

Microcontroller 8051

- Phần 2



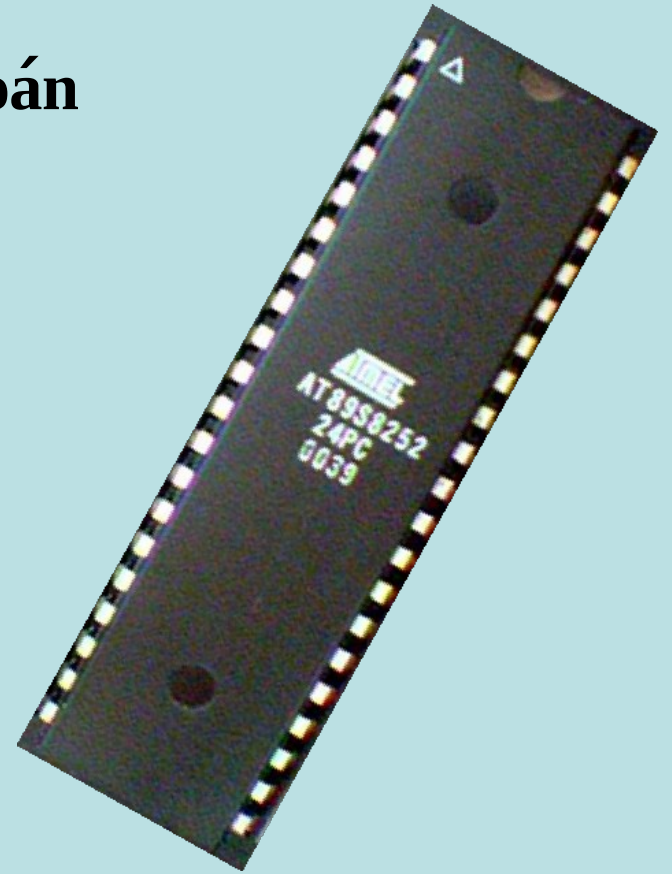
Microprocessors course

Phạm Thế Duy

(phamtheduy@gmail.com)

Nội dung:

- ◆ Lập trình I/O, xử lý bit
- ◆ Tạo trễ bằng phần mềm và tính toán thời gian trễ.
- ◆ Lập trình Timer/Counter
 - Timers
 - Counters
- ◆ Lập trình ngắt
- ◆ Cổng nối tiếp



Lập trình xử lý bit I/O

- Lập trạng thái các bit P1 có thể thực hiện bằng 3 cách:

- Cách 1: gửi dữ liệu ra P1 bằng ACC

```
BACK: MOV A,#55H ;A=01010101B
      MOV P1,A
      ACALL DELAY
      MOV A,#0AAH ;A=10101010B
      MOV P1,A
      ACALL DELAY
      SJMP BACK
```

- Cách 2: Truy cập trực tiếp P1


```
BACK: MOV P1,#55H ;P1=01010101B
      ACALL DELAY
      MOV P1,#0AAH ;P1=10101010B
      ACALL DELAY
      SJMP BACK
```

- Sử dụng đặc tính Read-modify-write:

```
MOV P1,#55H ;P1=01010101B
AGAIN: XRL P1,#0FFH
      ACALL DELAY
      SJMP AGAIN
```

- Lệnh **XRL P1,#0FFH** thực hiện phép toán EX-OR P1 và FFH` làm lật trạng thái P1.

XỬ LÝ BIT

- Đôi khi cần truy cập chỉ 1 hoặc 2 bit của một cổng 8 bit
- Bảng mô tả tên các chân của các cổng. 
- Ví dụ:

Viết chương trình thực hiện các công việc sau:.

(a) Giám sát ngõ vào P1.2 cho đến khi gặp mức cao.

(b) Khi P1.2 mức cao, ghi giá trị 45H tới P0 và

(c) Tạo 1 xung (H-to-L) trên đường P2.3.

Giải:

```
                SETB P1.2                ;khởi động P1.2 là ngõ vào
                MOV  A,#45H                ;A=45H
AGAIN:          JNB  P1.2,AGAIN            ;ngưng lặp khi P1.2=1
                MOV  P0,A                  ;gửi A tới P0
                SETB P2.3                  ; P2.3 = 1
                CLR  P2.3                  ; P2.3 =0
```

Chú ý

1. JNB: nhảy nếu bit bằng 0 nhảy nếu P1.2 = 0'
2. Trạng thái xung H-to-L tạo ra bằng các lệnh SETB và CLR.

Địa chỉ bit của các cổng

P0	P1	P2	P3	Port Bit
P0.0	P1.0	P2.0	P3.0	D0
P0.1	P1.1	P2.1	P3.1	D1
P0.2	P1.2	P2.2	P3.2	D2
P0.3	P1.3	P2.3	P3.3	D3
P0.4	P1.4	P2.4	P3.4	D4
P0.5	P1.5	P2.5	P3.5	D5
P0.6	P1.6	P2.6	P3.6	D6
P0.7	P1.7	P2.7	P3.7	D7



Tạo trễ bằng lệnh và tính toán thời gian trễ

- Chu kỳ máy (Machine cycle)
- Để CPU thực hiện xong một lệnh cần tốn một số chu kỳ clock nhất định (tùy theo từng lệnh). Trong các hệ thống họ 8051 12 chu kỳ clock thực hiện một tác vụ trong lệnh được gọi là một **Chu kỳ máy** (machine cycles).
- Tần số thạch anh dao động cho họ 8051 có thể thay đổi từ 4MHz tới 30 MHz (tùy thuộc và từng loại Chip cụ thể). Các hệ thống 8051 thông dụng thường sử dụng thạch anh 11.0592 MHz, để tốc độ công nối tiếp của 8051 tương thích với tốc độ công nối tiếp của máy tính PC.
- Trong 8051, một machine cycle chiếm 12 chu kỳ tần số dao động thạch anh.

Example:

For an 8051 system of 11.0592 MHz, find how long it takes to execute each of the following instructions.

- a) MOV R3,#55
- b) LJMP
- c) MUL AB
- d) DEC R3
- e) SJMP
- f) NOP

Solution:

Instruction	Machine cycles	Time to execute
MOV R3,#55	1	1x1.085 us=1.085us
LJMP	2	2x1.085 us=2.17 us
MUL AB	4	4x1.085 us=4.34 us
DEC R3	1	1x1.085 us=1.085us
SJMP	2	2x1.085 us=2.17 us
NOP	1	1x1.085 us=1.085us

Ví dụ:

Tìm thời gian thực hiện chương trình con sau, giả sử hệ thống chạy với tần số thạch anh 11.0592 MHz

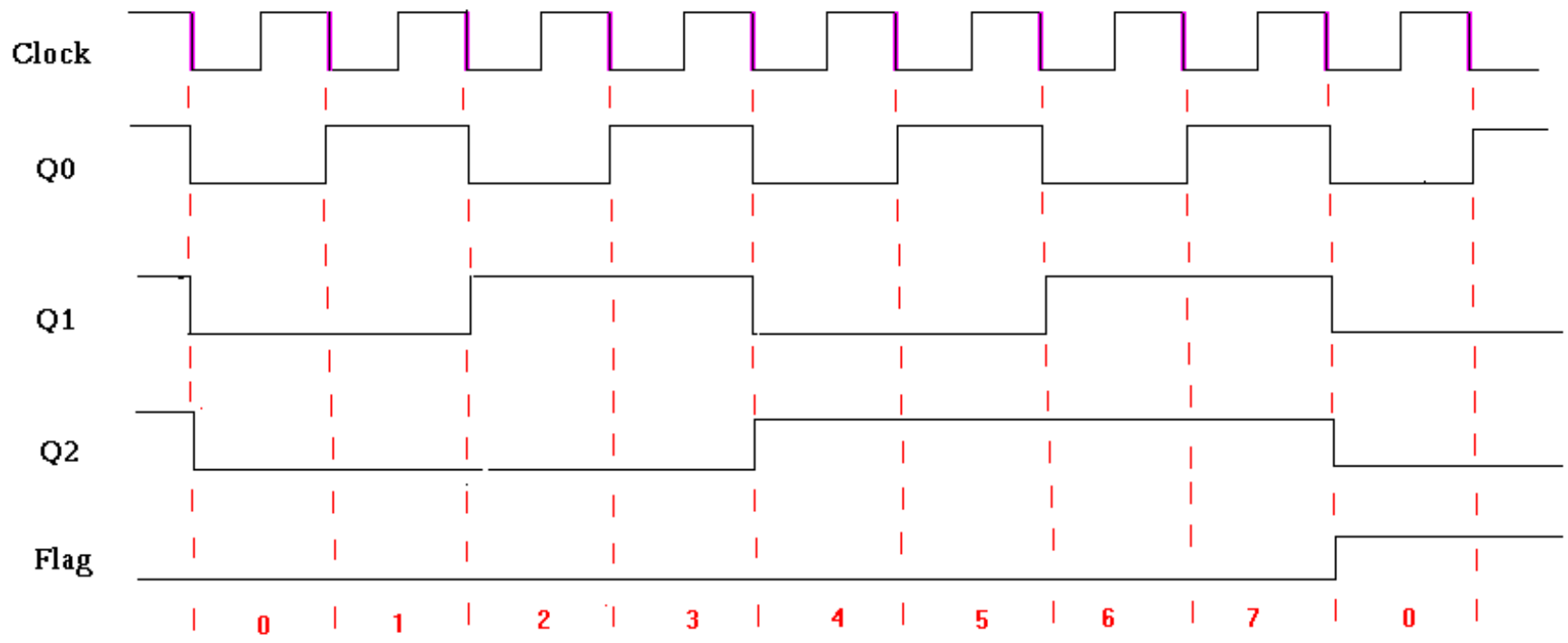
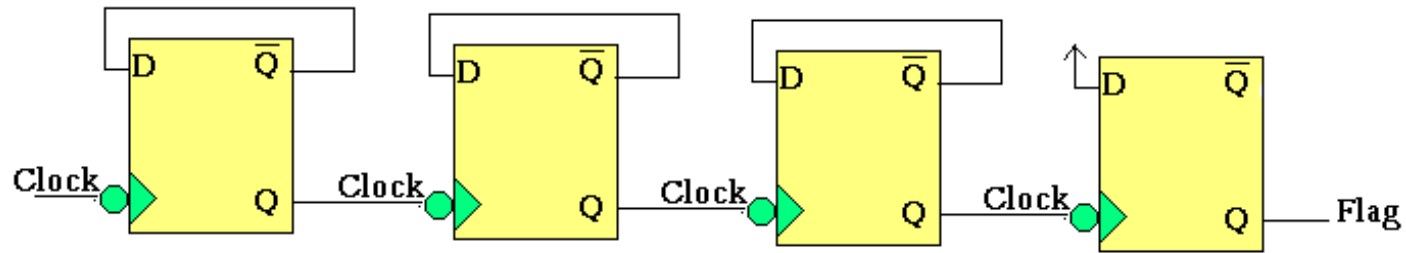
DELAY:	MOV	R3,#250	;	1 MC
HERE:	NOP		;	1 MC
	NOP		;	1 MC
	NOP		;	1 MC
	NOP		;	1 MC
	DJNZ	R3,HERE	;	2 MC
	RET		;	1 MC

Giải:

$$[250 \times (1+1+1+1+2)+2] \times 1.085 \text{ us} = 1627.5 \text{ us}$$

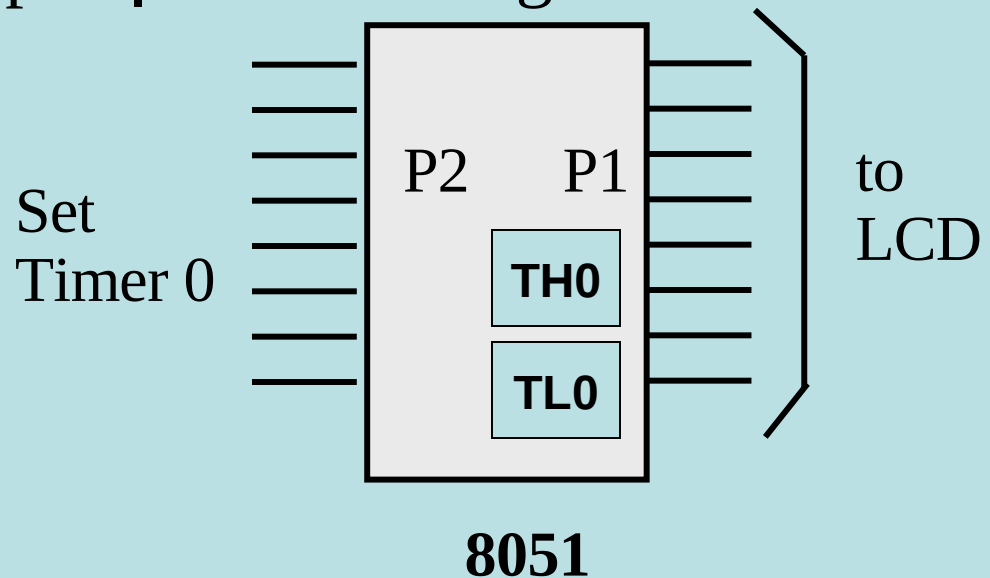
Lập trình Timers /Counters

- 8051 có 2 timers/counters: timer/counter 0 và timer/counter 1. Chúng có thể sử dụng như:
 1. Một **timer** để tạo ra một khoảng thời gian trễ.
 - Nguồn clock cung cấp cho bộ đếm được lấy từ bộ dao động thạch anh bên trong 8051.
 2. Một **counter**.
 - **Nguồn clock được cấp từ bên ngoài (External input)** đặc trưng cho số sự kiện và thanh ghi bên trong sẽ chứa kết quả đếm được.
 - Ví dụ: Số xung cung cấp cho bộ đếm có thể tương ứng với số người đi qua một cổng, hoặc số vòng quay của một bánh xe, hoặc một sự kiện nào đó có thể tạo thành xung.



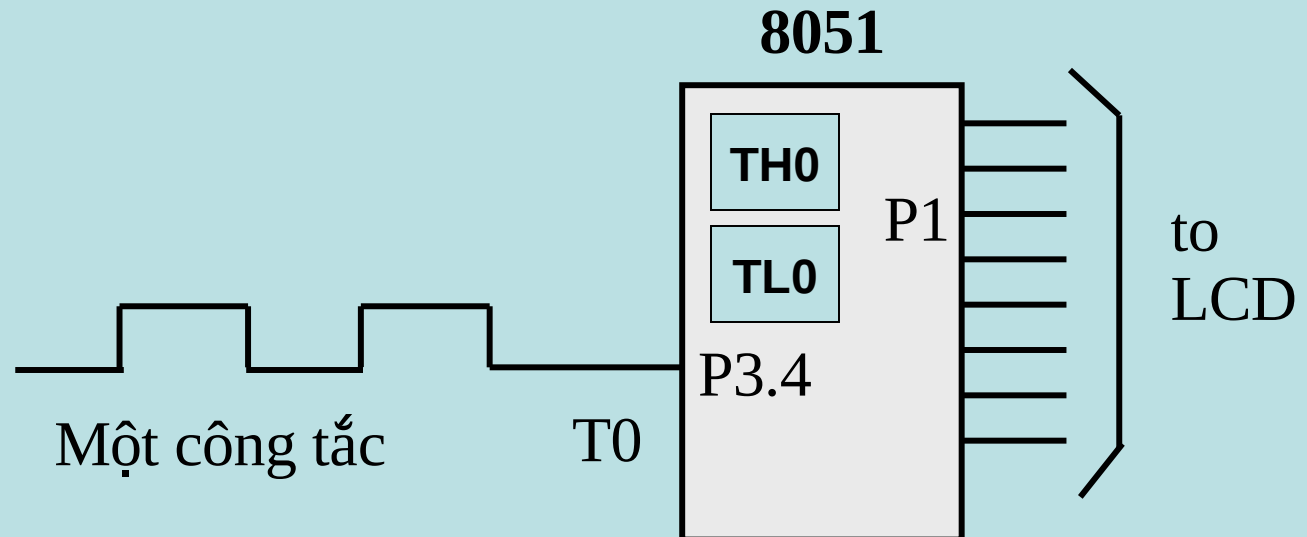
Timer

- Khởi động giá trị cho các thanh ghi
- Cho phép Timer bắt đầu đếm lên.
- Clock được lấy từ bên trong (bằng 1 machine cycle)
- Khi thanh ghi đếm đặt giá trị cực đại, thêm 1 clock sẽ quay về 0, 8051 Lập một bit để thông báo hết thời gian.



Counter

- Đếm số sự kiện xảy ra
 - Giữ số sự kiện đếm được trong thanh ghi đếm.
 - Chân T0 (P3.4) sử dụng làm ngõ vào cho Counter 0
 - Chân T1 (P3.5) sử dụng làm ngõ vào cho Counter 1



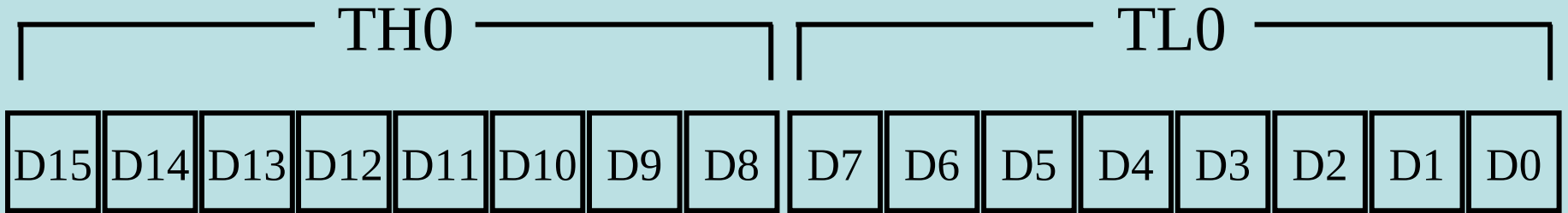
Các thanh ghi sử dụng cho Timer/Counter

- TH0, TL0, TH1, TL1
- TMOD (Thanh ghi chế độ – Timer Mode)
- TCON (Thanh ghi điều khiển – Timer Control)
- Do 8052 có 3 timers/counters, nên cấu tạo các thanh ghi TMOD và TCON sẽ khác.
 - T2CON (Timer 2 control register), TH2 và TL2 chỉ sử dụng cho 8052.

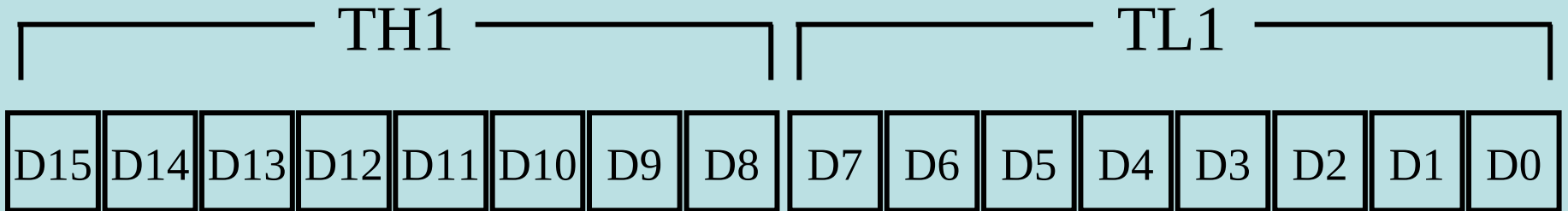
Các thanh ghi đếm của Timer

- Cả hai timer 0 và timer 1 đều có các thanh ghi đếm 16 bit.
 - Các thanh ghi này chứa:
 - Thời gian trễ khi sử dụng Timer.
 - Số sự kiện đếm được khi sử dụng Counter
 - Timer 0: **TH0** & **TL0**
 - Timer 0 high byte, timer 0 low byte
 - Timer 1: **TH1** & **TL1**
 - Timer 1 high byte, timer 1 low byte
 - Các thanh ghi 16-bit này có thể truy cập phần cao và thấp một cách riêng rẽ.

Các thanh ghi Timer



Timer 0



Timer 1

Thanh ghi TMOD

- Thanh ghi chế độ Timer: **TMOD**

MOV TMOD, #21H

- Là thanh ghi 8-bit.
- Sử dụng để khởi động chế độ làm việc cho cả hai Timer
 - 4 bits thấp khởi động Timer 0 (cho bằng 0000 nếu không sử dụng)
 - 4 bits cao khởi động Timer 1 (cho bằng 0000 nếu không sử dụng)

(MSB) không được địa chỉ hoá theo bit.

(LSB)

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

Các bit thanh ghi TMOD

GATE Khi bit này = 1 Timer/counter chỉ được cho phép khi đồng thời INTx và TRx bằng 1. Khi bit này được xoá Timer được cho phép khi bit TRx được lập.

C/T Khi bit này bằng 0, sử dụng Timer (lấy clock từ bên trong). Khi bit này được lập = 1, sử dụng counter (nguồn clock lấy từ bên ngoài).

M1 Bit chế độ 1

M0 Bit chế độ 0

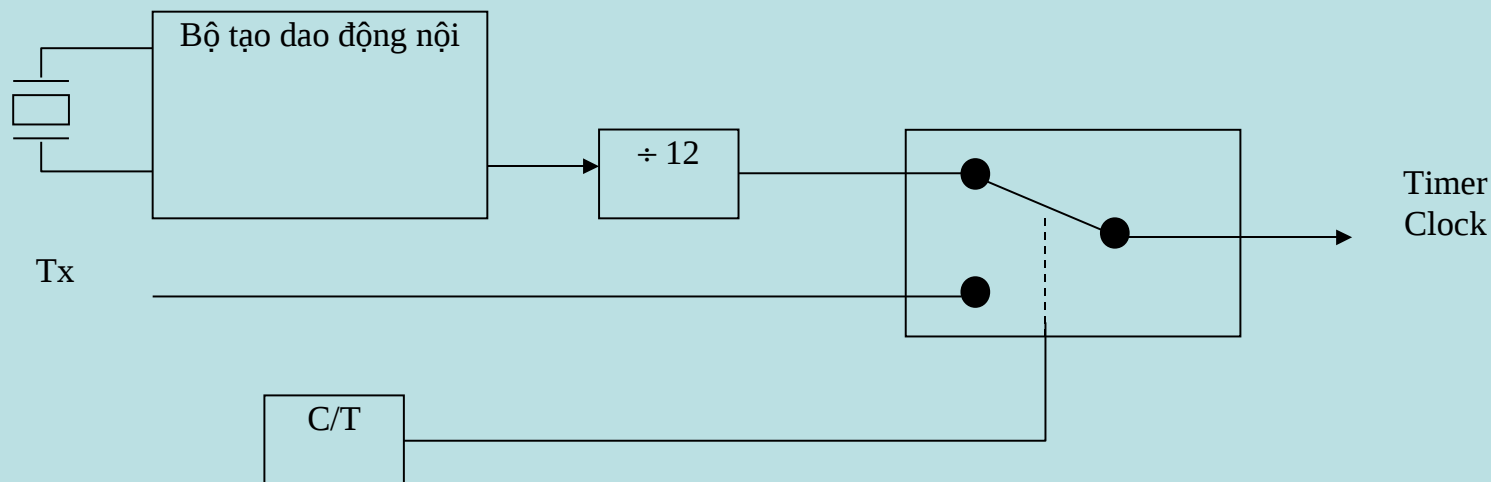
(MSB)

(LSB)

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

C/T (Clock/Timer)

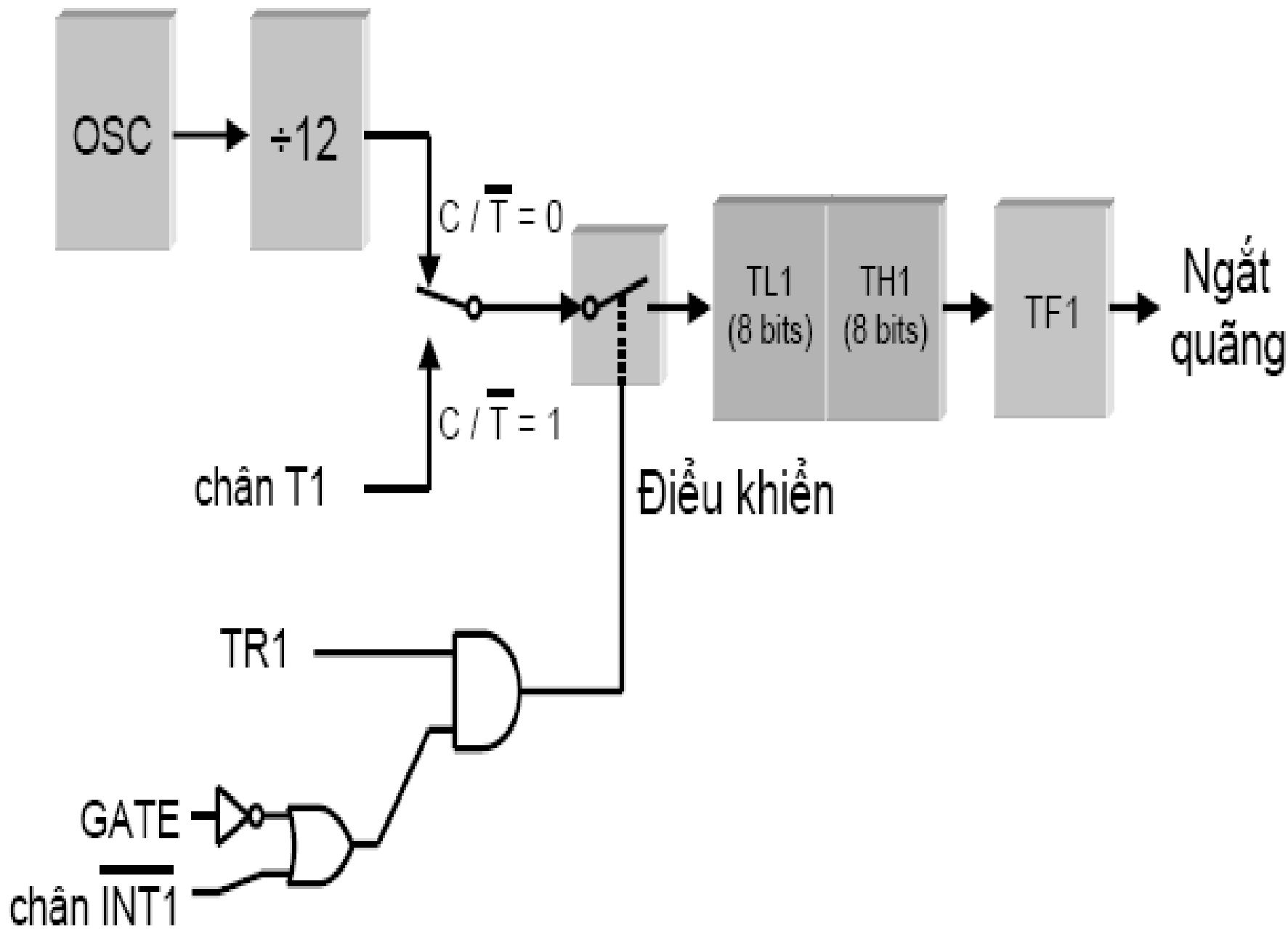
- Bit này sử dụng để chọn chế độ tạo thời gian trễ (Timer) hoặc chế độ đếm sự kiện (Counter)
- $C/T = 0$: timer
- $C/T = 1$: counter



Sơ đồ cung cấp xung nhịp cho timer

Gate

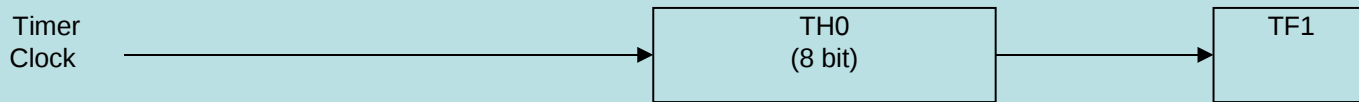
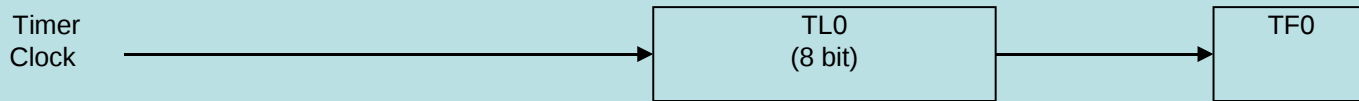
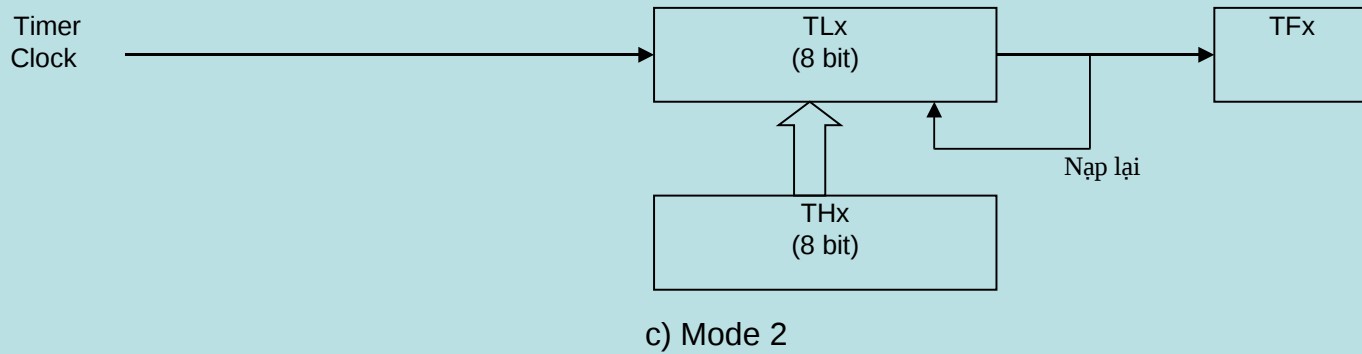
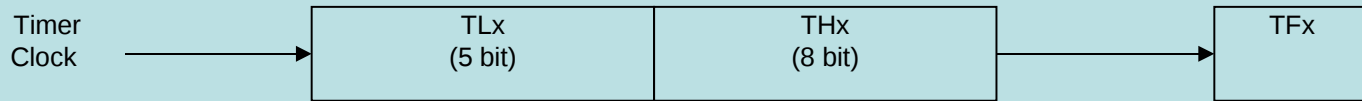
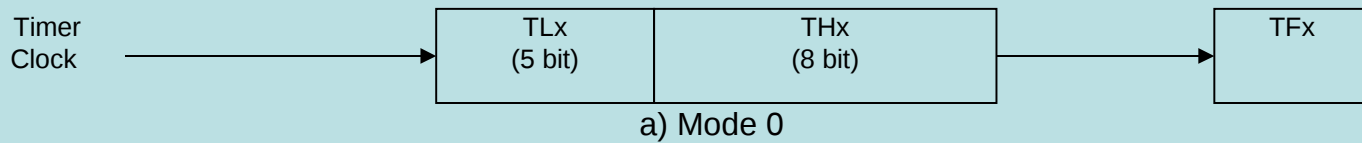
- Chọn cho phép chạy và dừng Timer ở trong hay ngoài.
 - GATE=0
 - **Điều khiển bên trong.**
 - Cho phép chạy và dừng Timer bằng lệnh phần mềm.
 - Lập xoá bit TR để cho phép chạy/dừng Timer.
 - GATE=1
 - **Điều khiển từ bên ngoài**
 - Cho phép chạy/dừng Timer bằng lệnh và tín hiệu phần cứng cung cấp từ bên ngoài.
 - Timer/counter được cho phép chỉ khi TR = 1 và ngõ vào INT = 1.



M1, M0

- M0 và M1 chọn chế độ hoạt động cho timers 0 & 1.

<u>M1</u>	<u>M0</u>	<u>C.độ</u>	<u>Hoạt động</u>
0	0	0	Chế độ đếm 13-bit timer 8-bit THx + 5-bit TLx (x= 0 or 1)
0	1	1	Chế độ đếm 16-bit timer 8-bit THx + 8-bit TLx
1	0	2	Chế độ tự động nạp lại 8-bit (auto reload) THx giữ giá trị để nạp lại TLx giữ giá trị đếm.
1	1	3	Chế độ đếm 8 bit.



d) Mode 3

Hình 8.10: Các chế độ hoạt động của các timer

Ví dụ:

Tìm giá trị của TMOD để Timer 0 làm việc ở chế độ 2, sử dụng clock bên trong, và sử dụng lệnh để cho phép chạy và dừng Timer.

Giải:

timer 1 **timer 0**

↙ ↘

TMOD= **0000** **0010** Timer 1 không sử dụng.

↑ ↑

Timer 0, **mode 2**,

C/T = 0 sử dụng nguồn clock nội (timer)

gate = 0 cho phép bằng lệnh (**software**)

Thanh ghi TCON (1/2)

- Thanh ghi điều khiển timer: **TCON**
 - Bốn bit cao sử dụng cho timer/counter, Bốn bit thấp sử dụng cho ngắt.
 - **TR** (bit điều khiển chạy timer)
 - TR0 cho Timer/counter 0; TR1 cho Timer/counter 1.
 - TR được lập xoá bằng lệnh để cho phép timer/counter on/off.
 - TR=0: off (stop)

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Timer 1		Timer0		cho Interrupt			

Thanh ghi TCON (2/2)

- **TF** (timer flag, control flag)
 - TF0 cho timer/counter 0; TF1 cho timer/counter 1.
 - TF giống như cờ nhớ. Khởi động, TF=0. Khi TH-TL = FFFFH thêm 1 clock trở về 0000, TF sẽ được lập lên 1.
 - TF=0 : chưa hết giá trị cần đếm.
 - TF=1: hết giá trị cần đếm
 - Nếu cho phép ngắt timer, TF=1 sẽ cho chạy chương trình ngắt ISR.

(MSB)

(LSB)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Timer 1		Timer0		cho Interrupt			

Các lệnh tác động cho thanh ghi điều khiển timer

Cho Timer 0

SETB TR0	=	SETB TCON.4
----------	---	-------------

CLR TR0	=	CLR TCON.4
---------	---	------------

SETB TF0	=	SETB TCON.5
----------	---	-------------

CLR TF0	=	CLR TCON.5
---------	---	------------

Cho timer 1

SETB TR1	=	SETB TCON.6
----------	---	-------------

CLR TR1	=	CLR TCON.6
---------	---	------------

SETB TF1	=	SETB TCON.7
----------	---	-------------

CLR TF1	=	CLR TCON.7
---------	---	------------

TCON: Timer/Counter Control Register

TF1

TR1

TF0

TR0

IE1

IT1

IE0

IT0

Timer chế độ 1

- Dưới đây chúng ta sử dụng timer 0, timer 1 hoàn toàn tương tự.
- **16-bit** timer (TH0 và TL0)
- TH0-TL0 sẽ bắt đầu tăng liên tục sau mỗi clock nếu $TR0 = 1$. Và 8051 sẽ ngừng tăng TH0-TL0 khi $TR0=0$.
- Timer làm việc với nguồn clock bên trong, hay nói cách khác nó sẽ đếm lên sau mỗi chu kỳ máy.
- Khi timer (TH0-TL0) đạt giá trị cực đại FFFFH, nó sẽ quay về 0000, và TF0 sẽ bằng 1.
- Người lập trình có thể kiểm tra TF0 và ngừng timer 0.

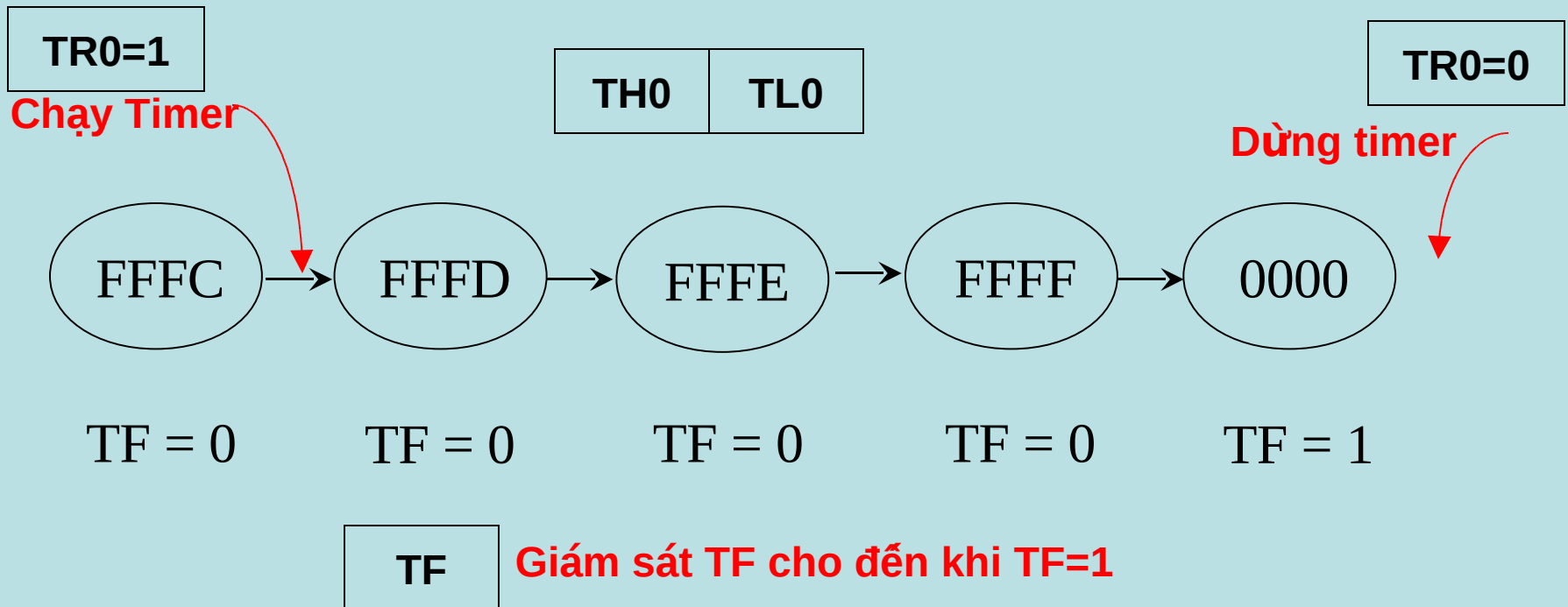
Các bước hoạt động Chế độ 1 (1/3)

1. Chọn chế độ 1 cho timer 0
 - **MOV TMOD, #01H**
2. Khởi động giá trị cho TH0 và TL0.
 - **MOV TH0, #FFH**
 - **MOV TL0, #FCH**
3. Tốt nhất nên xoá cờ giám sát: TF0=0.
 - **CLR TF0**
4. Cho timer chạy.
 - **SETB TR0**

Các bước hoạt động Chế độ 1 (2/3)

1. 8051 bắt đầu đếm tăng trên TH0-TL0.

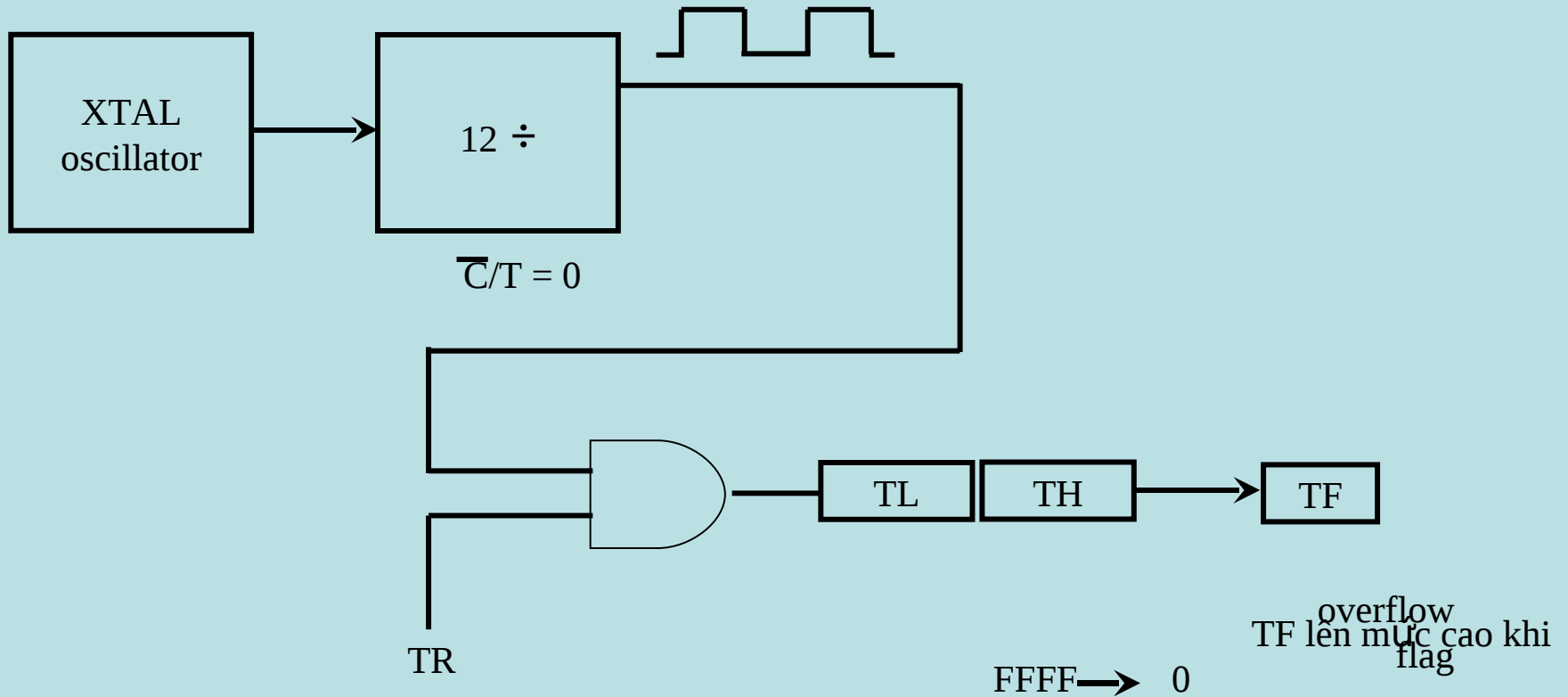
- TH0-TL0 = FFFCH, FFFDH, FFFEh, FFFFh, 0000h



Các bước hoạt động Chế độ 1 (3/3)

1. Khi TH0-TL0 đặt tới giá trị FFFFH và quay về 0000, 8051 lập TF0=1.
 - TH0-TL0= FFFE_H, FFFF_H, 0000_H (bây giờ TF0=1)
2. Giám sát cờ timer (TF) có thể thực hiện như sau:
 - AGAIN: JNB TF0, AGAIN
3. Xoá TR0 để ngừng quá trình đếm.
 - CLR TR0
4. Xoá cờ TF cho vòng đếm kế tiếp.
 - CLR TF0

Lập trình cho phép Chế độ 1



Tính toán thời gian trễ với XTAL = 11.0592 MHz

(a) theo hex

$(FFFF - YYXX + 1) \times$

1.085 μs với YYXX là giá trị khởi
động cho TH, TL

Chú ý giá trị YYXX tính theo hex.

(b) theo decimal

Đổi giá trị Hex YYXX trong
thanh ghi TH, TL thành giá trị
thập phân NNNNN

Tính theo thập phân sẽ là:

$(65536 - NNNNN) \times 1.085 \mu s$

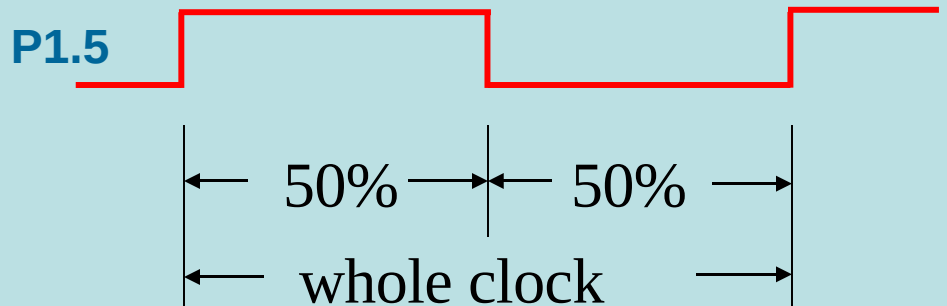
Ví dụ 8-4 (1/3)

Chương trình sau tạo sóng vuông với chu kỳ nhiệm vụ 50% (bán kỳ cao bằng bán kỳ thấp) trên đường P1.5, sử dụng Timer 0 tạo thời gian trễ.

Phân tích chương trình:

; Mỗi vòng lặp tạo $\frac{1}{2}$ chu kỳ xung vuông.

```
MOV TMOD, #01      ;Timer 0, mode 1(16-bit)
HERE: MOV TL0, #0F2H ;Timer value = FFF2H
MOV TH0, #0FFH
CPL P1.5
ACALL DELAY
SJMP HERE
```



Ví dụ 8-4 (2/3)

;Tạo thời gian trễ sử dụng timer 0

DELAY:

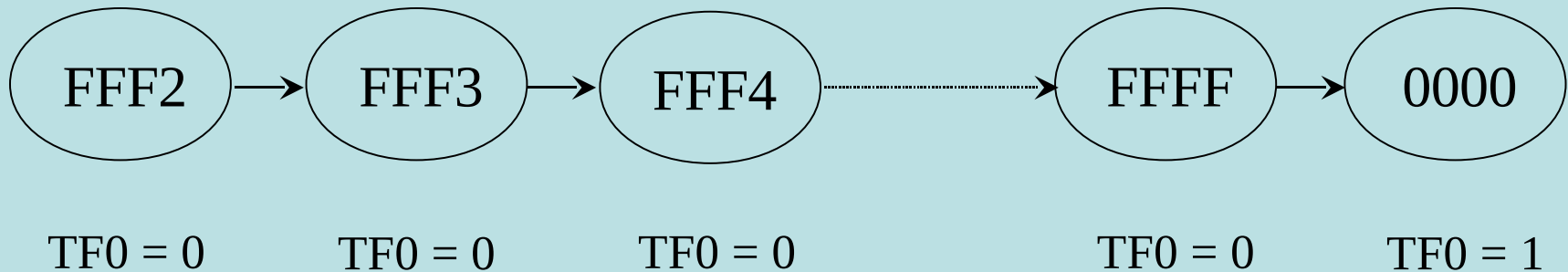
SETB TR0 ;cho timer 0 chạy

AGAIN: JNB TF0, AGAIN

CLR TR0 ;dừng timer 0

CLR TF0 ;xoá cờ timer 0

RET



Ví dụ 8-4 (3/3)

Giải:

Trong chương trình trên chú ý các bước sau.

1. Nạp TMOD = **0000 0001**.
2. Nạp TH0 – TL0 = **FFF2H**.
3. P1.5 lật trạng thái để tạo xung ngõ ra.
4. Gọi chương trình con tạo trễ sử dụng timer (DELAY)
5. Trong chương trình DELAY, timer 0 bắt đầu chạy bằng lệnh “**SETB TR0**” .
6. Timer 0 đếm lên sau mỗi chu kỳ clock tạo bởi bộ tạo dao động nội.

Timer đếm lên qua các trạng thái FFF3, FFF4, FFF5, FFF6, FFF7, FFF8, FFF9, FFFA, FFFB, FFFC, FFFD, FFFE, FFFFH. Thêm một clock bộ đếm quay về 0 và cơ timer được lập (**TF0 = 1**). Tại thời điểm này lệnh JNB sẽ ngưng lặp.

7. Timer 0 ngưng chạy bằng lệnh “**CLR TR0**”. Chương trình con DELAY kết thúc và quá trình được lặp lại.

Chú ý: khi quá trình lặp lại cần nạp lại giá trị cho TH và TL trong chương trình chính.

Ví dụ 8-9 (1/2)

Chương trình dưới đây tạo ra một xung vuông trên đường P1.5 sử dụng Timer 1 tạo thời gian trễ. Tìm tần số sóng vuông nếu XTAL = 11.0592 MHz. (Bỏ qua các lệnh trong vòng lặp)

```
MOV    TMOD, #10H           ;timer 1, chế độ 1
AGAIN: MOV    TL1, #34H      ; giá trị timer=7634H
        MOV    TH1, #76H
        SETB   TR1          ;chạy timer
BACK:   JNB    TF1, BACK
        CLR    TR1          ;dừng timer
        CPL    P1.5         ;1/2 chu kỳ tiếp theo
        CLR    TF1         ;xoá cờ timer 1
        SJMP  AGAIN        ; nạp lại giá trị timer1
```

Ví dụ 8-9 (2/2)

Giải:

Trong chế độ 1, chương trình cần phải nạp lại thanh ghi TH1, TL1 sau mỗi lần tiếp tục tạo sóng vuông.

$$\text{FFFFH} - 7634\text{H} + 1 = 89\text{CCH} = 35276 \text{ clock}$$

$$\text{Nửa chu kỳ} = 35276 \times 1.085 \mu\text{s} = 38.274 \text{ ms}$$

$$\text{Toàn chu kỳ} = 2 \times 38.274 \text{ ms} = 76.548 \text{ ms}$$

$$\text{Tần số} = 1 / 76.548 \text{ ms} = 13.064 \text{ Hz.}$$

Chú ý rằng xung vuông có phần cao và phần thấp bằng nhau.

Thời gian tính toán trên thời gian các lệnh trong vòng lặp được bỏ qua.

Tìm các giá trị của Timer

- Giải sử biết tổng thời gian trễ và $XTAL = 11.0592 \text{ MHz}$.
- Tìm giá trị cần thiết nạp cho TH, TL?
 - Chia thời gian trễ muốn tạo cho **$1.085 \mu s$** được **n** .
 - Lấy **$65536 - n$** , với **n** là giá trị thập phân tìm được trong bước 1.
 - Đổi giá trị tìm được trong bước 2 ra hex, ở đây **$yyxx$** là giá trị hex sẽ khởi động cho các thanh ghi của timer.
 - Nạp **TH = yy** và **TL = xx**.
- Ví dụ 8-4
- Có thể viết: `MOV TL, LOW(-100)`
`MOV TH, HIGH(-100)`

Ví dụ 8-12 (1/2)

Giả sử XTAL = 11.0592 MHz, viết chương trình tạo sóng vuông 50 Hz trên chân P2.3.

Giải:

Chương trình có thể thực hiện theo các bước sau.

(a) Một chu kỳ sóng vuông = $1 / 50 \text{ Hz} = 20 \text{ ms}$.

(b) Phần cao = phần thấp = 10 ms.

(c) $10 \text{ ms} / 1.085 \mu\text{s} = 9216$

$65536 - 9216 = 56320\text{D} = \text{DC00H}$.

(d) TL1 = 00H và TH1 = DCH.

Ví dụ 9-12 (2/2)

```
MOV    TMOD,#10H    ;timer 1, mode 1
AGAIN: MOV    TL1,#00    ;Giá trị = DC00H

MOV    TH1,#0DCH
SETB   TR1          ;start
BACK:  JNB    TF1, BACK
CLR    TR1          ;stop
CPL    P2.3
CLR    TF1          ;xoá TF1
SJMP   AGAIN       ; nạp lại giá trị
                    ;timer vì mode 1
                    ;không tự động nạp
                    ;lại
```


Tạo thời gian trễ lớn

- Thời gian trễ phụ thuộc vào 2 yếu tố:
 - Tần số thanh anh
 - Giá trị nạp cho thanh ghi timer 16-bit, TH & TL
- Thời gian trễ lớn nhất được thực hiện bằng cách nạp $TH=TL=0$. Làm thế nào nếu thời gian này không đủ?
- Ví dụ 8 – 13 trình bày cách tạo ra một thời gian trễ lớn hơn.

Ví dụ 8-13

Tìm thời gian trễ trong chương trình dưới đây (bỏ qua các lệnh trong vòng lặp).

```
MOV    TMOD, #10H
MOV    R3, #200
AGAIN: MOV    TL1, #08H
        MOV    TH1, #01H
        SETB  TR1
BACK:  JNB   TF1, BACK
        CLR   TR1
        CLR   TF1
        DJNZ  R3, AGAIN
```

Giải:

$TH - TL = 0108H = 264$ Decimal

$65536 - 264 = 65272$.

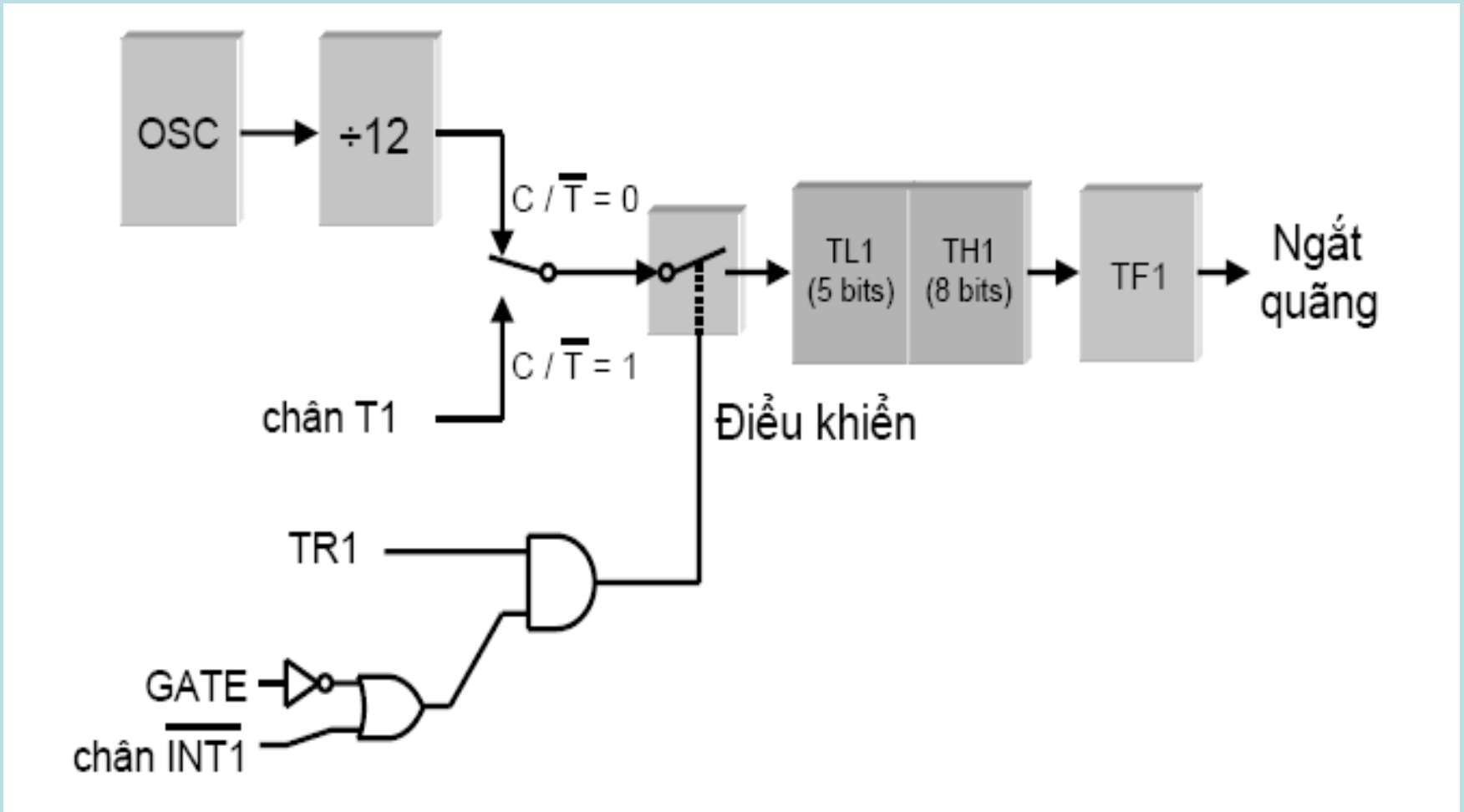
Thời gian một lần trễ = $65272 \times 1.085 \mu s = 70.820$ ms

Tổng thời gian trễ = 200×70.820 ms = 14.164024 seconds

Timer Chế độ 0

- Chế độ 0 của timer hoàn toàn giống chế độ 1 ngoại trừ việc chế độ 0 đếm **13-bit** chế độ 1 đếm 16-bit.
 - 8-bit TH0 + 5-bit TL0
- Giá trị giữ trong bộ đếm từ 0000 tới 1FFF trong TH0-TL0.
 - $2^{13}-1 = 2000H-1 = 1FFFH$
- TH0-TL0 sẽ đếm lên từ giá trị khởi động.
- Khi timer đạt tới giá trị cực đại 1FFFH, nó sẽ quay về 0000 ở clock tiếp theo, và TF0 được lập lên 1.

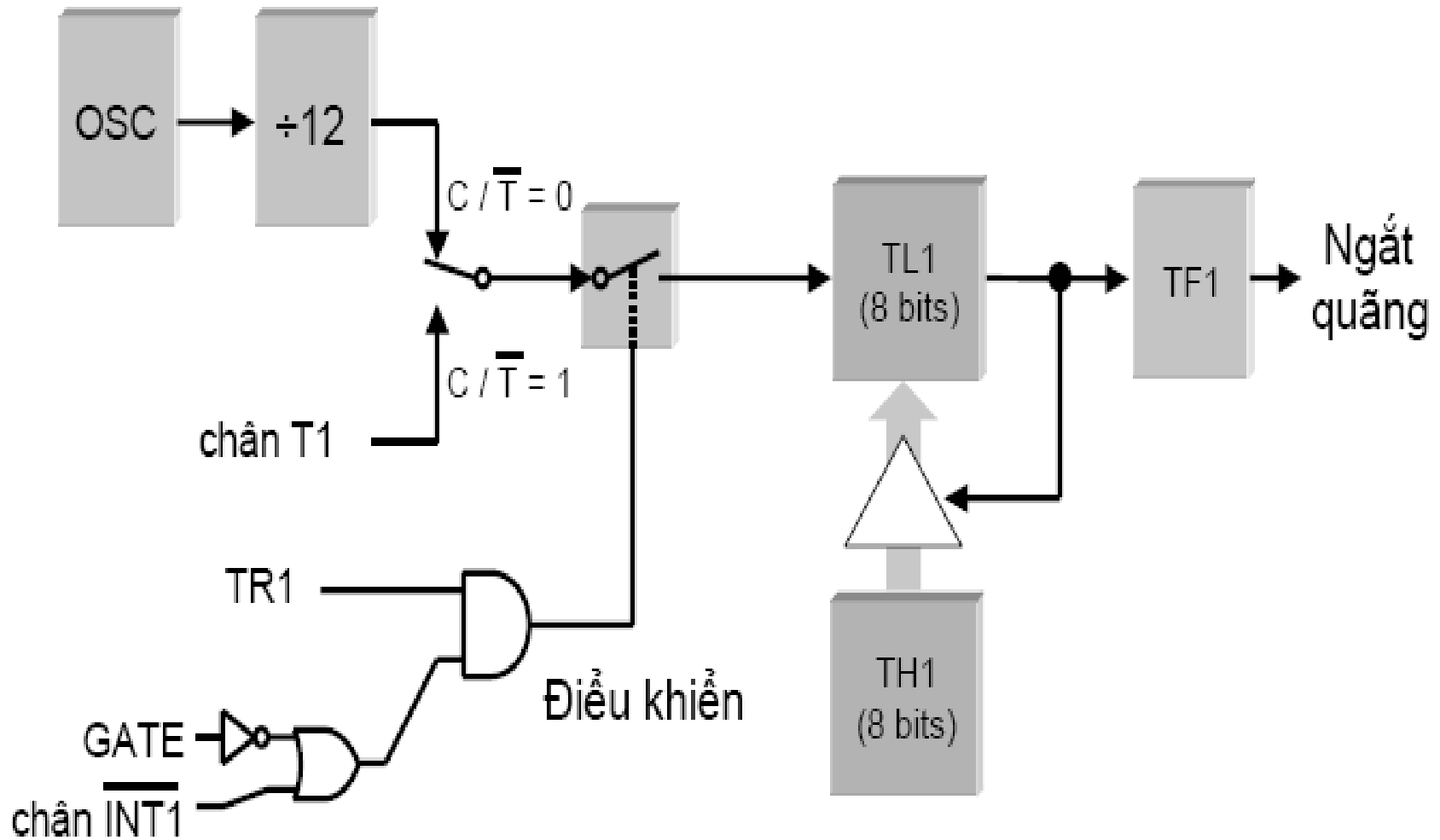
Cấu trúc hoạt động của timer trong chế độ 0



Timer Chế độ 2

- Đếm 8-bit.
 - Chỉ cho phép nạp giá trị từ 00 tới FFH vào TH0.
- Tự động nạp lại (Auto-reloading)
- TL0 tăng liên tục sau mỗi clock và khi tràn về 0 thì TF0=1.
- Trong ví dụ sau sẽ tạo ra thời gian trễ bằng 200 chu kỳ máy (machine cycle) bằng timer 0.
- Xem ví dụ 8-14 đến 8-16

Cấu trúc hoạt động của timer chế độ 2



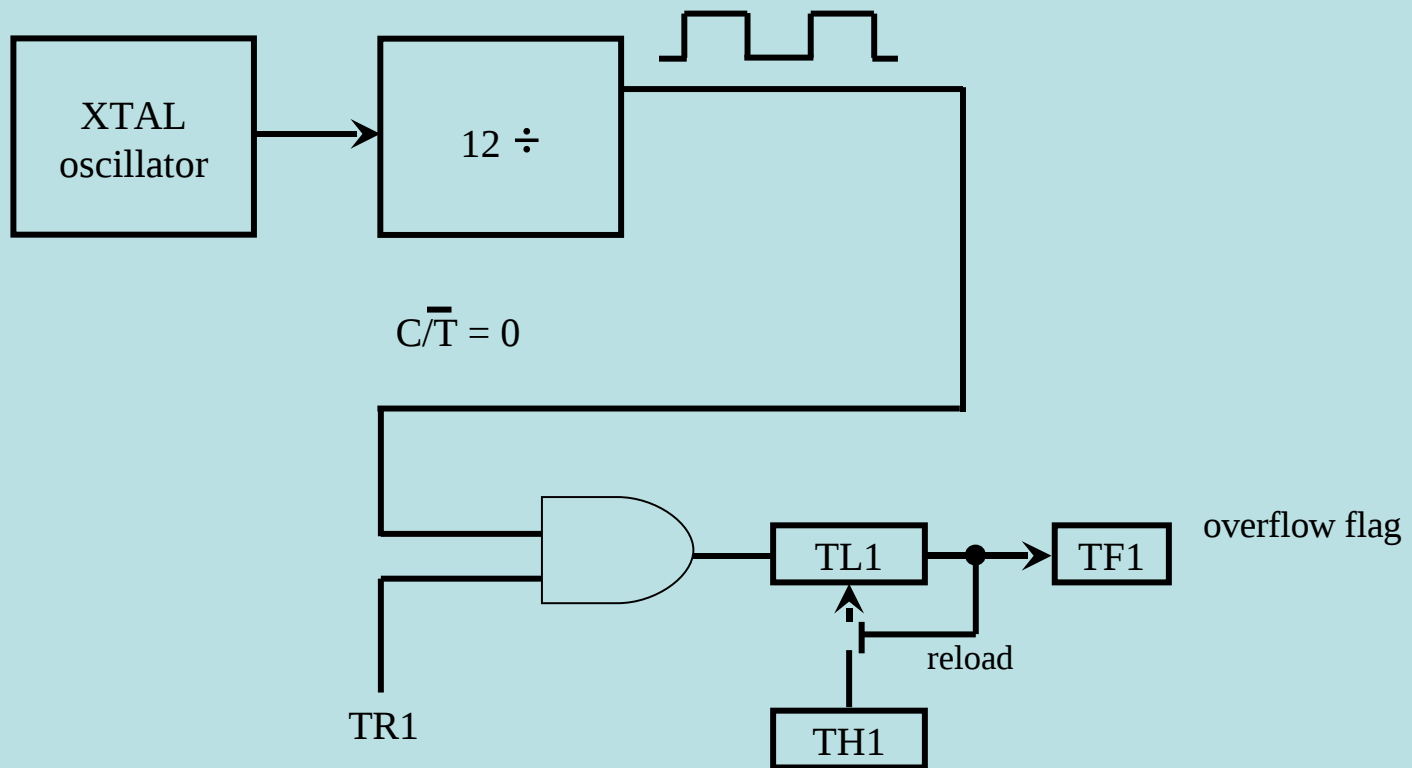
Các bước chạy Mode 2 (1/2)

1. Chọn timer 0 làm việc ở chế độ 2.
 - **MOV TMOD, #02H**
2. Khởi động giá trị cho TH0.
 - **MOV TH0, #38H**
3. Xoá cờ báo tràn TF0=0.
 - **CLR TF0**
4. Sau khi TH0 được nạp một giá trị 8-bit, 8051 sẽ chép qua TL0.
 - **TL0=TH0=38H**
5. Cho chạy Timer.
 - **SETB TR0**

Các bước chạy Mode 2 (2/2)

1. 8051 bắt đầu tăng giá trị trong TL0.
 - **TL0 = 38H, 39H, 3AH, ...**
 2. Khi TL0 tràn từ FFH về 00, 8051 lập TF0=1. Do đó, TL0 được tự động nạp lại bằng giá trị trong TH0.
 - **TL0 = FEH, FFH, 00H (TF0=1)**
 - 8051 tự động nạp lại **TL0=TH0=38H**.
 - Quay lại bước 6 (nghĩa là., TL0 tiếp tục tăng).
- Chú ý cần xoá TF0 khi TL0 tràn về 0. Vì cần phải giám sát TF0 trong chu kỳ tiếp theo.
 - Xoá TR0 để dừng quá trình đếm.

Timer 1 Chế độ 2 với nguồn clock trong



TF goes high
when FF \rightarrow 0

Ví dụ 8-15

Tìm tần số sóng vuông tạo ra trên chân P1.0.

Giải:

```
      MOV    TMOD, #2H    ;Timer 0, mode 2
      MOV    TH0, #6
AGAIN: MOV    R5, #256    ;đếm 250 lần
      ACALL DELAY
      CPL    P1.0
      SJMP AGAIN


---


DELAY: SETB  TR0        ;start
BACK: JNB   TF0, BACK
      CLR    TR0        ;stop
      CLR    TF0        ;xoá TF
      DJNZ R5, DELAY    ;timer 2: auto-reload
      RET
```

$T = 2 (256 \times 250 \times 1.085 \mu s) = 138.88 \text{ ms}, f = 72 \text{ Hz}.$

Ví dụ 8-16

Giả sử các timer được lập trình chế độ 2, tìm giá trị (theo hex) cần nạp vào cho TH trong mỗi trường hợp sau:

- (a) **MOV TH1, #-200** (b) **MOV TH0, #-60** (c) **MOV TH1, #-3**
(d) **MOV TH1, #-12** (e) **MOV TH0, #-48**

Giải:

Một số trình hợp dịch 8051 cho phép tính giá trị theo cách.

$$-200 = -C8H \Rightarrow \text{bù 2 của } -200 = 100H - C8H = 38H$$

<i>Decimal</i>	<i>Lấy bù 2 (giá trị TH)</i>
-200 = - C8H	38H
- 60 = - 3CH	C4H
- 3	FDH
- 12	F4H
- 48	D0H

Ví dụ 9-17 (1/2)

Tìm (a) tần số sóng vuông tạo ra trong đoạn chương trình (trang sau), và (b) chu kỳ nhiệm vụ của sóng vuông này.

Giải:

“**MOV TH0, #-150**” đếm 150 clocks.

Chương trình DELAY có thời gian trễ = $150 \times 1.085 \mu s = 162 \mu s$.

Phần cao của xung vuông gấp 2 phần thấp (chu kỳ nhiệm vụ = 66%).

Chu kỳ xung vuông = phần cao + phần thấp

$$= 325.5 \mu s + 162.25 \mu s = 488.25 \mu s$$

Tần số = 2.048 kHz.

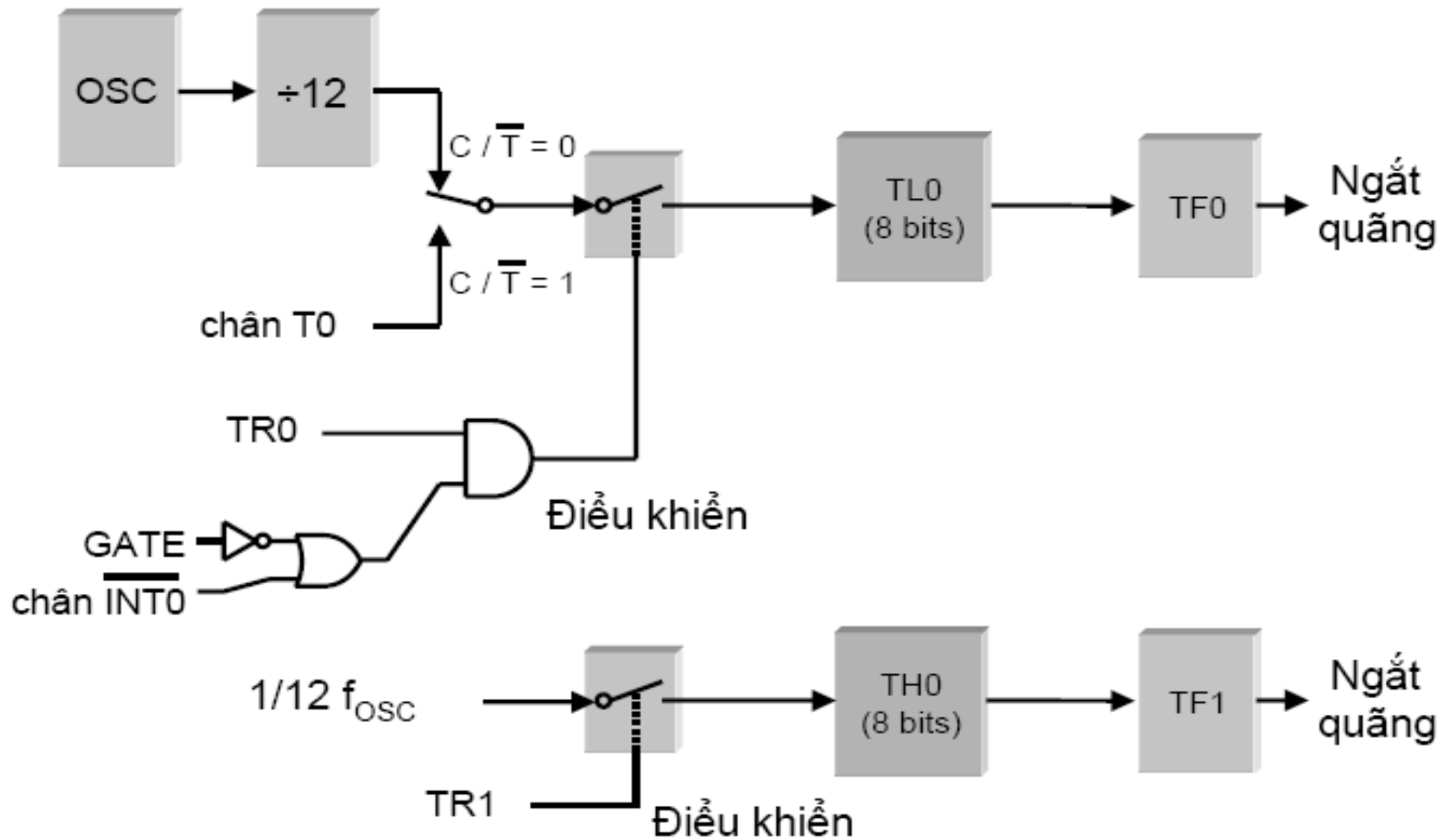
Ví dụ 8-17 (2/2)

```
MOV    TMOD,#2H    ;Timer 0,mode 2
MOV    TH0,#-150   ;Count=150
AGAIN:SETB P1.3
      ACALL DELAY  } high
      ACALL DELAY  } period
      CLR  P1.3    } low
      ACALL DEALY  } period
      SJMP AGAIN

DELAY:SETB TR0     ;start
BACK: JNB  TF0,BACK
      CLR  TR0     ;stop
      CLR  TF0     ;Xoá TF
      RET
```

Cấu trúc hoạt động timer chế độ 3

Đếm chế độ 3 : đếm 8 bit 2 chế độ



Counter

- Các timer của 8051 có thể sử dụng làm bộ đếm sự kiện (**counting events**) xảy ra bên ngoài 8051.
- Khi timer được sử dụng làm bộ đếm, xung cung cấp từ bên ngoài 8051 sẽ tăng giá trị trong TH, TL.
- Khi C/T=1, bộ đếm timer sẽ đếm lên sau mỗi chu kỳ xung cung cấp vào:
 - T0: ngõ vào timer 0 (chân 14, P3.4)
 - T1: ngõ vào timer 1 (chân 15, P3.5)

Các chân P3 sử dụng cho Timers 0 and 1

Chân	cổng	Chức năng	Mô tả
14	P3.4	T0	Timer/Counter 0 external input
15	P3.5	T1	Timer/Counter 1 external input

(MSB)

(LSB)

GATE

C/T=1

M1

M0

GATE

C/T=1

M1

M0

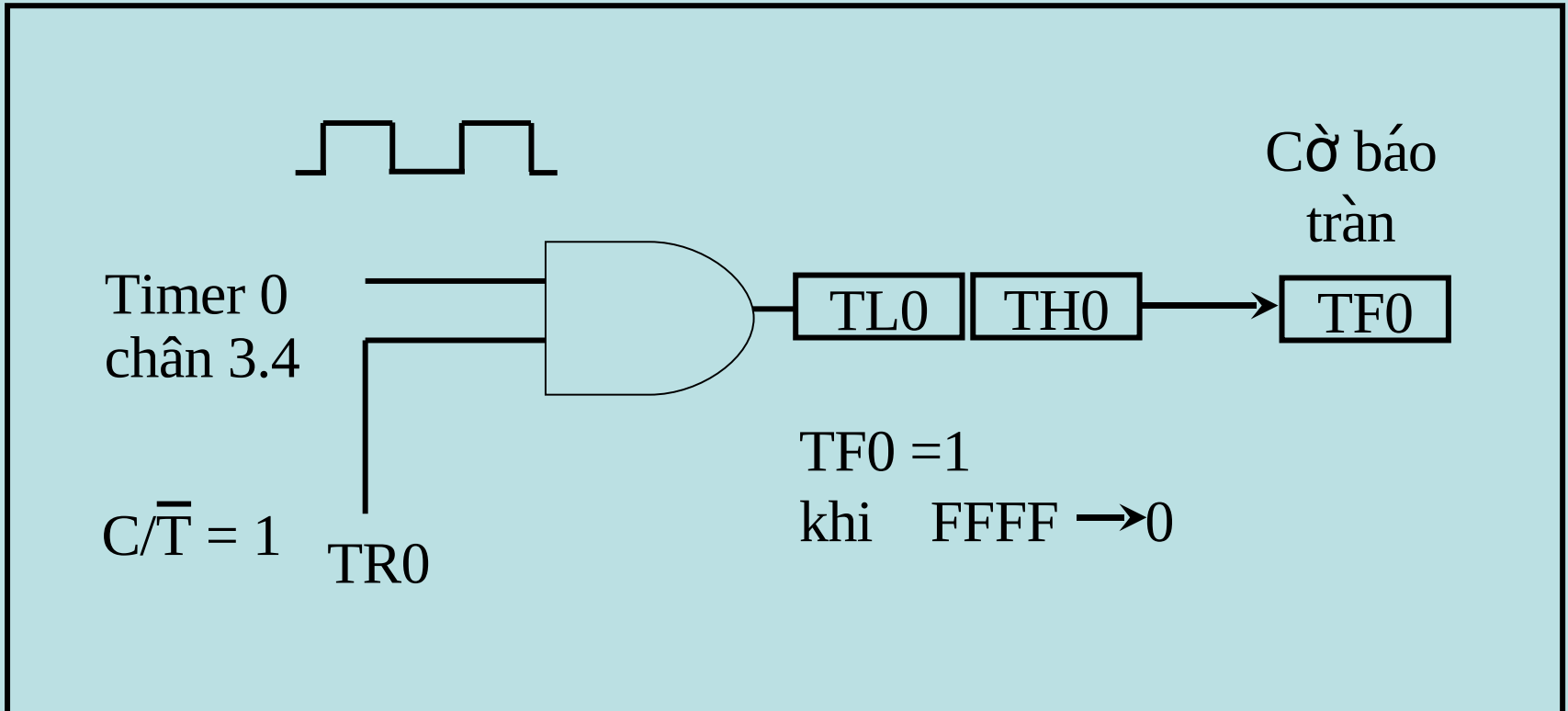
Timer 1

Timer 0

Counter Chế độ 1

- **Bộ đếm 16-bit** (TH0 và TL0)
- TH0-TL0 tăng khi TR0 = 1 **và** có một xung trên ngõ vào (T0).
- Sau khi bộ đếm (TH0-TL0) đạt giá trị cực đại FFFFH, nó sẽ tràn về 0000, và TF0 sẽ lên mức 1.
- Người lập trình có thể giám sát TF0 để ngừng counter 0.
- Người lập trình có thể khởi động giá trị cho TH0-TL0 và khi TF0=1 sẽ chỉ thị một sự kiện đặc biệt. (ví dụ: có 100 người đi qua).

Hoạt động Timer 0 với nguồn xung bên ngoài (Mode 1)



Bộ đếm Chế độ 2

- 8-bit **counter**.
 - Chỉ cho phép nạp giá trị từ 00 tới FFH vào TH0.
- Tự động nạp lại (Auto-reloading)
- TL0 tăng nếu TR0=1 và có xung cấp từ ngoài.
- Xem các ví dụ 8-18, 8-19

Ví dụ 8-18 (1/2)

Giả sử rằng xung clock được cấp tới chân T1. Viết chương trình cho bộ đếm 1 làm việc ở chế độ 2. gửi giá trị đếm được trong TL1 tới P2.

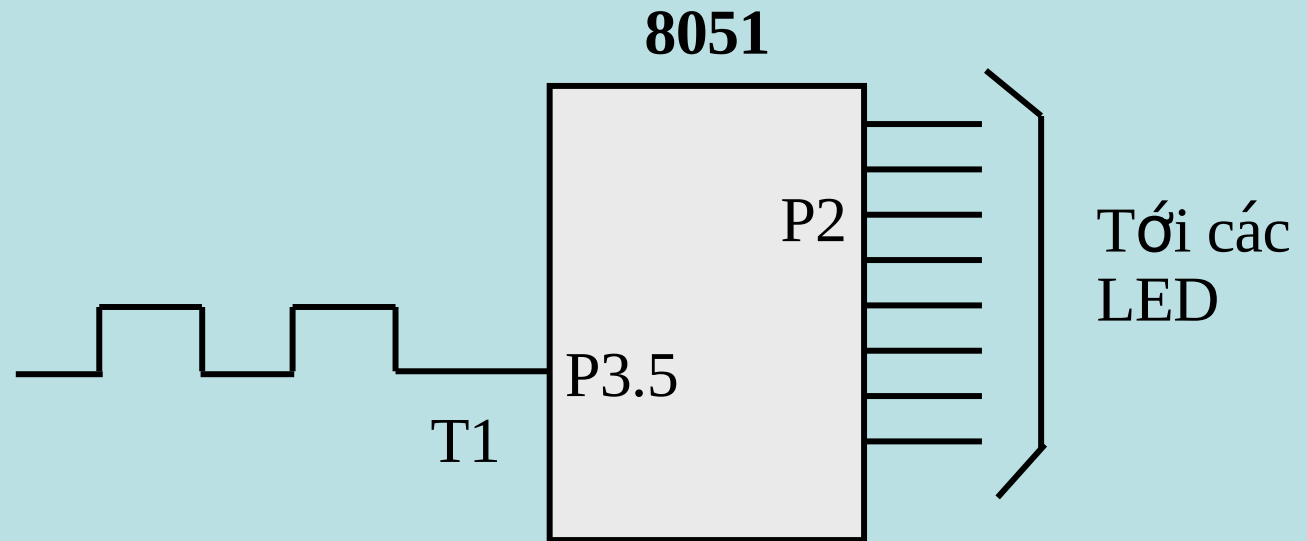
Giải:

```
MOV    TMOD, #01100000B ;mode 2, counter 1
MOV    TH1, #0
SETB  P3.5                ;khởi động T1 là in
AGAIN: SETB TR1            ;start
BACK: MOV  A, TL1
      MOV  P2, A            ;gửi ra P2
      JNB  TF1, Back        ;giám sát tràn
                          ;overflow
      CLR  TR1              ;stop
      CLR  TF1              ; TF=0
      SJMP AGAIN           ;lặp lại
```

Ví dụ 8-18 (2/2)

Sử dụng timer 1 làm bộ đếm số xung đưa cạp chân P3.5. Chú ý vai trò lệnh “**SETB P3.5**”. Vì các cổng được khởi động là out sau khi cấp nguồn cho 8051, nên cần phải khởi động P3.5 thành ngõ vào.

P2 được nối tới 8 LED và cấp xung vào T1.



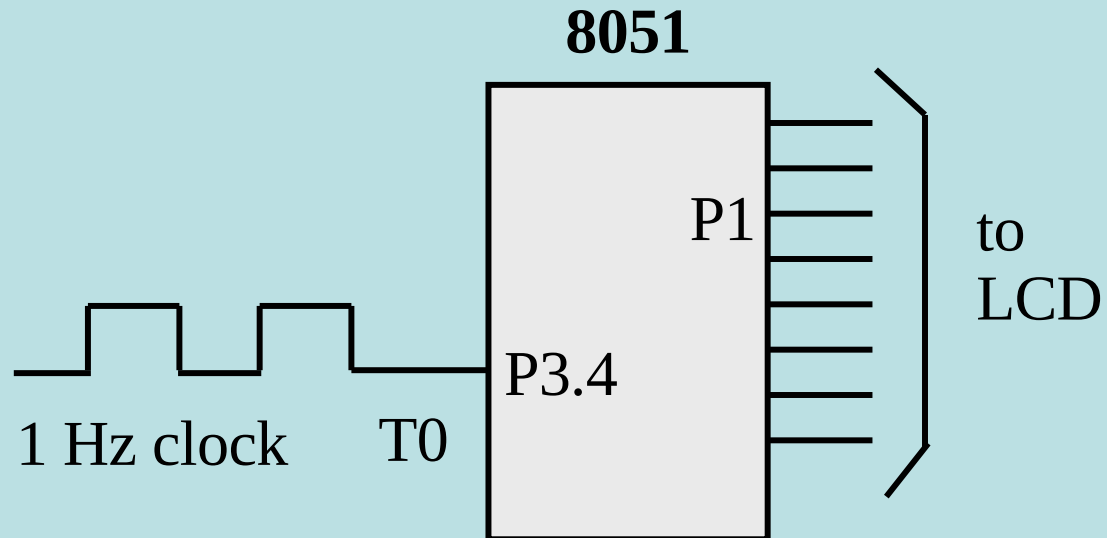
Ví dụ 8-19 (1/3)

Giả sử rằng có xung tần số **1-Hz** cấp tới ngõ P3.4.

Viết chương trình hiển thị giá trị counter 0 lên LCD. Khởi động $TH0 = -60$.

Giải:

Chú ý rằng ở vòng đếm đầu tiên, bộ đếm sẽ bắt đầu từ 0 tới 256 sự kiện, vì sau khi RESET $TL0=0$. Để tránh vắn đề này có thể nạp giá trị -60 cho bộ đếm khi bắt đầu chương trình.



Ví dụ 8-19 (2/3)

```
ACALL LCD_SET_UP      ;giới chương trình
                      ;khởi động LCD

MOV    TMOD,#00000110B ;Counter 0,mode2
MOV    TH0,#-60
SETB   P3.4           ;khởi động T0 in
AGAIN:SETB TR0        ;cho phép chạy
BACK:  MOV    A,TL0    ;chuyển ra A sau 60 xung
ACALL  CONV           ;đổi ra R2,R3,R4
JNB    TF0,BACK       ;lặp khi TF0=0
CLR    TR0            ;dừng đếm
CLR    TF0
SJMP   AGAIN
```

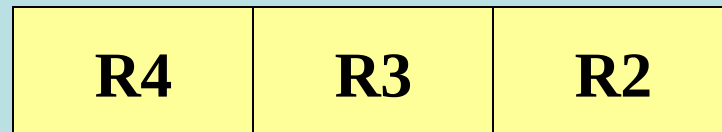
Ví dụ 8-19 (3/3)

;đổi 8-bit nhị phân ra ASCII

```
CONV: MOV    B, #10      ;chia cho 10
      DIV    AB
      MOV    R2, B      ;lưu số dư
      MOV    B, #10      ;chia 10 một lần nữa
      DIV    AB


---


      ORL    A, #30H     ;đổi thành ASCII
      MOV    R4, A
      MOV    A, B
      ORL    A, #30H
      MOV    R3, A
      MOV    A, R2
      ORL    A, #30H
      MOV    R2, A      ;ACALL LCD_DISPLAY
      RET
```



Một đồng hồ số

- **Ví dụ 8-19** sử dụng cho một đồng hồ số.
 - Nếu cung cấp xung vuông có tần số 60Hz cho timer/counter sẽ tạo được các giá trị giây, phút, giờ và hiển thị lên LCD.
- Cần chú ý lệnh “**JNB TF0, target**” để giám sát TF0 cũng làm mất thời gian của vi điều khiển.
 - Để giải quyết cần sử dụng ngắt. Xem phần sau.
 - Khi sử dụng ngắt 8051 có thể thực hiện thêm các công việc khác.
 - Khi TF =1 chương trình ngắt sẽ được thực hiện.

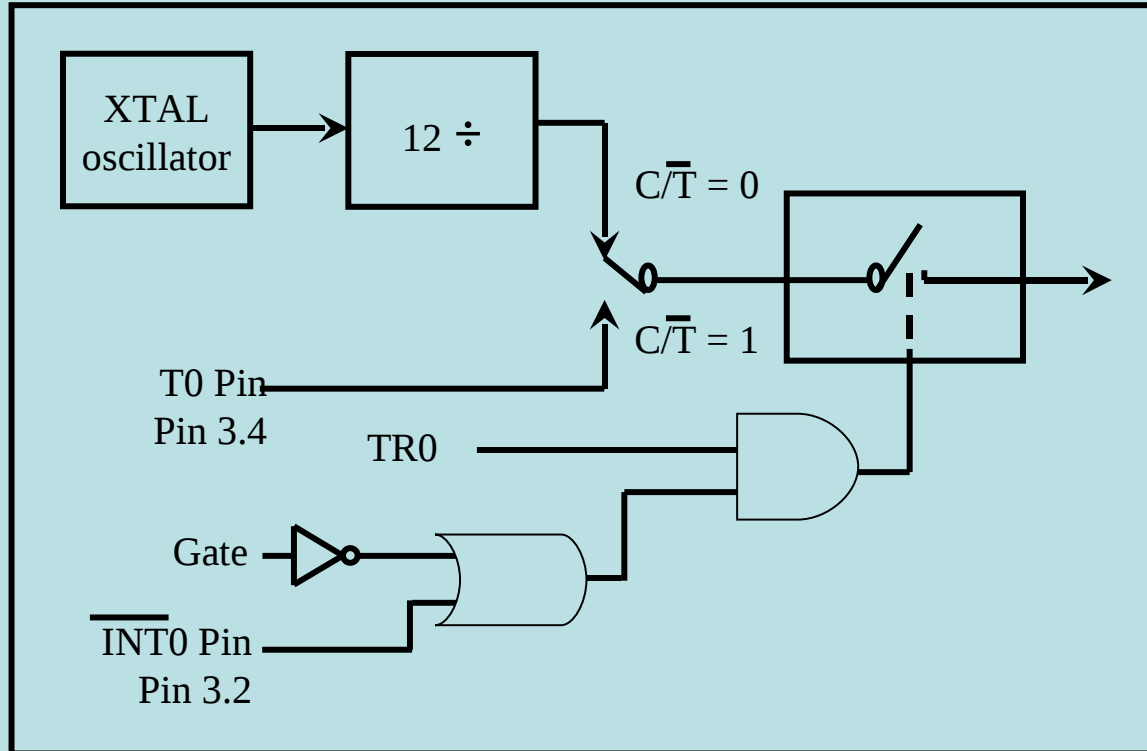
Khi GATE=1

- Trong tất cả phần trên đều cho GATE=0.
 - Timer bắt đầu đếm bằng lệnh “**SETB TR0**” và “**SETB TR1**” tương ứng cho timers 0 và 1.
- Khi GATE=1, chúng ta sẽ sử dụng phần cứng để điều khiển chạy và ngừng timer.
 - INT0 (P3.2, chân 12) starts và stops timer 0
 - INT1 (P3.3, chân 13) starts và stops timer 1
 - Khi đó có thể cho phép chạy hoặc dừng timer bằng một công tắc bên ngoài.

Ví dụ cho **GATE=1**

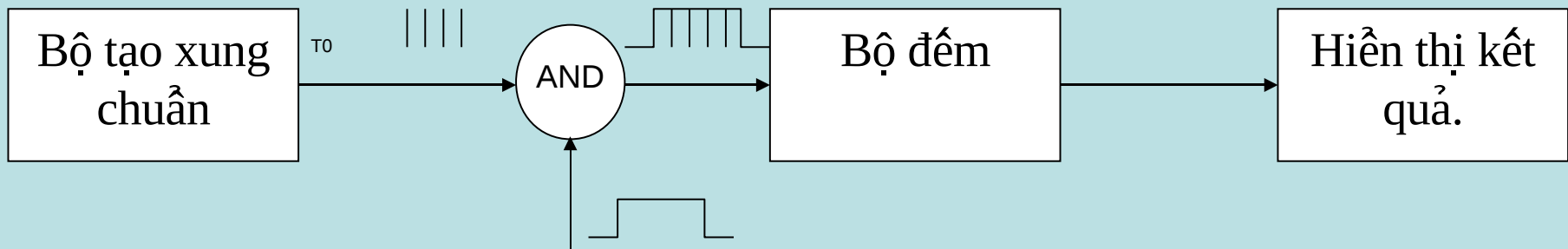
- 8051 sử dụng tạo ra âm thanh cảnh báo sau mỗi giây bằng timer 0.
- Timer 0 bắt đầu chạy bằng lệnh “**SETB TR0**” và dưới sự điều khiển của người sử dụng hệ thống.
- Một công tắc nối vào chân P3.2 có thể sử dụng để bật hoặc tắt tín hiệu cảnh báo.

Timer/Counter 0



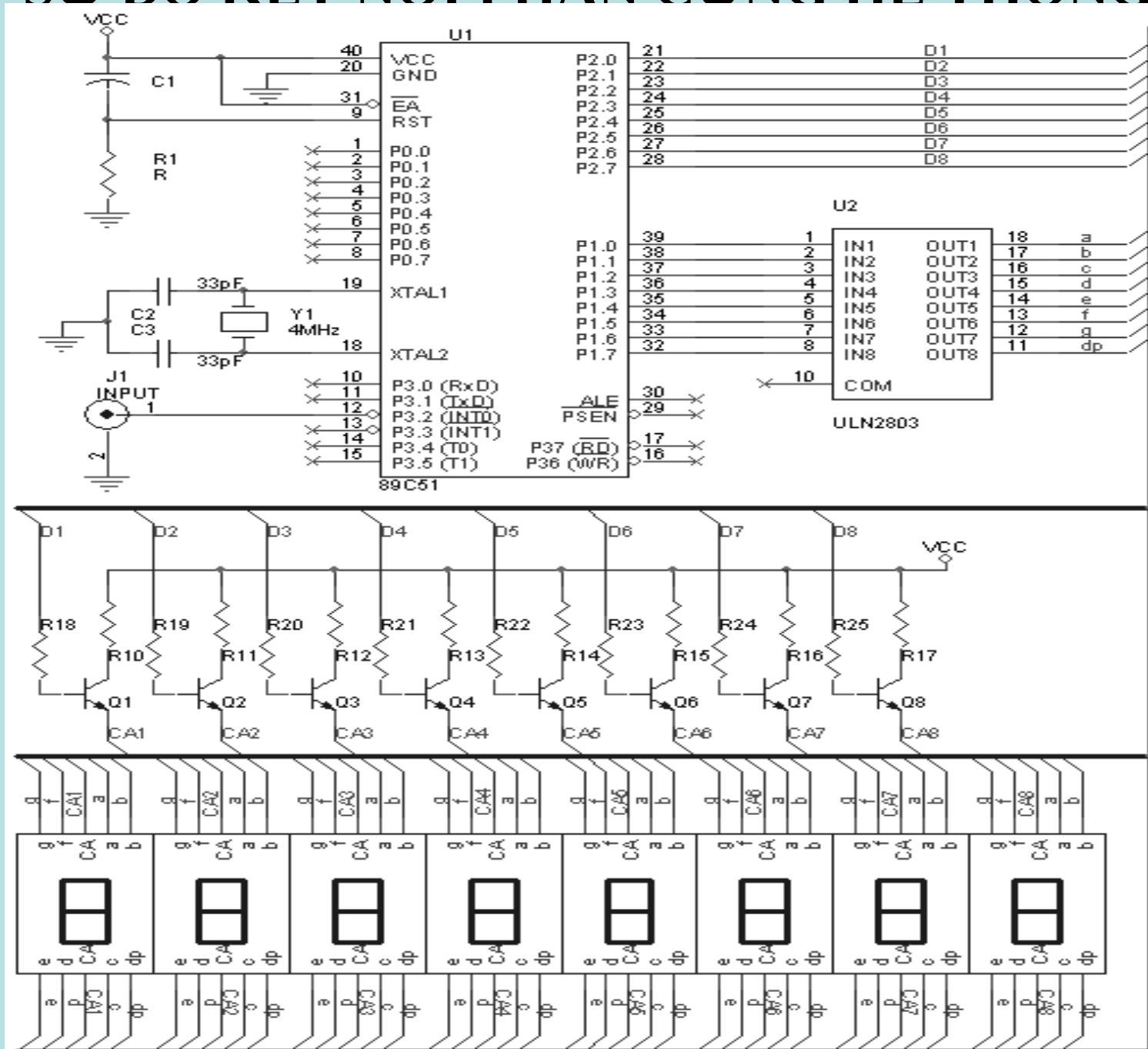
PROJECT 1

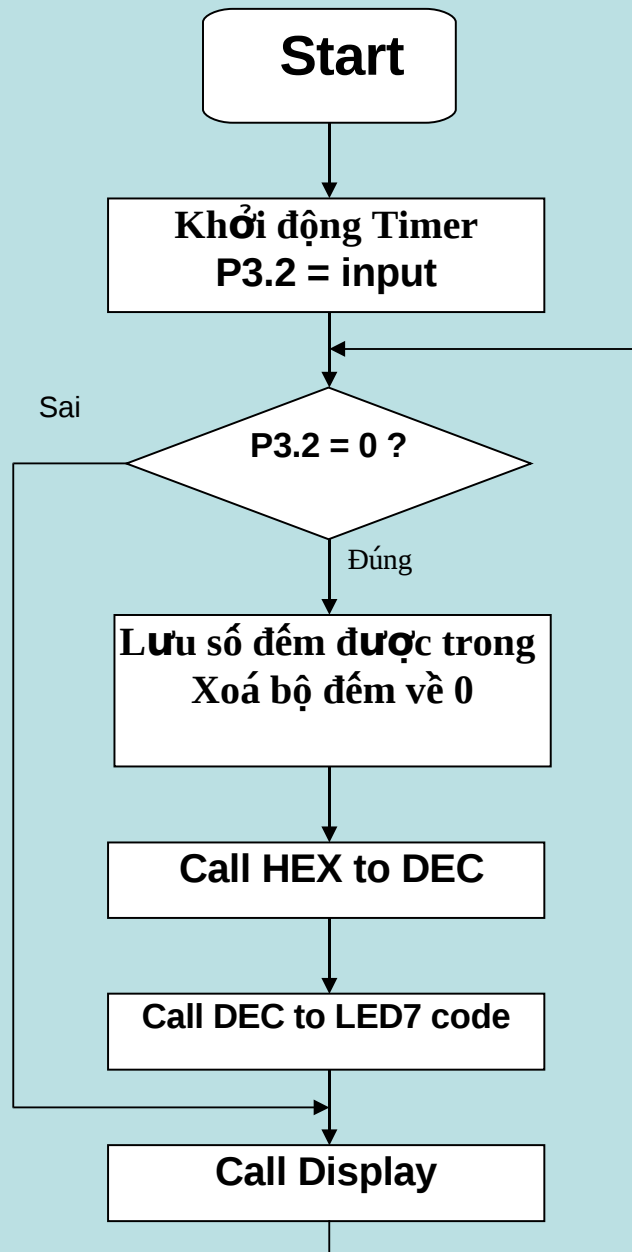
- Thực hiện hệ thống vi điều khiển đo độ rộng xung.



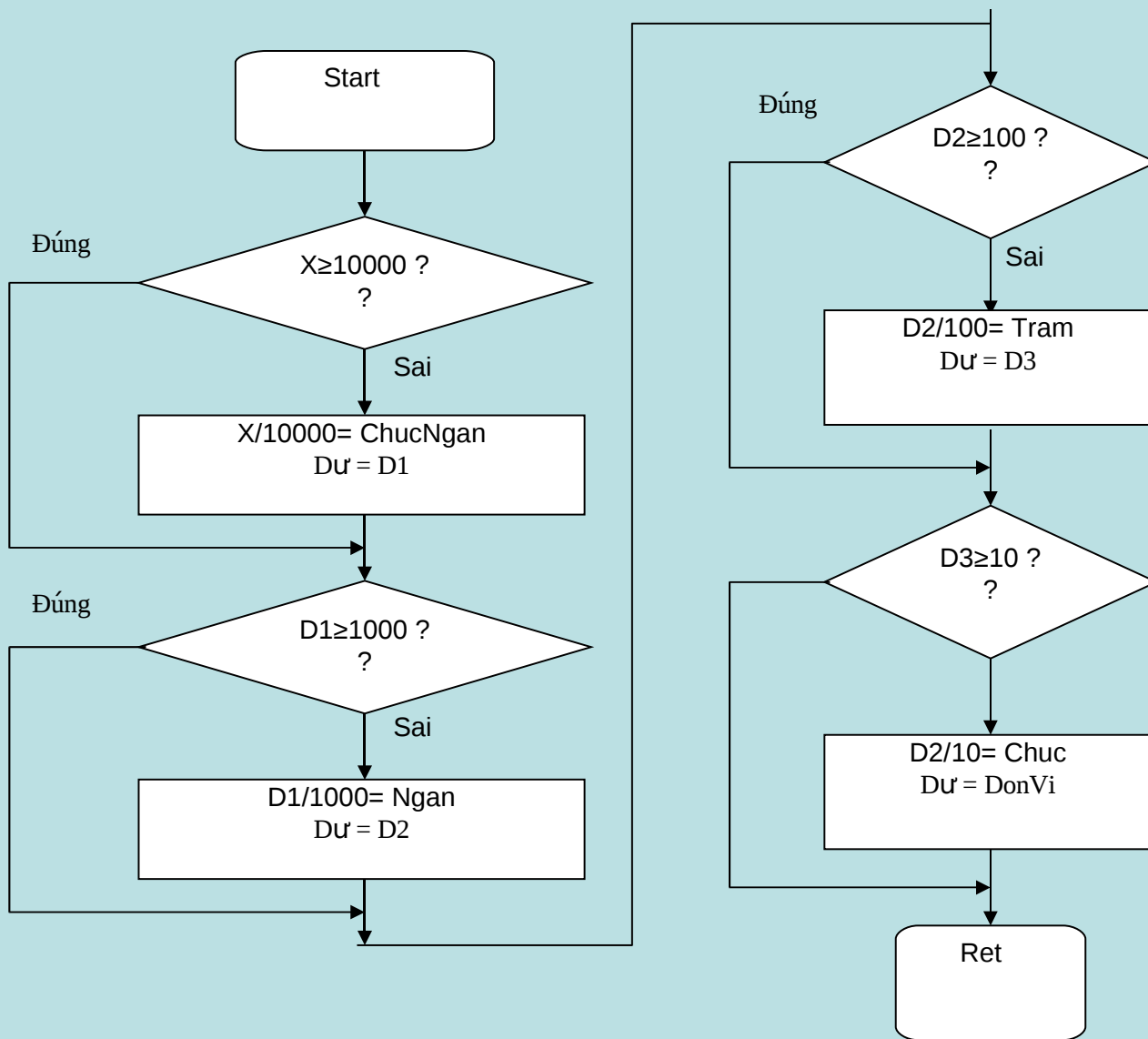
Nguyên tắc đo độ rộng xung.

SƠ ĐỒ KẾT NỐI PHẦN CỨNG HỆ THỐNG





Giải thuật chương trình chính hệ thống đo độ rộng xung.



Hình 7.4. Giải thuật chương trình đổi HEX ra DEC.

Lập trình ngắt

- Một *ngắt* là một sự kiện của một thiết bị từ ngoài hoặc bên trong ngắt vi điều khiển, thông báo nó cần vi điều khiển phục vụ.

Ngắt (Interrupt) và Quét vòng (Polling)

- Một bộ vi điều khiển có thể điều khiển một số thiết bị. Có hai phương pháp để điều khiển các thiết bị này: interrupts hoặc polling.
- Chương trình điều khiển thiết bị theo cơ chế ngắt được gọi là **Chương trình con phục vụ ngắt** (*interrupt service routine* - ISR) hoặc chương trình xử lý ngắt (*interrupt handler*).

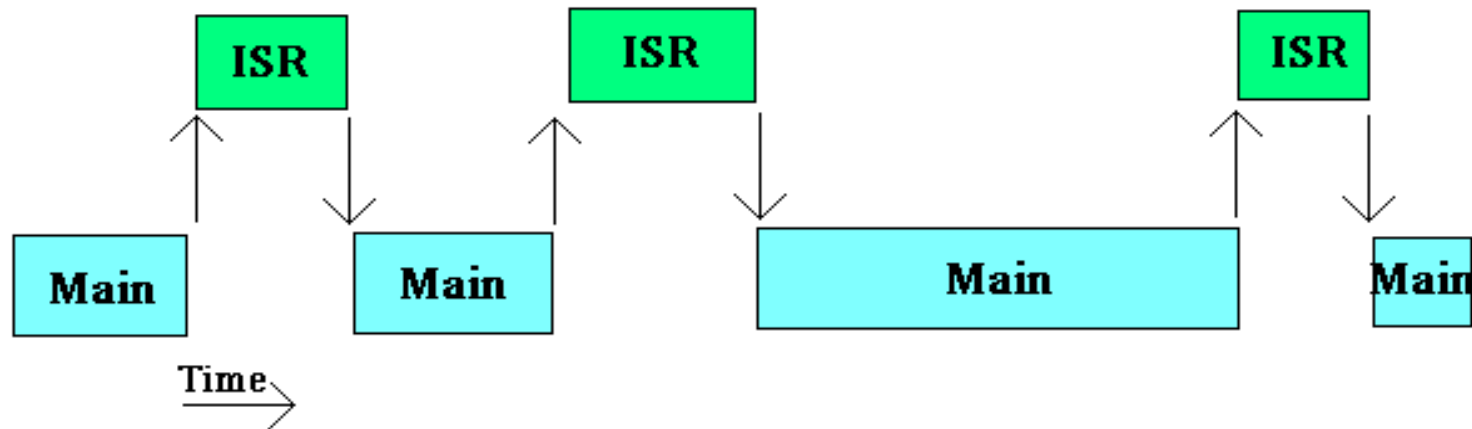
Các bước thực hiện một ngắt

- Kết thúc lệnh đang thực hiện và cất địa chỉ của lệnh kế tiếp (PC) vào ngăn xếp.
- Cất các trạng thái hiện tại của các ngắt vào thanh ghi bên trong (không cất vào ngăn xếp)
- Nhảy tới một vị trí định trước trong bộ nhớ được gọi là bảng vector ngắt, nơi giữ địa chỉ chương trình con phục vụ ngắt.
- Bộ vi điều khiển sẽ lấy địa chỉ chương trình con phục vụ ngắt và chuyển điều khiển tới chương trình này. ISR sẽ bắt đầu được thực hiện cho đến khi gặp lệnh cuối cùng của ISR là RETI (return from interrupt)
- Sau khi thực hiện lệnh RETI, bộ vi điều khiển sẽ quay về vị trí đã bị ngắt. Trước tiên nó lấy giá trị đỉnh ngăn xếp nạp trở lại thanh ghi PC. Sau đó nó bắt đầu thực hiện chương trình tại địa chỉ này.

Program execution without intrrupts :



Program execution with intrrupts :



ISR : Intrrupt Service Routin

Sáu nguồn ngắt của 8051

```
ORG 0
SJMP FIRST

ORG 13H      ;ISR cho INT1
MOV  A,P1    ;đọc dữ liệu
ACALL MUL39  ;R3R2R1=A*39
MOV  A,R3
FCAL PUTH
MOV  A,R2
FCAL PUTH
MOV  A,R1
FCAL PUTH    ;hiển thị điện áp theo mV
SETB P3.0    ;RD=1 cho cky kế tiếp
CLR  P3.1    ;WR=0
SETB P3.1    ;WR=1 ,bắt đầu ADC
RETI
ORG 30H
FIRST:
...
END
```

Six interrupts in 8051

<i>Interrupt</i>	<i>ROM location</i>	<i>Pin</i>
Reset	0000H	9
Timer 0	000BH	
Timer 1	001BH	
INT 0 (EX1)	0003H	P3.2
INT 1 (EX2)	0013H	P3.3
Serial communication	0023H	

Cho phép và cấm một ngắt:

D7				D0			
EA	--	ET2	ES	ET1	EX1	ET0	EX0
EA	IE.7	Disables all interrupts.					
--	IE.6	No implemented, reserved for future use					
ET2	IE.5	Enables or disables timer 2 overflow interrupt					
ES	IE.4	Enables or disables the serial port interrupt					
ET1	IE.3	Enables or disables timer 2 overflow interrupt					
EX1	IE.2	Enables or disables external interrupt 1					
ET0	IE.1	Enables or disables timer 0 overflow interrupt					
EX0	IE.0	Enables or disables external interrupt					

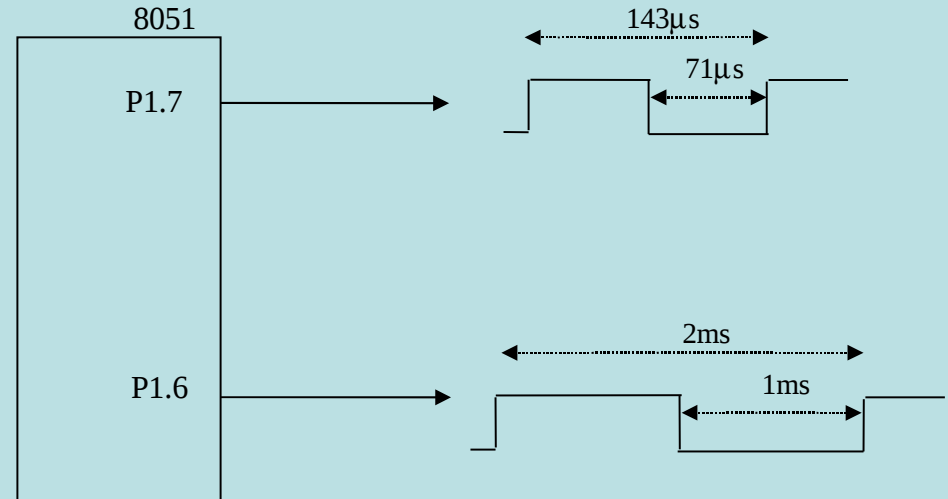
Bit	Ký hiệu	Địa chỉ	Chức năng (1: cho phép; 0: cấm)
IE7	EA	AFH	Cho phép/cấm toàn bộ các ngắt.
IE6	-	AEH	Không định nghĩa.
IE5	-	ADH	Dự phòng cho thế hệ kế tiếp.
IE4	ES	ACH	Cho phép ngắt cổng nối tiếp.
IE3	ET1	ABH	Cho phép ngắt timer 1.
IE2	EX1	AAH	Cho phép ngắt ngoài cấp từ INT1.
IE1	ET0	A9H	Cho phép ngắt timer 0
IE0	EX0	A8H	Cho phép ngắt ngoài INT0

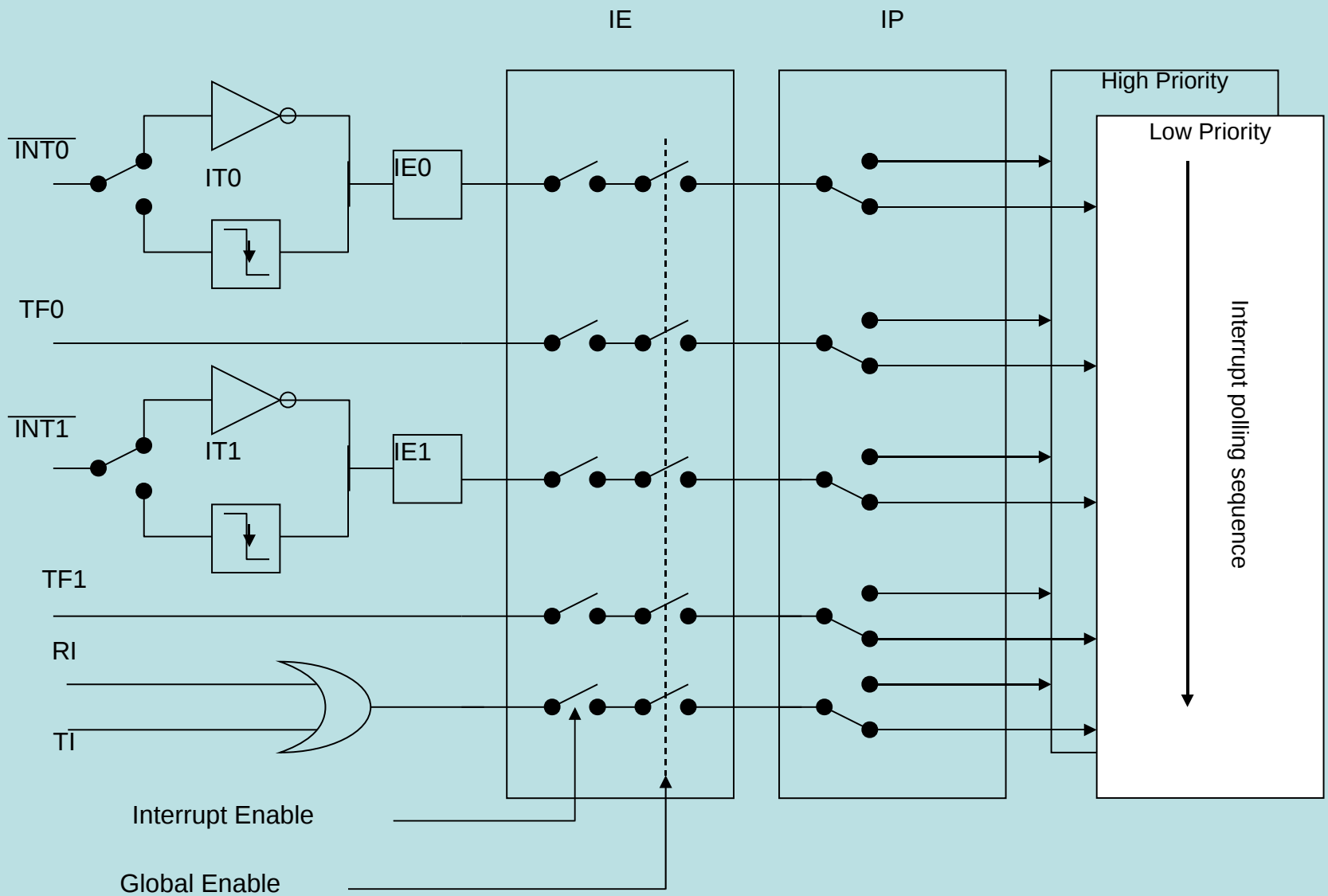
Ví dụ:

Viết chương trình sử dụng ngắt tạo ra đồng thời 2 xung vuông 7 kHz và 500 Hz trên P1.7 và P1.6.

Giải:

```
ORG 0
LJMP MAIN
ORG 000BH
LJMP T0ISR
ORG 001BH
LJMP T1ISR
ORG 0030H
MAIN: MOV TMOD,#12H
      MOV TH0,#-71
      SETB TR0
      SETB TR1
      MOV IE,#8AH
      MOV IE,#8AH
      SJMP $
T0ISR: CPL P1.7
      RETI
T1ISR: CLR TF1
      CLR TR1
      MOV TH1,#HIGH(-1000)
      MOV TL1,#LOW(-1000)
      SETB TR1
      CPL P1.6
      RETI
END
```





Cấu trúc cơ chế ngắt của 8051.

Thanh ghi Ưu tiên ngắt IP.

Bit	Ký hiệu	Địa chỉ	Chức năng (1: mức Ưu tiên cao; 0: mức Ưu tiên thấp)
IP7	-	-	Không định nghĩa.
IP6	-	-	Không định nghĩa.
IP5	-	BDH	Dự phòng.
IP4	PS	BCH	Ưu tiên cho ngắt cổng nối tiếp.
IP3	PT1	BBH	Ưu tiên cho ngắt timer 1.
IP2	PX1	BAH	Ưu tiên cho ngắt ngoài INT1.
IP1	PT0	B9H	Ưu tiên cho ngắt timer 0.
IP0	PX0	B8H	Ưu tiên cho ngắt ngoài INT0.

Các cờ tác động ngắt

Ngắt	CỜ	Địa chỉ vector ngắt
Reset	RST	0000H
INT0	IE0	0003H
Timer 0	TF0	000BH
INT1	IE1	0013H
Timer 1	TF1	001BH
Serial port	RI hoặc TI	0023H

Cấu trúc chương trình sử dụng ngắt

```
ORG    0
JMP    main
ORG    03h
JMP    ext_int0
ORG    0Bh
JMP    timer_int0
ORG    13h
JMP    ext_int1
ORG    1Bh
JMP    timer_int1
ORG    23h
JMP    serial_int
ORG    2Bh
JMP    timer_int2
ORG    030h

main:  . . .
```

Bài tập

- Giải sử một động cơ điện gởi tới hệ thống vi điều khiển số xung tương ứng với số vòng quay / phút của nó (r.p.m). Viết một chương trình giám sát:
 - Nếu số xung trong 10 giây nhỏ hơn 100 lập P0.1 lên 1. Nếu số xung nhiều hơn 200 lập P1.1 lên 1.
- Thiết kế phần cứng kết nối bàn phím ma trận 4x4 tới vi điều khiển. Viết chương trình xác định phím nhấn trên bàn phím.

CỔNG NỘI TIẾP 8051

Basics of serial communication

- *Baud Rate*

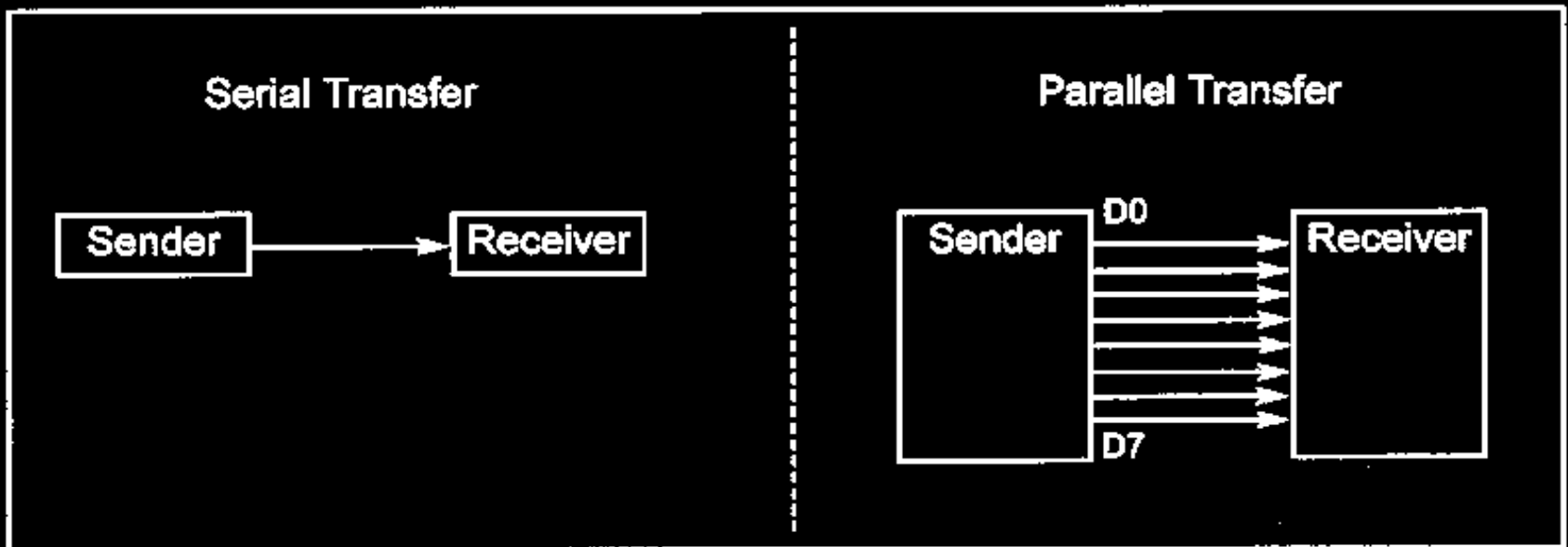


Figure 10-1. Serial versus Parallel Data Transfer

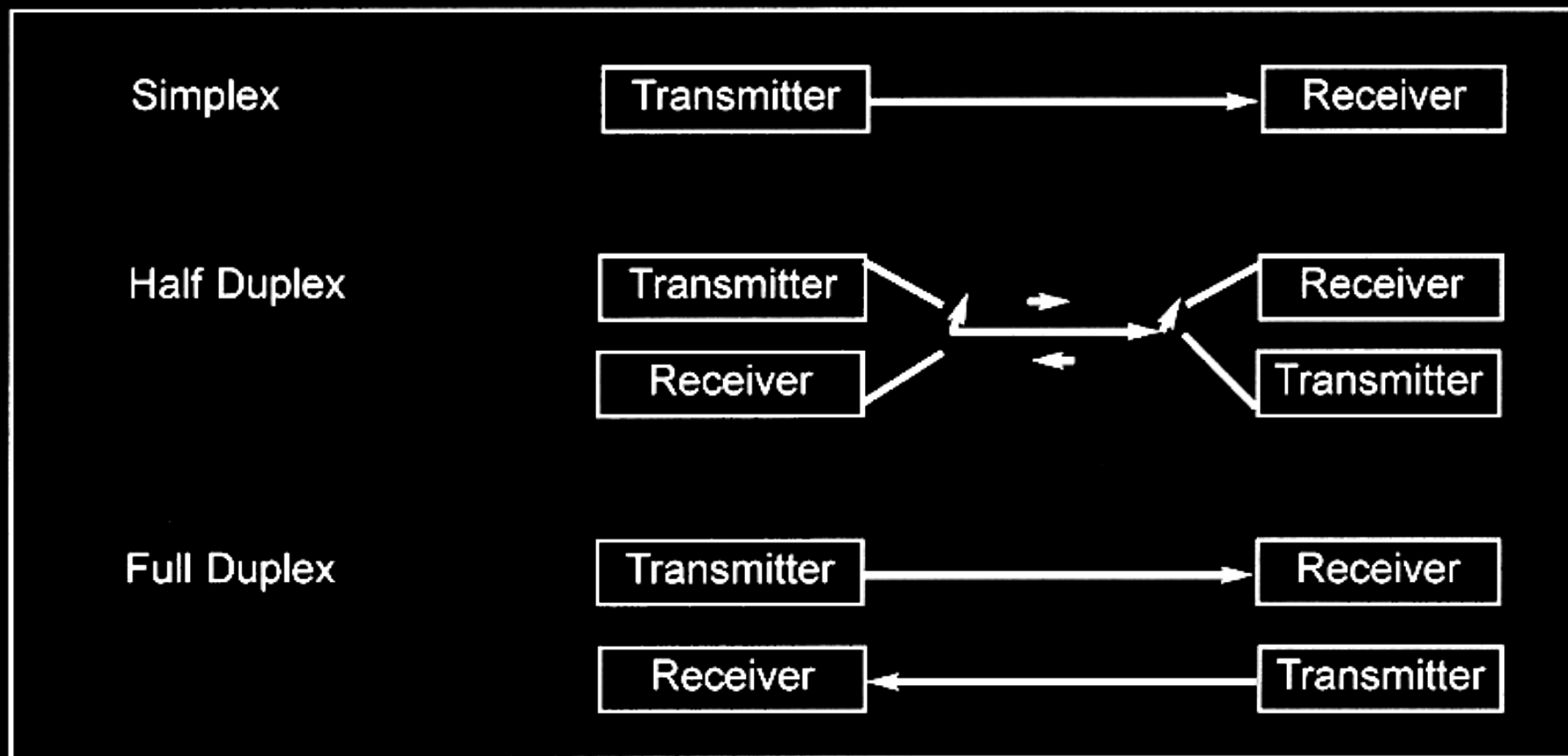


Figure 10-2. Simplex, Half-, and Full-Duplex Transfers

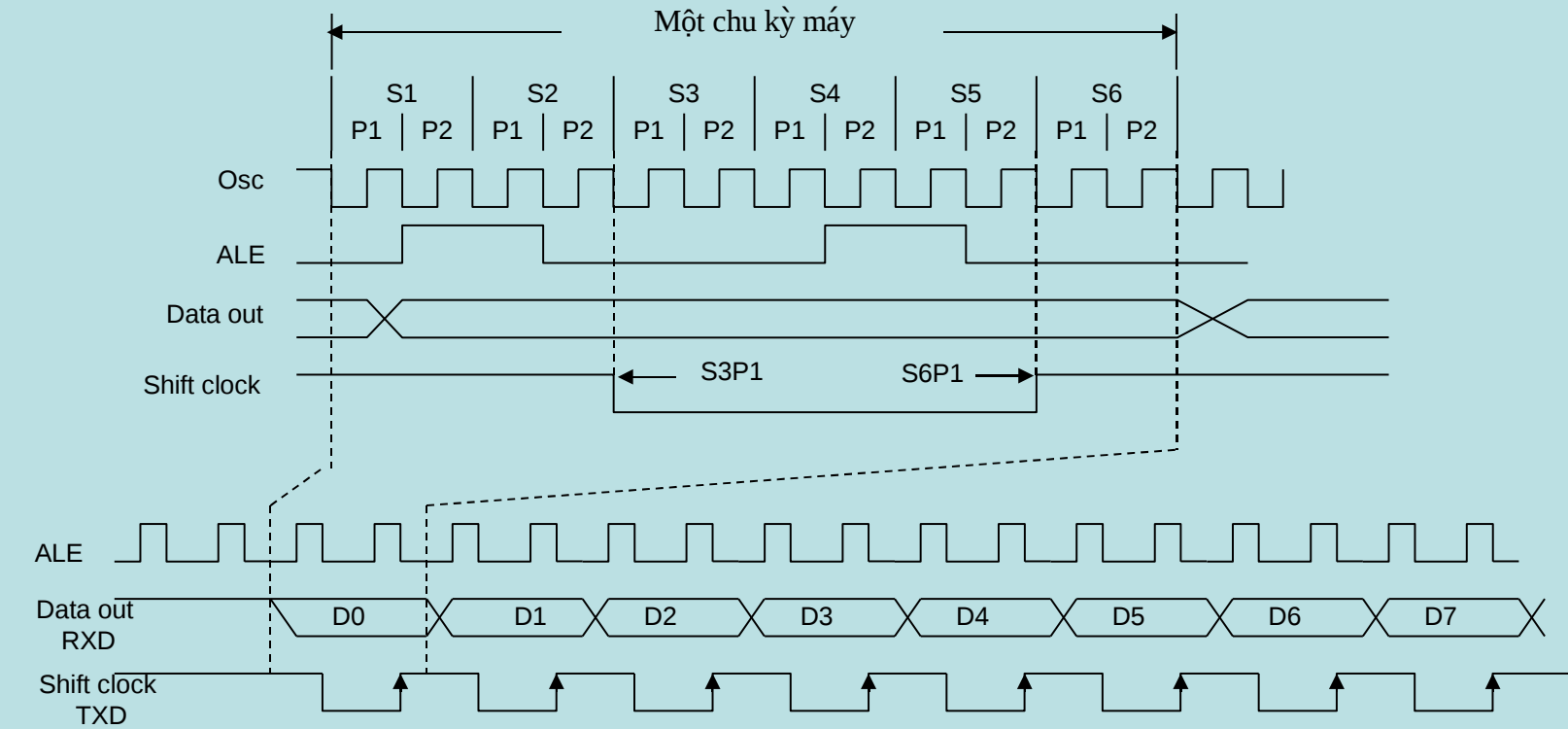
Thanh ghi SCON

SCON.7	SM0	9FH	Bit 0 của chế độ port nối tiếp
SCON.6	SM1	9EH	Bit 1 của chế độ port nối tiếp
SCON.5	SM2	9DH	Cho phép chế độ nhiều vi xử lý.
SCON.4	REN	9CH	Cho phép nhận. Lập/xóa mềm.
SCON.3	TB8	9BH	Bit thứ 9 trong mode 2 và 3. Lập/xóa mềm.
SCON.2	RB8	9AH	Bit thứ 9 nhận vào (tùy mode).
SCON.1	TI	99H	Cờ ngắt truyền.
SCON.0	RI	98H	Cờ ngắt nhận.

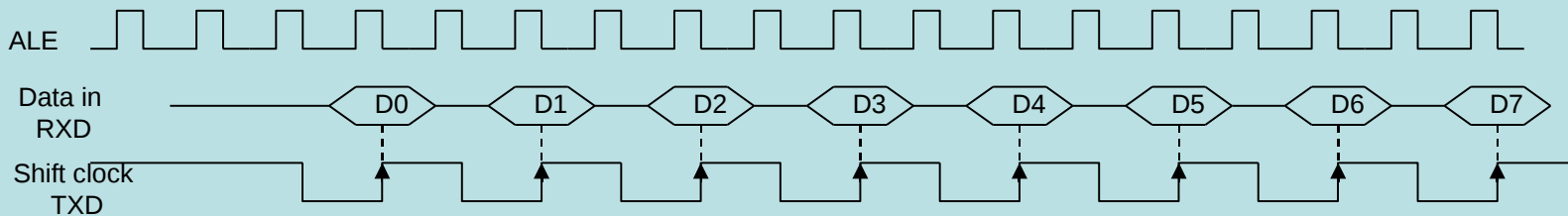
Các chế độ hoạt động của cổng nối tiếp

SM0	SM1	Mode	Mô tả	Tốc độ
0	0	0	Thanh ghi dịch	$f_{\text{ocs}} / 12$
0	1	1	UART 8 bit	Thay đổi
1	0	2	UART 9 bit	$f_{\text{osc}} / 64$ hoặc $/ 32$
1	1	3	UART 9 bit	Thay đổi

Chế độ 0 của cổng nối tiếp

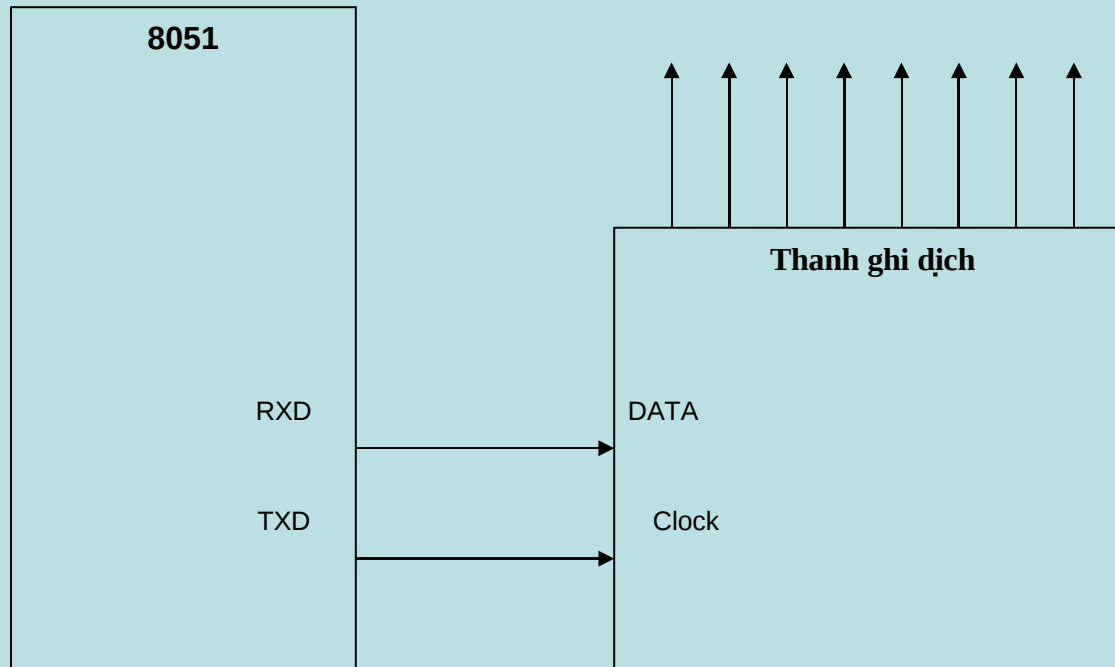


a) Định thời phát nối tiếp trong chế độ 0



b) Định thời nhận nối tiếp trong chế độ 0

Hình 8.13: Định thời thu phát dữ liệu trên cổng nối tiếp ở chế độ 0



Tăng thêm cổng vào ra song song sử dụng cổng nối tiếp ở chế độ 0

Khung truyền nối tiếp cận đồng bộ

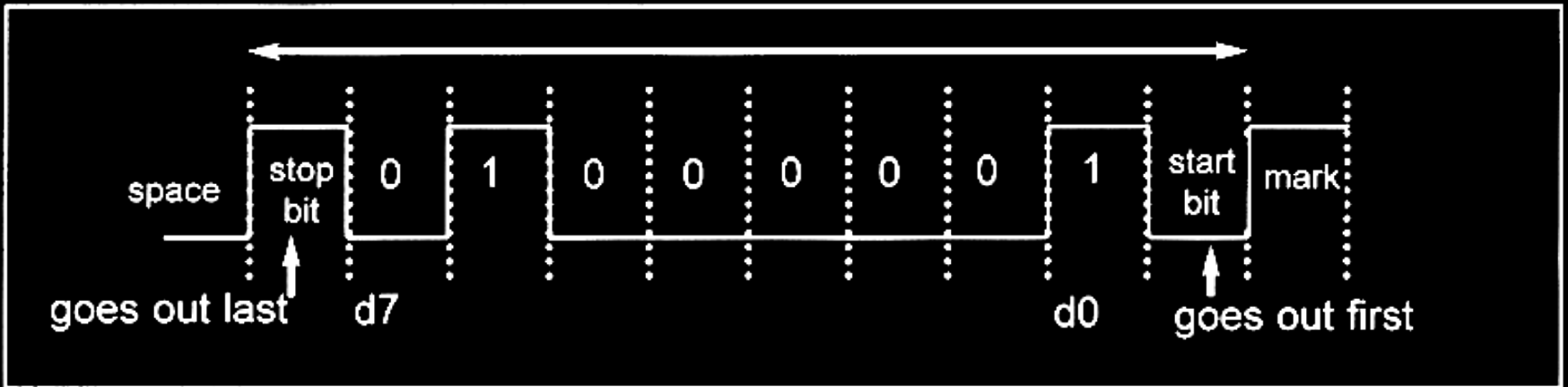


Figure 10-3. Framing ASCII "A" (41H)

Các chân RxD và TxD của 8051

- TxD chân 11 của 8051 (P3.1)
- RxD chân 10 của 8051 (P3.0)

Thanh ghi SBUF

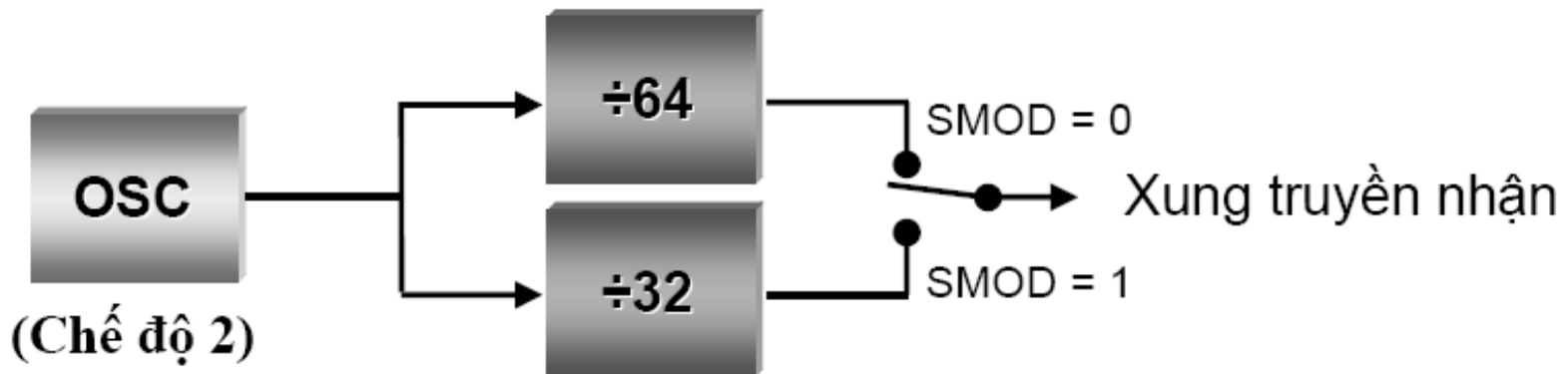
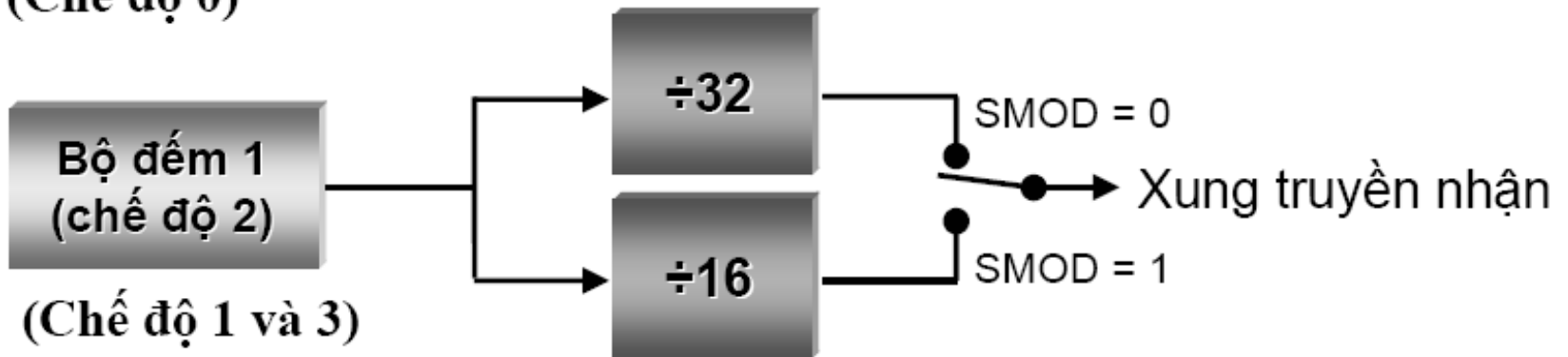
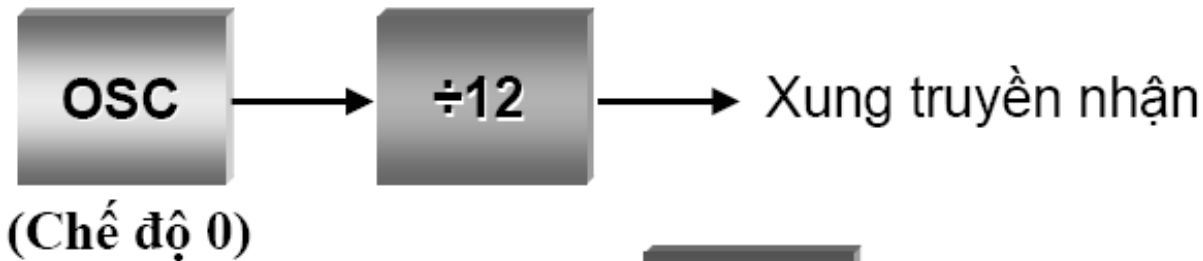
```
MOV  SBUF,#'D'  ; nạp SBUF=44H, ASCII của 'D'  
MOV  SBUF,A     ; chuyển dữ liệu từ thanh chứa tới SBUF  
MOV  A,SBUF     ; chép giá trị SBUF vào thanh chứa
```

Truyền thông tin nối tiếp hệ thống đa xử lý



- Truyền 9 bit.
- Bit thứ 9 vào RB8.
- Ngắt quãng chỉ xảy ra khi RB8 = 1 (cho bit SM2 = 1-).
- Thông tin chia ra byte địa chỉ có RB8 = 1 và byte dữ liệu có RB8 = 0.
- Master gọi byte địa chỉ trước, slave nào đúng địa chỉ thì xóa bit SM2 để nhận dữ liệu.

Chọn tốc độ truyền cho cổng nối tiếp



Công thức tính tốc độ cổng nối tiếp

$$\text{tốc độ Baud} = \frac{\text{tần số dao động}}{12} \times \frac{2^{\text{SMOD}}}{32 \times (256 - \text{TH1})}$$

Công thức tính số đếm bộ đếm 1

$$\text{TH1} = 256 - \frac{2^{\text{SMOD}}}{32 \times \text{tốc độ Baud}} \times \frac{\text{tần số dao động}}{12}$$

Bảng chọn tốc độ truyền công nối tiếp

tốc độ Baud	tần số thạch anh	SMOD	TH1	tốc độ thực tế	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

MAX232

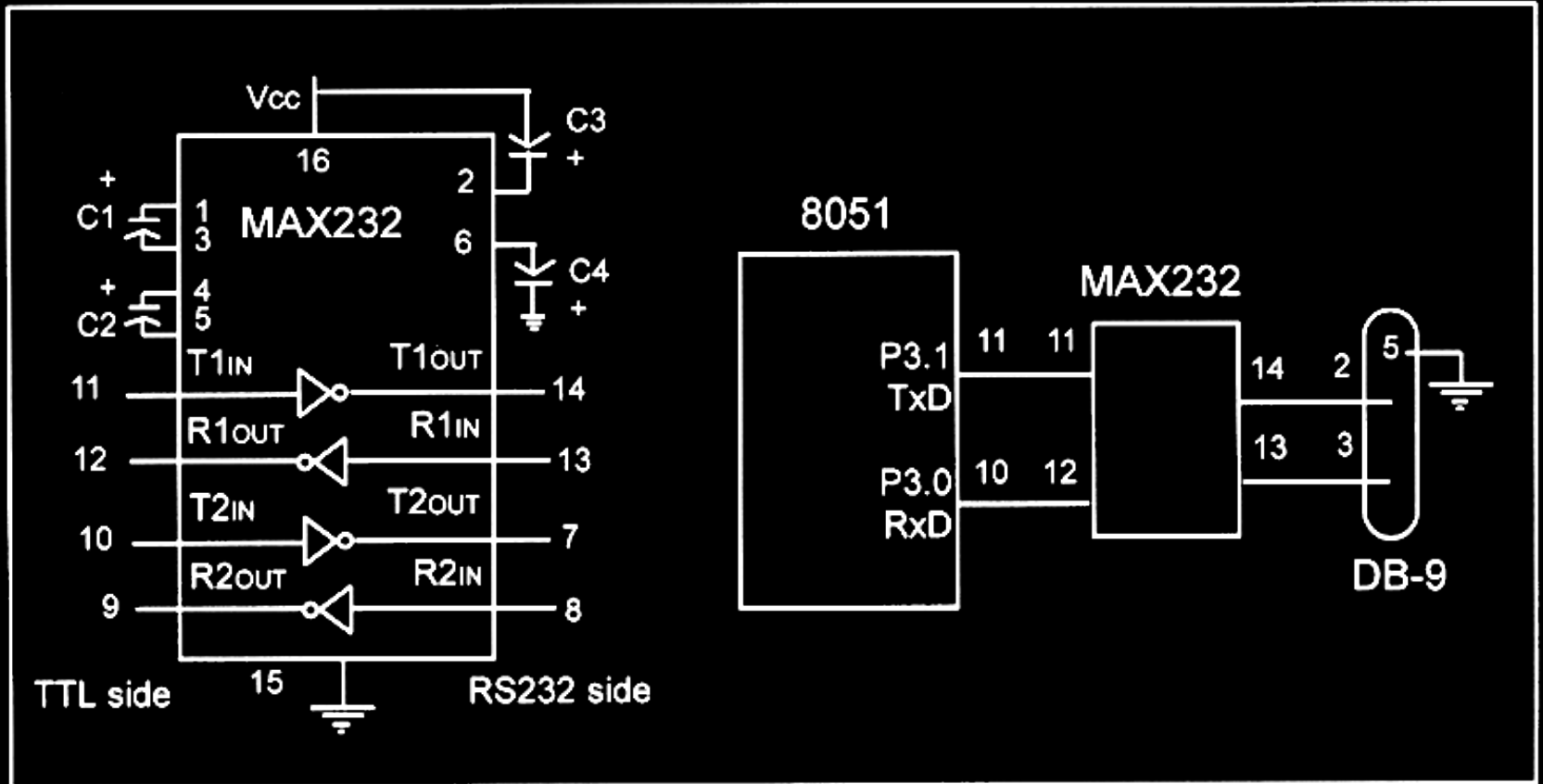
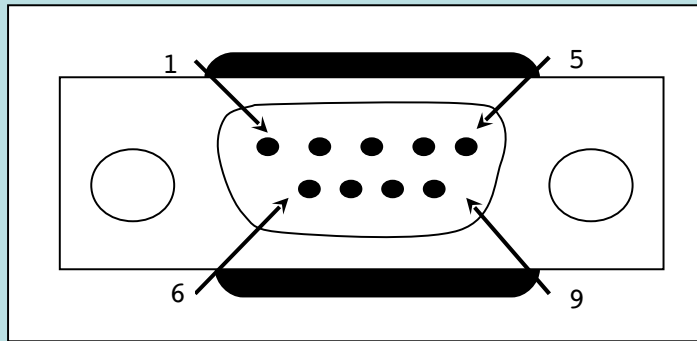


Figure 10.7: (a) Inside MAX232 and (b) its Connection to the 8051 (Null Modem)



Example 10-1

With XTAL = 11.0592 MHz, find the TH1 value needed to have the following baud rates. (a) 9600 (b) 2400 (c) 1200

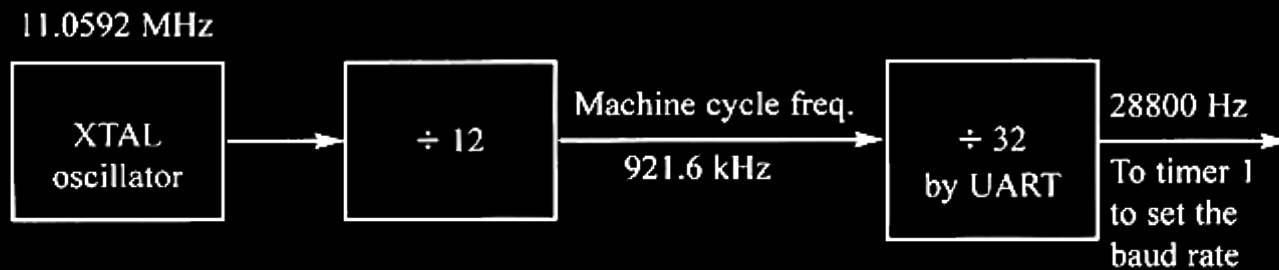
Solution:

With XTAL = 11.0592 MHz, we have:

The machine cycle frequency of the 8051 = $11.0592 \text{ MHz} / 12 = 921.6 \text{ kHz}$, and $921.6 \text{ kHz} / 32 = 28,800 \text{ Hz}$ is the frequency provided by UART to timer 1 to set baud rate.

- (a) $28,800 / 3 = 9600$ where $-3 = \text{FD}$ (hex) is loaded into TH1
(b) $28,800 / 12 = 2400$ where $-12 = \text{F4}$ (hex) is loaded into TH1
(c) $28,800 / 24 = 1200$ where $-24 = \text{E8}$ (hex) is loaded into TH1

Notice that dividing 1/12th of the crystal frequency by 32 is the default value upon activation of the 8051 RESET pin. We can change this default setting. This is explained at the end of this chapter.



Thực hiện đồng hồ điện tử trên 8051

- Timer 0 chế độ 2 đếm 250 xung (1uSec): TMODE= #02; TH0=-250
- Cho phép ngắt Timer 0: IE=10000010B.
- Cấu trúc chương trình trang 78/100 slide 2
- Chương trình ngắt T0:
 - Tăng R0, R0 = 4 xoá R0 tăng R1,
 - R1=100 xoá R1 tăng R2 (0,1 sec),
 - R2=10 xoá R2 tăng R3,
 - R3=10 xoá R3 tăng R4. R4=6 xoá R4 tăng R5,
 - R5 =10 xoá R5 tăng R6. R6 = 6 xoá R6 tăng R7.
 - R7= 10 xoá R7 tăng (08H).
 - (08H) = 2, R7 = 4 xoá R7 và (08H)
 - RETI

Thực hiện đồng hồ điện tử trên 8051

- Chương trình chính:
 - Khởi động Timer, khởi động các ô nhớ các R0 – R7, 08H
 - Đổi BCD ra LED7: 06 giá trị ghi vào các ô nhớ 30H-35H. (sử dụng lệnh `MOVC A,@A+DPTR`).
 - Hiện lên LED 7 đoạn.
 - Chỉnh giờ: Nhận phím.