

# 微介實驗七

## 計時計數器

日期:2023/11/19

報告者:林柏均

# Outline

- 實驗目的
- 材料清單
- 實驗原理
- 實驗電路
- 實驗程式

# Outline

- 實驗目的
- 材料清單
- 實驗原理
- 實驗電路
- 實驗程式

# 實驗目的

- 了解8051之計時/計數器的功能與相關暫存器設定

# Outline

- 實驗目的
- 材料清單
- 實驗原理
- 實驗電路
- 實驗程式

# 材料清單

器材名稱		數量
AT89S51		1
12MHz 石英震盪器		1
LED二極體		8
按壓開關		2
電阻	1k $\Omega$	9
	10 k $\Omega$	1
電容	20pF	2
	10 $\mu$ F	1

# Outline

- 實驗目的
- 材料清單
- 實驗原理
- 實驗電路
- 實驗程式

# 實驗原理

- 計時器/計數器Timer/Counter
  - 中斷程式，當溢位時產生中斷
  - 計時與計數的差別在於觸發方式
    - 計時器是依靠內部的時脈計時
    - 計數器是依靠外部的接腳計數
  - 8051提供2組計時/計數器(Timer0、Timer1)
  - 8052則是3組(Timer0、Timer1、Timer2)



# 實驗原理

- 計時/計數器中斷相關暫存器
  - TCON
    - 控制Timer/Counter開關與否
  - TMOD
    - 控制Timer/Counter的模式
  - TH/TL
    - 控制Timer觸發中斷所需的時間或是Counter觸發中斷所需的記數次數
  - IE & IP
    - 中斷開關與優先權，詳見實驗六

# 實驗原理

- 計時/計數器控制暫存器TCON(可位元定址)
  - 8-bits，僅TCON.4 ~ 7與計時/計數器有關
  - 當TRX = 1時，啟動TimerX；反之則關閉TimerX
  - 當Timer X的THX溢位時，TFX = 1；進入中斷向量後TFX = 0
  - 例如今天想啟動Timer 0與Timer 1，須設定TCON為0x50

• 0x50 = 01010000B

TCON

TF1	TR1	TF0	TR0	IE1	IT	IE0	IT0
0	1	0	1	0	0	0	0

# 實驗原理

- 計時/計數器模式控制暫存器TMOD(不可位元定址)
  - 一個Timer/Counter各自使用4 bits
  - GATE
    - 如果GATE = 1，代表該Timer/Counter外部啟動  
TCON的TRX = 1且接腳INTX為HIGH時TimerX才會啟動
    - 反之GATE = 0時僅需TRX = 1便可啟動TimerX
  - C/T = 0時為計時器，C/T = 1時為計數器

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

# 實驗原理

- 計時/計數器模式控制暫存器TMOD(不可位元定址)
  - M1、M0為控制計時/計數模式

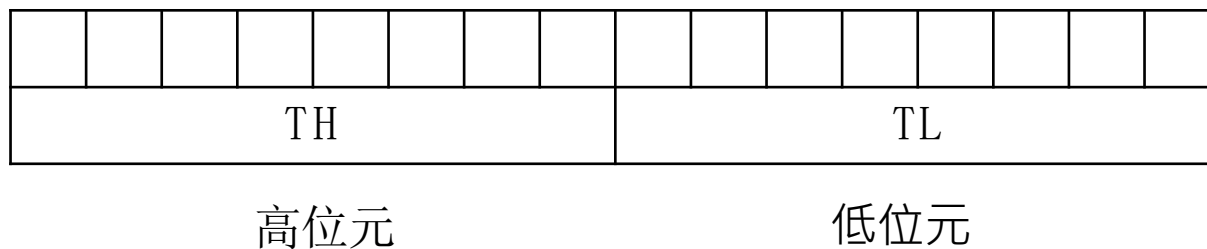
M1	M0	模式
0	0	Mode 0: 13-bits Timer/Counter
0	1	Mode 1: 16-bits Timer/Counter
1	0	Mode 2: 8-bits auto-reload Timer/Counter
1	1	Mode 3: 一個8-bits Timer/Counter，一個8-bits Timer

TMOD

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

# 實驗原理

- TH/TL暫存器
  - Timer/Counter的計時記數是利用TH和TL兩個暫存器達成
  - 當TH暫存器發生溢位時會使Timer Flag(TF)升起，讓程式有分支跳躍的根據
    - 根據Timer/Counter模式的不同，TH和TL會有不同的使用方法



# 實驗原理

- 計時/計數器中斷相關暫存器設定範例
  - 啟動Timer 0和1與其中斷，其中Timer 1為counter，Mode 1；Timer 0需外部啟動，Mode 2。相關暫存器需如此設定：

– IE

EA	-	-	ES	ET1	EX1	ET0	EX0
1	0	0	0	1	0	1	0

– TCON

TF1	TR1	TF0	TR0	IE1	IT	IE0	IT0

– TMOD

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

# 實驗原理

- 計時/計數器中斷相關暫存器設定範例
  - 啟動Timer 0和1與其中斷，其中Timer 1為counter，Mode 1；Timer 0需外部啟動，Mode 2。相關暫存器需如此設定：

– IE

EA	-	-	ES	ET1	EX1	ET0	EX0
1	0	0	0	1	0	1	0

– TCON

TF1	TR1	TF0	TR0	IE1	IT	IE0	IT0
0	1	0	1	0	0	0	0

– TMOD

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

# 實驗原理

- 計時/計數器中斷相關暫存器設定範例
  - 啟動Timer 0和1與其中斷，其中Timer 1為counter，Mode 1；Timer 0需外部啟動，Mode 2。相關暫存器需如此設定：

– IE

EA	-	-	ES	ET1	EX1	ET0	EX0
1	0	0	0	1	0	1	0

– TCON

TF1	TR1	TF0	TR0	IE1	IT	IE0	IT0
0	1	0	1	0	0	0	0

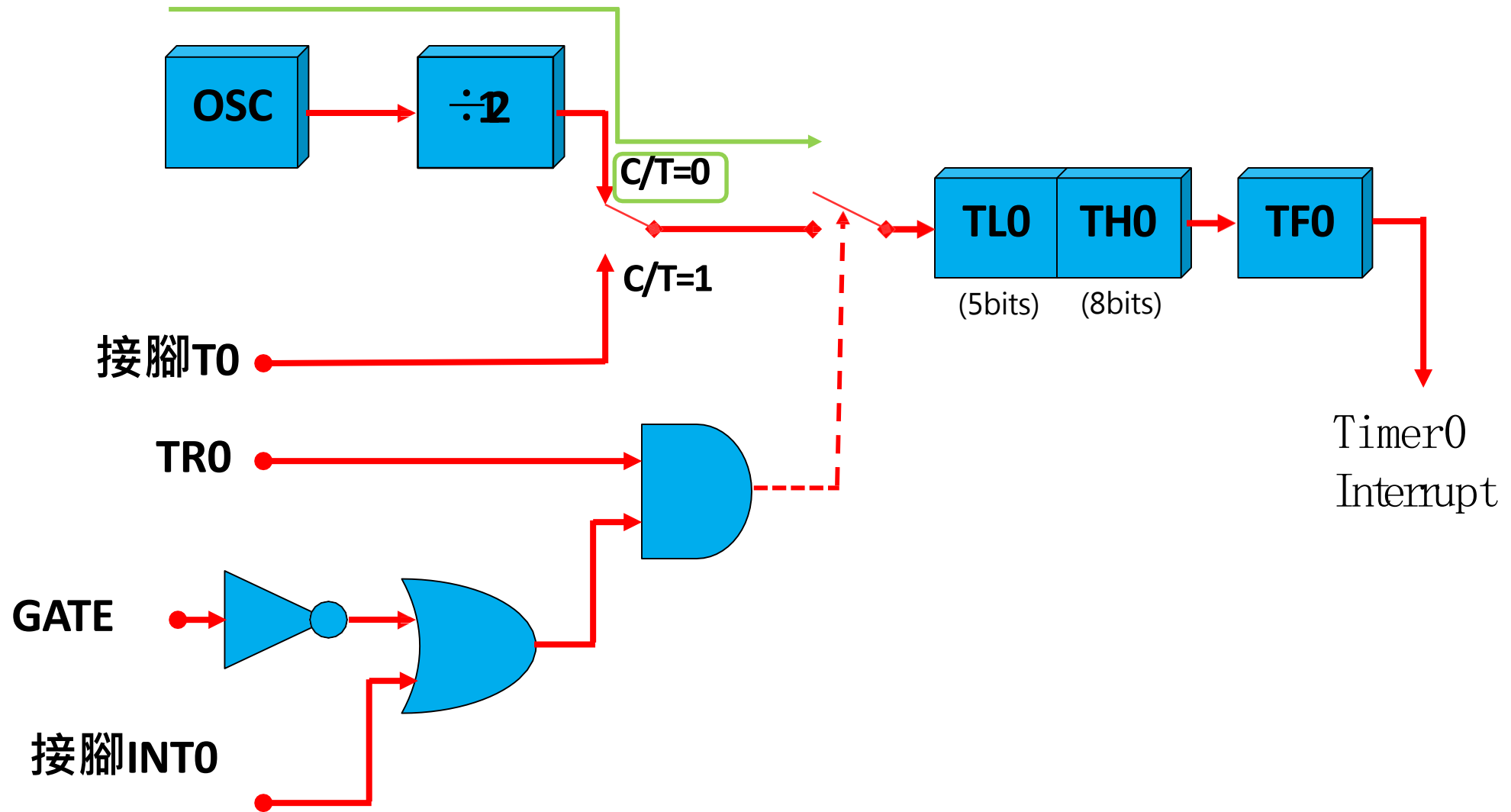
– TMOD

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



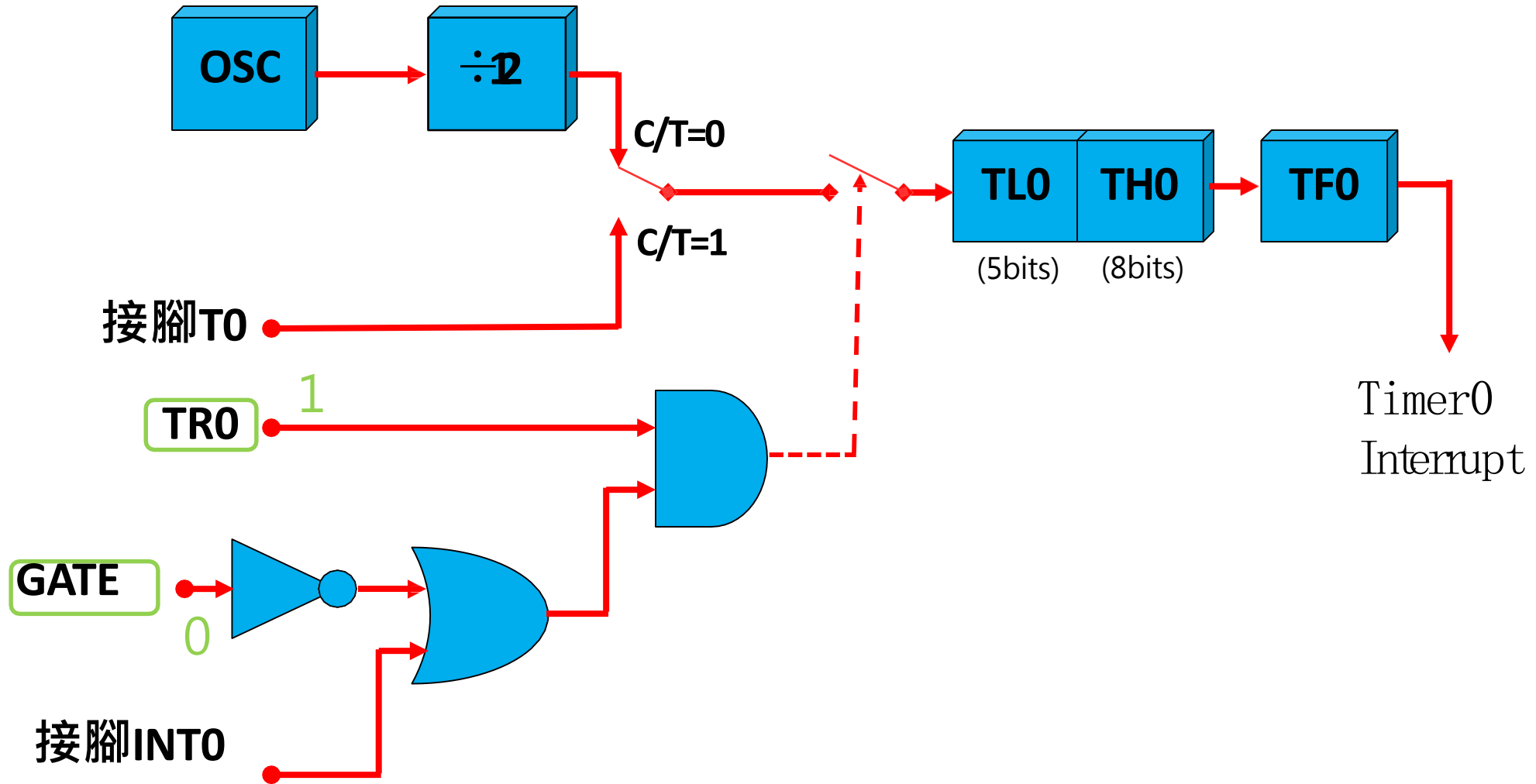
# 實驗原理

以Timer0的mode0為例



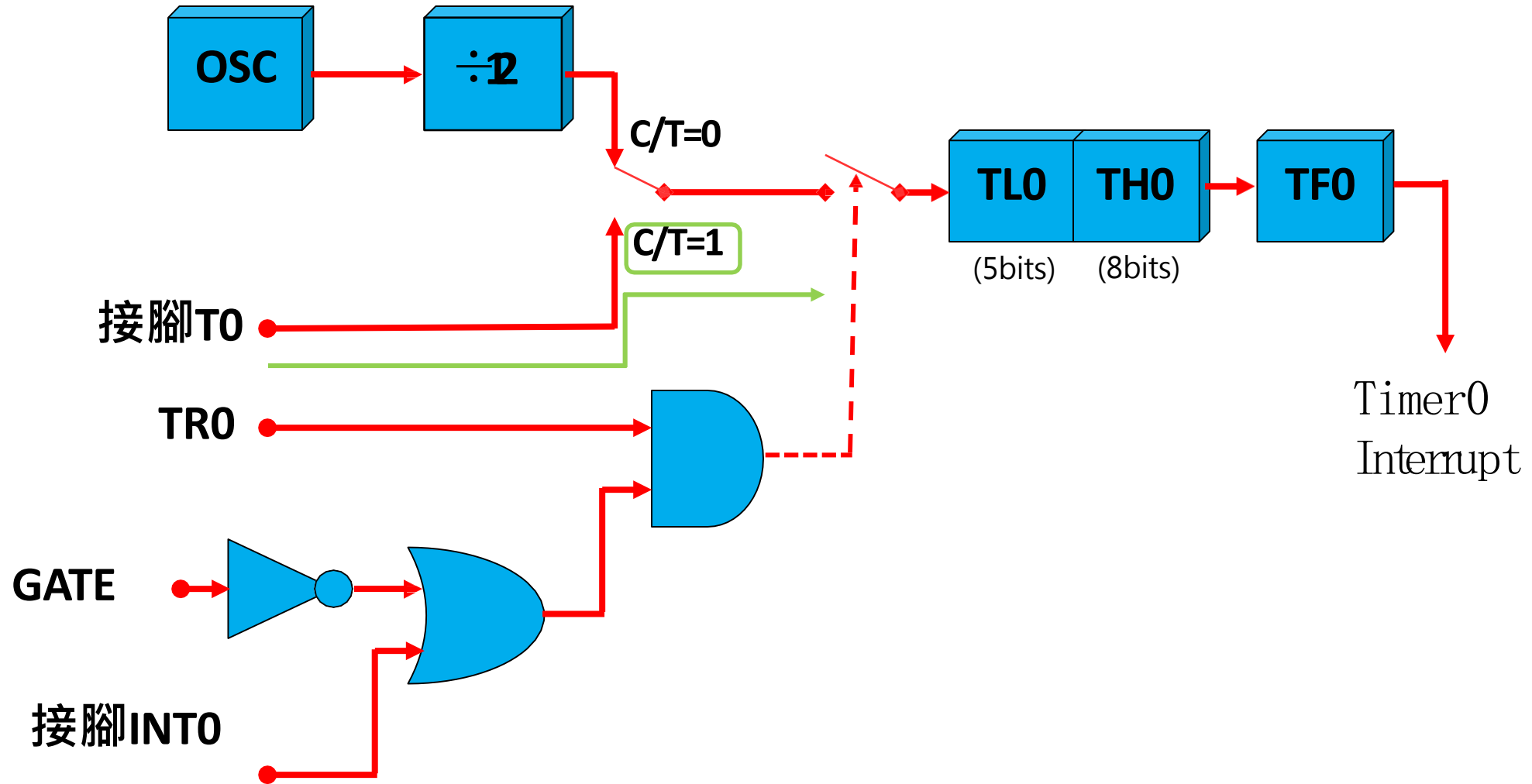
# 實驗原理

以Timer0的mode0為例



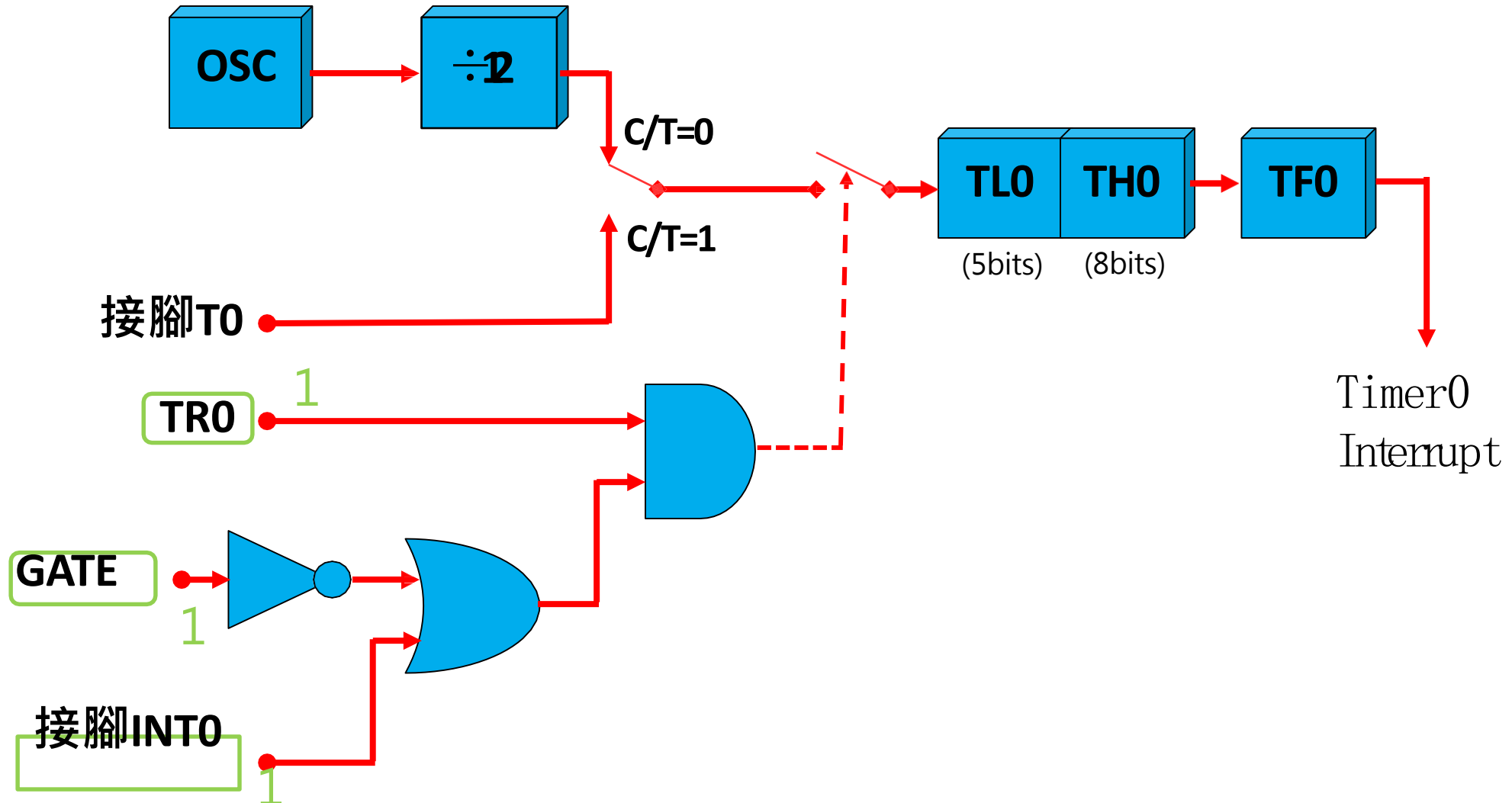
# 實驗原理

以Timer0的mode0為例



# 實驗原理

以Timer0的mode0為例



# 實驗原理

- Timer/Counter 模式介紹

- Mode 0: 13-bits的Timer/Counter

- TL僅使用前5個bits，可以視為一個13-bits的暫存器隨Machine cycle累加，溢位時TF升起

0	0	0	0	0	0	0	0	-	-	-	0	0	0	0	0
TH								TL							

- Mode 1: 16-bits的Timer/Counter

- TL與TH皆使用8個bits，可以視為一個16-bits的暫存器隨Machine cycle累加，溢位時TF升起

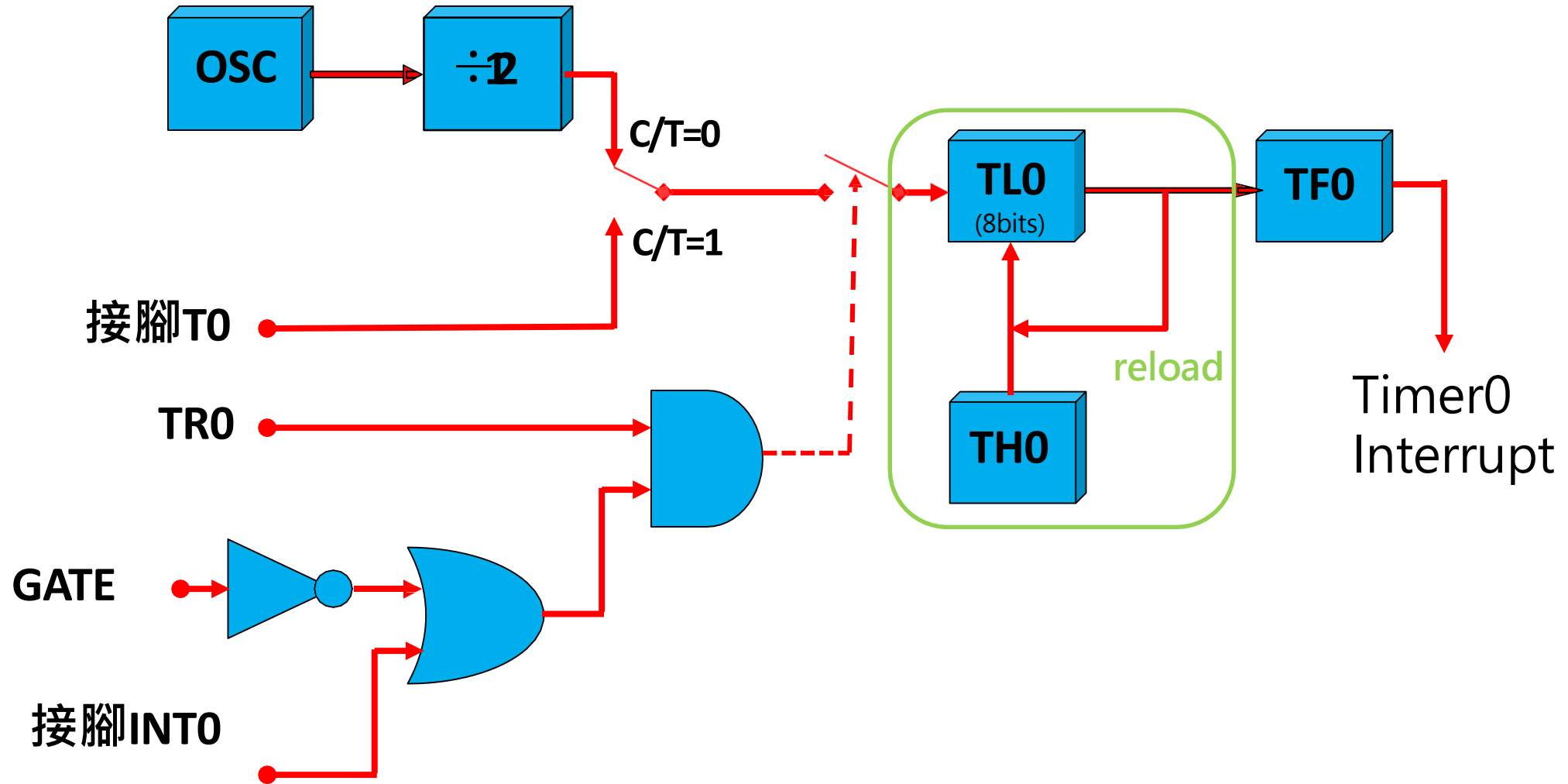
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH								TL							

# 實驗原理

- Timer/Counter 模式介紹
  - Mode 2: 8-bits且具auto-reload功能的Timer/Counter
    - 僅使用TL計時/計數，TH用作紀錄TL的預設值使用
    - 在其他模式中，TF升起後下次的計時計數會從0開始計算，但通常我們不希望每次都是從0開始，因此都會在ISR中設定下次TH和TL的初始值，Mode 2便是省去設定TH和TL的動作
  - Mode 3: 一個8-bits Timer/Counter，一個8-bits Timer
    - Mode 3的狀況較為複雜且不常使用，以下以Timer 0的Mode 3為例
    - TL0是一個由C/T控制的Timer/Counter，溢位時TF0升起
    - TH0是一個Timer，溢位時TF1升起

# 實驗原理

以Timer0的mode2為例



# 實驗原理

- TH/TL 計算與設定
  - 因本實驗使用的石英震盪器為12MHz
    - Machine cycle頻率為 $12\text{MHz} / 12 = 1\text{MHz}$
    - 一個Machine cycle需要一微秒，意即TL每一微秒+1
  - 為了方便計算中斷時間以提高計時精準度，通常會將TH和TL設定為特定數值



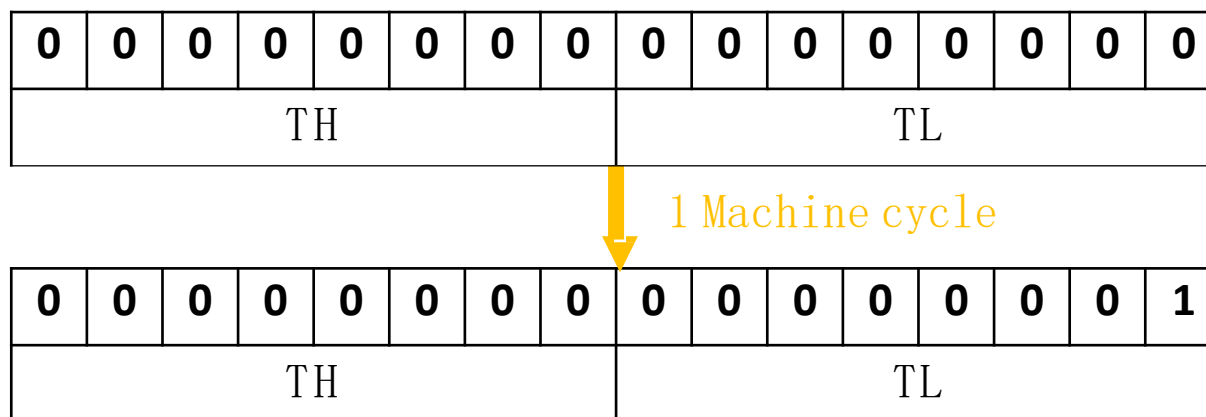
# 實驗原理

- TH/TL 計算與設定 ( 以Mode 1為例 )

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH								TL							

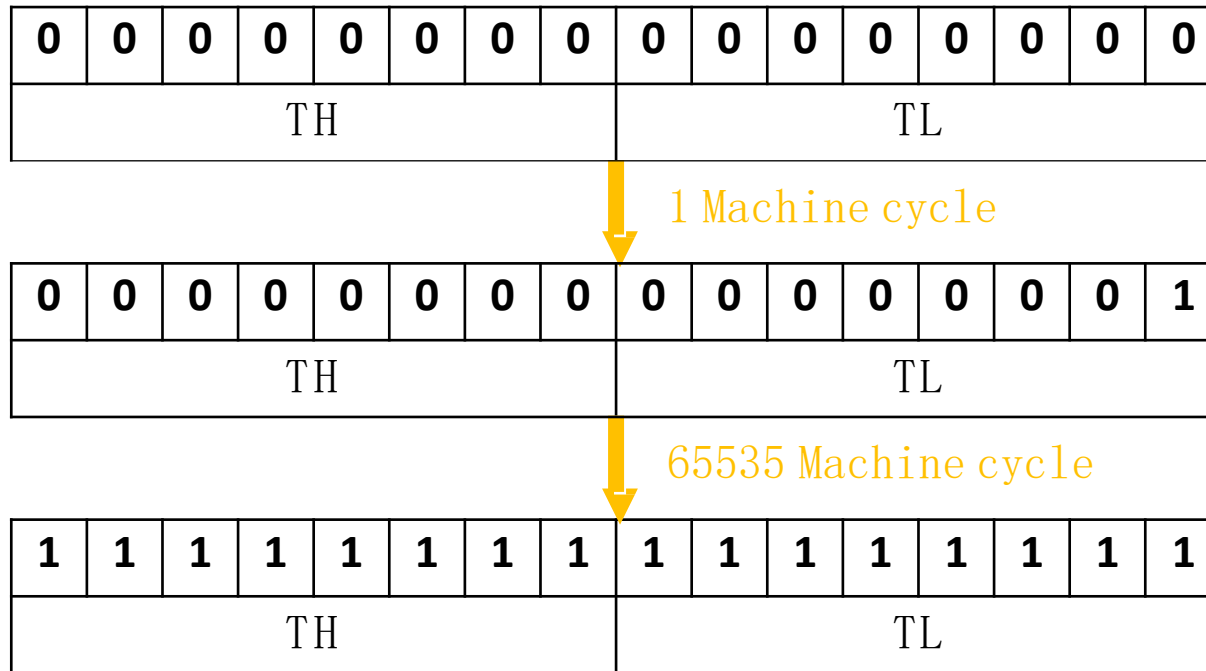
# 實驗原理

- TH/TL 計算與設定 ( 以Mode 1為例 )



# 實驗原理

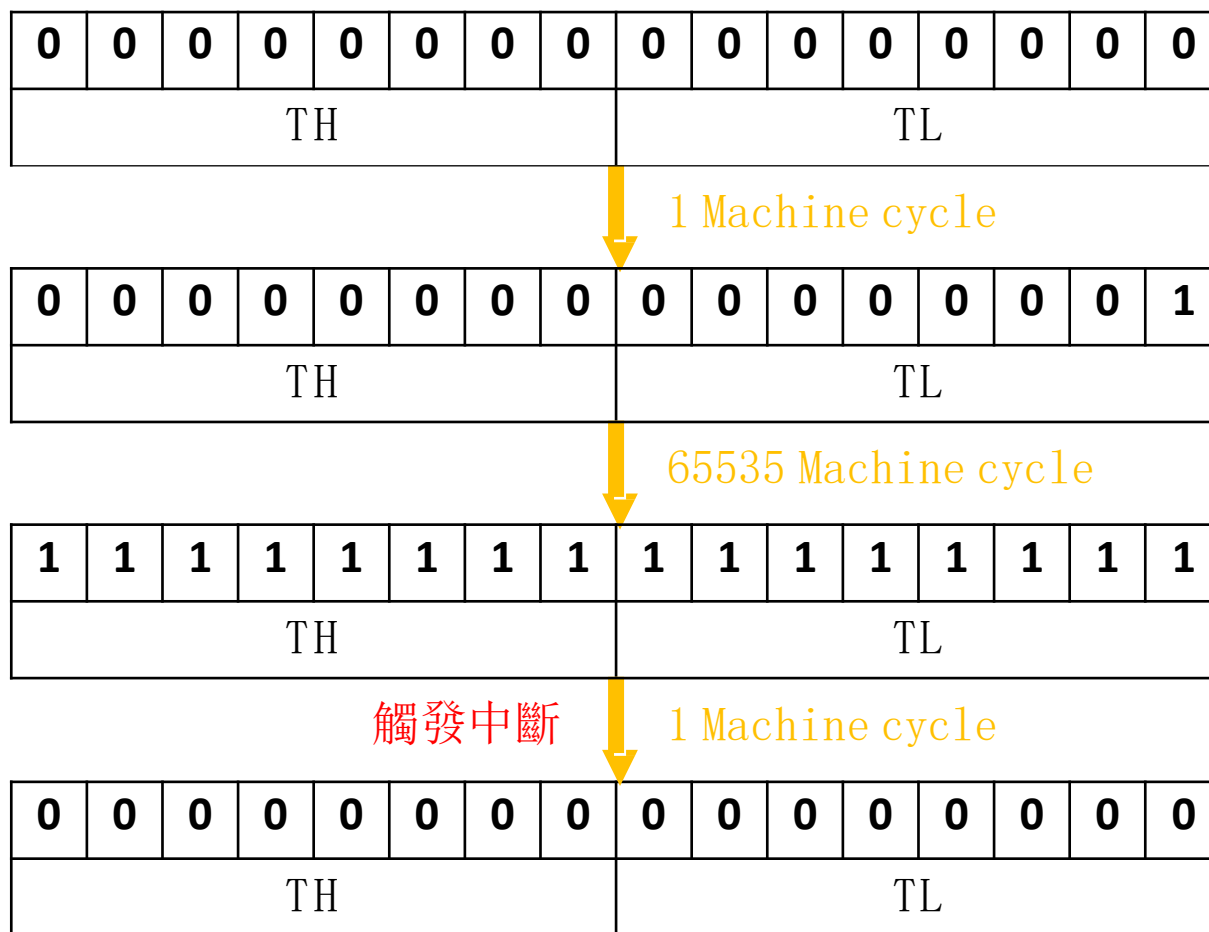
- TH/TL 計算與設定 ( 以Mode 1為例 )



← 即將溢位

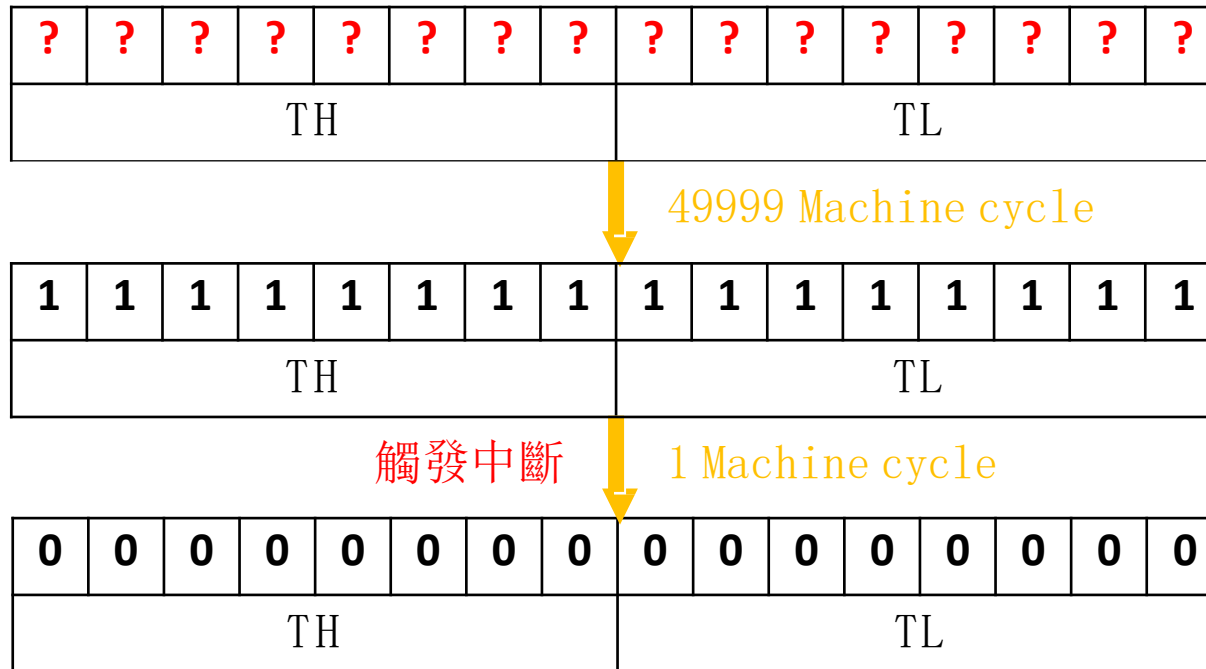
# 實驗原理

- TH/TL 計算與設定 ( 以Mode 1為例 )



# 實驗原理

- TH/TL 計算與設定 ( 以Mode 1為例 )



# 實驗原理

- TH/TL 計算與設定 ( 以Mode 1為例 )
  - 目標是讓TH和TL在50000個Machine cycle後溢位
  - 將TH和TL共16 bits設定為 $65536 - 50000 = 15536 = 11110010110000$ 
    - 將前八位放進TH，後八位放進TL並將剩餘位元補0

0	0	1	1	1	1	0	0	1	0	1	1	0	0	0	0
TH								TL							

# 實驗原理

- TH/TL 計算與設定

Mode1:每50ms觸發一次中斷

```
TH0=(65536-50000)/256;    //除完的商寫入TH0  
TL0=(65536-50000)%256;    //除完的餘數寫入TL0
```

Mode2:每250us觸發一次中斷

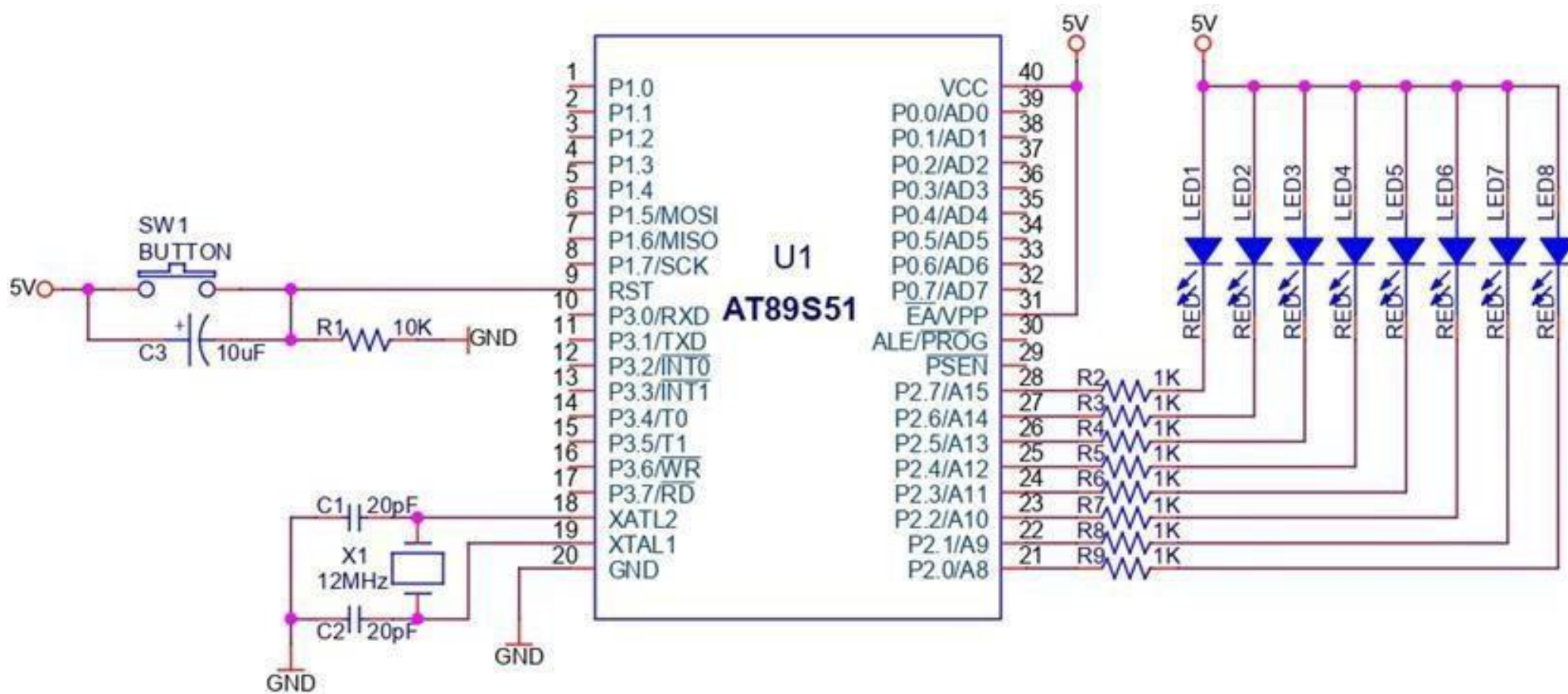
```
TH0=0x06;                //TH0 = 256 - 250  
TL0=0x06;                //TL0 = 256 - 250
```

# Outline

- 實驗目的
- 材料清單
- 實驗原理
- **實驗電路**
- 實驗程式



# 實驗電路



# Outline

- 實驗目的
- 材料清單
- 實驗原理
- 實驗電路
- 實驗程式

# 實驗程式

- LED閃爍
  - 連接於P2.0的LED使用delay function作為閃爍的時間間隔
  - 連接於P2.1的LED使用Timer 0的interrupt作為閃爍的時間間隔
  - 調整參數，使兩個LED的閃爍時間間隔接近一致

# 實驗程式

1. `#include <regx51.h>` *// include header file for 8051*
2. `#define TH0_init 0x06` *//TH0\_init= 256 - 250*
3. `#define TL0_init 0x06` *//TL0\_init = 256 - 250*
4. `#define Timer0_int_exe_time 2000` *//the parameter can be changed*
5. `void delay(int t);`
6. `int counter;`

# 實驗程式

```
7. void main()  
8. {  
9.     TMOD = 0x02;           //set timer0 to mode 2(8-bits and auto-reload)  
10.    IP = 0x02;             //timer0 interrupt has high priority  
11.    IE = 0x82;             //enable timer0 interrupt  
12.    P2 = 0xff;  
13.    TLO = TLO_init; TH0 = TH0_init; //set TH0 & TLO  
14.    TCON = 0x10;           //enable timer0  
15.    while (1)  
16.    {  
17.        delay(10000);       // adjust this parameter to match timer  
18.        P2_0 = ~P2_0;       //inverse P2.0  
19.    }  
20. }
```

# 實驗程式

```
21. void timer0_interrupt(void) interrupt 1           // 'interrupt 1' is int vector of INT0
22. {
23.     counter++;
24.     if(counter == Timer0_int_exe_time)           //250clock cycle * 2000 = 0.5 second
25.     {
26.         P2_1 = ~P2_1;                             //inverse P2.0
27.         counter = 0;
28.     }
29. }
30. //delay function
31. void delay(int t)
32. {
33.     while(t--);
34. }
```

# 實驗程式

- 本次實驗開始使用C語言，請開啟一個新project並在project中加入STARTUP.A51



- 如果已經習慣組合語言或是碰到必須使用組合語言指令的場合，可參考以下網址以便在C語言中使用組合語言指令  
– <http://qmierhuang.blogspot.com/2016/05/8051c.html>

# 實驗程式

- 在C語言中，觸發中斷時會自動呼叫中斷的function，該function有著指定的宣告格式
  - void 中斷function名稱 (void) interrupt 中斷型號 {}

中斷源	中斷向量	中斷型號
外部中斷0	0x03	0
計時/計數器0	0x0B	1
外部中斷1	0x13	2
計時/計數器1	0x1B	3
串列埠	0x23	4

- 例： void ex0\_interrupt(void) interrupt0



# 實驗程式

- 8051 C語言程式常用的資料型態(data type)

資料型態	Bits(位元)	Data Range(數值範圍)
Unsigned char	8-bit	0 to 255
(signed)char	8-bit	-128 to +127
Unsigned int	16-bit	0 to 65535
(signed) int	16-bit	-32768 to +32767
sbit	1-bit	0 to 1
bit	1-bit	0 to 1
sfr	8-bit	0 to 255

Q & A