

# YS65F804

## 数据手册（版本 V1.3）

### 8 位 MTP CMOS 单片机

## 1.0 器件概述

### ◆ 存储器:

MTP 空间: 2K\*14 位。可经受 1000 次写操作。  
SRAM 空间: 128 字节。  
CS SRAM: 32 字节。

### ◆ 8 级深硬件堆栈

### ◆ I/O 引脚配置

具有独立方向控制的 6 个 I/O 引脚: PA 口 6 个

PA3: 外部复位, 当设定为输出时为开漏输出模式 (open drain)。

PA 端口独立的可编程弱上/下拉。

### ◆ 定时器

Timer0: 带 8 位可编程预分频器的 8 位定时器/计数器。

增强型 Timer1:

- 带有预分频器的 16 位定时器/计数器。

### ◆ 中断源

- Timer0、Timer1 中断。  
- PA 电平变化中断。  
- 触摸中断。

### ◆ PWM 模块

- 2 路 10bit PWM

### ◆ 5 通道 Touch KEY（最大同时支持 4 通道）

可通过软件单独对每路调节灵敏度

### ◆ 时钟系统

内部高速时钟: 16MHz RC (Fcpu 支持 16Mhz、8MHz、4MHz、2MHz、1MHz、500KHz、250KHz)。

内部低速时钟: RC 振荡器 31KHz。

### ◆ 高性能的 RISC CPU

仅需学习 35 条指令。  
除跳转指令外的所有指令都是单周期的。  
直接和间接寻址模式。

### ◆ 特殊特性

- 高精度内部振荡器, 出厂精度校准为  $\pm 1\%$ 。  
全温全压偏差小于  $\pm 2\%$
- 可用软件选择的频率范围为 250kHz 到 16MHz。
- 软件可选的 31kHz 内部振荡器。
- 节能的休眠模式。
- 宽工作电压范围 (2.5V 到 5.5V)。
- 工业级温度范围。
- 上电复位 (Power-onReset, POR)。
- 带软件控制选择的 LVR 低电压侦测选择(侦测电压有 2.5V、3.0V、3.6V 可选)。
- 带片上振荡器 (振荡器频率可由软件选择, 当预分频比最大时其标称值为 268 秒)并且可软件使能的增强型低电流看门狗定时器 (WatchdogTimer, WDT)。
- 带上拉的主复位, 可复用为输入引脚。
- 可编程代码保护。

### ◆ 低功耗特性

- 待机电流:
- 电压为 2.5V 时, 典型值 500nA。
- 工作电流:
- 频率为 8MHz、电压为 2.5V 时, 典型值为 500uA。
  - 频率为 31kHz、电压为 2.5V 时, 典型值为 120uA。
- 看门狗定时器电流:
- 电压为 2.5V 时, 典型值为 1uA。

### ◆ 通过三个引脚(PA0, PA1, PA3)实现在线串行编程和调试。

器件	ROM	RAM		I/O	Timer	PWM	唤醒功能 引脚数目	Touch	封装
	MTP (word)	SRAM(word)	CS RAM(word)						
YS65F804	2048	128	32	6	2	2	5	4	SOT23-6、DFN6、 SOP8、DFN8

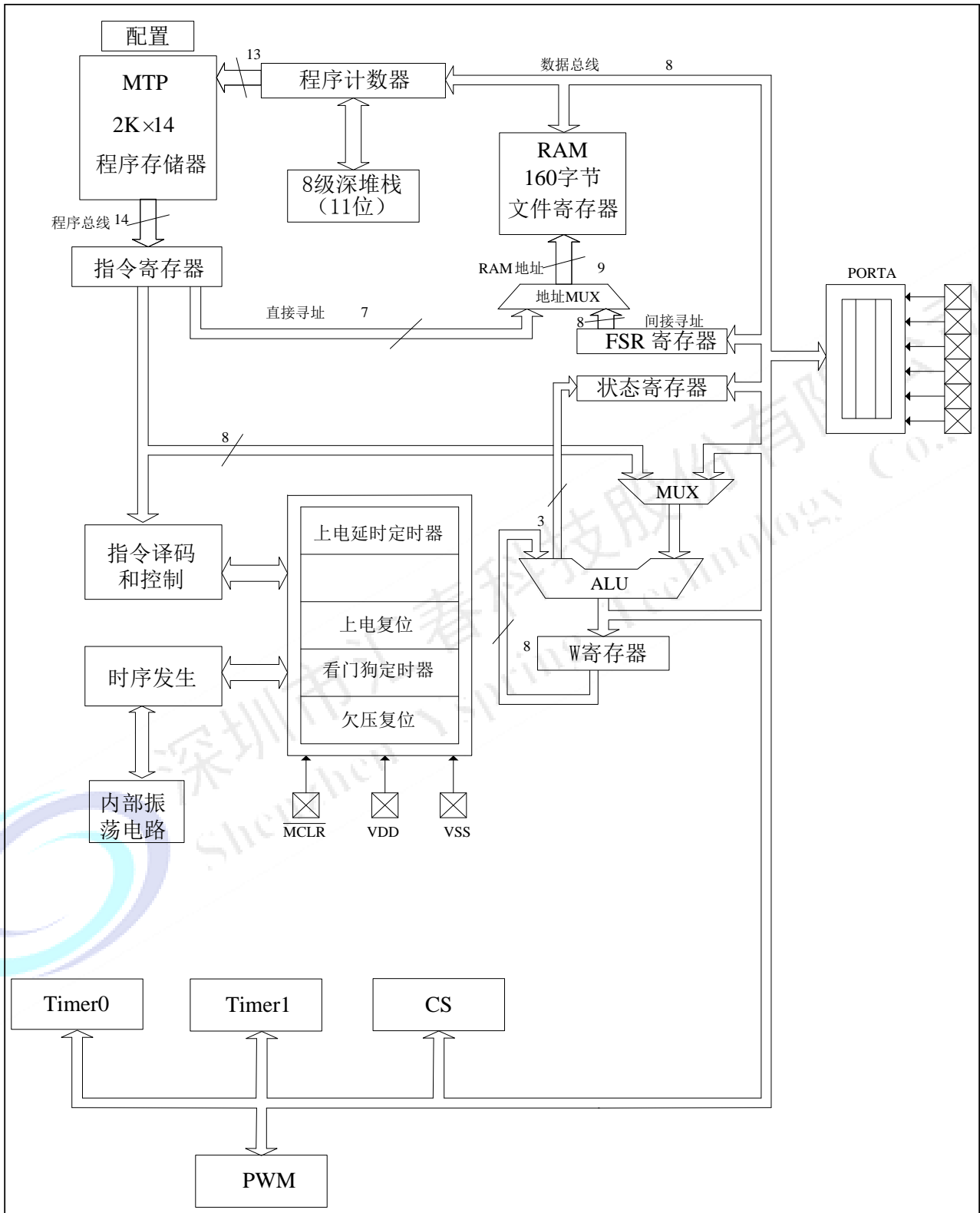
<b>1.0 器件概述</b> .....	<b>1</b>
1.1 系统结构图 .....	4
1.2 封装脚位图 .....	5
1.3 引脚说明 .....	6
<b>2.0 存储器构成</b> .....	<b>7</b>
2.1 程序存储器构成 .....	7
2.2 数据存储器构成 .....	8
2.3 PCL 和 PCLATH .....	16
2.4 间接寻址、IAR 和 MSR 寄存器 .....	17
2.5 看门狗定时器 (WDT) .....	18
<b>3.0 复位</b> .....	<b>21</b>
3.1 概述 .....	21
3.2 上电复位 .....	21
3.3 MCLR .....	22
3.4 欠压复位 .....	22
3.5 上电延时定时器 (PWRT) .....	23
3.6 低电压检测 (LVD) .....	24
3.7 电源控制 (PSTA) 寄存器 .....	24
<b>4.0 系统时钟</b> .....	<b>25</b>
4.1 概述 .....	25
4.2 振荡器控制 .....	25
4.3 掉电模式 (休眠) .....	28
<b>5.0 中断</b> .....	<b>30</b>
5.1 中断特殊功能寄存器 .....	31
5.2 TIMER0 中断 .....	32
5.3 PORTA 电平变化中断 .....	32
5.4 中断的现场保护 .....	32
<b>6.0 I/O 端口</b> .....	<b>33</b>
6.1 PORTA 和 CPIOA 寄存器 .....	33
6.2 其它引脚功能 .....	34
6.3 PORTA 引脚说明和引脚图 .....	36
<b>7.0 定时器及 PWM</b> .....	<b>42</b>
7.1 TIMER0 模块 .....	42
7.2 TIMER1 模块 .....	44
7.3 PWM 模块 .....	46
<b>8.0 电容触摸模块</b> .....	<b>51</b>
8.1 原理说明 .....	51
8.2 扫描周期定时器 .....	51
8.3 配置说明 .....	51
8.4 电容感应数据区 .....	51

8.5 寄存器说明 .....	52
<b>9.0 配置寄存器 .....</b>	<b>56</b>
9.1 CONFIG OPTION .....	56
<b>10.0 指令表 .....</b>	<b>57</b>
<b>11.0 电气特性 .....</b>	<b>58</b>
11.1 绝对极限参数 .....	58
11.2 直流电器特性 .....	59
11.3 交流电气特性 .....	60
<b>12.0 开发支持 .....</b>	<b>61</b>
12.1 仿真信息 .....	61
12.2 烧录信息 .....	62
<b>13.0 封装信息 .....</b>	<b>65</b>
13.1 SOT23-6 .....	65
13.2 SOP8 PIN .....	66
<b>14.0 汇春知识产权政策 .....</b>	<b>68</b>
14.1 专利权 .....	69
14.2 著作权 .....	69

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

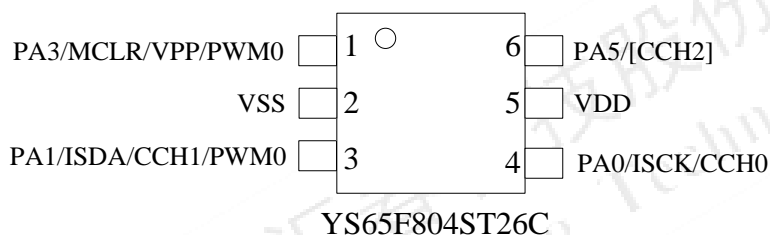
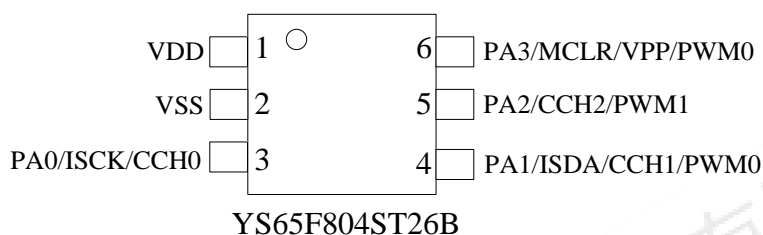
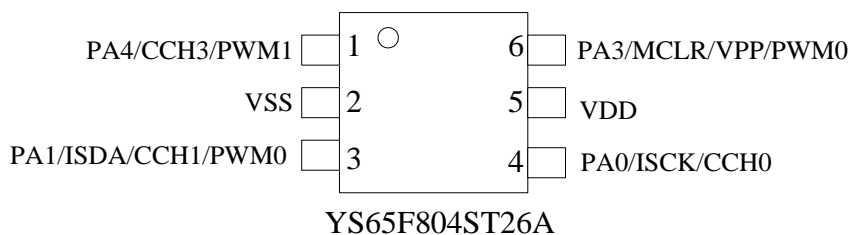


# 1.1 系统结构图

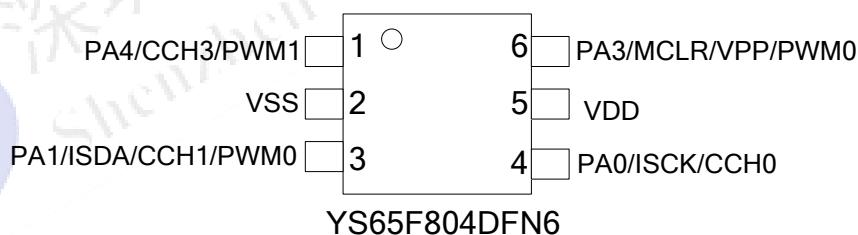


## 1.2 封装脚位图

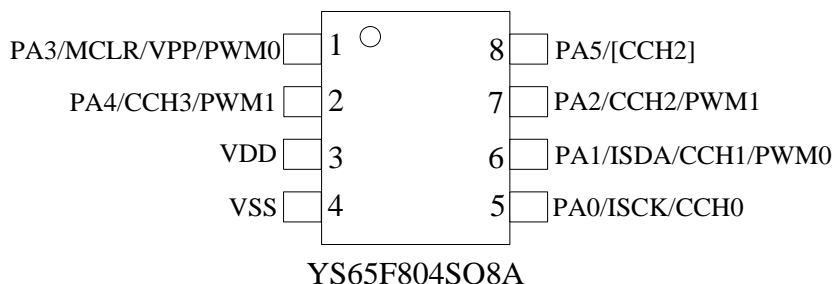
### 1.2.1 6 引脚图 (SOT23-6)

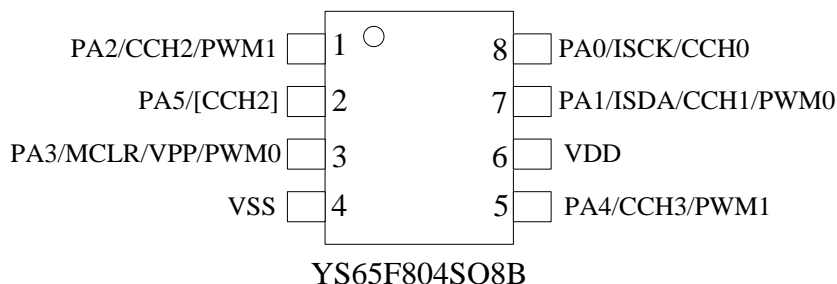


### 1.2.2 6 引脚图 (DFN6)

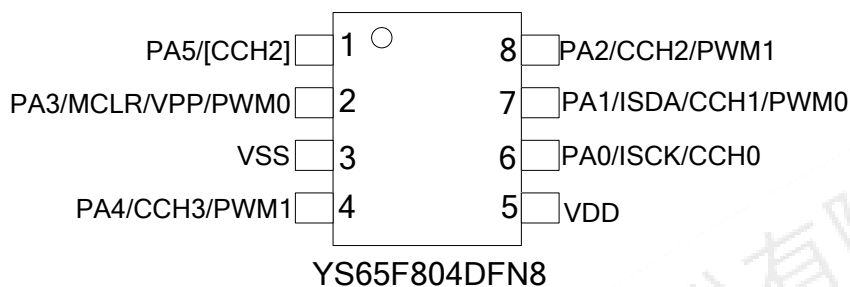


### 1.2.3 8 引脚图 (SOP8)





### 1.2.4 8 引脚图 (DFN8)



### 1.3 引脚说明

名称	功能	输入类型	输出类型	说明
PA0/ISCK/CCH0	PA0	TTL	CMOS	具有可编程上下拉和电平变化中断的PORTA I/O
	ISCK	ST	—	串行编程和调试时钟输入,
	CCH0	AN		触摸按键通道0输入
PA1/ISDA/CCH1/PWM0	PA1	TTL	CMOS	具有可编程上下拉和电平变化中断的PORTA I/O
	ISDA	ST/TTL	CMOS	串行编程和调试数据输入输出,
	CCH1	AN		触摸按键通道1输入
	PWM0	—	CMOS	功能转移前PWM0输出
PA2/CCH2/PWM1	PA2	ST	CMOS	具有可编程上下拉和电平变化中断的PORTA I/O
	CCH2	AN		触摸按键通道2输入
	PWM1	—	CMOS	功能转移前PWM1输出
PA3/MCLR/VPP/【PWM0】	PA3	TTL	Open Drain	具有可编程上拉和电平变化中断的PORTA I/O
	MCLR	ST	—	带有内部上拉的主复位
	VPP	HV	—	编程电压 (9.5V)
	[PWM0]	—	NMOS	功能转移后PWM0输出
PA4/CLKOUT/CCH3/【PWM1】	PA4	TTL	CMOS	具有可编程上下拉和电平变化中断的PORTA I/O
	CCH3	AN		触摸按键通道3输入
	[PWM1]	—	CMOS	功能转移后PWM1输出
PA5/[CCH2]	PA5	TTL	CMOS	具有可编程上下拉和电平变化中断的PORTA I/O
	[CCH2]	AN		触摸按键通道2输入
VDD	VDD	电源	—	正电源端
VSS	VSS	电源	—	接地参考端

图注: AN = 模拟输入或输出, TTL = TTL兼容输出, ST = 带CMOS电平的施密特触发器输入.

HV = 高压, CMOS = CMOS兼容输入或输出, NMOS = NMOS兼容输入或输出 opendrain: 开漏极

## 2.0 存储器构成

### 2.1 程序存储器构成

YS65F804 具有一个 11 位程序计数器, 可以对  $2K \times 14$  的程序存储空间进行寻址。而 YS65F804 仅在物理上实现了第一个  $2K \times 14$  (0000h-07FFh) 的存储空间。访问该边界以外的单元将导致实际访问存储器的第一个  $2K \times 14$  存储空间。复位向量地址为 0000h, 中断向量地址为 0004h。

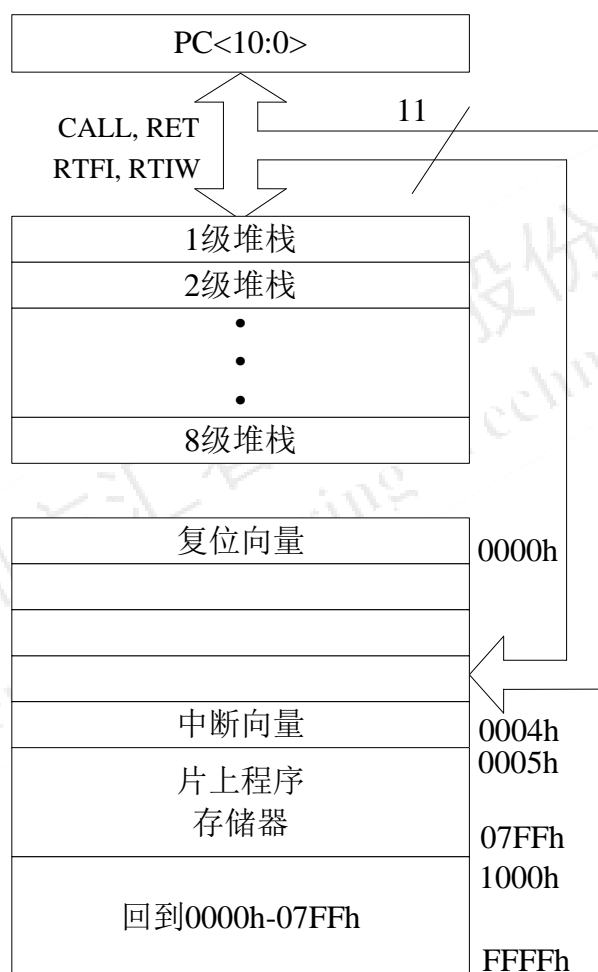


图 2-1

#### 2.1.1 复位向量 (0000H)

具有一个字长的系统复位向量 (0000H)。

- 上电复位;
- 看门狗复位;
- 外部复位;
- 欠压复位;

发生上述任一种复位后,程序将从 0000H 处重新开始执行,系统寄存器也都将恢复为默认值。根据 STATUS 寄存器中的TF和PF标志位的内容可以判断系统复位方式。

## 2.1.2 中断向量 (0004H)

中断向量的地址为 0004H。一旦有中断响应,程序计数器 PC 的当前值就会存入堆栈缓存器并跳转到 0004H 开始执行中断服务程序。

## 2.2 数据存储器的构成

数据存储器被分为 4 个存储区 (bank), 其中包含通用寄存器 (GeneralPurposeRegister, GPR) 和特殊功能寄存器 (SpecialFunctionRegister, SFR) 特殊功能寄存器位于每个存储区开头的 32 个单元。通用寄存器位于 Bank0 中的 20h-7Fh 和 Bank1 中的 A0h-BFh 寄存器单元中。它们以静态 RAM 的方式实现。Bank1、Bank2 和 Bank3 中的寄存器单元 F0h-FFh 指向 Bank0 中的地址单元 70h-7Fh。所有其他的 RAM 均未实现, 读取它们时将返回 0。

STATUS 寄存器的 RP1、RP0 位为存储区选择位:

- 00 → 选定 Bank0
- 01 → 选定 Bank1
- 10 → 选定 Bank2
- 11 → 选定 Bank3

### 2.2.1 通用文件寄存器

YS65F804 通用寄存器是按 128×8 的形式实现的。可直接访问每个寄存器或通过文件选择寄存器 (FileSelectRegister, MSR) 间接访问每个存储器 (见第 2.4 章节“[间接寻址、IAR 和 MSR 寄存器](#)”。)

### 2.2.2 特殊功能寄存器

特殊功能寄存器是 CPU 和外设模块用来控制所需的器件操作的寄存器, 这些寄存器都实现为静态 RAM。

特殊功能寄存器可分为两类: 内核与外设。本章仅讲述与“内核”有关的特殊功能寄存器。那些与外设功能部件的操作有关的特殊功能寄存器将在相应的外设功能部件章节中讲述。



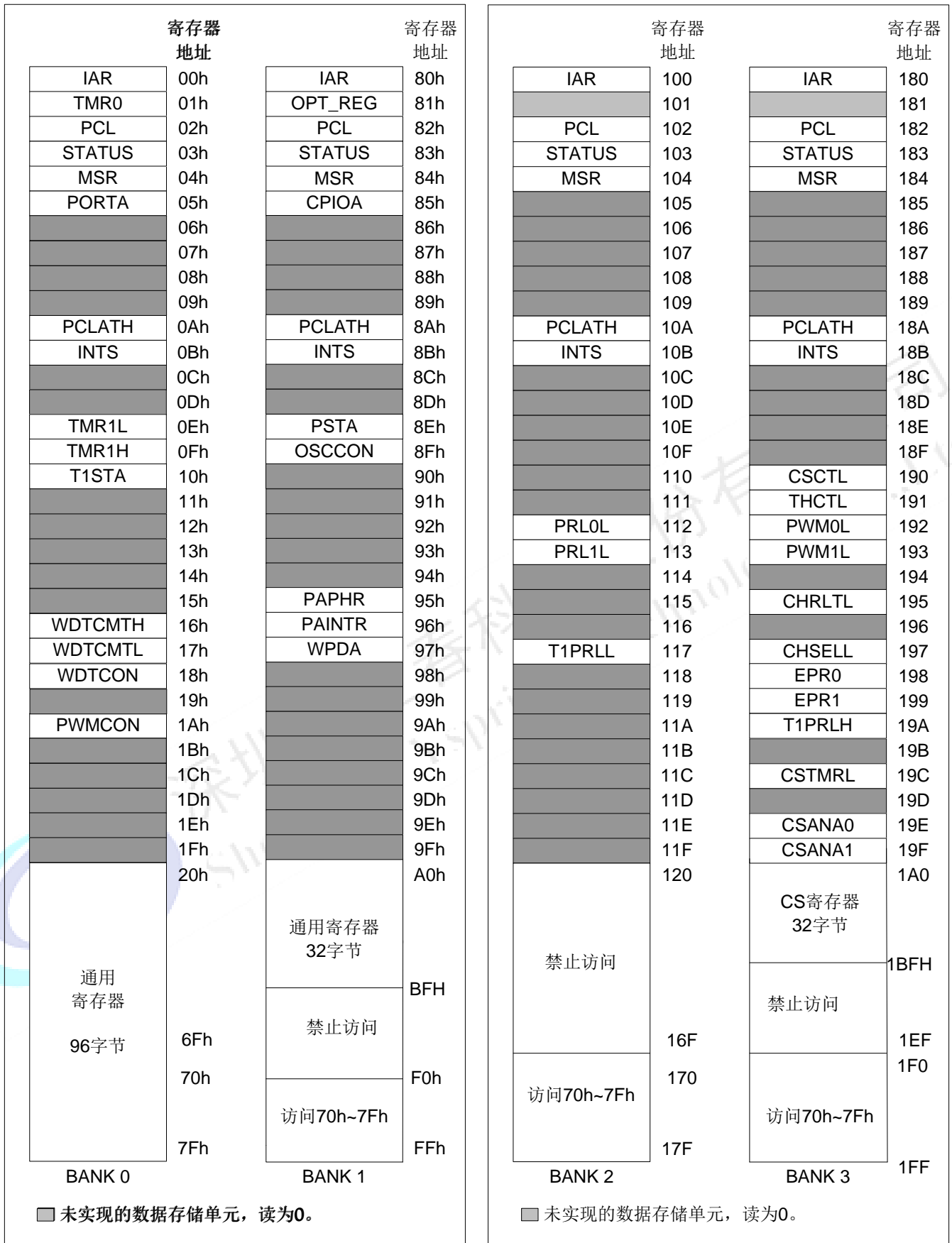


图 2-2：数据存储映射

## 2.2.2.1 特殊功能寄存器位定义 (BANK0)

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
<b>BANK0</b>										
00H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元 (不是物理寄存器)								xxxx xxxx
01H	<a href="#">TMR0</a>	TIMER0 模块寄存器								xxxx xxxx
02H	<a href="#">PCL</a>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
03H	<a href="#">STATUS</a>	IRP	RP1	RP0	TF	PF	Z	HC	C	0001_1xxx
04H	MSR	间接数据存储器地址指针								xxxx-xxxx
05H	<a href="#">PORTA</a>	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--xx_xxxx
06H	—	未实现								—
07H	—	未实现								—
08H	—	未实现								—
09H	—	未实现								—
0AH	<a href="#">PCLATH</a>	—	—	—	—	—	PC10	PC9	PC8	----_0000
0BH	<a href="#">INTS</a>	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000_0000
0CH	—	未实现								—
0DH	—	未实现								—
0EH	<a href="#">TMR1L</a>	16 位 TMR1 低字节的保持寄存器								xxxx-xxxx
0FH	<a href="#">TMR1H</a>	16 位 TMR1 高字节的保持寄存器								xxxx-xxxx
10H	<a href="#">T1STA</a>	—	—	TICKPS	—	—	—	—	TMR1ON	0000_0000
11H	—	未实现								—
12H	—	未实现								—
13H	—	未实现								—
14H	—	未实现								—
15H	—	未实现								—
16H	<a href="#">WDTMTH</a>	WDT 定时器低字节寄存器								xxxx xxxx
17H	<a href="#">WDTMTL</a>	WDT 定时器高字节寄存器								xxxx xxxx
18H	<a href="#">WDTCON</a>	—	—	—	WDTPS				SWDTEN	---0_1000
19H	—	未实现								—
1AH	<a href="#">PWMCON</a>	—	—	—	—	PWM1SEL	PWM0SEL	PWM1EN	PWM0EN	--00_0011
1BH	—	未实现								—
1CH	—	未实现								—
1DH	—	未实现								—
1EH	—	未实现								—
1FH	—	未实现								—

图注: - = 未实现单元读为 0, u = 不变, x = 未知, q = 取值视情况而定, 阴影 = 未实现

注 1: 配置字寄存器中的 CFG\_MCLRE 为 1 时 RA3 上拉使能。

## 2.2.2.2 特殊功能寄存器位定义（BANK1）

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
<b>BANK1</b>										
80H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元（不是物理寄存器）								xxxx xxxx
81H	<a href="#">OPT_REG</a>	—	—	—	TOEN	PSC	PS			---1_1111
82H	<a href="#">PCL</a>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
83H	<a href="#">STATUS</a>	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
84H	MSR	间接数据存储器地址指针								xxxx-xxxx
85H	<a href="#">CPIOA</a>	—	—	PORTA 三态控制位						--11_1111
86H	—	未实现								—
87H	—	未实现								—
88H	—	未实现								—
89H	—	未实现								—
8AH	<a href="#">PCLATH</a>	—	—	—	—	—	PC10	PC9	PC8	----_0000
8BH	<a href="#">INTS</a>	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000_0000
8CH	—	未实现								—
8DH	—	未实现								—
8EH	<a href="#">PSTA</a>	—	LVDF	LVDE N	SBOR EN	$\overline{MCR}$	IER	$\overline{POR}$	$\overline{BOR}$	-x01_100q
8FH	<a href="#">OSCCON</a>	—	IRCF			—	HTS	LTS	—	-101_x00-
90H	—	未实现								—
91H	—	未实现								—
92H	—	未实现								—
93H	—	未实现								—
94H	—	未实现								—
95H	<a href="#">PAPHR</a>	—	—	PAPH5	PAPH4	PAPH3	PAPH2	PAPH1	PAPH0	--00_0000
96H	<a href="#">PAINTR</a>	—	—	PORTA 电平变化中断允许位						
97H	<a href="#">WPDA</a>	—	—	WPDA5	WPDA4	—	WPDA2	WPDA1	WPDA0	--00_0000
98H	—	未实现								—
99H	—	未实现								—
9AH	—	未实现								—
9BH	—	未实现								—
9CH	—	未实现								—
9DH	—	未实现								—
9EH	—	未实现								—
9FH	—	未实现								—

## 2.2.2.3 特殊功能寄存器位定义（BANK2）

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
<b>BANK2</b>										
100H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元（不是物理寄存器）								xxxx xxxx
101H	—	未实现								—
102H	<a href="#">PCL</a>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
103H	<a href="#">STATUS</a>	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
104H	MSR	间接数据存储器地址指针								xxxx-xxxx
105H	—	未实现								—
106H	—	未实现								—
107H	—	未实现								—
108H	—	未实现								—
109H	—	未实现								—
10AH	<a href="#">PCLATH</a>	—	—	—	—	—	PC10	PC9	PC8	---_000
10BH	<a href="#">INTS</a>	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000_0000
10CH	—	未实现								—
10DH	—	未实现								—
10EH	—	未实现								—
10FH	—	未实现								—
110H	—	未实现								—
111H	—	未实现								—
112H	<a href="#">PRL0L</a>	PWM0 占空比重载寄存器低字节								0000_0000
113H	<a href="#">PRL1L</a>	PWM1 占空比重载寄存器低字节								0000_0000
114H	—	未实现								—
115H	—	未实现								—
116H	—	未实现								—
117H	<a href="#">TIPRL</a>	PWM0/1 周期寄存器的低字节。								0000_0000
118H	—	未实现								—
119H	—	未实现								—
11AH	—	未实现								—
11BH	—	未实现								—
11CH	—	未实现								—
11DH	—	未实现								—
11EH	—	未实现								—
11FH	—	未实现								—

## 2.2.2.4 特殊功能寄存器位定义（BANK3）

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
<b>BANK3</b>										
180H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元（不是物理寄存器）								xxxx xxxx
181H	—	未实现								—
182H	<a href="#">PCL</a>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
183H	<a href="#">STATUS</a>	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
184H	MSR	间接数据存储器地址指针								xxxx-xxxx
185H	—	未实现								—
186H	—	未实现								—
187H	—	未实现								—
188H	—	未实现								—
189H	—	未实现								—
18AH	<a href="#">PCLATH</a>	—	—	—	—	—	PC10	PC9	PC8	----_0000
18BH	<a href="#">INTS</a>	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000_0000
18CH	—	未实现								—
18DH	—	未实现								—
18EH	—	未实现								—
18FH	—	未实现								—
190H	<a href="#">CSCTL</a>	CSEN	RAMEN	CSIE	CSBUSY	CSINTS	CSACU<2:0>			0000_0000
191H	<a href="#">THCTL</a>	THFIB	THGAP<1:0>		THMOD<2:0>			THDS<1:0>		0000_0000
192H	PWM0L	PWM0 占空比较寄存器低字节								0000_0000
193H	PWM1L	PWM1 占空比较寄存器低字节								0000_0000
194H	—	未实现								—
195H	<a href="#">CHRLTL</a>	—	—	—	—	CHRLTL<3:0>				0000_0000
196H	—	未实现								—
197H	<a href="#">CHSELL</a>	CCH2SEL	—	—	—	CHSEL<3:0>				0---_0000
198H	<a href="#">EPR0</a>					PWM0H 占空比较寄存器高两位		EPR0L		----_0000
199H	<a href="#">EPR1</a>					PWM1H 占空比较寄存器高两位		EPR1L		----_0000
19AH	<a href="#">T1PRLH</a>	—	—	—	—	—	—	T1PRLH		----_--00
19BH	—	未实现								—
19CH	<a href="#">CSTMR</a>	触摸自动扫描定时器								0000_0000
19DH	—	未实现								—
19EH	<a href="#">CSANA0</a>	—	BSTRM<2:0>			CMPTRM<1:0>		VTSEL<1:0>		0010_0111
19FH	<a href="#">CSANA1</a>	—	—	—	DTTRM<1:0>		CTTRM<2:0>			--0_1011

### 2.2.2.3 STATUS 状态寄存器

状态（STATUS）寄存器包含 ALU 的算术运算状态、复位状态和数据存储器（SRAM）的存储区选择位，和其他寄存器一样，状态寄存器也可以作为任何指令的目标寄存器。如果一条影响 Z、HC 或 C 位的指令以状态寄存器作为目标寄存器，将禁止写这三位。根据器件逻辑，这些位会被置 1 或清零。此外，也不能写 TF 和 PF 位。因此，当执行一条把状态寄存器作为目标寄存器的指令后，状态寄存器的结果可能和预想的不一樣。例如，执行 CLRR STATUS 指令会清零该寄存器的高 3 位并将 Z 位置 1。从而使状态寄存器的值为“000uu1uu”（其中 u 表示不变）。因此，建议仅使用 BCR、BSR、SWAPR 和 STWR 指令来改变状态寄存器，因为这些指令不影响任何状态位。欲知其他不会影响任何状态位的指令，请参见第 10.0 章节“指令表”。

#### 寄存器 03H/83H/183H: 状态寄存器 (STATUS)

R/W-0	R/W-0	R/W-0	R-x	R-x	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TF	PF	Z	HC	C
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7	<b>IRP:</b> 间接寻址最高位: 1 = 间接寻址第2、3页 0 = 间接寻址第0、1页
bit6 bit5	<b>RP1RP0:</b> 寄存器存储区选择位（用于直接寻址） 00 = 直接寻址第0页 01 = 直接寻址第1页 10 = 直接寻址第2页 11 = 直接寻址第3页
bit4	<b>TF:</b> 超时状态位 1 = 上电后，执行了CLRWT指令或SLEEP指令 0 = 发生WDT超时溢出
bit3	<b>PF:</b> 掉电标志位 1 = 上电复位后或执行了CLRWT指令 0 = 执行了SLEEP指令
bit2	<b>Z:</b> 零标志位 1 = 算术运算或逻辑运算的结果为零 0 = 算术运算或逻辑运算的结果不为零
bit1	<b>HC:</b> 半进位/借位位（ADDWR、ADDWI、SUBWR和SUBWI指令）。对于借位，极性是相反的。 1 = 结果的第4低位向高位发生了进位 0 = 结果的第4低位未向高位发生进位
bit0	<b>C:</b> 进位/借位位 <sup>(1)</sup> （ADDWR、ADDWI、SUBWR和SUBWI指令） 1 = 结果的最高位发生了进位（减法时，没有发生借位时为1） 0 = 结果的最高位未发生进位
注	1: 借位的极性是相反的。减法是通过加上第二个操作数的二进制补码（Two's Complement）来实现的。对于移位指令（RRR和RLR），此位的值来自源寄存器的最高位或最低位。

### 2.2.2.4 OPT\_REG 选项寄存器

选项（OPT\_REG）寄存器是可读写的寄存器，包含可对以下各项进行配置的各种配置位：

- Timer0/WDT 预分频器分配位
- Timer0 使能位

寄存器81H：选项寄存器（OPT\_REG）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	<b>T0EN</b>	<b>PSC</b>	<b>PS2</b>	<b>PS1</b>	<b>PS0</b>
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-5 未实现：读为0

bit4 **T0EN**：Timer0使能位

1 = Timer0打开

0 = Timer0关闭

bit3 **PSC**：预分频器分配位

1 = 将预分频器分配给WDT

0 = 将预分频器分配给Timer0模块

bit2-0 **PS<2:0>**：预分频比选择位

位值	TMR0分频比	WDT分频比
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

### 2.2.2.5 PSTA 寄存器

电源控制（PSTA）寄存器包含区分以下复位的标志位：

- 低电压检测（LVD）
- 上电复位（POR）
- 欠压复位（BOR）
- 看门狗定时器复位（WDT）
- 外部MCLR复位

PSTA 寄存器也用于控制超低功耗唤醒和BOR的软件使能。

寄存器 8EH：电源控制寄存器1（PSTA）

U-0	R/W-x	R/W-0	R/W-1	R/W-x	R/W-x	R/W-x	R/W-x
—	<b>LVDF</b>	<b>LVDEN</b>	<b>SBOREN</b>	<b>MCLR</b>	<b>IER</b>	<b>POR</b>	<b>BOR</b>
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7	未实现：读为0
Bit6	<b>LVDF</b> ：低电压检测标志 1 = 未检测到了低电压（必须在上电复位发生后用软件置1） 0 = 检测到了低电压
bit5	<b>LVDEN</b> ：低电压检测使能位 1 = 允许低电压检测 0 = 禁止低电压检测
bit4	<b>SBOREN</b> ：软件欠压检测使能位 <sup>(1)</sup> 1 = 允许欠压电压 0 = 禁止欠压电压
bit3	<b>MCLR</b> ：外部复位状态位 1 = 上电复位值 0 = 发生了外部复位
bit2	<b>IER</b> ：指令错误复位状态位 1 = 发生了指令错误复位 0 = 上电复位值
bit1	<b>POR</b> ：上电复位状态位 1 = 未发生上电复位 0 = 发生了上电复位（必须在上电复位发生后用软件置1）
bit0	<b>BOR</b> ：欠压复位状态位 1 = 未发生欠压复位 0 = 已发生欠压复位（必须在发生欠压复位后用软件置1）
注	1：当配置字寄存器中的LVREN<1:0> = 01时允许使用该位对BOR进行控制。

## 2.3 PCL 和 PCLATH

程序计数器（PC）为 11 位宽。其低 8 位来自可读写的 PCL 寄存器，高 3 位（PC<10:8>）来自 PCLATH，不能直接读写。只要发生复位，PC 就将被清零。图 2-3 显示了装载 PC 值的两种情形。图 2-3 中上部分的例子说明了在写 PCL（PCLATH<2:0>→PCH）时是如何装载 PC 的。

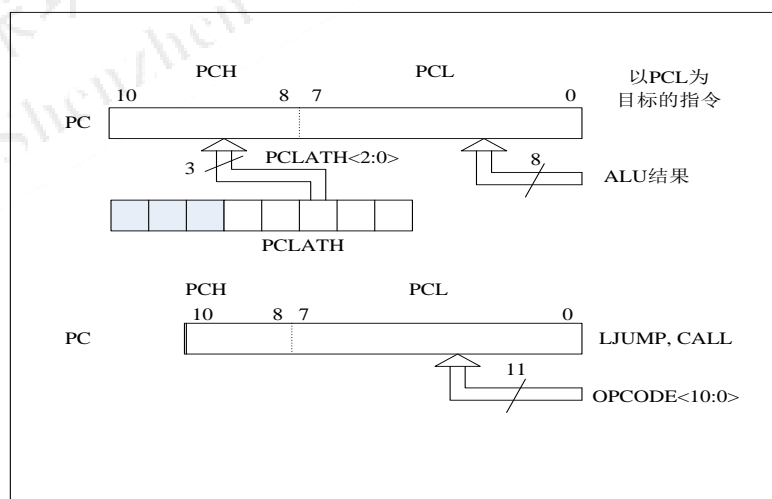


图 2-3：在不同情况下装载 PC

### 2.3.1 修改 PCL

执行任何以 PCL 寄存器为目标寄存器的指令将同时使程序计数器的 PC<10:8>位（PCH）被 PCLATH 寄存器的内容所取代。这样可通过将所需的高 3 位写入 PCLATH 寄存器来改变程序计数器的所有内容。当低 8



位写入 PCL 寄存器时，程序计数器的所有 11 位将变为 PCLATH 寄存器中所包含的值以及写入 PCL 寄存器中的值。

计算 LJUMP 指令是通过向程序计数器加入偏移量 (ADDWR PCL) 来实现的。通过修改 PCL 寄存器跳转到查找表或程序分支表 (计算 LJUMP) 时应特别谨慎。假定 PCLATH 设置为表的起始地址，如果表长度大于 255 条指令，或如果存储器地址的低 8 位在表的中间从 0xFF 计满返回到 0x00，那么在每次表起始地址与表内的目标地址之间发生计满返回时，PCLATH 必须均必须递增。

## 2.3.2 堆栈

YS65F804 器件具有 8 级深 × 11 位宽的硬件堆栈。堆栈空间既不占用程序存储区空间，也不占用数据存储区空间，而且堆栈指针是不可读写的。当执行 CALL 指令或当中断导致程序跳转时，PC 值将被压入 (PUSH) 堆栈。而在执行 RET、RTIW 或 RTFI 指令时，堆栈中的断点地址将从堆栈中弹出 (POP) 到 PC 中。PCLATH 不受 PUSH 或 POP 操作的影响。

堆栈的工作原理犹如循环缓冲区。这意味着当堆栈压栈 8 次后，第 9 次压栈的数值将会覆盖第一次压栈时所保存的数值，而第十次压栈数值将覆盖第二次压栈时保存的数值，以后依次类推。

注： 1: 不存在指明堆栈是否上溢或下溢的状态标志位。

2: 不存在被称为 PUSH 或 POP 的指令/助记符。堆栈的压入或弹出是源于执行了 CALL、RET、RTIW 和 RTFI 指令，或源于指向中断向量地址。

### 寄存器 02H/82H/182H: 程序计数器低字节控制寄存器 (PCL)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
bit7							Bit0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-0 PC<7:0>: 程序计数器低8位。

### 寄存器 0AH/8AH/18AH: 程序计数器高字节控制寄存器 (PCLATH)

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	PC10	PC9	PC8
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-5 未实现：读为0

bit4-0 PC<10:8>: 程序计数器高3位。

## 2.4 间接寻址、IAR 和 MSR 寄存器

IAR 寄存器不是实际存在的寄存器，使用 IAR 寄存器寻址将产生间接寻址。任何使用 IAR 寄存器的指令，实际上是对文件选择寄存器 (MSR) 所指向的数据进行存取。间接对 IAR 寄存器进行读操作将返回 00h。间接对 IAR 寄存器进行写操作将导致空操作 (尽管可能会影响状态标志位)。通过将 8 位的 MSR 寄存器与 STATUS 寄存器的 IRP 位进行组合可得到一个有效的 9 位地址，如图 2-4 所示。例 2-5 给出了一个使用间接寻址将 RAM 地

址单元 20h-2Fh 清零的简单程序。

例 2-5: 间接寻址

```

;功能: 地址 20 开始的连续 16 个字节的 RAM 清 0
LDWI    0X20    ;
STWR    MSR     ;地址指针赋初值
NEXT:   CLRR    IAR     ;清除 MSR 值对应的寄存器值
        INCR    MSR,R   ;递增地址指针
        BTSS    MSR,4   ;完成没有?
        LJUMP   NEXT    ;没完成, 继续
CONTINUE:                ;完成了

```

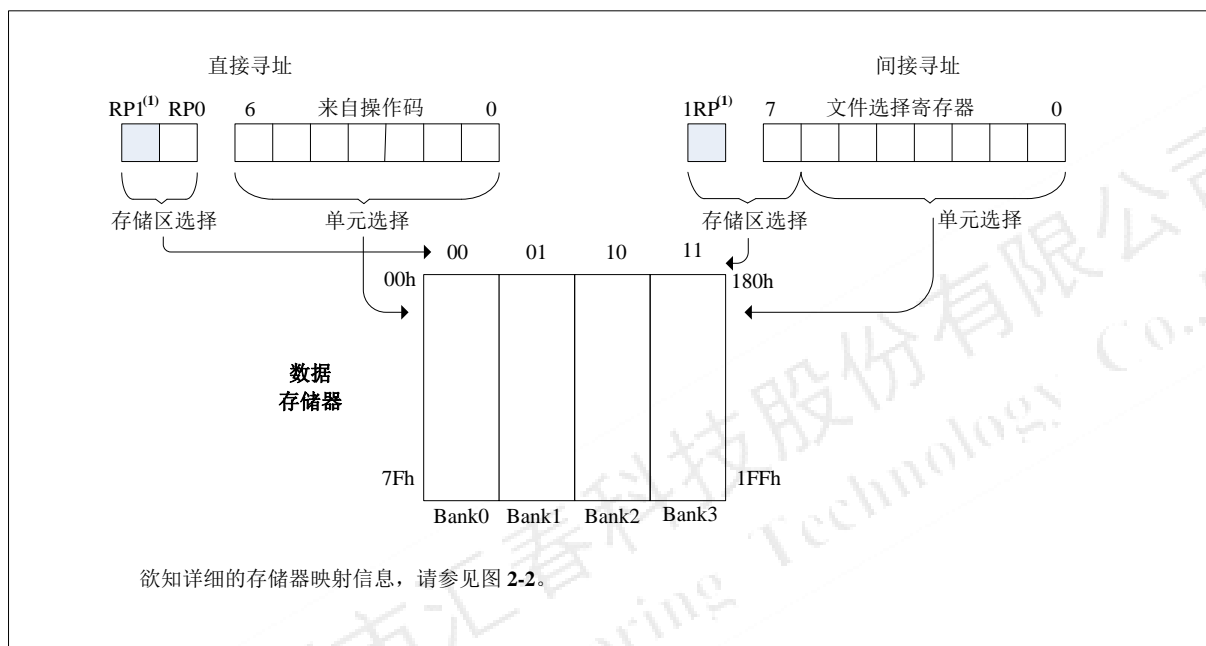


图 2-4: YS65F804 的直接/间接寻址

## 2.5 看门狗定时器 (WDT)

WDT 具有以下特性:

- 工作于 LFINTOSC (31kHz)
- 包含 16 位预分频器
- 与 Timer0 共用 8 位预分频器
- 超时周期从 1ms 至 268 秒
- 配置位和使用软件控制

WDT 在如表 2-1 所述的情况下清零。

### 2.5.1 WDT 振荡器

WDT 的时基来源于 31kHz LFINTOSC。OSCCON 寄存器的 LTS 位不反映 LFINTOSC 是否被使能。在所有复位时 WDTCON 的值为“---01000”。这样标称时基为 17ms。

注: 当执行振荡器起振定时器 (OST) 时, WDT 保持复位状态, 因为 OST 使用 WDT 纹波计数器来执行振荡器延时计数。OST 计数到期后, WDT 将开始计数 (如果使能)。

## 2.5.2 WDT 控制

配置字寄存器中的 WDTE 位置 1 时，WDT 连续运行且 WDTCON 寄存器中的 SWDTEN 位不起作用。如果 WDTE 清零，那么 SWDTEN 位可用于使能和禁止 WDT。该位置 1 使能 WDT，该位清零禁止 WDT。

看门狗定时器框图中（图 2-5）的 PSC 和 PS<2:0>位在 OPT\_REG 寄存器中。。更多信息请参见第 7.1 节“Timer0 模块”。

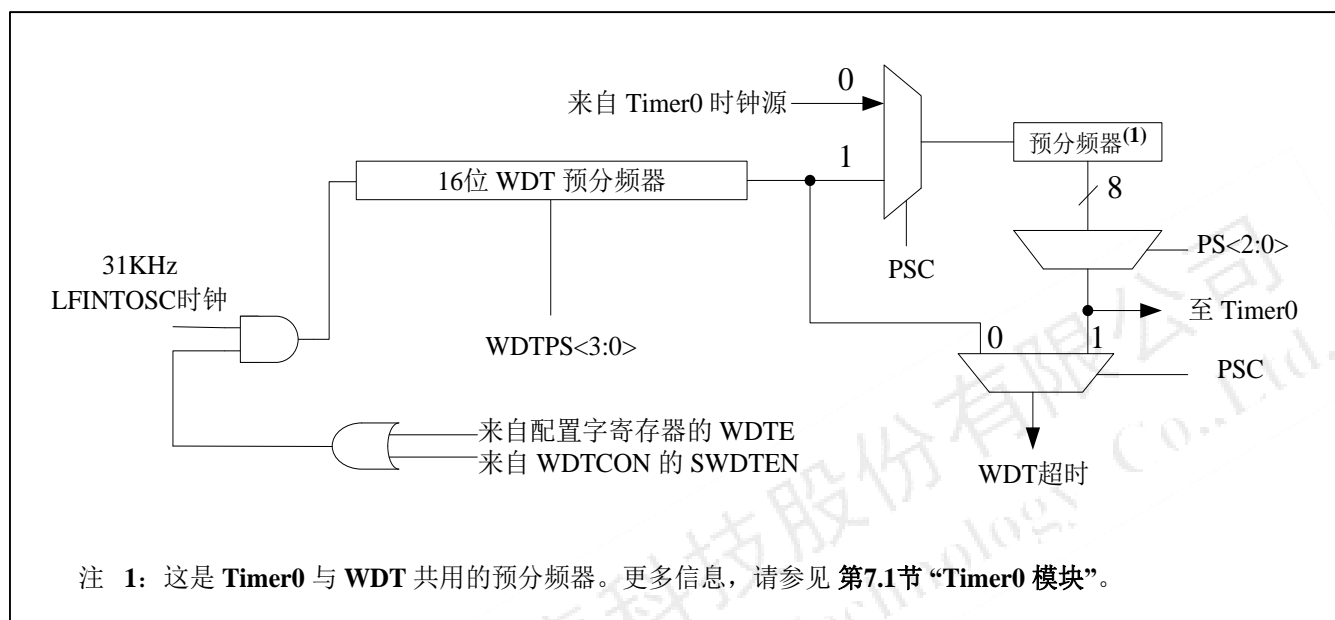


图 2-5: 看门狗定时器框图

## 2.5.3 WDT 计数寄存器 (WDTCMTL 和 WDTCMTH)

WDT 是 16 位的计数器，当看门狗溢出或执行 CLRWT 指令时，WDTCMTL 与 WDTCMTH 被赋 FFFFH。看门狗运行时，每个慢时钟周期，寄存器 (WDTCMTH:WDTCMTL) 执行一次减一操作。

### 寄存器 16H: WDT 定时器寄存器的高字节 (WDTCMTH)

R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1
WDTCMTH							
bit7				bit0			
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

Bit7-0      **WDTCMTH:** 看门狗计数器的高字节

### 寄存器 17H: WDT 定时器寄存器的低字节 (WDTCMTL)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
WDTCMTL							
bit7	bit7	bit7	bit7	bit7	bit7	bit7	bit7
图注:							
R = 可读位		R = 可读位		R = 可读位		R = 可读位	
-n = POR时的值		-n = POR时的值		-n = POR时的值		-n = POR时的值	

Bit7-0      **WDTCMTL:** 看门狗计数器的低字节

## 寄存器 18H: 看门狗定时器控制寄存器 (WDTCON)

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-5

未实现: 读为0

Bit4-1

**WDTPS<3:0>**: 看门狗定时器周期选择位

位值 = 预分频

0000 = 1:32

0001 = 1:64

0010 = 1:128

0011 = 1:256

0100 = 1:512 (复位值)

0101 = 1:1024

0110 = 1:2048

0111 = 1:4096

1000 = 1:8192

1001 = 1:16384

1010 = 1:32768

1011 = 1:65536

1100 = 保留

1101 = 保留

1110 = 保留

1111 = 保留

Bit0

**SWDTEN**: 软件使能或禁止看门狗定时器位<sup>(1)</sup>

1 = WDT 开启

0 = WDT 关闭 (复位值)

注 1: 如果配置字寄存器 (CONFIG) 的WDTE配置位=1, 则WDT始终被使能, 而与该控制位的状态无关。

如果配置字寄存器 (CONFIG) 的WDTE配置位=0, 则可以使用该控制位开启/关闭WDT。

表 2-1: WDT 状态

条件	WDT
CLRWT 命令	清零
振荡器失效检测	

表 2-2: 与看门狗定时器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 0000	---0 0000
OPT_REG	—	—	—	T0EN	PSC	PS2	PS1	PS0	--1- 1111	--1- 1111
CONFIG0	—	CPB	MCLRE	PWRTE	WDTE	—	—	—	—	—

图注: 看门狗定时器不使用阴影单元。

注 1: 关于配置字寄存器中所有位的操作, 请参见寄存器 800H。

## 3.0 复位

### 3.1 概述

YS65F804 有以下几种不同类型的复位：

- 上电复位 (POR)
- 正常工作期间的 WDT 复位
- 休眠期间的 WDT 复位
- 休眠期间的 MCLR 复位
- 欠压复位 (BOR)

有些寄存器不受任何复位的影响；在上电复位时它们的状态未知，而在其他复位时状态不变。大多数寄存器在以下复位时会复位到各自的“复位状态”：

- 上电复位
- MCLR 复位
- 休眠期间的 MCLR 复位
- WDT 复位

WDT 唤醒不会导致寄存器像 WDT 复位那样复位，这是因为唤醒被视为恢复正常工作。TF 和 PF 位在不同的复位情形下会分别被置 1 或清零，如表 3-1 所示。软件可使用这些位判断复位的性质。

图 3-1 给出了片上复位电路的简化框图。MCLR 复位路径上有一个噪声滤波器，用来检测并滤除小脉冲。关于脉冲宽度规范，请参见第 11.0 节“电气特性”。

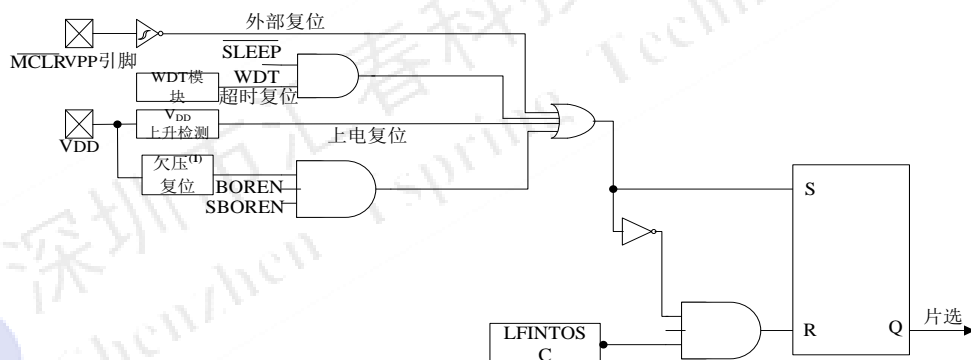


图 3-1: 片上复位电路的简化框图

表 3-1: STATUS/PSTA 位及其含义

POR	BOR	TF	PF	条件
0	x	1	1	上电复位
u	0	1	1	欠压复位
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	正常工作期间的 MCLR 复位
U	u	1	0	休眠期间的 MCLR 复位

图注： u = 不变， x = 未知

### 3.2 上电复位

在 VDD 达到足以使器件正常工作的电平之前，片上上电复位电路将使器件保持在复位状态。要有效利用 POR，只要将 MCLR 引脚通过一个电阻连接到 VDD 即可。需要一个最大上升时间才能达到 VDD。详见第 11.0 章节“电气特性”。如果使能了欠压复位，那么该最大上升时间规范将不再适用。欠压复位电路将使器件保持在复位状态，直到 VDD 达到 VBOR（见第 3.5 章节“欠压复位”）。

注： 当VDD降低时，上电复位电路不会产生内部复位。要重新使能上电复位，VDD必须至少保持100us的Vss电压。

当器件开始正常工作（退出复位状态）时，器件的工作参数（即电压、频率和温度等）必须得到满足，以确保其正常工作。如果不满足这些条件，那么器件必须保持在复位状态，直到满足工作条件为止。

### 3.3 MCLR

YS65F804 在MCLR复位路径中有一个噪声滤波器，该滤波器检测并滤除小脉冲。应注意，WDT 复位不会将MCLR引脚驱动为低电平。若施加在MCLR引脚上的电压超出规范值，则在 ESD 事件发生期间可导致MCLR复位且器件中流过超过规范值的过电流。因此，建议不要把MCLR引脚直接连接到 VDD。建议使用图 3-2 给出的 RC 网络。通过清零配置字寄存器中的MCLR位，可使能内部MCLR选项。当 MCLRE=0 时，在内部产生芯片的复位信号。当 MCLRE=1 时，PA3/MCLR引脚变成外部复位输入。在这种模式下，PA3/MCLR引脚具有到 VDD 的弱上拉功能。

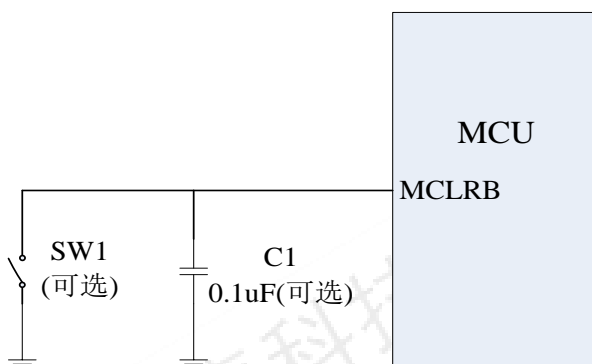


图 3-2: 建议的MCLR电路

### 3.4 欠压复位

配置字寄存器中的 BOREN0 和 BOREN1 位用于选择 4 种欠压复位模式中的一种。其中添加了两种允许使用软件或硬件对 BOR 的使能进行控制的模式。当 BOREN<1:0> = 01 时，可由 PSTA 寄存器的 SBOREN 位使能/禁止 BOR，从而能用软件对其进行控制。通过选择 BOREN<1:0>，可使欠压复位在休眠时被自动禁止，从而节约功耗；而在唤醒后被重新使能。在此模式下，SBOREN 位被禁止。关于配置字的定义，请参见寄存器 PSTA。如果 VDD 下降到 VBOR 以下，且持续时间超过参数值 (TBOR)（见第 11.0 节“电气特性”，欠压状况将使）器件复位。不管 VDD 的变化速率如何，上述情况都会发生。如果 VDD 低于 VBOR 的时间少于参数值 (TBOR)，则不一定会发生复位。

任何复位(上电复位、欠压复位或看门狗定时器复位等)都会使芯片保持复位状态，直到 VDD 上升到 VBOR 以上（见图 3-3）。如果使能了上电延时定时器，此时它将启动，并且会使器件保持复位状态的时间延长 64ms。

注： 配置字寄存器中的PWRTE位用于使能上电延时定时器。

如果在上电延时定时器运行过程中，VDD 降低到 VBOR 以下，芯片将重新回到欠压复位状态并且上电延时定时器会恢复为初始状态。一旦 VDD 上升到 VBOR 以上，上电延时定时器将执行一段 64ms 的复位。

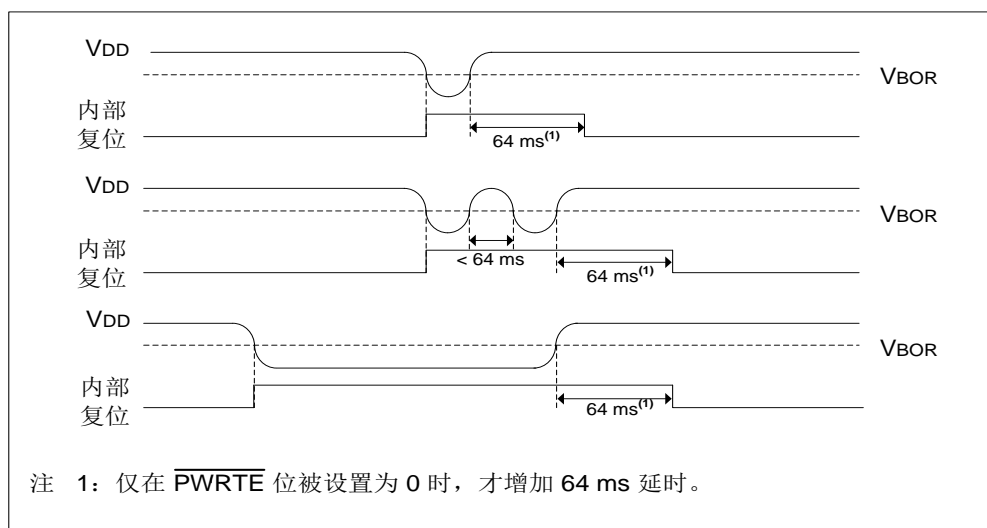


图 3-3: 欠压情形

表 3-1: 与欠压有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
PSTA	—	LVDF	LVD	SBOREN	MCR	IER	POR	BOR	0001 100x	0001 100x
STATUS	IRP	PR1	PAGE	$\overline{\text{T}}\text{F}$	$\overline{\text{P}}\text{F}$	Z	HC	C	0001 1xxx	0001 1xxx

图注: u = 不变, x = 未知, — = 未实现位, 读为 0, q = 取值视具体情况而定。BOR 不使用阴影单元。

注 1: 其他 (非上电) 复位包括正常工作时的 MCLR 复位和看门狗定时器复位。

表 3-2: 特殊寄存器的初始状态

条件	程序计数器	状态寄存器	PSTA 寄存器
上电复位	0000h	0001 1xxx	--01 --0x
正常工作期间的 MCLR 复位	0000h	000u uuuu	--0u --uu
休眠期间的 MCLR 复位	0000h	0001 0uuu	--0u --uu
WDT 复位	0000h	0000 uuuu	--0u --uu
WDT 唤醒	PC + 1	uuu0 0uuu	--uu --uu
欠压复位	0000h	0001 1uuu	--01 --u0
通过中断从休眠唤醒	PC + 1 <sup>(1)</sup>	uuu1 0uuu	--uu --uu

### 3.5 上电延时定时器 (PWRT)

上电延时定时器仅在上电时 (上电复位或欠压复位) 提供一个 55ms (标称值) 的固定延时。上电延时定时器采用 LFINTOSC 振荡器作为时钟源, 工作频率为 31kHz。只要 PWRT 处于活动状态, 芯片就保持在复位状态。配 PWRT 延时使 VDD 有足够的时间上升到所需的电平。置位  $\overline{\text{PWRT}}\text{E}$  可以禁止 (如果置 1) 或使能 (如果清零或被编程) 上电延时定时器。虽然不是必需的, 但是在使能欠压复位时也应使能上电延时定时器。

由于以下原因不同芯片的上电延时定时器的延时也各不相同:

- VDD 差异
- 温度差异

- 制造工艺差异

详见直流参数（第 11.0 章节“电气特性”）。

注：在  $\overline{\text{MCLR}}$  引脚的低于  $\text{VSS}$  的电压尖峰，包括大于  $80\text{mA}$  的电流，可导致闭锁。因此，在  $\overline{\text{MCLR}}$  引脚上施加“低”电平时，应使用阻值在  $50\text{-}100\Omega$  的串联电阻，而不是将该引脚直接拉到  $\text{VSS}$ 。

## 3.6 低电压检测（LVD）

低电压检测是用于监测  $\text{VDD}$  电压，当使能低电压检测（ $\text{LV DEN}=1$ ）时，如果  $\text{VDD}$  电压低于设定的电压值， $\text{PSTA}$  寄存器的  $\text{LVDF}$  位将复位（置零）。 $\text{LVDF}$  位必须由软件写一。LVD 检测电压的设定由 BOR 复位电压点来确定，见下表：

BOR 复位电压	LVD 检测电压
2.0V	2.5V
2.5V	3.0V
3.0V	3.6V
3.6V	不确定（3.6V 以上）

## 3.7 电源控制（PSTA）寄存器

电源控制寄存器  $\text{PSTA}$  有两个状态位，用于指示上次发生的复位的类型。

$\text{bit0}$  是  $\overline{\text{BOR}}$ （欠压复位）标志位。 $\overline{\text{BOR}}$  在上电复位时未知。然后，用户必须将该位置 1，并在随后的复位发生时检查  $\overline{\text{BOR}}$  是否为 0，如果是，则表示已发生欠压复位。当禁止欠压复位电路（配置字寄存器中的  $\text{BOREN}\langle 1:0 \rangle = 00$ ）时， $\overline{\text{BOR}}$  状态位是“无关位”并且不一定预测得到。

$\text{bit1}$  是  $\overline{\text{POR}}$ （上电复位）标志位，在上电复位时值为 0，其他情况下不受影响。上电复位后，用户必须对该位写 1。发生后续复位后，如果  $\overline{\text{POR}}$  为 0，则表示发生了上电复位（即  $\text{VDD}$  可能已经变为了低电平）。



## 4.0 系统时钟

### 4.1 概述

振荡器模块有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. HFINTOSC（高频内部振荡器）出厂时已校准，工作频率为 16MHz，精度范围为±1%。
2. LFINTOSC（低频内部振荡器）未经校准，工作频率为 31kHz。

通过软件对 OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>进行操作，可选择系统时钟速度。

图 4-1 给出了振荡器模块的框图。

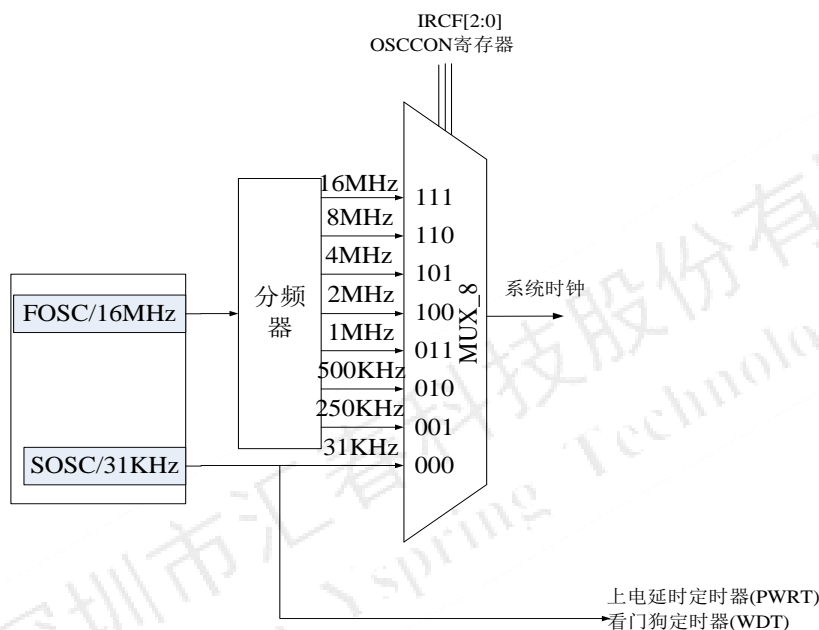


图 4-1: MCU 时钟源框图

### 4.2 振荡器控制

振荡器控制（OSCCON）寄存器控制系统时钟和频率选择等选项。OSCCON 寄存器包含以下位：

- 频率选择位（IRCF）
- 频率状态位（HTS 和 LTS）

#### 寄存器 8FH: 振荡器控制寄存器（OSCCON）

U-0	R/W-1	R/W-0	R/W-1	U-0	R-0	R-0	U-0
—	IRCF2	IRCF1	IRCF0	—	HTS	LTS	—
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7

未实现：读为0

bit6-4	<b>IRCF&lt;2:0&gt;</b> : 内部振荡器频率选择位 000 = 31kHz 001 = 250kHz 010 = 500kHz 011 = 1MHz 100 = 2MHz 101 = 4MHz (缺省值) 110 = 8MHz 111 = 16MHz
bit3	未实现: 读为0
bit2	<b>HTS</b> : HFINTOSC (高频——16MHz到250kHz) 状态位 1 = HFINTOSC稳定 0 = HFINTOSC不稳定
bit1	<b>LTS</b> : LFINTOSC (低频——31kHz) 状态位 1 = LFINTOSC稳定 0 = LFINTOSC不稳定
bit0	未实现: 读为0

### 4.2.1 HFINTOSC

高频内部振荡器 (HFINTOSC) 是出厂时已校准的 16MHz 内部时钟源。

HFINTOSC 的输出连接到后分频器和多路复用器 (见图 4-1)。使用 OSCCON 寄存器的 IRCF<2:0>位, 可通过软件选择七个频率之一, 更多信息, 请参见第 4.2 节“振荡器控制”。将 OSCCON 寄存器的 IRCF<2:0>位设置为  $\neq 000$  选择 16MHz 到 250kHz 之间的任一频率, 可使能 HFINTOSC。OSCCON 寄存器的 HF 内部振荡器 (HTS) 位用于显示 HFINTOSC 是否稳定。

### 4.2.2 LFINTOSC

低频内部振荡器 (LFINTOSC) 是未经校准的 31kHz 内部时钟源。

LFINTOSC 的输出连接到后分频器和多路复用器 (见图 4-1)。通过软件对 OSCCON 寄存器的 IRCF<2:0>位进行操作, 选取 31kHz。LFINTOSC 还是看门狗定时器 (WDT) 的时钟源。

选取 31kHz 只需将 OSCCON 寄存器的 IRCF<2:0>位设置为 000 为系统时钟源, 或者使能看门狗定时器 (WDT 时, LFINTOSC 将被使能:

OSCCON 寄存器的 LF 内部振荡器 (LTS) 位用于指示 LFINTOSC 是否稳定。

### 4.2.3 频率选择位 (IRCF)

16MHz HFINTOSC 和 31kHz LFINTOSC 的输出连接到后分频器和多路复用器 (见图 4-1)。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>用于选择内部振荡器的频率输出。可通过软件选择以下 8 个频率之一:

- 16MHz
- 8MHz
- 4MHz (复位后的缺省值)
- 2MHz
- 1MHz

- 500kHz
- 250kHz
- 31kHz

注：任何复位后，OSCCON寄存器的IRCF<2:0>位将被置为101且频率选择置为4MHz。用户可修改IRCF位来选择其他频率。

## 4.2.4 HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见图 4-2）。在这种情况下，OSCCON 寄存器的 IRCF 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

1. OSCCON 寄存器的 IRCF<2:0>位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低，时钟切换电路等待新时钟上升沿的到来。
5. 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 HTS 和 LTS 位按要求被更新。
6. 时钟切换完成。

如果选取的内部振荡器速度在 16MHz 到 250kHz 之间，选取新频率不存在启动延时，这是因为新旧频率都来自经过分频器和多路复用器的 HFINTOSC。启动延时规范在第 11.0 章节“电气特性”中与振荡器相关的表格中。

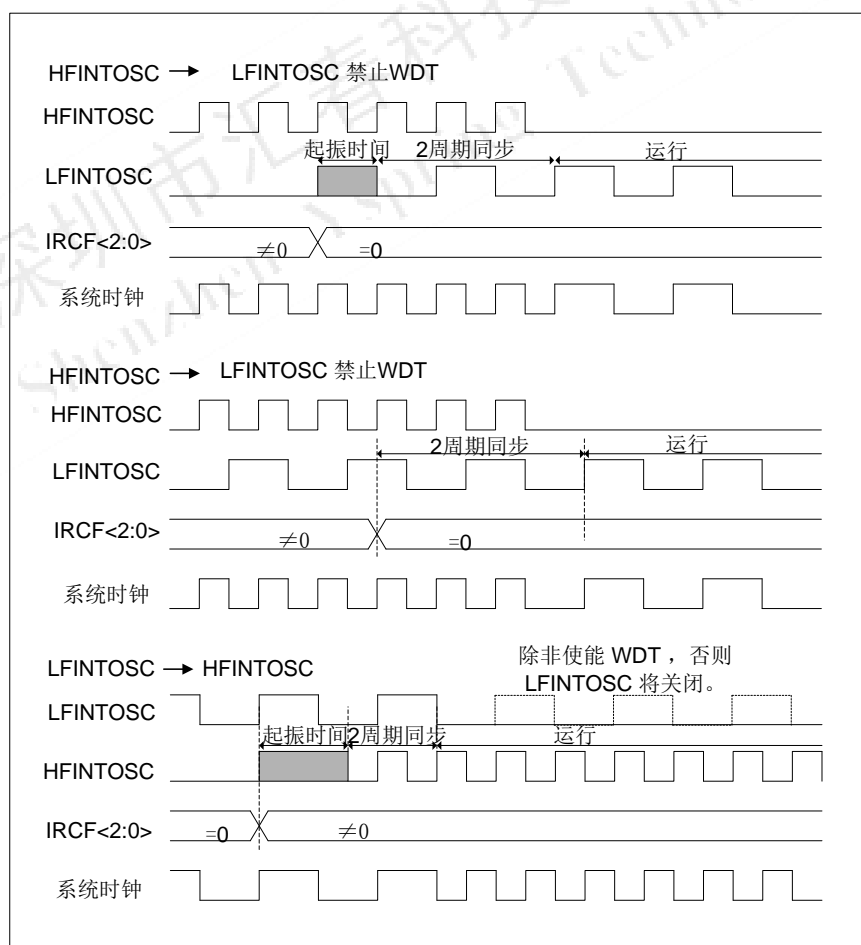


图 4-2：内部振荡器切换时序

## 4.3 掉电模式（休眠）

通过执行 SLEEP 指令可进入掉电模式。

如果使能看门狗定时器：

- WDT 将被清零并保持运行。
- 状态寄存器中的 PF 位被清零。
- TF 位被置 1。
- 关闭振荡器驱动器。
- I/O 端口保持执行 SLEEP 指令之前的状态（驱动为高电平、低电平或高阻状态）。

为使这种模式下的电流消耗降至最低，所有 I/O 引脚都应保持为 VDD 或 VSS，以确保没有外部电路从 I/O 引脚消耗电流。为了避免输入引脚悬空而引入开关电流，应在外部将高阻输入的 I/O 引脚拉为高电平或低电平。为使电流消耗降至最低，T0CKI 输入也应保持为 VDD 或 VSS，还应考虑 PORTA 片上上拉的影响，MCLR 引脚必须为逻辑高电平。

注： 请注意，由于 WDT 超时产生的复位不会将 MCLR 引脚驱动为低电平。

### 4.3.1 从休眠状态唤醒

由于在休眠期间没有片上时钟处于工作状态，下列外设中断可以将器件从休眠状态唤醒：

1. PORTA 电平变化中断
2. 来自 CS 模块的触摸中断
3. 看门狗定时器唤醒（如果 WDT 使能）

MCLR 引脚上的有效信号会导致器件复位。其他事件被认为是程序执行的继续。状态寄存器中的 TF 和 PF 位用于确定器件在执行 SLEEP 复位的原因。位在上电时被置 1，PF 指令时被清零。TF 位在发生 WDT 唤醒时被清零。

当执行 SLEEP 指令时，下一条指令 (PC+1) 将预先取出。如果希望通过中断事件唤醒器件，则必须将相应的中断允许位置 1（使能）。发生唤醒与 GIE 位的状态无关。如果 GIE 位被清零（禁止），器件将继续执行 SLEEP 指令之后的指令。如果 GIE 位被置 1（使能），器件执行 SLEEP 指令之后的指令，然后跳转到中断地址 (0004h) 处执行代码。如果不希望执行 SLEEP 指令之后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。器件从休眠状态唤醒时，WDT 都将被清零，而与唤醒原因无关。

注： 如果禁止了全局中断 (GIE 被清零)，但有任一中断源将其中断允许位以及相应的中断标志位置 1，器件将立即从休眠状态唤醒。

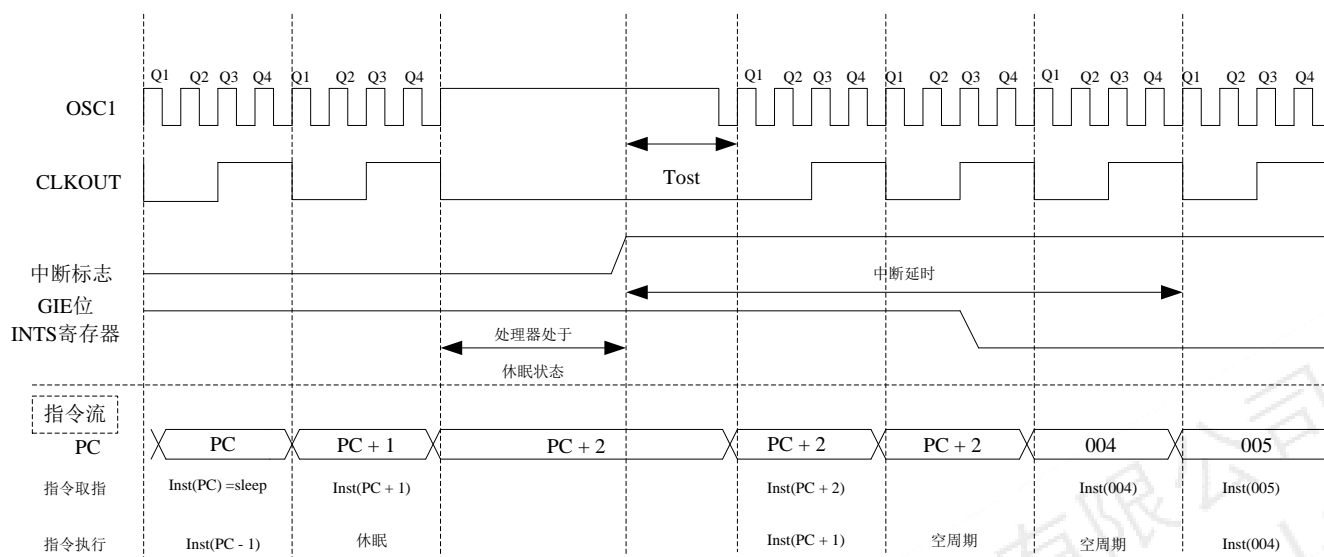
### 4.3.2 使用中断唤醒

当禁止全局中断 (GIE 被清零) 时，并且有任一中断源将其中断允许位和中断标志位置 1，将会发生下列事件之一：

- 如果在执行 SLEEP 之前产生了中断，那么 SLEEP 指令将被作为一条 NOP 指令执行。因此，WDT 及其预分频器和后分频器（如果使能）将不会被清零，并且 TF 位将不会被置 1，同时 PF 位也不会被清零。
- 如果在执行 SLEEP 指令期间或之后产生了中断，那么器件将被立即从休眠状态唤醒。SLEEP 指令将在唤醒之前执行完毕。因此，WDT 及其预分频器和后分频器（如果使能）将被清零，并且 TF 位将被置 1，同时 PF 位也将被清零。

即使在执行 SLEEP 指令之前，检测到标志位为 0，它也可能在 SLEEP 指令执行完毕之前被置 1。要确定是否执行了 SLEEP 指令，可测试  $\overline{PF}$  位。如果  $\overline{PF}$  位置 1，则说明 SLEEP 指令被当作一条 NOP 指令执行了。

在执行 SLEEP 指令之前，必须先执行一条 CLRWT 指令，来确保将 WDT 清零。详情请参见图 4-3。



注：中断延时在 GIE = 1 处理器唤醒后，将调用 004H 的 ISR 程序，如果 GIE=0 时，程序将继续执行。

图 4-3：通过中断将器件从休眠状态唤醒

## 5.0 中断

YS65F804 有以下多种中断源：

- PORTA 电平变化中断
- Timer0 溢出中断
- Timer1 溢出中断
- 触摸中断

INTS 寄存器除 CS 中断控制位外，包含其他所有中断允许位、中断标志位及全局中断允许位。全局中断允许位 **GIE** 在置 1 时允许所有未屏蔽的中断，而在清零时禁止所有中断。复位时 **GIE** 被清零。

响应中断时，自动发生以下动作：

- **GIE** 被清零以禁止任何其他中断。
- 返回地址被压入堆栈。
- 在 **PC** 中装入 0004h。

执行“从中断返回”指令 **RTFI** 退出中断程序并将 **GIE** 位置 1，从而重新使能未屏蔽的中断。

注： 1: 各中断标志位的置1不受相应的中断屏蔽位或GIE位状态的影响。  
2: 当执行一条清零GIE位的指令后，任何等待在下一周期执行的中断都将被忽略。当GIE位被再次置1后，被忽略的中断仍会继续等待处理。

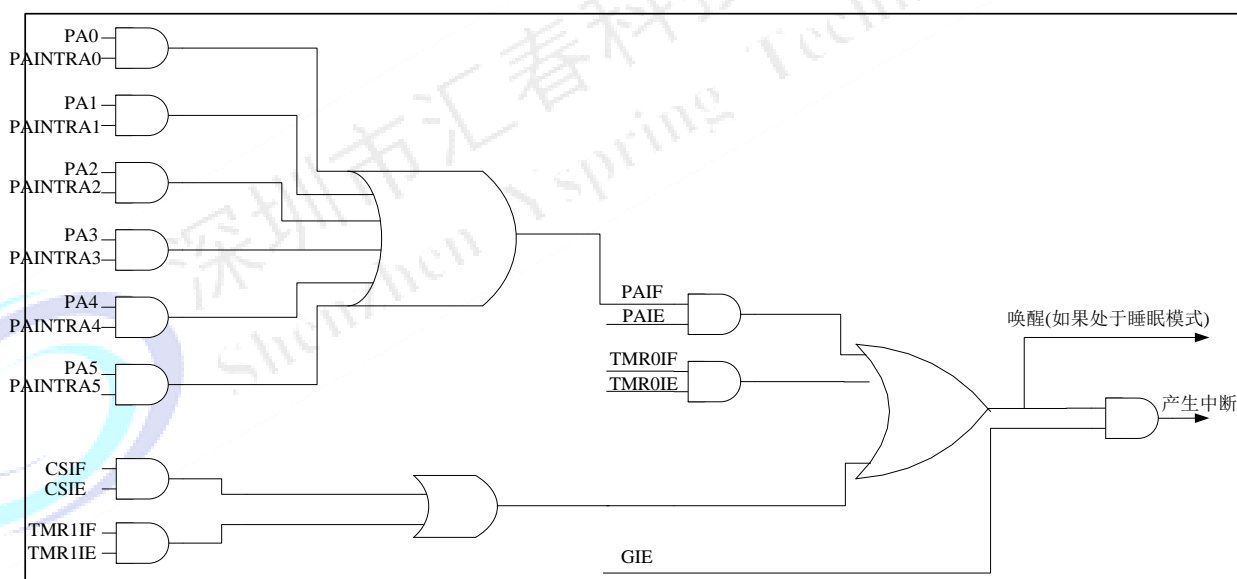


图 5-1：中断逻辑

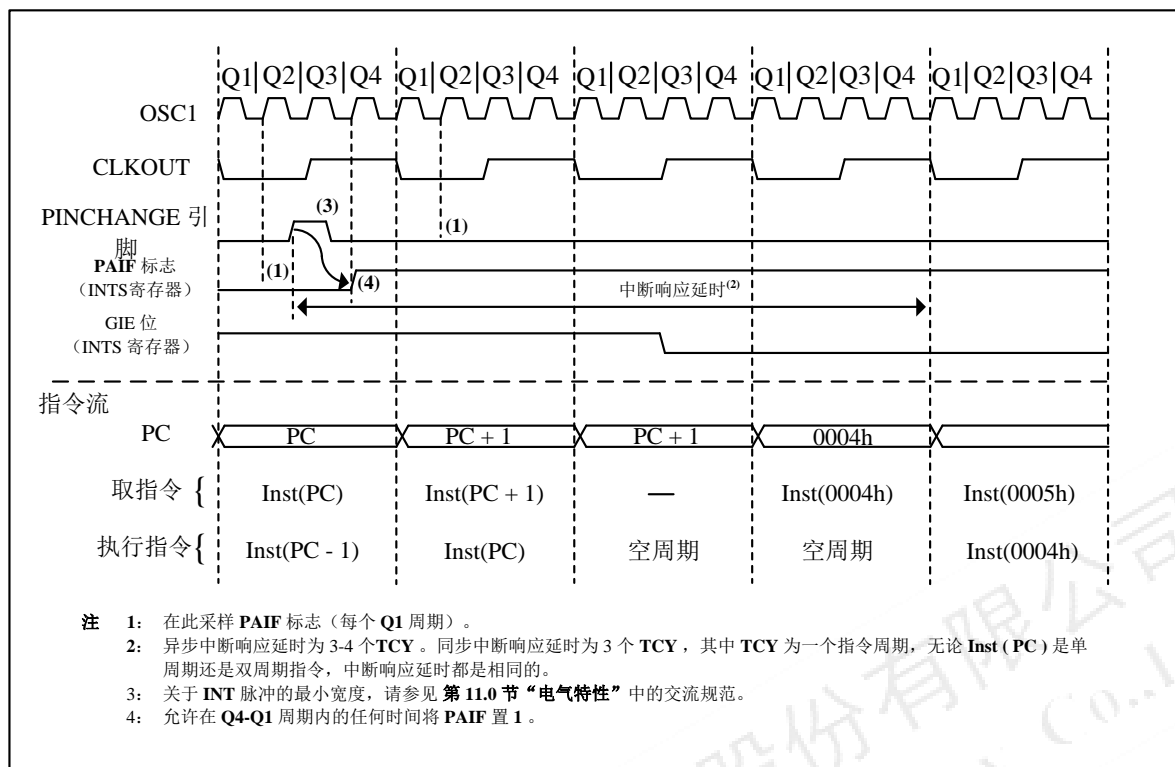


图 5-2: PIN CHANGE 引脚中断时序

## 5.1 中断特殊功能寄存器

### 5.1.1 INTS 寄存器

INTS 寄存器是可读写的寄存器，包含 TMR0 寄存器溢出、PORTA 电平变化等中断的各种允许和标志位。

注： 当有中断条件产生时，不管相应的中断允许位或INTS寄存器的全局允许位GIE状态如何，中断标志位都将置1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

#### 寄存器 0BH/8BH/10BH/18BH: 中断控制寄存器 (INTS)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位，读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

bit7            **GIE:** 全局中断允许位  
 1 = 允许所有未屏蔽的中断  
 0 = 禁止所有中断  
 bit6            **CSIF:** 触摸中断标志位  
 1 = 发生了触摸中断  
 0 = 未发生触摸中断

bit5	<b>TIS:</b> Timer0溢出中断允许位 1 = 允许Timer0中断 0 = 禁止Timer0中断
bit4	<b>T1IE:</b> Timer1中断允许位 1 = 允许Timer1溢出中断 0 = 禁止Timer1溢出中断
bit3	<b>PAIE:</b> PORTA电平变化中断允许位 <sup>(1)</sup> 1 = 允许PORTA电平变化中断 0 = 禁止PORTA电平变化中断
bit2	<b>TIF:</b> Timer0溢出中断标志位 <sup>(2)</sup> 1 = Timer0寄存器已经溢出（必须用软件清零） 0 = Timer0寄存器没有溢出
bit1	<b>T1IF:</b> Timer1溢出中断标志位 1 = Timer1寄存器已经溢出（必须用软件清零） 0 = Timer1寄存器未溢出
bit0	<b>PAIF:</b> PORTA电平变化中断标志位 <sup>(1)</sup> 1 = 至少一个PORTA<5:0>引脚的电平状态发生了改变（必须用软件清零） 0 = 没有一个PORTA<5:0>引脚的电平状态发生改变
注	1: 必须同时使能PAINTR寄存器。 2: 当TMR0计满回零时，TIF位置1。复位时TMR0的状态不变，它应该在清零TIF位之前被初始化。

表 5-1: 与中断相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000 0000	0000 0000
PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	--00 0000	0000 0000

图注: x = 未知, u = 不变, — = 未实现 (读为 0), q = 取值视情况而定。中断模块不使用阴影单元。

## 5.2 Timer0 中断

TMR0 寄存器溢出 (FFh → 00h) 会将 INTS 寄存器的 TIF 位置 1。可以通过置 1/清零 INTS 寄存器的 TIS 位来使能/禁止该中断。关于 Timer0 模块的操作，请参见第 7.1 章节“Timer0 模块”。

## 5.3 PORTA 电平变化中断

PORTA 输入电平的变化会使的 PAIF 位置 1。可以通过置 1/清零的 PAIE 位来使能/禁止该中断。此外，可通过 PAINTR 寄存器对该端口的各个引脚进行配置。

注：当读操作正在执行时发生了 I/O 引脚电平变化，则 PAIF 中断标志位可能不会被置 1。

## 5.4 中断的现场保护

在中断期间，仅将返回的 PC 值压入堆栈。通常情况下，用户可能希望在中断期间保存关键寄存器（例如，W 寄存器和状态寄存器）。这必须用软件实现。应将临时保存寄存器 W\_TEMP 和 STATUS\_TEMP 置于 GPR 的末 16 字节中。这 16 个单元是所有存储区共用的，无需分区。这样就简化了现场保护和恢复操作。例 5-1 中所示的代码可用于：

- 保存 W 寄存器



- 保存状态寄存器
- 执行 ISR 代码
- 恢复状态寄存器（和存储区选择位寄存器）
- 恢复 W 寄存器

注：YS65F804通常不需要保存PCLATH。但是，如果要在ISR和主程序中使用计算LJUMP，就必须在ISR中保存和恢复PCLATH。

例 5-1：将状态寄存器和 W 寄存器保存在 RAM 中

```

STWR    W_TEMP          ;保存 W 寄存器到 W_TEMP
SWAPR   STATUS,W        ;将 STATUS 寄存器高低四位交换后保存到 W（目的：不改变标志）
STWR    STATUS_TEMP     ;保存 STATUS 寄存器内容到 STATUS_TEMP
;(ISR)   ; 中断服务程序
SWAPR   STATUS_TEMP,W   ;交换 STATUS_TEMP 高低四个字节
STWR    STATUS          ;恢复标志状态寄存器
SWAPR   W_TEMP,R        ;恢复 W 寄存器值（用交换指令的目的：不改变状态寄存器值）
SWAPR   W_TEMP,W

```

## 6.0 I/O 端口

YS65F804 共有 6 个通用 I/O 引脚。根据使能的外设不同，有些（或全部）引脚不能用作通用 I/O。通常使能了一个外设后，相关的引脚就不能用作通用 I/O 引脚了。

### 6.1 PORTA 和 CPIOA 寄存器

PORTA 是 6 位宽的双向端口。PORTA 对应的数据方向寄存器是 CPIOA。将 CPIOA 位置 1 可以使对应的 PORTA 引脚作为输入引脚（即禁止相应的输出驱动器）。将 CPIOA 位清零将使对应的 PORTA 引脚作为输出引脚（即使能输出驱动器并将输出锁存器的内容置于所选的引脚上）。例 6-1 给出了初始化 PORTA 的方法。

读 PORTA 寄存器将读取引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读—修改—写操作。因此，写一个端口就意味着读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。当 MCLRE=1 时，PA3 读为 0。

例 6-1：初始化 PORTA

```

BCR     STATUS,PAGE    ;Bank0
CLRR   PORTA          ;清除端口 A 输出缓冲器
BSR     STATUS,PAGE    ;Bank1
LDWI   0CH           ;设置端口 A 的<3,2>位为输入，其他输出
STWR   CPIOA          ;
BCR     STATUS,PAGE    ;Bank0

```

寄存器 05H：PORTA 寄存器（PORTA）

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	PA5	PA4	PA3	PA2	PA1	PA0
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-6 未实现：读为0  
 bit5-0 **PA<5:0>**: PORTA I/O引脚位  
 1 = PORTA引脚电平>V<sub>IH</sub>  
 0 = PORTA引脚电平<V<sub>IL</sub>

#### 寄存器 85H: PORTA 三态寄存器 (CPIOA)

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	<b>CPIOA5</b>	<b>CPIOA4</b>	<b>CPIOA3</b>	<b>CPIOA2</b>	<b>CPIOA1</b>	<b>CPIOA0</b>
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位，读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

bit7-6 未实现：读为0  
 bit5-0 **CPIOA<5:0>**: PORTA三态控制位  
 1 = PORTA引脚配置为输入（三态）  
 0 = PORTA引脚配置为输出

## 6.2 其它引脚功能

YS65F804 的 PORTA 端口上的每一个引脚都具有电平变化中断和弱上拉功能 (PA3 做为复位脚时，使能上拉。作为 IO 口时，没有上拉电阻)。下面两个小节将介绍这些功能。

### 6.2.1 弱上拉与下拉

每一个 PORTA 引脚具有各自的可配置内部弱上拉。控制位 PAPHRX 使能或禁止每一个弱上拉。当将端口引脚配置为输出时，其弱上拉电路会自动切断。在 PA3 配置为 MCLR 时，自动启动 PA3 的弱上拉功能。MCLR 上拉不受软件控制。

每一个 PORTA 引脚 (除 PA3) 具有各自的可配置内部弱下拉。控制位 WPDAX 使能或禁止每一个弱下拉。当将端口引脚配置为输出时，其弱下拉电路会自动切断。

#### 寄存器 95H: 弱上拉PORTA寄存器 (PAPHR)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	<b>PAPHR5</b>	<b>PAPHR4</b>	<b>PAPHR3</b>	<b>PAPHR2</b>	<b>PAPHR1</b>	<b>PAPHR0</b>
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位，读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

bit7-6 未实现：读为0  
 bit5-0 **PAPHR<5:0>**: 弱上拉控制位  
 1 = 使能上拉  
 0 = 禁止上拉

#### 寄存器97H: 弱下拉PORTA寄存器 (WPDA)

U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	—	<b>WPDA5</b>	<b>WPDA4</b>	—	<b>WPDA2</b>	<b>WPDA1</b>	<b>WPDA0</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-6	未实现: 读为0
bit5-4	<b>WPDA&lt;5:4&gt;</b> : 弱下拉控制位
	1 = 使能下拉
	0 = 禁止下拉
bit3	未实现: 读为0
bit2-0	<b>WPDA&lt;2:0&gt;</b> : 弱下拉控制位
	1 = 使能下拉
	0 = 禁止下拉

## 6.2.2 电平变化中断

每一个 PORTA 引脚均可分别配置为电平变化中断引脚。控制位 **PAINTR\_x** 使能或禁止每个引脚的中断功能, 在上电复位时禁止电平变化中断。对于已允许电平变化中断的引脚, 则将该引脚上的值同上一次读 PORTA 时锁存的值进行比较。将上一次“不匹配”的输出一起作逻辑或运算, 以便将 INTS 寄存器中 PORTA 电平变化中断标志位 (PAIF) 置 1。

该中断能唤醒休眠下的器件。用户在中断服务程序中通过以下方式清除中断:

- 对 PORTA 进行读或写操作。这将结束引脚电平不匹配条件。
- 将标志位 PAIF 清零。

电平不匹配条件会继续将 PAIF 标志位置 1。而读 PORTA 将结束不匹配条件并允许将 PAIF 标志位清零。锁存器将保持最后一次读取的值不受 MCLR 和欠压复位的影响。在这些复位之后, 如果出现电平不匹配, PAIF 标志位将继续被置 1。

注: 在执行任何 PORTA 操作时如果 I/O 引脚的电平发生变化, 则 PAIF 中断标志位可能不会被置 1。

### 寄存器 96H: 电平变化中断 PORTA 寄存器 (PAINTR)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	<b>PAINTR5</b>	<b>PAINTR4</b>	<b>PAINTR3</b>	<b>PAINTR2</b>	<b>PAINTR1</b>	<b>PAINTR0</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-6	未实现: 读为0。
bit5-0	<b>PAINTR&lt;5:0&gt;</b> : 电平变化中断 PORTA 控制位
	1 = 允许电平变化中断
	0 = 禁止电平变化中断
注	1: 必须允许全局中断允许 (GIE) 以使各中断被识别。

## 6.3 PORTA 引脚说明和引脚图

每个 PORTA 引脚都与其他功能复用。这里将简要说明引脚及其复合功能。

### 6.3.1 PA0 /ISCK/CCH0

图 6-1 给出了此引脚的引脚图。PA0/ISCK/CCH0 引脚可配置为下列功能之一：

- 通用 I/O
- 在线串行编程（In-Circuit Serial Programming™）烧录和调试的时钟 SCK
- 触摸的模拟输入通道 0

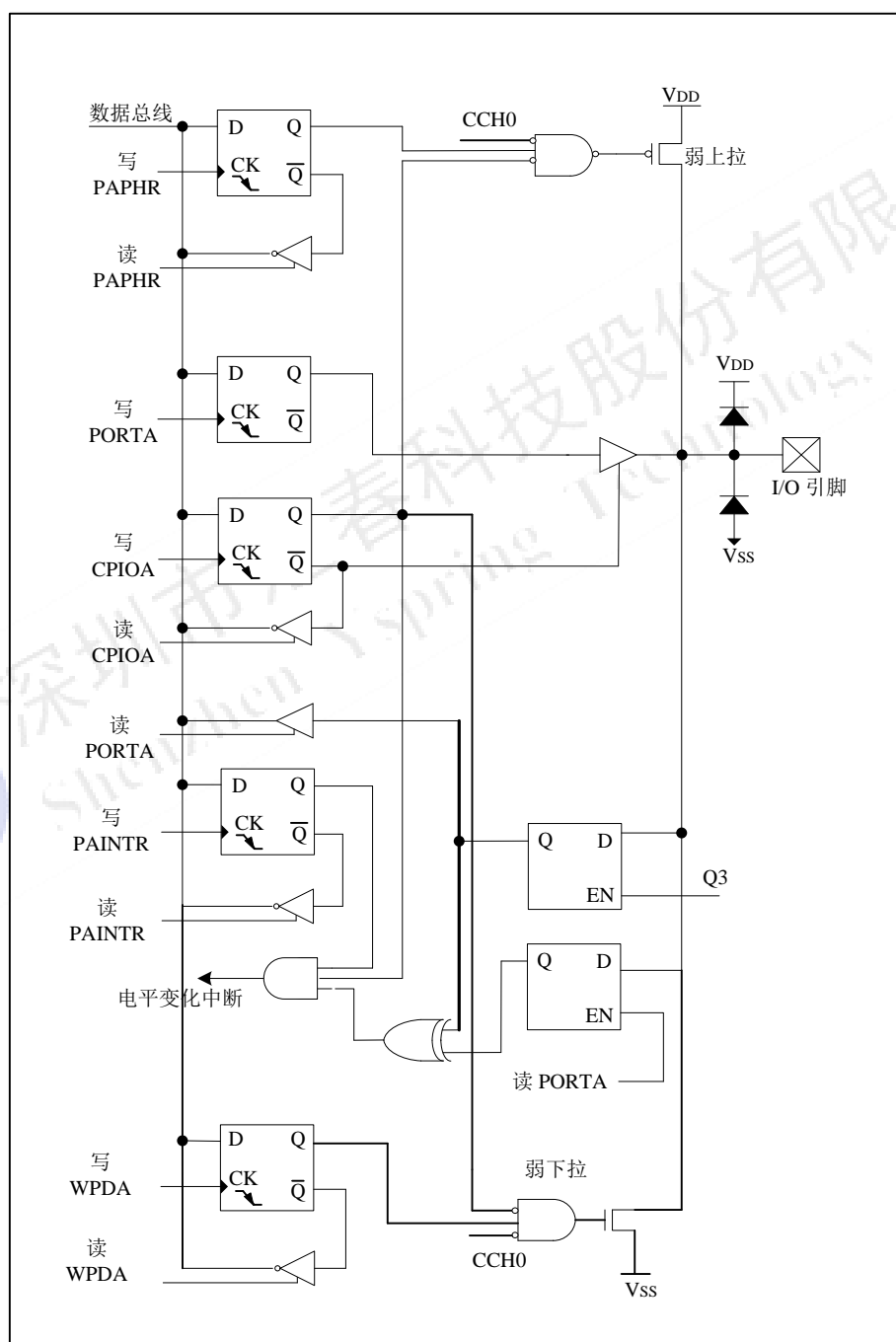


图 6-1: PA0 框图

### 6.3.2 PA1/ISDA/CCH1/PWM0

图 6-2 给出了此引脚的引脚图。PA1/ISDA/CCH1/PWM0 引脚可配置为下列功能之一：

- 通用 I/O
- 在线串行编程烧录和调试的数据端口 SDA
- 触摸的模拟输入通道 1
- 功能转移前 PWM0 输出

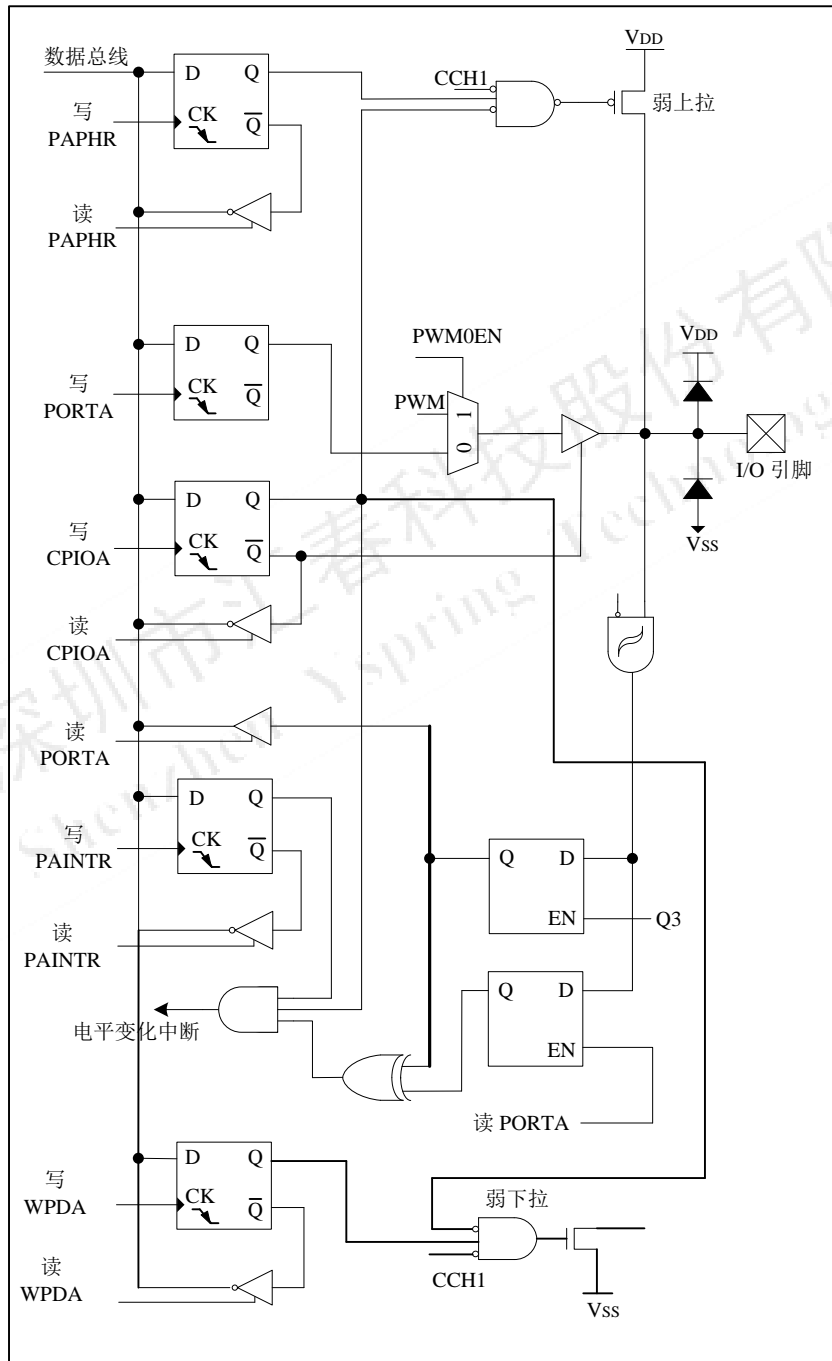


图 6-2: PA1 框图

## 6.3.3 PA2/CCH2/PWM1

图 6-3 给出了此引脚的引脚图。PA2/CCH2/PWM1 引脚可配置为下列功能之一：

- 通用 I/O
- 功能转移前触摸输入通道 2
- 功能转移前 PWM1 输出

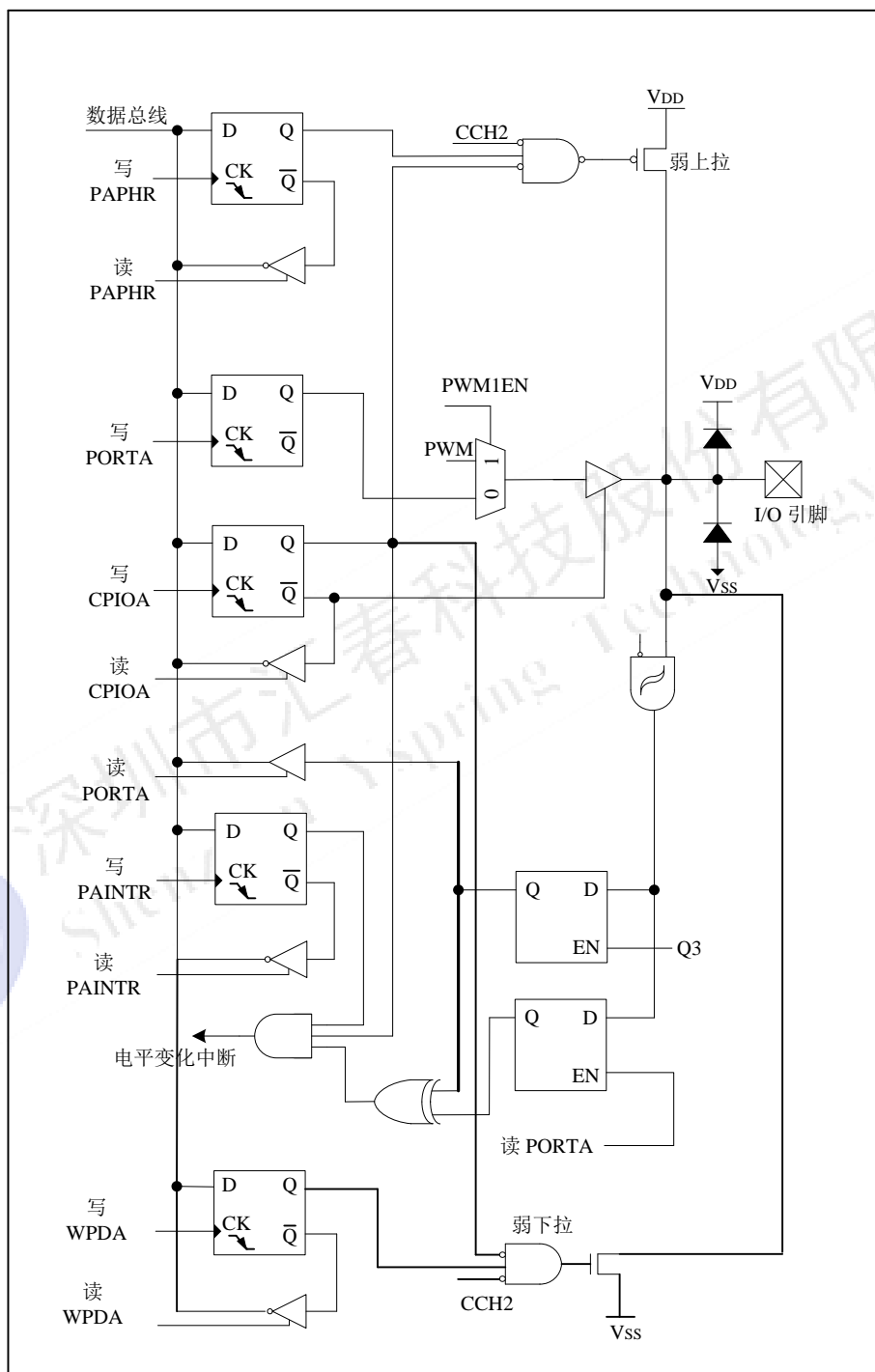


图 6-3: PA2 框图

### 6.3.4 PA3/MCLR/VPP/PWM0

图 6-4 给出了此引脚的引脚图。PA3/MCLR/VPP 引脚可配置为下列功能之一：

- 通用输入；
- 带弱上拉的主清零复位；
- 高压烧写引脚
- 功能转移后 PWM0 输出

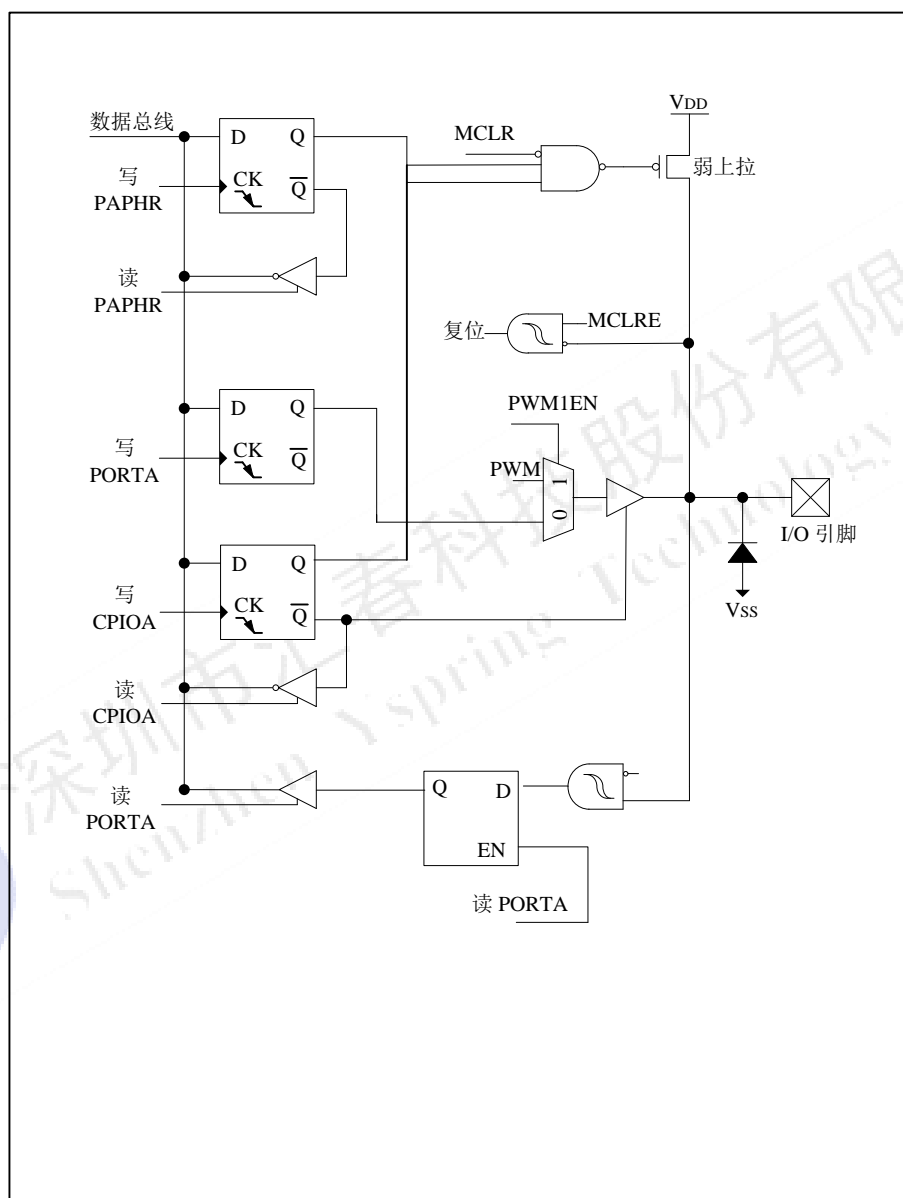


图 6-4: PA3 框图

### 6.3.5 PA4 /PWM1/CCH3

图 6-5 给出了此引脚的引脚图。PA4/CLKOUT/CCH3 引脚可配置为下列功能之一：

- 通用 I/O
- 功能转移后 PWM1 输出
- 触摸输入通道 3

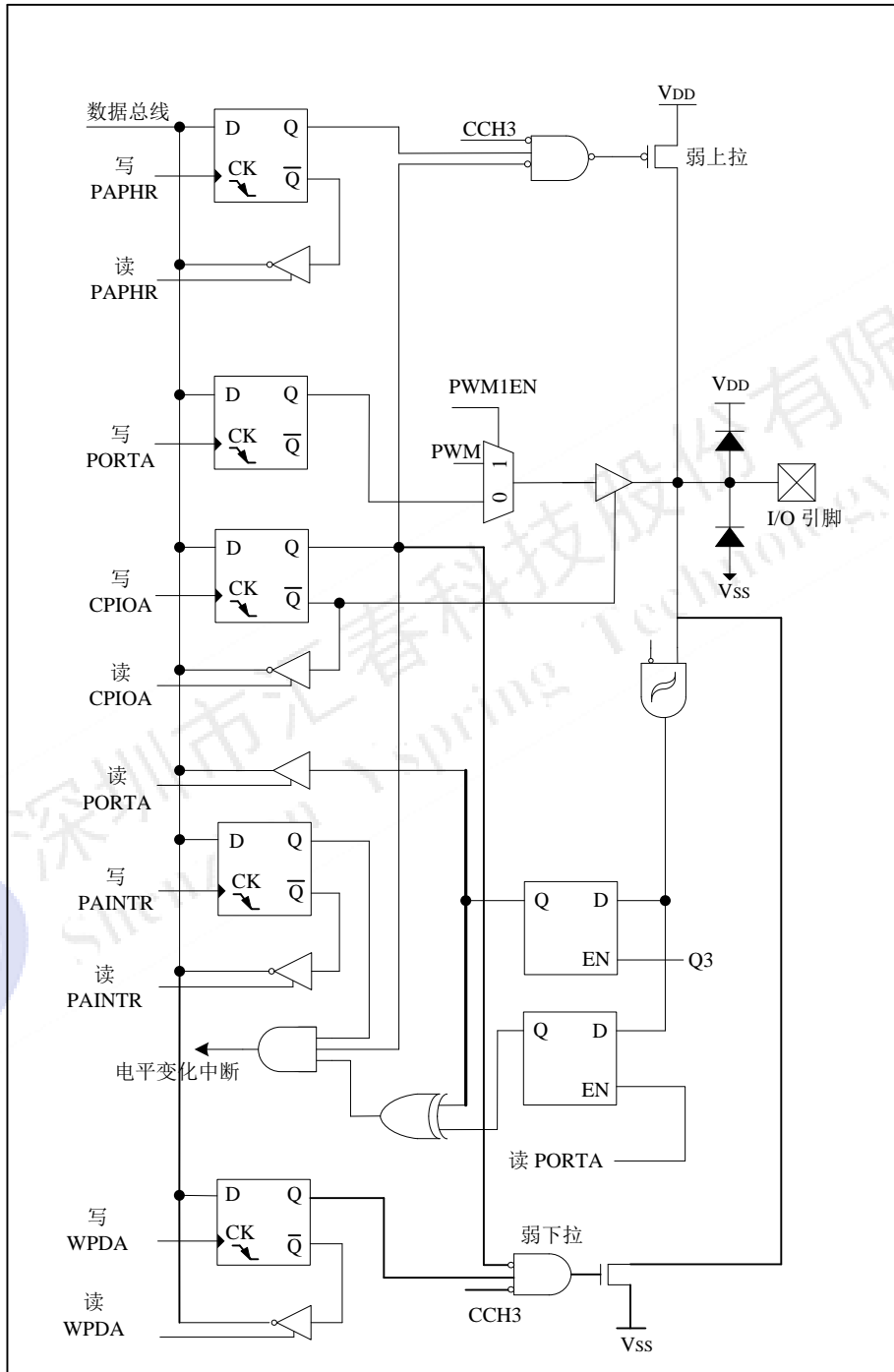


图 6-5: PA4 框图



## 6.3.6 PA5/CCH2

图 6-6 给出了此引脚的引脚图。PA5 引脚可配置为下列功能之一：

- 通用 I/O
- 功能转移后触摸输入通道 2

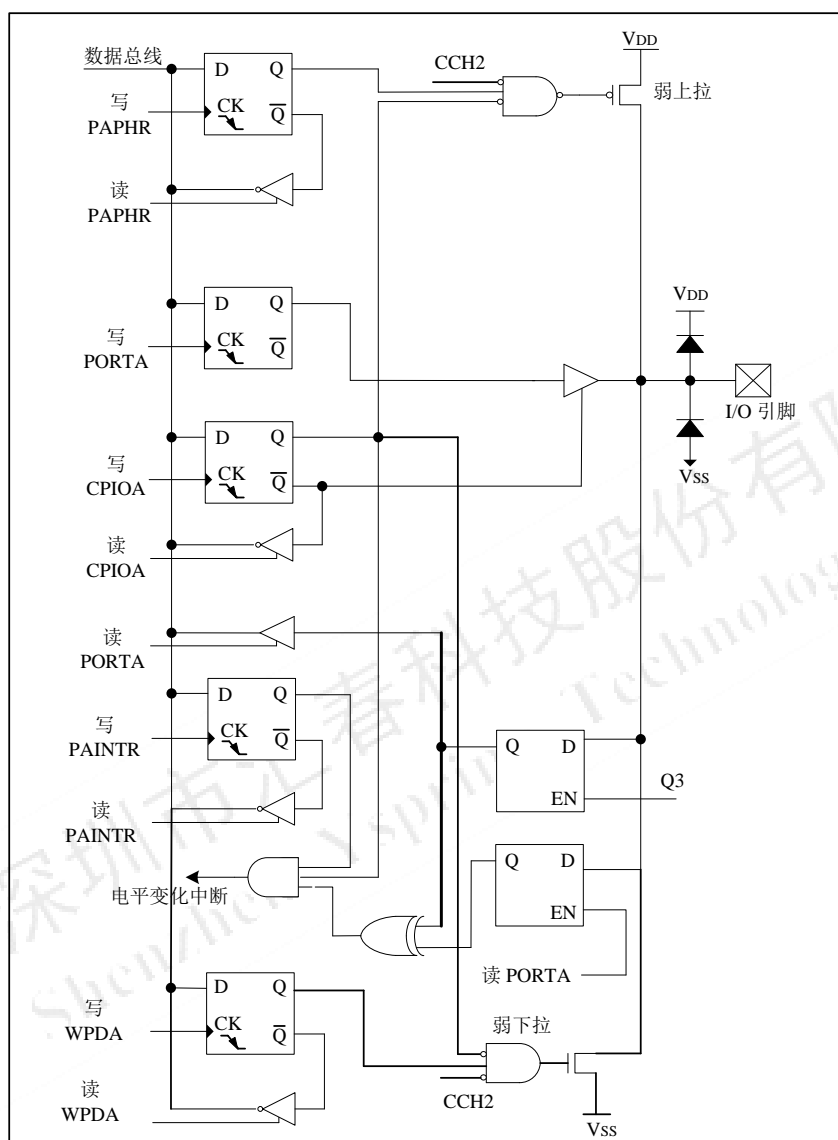


图 6-6: PA5 框图

表 6-1: 与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
PSTA	—	LVDF	LVDEN	SBOREN	MCLR	IER	POR	BOR	--01 --qq	--0u --uu
INTS	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000 0000	0000 0000
PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	--00 0000	--00 0000
OPT_REG	—	—	—	T0EN	PSC	PS2	PS1	PS0	--11 1111	--11 1111
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--xx xxxx	--uu uuuu
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
PAPHR	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0	--00 0000	--00 0000
WPDA	—	—	WPDA5	WPDA4	—	WPDA2	WPDA1	WPDA0	--00 -000	--00 -000

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。PORT A 不使用阴影单元。

## 7.0 定时器及 PWM

### 7.1 Timer0 模块

Timer0 模块是 8 位定时器/计数器，具备以下特性：

- 8 位定时器/计数器寄存器（TMR0）
- 8 位预分频器（与看门狗定时器共用）
- 可编程内部时钟源
- 可编程外部时钟边沿选择
- 溢出中断
- 定时器/计数器重载功能

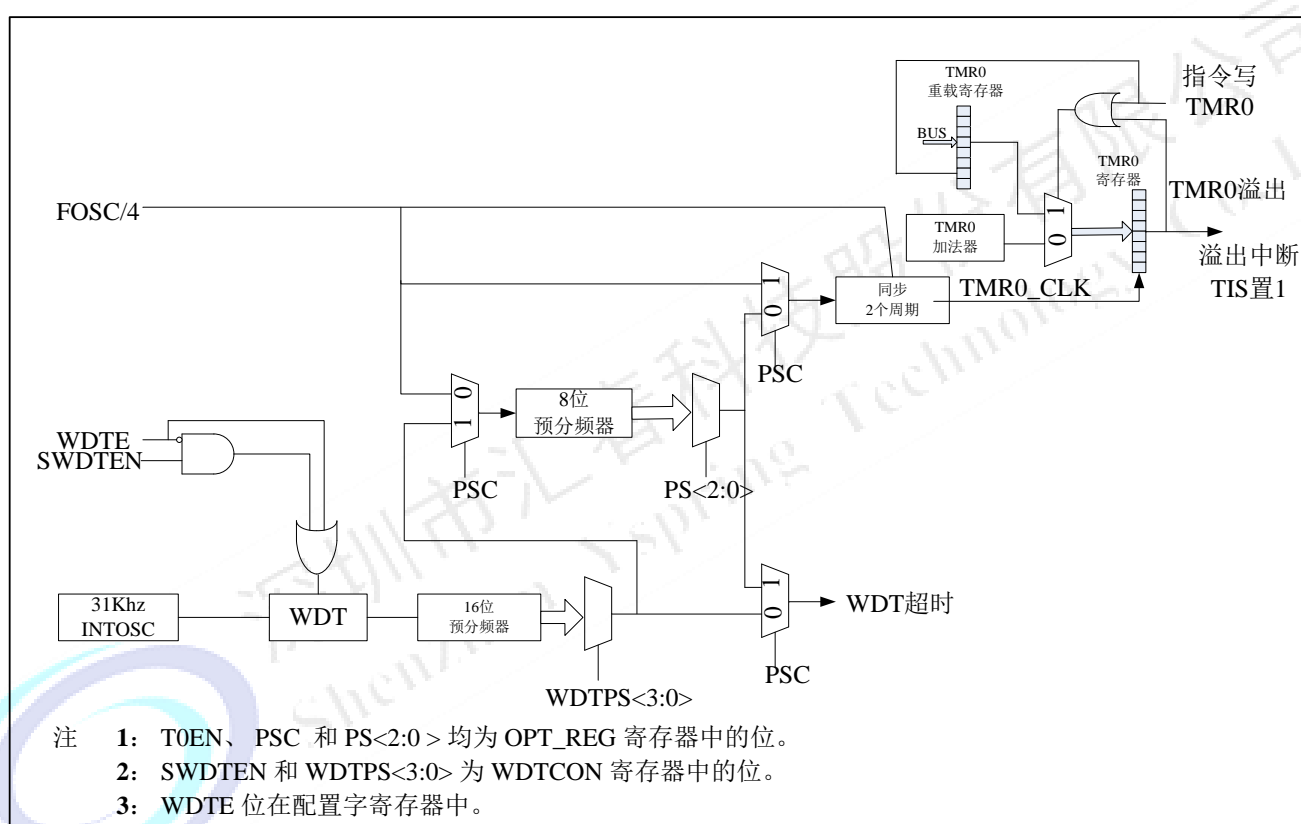


图 7-1: Timer0 模块的框图

#### 7.1.1 Timer0 的工作原理

Timer0 是一个 8 位并具有重载功能的定时器/计数器，读 TMR0，读的是计数器的值，写 TMR0，写的是重载寄存器里的值。注意：上电时重载寄存值是随机的，程序需要进行初始化。

##### 寄存器 01H: Timer0 模块寄存器（TMR0）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
<b>TMR0&lt;7:0&gt;</b>							
bit7							bit0

图注: W = 可写位 U = 未实现位, 读为0 x = 未知

R = 可读位	1 = 置1	0 = 清零
-n = POR时的值		

bit7-0      **TMR0<7:0>**: 8位定时器/计数器寄存器

## 7.1.2 8 位定时器模式

Timer0 的开关使能位 T0EN 见寄存器 OPT\_REG，只有打开使能位，Timer0 才能工作。作为定时器使用时，Timer0 模块将在每个指令周期递增（无预分频器）。当 TMR0 被写入时，写入后将立即禁止 2 周期递增。

注：在 TMR0 被写入时，为了计入两个指令周期的延时，可以调整写入 TMR0 寄存器的值。

## 7.1.3 软件可编程预分频器

Timer0 或看门狗定时器（WDT）之一可使用一个软件可编程预分频器，但两者不能同时使用。预分频器的分配由 OPT\_REG 寄存器的 PSC 位控制，要将预分频器分配给 Timer0，必须将 PSC 位清零。Timer0 模块的预分频比有 8 种选项，从 1:2 至 1:256，预分频比可通过 OPT\_REG 寄存器的 PS<2:0>位进行选择。要使 Timer0 模块得到 1:1 的预分频比，必须将预分频器分配给 WDT 模块。预分频器不可读写。当预分频器被分配给 Timer0 模块时，所有写入 TMR0 寄存器的指令均会将预分频器清零。当预分频器被分配给 WDT 时，一条 CLRWT 指令将同时清零预分频器和 WDT。

### 7.1.3.1 在 Timer0 和 WDT 模块间切换预分频器

由于预分频器可分配给 Timer0 或 WDT，因此在切换预分频比时可能产生不想要的器件复位。当把预分频器从 Timer0 切换到 WDT 模块时，必须执行例 7-1 所示的指令序列。

例 7-1: 更改预分频器 (TIMER0→WDT)

```
BCR      STATUS,PAGE      ;Bank 0
CLRWT                                         ;清看门狗定时器
CLRR     TMR0              ;清 TMR0
BSR      STATUS,PAGE      ;Bank 1
LDWI     b'10001111'      ;禁止上拉，TMR0 分频给看门狗，分频比 1:128
TMODE                                         ;
CLRWT                                         ;清看门狗
BCR      STATUS,PAGE      ;Bank 0
```

当将预分频器从 WDT 切换到 Timer0 模块时，必须执行以下指令序列（见例 7-2）。

例 7-2: 更改预分频器 (WDT→TIMER0)

```
CLRWT                                         ;清看门狗
BSR      STATUS,PAGE      ;Bank 1
LDWI     b'xxxx0xxx'      ;将分频器分配给 TMR0
TMODE                                         ;
BCR      STATUS,PAGE      ;Bank 0
```

## 7.1.4 Timer0 中断

TMR0 寄存器从 FFh 溢出到 00h 时，Timer0 将产生中断。INTS 寄存器的 TIF 中断标志位将在每次 TMR0 寄存器溢出时被置 1，无论是否允许了 Timer0 中断。必须用软件将 TIF 位清零。Timer0 中断允许位是 INTS 寄存器的 TIS 位。

注： 由于定时器在休眠时被冻结，因此Timer0中断无法将处理器从休眠中唤醒。

表 7-1: 与 TIMER0 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
TMR0	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu
INTS	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000 0000	0000 0000
OPT_REG	—	—	—	T0EN	PSC	PS2	PS1	PS0	1111 1111	1111 1111
PWMCON	—	—	—	—	PWM1SEL	PWM0SEL	PWM1EN	PWM0EN	---- 0000	---- 0000
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111

图注： x = 未知， u = 不变， — = 未实现（读为 0）。Timer0 不使用阴影单元。

## 7.2 Timer1 模块

Timer1 模块是 16 位定时器/计数器，具备以下特性：

- 16 位定时器/计数器寄存器（TMR1H:TMR1L）
- 可编程内部时钟源
- 3 位预分频器
- 溢出中断

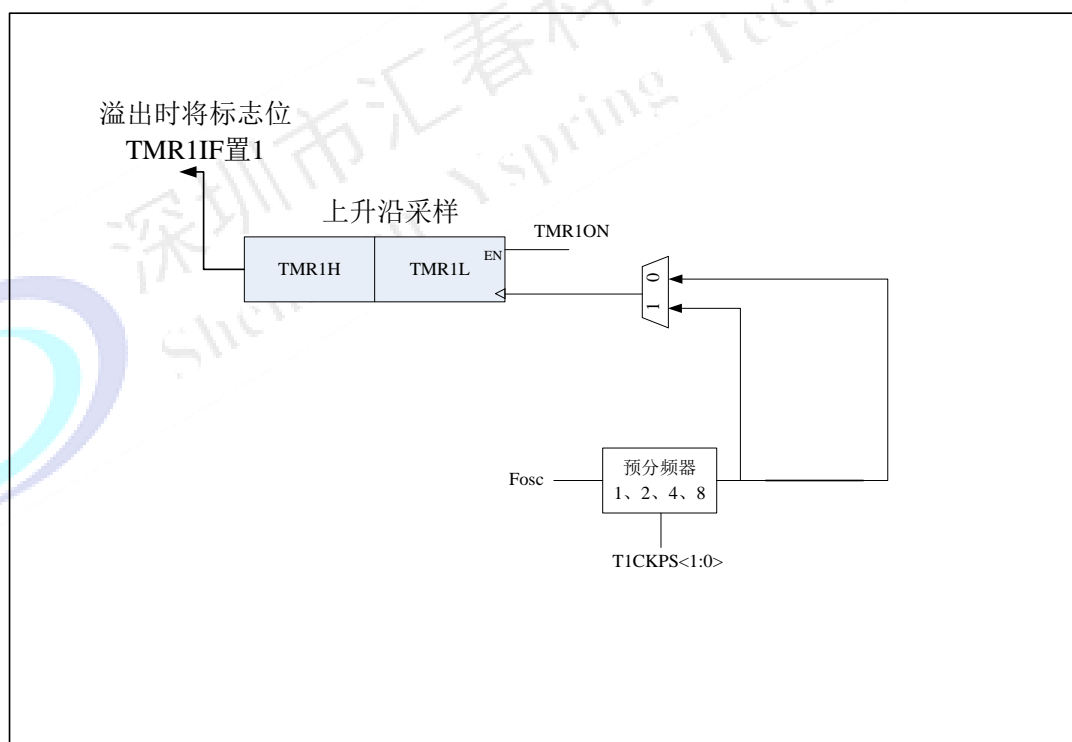


图 7-2: Timer1 模块的框图

### 7.2.1 Timer1 的工作原理

Timer1 模块是 16 位递增计数器，时钟源固定为 FOSC。通过一对寄存器 TMR1H:TMR1L 访问。对 TMR1H 或 TMR1L 的写操作将直接更新计数器。Timer1 控制寄存器 T1STA 用于控制 Timer1 禁止/使能、以及预分频

器。

#### 寄存器 10H: TIMER1控制寄存器 (T1STA)

U-0	U-0	R/W-0	R/W-0	U-0	U-0	R-0	R/W-0
—	—	T1CKPS1	T1CKPS0	—	—	—	TMR1ON
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7-6 未实现: 读为0
- bit5-4 **T1CKPS<1:0>**: Timer1输入时钟预分频比选择位  
 11 = 1:8  
 10 = 1:4  
 01 = 1:2  
 00 = 1:1
- bit3 未实现: 读为0
- bit2 未实现: 读为0
- bit1 未实现: 读为0
- bit0 **TMR1ON**: Timer1打开位  
 1 = 使能Timer1  
 0 = 停止Timer1

#### 寄存器 0EH: 16位Timer1低字节的保持寄存器 (TMR1L)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
<b>TMR1&lt;7:0&gt;</b>							
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7-0 **TMR1<7:0>**: 16位Timer1定时器/计数器低字节寄存器

#### 寄存器 0FH: 16位Timer1高字节的保持寄存器 (TMR1H)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
<b>TMR1&lt;15:8&gt;</b>							
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7-0 **TMR1<15:8>**: 16位Timer1定时器/计数器高字节寄存器

## 7.2.2 Timer1 的预分频器

Timer1 有四种预分频选项, 可对时钟输入进行 1、2、4 或 8 分频。T1STA 寄存器的 T1CKPS 位控制预分频计数器。预分频计数器不可直接读写; 但当发生对 TMR1H 或 TMR1L 的写操作时, 预分频计数器被清零。

## 7.2.3 Timer1 中断

Timer1 的一对寄存器 (TMR1H:TMR1L) 递增至 FFFFh 后返回 0000h。Timer1 计满返回时, INTS 寄存器的 Timer1 中断标志位被置 1。为确保计满返回时产生中断, 您必须将以下位置 1:

- T1STA 寄存器的 TMR1ON 位
- INTS 寄存器的 T1IE 位
- INTS 寄存器的 GIE 位

在中断服务程序中将 T1IF 位清零将清除中断标志。

注: TMR1H:TMR1L这对寄存器和T1IF位应在允许中断前清零。

## 7.2.4 与 Timer1 相关的寄存器汇总

表 7-2: 与 TIMER1 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
PWMCON	—	—	—	—	PWM1SEL	PWM0SEL	PWM1EN	PWM0EN	---- 0000	---- 0000
INTS	GIE	CSIF	TIS	T1IE	PAIE	TIF	T1IF	PAIF	0000 0000	0000 0000
TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx xxxx	uuuu uuuu
T1STA	—	—	T1CKPS1	T1CKPS0	—	—	—	TMR1ON	0000 0000	0000 0000

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。Timer1 模块不使用阴影单元。

## 7.3 PWM 模块

脉宽调制 (PWM) 是通过一种完全开启和完全关闭之间进行快速切换而为负载供电的方案。PWM 信号类似于方波, 信号的高电平部分视为开启状态, 信号的低电平部分视为关闭状态。高电平部分 (也称脉宽) 可随变, 并以步幅为单位进行定义。施加的步幅数量越多 (这会增大脉宽), 为负载提供的电量就越多。施加的步幅数量降低时 (这会缩短脉宽), 提供的电量就会下降。PWM 周期定义为一个完整周期的持续时间, 或者开启和关闭时间相加的总时间。PWM 分辨率定义可以在单个 PWM 周期中出现的最大步幅数量。分辨率越高, 就可以精确地控制脉宽时间, 从而更精确地控制在负载上的供电量。占空比这一术语描述开启时间与关闭时间之间以百分比形式表示的比例, 0%代表完全关闭, 100%代表完全开启。占空比越低, 对应的供电量就越低; 占空比越高, 对应的供电量就越高。

### 7.3.1 PWM 的输出

PWMCON 的 PWMxSEL 为零时, PWM 将在 PA1 与 PA2 引脚上输出; PWMxSEL 置一时, PWM 将在 PA3 与 PA4 引脚上输出。PWM 模式将在 PWM 输出引脚产生频率相同、脉宽可调制的 2 路 PWM 信号。PWM 使能时, 当{TMR1H,TMR1L} == {T1PRLH, T1PRLL}时, 下一计数周期 Timer1 计数器清零。当{TMR1H,TMR1L} == 0 时, PWM 输出高电平, 当{TMR1H,TMR1L} == {PWMXH, PWMXL}时, PWM 输出低电平。其占空比, 周期和分辨率由以下寄存器决定:

- T1PRLL
- T1PRLH
- PWMxL
- PWMxH

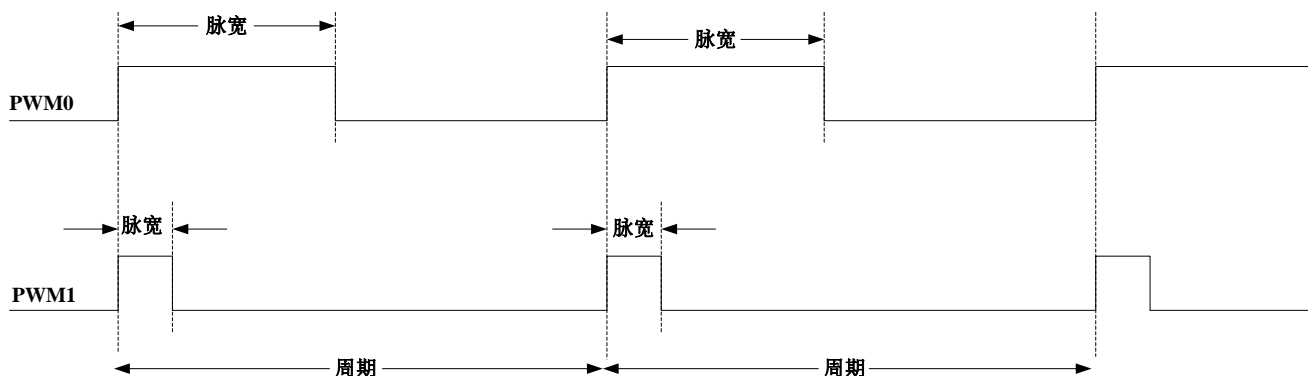


图7-4: PWM输出

### 7.3.2 PWM 的周期

PWM 周期由 Timer1 的 T1PRLH、T1PRLl 寄存器决定，其计算公式如下：

$$\text{PWM 周期} = [\{T1PRLH, T1PRLl\} + 1] * T_{osc} * TMR1 \text{ 预分频值}$$

当 TMR1 等于  $\{T1PRLH, T1PRLl\} + 1$  时，下一次递增周期将发生以下三个事件：

- TMR1 被清零；
- PWM 引脚被置 1（例外：若 PWM 占空比 = 0%，引脚不被置 1）；
- PWM 占空比重载寄存器值锁存到占空比比较寄存器；

### 7.3.3 PWM 的占空比

通过对以下几个寄存器写入 10 位值可设定 PWM 占空比：PRLxH, PRLxL 为占空比重载寄存器，PWMxL、PWMxH 为占空比比较寄存器（只读）。脉冲宽度直到周期完成时才将占空比重载寄存器的值锁存到占空比比较寄存器，这样可以防止程序改变占空比时造成脉冲宽度跳变。用户设定占空比只需写占空比重载寄存器，而不是占空比比较寄存器。脉冲宽度计算公式和占空比公式如下：

$$\begin{aligned} \text{脉冲宽度} &= [\{PWMxH, PWMxL\} + 1] * T_{osc} * TMR1 \text{ 预分频值。} \\ \text{占空比} &= [\{PWMxH, PWMxL\} + 1] / [\{T1PRLH, T1PRLl\} + 1] \end{aligned}$$

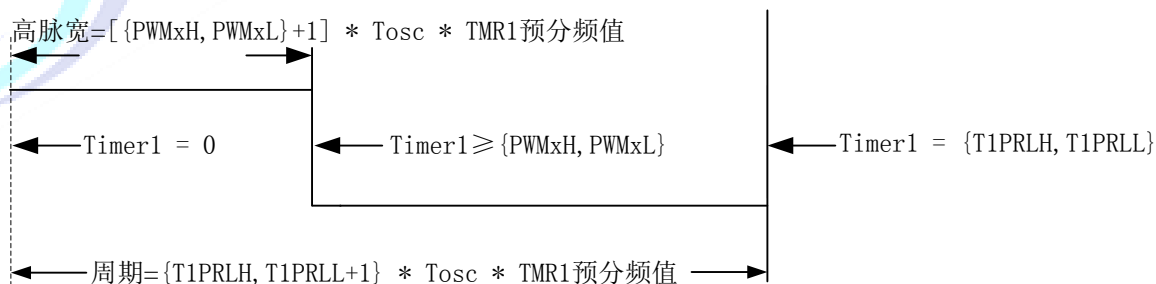


图 7-5: 周期、占空比示意图

### 7.3.4 PWM 的分辨率

分辨率决定某个周期的有效占空比。例如，10 位分辨率有 1024 个分立的占空比，而 8 位分辨率则有 256 个分立的占空比。 $[\{T1PRLH, T1PRLl\} + 1]$  为 1024 时产生 10 位的最大 PWM 分辨率。分辨率是

[[T1PRLH,T1PRLL]]的函数，其公式如下：

$$\text{PWM分辨率} = \frac{\text{Log}([T1PRLH,T1PRLL] + 1)}{\text{Log}[2]} \text{位}$$

表 7-3: PWM 频率和分辨率示例 (Fosc = 16MHz)

PWM 频率	1.95KHz	3.91 kHz	7.8 kHz	250 kHz	500 kHz	667 kHz
定时器预分频比 (1、2、4 或 8)	8	4	2	1	1	1
[T1PRLH,T1PRLL]的值	0x3FF	0x3FF	0x3FF	0x3F	0x1F	0x17
最大分辨率 (位)	10	10	10	8	7	6.6

表 7-4: PWM 频率和分辨率示例 (Fosc = 8MHz)

PWM 频率	9.8kHz	19.6 kHz	39.2 kHz	308 kHz	615 kHz	800kHz
定时器预分频比 (1、2、4 或 8)	8	4	2	1	1	1
[T1PRLH,T1PRLL]的值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率 (位)	8	8	8	6	5	5

### 7.3.5 PWM 输出配置

以下是配置PWM输出的步骤示例：

- 1、端口配置：  
选择PA1、PA2作为PWM输出  
配置PA1，PA2为输出口
- 2、配置周期：  
配置TIMER1控制寄存器 (T1STA)，设定TIMER1为定时器模式，并设定预分频比  
配置T1PRLH、T1PRLL (详见[7.3.2 PWM的周期](#))
- 3、配置占空比：  
配置占空比重载寄存器PRLxH，PRLxL
- 4、使能PWM：  
设置PWM控制寄存器 PWMCON
- 5、使能TIMER1  
置位TMR1ON，此时将会使PRLxH，PRLxL载入PWMxH，PWMxL

### 7.3.6 PWM 的寄存器定义

寄存器1AH: PWM控制寄存器 (PWMCON)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PWM1SEL	PWM0SEL	PWM1EN	PWM0EN
bit7						bit0	
图注:							
R = 可读位		W = 可写位		U = 未实现位，读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

- Bit5-4 未实现：读为0
- Bit3 **PWM1SEL**: PWM1输出引脚选择位  
0 = PWM1 由 PA2 输出  
1 = PWM1 由 PA4 输出
- Bit2 **PWM0SEL**: PWM0输出引脚选择位



	0 = PWM0 由 PA1 输出
	1 = PWM0 由 PA3 输出
Bit1	<b>PWM1EN</b> : PWM1输出使能位
	0 = PWM1 禁止输出
	1 = PWM1允许输出
bit0	<b>PWM0EN</b> : PWM0输出使能位
	0 = PWM0 禁止输出
	1 = PWM0允许输出

## 寄存器 117H: PWM0/1周期寄存器的低字节 (T1PRL)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>T1PRL</b>							
bit7				bit0			
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

Bit7-0 **T1PRL**: PWM0/1周期寄存器的低8位

## 寄存器 19AH: PWM0/1周期寄存器的高2位 (T1PRLH)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
							<b>T1PRLH</b>
bit7				bit0			
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

Bit7-0 **T1PRLH**: PWM0/1周期寄存器的高2位

## 寄存器 112H: PWM0占空比重载寄存器低字节 (PRL0L)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>PRL0L</b>							
bit7				bit0			
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

Bit7-0 **PRL0L**: PWM0占空比重载寄存器的低8位

## 寄存器 192H: PWM0占空比比较寄存器低字节 (PWM0L)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
<b>PWM0L</b>							
bit7				bit0			
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

bit7-0 **PWM0L**: PWM0占空比比较寄存器低8位

## 寄存器 198H: PWM0占空比较寄存器高2位与占空比重载寄存器高2位(EPR0)

R-0	R-0	R-0	R-0	R-0	R-0	R/W-0	R/W-0
—	—	—	—	PWM0H		PRL0H	
bit7						bit0	
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

bit7-4 未实现位

bit3-2 **PWM0H**: PWM0 占空比较寄存器高 2 位。bit1-0 **PRL0H**: PWM0占空比重载寄存器高2位。

## 寄存器 113H: PWM1占空比重载寄存器低字节 (PRL1L)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PRL1L							
bit7						bit0	
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

bit7-0 **PRL1L**: PWM1占空比重载寄存器的低8位

## 寄存器 193H: PWM1占空比较寄存器低字节 (PWM1L)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PWM1L							
bit7						bit0	
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

bit7-0 **PWM1L**: PWM1占空比较寄存器低字节

## 寄存器 199H: PWM1占空比较寄存器高2位与占空比重载寄存器高2位 (EPR1)

R-0	R-0	R-0	R-0	R-0	R-0	R/W-0	R/W-0
—	—	—	—	PWM1H		PRL1H	
bit7						bit0	
图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
-n = POR时的值		1 = 置1		0 = 清零		x = 未知	

bit7-4 未实现位

bit3-2 **PWM1H**: PWM1 占空比较寄存器高 2 位。bit1-0 **PRL1H**: PWM1 占空比重载寄存器高 2 位。

## 8.0 电容触摸模块

对于YS65F804，触摸模块最多可监视5个输入（CHSEL<3:0>）；要使用触摸功能，用户必须：

- 通过设置CHSELL寄存器选择引脚为触摸功能。
- 将相应的CPIO位置1。
- 设置触摸相关参数（触摸灵敏度、初始化环境值、扫描时间等，详细说明见下文）运行软件算法

### 8.1 原理说明

YS65F804 主要检测的是电容充电和放电的时间。首先，由一个 GPIO 对电容 Cx 进行充电；同时开启计时器进行计时；随着充电的进行，Cx 的电压中不断升高，最终它将会操作某个门限电压 V，当其超过门限电压 V 后，GPIO 将会检测到这个事件，同时停止计时器并读出此时的数值。这样，就完成了一次充电计时过程，当手指接触到触摸按键时，Cx 将会变大，显然，充电时间也会变长。通过不断比较每次充电的时间，很自然地就能得知当前是否有按键被“按下”。

### 8.2 扫描周期定时器

触摸模块以慢时钟作为时钟源（LFINTOSC），硬件将慢时钟 64 分频后作为扫描周期的定时时基。也就是扫描一帧最短时间为 2ms；最长时间为 500ms。

### 8.3 配置说明

要使用触摸功能，必须先配置触摸门限差值以及触摸门限初值，配置{CSTMRL}设定帧频率，配置{CHSELL}选择触摸通道，配置 CSACU[2:0]选择通道扫描帧次数，配置 THCTL 选择工作模式，配置完成后打开触摸使能开关 CSEN，触摸模块开始工作。

### 8.4 电容感应数据区

表 8-1：每个通道电容感应的相关数据：

感应通道编号	感应电容值 (16'b)L:H	门限值 (16'b)L:H	门限差 (12'b)L:H(bit3~0)	门限累加值 (20'b)L(bit7~4):M:H
第 0 通道	0X1A9:0X1A8	0X1A1:0X1A0	0X1B1:0X1B0	0X1B0:0X1B9:0X1B8
第 1 通道	0X1AB:0X1AA	0X1A3:0X1A2	0X1B3:0X1B2	0X1B2:0X1BB:0X1BA
第 2 通道	0X1AD:0X1AC	0X1A5:0X1A4	0X1B5:0X1B4	0X1B4:0X1BD:0X1BC
第 3 通道	0X1AF:0X1AE	0X1A7:0X1A6	0X1B7:0X1B6	0X1B6:0X1BF:0X1BE

## 8.5 寄存器说明

### 寄存器190H: 触摸控制寄存器 < CSCTL >

R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0	R/W-1	R/W-1
<b>CSEN</b>	<b>RAMEN</b>	<b>CSIE</b>	<b>CSBUSY</b>	<b>CSINTS</b>	<b>CSACU2</b>	<b>CSACU1</b>	<b>CSACU0</b>
bit7						bit0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7      **CSEN:** 触摸模块使能位  
1 = 触摸模块打开  
0 = 触摸模块关闭
- bit6      **RAMEN:** CS RAM控制选择位  
1 = CS RAM区域由MCU控制  
0 = CS RAM区域由触摸模块控制
- bit5      **CSIE:** 触摸中断禁止/允许位  
1 = 允许触摸中断  
0 = 禁止触摸中断
- bit4      **CSBUSY:** 触摸扫描标志位  
1 = 触摸扫描正在进行  
0 = 触摸空闲
- bit3      **CSINTS:** 触摸产生方式选择  
1 = 有触摸按键产生中断  
0 = 每帧扫描完产生中断
- bit2-0    **CSACU <2:0>:** 通道每帧扫描次数  
000 = 1  
001 = 2  
010 = 4  
011 = 8 (缺省值)  
100 = 16  
101 = 32  
110 = 64  
111 = 128

### 寄存器191H: 触摸门限自动更新寄存器 < THCTL >

R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1
<b>THFIB</b>	<b>THGAP1</b>	<b>THGAP0</b>	<b>THMOD2</b>	<b>THMOD1</b>	<b>THMOD0</b>	<b>THDS1</b>	<b>THDS0</b>
bit7						bit0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7      **THFIB:** 触摸门限更新禁止/使能  
1 = 禁止更新  
0 = 允许更新
- bit6-5    **THGAP<1: 0>:** 门限抽取间隔帧数  
00 = 3  
01 = 7 (缺省值)  
10 = 15  
11 = 31

- bit4-2 **THMOD<2:0>**: 门限累加次数  
 000 = 2  
 001 = 4  
 010 = 8 (缺省值)  
 011 = 16  
 100 = 32  
 101 = 64  
 110 = 128  
 111 = 256
- bit1-0 **THDS<1:0>**: 在触摸按键释放后门限更新累加所需间隔的帧数  
 00 = 16  
 01 = 32 (缺省值)  
 10 = 64  
 11 = 128

**寄存器197H: 触摸通道掩码低位寄存器 < CHSELL >**

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>CCH2SEL</b>	—	—	—	<b>CHSEL3</b>	<b>CHSEL2</b>	<b>CHSEL1</b>	<b>CHSEL0</b>
bit7				bit0			

图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

- bit7 **CCH2SEL**: 触摸通道2 (CH2) 引脚选择位  
 1 = 触摸通道2 (CCH2) 选择PA5  
 0 = 触摸通道2 (CCH2) 选择PA2
- bit3-0 **CHSEL<3:0>**: 允许/禁止该通道为触摸通道  
 1 = 对应通道配置成触摸通道  
 0 = 对应通道没有配置成触摸通道

**寄存器195H: 触摸通道结果标志低位寄存器 < CHRLTL >**

U-0	U-0	U-0	U-0	R-x	R-x	R-x	R-x
—	—	—	—	<b>CHRLT3</b>	<b>CHRLT2</b>	<b>CHRLT1</b>	<b>CHRLT0</b>
bit7				bit0			

图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

- bit3-0 **CHRLT<3:0>**: 触摸通道结果标志  
 1 = 对应通道有触摸  
 0 = 对应通道没有触摸

**寄存器19CH: 触摸自动扫描定时器寄存器低8位 < CSTMRL >**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
<b>CSTMR7</b>	<b>CSTMR6</b>	<b>CSTMR5</b>	<b>CSTMR4</b>	<b>CSTMR3</b>	<b>CSTMR2</b>	<b>CSTMR1</b>	<b>CSTMR0</b>
bit7				bit0			

图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

- bit7-0 **CSTMR<7:0>**: 触摸扫描周期计数器

## 寄存器19EH: 触摸灵敏度调节寄存器0 &lt; CSANA0 &gt;

U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
—	<b>BSTRM2</b>	<b>BSTRM1</b>	<b>BSTRM0</b>	<b>CMPTRM1</b>	<b>CMPTRM0</b>	<b>VTSEL1</b>	<b>VTSEL0</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7 未实现位

bit6-4 **BSTRM<2:0>**: BIAS输出电流调节

000 = 3.5uA

001 = 3.75uA

010 = 4uA (缺省值)

011 = 4.25uA

100 = 4.5uA

101 = 5uA

110 = 5.5uA

111 = 6uA

bit3-2 **CMPTRM <1:0>**: 触摸内部比较器消耗电流

00 = 6uA

01 = 12uA (缺省值)

10 = 18uA

11 = 24uA

bit1-0 **VTSEL <1:0>**: 触摸内部比较高电平选择

00 = 0.65V

01 = 1.3V

10 = 1.9V

11 = 2.55V (缺省值)

## 寄存器19FH: 触摸灵敏度调节寄存器1 &lt; CSANA1 &gt;

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-1
—	—	—	<b>DTTRM1</b>	<b>DTTRM0</b>	<b>CTTRM2</b>	<b>CTTRM1</b>	<b>CTTRM0</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-5 未实现

Bit4-3 **DTTRM <1:0>**: 触摸放电时间调节

00 = 0.75us

01 = 1us (缺省值)

10 = 1.25us

11 = 1.5us

Bit2-0 **CTTRM <2:0>**: 触摸充电电流调节, 用于调节片外触摸电容量程, 提高触摸精度

000 = 0.25uA (量程: 0 ~ 256pF)

001 = 0.5uA (量程: 0 ~ 128pF)

010 = 0.75uA (量程: 0 ~ 96pF)

011 = 1uA (量程: 0 ~ 64pF) (缺省值)

100 = 1.25uA (量程: 0 ~ 56pF)

101 = 1.5uA (量程: 0 ~ 48pF)

110 = 1.75uA (量程: 0 ~ 40pF)

111 = 2uA (量程: 0 ~ 32pF)

表 8-2: 与触摸相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 LVR 时的值	所有其他复位值
INTS	GIE	CSIF	TIS	TMR1IE	PAIE	TIF	TMR1IF	PAIF	0000 0000	0000 0000
CSCTL	CSEN	RAMEN	CSIE	CSBUSY	CSINTS	CSACU2	CSACU1	CSACU0	0000 0011	0000 0011
THCTL	THFIB	THGAP1	THGAP0	THMOD2	THMOD1	THMOD0	THDS1	THDS0	0010 1001	0010 1001
CHRLTL	—	—	—	—	CHRLT3	CHRLT2	CHRLT1	CHRLT0	---- 0000	---- 0000
CHSELL	CCH2SEL	—	—	—	CHSEL3	CHSEL2	CHSEL1	CHSEL0	0--- 0000	0--- 0000
CSTMRL	触摸扫描周期计数器								0000 0000	xxxx xxxx
CSANA0	—	BSTRM2	BSTRM1	BSTRM0	CMPTRM1	CMPTRM1	VTSEL1	VTSEL0	-010 0111	-010 0111
CSANA1	—	—	—	DTTRM1	DTTRM0	CTTRM2	CTTRM1	CTTRM0	---0 1011	---0 1011

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

## 9.0 配置寄存器

配置寄存器是用来配置 MCU 的设置，其原始值存在 Flash 的某些位置，在 MCU 启动时从 Flash Boot 到配置寄存器中，配置寄存器位宽为 8 位，由于 Flash 的位宽为 14 位，Boot 时，只 Boot 低 8 位至配置寄存器。

### 9.1 Config Option

Config Option 用于给客户id提供配置选项。

寄存器 801H: 配置字寄存器高8位 (Config1 Option)

U-0	R/W-1	R/W-1	R/W-1	U-0	U-0	R/W-1	R/W-1
—	RD_CTRL	BORS1	BORS0	—	—	BOREN1	BOREN0
bit15						bit8	

寄存器 800H: 配置字寄存器低8位 (Config0 Option)

U-0	R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0
—	CPB	MCLRE	PWRTE	WDTE	—	—	—
bit7						bit0	

bit15 未实现: 保留

bit14 **RD\_CTRL**: 输出模式时读端口控制

1: 读数据端口返回的PIN上的值

0: 读数据端口返回的锁存器上的值

bit13-12 **BORS <1:0>**: 低电压复位电压设定

11: 3.6V

10: 3.0V

01: 2.5V

00: 2.0V

bit11-10 未实现: 保留

bit9-8 **BOREN <1:0>**: 低电压复位选择

11: 低电压复位使能

10: MCU工作时，低电压复位开启；MCU睡眠时，低电压复位关闭

01: 低电压复位由PCON之SBOREN位决定

00: 禁止低电压复位

bit7 未实现: 保留

bit6 **CPB**: 代码保护位

1: Flash 内容不保护

0: 启动Flash内容保护，MCU能读，串口不能读

bit5 **MCLRE**: PA3/MCLR引脚功能选择位<sup>(4)</sup>

1: PA3/MCLR脚执行MCLR功能，是复位脚

0: PA3/MCLR脚执行PA3功能，是数字输入引脚

bit4 **PWRTE**: 上电延时定时器使能位

1: PWRT禁止

0: PWRT使能

bit3 **WDTE**: 看门狗定时器使能位

1: WDT使能，程序不能禁止

0: WDT禁止，但程序可通过设置WDTCN的SWDTEN位将WDT使能

bit2-0 未实现: 保留



## 10.0 指令表

指令	功能	操作	影响状态位
NOP	空操作	无	无
CLRWT	清看门狗	$0 \rightarrow WT$	/TF, /PF
SLEEP	进入睡眠模式	$0 \rightarrow WT$ , stop OSC	/TF, /PF
TMODE	将 W 寄存器值装入选项寄存器 (81H)	$W \rightarrow 81H$	无
CPIO R	设置端口方向寄存器 (1 输入, 0 输出)	$W \rightarrow CPIO_r$	无
STWR R	将 W 寄存器内容送 R 寄存器	$W \rightarrow R$	无
LDR R, t	读寄存器 R, 结果保存在 R(t=1) 或者 W(t=0)	$R \rightarrow t$	Z
LDWI I	立即数送 W 寄存器	$I \rightarrow W$	无
SWAPR R, t	交换寄存器 R 的高低四位, 结果保存在 R(t=1) 或者 W(t=0) 中	$[R(0 \sim 3)R(4 \sim 7)] \rightarrow t$	无
INCR R, t	递增寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中	$R + 1 \rightarrow t$	Z
INCRSZ R, t	递增寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中; 如果结果等于 0 则跳过该指令接下来的指令	$R + 1 \rightarrow t$	无
ADDWR R, t	W 寄存器与 R 寄存器相加, 结果保存在 R(t=1) 或者 W(t=0) 中	$W + R \rightarrow t$	C, HC, Z
SUBWR R, t	R 寄存器减去 W 寄存器, 结果保存在 R(t=1) 或者 W(t=0) 中	$R - W \rightarrow t$ ( $R+/W+1 \rightarrow t$ )	C, HC, Z
DECR R, t	递减寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中	$R - 1 \rightarrow t$	Z
DECRSZ R, t	递减寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中; 如果结果等于 0 则跳过该指令接下来的指令	$R - 1 \rightarrow t$	无
ANDWR R, t	R 寄存器与 W 寄存器做“与”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R \cap W \rightarrow t$	Z
ANDWI I	W 寄存器与立即数 I 做“与”操作, 结果保存到 W 寄存器中	$I \cap W \rightarrow W$	Z
IORWR R, t	R 寄存器与 W 寄存器做“或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R \cup W \rightarrow t$	Z
IORWI I	W 寄存器与立即数 I 做“或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$I \cup W \rightarrow W$	Z
XORWR R, t	R 寄存器与 W 寄存器做“异或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R \oplus W \rightarrow t$	Z
XORWI I	W 寄存器与立即数 I 做“异或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$I \oplus W \rightarrow W$	Z
COMR R, t	R 寄存器“取反”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$/R \rightarrow t$	Z
RRR R, t	R 寄存器循环“右移”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R(n) \rightarrow R(n-1)$ , $C \rightarrow R(7)$ , $R(0) \rightarrow C$	C
RLR R, t	R 寄存器循环“左移”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R(n) \rightarrow r(n+1)$ , $C \rightarrow R(0)$ , $R(7) \rightarrow C$	C
CLRW	W 寄存器清 0	$0 \rightarrow W$	Z
CLRR R	R 寄存器清 0	$0 \rightarrow R$	Z
BCR R, b	R 寄存器的第 b 位清 0	$0 \rightarrow R(b)$	无
BSR R, b	R 寄存器的第 b 位置 1	$1 \rightarrow R(b)$	无
BTSC R, b	如果 R 寄存器的第 b 位为 0, 则跳过该指令接下来的指令	Skip if $R(b)=0$	无
BTSS R, b	如果 R 寄存器的第 b 位为 1, 则跳过该指令接下来的指令	Skip if $R(b)=1$	无
LCALL N	在整个 2K 区域内的调用指令	$N \rightarrow PC$ , $PC+1 \rightarrow Stack$	无
LJUMP N	在整个 2K 区域内的跳转指令	$N \rightarrow PC$	无
RTIW I	带立即数从子程序返回	$Stack \rightarrow PC$ , $I \rightarrow W$	无
ADDWI I	W 寄存器与立即数 I 相加, 结果保存到 W 中	$PC+1 \rightarrow PC$ , $W+I \rightarrow W$	C, HC, Z
SUBWI I	立即数 I 减去 W 寄存器, 结果保存到 W 寄存器中	$I - W \rightarrow W$	C, HC, Z
RTFI	中断返回	$Stack \rightarrow PC$ , $1 \rightarrow GIS$	无
RET	从子程序返回	$Stack \rightarrow PC$	无

## 11.0 电气特性

### 11.1 绝对极限参数

偏置电压下的环境温度.....	-40°C 至+85°C
储存温度.....	-65°C 至+150°C
VDD 引脚相对于 VSS 的电压.....	-0.3V 至+6.5V
MCLR 引脚相对于 Vss 的电压.....	-0.3V 至+9.5V
所有其他引脚相对于 VSS 的电压.....	-0.3V 至(VDD+0.3V)
总功耗 <sup>(1)</sup> .....	600mW
流出 VSS 引脚的最大电流.....	95mA
流入 VDD 引脚的最大电流.....	95mA
输入钳位电流, I <sub>IK</sub> (V <sub>I</sub> <0 或 V <sub>I</sub> >VDD).....	±20mA
输出钳位电流, I <sub>OK</sub> (V <sub>O</sub> <0 或 V <sub>O</sub> >VDD).....	±20mA
任一 I/O 引脚的最大输出灌电流.....	25mA
任一 I/O 引脚的最大输出拉电流.....	25mA
PORTA 最大灌电流.....	90mA
PORTA 最大拉电流.....	90mA

注 1: 功耗计算公式为:  $P_{Dis} = VDD \times \{ I_{DD} - \sum I_{OH} \} + \sum \{ (V_{DD} - V_{OH}) \times I_{OH} \} + \sum (V_{OL} \times I_{OL})$

注意: 如果运行条件超过了上述“绝对极限参数值”, 即可能对器件造成永久性损坏。上述值仅为运行条件的极大值, 我们不建议器件运行在该规范范围以外。器件长时间工作在绝对极限参数条件下, 其稳定性可能受到影响。

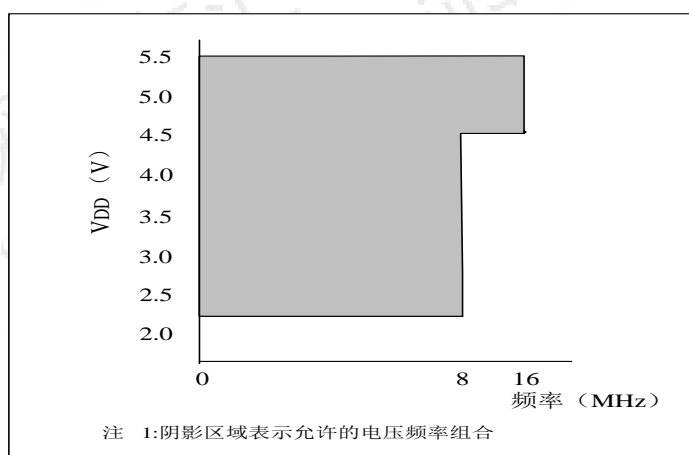


图 11-1: YS65F804 外部振荡电压-频率图,  $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$

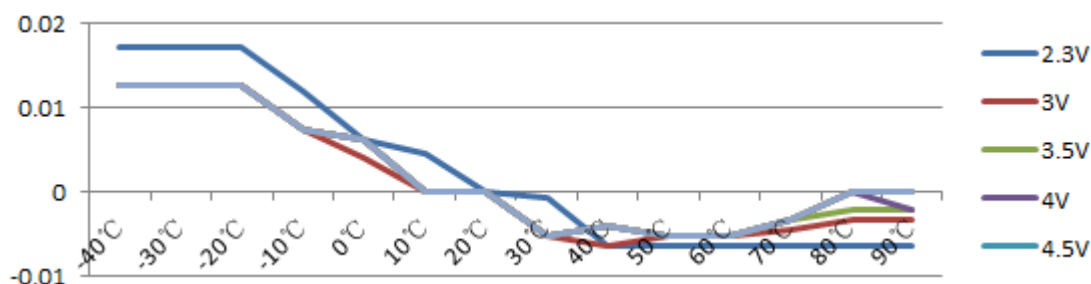


图 11-2: 器件 VDD 和温度范围内 HFINTOSC 的频率准确性

## 11.2 直流电器特性

直流特性		标准工作条件						
		工作温度 $-40^{\circ}\text{C} \leq \text{Ta} \leq 85^{\circ}\text{C}$						
符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件		
VDD	电源电压	2.5		5.5	V			
VDR	RAM 数据保持电压 <sup>(2)</sup>	—	0.5*	—	V	器件处于休眠模式		
VPOR	Vdd 起始电压确保能够产生上电复位信号	—	Vss	—	V			
SVDD	Vdd 上升速率确保能够产生上电复位信号	0.05*	—	—	V/ms			
IDD	工作电流 <sup>(3)</sup>	—	450	—	uA	3.3V	1M	WDT En PED Dis IRC
			640				4M	
			870			5V	8M	
			540				1M	
			840				4M	
			1250				8M	
IPD	掉电流 <sup>(4)</sup>	—	1	—	uA	WDT Disable VDD=2.5V		
$\Delta$ IWDT	WDT 电流 <sup>(4)</sup>	—	2	—	uA	VDD=5V		
VIL	输入低电压	VSS	—	1.24	V	3V TLL		
		VSS	—	1.28		3V SCHMITT		
		VSS	—	1.76		5V TLL		
		VSS	—	1.52		5V SCHMITT		
VIH	输入高电压	1.28	—	VDD	V	3V TLL		
		1.68	—	VDD		3V SCHMITT		
		1.76	—	VDD		5V TLL		
		2.88	—	VDD		5V SCHMITT		
IOL	输出灌电流	—	20	—	mA	VOL=0.3V	3V	
		—	45	—		VOL=0.5V	5V	
IOH	输出拉电流	—	8	—	mA	VOH=2.7V	3V	
		—	15	—		VOH=4.5V	5V	
LVR	低电压复位电压	2.0 -20%	2.0	2.0 +20%	V	LOW		
		2.5 -20%	2.5	2.5 +20%		LOW		
		3.0 -20%	3.0	3.0 +20%		MIDDLE		
		3.6 -20%	3.6	3.6 +20%		HIGH		
Rpu	上拉电阻	—	32	—	K	3V		
		—	20	—		5V		
Rd	下拉电阻	—	6	—	K	3V		
		—	5	—		5V		

注：“—”表示没有，或待定。

(1) 典型栏中数据均为 25°C 条件下值，此部分数据仅供参考。

(2) 该电压是保证不丢失 RAM 数据的最小 VDD。

(3) 工作电流主要随工作电压和频率而变化。其它因素，如总线负载、总线速率、内部代码执行模式和温度也会影响电流消耗。

(4) 掉电电流是在器件休眠时，所有 I/O 引脚都处于高阻态并且连接到 Vdd 或 Vss 时测得。

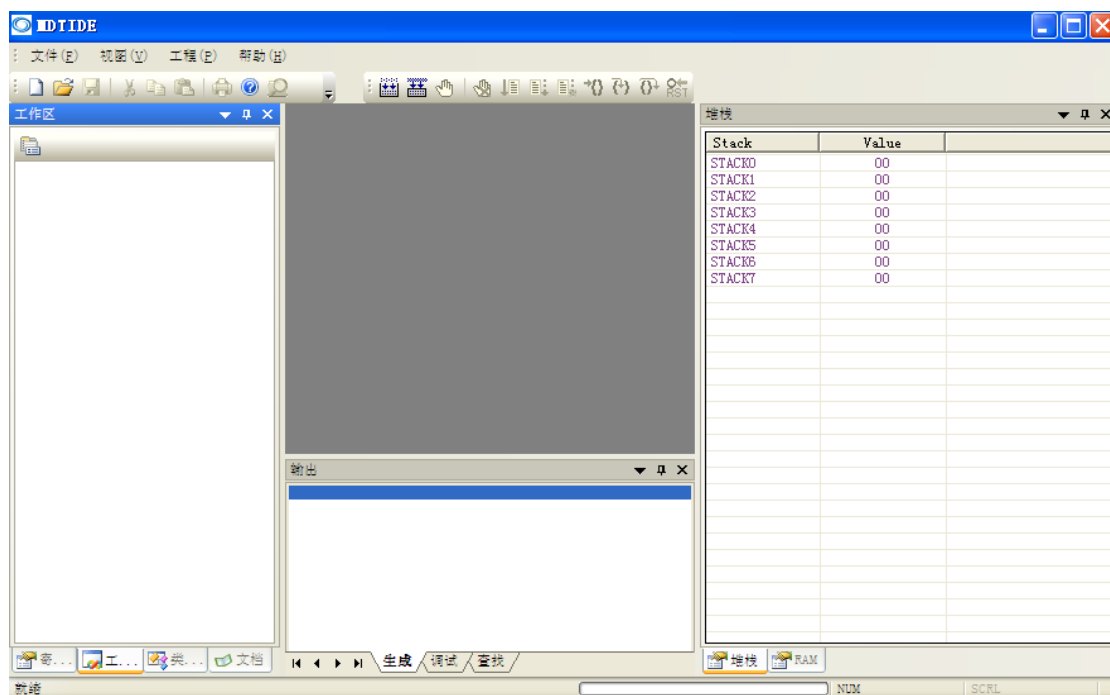
### 11.3 交流电气特性

交流特性		标准工作条件				
		工作温度 $-40^{\circ}\text{C} \leq \text{Ta} \leq +85^{\circ}\text{C}$				
符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
F <sub>OSC</sub>	系统时钟	31K	8M	16M	Hz	5V
F <sub>HIRC</sub>	内部高速时钟	250K	4M	16M	Hz	5V
F <sub>LIRC</sub>	内部低速时钟	—	31K	—	Hz	5V
T <sub>INT</sub>	中断脉冲		2	3	Tins	指令周期
T <sub>SST</sub>	系统启动时间 (上电复位)		13	20	ms	5V
	系统启动时间 (由 SLEEP 模式唤醒, F <sub>sys</sub> 在 SLEEP 模式下关闭)		16		T <sub>sys</sub>	5V
						5V
	系统启动时间(由 SLEEP 模式唤醒,主要在 SLEEP 模式下开启)		16		T <sub>sys</sub>	5V

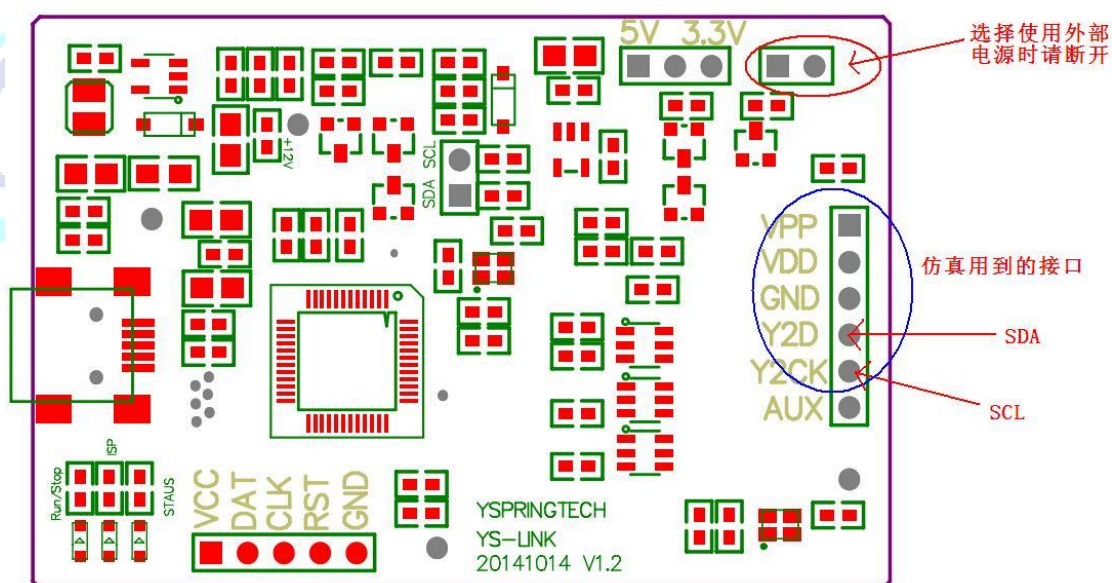
## 12.0 开发支持

### 12.1 仿真信息

#### 12.1.1 软件：MDT-IDE



#### 12.1.2 硬件：YS-Link



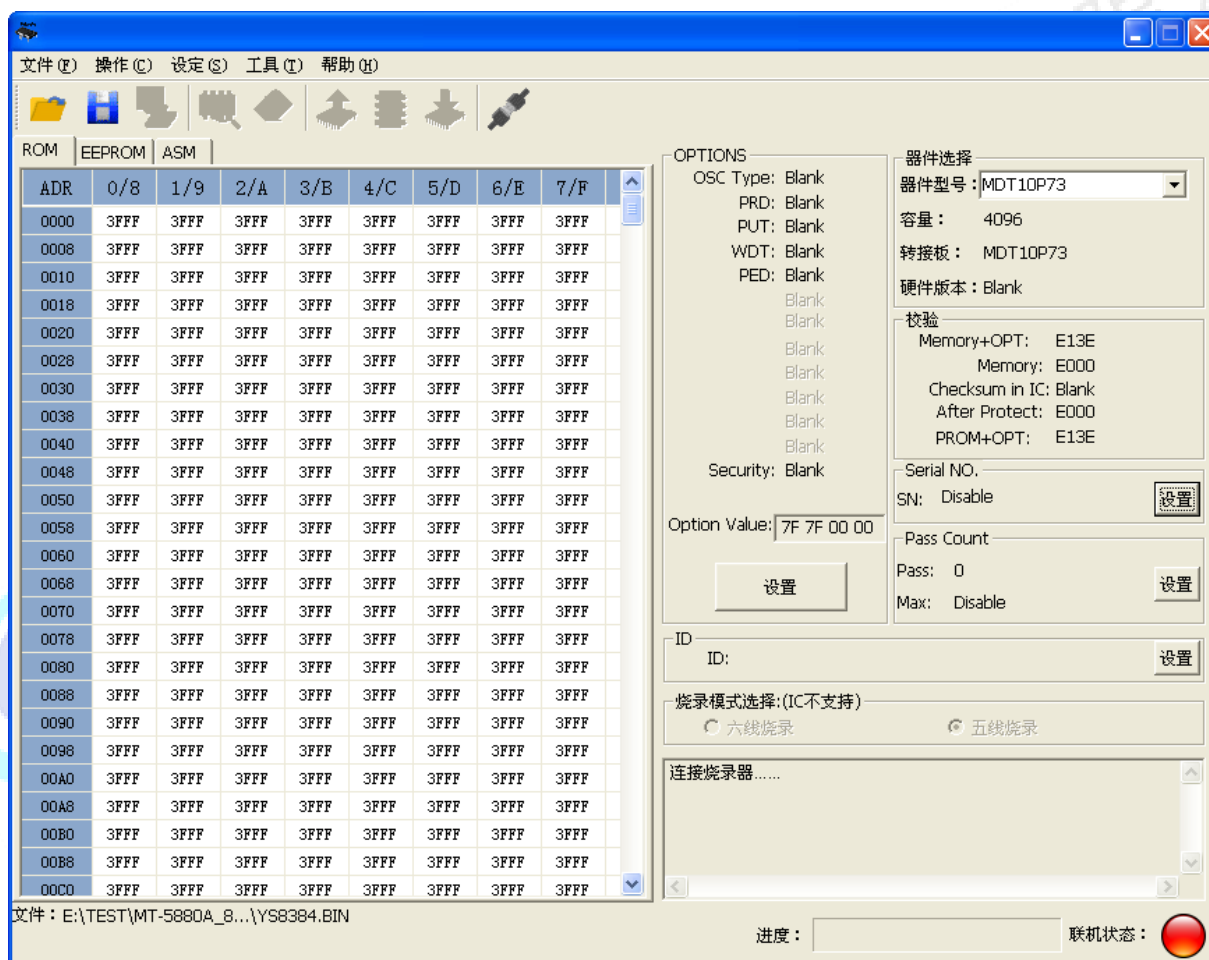
### 12.1.3 接口:

YS65F804 具备片内仿真功能，可通过串口模式仿真调试程序，串口接口如下：

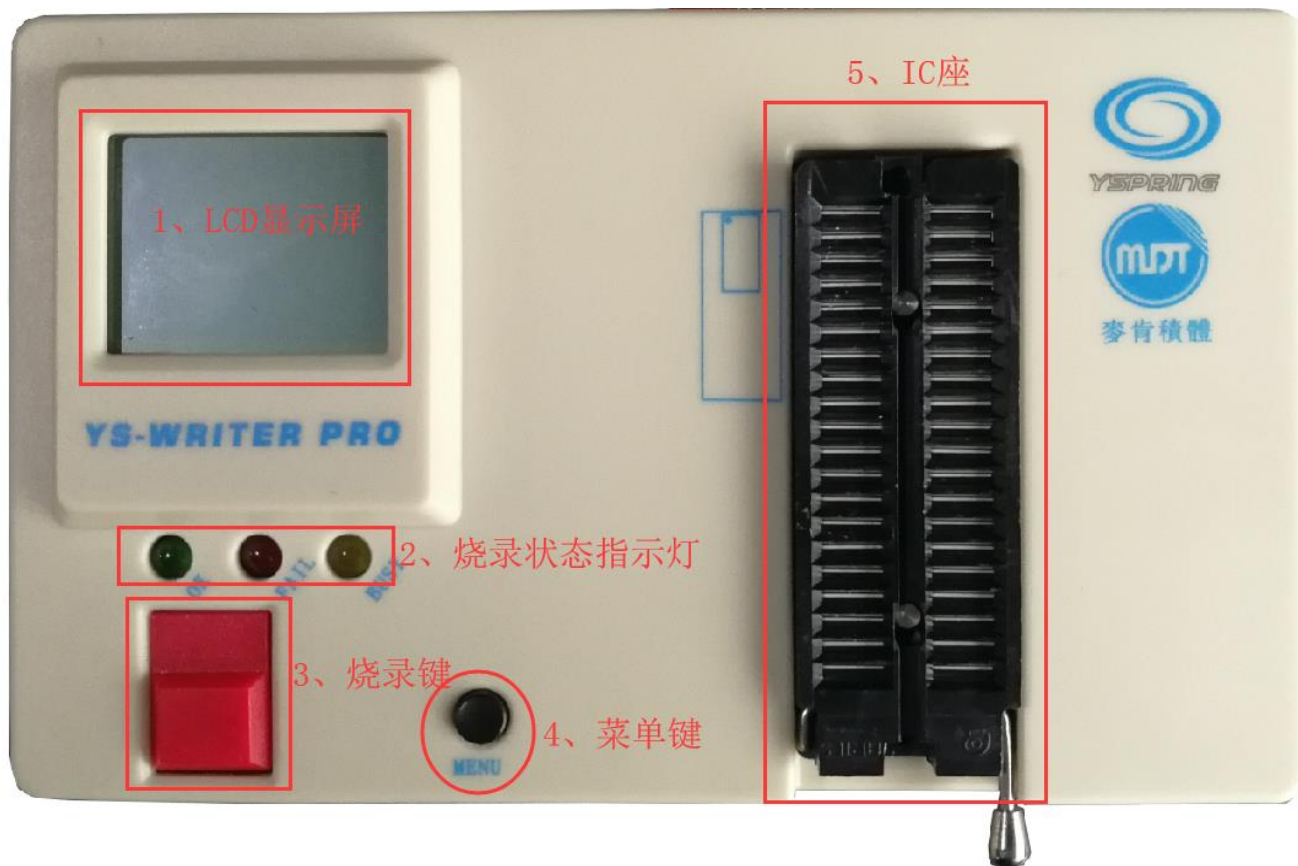
YS65F804	YS-Link
VDD	VDD
GND	GND
PA3	VPP
PA0	SCK
PA1	SDA

## 12.2 烧录信息

### 12.2.1 烧录软件：YSpringPro



### 12.2.2 烧录器：YS-Writer



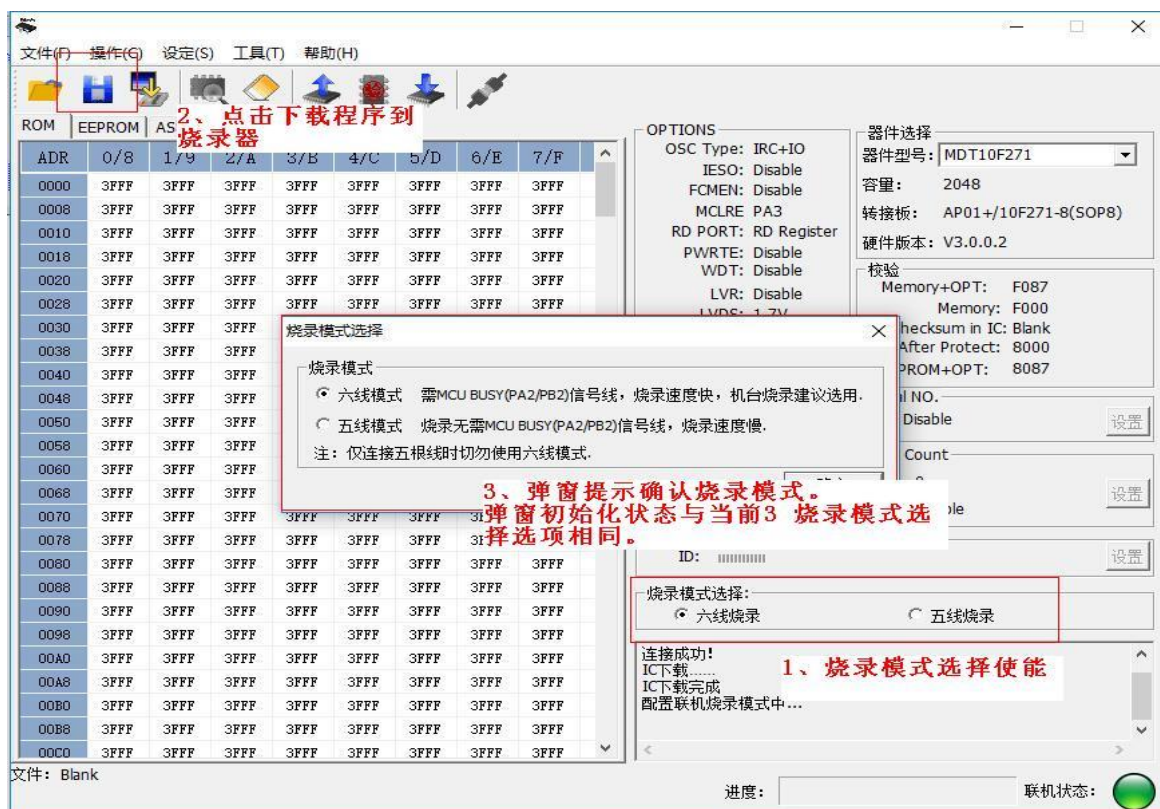
深圳市汇  
Shenzhen Yspring

MTP 类型 MCU 烧录时需要选择五线烧录模式或六线烧录模式。

具体操作步骤：

- 1) 如果 1 烧录模式选择使能。
- 2) 点击 2 下载程序到烧录器时，弹窗 3 提示确认烧录模式。
- 3) 弹窗 3 初始化状态与当前 1 烧录模式选择选项相同。

注：仅连接五根线时切勿使用六线模式

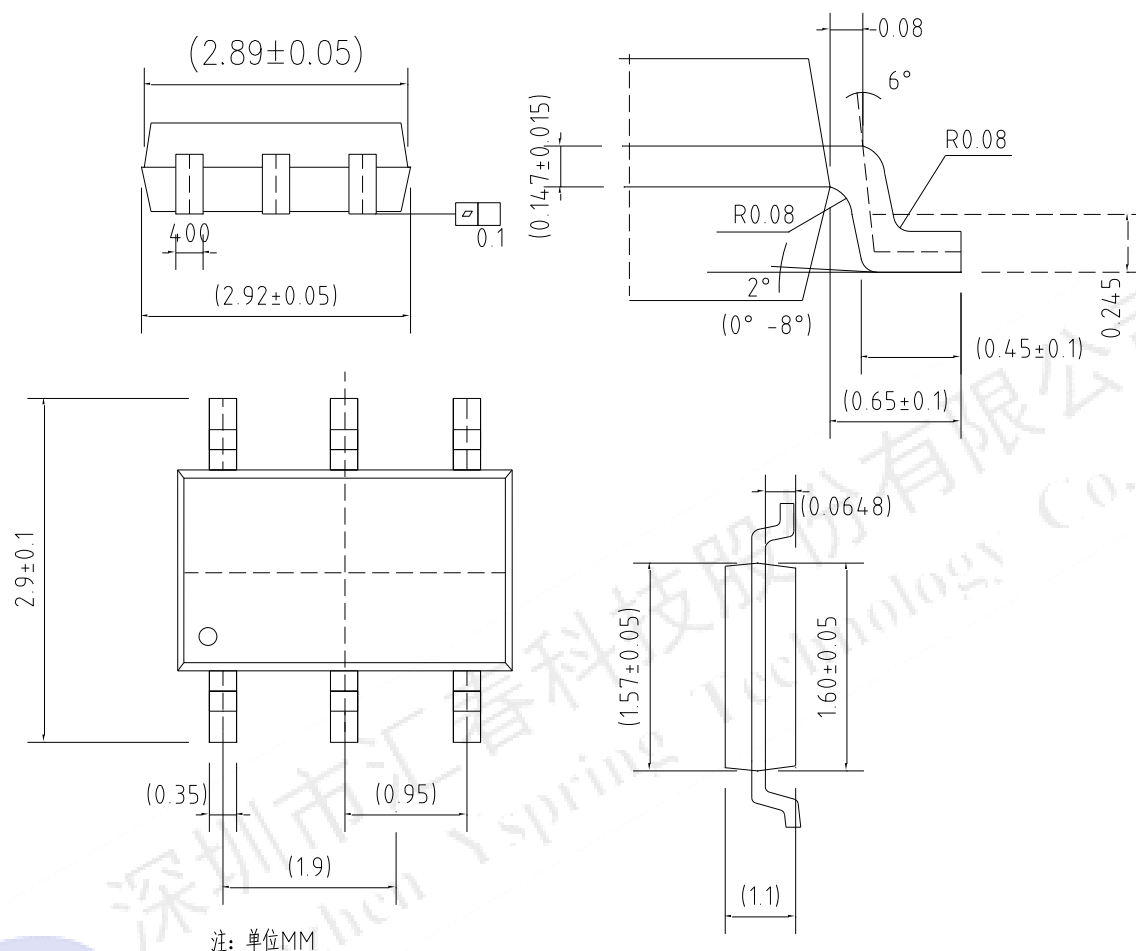


烧录模式	烧录器使用引脚
六线烧录模式：需要 MCU 烧录 Busy 信号，烧录速度快，通常机台烧录时选用。	VDD、VSS、VPP、PA0、PA1、PA2 (Busy 信号)
五线烧录模式：无需 Busy 信号，通过延时完成烧录，烧录速度慢，通常板上烧录没有引出 MCU Busy 信号引脚时使用。	VDD、VSS、VPP、PA0、PA1



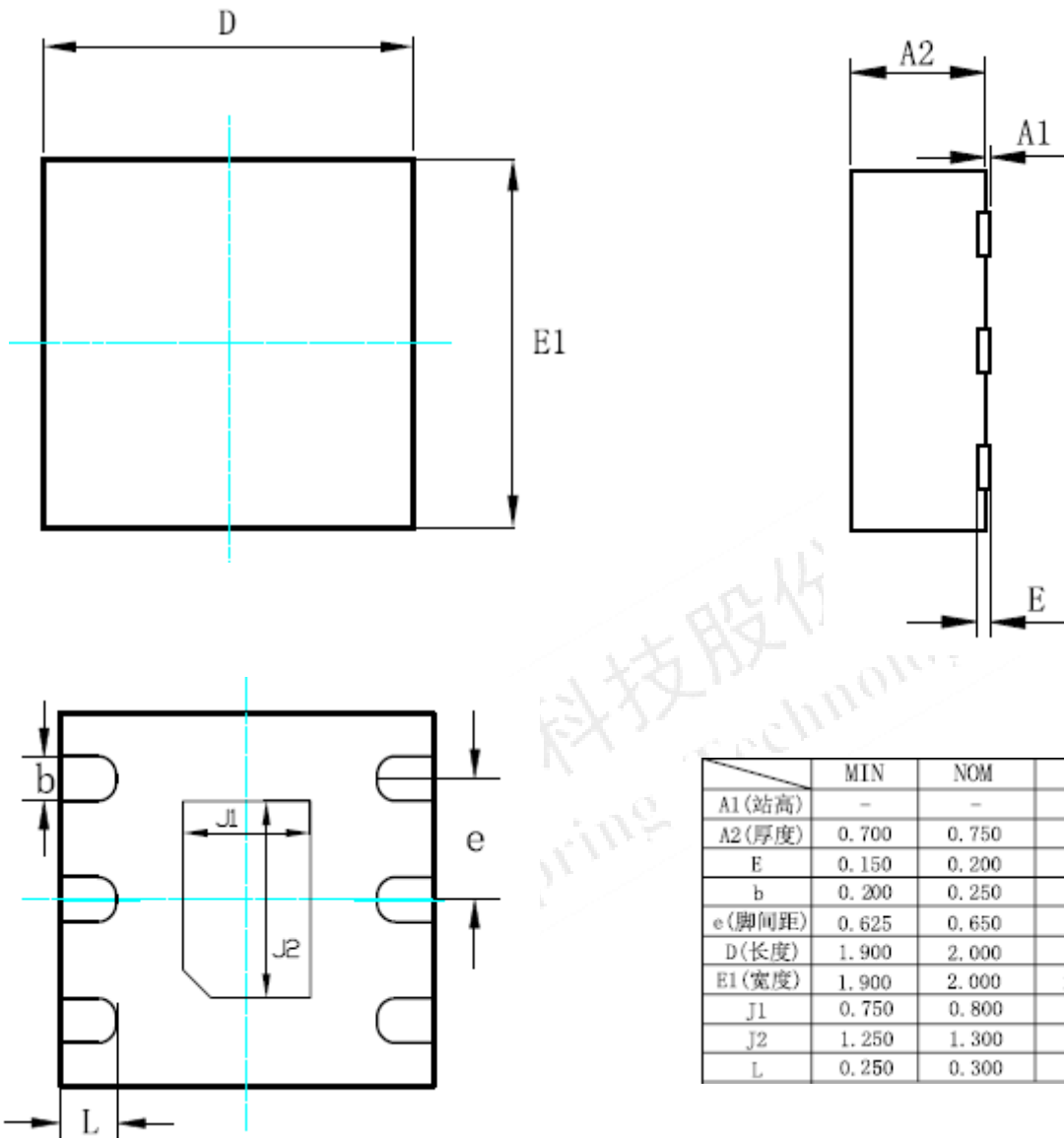
## 13.0 封装信息

### 13.1 SOT23-6



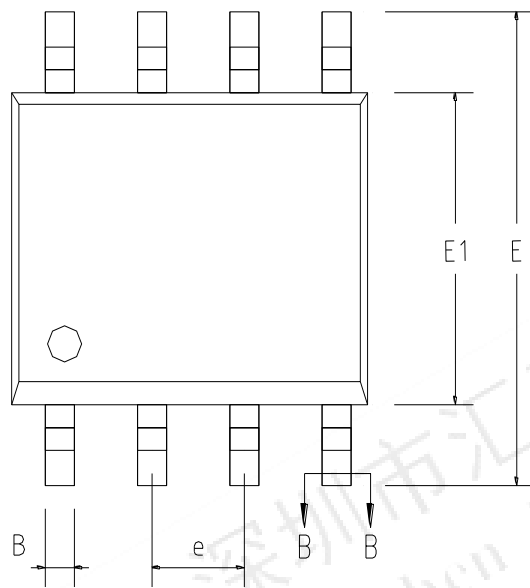
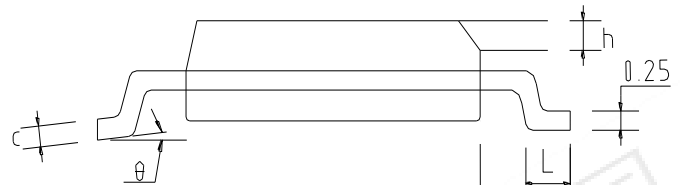
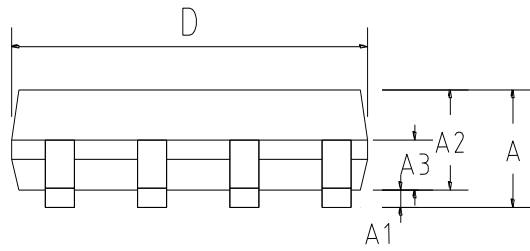
注: 单位 MM

### 13.2 DFN6



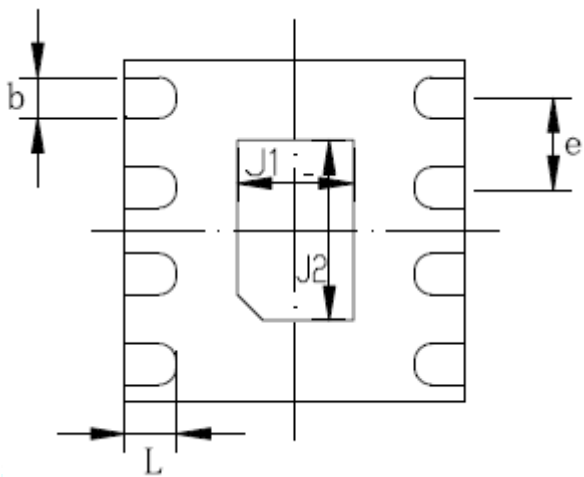
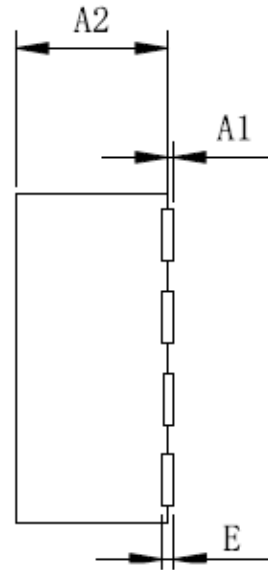
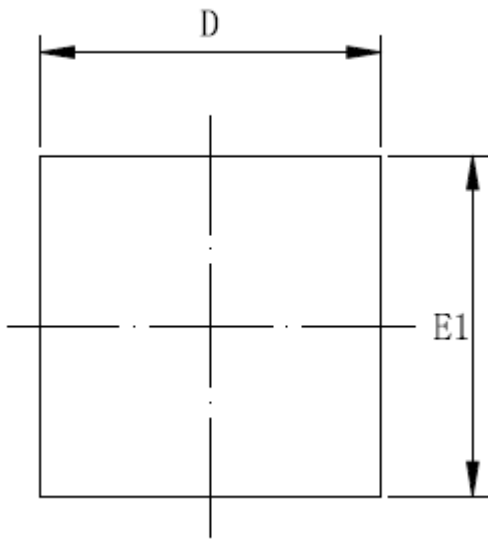
	MIN	NOM	MAX
A1 (站高)	-	-	0.050
A2 (厚度)	0.700	0.750	0.800
E	0.150	0.200	0.250
b	0.200	0.250	0.300
e (脚间距)	0.625	0.650	0.675
D (长度)	1.900	2.000	2.100
E1 (宽度)	1.900	2.000	2.100
J1	0.750	0.800	0.850
J2	1.250	1.300	1.350
L	0.250	0.300	0.350

## 13.3 SOP 8 PIN



SYMBOLS	MIN	NOM	MAX	SYMBOLS	MIN	NOM	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.70	3.90	4.10
A1	0.10	—	0.225	e	1.27BSC		
A2	1.30	1.40	1.50	h	0.25	—	0.50
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	4.70	4.90	5.10	L1	1.05 BSC		
E	5.80	6.00	6.20	$\theta$	0	—	8°
L/F 载体 尺寸(mil)	80*80	90*90	95*130	—			

## 13.4 DFN8



	MIN	NOM	MAX
A1 (站高)	-	-	0.050
A2 (厚度)	0.700	0.750	0.800
E	0.150	0.200	0.250
b	0.200	0.250	0.300
e (脚间距)	-	0.5TYP	-
D (长度)	1.900	2.000	2.100
E1 (宽度)	1.900	2.000	2.100
J1	0.840	0.890	0.940
J2	1.450	1.500	1.550
L	0.250	0.300	0.350

## 14.0 汇春知识产权政策

### 14.1 专利权

汇春公司在全球各地区已核准和申请中之专利权至少有 20 件以上,享有绝对之合法权益。与汇春公司 MCU 或其它产品有关的专利权并未被同意授权使用,任何经由不当手段侵害汇春公司专利权之公司、组织或个人,汇春将采取一切可能的法律行动,遏止侵权者不当的侵权行为,并追讨汇春公司因侵权行为所受之损失、或侵权者所得之不法利益。

### 14.2 著作权

Copyright 2017 by INC.

规格书中所出现的信息在出版当时相信是正确的,然而汇春对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明,汇春不保证或不表示这些应用没有更深入的修改就能适用,也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。汇春产品不授权使用于救生、维生器件或系统中做为关键器件。汇春拥有不事先通知而修改产品的权利,对于最新的信息,请参考我们的网址

<http://www.yspringtech.com>;

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology

