBUF, A          ;Phát dữ liệu ra port nối tiếp.
INC     DPTR            ;Tăng con trỏ dữ liệu.
SJMP    NEXT            ;Lặp lại quá trình phát ký tự kế tiếp.
EXIT:
SJMP    LOOP            ;Lặp lại từ đầu.
MYDATA:
DB      "TTCNDT",0      ;Dữ liệu cần truyền đi, có ký tự Null.
END      ;Kết thúc chương trình.
    
```

8. Ví dụ 8: (Nhận dữ liệu)

Viết chương trình cho 8051 ($f_{osc}=11,0592\text{MHz}$) để nhận liên tục các dữ liệu thông qua port nối tiếp với tốc độ 4800 baud (Mode 1) và gửi các dữ liệu nhận được đến P1.

Giải

• Tính toán:

Port nối tiếp (Mode 1) $\Rightarrow$ (SCON) = 52H
 Timer 1 (Mode 2) $\Rightarrow$ (TMOD) = 20H
 Baud rate = 4800 và $f_{osc}=11,0592\text{MHz}$ $\Rightarrow$ (TH1) = -6 với (SMOD) = 0

• Chương trình: Dựa vào những tính toán trên, ta có:

```
MOV    SCON, #52H    ;UART 8 bit, tốc độ baud thay đổi.
MOV    TMOD, #20H    ;Chế độ 8 bit tự nạp lại.
MOV    TH1, #(-6)    ;Baud rate = 4800.
SETB   TR1           ;Khởi động Timer 1.

LOOP:
JNB     RI, $         ;Kiểm tra đã thu dữ liệu hoàn tất?
CLR     RI            ;Xoá cờ RI, chuẩn bị cho lần thu kế tiếp.
MOV     A, SBUF       ;Thu và cất dữ liệu vào ACC.
MOV     P1, A         ;Gửi dữ liệu thu được ra P1.
SJMP    LOOP         ;Lặp lại quá trình phát.
END        ;Kết thúc chương trình.
```

9. Ví dụ 9: (Thu và phát dữ liệu)

Cho port nối tiếp của 8051 ($f_{osc}=11,0592\text{MHz}$) được nối với cổng COM của máy tính PC (giả sử rằng trên máy tính đã có sẵn chương trình để gửi và nhận dữ liệu nối tiếp thông qua cổng COM). Hãy viết chương trình cho 8051 để thực hiện các công việc sau:

- Gửi câu thông báo “READY” đến máy tính.
- Liên tục nhận các dữ liệu nối tiếp từ máy tính gửi đến và chuyển các dữ liệu này ra P1 của 8051.
- Liên tục lấy các dữ liệu từ P2 của 8051 và phát các dữ liệu này đến máy tính thông qua port nối tiếp.

Biết rằng 8051 truyền dữ liệu nối tiếp ở Mode 1 với tốc độ baud là 9600.

Giải

• Tính toán:

Port nối tiếp (Mode 1) $\Rightarrow$ (SCON) = 52H
 Timer 1 (Mode 2) $\Rightarrow$ (TMOD) = 20H
 Baud rate = 9600 và $f_{osc}=11,0592\text{MHz}$ $\Rightarrow$ (TH1) = -3 với (SMOD) = 0

• Chương trình: Dựa vào những tính toán trên, ta có:

```
MOV     P2, #0FFH    ;Cấu hình P2 là cổng vào.
MOV     SCON, #52H    ;UART 8 bit, tốc độ baud thay đổi.
MOV     TMOD, #20H    ;Chế độ 8 bit tự nạp lại.
MOV     TH1, #(-3)    ;Baud rate = 4800.
SETB    TR1          ;Khởi động Timer 1.
MOV     DPTR, #MYDATA ;Nạp con trỏ vùng dữ liệu.
NEXT:
CLR     A             ;Xoá ACC, A = 0
MOVC    A, @A+DPTR    ;Lấy dữ liệu tại ô nhớ ROM do
```

		; (A+DPTR) trở đến đưa vào A.
JZ	EXIT	; Thoát nếu là ký tự Null.
ACALL	OUTDATA	; Phát dữ liệu, câu thông báo.
INC	DPTR	; Tăng con trỏ dữ liệu.
SJMP	NEXT	; Lặp lại quá trình phát ký tự kế tiếp.
EXIT:		; Phân phát/thu dữ liệu giữa 8051 và PC.
ACALL	INDATA	; Thu dữ liệu từ PC.
MOV	P1, A	; Chuyển dữ liệu này đến P1.
MOV	A, P2	; Lấy dữ liệu từ P2.
ACALL	OUTDATA	; Phát dữ liệu này đến PC.
SJMP	EXIT	; Lặp lại chu trình làm việc.
INDATA:		; Chương trình con thu dữ liệu.
JNB	RI, \$	; Kiểm tra đã thu dữ liệu hoàn tất?
CLR	RI	; Xóa cờ RI, chuẩn bị cho lần thu tiếp.
MOV	A, SBUF	; Thu và cất dữ liệu vào ACC.
RET		
OUTDATA:		; Chương trình con phát dữ liệu.
JNB	TI, \$	; Kiểm tra phát dữ liệu trước hoàn tất?
CLR	TI	; Xóa cờ TI, chuẩn bị cho lần phát tiếp.
MOV	SBUF, A	; Phát dữ liệu chứa trong ACC.
RET		
MYDATA:		
DB	"READY",0	; Câu thông báo, có ký tự Null.
END		; Kết thúc chương trình.

X. PHẦN BÀI TẬP:

Bài 1: Viết đoạn lệnh đọc một chuỗi data chứa trong RAM nội từ địa chỉ 30H đến 50H và xuất ra một thiết bị (ví dụ như màn hình tinh thể lỏng LCD) được nối với port nối tiếp của 8051 (chế độ UART 8 bit, 2400 baud). Cho $f_{osc}=11,0592$ MHz.

Bài 2: Viết đoạn lệnh nhận một chuỗi data từ một thiết bị ngoài (ví dụ như máy đọc mã vạch) nối với 8051 qua port nối tiếp (chế độ UART 8 bit, 4800 baud) và ghi data vào RAM nội từ địa chỉ 40H. Biết rằng chuỗi data gồm 20 byte và $f_{osc}=11,0592$ MHz.

Bài 3: Viết đoạn lệnh lấy một chuỗi data chứa trong RAM ngoài bắt đầu từ địa chỉ 2000H và xuất ra một thiết bị được nối với port nối tiếp của 8051 (chế độ UART 8 bit, 1200 baud). Chuỗi kết thúc bởi ký tự EOT (có mã ASCII là 04H) và ký tự này cũng được xuất ra ($f_{osc}=11,0592$ MHz).

Bài 4: Làm lại bài 3 nhưng không xuất ký tự EOT.

Bài 5: Viết đoạn lệnh nhận một chuỗi data từ một thiết bị ngoài nối với 8051 qua port nối tiếp (chế độ UART 8 bit, 9600 baud) và ghi data vào RAM ngoài bắt đầu từ địa chỉ 4000H. Chuỗi data bắt đầu bằng ký tự STX (02H) và kết thúc bằng ký tự ETX (03H). Không ghi hai ký tự này vào RAM. Cho $f_{osc}=11,0592$ MHz.

Bài 6: Viết chương trình con mang tên XUAT có nhiệm vụ lấy một chuỗi data chứa trong RAM ngoài xuất ra port nối tiếp ở chế độ UART 9 bit. Bit thứ 9 là bit parity chẵn. Chuỗi data kết thúc bằng ký tự NULL (00H). đoạn lệnh gọi chương trình con XUAT sẽ đặt địa chỉ bắt đầu của chuỗi vào DPTR trước khi gọi chương trình con XUAT. Giả sử port nối tiếp đã được khởi động.

Bài 7: Viết chương trình con mang tên NHAP có nhiệm vụ nhập một chuỗi data gồm 30 byte từ port nối tiếp ở chế độ UART 9 bit, bit thứ 9 là bit parity lẻ. Nếu data nhận được không bị lỗi thì ghi

vào một vùng nhớ của RAM nội, nếu bị lỗi thì không ghi. đoạn lệnh gọi chương trình con NHAP sẽ đặt địa chỉ đầu của vùng nhớ vào thanh ghi R0 trước khi gọi chương trình con NHAP. Giả sử port nối tiếp đã được khởi động.