

Hiệu đính từ slide của thầy Hồ Trung Mỹ  
Bộ môn Điện tử - DH BK TPHCM

# CHƯƠNG 3

## HỘ VI ĐIỀU KHIỂN 8051

# 3.6 Cổng nối tiếp (Serial Port)

# Các hoạt động cổng nối tiếp của 8051

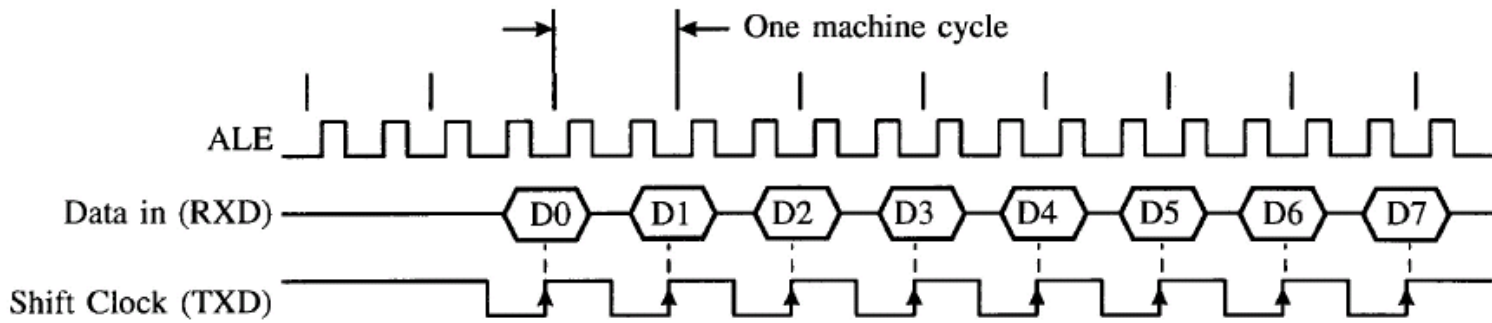
- Giới thiệu
- Chuẩn EIA(Electronic Industries Association)-232E hoặc RS232
- Giao tiếp cổng nối tiếp MCS-51/52
- Ứng dụng của chế độ 0
- Ứng dụng của chế độ 1
- Ứng dụng của chế độ 2 & 3

# Truyền dữ liệu nối tiếp

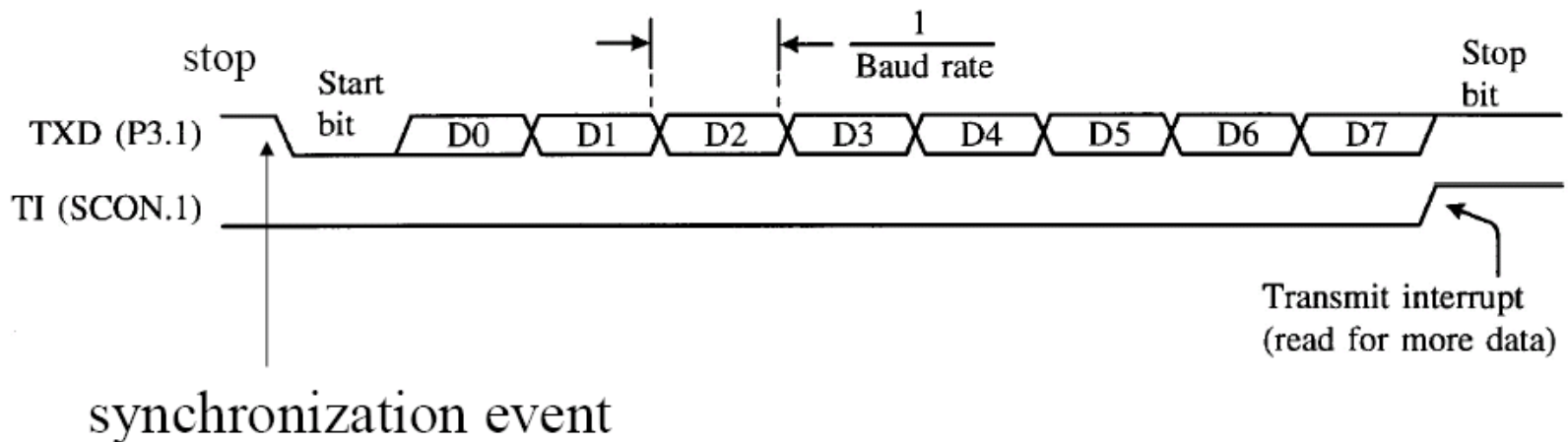
- Truyền dữ liệu khoảng cách xa
- Tốc độ truyền dữ liệu thấp
- Truyền dữ liệu nối tiếp đồng bộ
  - Dùng tín hiệu clock riêng để đồng bộ hóa mạch thu và phát.
  - Thường dùng khung dữ liệu lớn hơn và nhanh hơn.
- Truyền dữ liệu nối tiếp bất đồng bộ
  - Không cần tín hiệu clock riêng để đồng bộ hoá chuyển dữ liệu.
  - Dựa trên ký tự. Mỗi ký tự được đóng khung bằng “start bit” và “stop bit”.
  - Bên thu cần nhận dạng “start bit” và “stop bit” để nhận đúng ký tự dữ liệu.

# Truyền dữ liệu nối tiếp đồng bộ (synchronous) và bất đồng bộ (asynchronous) với 8051

## Synchronous serial communication

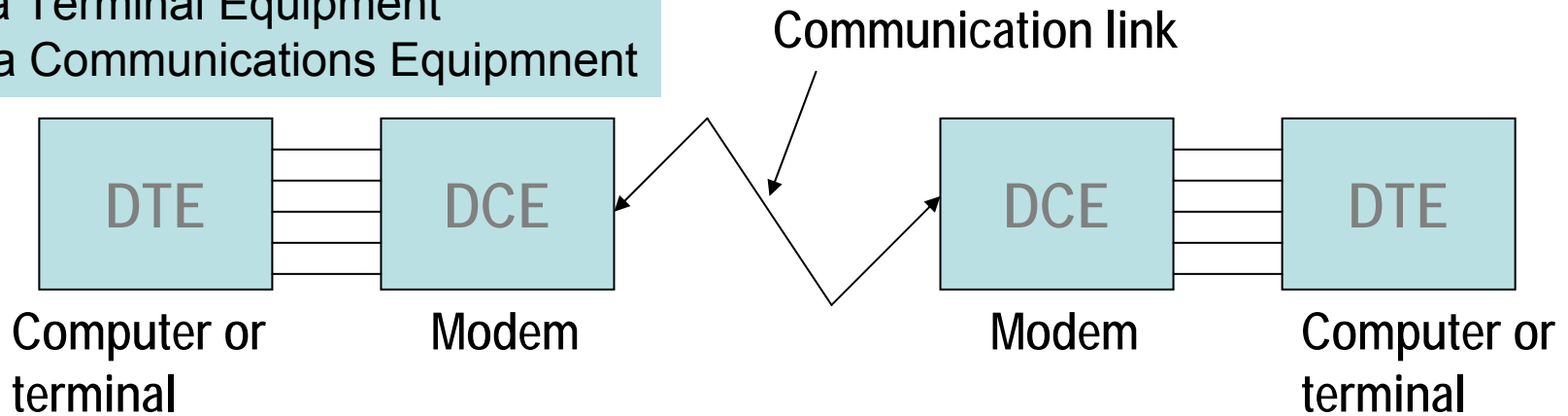


## Asynchronous serial communication



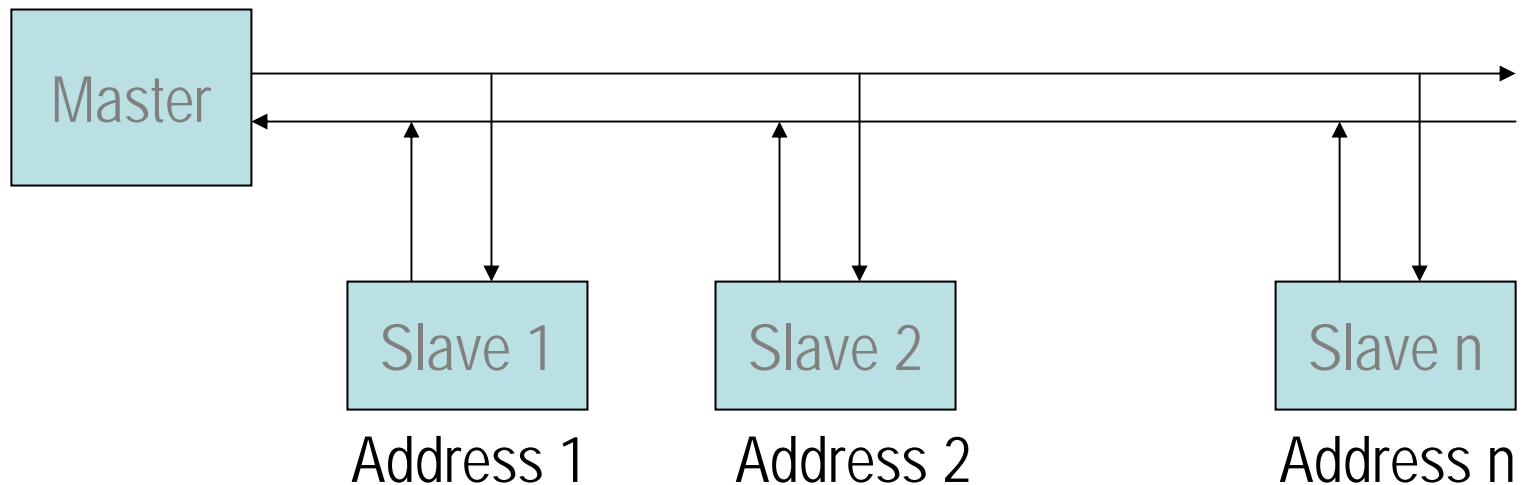
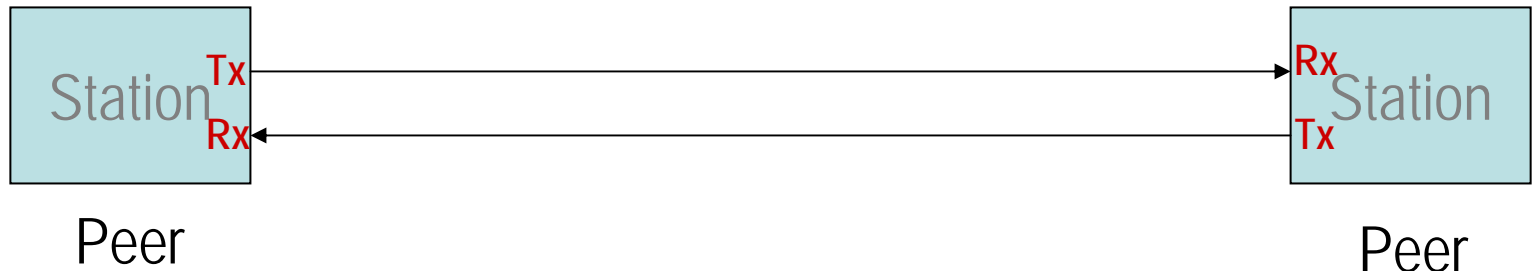
# Đường truyền dữ liệu cơ bản

DTE: Data Terminal Equipment  
DCE: Data Communications Equipment



- Người ta có thể sử dụng **đường điện thoại công cộng hoặc dành riêng** làm môi trường truyền dữ liệu nối tiếp bất đồng bộ.
- **Modem** được dùng để chuyển đổi dữ liệu số thành dạng sóng analog thích hợp cho việc truyền trên đường dây điện thoại và ngược lại.
- Có 2 kiểu đường truyền: **Điểm đến điểm** và **Đa điểm**.

# Đường truyền Điểm đến điểm và Đa điểm

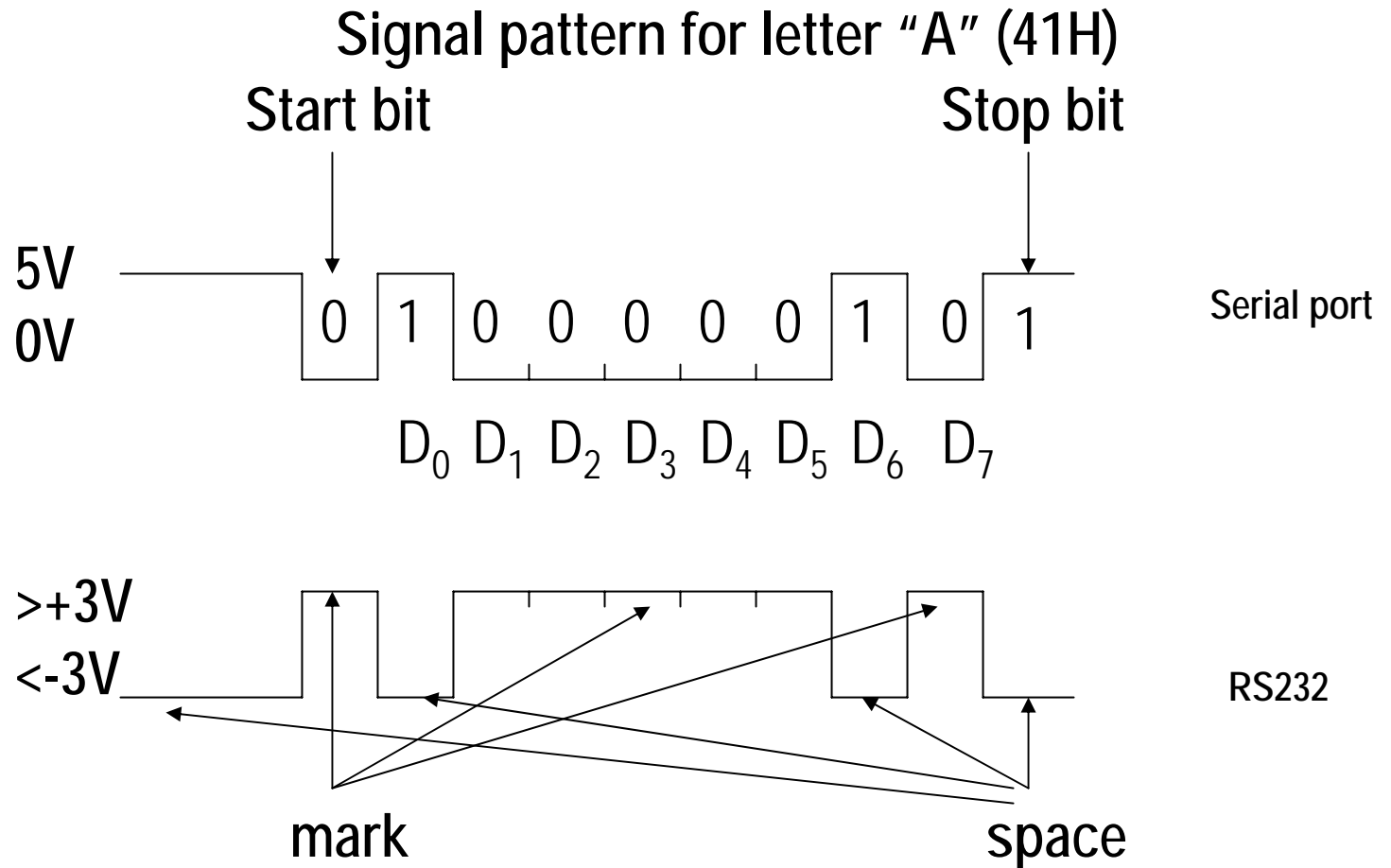


# Cơ bản về các đường truyền

- Hệ thống 2 dây và 4 dây:
  - 2 dây: tín hiệu và đất.
  - 4 dây: 2 bộ 2 dây.
- Các kiểu đường truyền:
  - Đơn công (**Simplex link**): đường dây dành riêng cho thu hoặc phát chứ không cả hai.
  - Bán song công (**Half-duplex link**): đường dây có thể dùng cho phát hoặc thu nhưng mỗi lần chỉ có 1 chiều.
  - Song công (**Full-duplex link**): Phát và thu có thể tiến hành đồng thời. Đường truyền cần 4 dây.



# Cơ bản về các đường truyền

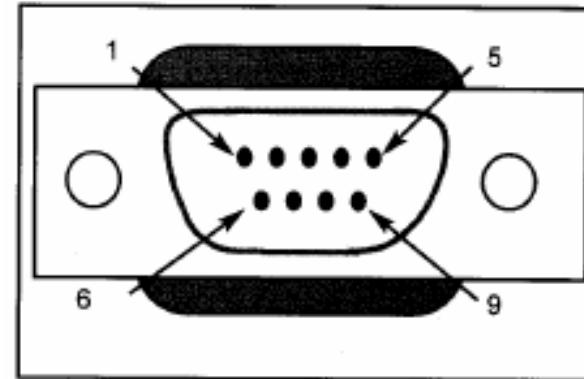
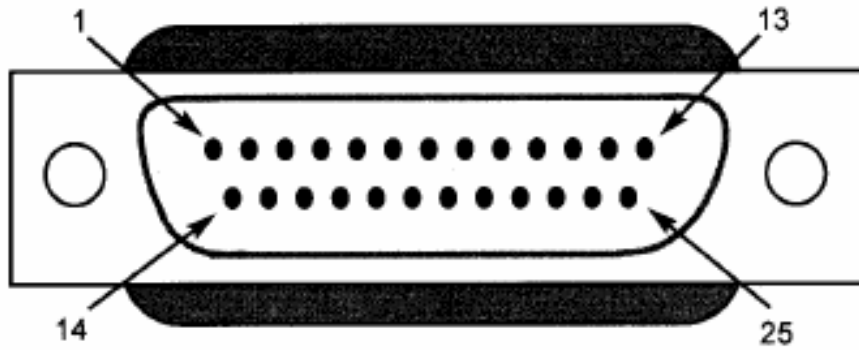


# EIA-232-E or RS-232

- EIA-232-E or RS-232 can apply to the following data communication schemes:
  - Serial communications
  - Synchronous and asynchronous
  - Dedicated leased or private lines
  - Switched service
  - Two wire or four wire
  - Point to point or multipoint
- 4 aspects in EIA-232-E interface: mechanical, functional, procedural, and electrical.

# EIA-232-E Mechanical Characteristics

- 25-pin DB25 /9-pin DB9 plug connector.



# DB-9 Signals

<b>Pin</b>	<b>Description</b>
1	Data carrier detect ( $\overline{\text{DCD}}$ )
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready ( $\overline{\text{DSR}}$ )
7	Request to send ( $\overline{\text{RTS}}$ )
8	Clear to send ( $\overline{\text{CTS}}$ )
9	Ring indicator (RI)

# EIA-232-E Connector and Pin Assignment

Signal direction	Signal Name	Signal Name	Signal direction
to DCE	Secondary TxD	protective ground	Both
to DTE	Tx clock	TxD	to DCE
to DTE	Secondary RxD	RxD	to DTE
to DTE	Rx clock	Request To Send	to DCE
	unassigned	Clear To Send	to DTE
to DCE	Secondary Request To Send	Data Set Ready	to DTE
to DCE	Data Terminal Ready	Signal ground	Both
to DTE	Signal Quality Detect	Carrier Detect	to DTE
to DTE	Ring indicator	Reserved	
Both	Data Rate Select	Reserved	
to DCE	Tx clock	Unassigned	
	unassigned	Secondary Carrier Detect	to DTE
		Secondary Clear To Send	to DTE

# Functions of EIA-232-E Interchange Circuits

1	-	Shield
2	BA	Transmitted data
3	BB	Received data
4	CA/CJ	Request to send/ready for receiving
5	CB	Clear to send
6	CC	DCE
7	AB	Signal common
8	CF	Received line signal detector
9,10,11	-	(Reserved for testing, unassigned)
12	SCF/CI	Secondary received line signal detection/data range selector (DCE source)
13	SCB	Secondary clear to send
14	SBA	Secondary transmitted data
15	DB	Transmitter signal element timing (DCE source)
16	SBB	Secondary received data
17	DD	Received signal element timing
18	LL	Local loopback
19	SCA	Secondary request to send
20	CD	DTE ready
21	RL/CG	Remote loopback/signal quality detector
22	CE	Ring indicator
23	CH/CI	Data signal rate selector (DTE/DCE source)
24	DA	Transmitter signal element timing (DTE source)
25	TM	Test mode

# EIA-232-E Procedural Characteristics



Signal GND (7)			Signal GND (7)
DTE ready (20)	->		<- DTE ready (20)
	<- DCE ready (6)	DCE ready (6)	->
Request to send(4)	-> carrier on	carrier detect (8)	->
	<- clear to send (5)		
data send (2)	-> data send	data receive (3)	->
send timing (24)	->	receive timing (17)	->
EOT send (2)	-> EOT send	EOT receive (3)	->
send timing (24)	->	receive timing (17)	->
RTS off (4)	->		error check
	carrier off	carrier off (8)	->
	<- CTS off (5)		<- RTS (4)
	<- carrier detect (8)	carrier on	
		clear to send (5)	->
	<- ACK message (3)	modulated ACK MSG	<- ACK message (2)
	<-receive timing (17)		<- send timing (24)
	<- EOT received (3)	modulate EOT send	<- EOT send (2)
	<- receive timing (17)		<- send timing (24)
	<- carrier off (8)	carrier off	<-RTS off (4)
		CTS off (5)	->

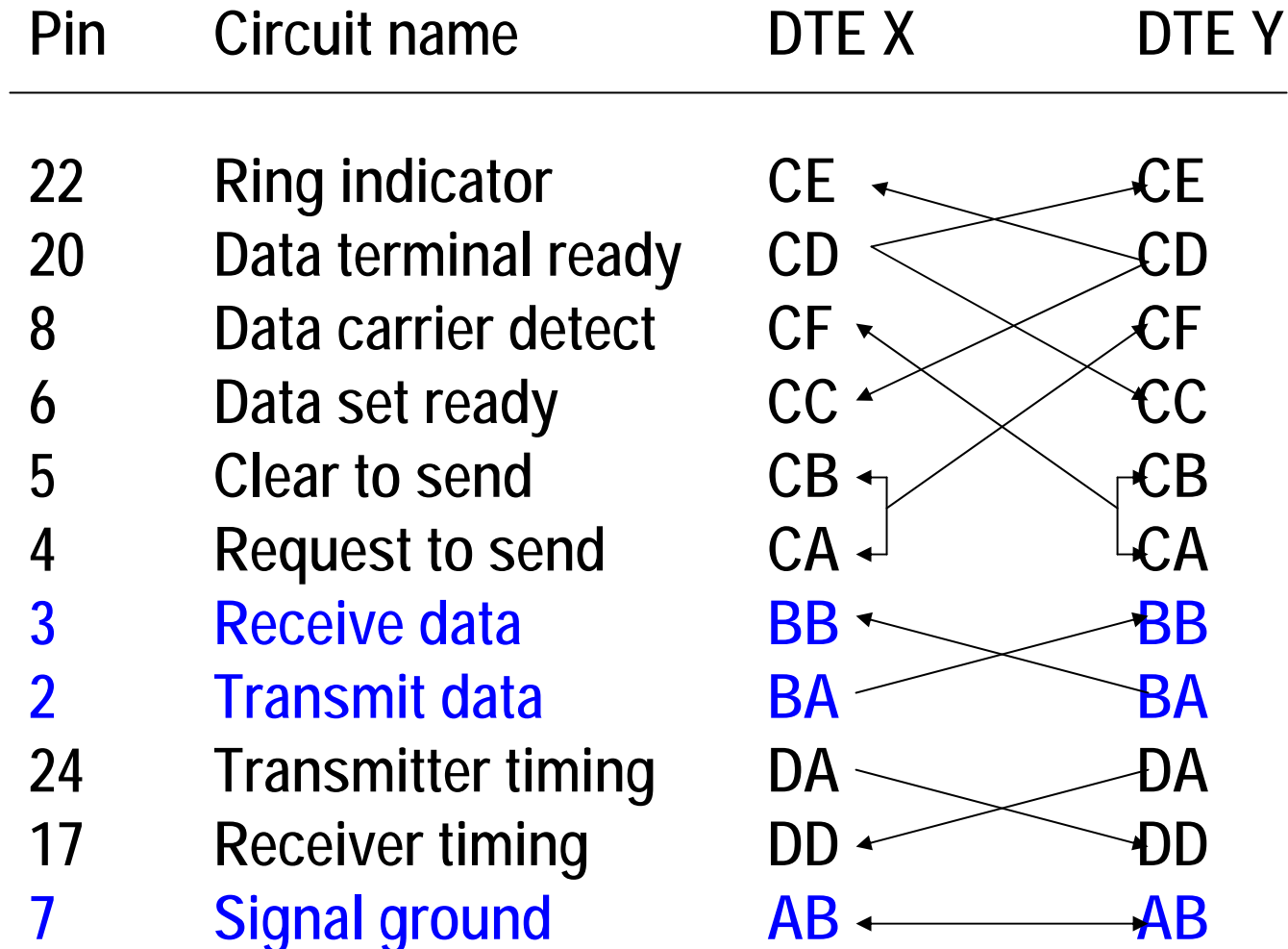
# Data Transmission Errors

- **Framing error**: when a received character is improperly framed by the start and stop bits; it is **set by the absence of stop bit**.
- **Receive overrun**: one or more characters in the data stream were lost.
- **Parity error**: received character parity error.



# Null Modem Connection

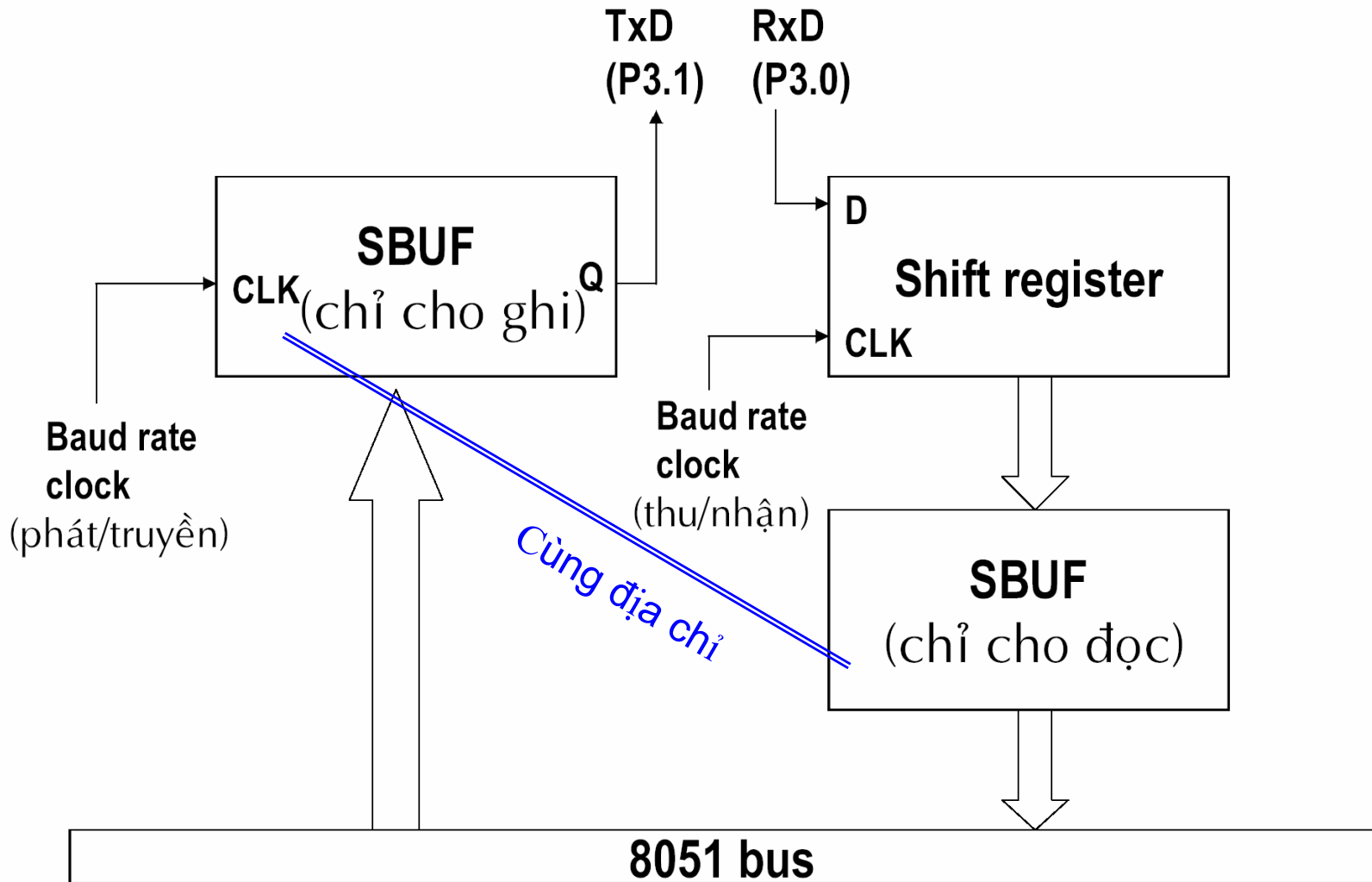
When 2 DTEs are very close, no need to use modems to connect them.



# Interfacing a $\mu\text{p}$ with 232

- Transmission:
  - Convert data from parallel to serial format
  - Add start, stop, and optional parity bits to the serial data.
  - Translate the voltage levels to those of the EIA-232-E standard. (transceiver chip: MAX 232, MC145403, etc.)
- Reception:
  - Translate the voltage levels of the incoming data stream to those used in the  $\mu\text{p}$ .
  - Detect the start and stop bits.
  - Optionally perform parity checking
  - Convert serial data into parallel format.

# Sơ đồ khối cổng nối tiếp của 8051



# Các đặc tính của cổng nối tiếp 8051

- Song công (ful-duplex)
- Có đệm khi thu (bộ đệm 1 byte)
- Có 4 chế độ làm việc (0 - 3)
- Lập trình dễ dàng để:
  - Chọn tốc độ (cố định (on-chip OSC/12 hoặc OSC/64), hoặc thay đổi được (T1))
  - Chọn chế độ làm việc
  - Cho chạy (SCON)
- Dễ sử dụng
  - Truyền dữ liệu:           MOV SBUF, A
  - Nhận dữ liệu:   MOV A, SBUF

# Thanh ghi điều khiển công nối tiếp SCON

Bit	Ký hiệu	Địa chỉ	Mô tả
SCON.7	SM0	9FH	Chế độ công nối tiếp bit 0 (xem bảng 3.22)
SCON.6	SM1	9EH	Chế độ công nối tiếp bit 1 (xem bảng 3.22)
SCON.5	SM2	9DH	Chế độ công nối tiếp bit 2. Cho phép truyền thông đa xử lý trong các chế độ 2 và 3; RI sẽ không được kích hoạt nếu bit 9 của bộ thu là 0.
SCON.4	REN	9CH	(Receiver ENable) cho phép bộ thu. Phải được đặt lên 1 để thu (nhận) các ký tự
SCON.3	TB8	9BH	Bit truyền 8. Bit thứ 9 được truyền/phát trong các chế độ 2 và 3; được đặt lên 1 hay xóa về 0 bằng phần mềm.
SCON.2	RB8	9AH	Bit nhận 8. Bit thứ 9 nhận được.
SCON.1	TI	99H	Cờ ngắt phát/truyền. Được đặt lên 1 khi kết thúc truyền ký tự; được xóa bằng phần mềm.
SCON.0	RI	98H	Cờ ngắt thu/nhận. Được đặt lên 1 khi kết thúc nhận ký tự; được xóa bằng phần mềm.

# Các chế độ hoạt động của cổng nối tiếp

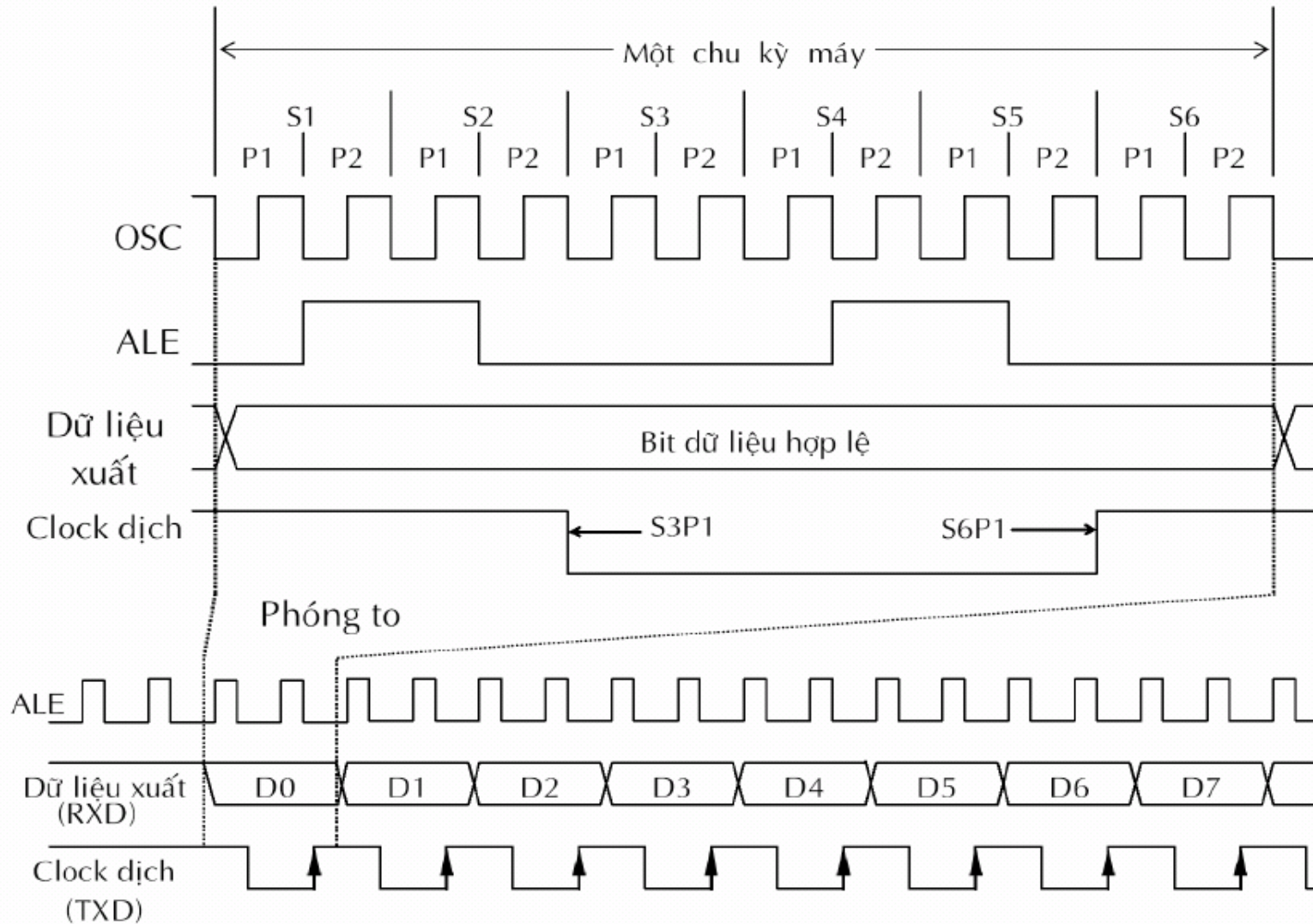
SM0	SM1	Chế độ	Mô tả	Tốc độ baud
0	0	0	Thanh ghi fích	Cố định (tần số dao động ÷ 12)
0	1	1	UART 8 bit	Thay đổi được (chỉnh bằng timer)
1	0	2	UART 9 bit	Cố định (tần số dao động ÷ 12)
1	1	3	UART 9 bit	Thay đổi được (chỉnh bằng timer)

Trước khi sử dụng cổng nối tiếp, SCON được khởi tạo trị với chế độ đúng, và v.v... Thí dụ, các lệnh sau

```
MOV SCON, #01010010B
```

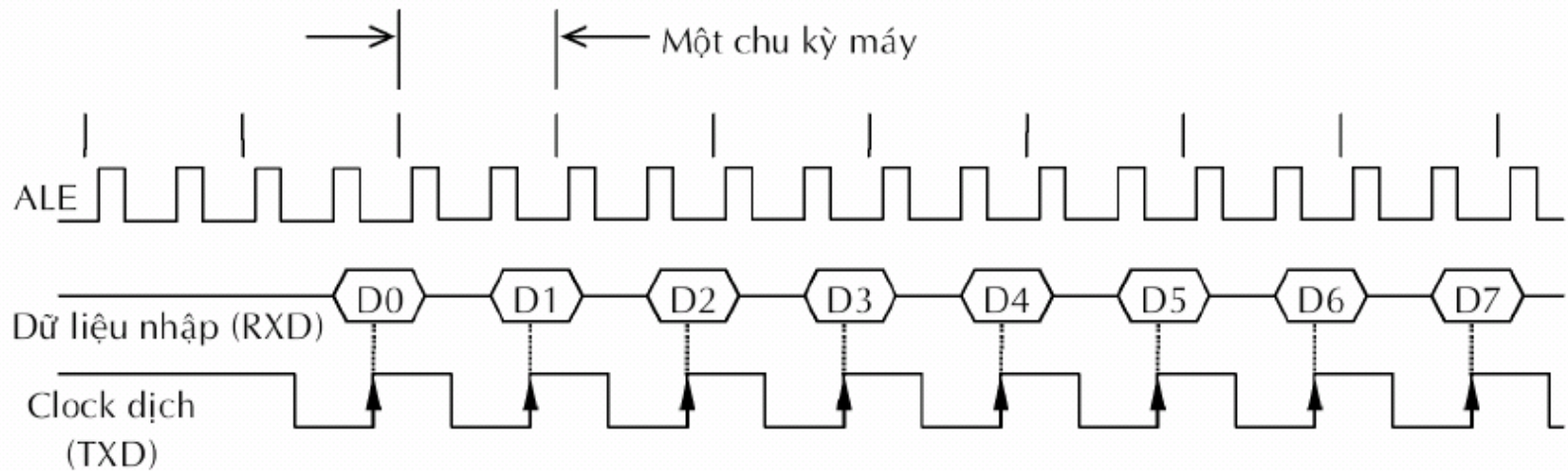
khởi tạo trị cho cổng nối tiếp ở chế độ 1 (SM0/SM1=0/1), cho phép bộ thu (REN=1) và đặt cờ ngắt phát (T1=1) để chỉ bộ phát sẵn sàng hoạt động.

# Chế độ 0: Thanh ghi dịch 8 bit (1/2)

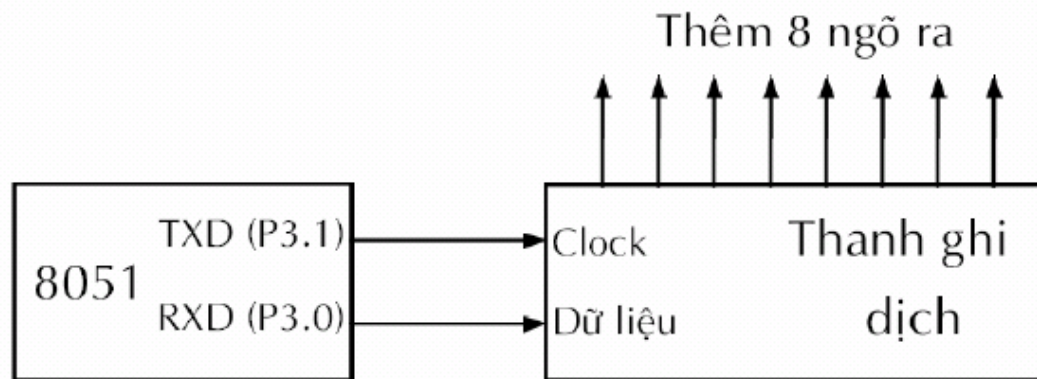


Hình 3.30 Định thì phát (truyền) cổng nối tiếp với chế độ 0.

# Chế độ 0: Thanh ghi dịch 8 bit (2/2)



**Hình 3.31** Định thì thu (nhận) cổng nối tiếp với chế độ 0.

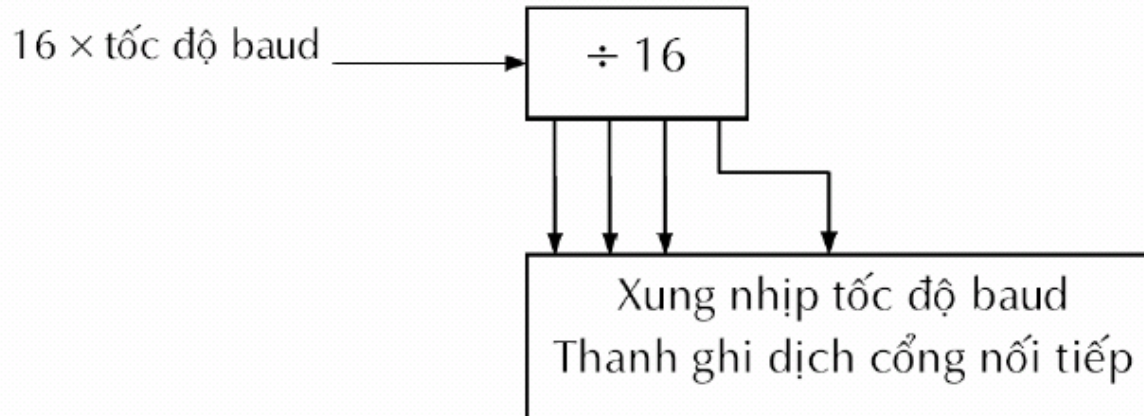


**Hình 3.32** Mở rộng ngõ ra với chế độ thanh ghi dịch cổng nối tiếp.



# Chế độ 1: UART 8 bit có tốc độ baud thay đổi được (1/3)

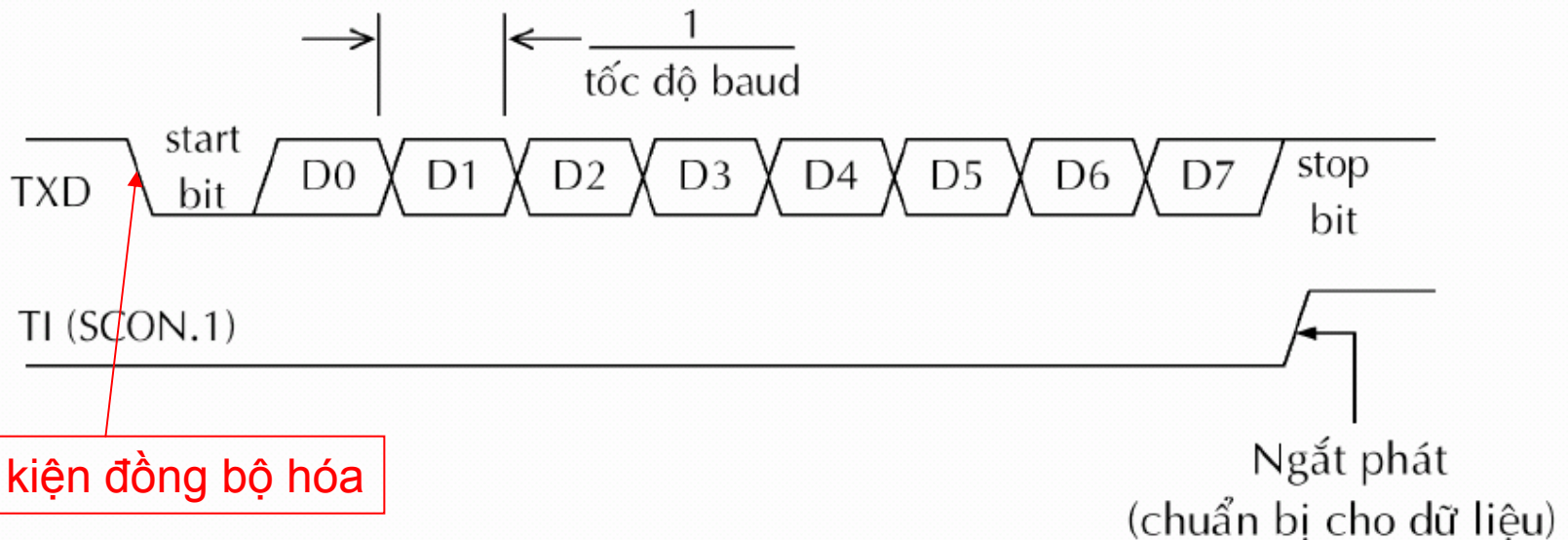
- Ở chế độ 1, 10 bit được phát đi trên TXD hoặc được thu vào ở RXD. Các bit này bao gồm 1 bit bắt đầu (luôn luôn là 0), 8 bit dữ liệu (LSB đi trước), và 1 bit dừng (luôn luôn là 1).
- Để hoạt động bộ thu, bit dừng đi vào RB8 trong SCON. Trong 8051, tốc độ baud được đặt bằng tốc độ tràn của Timer 1.



Hình 3.33 Cấp xung nhịp cổng nối tiếp.

# Chế độ 1: UART 8 bit có tốc độ baud thay đổi được (2/3)

- Việc phát được khởi động bằng cách ghi vào SBUF, nhưng không thật sự bắt đầu cho đến khi cạnh xuống kế của bộ đếm chia 16 cung cấp tốc độ baud cổng nối tiếp. Dữ liệu được dịch ra trên đường TXD bắt đầu bằng bit bắt đầu (start bit), theo sau là 8 bit dữ liệu, và kế đến là bit dừng. Chu kỳ của mỗi bit là nghịch đảo của tốc độ baud được lập trình trong timer. Cờ ngắt để phát (TI) được đặt lên 1 ngay khi bit dừng xuất hiện trên TXD.



Hình 3.34 Đặt cờ TI của cổng nối tiếp lên 1.

# Chế độ 1: UART 8 bit có tốc độ baud thay đổi được (3/3)

- Việc thu được khởi động bởi chuyển tiếp 1 sang 0 ở RXD. Bộ đếm chia 16 được reset tức thời để đồng chỉnh các số đếm với chuỗi bit đến. Chuỗi bit đến được lấy mẫu ở giữa 16 lần đếm.
- Giả sử bộ thu phát hiện bit bắt đầu hợp lệ thì tiếp tục nhận ký tự. Bit bắt đầu được bỏ qua và 8 bit dữ liệu được nhip vào thanh ghi dịch cổng nối tiếp. Khi tất cả 8 bit đã được nhip vào thì xảy ra:
  - Bit thứ 9 (bit dừng) được nhip vào RB8 trong SCON.
  - SBUF được nạp với 8 bit dữ liệu, và
  - Cờ ngắt của bộ thu (RI) được đặt lên 1.
- Tuy nhiên các điều trên chỉ xảy ra nếu có các điều kiện sau:
  - $RI = 0$ , và
  - $SM2 = 1$ , và bit dừng nhận được = 1, hoặc  $SM2=0$ .
- Yêu cầu  $RI = 0$  để bảo đảm rằng phần mềm đã đọc ký tự trước (và đã xóa RI).kiện thứ 2 thì hơi phức tạp, nhưng chỉ áp dụng trong chế độ truyền thông đa xử lý. Nó hàm ý “Không được đặt RI lên 1 trong chế độ truyền thông đa xử lý khi bit dữ liệu thứ chín là 0”.

# Chế độ 2: UART 9 bit có tốc độ baud cố định

- Khi  $SM0 = 1$  và  $SM1 = 0$ , cổng nối tiếp hoạt động ở chế độ 2 như UART 9 bit với tốc độ baud cố định.
- Mười một bit được phát hoặc thu: 1 bit bắt đầu, 8 bit dữ liệu, 1 bit dữ liệu thứ 9 lập trình được, và 1 bit dừng.
- Khi phát, bit thứ 9 là nội dung của bit TB8 trong SCON.
- Khi thu, bit thứ 9 thu được được đặt trong RB8.
- Tốc độ baud ở chế độ 2 là  $1/32$  hay  $1/64$  tần số mạch dao động trên chip (xem thêm mục tốc độ baud cổng nối tiếp).

# Chế độ 3: UART 9 bit có tốc độ baud thay đổi được

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

# Khởi tạo và truy cập các thanh ghi cổng nối tiếp

- Các tác vụ với cổng nối tiếp
  - Cho phép bộ thu
  - Truyền (thu/phát) dữ liệu nối tiếp
  - Thêm bit kiểm tra chẵn lẻ
  - Các cờ ngắt
  - Đặt tốc độ baud cho cổng nối tiếp
  - Truyền thông đa xử lý

# Cho phép bộ thu – Truyền dữ liệu nối tiếp

## Sử dụng các bit trong thanh ghi SCON

- Cho phép bộ thu (REN – bit 4)

```
SETB  REN  Hoặc
```

```
MOV   SCON, #xxx1xxxxB ; với x=0/1
```

- Các cờ ngắt RI, TI: Các cờ này được bật lên bởi phần cứng nhưng phải được xóa bằng phần mềm.
- RI = 1: Bộ đệm nhận đầy; TI = 1: Bộ đệm phát rỗng

### 1. Thu nối tiếp:

```
WAIT: JNB RI, WAIT
      CLR RI
      MOV A, SBUF
```

### 2. Phát nối tiếp:

```
WAIT: JNB TI, WAIT
      CLR TI
      MOV SBUF, A
```

# Thêm bit kiểm tra chẵn lẻ

## – bit kiểm tra chẵn lẻ - TB8

- kiểm tra chẵn/lẻ (nếu không có dòng (\*) là chẵn)

; qua bit nhớ C

```
MOV C, P
```

```
CPL C ; kiểm tra lẻ (*)
```

```
MOV TB8, C
```

```
MOV SBUF, A
```

; ở chế độ 1: 7-bit + 1 parity-bit

```
CLR ACC.7
```

```
MOV C, P
```

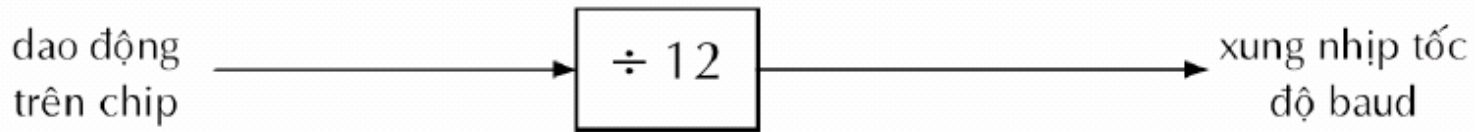
```
MOV ACC.7, C
```

```
MOV SBUF, A
```

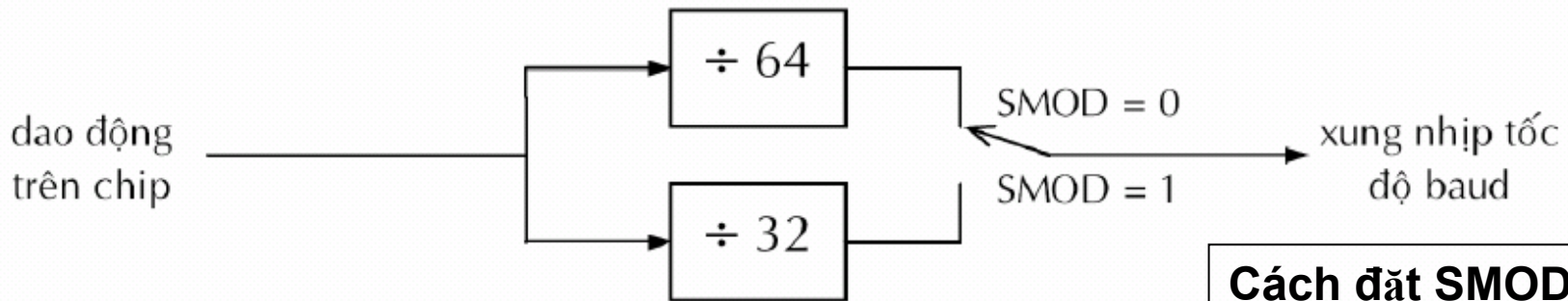
- Nếu Parity luôn luôn bằng 1  $\Rightarrow$  có 2 bit dừng



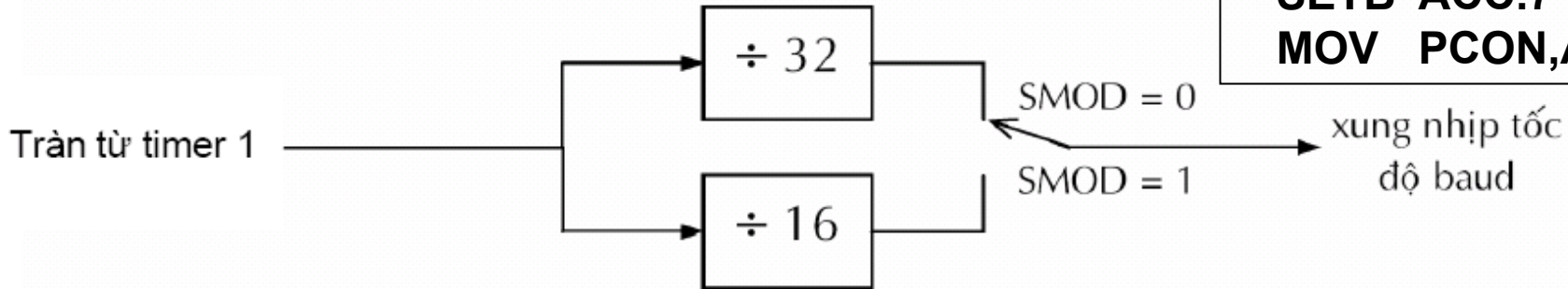
# Đặt tốc độ baud cho cổng nối tiếp



a) chế độ 0



b) chế độ 2



c) các chế độ 1 và 3.

**Cách đặt SMOD lên 1**  
**MOV A,PCON**  
**SETB ACC.7**  
**MOV PCON,A**

Hình 3.35 Các nguồn xung nhịp cho cổng nối tiếp.

# Sử dụng Timer 1 làm xung nhịp tốc độ baud

- Khởi tạo trị TMOD ở chế độ timer 8 bit tự động nạp lại trị đầu (chế độ 2 của timer) và đặt giá trị nạp lại vào TH1 để cho tốc độ tràn bằng tốc độ baud. TMOD được khởi tạo trị như sau:

```
MOV TMOD, #0010xxxxB
```

với “x” có giá trị 0 hay 1 cần cho Timer 0.

- Đặt tốc độ baud rất thấp bằng cách dùng Timer 1 chế độ 16 bit , với TMOD = #0001xxxxB. Tuy nhiên phải khởi tạo trị lại các thanh ghi TL1/TH1 sau mỗi lần tràn. Việc này sẽ được thực hiện trong chương trình phục vụ ngắt.
- Một lựa chọn khác là cung cấp xung nhịp Timer1 bằng xung nhịp bên ngoài qua chân T1 (P3.5). Tốc độ baud là 1/32 tốc độ tràn của Timer 1.
- Do đó công thức để xác định tốc độ baud trong các chế độ 1 và 3 là

Tốc độ baud = 1/32 Tốc độ tràn của Timer 1

**Thí dụ:** hoạt động 1200 baud cần tốc độ tràn như sau

$$1200 = 1/32 \text{ Tốc độ tràn của Timer 1}$$

$$\text{Tốc độ tràn của Timer 1} = 32 \times 1200 = 38400 = 38.4 \text{ KHz}$$

Với XTAL=12 MHz thì trị nạp cho T1 là  $1000\text{KHz}/38.8\text{KHz}=26.04 \sim 26$ <sup>34</sup>

**Bảng 3.23** Tóm tắt tốc độ baud.

Tốc độ baud	Tần số thạch anh	SMOD	Giá trị nạp lại vào TH1	Tốc độ baud thật	Sai số (%)
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

# Thí dụ: Khởi tạo trị cho cổng nối tiếp

Viết chuỗi lệnh để khởi tạo trị cổng nối tiếp hoạt động như UART 8 bit với tốc độ 2400 baud. Sử dụng Timer 1 để cung cấp xung nhịp tốc độ baud.

Bài giải.

Với thí dụ này, có 4 thanh ghi phải được khởi tạo trị là: SCON, TMOD, TCON, và TH1. Giá trị cần để khởi tạo trị cho các thanh ghi như sau

	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SCON:	0	1	0	1	0	0	1	0

	GATE	C/T	M1	M0	GATE	C/T	M1	M0
TMOD:	0	0	1	0	0	0	0	0

	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TCON:	0	1	0	0	0	0	0	0

TH1:	1	1	1	1	0	0	1	1
------	---	---	---	---	---	---	---	---

# Thí dụ: Chương trình xuất ký tự

Viết chương trình con OUTCHR để phát mã ASCII 7 bit trong thanh ghi tích lũy A ra cổng nối tiếp của 8051 với bit kiểm tra lỗi được thêm vào như bit thứ 8. Trở về từ chương trình con không làm thay đổi trị của thanh ghi tích lũy trước khi gọi chương trình con.

## Bài giải.

Thí dụ này và thí dụ kế minh họa 2 trong số các chương trình con thông dụng nhất trên các hệ vi tính có gắn thiết bị đầu cuối RS232: xuất ký tự (OUTCHR) và nhập ký tự (INCHAR).

```
                ORG    8100H
OUTCHR:        MOV    C, P            ; đặt bit kiểm tra lỗi vào cờ C
                CPL    C            ; đổi sang kiểm tra lỗi
                MOV    ACC.7, C      ; thêm vào mã ký tự
AGAIN:        JNB    TI, AGAIN       ; TX có rảnh ? Không : kiểm tra lại
                CLR    TI            ; Truyền được → xóa cờ TI và
                MOV    SBUF, A; gửi ký tự
                CLR    ACC.7         ; bỏ bit kiểm tra lỗi và
                RET                    ; quay về
                END
```

Thí dụ, các lệnh sau phát mã ASCII cho chữ Z đến thiết bị nối tiếp gắn vào cổng nối tiếp của 8051:

```
MOV    A, # 'Z'  
CALL   OUTCHR  
(tiếp tục)
```

# Thí dụ: Chương trình con nhập ký tự

Viết chương trình con INCHAR để nhập 1 ký tự từ cổng nối tiếp của 8051 và quay về với mã ASCII 7 bit trong thanh ghi tích lũy. Ngoại trừ bit kiểm tra lỗi trong bit thứ 8 nhận được và đặt cờ nhớ lên 1 nếu có sai về kiểm tra chẵn lẻ.

```
ORG 8100H
INCHAR: JNB RI, $ ; đợi ký tự
        CLR RI ; xóa cờ
        MOV A, SBUF ; đọc ký tự vào A
        MOV C, P ; với kiểm tra lỗi trong A, P sẽ = 1
        CPL C ; bù để chỉ có lỗi
        CLR ACC.7 ; bỏ bit kiểm tra chẵn lẻ
        RET
END
```