十速

TM54M0321

Rev 0.93

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **Tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **Tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

修改记录

版本	日期	描述
0.90	Mar, 2026	New Release
0.91	Apr, 2026	电气特性修订
0.92	Apr, 2026	1. MOVFW 影响标志 Z 2. 增加进出中断范例程序 3. 修正最大工作电压为5.5V
0.93	Aug, 2026	1. ADC Vss input 更正為ADCHS=1101b 2. 文字叙述暂停模式更改为空闲模式 3. VBG DC特性調整 4. 新增LVD功能

CONTENTS

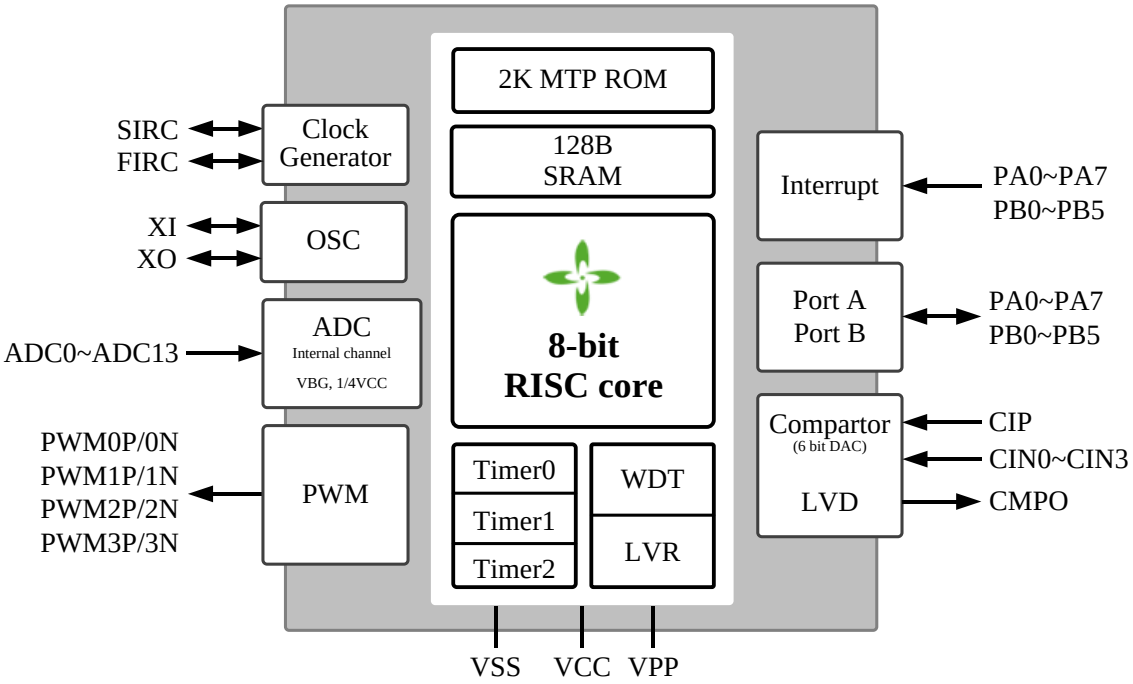
修改记录 2
基本功能 5
系统结构图 6
引脚分配图 7
引脚描述 8
引脚摘要 9
功能描述 10
1. CPU 核心 10
1.1 程序 ROM (PROM) 10
1.2 系统配置寄存器 (SYSCFG) 11
1.3 时钟配置和指令周期 12
1.4 RAM 寻址模式 13
1.5 程序计数器（PC）和堆叠 16
1.6 ALU 和工作寄存器（W） 18
1.7 状态寄存器 (F-Plane 03H) 18
2. 复位 22
2.1 上电复位 (POR) 22
2.2 低电压复位 (LVR) 22
2.3 看门狗定时器复位 (WDT) 22
3. 时钟电路和工作模式 23
3.1 系统时钟 23
3.2 系统工作模式 23
3.3 主时钟切换 24
3.4 主时钟振荡器 26
4. 中断 27
5. Timer0 与 WDT 30
6. Timer1 (PWM0/PWM1) 33
7. Timer2 (PWM2/PWM3) 38
8. ADC 41
9. I/O 端口 44
9.1 PA7-PA0, PB5-PB0 44

10. 比较器与 DAC 与 LVD 48
指令集 51
内存功能图 52
F-Plane 52
指令集 61
电气特性 74
1. 绝对最大值 74
2. 直流特性 74
3. 时钟时序 75
4. 复位时间特性 76
5. LVR 电路特性 76
6. ADC 电气特性 76
7. 比较器特性 76
8. 特性曲线图 77
封装说明 80

基本功能

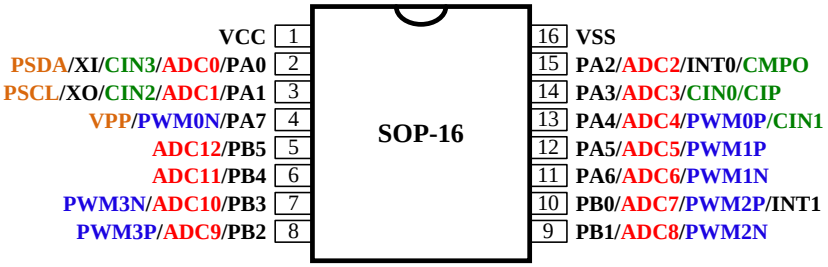
- ROM：2K x 14 位 MTP
- RAM：128 x 8 位
- 堆叠：8 级
- I/O 端口：14 个可编程的 I/O 端口
- 3 个定时器:
  - Timer0: 8 位定时器与 WDT 共享预分频器
  - Timer1: 10 位定时器与 PWM0/1 共享计数器
  - Timer2: 8 位定时器与 PWM2/3 共享计数器
- 系统时钟源 (Fsys)
  - 快时钟：内部高速振荡器 FIRC 16 MHz, 其/1/2/4/8 分频时钟可作为系统主时钟源
  - 慢时钟：可作为系统主时钟源
    - 内部低频振荡器 SIRC 50KHz
    - 可外接晶体振荡器 SXT 32KHz
- 系统工作模式
  - 运行模式: 系统在主时钟下运行
  - 空闲模式: 系统时钟停止, 慢时钟仍运行
  - 停止模式: 系统时钟与快/慢时钟都停止
- 低电压检测 (LVD)
  - 使用比较器与 DAC 完成低電壓檢測
- 复位
  - 上电复位 1.5V
  - 低电压复位 关闭/1.8V/2.3V/2.8V
- 中断
  - INT0/INT1 为上边沿或下边沿触发
  - 全脚位引脚电平变化中断/唤醒
  - 定时器中断 (Timer0 / Timer1 / Timer2)
  - ADC 中断
  - 比较器中断
- I/O 口模式
  - PA7 为刻录时高压 VPP 输入端口
  - 所有端口均支持推挽/开漏输出
  - 所有端口均支持引脚电平变化中断/唤醒功能，可单独使能
  - 所有端口均内置上拉和下拉电阻，可单独使能
- 3 阶 LVR (2.8V/2.3V/1.8V)
- ADC
  - 13 外部脚位通道模拟输入, 2 内部电位输入(VBG, 1/4V_{CC})
- 比较器与 DAC
  - 比较器负端可选择由 4 个外部脚位输入或 VBG 输入或由 DAC 产生电位输入
  - 比较器正端可由 PA3 脚位输入
- 4 组 PWM 输出
  - PWM0/PWM1 共享 timer1 计数器, 为 10 位 PWM
  - PWM2/PWM3 共享 timer2 计数器, 为 8 位 PWM
  - PWM0 含死区设置與互补输出功能
- 内部自振式看门狗计数器（WDT）
  - 与定时器 Timer0 共享预分频器
  - 溢出时间可配置：2.5ms/10ms/40ms/160ms（无预分频时）
  - 工作模式可配置：开启 WDT、关闭 WDT，也可软件控制开启或关闭
  - WDT 在空闲模式下溢出, 此时为溢出唤醒, 不会产生复位
- 读表指令：14 位 ROM 数据查表
- 操作电压：压复位电压至 5.5V
- 操作温度范围：-40°C 至 +85°C
- 指令集：39 条指令

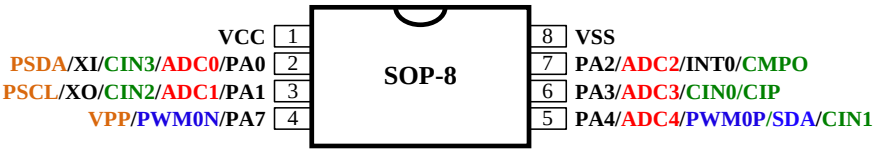
系统结构图

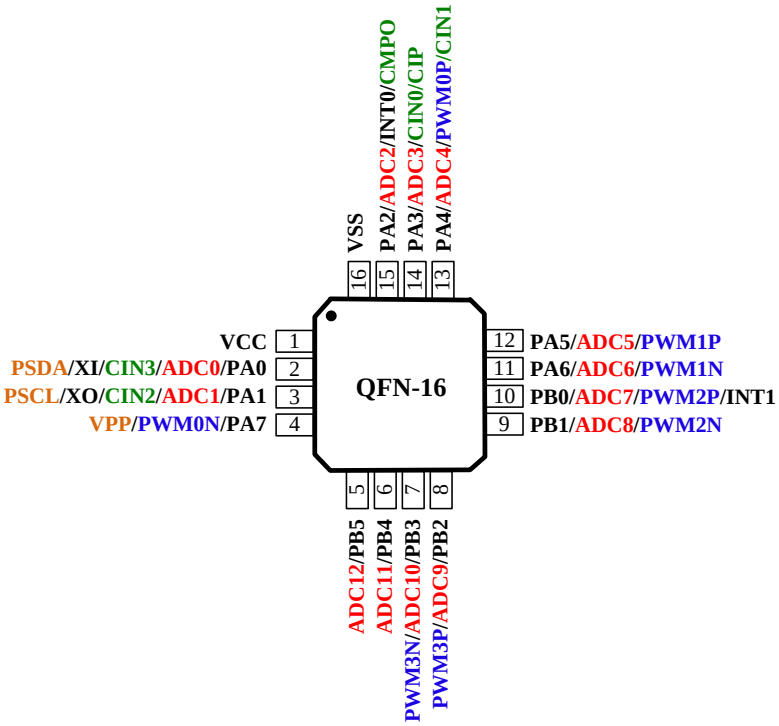


Block Diagram

引脚分配图


SOP-16
VCC 1
PSDA/XI/CIN3/ADC0/PA0 2
PSCL/XO/CIN2/ADC1/PA1 3
VPP/PWM0N/PA7 4
ADC12/PB5 5
ADC11/PB4 6
PWM3N/ADC10/PB3 7
PWM3P/ADC9/PB2 8
16 VSS
15 PA2/ADC2/INT0/CMPO
14 PA3/ADC3/CIN0/CIP
13 PA4/ADC4/PWM0P/CIN1
12 PA5/ADC5/PWM1P
11 PA6/ADC6/PWM1N
10 PB0/ADC7/PWM2P/INT1
9 PB1/ADC8/PWM2N


SOP-8
VCC 1
PSDA/XI/CIN3/ADC0/PA0 2
PSCL/XO/CIN2/ADC1/PA1 3
VPP/PWM0N/PA7 4
8 VSS
7 PA2/ADC2/INT0/CMPO
6 PA3/ADC3/CIN0/CIP
5 PA4/ADC4/PWM0P/SDA/CIN1


QFN-16
VCC 1
PSDA/XI/CIN3/ADC0/PA0 2
PSCL/XO/CIN2/ADC1/PA1 3
VPP/PWM0N/PA7 4
16 VSS
15 PA2/ADC2/INT0/CMPO
14 PA3/ADC3/CIN0/CIP
13 PA4/ADC4/PWM0P/CIN1
12 PA5/ADC5/PWM1P
11 PA6/ADC6/PWM1N
10 PB0/ADC7/PWM2P/INT1
9 PB1/ADC8/PWM2N
5 ADC12/PB5
6 ADC11/PB4
7 PWM3N/ADC10/PB3
8 PWM3P/ADC9/PB2

引脚描述

名称	输入/输出	引脚描述
PA0~PA7 PB0~PB5	I/O	可位编程 I/O 端口，用于施密特触发器输入，CMOS 推挽输出，或漏极开路输出。上拉/下拉电阻可由软件分配。
VCC, VSS	P	电源输入引脚和地
VPP	P	刻录高压输入脚位
PSCL/PSDA	I/O	刻录接口
XI, XO	–	用于系统时钟（SXT）的晶体/谐振器振荡器连接
INT0~INT1	I	外部中断输入
PWM0P/0N PWM1P/1N PWM2P/2N PWM3P/3N	O	PWM 输出
CMPO	O	比较器状态输出
ADC0~ADC13	I	ADC 输入通道
CIN0~CIN3	I	比较器负端口输入
CIP	I	比较器正端口输入

编程引脚: VCC / VSS / VPP / PSCL / PSDA

引脚摘要

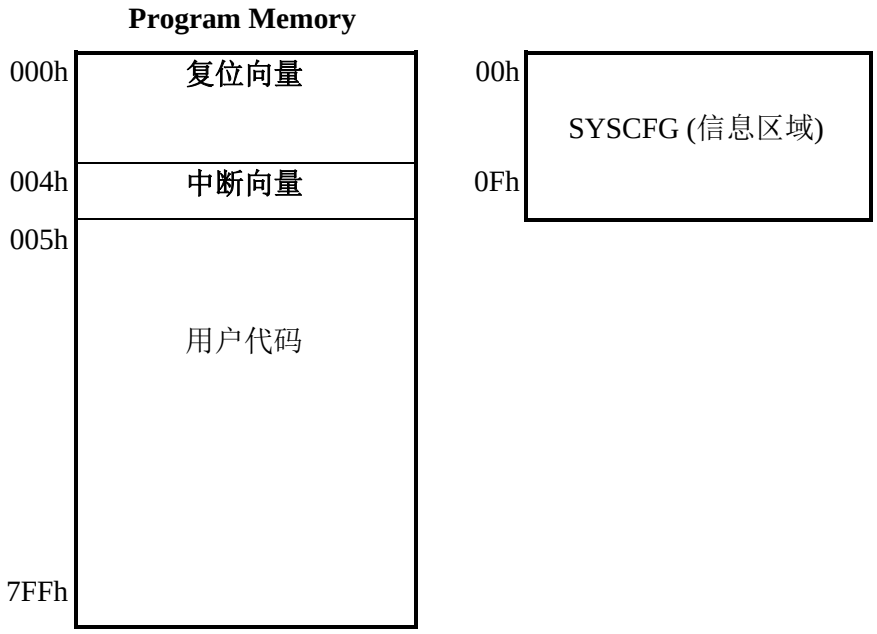
引脚编号	引脚名称		输入				输出		PWM	ADC	比较器	刻录脚	晶振
			上拉控制	下拉控制	中断	唤醒	开漏	推挽					
1	VCC	P											
2	**PA0/ADC0/CIN3/XI/PSDA**	I/O	●	●	●	●	●	●		●	●	●	●
3	**PA1/ADC1/CIN2/XO/PSCL**	I/O	●	●	●	●	●	●		●	●	●	●
4	**PA7/PWM0N/VPP**	I/O	●	●	●	●	●	●	●			●	
5	**PB5/ADC12**	I/O	●	●	●	●	●	●		●			
6	**PB4/ADC11**	I/O	●	●	●	●	●	●		●			
7	**PB3/ADC10/PWM3N**	I/O	●	●	●	●	●	●	●	●			
8	**PB2/ADC9/PWM3P**	I/O	●	●	●	●	●	●	●	●			
9	**PB1/ADC8/PWM2N**	I/O	●	●	●	●	●	●	●	●			
10	**PB0/ADC7/PWM2P/INT1**	I/O	●	●	●	●	●	●	●	●			
11	**PA6/ADC6/PWM1N**	I/O	●	●	●	●	●	●	●	●			
12	**PA5/ADC5/PWM1P**	I/O	●	●	●	●	●	●	●	●			
13	**PA4/ADC4/PWM0P/CIN1**	I/O	●	●	●	●	●	●	●	●	●		
14	**PA3/ADC3/CIN0/CIP**	I/O	●	●	●	●	●	●		●	●		
15	**PA2/ADC2/INT0/CMPO**	I/O	●	●	●	●	●	●		●	●		
16	**VSS**	P											

功能描述

1. CPU 核心

1.1 程序 ROM (PROM)

该器件的闪存程序 ROM 为 2K 字，带有一个额外的信息区域来存储 SYSCFG。ROM 可以多次写入，只要 SYSCFG 的 PROTECT (SYSCFG.13) 位不设置就可以读取。无论 PROTECT 设置还是清零都可以读取 SYSCFG，但只有在擦除用户 ROM 代码区域时才能清除 PROTECT 位。另一方面，如果设置了 PROTECT 位，则写入器不会读取用户 ROM 代码区域，并且在 PROTECT 位清零之前无法更新用户 ROM 代码。



复位向量 (000h)

复位后，系统将在地址 000h 处重新启动程序计数器 (PC)，所有寄存器都将恢复到默认值。

中断向量 (004h)

当中断发生时，程序计数器 (PC) 将被推到堆栈上并跳转到地址 004h。

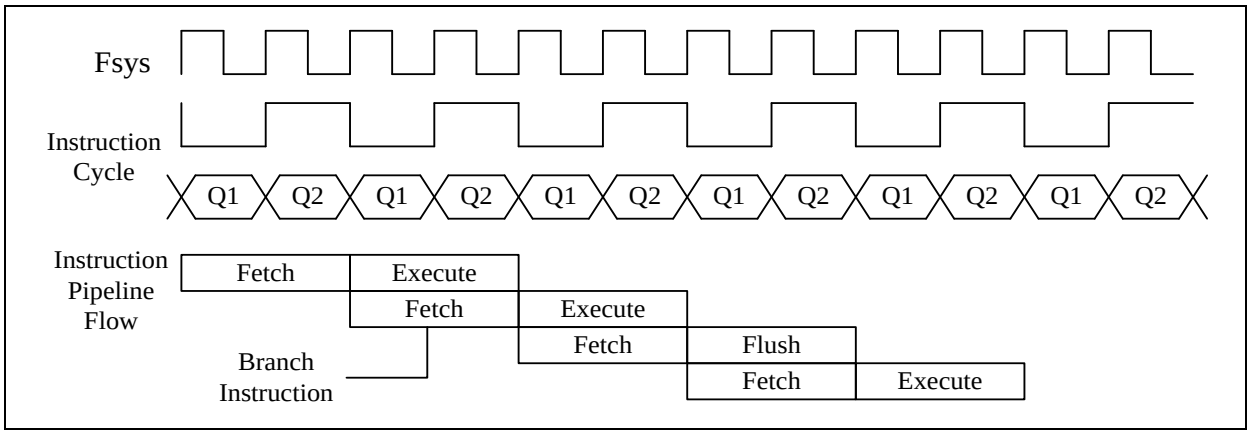
1.2 系统配置寄存器 (SYSCFG)

系统配置寄存器 (SYSCFG) 位于闪存信息区域, 用户可在烧录用户程序代码时进行配置与烧录, 在上电发生后将配置信息载入寄存器中，通过寄存器控制关键模块的工作状态。

位		描述	
SYSCFG	13	**PROTECT**: 代码保护选择	
		0	关闭
		1	使能
	12-6	系统保留设置	
	5	**ATDOFF**: 慢时钟模式省电设置	
		0	慢时钟模式省电开启
		1	慢时钟模式省电关闭
	4	**WDTC**: 模式设置	
		0	WDT 始终关闭
		1	WDT 开启 (SFR WDTE 必需同时为 1 才会启用 WDT)
	3-2	**WDTSEL**: WDT 溢出时间选择（无预分频）	
		00	2.5ms
		01	10ms
		10	40ms
		11	160ms
	1-0	**LVRE**: LVR 复位电压选择	
		00	1.8V
		01	2.3V
		10	2.8V
		11	关闭

1.3 时钟配置和指令周期

系统时钟在内部被分成用于指令周期的 Q1 和 Q2 两个状态。程序计数器（PC）在 Q1 更新且指令从程序寄存器获得并且在状态 Q2 锁于指令寄存器。在接下来的 Q1-Q2 周期，程序被解码并执行。分支指令占用两个周期，因为从传输管道获得指令的同时新指令被获取并执行。



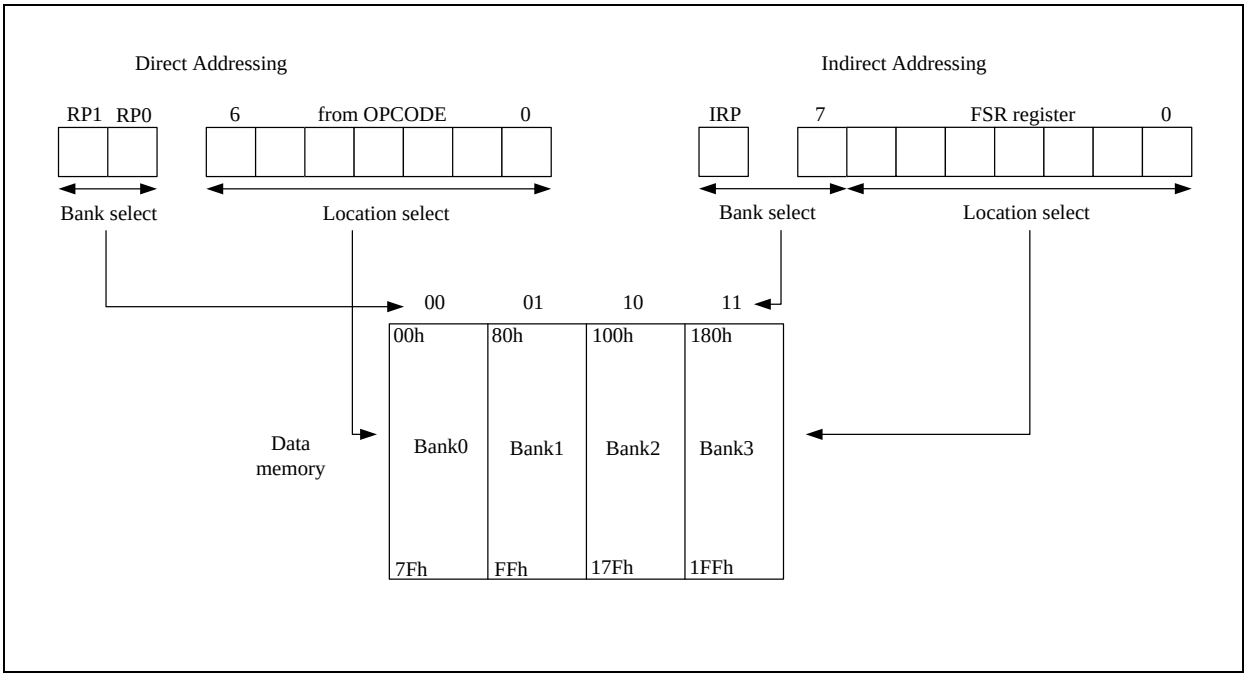
1.4 RAM 寻址模式

芯片中有一个数据存储器分为四个部分 BANK。每个 BANK 最多可扩展到 7Fh (128 字节)。每个 BANK 的下部位置保留用于特殊功能寄存器 (SFR)。SFR 上方是通用寄存器，实现为静态 RAM。所有已实现的 BANK 都包含特殊功能寄存器。BANK 的一些常用特殊功能寄存器可能会在另一个 BANK 中进行镜像，以减少代码并加快访问速度。

RP1 和 RP0 位 (STATUS [6:5]) 是 BANK 选择位。

[RP1, RP0]	BANK
00	0
01	1
10	2
11	3

该页面可以直接或间接寻址。对 INDF 寄存器进行写值为间接寻址，INDF 寄存器不是实质寄存器，任何使用 INDF 寄存器的指令实际上都为访问文件选择寄存器 FSR 指向的寄存器。间接读取 INDF 寄存器本身 (FSR =“ 0”)将读为 00h，间接写入 INDF 寄存器将导致空操作（可能会影响状态位）。通过将 8 位 FSR 寄存器和 IRP 位 (STATUS [7]) 连接起来，可以获得有效的 9 位地址。请参考下图。



直接 / 间接寻址

	【BANK0】 000~07Fh		【BANK1】 080h~0FFh		【BANK2】 100h~17Fh		【BANK3】 180h~1FFh
000h	**INDF**	080h	**INDF**	100h	**INDF**	180h	**INDF**
001h		081h		101h		181h	
002h	**PCL**	082h	**PCL**	102h	**PCL**	182h	**PCL**
003h	**STATUS**	083h	**STATUS**	103h	**STATUS**	183h	**STATUS**
004h	**FSR**	084h	**FSR**	104h	**FSR**	184h	**FSR**
005h	PAD	085h	PAD	105h	PAD	185h	PAD
006h	PBD	086h	PBD	106h	PBD	186h	PBD
007h		087h		107h		187h	
008h		088h		108h		188h	
009h		089h		109h		189h	
00Ah	**PCLATH**	08Ah	**PCLATH**	10Ah	**PCLATH**	18Ah	**PCLATH**
00Bh	**INTIE**	08Bh	**INTIE**	10Bh	**INTIE**	18Bh	**INTIE**
00Ch	INTIF	08Ch	INTIF	10Ch	INTIF	18Ch	INTIF
00Dh	DPL	08Dh	DPL	10Dh	DPL	18Dh	DPL
00Eh	DPH	08Eh	DPH	10Eh	DPH	18Eh	DPH
00Fh	CLKCTL	08Fh	CLKCTL	10Fh	CLKCTL	18Fh	CLKCTL
010h	OPTION	090h	TM0	110h		190h	
011h	PAMODL	091h	TM0CTL	111h		191h	
012h	PAMODH	092h	TM1	112h		192h	
013h	PBMODL	093h	TM1RLD	113h		193h	
014h	PBMODH	094h	TM1CTL	114h		194h	
015h	PAPU	095h	PWM0MDZ	115h		195h	
016h	PAPD	096h	PWM0D	116h		196h	
017h	PBPU	097h	PWM1D	117h		197h	
018h	PBPD	098h	PWM01CTL	118h		198h	
019h	TM2	099h	T1PWMH	119h		199h	
01Ah	TM2RLD	09Ah	ADCDH	11Ah		19Ah	
01Bh	TM2CTL	09Bh	ADCTL	11Bh		19Bh	
01Ch	PWM2D	09Ch	ADCTL2	11Ch		19Ch	
01Dh	PWM3D	09Dh	CMPCTL	11Dh		19Dh	
01Eh	PWM23CTL	09Eh	CMPPNS	11Eh		19Eh	
01Fh		09Fh	DACTL	11Fh		19Fh	
020h	RAM Bank0 area (80 Bytes)	0A0h	RAM Bank1 area (32 Bytes)	120h		1A0h	
06Fh		0EFh		16Fh		1EFh	
070h	**common area (16 Bytes)**	0F0h	**accesses 070h~07Fh**	170h	**accesses 070h~07Fh**	1F0h	**accesses 070h~07Fh**
07Fh		0FFh		17Fh		1FFh	

◇ 范例: 通过使用直接寻址来读 / 写寄存器

```
RAM020    Equ     020h     ; RAM 在 Bank0
RAM0A0    Equ     0A0h     ; RAM 在 Bank1

MOVLW     16h              ; W = 16h

BCF       RP0              ;
BCF       RP1              ;
MOVWF     RAM020           ; RAM[0x20] = W = 16h

BCF       RP0              ;
BSF       RP1              ;
MOVWF     RAM020           ; RAM[0xA0] = W = 16h
```

◇ 范例: 通过使用间接寻址来读 / 写寄存器 **(RP0=RP1=0)**

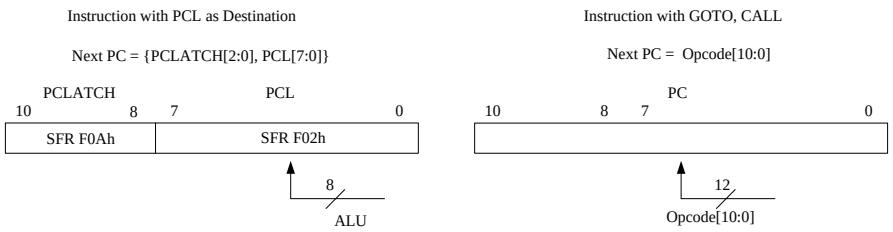
```
BCF       IRP              ; IRP = 0 => Bank0/1
MOVLW     0Fh              ; W = 0Fh
MOVWF     FSR              ; FSR = W = 0Fh
MOVFW     INDF             ; 读取 SFR 0Fh 至 W

MOVLW     0Fh              ; W = 0Fh
MOVWF     FSR              ; FSR = W = 0Fh
MOVLW     0Bh              ; W = 0Bh
MOVWF     INDF             ; CLKCTL (00Fh) = W = 0Bh
```

1.5 程序计数器（PC）和堆叠

该程序计数器为 11 位的编程计数器，能够寻址一个 2Kx14 的闪存 ROM。当执行一个程序指令时，PC 将包含下一个要执行的程序指令的地址。对于 CALL/GOTO 指令，PC 从指令字加载下 11 位地址。对于 RET/RETI/RETLW 指令，PC 从顶级堆栈中检索其内容。初始设置复位向量 (000h) 和中断向量 (004h) 用于 PC 初始化和中断。

执行任何以 PCL 寄存器为目标的指令，编程计数器的高 3 位(PC[10:8])是由 PCLATH (00Ah/08Ah/10Ah/18Ah) 寄存器的内容所取代。当低字节被写入 PCL 寄存器时，程序计数器的所有内容将变为 PCLATH 寄存器中包含的值和那些被写入 PCL 寄存器的值。

。



002h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PCL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

002h.7~0 **PCL:** 程序计数器数据位 7~0

00Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLATH	-					PCLATH		
R/W	-					R/W		
Reset	-	-	-	-	-	0	0	0

00Ah.2~0 **PCLATH:** 当执行以 PCL 为目标的指令时编程计数器高字节数据

◇ 范例 1: 要查找位于“ TABLE”数据, 使用指令 TABRL, TABRH

```
ORG     000h                    ; 复位向量
        GOTO    START
START:
        MOVLW   02h
        MOVWF   DPH
        MOVLW   00h
        MOVWF   DPL             ; DPTR = {DPH, DPL} = TABLE
; 通过指令 TABRL / TABRH 读表
        TABRL                   ; 将 DPTR选到的位置低字节数据读到 W (W = 34h)
        TABRH                   ; 将DPTR选到的位置高字节数据读到W (W = 12h)
                                ; 高字节数据读到6位
                                ; 低字节数据读到8位
        ORG     200H            ;
TABLE:
        .DT     1234H           ; 14位存值
```

◇范例 2：查询位于 “TABLE” 的资料, 使用 SFR PCLATH, 与 PCL

```
INDEX   Equ     020h            ; RAM 在 Bank0
        ORG     000H            ; 清除向量
        GOTO    START           ; 至用户程序地址
START:
        MOVLW   00H
        MOVWF   INDEX           ; 设定查表地址 (INDEX)
        MOVLW   02H
        MOVWF   PCLATH          ; 设定查表地址高位 (PC[10:8])
LOOP:
        MOVFW   INDEX           ; 移动INDEX值至W寄存器
        CALL    TABLE           ; 查询资料 (W = 55H当INDEX = 00H)
        …
        INCF    INDEX, 1        ; 递增INDEX至下一个地址
        …
        GOTO    LOOP            ; 至LOOP标签

        ORG     200H            ; 起始查表位置 (PC[10:8]=2)
TABLE:
        ADDWF   PCL, 1          ; (Addr = X00H) 相加W和PCL，结果存回PCL
        RETLW   55H             ; 回传时，W = 55H
        RETLW   56H             ; 回传时，W = 56H
        RETLW   58H             ; 回传时，W = 58H
```

备注：芯片定义 PC 每 256 个地址为 1 页(PC[7:0]), 芯片 2K ROM 共可分为 8 页(PC[10:8]), 范例 2 中的 TABLE 表必需放置在同一个页, 即同一个 TABLE 必需有相同的高位 PC[10:8], 若 TABLE 存放跨越不同页, 则使用 SFR PCLATH 与 PCL 查表时会造成查表错误。

1.6 ALU 和工作寄存器（W）

ALU 有 8 位宽，可进行加，减，移位和逻辑运算的操作。在两个操作数的指令中，主操作数是一个 8 位不可寻址的 W 寄存器用于 ALU 运算。另一个操作数是一个文件寄存器或一个立即常数。在单个操作数的指令中，操作数即可是 W 寄存器，也可是文件寄存器。决定于执行的指令，ALU 可能影响状态寄存器中的标志位 C，DC，和 Z。C 和 DC 标志运用于借位和十进制借位，特别是在减法中。

备注：借位表示借位寄存器的反转

十进制借位表示十进制借位寄存器的反转

1.7 状态寄存器 (F-Plane 03H)

状态寄存器包含 ALU 计算结果的状态，复位状态及电压状态。状态寄存器和其他寄存器一样可以是任何指令的目的单元。如果状态寄存器是影响 Z，DC，或 C 标志位的指令目的地，那么就不可以写进这三个标志位了。这些位的设定或清除是根据设备的逻辑运算。建议，只有 BCF，BSF，和 MOVWF 指令用来改变状态寄存器因为这些指令不会影响那些标志位。

<table>
<tr><th>状态</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>复位值</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R</td><td>R</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>位</td><td colspan="8">描述</td></tr>
<tr><td>7</td><td colspan="8">IRP: 寄存器位于 Bank 的 选择位 (用于间接寻址)
0 = Bank 0,1 (000h - 0FFh)
1 = Bank 2,3 (100h - 1FFh)</td></tr>
<tr><td>6:5</td><td colspan="8">RP1:RP0: 寄存器位于 Bank 的选择位 (用于直接寻址)
00 = Bank 0 (000h - 07Fh)
01 = Bank 1 (080h - 0FFh)
10 = Bank 2 (100h - 17Fh)
11 = Bank 3 (180h - 1FFh)
每个 Bank 为 128 字节</td></tr>
<tr><td>4</td><td colspan="8">TO: 超时标志
0: 上电复位或 CLRWDT/SLEEP 指令后
1: WDT 超时</td></tr>
<tr><td>3</td><td colspan="8">PD: 掉电标志
0: 上电复位或 CLRWDT 指令后
1: 执行 SLEEP 指令后</td></tr>
<tr><td>2</td><td colspan="8">Z: 零标志位
0: 逻辑操作的结果不是 0
1: 逻辑操作的结果是 0</td></tr>
<tr><td rowspan="3">1</td><td colspan="8">DC: 十进制进位标志或十进制借位标志</td></tr>
<tr><td colspan="4">加法指令</td><td colspan="4">减法指令</td></tr>
<tr><td colspan="4">0: 无进位
1: 低四位有进位</td><td colspan="4">0: 低四位有借位
1: 无借位</td></tr>
<tr><td rowspan="3">0</td><td colspan="8">C: 进位标志或借位标志</td></tr>
<tr><td colspan="4">加法指令</td><td colspan="4">减法指令</td></tr>
<tr><td colspan="4">0: 无进位
1: MSB 有进位</td><td colspan="4">0: MSB 有借位
1: 无借位</td></tr>
</table>

◇范例：写入立即数到状态寄存器

```
MOVLW   00H
MOVWF   STATUS          ; 清除状态寄存器
```

◇范例：位寻址设定和清除状态寄存器

```
BSF     STATUS, 0       ; 设定 C = 1
BCF     STATUS, 0       ; 清除C = 0
```

◇范例：以 BTFSS 指令判断 C 标志

```
BTFSS   STATUS, 0       ; 检查进位标志
GOTO    LABEL_1         ; 若C = 0，前往LABEL_1
GOTO    LABEL_2         ; 若C = 1，前往LABEL_2
```

◇范例：设定 INT0 中断要求和上升沿触发

```
            ORG     000H            ; 复位向量
            GOTO    START           ; 前往用户程序地址

            ORG     004H            ; 所有中断向量
            GOTO    INT             ; 若INT0 输入发生上升沿

START:
            BCF     RP1
            BCF     RP0
            MOVLW   xx00xxxxB
            MOVWR   PAMODL          ; INT0(PA2) 开漏(输入)模式
            MOVLW   01000000B
            MOVWF   OPTION          ; 设置INT0上升沿中断
            MOVLW   11111110B
            MOVWF   INTIF           ; 清除INT0中断要求标志
            MOVLW   00000001B
            MOVWF   INTE            ; 使能INT0中断
MAIN:
            …
            GOTO    MAIN

INT:
            …       …               ; 参考插入第4章进出中断范例程序

            BTFSS   INT0IF          ; 检查INT0IF位
            GOTO    EXIT_INT        ; INT0IF = 0，跳出中断程序
            …                       ; INT0中断服务程序
            MOVLW   11111110B
            MOVWF   INTIF           ; 清除INT0上升沿中断要求标志
EXIT_INT:
            …       …               ; 参考插入第4章进出中断范例程序
            RETI                    ; 从中断回来
```

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	CMPIE	TM2IE	TM1IE	TM0IE	PCXIE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.7 **ADCIE:** ADC 中断使能
0: 关闭
1: 使能

0Bh.6 **CMPIE:** 比较器中断使能
0: 关闭
1: 使能

0Bh.5 **TM2IE:** Timer2中断使能
0: 关闭
1: 使能

0Bh.4 **TM1IE:** Timer1 中断使能
0: 关闭
1: 使能

0Bh.3 **TM0IE:** Timer0 中断使能
0: 关闭
1: 使能

0Bh.2 **PCXIE:** 引脚电平变化中断使能
0: 关闭
1: 使能

0Bh.1 **INT1IE:** INT1 中断使能
0: 关闭
1: 使能

0Bh.0 **INT0IE:** INT0 中断使能
0: 关闭
1: 使能

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	CMPIF	TM2IF	TM1IF	TM0IF	PCXIF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.7 **ADCIF:** ADC 中断事件挂起标志
当 ADC 转换结束后由 H/W 置位，对此位写 0 将清除该标志

0Ch.6 **CMPIF:** 比较器中断事件挂起标志
当 CMPO 匹配触发条件时由 H/W 置位，对此位写 0 将清除该标志

0Ch.5 **TM2IF:** Timer2 中断事件挂起标志
当 Timer2 溢出时由 H/W 置位，对此位写 0 将清除该标志

0Ch.4 **TM1IF:** Timer1 中断事件挂起标志
当 Timer1 溢出时由 H/W 置位，对此位写 0 将清除该标志

0Ch.3 **TM0IF:** Timer0 中断事件挂起标志
当 Timer0 溢出时由 H/W 置位，对此位写 0 将清除该标志

0Ch.2 **PCXIF:** 引脚电平变化中断挂起标志
当引脚电平变化由 H/W 置位，对此位写 0 将清除该标志

0Ch.1 **INT1IF:** INT1 引脚下降沿/上升沿中断挂起标志
当 INT1 引脚发生下降/上升沿时由 H/W 置位，对此位写 0 将清除该标志

0Ch.0 **INT0IF:** INT0 引脚下降沿/上升沿中断事件挂起标志
当 INT0 引脚发生下降/上升沿时由 H/W 置位，对此位写 0 将清除该标志

2. 复位

芯片有 3 种复位方式.

- 上电复位 (POR)
- 低电压复位 (LVR)
- 看门狗复位 (WDT)

复位可以是由上电复位 (POR)、看门狗定时器复位 (WDTR) 或低电压复位 (LVR) 引起的。复位后，SFR 返回其默认值，程序计数器 (PC) 被清除，并且系统从复位向量 000h 处开始运行。状态寄存器 (STATUS) 上的 TO 和 PD 标志指示系统复位状态。

2.1 上电复位 (POR)

在通电重置后，所有系统和外围控制寄存器将被设置为它们的默认硬件重置值。

2.2 低电压复位 (LVR)

芯片的低电压复位可通过系统配置寄存器 (SYSCFG)选择。当电源电压下降至 LVR 电压时产生复位。

当 LVRSAV(F10h.2)设置时, LVR 在系统进入空闲或停止模式时将自动关闭，进入运行模式后自动开启。

当 LVRSAV(F10h.2)设置时(默认为 1), 若主时钟源为慢时钟，则 LVR 将一直关闭。当主时钟源在快时钟时, LVR 则会启用。因此当 VCC 小于设定的 LVR 电压时, 主时钟源由慢时钟切换至快时钟后会立刻产生复位。复位后主时钟源回到慢时钟继续运行。

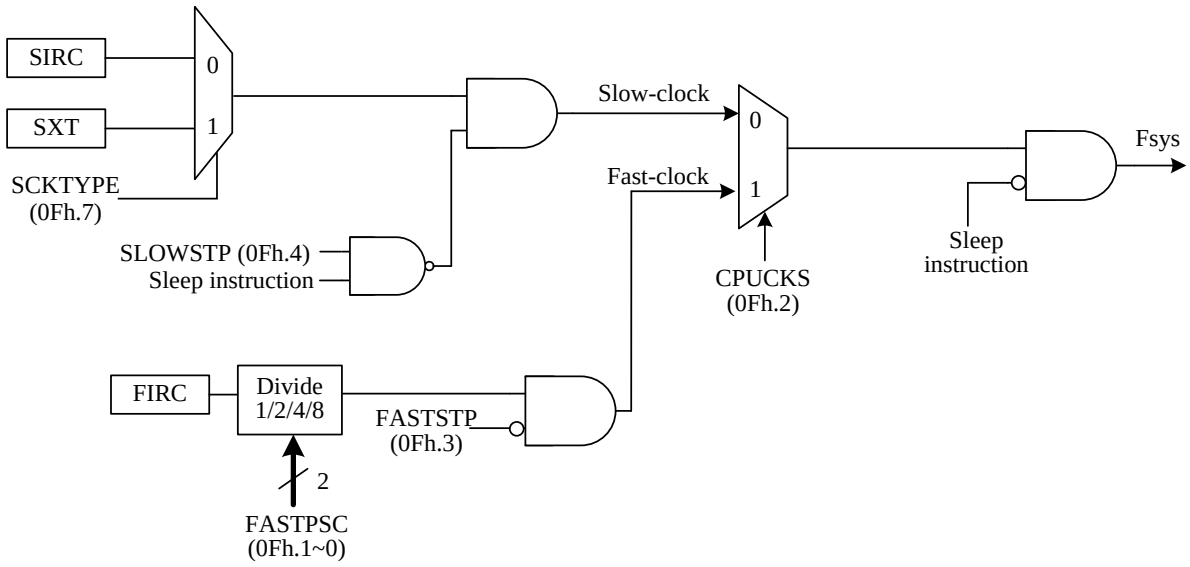
2.3 看门狗定时器复位 (WDT)

WDT 的预分频器与 Timer0 共享。
WDT 溢出复位可通过系统配置寄存器 (SYSCFG WDTC)与 SFR WDTE 设置。
WDTC/WDTE 必需同时为 1 才会启用 WDT
WDT 启用时, 执行 SLEEP 指令后即进入空闲模式, 即 SIRC 与 WDT 会继续计数
WDT 启用时,在主时钟为快时钟时进入空闲模式下溢出, 此时为溢出复位
WDT 启用时,在主时钟为慢时钟时进入空闲模式下溢出, 此时为溢出唤醒, 不会产生复位。

3. 时钟电路和工作模式

3.1 系统时钟

芯片为双时钟系统，可选择系统主时钟为快时钟或慢时钟。快时钟为内部高速 RC 振荡器 16MH (FIRC), 并可选择/1/2/4/8 的分频时钟。慢时钟可分为内部低频 RC 振荡器约 50KHz(SIRC) 与外接晶体振荡器。WDT（看门狗）电路的时钟源固定为内部低频 RC 振荡器。**快钟源预分频器 (0Fh.1~0) 必需在主时钟为慢时钟时才能切换。**



3.2 系统工作模式

芯片可分为运行模式和停止模式与空闲模式。当执行 SLEEP 指令时, 若 SLOWSTP(F0Fh.4)为 0 则进入空闲模式, 如果 SLOWSTP(F0Fh.4)为 1 则进入停止模式。空闲模式下 WDT 仍可计数, 停止模式下 WDT 停止计数。

工作模式	切入条件	系统状态
运行模式	1. 复位 2. 空闲/停止模式唤醒	主时钟运行
空闲模式	1. SLOWSTP(F0Fh.4)=0 或WDTC/WDTE同时为1 2. SLEEP指令	主时钟停止, 慢时钟运行
停止模式	1. SLOWSTP(F0Fh.4)=1 2. WDTC或WDTE为0 3. SLEEP指令	主时钟停止, 慢时钟停止

3.3 主时钟切换

- **主时钟由快时钟切换至慢时钟**

应按顺序执行以下步骤, **不可同时改变 CPUCKS 与 FASTSTP**:

(1) 切换至慢速模式 (CPUCKS=0)
(2) 停止快时钟 (FASTSTP=1)

◇ 范例: 快时钟切换至慢时钟

```
BCF     CPUCKS      ; Fsys=慢时钟
BSF     FASTSTP     ; 停止快时钟
```

- **主时钟由慢时钟式切换至快时钟**

应按顺序执行以下步骤，**不可同时改变 CPUCKS 与 FASTSTP**:

(1) 启用快时钟 (FASTSTP=0)
(2) 依需求选择快钟源预分频器
(3) 切换到快速模式 (CPUCKS=1)

◇ 范例：慢时钟切换到快时钟

```
BCF     FASTSTP     ; 使能快时钟
BSF     FASTPSC0    ; 快钟源预分频器选择
BSF     FASTPSC1    ; 快钟源预分频器选择
BSF     CPUCKS      ; 主时钟(Fsys)=快时钟
```

注: 快钟源预分频器(0Fh.1~0) 必需在主时钟为慢时钟时才能切换

- **空闲模式设置**

空闲模式可以按顺序设置进行配置:

(1) 启用慢时钟 (SLOWSTP=0)
(2) 执行 SLEEP 指令

◇ 范例:切换至空闲模式.

```
BCF     SLOWSTP     ; 在执行 SLEEP 指令后慢时钟持续运行
SLEEP               ; 进入空闲模式
```

- **停止模式设置**

可以通过以下顺序设置停止模式:

(1) 执行 SLEEP 执令后停止慢时钟 (SLOWSTP=1)
(2) 停止 WDT 计数(WDTE=0)
(3) 执行 SLEEP 指令

◇ 范例:切换至停止模式.

```
banksel     OPTION      ; 切换SFR 至bamk0
BSF         SLOWSTP     ; SLEEP 后关闭SIRC
BCF         WDTE        ; 停止WDT计数
SLEEP                   ; 进入停止模式.
```

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	SCKTYPE	–	–	SLOWSTP	FASTSTP	CPUCKS	FASTPSC	
R/W	R/W	–	–	R/W	R/W	R/W	R/W	
Reset	0	–	–	0	1	0	1	1

0Fh.7 **SCKTYPE**: 慢时钟选择
0: 慢时钟为 SIRC
1: 慢时钟为 SXT

0Fh.4 **SLOWSTP**: 在 SLEEP 指令后停止慢时钟
0: 慢时钟在 SLEEP 指令后持续运行
1: 慢时钟在 SLEEP 指令后停止运行

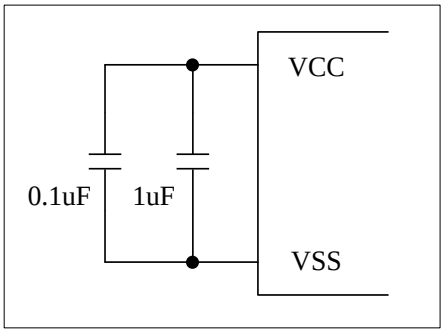
0Fh.3 **FASTSTP**: 停止快时钟
0: 快时钟运行
1: 快时钟停止

0Fh.2 **CPUCKS**: 主时钟源选择
0: 慢时钟
1: 快时钟

0Fh.1~0 **FASTPSC**: 快钟源预分频器,只能在主时钟为慢时钟时才能切换。
00: 除以 8
01: 除以 4
10: 除以 2
11: 除以 1

3.4 主时钟振荡器

在内部快速 RC (FIRC) 模式下，片上振荡器产生 16 MHz 的系统时钟。由于电源噪声会降低内部时钟振荡器的性能，因此将电源旁路电容器 1 uF 和 0.1 uF 放置在靠近 VCC/VSS 引脚，可以提高时钟和整个系统的稳定性。



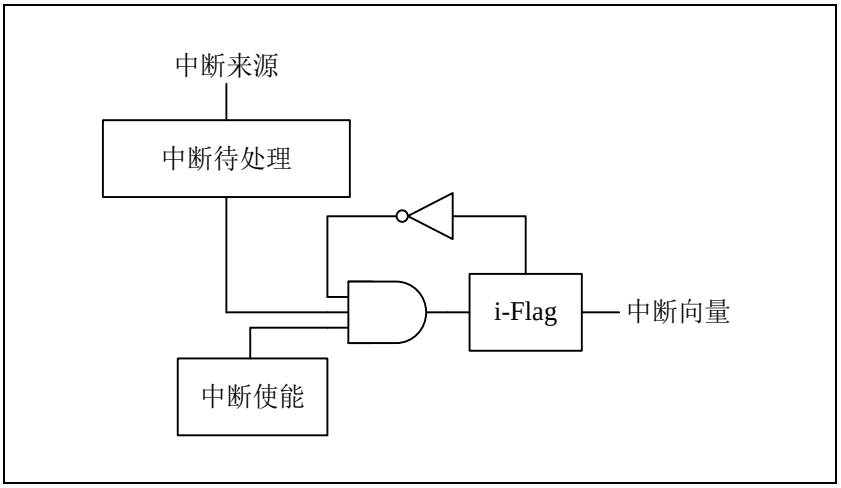
Internal RC Mode

4. 中断

TM54M0321 有一级，一个向量，8 个中断源。每个中断源有它自己启用控制位。每个中断事件都可以触发它自己的中断标志位，不论它的中断启用控制位是 0 或 1。进出中断方式请参考范例程序。

若相对的中断启用控制位已被设定（INTE），它将触发 CPU 服务此中断。CPU 接受目前执行的指令周期结尾的中断。同时，“CALL 004” 指令被插进 CPU，并且 i-flag 被设定来防止中断嵌套。

中断嵌套在执行 “RETI” 指令后即被清零。所以，待处理中断时，至少一个指令在主程序中执行着。在处理中断程序后，F/W 必须清除中断事件寄存器。



◇ 范例: 进入中断储存 W / FSR / PCLATH 范例程序

```
        RAM70     EQU     070h      ;
        RAM71     EQU     071h      ;
        RAM72     EQU     072h      ;
        RAM7E     EQU     07Eh      ;

ORG     0004h
        MOVWF     RAM7E             ; 备份 W
        SWAPF     STATUS, W
        MOVWF     RAM70             ; 备份 STATUS
        MOVFW     FSR
        MOVWF     RAM71             ; 备份 FSR
        MOVFW     PCLATH
        MOVWF     RAM72             ; 备份 PCLATH
        …         …                 ; 接续用户中断程序
```

◇ 范例: 离开中断还原 W / FSR / PCLATH 范例程序

```
        RAM70     EQU     070h      ;
        RAM71     EQU     071h      ;
        RAM72     EQU     072h      ;
        RAM7E     EQU     07Eh      ;

        …         …                 ; 接续用户中断程序
        MOVFW     RAM72
        MOVWF     PCLATH            ; 还原 PCLATH
        MOVFW     RAM71
        MOVWF     FSR               ; 还原 FSR
        SWAPF     RAM70, W
        MOVWF     STATUS            ; 还原 STATUS,
        SWAPF     RAM7E, F
        SWAPF     RAM7E, W          ; 还原 W
        RETI
```

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	CMPIE	TM2IE	TM1IE	TM0IE	PCXIE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.7 **ADCIE:** ADC 中断使能
0: 关闭
1: 使能

0Bh.6 **CMPIE:** 比较器中断使能
0: 关闭
1: 使能

0Bh.5 **TM2IE:** Timer2中断使能
0: 关闭
1: 使能

0Bh.4 **TM1IE:** Timer1 中断使能
0: 关闭
1: 使能

0Bh.3 **TM0IE:** Timer0 中断使能
0: 关闭
1: 使能

0Bh.2 **PCXIE:** 引脚电平变化中断使能
0: 关闭
1: 使能

0Bh.1 **INT1IE:** INT1 中断使能
0: 关闭
1: 使能

0Bh.0 **INT0IE:** INT0 中断使能
0: 关闭
1: 使能

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	CMPIF	TM2IF	TM1IF	TM0IF	PCXIF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.7 **ADCIF:** ADC 中断事件挂起标志
当 ADC 转换结束后由 H/W 置位，对此位写 0 将清除该标志

0Ch.6 **CMPIF:** 比较器中断事件挂起标志
当 CMPO 匹配触发条件时由 H/W 置位，对此位写 0 将清除该标志

0Ch.5 **TM2IF:** Timer2 中断事件挂起标志
当 Timer1 溢出时由 H/W 置位，对此位写 0 将清除该标志

0Ch.4 **TM1IF:** Timer1 中断事件挂起标志
当 Timer1 溢出时由 H/W 置位，对此位写 0 将清除该标志

0Ch.3 **TM0IF:** Timer0 中断事件挂起标志
当 Timer0 溢出时由 H/W 置位，对此位写 0 将清除该标志

0Ch.2 **PCXIF:** 引脚电平变化中断挂起标志
当引脚电平变化由 H/W 置位，对此位写 0 将清除该标志

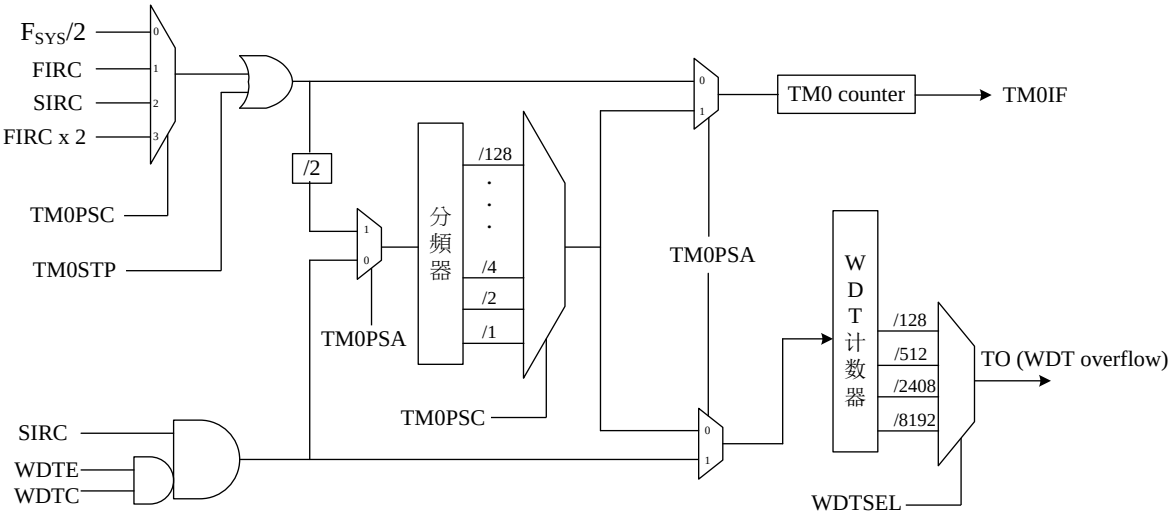
0Ch.1 **INT1IF:** INT1 引脚下降沿/上升沿中断挂起标志
当 INT1 引脚发生下降/上升沿时由 H/W 置位，对此位写 0 将清除该标志

0Ch.0 **INT0IF:** INT0 引脚下降沿/上升沿中断事件挂起标志
当 INT0 引脚发生下降/上升沿时由 H/W 置位，对此位写 0 将清除该标志

5. Timer0 与 WDT

Timer0 是一个 8 位计数器, 包含一个 8 位递增计数器与可编程分频器,可产生中断与唤醒功能。Timer0 有 4 种时钟源可选, 包含系统时钟除 2、快时钟、慢时钟与快时钟乘 2。在使用 Timer0 时, 应将时钟源、分频器选择等先设置完成后, 最后再开启 Timer0 计数(即设置 TM0STP=0) 。

- TM0 为递增计数
- 分频器为 Timer0 与 WDT 共享，通过寄存器位 TM0PSA 控制分频器的分配
- TM0PSA 为 1 时, 写入 TM0 时, 将清零分频器
- TM0PSA 为 0 时, 执行 CLRWDT 与 STOP 指令时, 将清零分频器
- Timer0 慢时钟输入时, 在空闲模式下 Timer0 将持续计数, 支持溢出唤醒功能
- WDTC/WDTE 必需同时为 1 才会启用 WDT
- WDT 启用时, 执行 SLEEP 指令后即进入空闲模式, 即 SIRC 与 WDT 会继续计数
- WDT 启用时,在主时钟为快时钟时进入空闲模式下溢出, 此时为溢出复位
- WDT 启用时,在主时钟为慢时钟时进入空闲模式下溢出, 此时为溢出唤醒, 不会产生复位
- 选择 FIRCx2 为时钟源时, 工作电压必需大于 3V



Timer0 区块图

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	CMPIE	TM2IE	TM1IE	TM0IE	PCXIE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.3 **TM0IE:** Timer0 中断使能
0: 关闭
1: 使能

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	CMPIF	TM2IF	TM1IF	TM0IF	PCXIF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.3 **TM0IF:** Timer0 中断事件挂起标志
当 Timer0 溢出时由 H/W 置位，对此位写 0 将清除该标志

F90	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0	TM0							
R/W	R/W							
复位	0	0	0	0	0	0	0	0

F90.7~0 **TM0:** Timer0内容

91h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0CTL	-	TM0STP	TM0PSA	TM0CKS		TM0PSC		
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0

91h.6 **TM0STP:** Timer0 启用
0: Timer0计数
1: Timer0 停止计数

91h.5 **TM0PSA:** 分频器选择
0: WDT使用分频器
1: Timer0 使用分频器

91h.4~3 **TM0CKS:** Timer0 时钟源选择
00: 系统时钟除2(F_{SYS}/2)
01: 快时钟
10: 慢时钟
11: 快时钟乘2 (必需工作于3V以上)

91h.2~0 **TM0PSC:** Timer0 分频器选择
000: 1分频
001: 2分频
010: 4分频
011: 8分频
100: 16分频
101: 32分频
110: 64分频
111: 128分频

03h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset	0	0	0	1	0	1	0	0

03h.4 **TO:** 超时标志
0: 上电复位或CLRWDT/SLEEP 指令后
1: WDT 超时

10h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	-	INT0EDG	INT1EDG	WDTE	PORPD	LVRSAV	PORSAV	LVRO
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset	-	0	0	1	0	1	0	0

10h.4 **WDTE:** WDT 启用
0: WDT禁止
1: WDT启用 (WDTC/WDTE必需同时为1才会启用WDT)

位		描述	
SYSCFG	5	**WDTC**: 模式设置	
		0	WDT 始终关闭
		1	WDT 开启 (WDTC/WDTE 必需同时为 1 才会启用 WDT)
	3-2	**WDTSEL**: WDT 溢出时间选择（无预分频）	
		00	2.5ms
		01	10ms
		10	40ms
		11	160ms

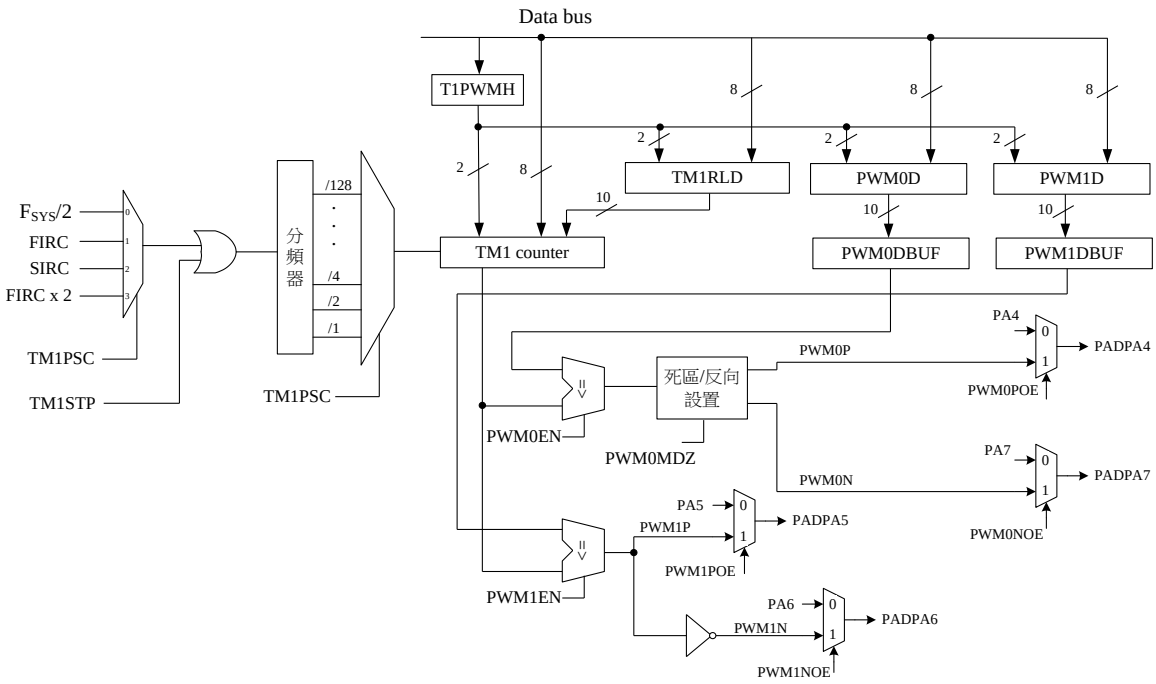
6. Timer1 (PWM0/PWM1)

Timer1 是一个 10 位计数器, 包含一个 10 位计数器与可编程分频器,可产生中断与唤醒功能。Timer1 可产生 2 组 PWM (PWM0/PWM1) 。PWM0P/PWM0N，可设置死区与四种不同的输出模式 (4 种互补输出方式) 。Timer1 有 4 种时钟源可选, 包含系统时钟除 2、快时钟、慢时钟与快时钟乘 2。在使用 Timer1 时, 应将时钟源、分频器选择等先设置完成后, 最后再开启 Timer1 计数(即设置 TM1STP=0) 。

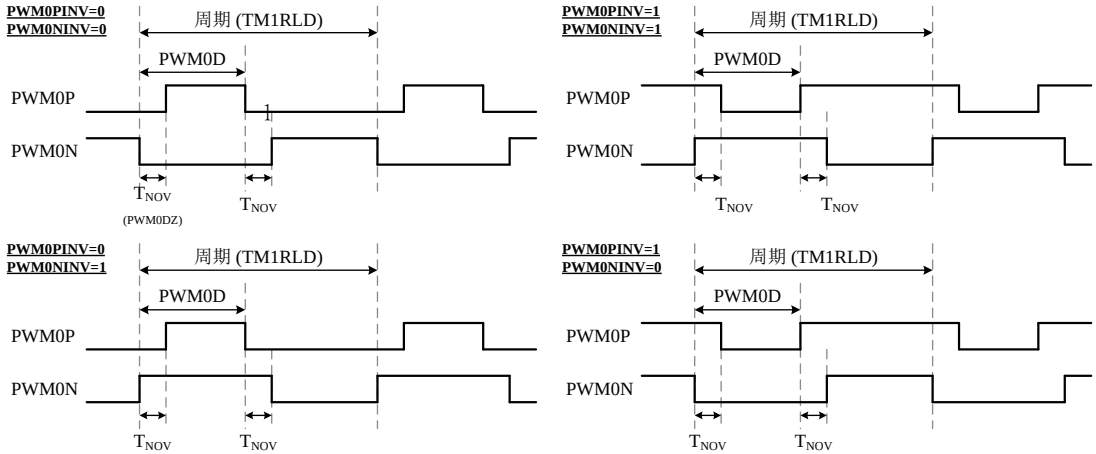
- TM1 为递减计数
- TM1RLD 计数器共享为 PWM0P/PWM0N, PWM1P/PWM1N 周期寄存器
- PWM0P/PWM0N 共享一组 10 位占空比寄存器
- PWM1P/PWM1N 共享一组 10 位占空比寄存器
- PWM0P/PWM0N，可设置死区与四种不同的输出模式(4 种互补输出方式)。
- 10 位 TM1 计数器和占空比寄存器均具有低字节和高字节结构。访问方式为:
  先写入 2 位高字节 T1PWMH(F99h)，再写 8 位低字节(PWM0D/PWM1D/TM1/TM1RLD)。
  先读取 8 位低字节(PWM0D/PWM1D/TM1/TM1RLD)，再读取 2 位高字节 T1PWMH(F99h)。
- 10 位 TM1RLD 与 与占空比寄存器的读写, 建议只在主程序内更新数据, 或只在中断内更新数据, 以避免可能的错误产生
- PWM0 (PWM0EN) 关闭时写 PWM0D，将立即载入缓冲器(PWM0DBUF)中。
- PWM0 使能后写 PWM0D 则将在 TM1 溢出时才载入比较缓冲器中
- PWM1 (PWM1EN) 关闭时写 PWM1D，将立即载入缓冲器(PWM1DBUF)中。
- PWM1 使能后写 PWM1D 则将在 TM1 溢出时才载入比较缓冲器中
- PWM 关闭时输出为低电平
- PWM0/PWM1 同时关闭时, 此时若开启 PWM0EN 或 PWM1EN 会将重载值(TM1RLD)写入 TM1 计数器后再计数
- 选择 FIRCx2 为时钟源时, 工作电压必需大于 3V

PWM0 占空比计算如下：

PWM0 高电平 =（PWM0D） ×TM1 时钟周期
PWM0 周期 =（TM1RLD+1）×TM1 时钟周期
占空比 = 高电平时间 / 周期 =（PWM0D）/（TM1RLD+1）
當 PWM0D > TM1RLD 時, 占空比為 100%



Timer1 区块图



PWM0 输出模式(4 种互补输出方式)

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	CMPIE	TM2IE	TM1IE	TM0IE	PCXIE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.4 **TM1IE:** Timer1 中断使能
0: 关闭
1: 使能

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	CMPIF	TM2IF	TM1IF	TM0IF	PCXIF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.4 **TM1IF:** Timer1 中断事件挂起标志
当 Timer1 溢出时由 H/W 置位，对此位写 0 将清除该标志

F92	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1	TM1							
R/W	R/W							
复位	1	1	1	1	1	1	1	1

F92.7~0 **TM1:** Timer1低8位

F93	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1RLD	TM1RLD							
R/W	R/W							
复位	1	1	1	1	1	1	1	1

F93.7~0 **TM1RLD:** Timer1重载值低8位

94h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1CTL	-	-	TM1STP	TM1CKS		TM1PSC		
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

94h.5 **TM1STP:** Timer1 启用
0: Timer1计数
1: Timer 1 停止计数

94h.4~3 **TM1CKS:** Timer1 时钟源选择
00: 系统时钟除2(F_{SYS}/2)
01: 快时钟
10: 慢时钟
11: 快时钟乘2 (必需工作于3V以上)

94h.2~0 **TM1PSC:** Timer1 分频器选择
000: 1分频
001: 2分频
010: 4分频
011: 8分频
100: 16分频
101: 32分频
110: 64分频
111: 128分频

95h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0MDZ	-	-	PWM0PINV	PWM0NINV	PWM0DZ			
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

95h.5 **PWM0PINV:** PWM0P输出选择
0: 正向输出
1: 反向输出

95h.4 **PWM0NINV:** PWM0N输出选择
0: 正向输出
1: 反向输出

95h.3~0 **PWM0DZ:** PWM0 死区控制
0000: 死区禁用
0001: 死区宽度 1*Tpwmclk
0010: 死区宽度 2*Tpwmclk
…
1111: 死区宽度 15*Tpwmclk

F96	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0D	PWM0D							
R/W	R/W							
复位	0	0	0	0	0	0	0	0

F96.7~0 **PWM0D:** PWM0占空比低8位

F97	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1D	PWM1D							
R/W	R/W							
复位	0	0	0	0	0	0	0	0

F97.7~0 **PWM1D:** PWM1占空比低8位

98h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM01CTL	PWM0POE	PWM0NOE	PWM0EN	-	PWM1POE	PWM1NOE	PWM1EN	-
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	-
Reset	0	0	0	-	0	0	0	-

98h.7 **PWM0POE:** PWM0P 输出
0: PWM0P输出禁止
1: PWM0P输出使能

98h.6 **PWM0NOE:** PWM0N 输出
0: PWM0N输出禁止
1: PWM0N输出使能

98h.5 **PWM0EN:** PWM0 使能
0: 禁止
1: 使能

98h.3 **PWM1POE:** PWM1P 输出
0: PWM1P输出禁止
1: PWM1P输出使能

98h.2 **PWM1NOE:** PWM1N 输出
0: PWM1N输出禁止
1: PWM1N输出使能

98h.1 **PWM1EN:** PWM1 使能
0: 禁止
1: 使能

F99	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1PWMH	-	-	-	-	-	-	T1PWMH	
R/W	-	-	-	-	-	-	R/W	
复位	-	-	-	-	-	-	-	-

T1PWMH: TM1/TM1RLD/PWM0D/PWM1D 高2位(bit 9~8)访问

F99.1~0 先写入 2 位高字节，再写 8 位低字节(PWM0D/PWM1D/TM1/TM1RLD)。
先读取8位低字节(PWM0D/PWM1D/TM1/TM1RLD)，再读取2位高字节T1PWMH。

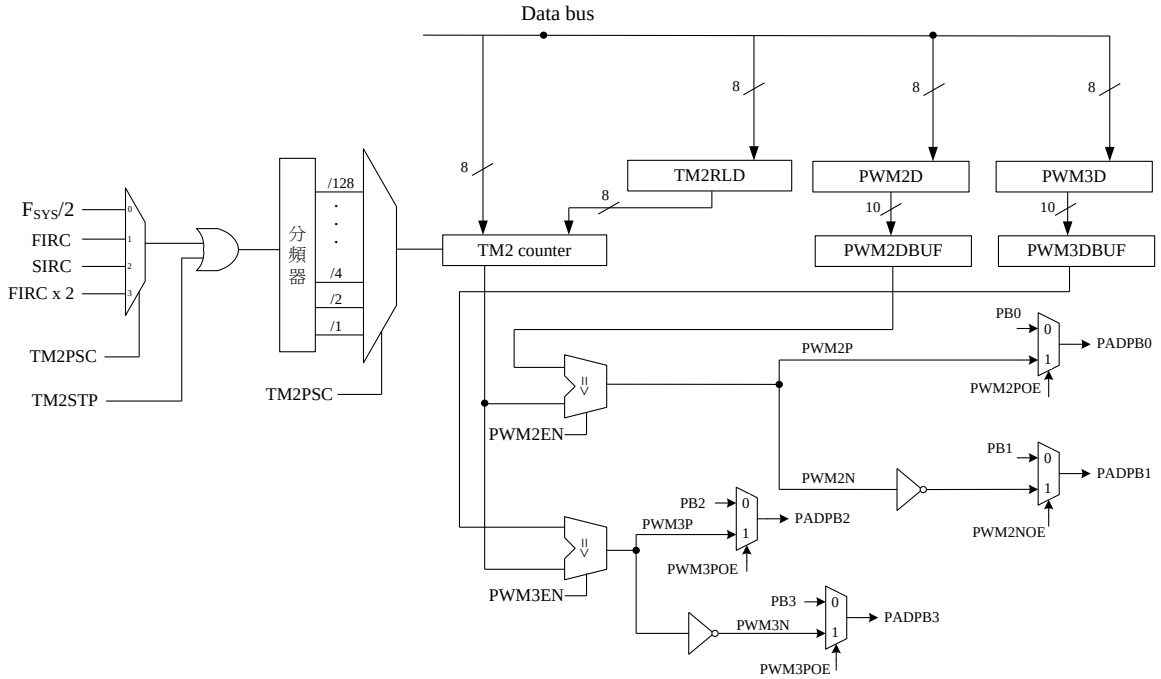
7. Timer2 (PWM2/PWM3)

Timer2 是一个 8 位计数器, 包含一个 8 位计数器与可编程分频器,可产生中断与唤醒功能。Timer2 可产生 2 组 PWM (PWM2/PWM3) 。Timer2 有 4 种时钟源可选, 包含系统时钟除 2、快时钟、慢时钟与快时钟乘 2。在使用 Timer2 时, 应将时钟源、分频器选择等先设置完成后, 最后再开启 Timer2 计数(即设置 TM2STP=0) 。

- TM2 为递减计数
- TM2RLD 共享为 PWM2P/PWM2N, PWM3P/PWM3N 周期寄存器
- PWM2P/PWM2N 共享一组 8 位占空比寄存器
- PWM3P/PWM3N 共享一组 8 位占空比寄存器
- PWM2 (PWM2EN) 关闭时写 PWM2D，将立即载入缓冲器(PWM2DBUF)中。
- PWM2 使能后写 PWM2D 则将在 TM2 溢出时才载入比较缓冲器中
- PWM3 (PWM3EN) 关闭时写 PWM3D，将立即载入缓冲器(PWM3DBUF)中。
- PWM3 使能后写 PWM3D 则将在 TM2 溢出时才载入比较缓冲器中
- PWM 关闭时输出为低电平
- PWM2/PWM3 同时关闭时, 此时若开启 PWM2EN 或 PWM3EN 会将重载值(TM2RLD)写入 TM2 计数器后再计数
- 选择 FIRCx2 为时钟源时, 工作电压必需大于 3V

PWM2 占空比计算如下：

PWM2 高电平 =（PWM2D） ×TM2 时钟周期
PWM2 周期 =（TM2RLD+1）×TM2 时钟周期
占空比 = 高电平时间 / 周期 =（PWM2D）/（TM2RLD+1）
當 PWM2D > TM2RLD 時, 占空比為 100%



Timer2 区块图

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	CMPIE	TM2IE	TM1IE	TM0IE	PCXIE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.5 **TM2IE:** Timer1 中断使能
0: 关闭
1: 使能

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	CMPIF	TM2IF	TM1IF	TM0IF	PCXIF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.5 **TM2IF:** Timer2 中断事件挂起标志
当 Timer2 溢出时由 H/W 置位，对此位写 0 将清除该标志

F19	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2	TM2							
R/W	R/W							
复位	1	1	1	1	1	1	1	1

F19.7~0 **TM2:** Timer1计数器

F1A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2RLD	TM2RLD							
R/W	R/W							
复位	1	1	1	1	1	1	1	1

F1A.7~0 **TM2RLD:** Timer2重载值

1Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2CTL	-	-	TM2STP	TM2CKS		TM2PSC		
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

1Bh.5 **TM2STP:** Timer2 启用
0: Timer2计数
1: Timer2 停止计数

1Bh.4~3 **TM2CKS:** Timer2 时钟源选择
00: 系统时钟除2(F_{SYS}/2)
01: 快时钟
10: 慢时钟
11: 快时钟乘2 (必需工作于3V以上)

1Bh.2~0 **TM2PSC:** Timer2 分频器选择
000: 1分频
001: 2分频
010: 4分频
011: 8分频
100: 16分频
101: 32分频
110: 64分频
111: 128分频

F1C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2D	PWM2D							
R/W	R/W							
复位	0	0	0	0	0	0	0	0

F1C7~0 **PWM2D:** PWM2占空比

F1D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM3D	PWM3D							
R/W	R/W							
复位	0	0	0	0	0	0	0	0

F1D.7~0 **PWM3D:** PWM3占空比

1Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM23CTL	PWM2POE	PWM2NOE	PWM2EN	-	PWM3POE	PWM3NOE	PWM3EN	-
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	-
Reset	0	0	0	-	0	0	0	-

1Eh.7 **PWM2POE:** PWM2P 输出
0: PWM2P输出禁止
1: PWM2P输出使能

1Eh.6 **PWM2NOE:** PWM2N 输出
0: PWM2N输出禁止
1: PWM2N输出使能

1Eh.5 **PWM2EN:** PWM2 使能
0: 禁止
1: 使能

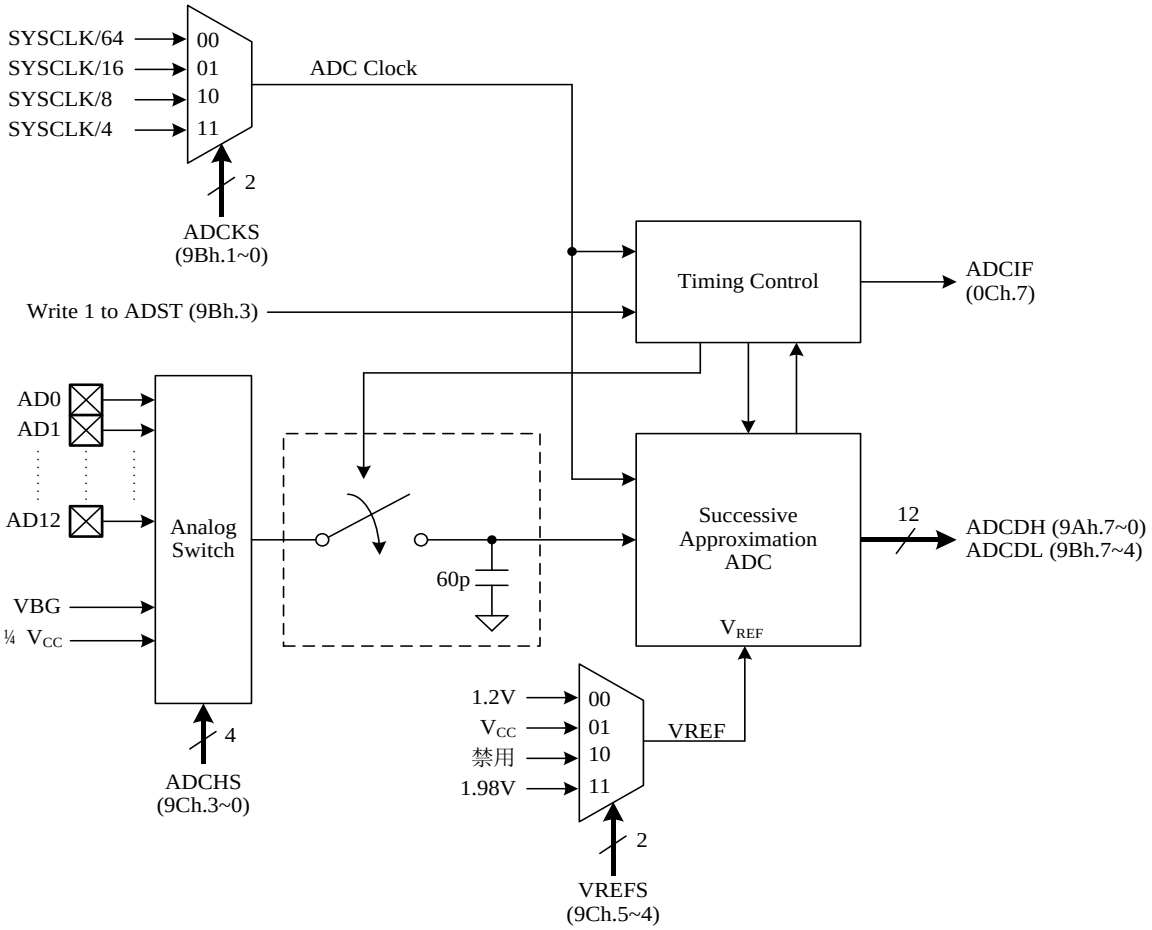
1Eh.3 **PWM3POE:** PWM3P 输出
0: PWM3P输出禁止
1: PWM3P输出使能

98h.2 **PWM3NOE:** PWM3N 输出
0: PWM3N输出禁止
1: PWM3N输出使能

98h.1 **PWM3EN:** PWM3 使能
0: 禁止
1: 使能

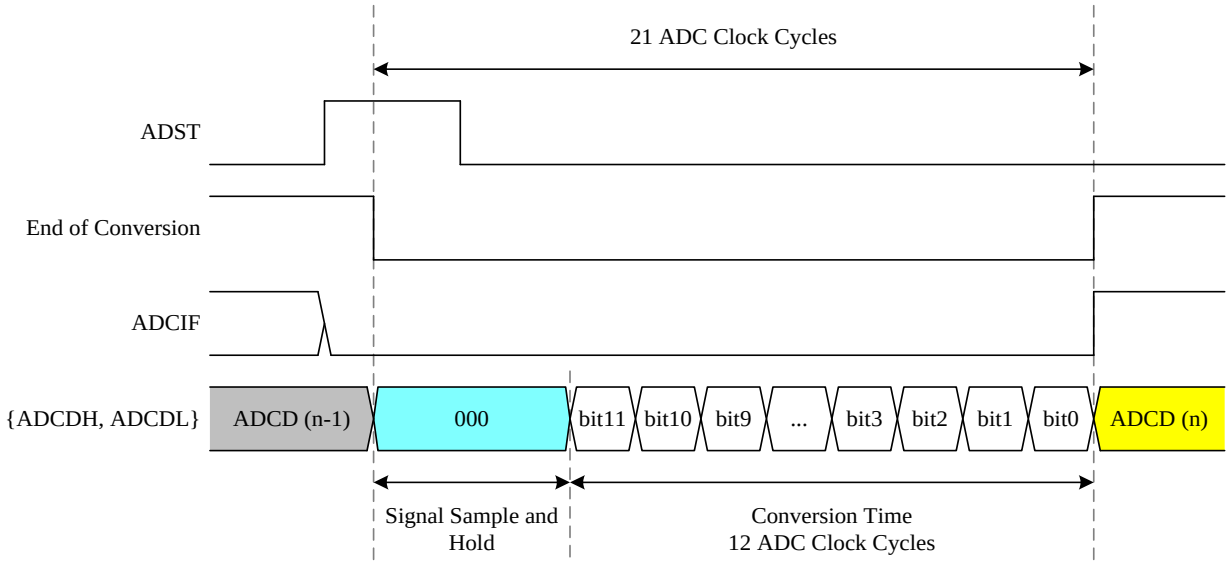
8. ADC

- 12 位逐次逼近式
- 13 外部脚位通道模拟输入, 2 内部电位输入(VBG, 1/4V_{CC})
- ADC 的参考电压(V_{REF})由 VREFS 选择： V_{CC}/3V/1.98V/1.2V。
- VBG 电位由 VREFS 选择, 当 VREFS 设置为 V_{CC}时, VBG 电位则为 1.2V, 其余选项 V_{REF}与 VBG 同电位。
- 模拟输入电平必须保持从 V_{SS}到参考电压 V_{REF}的范围内。
- ADCHS 选择到 VBG 时, VREFS 必需设置为 V_{CC}, 否则 ADC 转换无效, 且此时 VBG 电位则为 1.2V。
- ADC 转换时间，一共需要 25 个 ADC 时钟周期以执行完全转化。
- 设置 ADST 位启动 ADC 转换, ADST 将在 ADC 转换开始后自动清除



12 位 ADC,一共有 13 個外部脚位通道模拟输入 AD0~AD12, 2 内部电位输入 VBG 和 1/4V_{CC}。ADC 通道通过模拟开关多路复用器连接到模拟输入引脚。模拟开关多路复用器由 ADCHS 寄存器控制。选择 VREFS 时, 即为选择 VBG 内部基准电压。当 ADC 通道选择到 VBG 时，VBG 发生器将自动启用。

ADC 转换时间是 ADC 转换电压所需要的时间。该 ADC 转换每个位需要两个 ADC 时钟周期,以及多个时钟周期进行输入电压采样和保持。一共需要 21 个 ADC 时钟周期以执行完全转化。当转换时间结束,ADIF 中断标志由 H/W 设置,12 位 A/D 转换结果被加载到 ADCDH 和 ADCDL 寄存器。



0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	CMPIE	TM2IE	TM1IE	TM0IE	PCXIE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.7 **ADCIE:** ADC 中断使能
0: 关闭
1: 使能

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	CMPIF	TM2IF	TM1IF	TM0IF	PCXIF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.7 **ADCIF:** ADC 中断事件挂起标志
当 ADC 转换结束后由 H/W 置位，对此位写 0 将清除该标志

SFR **9Ah**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCDH	ADCDH							
R/W	R							
Reset	–	–	–	–	–	–	–	–

9Ah.7~0 **ADCDH:**ADC 数据位 11~4

SFR **9Bh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL	ADCDL				ADST	–	ADCKS	
R/W	R				R/W	–	R/W	R/W
Reset	–	–	–	–	0	–	0	0

9Bh.7~4 **ADCDL:** ADC 数据位 3~0

9Bh.3 **ADST:**启动 ADC 转换

设置 ADST 位启动 ADC 转换, ADST 将在 ADC 转换开始后自动清除。

9Bh.1~0 **ADCKS:**ADC 时钟频率选择

00:F_{SYSCLK}/64
01:F_{SYSCLK}/16
10:F_{SYSCLK}/8
11:F_{SYSCLK}/4

SFR **9Ch**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL2	–	–	VREFS		ADCHS			
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1

9Ch.5~4 **VREFS:** ADC 参考电压与 VBG 输出电压选择

ADCHS 选择到 VBG 时, VREFS 必需设置为 V_{CC}, 否则 ADC 转换无效

当 VREFS 设置为 V_{CC} 时, VBG 则固定为 1.2V

00: 1.2V
01: V_{CC}
10: 禁用
11: 1.98V

9Ch.5~0 **ADCHS:** ADC 通道选择

0000: AD00
0001: AD01
…
1101: Vss
1110: V_{BG} (内部基准电压源)
1111: 1/4V_{CC} (内部基准电压源)

9. I/O 端口

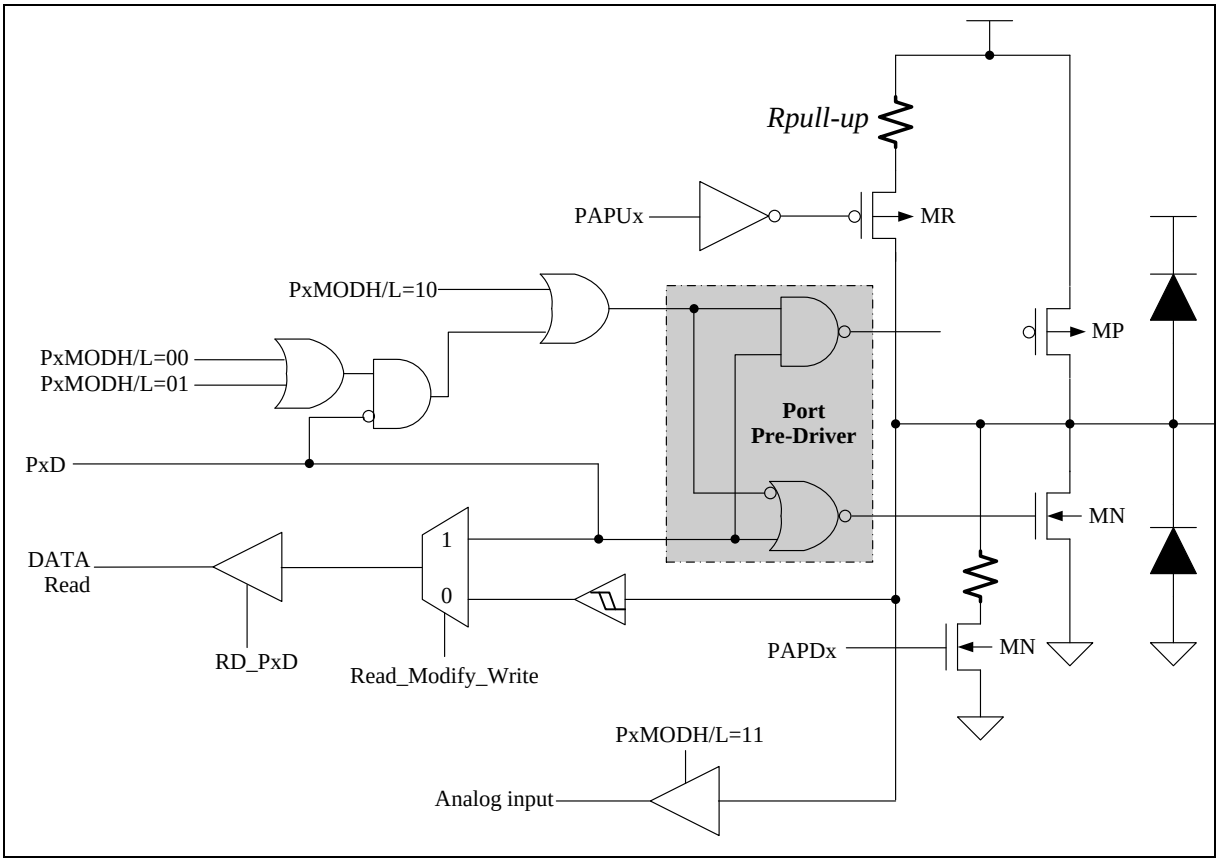
9.1 PA7-PA0, PB5-PB0

这些引脚可被用作施密特触发输入或 CMOS 推挽输出。S/W 需要设置 PAMODH/PAMODL。读引脚数据 (PAD) 具有不同的含义。在 "读-修改-写" 指令中， CPU 实际上读取输出数据寄存器。在别的指令中，CPU 读取引脚的状态。所谓的 "读-修改-写" 指令包括 BSF，BCF 以及所有把 F – Plane 做目的地的指令。PA7 共享刻录时的 VPP 引脚。

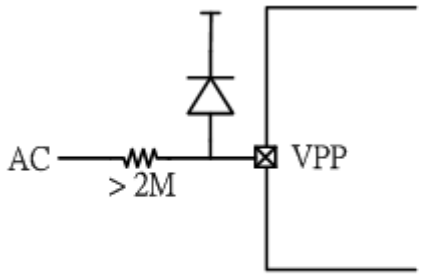
引脚设置 **PAMODH/L** **PBMODH/L**	**引脚功能**	**PAD/PBD SFR data**	**引脚状态**	**引脚改变唤醒**
00b	开漏输入	0	低驱动	Disable
		1	高组态	
01b		0	低驱动	Enable
		1	高组态	
10b	CMOS 输出	0	低驱动	Disable
		1	高驱动	
11b	模拟输入 ADC/CMP	1	高组态	Disable

这如果引脚用作施密特触发器输入，软件必须将 I/O 引脚设置为 00b 或 01b，并将相应的端口数据 SFR 设置为 1，以禁用该引脚的输出驱动电路。除了 I/O 端口功能外，每个引脚还具有一个或多个其他功能，例如 ADC。大多数功能通过将各个引脚的模式控制 SFR 设置为模式 11 来激活。

PA7-PA0, PB5-PB0 每只引脚有单独的上拉/下拉开关, 由 SFR PAPU/PAPD/PBPU/PBPD 分别控制。



PA Structure (except PA7)



Zero crossing detector circuit for VPP pin

F05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD	PAD							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

F05.7~0 **PAD:** PA data

F06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBD	PBD							
R/W	R/W							
Reset	-	-	1	1	1	1	1	1

F06.7~0 **PBD:** PB data

F11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODL	PA3MOD		PA2MOD		PA1MOD		PA0MOD	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

F11.7~6 PA3MOD/ PA2MOD/ PA1MOD/ PA0MOD
F11.5~4 00: 开漏
F11.3~2 01: 开漏，带引脚改变唤醒
F11.1~0 10: CMOS 输出
11: 模拟输入

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODH	PA7MOD		PA6MOD		PA5MOD		PA4MOD	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

F12.7~6 PA7MOD/ PA6MOD/ PA5MOD/ PA4MOD
F12.5~4 00: 开漏
F12.3~2 01: 开漏，带引脚改变唤醒
F12.1~0 10: CMOS 输出
11: 模拟输入

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBMODL	PB3MOD		PB2MOD		PB1MOD		PB0MOD	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

F13.7~6 PB3MOD/ PB2MOD/ PB1MOD/ PB0MOD
F13.5~4 00: 开漏
F13.3~2 01: 开漏，带引脚改变唤醒
F13.1~0 10: CMOS 输出
11: 模拟输入

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBMODH	-		-		PB5MOD		PB4MOD	
R/W	-		-		R/W		R/W	
Reset	-	-	-	-	0	0	0	0

F14.3~2 PB5MOD/ PB4MOD
F14.1~0
00: 开漏
01: 开漏，带引脚改变唤醒
10: CMOS 输出
11: 模拟输入

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAPU	PAPU							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F15.7~0 **PADU:** PA 上拉功能
0: 禁止
1: 使能

F16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAPD	PAPD							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F16.7~0 **PADU:** PA 下拉功能
0: 禁止
1: 使能

F17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBPU	-	-	PBPU					
R/W	-	-	R/W					
Reset	-	-	0	0	0	0	0	0

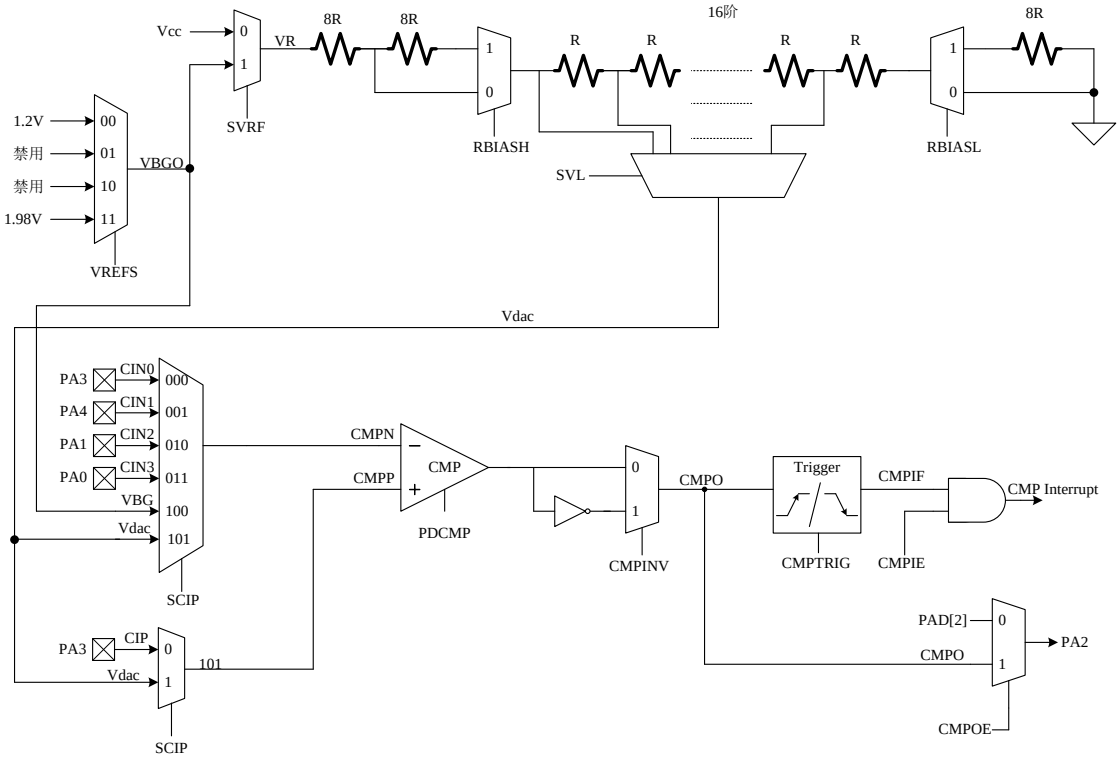
F17.5~0 **PBDU:** PB 上拉功能
0: 禁止
1: 使能

F18	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBPD	-	-	PBPD					
R/W	-	-	R/W					
Reset	-	-	0	0	0	0	0	0

F18.5~0 **PBDU:** PB 下拉功能
0: 禁止
1: 使能

10. 比较器与 DAC 与 LVD

- 该器件中有一个比较器 (CMP)与 DAC。
- 比较器正端可选择由 4 个外部脚位输入或 VBG 输入或由 DAC 产生电位输入。
- 比较器正端可由 PA3 脚位输入或由 DAC 产生电位输入。
- 比较器使用步骤, 应先设置 SFR CMPTRIG, CMPINV 后, 再清除中断旗标(CMPIF), 最后才致能 CMP(PDCMP=0) 。
- 可利用比较器两端一端选择 Vdac 输入, 一端选择 VBG 输入完成低电压 LVD 检测



比较器与 DAC 的功能框图

Vdac 电压可以下列公式计算近似

$$\text{Vdac} = \frac{(8 \times \text{RBIASL} + \text{SVL} + 1)}{(8 \times \text{RBIASH} + 8 \times \text{RBIASL} + 24)} \times VR$$

其中 VR 为 SVRF 所选择到的电位 V_{CC} 或 VBGO
RBIASL/RBIASH/SVL 为特殊寄存器 (SFR) 的值

◇ 比较器的设置方式应依下述流程设置:

先在比较器关闭的状态下(PDCMP=1), 设置 CMPCTL 选项, 然后清除 CMPIF, 最后才开启比较器(PDCMP=0)

```
BCF     RP1
BSF     RP0
MOVLW   10101000b       ;
MOVWF   CMPCTL          ; 在比较器关闭的状态下(PDCMP=1)
                        ; 设置CMPCTL选项
MOVLW   10111111h
MOVWF   INTIF           ; 清除CMPIF
BCF     PDCMP           ; 启用CMP
```

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	CMPIE	TM2IE	TM1IE	TM0IE	PCXIE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.6 **CMPIE:** 比较器中断使能
0: 关闭
1: 使能

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	CMPIF	TM2IF	TM1IF	TM0IF	PCXIF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.6 **CMPIF:** 比较器中断事件挂起标志
当 CMPO 匹配触发条件时由 H/W 置位，对此位写 0 将清除该标志

9Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPCTL	PDCMP	CMPO	CMPOE	CMPINV	CMPTRIG	-	-	-
R/W	R/W	R	R/W	R/W	R/W	-	-	-
Reset	1	1	0	0	0	-	-	-

9Dh.7 **PDCMP:** 比较器使能
0: 使能
1: 禁用

9Dh.6 **CMPO:** 比较器结果
0: 当CIP<CIN时 (如果CMPINV=0)
1: 当CIP>CIN时 (如果CMPINV=0) 或 PDCMP=1

9Dh.5 **CMPOE:** CMPO端口输出选择
0: 禁止CMPO输出至PA2
1: 使能CMPO输出至PA2

9Dh.4 **CMPINV:** CMPO反控制位
0: CMPO正向输出
1: CMPO反向输出

9Dh.3 **CMPTRIG:** CMPO中断触发选择
0: CMPO上升缘触发中断
1: CMPO下降缘触发中断

9Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPPNS	-	-	-	-	SCIN			SCIP
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset	-	-	-	-	0	0	0	**0**

9Eh.3~1 **SCIN:** 比较器负端输入(CIN)选择
000: CIN0(PA3) 连接至比较器负端
001: CIN1(PA4) 连接至比较器负端
010: CIN2(PA1) 连接至比较器负端
011: CIN3(PA0) 连接至比较器负端
100: VBGO 连接至比较器负端 (由VREFS选择VBGO电位)
101: Vdac连接至比较器负端
110: 保留
111: 保留

9Eh.0 **SCIP:** 比较器正端输入(CIP)选择
0: CIP(PA3) 连接至比较器正端
1: Vdac连接至比较器负端

9Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DACTL	SVRF	-	RBIASH	RBIASL	SVL			
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	**0**

9Fh.7 **SVRF:** DAC输入电源选择
0: V_{CC}
1: VBGO (由VREFS选择VBG电位)

9Fh.5 **RBIASH:** 具体用法参考比较器框图与Vdac 计算公式
可选择0~1

9Fh.4 **RBIASL:** 具体用法参考比较器框图与Vdac 计算公式
可选择0~1

9Fh.3~0 **SVL:** 内部电阻分压比选择位, 具体用法参考比较器框图与Vdac 计算公式
可选择0~15

指令集

每个指令都是一个 14 位字被分开成一个 Op Code，他详细的阐述了指令类型，一个或更多的能进一步详解指令运行的操作数。指令在下面的表格中被分类为字节导向，位向导和文字操作列表。

对字节指令来说，“f”或“r”代表指示地址，“d”代表指示目的单元格。地址指示被用来指定程序存储器的哪位地址被指令使用。目的单元格指示出操作结果被放置的位置。如果“d”是 0，结果被放在 W 寄存器，否则结果被放在指令指示的地址中。

对位指示指令来说，“b”代表位指示，他用来选择受操作影响的位，而“f”代表地址指示。对于文字操作，“k”代表文字或常数。

当 F-plan 内的缓存器为 write only 时, 禁止使用 BCF 与 BSF 指令, 应对 write only 的相关缓存器(如 INTE 与 EEPCTL)直接赋值。

简记符号	描述
f	F-Plane寄存器
r	R-Plane寄存器
b	位地址
k	文字，常数或标签
d	目的选择项。0 : 工作寄存器　1 : 寄存器档案
W	工作寄存器
Z	零标志
C	进位标志
DC	十进制进位标志
PC	程序计数器
TOS	堆叠顶
GIE	全局中断使能标志（i-Flag）
[]	可选项
( )	内容
.	比特项
B	之前
A	之后
←	赋值方向

内存功能图

F-Plane

名称	地址	R/W	复位值	描述
(F00) INDF				**RAM W/R 相关功能**
INDF	00.7~0	R/W	-	非物理寄存器，INDF 寻址实际上指向 FSR 寄存器内的地址的寄存器
(F02) PCL				**Programming Counter 相关功能**
PCL	02.7~0	R/W	0	Programming Counter LSB[7~0]
(F03) STATUS				**Status 相关功能**
IRP	03.7	R/W	0	寄存器 Bank 选择位 (用于间接寻址)
RP1	03.6	R/W	0	寄存器 Bank 选择位 1 用于直接寻址
RP0	03.5	R/W	0	寄存器 Bank 选择位 0 用于直接寻址
TO	03.4	R	0	WDT 超时标志，通过 PWRST，‘SLEEP’ 或 ‘CLRWDT’ 指令清除
PD	03.3	R	0	掉电标志，通过指令 ‘SLEEP’ 设置，通过指令 ‘CLRWDT’ 清除
Z	03.2	R/W	0	零标志
DC	03.1	R/W	0	十进制进位标志
C	03.0	R/W	0	进位标志
(F04) FSR				**RAM W/R/读表相关功能**
FSR	04.7~0	R/W	0	档案选择寄存器，间接寻址模式指标
(F05) PAD				**Port A 相关功能**
PAD	05.7~0	R	-	Port A 引脚或“数据寄存器”状态
		W	F	Port A 输出数据寄存器
(F06) PBD				**Port B 相关功能**
PBD	06.5~0	R	-	Port B 引脚或“数据寄存器”状态
		W	F	Port B 输出数据寄存器
(F0A) PCLATH				**Function related to: PROGRAM COUNT**
PCLATH	0A.2~0	R/W	0	当执行以 PCL 为目标的指令时编程计数器高字节数据

名称	地址	R/W	复位值	描述
(F0B) INTE				**中断使能相关功能**
ADCIE	0B.7	R/W	0	ADC 中断使能 0: 禁止 1: 使能
CMPIE	0B.6	R/W	0	比较器中断使能 0: 禁止 1: 使能
TM2IE	0B.5	R/W	0	Timer2 中断使能 0: 禁止 1: 使能
TM1IE	0B.4	R/W -	0	Timer1 中断使能 0: 禁止 1: 使能
TM0IE	0B.3	R/W	0	Timer0 中断使能 0: 禁止 1: 使能
PCXIE	0B.2	R/W	0	引脚电平变化中断使能 0: 禁止 1: 使能
INT1IE	0B.1	R/W	0	INT0 中断使能 0: 禁止 1: 使能
INT0IE	0B.0	R/W	0	INT0 中断使能 0: 禁止 1: 使能
(F0C) INTF				**中断使能相关功能**
ADCIF	0C.7	R/W	0	当 ADC 转换结束后由 H/W 置位，对此位写 0 将清除该标志
CMPIF	0C.6	R/W	0	当 CMPO 匹配触发条件时由 H/W 置位，对此位写 0 将清除该标志
TM2IF	0C.5	R/W	0	当 Timer2 溢出时由 H/W 置位，对此位写 0 将清除该标志
TM1IF	0C.4	R/W -	0	当 Timer1 溢出时由 H/W 置位，对此位写 0 将清除该标志
TM0IF	0C.3	R/W	0	当 Timer0 溢出时由 H/W 置位，对此位写 0 将清除该标志
PCXIF	0C.2	R/W	0	当引脚电平变化由 H/W 置位，对此位写 0 将清除该标志
INT1IF	0C.1	R/W	0	当 INT1 引脚发生下降/上升沿时由 H/W 置位，对此位写 0 将清除该标志
INT0IF	0C.0	R/W	0	当 INT0 引脚发生下降/上升沿时由 H/W 置位，对此位写 0 将清除该标志
(F0D) DPL				**读表相关功能**
DPL	0D.7~0	R/W	–	读表位地址，数据 ROM 指针 (DPTR) 低位
(F0E) DPH				**读表相关功能**
DPH	0E.2~0	R/W	–	读表高地址，数据 ROM 指针 (DPTR)高位

名称	地址	R/W	复位值	描述
(F0F) CLKCTL				**CPUCLK 相关功能**
SCKTYPE	0F.7	R/W	0	慢时钟选择 0: 慢时钟为 SIRC 1: 慢时钟为 SXT
SLOWSTP	0F.4	R/W	0	在 SLEEP 指令后停止慢时钟 0: 慢时钟在 SLEEP 指令后持续运行 1: 慢时钟在 SLEEP 指令后停止运行
FASTSTP	0F.3	R/W	1	停止快时钟 0: 快时钟运行 1: 快时钟停止
CPUCKS	0F.2	R/W	0	主时钟源选择 0: 慢时钟 1: 快时钟
FASTPSC	0F.1~0	R/W	11	快钟源预分频器,只能在主时钟为慢时钟时才能切换 00: 除以 8 01: 除以 4 10: 除以 2 11: 除以 1
(F10) OPTION				
INT0EDG	10.6	R/W	0	INT0 中断沿选择 0: INT0 负沿中断 1: INT0 正沿中断
INT1EDG	10.5	R/W	0	INT1 中断沿选择 0: INT1 负沿中断 1: INT1 正沿中断
WDTE	10.4	R/W	1	停止快时钟 0: WDT 禁止 1: WDT 启用 (WDTC/WDTE 必需同时为 1 才会启用 WDT)
PORPD	10.3	R/W	0	1: 关闭 POR
LVRSAV	10.2	R/W	1	设置为 1 时, 执行 SLEEP 指令后或系统时钟为慢时钟时自动关闭 LVR
PORSAV	10.1	R/W	0	设置为 1 时, 执行 SLEEP 指令后自动关闭 POR
LVRO	10.0	R	0	LVR 产生时为 1
(F11) PAMODL				**Port A 相关功能**
PA3MOD	11.7~6	R/W	00	PA3MOD/ PA2MOD/ PA1MOD/ PA0MOD 00: 开漏 01: 开漏，带引脚改变唤醒 10: CMOS 输出 11: 模拟输入
PA2MOD	11.5~4	R/W	00	
PA1MOD	11.3~2	R/W	00	
PA0MOD	11.1~0	R/W	00	
(F12) PAMODH				**Port A 相关功能**
PA7MOD	12.7~6	R/W	00	PA7MOD/ PA6MOD/ PA5MOD/ PA4MOD 00: 开漏 01: 开漏，带引脚改变唤醒 10: CMOS 输出 11: 模拟输入
PA6MOD	12.5~4	R/W	00	
PA5MOD	12.3~2	R/W	00	
PA4MOD	12.1~0	R/W	00	

名称	地址	R/W	复位值	描述
(F13) PBMODL				**Port B 相关功能**
PB3MOD	13.7~6	R/W	00	PB3MOD/ PB2MOD/ PB1MOD/ PB0MOD 00: 开漏 01: 开漏，带引脚改变唤醒 10: CMOS 输出 11: 模拟输入
PB2MOD	13.5~4	R/W	00	
PB1MOD	13.3~2	R/W	00	
PB0MOD	13.1~0	R/W	00	
(F14) PBMODH				**Port B 相关功能**
PB5MOD	14.3~2	R/W	00	PB5MOD/ PB4MOD 00: 开漏 01: 开漏，带引脚改变唤醒 10: CMOS 输出 11: 模拟输入
PB4MOD	14.1~0	R/W	00	
(F15) PAPU				**Port A 相关功能**
PAPU	15.7~0	R/W	00	PA7~0上拉选择 0: 关闭上拉 1: 开启上拉
(F16) PAPD				**Port A 相关功能**
PAPD	16.7~0	R/W	00	PA7~0下拉选择 0: 关闭下拉 1: 开启下拉
(F17) PBPU				**Port B 相关功能**
PBPU	17.5~0	R/W	00	PB5~0上拉选择 0: 关闭上拉 1: 开启上拉
(F18) PBPD				**Port B 相关功能**
PBPD	18.5~0	R/W	00	PB5~0下拉选择 0: 关闭下拉 1: 开启下拉
(F19) TM2				**Timer2 相关功能**
TM2	19.7~0	R/W	ff	Timer2 计数器, 可擦写的递减计数器
(F1A) TM2				**Timer2 相关功能**
TM2RLD	1A.7~0	R/W	ff	Timer2 重新加载计数器

名称	地址	R/W	复位值	描述
(F1B) TM2CTL				**TM2 相关功能**
TM2STP	1B.5	R/W	0	Timer2 计数器停止 0: Timer2 计数 1: Timer2 停止计数
TM2CKS	1B.4~3	R/W	0	Timer2 时钟源选择 00: 系统时钟除2(F_{SYS}/2) 01: 快时钟 10: 慢时钟 11: 快时钟乘2 (必需工作于3V以上)
TM2PSC	1B.2~0	R/W	0	Timer2 预分频 000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频
(F1C) PWM2D				**PWM 相关功能**
PWM2D	1C.7~0	R/W	-	PWM2 占空比设置
(F1D) PWM3D				**PWM 相关功能**
PWM3D	1D.7~0	R/W	-	PWM3 占空比设置
(F1E) PWM23CTL				**TM2 相关功能**
PWM2POE	1E.7	R/W	0	PWM2P 输出 0: PWM2P 输出禁止 1: PWM2P 输出使能
PWM2NOE	1E.6	R/W	0	PWM2N 输出 0: PWM2N输出禁止 1: PWM2N输出使能
PWM2EN	1E.5	R/W	0	PWM2 使能 0: 禁止 1: 使能
PWM3POE	1E.3	R/W	0	PWM3P 输出 0: PWM3P 输出禁止 1: PWM3P 输出使能
PWM3NOE	1E.2	R/W	0	PWM3N 输出 0: PWM3N输出禁止 1: PWM3N输出使能
PWM3EN	1E.1	R/W	0	PWM3 使能 0: 禁止 1: 使能
User Data Memory				
RAM	20~6F	R/W	-	RAM Bank0 区域(80 字节)
RAM	70~7F	R/W	-	RAM 公共区域(16 字节)
(F90) TM0				**Timer0 相关功能**
TM0	90.7~0	R/W	-	Timer0 计数器, 可擦写的递增计数器
(F91) TM0CTL				**TM0 相关功能**

名称	地址	R/W	复位值	描述
TM0STP	91.6	R/W	0	Timer0 计数器停止 0: Timer0 计数 1: Timer0 停止计数
TM0PSA	91.5	R/W	0	Timer0分频器选择 0: WDT使用分频器 1: Timer0 使用分频器
TM0CKS	91.4~3	R/W	0	Timer0 时钟源选择 00: 系统时钟除2(F_{SYS}/2) 01: 快时钟 10: 慢时钟 11: 快时钟乘2 (必需工作于3V以上)
TM0PSC	91.2~0	R/W	0	Timer0 预分频 000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频
(F92) TM1				**Timer1 相关功能**
TM1	92.7~0	R/W	ff	Timer1 计数器, 可擦写的递减计数器
(F93) TM1RLD				**Timer1 相关功能**
TM1RLD	93.7~0	R/W	ff	Timer1 重新加载计数器
(F94) TM1CTL				**TM1 相关功能**
TM1STP	94.5	R/W	0	Timer1 计数器停止 0: Timer1 计数中 1: Timer1 停止计数
TM1CKS	94.4~3	R/W	0	Timer1 时钟源选择 00: 系统时钟除2(F_{SYS}/2) 01: 快时钟 10: 慢时钟 11: 快时钟乘2 (必需工作于3V以上)
TM1PSC	94.2~0	R/W	0	Timer1 预分频 000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频
(F95) PWM0MDZ				**PWM 相关功能**
PWM0PINV	95.5	R/W	0	PWM0P输出选择 0: 正向输出 1: 反向输出
PWM0NINV	95.4	R/W	0	PWM0N输出选择 0: 正向输出

名称	地址	R/W	复位值	描述
				1: 反向输出
PWM0DZ	95.3~0	R/W	0	PWM0 死区控制 0000: 死区禁用 0001: 死区宽度 1*Tpwmclk 0010: 死区宽度 2*Tpwmclk … 1111: 死区宽度 15*Tpwmclk
(F96) PWM0D				**PWM 相关功能**
PWM0D	96.7~0	R/W	-	PWM0 占空比设置
(F97) PWM1D				**PWM 相关功能**
PWM1D	97.7~0	R/W	-	PWM1 占空比设置
(F98) PWM01CTL				**PWM 相关功能**
PWM0POE	98.7	R/W	0	PWM0P 输出 0: PWM0P 输出禁止 0: PWM0P 输出禁止
PWM0NOE	98.6	R/W	0	PWM0N 输出 0: PWM0N输出禁止 1: PWM0N输出使能
PWM0EN	98.5	R/W	0	PWM0 使能 0: 禁止 1: 使能
PWM1POE	98.3	R/W	0	PWM1P 输出 0: PWM1P 输出禁止 0: PWM1P 输出禁止
PWM1NOE	98.2	R/W	0	PWM1N 输出 0: PWM1N输出禁止 1: PWM1N输出使能
PWM1EN	98.1	R/W	0	PWM1使能 0: 禁止 1: 使能
(F99) T1PWMH				**PWM 相关功能**
T1PWMH	99.1~0	R/W	0	TM1/TM1RLD/PWM0D/PWM1D 高 2 位(bit 9~8)访问 先写入 2 位高字节，再写 8 位低字节 先读取 8 位低字节，再读取 2 位高字节 T1PWMH
(F9A) ADCDH				**ADC 相关功能**
ADCDH	9A.7~0	R	-	ADC输出数据位 11~4
(F9B) ADCTL				**ADC 相关功能**
ADCDL	9B.7~4	R	-	ADC输出数据位 3~0
ADST	9B.3	R/W	0	ADC起始位 1: 设置ADST位启动ADC转换, ADST将在ADC转换开始后自动清除
ADCKS	9B.1~0	R/W	0	ADC 时钟频率选择 00:F_{SYSCLK}/64 01:F_{SYSCLK}/16 10:F_{SYSCLK}/8 11:F_{SYSCLK}/4

名称	地址	R/W	复位值	描述
(F9C) ADCTL2				**ADC 相关功能**
VREFS	9C.5~4	R	-	ADC 参考电压与VBG输出电压选择 ADCHS 选择到 VBG 时, VREFS 必需设置为 V_{CC}, 否则 ADC 转换无效 当 VREFS 设置为 V_{CC} 时, VBG 则固定为 1.2V 00: 1.2V 01: V_{CC} 10: 3V 11: 1.98V
ADCHS	9C.3~0	R/W	0	ADC 通道选择 0000: AD00 0001: AD01 … 1101: Vss 1110: V_{BG} (内部基准电压源) 1111: 1/4V_{CC} (内部基准电压源)
(F9D) CMPCTL				**CMP 相关功能**
PDCMP	9D.7	R/W	1	比较器使能 0: 使能 1: 禁用
CMPO	9D.6	R	1	比较器结果 0: 当 CIP<CIN 时 (如果 CMPINV=0) 1: 当 CIP>CIN 时 (如果 CMPINV=0) 或 PDCMP=1
CMPOE	9D.5	R/W	0	CMPO 端口输出选择 0: 禁止 CMPO 输出至 PA2 1: 使能 CMPO 输出至 PA2
CMPINV	9D.4	R/W	0	CMPO 反控制位 0: CMPO 正向输出 1: CMPO 反向输出
CMPTRIG	9D.3	R/W	0	CMPO 中断触发选择 0: CMPO 上升缘触发中断 1: CMPO 下降缘触发中断
(F9E) CMPPNS				**CMP 相关功能**
SCIN	9E.3~1	R/W	111	比较器负端输入(CIN)选择 000: CIN0(PA3) 连接至比较器负端 001: CIN1(PA4) 连接至比较器负端 010: CIN2(PA1) 连接至比较器负端 011: CIN3(PA0) 连接至比较器负端 100: VBGO 连接至比较器负端 (由VREFS选择VBGO电位) 101: Vdac连接至比较器负端 110: 保留 111: 保留
SCIP	9E.0	R/W	1	比较器正端输入(CIP)选择 0: CIP(PA3) 连接至比较器正端 1: Vdac 连接至比较器负端

名称	地址	R/W	复位值	描述
(F9F) DACTL				**CMP 相关功能**
SVRF	9F.7	R/W	0	DAC 输入电源选择 0: V_{CC} 1: VBGO (由VREFS选择VBG电位)
RBIASH	9F.5	R/W	0	具体用法参考比较器框图与 Vdac 计算公式 可选择 0~1
RBIASL	9F.4	R/W	0	具体用法参考比较器框图与 Vdac 计算公式 可选择 0~1
SVL	9F.3~0	R/W	0	内部电阻分压比选择位, 具体用法参考比较器框图与 Vdac 计算公式 可选择 0~15
User Data Memory				
RAM	A0~BF	R/W	-	RAM Bank1 区域(32 字节)

指令集

每个指令都是一个 14 位字被分开成一个 Op Code，他详细的阐述了指令类型，一个或更多的能进一步详解指令运行的操作数。指令在下面的表格中被分类为字节导向，位向导和文字操作列表。

对字节指令来说，“f”或“r”代表指示地址，“d”代表指示目的单元格。地址指示被用来指定程序存储器的哪位地址被指令使用。目的单元格指示出操作结果被放置的位置。如果“d”是 0，结果被放在 W 寄存器，否则结果被放在指令指示的地址中。

对位指示指令来说，“b”代表位指示，他用来选择受操作影响的位，而“f”代表地址指示。对于文字操作，“k”代表文字或常数。

当 F-plan 内的缓存器为 write only 时, 禁止使用 BCF 与 BSF 指令, 应对 write only 的相关缓存器(如 INTE 与 EEPCTL)直接赋值。

简记符号	描述
f	F-Plane寄存器
r	R-Plane寄存器
b	位地址
k	文字，常数或标签
d	目的选择项。0 : 工作寄存器　1 : 寄存器档案
W	工作寄存器
Z	零标志
C	进位标志
DC	十进制进位标志
PC	程序计数器
TOS	堆叠顶
GIE	全局中断使能标志（i-Flag）
[]	可选项
( )	内容
.	比特项
B	之前
A	之后
←	赋值方向

助记符		操作码	指令周期	影响标志	说明
字节操作指令					
ADDWF	f, d	00 0111 dfff ffff	1	C, DC, Z	W 和 “f” 相加 → W 或 "f"
ANDWF	f, d	00 0101 dfff ffff	1	Z	W 和 “f” 相与 → W 或 "f"
CLRF	F	00 0001 1fff ffff	1	Z	"f"清零
CLRW		00 0001 0100 0000	1	Z	W 清零
COMF	f, d	00 1001 dfff ffff	1	Z	"f" 取反 → W 或 "f"
DECF	f, d	00 0011 dfff ffff	1	Z	"f" 减一 → W 或 "f"
DECFSZ	f, d	00 1011 dfff ffff	1 or 2	-	“f” 减一 → W 或 "f", 结果为 0 就间跳
INCF	f, d	00 1010 dfff ffff	1	Z	"f" 加一 → W 或 "f"
INCFSZ	f, d	00 1111 dfff ffff	1 or 2	-	"f" 加一 → W 或 "f", 结果为 0 就间跳
IORWF	f, d	00 0100 dfff ffff	1	Z	W 和 “f” 相或 → W 或 "f"
MOVF	f,d	00 1000 dfff ffff	1	Z	"f"→W 或 "f"
MOVFW	f	00 1000 0fff ffff	1	Z	W ← "f"
MOVWF	f	00 0000 1fff ffff	1	-	"f" ← W
RLF	f, d	00 1101 dfff ffff	1	C	“f”带进位位左移 → W 或 "f"
RRF	f, d	00 1100 dfff ffff	1	C	"f" 带进位位右移 → W 或 "f"
SUBWF	f, d	00 0010 dfff ffff	1	C, DC, Z	"f" – W → W 或 "f"
SWAPF	f, d	00 1110 dfff ffff	1	-	"f"的高低半字节互换 → W 或 "f"
TSTF	f	00 1000 1fff ffff	1	Z	测试"f" 是否为 0, 影响 Z 标志
XORWF	f, d	00 0110 dfff ffff	1	Z	W 和 "f" 相异或 → W 或 "f"
位操作指令					
BCF	f, b	11 00bb bfff ffff	1	-	"f"的"b"位清零
BSF	f, b	11 01bb bfff ffff	1	-	"f"的"b"位置 1
BTFSC	f, b	11 10bb bfff ffff	1 or 2	-	"f"的"b"位为 0 间跳
BTFSS	f, b	11 11bb bfff ffff	1 or 2	-	"f"的"b"位为 1 间跳
立即数操作指令和控制指令					
ADDLW	k	01 1100 kkkk kkkk	1	C, DC, Z	立即数 k + W → W
ANDLW	k	01 1011 kkkk kkkk	1	Z	立即数 k 和 W 相与 → W
CALL	k	10 0kkk kkkk kkkk	2	-	调用子程序，子程序入口地址(标号)为 k
CLRWDT		01 1110 0000 0100	1	TO, PD	看门狗清零
GOTO	k	10 1kkk kkkk kkkk	2	-	跳转指令，目的地址(标号)为 k
IORLW	k	01 1010 kkkk kkkk	1	Z	立即数 k 和 W 相或 → W
MOVLW	k	01 1001 kkkk kkkK	1	-	立即数 k → W
NOP		00 0000 0000 0000	1	-	空指令
RET		00 0000 0100 0000	2	-	从子程序返回
RETI		00 0000 0110 0000	2	-	从中断返回
RETLW	k	01 1000 kkkk kkkK	2	-	带立即数 k 返回，返回值在 W 中
SLEEP		01 1110 0000 0011	1	TO, PD	进入待机模式，时钟振荡停止
SUBLW	k	01 1111 kkkk kkkk	1	C, DC, Z	立即数 k – W → W
TABRH		00 0000 0101 1000	2	-	查找 ROM 高位数据到 W
TABRL		00 0000 0101 0000	2	-	查找 ROM 低位数据到 W
XORLW	k	01 1101 kkkk kkkk	1	Z	立即数 k 和 W 相异或 → W

ADDLW "k" 与 W相加

句法	ADDLW k
操作数	k : 00h ~ FFh
运作	(W) ← (W) + k
影响的状态位	C, DC, Z
OP-Code	01 1100 kkkk kkkk
描述	W 寄存器中的内容和 8 位“K”值相加，结果放在 W 寄存器中。
周期	1
举例	ADDLW 0x15　　B : W =0x10 A : W =0x25

ADDWF W 和 "f"相加

句法	ADDWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运作	(目标寄存器) ← (W) + (f)
影响的状态位	C, DC, Z
OP-Code	01 0111 dfff ffff
描述	W 寄存器的内容和”f’相加。结果放在目标寄存器中。如"d"为 0,结果放 W 中,如"d"为 1,结果放 F 中.
周期	1
举例	ADDWF FSR, 0　　B : W =0x17, FSR =0xC2 A : W =0xD9, FSR =0xC2

ANDLW "k" 和 W 相与

句法	ANDLW k
操作数	k : 00h ~ FFh
运作	(W) ← (W) AND k
影响的状态位	Z
OP-Code	01 1011 kkkk kkkk
描述	W 寄存器的内容与 8 位立即数 k 进行 AND 运算。 结果存放在 W 寄存器中。
周期	1
举例	ANDLW 0x5F　　B : W =0xA3 A : W =0x03

ANDWF W 和"f"相与

句法	ANDWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运作	(目标寄存器) ← (W) AND (f)
影响的状态位	Z
OP-Code	00 0101 dfff ffff
描述	将 W 寄存器与寄存器'f'相与。 如果'd'为 0，结果存储在 W 寄存器中。 如果'd'为 1，结果存储在寄存器'f'中。
周期	1
举例	ANDWF FSR, 1　　B : W =0x17, FSR =0xC2 A : W =0x17, FSR =0x02

BCF "f"的"b"位清零

句法	BCF f [,b]
操作数	f : 00h ~ 7Fh, b : 0 ~ 7
运作	(f.b) ← 0
影响的状态位	-
OP-Code	11 00bb bfff ffff
描述	寄存器 'f' 的 'b'位被清零。
周期	1
举例	BCF FLAG_REG, 7　　B : FLAG_REG =0xC7 A : FLAG_REG =0x47

BSF "f"的"b"位置 1

句法	BSF f [,b]
操作数	f : 00h ~ 7Fh, b : 0 ~ 7
运作	(f.b) ← 1
影响的状态位	-
OP-Code	11 01bb bfff ffff
描述	寄存器 'f' 的 'b'位被置 1。
周期	1
举例	BSF FLAG_REG, 7　　B : FLAG_REG =0x0A A : FLAG_REG =0x8A

BTFSC "f"的"b"位若为 0 则跳转

句法	BTFSC f [,b]
操作数	f : 00h ~ 7Fh, b : 0 ~ 7
运作	如果（f.b）= 0，跳过下一条指令
影响的状态位	-
OP-Code	11 10bb bfff ffff
描述	若寄存器 f 的位 b 是 1，下一条指令被执行。否则下一条指令被放弃，一个 NOP 指令被执行，使这条指令要两个指令周期。
周期	1 or 2
举例	LABEL1 BTFSC FLAG, 1　　B : PC =LABEL1 TRUE GOTO SUB1　　A : if FLAG.1 =0, PC =FALSE FALSE ...　　if FLAG.1 =1, PC =TRUE

BTFSS "f"的"b"位若为 1 则跳转

句法	BTFSS f [,b]
操作数	f : 00h ~ 7Fh, b : 0 ~ 7
运作	如果（f.b）= 1，跳过下一条指令
影响的状态位	-
OP-Code	11 11bb bfff ffff
描述	若寄存器 F 的 b 位是 0，下一条指令被执行。否则下一条指令被放弃，一个 NOP 指令被执行，使这条指令要两个指令周期。
周期	1 or 2
举例	LABEL1 BTFSS FLAG, 1　　B : PC =LABEL1 TRUE GOTO SUB1　　A : if FLAG.1 =0, PC =TRUE FALSE ...　　if FLAG.1 =1, PC =FALSE

CALL 调用子程序

句法	CALL k	
操作数	k : 000h ~ 7FFh	
运作	运算: TOS ← (PC) + 1, PC.10~0 ← k	
影响的状态位	-	
OP-Code	10 0kkk kkkk kkkk	
描述	调用子程序。 首先，返回地址（PC + 1）被推入堆栈。 11 位立即数地址装入 PC 位<10：0>，PCLATH 装入 PC 的高位。 CALL 是一个双周期指令。	
周期	2	
举例	LABEL1 CALL SUB1	B : PC =LABEL1 A : PC =SUB1, TOS =LABEL1 + 1

CLRF 清除 "f"

句法	CLRF f	
操作数	f : 00h ~ 7Fh	
运作	(f) ← 00h, Z ← 1	
影响的状态位	Z	
OP-Code	00 0001 1fff ffff	
描述	寄存器 f 的内容被清零，Z 位置 1。	
周期	1	
举例	CLRF FLAG_REG	B : FLAG_REG =0x5A A : FLAG_REG =0x00, Z =1

CLRW 清除 W

句法	CLRW	
操作数	-	
运作	(W) ← 00h, Z ← 1	
影响的状态位	Z	
OP-Code	00 0001 0100 0000	
描述	W 寄存器清零，Z 位置 1。	
周期	1	
举例	CLRW	B : W =0x5A A : W =0x00, Z =1

CLRWDT 清除看门狗定时器

句法	CLRWDT	
操作数	-	
运作	WDT/WKT Timer ← 00h	
影响的状态位	TO, PD	
OP-Code	01 1110 0000 0100	
描述	CLRWDT 指令清零看门狗/唤醒定时器	
周期	1	
举例	CLRWDT	B : WDT counter =? A : WDT counter =0x00

COMF "f"取反

句法	COMF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运作	(目标寄存器) ← ($\bar{f}$)
影响的状态位	Z
OP-Code	00 1001 dfff ffff
描述	寄存器'f'的值做补码运算,运算结果放到目标寄存器中, 如果'd'为 0，结果存储在 W 中。如果'd'为 1，则结果存储在寄存器 f 中。
周期	1
举例	COMF REG1, 0　　B : REG1 =0x13 A : REG1 =0x13, W =0xEC

DECF "f"递减 1

句法	DECF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运作	(destination) ← (f) - 1
影响的状态位	Z
OP-Code	00 0011 dfff ffff
描述	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
周期	1
举例	DECF CNT, 1　　B : CNT =0x01, Z =0 A : CNT =0x00, Z =1

DECFSZ F 递减 1，结果为 0 则跳转

句法	DECFSZ f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(目标寄存器) ← (f) - 1, 若结果为 0，则跳过下一条指令。
影响的状态位	-
OP-Code	00 1011 dfff ffff
描述	寄存器 f 的值递减 1。若‘d’为 0，结果放在 W 寄存器中。若‘d’为 1，结果放回寄存器‘f’。若结果为 1，下一条指令就会继续执行。若结果是 0，一个 NOP 指令取代执行，此时会需要两个指令周期的时间。
周期	1 或 2
举例	LABEL1 DECFSZ CNT, 1　　B : PC = LABEL1 GOTO LOOP　　A : CNT = CNT – 1 CONTINUE　　if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1

GOTO 强行跳转

句法	GOTO k
操作数	k : 00h ~ 7FFh
运行	PC.10~0 ← k
影响的状态位	-
OP-Code	10 1kkk kkkk kkkk
描述	GOTO 是无条件转移指令。11 个位的 k 值会立即被放在 PC<10:0>的位置中。GOTO 指令占用两个指令周期。
周期	2
举例	LABEL1 GOTO SUB1　　B : PC = LABEL1

A : PC = SUB1

INCF　F 递增 1

句法	INCF f [,d]
操作数	f : 00h ~ 7Fh
运行	(目标寄存器) ← (f) + 1
影响的状态位	Z
OP-Code	00 1010 dfff ffff
描述	寄存器 f 的值递加 1。如"d"为 0,结果放 W 中,如"d"为 1,结果放 F 中.。
周期	1
举例	INCF CNT, 1　　B : CNT = 0xFF, Z = 0 A : CNT = 0x00, Z = 1

INCFSZ　F 递增 1，结果为 0 则跳转

句法	INCFSZ f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(目标寄存器) ← (f) + 1, 若结果为 0，则跳过下一个指令
影响的状态位	-
OP-Code	00 1111 dfff ffff
描述	增量寄存器 f。若‘d’为 0，运算结果放在 W 寄存器。若‘d’为 1，将结果放回寄存器‘f’。若结果是 1，下一条指令被执行。若结果是 0，一个 NOP 指令被执行，使这条指令变成一个二周期指令。
周期	1 或 2
举例	LABEL1　INCFSZ CNT, 1　　B : PC = LABEL1 GOTO LOOP　　A : CNT = CNT + 1 CONTINUE　　if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1

IORLW　"k"和 W 逻辑或

句法	IORLW k
操作数	k : 00h ~ FFh
运行	(W) ← (W) OR k
影响的状态位	Z
OP-Code	01 1010 kkkk kkkk
描述	W 寄存器的值和 K 的值做 or 运算。结果放在 W 寄存器中。
周期	1
举例	IORLW 0x35　　B : W = 0x9A A : W = 0xBF, Z = 0

IORWF　W 和 F 逻辑或

句法	IORWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(目标寄存器) ← (W) OR k
影响的状态位	Z
OP-Code	00 0100 dfff ffff
描述	包含 W 寄存器和 F 寄存器值相或。如"d"为 0,结果放 W 中,如"d"为 1,结果放 F 中.
周期	1
举例	IORWF RESULT, 0　　B : RESULT = 0x13, W = 0x91 A : RESULT = 0x13, W = 0x93, Z = 0

,

MOVF　F 内容传送至目标寄存器

句法	MOVF f,d	
操作数	f : 00h ~ 7Fh	
运行	(目标寄存器) ← (f)	
影响的状态位	Z	
OP-Code	00 1000 0fff ffff	
描述	把 f 寄存器中的值移到目标寄存器中。如 d 为 0,目标寄存器为 W。如 d 为 1,目标寄存器为 f。	
周期	1	
举例	MOVF FSR, 0	B : FSR=0xC2, W = ? A : FSR=0xC2, W =0xC2

MOVFW　F 内容取出送 W

句法	MOVFW f	
操作数	f : 00h ~ 7Fh	
运行	(W) ← (f)	
影响的状态位	Z	
OP-Code	00 1000 0fff ffff	
描述	把 f 寄存器中的值移到 W 寄存器中。	
周期	1	
举例	MOVFW FSR, 0	B : FSR=0xC2, W = ? A : FSR=0xC2, W =0xC2

MOVLW　W 置立即数

句法	MOVLW k	
操作数	k : 00h ~ FFh	
运行	(W) ← k	
影响的状态位	-	
OP-Code	01 1001 kkkk kkkk	
描述	K 值装进 W 寄存器中。忽略之位将视为 0。	
周期	1	
举例	MOVLW 0x5A	B : W = ? A : W = 0x5A

MOVWF　把 W 内容送 F

句法	MOVWF f	
操作数	F : 00h ~ 7Fh	
运行	(f) ← (W)	
影响的状态位	-	
OP-Code	00 0000 1fff ffff	
描述	W 寄存器值移到 f 寄存器中。	
周期	1	
举例	MOVWF REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

NOP 空操作

句法	NOP
操作数	-
运行	空操作
影响的状态位	-
OP-Code	00 0000 0000 0000
描述	空操作
周期	1
举例	NOP -

RET 子程序返回

句法	RET
操作数	-
运行	PC ← TOS
影响的状态位	-
OP-Code	00 0000 0100 0000
描述	从子程序中返回。堆栈顶端的值会被弹出放在 PC 中。此为两个周期的指令。
周期	2
举例	RETURN A : PC = TOS

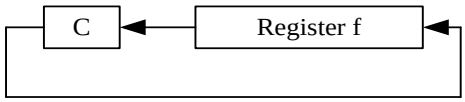
RETI 中断返回

句法	RETI
操作数	-
运行	PC ← TOS, GIE ← 1
影响的状态位	-
OP-Code	00 0000 0110 0000
描述	从中断中返回。堆栈被弹出，堆栈顶端的值会被弹进放在 PC 中。中断将被使能。此为两周期的指令。
周期	2
举例	RETFIE A : PC = TOS, GIE = 1

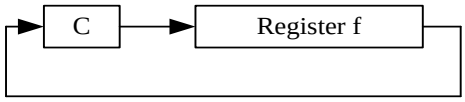
RETLW W 带立即数返回

句法	RETLW k
操作数	k : 00h ~ FFh
运行	PC ← TOS, (W) ← k
影响的状态位	-
OP-Code	01 1000 kkkk kkkk
描述	W 寄存器装进“K”值。堆栈顶弹出的值放入 PC 中。此为两个周期的指令。
周期	2
举例	CALL TABLE B : W = 0x07 : A : W = value of k8 TABLE ADDWF PCL,1 RETLW k1 RETLW k2 : RETLW kn

RLF 带进位循环左移

句法	RLF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	
影响的状态位	C
OP-Code	00 1101 dfff ffff
描述	C 位放在 F 值的 MSB 之前当作位 9，然后每个位向左边移动，C 位则位移到位 0，操作结果存于目标寄存器中。若‘d’为 0，结果放在 W 寄存器。若‘d’为 1，结果将放回寄存器‘f’。
周期	1
举例	RLF REG1,0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 1100 1100, C = 1

RRF 带进位循环右移

句法	RRF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	
影响的状态位	C
OP-Code	00 1100 dfff ffff
描述	C 位放在 F 值的 MSB 之前当作位 9，然后每个位向右移动，0 位则位移到 c 位，操作结果存于目标寄存器中。若‘d’为 0，结果放在 W 寄存器。若‘d’为 1，结果将放回寄存器‘f’。
周期	1
举例	RRF REG1,0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 0111 0011, C = 0

SLEEP 进入省电模式，震荡时钟停止

句法	SLEEP
操作数	-
运行	-
影响的状态位	TO，PD
OP-Code	00 0000 0000 0011
描述	进入省电模式，晶体停止振荡
周期	1
举例	SLEEP -

SUBLW 立即数减去 W

句法	SUBLW k
操作数	k : 00h ~ FFh
运行	(W) ← k - (W)
影响的状态位	C, DC, Z
OP-Code	01 1111 kkkk kkkk
描述	立即数 k 减去 W 寄存器的值(二进制补码运算)。结果放在 W 寄存器中。
周期	1
举例	SUBLW 0x15 B : W = 0x25 A : W = 0x10

SUBWF "f"减去 W

句法	SUBWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运作	(目标寄存器) ← (f) – (W)
影响的状态位	C, DC, Z
OP-Code	00 0010 dfff ffff
描述	寄存器 ‘f’的值减去 W 寄存器中的值(2 的 补码运算) 。如"d"为 0,结果放 W 中,如"d"为 1,结果放 F 中.
周期	1
举例	SUBWF REG1, 1 B : REG1 =0x03, W =0x02, C =?, Z =? A : REG1 =0x01, W =0x02, C =1, Z =0 SUBWF REG1, 1 B : REG1 =0x02, W =0x02, C =?, Z =? A : REG1 =0x00, W =0x02, C =1, Z =1 SUBWF REG1, 1 B : REG1 =0x01, W =0x02, C =?, Z =? A : REG1 =0xFF, W =0x02, C =0, Z =0

SWAPF "f"高低 4 位内容互换

句法	SWAPF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运作	(目标寄存器,7~4) ← (f.3~0), (目标寄存器.3~0) ← (f.7~4)
影响的状态位	-
OP-Code	00 1110 dfff ffff
描述	寄存器 f 的高低半位值互换。如"d"为 0,结果放 W 中,如"d"为 1,结果放 F 中.
周期	1
举例	SWAPF REG, 0 B : REG1 =0xA5 A : REG1 =0xA5, W =0x5A

TABRH 将 DPTR 高字节内容加载到 W

句法	TABRH
操作数	-
运作	(W) ← ROM[DPTR]高字节内容, 其中 DPTR = {DPH[max:8], FSR[7:0]}
影响的状态位	-
OP-Code	00 0000 0101 1000
描述	W寄存器装载了高位字节的ROM [DPTR]。 这是一个双周期指令。
周期	2
举例	

```
MOVLW   (TAB1&0xFF)
MOVWF   FSR                 ;其中 FSR 是 F-Plane 寄存器
MOVLW   (TBA1>>8)&0xFF
MOVWF   DPH                 ;其中 DPH 是 F-Plane 寄存器

TABRL                       ;W =0x89
TABRH                       ;W =0x37

        ORG 0234H
TAB1:
DT      0x3789, 0x2277      ;ROM data 14 bits
```

TABRL 将 DPTR 低字节内容加载到 W

句法	TABRL
操作数	-
运作	(W) ← ROM[DPTR]低字节内容，其中 DPTR = {DPH[max:8], FSR[7:0]}
影响的状态位	-
OP-Code	00 0000 0101 0000
描述	W寄存器加载低位字节的ROM [DPTR]。 这是一个双周期指令。
周期	2
举例	

```
MOVLW   (TAB1&0xFF)
MOVWF   FSR                 ;其中 FSR 是 F-Plane 寄存器
MOVLW   (TBA1>>8)&0xFF
MOVWF   DPH                 ; 其中 DPH 是 F-Plane 寄存器

TABRL                       ;W =0x89
TABRH                       ;W =0x37

        ORG 0234H
TAB1:
DT      0x3789, 0x2277      ;ROM data 14 bits
```

TSTF 检测 "f"是否为 0

句法	TSTF f
操作数	f : 00h ~ 7Fh
运行	Set Z flag if (f) is 0
影响的状态位	Z
OP-Code	00 1000 1fff ffff
描述	若寄存器 f 的内容是 0，零标志被设置为 1。
周期	1
举例	TESTZ REG1　　B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1

XORLW W 和立即数逻辑异或

句法	XORLW k
操作数	k : 00h ~ FFh
运行	(W) ← (W) XOR k
影响的状态位	Z
OP-Code	01 1111 kkkk kkkk
描述	‘K’的值和 W 寄存器的值做 XOR 运算。运算结果放进 W 寄存器中。
周期	1
举例	XORLW 0xAF　　B : W = 0xB5 A : W = 0x1A

XORWF W 和 F 逻辑异或

句法	XORWF f [,d]
操作数	f : 00h ~ 7Fh, d : 0, 1
运行	(目标寄存器) ← (W) XOR (f)
影响的状态位	Z
OP-Code	00 0110 dfff ffff
描述	f 寄存器的值和 W 寄存器的值做 XOR 运算。如"d"为 0,结果放 W 中,如"d"为 1,结果放 F 中.
周期	1
举例	XORWF REG 1　　B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

电气特性

1. 绝对最大值 (T_A = 25℃)

参数	范围	单位
电源电压	V_{SS} -0.3 to V_{SS} +5.5	V
输入电压	V_{SS} -0.3 to V_{DD} +0.3	
输出电压	V_{SS} -0.3 to V_{DD} +0.3	
每个引脚的输出拉电流	-25	mA
所有引脚的输出拉电流	-80	
每个引脚的输出灌电流	+30	
所有引脚的输出灌电流	+150	
最大工作电压	5.5	V
工作温度	-40 to +105	°C
储存温度	-65 to +150	

2. 直流特性 (T_A = 25℃, V_{CC} = 5.0V, 除非另有规定)

参数	符号	条件		最小	典型	最大	单位
工作电压	V_{cc}	Fsys = 16 MHz (FIRC)		2	–	5.5	V
		Fsys = 8 MHz (FIRC/2)		1.7	–	5.5	V
输入高电压	V_{IH}	所有输入	V_{CC} = 3.0~5.0V	0.6V_{CC}	–	V_{CC}	V
输入低电压	V_{IL}	所有输入	V_{CC} = 3.0~5.0V	V_{SS}	–	0.2V_{CC}	V
I/O 端口拉电流	I_{OH}	所有引脚	V_{CC} = 5.0V, V_{OH} = 4.5V	6	12	–	mA
			V_{CC} = 3.0V, V_{OH} = 2.7V	2.5	5	–	
I/O 端口灌电流	I_{OL}	所有引脚	V_{CC} = 5.0V, V_{OL} = 0.5V	30	45	–	mA
			V_{CC} = 3.0V, V_{OL} = 0.3V	15	20	–	
输入漏电流 (引脚为高)	I_{ILH}	所有输入	V_{IN} = V_{CC}	–	–	1	μA
输入漏电流 (引脚为低)	I_{ILL}	所有输入	V_{IN} = 0V	–	–	–1	μA

参数	符号	条件		最小	典型	最大	单位
工作电流 (空载)	I_{CC}	快速模式 FIRC 16 MHz	V_{CC} = 5.0V	–	4	–	mA
			V_{CC} = 3.0V	–	2.3	–	
		快速模式 FIRC 8 MHz	V_{CC} = 5.0V	–	2.9	–	
			V_{CC} = 3.0V	–	1.9	–	
		快速模式 FIRC 4 MHz	V_{CC} = 5.0V	–	1.7	–	
			V_{CC} = 3.0V	–	1.3	–	
		快速模式 FIRC 2 MHz	V_{CC} = 5.0V	–	1.4	–	
			V_{CC} = 3.0V	–	1	–	
		慢速模式 SXT 32 KHz FIRC 停止 ATDOFF=0	V_{CC} = 5.0V	–	78	–	uA
			V_{CC} = 3.0V	–	35	–	
		慢速模式 SIRC 除 1 FIRC 停止 ATDOFF=0	V_{CC} = 5.0V	–	58	–	
			V_{CC} = 3.0V	–	34	–	
		慢速模式 SIRC 除 1 FIRC 停止 ATDOFF=0 POR/LVR 关闭	V_{CC} = 5.0V	–	24	–	
			V_{CC} = 3.0V	–	15	–	
		空闲模式 SIRC 除 1 POR/LVR 关闭	V_{CC} = 5.0V	–	8	–	μA
			V_{CC} = 3.0V	–	2.7	–	
		停止模式 POR/LVR 关闭	V_{CC} = 5.0V	–	–	1	μA
			V_{CC} = 3.0V	–	–	1	
上拉电阻	R_{UP}	V_{IN} = 0 V 端口 A, B, D	V_{CC} = 5.0V	–	30	–	KΩ
			V_{CC} = 3.0V	–	48	–	

3. 时钟时序

此参数数值是基于测试样本的特性值

参数	条件		最小	典型	最大	单位
FIRC 频率(*)	T_A = 25℃	V_{CC} = 5.0 V	-1%	16	+1%	MHz
	T_A = -40℃ ~ 85℃	V_{CC} = 5.0 V	-2%	16	+2%	
	T_A = -40℃ ~ 105℃	V_{CC} = 5.0 V	-3%	16	+2%	
	T_A = -40℃ ~ 85℃	V_{CC} = 3.0 ~ 5.0V	-4%	16	+2%	
	T_A = -40℃ ~ 105℃	V_{CC} = 2.0 ~ 5.0V	-5%	16	+2%	
SIRC 频率	T_A = 25℃	V_{CC} = 5.0 V		51		KHz

4. 复位时间特性 (T_A = 25°C)

参数	条件	最小	典型	最大	单位
复位输入低脉宽	输入 V_{CC} = 5.0 V ±10 %	–	30	–	μs
CPU 启动时间	V_{CC} = 5.0 V	–	20	–	ms

5. LVR 电路特性 (T_A = 25℃)

参数	符号	条件	最小	典型	最大	单位
LVR 参考电压	LVR_{th}	T_A = 25℃	–	1.83	–	V
			–	2.35	–	
			–	2.87	–	
LVR 迟滞电压	V_{HYS_LVR}	T_A = 25℃	–	20	–	mV
低电压检测时间	T_{LVR}	T_A = 25℃	100	–	–	μs

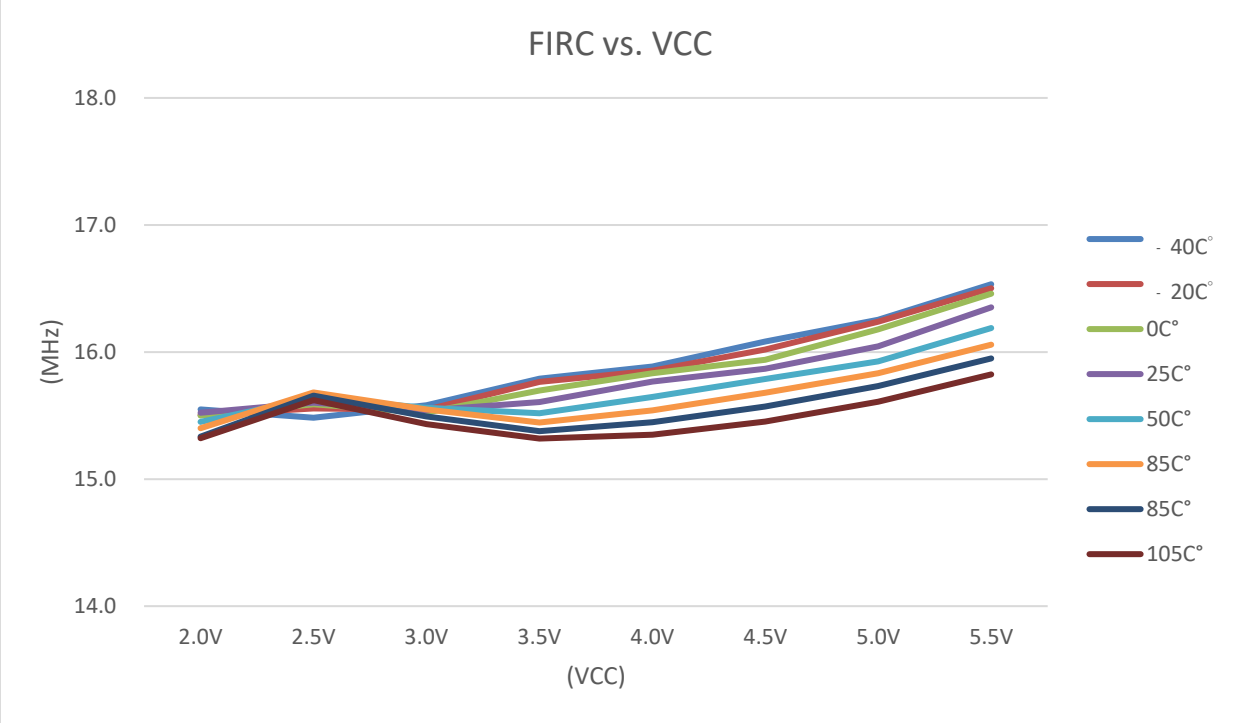
6. ADC 电气特性 (T_A = 25℃, V_{CC} = 3.0V to 5.5V, V_{SS} = 0V)

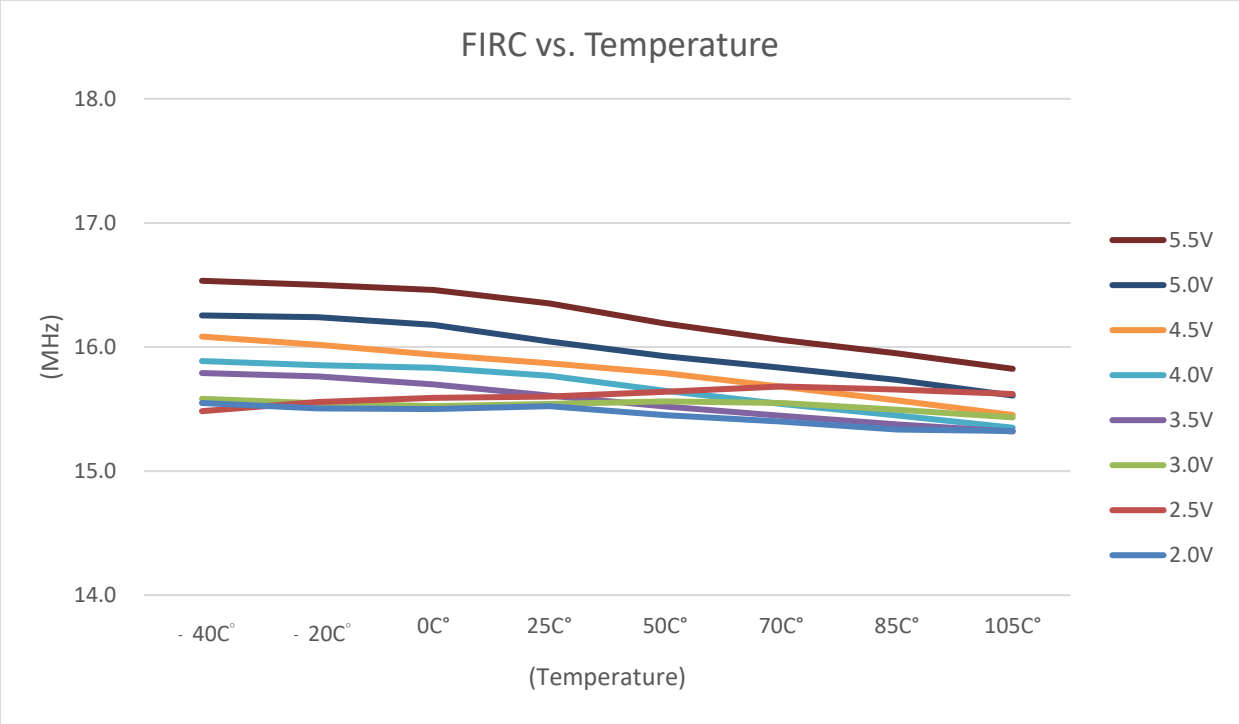
参数	条件	最小	典型	最大	单位
总精度	V_{CC} = 5.0V, V_{SS} = 0V, F_{ADC} = 1 MHz	–	±3	–	LSB
积分非线性误差		–	±3.2	–	
微分非线性误差		–	±1	±4	
最大输入时钟频率 (F_{ADC})	信号驱动源阻抗 (Rs<5K ohm)	–	–	4	MHz
	信号驱动源阻抗 (Rs<10K ohm)	–	–	2	
	信号驱动源阻抗 (Rs<25K ohm)	–	–	1	
	信号来源是 VBG (ADCHS=1110b)	–	–	F_{SYSCLK}/4	
转换时间	F_{ADC} = 1 MHz	–	21	–	μs
带隙电压参考 (V_{BG})	25°C, V_{CC} = 3V~5.0V	-2.8%	1.20	+2.8%	V
	-20°C~70°C, V_{CC} = 2.5V~5.0V	-3.5%	1.20	+3.5%	V
	-40°C~105°C, V_{CC} = 2.5V~5.0V	-3.8%	1.20	+3.5%	V
输入电压	–	V_{SS}	–	V_{CC}	V

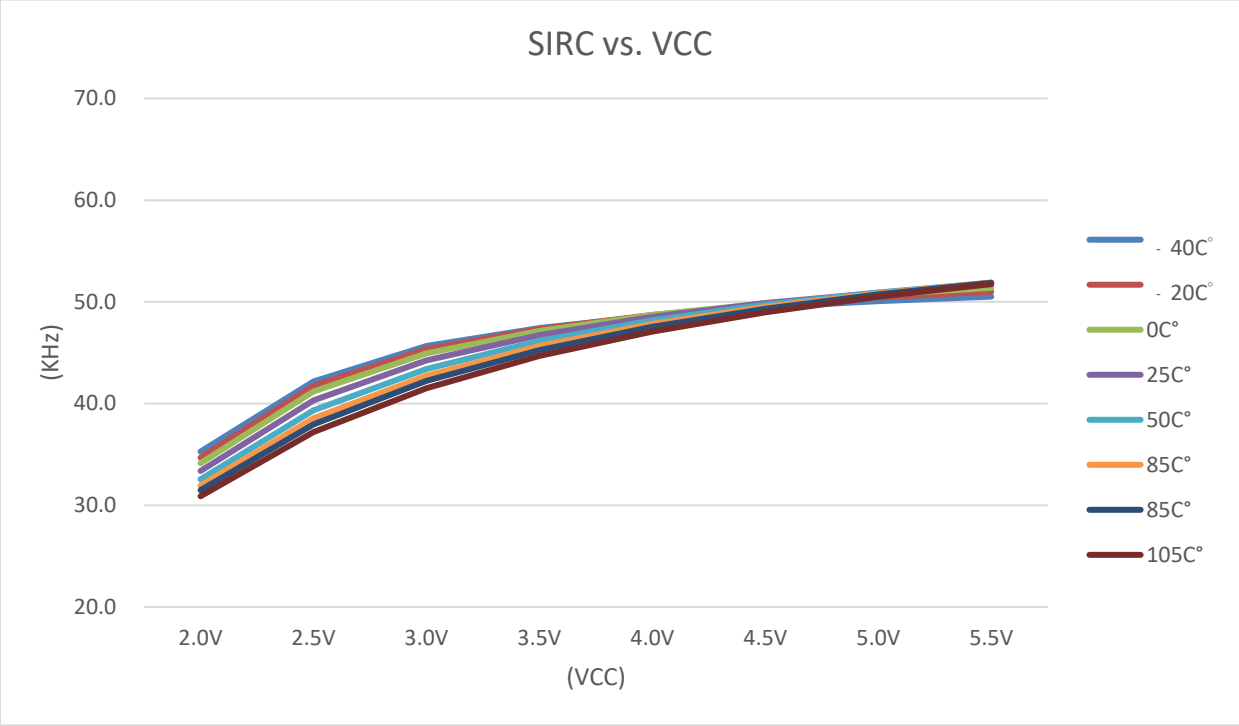
7. 比较器特性 (T_A = 25℃, V_{CC} = 3.0V to 5.5V, V_{SS} = 0V)

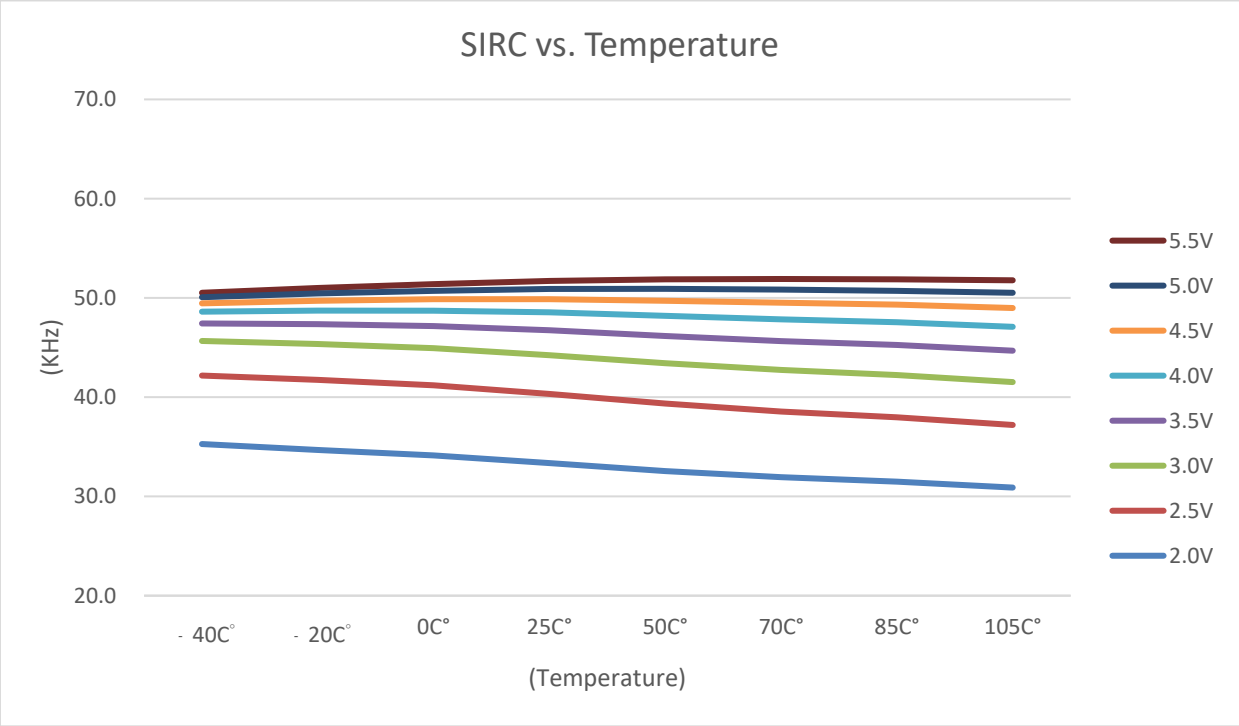
参数	条件	最小	典型	最大	单位
电源电压	–	2.2	–	5.5	V
静态电流	V_{CC} = 5.0V	–	100	–	μA
DAC Current	V_{CC} = 5.0V	60	–	220	μA
V_{OS_CMP}	V_{CC} = 5.0V	-15	–	15	mV
V_{CM_CMP}	V_{CC} = 5.0V	0	–	V_{CC}-0.5	V
V_{HYS_CMP}	V_{CC} = 5.0V	20	30	40	mV

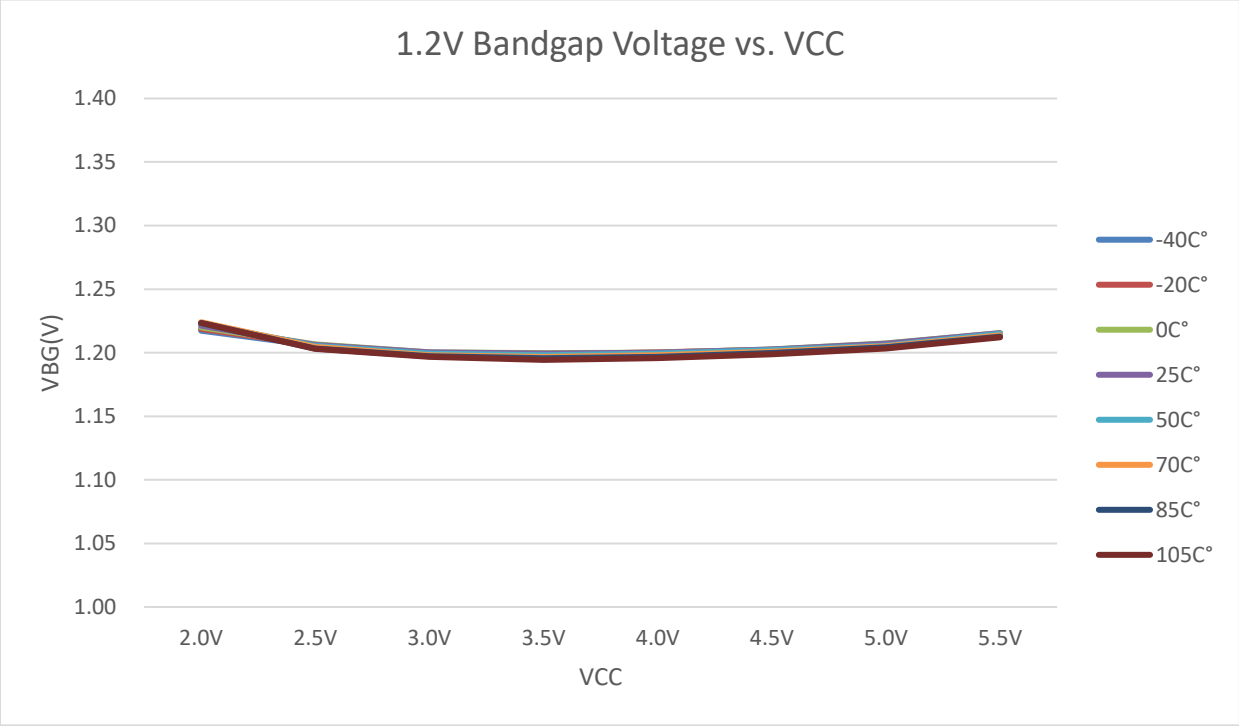
8. 特性曲线图

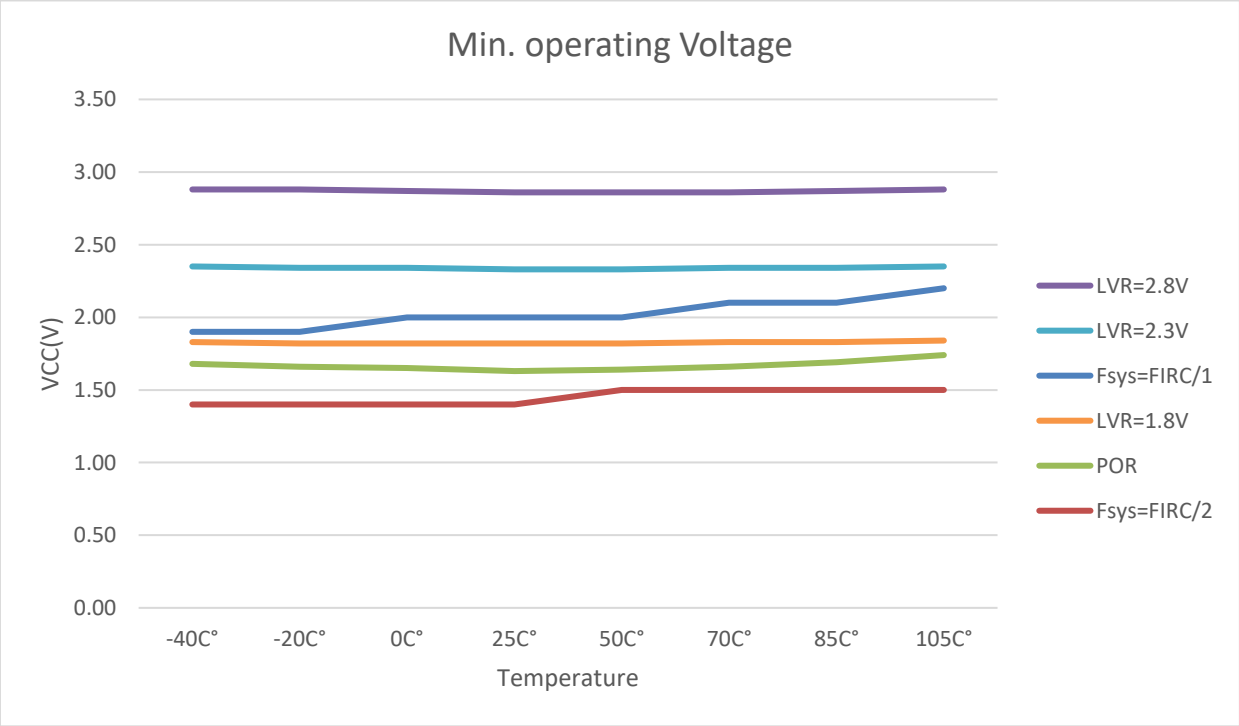

FIRC vs. VCC
18.0
17.0
16.0
15.0
14.0
(MHz)
2.0V
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V
(VCC)
- 40C°
- 20C°
0C°
25C°
50C°
85C°
85C°
105C°


FIRC vs. Temperature
18.0
17.0
16.0
15.0
14.0
(MHz)
- 40C°
- 20C°
0C°
25C°
50C°
70C°
85C°
105C°
(Temperature)
5.5V
5.0V
4.5V
4.0V
3.5V
3.0V
2.5V
2.0V


SIRC vs. VCC
70.0
60.0
50.0
40.0
30.0
20.0
(KHz)
2.0V
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V
(VCC)
- 40C°
- 20C°
0C°
25C°
50C°
85C°
85C°
105C°


SIRC vs. Temperature
70.0
60.0
50.0
40.0
30.0
20.0
(KHz)
- 40C°
- 20C°
0C°
25C°
50C°
70C°
85C°
105C°
(Temperature)
5.5V
5.0V
4.5V
4.0V
3.5V
3.0V
2.5V
2.0V


1.2V Bandgap Voltage vs. VCC
1.40
1.35
1.30
1.25
1.20
1.15
1.10
1.05
1.00
VBG(V)
2.0V
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V
VCC
-40C°
-20C°
0C°
25C°
50C°
70C°
85C°
105C°


Min. operating Voltage
3.50
3.00
2.50
2.00
1.50
1.00
0.50
0.00
VCC(V)
-40C°
-20C°
0C°
25C°
50C°
70C°
85C°
105C°
Temperature
LVR=2.8V
LVR=2.3V
Fsys=FIRC/1
LVR=1.8V
POR
Fsys=FIRC/2

封装说明

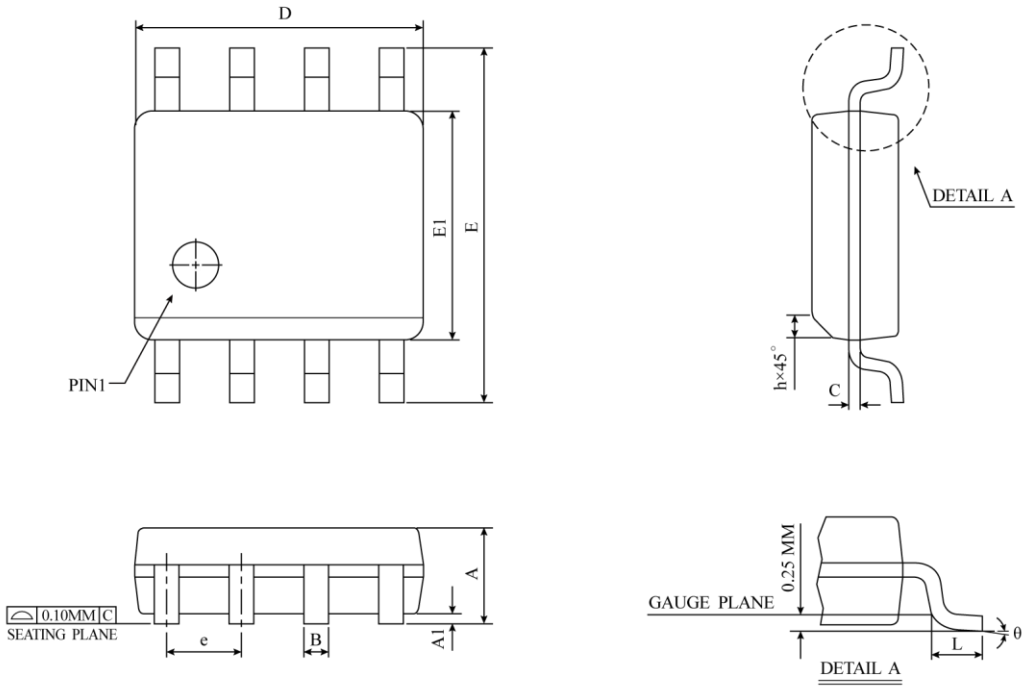
请注意，此处提供的包装信息仅供参考。 由于此信息经常更新，因此用户可以联系销售人员以咨询最新的包装信息和库存

订购须知：

订购代码	封装
TM54M03211S	SOP 8-pin (150 mil)
TM54M03212S	SOP 16-pin (150 mil)
TM54M03212Q	QFN 16-pin (3x3x0.75-0.5 mm)

包装信息

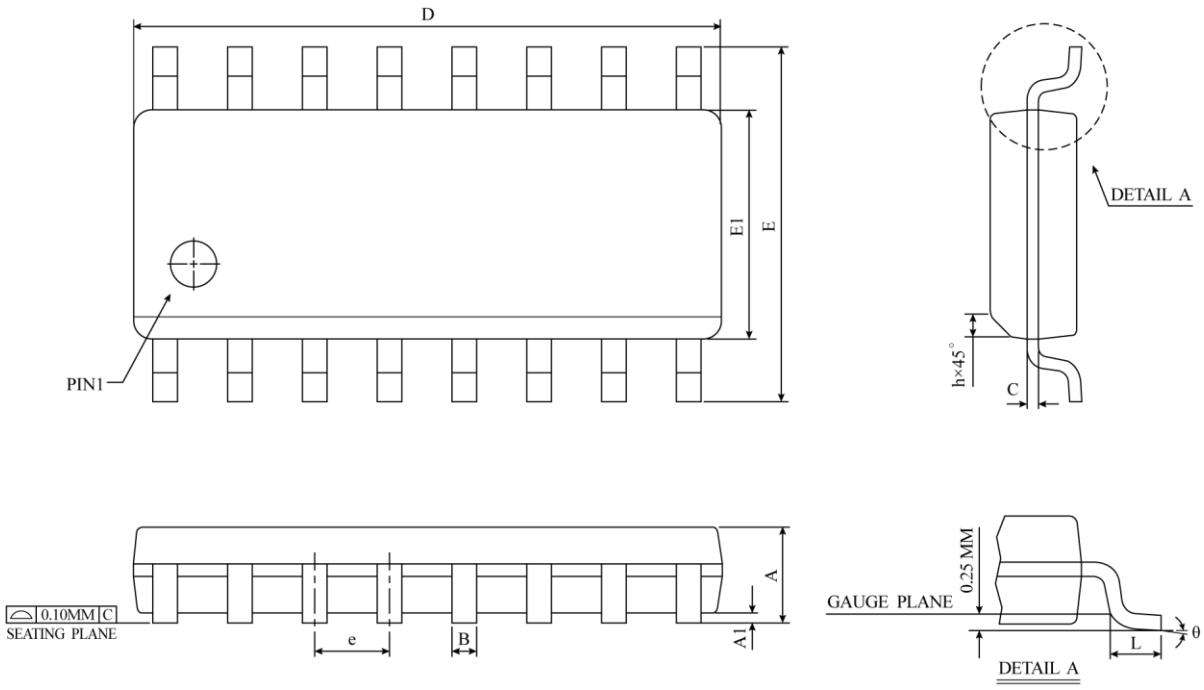
SOP-8 (150mil) 封装尺寸



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.55	1.75	0.0532	0.0610	0.0688
A1	0.10	0.18	0.25	0.0040	0.0069	0.0098
B	0.33	0.42	0.51	0.0130	0.0165	0.0200
C	0.19	0.22	0.25	0.0075	0.0087	0.0098
D	4.80	4.90	5.00	0.1890	0.1939	0.1988
E	5.80	6.00	6.20	0.2284	0.2362	0.2440
E1	3.80	3.90	4.00	0.1497	0.1536	0.1574
e	1.27 BSC			0.050 BSC		
h	0.25	0.38	0.50	0.0099	0.0148	0.0196
L	0.40	0.84	1.27	0.0160	0.0330	0.0500
θ	0°	4°	8°	0°	4°	8°
JEDEC	MS-012 (AA)					

⚠ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

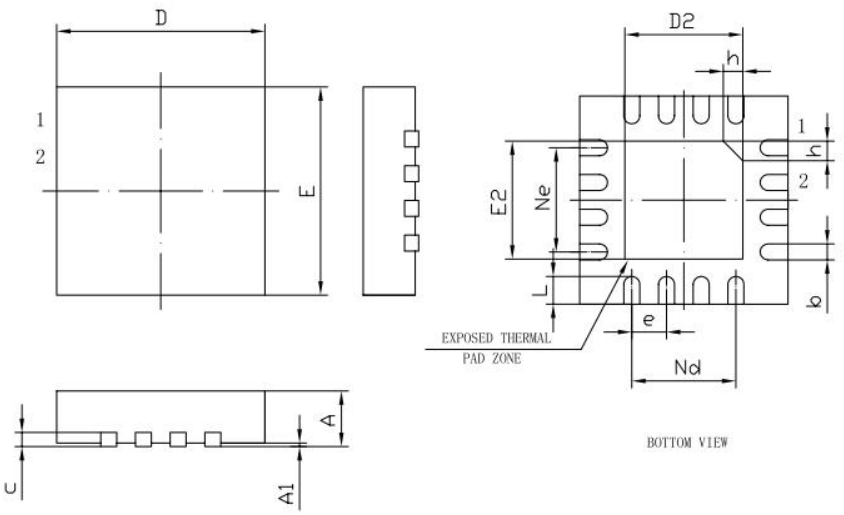
16-SOP (150 mil) 封装尺寸



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.55	1.75	0.0532	0.0610	0.0688
A1	0.10	0.18	0.25	0.0040	0.0069	0.0098
B	0.33	0.42	0.51	0.0130	0.0165	0.0200
C	0.19	0.22	0.25	0.0075	0.0087	0.0098
D	9.80	9.90	10.00	0.3859	0.3898	0.3937
E	5.80	6.00	6.20	0.2284	0.2362	0.2440
E1	3.80	3.90	4.00	0.1497	0.1536	0.1574
e	1.27 BSC			0.050 BSC		
h	0.25	0.38	0.50	0.0099	0.0148	0.0196
L	0.40	0.84	1.27	0.0160	0.0330	0.0500
θ	0°	4°	8°	0°	4°	8°
JEDEC	MS-012 (AC)					

⚠ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

QFN-16 (3*3*0.75-0.5mm) 封装尺寸



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	—	0.02	0.05
b	0.18	0.25	0.30
c	0.18	0.20	0.25
D	2.90	3.00	3.10
D2	1.55	1.65	1.75
e	0.50BSC		
Ne	1.50BSC		
Nd	1.50BSC		
E	2.90	3.00	3.10
E2	1.55	1.65	1.75
L	0.35	0.40	0.45
h	0.20	0.25	0.30
L/F载体尺寸 (mil)	75x75		