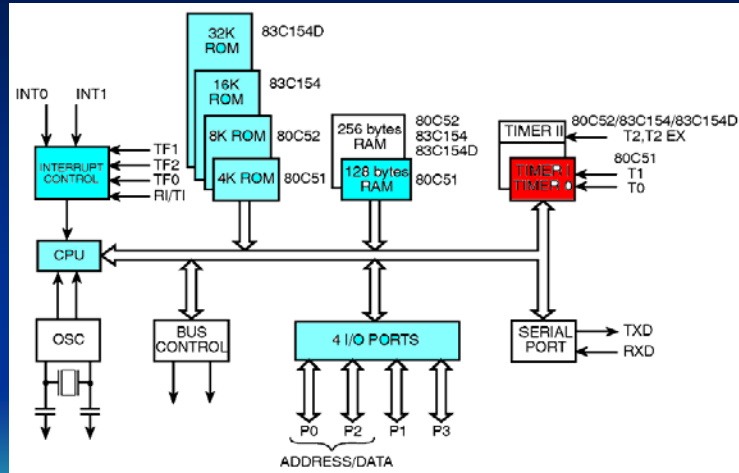


## 12. วงจรนับ/จับเวลา



วงจรถ่ายทอดและวงจรมินิ

## 12. วงจรนับ/จับเวลา

8051 ประกอบด้วยรีจิสเตอร์ขนาด 16 บิต จำนวน 2 ตัว คือ T0 (Timer0) และ T1 (Timer1) ซึ่งสามารถนำไปใช้งานได้สองวิธี โดยสามารถควบคุมให้ทำหน้าที่เป็น **ตัวจับเวลา(Timer)** เพื่อบันทึกจำนวนพัลส์สัญญาณนาฬิกาภายใน หรือควบคุมให้ทำหน้าที่เป็น **ตัวนับ(Counter)** เพื่อบันทึกจำนวนพัลส์ของระบบได้ ภายในรีจิสเตอร์แต่ละตัวยังสามารถแยกออกได้เป็นรีจิสเตอร์ขนาด 8 บิต คือ TH0, TL0, TH1 และ TL1 โดยการทำงานของรีจิสเตอร์ทั้ง 2 ตัวนี้มีผลมาจากค่าที่กำหนดค่าของ bit ที่อยู่ใน **TMOD (Timer mode control register)** และ **TCON (Timer/Counter control register)**

วงจรถ่ายทอดและวงจรมินิ

## การทำงานของ Timer และ Counter

### การทำงานเป็น Timer

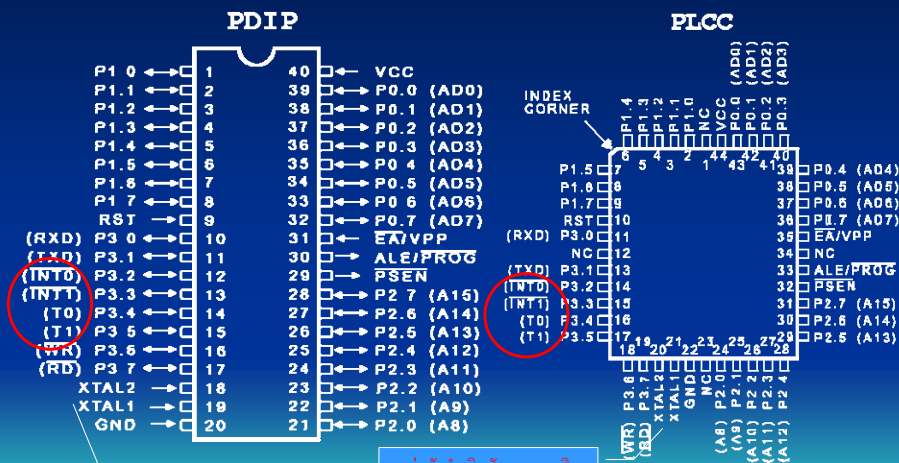
รีจิสเตอร์จะเพิ่มค่าทุกครั้งในทุกๆเมกซ์ซินไซเคิล (12 คาบของสัญญาณนาฬิกา) เมื่อเพิ่มไปจนถึงค่าสูงสุดของรีจิสเตอร์แล้วก็จะกลับไปเป็นค่า 0 เมื่อมีพัลส์สุดท้ายเกิดขึ้น เรียกว่า "Overflow"

### การทำงานเป็น Counter

รีจิสเตอร์จะเพิ่มค่าขึ้นทีละ 1 ตามการเปลี่ยนแปลงสถานะของสัญญาณ T0 หรือ T1 ซีพียูจะตรวจสอบระดับในทุกๆเมกซ์ซินไซเคิล โดยจะเพิ่มค่าเมื่อเมกซ์ซินไซเคิลแรกตรวจพบว่าเป็นลอจิก 1 แล้วเมกซ์ซินไซเคิลต่อมาเป็นลอจิก 0 ดังนั้นการนับสัญญาณจะทำได้เร็วที่สุดภายใน 2 เมกซ์ซินไซเคิล หรือความถี่สูงสุดของสัญญาณที่ซีพียูจะตรวจนับได้จะมีค่าเท่ากับ 1/24 ของความถี่สัญญาณนาฬิกา

วงจรมอนิเตอร์จอร์นัม

## ขาสัญญาณที่เกี่ยวข้องกับวงจรมอนิเตอร์จอร์นัม

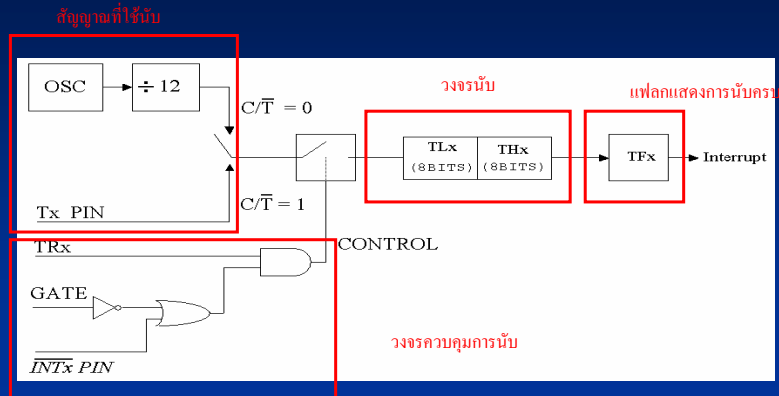


ขาต่อตัวกำเนิดสัญญาณนาฬิกา

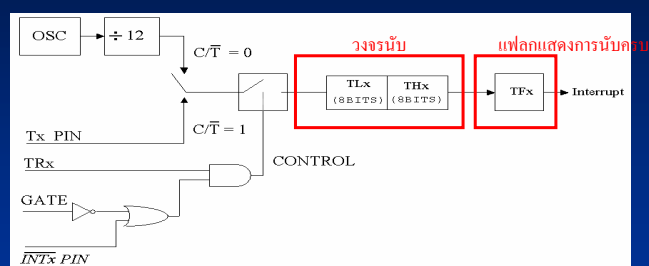
ขาต่อตัวกำเนิดสัญญาณนาฬิกา

วงจรมอนิเตอร์จอร์นัม

## โครงสร้างของวงจรรนับ/จับเวลา



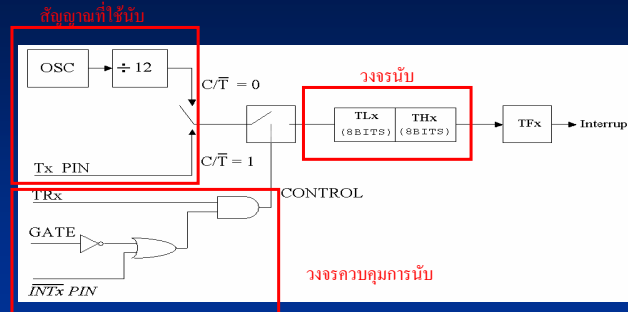
## โครงสร้างของวงจรรนับ/จับเวลา



TL และ TH เป็นวงจรรนับแบบนับขึ้น ซึ่งสามารถกำหนดค่าเริ่มต้นได้ การนับนี้ TL จะเป็นตัวนับก่อนเมื่อนับจนได้ 1 ทุกบิตก็จะกลับไปเป็น 0 แล้วทำให้ TH นับ ขึ้นไป 1 ถ้าทั้ง TH และ TL เป็น 1 หมดสัญญาณพัลส์ลูกต่อไปที่เข้ามาสู่วงจรรนับจะทำให้ ทั้ง TH และ TL กลับเป็น 0 หมดทุกบิตเรียกว่า เกิด Overflow แผง TF จะถูกเซ็ท ให้เป็น 1

วงจรรจับเวลาและวงจรรนับ

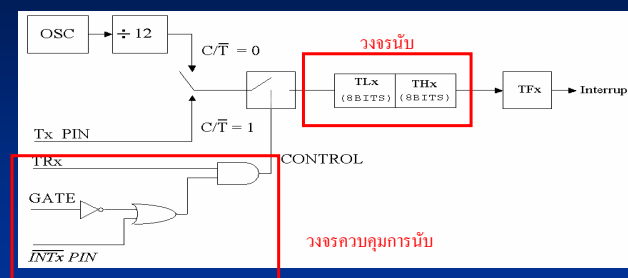
## โครงสร้างของวงจรมับ/จับเวลา



สัญญาณที่จะป้อนให้วงจรมับได้มาจาก 2 แหล่งคือ จากวงจรถวนคุมสัญญาณนาฬิกาภายใน กับจากสัญญาณภายนอกที่ป้อนเข้าทางขา T0 หรือ T1 สัญญาณทั้งสองแหล่งนี้สามารถเลือกได้ด้วย บิตควบคุม C/T และยังถูกควบคุมด้วยวงจรถวนคุมวงจรมับเป็นตัวควบคุมว่าจะให้สัญญาณเข้าสู่วงจรมับหรือไม่ นั่นคือการควบคุมการทำงานของวงจรมับนั่นเอง

วงจรมับและวงจรมับ

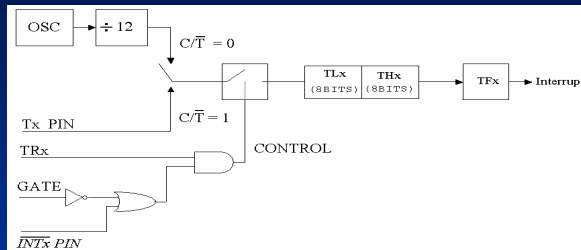
## โครงสร้างของวงจรมับ/จับเวลา



โดยทั่วไปเราควบคุมให้วงจรมับทำงานด้วยโปรแกรม โดยการให้ GATE0 = 0 เมื่อต้องการให้วงจรมับทำการนับเราจะให้ TR = 1 TLO ก็จะได้รับสัญญาณไปนับ (นับขึ้น) แต่ถ้าต้องการควบคุมการทำงานด้วยสัญญาณจากภายนอก (ที่ขา INT) เราจะให้ GATE และ TR เป็น 1 ทั้งคู่ เมื่อกำหนดเช่นนี้ถ้าต้องการให้วงจรมับทำงานก็ให้ป้อน สัญญาณ 1 เข้าที่ขา INT

วงจรมับและวงจรมับ

## โครงสร้างของรีจิสเตอร์ควบคุมสำหรับวงจรมนับ



ชื่อรีจิสเตอร์	รายละเอียด	ตำแหน่ง
TH0	Timer 0 High Byte	8Ch
TL0	Timer 0 Low Byte	8Ah
TH1	Timer 1 High Byte	8Dh
TL1	Timer 1 Low Byte	8Bh
TCON	Timer Control	88h
TMOD	Timer Mode	89h

วงจรมนับและวงจรมนับ

## TCON (Timer Control Register) แอดเดรส 88H

7 6 5 4 3 2 1 0  
TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0

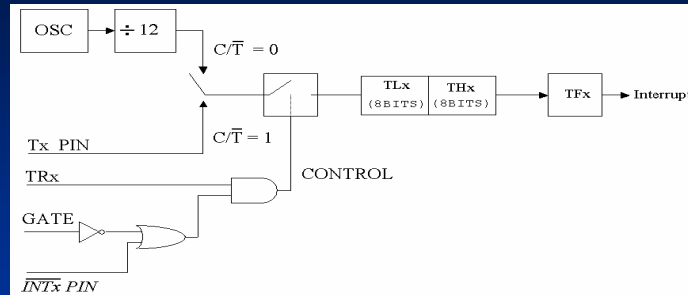
TCON (Timer Control Register)

การกำหนดลักษณะของสัญญาณภายนอกเพื่ออินเทอร์รัพท์ที่ซีพียู

บิต	ชื่อ	ตำแหน่งบิต	ความหมายของฟังก์ชัน
7	TF1	8Fh	Timer 1 Overflow. บิตนี้จะถูกเซตเมื่อ Timer 1 Overflow
6	TR1	8Eh	Timer 1 Run เมื่อบิตนี้เซต Timer 1 จะทำงาน แต่ถ้าบิตเป็น 0 จะหยุด
5	TF0	8Dh	Timer 0 Overflow. บิตนี้จะถูกเซตเมื่อ Timer 0 Overflow
4	TR0	8Ch	Timer 0 Run เมื่อบิตนี้เซต Timer 0 จะทำงาน แต่ถ้าบิตเป็น 0 จะหยุด
3	IE1	8Bh	แฟล็กแสดงการอินเทอร์รัพท์ของ INT1
2	IT1	8Ah	บิตเลือกประเภทสัญญาณอินเทอร์รัพท์ เป็น 0 = ระดับ ถ้าเป็น 1 = ขอบ
1	IE0	89h	แฟล็กแสดงการอินเทอร์รัพท์ของ INTO
0	IT0	88h	บิตเลือกประเภทสัญญาณอินเทอร์รัพท์ เป็น 0 = ระดับ ถ้าเป็น 1 = ขอบ

วงจรมนับและวงจรมนับ

## TCON (Timer Control Register) แอดเดรส 88H



TF1 = Timer 1 Overflow. บิตนี้จะถูกเซ็ทเมื่อ Timer 1 Overflow

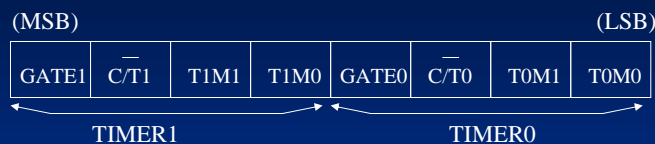
TR1 = Timer 1 Run เมื่อบิตนี้เซ็ท Timer 1 จะทำงาน แต่ถ้าบิตเป็น 0 จะหยุด

TF0 = Timer 0 Overflow. บิตนี้จะถูกเซ็ทเมื่อ Timer 0 Overflow

TR0 = Timer 0 Run เมื่อบิตนี้เซ็ท Timer 0 จะทำงาน แต่ถ้าบิตเป็น 0 จะหยุด

วงจรถ่ายเวลาและวงจรมับ

## TMOD Register - Operation Control



บิต	ชื่อ	หน้าที่
7	GATE1	ถ้าบิตนี้เป็น 1 การทำงานของวงจรถ่ายเวลาจะขึ้นอยู่กับสัญญาณ โลจิก 1 ของ INT1 (P3.3) แต่ถ้าบิตนี้เป็น 0 การทำงานของวงจรถ่ายเวลาจะไม่ขึ้นอยู่กับสัญญาณ INT1
6	C/T1	ถ้าบิตนี้เป็น 0 วงจรนับช่อง 1 จะทำหน้าที่เป็นวงจรถ่ายเวลา แต่ถ้าเป็น 1 จะทำหน้าที่เป็นวงจรมับสัญญาณที่ขา T1 (P3.5)
5	TIM1	บิตกำหนดโหมดการทำงานของ Timer 1 บิตที่ 1
4	TIM0	บิตกำหนดโหมดการทำงานของ Timer 1 บิตที่ 0

วงจรถ่ายเวลาและวงจรมับ

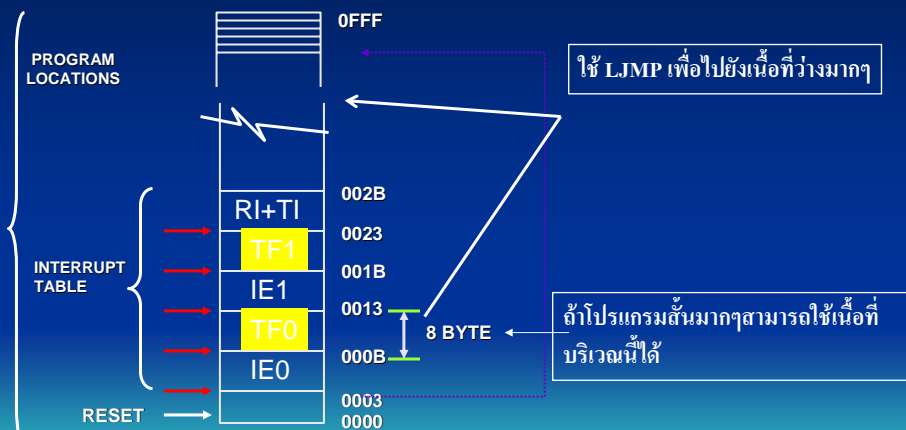
## TMOD Register - Operation Control



บิต	ชื่อ	หน้าที่
3	GATE0	ถ้าบิตนี้เป็น 1 การทำงานของวงจรถับเวลาจะขึ้นอยู่กับสัญญาณ โลจิก 1 ของ INT0 (P3.2) แต่ถ้าบิตนี้เป็น 0 การทำงานของวงจรถับเวลาจะไม่ขึ้นอยู่กับสัญญาณ INT1
2	C/T0	ถ้าบิตนี้เป็น 0 วงจรนับช่อง 1 จะทำหน้าที่เป็นวงจรถับเวลา แต่ถ้าเป็น 1 จะทำหน้าที่เป็นวงจรถับสัญญาณที่ขา T0 (P3.4)
1	T0M1	บิตกำหนดโหมดการทำงานของ Timer 0 บิตที่ 1
0	T0M0	บิตกำหนดโหมดการทำงานของ Timer 0 บิตที่ 0

วงจรถับเวลาและวงจรถับ

## Timer Interrupt



วงจรถับเวลาและวงจรถับ

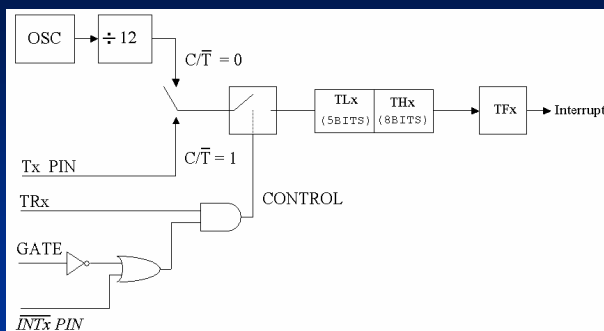
## รูปแบบการทำงาน

- Mode 0 (13-bit Timer)
- Mode 1 (16-bit Timer)
- Mode 2 (8-bit Timer with Auto-Reload)
- Mode 3 (Two 8-bit Timers)

ข้อควรระวังเมื่อจะเปลี่ยนโหมดการทำงานของวงจร ต้องหยุดการทำงาน  
ของวงจรมีก่อน

วงจรจับเวลาและวงจรมับ

## Timer/Counter Mode 0 13-bit Counter



### ตัวอย่างการกำหนดค่า

GATE = 0

C/T = 0

M1 = 0

M0 = 0

พอกำหนดค่า TL TH แล้วให้ TR = 1

วงจรจับเวลาและวงจรมับ



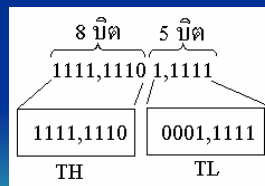
ตัวอย่าง การจับเวลาในโหมด 0 ให้จับเวลา 32 uS กำหนดให้สัญญาณนาฬิกามีความถี่ 12 MHz

วิธีทำ  $F_{osc} = 12 \text{ MHz}$   
 $1 \text{ Machine Cycle} = 12/12 = 1 \text{ uS}$   
 ต้องการเวลา 32 uS ดังนั้น ต้องนับ  $32/1 = 32$

แปลงเลข 32 เป็น binary 13 บิต = 0000000100001B

แปลงเป็น 2's Complement = 111111011111B

จัดแบ่งเป็น 2 ไบต์ โดยไบต์แรกสำหรับ TL นับจาก บิต LSB (ขวาสุด) มาทางซ้าย 5 บิต ที่เหลือจะเป็นสำหรับ TH สำหรับบิตที่เหลือของ TL อีก 3 บิต จะเป็นบิต MSB มีค่าเป็น 0 หรือ 1 ก็ได้



วงจรถ่ายโอนและวงจรมินิ

ตัวอย่าง โปรแกรมเพื่อใช้ทดสอบการ

ทำงาน

```

ORG 0000H
START: MOV TMOD, #00H
        MOV TLO, #1FH
        MOV TH0, #0FEH
        SETB TR0
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
    
```

```

NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
    
```

```

NOP
NOP
NOP
NOP
NOP
NOP
END
    
```

วงจรถ่ายโอนและวงจรมินิ

## ตัวอย่างคำสั่งของวงจรรนับเมื่อทำงานในโหมด 0

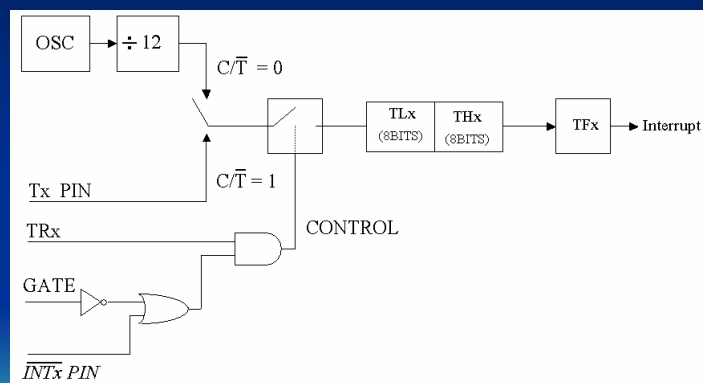
```

ORG      0000H
START:   SJMP  MAIN
;
ORG      000BH
INC      R0
INC      R1
INC      R2
INC      R3
RETI
;
MAIN:    SETB  EA
         SETB  ETO
         MOV   TMOD,#00H      ;Set timer0 mode
         MOV   TL0,#18H      ;Timer 0 start-up value
         MOV   TH0,#0FCH
         SETB  TR0           ;Start Timer 0
         SJMP  $
;
END

```

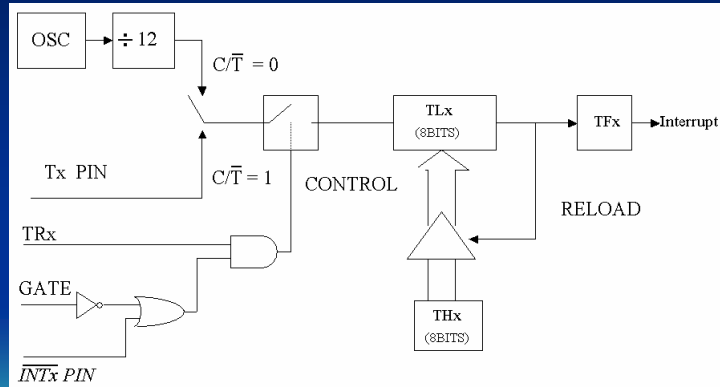
วงจรรนับเวลาและวงจรรนับ

## Timer/Counter Mode 1 16-bit Counter



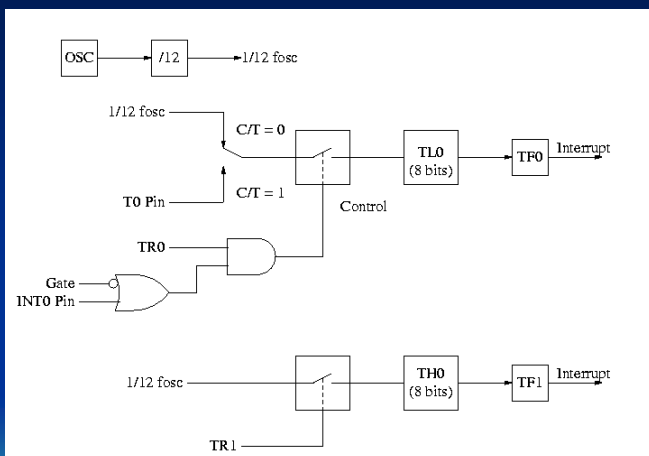
วงจรรนับเวลาและวงจรรนับ

## Timer/Counter Mode 2 8-bit Timer with Auto-Reload



วงจรมินิคอมพิวเตอร์

## Timer/Counter Mode 3 Two 8-bit Timers



ใช้ได้กับ Timer 0 เท่านั้น ถ้านำไปใช้กับ Timer 1 วงจรจะหยุดทำงาน โหมดนี้ TH0 และ TLO จะทำงานแยกกัน โดยอิสระ โดย TLO ถูกควบคุมด้วย TCON และ INTO ส่วน TH0 ถูกควบคุมด้วย TR1 ถ้าใช้โหมดนี้ Timer1 ใช้ Interrupt ไม่ได้

วงจรมินิคอมพิวเตอร์