

HC16P122A1/B1

數據手冊

16 引腳 8 位
AD 型 OTP 單片機

目錄

1	產品簡述.....	1
1.1	特性.....	1
1.2	系統框圖.....	2
1.3	引腳圖.....	3
1.4	引腳說明.....	3
1.5	引腳電路.....	4
2	中央處理器（CPU）	5
2.1	記憶體.....	6
2.1.1	程式記憶體（ROM）	6
2.2	晶片配置選擇表.....	11
2.2.1	晶片配置字.....	11
2.3	數據寄存器（RAM）	12
2.3.1	STATUS 寄存器.....	14
2.3.2	PC 寄存器.....	15
2.4	尋址模式.....	15
2.4.1	立即尋址.....	15
2.4.2	直接尋址.....	15
2.4.3	間接尋址.....	15
2.5	堆疊.....	16
3	複位.....	17
3.1	上電複位.....	17
3.2	外部複位.....	18
3.3	欠壓複位.....	19
3.4	看門狗定時器複位.....	19
3.5	PCON 寄存器.....	19
4	系統時鐘.....	20
4.1	概述.....	20
4.2	時鐘框圖.....	20
4.3	系統高頻時鐘.....	21
4.3.1	內部高頻 RC 振盪器.....	21
4.3.2	外部高頻時鐘.....	21
4.4	系統低頻時鐘.....	22
4.4.1	低頻晶體振盪器.....	23
4.4.2	低頻 RC 振盪器.....	23
5	系統工作模式.....	24
5.1	模式切換舉例.....	25
5.2	高低頻模式切換.....	26
5.3	喚醒時間.....	27

5.4	OSCCON 寄存器.....	27
6	中斷源.....	28
6.1	內核中斷.....	28
6.2	外設中斷.....	30
6.3	GIE 全局中斷.....	32
6.4	中斷保護.....	32
6.5	TIMER0 定時器中斷.....	33
6.6	INT0 外部中斷.....	33
6.7	PORT 電平變化中斷.....	34
6.8	TIMER2 定時器中斷.....	35
6.9	TIMER1 中斷.....	36
6.10	AD 中斷.....	37
6.11	CCP 中斷.....	37
6.12	UART 中斷.....	37
6.13	PWM 中斷.....	37
6.14	多中斷操作.....	37
7	I/O 口.....	39
7.1	I/O 口輸入輸出控制寄存器.....	39
7.2	I/O 口上拉控制寄存器.....	40
7.3	I/O 口下拉控制寄存器.....	40
7.4	I/O 驅動控制寄存器.....	40
7.5	I/O 口數據寄存器.....	41
7.6	I/O 口管腳配置寄存器.....	41
8	定時器/計數器.....	42
8.1	看門狗定時器.....	42
8.2	TIMER0 定時器/計數器.....	43
8.3	TIMER1 定時器/計數器.....	46
8.3.1	Timer1 控制寄存器.....	46
8.4	TIMER2 定時器.....	47
8.4.1	Timer2 控制寄存器.....	48
8.4.2	Timer2 計數寄存器.....	48
8.4.3	Timer2 週期寄存器.....	48
8.5	CCP 模組.....	49
8.5.1	捕捉模式.....	49
8.5.2	比較模式.....	50
8.5.3	PWM 模式.....	51
9	PWM 模組.....	57
9.1	PWM 特性.....	57
9.2	PWM 輸出模式.....	57
9.2.1	互補輸出模式.....	57
9.2.2	獨立輸出模式.....	57

9.3	PWM 相關寄存器.....	57
9.3.1	PWM 使能寄存器.....	57
9.3.2	PWM0 控制寄存器.....	58
9.3.3	PWM 模式寄存器 PWMM.....	59
9.3.4	PWM0 週期/占空比寄存器.....	59
9.3.5	死區時間.....	61
10	模數轉換 (ADC).....	62
10.1	A/D 引腳控制寄存器.....	62
10.2	A/D 控制寄存器.....	63
10.3	ADC 使用.....	67
11	串行口通信.....	69
11.1	串行口的控制寄存器 SCON.....	69
11.2	串行口數據緩衝寄存器 SBUF.....	70
11.3	輔助寄存器 AUXR.....	70
11.4	獨立串列傳輸速率發生器寄存器 BRT.....	71
11.5	自動地址識別.....	71
11.6	串行口工作模式.....	73
11.6.1	串行口工作模式 0: 同步移位寄存器.....	73
11.6.2	串行口工作模式 1: 8 位 UART, 串列傳輸速率可變.....	74
11.6.3	串行口工作模式 2: 9 位 UART, 串列傳輸速率固定.....	76
11.6.4	串行口工作模式 3: 9 位 UART, 串列傳輸速率可變.....	78
11.7	串口通信中波特率的設置.....	79
12	軟體 LCD 驅動.....	82
12.1	相關寄存器.....	82
12.2	軟體 LCD 操作說明.....	82
13	指令表.....	84
14	電性參數.....	85
14.1	極限參數.....	85
14.2	直流特性.....	85
14.3	交流特性.....	86
15	開發工具.....	88
15.1	OTP 燒錄器 (HC-PM18 5.0)	88
15.2	HC-IDE.....	88
16	封裝資訊.....	89
16.1	SOP16.....	89
16.2	SOP8.....	90
17	修改記錄.....	91

1 產品簡述

HC16P122A1/B1是一顆採用高速低功耗CMOS工藝設計開發的8位高性能精簡指令單片機，內部有2K×16位一次性編程ROM(OTP-ROM)，256×8位的數據寄存器（RAM），2組雙向I/O口，三個Timer定時器/計數器，一個PWM模組，兩個CCP模組，一個AD模組，支持一路UART。一個14通道的12位模數轉換器，多個系統時鐘，四種系統工作模式以及多個中斷源。這款單片機可以廣泛應用於液晶梳子、電動車碼表等產品。

1.1 特性

- CPU 特性
 - 36條高性能精簡指令
 - 2K×16位的OTP程式記憶體
 - 256×8位的數據記憶體
 - 8級堆疊緩存器
 - 2T/4T時鐘模式
 - 立即、直接和兩組間接尋址模式
 - 16位RDT查表
- I/O 口
 - 2組雙向I/O口：PORTA，PORTB
 - 最多14個雙向I/O口
 - 所有端口4級驅動電流配置(除PORTB5)可直接驅動LED（4mA）
 - 最多14個可編程弱上拉/下拉口（PA、PB）
 - 所有端口具有喚醒功能的電平變化中斷
 - 具體端口配置詳見第7章
 - 所有端口支持軟體1/2bias COM口功能（除VPP）
- 三個 Timer 定時器/計時器
 - Timer0：帶有8位預分頻器的8位定時器/計數器
 - Timer1：帶有預分頻器的16位定時器/計數器
 - Timer2：帶有8位週期寄存器的8位定時器
- 兩個CCP模組
 - 16位捕捉、16位比較、最高10位PWM
- PWM模組
 - 1組可編程帶死區控制的固定相位PWM 1*12Bit
- BOR複位系統
 - 1.5V/1.8V/2.0V/2.2V/2.4V/2.7V/3.6V
- 一個UART模組
- AD 模數轉換器
 - 12位轉換解析度
 - 最多14個模擬輸入通道（13個外部ADC輸入，1個內部1/4VDD檢測）
 - 內部參考電壓(VDD、4V、3V、2V、1.2V)和外部參考電壓
- 雙系統時鐘
 - 高頻系統時鐘
 - 高頻晶體振盪器：最高 20MHz
 - 內部 RC 振盪器：高達 32MHz
 - 低頻系統時鐘
 - 低頻晶體振盪器： 32.768KHz
 - 低頻 RC 振盪器： 32KHz
- 系統工作模式
 - 高頻模式
 - 低頻模式
 - 休眠模式
 - 綠色模式
- 中斷源
 - 定時器中斷：Timer0、Timer1和Timer2
 - INT0外部中斷
 - 所有IO電平變化中斷
 - CCP1/CCP2中斷
 - ADC中斷
 - UART中斷
 - PWM中斷
- 複位
 - 上電複位（POR）
 - 外部複位（MCLR Reset）
 - 欠壓複位（BOR）
 - 看門狗定時器複位（WDT Reset）
- 封裝形式
 - SOP16/8
 - DFN16

■ 抗干擾能力

- ESD $\geq$ 3000V
- EFT $\geq$ 4000V

■ 最低工作電壓

- 1.5V~5.5V (Fsys= 4MHz)
- 1.8V~5.5V (Fsys= 8MHz)
- 4.5V~5.5V (Fsys=16MHz)

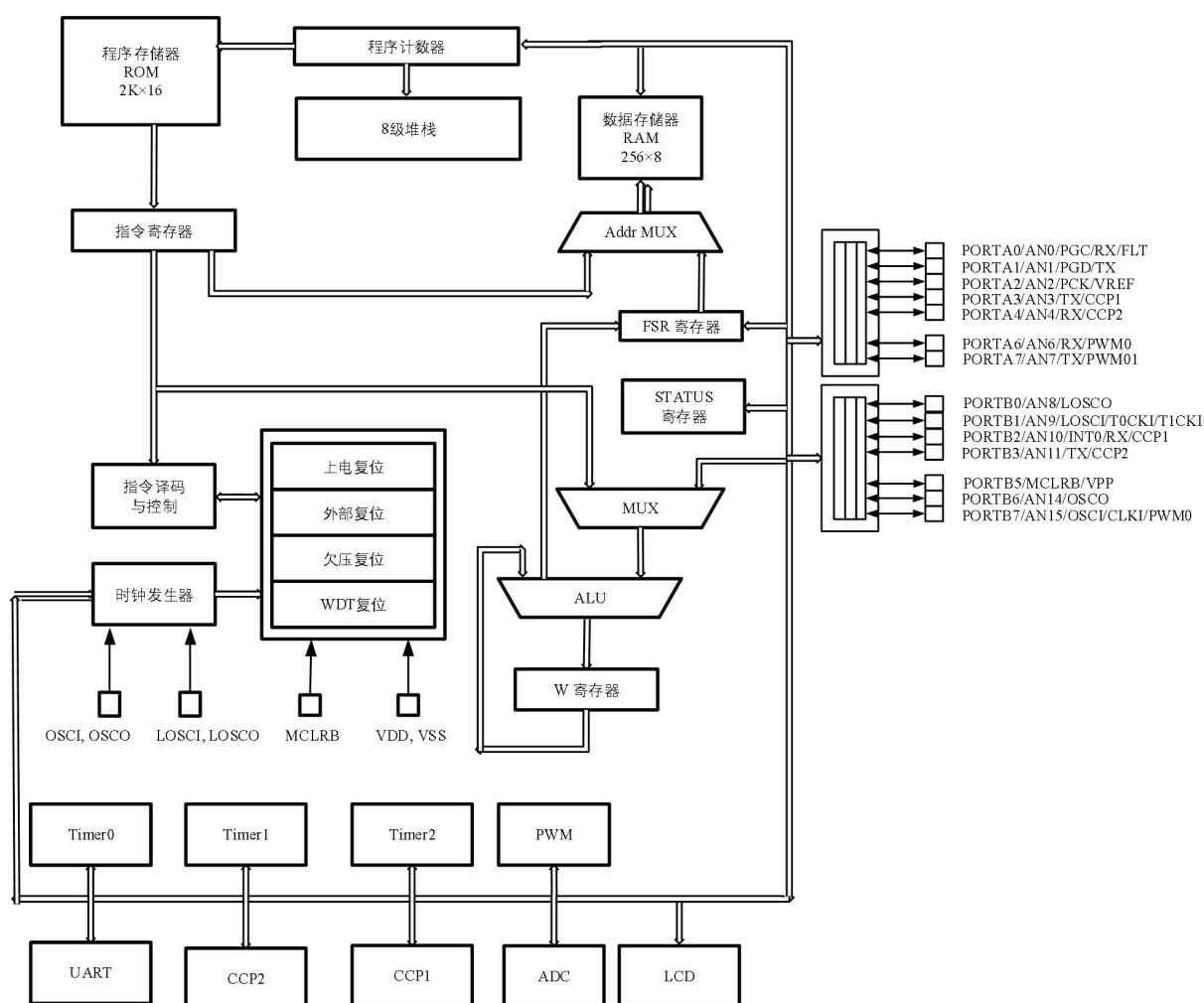
✓ 選型表

產品型號	ROM	RAM	堆疊	定時器	I/O	CCP	Wake up port	ADC	封裝形式
HC16P122A1	1K*16	256*8	8	3	5+1	1	6	12*(5+1)	SOP8
HC16P122B1	2K*16	256*8	8	3	13+1	2	14	12*(13+1)	SOP16
HC16P122B1	2K*16	256*8	8	3	13+1	2	14	12*(13+1)	DFN16

使用注意事項:

ADCON1 寄存器的低四位 Bit3~Bit0 如果全為 0，功耗會比一般情況偏大。在 sleep 模式下為保證系統的低功耗，請避免將 ADCON1 寄存器的低四位 Bit3~Bit0 全設置為 0。

1.2 系統框圖



1.3 引腳圖

HC16P122B1_16PIN		
MCLR _B /VPP/PORT _B 5	1	16
CCP2/PORT _B 3/AN11/TX	2	15
CCP1/INT0/PORT _B 2/AN10/RX	3	14
T0CKI/T1CKI/LOSCI/PORT _B 1/AN9	4	13
LOSCO/PORT _B 0/AN8	5	12
PORTA7/AN7/TX/PWM01	6	11
PORTA6/AN6/RX/PWM0	7	10
GND	8	9
		OSCO/PORT _B 6/AN14/PWM01
		OSCI/CLKI/PORT _B 7/AN15/PWM0
		VDD
		PORTA0/AN0/PGC/RX/FLT
		PORTA1/AN1/PGD/TX
		PORTA2/AN2/PCK/V _{REF}
		PORTA3/AN3/TX/CCP1
		PORTA4/AN4/RX/CCP2

HC16P122A1_8PIN		
VDD	1	8
OSCO/PWM01/AN14/PORT _B 6	2	7
MCLR _B /VPP/PORT _B 5	3	6
CCP2/RX/AN4/PORTA4	4	5
		VSS
		PORTA0/AN0/PGC/RX/FLT
		PORTA1/AN1/PGD/TX
		PORTA2/AN2/PCK/V _{REF}

1.4 引腳說明

16PIN 腳位	名稱	類型	說明
1	PORT _B 5 MCLR _B VPP	I I P	輸入/輸出，帶可編程上下拉電阻，端口電平變化中斷。 複位輸入，低電平有效。 編程高壓電源輸入。
2	PORT _B 3 AN11 CCP2 TX	I/O AN I/O O	輸入/輸出，帶可編程上下拉電阻，端口電平變化中斷。 ADC通道11輸入。 CCP2輸入/輸出。 串口數據發送端。
3	PORT _B 2 CCP1 AN10 INT0 RX	I/O I/O AN I I	輸入/輸出，帶可編程上拉電阻，端口電平變化中斷。 CCP1輸入/輸出。 ADC通道10輸入。 外部中斷接收端。 串口數據接收端。
4	PORT _B 1 T0CKI T1CKI LOSCI AN9	I/O I I AN AN	輸入/輸出，帶可編程上拉電阻，端口電平變化中斷。 Timer0外部時鐘輸入（施密特觸發器）。 Timer1外部時鐘輸入（施密特觸發器）。 低頻晶體振盪器輸入。 ADC通道9輸入。

5	PORTB0 AN8 LOSCO	I/O I AN	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 ADC通道8輸入口。 低頻晶體振盪器輸出口。
6	PORTA7 AN7 TX PWM01	I/O AN O O	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 ADC通道7輸入口。 串口數據發送端口。 PWM01輸出口。
7	PORTA6 AN6 RX PWM0	I/O AN I O	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 ADC通道6輸入口。 串口數據接收端口。 PWM0輸出口。
8	GND	P	電源地。
9	PORTA4 AN4 RX CCP2	I/O AN I I/O	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 ADC通道4輸入口。 串口數據接收端口。 CCP2輸入/輸出口。
10	PORTA3 AN3 TX CCP1	I/O AN O I/O	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 ADC通道3輸入口。 串口數據發送端口。 CCP1輸入/輸出口。
11	PORTA2 AN2 PCK VREF	I/O AN O P	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 ADC通道2輸入口。 內部高頻 RC 振盪頻率輸出口。 外部參考電壓輸入口。
12	PORTA1 PGD AN1 TX	I/O I/O AN O	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 編程數據輸入/輸出口。 ADC通道1輸入口。 串口數據發送端口。
13	PORTA0 AN0 PGC RX FLT	I/O I I/O I I	輸入/輸出口，帶可編程上拉電阻，端口電平變化中斷。 ADC通道1輸入口。 編程數據輸入/輸出口。 串口數據接收端口。 PWM 外部故障檢測端口。
14	VDD	P	電源輸入。
15	PORTB7 AN15 OSCI PWM0 CLKI	I/O AN AN O I	輸入/輸出口，帶可編程上下拉電阻，端口電平變化中斷。 ADC通道15輸入口。 高頻晶體振盪器輸入口。 PWM0輸出口。 時鐘輸入口。
16	PORTB6 OSCO PWM01 AN14	I/O AN O AN	輸入/輸出口，帶可編程上下拉電阻，端口電平變化中斷。 高頻晶體振盪器輸出口。 PWM01輸出口。 ADC通道14輸入口。

注： I = 輸入 O = 輸出 I/O = 輸入/ 輸出 P = 電源 AN = 模擬輸入輸出

1.5 引腳電路

圖 1-1: PORTA[7:6] [4:3] [1:0]的等效電路

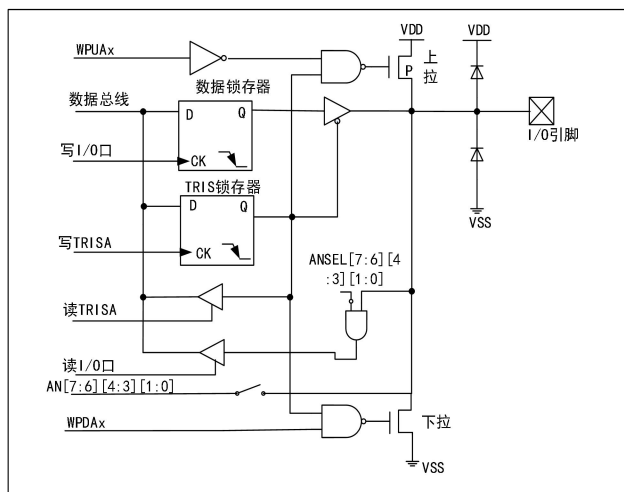


圖1-2: PORTA2的等效電路

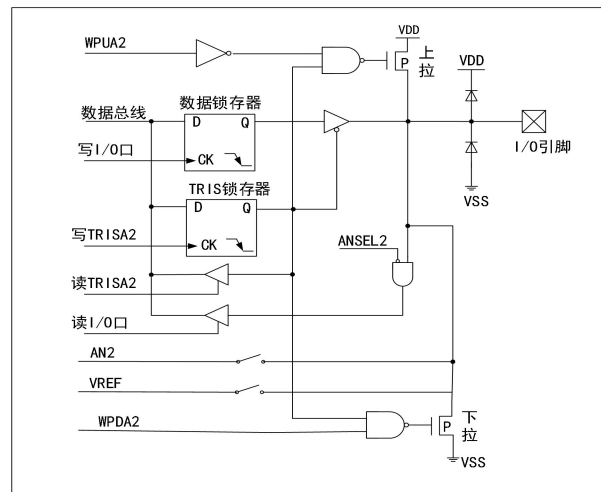


圖1-3: PORTB[7:6] [3:0]的等效電路

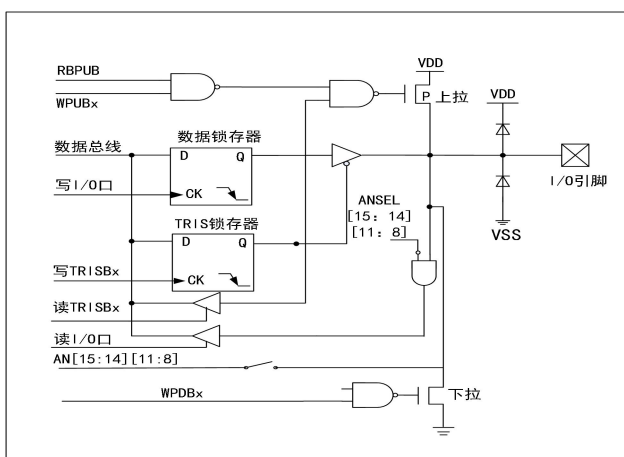
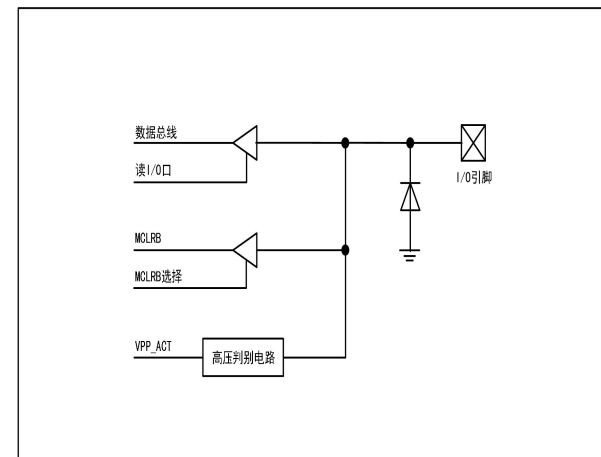


圖1-4: PORTB5口的等效電路



2 中央處理器（CPU）

HC16P122A1/B1 CPU內核包括：

- 2T/4T 時鐘模式
- 8 級堆疊
- 程式記憶體
- 尋址方式
- 數據儲存器

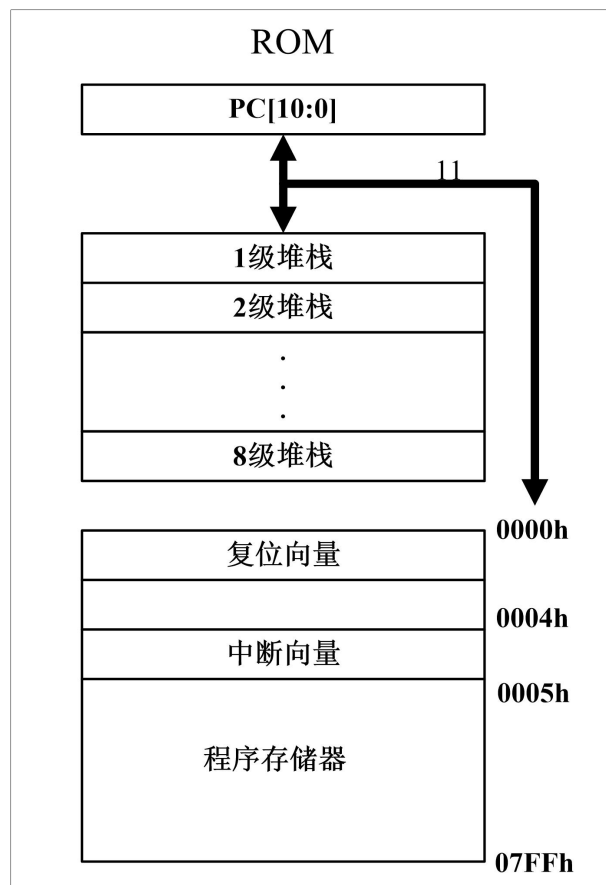
2.1 記憶體

2.1.1 程式記憶體（ROM）

HC16P122A1/B1具有2K×16位的程式記憶體，圖3-1給出了程式記憶體的映射。訪問超出物理地址以外的單元時，會導致返回到地址最低單元。

複位向量是0000H，中斷向量是0004H。

圖2-1 程式記憶體映射和堆疊



2.1.1.1 複位向量（0000H）

- 上電複位（POR=0，BOR=X，TO=1）
- 低電壓複位（POR=1，BOR=0，TO=1）
- 看門狗複位（POR=1，BOR=1，TO=0）
- 外部複位（POR=1，BOR=1，TO=1）

發生上述任一種複位後，程式將從0000H 處重新開始執行，系統寄存器也都將恢復為默認值。根據PCON寄存器中的POR，BOR標誌及STATUS 寄存器中的TO標誌位的內容可以判斷系統複位方式。下麵一段程式演示了如何定義ROM 中的複位向量。

➤ 例：定義複位向量

```

                ORG          0000H          ;複位向量
                GOTO         MAIN          ;跳轉到用戶程式
                ...
MAIN:           ORG          400H          ;用戶程式起始
                ...
                ...
                END              ;用戶程式結束

```

➤ 例：複位源判斷

```

                ORG          0000H
                GOTO         RST_JUGE
                ...
RST_JUGE:
                BTFSS        PCON,POR
                GOTO         ISPOR          ;POR 標誌為0，判定為上電複位
                BTFSS        PCON,BOR
                GOTO         ISBOR          ;POR=1,BOR=0,判定為低電壓複位
                BTFSS        STATUS,TO
                GOTO         ISWDTR         ;POR=1,BOR=1,TO=0，判定為WDT複位
EXT_RST:       ...                        ;POR=1,BOR=1,TO=1,判定為外部複位
                ...
ISPOR:         BSF           PCON,POR      ;上電複位處理程式
                ...
ISBOR:         BSF           PCON,BOR      ;低電壓複位處理程式
                ...
ISWDTR:        CLRWDT              ;TO標誌置1,WDT複位處理程式
                ...                  ;其他程式，注意處理BANK

```

2.1.1.2 中斷向量 (0004H)

中斷向量地址為0004H。一旦有中斷回應，程式計數器PC 的當前值就會存入堆疊緩存器並跳轉到0004H開始執行中斷服務程式。中斷服務副程式中需根據程式需要對相應狀態寄存器進行適當的中斷點保護和恢復。下麵的示例程式說明了如何編寫中斷服務程式。

➤ 例：中斷副程式的編寫

```

                ORG          0000H
                GOTO         MAIN
                ORG          0004H
                GOTO         INT_SERVICE
MAIN:           ...
INT_SERVICE:
                MOVWF        W_TEMP          ;保存W
                SWAPF        STATUS,W
                MOVWF        STATUS_TEMP     ;保存STATUS
                MOVF         PCLATH,W
                MOVWF        PCLATH_TEMP     ;保存PCLATH

```

```

...
MOVF      PCLATH_TEMP,W
MOVWF     PCLATH      ;恢復PCLATH
SWAPF     STATUS_TEMP,W
MOVWF     STATUS      ;恢復STATUS
SWAPF     W_TEMP,F
SWAPF     W_TEMP,W    ;恢復W
RETFIE    ;退出中斷
...
END

```

對於編寫中斷服務程式，需要以下幾個要點需注意：

1. 中斷入口地址為 0X04，回應中斷後，程式自動跳轉到 0X04 開始執行
2. 中斷服務程式需首先對相應的寄存器進行保護。
3. 保存系統寄存器時使用到的 RAM 建議定義在所有 BANK 均映射的位置。
4. 中斷服務副程式返回前對保護的寄存器進行恢復，注意恢復順序，對 W 必須使用 SWAPF。
5. 程式中使能兩個以上的中斷源時，程式需對發生中斷的中斷源進行判斷，從而執行相應的服務程式。
6. 需要軟體清空對應的中斷標誌。
7. RETFIE 指令將自動使能 GIE，請勿在中斷服務副程式中用其他指令使能 GIE，以免造成中斷回應混亂

2.1.1.3 查表

方式一：

利用 ADDWF PCL, F 和 RETLW 指令實現數據表，因為以 PCL 為目的運算元的運算將改變程式指針(PC)值，其具體操作為 PC 的低 8 位為 ALU 的運算結果，PC 的高 3 位將從 PC 高位緩衝器 PCLATH 中獲得。 如下是數據表實現的一個例子。

➤ 例：數據查表

```

...
MOVLW     HIGH TAB1    ;獲得數據表地址高8位（內部宏指令）
MOVWF     PCLATH        ;表地址高位賦給PCLATH
MOVF      TABBUF,W      ;獲得表數據偏移量，調用前賦值。
CALL      TAB1          ;調用數據表
...
ORG       100H

```

TAB1:

```

ADDWF     PCL,F         ;表頭運算
RETLW     DATA0_TAB1   ;W=0對應數據
RETLW     DATA1_TAB1   ;W=1對應數據
RETLW     DATA2_TAB1   ;W=2對應數據
...
RETLW     DATAFE_TAB1  ;W=0XFE對應數據

```

對於數據查表的編程，需注意：

1. 數據表寬度：8 位
2. 當 PCL 與 W 的加運算有進位時，進位將被捨棄，數據表溢出，將造成查表混亂；故表頭儘量放在數據頁前端，以免數據表溢出。
3. TABBUF 的值不得大於表長，否則將造成運行混亂。

➤ 例：跳轉表

跳轉表能夠實現多地址跳轉功能。由於 PCL 和 W 的值相加即可得到新的 PCL，同時 PCH 從 PCLATH 中載入，因此，可以通過對 PCL 加上不同的 W 值來實現多地址跳轉，可參考以下範例。

```

...
ORG          0100H
MOVLW       HIGH   TAB2    ;獲得跳轉表地址高位（內部宏指令）
MOVWF       PCLATH
MOVF        TABBUF,W
TAB2:        ADDWF   PCL,F
GOTO        LABEL0_TAB2    ;TABBUF =0,跳轉 LABEL0_TAB2
GOTO        LABEL1_TAB2    ;以下類推
GOTO        LABEL2_TAB2
GOTO        LABEL3_TAB2

```

注：

如上跳轉表，有 4 個跳轉分支，TABBUF 的合法範圍為 0X00~0X03

方式二：

可以通過以下5個特殊功能寄存器對ROM區中的數據進行查找：

```

PMCON
PMDATL
PMDATH
PMADRL
PMADRH

```

寄存器 PMADRH 指向 ROM 區數據地址的高位元組（Bit8~Bit15），寄存器 PMADRL 指向 ROM 區數據地址的低位元組（Bit0~Bit7）。將 PMCON 寄存器的 RDON 位置 1 啟動讀操作，使用兩條指令來讀數據，RDON 位置 1 後的二條指令被自動忽略，建議用戶 RDON 位置 1 後的兩條指令為 NOP。執行完讀操作後，所查找的數據保存在 PMDATLH:PMDATL 寄存器。

➤ 例：查找ROM 地址為“TABLE”的值

```

MOVF        TABLE_ADDR_H,W
MOVWF       PMADRH    ;設置TABLE地址高位元組
MOVF        TABLE_ADDR_L,W
MOVWF       PMADRL    ;設置TABLE地址低位元組
BSF         PMCON,RDON ;開始讀
NOP

```

```

NOP                                ;等待兩條指令
MOVF    PMDATL, W
MOVWF   TABLE_DATA_L            ;TABLE_DATA_L= TABLE地址數據低位元組
MOVF    PMDATH, W
MOVWF   TABLE_DATA_H            ;TABLE_DATA_H= TABLE地址數據高位元組
...
...
TABLE   DW    1234H                ;定義數據表（16 位）數據。
        DW    F178H
        DW    2123H
    
```

PMCON

9Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMCON	-	-	-	-	-	-	-	RDON
R/W	-	-	-	-	-	-	-	R/W
POR的值	-	-	-	-	-	-	-	0

Bit [0] **RDON**: 讀控制位

0 = 不啟動ROM記憶體讀操作

1 = 啟動ROM讀操作（由硬體清零RDON；軟體只能將RDON位置1，但不能清零）

PMDATL

9Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMDATL	PMD7	PMD6	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
R/W	R/W	R/W	- R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

PMDATH

9Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMDATH	PMD15	PMD14	PMD13	PMD12	PMD11	PMD10	PMD9	PMD8
R/W	- R/W	- R/W	- R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

Bit [15:0] **PMDx[15:0]**: ROM記憶體讀操作後，PMADRH:PMADRL 指向地址的數據

9Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMADRL	PMA7	PMA6	PMA5	PMA4	PMA3	PMA2	PMA1	PMA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

9Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMADRH	PMA15	PMA14	PMA13	PMA12	PMA11	PMA10	PMA9	PMA8
R/W	- R/W	- R/W	- R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

Bit [15:0] **PMAx[15:0]**: ROM記憶體地址

2.2 晶片配置選擇表

2.2.1 晶片配置字

HC16P122A1/B1提供的配置字可以對多個系統模組做配置選擇，詳細配置選擇見下麵表格。系統還提供了一個4*16Bit的器件ID供用戶存儲校驗或其他代碼標識號。在程式運行過程中不能訪問這些存儲單元，但可在編程燒錄/校驗時對它們進行讀寫。

複位時間選擇	18ms
	4.5ms
施密特選擇	非施密特
	施密特
時鐘模式	4T
	2T
啟動時鐘選擇	高頻時鐘啟動
	低頻時鐘啟動
ADC 使能位	使能 ADC
	禁止 ADC
低頻系統時鐘選擇	內部低頻 RC
	低頻晶體振盪器 LOSCI/LOSCO 為低頻晶體振盪器輸入/輸出口。
	系統僅工作於高頻時鐘（TMR0 振盪器選擇 WDT 振盪器）
	系統僅工作於高頻時鐘（TMR0 振盪器選擇外部 32K）
內部 RC 選擇	內部 RC 振盪器頻率選擇 32MHz
	內部 RC 振盪器頻率選擇 16MHz
	內部 RC 振盪器頻率選擇 8MHz
	內部 RC 振盪器頻率選擇 4MHz
	內部 RC 振盪器頻率選擇 2MHz
	內部 RC 振盪器頻率選擇 1MHz
	內部 RC 振盪器頻率選擇 500KHz
	內部 RC 振盪器頻率選擇 62.5KHz
加密功能使能位	不加密
	加密
外部複位使能位	使能外部複位
	遮罩外部複位做輸入
WDT 使能位	使能 WDT
	禁止 WDT
高頻系統時鐘選擇	高頻晶體振盪器
	內部 RC
複位點選擇位	BOR2.4V 當系統電壓低於 2.4V 時，系統複位
	BOR2.0V 當系統電壓低於 2.0V 時，系統複位
	BOR1.5V 當系統電壓低於 1.5V 時，系統複位
	BOR3.6V 當系統電壓低於 3.6V 時，系統複位
	BOR2.7V 當系統電壓低於 2.7V 時，系統複位
	BOR2.0V 當系統電壓低於 2.0V 時，系統複位
	BOR2.2V 當系統電壓低於 2.2V 時，系統複位
	BOR1.8V 當系統電壓低於 1.8V 時，系統複位

選擇晶片配置字注意事項：

1. 在系統允許的情況下，儘量選用較低系統時鐘頻率，有利於降低系統功耗和提升系統電磁相容性
2. 時鐘模式選擇為 2T 時，PWM 模組的最大解析度降低到 9 位
3. 強干擾情況下，建議開啟 WDT 功能

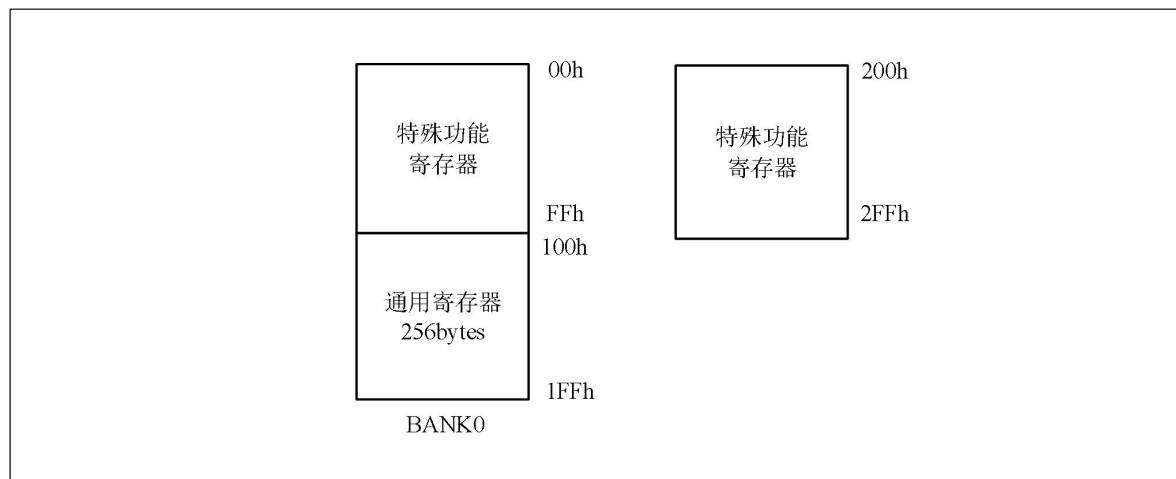
2.3 數據寄存器（RAM）

HC16P122A1/B1共有256個通用寄存器（GPR）和98個特殊功能寄存器（SFR），分在2個存儲區 Bank0~Bank1，每個存儲區的低256個地址單元保留為特殊功能寄存器，RP0是存儲區的選擇位。

CORE Register 00-09h&200-209h

00h&200h	INDF0	間接尋址 0 寄存器（不是實際存在的物理寄存器）							
01h&201h	INDF1	間接尋址 1 寄存器（不是實際存在的物理寄存器）							
02h&202h	PCL	程式計數器（PC）低位元組							
03h&203h	STATUS			RP0	TO	PD	Z	DC	C
04h&204h	FSR0L	間接尋址 0 地址低位指針							
05h&205h	FSR0H	間接尋址 0 地址高位指針							
06h&206h	FSR1L	間接尋址 1 地址低位指針							
07h&207h	FSR1H	間接尋址 1 地址高位指針							
08h&208h	PCLATH						程式計數器高 3 位		
09h&209h	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

圖2-2 數據記憶體映射



特殊功能寄存器列表如下圖：

地址	名稱	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	複位初值
BANK0										
010h	TRISA	TRISA7	TRISA6	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111
011h	TRISB	TRISB7	TRISB6	TRISB5	-	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111
01Ch	PORTA	PORTA7	PORTA6	-	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	0000 0000
01Dh	PORTB	PORTB7	PORTB6	PORTB5	-	PORTB3	PORTB2	PORTB1	PORTB0	0000 0000
028h	WPUA	WPUA7	WPUA6	-	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	1111 1111
029h	WPUB	WPUB7	WPUB6	WPUB5	-	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111
034h	WPDA	WPDA7	WPDA6	-	WPDA4	WPDA3	WPDA2	WPDA1	WPDA0	1111 1111
035h	WPDB	WPDB7	WPDB6	WPDB5	-	WPDB3	WPDB2	WPDB1	WPDB0	1111 1111
040h	IOCA	IOCA7	IOCA6	-	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	0000 0000
041h	IOCB	IOCB7	IOCB6	IOCB5	-	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000
04Ch	PORCTR	-	-	-	ONA3	CCPCT	PWMCT	UAPCT1	UAPCT0	--- 0000
04Dh	DRENAL	DRENA7L	DRENA6L	-	DRENA4L	DRENA3L	DRENA2L	DRENA1L	DRENA0L	1111 1111
04Eh	DRENB	DRENB7L	DRENB6L	DRENB5L	-	DRENB3L	DRENB2L	DRENB1L	DRENB0L	1111 1111
054h	PIR1	-	ADIF	-	-	-	CCP1IF	T2IF	T1IF	-0- -000
055h	PIR2	-	-	PWM0IF	-	RXIF	TXIF	-	CCP2IF	--0 00-0
056h	PIR3	-	-	-	-	-	-	-	RAIF	---0 0000
058h	T1L	Timer1 計數寄存器低位元組								XXXX XXXX
059h	T1H	Timer1 計數寄存器高位元組								XXXX XXXX
05Ah	T1CON	T1CS1	T1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	-	T1ON	0000 00-0
05Bh	T0	Timer0 計數寄存器								XXXX XXXX
05Ch	T2	Timer2 計數寄存器								XXXX XXXX
05Dh	PR2	Timer2 週期寄存器								0000 0000
05Eh	T2CON	-	T2CKPS3	T2CKPS2	T2CKPS1	T2CKPS0	T2ON	-	-	-000 00--
05Fh	PR1L	Timer1 週期寄存器低位元組								XXXX XXXX
060h	PR1CON	PWM1T1	PWM1T0	PWM2T1	PWM2T0	T1CKPS3	T1CKPS2	PWMPRI	PR1EN	0000 0000
070h	PIE1	-	ADIE	-	-	-	CCP1IE	T2IE	T1IE	0000 0000
071h	PIE2	-	-	PWM0IE	-	-	UARTIE	-	CCP2IE	--00 -000
072h	PIE3	-	-	-	-	-	-	-	RAIE	---0 0000
078h	OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	0000 0000
079h	PCON	LVD2EN	LVD1EN	-	WDTENS	LVD2F	LVD1F	POR	BOR	00-1 qqqq
07Ah	OSCCON	T0OSCEN	-	-	-	-	-	HXEN	SCS	0--- -0q
080h	CCPR2L	CCP2 寄存器低位元組								XXXX XXXX
081h	CCPR2H	CCP2 寄存器高位元組								XXXX XXXX
082h	CCP2CON	-	-	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000
083h	CCPR1L	CCP1 寄存器低位元組								0000 0000
084h	CCPR1H	CCP1 寄存器高位元組								0000 0000
085h	CCP1CON	-	-	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000
08Ch	ANSELL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0	1111 1111
08Dh	ANSELH	ANSEL15	ANSEL14	-	-	ANSEL11	ANSEL10	ANSEL9	ANSEL8	1111 1111
092h	ADRESL	ADC 結果寄存器低位元組								00000000

093h	ADRESH	ADC 結果寄存器高位元組								00000000
094h	ADCON0	-	-	CHS3	CHS2	CHS1	CHS0	ADON	ADEN	00000000
095h	ADCON1	ADFM	ADCS2	ADCS1	ADCS0	VHS2	VHS0	VHS0	ADREF	0000---0
096h	ADCLK	-	-	-	-	-	ADCLK2	ADCLK1	ADCLK0	---- -000
09Ah	PMDATL	程式記憶體讀數據寄存器的低位元組								0000 0000
09Bh	PMDATH	程式記憶體讀數據寄存器的高位元組								0000 0000
09Ch	PMADRL	程式記憶體讀地址寄存器的低位元組								0000 0000
09Dh	PMADRH	程式記憶體讀地址寄存器的高位元組								0000 0000
09Eh	PMCON	-	-	-	-	-	-	-	RDON	---- ---0
0A0h	DRENAH	DRENA7H	DRENA6H	DRENA5H	DRENA4H	DRENA3H	DRENA2H	DRENA1H	DRENA0H	1111 1111
0A1h	DRENBH	DRENB7H	DRENB6H	DRENB5H	DRENB4H	DRENB3H	DRENB2H	DRENB1H	DRENB0H	1111 1111

地址	名稱	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	複位初值
BANK1										
23Ah	BRT									0000 0000
23Bh	AUXR	-	UARTEN	UARTM0	BRTR	BRTX12	S1BR5	SMOD	SMOD0	0000 0000
23Ch	SCON	SM0/FE	SM1	SM2	REN	TR8	RB8	RXWK	-	0000 0000
23Dh	SBUF	串行口緩衝寄存器								0000 0000
23Eh	SADEN	從機地址寄存器								0000 -000
23Fh	SADDR	從機地址掩碼寄存器								00-0 0000
257h	PWM0DTL	DTL0.7	DTL0.6	DTL0.5	DTL0.4	DTL0.3	DTL0.2	DTL0.1	DTL0.0	0000 0000
258h	PWM0DTH					DTH0.3	DTH0.2	DTH0.1	DTH0.0	0000 0000
259h	PWM0DL	PD0.7	PD0.6	PD0.5	PD0.4	PD0.3	PD0.2	PD0.1	PD0.0	0000 0000
25Ah	PWM0DH	-	-	-	-	PD0.11	PD0.10	PD0.9	PD0.8	0000 0000
25Bh	PWM0PL	PP0.7	PP0.6	PP0.5	PP0.4	PP0.3	PP0.2	PP0.1	PP0.0	0000 0000
25Ch	PWM0PH	-	-	-	-	PP0.11	PP0.10	PP0.9	PP0.8	0000 0000
25Dh	PWM0C	-	-	FLTS	FLTC	PWM0S1	PWM0S0	CK01	CK00	0000 0000
25Eh	PWMEN	-	EFLT	-	-	EPWM01	-	-	EPWM0	0000 0000
25Fh	FLTM			-	-	-	-	FLT0M1	FLT0M0	--00 0000
260h	PWMM				PWM0M				RELOAD0	0000 0000
2B0h	LCDCON	LCDEN	RLCD1	RLCD0	FRAME	-	-	-	-	0000 ----
2B1h	COMAEN	COMAEN7	COMAEN6	COMAEN5	COMAEN4	COMAEN3	COMAEN2	COMAEN1	COMAEN0	0000 0000
2B2h	COMBEN	COMBEN7	COMBEN6	-	COMBEN4	COMBEN3	COMBEN2	COMBEN1	COMBEN0	00-0 0000

注： x = 未知， u = 不變， q = 取值視條件而定， - = 未實現

2.3.1 STATUS 寄存器

STATUS寄存器包含ALU的算術狀態、複位狀態和寄存器的存儲區選擇位。

03&203h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	-	-	RP0	TO	PD	Z	DC	C
R/W	-	-	R/W	R	R	R/W	R/W	R/W
POR的值	-	-	0	1	1	x	x	x

Bit [5] **RP0**: BANK選擇位

1 = Bank1

0 = Bank0

- Bit [4] **TO**: 超時位
1 = 上電、執行了CLRWDWT指令或SLEEP指令
0 = 發生了WDT溢出
- Bit [3] **PD**: 掉電位
1 = 上電或執行了CLRWDWT指令
0 = 執行了SLEEP指令
- Bit [2] **Z**: 結果為零位
1 = 算術或邏輯運算的結果為零
0 = 算術或邏輯運算的結果不為零
- Bit [1] **DC**: 半進位/借位位
1 = 加法運算時低四位有進位/減法運算時沒有向高四位借位
0 = 加法運算時低四位沒有進位/減法運算時有向高四位借位
- Bit [0] **C**: 進位/借位位
1 = 加法運算時有進位/減法運算時沒有借位發生/移位後移出邏輯1
0 = 加法運算時沒有進位/減法運算時有借位發生/移位後移出邏輯0

2.3.2 PC 寄存器

程式計數器（PC）為11位寬，低位元組來自可讀寫的PCL寄存器，高位元組（PC[10:8]）不可讀寫，可通過PCLATH 寄存器間接寫入。

02h&202h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

08h&208h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLATH	-	-	-	-	-	PCH10	PCH9	PCH8
R/W	-	-	-	-	-	R/W	R/W	R/W
POR的值	-	-	-	-	-	0	0	0

2.4 尋址模式

HC16P122A1/B1 共有三種尋址方式：立即尋址、直接尋址和間接尋址模式

2.4.1 立即尋址

立即數參與運算的尋址方式

➤ 例：立即尋址

ADDLW 06h ; W 的內容加 6，結果放入 W

2.4.2 直接尋址

寄存器參與運算的尋址方式

➤ 例：直接尋址

MOVWF OPTION ; W 的內容裝入 OPTION

2.4.3 間接尋址

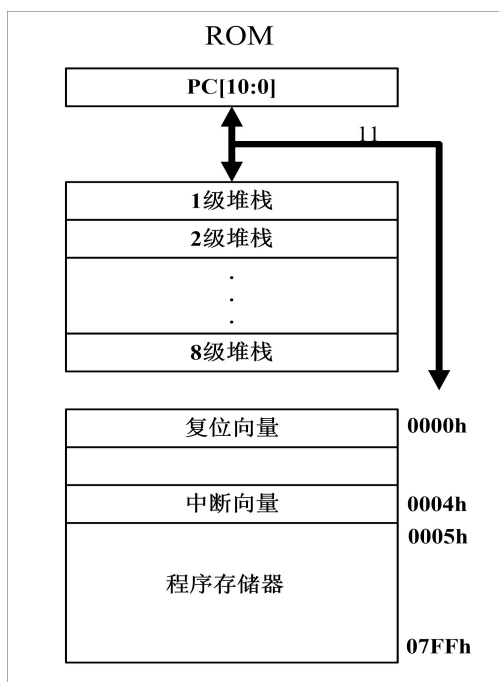
由指針 FSR 指向的寄存器參與運算的尋址方式。INDF 寄存器不是物理寄存器，對 INDF 寄存器操作可以實現間接尋址

➤ 例：利用間接尋址對 0X100~0X1FF 通用數據記憶體進行清零

	MOVLW	00h	;清零 0X100~0X1FF
	MOVWF	FSR0L	
	MOVLW	0x01	
	MOVWF	FSR0H	; FSR 指向 100h 地址
NEXTBYTE:	CLRF	INDF0	;對 FSR 指向的數據記憶體清零
	INCF	FSR0L,F	;FSR + 1,指向下一個地址
	MOVLW	00h	;注意這裏的邊界值為欲操作 RAM 最大地址
+ 1			
情況	XORWF	FSR0L,W	;利用間接尋址,注意意外指向特殊寄存器的
	BTFSS	STATUS,Z	
	GOTO	NEXTBYTE	;FSR 的值小於 1FFh,迴圈清零下一個地址
CONTINUE:	...		;完成清零操作

2.5 堆疊

HC16P122A1/B1 具有一個 8 級深度的硬體堆疊。當執行 CALL 指令或由於中斷導致程式跳轉時，PC 值會被壓入堆疊;當執行 RETURN、RETLW 或 RETFIE 指令時，PC 值從堆疊彈出。



注：

壓棧級數請勿超過 8 級，超過 8 級壓棧將導致堆疊溢出，溢出後堆疊指針迴圈，新的壓棧將覆蓋原堆疊內容。

3 複位

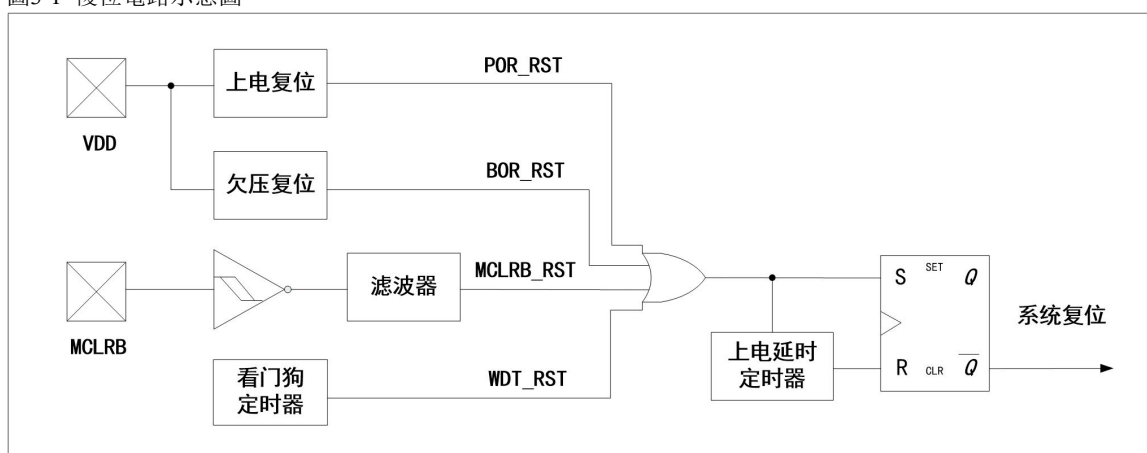
HC16P122A1/B1 共有四種複位方式：

- 上電複位（POR）
- 外部複位（MCLR Reset）
- 欠壓複位（BOR）
- 看門狗定時器複位（WDT Reset）

當上述任何一種複位產生時，系統進入複位狀態，所有的特殊功能寄存器被初始化，程式停止運行，同時程式計數器（PC）清零。經過上電延時定時器延時後，系統結束複位狀態，程式從 0000h 地址開始執行。

上電延時定時器在複位時提供一個 18 ms（典型值）的固定延時和一個振盪器起振穩定的時間延時。

圖3-1 複位電路示意圖



特殊功能寄存器複位狀態

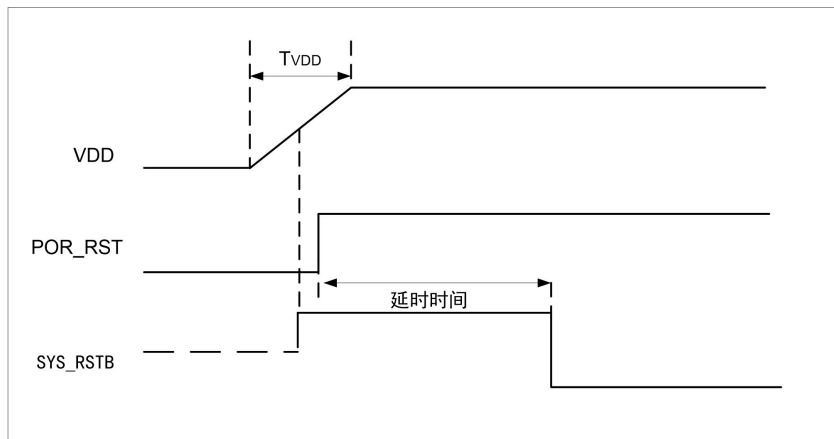
複位方式	STATUS 寄存器	PCON 寄存器
上電複位	0001 1xxx	00-1 qq00
正常工作模式下的外部複位	0001 1xxx	00-1 qq0u
休眠模式下的外部複位	0001 0uuu	00-1 qquu
欠壓複位	0001 0uuu	00-1 qq00
看門狗定時器複位	0000 1uuu	00-1 qquu

注：u = 不變，x = 未知，- = 未使用，q = 取值視條件而定

3.1 上電複位

系統上電過程中，VDD 達到系統正常工作電壓之前，上電複位電路產生內部複位信號。可通過查詢 PCON 寄存器來判斷是否發生上電複位。VDD 最大上升時間 T_{VDD} 必須滿足規格要求。任何一種複位方式都需要一定的回應時間，系統提供完善的複位流程以保證複位動作的順利進行。對於不同類型的振盪器，完成複位所需要的時間也不同。因此，VDD 的上升速度和不同晶振的起振時間都不固定。RC 震盪器的起振時間最短，晶體震盪器的起振時間則較長。在用戶的使用過程中，應考慮系統對上電複位時間的要求。

圖3-2 上電複位示意圖



3.2 外部複位

當外部複位端口 MCLRB 輸入一個持續時間超過 T_{MCLRB} 的低電平時，產生外部複位。MCLRB 選擇配置字為 1，MCLRB 口為外部複位輸入口。

圖3-3 外部複位示意圖

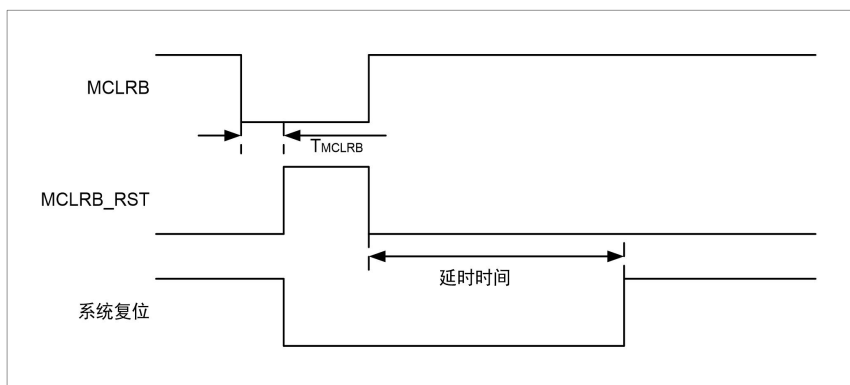
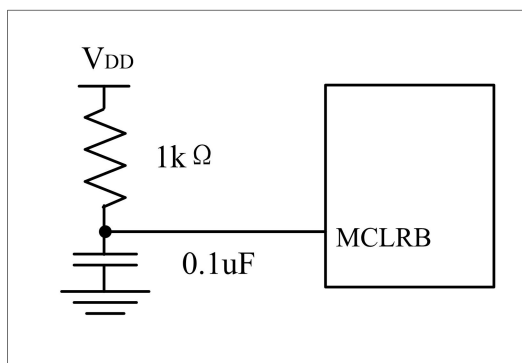


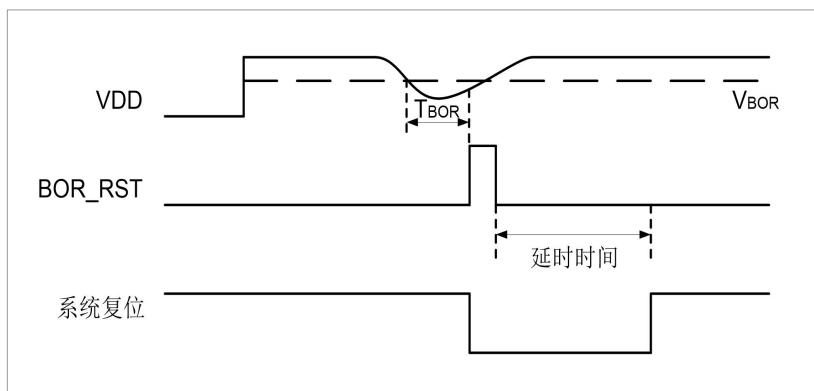
圖3-4 建議外部複位電路



3.3 欠壓複位

當VDD電壓下降到 V_{BOR} 以下，且持續時間超過 T_{BOR} 時，系統產生欠壓複位。

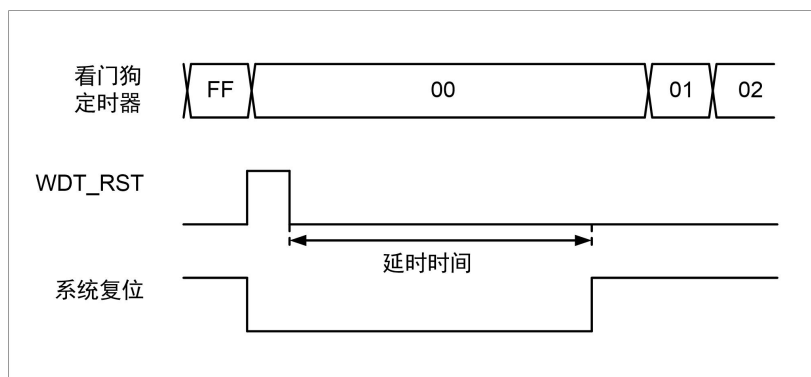
圖3-5 欠壓複位示意圖



3.4 看門狗定時器複位

在高頻和低頻模式下，看門狗定時器溢出會產生WDT複位；在綠色和休眠模式下，看門狗定時器溢出將喚醒SLEEP並使其返回高頻或低頻模式，程式從SLEEP指令下一條開始執行。WDT定時器配置字和WDTENS都為1時，才能使能看門狗定時器。

圖3-6 看門狗複位示意圖



3.5 PCON 寄存器

079h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	-	-	-	WDTENS	-	-	POR	BOR
R/W	-	-	-	R/W	-	-	R/W	R/W
POR的值	-	-	-	1	-	-	q	q

注：— = 未實現，q = 取值視條件而定

Bit [1] **POR**: 上電複位狀態位

1 = 非上電複位

0 = 上電複位（需要軟體置1）

Bit [0] **BOR**: 欠壓複位狀態位

1 = 未發生欠壓複位

0 = 發生了欠壓複位（需要軟體置1）

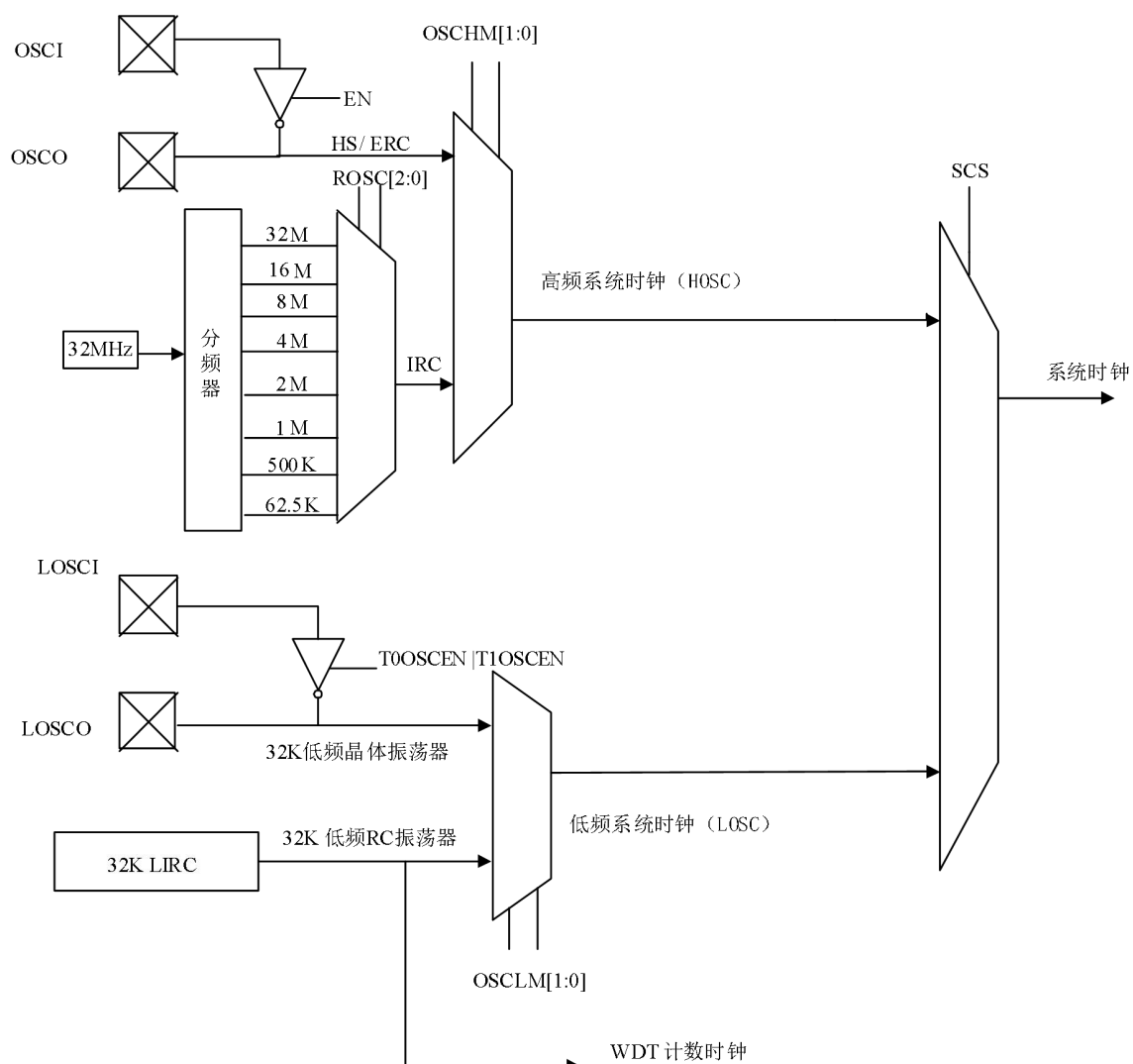
4 系統時鐘

4.1 概述

HC16P122A1/B1內帶雙時鐘系統：高頻時鐘和低頻時鐘。高頻時鐘的時鐘源由高頻晶振或內部32MHz RC 振盪電路（IRC 32MHz）提供。低頻時鐘的時鐘源則由低頻晶振或內部低速RC振盪電路（RC 32KHz@5V）提供。兩種時鐘都可作為系統時鐘源Fosc。OSCCON寄存器的SCS位控制高頻時鐘和低頻時鐘之間切換。

- 高頻模式：Fcpu = Fsys / N，N = 2或4，時鐘模式選擇決定N的值。
- 低頻模式：Fcpu = Fsys / N，N = 2或4，時鐘模式選擇決定N的值。

4.2 時鐘框圖



- OSCHM[1:0]: 高速系統時鐘選擇配置字
- OSCLM[1:0]: 低速系統時鐘選擇配置字
- ROSC[2:0]: 高速內部RC振盪器頻率選擇配置字

- Fosc: 時鐘源頻率
- Fsys: 系統時鐘頻率
- Fcpu: 指令時鐘頻率

4.3 系統高頻時鐘

系統高頻時鐘有三種選擇，通過OSCHM[1:0]高頻系統時鐘選擇配置字來控制。

高頻系統時鐘選擇配置字：

OSCHM[1:0]	說明
00	內部 RC 振盪器（IRC），OSCI/OSCO 作為輸入/輸出口
01	高頻晶體振盪器（HS），OSCI/OSCO 作為高頻晶體振盪器輸入/輸出口 外部時鐘輸入，OSCI 作為外部時鐘輸入口，OSCO 作為外部時鐘輸出口
10	保留
11	LP 模式

4.3.1 內部高頻 RC 振盪器

配置字OSCHM[1:0]和ROSC[2:0]控制單片機的內置RC高速時鐘。OSCHM[1:0]若選擇“00”，則內置RC振盪器作為系統時鐘源，OSCI/OSCO作為通用I/O口。

內置RC高頻時鐘有32M/16M/8M/4M/2M/1M /500K/62.5K 八種選擇。

高頻內部RC振盪器頻率選擇配置字

ROSC[2:0]	說明
111	內部RC振盪器頻率選擇32MHz
110	內部RC振盪器頻率選擇16MHz
101	內部RC振盪器頻率選擇8MHz
100	內部RC振盪器頻率選擇4MHz
011	內部RC振盪器頻率選擇2MHz
010	內部RC振盪器頻率選擇1MHz
001	內部RC振盪器頻率選擇500KHz
000	內部RC振盪器頻率選擇62.5KHz

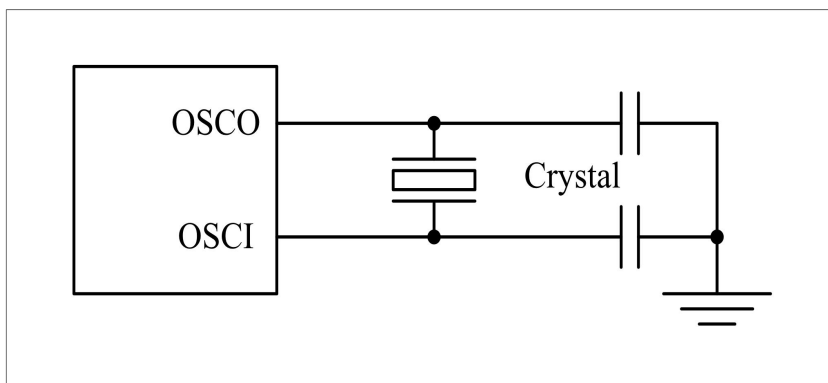
4.3.2 外部高頻時鐘

外部高頻時鐘共兩種模式，由配置字 OSCHM 控制具體模式的選擇

- 高頻晶體振盪器：最高 20MHz
- 外部時鐘輸入

4.3.2.1 高頻晶體振盪器

高頻晶體振盪器的頻率為4MHz~20MHz，推薦的典型值為4MHz、8MHz和16MHz，電容推薦值為20pF。



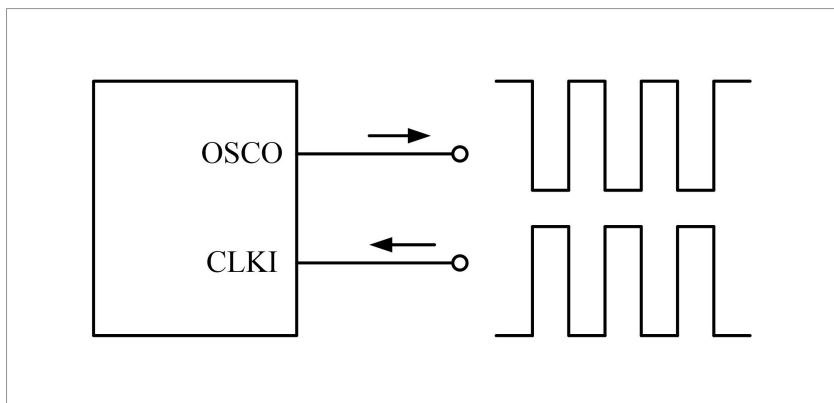
注：

OSCI 和 OSCO 引腳與振盪器和起振電容之間距離越近越好。

4.3.2.2 外部時鐘源

單片機可選擇外部時鐘信號作為系統時鐘，由配置字OSCHM控制，從OSCI 腳送入。

如果配置字OSCHM選擇為高頻晶體振盪器，外部時鐘由CLKI引腳輸入，OSCO引腳為CLKI的反向輸出。



注：

外部振盪电路中的 GND 必須盡可能的接近單片機的 VSS 端口。

4.4 系統低頻時鐘

高頻時鐘有兩種選擇，通過低頻時鐘選擇配置字來選擇。

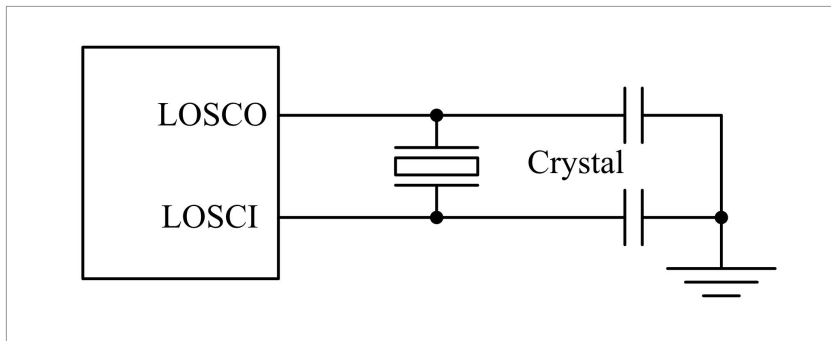
- 低頻晶體振盪器： 32.768KHz
- 低頻 RC 振盪器： 32K（5V 典型值）

低頻系統時鐘選擇配置字

OSCLM[1:0]	說明
00	低頻 RC 振盪器，32KHz，LOSCI/LOSCO 作為輸入/輸出口
01	低頻晶體振盪器，32.768KHz，LOSCI/LOSCO 作為低頻晶體振盪器輸入/輸出口
10	系統僅工作於高頻時鐘（TMR0 振盪器選擇 WDT 振盪器）
11	系統僅工作於高頻時鐘（TMR0 振盪器選擇外部 32K）

4.4.1 低頻晶體振盪器

低頻晶體振盪器的頻率為32.768KHz，電容推薦值為20pF。
低頻晶體振盪器電路



系統工作在綠色模式下，可以使能低頻晶體振盪器。

注：

外部高頻晶振接 OSCO、OSCI 端口，外部低頻晶振接 LOSCO、LOSCI 端口。

4.4.2 低頻 RC 振盪器

系統低頻時鐘源也可採用RC振盪電路。低頻RC振盪電路的輸出頻率受系統電壓和環境溫度的影響較大，通常為5V時輸出32KHZ（典型值）。

注：

低頻時鐘也用作看門狗定時器的時鐘。

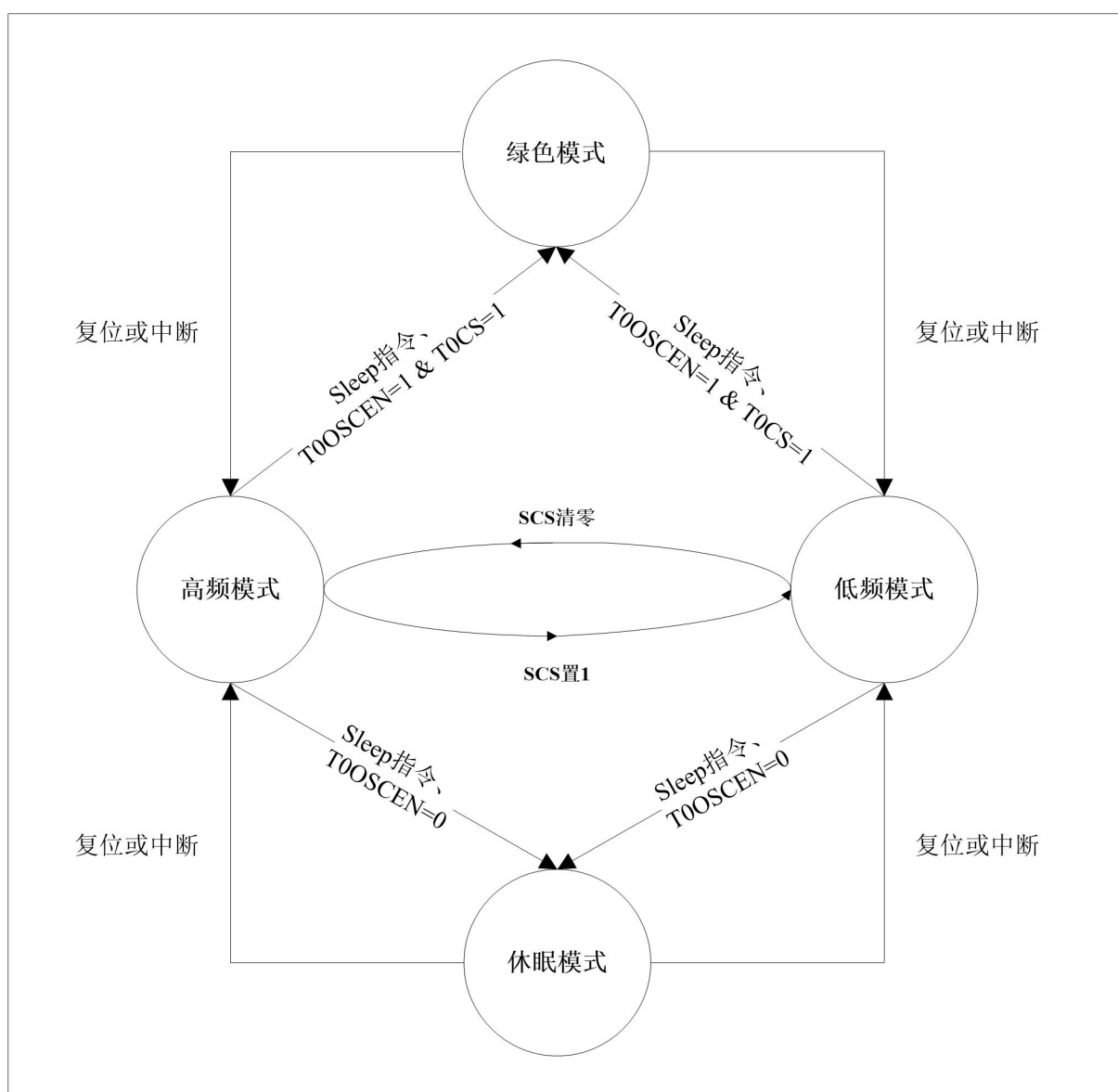
5 系統工作模式

HC16P122A1/B1共有四種工作模式：

- 高頻模式
- 低頻模式
- 休眠模式
- 綠色模式

系統複位後，工作於高頻模式還是低頻模式，由系統配置字決定。程式運行過程中，可以通過設置SCS位使系統在高頻和低頻模式之間切換。

圖5-1 系統工作模式轉換：



注：

1. 從休眠或綠色模式喚醒，中斷使能的情況則進入相應中斷，否則執行下一句。
2. 外部複位和 Timer2 中斷不能喚醒休眠或綠色模式。
3. 進入休眠或綠色模式前，關閉 WDT 可降低功耗。

各種模式下振盪器模組及Timer0/Timer1的工作狀態表

模組	高頻模式	低頻模式	綠色模式	休眠模式
高頻振盪器	運行	由HXEN決定	由HXEN決定	關閉
低頻振盪器	運行	運行	運行	關閉
Timer0	運行	運行	定時喚醒模式下運行	計數器模式下運行
Timer1	運行	運行	非同步定時喚醒模式下運行	非同步計數器模式下運行

5.1 模式切換舉例

- 例：高頻/低頻模式切換到睡眠模式。

```
BCF      STATUS,RP0      ;BANK0
BCF      OSCCON,T0OSCEN
SLEEP
```

- 例：高頻模式切換到低頻模式。

```
BCF      STATUS,RP0      ;BANK0
BSF      OSCCON,SCS      ;SCS = 1, 系統進入低頻模式
```

- 例：從低頻模式切換到高頻模式。

```
BCF      STATUS,RP0      ;BANK0
BCF      OSCCON,SCS      ;SCS = 0, 系統進入高頻模式
```

- 例：從高頻/低頻模式切換到綠色模式

;T0定時器定時喚醒

```
BCF      STATUS,RP0      ;BANK0
MOVLW    0X05
MOVWF    OPTION
BSF      OPTION,T0CS
BSF      OSCCON,T0OSCEN
BSF      INTCON,T0IE      ;使能T0 定時器。
BSF      INTCON,GIE
CLRF     T0
SLEEP
```

- 例：從高頻/低頻模式切換到綠色模式。

;T0定時器定時喚醒，OSCLM=01，低頻晶體振盪器為32,768KHz，定時喚醒時間為0.5s。

```
BCF      STATUS,RP0      ;BANK0
MOVLW    0X05
MOVWF    OPTION
```

```
BSF      OPTION,T0CS
BSF      OSCCON,T0OSCEN
BCF      OSCCON,T0IF
BSF      INTCON,T0IE      ;使能T0 定時器。
BSF      INTCON,GIE
CLRF     T0
```

RTC_MODE:

```
SLEEP
BCF      STATUS,RP0      ;BANK0
BCF      INTCON,T0IF     ;0.5s時間到
...
GOTO     RTC_MODE
```

5.2 高低頻模式切換

圖5-2 高低頻切換時序圖一

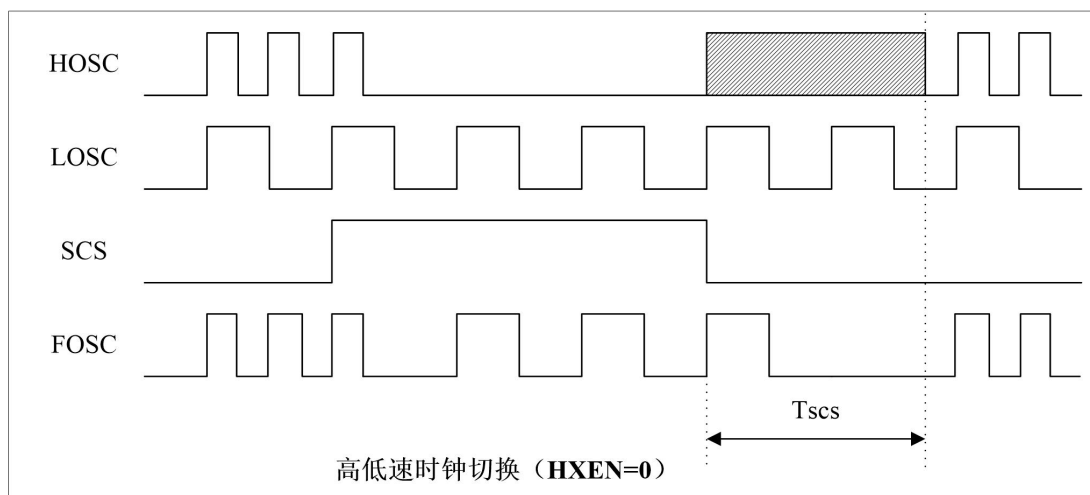
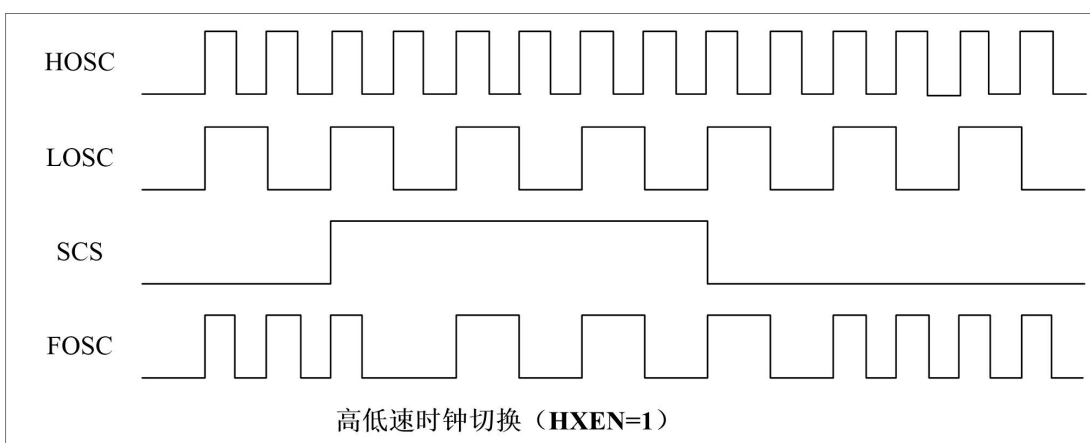


圖5-3 高低頻切換時序圖二



時鐘切換時間 (Tscs) 計算:

Tscs = 高頻振盪器起振時間 + 高頻振盪器穩定時間

不同類型高頻振盪器的穩定時間表

振盪器類型	高頻振盪器穩定時間
高頻晶體振盪器	1024 Clock
外部/內部 RC 振盪器	16 Clock

5.3 喚醒時間

系統進入休眠模式後，系統時鐘停止運行。外部中斷把系統從休眠模式下喚醒時，系統需要等待振盪器起振定時器（OST）定時結束，以使振盪電路進入穩定工作狀態，等待的這一段時間稱為喚醒時間。喚醒時間結束後，系統進入高頻或低頻模式。

喚醒時間的計算如下：

$$\text{喚醒時間} = \text{起振時間} + \text{OST定時時間}$$

同類型振盪器OST定時時間表

振盪器類型	OST 定時時間
高/低頻晶體振盪器	1024 Clock
外/內部 RC 振盪器	16 Clock
低頻 RC 振盪器	4 Clock

注：

系統進入綠色模式後，低頻時鐘正常運行。外部或內部中斷將系統從綠色模式中喚醒不需要喚醒時間。

5.4 OSCCON 寄存器

07Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCCON	T0OSCEN	-	-	-	-	-	HXEN	SCS
R/W	R/W	-	-	-	-	-	R/W	R/W
POR的值	0	-	-	-	-	-	0	q

注： x = 未知， - = 未實現， q = 取值視條件而定

Bit [1] **HXEN**： 高頻振盪器使能位

1 = 在低速或綠色模式下使能高頻振盪器

0 = 在低速或綠色模式下禁止高頻振盪器

Bit [0] **SCS**： 高低頻模式選擇位

1 = 系統時鐘選擇為低頻系統時鐘

0 = 系統時鐘選擇為高頻系統時鐘

6 中斷源

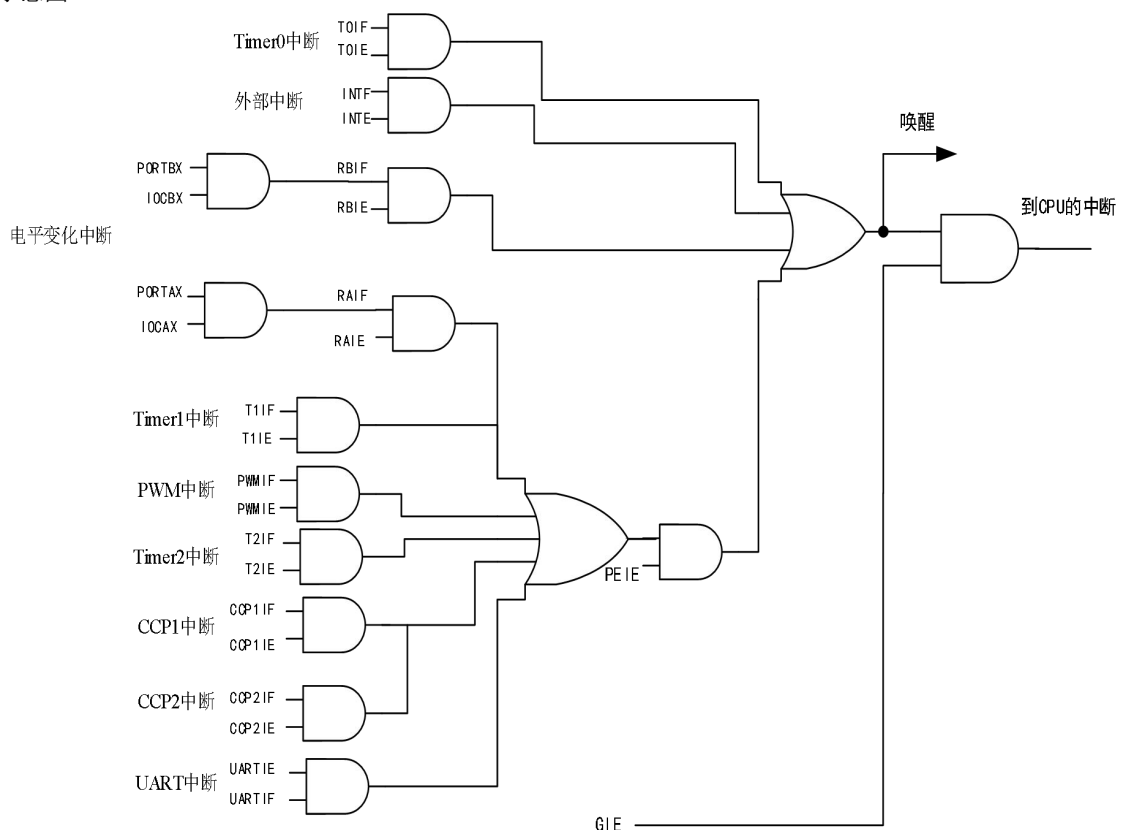
HC16P122A1/B1中斷源:

- Timer0定時器中斷
- INT0外部中斷
- PORT口電平變化中斷
- Timer1定時器中斷
- Timer2定時器中斷
- CCP1中斷
- CCP2中斷
- AD中斷
- UART中斷
- PWM中斷

系統產生中斷時，程式計數器（PC）值壓入堆疊，程式跳轉至0004h，進入中斷服務程式。當程式運行到RETFIE指令時，系統退出中斷服務程式，程式計數器值出棧，系統執行PC+1地址對應的指令。

為避免誤進入中斷，在使能中斷和退出中斷服務程式之前，必須清除中斷標誌位。

圖6-1 中斷示意圖



6.1 內核中斷

使能內核中斷必須將GIE和相應中斷的使能位置1，使能PORTB電平變化中斷還需要將相應端口配置為輸入並且IOCB的相應位置1。INT0外部中斷和PORTB電平變化中斷可以喚醒SLEEP，Timer0中斷在

計數器模式和定時喚醒模式下可以喚醒SLEEP。

09h、209h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	x

注： x = 未知

Bit [7] **GIE**: 全局中斷使能位

1 = 使能所有未遮罩的中斷

0 = 禁止所有中斷

Bit [5] **T0IE**: Timer0溢出中斷使能位

1 = 使能Timer0中斷

0 = 禁止Timer0中斷

Bit [4] **INTE**: INT0外部中斷使能位

1 = 使能INT0外部中斷

0 = 禁止INT0外部中斷

Bit [3] **RBIE**: PORTB電平變化中斷使能位

1 = 使能PORTB電平變化中斷

0 = 禁止PORTB電平變化中斷

Bit [2] **T0IF**: Timer0溢出中斷標誌位，Timer0計數寄存器在FFh至00h時產生溢出信號

1 = Timer0計數寄存器溢出（必須由軟體清0）

0 = Timer0計數寄存器未溢出

Bit [1] **INTF**: INT0外部中斷標誌位

1 = 發生INT0外部中斷（必須由軟體清0）

0 = 未發生INT0外部中斷

Bit [0] **RBIF**: PORTB電平變化中斷標誌位

1 = PORTB[7:0]中至少有一個口的電平狀態發生了改變（必須由軟體清0）

0 = PORTB[7:0]電平狀態沒有變化

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	-	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	1	1	1	1	1	1	1

Bit [6] **INTEDG**: 觸發INT0外部中斷的邊沿選擇位

1 = INT0引腳上升沿觸發中斷

0 = INT0引腳下降沿觸發中斷

6.2 外設中斷

使能外設中斷必須將GIE和PEIE置1，同時將相應中斷的使能位置1。Timer1中斷在非同步計數器模式和非同步定時喚醒模式下可以喚醒SLEEP，CCPx中斷在捕捉模式下可以喚醒SLEEP。

09h、209h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	x

Bit [7] **GIE**: 全局中斷使能位

1 = 使能所有未遮罩的中斷

0 = 禁止所有中斷

Bit [6] **PEIE**: 外設中斷使能位

1 = 使能所有未遮罩的外設中斷

0 = 禁止所有外設中斷

070h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE1	-	ADIE	-	-	-	CCP1IE	T2IE	T1IE
R/W	-	R/W	-	-	-	R/W	R/W	R/W
POR的值	-	0	-	-	-	0	0	0

Bit [6] **ADIE**: ADC中斷使能位

1 = 使能ADC中斷

0 = 禁止ADC中斷

Bit [2] **CCP1IE**: CCP1中斷使能位

1 = 使能CCP1中斷

0 = 禁止CCP1中斷

Bit [1] **T2IE**: Timer2計數寄存器與PR2匹配中斷使能位

1 = 使能Timer2匹配中斷

0 = 禁止Timer2匹配中斷

Bit [0] **T1IE**: Timer1溢出中斷使能位

1 = 使能Timer1溢出中斷

0 = 禁止Timer1溢出中斷

071h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE2	-	-	PWM0IE	-	-	UARTIE	-	CCP2IE
R/W	-	-	R/W	-	-	R/W	-	R/W
POR的值	-	-	0	-	-	0	-	0

Bit [5] **PWM0IE**: PWM0中斷使能位

1 = 使能PWM0中斷

0 = 禁止PWM0中斷

Bit [2] **UARTIE**: UART中斷使能位

1 = 使能UART中斷

0 = 禁止UART中斷

Bit [0] **CCP2IE**: CCP2中斷使能位

1 = 使能CCP2中斷
0 = 禁止CCP2中斷

072h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE3	-	-	-	-	-	-	-	RAIE
R/W	-	-	-	-	-	-	-	R/W
POR的值	-	-	-	-	-	-	-	0

Bit [0] **RAIE**: PORTA電平變化中斷使能位

1 = 使能PORTA電平變化中斷
0 = 禁止PORTA電平變化中斷

054h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR1	-	ADIF	-	-	-	CCP1IF	T2IF	T1IF
R/W	-	R/W	-	-	-	R/W	R/W	R/W
POR的值	-	0	-	-	-	0	0	0

Bit [6] **ADIF**: AD中斷標誌位

1 = ADC轉換已完成（必須由軟體清0）
0 = ADC轉換未完成或尚未開始

Bit [2] **CCP1IF**: CCP1中斷標誌位

捕捉模式:

1 = 發生了捕捉事件（必須用軟體清零）
0 = 未發生捕捉事件

比較模式:

1 = 發生了比較事件（必須用軟體清零）
0 = 未發生比較事件

PWM 模式:

在此模式下未使用

Bit [1] **T2IF**: Timer2計數寄存器與PR2匹配中斷標誌位

1 = Timer2發生匹配（必須用軟體清零）
0 = Timer2未發生匹配

Bit [0] **T1IF**: Timer1溢出中斷標誌位，Timer1計數寄存器在FFFFh至0000h時產生溢出信號

1 = Timer1計數寄存器溢出（必須由軟體清0）
0 = Timer1計數寄存器未溢出

055h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR2	-	-	PWM0IF	-	RXIF	TXIF	-	CCP2IF
R/W	-	-	R/W	-	R/W	R/W	-	R/W
POR的值	-	-	0	-	0	0	-	0

Bit [5] **PWM0IF**: PWM0中斷標誌位

1 = PWM0中斷產生中斷（必須由軟體清0）
0 = PWM0中斷未產生中斷

Bit [3] **RXIF**: UART接收標誌位

1 = UART接收中斷產生中斷（必須由軟體清0）
0 = UART接收中斷未產生中斷

- Bit [2] **TXIF**: UART發送中斷標誌位
 1 = UART發送中斷產生中斷（必須由軟體清0）
 0 = UART發送中斷未產生中斷
- Bit[0] **CCP2IF**: CCP2中斷標誌位
 捕捉模式：
 1 = 發生了捕捉事件（必須用軟體清零）
 0 = 未發生捕捉事件
 比較模式：
 1 = 發生了比較事件（必須用軟體清零）
 0 = 未發生比較事件
 PWM 模式：
 在此模式下未使用

056h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR3	-	-	-	-	-	-	-	RAIF
R/W	-	-	-	-	-	-	-	R/W
POR的值	-	-	-	-	-	-	-	0

- Bit [0] **RAIF**: PORTx電平變化中斷標誌位
 1 = PORTA[7:0]中至少有一個口的電平狀態發生了改變（必須由軟體清0）
 0 = PORTA[7:0]電平狀態沒有變化

6.3 GIE 全局中斷

只有當全局中斷控制位GIE置“1”的時候程式才能回應中斷請求。一旦有中斷發生，程式計數器入棧，程式轉至中斷向量地址（ORG 0004H），堆疊層數加1。

例：設置全局中斷控制位（GIE）

BSF INTCON,GIE ;使能 GIE。

注：

在所有中斷中，GIE 都必須處於使能狀態。

6.4 中斷保護

有中斷請求發生並被回應後，程式轉至0004H執行中斷服務程式。

中斷服務程式開始執行時，需保存W寄存器、STATUS寄存器的內容；結束中斷服務程式時，恢復PCLATH寄存器、STATUS寄存器、W寄存器的數值，注意順序。

注：

在退出中斷時，由於需要先恢復 STATUS，再使用 MOVF 指令恢復 W，可能會改變 STATUS，因此必須使用 SWAPF 指令恢復 W。注意在中斷中共有兩句 SWAPF 指令。

➤ 例：對W、PCLATH 和STATUS 進行入棧保護。

ORG 0000H

```

        GOTO    START
        ORG     0004H
        GOTO    INT_SERVICE
        ORG     0010H

START:
    ...

INT_SERVICE:
    MOVWF    W_TEMP           ;保存 W。
    SWAPF    STATUS,W
    MOVWF    STATUS_TEMP     ;保存 STATUS。
    MOVF     PCLATH, W
    MOVWF    PCLATH_TEMP     ;保存 PCLATH。
    CLRF     STATUS          ;切換到BANK0
    ...
    MOVF     PCLATH_TEMP, W
    MOVWF    PCLATH          ;恢復PCLATH。
    SWAPF    STATUS_TEMP, W
    MOVWF    STATUS          ;恢復STATUS。
    SWAPF    W_TEMP, F
    SWAPF    W_TEMP, W       ;恢復W。
    RETFIE    ;退出中斷。
    ...
    END

```

6.5 Timer0 定時器中斷

T0 溢出時，無論 T0IE 處於何種狀態，T0IF 都會置“1”。若 T0IE 和 T0IF 都置“1”，且 GIE 使能，系統就會回應 TIMER0 的中斷；若 T0IE = 0，則無論 T0IF 是否置“1”，系統都不會回應 TIMER0 中斷。

6.6 INT0 外部中斷

INT0 被觸發，則無論 INTE 處於何種狀態，INTF 都會被置“1”。如果 INTF = 1 且 INTE = 1，GIE 使能，系統回應該中斷；如果 INTF = 1 而 INTE = 0，系統並不會執行中斷服務。在處理多中斷時尤其需要注意。

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

Bit [6] **INTEDG**: 觸發INT0外部中斷的邊沿選擇位

1 = INT0引腳上升沿觸發中斷

0 = INT0 引腳下降沿觸發中斷

➤ 例：INT0 中斷請求設置，電平觸發。

```
BSF    STATUS,RP0    ;BANK1
```

```
BSF      OPTION,INTEG      ;INT0 置為上升沿觸發。
BCF      STATUS,RP0        ;BANK0
BCF      INTCON,INTF       ;INT0 中斷請求標誌清零。
BSF      INTCON,INTE       ;使能INT0 中斷。
BSF      INTCON,GIE        ;使能GIE。
```

➤ 例：INT0 中斷。

```
ORG      0004H ;
GOTO     INT_SERVICE
```

INT_SERVICE:

```
...      ;保存STATUS、W 和PCLATH。
BCF      STATUS,RP0        ;BANK0
BTFSS    INTCON,INTF       ;檢測T0IF。
GOTO     EXIT_INT          ;T0IF = 0，退出中斷。
BCF      INTCON,INTF       ;T0IF 清零。
...      ;INT0 中斷服務程式。
...
```

EXIT_INT:

```
...      ;恢復STATUS、W和PCLATH。
RETFIE   ;退出中斷。
```

6.7 PORT 電平變化中斷

PORTx電平變化中斷時，則無論RBIE處於何種狀態，RxIF都會被置“1”。如果RxIF=1 且RxIE=1，GIE使能，系統回應該中斷；如果RxIF=1 而RxIE=0，系統並不會執行中斷服務。

電平變化中斷必須將PORTx端口設為輸入，並將寄存器IOCx對應位置“1”。

041h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCB	IOCB7	IOCB6	IOCB5	-	IOCB3	IOCB2	IOCB1	IOCB0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
POR的值	0	0	0	-	0	0	0	0

Bit [7:0] **IOCB[7:0]**: PORTBx電平變化中斷使能控制位。

0 = 該端口禁止電平變化中斷；

1 = 該端口使能電平變化中斷。

040h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCA	IOCA7	IOCA6	-	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	-	0	0	0	0	0

Bit [7:0] **IOCA[7:0]**: PORTA電平變化中斷使能控制位。

0 = 該端口禁止電平變化中斷；

1 = 該端口使能電平變化中斷。

➤ 例：PORTB1 電平變化中斷請求設置。

```
BCF      STATUS,RP0        ;BANK0
MOVLW    0X02
```

```

        IORWF    TRISB,F           ;PORTB1 端口為輸入。
        MOVLW    0X02
        IORWF    IOCB,F           ;使能PORTB1 端口為電平變化中斷。
        MOVF     PORTB,W          ;讀PORTB 口。
        BCF      INTCON,RBIF      ;PROTB 中斷請求標誌清零。
        BSF      INTCON,RBIE      ;使能PROTB 中斷。
        BSF      INTCON,GIE       ;使能GIE。
➤ 例：PORTB 中斷。
        ORG      0004H
        GOTO     INT_SERVICE
INT_SERVICE:
        ...
        BCF      STATUS,RP0       ;保存STATUS、W 和PCLATH。
        BTFS     INTCON,RBIF      ;BANK0
        GOTO     EXIT_INT         ;檢測RBIF。
        MOVF     PORTB,W          ;RBIF = 0，退出中斷。
        BCF      INTCON,RBIF      ;讀PORTB 端口
        ...                      ;RBIF 清零。
        ...                      ;PORTB 電平變化中斷服務程式。
EXIT_INT:
        ...
        RETFIE                   ;恢復STATUS、W 和PCLATH。
                                   ;退出中斷。

➤ 例：PORTB 中斷喚醒SLEEP。
        BCF      STATUS,RP0       ;BANK0
        MOVLW    0X02
        IORWF    TRISB,F           ;PORTB1 端口為輸入。
        MOVLW    0X02
        IORWF    IOCB,F           ;使能PORTB1 端口為電平變化中斷。
        MOVF     PORTB,W          ;讀PORTB 口。
        BCF      INTCON,RBIF      ;PROTB 中斷請求標誌清零。
        BSF      INTCON,RBIE      ;使能PROTB 中斷。
        SLEEP
        BCF      INTCON,RBIE
        MOVF     PORTB,W          ;讀PORTB 端口。
        ...                      ;其他程式。

```

注：

1. 如要允許 PORTx 口電平變化中斷必須將 IOCx 的對應端口的位置 1。
2. PORTx 電平變化中斷中，在清零 RxF 之前必須執行 PORTx 端口讀操作。

6.8 Timer2 定時器中斷

當T2的值和PR2的值相同時，TIMER2中斷被觸發，則無論T2IE 處於何種狀態，T2IF 都會被置“1”。如果T2IF=1 且T2IE=1，且PEIE、GIE均使能，系統回應該中斷；如果T2IF=1 而T2IE=0，系統並不會

執行中斷服務。

➤ 例：TIMER2 中斷請求設置

```
BCF      STATUS,RP0      ;BANK0
MOVLW    0XFF
MOVWF    PR2              ;設置T2 週期。
MOVLW    0X04
MOVWF    T2CON            ;設置分頻比。
CLRF     T2
BSF      PIE1,T2IE        ;使能TIMER2 中斷。
BSF      INTCON,GIE
BSF      T2CON,T2ON       ;使能TIMER2。
```

➤ 例：TIMER2 中斷。

```
ORG      0004H
GOTO     T2INT_SERVICE
```

T2INT_SERVICE:

```
...      ;保存STATUS、W 和PCLATH。
BCF      STATUS,RP0      ;BANK0
BTFSS    PIR1,T2IF        ;檢測T2IF。
GOTO     EXIT_INT         ;T2IF = 0，退出中斷。
BCF      PIR1,T2IF        ;T2IF 清零。
...      ;TIMER2 中斷服務程式。
...
```

EXIT_INT:

```
...      ;恢復STATUS、W和PCLATH。
RETFIE   ;退出中斷。
```

6.9 Timer1 中斷

T1溢出時，無論T1IE 處於何種狀態，T1IF都會置“1”。若T1IE 和T1IF 都置“1”，且PEIE、GIE均使能，系統就會回應TIMER1的中斷；若T1IE = 0，則無論T1IF 是否置“1”，系統都不會回應TIMER1中斷。

➤ 例：TIMER1工作於非同步計數模式，並中斷喚醒SLEEP

```
MOVWLW    0XA4
BCF      STATUS,RP0      ;BANK0
MOVWF     T1CON           ;T1時鐘源為T1CKI;分頻比為1:4;非同步計
數器模式

MOVLW     nnH
MOVWF     T1H
MOVLW     nnH
MOVWF     T1L             ;Timer1賦初值
MOVLW     0XC0
MOVWF     INTCON          ;使能外設中斷
BCF      STATUS,RP0      ;BANK0
BSF      PIE1,T1IE
```

```

BCF      STATUS,RP0
BCF      PIR1,T1IF
BSF      T1CON,T1ON
BSF      STATUS,RP0
BCF      OSCCON,T0SCEN    ;禁止低頻晶體振盪器
SLEEP                                ;進入SLEEP

```

T1INT_SERVICE:

```

...                                ;保存STATUS、W 和PCLATH。
BCF      STATUS,RP0              ;BANK0
BTFSS    PIR1,T1IF              ;檢測T1IF。
GOTO     EXIT_INT               ;T1IF = 0，退出中斷。
BCF      PIR1,T1IF              ;T1IF 清零。
...                                ;TIMER1 中斷服務程式。
...

```

EXIT_INT:

```

...                                ;恢復STATUS、W和PCLATH。
RETFIE                                ;退出中斷。

```

6.10 AD 中斷

當 ADC 完成，ADON 被硬體清零，無論 ADIE 處於何種狀態，與此同時 ADIF 被置“1”。若 ADIE、ADIF 為“1”，且 PEIE、GIE 均使能，系統就會相應 ADC 中斷；若 ADIE = 0，則無論 ADIF 是否置“1”，系統都不會回應 ADC 中斷。

6.11 CCP 中斷

當發生 CCPx 中斷時，無論 CCPxIE 處於何種狀態，CCPxIF 被置“1”。若 CCPxIE、CCPxIF 為“1”，且 PEIE、GIE 均使能，系統就會相應 CCPx 中斷；若 CCPxIE = 0，則無論 CCPxIF 是否置“1”，系統都不會回應 CCPx 中斷。

6.12 UART 中斷

當 UART 接收、發送完成後，無論 UARTIE 處於何種狀態，TXIF、RXIF 都被置“1”。若 UARTIE、TXIF、RXIF 為“1”，且 PEIE、GIE 均使能，系統就會相應 UART 中斷。

6.13 PWM 中斷

當 PWM0 週期計數器溢出，無論 PWM0IE 處於何種狀態，PWM0IF 都被置“1”。若 PWM0IE 為“1”，且 PEIE、GIE 均使能，系統就會相應產生 PWM 中斷。

6.14 多中斷操作

在同一時刻，系統中可能出現多個中斷請求。此時，用戶必須根據系統的要求對各中斷進行優先權的設置。中斷請求標誌IF由中斷事件觸發，當IF處於有效值“1”時，系統並不一定會回應該中斷。各中斷觸發事件如下表所示：

中斷	有效觸發
T0IF	T0溢出

INTF	由INTEDG控制
RBIF	PORTB電平變化
RAIF	PORTA電平變化
T2IF	T2的值和PR2相同
RXIF/TXIF	UART發生發送接收事件
PWM0IF	PWM0週期計數溢出中斷

多個中斷同時發生時，需要注意的是：首先，必須預先設定好各中斷的優先權；其次，利用IE和IF控制系統是否回應該中斷。在程式中，必須對中斷控制位和中斷請求標誌進行檢測。

➤ 例：多中斷條件下檢測中斷請求。

```

                ORG    0004H ;
                GOTO   INT_SERVICE

INT_SERVICE:
                ...                ;保存STATUS、W和PCLATH。
INT0CHK:
                ...                ;檢查是否有INT0 中斷請求。
                BCF    STATUS,RP0    ;BANK0
                BTFSS  INTCON,INTE    ;檢查是否使能INT0 中斷。
                GOTO   INT0CHK        ;跳到下一個中斷。
                BTFSC  INTCON,INTF    ;檢查是否有INT0 中斷請求。
                GOTO   INT0          ;進入INT0 中斷。
INTT0CHK:
                ...                ;檢查是否有T0 中斷請求。
                BTFSS  INTCON,T0IE    ;檢查是否使能T0 中斷。
                GOTO   INTT2CHK        ;跳到下一個中斷。
                BTFSC  INTCON,T0IF    ;檢查是否有T0 中斷請求。
                GOTO   INTT0          ;進入T0 中斷。
INTT2CHK:
                ...                ;檢查是否有T2 中斷請求。
                BCF    STATUS,RP0    ;BANK0
                BTFSS  PIE1,T2IE    ;檢查是否使能T2 中斷。
                GOTO   INTRBCHK        ;跳到下一個中斷。
                BCF    STATUS,RP0    ;BANK0
                BTFSC  PIR1,T2IF    ;檢查是否有T2 中斷請求。
                GOTO   INTT2          ;進入T2 中斷。
INTRBCHK:
                BTFSS  INTCON,RBIE    ;檢查是否使能PORTB 電平變化中斷。
                GOTO   INT_EXIT        ;跳到中斷結束。
                BTFSC  INTCON,RBIF    ;檢查是否有PORTB 電平變化中斷請求。
                GOTO   INTRB          ;進入PORTB 電平變化中斷。
INTT2:
                BCF    PIR1,T2IF
                .....
                GOTO   INT_EXIT

INT_EXIT:
                ...                ;恢復STATUS、W和PCLATH。
                RETFIE                ;退出中斷。

```

7 I/O口

HC16P122A1/B1共有兩組雙向端口：

- PORTA口
- PORTB口

7.1 I/O 口輸入輸出控制寄存器

010h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISA	TRISA7	TRISA6	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	-	1	1	1	1	1

011h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISB	TRISB7	TRISB6	TRISB5	-	TRISB3	TRISB2	TRISB1	TRISB0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
POR的值	1	1	1	-	1	1	1	1

注：PORTB5可配置為開漏輸出

TRISx [7:0]：PORTx[7:0]的輸入輸出控制位

1 =輸入狀態

0 =輸出狀態

08Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSEL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	-	1	1	1	1	1

08Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELH	ANSEL15	ANSEL14	-	-	ANSEL11	ANSEL10	ANSEL9	ANSEL8
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR的值	1	1	-	-	1	1	1	1

ANSEL [15:0]：A/D引腳數模控制位

1：模擬模式，作為模擬信號口，僅可作為AD通道的模擬輸入。

0：數字模式，作為數字輸入或輸出口。

注：

ANSEL 上電初始值為 B' 1111 1111'，即作為模擬輸入。無論是否應用到 AD，均需要在上電後，對 IO 操作之前按需配置，否則 IO 口可能無法受控於對應的端口寄存器，狀態將不確定。

ANSEL[4:0]對應 AN4~AN0（PA4~PA0），ANSEL[7:6]對應 AN7、AN6（PA7、PA6）

ANSEL[11:8]對應 AN11~AN8（PB3~PB0），ANSEL[15:14]對應 AN15、AN14（PB7、PB6）

7.2 I/O 口上拉控制寄存器

028~029h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPUx	WPUx7	WPUx6	WPUx5	WPUx4	WPUx3	WPUx2	WPUx1	WPUx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

x = A、B

WPUx [7:0]: PORTx[7:0]的上拉使能位

1 = 上拉禁止

0 = 上拉使能

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

Bit [7] **RBPUB**: PORTB上拉使能位

1 = PORTB上拉由WPUB決定

0 = 使能PORTB上拉(此時無論WPUB為何值PORTB都上拉)

注:

1. 注意此處上拉控制寄存器邏輯，0 為使能，1 為禁止。
2. 當端口設置為輸出時，上拉/下拉默認關閉(硬體關閉)

7.3 I/O 口下拉控制寄存器

034h~035h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPDx	WPDx7	WPDx6	WPDx5	WPDx4	WPDx3	WPDx2	WPDx1	WPDx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

x = A、B

WPDx [7:0]: PORTx[7:0]的上拉使能位

1 = 下拉禁止

0 = 下拉使能

注:

1. 注意此處下拉控制寄存器邏輯，0 為使能，1 為禁止；
2. 當端口設置為輸出時，下拉無效；
3. 當下拉打開時，上拉根據控制寄存器使能或禁止，即設置為輸入時可同時打開上下拉。

7.4 I/O 驅動控制寄存器

04Dh~04Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DRENxL	DRENN7L	DRENN6L	DRENN5L	DRENN4L	DRENN3L	DRENN2L	DRENN1L	DRENN0L
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

0A0h~0A1h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DRENxH	DRENx7H	DRENx6H	DRENx5H	DRENx4H	DRENx3H	DRENx2H	DRENx1H	DRENx0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

x=A/B

DRENxL [7:0]: DRENxL[7:0]的驅動控制位

DRENxH=0時:

0 = 源電流/灌電流 (Level0) (IOH=VDD-0.6V 30mA@ 5V) (IOL=VSS+0.6V 50mA@ 5V)

1 = 源電流/灌電流 (Level1) (IOH=VDD-0.6V 8mA@ 5V) (IOL=VSS+0.6V 10mA@ 5V)

DRENxH=1時:

1 = 源電流/灌電流 (Level2) (IOH=VDD-0.6V 20mA@ 5V) (IOL=VSS+0.6V 25mA@ 5V)

0 = 源電流/灌電流 (Level3) (IOH=VDD-0.6V 3mA@ 5V) (IOL=VSS+0.6V 5mA@ 5V)

7.5 I/O 口數據寄存器

01Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	PORTA7	PORTA6	-	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	-	x	x	x	x	x

01Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	PORTB7	PORTB6	PORTB5	-	PORTB3	PORTB2	PORTB1	PORTB0
R/W	R/W	R/W	R	-	R/W	R/W	R/W	R/W
POR的值	x	x	x	-	x	x	x	x

7.6 I/O 口管腳配置寄存器

04Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORCTR	-	-	-	-	CCPCT	PWMCT	UAPCT1	UAPCT0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

Bit[3]: **CCPCT** CCP 管腳配置位

0 = CCP1/CCP2 管腳配置在 PORTB2/PORTB3 (默認)

1 = CCP1/CCP2 管腳配置在 PORTA3/PORTA4

Bit[2]: **PWMCT** PWM 管腳配置位

0 = PWM0/PWM01 管腳配置在 PORTB7/PORTB6 (默認)

1 = PWM0/PWM01 管腳配置在 PORTA6/PORTA7

Bit[1:0] **UAPCT[1:0]** UART 管腳配置位

00 = RX 配置在 PORTA4, TX 配置在 PORTA3 (默認);

01 = RX 配置在 PORTA6, TX 配置在 PORTA7

10 = RX 配置在 PORTB2, TX 配置在 PORTB3

11 = RX 配置在 PORTA0, TX 配置在 PORTA1

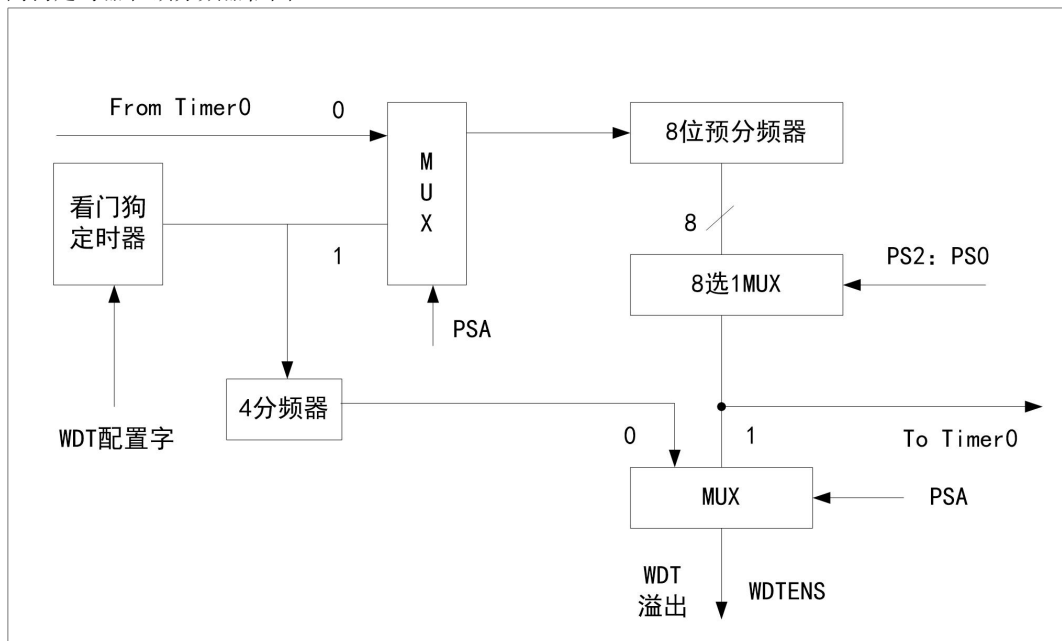
注: 管腳複用功能優先順序: UART > CCP > PWM > IO。

8 定時器/計數器

8.1 看門狗定時器

HC16P122A1/B1的看門狗定時器與Timer0定時器/計數器共用一個預分頻器。當PSA為0時，看門狗定時器每72ms（典型值）產生一個溢出信號；當PSA為1時，WDT溢出時間由預分頻器OPTION[2:0]設置決定，具體請參考Timer0定時器/計數器。

圖8-1 看門狗定時器和預分頻器框圖



079h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	LVD2EN	LVD1EN	-	WDTENS	LVD2F	LVD1F	POR	BOR
R/W	R/W	R/W	-	R/W	R	R	R/W	R/W
POR的值	0	0	-	1	q	q	q	q

Bit [4] **WDTENS**: 硬體看門狗軟體使能位（需配置字使能看門狗，否則該位無效）

1 = 軟體使能硬體看門狗定時器

0 = 軟體遮罩硬體看門狗定時器

注：看門狗的使能邏輯 看門狗使能 = 晶片配置字使能（WDTEN） & 軟體使能（WDTENS）

當系統處於休眠或綠色模式，看門狗定時器溢出將喚醒SLEEP並使其返回高頻或低頻模式，程式從SLEEP指令下一條開始執行。

注：

1. 對看門狗清零之前，檢查I/O 口的狀態和RAM 的內容可增強程式的可靠性；
2. 不能在中斷中對看門狗清零，否則無法偵測到主程序跑飛的狀況；
3. 程式中應該只在主程序中有一次清看門狗的動作，這種架構能夠最大限度的發揮看門狗的保護功能。

➤ 例：看門狗在主程序中的應用

MAIN:

```

BCF          STATUS,RP0      ;BANK0
BSF          PCON,WDTENS     ;軟體使能WDT
...          ;檢查IO狀態是否正確
...          ;檢查RAM是否正確
GOTO        ERR              ;檢查IO/RAM出錯，進入出錯處理程式
CLRWDT
...
CALL        SUB1
CALL        SUB2
...
GOTO        MAIN

```

➤ 例：在休眠狀態下，遮罩看門狗功能，可以節省系統功耗

```

...
BCF          STATUS,RP0      ;BANK0
BCF          PCON,WDTENS     ;軟體遮罩看門狗功能
BCF          OSCCON,T0OSCEN  ;禁止低頻晶體振盪器
SLEEP
BSF          PCON,WDTENS     ;喚醒後,重新使能看門狗功能
...

```

➤ 例：對看門狗定時器操作，看門狗定時器使能和清零

```

BCF          STATUS,RP0      ;BANK0
BSF          PCON,WDTENS     ;使能看門狗
CLRWDT
...

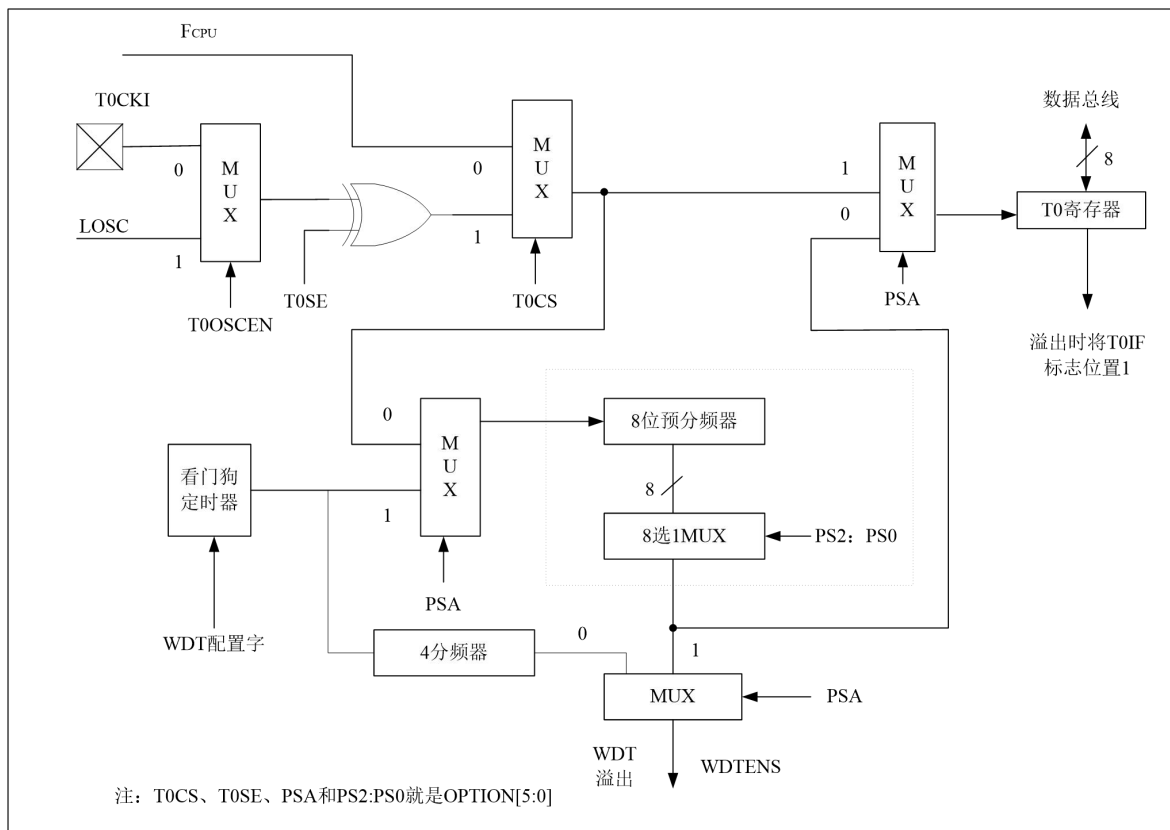
```

8.2 Timer0 定時器/計數器

Timer0 定時器/計數器模組具有如下功能：

- 8 位可編程定時器
- 外部事件計數器
- 綠色模式定時喚醒

圖8-2 Timer0模組和預分頻器（與WDT共用）框圖



07Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCCON	T0SCEN	-	-	-	-	-	HXEN	SCS
R/W	R/W	-	-	-	-	-	R/W	R/W
POR的值	0	-	-	-	-	-	0	q

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	-	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	1	1	1	1	1	1	1

看門狗定時器與Timer0定時器/計數器共用一個預分頻器，當PSA=1預分頻器分配給WDT時，Timer0在所選中時鐘源的每個週期遞增；當PSA=0預分頻器分配給Timer0時，Timer0根據PS[2:0]值選擇的預分頻時鐘遞增。

Timer0的預分頻器不可尋址，當預分頻器分配給Timer0時，對Timer0計數寄存器的寫操作可以對預分頻器清0。

Timer0預分頻比選擇

PS[2:0]	Timer0預分頻比	WDT預分頻比	WDT溢出時間（典型值）
000	1 : 2	1 : 1	18ms
001	1 : 4	1 : 2	36 ms
010	1 : 8	1 : 4	72ms
011	1 : 16	1 : 8	144ms
100	1 : 32	1 : 16	288ms

101	1: 64	1: 32	576ms
110	1: 128	1: 64	1.152s
111	1: 256	1: 128	2.304s

Timer0 工作模式選擇

T0CS	T0OSCEN	T0SE	Timer0工作狀態
0	x	x	定時器模式，計數時鐘 Fcpu， 休眠和綠色模式下停止
1	0	0	計數器模式，計數時鐘 T0CKI，上升沿計數 休眠模式下工作，溢出中斷可喚醒 SLEEP
1	0	1	計數器模式，計數時鐘 T0CKI，下降沿計數 休眠模式下工作，溢出中斷可喚醒 SLEEP
1	1	0	定時喚醒模式，計數時鐘LOSC，上升沿計數 綠色模式下工作，溢出中斷可喚醒SLEEP
1	1	1	定時喚醒模式，計數時鐘LOSC，下降沿計數 綠色模式下工作，溢出中斷可喚醒SLEEP

注：

Timer0 工作模式的選擇需符合上表描述，選擇除上表以外情況可能會造成程式運行混亂，請謹慎操作。

- 例：Timer0工作於定時器模式，計數時鐘為Fcpu，T0計滿到FF後溢出進入中斷

```

MOVLW    0X01
BCF      STATUS,RP0           ;BANK0
MOVWF    OPTION               ;定時器模式，分頻比為1:4
MOVLW    0X00
MOVWF    T0                   ;T0賦初值
BSF      INTCON,T0IE
BCF      INTCON,T0IF
BSF      INTCON,GIE

```

T0INT_SERVICE:

```

...                               ;保存STATUS、W 和PCLATH。
BCF      STATUS,RP0           ;BANK0
BTFSS    INTCON,T0IF         ;檢測T0IF。
GOTO     EXIT_INT            ;T0IF = 0，退出中斷。
BCF      INTCON,T0IF         ;T0IF 清零。
...                               ;TIMER0 中斷服務程式。
...

```

EXIT_INT:

```

...                               ;恢復STATUS、W和PCLATH。
RETFIE   ;退出中斷。

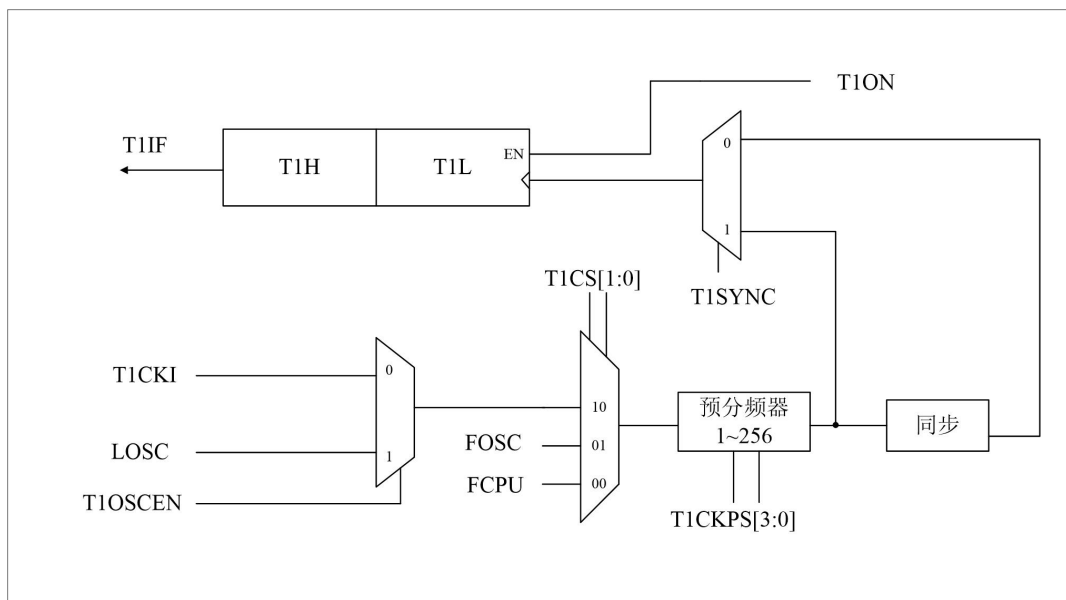
```

8.3 Timer1 定時器/計數器

Timer1 定時器/計數器模組具有如下功能：

- 16 位可編程定時器
- 外部事件計數器，可編程選擇同步、非同步功能
- 綠色模式定時喚醒

圖8-3Timer1模組框圖



8.3.1 Timer1 控制寄存器

05Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CON	T1CS1	T1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	-	T1ON
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
POR的值	0	0	0	0	0	0	-	0

Timer1時鐘源選擇

T1CS1	T1CS0	T1OSCEN	時鐘源
0	0	x	指令時鐘 (F _{CPU})
0	1	x	系統時鐘 (F _{sys})
1	0	0	TICKI引腳上的外部時鐘
1	0	1	低頻系統時鐘

注：

Timer1 時鐘源的選擇需符合上表描述，選擇除上表以外情況會造成程式運行不正常，請謹慎操作。

Timer1輸入時鐘預分頻比選擇

T1CKPS[3:0]	Timer1 預分頻比
0000	1 : 1
0001	1 : 2
0010	1 : 4
0011	1 : 8

0100	1: 16
0101	1: 32
0110	1: 64
0111	1: 128
1xxx	1: 256

Timer1的預分頻器不可尋址，可以通過對Timer1計數寄存器寫操作將預分頻器清0。

Timer1工作模式選擇

T1ON	T1CS[1:0]	T1OSCEN	T1SYNC	Timer1工作模式
1	00	x	x	定時器模式，休眠和綠色模式下停止
1	01	x	x	定時器模式，休眠和綠色模式下停止
1	10	0	0	同步計數器模式，休眠模式下停止
1	10	0	1	非同步計數器模式，休眠模式下工作，溢出中斷可喚醒SLEEP
1	10	1	0	同步定時喚醒模式，綠色模式下停止，溢出中斷不能喚醒SLEEP
1	10	1	1	非同步定時喚醒模式，綠色模式下工作，溢出中斷可喚醒SLEEP

注：

1、T1 為 16 位計時器，在溢出中斷重新賦值時應先 T1H，後 T1L，避免 T1L 在操作中的進位被覆蓋；清空時則應先 T1L 後 T1H，避免 T1L 進位意外進入 T1H 造成清空失敗。

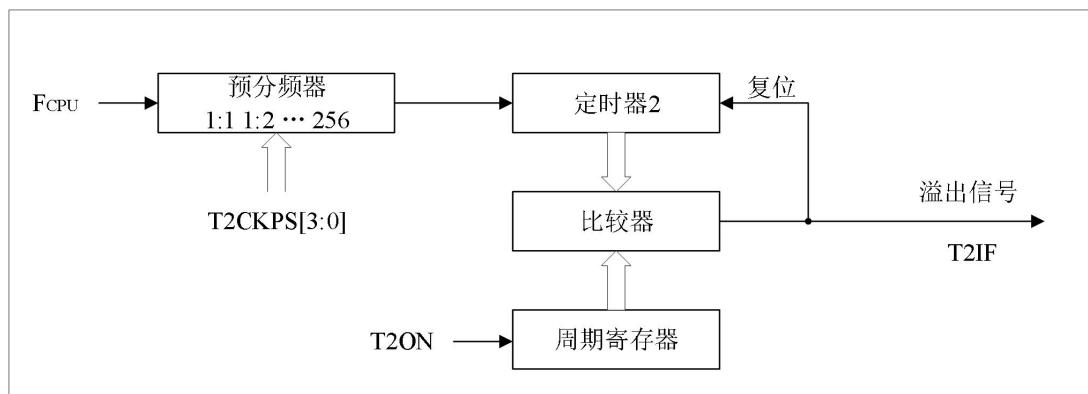
2、Timer1 工作於同步計數器模式和同步定時喚醒模式時，不能喚醒 SLEEP 或綠色模式

3、Timer1 工作模式的選擇需符合上表描述，選擇除上表以外情況可能會造成程式運行混亂，請謹慎操作

8.4 Timer2 定時器

Timer2定時器具有8位預分頻器和8位週期寄存器(PR2)，Timer2定時器的輸入時鐘為指令時鐘FCPU，輸入時鐘通過預分頻器產生Timer2計數時鐘，當計數到與週期寄存器（PR2）的值相同時，在下一指令週期產生Timer2溢出信號，可根據實際需要選擇不同的預分頻比及設置週期寄存器的值，產生不同溢出時間。

圖8-4 Timer2模組框圖



8.4.1 Timer2 控制寄存器

05Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	-	T2CKPS3	T2CKPS2	T2CKPS1	T2CKPS0	T2ON	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-
POR的值	-	0	0	0	0	0	-	-

Bit [2] **T2ON**: Timer2模組使能位

1 = 使能Timer2模組

0 = 禁止Timer2模組

Timer2具有一個8位可編程預分頻器，關閉Timer2模組和對Timer2計數寄存器T2CON寄存器寫操作都將對預分頻器清0。

T2CKPS[3:0]	Timer2 預分頻比
0000	1 : 1
0001	1 : 2
0010	1 : 4
0011	1 : 8
0100	1 : 16
0101	1 : 32
0110	1 : 64
0111	1 : 128
1xxx	1 : 256

8.4.2 Timer2 計數寄存器

05Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2	Timer2計數寄存器							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

8.4.3 Timer2 週期寄存器

05Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR2	Timer2週期寄存器							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

Timer2 定時器的輸入時鐘為指令時鐘 F_{CPU}，輸入時鐘通過預分頻器產生 Timer2 計數信號，當計數到與週期寄存器（PR2）的值相同時產生 Timer2 溢出信號。

Timer2 溢出時間 = (PR2 + 1) * 預分頻比/F_{cpu}。

8.5 CCP 模組

HC16P122A1/B1具有2個獨立的CCP模組CCP1和CCP2，每個CCP模組具有三種模式：

- 捕捉
- 比較
- PWM

CCP模組的時基由Timer1和Timer2提供。

CCP模組的時基

CCP模式	時鐘源
捕捉	Timer1
比較	Timer1
PWM	Timer2/Timer1

082h、085h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPxCON	-	-	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	0	0	0	0	0	0

Bit [5:4] **DCxB[1:0]**: PWM占空比最低有效位

捕捉模式：未使用

比較模式：未使用

PWM模式：PWM占空比的低2位，高8位是CCPRxL寄存器

Bit [3:0] **CCPxM[3:0]**: CCPx模式選擇位

0000 = 捕捉/比較/PWM關閉（複位CCP模組）

0001 = 未使用（保留）

0010 = 比較模式，匹配時輸出翻轉電平（PIRx寄存器的CCPxIF位置1）

0011 = 未使用（保留）

0100 = 捕捉模式，每個下降沿

0101 = 捕捉模式，每個上升沿

0110 = 捕捉模式，每4個上升沿

0111 = 捕捉模式，每16個上升沿

1000 = 比較模式，匹配時輸出高電平（PIRx寄存器的CCPxIF位置1）

1001 = 比較模式，匹配時輸出低電平（PIRx寄存器的CCPxIF位置1）

1010 = 比較模式，匹配時僅產生軟體中斷（PIRx寄存器的CCPxIF位置1，CCPx引腳不受影響）

1011 = 比較模式，觸發特殊事件（PIRx寄存器的CCPxIF位置1，Timer1計數寄存器複位，CCPx引腳不受影響。）

11xx = PWM模式

8.5.1 捕捉模式

在輸入捕捉模式，適合用於測量引腳輸入週期性方波信號的週期、頻率和占空比等，也適合用於測量引腳輸入的非週期性矩形方波脈衝信號的寬度、到達時刻或消失時刻等參數。

當CCPx模組工作於捕捉模式時，一旦有下列事件在引腳CCPx上發生，CCPRx寄存器立即捕捉下這一時刻的TMR1計數值：

- 每個下降沿
- 每個上升沿

- 每 4 個上升沿
- 每 16 個上升沿

CCPxCON寄存器的CCPxM[3:0]設置的是預分頻器，關閉CCP模組或者CCP模組不在捕捉模式，預分頻計數器將會被清0。為避免錯誤中斷，可在改變預分頻比前通過清0 CCPxCON寄存器來關閉CCP模組。

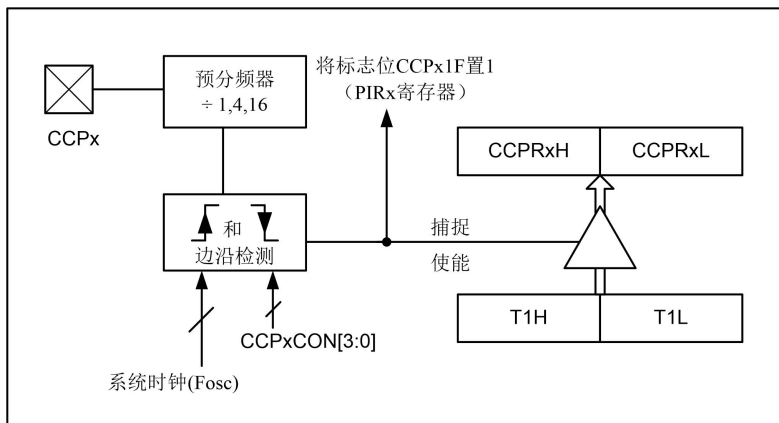
在捕捉模式下，Timer1必須運行在定時器模式或同步計數器模式。

使用注意：

- 1、在捕捉模式下，CCPx引腳必須由相應的方向控制器設定為輸入方式。
- 2、CCPxCON寄存器的CCPxM[3:0]設置的是預分頻器，關閉CCP模組或者CCP模組不在捕捉模式，預分頻計數器將會被清零。為避免錯誤中斷，可在改變預分頻比前通過清零 CCPxCON寄存器來關閉CCP模組。如果需要中途改變預分頻器的分頻比，建議使用以下程式片段：


```
BCF      STATUS, RP0           ;BANK0
CLRF     CCPxCON               ;關閉CCPx模組
MOVLW    0X05/0X06/0X07       ;選取新的分頻比（1:1、1:4、1:16）
MOVWF    CCPxCON               ;賦予CCPxCON寄存器，並打開CCPx模組
```
- 3、當一個捕捉事件發生後，硬體自動將CCPx的中斷標誌位CCPxIF置1，表示產生了一次CCPx捕捉中斷。CCPxIF位必須用軟體重新清零。當CCPRx寄存器中的值還未被程式讀取，而又發生了另一個新的捕捉事件時，原先的值將被新的值覆蓋掉。
- 4、在捕捉模式下，Timer1必須運行在定時器模式或同步計數器模式。

圖8-5 捕捉模式工作原理圖



CCPx引腳上發生變化時，CCPRxH:CCPRxL捕捉Timer1計數寄存器的16位值，PIRx寄存器的中斷標誌位CCPxIF被置1。如果在CCPRxH和CCPRxL寄存器的值被讀出之前又發生另一次捕捉，那麼原來的捕捉值會被新捕捉值覆蓋。

8.5.2 比較模式

輸出比較模式，適用於從引腳上輸出不同寬度的矩形正脈衝、負脈衝、延時驅動信號、可控矽驅動信號、步進電機驅動信號等。

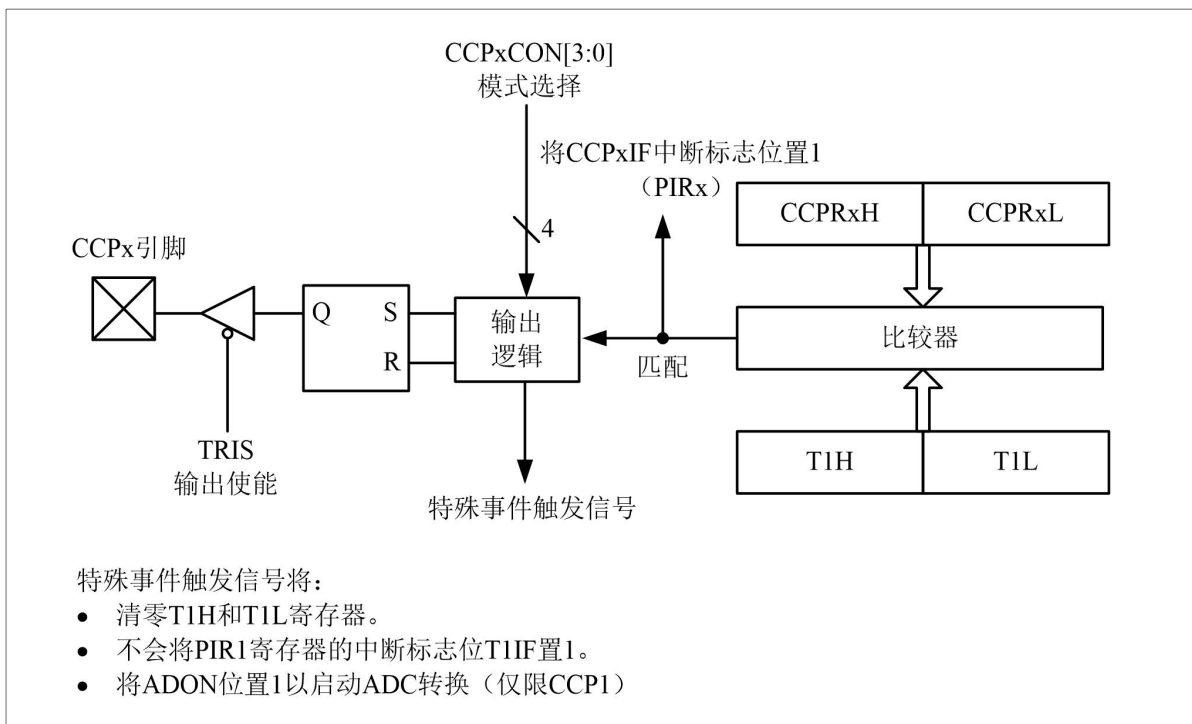
在比較模式下，CCPRxH:CCPRxL寄存器對成為了Timer1的週期寄存器，一旦Timer1計數寄存器對與CCPRxH和CCPRxL寄存器對發生匹配，Timer1計數寄存器對在Timer1時鐘的下一個上升沿複位，CCPx模組根據CCPxM[3:0] 控制位的配置進行相應操作：

- CCPx 引腳輸出翻轉電平
 - CCPx 引腳輸出高電平
 - CCPx 引腳輸出低電平
 - 僅產生軟體中斷
 - 產生特殊事件觸發信號
- 所有比較模式都能產生CCP中斷。

使用注意：

- 1、當選擇產生特殊事件觸發信號時，如果ADC被使能，則啟動一次ADC轉換（僅限於CCP1）。在此模式下CCPx模組不會對CCPx引腳進行控制。
- 2、在比較模式下，CCPx引腳必須由相應寄存器設定為輸出模式，以便作為比較器的輸出端使用。
- 3、應該注意的是，如果對控制寄存器CCPxCON進行重新賦值，將會迫使CCPx引腳輸出一個默認的低電平，而這並非是正常的比較輸出結果。

圖8-6 比較模式工作原理圖



8.5.3 PWM 模式

脈寬調製，PWM(pulse width modulation)輸出工作模式，適用於從引腳上輸出脈衝寬度隨時可調的PWM信號。例如，實現直流電動機調速、簡易D/A轉換器、步進電機的變頻控制等。

8.5.3.1 PWM不選擇擴展

一、PWM時鐘源為Timer2

在 PWM 模式下，當 Timer2 計數寄存器中的值與 PR2 寄存器中的值發生匹配時，在下一個計數時鐘 Timer2 計數寄存器被清零，CCPx 引腳被置 1（如果 PWM 占空比為 0%，CCPx 引腳將不會被置 1），PWM 占空比值從 CCPRxL 鎖存到 CCPRxH（在 Timer2 計數寄存器中的值與 PR2 寄存器中的值發生匹配前，占空比值不會被鎖存到 CCPRxH 中）。

圖8-7 PWM模式工作原理圖

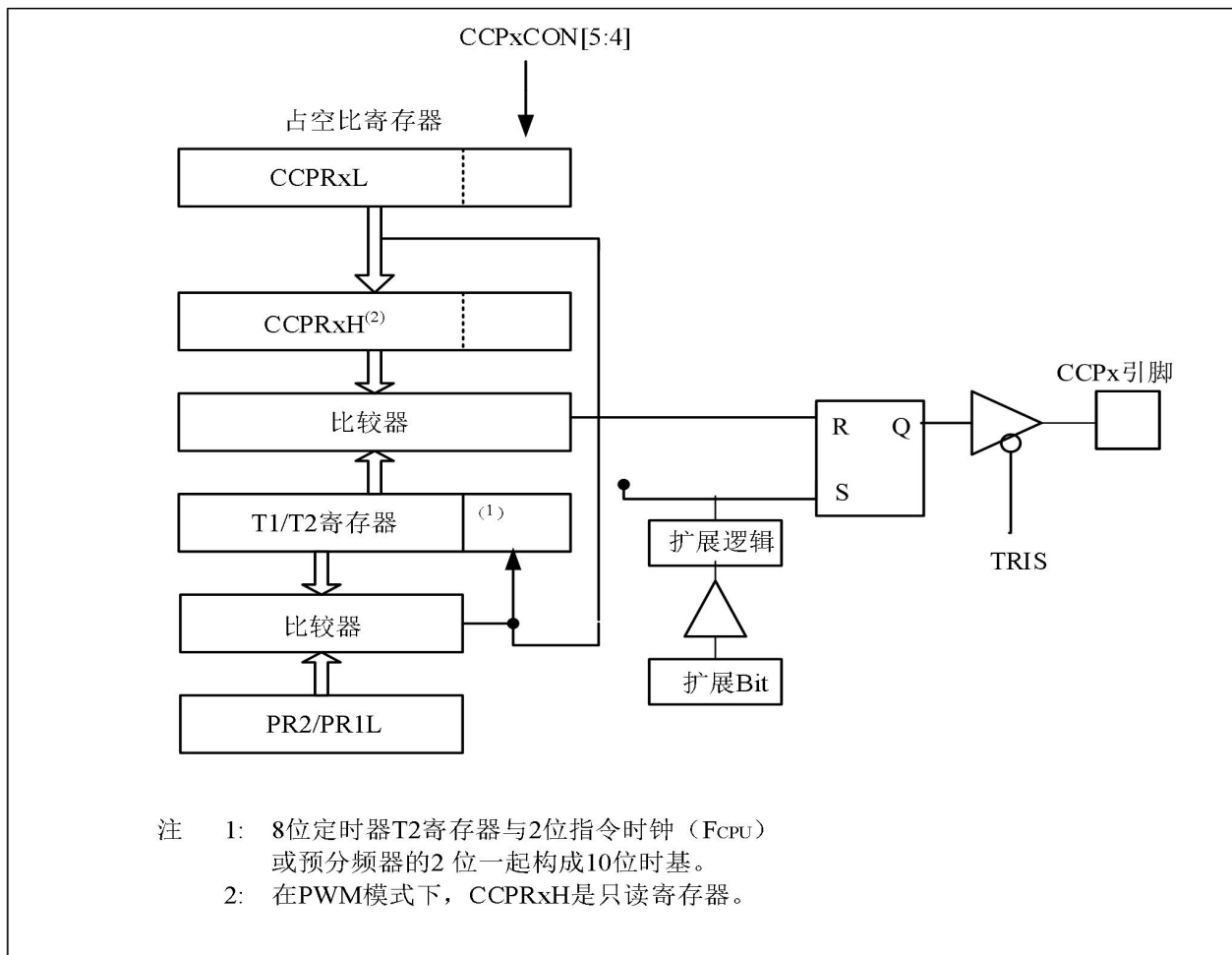
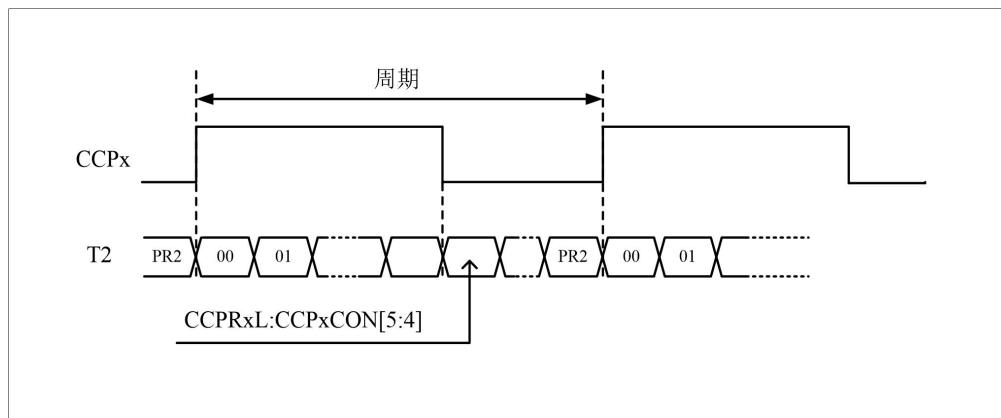


圖8-8 PWM波形圖



✓ 以下公式中，當晶片配置為 4T 模式時， $n=1$ ；當晶片配置為 2T 模式時， $n=1/2$

PWM 週期：

$$\text{PWM 周期} = [(PR2) + 1] \cdot 4 \cdot n \cdot T_{\text{sys}}$$

(Timer2 預分頻值)

注： $T_{\text{sys}} = 1/F_{\text{sys}}$

PWM脈衝寬度：

$$\text{脉冲宽度} = (CCPRxL:CCPxCON[5:4]) \cdot n \cdot T_{\text{sys}} \cdot (\text{Timer2 預分頻值})$$

注： $T_{\text{sys}} = 1/F_{\text{sys}}$

如果脈衝寬度值比週期長，則指定的PWM引腳將保持不變。

PWM占空比：

$$\text{占空比} = \frac{(CCPRxL:CCPxCON<5:4>)}{4(PR2 + 1)}$$

當時鐘模式選擇配置字選擇為2T時，PWM占空比由CCPRxL寄存器和CCPxCON寄存器的DCxB[1]位決定。

PWM解析度：

$$\text{分辨率} = \frac{\text{Log}[4 (PR2 + 1)]}{\text{Log}(2)} \text{ 位}$$

解析度是PR2寄存器值的函數，當PR2為255時PWM最大解析度為10位（時鐘模式選擇配置字選擇為2T時，PWM最大解析度為9位）。

PWM使用：

1. 設置相關TRISB位為1，禁止CCPx引腳輸出
2. 設置PWM週期，輸入PR2寄存器值
3. 設置CCPxCON寄存器，將CCP模組配置為PWM模式
4. 設置PWM占空比，輸入CCPRxL寄存器和CCPxCON[5:4]寄存器值
5. 配置和啟動Timer2
 - 將PIR1寄存器的T2IF中斷標誌位清零
 - 設置T2CON寄存器的T2CKPS位，選擇Timer2預分頻
 - 將T2CON寄存器的T2ON置1，使能Timer2
6. 設置PWM輸出
 - 等待Timer2溢出，PIR1寄存器的T2IF位置1
 - 將TRISB2或TRISB3位清零，讓CCPx引腳輸出
7. 如何關閉PWM輸出

- 將TRISB2或TRISB3位置1，設引腳輸入
- 設置CCPxCON寄存器的CCPxCON[3:0]設為0000，關PWM功能，CCPx用作I/O口
- 根據需要,設PB2或PB3輸出為高電平或者低電平

➤ 例：配置PWM輸出38K驅動方波，採用4M/4T模式。

```

ORG      0000H           ;複位向量
GOTO     MAIN
ORG      0004H           ;中斷
.....
RETFIE

```

MAIN:

```

BCF      STATUS,RP0      ;BANK0
BCF      PCON,WDTENS      ;DISABLE WDT
CLRF     OPTION           ;使用高頻時鐘
BSF      TRISB,2          ;CCPx口設為輸入

```

```

MOVLW    D'25'
MOVWF    PR2              ;設置PR2為26US

```

```

BCF      STATUS,RP0      ;BANK0
MOVLW    H'0D'
MOVWF    CCPR1L           ;占空比高8位
MOVLW    B'00001100'     ;選PWM模組，填寫占空比低兩位
MOVWF    CCP1CON          ;脈衝寬度13US

```

```

BCF      PIR1,T2IF
CLRF     T2CON            ;分頻比1: 1
BSF      T2CON,T2ON       ;開T2
BSF      STATUS,RP0
BSF      PIE1,T2IE

```

```

BTFSS    PIR1,T2IF        ;等待T2溢出
GOTO     $-1
BCF      PIR1,T2IF        ;清除中斷標誌
BCF      STATUS,RP0      ;BANK0
BCF      PIE1,T2IE
BCF      TRISB,2          ;打開PWM輸出，配置完成，PB2將持續輸出38K

```

二、PWM2時鐘源為Timer1

05Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR1L	Timer1週期寄存器低位元組							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR值	0	0	0	0	0	0	0	0

060h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR1CON	PWM1T1	PWM1T0	PWM2T1	PWM2T0	TICKPS3	TICKPS2	PWMPR1	PR1EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR值	0	0	0	0	0	0	0	0

Bit[3:2] **TICKPS[3:2]**: 8位可設週期Timer1的分頻比控制位的高兩位

TICKPS[3:0]	Timer1 預分頻比
0000	1 : 1
0001	1 : 2
0010	1 : 4
0011	1 : 8
0100	1 : 16
0101	1 : 32
0110	1 : 64
0111	1 : 128
1xxx	1 : 256

Bit[1] **PWMPR1**: Timer1提供PWM時鐘源使能位

1=CCP2的PWM的時鐘源由8位可設週期的Timer1提供（當PR1EN有效時，該位可用）

0=CCP2的PWM的時鐘源由Timer2提供

Bit[0] **PR1EN**: Timer1可設週期使能位

1= Timer1為可設週期的8位Timer

0= Timer1為16位Timer

使用注意：將寄存器PR1CON[Bit1~Bit0]置1，使PWM2的時鐘源選擇位可設週期的8位Timer1，其他使用操作可參考由Timer2提供時鐘源的PWM2的使用。

8.5.3.2 PWM選擇擴展

當PWM選擇擴展週期時，4 個調製週期為一個輸出週期，此時PWM最高可擴展到12位。也就是說當選擇擴展PWM模式時，PWM將為4個波形一個週期，此時不論你在哪一個波形輸出時改變PWM的數據記憶體都將在下一個週期才生效。

擴展週期控制寄存器PR1CON

060h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR1CON	PWM1T1	PWM1T0	PWM2T1	PWM2T0	TICKPS3	TICKPS2	PWMPR1	PR1EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR值	0	0	0	0	0	0	0	0

Bit[7:6] **PWM1T[1:0]**: PWM1的擴展週期選擇位

00: 無擴展週期

01: 擴展週期為1

10: 擴展週期為1、2

11: 擴展週期為1、2、3

Bit[5:4] **PWM2T[1:0]**: PWM2的擴展週期選擇位

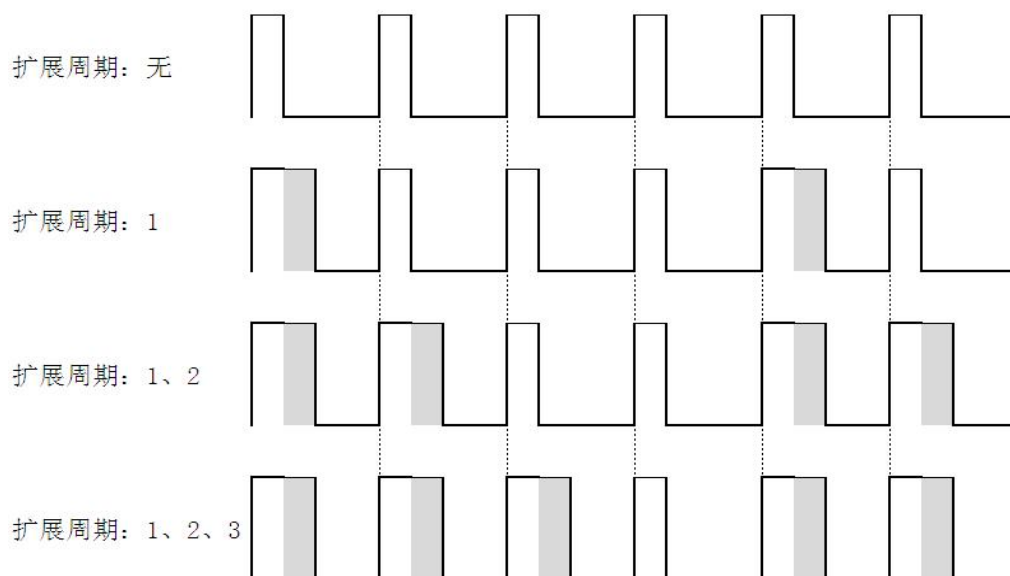
00: 無擴展週期

01: 擴展週期為1

10: 擴展週期為1、2

11: 擴展週期為1、2、3

PWM擴展示意如下:



9 PWM模組

9.1 PWM 特性

- 1 組帶死區互補 PWM 或 2 路獨立 PWM 輸出
- 提供每個 PWM 週期溢出中斷，但中斷共用同一向量入口
- 輸出極性可選擇
- 提供出錯偵測功能可緊急關閉 PWM 輸出
- PWM 工作時鐘源可設定時鐘分頻比
- PWM 可做定時器/計數器使用

HC16P122A1/B1 集成了一個 12 位 PWM 模組 PWM0，PWM0 有一個計數器，PWM0 的計數器由 PWM0_EN 和 PWM01_EN 來控制，只要使能 PWM0_EN 和 PWM01_EN 中的任何一個，計數器就會啟動，計數器的時鐘源通過 PWM0C 控制寄存器裏的 CK0 來選擇。

當需要從晶片管腳輸出 PWM 波形時，還需要將端口設置為輸出模式。如果不希望從晶片管腳上輸出 PWM 波形，可以不用設置端口模式，這時候 PWM 的計數器可以當一個定時器來使用，當計數器溢出時，如果中斷允許也會產生 PWM 中斷。

如果 EFLT0 置 1，PWM0 輸出和其互補輸出可由 FLT0 引腳輸入信號變化自動關閉。一旦檢測到 FLT0 引腳輸入有效電平，PWM 輸出會立即關閉，但 PWM 內部計數器仍在繼續運行，這樣方便在 FLT0 引腳錯誤去除後繼續 PWM 輸出。在 FLT0 輸入信號有效期間，FLT0S 位無法清除。只有當 FLT0 輸入信號消失後，才能軟體清除 FLT0S 狀態位，此時 PWM 恢復正常輸出。

PWM 模組有相應的控制位及標誌位，方便用戶定時修改 PWM 模組下一次迴圈的週期或占空比。

9.2 PWM 輸出模式

PWM 模組包含獨立的波形發生模組，對應的 PWM 輸出為 PWM0/PWM01，通過控制相關寄存器可使 PWM 輸出配置成互補輸出模式或獨立輸出模式。

9.2.1 互補輸出模式

當 PWM0M 置 0：PWM 將工作在互補輸出模式，互補輸出模式時，需要使能 PWM0_OEN 和 PWM01_OEN，此時可以控制對應的週期寄存器、占空比寄存器及死區時間寄存器，從而控制互補波形的輸出。互補輸出時可選擇 PWM0&PWM01 輸出極性，方便用戶各種電平驅動需求。

9.2.2 獨立輸出模式

當 PWM0M 置 1：PWM 將工作在獨立輸出模式，獨立輸出模式時，可以控制相關寄存器使能對應 PWM 端口單一輸出或同時輸出，同時讓 PWM0&PWM01 輸出時，其週期相同但占空比可單獨設置。此時互補輸出模式時占空比寄存器將控制 PWM0 的占空比，死區時間控制寄存器將控制 PWM01 的占空比，獨立輸出時也可控制 PWMx&PWMx1 輸出極性，方便用戶各種電平驅動需求。

9.3 PWM 相關寄存器

9.3.1 PWM 使能寄存器

25Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMEN	-	EFLT	-	-	EPWM01	-	-	EPWM00
R/W	-	R/W	-	-	R/W	-	-	R/W
POR的值	-	0	-	-	0	-	-	0

Bit [6] **EFLT**: FLT引腳配置位

1 = PWM故障檢測輸入引腳

0 = 普通I/O口或其他功能

Bit [3] **EPWM01**: PWM01輸出控制位

0 = PWM01輸出禁止，用作I/O功能

1 = PWM01 輸出允許

Bit [0] **EPWM00**: PWM0輸出控制位

0 = PWM0輸出禁止，用作I/O功能

1 = PWM0 輸出允許

當PWMEN清0後，PWM輸出立即關閉。

FLT 端口主要用於檢測異常信號，快速關閉 PWM 輸出，FLT 探測到故障後，由硬體執行使 PWM 輸出關閉，所以當故障發生後，它可以快速回應，使得 PWM 輸出無效以保護連接 PWM 的大功率器件。當使能 FLT 引腳故障檢測功能後，端口禁止任何上下拉電阻功能。

如果 EFLT 位清零，則表示 FLT 端口對 PWM 定時器輸出控制無效。

25Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FLTM	-	-	-	-	-	-	FLT0M1	FLT0M0
R/W	-	-	-	-	-	-	R/W	R/W
POR的值	-	-	-	-	-	-	0	0

輸出故障時，PWM 口輸出狀態：

Bit [1:0] **FLT0M[1:0]**: PWM0 口 FLT 故障時後，端口輸出狀態

00 = PWM0 輸出低電平

01 = PWM00 輸出低電平，PWM01 輸出高電平

10 = PWM00 輸出高電平，PWM01 輸出低電平

11 = PWM0 輸出高電平

9.3.2 PWM0 控制寄存器

25Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0C	-	-	FLTS	FLTC	PWM0S1	PWM0S0	CK01	CK00
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	0	0	0	0	0	0

Bit [5] **FLTS**: FLT狀態位

1 = PWM輸出關閉，硬體置1

0 = PWM 正常狀態，軟體清 0

Bit [4] **FLTC**: FLT引腳配置位

1 = FLT為高電平時，PWM輸出關閉

0 = FLT為低電平時，PWM輸出關閉

Bit [3:2] **PWM0S[1:0]**: PWM0和PWM01占空比輸出模式選擇位

11 = PWM0和PWM01均為低有效

10 = PWM0為低有效，PWM01為高有效

01 = PWM0為高有效，PWM01為低有效

00 = PWM0和PWM01均為高有效

Bit [1:0] **CK0[1:0]**: PWM1輸出控制位

11 = Fosc/128

10 = Fosc/32

01 = Fosc/8

00 = Fosc/1

注: Fosc = 32MHz

注意:

- 1、 PWM0C寄存器中的FLTS和FLTC位控制PWM0定時器，而寄存器中的PWM0S,CK0[1:0]只能影響12位PWM定時器。
- 2、 PWM輸出關閉時，PWM0/1/2和PWM01/11/21輸出固定電平：
PWM0S[1:0] = 00，PWM0和PWM01均輸出固定低電平；
PWM0S[1:0] = 01，PWM0輸出固定低電平，PWM01輸出固定高電平；
PWM0S[1:0] = 10，PWM0輸出固定高電平，PWM01輸出固定低電平；
PWM0S[1:0] = 11，PWM0和PWM01均輸出固定高電平；
- 3、 一旦檢測到FLT引腳輸入有效電平，PWM輸出會立即關閉，但PWM內部計數器仍在繼續運行。這樣方便在FLT引腳錯誤去除後繼續PWM輸出。
- 4、 在FLT輸入信號有效期間，FLTS位無法清除。只有當FLT輸入信號消失後，才能軟體清除FLTS狀態位，此時PWM恢復正常輸出。

9.3.3 PWM 模式寄存器 PWMM

260h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMM	-	-	-	PWM0M	-	-	-	RELOAD0
R/W	-	-	-	R/W	-	-	-	R/W
POR的值	-	-	-	0	-	-	-	0

Bit [4] **PWM0M**: PWM0工作模式選擇位

1 = PWM1&PWM11工作於獨立輸出模式

0 = PWM0&PWM01 工作於互補輸出模式

Bit [0] **RELOAD0**: PWM0自動重載使能位

1 = 使能自動重載

0 = 禁止自動重載

注：默認為1，默認狀態下修改週期、占空比、死區等參數後，參數會自動重載，並在下一個 PWM 週期使用這些參數。在修改週期、占空比、死區等參數前設置此位為0將禁止自動重載，PWM 不會使用任何新的參數，輸出狀態保持不變，在軟體修改參數完成，再設置此位為1，新修改的參數將在下一個 PWM 週期統一使用，這樣可實現多組 PWM 間的同步。

9.3.4 PWM0 週期/占空比寄存器

12 位 PWM 週期控制寄存器的高 4 位

25Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PH	-	-	-	-	PP0.11	PP0.10	PP0.9	PP0.8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

12 位 PWM 週期控制寄存器的低 8 位

25Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PL	PP0.7	PP0.6	PP0.5	PP0.4	PP0.3	PP0.2	PP0.1	PP0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM0輸出週期=[PWM0PH:PWM0PL] * PWM時鐘源

當[PWM0PH:PWM0PL]=0x00時，占空比 [PWM0DH:PWM0DL] =0x00，獨立模式：

如果PWM0S[1:0]=00，PWM0輸出低電平，PWM01輸出低電平

如果PWM0S[1:0]=01，PWM0輸出低電平，PWM01輸出高電平

如果PWM0S[1:0]=10，PWM0輸出高電平，PWM01輸出低電平

如果PWM0S[1:0]=11，PWM0輸出高電平，PWM01輸出高電平

12 位 PWM 脈寬控制寄存器的高 4 位

25Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	-	-	-	-	PD0.11	PD0.10	PD0.9	PD0.8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

12 位 PWM 脈寬控制寄存器的低 8 位

259h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DL	PD0.7	PD0.6	PD0.5	PD0.4	PD0.3	PD0.2	PD0.1	PD0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM輸出占空比=[PWM0DH:PWM0DL] * PWM時鐘源

當[PWM0PH:PWM0PL]<[PWM0DLH:PWM0DL]，並且[PWM0PH:PWM0PL] ≠0x00，獨立模式：

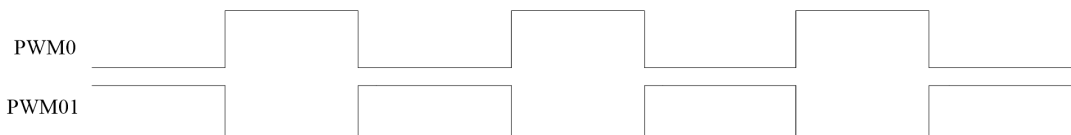
如果PWM0S[1:0]=00，PWM0輸出高電平，PWM01輸出高電平

如果PWM0S[1:0]=01，PWM0輸出高電平，PWM01輸出低電平

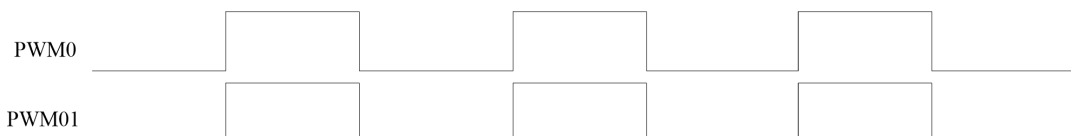
如果PWM0S[1:0]=10，PWM0輸出低電平，PWM01輸出高電平

如果PWM0S[1:0]=11，PWM0輸出低電平，PWM01輸出低電平

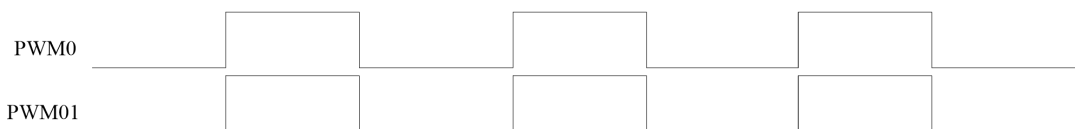
PWM0S=00& PWM0M=0：PWM0和PWM01工作於互補模式且均為高有效



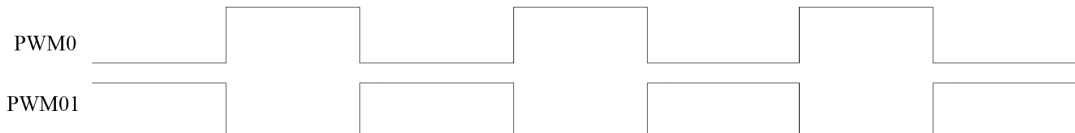
PWM0S=00& PWM0M=1：PWM0和PWM01工作於獨立模式且均為高有效



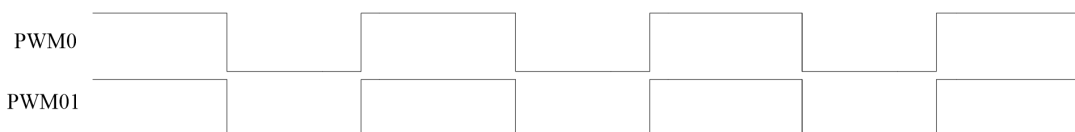
PWM0S=01& PWM0M=0：PWM0和PWM01工作於互補模式且PWM0為高有效、PWM01為低有效



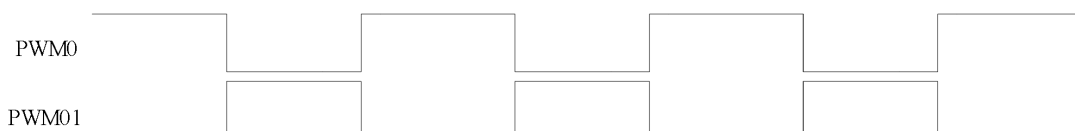
PWM0S=01 & PWM0M=1: PWM0和PWM01工作於獨立模式且PWM0為高有效、PWM01為低有效



PWM0S=10 & PWM0M=0: PWM0和PWM01工作於互補模式且PWM0為低有效、PWM01為高有效



PWM0S=10 & PWM0M=1: PWM0和PWM01工作於獨立模式且PWM0為低有效、PWM01為高有效



9.3.5 死區時間

PWM0 死區時間控制寄存器

257h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DTL	PWM0DTL[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

258h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DTH	PWM0DTH[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

當 PWM0_M=1 時，PWM0 工作在 2 路獨立模式，此時的死區時間寄存器被用來當做 PWM01 的占空比寄存器，即獨立模式的 PWM0 可以產生 2 路週期相同，但占空比可以不同的 PWM 波形。

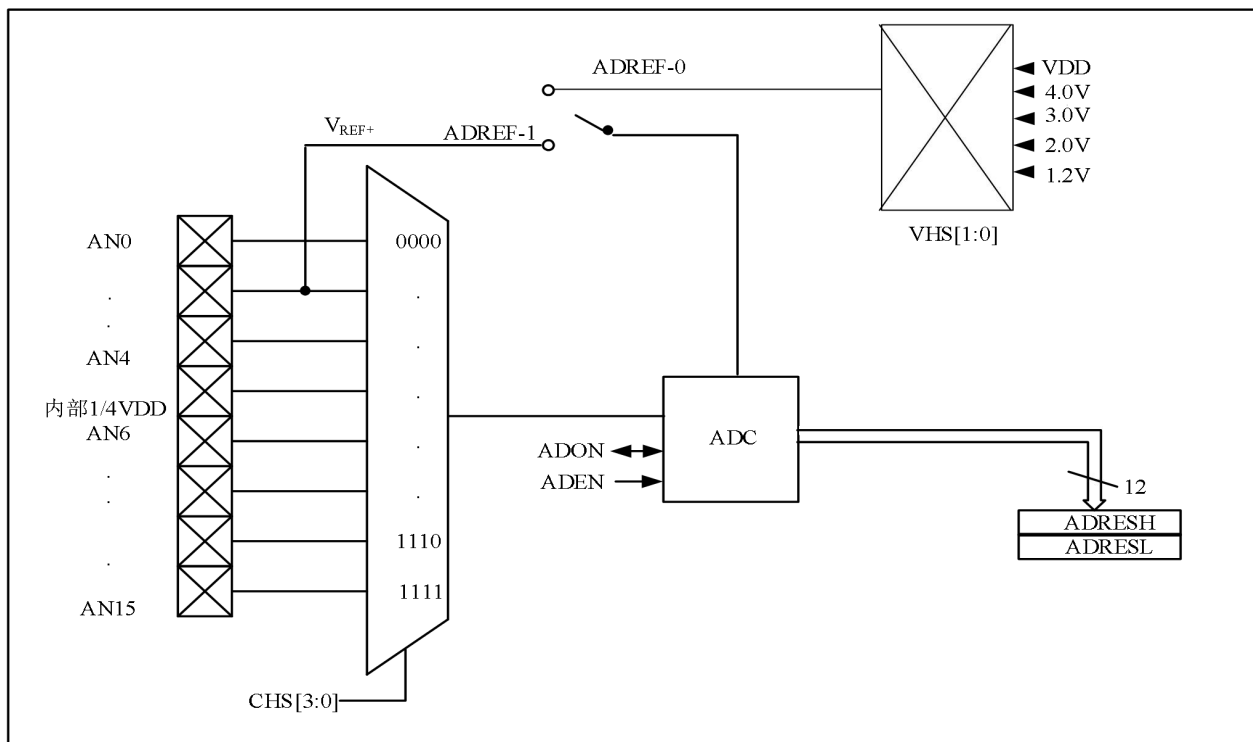
互補模式下：PWM0 死區時間 = [PWM0DTH : PWM0DTL] * PWM0 工作時鐘週期；
死區時間必須小於占空比時間，死區時間與占空比時間的和必須小於 PWM0 週期。

獨立模式下：PWM01 占空比時間 = [PWM0DTH : PWM0DTL] * PWM0 工作時鐘週期。

10 模數轉換 (ADC)

HC16P122A1/B1具有一個12位轉換解析度的模數轉換器，共有13個外部模擬輸入通道，1個內部電池檢測通道。

ADC的等效電路：



10.1 A/D引腳控制寄存器

08Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR值	1	1	-	1	1	1	1	1

08Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELH	ANSEL15	ANSEL14	-	-	ANSEL11	ANSEL10	ANSEL9	ANSEL8
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR值	1	1	-	-	1	1	1	1

ANSEL[15:0]: A/D引腳數模控制位

- 1: 模擬模式，作為模擬信號口，僅可作為AD通道的模擬輸入。
- 0: 數字模式，作為數字輸入或輸出口。

注：

ANSEL 上電初始值為 B' 1111 1111'，即作為模擬輸入。無論是否應用到 AD，均需要在上電後，對 IO 操作之前按需配置，否則 IO 口可能無法受控於對應的端口寄存器，狀態將不確定。

ANSEL[4:0]對應 AN4~AN0 (PA4~PA0)，ANSEL[7:6]對應 AN7、AN6 (PA7、PA6)

ANSEL[11:8]對應 AN11~AN8 (PB3~PB0)，ANSEL[15:14]對應 AN15、AN14 (PB7、PB6)

10.2 A/D 控制寄存器

094h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON0	-	-	CHS3	CHS2	CHS1	CHS0	ADON	ADEN
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	0	0	0	0	0	0

095h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON1	ADFM	ADCS2	ADCS1	ADCS0	VHS2	VHS1	VHS0	ADREF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

注：

ADCON1 寄存器的低四位 Bit3~Bit0 如果全為 0，功耗會比一般情況偏大。在 sleep 模式下為保證系統的低功耗，請避免將 ADCON1 寄存器的低四位 Bit3~Bit0 全設置為 0。

096h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCLK	-	-	-	-	-	ADCLK2	ADCLK1	ADCLK0
R/W	-	-	-	-	-	R/W	R/W	R/W
POR的值	-	-	-	-	-	0	0	0

ADC模擬通道選擇

CHS [3:0]	模擬通道
0000	AN0
0001	AN1
0010	AN2
0011	AN3
0100	AN4
0101	內部 1/4VDD(AN5)
0110	AN6
0111	AN7
1000	AN8
1001	AN9
1010	AN10
1011	AN11
1100	AN12
1101	AN13
1110	AN14
1111	AN15

ADC參考電壓選擇

ADREF	VHS[2:0]	參考電壓
0	000	內部VDD

0	001	內部4.0V
0	010	內部3.0V
0	011	內部2.0V
0	100	內部1.2V
1	xxx	外部參考電壓

ADC數據存放格式選擇

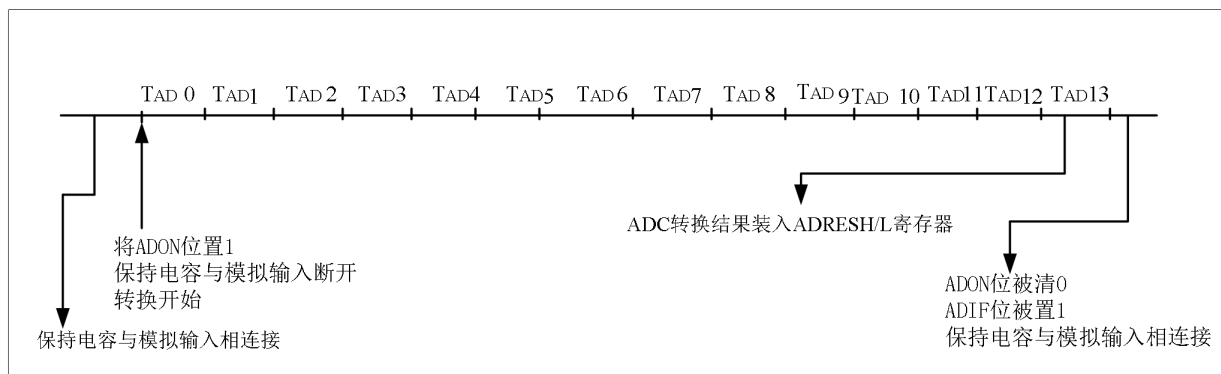
ADFM	數據格式
0	ADRESH[7:0]:ADRESL[7:4]
1	ADRESH[1:0]:ADRESL[7:0]

注意： 1、AN5為內部1/4VDD輸入通道，外部沒有輸入引腳。可作為電池系統的電池檢測器。

2、ADC 所採集數據，當選擇存格式為左對齊時，ADC 精度只能為 12 位，高 8 位存放在 ADRESH 寄存器中，低 4 位存放在 ADRESL 寄存器高 4 位上。當選擇存放格式為右對齊時，ADC 精度只能為 10 位，高 2 位存放在 ADRESH 的低 2 位上，低 8 位存放在 ADRESL 上。

ADC轉換一位數據所需的時間定義為TAD，轉換一次完整的12位數據需要14個TAD。為確保ADC正確轉換，必須滿足適當的TAD時間。

模數轉換TAD 週期



ADC轉換時間(TAD)與工作頻率關係表

ADC 轉換時間 (TAD)		系統頻率 (Fsys) :4MHz
ADC 時鐘源	ADCS[2:0]	典型值
Fsys	000	4us
Fsys /2	001	8us
Fsys /4	010	16us
Fsys /8	011	32us
Fsys /16	100	64us
Fsys /32	101	128us
Fsys /64	110	256us
FRC	111	500khz

系統頻率/AD時鐘源選擇對應的ADC時鐘選擇為對應關係如下表所示：

ADCLK軟體配置表

Fsys（系統頻率）	ADCS[2:0] AD 分頻	ADCLK[2:0] AD 時鐘
32M	000	000
	001	000
	010	000
	011	000
	100	001/010
	101	001/010
	110	011/1XX
	111	011/1XX
16M	000	000
	001	000
	010	000
	011	001/010
	100	001/010
	101	011/1XX
	110	011/1XX
	111	011/1XX
8M	000	000
	001	000
	010	001/010
	011	001/010
	100	011/1XX
	101	011/1XX
	110	011/1XX
	111	011/1XX
4M	000	000
	001	001/010
	010	001/010
	011	011/1XX
	100	011/1XX
	101	011/1XX
	110	011/1XX
	111	011/1XX
2M	000	001/010
	001	001/010
	010	011/1XX
	011	011/1XX

	100	011/1XX
	101	011/1XX
	110	011/1XX
	111	011/1XX
1M	000	001/010
	001	011/1XX
	010	011/1XX
	011	011/1XX
	100	011/1XX
	101	011/1XX
	110	011/1XX
	111	011/1XX
500K	000	011/1XX
	001	011/1XX
	010	011/1XX
	011	011/1XX
	100	011/1XX
	101	011/1XX
	110	011/1XX
	111	011/1XX
250K	000	011/1XX
	001	011/1XX
	010	011/1XX
	011	011/1XX
	100	011/1XX
	101	011/1XX
	110	011/1XX
	111	011/1XX

ADCLK[2:0]: AD時鐘選擇位（在確定系統頻率後，選擇不同的AD時鐘分頻對應的頻率。）

000: AD轉換頻率在4Mhz及以上

001/010: AD轉換頻率為1M或2MHz

011/1XX: AD轉換頻率為1M以下

注:

1. 為了加快轉換速度，建議選用較快時鐘源（ADC轉換時鐘不能超過4MHz）。
2. 當系統頻率高於1 MHz 時，僅當在休眠和綠色模式下進行轉換時才推薦使用FRC時鐘源。

選擇FRC時鐘源後，ADC需等待一個指令週期後才能啟動轉換操作，這使得可以執行SLEEP 指令，以降低轉換期間的系統雜訊。如果使能了ADC中斷，轉換完成時將喚醒SLEEP。如果禁止了ADC中斷，儘管ADEN位仍保持為1，轉換完成後ADC模組將關閉。ADC時鐘源不是FRC時，儘管ADEN位仍保持為1，SLEEP指令會導致當前轉換中止，ADC模組關閉。

除了選擇FRC時鐘源，改變系統時鐘頻率均會改變ADC的時鐘頻率，從而影響ADC轉換時間。

ADEN位置1將使能ADC模組，ADON位置1將啟動一次ADC轉換。ADC轉換完成，ADON位硬體清零，ADIF中斷標誌位置1，ADRESH/ADRESL寄存器值被更新。如果必須在轉換完成前終止轉換，可用軟體將ADON位清零，ADRESH/ADRESL寄存器將保持前次ADC轉換的結果。

10.3 ADC 使用

1. 配置端口：
 - 設置 TRISA 寄存器禁止引腳輸出
 - 設置 ANSEL 寄存器配置引腳為模擬輸入
2. 配置ADC模組：
 - 選擇 ADC 轉換時鐘，設置 ADCS[2:0]
 - 選擇 ADC 參考電壓，設置 ADREF、ADCON0
 - 選擇 ADC 輸入通道，設置 CHS[3:0]
 - 使能 ADC 模組，設置 ADEN
3. 配置ADC中斷（可選）：
 - 清零 ADC 中斷標誌
 - 使能 ADC 中斷
 - 使能外設中斷
 - 使能全局中斷
4. 等待所需採集時間
5. 設置ADON為1 啟動一次ADC轉換
6. 通過以下方式之一等待ADC轉換完成：
 - 查詢 ADON 位
 - 等待 ADC 中斷（已使能中斷）
7. 讀取ADC結果
8. 清零ADC中斷標誌（如果已使能中斷則需要）

➤ 例：配置AD，結果保留在BANK0的NTCADHIGH、NTCADLOW中。

```

.....                                ;其他程式
AD_TEST:
    BCF      STATUS,RP0                ;BANK0
    BSF      TRISA,0                  ;設置AD口為輸入
    BCF      STATUS,RP0                ;BANK0
    MOVLW    B'01010000'              ;INNER REF Fsys/32 ADRESH[7:0]
    ADRESL[7,6]
    MOVWF    ADCON1
                                           ;配置AD通道
    BCF      STATUS,RP0                ;BANK0
    MOVLW    B'00000001'
    MOVWF    ANSELL                   ;PA0作為模擬輸入
    CLRF     ANSELH
    BCF      STATUS,RP0                ;BANK0
    BCF      ADCON0,CHS2
    BCF      ADCON0,CHS1
    BCF      ADCON0,CHS0
  
```

	BCF	ADCON0,VHS1	
	BCF	ADCON0,VHS0	;參考電壓為內部VDD
	NOP		;延時
	NOP		
	BSF	ADCON0,ADEN	;使能ADC
	CALL	DELAY_1	;延時，用戶可自行完成
	BSF	ADCON0,ADON	;開始一次轉換
AD_TEST_WAIT:			
	BTFSC	ADCON0,ADON	;等待轉換完成
	GOTO	AD_TEST_WAIT	
			;轉換完成，保存結果
	MOVF	ADRESH,W	;LOAD THE AD HIGH 8 BITS TO W
	MOVWF	NTCADHIGH	;客戶應用時注意BANK
	MOVF	ADRESL,W	;LOAD THE AD LOW 8 BITS TO W
	BCF	STATUS,RP0	;BANK0
	MOVWF	NTCADLOW	;客戶應用時注意BANK

注意：

- 1、使能ADEN後（不是使能ADON），系統必須延遲一定的時間（視外部輸入信號而定）等待ADC電路穩定。
- 2、睡眠或綠色模式下，禁止ADC並將AD參考電壓設為非內部VDD以降低功耗。

11 串行口通信

HC16P122A1/B1 具有 1 個採用 UART(Universal Asynchronous Receiver/Transmitter)工作方式的全雙工串行通信介面。串行口由 2 個數據緩衝器、一個移位寄存器、一個串行控制寄存器和一個串列傳輸速率發生器等組成。串行口的數據緩衝器由 2 個互相獨立的接收、發送緩衝器構成，可以同時發送和接收數據。發送緩衝器只能寫入而不能讀出，接收緩衝器只能讀出而不能寫入，因而兩個緩衝器可以共用一個地址碼。串行口的兩個緩衝器共用的地址碼是 23Dh。兩個緩衝器統稱串行通信特殊功能寄存器 SBUF。

HC16P122A1/B1 的串行口都有 4 種工作方式，其中兩種方式的串列傳輸速率是可變的，另兩種是固定的，以供不同應用場合選用。用戶可用軟體設置不同的串列傳輸速率和選擇不同的工作方式。主機可通過查詢或中斷方式對接收/發送進行程式處理，使用十分靈活。

串行口對應的硬體部分是 TxD 和 RxD 引腳。

11.1 串行口的控制寄存器 SCON

SCON

23Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON	SM0/FE	SM1	SM2	REN	TB8	RB8	RXWK	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
POR 的值	0	0	0	0	0	x	0	-

- SM0/FE:

當 AUXR 寄存器的 SMOD0/AUXR.1 位為 1 時，該位用於幀錯誤檢測。當檢測到一個無效停止位時，通過 UART 接收器設置該位。它必須由軟體清零。

當 AUXR 寄存器的 SMOD0/AUXR.1 位為 0 時，該位和 SM1 一起指定串行通信的工作方式，如下表所示

SM0	SM1	工作方式	功能說明	串列傳輸速率
0	0	方式0	同步移位串行方式：移位寄存器	當 UARTM0=0 時，串列傳輸速率是 $F_{cpu}/12$ 當 UARTM0=1 時，串列傳輸速率是 $F_{cpu}/2$
0	1	方式1	8 位 UART，串列傳輸速率可變	$(2SMOD/32) \times (BRT \text{ 獨立串列傳輸速率發生器的溢出率})$
1	0	方式2	9 位 UART	$(2SMOD/64) \times F_{cpu}$ 系統工作時鐘頻率
1	1	方式3	9 位 UART，串列傳輸速率可變	$(2SMOD/32) \times BRT \text{ 獨立串列傳輸速率發生器的溢出率}$

當 BRTX12=0 時，BRT 獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/12/(256-BRT)$

當 BRTX12=1 時，BRT 獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/(256-BRT)$

- SM2：允許方式2或方式3多機通信控制位
SM2=1：且 REN 位為 1，則允許接收機處於地址篩選狀態
SM2=0：且 REN 位為 1，則禁止接收機處於地址篩選狀態
- REN：允許/禁止串行接收控制位，由軟體置位
REN=1：允許串行接收狀態
REN=0：禁止串行接收狀態
- TB8：在方式2或者方式3，它為要發送的第 9 位數據，由軟體置位，在方式0、方式1，該位無用
- RB8：在方式2或者方式3，是接收到的第 9 位數據，作為奇偶校驗位或地址幀/數據幀的標誌位
- RXWK：允許/禁止接收喚醒使能位

RXWK = 1: 允許RXD喚醒SLEEP

RXWK = 0: 禁止RXD喚醒SLEEP

11.2 串行口數據緩衝寄存器 SBUF

SBUF

23Dh	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

串行口緩衝寄存器(SBUF)的地址是23Dh，實際是2個緩衝器，寫SBUF的操作完成待發送數據的加載，讀SBUF的操作可獲得已接收到的數據。兩個操作分別對應兩個不同的寄存器，1個是只寫寄存器，1個是只讀寄存器。

串行通道內設有數據寄存器。在所有的串行通信方式中，在寫入SBUF信號的控制下，把數據裝入相同的9位移位寄存器，前面8位為數據位元組，其最低位為移位寄存器的輸出位。根據不同的工作方式會自動將“1”或TB8的值裝入移位寄存器的第9位,並進行發送。

串行通道的接收寄存器是一個輸入移位寄存器。在方式0時它的字長為8位，其他方式時為9位。當一幀接收完畢，移位寄存器中的數據字節裝入串行數據緩衝器SBUF中,其第9位則裝入SCON寄存器中的RB8位。如果由於SM2使得已接收到的數據無效時,RB8和SBUF中內容不變。

由於接收通道內設有輸入移位寄存器和SBUF緩衝器，從而能使一幀接收完將數據由移位寄存器裝入SBUF後，可立即開始接收下一幀資訊，主機應在該幀接收結束前從SBUF緩衝器中將數據取走，否則前一幀數據將丟失。SBUF以並行方式送往內部數據匯流排。

11.3 輔助寄存器 AUXR

AUXR

23Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUXR	-	UARTEN	UARTM0	BRTR	BRTX12	S1BRS	SMOD	SMOD0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	0	0	0	0	x	0	0

Bit [6] **UARTEN**: UART管腳控制位

1 = 使能UART管腳功能

0 = 禁止UART管腳功能，做普通IO口

Bit [5] **UARTM0**: 串口模式0的通信速度設置位

1 = UART串口模式0 2分頻

0 = UART 串口模式 0 12 分頻

Bit [4] **BRTR**: 獨立串列傳輸速率發生器運行控制位

1 = 允許獨立串列傳輸速率發生器運行

0 = 不允許獨立串列傳輸速率發生器運行

Bit [3] **BRTX12**: 獨立串列傳輸速率發生器計數控制位

1 = 獨立串列傳輸速率發生器每1個時鐘計數一次

0 = 獨立串列傳輸速率發生器每 12 個時鐘計數一次

Bit [2] **S1BRS**: 串口（UART）的串列傳輸速率發生器控制位

1 = 選擇獨立串列傳輸速率發生器作為串口（UART）串列傳輸速率發生器

0=選擇定時器1作為串口（UART）串列傳輸速率發生器

Bit [1] **SMOD**: 串列傳輸速率選擇位

1=串行通信方式1、2、3的串列傳輸速率加倍

0=串行通信方式1、2、3的串列傳輸速率不加倍

Bit [0] **SMOD0**: 幀錯誤檢測有效控制位

1=SCON寄存器中的 SM0/FE應用於FE（幀錯誤檢測）功能

0=SCON寄存器中的 SM0/FE應用於SM0功能，和SM1一起指定串行口的工作方式

11.4 獨立串列傳輸速率發生器寄存器 BRT

23Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BRT								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

獨立串列傳輸速率發生器寄存器BRT用於保存重裝時間常數。

11.5 自動地址識別

從機地址寄存器 SADEN

23Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADEN								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

從機地址掩碼寄存器 SADDR

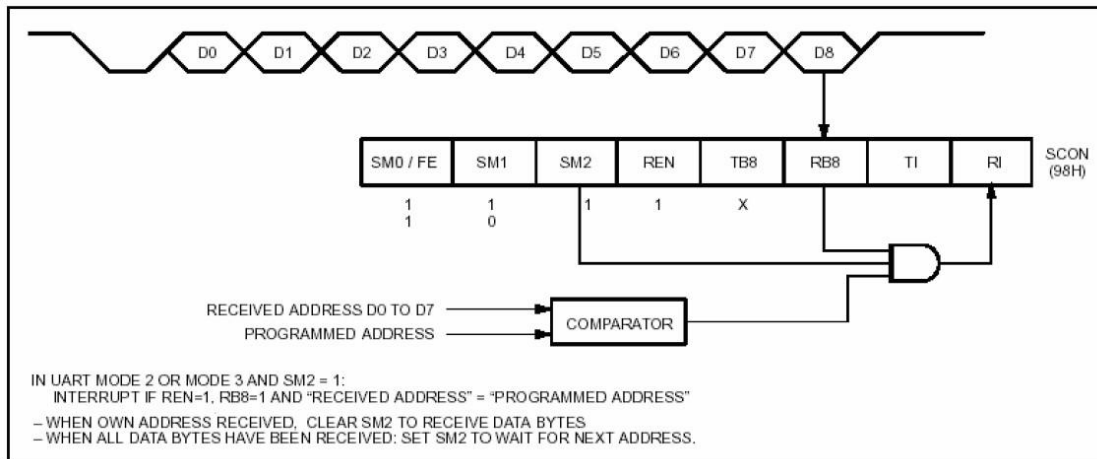
23Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADDR								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

自動地址識別通過硬體比較可以讓UART識別串行碼流中的地址部分，該功能免去了使用軟體識別時需要大量代碼的麻煩。該功能通過設定SCON的SM2位來開啟。

在9位數據UART模式下，即模式2和模式3，收到特定地址或廣播地址時自動置位接收中斷(RXIF)標誌，9位模式的第9位資訊為1表明接收的是一個地址而不是數據。自動地址識別功能請參考下圖。在8位模式，即模式1下，如果SM2置位並且在8位地址與給定地址或廣播地址核對一致後收到有效停止位則RXIF置位。模式0是移位寄存器模式，SM2被忽略。

使用自動地址識別功能可以讓一個主機選擇性的同一個或多個從機進行通訊，所有從機可以使用廣播地址接收資訊。增加了SADDR從機地址寄存器和SADEN地址掩碼寄存器。

SADEN用來定義SADDR中的那些位是“無關緊要”的，SADEN掩碼和SADDR寄存器進行邏輯與來定義供主機尋址從機的“給定”地址，該地址讓多個從機進行排他性的識別。



下麵的實例幫助理解這個方案的通用性：

從機 0

SADDR = 1100 0000

SADEN = 1111 1101

地址 = 1100 00X0

從機 1

SADDR = 1100 0000

SADEN = 1111 1110

地址 = 1100 000X

上面的例子中SADDR是相同的值，而使用SADEN數據來區分兩個從機。從機0要求第0位必須為0，並忽略第1位的值；從機1要求第1位必須為0，並忽略第0位的值。從機0的唯一地址是11000010，而從機1的唯一地址是1100 0001，地址1100 0000是可以同時尋找到從機0和從機1的。

下麵一個更為複雜的系統可以尋址到從機1和從機2，而不會尋址到從機0：

從機 0

SADDR = 1110 0000

SADEN = 1111 1001

地址 = 1100 0XX0

從機 1

SADDR = 1110 0000

SADEN = 1111 1010

地址 = 1110 0X0X

從機 2

SADDR = 1110 0000

SADEN = 1111 1100

地址 = 1110 00XX

上面的例子中，3個從機的低3位地址不一樣，從機0要求第0位必須為0，1110 0110可以唯一尋址從機0；從機1要求第1位必須為0，1110 0101可以唯一尋址從機1；從機2要求第2位必須為0，它的唯一地址是1110 0011。為了尋址到從機0和從機1而不會尋址到從機2，可以使用地址1110 0100，因為這個地址第2位是1。

每個從機的廣播地址SADDR和SADEN的邏輯或，0按不需關心處理。大部分情況下，使用FF作為廣播地址。

複位後，SADDR（SFR地址23Fh）和SADEN（SFR地址23Eh）值均為0，這樣可以接收所有地址的資訊，也就有效的禁用了自動地址識別模式，從而使該處理器運行於標準80C51的UART下。

11.6 串行口工作模式

串行通信介面有4種工作模式，可通過軟體編程對SCON中的SM0、SM1的設置進行選擇。其中模式1、模式2、模式3為非同步通信，每個發送和接收的字元都帶有1個啟動位和1個停止位。在模式0中，串行口被作為1個簡單的移位寄存器應用。

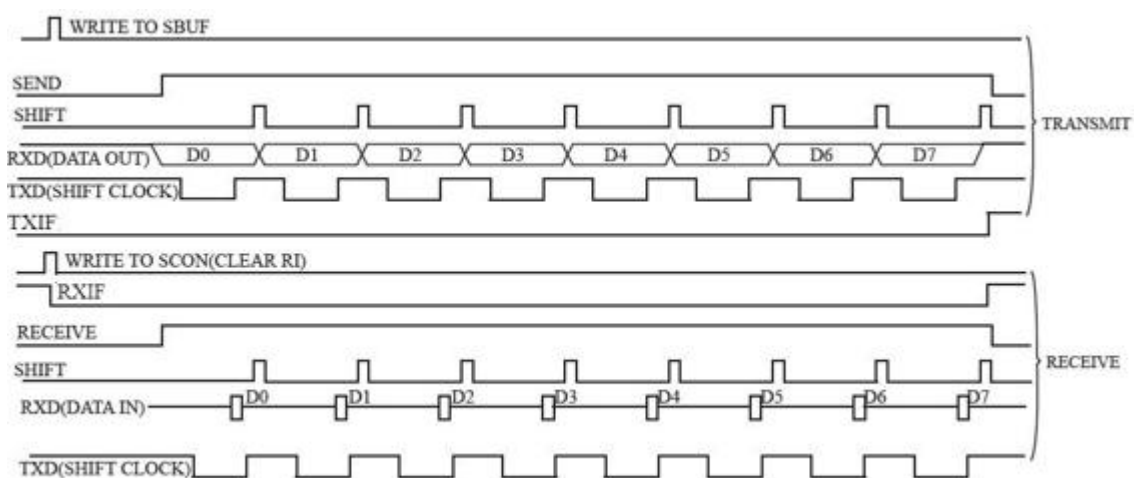
11.6.1 串行口工作模式 0：同步移位寄存器

在模式0狀態，串行通信介面工作在同步移位寄存器模式，當串行口模式0的通信速度設置位UARTM0/AUXR.5 = 0時，其串列傳輸速率固定為 $F_{cpu}/12$ 。當串行口模式0的通信速度設置位UARTM0/AUXR.5 = 1時，其串列傳輸速率固定為 $F_{cpu}/2$ 。串行口數據由RxD端輸入，同步移位脈衝（SHIFTCLOCK）由TxD輸出，發送、接收的是8位數據，低位在先。

模式0的發送過程：當主機執行將數據寫入發送緩衝器SBUF指令時啟動發送，串行口即將8位數據以 $F_{cpu}/12$ 或 $F_{cpu}/2$ （由UARTM0/AUXR.5確定是12分頻還是2分頻）的串列傳輸速率從RxD管腳輸出（從低位到高位），發送完中斷標誌TXIF置"1"，TxD管腳輸出同步移位脈衝（SHIFTCLOCK）。波形如下圖中“發送”所示。

當寫信號有效後，相隔一個時鐘，發送控制端SEND有效（高電平），允許RxD發送數據，同時允許TxD輸出同步移位脈衝。一幀（8位）數據發送完畢時，各控制端均恢復原狀態，只有TXIF保持高電平，呈中斷申請狀態。在再次發送數據前，必須用軟體將TXIF清0。

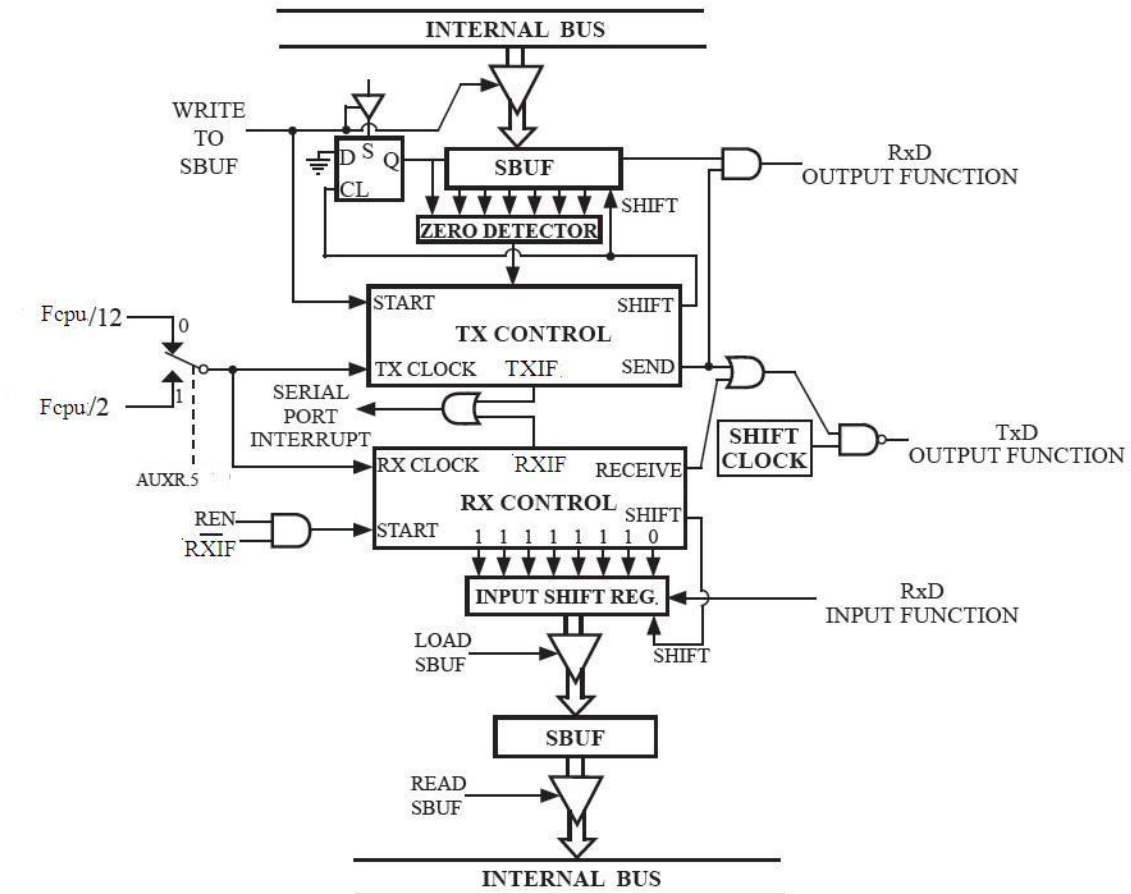
模式0接收過程：模式0接收時，複位接收中斷請求標誌RXIF，即RXIF=0，置位允許接收控制位REN=1時啟動串行模式0接收過程。啟動接收過程後，RxD為串行輸入端，TxD為同步脈衝輸出端。串行接收的串列傳輸速率為 $F_{cpu}/12$ 或 $F_{cpu}/2$ （由UARTM0/AUXR.5確定是12分頻還是2分頻）。其時序圖如下圖中所示：



當接收完成一幀數據（8位）後，控制信號複位，中斷標誌RXIF被置"1"，呈中斷申請狀態。當再次接

收時，必須通過軟體將RXIF清0。工作於模式0時，必須清0多機通信控制位SM2，使不影響TB8位和RB8位。由於串列傳輸速率固定為 $F_{cpu}/12$ 或 $F_{cpu}/2$ ，無需定時器提供，直接由單片機的時鐘作為同步移位脈衝。

串行口工作模式0的示意圖如下圖所示：

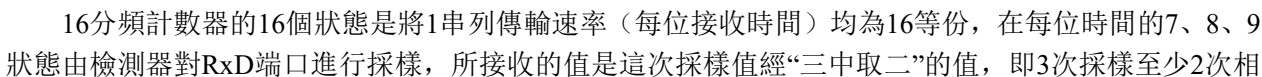


由示意圖中可見，由TX和RX控制單元分別產生中斷請求信號並置位TXIF=1或RXIF=1，經“或門”送主機請求中斷，所以主機回應中斷後必須軟體判別是TXIF還是RXIF請求中斷，必須軟體清0中斷請求標誌位TXIF或RXIF。

11.6.2 串行口工作模式1：8位UART，串列傳輸速率可變

當軟體設置SCON的SM0、SM1為“01”時，串行口則以模式1工作。此模式為8位UART格式，一幀資訊為10位：1位起始位，8位數據位(低位在先)和1位停止位。串列傳輸速率可變，即可根據需要進行設置。TxD/P3.1為發送資訊，Rx/D/P3.0為接收端接收資訊，串行口為全雙工接受/發送串行口。

下圖為串行模式1的功能結構示意圖及接收/發送時序圖：



同的值，以此消除干擾影響，提高可靠性。在起始位，如果接收到的值不為“0”（低電平），則起始位無效，複位接收電路，並重新檢測“1”→“0”的跳變。如果接收到的起始位有效，則將它輸入移位寄存器，並接收本幀的其餘資訊。

接收的數據從接收移位寄存器的右邊移入，已裝入的1FFH向左邊移出，當起始位“0”移到移位寄存器的最左邊時，使RX控制器作最後一次移位，完成一幀的接收。若同時滿足以下兩個條件：

- RXIF=0;
- SM2=0或接收到的停止位為1。

則接收到的數據有效，實現裝載入SBUF，停止位進入RB8，置位RXIF，即RXIF=1，向主機請求中斷，若上述兩條件不能同時滿足，則接收到的數據作廢並丟失，無論條件滿足與否，接收器重又檢測RxD端口上的“1”→“0”的跳變，繼續下一幀的接收。接收有效，在回應中斷後，必須由軟體清0，即RXIF=0。通常情況下，串行通信工作於模式1時，SM2設置為“0”。

串行通信模式1的串列傳輸速率是可變的，可變的波特由定時器/計數器1或獨立串列傳輸速率發生器產生。

模式1的串列傳輸速率為：

串行通信模式1的串列傳輸速率=2SMOD/32×(定時器/計數器1溢出率或BRT獨立串列傳輸速率發生器溢出率)。

當BRTx12 = 0時， BRT獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/12/(256 - BRT)$;

當BRTx12 = 1時， BRT獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/(256 - BRT)$ 。

11.6.3 串行口工作模式 2：9 位 UART，串列傳輸速率固定

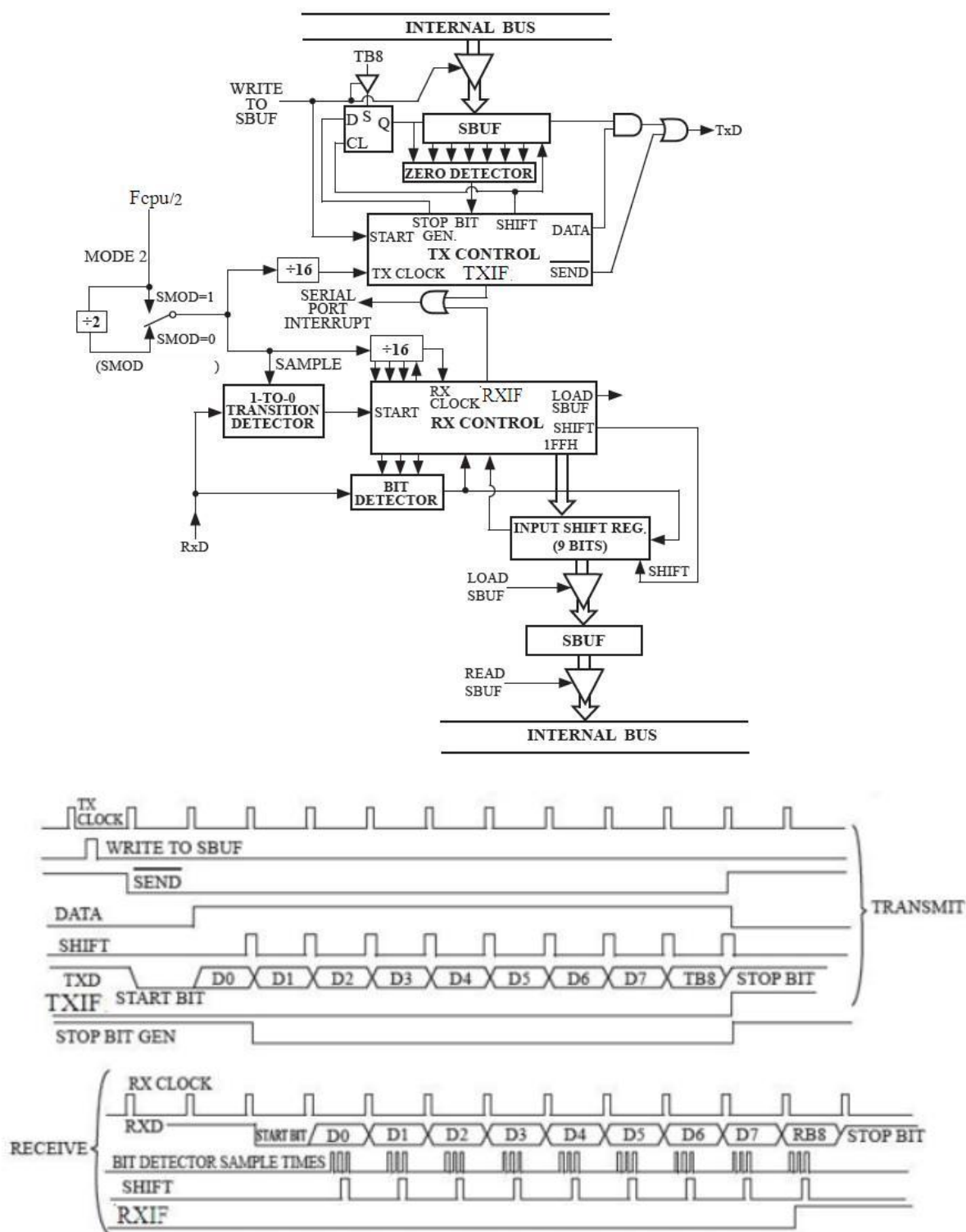
當SM0、SM1兩位為10時，串行口工作在模式2。串行口工作模式2為9位數據非同步通信UART模式，其一幀的資訊由11 位組成：1位起始位，8位數據位(低位在先)，1位可編程位(第9位數據)和1位停止位。發送時可編程位(第9位數據)由SCON中的TB8提供，可軟體設置為1或0，或者將奇/偶校驗值裝入TB8（TB8既可作為多機通信中的地址數據標誌位，又可作為數據的奇偶校驗位）。接收時第9位數據裝入SCON的RB8。TxD為發送端口，RxD為接收端口，以全雙工模式進行接收/發送。

模式2的串列傳輸速率為：

串行通信模式2串列傳輸速率=2SMOD/64×(Fcpu)。

上述串列傳輸速率可通過軟體對AUXR中的SMOD位進行設置，當SMOD=1時，選擇1/32(Fcpu)；當SMOD=0時，選擇1/64(Fcpu)，故而稱SMOD為串列傳輸速率加倍位。可見，模式2的串列傳輸速率基本上是固定的。

下圖為串行通信模式2的功能結構示意圖及其接收/發送時序圖：



由上圖可知，模式2和模式1相比，除串列傳輸速率發生源略有不同，發送時由TB8提供給移位寄存器第9數據位不同外，其餘功能結構均基本相同，其接收/發送操作過程及時序也基本相同。

當接收器接收完一幀資訊後必須同時滿足下列條件：

- RXIF=0;
- SM2=0或者SM2=1，並且接收到的第9數據位RB8=1。

當上述兩條件同時滿足時，才將接收到的移位寄存器的數據裝入SBUF和RB8中，並置位RXIF=1，向主機請求中斷處理。如果上述條件有一個不滿足，則剛接收到移位寄存器中的數據無效而丟失，也不置

位RXIF。無論上述條件滿足與否，接收器又重新開始檢測RxD輸入端口的跳變資訊，接收下一幀的輸入資訊。

在模式2中，接收到的停止位與SBUF、RB8和RXIF無關。

通過軟體對SCON中的SM2、TB8的設置以及通信協議的約定，為多機通信提供了方便。

11.6.4 串行口工作模式3：9位UART，串列傳輸速率可變

當SM0、SM1兩位為“11”時，串行口工作在模式3。串行通信模式3為9位數據非同步通信UART模式，其一幀的資訊由11位組成：1位起始位，8位數據位(低位在先)，1位可編程位(第9位數據)和1位停止位。發送時可編程位(第9位數據)由SCON中的TB8提供，可軟體設置為1或0，裝入TB8(TB8既可作為多機通信中的地址數據標誌位，又可作為數據的奇偶校驗位)。接收時第9位數據裝入SCON的RB8。TxD為發送端口，RxD為接收端口，以全雙工模式進行接收/發送。

模式3的串列傳輸速率為：

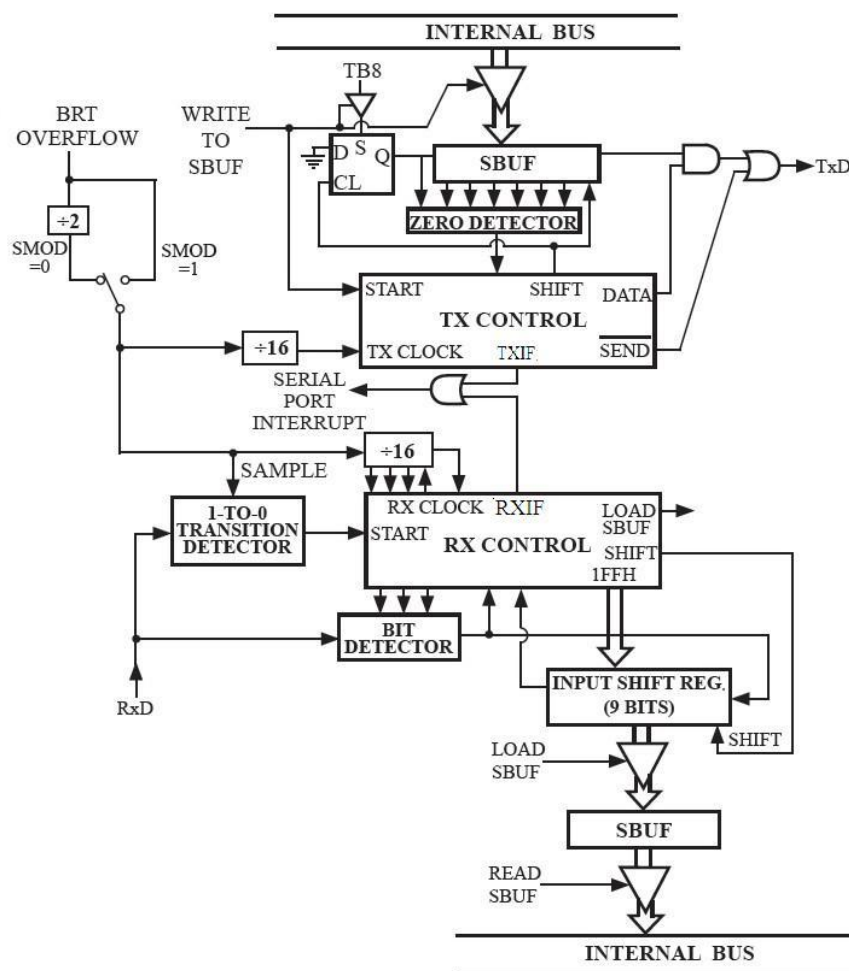
串行通信模式3串列傳輸速率=2SMOD/32×(BRT獨立串列傳輸速率發生器的溢出率)。

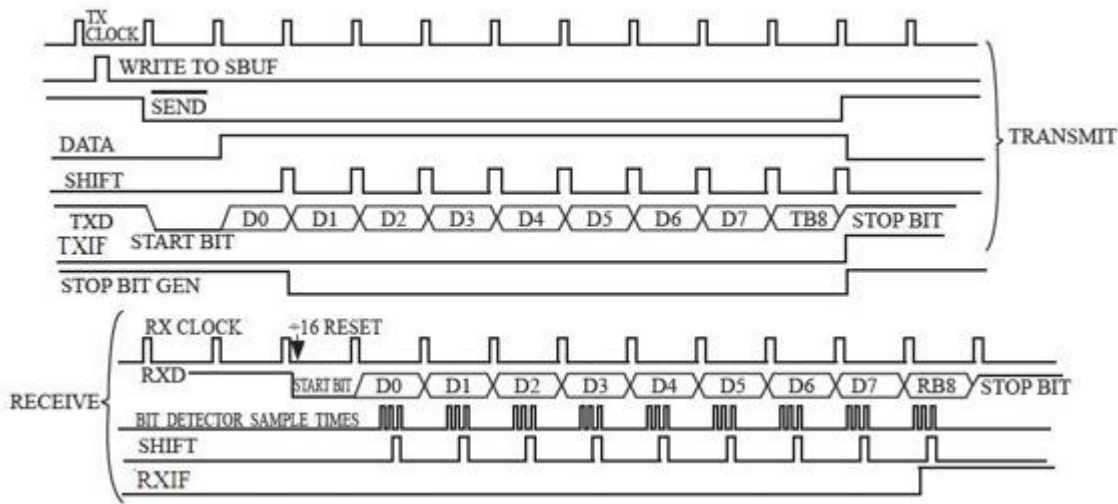
當BRTx12 = 0時， BRT獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/12/(256 - BRT)$;

當BRTx12 = 1時， BRT獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/(256 - BRT)$ 。

可見，模式3和模式1一樣，其串列傳輸速率可通過軟體對定時器/計數器1或獨立串列傳輸速率發生器的設置進行串列傳輸速率的選擇，是可變的。

下圖為串行口工作模式3的功能結構示意圖及其接收/發送時序圖：





由下圖可知，模式3和模式1相比，除發送時由TB8提供給移位寄存器第9數據位不同外，其餘功能結構均基本相同，其接收、發送操作過程及時序也基本相同。

當接收器接收完一幀資訊後必須同時滿足下列條件：

- RXIF=0；
- SM2=0或者SM2=1，並且接收到的第9數據位RB8=1。

當上述兩條件同時滿足時，才將接收到的移位寄存器的數據裝入SBUF和RB8中，並置位RXIF=1，向主機請求中斷處理。如果上述條件有一個不滿足，則剛接收到移位寄存器中的數據無效而丟失，也不置位RXIF。無論上述條件滿足與否，接收器又重新開始檢測RxD輸入端口的跳變資訊，接收下一幀的輸入資訊。在模式3中，接收到的停止位與SBUF、RB8和RXIF無關。通過軟體對SCON中的SM2、TB8的設置以及通信協議的約定，為多機通信提供了方便。

11.7 串口通信中波特率的設置

此系列單片機串行通信的串列傳輸速率隨所選工作模式的不同而異，對於工作模式0和模式2，其串列傳輸速率與系統時鐘頻率F_{cpu}和AUXR中的串列傳輸速率選擇位SMOD有關，而模式1和模式3的串列傳輸速率除與F_{cpu}和AUXR位有關外，還與定時器/計數器1或BRT獨立串列傳輸速率發生器設置有關。通過對定時器/計數器1或BRT獨立串列傳輸速率發生器的設置，可選擇不同的串列傳輸速率，所以這種串列傳輸速率是可變的。

串行通信模式0，其串列傳輸速率與系統時鐘頻率SYSclk有關。

當模式0的通信速度設置位UARTM0/AUXR.5 = 0時，其串列傳輸速率 = F_{cpu}/12。

當模式0的通信速度設置位UARTM0/AUXR.5 = 1時，其串列傳輸速率 = F_{cpu}/2。

一旦F_{cpu}選定且UART_M0x6/AUXR.5設置好，則串行通信工作模式0的串列傳輸速率固定不變。

串行通信工作模式2，其串列傳輸速率除與SYSclk有關外，還與SMOD位有關。

其基本運算式為：串行通信模式2串列傳輸速率=2SMOD/64×(F_{cpu}系統工作時鐘頻率)

當SMOD=1時，串列傳輸速率=2/64(F_{cpu})=1/32(F_{cpu})；

當SMOD=0時，串列傳輸速率=1/64(F_{cpu})。

當F_{cpu}選定後，通過軟體設置AUXR中的SMOD位，可選擇兩種串列傳輸速率。所以，這種模式的串列傳輸速率基本固定。

串行通信模式1和3，其串列傳輸速率是可變的：

模式1、3串列傳輸速率=2SMOD/32×(BRT獨立串列傳輸速率發生器的溢出率)

當BRTx12 = 0時， BRT獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/12/(256 - BRT)$;

當BRTx12 = 1時， BRT獨立串列傳輸速率發生器的溢出率 = $F_{cpu}/(256 - BRT)$ 。

通過對定時器/計數器1和BRT獨立串列傳輸速率發生器的設置，可靈活地選擇不同的串列傳輸速率。在實際應用中多半選用串行模式1或串行模式3。顯然，為選擇串列傳輸速率，關鍵在於定時器/計數器1和BRT獨立串列傳輸速率發生器的溢出率的計算。SMOD的選擇，只需根據需要執行下列指令就可實現SMOD=0或1：

```
BCF  AUXR, 1      ; 使SMOD=0
BSF  AUXR, 1      ; 使SMOD=1
```

SMOD只佔用電源控制寄存器AUXR的BIT2，其他各位的具體設置應根據實際情況而定。

當用戶選擇BRT獨立串列傳輸速率發生器作串列傳輸速率發生器時，為選擇串列傳輸速率，關鍵在於獨立串列傳輸速率發生器的溢出率。當用戶選擇BRT獨立串列傳輸速率發生器作串列傳輸速率發生器時，定時器/計數器1可以釋放出來作為定時器/計數器/時鐘輸出使用。

用戶在程式中如何具體使用串口1和獨立串列傳輸速率發生器BRT：

- 1.設置串口的工作模式，SCON寄存器中的SM0和SM1兩位決定了串口1的4種工作模式；
- 2.設置串口的串列傳輸速率,使用獨立串列傳輸速率發生器寄存器和相應的位：BRT獨立串列傳輸速率發生器寄存器，BRTx12 位，SMOD位；
- 3.啟動獨立串列傳輸速率發生器，讓BRTR位為1，BRT獨立串列傳輸速率發生器寄存器就立即開始計數；
- 4.設置串口的中斷，及打開中斷相應的控制位是：PS，PSH，ES，EA；
- 5.如要串口接收，將REN置1即可。

如要串口發送，將數據送入SBUF即可，接收完成標誌RXIF，發送完成標誌TXIF，要由軟體清0。

當串口工作在模式1和模式3時,計算相應的串列傳輸速率需要設置的重裝載數,結果送入BRT寄存器計算自動重裝數 RELOAD (SMOD = 0, SMOD是AUXR特殊功能寄存器的最高位):

- 1) . 計算 RELOAD (以下是 SMOD = 0 時的計算公式):

a) 12T 模式的計算公式: $RELOAD = 256 - INT(F_{cpu}/Baud0/32/12 + 0.5)$

b) 1T 模式的計算公式: $RELOAD = 256 - INT(F_{cpu}/Baud0/32 + 0.5)$

計算出的RELOAD 數直接送BRT 寄存器。

式中: INT()表示取整運算即舍去小數, 在式中加 0.5 可以達到四捨五入的目的。

SYSclk = 晶振頻率

Baud0 = 標準串列傳輸速率

- 2) . 計算用 RELOAD 產生的串列傳輸速率:

a) $Baud = F_{cpu}/(256 - RELOAD)/32/12$ 12T 模式

b) $Baud = F_{cpu}/(256 - RELOAD)/32$ 1T 模式

- 3) . 計算誤差:

$error = (Baud - Baud0)/Baud0 * 100\%$

- 4) . 如果誤差絕對值 > 3%, 要更換串列傳輸速率或者更換晶體頻率, 重複步驟 1-4。

例: $F_{cpu} = 22.1184MHz$, $Baud0 = 57600$ (12T 模式)

1. $RELOAD = 256 - INT(22118400/57600/32/12 + 0.5)$
 $= 256 - INT(1.5)$

$$= 256 - 1$$

$$= 255$$

$$= 0FFH$$

$$2. \text{Baud} = 22118400 / (256 - 255) / 32 / 12 = 57600$$

3. 誤差等於零

12 軟體LCD驅動

HC16P122A1/B1 中內置軟體 LCD 驅動，所有端口可通過 COMxEN[7:0]使能或關閉 COM 口功能。

12.1 相關寄存器

LCDCON 寄存器

2B0h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDCON	LCDEN	RLCD1	RLCD0	FRAM	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR的值	0	0	0	0	-	-	-	-

Bit[7] **LCDEN**: 軟體 LCD 模組使能控制位

0 = 禁止

1 = 使能

Bit[6:5] **RLCD**[1:0]: 軟體 LCD 電阻選擇位

00 = 600K

01 = 300K

10 = 100K

11 = 50K

Bit[4] **FRAM**: Frame0 或 Frame1 輸出使能位

0 = Frame0

1 = Frame1

COM 口使能位

2B1h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
COMAEN	COMAEN7	COMAEN6	COMAEN5	COMAEN4	COMAEN3	COMAEN2	COMAEN1	COMAEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

2B2h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
COMBEN	COMBEN7	COMBEN6	-	COMBEN4	COMBEN3	COMBEN2	COMBEN1	COMBEN0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	-	0	0	0	0	0

Bit[7:0] **COMxEN**[7:0] COM 功能使能位

0 = 禁止對應 COMxENy 的 COM 功能，當 IO 使用

1 = 使能對應 COMxENy 的 COM 功能

說明：請根據實際需要選擇適當個數 COM 功能。

12.2 軟體 LCD 操作說明

1 設置LCD總使能，LCDEN=1；這是總的使能，打開電阻分壓電路；

- 2 設置驅動能力，選擇不同的電阻分壓RLCD[1:0];
- 3 對IO口使能COM或SEG, COMXEN=1, SEGXEN=1;這是具體設置某個IO的狀態，使能LCD的模擬通道;
- 4 設定frame0，用於確定點亮和非點亮電平;
- 5 設置定時開始，設置COM[3:0]=0001，設置SEG[19:0]=XXX，等待定時結束;
- 6 設置定時開始，設置COM[3:0]=0010，設置SEG[19:0]=XXX，等待定時結束;
- 7 設置定時開始，設置COM[3:0]=0100，設置SEG[19:0]=XXX，等待定時結束;
- 8 設置定時開始，設置COM[3:0]=1000，設置SEG[19:0]=XXX，等待定時結束;
- 9 設定frame0，用於確定點亮和非點亮電平;
- 10設置定時開始，設置COM[3:0]=0001，設置SEG[19:0]=XXX，等待定時結束;
- 11設置定時開始，設置COM[3:0]=0010，設置SEG[19:0]=XXX，等待定時結束;
- 12設置定時開始，設置COM[3:0]=0100，設置SEG[19:0]=XXX，等待定時結束;
- 13設置定時開始，設置COM[3:0]=1000，設置SEG[19:0]=XXX，等待定時結束。

注意:

- 1 驅動加速：在每次切換COM時，設置最大的驅動能力RLCD [1:0]=11，等一段時間如50us，重新設置最小的驅動能力ISEL[1:0]=00;
- 2 不用的SEG口，可以不設置，SEG數不限定於20個;
- 3 不用的 COM 口，軟體直接跳過，不需要延時等待，COM 數不限定於 4 個。

13 指令表

助記符 運算元	說明	週期數	受影響狀態位
ADDWF f,d	W 和 f 相加	1	C, DC, Z
ADDLW k	將立即數和 W 相加	1	C, DC, Z
SUBWF f,d	f 減去 W	1	C, DC, Z
SUBLW k	立即數減去 W	1	C, DC, Z
DAW -	W 寄存器值進行 BCD 調整	1	C, DC
ANDWF f,d	W 和 f 作邏輯與運算	1	Z
ANDLW k	立即數和 W 作邏輯與運算	1	Z
IORWF f,d	W 和 f 作邏輯或運算	1	Z
IORLW k	立即數和 W 作邏輯或運算	1	Z
XORWF f,d	W 和 f 作邏輯異或運算	1	Z
XORLW k	立即數和 W 作邏輯異或運算	1	Z
COMF f,d	f 取反	1	Z
CLRW -	將 W 清零	1	Z
CLRF f	將 f 清零	1	Z
INCF f,d	f 加 1	1	Z
INCFSZ f,d	f 加 1, 為 0 則跳過	1(2)	-
DECF f,d	f 減 1	1	Z
DECFSZ f,d	f 減 1, 為 0 則跳過	1(2)	-
BCF f,d	將 f 中的 d 位清 0	1	-
BSF f,d	將 f 中的 d 位置 1	1	-
BTFSC f,d	檢測 f 中的 d 位, 為 0 則跳過	1(2)	-
BTFSS f,d	檢測 f 中的 d 位, 為 1 則跳過	1(2)	-
MOVWF f	將 W 的內容傳送到 f	1	-
MOVF f,d	將 f 的內容送到目標寄存器	1	Z
MOVLW k	將立即數 k 傳送到 W	1	-
RLF f,d	對 f 執行帶進位的迴圈左移	1	C
RRF f,d	對 f 執行帶進位的迴圈右移	1	C
SWAPF f,d	將 f 的兩個半位元組進行交換	1	-
CALL k	調用副程式	2	-
GOTO k	無條件跳轉	2	-
RETFIE -	從中斷返回	2	GIE
RETURN -	從副程式返回	2	-
RETLW k	返回時將立即數傳送到 W	2	-
CLRWDI -	清零看門狗定時器	1	TO, PD
SLEEP -	進入待機模式	1	TO, PD
NOP -	空操作	1	-

14 電性參數

14.1 極限參數

儲存溫度.....	-50℃~125℃
工作溫度.....	-40℃~85℃
電源供應電壓.....	VSS-0.3V~VSS+6.0V
端口輸入電壓.....	VSS-0.3V~VDD+0.3V

14.2 直流特性

符號	參數	測試條件		最小值	典型值	最大值	單位
		VDD	條件（常溫 25℃）				
VDD	工作電壓	—	Fsys = 8MHz, 2T	1.8	—	5.5	V
		—	Fsys = 16MHz, 2T	2.6	—	5.5	V
IDD1	工作電流	3V	Fsys = 16MHz, 4T, 高頻模式, WDT 禁止, 無負載	—	1.70	—	mA
		5V		—	3.00	—	mA
IDD2	工作電流	3V	Fsys = 8MHz, 4T, 高頻模式, WDT 禁止, 無負載	—	1.20	—	mA
		5V		—	2.20	—	mA
IDD3	工作電流	3V	Fsys = 32KHz, 4T, 低頻模式, WDT 禁止, 無負載	—	3.0	—	μA
		5V		—	10.0	—	μA
IDD4	工作電流	3V	Fsys = 32KHz, 4T, 綠色模式, WDT 禁止, 無負載	—	2.0	—	μA
		5V		—	7.0	—	μA
ISB4	靜態電流	3V	休眠模式, WDT 禁止, 無負載	—	—	1	μA
		5V		—	—	1	μA
VIL1	輸入低電平	—	輸入口	VSS	—	0.3VDD	V
VIH1	輸入高電平	—	輸入口	0.7VDD	—	VDD	V
VIL2	輸入低電平	—	施密特輸入口	VSS	—	0.2VDD	V
VIH2	輸入高電平	—	施密特輸入口	0.8VDD	—	VDD	V
VBOR1	低電壓複位 1.5V	—	—	1.4	1.5	1.6	V
VBOR2	低電壓複位 1.8V	—	—	1.7	1.8	1.9	V
VBOR3	低電壓複位 2.0V	—	—	1.9	2.0	2.1	V
VBOR4	低電壓複位 2.2V	—	—	2.1	2.2	2.3	V
VBOR5	低電壓複位 2.4V	—	—	2.3	2.4	2.5	V
VBOR6	低電壓複位 2.7V	—	—	2.6	2.7	2.8	V
VBOR7	低電壓複位	—	—	3.5	3.6	3.7	V

	3.6V						
V _{LVD0}	低電壓標誌	—	—	2.3	2.4	2.5	V
V _{LVD1}	低電壓標誌	—	—	3.5	3.6	3.7	V
I _{OL1}	(DREN _{xn} =00)	3V	VOL=VSS+0.6V	—	—	—	mA
		5V		—	50	—	mA
I _{OL2}	(DREN _{xn} =01)	3V	VOL=VSS+0.6V	—	—	—	mA
		5V		—	10	—	mA
I _{OL3}	(DREN _{xn} =10)	3V	VOL=VSS+0.6V	—	—	—	mA
		5V		—	25	—	mA
I _{OL4}	(DREN _{xn} =11)	3V	VOL=VSS+0.6V	—	—	—	
		5V		—	5	—	
I _{OH1}	(DREN _{xn} =00)	3V	VOH= VDD-0.6V			—	mA
		5V		—	30	—	mA
I _{OH2}	(DREN _{xn} =01)	3V	VOH= VDD-0.6V			—	mA
		5V		—	8	—	mA
I _{OH3}	(DREN _{xn} =10)	3V	VOH=VDD-0.6V	—		—	mA
		5V		—	20	—	mA
I _{OH4}	(DREN _{xn} =11)	3V	VOH=VDD-0.6V	—		—	mA
		5V		—	3	—	mA
R _{PH1}	內部上拉電阻	5V	可編程上拉電阻(PA2 除外)	5	10	15	kΩ
R _{PD1}	內部下拉電阻	5V	可編程下拉電阻 (PA2, PB5 除外)	5	10	15	kΩ
V _{AD}	ADC 輸入電壓	—	—	VSS	—	V _{REF}	V
DNL	差分非線性誤差	5V	AD 時鐘頻率 2MHz	—	±1	—	LSB
INL	積分非線性誤差	5V	AD 時鐘頻率 2MHz	—	±1	—	LSB
I _{ADC}	ADC 工作電流	3V	AD 時鐘頻率 2MHz	—	0.3	—	mA
		5V		—	0.5	—	mA

14.3 交流特性

符號	參數	測試條件		最小值	典型值	最大值	單位
		VDD	條件 (常溫 25°C)				
FRCH	高頻內部 RC 振盪器	5V	—	31.68	32	32.32	MHz
FRCH	高頻內部 RC 振盪器		2.0-5.5V	31.36	32	32.64	MHz
FRCL	低頻內部 RC 振盪器	5V	—	16	32	48	KHz
FOSH	外部高頻晶振	—	2.0~5.5V	4	—	20	MHz
FOSL	外部低頻晶振	—	2.0~5.5V	—	32.768	—	KHz
TVDD	VDD 上升時間	5V	—	—	—	100	ms
TBOR	欠壓複位回應時間	5V	—	100	—	—	ns
TWDT	看門狗溢出時間	5V	使用預分頻 1:1	—	18	—	ms
			不使用預分頻器	—	72	—	ms

TMCLRb	複位脈衝時間	5V	—	200	—	—	us
--------	--------	----	---	-----	---	---	----

15 開發工具

15.1 OTP 燒錄器（HC-PM18 5.0）

- PM18 5.0: 支持 HC18 系列 MCU 大批量的脫機燒錄。

注：

詳情請參考 HC-PM18 用戶手冊。

15.2 HC-IDE

Holychip 8 位單片機的集成開發環境 HC-IDE 包括編譯器、HC-PM18 下載燒錄軟體。

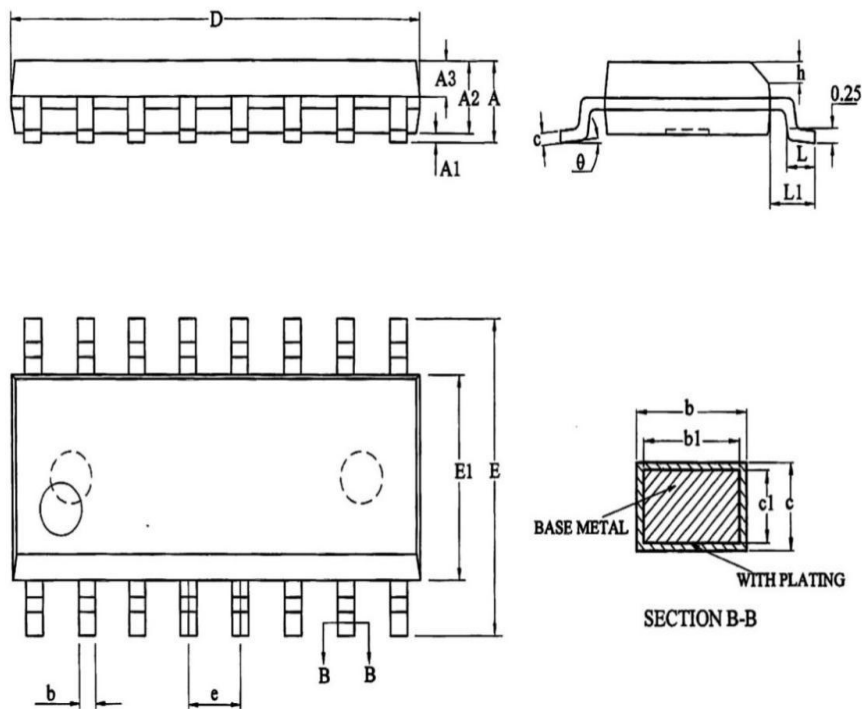
- HC-PM18: V5.0.3.0
- HC-IDE: V3.0.4.0 (支持彙編/C 語言)

注：

- 1、詳情請參考 HC-IDE 用戶手冊。
- 2、IDE 更新請關注芯聖官網：<http://www.holychip.cn/>

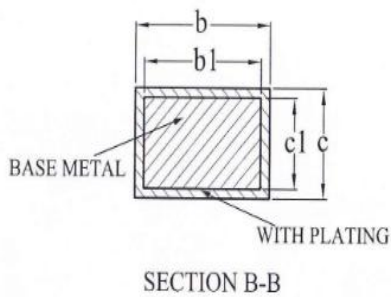
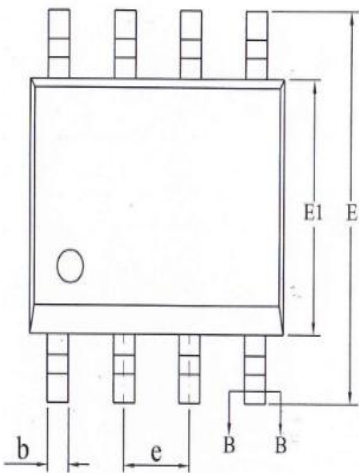
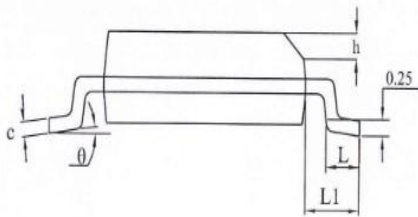
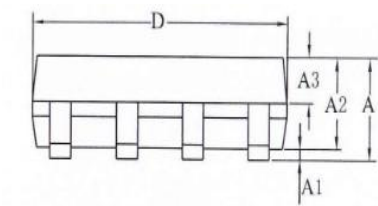
16 封裝資訊

16.1 SOP16



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.05	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.48
b1	0.38	0.41	0.43
c	0.21	—	0.26
c1	0.19	0.20	0.21
D	9.70	9.90	10.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05BSC		
θ	0	—	8°

16.2 SOP8



△1
△1
△1
△1
△1
△1

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	4.80	4.90	5.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

17 修改記錄

版本	日期	描述
Ver1.00	2020-11	第一版

HOLYCHIP 公司保留對以下所有產品在可靠性、功能和設計方面的改進作進一步說明的權利。

HOLYCHIP 不承擔由本手冊所涉及的产品或電路的運用和使用所引起的任何責任，HOLYCHIP 的产品不是專門設計來應用於外科植入、生命維持和任何 HOLYCHIP 產品產生的故障會對個體造成傷害甚至死亡的領域。如果將 HOLYCHIP 的产品用於上述領域，即使這些是由 HOLYCHIP 在产品設計和製造上的疏忽引起的，用戶應賠償所有費用、損失、合理的人身傷害或死亡所直接或間接所產生的律師費用，並且用戶保證 HOLYCHIP 及其雇員、子公司、分支機構和銷售商與上述事宜無關。

芯聖電子

2020年11月