



西安恩狄集成电路有限公司
Analog & Digital Microelectronics Limited

AD1845

用户手册

版本号: V1.0.0.1

支持 12 位 A/D、16 位电量 A/D 的 单节锂电池控制专用单片机

版权所有©

西安恩狄集成电路有限公司

本资料内容为西安恩狄集成电路有限公司在现有数据资料基础上编制而成，本资料中所记载的实例以正确的试用方法和标准操作为前提，使用方在应用该等实例时应充分考虑外部诸条件，西安恩狄集成电路有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，西安恩狄集成电路有限公司亦不对使用方使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。文档中所有涉及到第三方软件的，请自行购买正版软件，因第三方软件版权问题涉及到的一切后果，与西安恩狄集成电路有限公司无关。基于使本资料的内容更加完善等原因，西安恩狄集成电路有限公司保留未经预告的修改权。

西安恩狄集成电路有限公司

地址：陕西省西安市高新区高新一路 19 号思安大厦 501

电话：+ (86 29) 88322766 网站：www.admicrochip.com

微信号：恩狄 ADUC



版本修订记录

Bin	Version	Change List	Owner	Data
1	1.0.0.0	初版	Gaoqili	2021.7.9
2	1.0.0.1	引脚名称修订	Liuheidong	2022.7.4

目录

版本修订记录.....	1
1 产品简介.....	5
1.1 产品特性.....	5
1.2 系统框图.....	7
1.3 引脚排列.....	7
1.4 引脚说明.....	8
2 中央处理器.....	10
2.1 指令集.....	10
2.2 程序存储 ROM.....	10
2.3 数据存储器 RAM.....	14
2.4 数据寻址方式.....	15
2.5 用户配置字 Configuration	18
3 系统时钟.....	26
3.1 外部晶振.....	27
3.2 内置高精度 RC 振荡	27
3.3 WDT 振荡器.....	28
4 复位.....	29
4.1 RCON 寄存器.....	29
4.2 主清零 (RSTn)	30
4.3 上电复位 (POR)	30
4.4 欠压复位 (BOR)	30
5 I/O 口	31
5.1 I/O 工作模式	31
5.2 上拉/下拉电阻和开漏.....	32
5.3 端口模拟输出控制.....	33
6 定时器.....	35

6.1 Timer0 16 位自动重载定时/计数器	35
6.2 Timer1 16 位自动重载定时/计数器	37
6.3 TIMER2 模块	41
6.4 WDT 模块.....	43
7 四路捕捉/比较/ PWM 模块	45
7.1 捕捉模式.....	56
7.2 比较模式.....	58
7.3 PWM 模式.....	59
8 中断.....	62
8.1 INTn 引脚中断	63
8.2 定时器中断.....	65
8.3 PORTA/PORTB 电平变化中断	66
8.4 外设中断.....	69
9 I2C 控制器.....	71
9.1 从动模式.....	77
9.2 支持广播呼叫地址.....	79
10 CRC16.....	80
11 通用同步/异步收发器 (USART)	81
11.1 波特率发生器 (BRG)	84
11.2 异步发送器.....	85
11.3 异步接收.....	87
11.4 同步模式.....	88
11.5 同步从动模式.....	92
12 8×8 硬件乘法器和 16/16 硬件除法器	94
13 固定参考电压 (FVR)	96
14 A/D 转换器	97
15 ADC 电荷计数器- Sigma-delta 型计量标准的专用型模数转换器.....	103
15.1 特性.....	103
15.2 概述.....	103

15.3 普通操作.....	104
15.4 常规电流检测.....	106
15.5 切换极性消除偏移.....	106
15.6 配置和使用.....	107
15.7 寄存器描述.....	107
15.7.2 CADCSRB – CC-ADC 控制/状态寄存器 B.....	108
16 系统低功耗 SLEEP 模式.....	114
17 LVD 检测.....	116
18 OTP 操作描述.....	118
18.1 OTP 的相关寄存器.....	118
19 电气特性.....	121
19.1 绝对极限参数 over operating free-air temperature range.....	121
19.2 电源特性.....	121
19.3 直流特性.....	121
19.4 交流特性.....	122
19.5 ADC 电气特性.....	122
19.6 CCADC(Instant-Current&Coulomb-Counter)电气特性.....	122
19.7 LVR 电气特性.....	123
19.8 POR 电气特性.....	123
19.9 BOR 电气特性.....	123
19.10 FVR 电气特性.....	123
20 封装尺寸.....	124
21 订购信息.....	125

1 产品简介

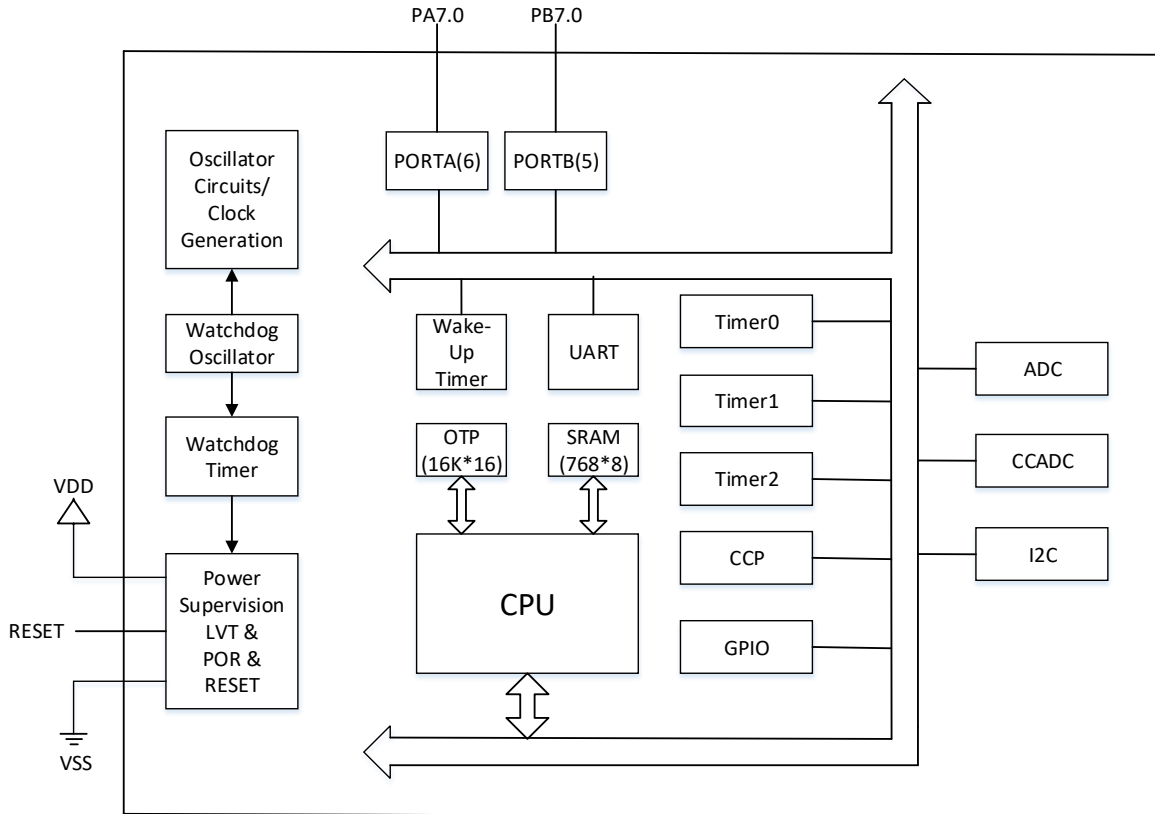
1.1 产品特性

- 8位 RISC CPU 内核，16MHz，支持 2T、4T CPU 运算模式，内置 8x8 单周期硬件乘法器，16/16 除法器
- 存储器
 - ◇ 16K*16 位 OTP 程序存储器空间（可以配置成 1K*16,2K*16,4K*16,8K*16,16K*16 存储模式），16 级深度硬件堆栈
 - ◇ 支持分页加密 支持 OTP 程序空间二次写入
 - ◇ 768 字节通用数据寄存器空间
 - ◇ 256 字节 EEPROM
- 2 组 IO 口（最多可支持 11 个通用 IO 口，支持 20mA 的输出电流和灌电流；支持上下拉电阻；所有 IO SMT 支持两路 SMT 电平输入选择）
- 正常高速工作模式(20MHz, IO 不翻转)，小于 2mA；低功耗模式，唤醒时间可配置，最小 16 个系统时钟；PowerOff 模式功耗小于 10uA；PowerSave 模式功耗小于 30uA
- 接口
 - ◇ WDT：内部自振式看门狗计数器（WDT）
 - ◇ Timer0：16 位定时/计数器，可设置溢出中断，可以工作在睡眠模式；
 - ◇ Timer1：16 位定时/计数器，可以工作在睡眠模式，可设置溢出中断，可以自动重载初始值；与 CCPR1、CCPR2，CCPR3 寄存器，采用 CCP 比较模式可构成 4 路 16 位精度的 PWM 输出。
 - ◇ Timer2：12 位定时/计数器，自动加载功能 12 位定时器，可设置溢出中断
 - ◇ 4 路 CCP：捕获、比较、12bit 分辨率的互补带死区的独立 PWM 输出模式；通过 2 路 CCP 可快速实现 HDQ 单总线解析。
 - ◇ 从 I2C 接口，硬件从 I2C 接口，支持 100K，400K，1M BPS，支持四个 7 位从地址应答寄存器，可设定是否产生应答，支持应答位的时钟扩展
 - ◇ UART：同步、异步、半双工串口，支持 1bit、2bit、3bit 停止位
 - ◇ IO 中断：3 个独立外部硬件中断、PA、PB 中断
 - ◇ ADC：12BIT 1MHz 的 SAR ADC，16 个通路输入选择，可选的外部触发转换，可

- 编程的采集时间；
- ◇ CCADC：瞬态 3.9ms 的转换时间的 16BIT 库伦 ADC，支持 200mv/100mv 两个档位采集；累加模式支持 18 位精度；
 - ◇ FVR：专用参考电压源，可输出 1.0v, 1.1v, 1.2v, 2.0v, 2.2v, 2.4v (1%) 的参考电压源，支持温度传感器数据采集；
 - ◇ LVR：多路低电压复位检测模块；支持高压检测模式，可用于电池电压二级保护检测，可以实现不采用程序控制的，VCC 电压自动监测功能；当电压异常，强制 IO 输出到固定电位。
 - ◇ 一路可控 2.0V LDO 输出，实现电池协议电平需要。
- 中断
- ◇ 定时器 0 中断
 - ◇ 定时器 1 中断
 - ◇ 定时器 2 中断
 - ◇ CCP 中断
 - ◇ I2C 中断
 - ◇ UART 接收和发送中断
 - ◇ AD 转换器中断
 - ◇ CCADC 转换器中断
 - ◇ 低电压检测中断
 - ◇ 3 个外部中断口（可软件设置为上升沿、下降沿触发）
 - ◇ 16 路键盘中断功能
- 振荡器
- ◇ 内部 1M RC 振荡器，用于看门狗计数
 - ◇ 内部 16M RC 振荡器，用于系统时钟
- 芯片具有唯一 ID 号，使能代码加密功能后，具有代码自动加密功能，可以保证每片芯片程序与其他芯片不一致；防止代码拷贝破解，保护知识产权
- 支持双线快速烧录
- 支持封装
- DFN12, DFN8

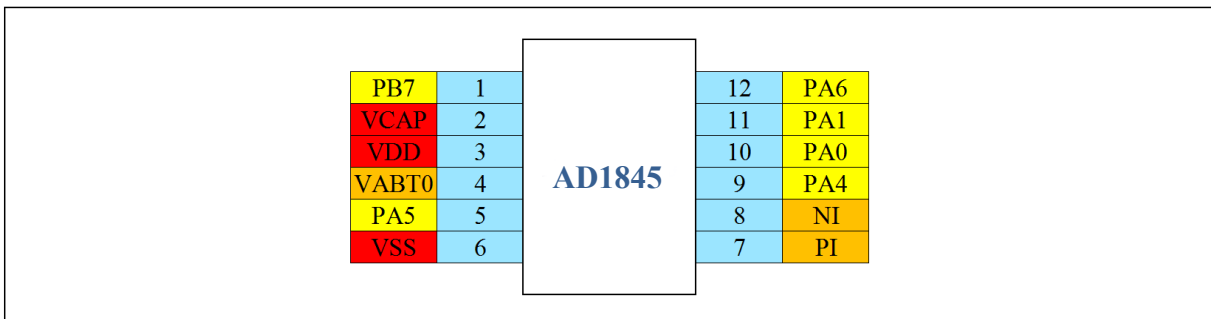
- 工作电压
 - ✧ 2.6~5.5V@ (振荡频率 412KHz-16MHz)

1.2 系统框图

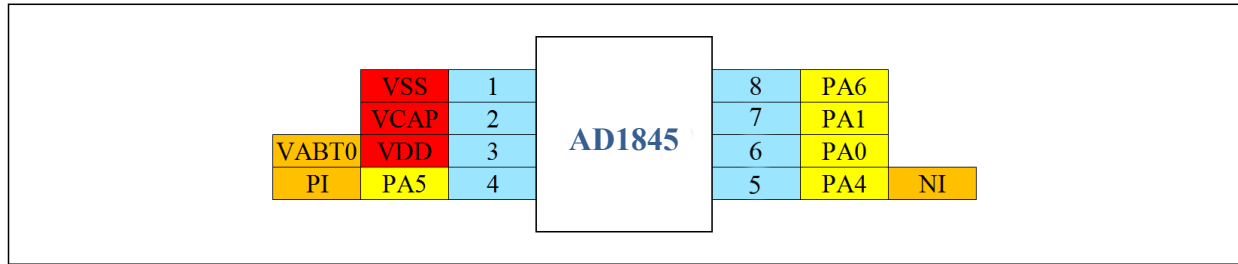


AD1845 系统框图

1.3 引脚排列



DFN12 引脚图



DFN8 引脚图

1.4 引脚说明

SMT1(0.7VDD,0.3VDD), SMT0(0.4VDD,0.2VDD)

管脚名	管脚复用	输入类型	输出类型	A/D	端口说明	备注
PB7	PB7	SMT0/SMT1	CMOS	D	通用 I/O	可单独使 能弱上拉、 下拉、开漏
	PWM21	-	CMOS	D	FPWM21 输出	
	RSTn	SMT0/SMT1	-	D	引脚复位，低有效	
	RX	-	CMOS	D	UART 发送输出	
	FVROUT	SMT0/SMT1	CMOS	D	FVR 的电压输出	
	KAINT7	SMT0/SMT1	-	D	键盘中断 KB7	
VCAP	VCAP	-	-	-	2.0V LDO 输出管脚，需要 0.1uF 电容	
VDD	VDD	-	-	-	电源，2.6~5.5V 供电	
PA5	PA5	SMT0/SMT1	CMOS	D	通用 I/O	可单独使 能弱上拉、 下拉、开漏
	INT0	SMT0/SMT1	-	D	外部中断 INT0	
	CKOE	-	CMOS		系统时钟输出	
	AN3	-	-	A	ADC3	
	KAINT5	SMT0/SMT1	-	D	键盘中断 KA5	
VSS	VSS	-	-	-	地	
PI	PI	-	-	A	CCADC 差分输入	
NI	NI	-	-	A	CCADC 差分输入	
	TX				发送	
	PA4	SMT0/SMT1	CMOS	D	通用 I/O	可单独使

PA4	KAINT4	SMT0/SMT1	-	D	键盘中断 KA4	能弱上拉、 下拉、开漏
	PWM20	-	CMOS	D	PWM20 输出	
	TX_RX	SMT0/SMT1	-	D	串口接收发送	
	INT0	SMT0/SMT1	-	D	外部中断 INT0	
	CCPCH3	SMT0/SMT1	CMOS	D	捕获、比较输入输出通道 3	
	AN2	-	-	A	ADC2	
PA0	PA0	SMT0/SMT1	开漏	D	开漏 I/O	可单独使 能下拉
	KAINT0	SMT0/SMT1	-	D	键盘中断 KA0	
	SDA	SMT0/SMT1	开漏	D	I2C 主从 SDA 引脚	
	INT0	SMT0/SMT1	-	D	外部中断 INT0	
	CCPCH2	SMT0/SMT1	开漏	D	捕获、比较输入输出通道 2	
	AN0	-	-	A	ADC0	
PA1	PA1	SMT0/SMT1	开漏	D	通用 I/O	可单独使 能下拉
	KAINT1	SMT0/SMT1	-	D	键盘中断 KA1	
	SCL	SMT0/SMT1	开漏	D	I2C 主从 SCL 引脚	
	INT1	SMT0/SMT1	-	D	外部中断 INT1	
	CCPCH1	SMT0/SMT1	开漏	D	捕获、比较输入输出通道 1	
	AN2	-	-	A	ADC2	
PA6	PA6	SMT0/SMT1	开漏	D	通用 I/O	可单独使 能下拉
	KAINT6	SMT0/SMT1	-	D	键盘中断 KA6	
	HDQ				单总线	
	INT2	SMT0/SMT1	-	D	外部中断 INT2	
	CCPCH0	SMT0/SMT1	开漏	D	捕获、比较输入输出通道 0	
	AN4	-	-	A	ADC4	
BAT0	-	-	-	A	需外接到待测电压上	

2 中央处理器

2.1 指令集

AD1845 具有一个支持 83 条内核指令的标准指令集。其中包含 8 条针对优化递归和软件堆栈代码的扩展指令。

2.2 程序存储 ROM

AD1845 的程序存储器是 16K*16bits 的 EPROM，可用于存放用户程序。

AD1845 有两个中断向量。复位向量地址为 0000h,中断向量地址为 0008h 和 0018h。

2.2.1 程序计数器

程序计数器（Program Counter,PC），指定要取出执行的指令地址。PC 内的地址为 14 位二进制数，并且保存在 2 个独立的 8 位寄存器中。其中的低字节称为 PCL 寄存器，该寄存器可读写。高字节为 PCH 寄存器，存储 PC<14: 8>位，不可直接读写。可以通过 PCLATH 寄存器更新 PCH 寄存器。

通过执行写 PCL 的操作，可以将 PCLATH。类似的，通过执行读 PCL 的操作，可以将程序计数器的两个高字节传送到 PCLATH。

CALL, RCALL, GOTO 和程序转移指令直接写入程序计数器。对于这些指令，PCLATH 的内容将不会被传送到程序计数器。

2.2.2 返回地址堆栈

用于存放返回地址的堆栈允许保存最多 16 个程序调用地址和中断向量。当执行 CALL 或 RCALL 指令或响应中断时，PC 值被压入堆栈。而执行 RETURN、RETLW 或 RETFIE 指令时，PC 值从堆栈弹出。PCLATH 不受 RETURN 或 CALL 指令的影响。

通过 21 位的 RAM 和 4 位的堆栈指针（STKPTR）来实现 16 级的堆栈操作。堆栈既不占用程序存储空间也不占用数据存储空间。堆栈指针可以读写，并且通过栈顶的特殊文件寄存器可以读写栈顶地址。也可使用这些寄存器将数据压入堆栈，或将数据从堆栈弹出。

执行 CALL 类型指令引起进栈操作：堆栈指针首先加 1，并且将 PC 的内容写入堆栈指针指向的地址单元（PC 已经指向 CALL 的下一条指令）。执行 RETURN 类型指令时，引起

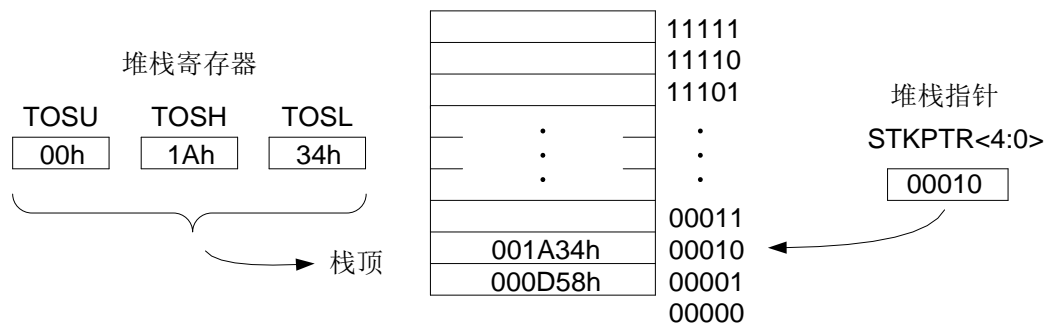
出栈操作：STKPTR 寄存器所指向的地址单元的内容被传送给 PC，然后堆栈指针减 1。

所有复位后，堆栈指针被初始化 00000。堆栈指针值 00000 不指向任何 RAM 单元，它只是一个复位值。状态表明堆栈是满、上溢还是下溢。

2.2.2.1 栈顶访问

只有栈顶（Top-of-Stack,TOS）时可读写的。有 2 个寄存器 TOSH:TOSL 用于保存 STKPTR 寄存器所指向的堆栈单元的内容。这可以让用户在必要时实现软件堆栈。在 CALL、RCALL 或中断后，软件可以通过读取 TOSH:TOSL 寄存器来读取进栈值。这些值可以被置入用户定义的软件堆栈。返回时，软件将这些值存回 TOSH:TOSL 并执行返回。

为防止对堆栈的意外操作，访问堆栈时用户必须禁止全局中断使能位。



2.2.2.2 返回堆栈指针（STKPTR）

STKPTR 寄存器包含堆栈指针值、STKFUL（堆栈满）状态位和 STKUNF（堆栈下溢）状态位。堆栈指针值可为 0 到 15 之间的整数。向堆栈压入值前，堆栈指针加 1；而从堆栈弹出值后，堆栈指针减 1。复位时，堆栈指针值为 0。用户可以读写堆栈指针的值。实时操作系统（Real-Time Operating System,RTOS）可以利用此特性对返回堆栈进行维护。

当向堆栈压入 PC 值 16 次（且没有值从堆栈弹出）后，STKFUL 位就会置 1。通过软件或 POR 使 STKFUL 位清零。

堆栈满时执行的操作由 STVREN（堆栈上溢复位使能）配置位的状态决定。如果 STVREN 位已经置 1（默认），第 31 次进栈将把（PC+2）值压入堆栈，将 STKFUL 位置 1，并复位器件。STKFUL 位将保持置 1，而堆栈指针将被清零。如果 STVREN 位被清零，第 31 次进栈时 STKFUL 位会被置 1，堆栈指针则加 1 变为 15。任何其他进栈操作都不会覆盖第 31 次进栈的值，并且 STKPTR 将保持 15。当堆栈弹出次数足够卸空堆栈时，下一次出

栈会向 PC 返回一个零值，并将 STKUNF 位置 1，而堆栈指针则保持为 0。STKUNF 位将保持置 1，直到被软件清零或发生 POR。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKPTR	STKFUL	STKUNF-	-	SP4	SP3	SP2	SP1	SP0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	-	0	0	0	0	0

BIT[7] STKFUL – 堆栈满标志位

 1: 堆栈满或上溢

 0: 堆栈未满或未上溢

BIT[6] STKUNF – 堆栈下溢标志位

 1: 发生堆栈下溢

 0: 未发生堆栈下溢

BIT[4:0] SP4:SP0 – 堆栈指针地址位

2.2.3 快速寄存器堆栈

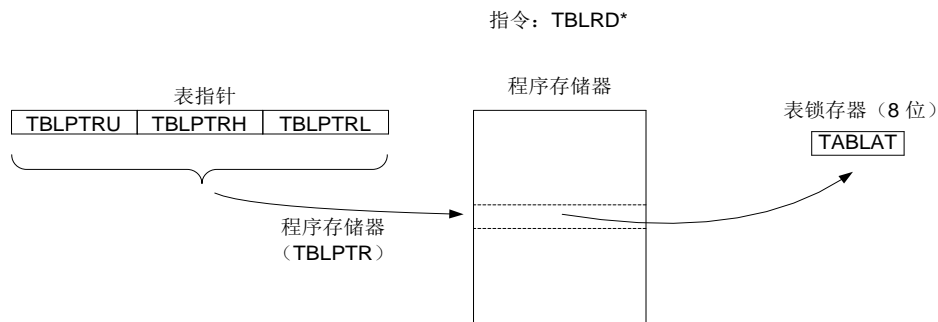
为 Status、WREG 和 BSR 寄存器提供的快速寄存器堆栈具有从中断“快速返回”的功能。每个寄存器堆栈的深度仅为 1 级，并且不可读写。当处理器转入中断向量处执行指令时，此堆栈装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用 RETFIE, FAST 指令从中断返回，这些寄存器中的值会被重新装回对应的寄存器。

如果同时使能了低优先级中断和高优先级中断，从低优先级中断返回时，无法可靠地使用堆栈寄存器。如果在为低优先级中断提供服务时，发生了高优先级中断，则低优先级中断存储在堆栈寄存器中的值将被覆盖。在这种情况下，用户必须在低优先级中断期间用软件保存关键寄存器的值。

如果未使用中断优先级，所有中断都可以使用快速寄存器堆栈从中断返回。如果没有使用中断，快速寄存器堆栈可以用于在子程序调用结束后恢复 Status、WREG 和 BSR 寄存器。要将快速寄存器堆栈用于子程序调用，必须执行 CALL lable,FAST 指令将 Status、WREG 和 BSR 寄存器的内容存入快速寄存器堆栈。在调用结束后执行 RETURN,FAST 指令，从快速寄存器堆栈中弹出并恢复这些寄存器的值。

2.2.4 程序存储器

为了读取程序存储器，AD1845 支持表读操作（TBLRD）。程序存储空间为 16 位宽，而数据 RAM 空间为 8 位宽。表读通过一个 8 位寄存器（TABLAT）在这两个存储空间之间移动数据。



表读操作

表锁存器（Table Latch, TABLAT）是映射到 SFR 空间的一个 8 位寄存器。表锁存器用于在程序存储器和数据 RAM 之间传输数据时保存 8 位数据。

表指针（Table Pointer, TBLPTR）在程序存储器中寻址字节。TBLPTR 由 3 个 SFR 寄存器组成：表指针最高字节、表指针次高字节和表指针低字节（TBLPTRH:TBLPTRL）。这 3 个寄存器合起来组成一个 21 位宽的指针。

TBLRD 指令使用表指针寄存器 TBLPTR。利用表操作的四种方法之一，这些指令可以更新 TBLPTR。下表列出了这些操作。

示例	表指针操作
TBLRD*	不修改 TBLPTR
TBLRD*+	TBLPTR 在读后递增
TBLRD*-	TBLPTR 在读后递减
TBLRD+*	TBLPTR 在读前递增

使用 TBLRD 指令执行表指针操作

读取一个闪存程序存储器字：

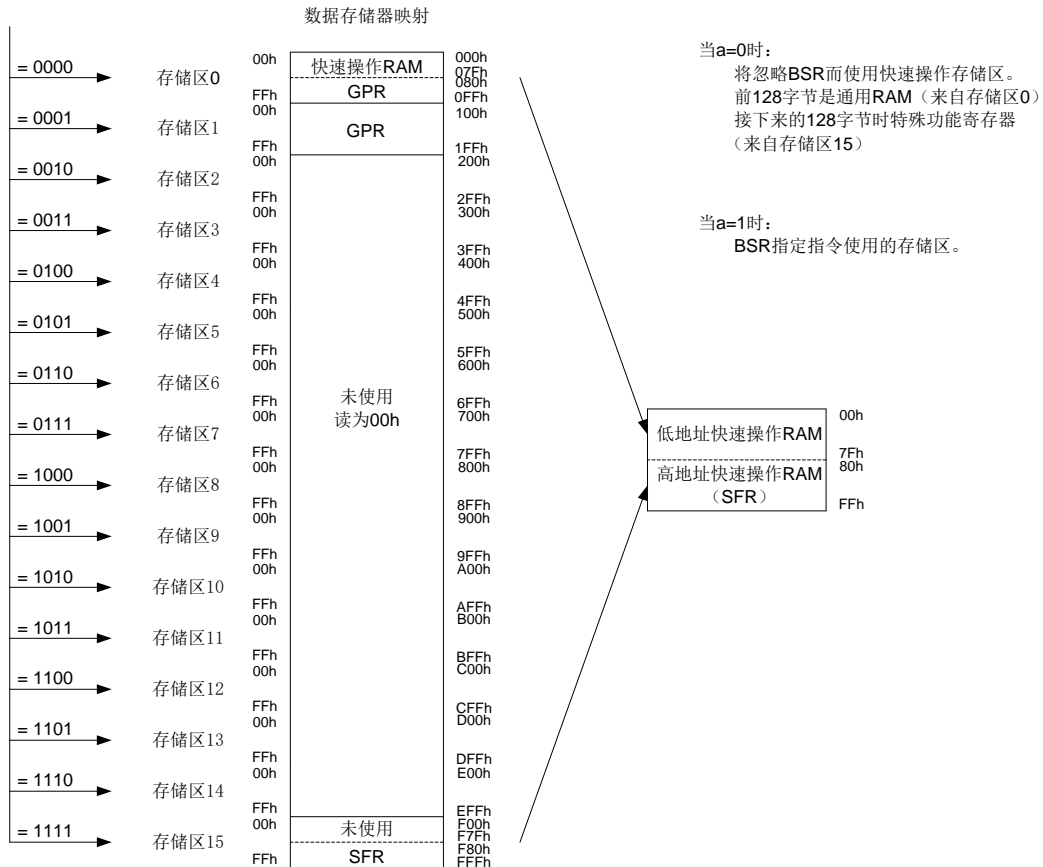
MOVLW	CODE_ADDR_UPPER	;Load TBLPR with the base
-------	-----------------	---------------------------

MOVWF	TBLPTRU	;address of the word
MOVLW	CODE_ADDR_HIGH	
MOVWF	TBLPTRH	
MOVLW	CODE_ADDR_LOW	
MOVWF	TBLPTRL	
READ_WORD		
TBLRD*+		;read into TABLAT and increment
MOVF	TABLAT, W	;get data
MOVWF	WORD_EVEN	
TBLRD*+		
MOVFW	TABLAT, W	;get data
MOVF	WORD_ODD	

2.3 数据存储器 RAM

AD1845 的数据存储器是用静态 RAM 实现的。在数据存储器中，每个寄存器有 12 位地址，数据存储容量可达 768 个字节。存储空间被分为 16 个存储器，每个存储器包含 256 个字节；

数据存储器由特殊功能寄存器（SFR）和通用寄存器（General Purpose Register,GPR）组成。SFR 用于单片机和外设功能模块的控制和状态显示，而 GPR 则用于在用户应用程序中存储数据和高速暂存操作。任何未使用单元的读取值均为 0。



2.4 数据寻址方式

AD1845 支持 4 种寻址方式：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

2.4.1 固有和立即数寻址

很多 AD1845 控制指令根本不需要任何参数，执行这些指令要么对整个器件造成影响，要么仅针对一个寄存器进行操作。这种寻址模式就是固有寻址。例如 SLEEP、RESET 和 DAW 指令。

其他指令的工作方式与此类似但需要操作码中有直接的参数。由于需要一些立即数作为参数，这种寻址模式被称为立即数寻址。例如 ADDLW 和 MOVLW，它们分别将立即数移入

W 寄存器或从中移出。其他的立即数寻址指令，例如 CALL 和 GOTO，包括一个 15 位的程序存储器地址。

2.4.2 直接寻址

直接寻址在操作码中指定操作的全部或部分源地址和/或目标地址。此选项由指令附带的参数指定。

2.4.3 间接寻址

间接寻址允许用户访问数据存储器中的单元而不需要在指令给出一个固定的地址。这是通过使用文件选择寄存器（File Select Register, FSR）指向被读取或写入的单元实现的。由于 FSR 本身作为特殊功能寄存器位于 RAM 中，所以也可以在程序控制下直接对它们进行操作。这使得 FSR 对于在数据存储器中实现诸如表和数组等数据结构非常有用。

也可以使用间接指针操作（Indirect File Operand, INDF）进行间接寻址。这种操作允许自动递增、递减或偏移指针，从而自动控制指针的值。它通过循环提高代码执行效率。

2.4.2.1 FSR 寄存器和 INDF 操作数

间接寻址的核心是三组寄存器：FSR0、FSR1 和 FSR2。每组寄存器都含有一对 8 位寄存器，FSRnH 和 FSRnL。FSRnH 寄存器的高四位未使用，所以每对 FSR 只保存一个 12 位二进制数，从而可以线性寻址整个数据存储空间。因此，FSR 寄存器对被用作数据存储器的地址指针。

间接寻址是通过一组间接指针操作数（从 INDF0 到 INDF2）完成的。这些操作数可以被看作“虚拟”寄存器：它们被映射到 SFR 空间中而不是通过物理方式实现的。对特定的 INDF 寄存器执行读或写操作实际上访问的是相应的 FSR 寄存器对。例如，读 INDF1 就是读 FSR1H:FSR1L 指向的地址单元中的数据。使用 INDF 寄存器作为操作数的指令实际上使用的是相应的 FSR 的内容，该内容为指向目标地址的指针。INDF 操作数只是使用指针的一种较方便的方法。

由于间接寻址使用完整的 12 位地址，因此没有必要进行数据 RAM 分区。因此 BSR 的当前内容和快速操作 RAM 位对于确定目标地址没有影响。

2.4.2.2 FSR 寄存器和 POSTINC、POSTDEC、PREINC 以及 PLUSW

除了 INDF 操作数之外，每对 FSR 寄存器还有四个额外的间接操作数。和 INDF 一样，它们也是不能直接读写的“虚拟”寄存器。访问这些寄存器其实就是访问相关的 FSR 寄存器对，也是在其存储的数据所指向的地址单元上进行特定的操作。

- POSTDEC: 访问 FSR 值，然后自动将它减 1
- POSTINC: 访问 FSR 值，然后自动将它加 1
- PREINC: 将 FSR 的值加 1，然后在操作中使用该值
- PLUSW: 将 W 寄存器中带符号的值（从-127 到 128）与 FSR 寄存器中带符号的值相加，并在操作中使用得到的新值

在应用中使用 FSR 寄存器中的值（不会更改此值）访问 INDF 寄存器。同样，访问 PLUSW 寄存器是将 W 寄存器中的值作为 FSR 值的偏移量，该操作不会改变这两个寄存器中的值。访问其他虚拟寄存器会更改 FSR 寄存器的值。

用 POSTDEC、POSTINC 和 PREINC 对 FSR 进行操作会影响整对寄存器，也就是 FSRnL 寄存器从 FFh 到 00h 溢出并向 FSRnH 寄存器进位。但这些操作的结果不会更改 Status 寄存器中的标志位（如 Z、N 和 OV 等）。

PLUSW 寄存器可以用于在数据存储空间实现变址寻址。通过控制 W 寄存器中的值，用户可以访问相对当前指针地址有固定偏移量的地址单元。在某些应用中，该功能可以被用于在程序存储器内部实现某些非常有用的程序控制结构，如软件堆栈。

2.4.2.3 通过 FSR 对其他 FSR 进行操作

在某些特殊情况下，间接寻址操作以其他 FSR 或虚拟寄存器作为寻址目标。例如，使用 FSR 指向一个虚拟寄存器会导致操作不成功。假设如下特殊情况：FSR0H:FSR0L 保存的是 INDF1 的地址 FE7h。尝试使用 INDF0 作为操作数读取 INDF1 的值，将返回 00h。尝试使用 INDF0 作为操作数写入 INDF1，将会导致执行一条 NOP。

另一方面，使用虚拟寄存器对一对 FSR 寄存器进行写操作可能会产生与预期不同的结果。在这些情形下，会将写入一对 FSR 寄存器，但 FSR 中的值不会有任何递增或递减。因此，写入 INDF2 或 POSTDEC2 时会把同样的值写入 FSR2H:FSR2L。

由于 FSR 是在 SFR 空间中映射的物理寄存器，所以可以通过直接寻址对它们进行操作。

用户在使用这些寄存器时应该特别小心，尤其是在代码使用间接寻址的情况。

同样，通常允许通过间接寻址对所有其他的 SFR 进行操作。用户在进行此类操作时应该特别小心，以免更改设置从而影响器件操作。

2.5 用户配置字 Configuration

配置选项 1L

比特位	名称	描述
3:0	LVT<3:0>	LVR 电压选择 0000: 2.0V 0001: 2.2V 0010: 2.4V 0011: 2.6V 0100: 2.8V 0101: 3.0V 0110: 3.2V 0111: 3.4V 1000: 3.6V 1001: 4.0V 1010: 4.1V 1011: 4.2V 1100: 4.3V 1101: 4.4V 1110: 4.5V 1111: 4.6V
4	LVREN	1: 使能 0: 禁止
7:5	FINTOSC	内部 RC 振荡器频率分频选择 111: 系统时钟1:1 101: 系统时钟1:2 110: 系统时钟1:4 100: 系统时钟1:8 011: 系统时钟1:16 010: 系统时钟1:32 001: 系统时钟1:64 000: 系统时钟1:128

配置选项 2L

比特位	名称	描述
-----	----	----

0	STVEN	堆栈溢出复位 = 1: 使能 = 0, 禁止
1	RESSEL	端口上/下拉电阻选择 0: 190K/300K 1: 30K/30K
2	RESETE	外部复位使能 0: 屏蔽外部复位功能 1: 使能外部复位功能
3	CRYPTON	OTP内容加密 0: 使能加密 1: 禁止加密
4	OTPDIS	禁止OTP读写 1: 打开 0: 禁止
5	IDIS	GPIO输入 1: 打开 0: 禁止
6	CRYEN	振荡器使能选择 1: 禁止外部晶体振荡器 0: 使能外部晶体振荡器
7	CRYSEL	外部晶体振荡器模式选择 1: 外部晶体高频 振荡器 0: 外部晶体低频 振荡器

配置选项 3L

比特位	名称	描述
2:0	TWDT[2:0]	看门狗溢出分频器 Prescale over time 000: 64ms 001: 128ms 010: 192ms 011: 320ms 100: 576ms 101: 1087ms

		110: 211ms 111: 4157ms
3	WDTEN	WDT 使能 0: 关闭 WDT 1: 使能 WDT
4	FCPUS	指令周期选择 0: 1 个指令周期为 2 个机器周期 1: 1 个指令周期为 4 个机器周期
7:5	PSUT[2:0]	上电复位时间选择 111: PWRT= 18ms 110: PWRT= 10ms 101: PWRT=300ms 100: PWRT =75ms 011: PWRT = 466us 010: PWRT = 466us 001: PWRT = 466us 000: PWRT = 466us

配置选项 4L

比特位	名称	描述
3:0	OTP_MODE[3:0]	OTP 工作模式 1111: CPU 访问 16K*16bit 空间 0111: CPU 访问 8K*16bit 空间 0011: CPU 访问 4K*16bit 空间 0001: CPU 访问 2K*16bit 空间 0000: CPU 访问 1K*16bit 空间
5:4	OSCM	内部芯片工作时钟选择 11: 内部HIRC (16Mhz) 10: 内部LIRCH (1M) 01: 内部LIRCL (32Khz)

配置选项 5L

比特位	名称	描述
7:0	OTP_KODE[7:0]	OTP_MODE=1111(16K*16bit 空间)

		1111_1111: CPU 访问 OTP 的 0000-7FFF 空间 OTP_MODE=1110(8K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-3FFF 空间 1111_1110: CPU 访问 OTP 的 4000-7FFF 空间 OTP_MODE=1100(4K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-1FFF 空间 1111_1011: CPU 访问 OTP 的 2000-3FFF 空间 1111_1001: CPU 访问 OTP 的 4000-5FFF 空间 1111_1000: CPU 访问 OTP 的 6000-7FFF 空间 OTP_MODE=1000(2K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-0FFF 空间 1011_1111: CPU 访问 OTP 的 1000-1FFF 空间 1001_1111: CPU 访问 OTP 的 2000-2FFF 空间 1000_1111: CPU 访问 OTP 的 3000-3FFF 空间 1111_0111: CPU 访问 OTP 的 4000-4FFF 空间 1110_0011: CPU 访问 OTP 的 5000-5FFF 空间 1100_0001: CPU 访问 OTP 的 6000-6FFF 空间 1000_0000: CPU 访问 OTP 的 7000-7FFF 空间 OTP_MODE=0000(1K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-07FF 空间 1111_1111: CPU 访问 OTP 的 0800-0FFF 空间 1111_1111: CPU 访问 OTP 的 1000-17FF 空间 1111_1111: CPU 访问 OTP 的 1800-1FFF 空间 1111_1111: CPU 访问 OTP 的 2000-27FF 空间 1111_1111: CPU 访问 OTP 的 2800-2FFF 空间 1111_1111: CPU 访问 OTP 的 3000-37FF 空间 1111_1111: CPU 访问 OTP 的 3800-3FFF 空间 0111_1111: CPU 访问 OTP 的 4000-47FF 空间
--	--	--

		0011_1111: CPU 访问 OTP 的 4800-4FFF 空间 0001_1111: CPU 访问 OTP 的 5000-57FF 空间 0000_1111: CPU 访问 OTP 的 5800-5FFF 空间 0000_0111: CPU 访问 OTP 的 6000-67FF 空间 0000_0011: CPU 访问 OTP 的 6800-6FFF 空间 0000_0001: CPU 访问 OTP 的 7000-77FF 空间 0000_0000: CPU 访问 OTP 的 7800-7FFF 空间
--	--	--

配置选项 5H

比特位	名称	描述
6:0	OTP_KODE[14:8]	OTP_MODE=1111(16K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-7FFF 空间 OTP_MODE=1110(8K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-3FFF 空间 1111_1111: CPU 访问 OTP 的 4000-7FFF 空间 OTP_MODE=1100(4K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-1FFF 空间 1111_1111: CPU 访问 OTP 的 2000-3FFF 空间 1111_1111: CPU 访问 OTP 的 4000-5FFF 空间 1111_1111: CPU 访问 OTP 的 6000-7FFF 空间 OTP_MODE=1000(2K*16bit 空间) 1111_1111: CPU 访问 OTP 的 0000-0FFF 空间 1111_1111: CPU 访问 OTP 的 1000-1FFF 空间 1111_1111: CPU 访问 OTP 的 2000-2FFF 空间 1111_1111: CPU 访问 OTP 的 3000-3FFF 空间 1111_1111: CPU 访问 OTP 的 4000-4FFF 空间 1111_1111: CPU 访问 OTP 的 5000-5FFF 空间 1111_1111: CPU 访问 OTP 的 6000-6FFF 空间 1111_1111: CPU 访问 OTP 的 7000-7FFF 空间

		OTP_MODE=0000(1K*16bit 空间) 111_1111: CPU 访问 OTP 的 0000-07FF 空间 011_1111: CPU 访问 OTP 的 0800-0FFF 空间 001_1111: CPU 访问 OTP 的 1000-17FF 空间 000_1111: CPU 访问 OTP 的 1800-1FFF 空间 000_0111: CPU 访问 OTP 的 2000-27FF 空间 000_0011: CPU 访问 OTP 的 2800-2FFF 空间 000_0001: CPU 访问 OTP 的 3000-37FF 空间 000_0000: CPU 访问 OTP 的 3800-3FFF 空间 000_0000: CPU 访问 OTP 的 4000-47FF 空间 000_0000: CPU 访问 OTP 的 4800-4FFF 空间 000_0000: CPU 访问 OTP 的 5000-57FF 空间 000_0000: CPU 访问 OTP 的 5800-5FFF 空间 000_0000: CPU 访问 OTP 的 6000-67FF 空间 000_0000: CPU 访问 OTP 的 6800-6FFF 空间 000_0000: CPU 访问 OTP 的 7000-77FF 空间 000_0000: CPU 访问 OTP 的 7800-7FFF 空间
--	--	--

控制寄存器

AD1845 的全部寄存器列在下表中，具体功能详见各功能模块的说明。

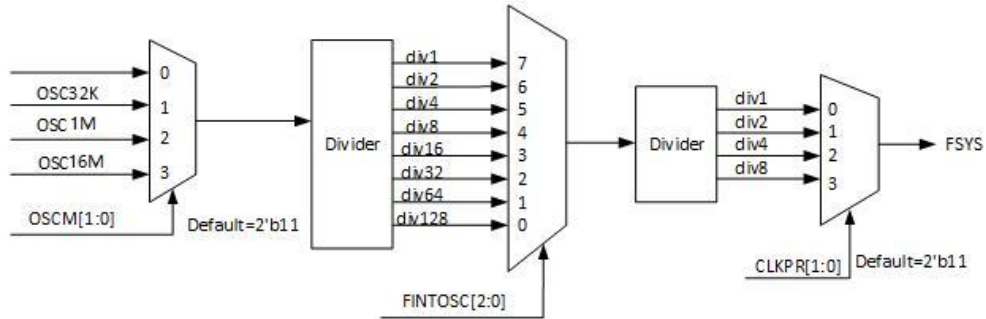
0xfff	TOSU								
0xffe	TOSH						TOS[14:8]		
0xffd	TOSL					TOS[7:0]			
0xffc	STKPTR	STKFUL	STKUNF		SP4	SP3	SP2	SP1	SPO
0xffb	PLLATU								
0xffa	PLCATH								PC[14:8]
0xff9	PCL								PC[7:0]
0xff8	TBLPTRU								
0xff7	TBLPTRH								TBLPTR[14:8]
0xff6	TBLPTRL								TBLPTR[7:0]
0xff5	TABLAT								Program Memory TaLEB Lath
0xff4	PRODH								Product Register High Byte
0xff3	PRODL								Product Register Low Byte
0xff2	INTCON	GIE/GIEH	PEIE/GIEL						
0xff1	INTCON1				INTMAP1	INTMAP0	INTEDG2	INTEDG1	INTEDG0
0xff0									
0xfef	INDF								not a physical register
0xfef	POSTINC0								not a physical register
0xfed	POSTDEC0								not a physical register
0xfec	PREINC0								not a physical register
0xfeb	PLUS0								not a physical register
0xfea	FSROH								FSRO[11:8]
0xfe9	FSROL								FSRO[7:0]
0xfe8	WREG								WREG[7:0]
0xfe7	INDF1								not a physical register
0xfe6	POSTINC1								not a physical register
0xfe5	POSTDEC1								not a physical register
0xfe4	PREINC1								not a physical register
0xfe3	PLUS1								not a physical register
0xfe2	FSR1H								FSR1[11:8]
0xfe1	FSR1L								FSR1[7:0]
0xfe0	BSR								BSR[3:0]
0xfdf	INDF2								not a physical register
0xfde	POSTINC2								not a physical register
0xfdd	POSTDEC2								not a physical register
0xfdc	PREINC2								not a physical register
0xfdb	PLUS2								not a physical register
0xfda	FSR2H								FSR2[11:8]
0xfd9	FSR2L								FSR2[7:0]
0xfd8	STATUS			N		O			DC
0xfd7	TMR0H								C
0xfd6	TMR0L								
0xfd5	TOCON	TOCH1	TOCH0	T0CKPS1	T0CKPS0	TOSE	TOSYNC	TMROCS0	TMROON
0xfd4	LVDCON		LVTST		LVDIM[10]			LVT[30]	
0xfd3	WDTCON						prediv[2:0]		SWDTEN
0xfd2	RCON	IPEN	CKOE	LDO2POEN	R1	TO	PD	POR	BOR
0xfd1	TMR1H								
0xfd0	TMR1L								
0xfcf	T1CON	T1CH1	T1CH0	T1CKPS1	T1CKPS0	T1SE	T1SYNC	TM1RCS0	TM1R1ON
0xfce	I2CADD2								I2C SLAVE Address Register
0xfcd	I2CADD3								I2C SLAVE Address Register
0xfcc	I2CACDLY	ACKDLYON							ACKDLY[6:0]
0xfcb	I2CTX								I2C Transmit Register
0xfca	I2CRC								I2C Receive Register
0xfc9	I2CADD0								I2C SLAVE Address Register
0xfc8	I2CADD1								I2C SLAVE Address Register
0xfc7	I2CADDMASK								MSSP Address MASK Register in I2C Slave Mode.
0xfc6	I2CSTAT	SMP1	SMP0	D/A	P	S	R/W	BPRK	BPTY
0xfc5	I2CCON0	WCOL	I2COV	I2CEN		I2CM3	I2CM2	I2CM1	I2CM0
0xfc4	I2CCON1	GCEN	ACKSTAT	ACKGET	ACKEN				
0xfc3	I2CIE				I2C FLOATIE	I2C STOPTIE	I2C STARTIE	I2C TXIE	I2C RCIE
0xfc2	I2CIF				I2C FLOATIF	I2C STOPIF	I2C STARTIF	I2C TXIF	I2C RCIF
0xfc1	I2CIP				I2C FLOATIP	I2C STOPIP	I2C STARTIP	I2C TXIP	I2C RCIP
0xfc0	CCPR4H								Capture/Compare/PWM Register 4 High Byte
0xfb9	CCPR4L								Capture/Compare/PWM Register 4 Low Byte
0xfb8	CCPR3H								Capture/Compare/PWM Register 3 High Byte
0xfb7	CCPR3L								Capture/Compare/PWM Register 3 Low Byte
0xfb6	CCPR2H								Capture/Compare/PWM Register 2 High Byte
0xfb5	CCPR2L								Capture/Compare/PWM Register 2 Low Byte
0xfb4	CCPR1H								Capture/Compare/PWM Register 1 High Byte
0xfb3	CCPR1L								Capture/Compare/PWM Register 1 Low Byte
0xfb2	CCPCON4								CCPR1L[11:8]
0xfb1	CCPCON5								CCPR2L[11:8]
0xfb0	CCPCON6								CCPR3L[11:8]
0xfaf	CCPCON7								CCPR4L[11:8]
0xfae	CCPCON0	BPSK			CCP1CH[3:0]				CCP1M[3:0]
0xfaq	CCPCON1	BPSK			CCP2CH[3:0]				CCP2M[3:0]
0xfad	CCPCON2	BPSK			CCP3CH[3:0]				CCP3M[3:0]
0xfae	CCPCON3	BPSK			CCP4CH[3:0]				CCP4M[3:0]
0xfaf	SPBRG								Rate Generator Register Byte
0xfaf	RCREG								RC Receive Register
0xfaf	RXREG								RX Transmit Register
0xfad	TXSTA	CSRC	TX9	TXEN	SYNC	TX PARITY	BRGH	TMRT	TX9D
0xfac	RCSTA	SPEN	RX9	SREN	CREN	BGRM	FERR	OERR	RX9D
0xfab	UARTCON0	INVMODE	PARITY	ODD EVEN	BOUD CLK	HALF DUPLEX	DLSB		STOP BIT[1:0]
0xfaa	ADRESH								A/D Result Register High Byte
0xfa9	ADRESL								A/D Result Register Low Byte
0xfa8	ADCON0	ADFM	ADSP	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
0xfa7	ADCON1	ADVREF1	ADVREF0	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
0xfa6	ANASEL0	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
0xfa5	ANASEL1						ANS10	ANS9	ANS8
0xfa4									
0xfa3	IPR1	RCIP	TXIP				TMR2IP	TMR1IP	TMR0IP
0xfa2	PIR1	RCIF	TXIF				TMR2IF	TMR1IF	TMR0IF
0xfa1	PIE1	RCIE	TXIE				TMR2IE	TMR1IE	TMR0IE
0xfa0	IPR2	PCIP	PBIP	PAIP	INT2IP	INT1IP	INT0IP		
0xf9f	PIR2	PCIF	PBIF	PAIF	INT2IF	INT1IF	INT0IF		
0xf9e	PIE2	PCIE	PBIE	PAIE	INT2IE	INT1IE	INT0IE		
0xf9d	IPR3	CCADCIP	ADIP	CCP3PIP	CCP2PIP	CCP1IP	CCP4IP		LVDIP
0xf9c	PIR3		ADIF	CCP3PIF	CCP2PIF	CCP1IF	CCP4IF		LVDIF
0xf9b	PIE3		ADIE	CCP3PIE	CCP2PIE	CCP1IE	CCP4IE		LVDIE
0xf9a									

0xf99	SMTB	PB7SMT				PB3SMT	PB2SMT	PB1SMT	PB0SMT
0xf98	SMTA	PA7SMT	PA6SMT	PA5SMT	PA4SMT			PA1SMT	PA0SMT
0xf97									
0xf96									
0xf95	PBOD	PB7OD				PB3OD	PB2OD	PB1OD	PB0OD
0xf94	PAOD	PA7OD	PA6OD	PA5OD	PA4OD			PA1OD	PA0OD
0xf93									
0xf92									
0xf91	TRISB	TRISA7				TRISB3	TRISB2	TRISB1	TRISB0
0xf90	TRISA	TRISA7	TRISA6	TRISA5	TRISA4			TRISA1	TRISA0
0xf8f									
0xf8e									
0xf8d	PBPD	PB7PD				PB3PD	PB2PD	PB1PD	PB0PD
0xf8c	PAPD	PA7PD	PA6PD	PA5PD	PA4PD			PA1PD	PA0PD
0xf8b									
0xf8a									
0xf89	PTNB					PORTB PIN Status			
0xf88	PTNA					PORTA PIN Status			
0xf87									
0xf86									
0xf85	PBPU	PB7PU				PB3PU	PB2PU	PB1PU	PB0PU
0xf84	PAPU	PA7PU	PA6PU	PA5PU	PA4PU			PA1PU	PA0PU
0xf83									
0xf82									
0xf81	PORTB	PB7				PB3	PB2	PB1	PB0
0xf80	PORTA	PA7	PA6	PA5	PA4			PA1	PA0

0xf77	CADCSRA	CADEN	CADPOL	CADUB	CADAS1	CADAS0	CADS11	CADS10	CADSO
0xf76	CADCSRB	CADIP	CADACIE	CADREIE	CADICIE		CADACIF	CADRCIF	CADICIF
0xf75	CADCSRC							CADBUFEN	CADVSE
0xf74									
0xf73	CADIGH					CADIC[15:8]			
0xf72	CADIGL					CADIC[7:0]			
0xf71	CADAG3					CADAC[31:24]			
0xf70	CADAC2					CADAC[23:16]			
0xf6f	CADAC1					CADAC[15:8]			
0xf6e	CADAC0					CADAC[7:0]			
0xf6d	CADRC					CADRC[7:0]			
0xf6c	CADRD					CADRD[7:0]			
0xf6b	PWST0					PWOFST[7:0]			
0xf6a	PWST1					PWOFST[7:0]			
0xf69	PWST2					PWOFST[7:0]			
0xf68	PWST3					PWOFST[7:0]			
0xf67	PWM2CON0	PWM27POL	PWM26POL	PWM25POL	PWM24POL	PWM23POL	PWM22POL	PWM21POL	PWM20POL
0xf66	PWM2CON1	PWM27OE	PWM26OE	PWM25OE	PWM24OE	PWM23OE	PWM22OE	PWM21OE	PWM20OE
0xf65	PWM2CON2					PWM21DLY[7:0]			
0xf64	PWM2CON3					PWM23DLY[7:0]			
0xf63	PWM2CON4					PWM25DLY[7:0]			
0xf62	PWM2CON5					PWM27DLY[7:0]			
0xf61	PWM2CON6	PWM2ACF	PWM2ASF	LVDF1TEN			IOLF1TEN	ASTART	ACLOSE
0xf60	PWM2CON7	PWM2ADPOS	PWM2ADEN	PWM24ADPOS	PWM24ADEN	PWM22ADPOS	PWM22ADEN	PWM20ADPOS	PWM20ADEN
0xf5f	PWM2CON8					PWMADDLY[7:0]			
0xf5e	DUMMY_3V					DUMMY_3V[7:0]			
0xf5d									
0xf5c	STGCON		LD1VOSC[1:0]		OSCM[1:0]		FINTOSC[2:0]		SIGS
0xf5b	PAINTMASK				PC4MSK	PC3MSK	PC2MSK	PC1MSK	PC0MSK
0xf5a	PBINTMASK	PB7MSK	PB6MSK	PB5MSK	PB4MSK	PB3MSK	PB2MSK	PB1MSK	PB0MSK
0xf59	PCINTMASK	PA7MSK	PA6MSK	PA5MSK	PA4MSK	PA3MSK	PA2MSK	PA1MSK	PA0MSK
0xf58	CLKPR	CLKPCE						CLKPS1	CLKPS0
0xf57	SMCR					SM2	SM1	SM0	SE
0xf56	CLKCFG0		UARTCLKEN		T2CLKEN	CCPCLKEN	T2CLKEN	T1CLKEN	TOCLKEN
0xf55	CLKCFG1							CCADCCLEN	ADCCLEN
0xf54	TMR2					TMR2L[7:0]			
0xf53	PR2					PR2L[7:0]			
0xf52	T2CON	PWM2CAEN	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
0xf51	TMR2H					TMR2H[3:0]			
0xf50	PR2H					PR2H[3:0]			
0xf4f	DIVDENDH								
0xf4e	DIVDENDL								
0xf4d	DIVISORH								
0xf4c	DIVISORL								
0xf4b	QUOTIENTH								
0xf4a	QUOTIENTL								
0xf49	REMAINDERH								
0xf48	REMAINDERL								
0xf47	DIVCON	DIVEN					DIV0	BUSY	START
0xf39	FCSR0		DCE2	DCE1	DCE0		CE2	CE1	CE0
0xf38	FCSR1		DPB7	DPB2	DPB1		CPA4	CPB3	CPB0
0xf37	FVR1CON2		BAT1SEL	BAT1EN	BATOEN		BUFSEL[2:0]		BUFEN
0xf36	FVR1CON0						FVRROUTEN	FVRPGA	FVREN
0xf35	FVR1CON1					FVR_SEL4	FVR_SEL3	FVR_SEL2	FVR_SEL1
0xf34	CRC16IN								FVR_SEL0
0xf33	CRC16H					CRC16IN[7:0]			
0xf32	CRC16L					CRC16H[7:0]			
0xf31						CRC16L[7:0]			
0xf30									
0xf2f	IAPTRIG	0x18, 0x01, 0xc0							
0xf2e	IAPCTRL							LOCK	PG
0xf2d	IAPADDRH					IAPADDR[15:8]			
0xf2c	IAPADDRL					IAPADDR[7:0]			
0xf2b	IAPDATL					IAPDAT[15:8]			
0xf2a	IAPDATL					IAPDAT[7:0]			
0xf29	IAPWAIT					IAPWAIT[7:0]			

3 系统时钟

AD1845 系统时钟图



由内置高精度 RC 电路产生的振荡时钟信号，经由 CLKPR 寄存器控制产生系统主时钟 Fsys。另有一个低速 RC 振荡器专供 WDT（看门狗）电路使用。当选择内部高频振荡器模式时，系统使能内部高频 RC 振荡器作为主时钟。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKPR	CLKPCE	-	-	-	-	-	CLKPS1	CLKPS0
R/W	R/W	-	-	-	-	-	R/W	R/W
初始值	0	-	-	-	-	-	1	1

BIT[7]

CLKPCE – 时钟预分频器切换使能位

1: 允许更新 CLKPS 位 (CLKPCE 更新时, CLKPS[1:0] 必须同时写入 2'b00, 4 个时钟周期后自动清零)

0: 禁止更新 CLKPS 位

BIT[1:0]

CLKPS1:CLKPS0 – 时钟预分频器选择位

00: 系统时钟

01: 系统时钟 2 分频

10: 系统时钟 4 分频

11: 系统时钟 8 分频

0xf56	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCFG0	-	UARTCLK EN	-	I2CCLKEN	CCPCLKEN	T2CLKEN	T1CLKEN	TOCLKEN
R/W		R/W	-	R/W	R/W	R/W	R/W	R/W
初始值		1	-	1	1	1	1	1

BIT[6]

UARTCLKEN -UART 的时钟使能, 使能后 UART 才可以工作

1: 允许使能

0: 禁止使能

BIT[4]

I2CCLKEN-I2C 的时钟使能, 使能后 I2C 才可以工作

1: 允许使能

0: 禁止使能

- BIT[3]** **CCPCLKEN**- CCP 的时钟使能，使能后 CCP 才可以工作
 1: 允许使能
 0: 禁止使能
- BIT[2]** **T2CLKEN**-TIMER2 的时钟使能，使能后 TIMER2 才可以工作
 1: 允许使能
 0: 禁止使能
- BIT[1]** **T1CLKEN**-TIMER1 的时钟使能，使能后 TIMER1 才可以工作
 1: 允许使能
 0: 禁止使能
- BIT[0]** **T0CLKEN**-TIMER0 的时钟使能，使能后 TIMER0 才可以工作
 1: 允许使能
 0: 禁止使能

0xf55	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCFG1	-	-	-	-	-	-	CCADCCLKEN	ADCCLKEN
R/W		-	-	-	-	-	R/W	R/W
初始值		-	-	-	-	-	1	1

- BIT[1]** **CCADCCLKEN**-CCADC 的时钟使能，使能后 CCADC 才可以工作
 1: 允许使能
 0: 禁止使能
- BIT[0]** **ADCCLKEN**-ADC 的时钟使能，使能后 ADC 才可以工作
 1: 允许使能
 0: 禁止使能

3.1 外部晶振

3.2 内置高精度 RC 振荡

当选用内置 RC 振荡时，AD1845 的内置高精度 RC 振荡器有 1MHZ、16MHZ 可选。

SIGCON 为内部 RC 时钟选择寄存器。

0xF5C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SIGCON	LDIVOSC1	LDIVOSC0	OSCM1	OSCM0	FINTOSC2	FINTOSC1	FINTOSC0	SIGS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

- Bit[7:6]**: 内部 LIRCH (1M) 的分频
 00: 4 分频
 01: 8 分频
 10: 16 分频
 11: 32 分频

- Bit[5:4]** : 内部芯片工作时钟选择
 11: 内部 HIRC (16Mhz) (默认)
 10: 内部 LIRCH (1M)
 01: 内部 LIRCL (32Khz)

- Bit[3:1]** : 内部 RC 振荡器频率分频选择

111: 系统时钟1:1
101: 系统时钟1:2
110: 系统时钟1:4
100: 系统时钟1:8
011: 系统时钟1:16
010: 系统时钟1:32
001: 系统时钟1:64
000: 系统时钟 1:128

Bit[0] : 读取配置字中的校准信息

0: 禁止读取配置字中的校准信息

1: 读取配置字中的校准信息

3.3 WDT 振荡器

AD1845 内置一个低频的 RC 振荡器，WDT 的基本溢出时间由相应配置位决定，若要更长的时间可对 WDT 时钟进行分频，分频后 WDT 溢出为基本时间的分频倍，例如基本时间选 18ms,然后软件配置为 4分频，则溢出时间为 $18*4=72ms$ 。该振荡器仅供给WDT电路使用而不能作为系统主时钟用。关于 WDT 振荡器配置可以参考定时器的章节。(定时器 6.4)

4 复位

AD1845 具有 8 种不同的复位方式：

- 1) 上电复位 (POR)
- 2) 正常工作状态下的 $\overline{\text{RSTn}}$ 复位
- 3) 功耗管理模式下的 $\overline{\text{RSTn}}$ 复位
- 4) 看门狗定时器 (WDT) 复位 (执行程序期间)
- 5) 可编程欠压复位 (BOR)
- 6) RESET 指令
- 7) 堆栈满复位
- 8) 堆栈下溢复位

4.1 RCON 寄存器

可通过 RCON 寄存器跟踪器件复位事件。该寄存器的低 5 位表示特定的复位事件是否已经发生。在大部分情况下，只能通过事件将这些位清零，而且必须在随后的应用程序中将它们置 1。读这 5 位标志位可以知道刚刚发生过的复位的类型。

RCON 寄存器中还有用于设置中断优先级的控制位 (IPEN)。

0xfd2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCON	IPEN	CKOE	LDO2P0EN	RI	TO	PD	POR	BOR
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
初始值	0	0	0	0	0	0	x	x

BIT[7]

IPEN – 中断优先级使能位

- 1: 使能高中断优先级
- 0: 禁止高中断优先级

BIT[6]

CKOE – 时钟输出使能位

- 1: 通过 PA5 输出系统时钟
- 0: 禁止输出系统时钟

BIT[6]

LDO2P0EN – LDO2P0 输出使能位

- 1: 使能 LDO2P0
- 0: 禁止 LDO2P0

BIT[4]

RI – RESET 指令标志位

- 1: 未执行 RESET 指令 (只能由固件置 1)
- 0: 已执行 RESET 指令, 导致器件复位 (必须在欠压复位发生之后用软件置 1)

BIT[3]

TO – 看门狗超时标志位

- 1: 通过上电、CLRWDT 指令或 SLEEP 指令置 1
- 0: 发生了 WDT 超时

BIT[2]

PD – 掉电检测标志位

	1: 通过上电或 CLRWDT 指令置 1
	0: 通过执行 SLEEP 指令置 0
<i>BIT[1]</i>	POR – 上电复位状态位
	1: 未发生上电复位（只能用固件置 1）
	0: 已发生上电复位（必须在发生上电复位后由软件置 1）
<i>BIT[0]</i>	BOR – 欠压复位状态位
	1: 未发生欠压复位（只能用固件置 1）
	0: 已发生欠压复位（必须在欠压复位后由软件置 1）

4.2 主清零 (RSTn)

RSTn 引脚提供了用外部硬件触发器复位的方法。将该引脚拉低可以产生复位信号。这些器件在 RSTn 复位路径上有一个噪声滤波器，该滤波器检测并滤除小的干扰脉冲。

任何内部复位，包括 WDT 复位，都不能将 RSTn 引脚驱动为低电平。

4.3 上电复位 (POR)

只要当 VDD 超过设定的门限值后，就会在片上产生上电复位脉冲。这使得 VDD 达到满足器件正常工作的数值时，器件会初始化并启动。

POR 事件由 POR 位 (RCON<1>) 捕获。每当发生 POR 时，该位的状态就会被置为 0，任何其他的复位事件均不能改变它。POR 不能被硬件置为 1，用户必须在 POR 之后用软件将该位置为 1。

4.4 欠压复位 (BOR)

当 VDD 跌落到 VBOR 以下的时间超过 TBOR 时就会复位器件。如果 VDD 降到 VBOR 以下的时间小于 TBOR，可能不一定发生复位。芯片将保持掉电复位状态，直至 VDD 电压上升到 VBOR 以上。

如果使能上电延时定时器，则它将在 VDD 上升到超过 VBOR 之后开始工作，并使芯片在延时 TPWRT 期间保持复位。如果在上电延时定时器运行过程中，VDD 电压降到 VBOR 以下，芯片将重新回到欠压复位状态且将初始化上电延时定时器。一旦 VDD 电压上升 VBOR 以上，上电延迟定时器将重新执行延时操作。BOR 和上电延时定时器 TPWRT 是分别配置的。使能 BOR 复位并不会自动使能 PWRT。

5 I/O 口

5.1 I/O 工作模式

AD1845 有 11 个通用双向 I/O 口 (PA0~PA1、PA4~PA7、PB0~PB3、PB7)。每个 IO 口都有相应的数据寄存器 (PORTA、PORTB)、数据锁存器 (PINA、PINB) 和方向寄存器 (TRISA、TRISB) 控制, 功能如下表所示。

通过寄存器可选输入施密特触发电平。

R/W	TRISA/TRISB	功能
W	1	IO 口处于输入状态; 数据写到数据寄存器中, 端口状态不受影响
W	0	IO 口处于输出状态; 数据写到数据寄存器中, 端口状态与数据寄存器同时改变

0xf80	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	PORTA7	PORTA6	PORTA5	PORTA4	-	-	PORTA1	PORTA0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	0	1	0	0	-	-	1	1

BIT[7:0] **PORTAn** – PORTA口数据位 (n=7-0)

0xf81	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	PORTB7	-	-	-	PORTB3	PORTB2	PORTB1	PORTB0
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
初始值	0	-	-	-	0	0	0	0

BIT[7:0] **PORTBn** – PORTB口数据位 (n=7-0)

0xf90	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	-	-	TRISA1	TRISA0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	1	1	1	1	-	-	1	1

BIT[5:0] **TRISAn** – PORTA口方向寄存器 (n=7-0)

0xf91	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISB	TRISB7	-	-	-	TRISB3	TRISB2	TRISB1	TRISB0
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
初始值	1	-	-	-	1	1	1	1

BIT[7:0] **TRISBn** – PORTB 口方向寄存器 (n=7-0)

0xf88	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PINA	PINA7	PINA6	PINA5	PINA4	-	-	PINA1	PINA0
R/W	R	R	R	R	-	-	R	R
初始值	0	0	0	0	-	-	0	0

BIT[5:0] **PINAn** – PORTA 数据锁存位 (n=7-0)

0xf89	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PINB	PINB7	-	-	-	PINB3	PINB2	PINB1	PINB0
R/W	R	-	-	-	R	R	R	R
初始值	0	-	-	-	0	0	0	0

BIT[7:0] **PINBn** – PORTB 数据锁存位 (n=7-0)

5.2 上拉/下拉电阻和开漏

PA、PB 和 PC 口可通过相应上拉控制寄存器 (PAPU、PBPU、PCPU)、下拉控制寄存器 (PAPD、PBPD、PCPD) 和开漏控制寄存器 (PAOD、PBOD、PCOD) 来控制端口状态。

0xf84	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAPU	PA7PU	PA6PU	PA5PU	PA4PU	-	-	PA1PU	PA0PU
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	1	1	1	1	-	-	1	1

BIT[7:0] **PAnPU** – PA口上拉电阻选择 (n=7-0)

1: PAn 端口上拉电阻无效

0: PAn 端口上拉电阻有效

0xf85	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBPU	PB7PU	-	-	-	PB3PU	PB2PU	PB1PU	PB0PU
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
初始值	1	-	-	-	1	1	1	1

BIT[7:0] **PBnPU** – PB口上拉电阻选择 (n=7-0)

1: PBn 端口上拉电阻无效

0: PBn 端口上拉电阻有效

0xf8c	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAPD	PA7PD	PA6PD	PA5PD	PA4PD	-	-	PA1PD	PA0PD
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	1	1	1	1	-	-	1	1

BIT[7:0] **PAnPD** – PA口下拉电阻选择 (n=7-0)

1: PAn 端口下拉电阻无效

0: PAn 端口下拉电阻有效

0xf8d	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBPD	PB7PD	-	-	-	PB3PD	PB2PD	PB1PD	PB0PD
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
初始值	1	-	-	-	1	1	1	1

BIT[7:0] P_{Bn}PD – PB口下拉电阻选择 (n=7-0)

- 1: P_{Bn} 端口下拉电阻无效
- 0: P_{Bn} 端口下拉电阻有效

0xf94	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAOD	PA7OD	PA6OD	PA5OD	PA4OD	-	-	PA1OD	PA0OD
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	0	1	0	0	-	-	1	1

BIT[7:0] P_{AN}OD – PA口开漏选择 (n=7-0)

- 1: P_{AN} 端口使能开漏输出
- 0: P_{AN} 端口为普通推挽输出口

0xf95	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBOD	PB7OD	-	-	-	PB3OD	PB2OD	PB1OD	PB0OD
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
初始值	0	-	-	-	0	0	0	0

BIT[7:0] P_{Bn}OD – PB口开漏选择 (n=7-0)

- 1: P_{Bn} 端口使能开漏输出
- 0: P_{Bn} 端口为普通推挽输出口

5.3 端口模拟输出控制

通过配置 ANASEL0 寄存器，可以控制端口模拟输出开关的打开和关闭。

0xfa6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANASEL0	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] ANS[7:0] –模拟输出开关控制 (1-打开, 0-关闭)

- ANS0: PA0 模拟输出开关
- ANS1: PA1 模拟输出开关
- ANS2: PA4 模拟输出开关

- ANS3: PA5 模拟输出开关
 ANS4: PA6 模拟输出开关
 ANS5: PA7 模拟输出开关
 ANS6: PB2 模拟输出开关
 ANS7: PB1 模拟输出开关

0xfa5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANASEL1	-	-	-	-	-	ANS10	ANS9	ANS8
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2:0] ANS[10:8] –模拟输出开关控制（1-打开，0-关闭）

- ANS8: PB7 模拟输出开关
 ANS9: PB0 模拟输出开关
 ANS10: PB3 模拟输出开关

0xf98	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMTA	PA7SMT	PA6SMT	PA5SMT	PA4SMT	-	-	PA1SMT	PA0SMT
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	1	1	1	1	-	-	1	1

SMTA – 施密特寄存器

- 0: 0.2VDD/0.4VDD(IO 低电平变为高电平时为 0.4VDD，高电平变为低电平时为 0.2VDD)
 1: 0.3VDD/0.7VDD(IO 低电平变为高电平时为 0.7VDD，高电平变为低电平时为 0.3VDD)

SMTB	PB7SMT	-	-	-	PB3SMT	PB2SMT	PB1SMT	PB0SMT
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
初始值	1	-	-	-	1	1	1	1

SMTB – 施密特寄存器

- 0: 0.2VDD/0.4VDD(IO 低电平变为高电平时为 0.4VDD，高电平变为低电平时为 0.2VDD)
 1: 0.3VDD/0.7VDD(IO 低电平变为高电平时为 0.7VDD，高电平变为低电平时为 0.3VDD)

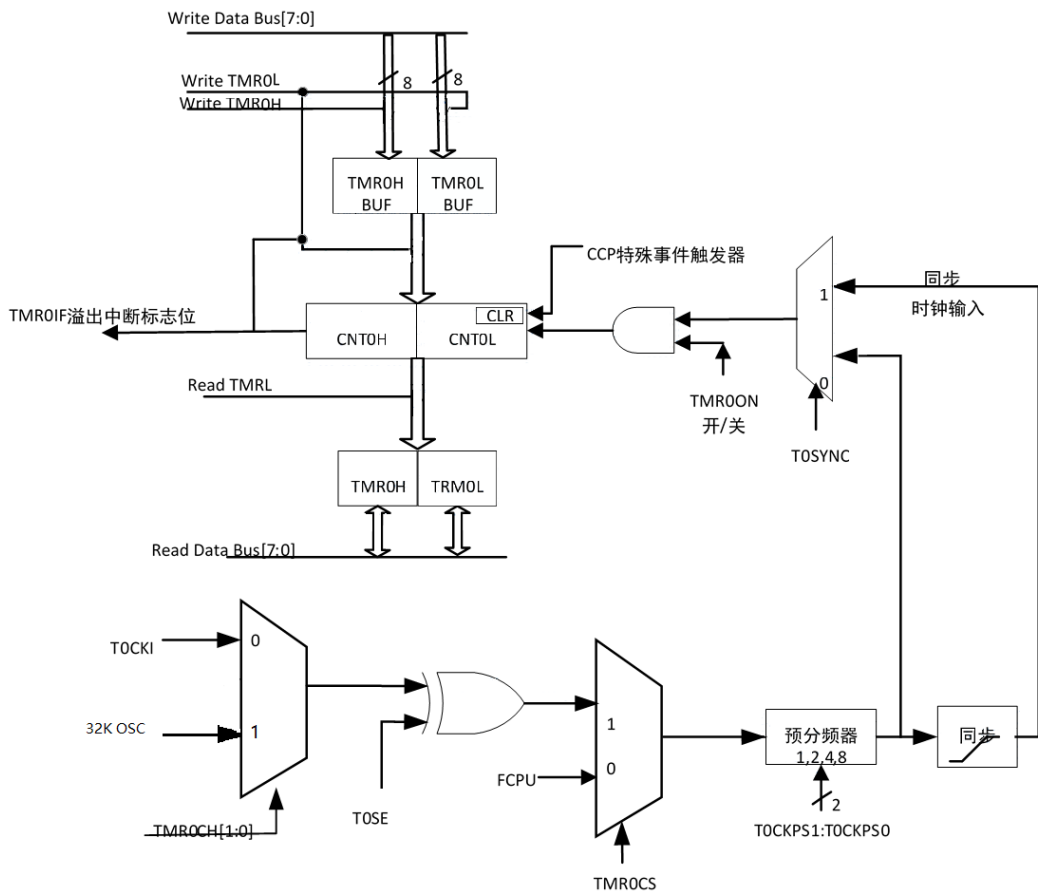
注意事项: PA0、PA1、PA6 上电默认开漏使能;

6 定时器

6.1 Timer0 16 位自动重载定时/计数器

Timer0 可以工作在定时器、同步计数器、异步计数器模式。

Timer0 为 16 位定时/计数器，内部拥有写入缓冲区，当溢出时，自动从写入缓冲区装载到 TIMER0 计数器作为初值，然后每个 TIMER0 时钟周期，计数递增。当 TIMER0 由计数值从 0XFFFF 变为 0X0000 时产生时钟溢出中断，并自动重载写入缓冲区数值。Timer0 的时钟源可以是内部、外部时钟源(T0CKI pin)、内部 32K 时钟等。



写入 16 位初值时，首先写入 TMR0H 寄存器，然后写入 TMR0L，此时硬件自动将写入缓冲区值重载到 Timer0 的计数器中；当读取 TMR0L 时，自动将当前 Timer0 的计数值写入 TMR0H 和 TMR0L 寄存器，保证读取时间准确。

6.1.1 使用内部时钟: 定时模式

定时模式在没有预置器的情况下，定时寄存器每个指令周期自动加 1，设置 TMR0H 和

TMR0L 以后，定时器开始自增。当 TIMER0 从 0XFFFF 加 1 变为 0x0000 时，产生溢出，同时 TMR0 自动重载写入值，开始计数。

6.1.2 使用外部时钟/内部 32K 时钟输出：计数模式

T0CKI 上升或下降沿/32K 时钟/，触发 Timer0 寄存器的增加，触发为由 T0SE 位 (T0CON<3>)决定上升下降触发，可以选择时钟要求与内部时钟(Tosc)是否同步。同步以后，Timer0 实际增加有一个延迟。当不选择同步计数时，可以在低功耗模式下继续工作，低功耗模式会自动关闭内部高速时钟。

当使能 TMR0CH 为 02 时，启动低速 1MHz 值晶体，选择内部 32K OSC 作为时钟输入，可以方便构成实时时钟。

预置分频时器，外部时钟输入被异步分频器平分，这种常用来计算波形。

6.1.3 定时器 TIMER0 相关的寄存器

TMR0 是一个 16 位定时/计数器重载寄存器，当写入初始值后，TIMER0 溢出后，会自动重载前面写入的初始值。

TMR0L (Timer0 16 位低 8 位寄存器)

0xfd6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR0L	16 位定时/计数器低 8 位							
R/W	R/W							
初始值	0x0							

TMR0H (Timer0 16 位高 8 位寄存器)

0xfd7	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR0H	16 位定时/计数器高 8 位							
R/W	R/W							
初始值	0x0							

T0CON (Timer0 控制寄存器)

0xfd5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CON	T0CH1	T0CH0	T0CKPS1	T0CKPS0	T0SE	T0SYNC	TMR0CS	TMR0ON
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

T0CH1、T0CH0 : 00:T0CKI 作为时钟输入

01:选择内部 32K OSC 作为时钟输入

T0CKPS1、T0CKPS0 : 00 : TIMER0 输入时钟 1:1 分频

01 : TIMER0 输入时钟 1:2 分频

10 : TIMER0 输入时钟 1:4 分频

11 : TIMER0 输入时钟 1:8 分频

T0SE : T0CH[1: 0] 时钟异或输入选项

T0SYNC : 1: 采用 FCPU 同步分频后时钟作为 TIMER0 钟（注意不支持选择内部时钟，同时 1:1 分频，并且 T0YNC 为 1）

0: 采用选择分频时钟作为 TIMER0 时钟

TMR0CS 1: 选择 T0CK[1:0]作为外设 TIMER0 的时钟

0: 选择 FCPU 时钟外设 TIMER0 的时钟

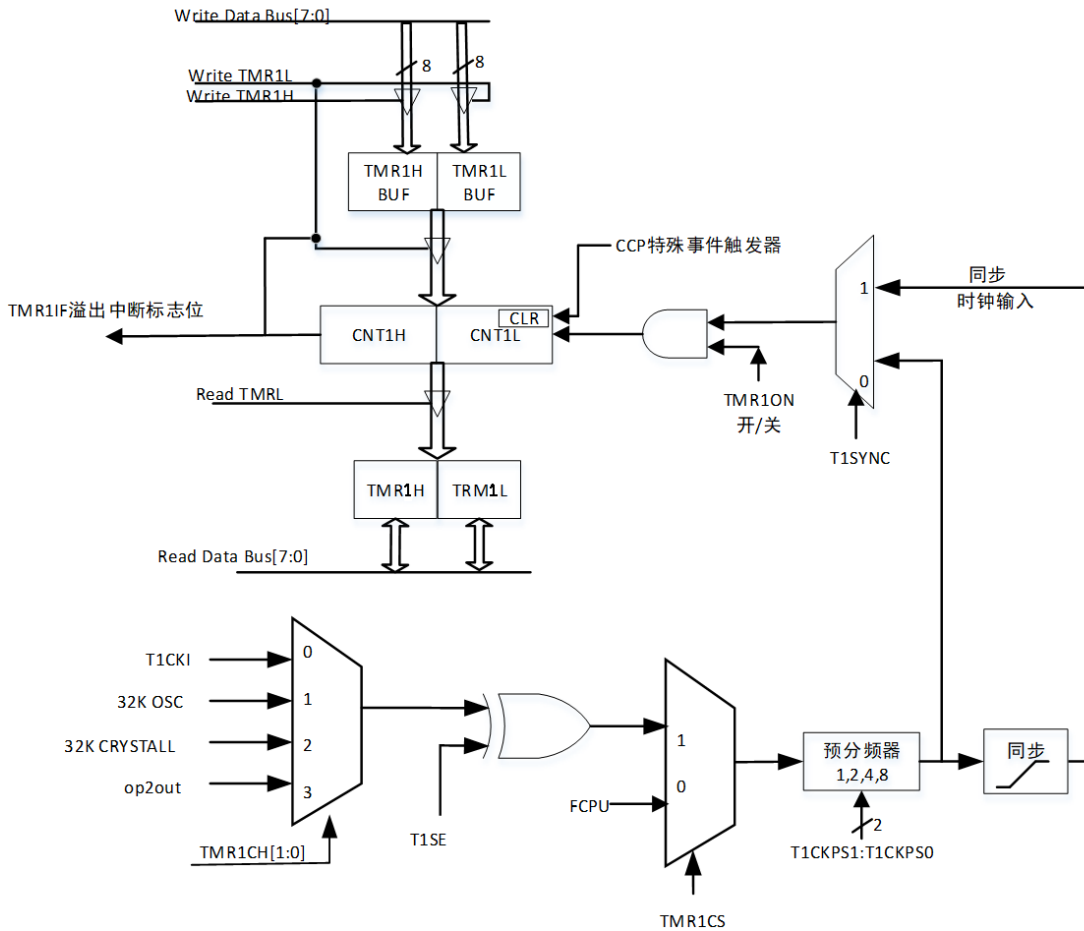
TMR0ON : 1:使能 Timer0 定时计数器

0 :关闭 Timer0 定时计数器

6.2 Timer1 16 位自动重载定时/计数器

Timer1 可以工作在定时器、同步计数器、异步计数器模式。

Timer1 为 16 位定时/计数器，内部拥有写入缓冲区，当溢出时，自动从写入缓冲区装载到 TIMER1 计数器作为初值，然后每个 TIMER1 时钟周期，计数递增。当 TIMER1 由计数值从 0XFFFF 变为 0X0000 时产生时钟溢出中断，并自动重载写入缓冲区数值。Timer1 的时钟源可以是内部、外部时钟源(T1CKI pin)、内部 32K 时钟等。同时 TIMER1 可以与 CCPR1H,CCPR1L 采用比较器模式，产生 3 路 16 位精度的 PWM 输出，输出引脚为 PWM20, PWM21, PWM22 引脚。



写入 16 位初值时，首先写入 TMR1H 寄存器，然后写入 TMR1L，此时硬件自动将写入缓冲区值重载到 Timer1 的计数器中；当读取 TMR1L 时，自动将当前 Timer1 的计数值写入 TMR1H 和 TMR1L 寄存器，保证读取时间准确。

6.2.1 使用内部时钟: 定时模式

定时模式在没有预置器的情况下，定时寄存器每个指令周期自动加 1，设置 TMR1H 和 TMR1L 以后，定时器开始自增。当 TIMER1 从 0XFFFF 加 1 变为 0x0000 时，产生溢出，同时 TMR1 自动重载写入值，开始计数。

6.2.2 使用外部时钟/内部 32K 时钟: 计数模式

选择通过 T1CKI 上升或下降沿/32K 时钟输出，触发 Timer1 寄存器的增加，触发为由 T1SE 位(T1CON<3>)决定上升下降触发，可以选择时钟要求与内部时钟(Tosc)是否同步。同

步以后, Timer1 实际增加有一个延迟。当不选择同步计数时, 可以在低功耗模式下继续工作, 低功耗模式会自动关闭内部高速时钟。

预置分频时器, 外部时钟输入被异步分频器平分, 这种常用来计算波形。

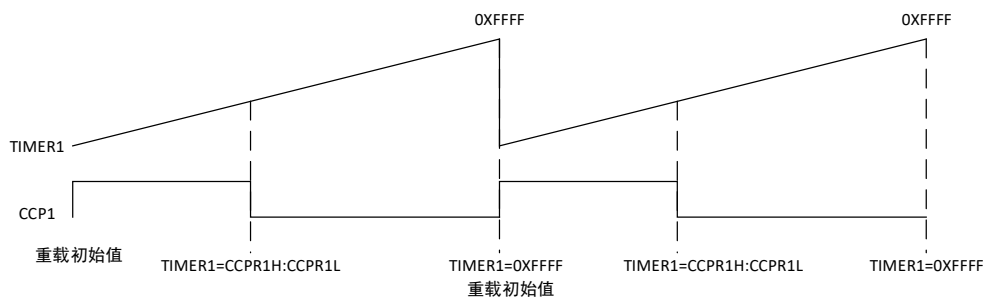
6.2.3 TIMER1 特殊事件

如果 CCP1 模块被配置成为特殊事件触发模式 (CCP1M=1011) 的比较模式, 该位触发信号将复位 Timer1。如果使能了 AD 模式 (ADON=1), 则来自 CCP1 的触发信号还将启动 A/D 转换。

要使用这一功能, 必须将 Timer1 配置为定时器或同步计数器。在这种情况下, CCPR1H:CCPR1L 这对寄存器实际上变成了 Timer1 的周期寄存器。如果 Timer1 在异步计数器模式下运行, 复位操作可能不起作用。如果 Timer1 的写操作和特殊事件触发信号同时发生, 则写操作优先。

6.2.4 3 路 16 位 PWM 输出

如果 CCP1 模块被配置成为比较模式, 同时 Timer1 配置位定时器或同步计数器, CCPR1H:CCPR1L 这对寄存器实际上变成了 Timer1 的占空比控制寄存器, 实现 16 位 PWM 控制输出。同样采用 CCPR2H:CCPR2L, CCPR3H:CCPR3L 构成另外两路 16 位精度 PWM 输出。



6.2.5 定时器 TIMER1 相关的寄存器

TMR1 是一个 16 位定时/计数器重载寄存器, 当写入初始值后, TIMER1 溢出后, 会自动重载前面写入的初始值。

TMR1L (Timer1 16 位低 8 位寄存器)

0xfd0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR1L	16 位定时/计数器低 8 位							
R/W	R/W							
初始值								

TMR1H (Timer1 16 位高 8 位寄存器)

0xfd1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR1H	16 位定时/计数器高 8 位							
R/W	R/W							
初始值								

T1CON (Timer1 控制寄存器)

0xfc	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CON	T1CH1	T1CH0	T1CKPS1	T1CKPS0	T1SE	T1SYNC	TMR1CS	TMR1ON
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

使时钟需要设置 TCON 的 T0CS0、T0CS1 共同决定

T1CH1、T1CH0 : 00: 选择 T1CKI 作为时钟输

01: 择内部 32K OSC 作为时钟输

T1CKPS1、T1CKPS0 : 00: TIMER1 输入时钟 1:1 分频

01: TIMER1 输入时钟 1:2 分频

10: TIMER1 输入时钟 1:4 分频

11: TIMER1 输入时钟 1:8 分频

T1SE : T1SE 时钟异或输入选项

0: 输出时钟与输入时钟相同

1: 输出时钟与输入时钟相反

T1SYNC : 1: 采用 FCPU 同步分频后时钟作为 TIMER1 时钟（注意不支持选择内部时钟，同时 1:1 分频，并且 T1SYNC 为 1）

0: 采用选择分频时钟作为 TIMER1 时钟

TMR1CS 1: 选择 T1CK[1:0]作为外设 TIMER1 的时钟

0: 选择 FCPU 时钟外设 TIMER1 的时钟

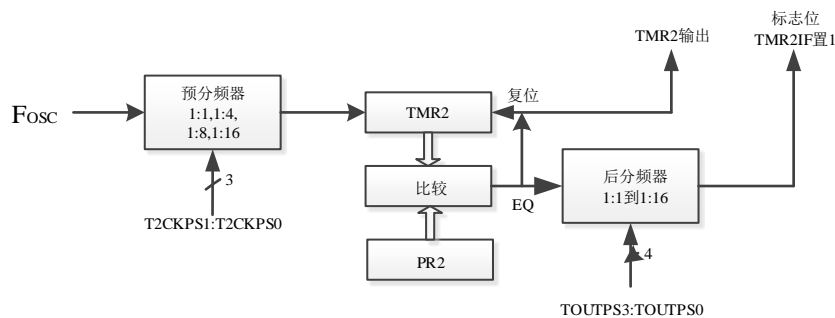
TMR1ON : 1: 使能 Timer1 定时计数器

0: 关闭 Timer1 定时计数器

6.3 TIMER2 模块

Timer2 定时器模块具有以下特征：

- 12 位定时器和周期寄存器（分别为 TMR2 和 PR2）
- 可读写（以上两个寄存器）
- 可软件编程的预分频器（分频比为 1:1、1:4、1:8 和 1:16）
- 可软件编程的后分频器（分频比为 1:1 至 1:16）
- 当 TMR2 与 PR2 匹配时产生中断
- 作为 MSSP 模块的可选移位时钟



T2 结构图

与定时器 T2 相关的寄存器说明如下。

0xf54	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR2L	TMR27	TMR26	TMR25	TMR24	TMR23	TMR22	TMR21	TMR20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] TMR2L[7:0] – Timer2 寄存器低 8 位

0xf51	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR2H	-	-	-	-	TMR2H3	TMR2H2	TMR2H1	TMR2H0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[7:0] TMR2H[3:0] – Timer2 寄存器高 4 位

0xf53	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR2	PR27	PR26	PR25	PR24	PR23	PR22	PR21	PR20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] PR2L[7:0] – Timer2 周期寄存器低 8 位

0xf50	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-------	-------	-------	-------	-------	-------	-------	-------	-------

PR2H	-	-	-	-	PR2H3	PR2H2	PR2H1	PR2H0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	1	1	1	1

BIT[7:0] **PR2H[7:0]** – Timer2 周期寄存器高 4 位

0xf52	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	PWM2CAEN	T2OUTP S3	T2OUTP S2	T2OUTP S1	T2OUTP S0	TMR2ON	T2CKPS 1	T2CKPS 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **PWM2CAEN** – Timer2 自减模式使能位

- 1: 使能 Timer2 自减模式。
- 0: 禁止 Timer2 自减模式。

BIT[6:3] **T2OUTPS3:T2OUTPS0** – Timer2 输出后分频选择位

- 0000: 1:1 后分频值
- 0001: 1:2 后分频值
-
-
-
- 1111: 1:16 后分频值

BIT[2] **TMR2ON** – Timer2 使能位

- 1: 使能 Timer2
- 0: 禁止 Timer2

BIT[1:0] **T2CKPS1:T2CKPS0** – Timer2 时钟预分频选择位

- 00: 预分频值为 1
- 01: 预分频值为 4
- 10: 预分频值为 8
- 11: 预分频值为 16

0xfa1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE1	RCIE	TXIE	-	-	-	TMR2IE	TMR1IE	TMR0IE
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

BIT[1] **TMR2IE** – Timer2 溢出中断允许位

- 1: 允许 Timer2 溢出中断
- 0: 禁止 Timer2 溢出中断

0xfa2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR1	RCIF	TXIF	-	-	-	TMR2IF	TMR1IF	TMR0IF
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

BIT[1] TMR2IF – TMR2 溢出中断标志位

- 1: TMR2 寄存器发生溢出（必须用软件清零）
- 0: TMR2 寄存器没有溢出

0xfa3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR1	RCIP	TXIP	-	-	-	TMR2IP	TMR1IP	TMR0IP
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	1	1	-	-	-	1	1	1

BIT[1] TMR2IP – TMR2 溢出中断优先级

- 1: 高优先级
- 0: 低优先级

6.3.1 TIMER2 自减模式

当 PWM2CAEN (T2CON<7>) 是 TIMER2 自减模式使能位（若此时相应 PWM 开启, PWM 开启中心对齐模式），TIMER2 会在每个周期开始检查该位，即当 PWM2CAEN 置 1 时：

- *若 TIMER2 尚未使能，TIMER2 会在使能后直接进入自减模式。
- *若 TIMER2 正在运行，TIMER2 会在下一个 TIMER2 周期进入自减模式。

TIMER2 自减模式工作流程：

TMR2H:TMR2 从 0 递增至 {PR2H:PR2L}，当 {TMR2H:TMR2L} 与 {PR2H:PR2L} 相等时，{TMR2H:TMR2L} 开始递减至 0，该周期结束。

6.4 WDT 模块

看门狗定时器 (WDT) 的运行依赖于芯片里的 RC 振荡器，无需任何额外电路即能工作。如在睡眠模式。在一般操作或睡眠模式情况下，看门狗定时器的溢出都会导致 MCU 复位同时 TO (STATUS<4>) 位被清零。

如 WDTE 位(PCON<7>)清零. 看门狗定时器不能工作。

在没有预置器时看门狗的溢出约为 64 ms, 128ms, 192ms, 320ms...

这个时间可以通过 TWDT<2:0> 设置。

需要看门狗的 t 溢出周期变长可以通过设置 WDTCON 寄存器的看门狗定时器分频大于 1:128., 因此最长的看门狗溢出周期为 33.26 秒。

CLRWDT 指令能使 WDT 和预置器清零, 启用看门狗可以防止超时, 如果超时 MCU 能复位。

SLEEP 指令重置 WDT 和预置器, 启用看门狗就给机器分派了一个最大睡眠时间。

0xfd3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTCON	-	-	-	-	PREDIV2	PREDIV1	PREDIV0	SWDTEN
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[0] SWDTEN – WDT 使能

1: 使能 WDT

0: 禁止 WDT

BIT[3:1] PREDIV – WDT 预分频

000: 分频值为 1

001: 分频值为 2

010: 分频值为 4

011: 分频值为 8

100: 分频值为 16

101: 分频值为 32

110: 分频值为 64

111: 分频值为 128

7 四路捕捉/比较/ PWM 模块

捕捉/比较/PWM (CCP) 模块包含一个 4 个 16 位寄存器，它可被用作：4 个 16 位捕捉寄存器、4 个 16 位比较寄存器、4 个 PWM 主/从占空比寄存器。捕捉/比较/PWM 寄存器 1 (CCPR1) 由两个 8 位寄存器组成：CCPR1L (低字节) 和 CCPR1H (高字节)。CCP1CON 寄存器控制 CCP1 的操作。比较匹配将产生特殊事件触发信号，该信号会使 TMR1H 和 TMR1L 寄存器清零。

0xfb9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR1 L	CCPR1L 7	CCPR1L 6	CCPR1L 5	CCPR1L 4	CCPR1L 3	CCPR1L 2	CCPR1L 1	CCPR1L 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **CCPR1L** – CCP1 寄存器低字节

0xfba	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR1 H	CCPR1H 7	CCPR1H 6	CCPR1H 5	CCPR1H 4	CCPR1H 3	CCPR1H 2	CCPR1H 1	CCPR1H 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **CCPR1H** – CCP1 寄存器高字节

0xfbb	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR2 L	CCPR2L 7	CCPR2L 6	CCPR2L 5	CCPR2L 4	CCPR2L 3	CCPR2L 2	CCPR2L 1	CCPR2L 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **CCPR2L** – CCP2 寄存器低字节

0xfbc	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR2 H	CCPR2H 7	CCPR2H 6	CCPR2H 5	CCPR2H 4	CCPR2H 3	CCPR2H 2	CCPR2H 1	CCPR2H 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **CCPR2H** – CCP2 寄存器高字节

0xfbd	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR3 L	CCPR3L 7	CCPR3L 6	CCPR3L 5	CCPR3L 4	CCPR3L 3	CCPR3L 2	CCPR3L 1	CCPR3L 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初始值	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

BIT[7:0] **CCPR3L** – CCPR3 寄存器低字节

0xfb8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR3H	CCPR3H	CCPR3H	CCPR3H	CCPR3H	CCPR3H	CCPR3H	CCPR3H	CCPR3H
	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **CCPR3H** – CCPR3 寄存器高字节

0xfb4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR4L	CCPR4L	CCPR4L	CCPR4L	CCPR4L	CCPR4L	CCPR4L	CCPR4L	CCPR4L
	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **CCPR4L** – CCPR4 寄存器低字节

0xf80	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPR4H	CCPR4H	CCPR4H	CCPR4H	CCPR4H	CCPR4H	CCPR4H	CCPR4H	CCPR4H
	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **CCPR4H** – CCPR4 寄存器高字节

0xfb4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON0	BPSK	CCP1CH2	CCP1CH1	CCP1CH0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **BPSK**: 改通道 PWM 支持采用 UART 实现 BPSK 调试信号输出

1: 使能与 UART 进行 BPSK 调制输出

0: 禁止 BPSK 调制输出

BIT[6:4] **CCP1CH[2:0]**: CCP1 捕获触发信号选择

000: 外部 CCP1CH0 引脚输入信号 (PA6)

001: 外部 CCP1CH1 引脚输入信号 (PA1)

010: 外部 CCP1CH2 引脚输入信号 (PA0)

011: 外部 CCP1CH3 引脚输入信号 (PA4)

110: 内部 32K 输入信号

BIT[3:0] CCP1M[3:0] – CCP1 模式选择位

0000: 禁止捕捉/比较/PWM (复位 CCP1 模块)

0100: 比较模式, 选择 CCPR1 匹配时将输出置为高电平 (CCP1IF 位置 1), 输出引脚为 PA4, TIMER1 溢出时, PA4 为低电平

0101: 比较模式, 选择 CCPR1 匹配时将输出置为低电平 (CCP1IF 位置 1), 输出引脚为 PA4, TIMER1 溢出时, PA4 为高电平

0110: 比较模式, 选择 CCPR1 匹配时将产生软件中断 (CCP1IF 位置 1, 而 PA4 引脚不受影响)

0111: 比较模式, 选择 CCPR1 触发特殊事件 (CCP1IF 位置 1, PA4 引脚不受影响); CCP1 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1)

1000: 捕捉模式, 在每个下降沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1001: 捕捉模式, 在每个上升沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1010: 捕捉模式, 在每 4 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1011: 捕捉模式, 在每 16 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1101: 捕捉模式, 上升沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1110: 捕捉模式, 下降沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCP1IF

1111: 捕捉模式, 上升沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCP1IF

001X: PWM 模式, 启动 CCP1 PWM 模式

0xfb3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON1	BPSK	CCP2CH2	CCP2CH1	CCP2CH0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] BPSK: 改通道 PWM 支持采用 UART 实现 BPSK 调试信号输出

1: 使能与 UART 进行 BPSK 调制输出

0: 禁止 BPSK 调制输出

BIT[6:4] CCP2CH[2:0]: CCP2 捕获触发信号选择

000: 外部 CCP1CH0 引脚输入信号 (PA6)

001: 外部 CCP1CH1 引脚输入信号 (PA1)

010: 外部 CCP1CH2 引脚输入信号 (PA0)

011: 外部 CCP1CH3 引脚输入信号 (PA4)

110: 内部 32K 输入信号

BIT[3:0] CCP2M[3:0] – CCP2 模式选择位

0000: 禁止捕捉/比较/PWM (复位 CCP2 模块)

0100: 比较模式, 选择 CCPR2 匹配时将输出置为高电平 (CCP2IF 位置 1), 输出引脚为 PB0, TIMER1 溢出时, PB0 为低电平

0101: 比较模式, 选择 CCPR2 匹配时将输出置为低电平 (CCP2IF 位置 1), 输出引脚为 PB0, TIMER1 溢出时, PB0 为高电平

0110: 比较模式, 选择 CCPR2 匹配时将产生软件中断 (CCP2IF 位置 1, 而 PB0 引脚不受影响)

0111: 比较模式, 选择 CCPR2 触发特殊事件 (CCP2IF 位置 1, PB0 引脚不受影响); C 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1)

1000: 捕捉模式, 在每个下降沿发生, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF

1001: 捕捉模式, 在每个上升沿发生, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF

1010: 捕捉模式, 在每 4 个上升沿发生一次, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF

1011: 捕捉模式, 在每 16 个上升沿发生一次, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF

1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF

1101: 捕捉模式, 上升沿, 复位 TIMER1, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF

1110: 捕捉模式, 下降沿, 捕捉值存入 CCPR2 寄存器, 不产生中断标志 CCP2IF

1111: 捕捉模式, 上升沿, 捕捉值存入 CCPR2 寄存器, 不产生中断标志 CCP2IF

001X: PWM 模式, 启动 CCP2PWM 模式

0xfb2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON2	BPSK	CCP3CH2	CCP3CH1	CCP3CH0	CCP3M3	CCP3M2	CCP3M1	CCP3M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] BPSK: 改通道 PWM 支持采用 UART 实现 BPSK 调试信号输出

1: 使能与 UART 进行 BPSK 调制输出

0: 禁止 BPSK 调制输出

BIT[6:4]

CCP3CH[2:0]: CCP3 捕获触发信号选择

000: 外部 CCP1CH0 引脚输入信号 (PA6)

001: 外部 CCP1CH1 引脚输入信号 (PA1)

010: 外部 CCP1CH2 引脚输入信号 (PA0)

011: 外部 CCP1CH3 引脚输入信号 (PA4)

110: 内部 32K 输入信号

BIT[3:0]

CCP3M[3:0] – CCP3 模式选择位

0000: 禁止捕捉/比较/PWM (复位 CCP3 模块)

0100: 比较模式, 选择 CCPR3 匹配时将输出置为高电平 (CCP3IF 位置 1), 输出引脚为 PB2, TIMER1 溢出时, PB2 为低电平

0101: 比较模式, 选择 CCPR3 匹配时将输出置为低电平 (CCP3IF 位置 1), 输出引脚为 PB2, TIMER1 溢出时, PB2 为高电平

0110: 比较模式, 选择 CCPR3 匹配时将产生软件中断 (CCP3IF 位置 1, 而 PB2 引脚不受影响)

0111: 比较模式, 选择 CCPR3 触发特殊事件 (CCP3IF 位置 1, PB2 引脚不受影响); 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1)

1000: 捕捉模式, 在每个下降沿发生, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF

1001: 捕捉模式, 在每个上升沿发生, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF

1010: 捕捉模式, 在每 4 个上升沿发生一次, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF

1011: 捕捉模式, 在每 16 个上升沿发生一次, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF

1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF

1101: 捕捉模式, 上升沿, 复位 TIMER1, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF

1110: 捕捉模式, 下降沿, 捕捉值存入 CCPR3 寄存器, 不产生中断标志 CCP3IF

1111: 捕捉模式, 上升沿, 捕捉值存入 CCPR3 寄存器, 不产生中断标志 CCP3IF

001X: PWM 模式, 启动 CCP3PWM 模式

0xfb1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON3	BPSK	CCP4CH2	CCP4CH1	CCP4CH0	CCP4M3	CCP4M2	CCP4M1	CCP4M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] BPSK: 改通道 PWM 支持采用 UART 实现 BPSK 调试信号输出

1: 使能与 UART 进行 BPSK 调制输出

0: 禁止 BPSK 调制输出

BIT[6:4] CCP4CH[2:0]: CCP4 捕获触发信号选择

000: 外部 CCP1CH0 引脚输入信号 (PA6)

001: 外部 CCP1CH1 引脚输入信号 (PA1)

010: 外部 CCP1CH2 引脚输入信号 (PA0)

011: 外部 CCP1CH3 引脚输入信号 (PA4)

110: 内部 32K 输入信号

BIT[3:0] CCP4M[3:0] – CCP4 模式选择位

0000: 禁止捕捉/比较/PWM (复位 CCP3 模块)

0100: 比较模式, 选择 CCPR4 匹配时将输出置为高电平 (CCP4IF 位置 1), 输出引脚为 PB7, TIMER1 溢出时, PB7 为低电平

0101: 比较模式, 选择 CCPR4 匹配时将输出置为低电平 (CCP4IF 位置 1), 输出引脚为 PB7, TIMER1 溢出时, PB7 为高电平

0110: 比较模式, 选择 CCPR4 匹配时将产生软件中断 (CCP4IF 位置 1, 而 PB5 引脚不受影响)

0111: 比较模式, 选择 CCPR4 触发特殊事件 (CCP4IF 位置 1, PB5 引脚不受影响); 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1)

1000: 捕捉模式, 在每个下降沿发生, 捕捉值存入 CCPR4 寄存器, 并产生中断标志 CCP4IF

1001: 捕捉模式, 在每个上升沿发生, 捕捉值存入 CCPR4 寄存器, 并产生中断标志 CCP4IF

1010: 捕捉模式, 在每 4 个上升沿发生一次, 捕捉值存入 CCPR4 寄存器, 并产生中断标志 CCP4IF

1011: 捕捉模式, 在每 16 个上升沿发生一次, 捕捉值存入 CCPR4 寄存器, 并产生中断标志 CCP4IF

1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR4 寄存器, 并产生中断标志 CCP4IF

1101: 捕捉模式, 上升沿, 复位 TIMER1, 捕捉值存入 CCPR4 寄存器, 并产生中断标志 CCP4IF

1110: 捕捉模式, 下降沿, 捕捉值存入 CCPR4 寄存器, 不产生中断标志 CCP3IF

1111: 捕捉模式, 上升沿, 捕捉值存入 CCPR4 寄存器, 不产生中断标志 CCP3IF

001X: PWM 模式, 启动 CCP3PWM 模式

0xfb8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON4	-	-	-	-	CCPR1L[11:8]			
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3:0] CCPR1L[11:8] – CCPR1L2 高四位

捕捉模式:

未使用

比较模式:

未使用

PWM 模式:

这些位是 PWM 占空比的高四个 LSB, 8 个 LSB 在 CCPR1L 中。

0xfb7	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON5	-	-	-	-	CCPR2L[11:8]			
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3:0] CCPR2L[11:8] – CCPR2L 高四位

捕捉模式:

未使用

比较模式:

未使用

PWM 模式:

这些位是 PWM 占空比的高四个 LSB，8 个 LSB 在 CCPR1L 中。

0xfb6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON6	-	-	-	-	CCPR3L[11:8]			
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3:0] CCPR3L [11:8] – CCPR3L 高四位

捕捉模式:

未使用

比较模式:

未使用

PWM 模式:

这些位是 PWM 占空比的高四个 LSB，8 个 LSB 在 CCPR1L 中。

0xfb5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPCON7	-	-	-	-	CCPR4L[11:8]			
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3:0] CCPR4L [11:8] – PWM26 高四位

捕捉模式:

未使用

比较模式:

未使用

PWM 模式:

这些位是 PWM 占空比的高四个 LSB，8 个 LSB 在 CCPR1L 中。

0xf67	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON0	PWM27	PWM26	PWM25	PWM24	PWM23	PWM22	PWM21	PWM20
	POL	POL	POL	POL	POL	POL	POL	POL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT0 PWM20POL – PWM20 输出极性

BIT1 PWM21POL – PWM21 输出极性

BIT2	PWM22POL	– PWM22 输出极性
BIT3	PWM23POL	– PWM23 输出极性
BIT4	PWM24POL	– PWM24 输出极性
BIT5	PWM25POL	– PWM25 输出极性
BIT6	PWM26POL	– PWM26 输出极性
BIT7	PWM27POL	– PWM27 输出极性

0xf66	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON1	PWM27	PWM26	PWM25	PWM24	PWM23	PWM22	PWM21	PWM20
	OE	OE	OE	OE	OE	OE	OE	OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT0 PWM20OE – PWM20 输出使能（禁止后，输出电平由对应 IO 的寄存器决定）

1, 使能 PWM20 输出

0, 禁止 PWM20 输出

BIT1 PWM21OE – PWM21 输出使能（禁止后，输出电平由对应 IO 的寄存器决定）

1, 使能 PWM21 输出

0, 禁止 PWM21 输出

BIT2 PWM22OE – PWM22 输出使能（禁止后，输出电平由对应 IO 的寄存器决定）

1, 使能 PWM22 输出

0, 禁止 PWM22 输出

BIT3 PWM23OE – PWM23 输出使能（禁止后，输出电平由对应 IO 的寄存器决定）

1, 使能 PWM23 输出

0, 禁止 PWM23 输出

BIT4 PWM24OE – PWM24 输出使能（禁止后，输出电平由对应 IO 的寄存器决定）

1, 使能 PWM24 输出

0, 禁止 PWM24 输出

BIT5 PWM25OE – PWM25 输出使能（禁止后，输出电平由对应 IO 的寄存器决定）

1, 使能 PWM25 输出

0, 禁止 PWM25 输出

BIT6 PWM26OE – PWM26 输出使能（禁止后，输出电平由对应 IO 的寄存器决定）

1, 使能 PWM26 输出

0, 禁止 PWM26 输出

BIT7 PWM27OE – PWM27 输出使能(禁止后, 输出电平由对应 IO 的寄存器决定)

1, 使能 PWM27 输出

0, 禁止 PWM27 输出

PWM21 死区时间控制

0xf65	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON2	PWM21DLY[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

PWM23 死区时间控制

0xf64	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON3	PWM23DLY[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

PWM25 死区时间控制

0xf63	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON4	PWM25DLY[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

PWM27 死区时间控制

0xf62	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON5	PWM27DLY[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

0xf61	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON6	PWM2ASF	PWM2ACF	LVDFLTEN	-	-	IOFLTEN	ASTART	ACLOSE
R/W	R/W	R/W	R/W	-	-	R/W	R/W	R/W
初始值	0	0	0	-	-	0	0	0

BIT0 ACLOSE– 发生故障后, 自动关闭 PWM 输出

1, 使能关闭 PWM 输出

0, 禁止关闭 PWM 输出

BIT1 ASTART- 发生故障后(ACLOSE=1), 当使能的故障都解除后, 自动启动 PWM 输出

1, 使能启动 PWM 输出

0, 禁止启动 PWM 输出

BIT2 IOFLTEN- PB0 映射为故障 IO, 当从 0 变为 1 后, 禁止 PWM 输出

1, 使能 PB0 作为故障输入

0, 禁止 PB0 作为故障输入

BIT5 LVDFLTEN- LVDOUT 为故障输入, 当从 0 变为 1 后, 禁止 PWM 输出

1, 使能 LVDOUT 作为故障输入

0, 禁止 LVDOUT 作为故障输入

BIT6 PWM2ACF - PWM2 发生自动关闭

1, 发生自动关闭

0, 未发生自动关闭

BIT7 PWM2ASF - PWM2 发生自动启动

1, 发生自动启动

0, 未发生自动启动

0xf60	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2 CON7	PWM26A	PWM26	PWM24A	PWM24	PWM22A	PWM22	PWM20A	PWM20
	DPOS	ADEN	DPOS	ADEN	DPOS	ADEN	DPOS	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT0 PWM20ADEN- 使能 PWM20 输出跳变时, 自动启动 ADC 采集功能

1, 使能 ADC 采集

0, 禁止 ADC 采集

BIT1 PWM20ADPOS- PWM20 跳变沿使能 ADC 采集

1, 使能 PWM20 由 1 变为 0 采集

0, 使能 PWM20 由 0 变为 1 采集

BIT2 PWM22ADEN- 使能 PWM22 输出跳变时, 自动启动 ADC 采集功能

1, 使能 ADC 采集

0, 禁止 ADC 采集

BIT3 PWM22ADPOS- PWM22 跳变沿使能 ADC 采集

1, 使能 PWM22 由 1 变为 0 采集

0, 使能 PWM22 由 0 变为 1 采集

BIT4 PWM24ADEN- 使能 PWM24 输出跳变时, 自动启动 ADC 采集功能

1, 使能 ADC 采集

0, 禁止 ADC 采集

BIT5 PWM24ADPOS- PWM24 跳变沿使能 ADC 采集

1, 使能 PWM24 由 1 变为 0 采集

0, 使能 PWM24 由 0 变为 1 采集

BIT6 PWM26ADEN- 使能 PWM26 输出跳变时, 自动启动 ADC 采集功能

1, 使能 ADC 采集

0, 禁止 ADC 采集

BIT7 PWM26ADPOS- PWM26 跳变沿使能 ADC 采集

1, 使能 PWM26 由 1 变为 0 采集

0, 使能 PWM26 由 0 变为 1 采集

PWM AD 采集延时

0xf5f	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2CON8	PWMADDLY[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

7.1 捕捉模式

在捕捉模式下, 当引脚 CCP1 发生事件时, CCPR1H:CCPR1L 将捕捉 TMR1 寄存器的 16 位值。事件定义如下, 由 CCP1CON[3:0]进行配置:

1000: 捕捉模式, 在每个下降沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1001: 捕捉模式, 在每个上升沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1010: 捕捉模式, 在每 4 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1011: 捕捉模式, 在每 16 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1101: 捕捉模式, 上升沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF

1110: 捕捉模式, 下降沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCP1IF

1111: 捕捉模式, 上升沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCP1IF 进行捕捉后, 中断请求标志位 CCP1IF 被置 1。该中断标志位必须用软件清零。如果在 CCPR1H 和 CCPR1L 寄存器对中的值被读出之前又发生另一次捕捉, 那么原来的捕捉值会被新捕捉值覆盖。输入捕获通道有 CCP1CH[3:0]进行配置:

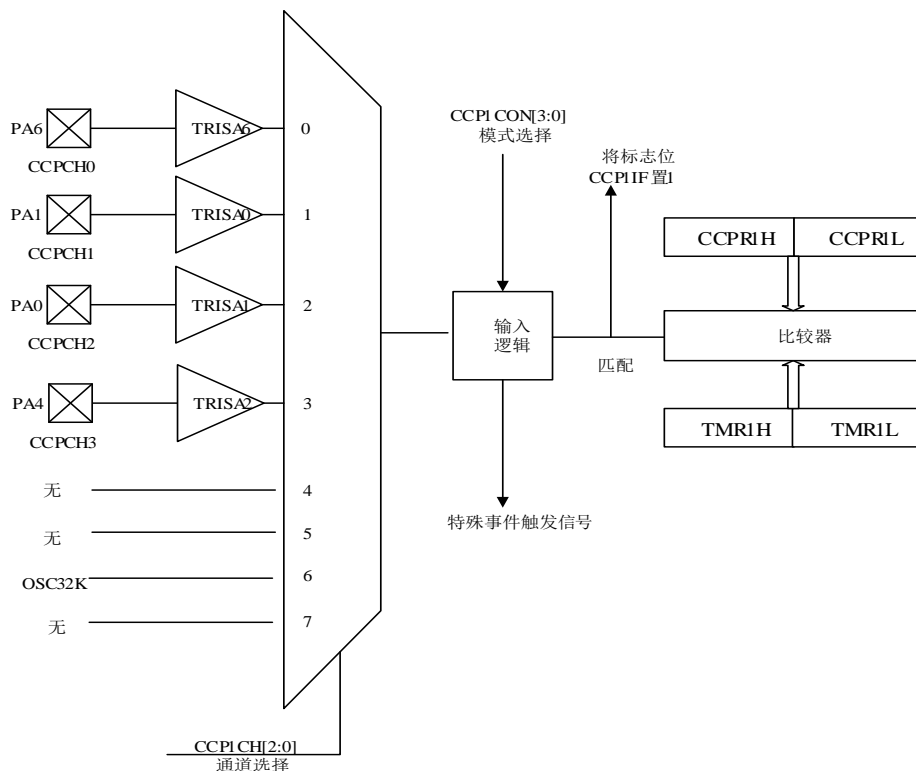
0000: 外部 CCP1CH0 引脚输入信号

0001: 外部 CCP1CH1 引脚输入信号

0010: 外部 CCP1CH2 引脚输入信号

0011: 外部 CCP1CH3 引脚输入信号

0110: 内部 32K 输入信号



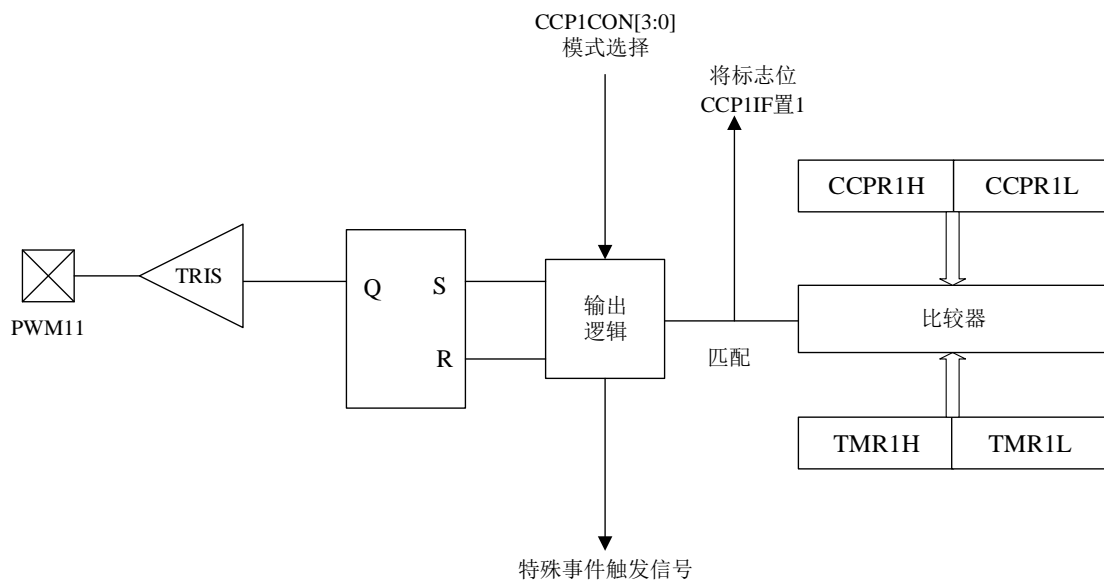
CCP1 捕捉模式工作原理图

同理, CCP2, CCP3, CCP4 的捕捉模式采用同样原理工作。

7.2 比较模式

在比较模式下, CCPR1 寄存器的 16 位值不断与一对 TMR1 寄存器的值进行比较。事件定义如下, 由 CCP1CON[3:0] 进行配置:

- ◇ 0100: 比较模式, 选择 CCPR1 匹配时将输出置为高电平 (CCP3IF 位置 1), 输出引脚为 T1O, TIMER1 溢出时, T1O 为低电平, 中断标志位 CCP1IF 置 1
- ◇ 0101: 比较模式, 选择 CCPR1 匹配时将输出置为低电平 (CCP3IF 位置 1), 输出引脚为 T1O, TIMER1 溢出时, T1O 为高电平, 中断标志位 CCP1IF 置 1
- ◇ 0110: 比较模式, 选择 CCPR1 匹配时将产生软件中断 (CCP3IF 位置 1, 而 T1O 引脚不受影响), 中断标志位 CCP1IF 置 1, 并启动 ADC 采集 (如果 ADCON=1);
- ◇ 0111: 比较模式, 选择 CCPR1 触发特殊事件 (CCP3IF 位置 1, T1O 引脚不受影响); 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1), 中断标志位 CCP1IF 置 1



比较模式工作原理图

7.3 PWM 模式

PWM 模式在 CCP1 引脚上产生脉宽调制信号。占空比、周期和分辨率由以下寄存器决定：

- ◇ PR2,PR2H
- ◇ T2CON
- ◇ CCPR1L、CCPR2L、CCPR3L、CCPR4L
- ◇ CCP1CON,CCP2CON,CCP3CON,CCP4CON,CCP5CON
- ◇ PWM2CON0, PWM2CON1, PWM2CON2, PWM2CON3, PWM2CON4,
PWM2CON5, PWM2CON6, PWM2CON7, PWM2CON8

在脉宽调制（PWM）模式下，CCP 模块会在 PWM20 和 PWM21 引脚上产生最大 12 位分辨率的 PWM 输出信号。由于 PWM20 和 PWM21 引脚与端口数据锁存器复用，该引脚的 TRIS 必须被清零以启用 PWM20 和 PWM21 引脚输出驱动器。

可通过写 PR2 寄存器来指定 PWM 周期。可以用以下公式计算 PWM 周期：

当 FCPUS=1(4T)时

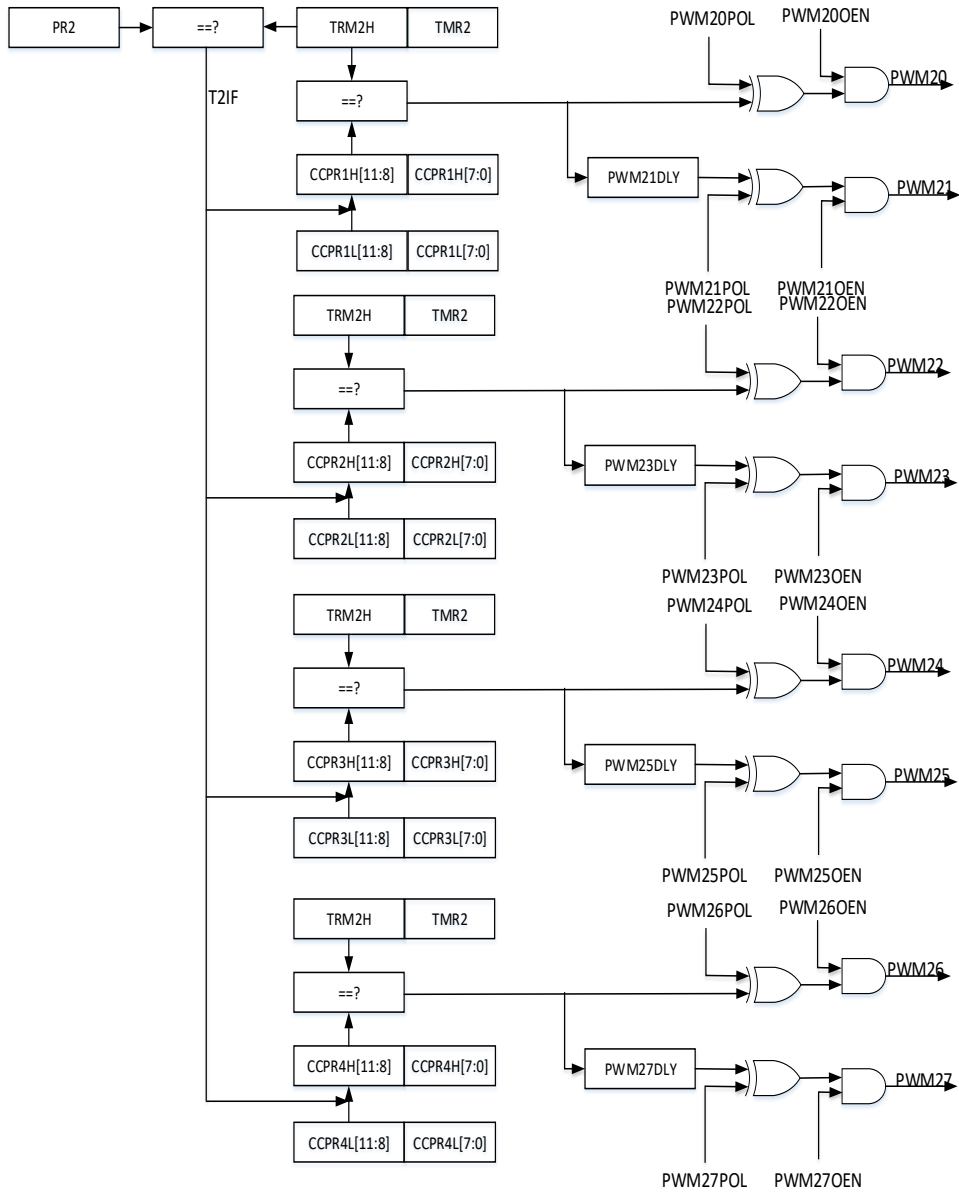
PWM 周期=[(PR2[11:0]) + 1]•(TMR2 预分频值)

当 TMR2 等于 PR2 时，在下一个递增计数周期中将发生以下三件事：

- TMR2 被清零
- PWM20 和 PWM21 引脚被置 1。（例外情况：如果 PWM 占空比=0%，引脚将不会被置 1。）
- PWM 占空比从 CCPR1L 锁存到 CCPR1H。

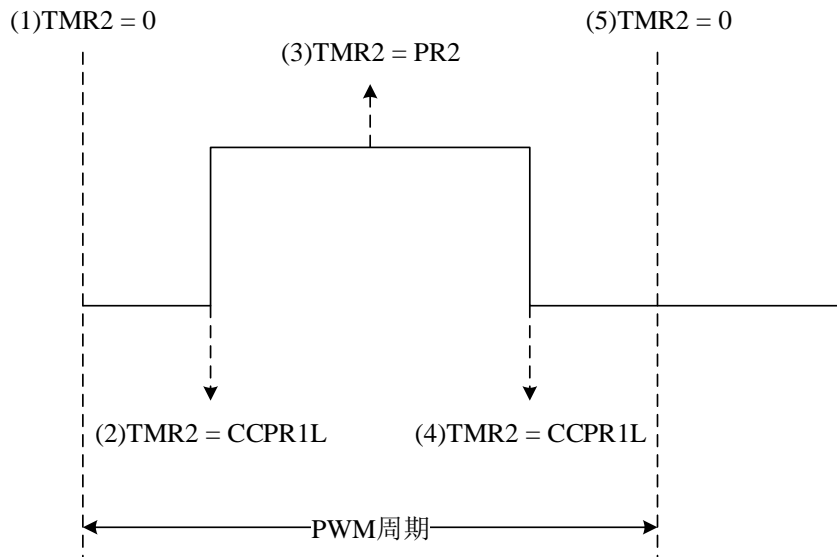
通过写入 CCPR1L 寄存器和 CCP4CON[3:0]位来指定 PWM 的占空比。最高分辨率可达 12 位。CCPR1L 包含八个 LSB，CCP4CON[3:0]包含四个 MSB。由 CCP1CON[3:0]:CCPR1L 表示这个 12 位值。计算 PWM 占空比的公式如下：

PWM 占空比= (PWM1L[11:8]:CCPR1L) •(TMR2 预分频值)



PWM 模块的简化框图

7.3.1 PWM 中心对齐模式



PWM 中心对齐示意图

当 PWM2CAEN (T2CON<7>) 开启时, PWM 使能中心对齐模式, 此时 PWM 工作过程如下:

- (1) PWM 周期开始, TMR2 开始从 0 递增;
- (2) 当 TMR2 与 CCPR1L 相等时, PWM 开始第一次高低电平变化, TMR2 继续递增;
- (3) 当 TMR2 与 PR2 相等时, TMR2 开始自减;
- (4) 当 TMR2 再次与 CCPR1L 相等时, PWM 再次变化高低电平;
- (5) 当 TMR2 自减为 0 时, 此时 PWM 周期结束, 开始下一个 PWM 周期。

此时, 实际上 PWM 周期为 $2 \cdot (PR2+1) \cdot T_{soc} \cdot (TMR2 \text{ 预分频值})$ 。

注意: 普通 pwm 模式切换中心对齐模式需要将 pwm 关闭后重新打开。

8 中断

AD1845 提供多个中断源及一个中断优先级功能，可以给大多数中断源分配高优先级或者低优先级。高优先级中断向量地址为 0008h，低优先级中断向量地址为 0018h。高优先级中断事件将中断所有可能正在进行的低优先级中断。

通过将 IPEN 位 (RCON<7>) 置 1，可启用中断优先级功能。当启用中断优先级时，有 2 位可启用全局中断。将 GIEH 位 (INTCON<7>) 置 1，可启用所有优先级位置 1 (高优先级) 的中断。将 GIEL 位 (INTCON<6>) 置 1，可启用所有优先级位清零 (低优先级) 的中断。当中断标志位、使能位以及相应的全局中断是均为被置 1 时，程序将立即跳转到中断地址 0008h 或 0018h，具体地址取决于优先级位的设置。通过设置相应的使能位可以禁止单个中断。

当 IPEN 位被清零 (默认状态) 时，便会禁止中断优先级功能，各个中断源的中断优先级位均不起作用。INTCON<6>是 PEIE 位，它可启用/禁止所有的外设中断源。INTCON<7>是 GIE 位，它可启用/禁止所有的中断源。所有中断均跳转到地址 0008h。

当响应中断时，全局中断允许位被清零以禁止其他中断。如果清零 IPEN 位，全局中断使能位就是 GIE 位。如果使用中断优先级，这个位将是 GIEH 位或者 GIEL 位。高优先级中断源会中断低优先级中断。处理高优先级中断时，低优先级中断将不被响应。

返回地址被压入堆栈，PC 中装入中断向量地址 (0008h 或 0018h)。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志清零，以避免重复响应中断。

执行“中断返回”指令 RETFIE，退出中断程序并置位 GIE 位 (若使用中断优先级，则为 GIEH 或 GIEL 位) 以重新允许中断。

对于外部中断事件，诸如 INT 引脚中断或者 PORTB 输入电平变化中断，中断响应延时将会是 3 到 4 个指令周期。对于单周期或双周期指令，中断响应延时完全相同。不管对应的中断使能位和 GIE 位状态如何，各中断标志位均被置 1。

0xfd2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCON	IPEN	CKOE	LDO2P0EN	RI	TO	PD	POR	BOR
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
初始值	0	0	0	0	0	0	x	x

BIT[7] IPEN – 中断优先级使能位

- 1: 使能中断优先级
- 0: 禁止中断优先级

0xff2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
初始值	0	0	-	-	-	-	-	-

BIT[7] GIE/GIEH – 全局中断使能位

- 1: 使能所有未被屏蔽的中断
- 0: 禁止所有中断

BIT[6] PEIE/GIEL – 外设中断使能位

当 IPEN=0 时:

- 1: 允许所有未屏蔽的外设中断
- 0: 禁止所有外设中断

当 IPEN=1 时:

- 1: 允许所有低优先级的外设中断
- 0: 禁止所有低优先级的外设中断

8.1 INTn 引脚中断

INT0、INT1 和 INT2 引脚的外部中断是边沿触发的。如果 INTCON1 寄存器中对应的 INTEDGx 位置位 (=1)，则该中断由上升沿触发；如果该位清零，则中断由下降沿触发。当 INTx 引脚上出现一个有效边沿时，对应的标志位 INTxF 被置 1。通过清零对应的使能位 INTxIE，可禁止该中断。在重新使能该中断前，必须在中断服务程序中先用软件将 INTxIF 标志位清零。

如果 INTxIE 位在进入空闲或休眠模式前被置 1，则所有的外部中断 (INT0、INT1 以及 INT2) 均能把处理器从这些模式唤醒。如果全局中断使能位 GIE 被置 1，则处理器将在被唤醒之后跳转到中断向量处执行程序。

INT0、INT1 和 INT2 的中断优先级由中断优先级位 INT0IP、INT1IP 和 INT2IP 包含的值决定。

0xff1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON1	-	-	-	INT0MAP1	INT0MAP0	INTEDG2	INTEDG1	INTEDG0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4: 3] INT0MAP[1:0] – 外部中断 0 映射

00: 映射到 PA0

01: 映射到 PA4

10: 映射到 PA5

BIT[2] INTEDG2 – 外部中断 2 边沿选择位

0: 上升沿触发中断

1: 下降沿触发中断

BIT[1] INTEDG1 – 外部中断 1 边沿选择位

0: 上升沿触发中断

1: 下降沿触发中断

BIT[0] INTEDG0 – 外部中断 0 边沿选择位

0: 上升沿触发中断 STATUS

1: 下降沿触发中断

0xfa0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR2	PCIP	PBIP	PAIP	INT2IP	INT1IP	INT0IP	OP2IP	OP1IP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[4] INT2IP – INT2 外部中断优先级位

1: 高优先级

0: 低优先级

BIT[3] INT1IP – INT1 外部中断优先级位

1: 高优先级

0: 低优先级

BIT[2] INT0IP – INT0 外部中断优先级位

1: 高优先级

0: 低优先级

0xf9f	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR2	PCIF	PBIF	PAIF	INT2IF	INT1IF	INT0IF	OP2IF	OP1IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

- BIT[4] INT2IF – INT2 外部中断标志位
- 1: 发生了 INT2 外部中断 (必须用软件清零)
 - 0: 未发生 INT2 外部中断
- BIT[3] INT1IF – INT1 外部中断标志位
- 1: 发生了 INT1 外部中断 (必须用软件清零)
 - 0: 未发生 INT1 外部中断
- BIT[2] INT0IF – INT0 外部中断标志位
- 1: 发生了 INT0 外部中断 (必须用软件清零)
 - 0: 未发生 INT0 外部中断

0xf9e	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE2	PCIE	PBIE	PAIE	INT2IE	INT1IE	INT0IE	OP2IE	OP1IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

- BIT[4] INT2IE – INT2 外部中断使能位
- 1: 使能 INT2 外部中断
 - 0: 禁止 INT2 外部中断
- BIT[3] INT1IE – INT1 外部中断使能位
- 1: 使能 INT1 外部中断
 - 0: 禁止 INT1 外部中断
- BIT[2] INT0IE – INT0 外部中断使能位
- 1: 使能 INT0 外部中断
 - 0: 禁止 INT0 外部中断

8.2 定时器中断

定时器 T0、T1、T2 在计数溢出时会产生中断请求 TMR0IF、TMR1IF、TMR2IF，它们有各自的屏蔽位。见[§6.1](#)、[§6.2](#)、[§6.3](#)。

0xfa3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR1	RCIP	TXIP	-	-	-	TMR2IP	TMR1IP	TRM0IP

R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	1	1	-	-	-	1	1	1

0xfa2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR1	RCIF	TXIF	-	-	-	TMR2IF	TMR1IF	TRM0IF
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

0xfa1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE1	RCIE	TXIE	-	-	-	TMR2IE	TMR1IE	TRM0IE
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

8.3 PORTA/PORTB 电平变化中断

如果 PAIE/PBIE 使能，并且 PAINTMASK_n/PBINTMASK_n 有效，那么相应的 Pan/PBn/引脚发生电平变化时，PAIF/PBIF 便会被置 1。PA/PB 电平变化中断的优先级由中断优先级位 PAIP、PBIP 的值决定。

0xfa0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR2	-	PBIP	PAIP	INT2IP	INT1IP	INT0IP	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-
初始值	-	1	1	1	1	1	-	-

BIT[6] PBIP – PB 键盘中断优先级位

1: 高优先级

0: 低优先级

BIT[5] PAIP – PA 键盘中断优先级位

1: 高优先级

0: 低优先级

0xf9f	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR2	-	PBIF	PAIF	INT2IF	INT1IF	INT0IF	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-

初始值	-	0	0	0	0	0	-	-
-----	---	---	---	---	---	---	---	---

BIT[6] PBIF –PB 键盘中断标志位

1: 发生了键盘中断（必须用软件清零）

0: 未发生键盘中断

BIT[5] PAIF –PA 键盘中断标志位

1: 发生了键盘中断（必须用软件清零）

0: 未发生键盘中断

0xf9e	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE2	-	PBIE	PAIE	INT2IE	INT1IE	INT0IE	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-
初始值	-	0	0	0	0	0	-	-

BIT[6] PBIE –PB 键盘中断使能允许位

1: 使能 PB 键盘中断

0: 禁止 PB 键盘中断

BIT[5] PAIE –PA 键盘中断使能允许位

1: 使能 PA 键盘中断

0: 禁止 PA 键盘中断

0xf9d	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR3	CCADCIP	ADIP	CCP3PIP	CCP2PIP	CCP1PIP	CCP4PIP	-	LVDIP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
初始值	1	1	1	1	1	1	-	1

BIT[0] PCIP –PC 键盘中断优先级

1: 高优先级

0: 低优先级

0xf9c	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR3	CCADCIF	ADIF	CCP3PIF	CCP2PIF	CCP1PIF	CCP4PIF	-	LVDIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
初始值	0	0	0	0	0	0	-	0

BIT[0] PCIF –PC 键盘中断标志位

1: 发生了键盘中断（必须用软件清零）

0: 未发生键盘中断

0xf9b	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE3	CCADCIE	ADIE	CCP3PIE	CCP2PIE	CCP1PIE	CCP4PIE	-	LVDIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
初始值	0	0	0	0	0	0	-	0

BIT[0] PCIE –PC 键盘中断使能允许位

1: 使能 PC 键盘中断

0: 禁止 PC 键盘中断

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAINT MASK	PAINT MASK7	PAINT MASK6	PAINT MASK5	PAINT MASK4	-	-	PAINT MASK1	PAINT MASK0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	0	0	0	0	-	-	0	0

BIT[7:0] PAINTMASK7:PAINTMASK0 – PAn 端口电平变化中断掩膜位

1: 使能 PAn (n=0~7) 端口电平变化中断

0: 禁止 PAn (n=0~7) 端口电平变化中断

0xff2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
初始值	0	0	-	-	-	-	-	-

BIT[3] PBIE – PB 端口电平变化中断允许位

1: 允许 PB 端口电平变化中断

0: 禁止 PB 端口电平变化中断

BIT[0] PBIF – PB 端口电平变化中断标志位

1: PB7:PB0 引脚至少有一个引脚的电平状态发生了变化（必须用软件清零）

0: PB7:PB0 引脚的电平状态没有改变。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBINT MASK	PBINT MASK7	-	-	-	PBINT MASK3	PBINT MASK2	PBINT MASK1	PBINT MASK0
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W

初始值	0	-	-	-	0	0	0	0
-----	---	---	---	---	---	---	---	---

BIT[7:0] PBINTMASK7:PBINTMASK0 – PBn 端口电平变化中断掩膜位

1: 使能 PBn (n=0~7) 端口电平变化中断

0: 禁止 PBn (n=0~7) 端口电平变化中断

8.4 外设中断

外设有 UART 接收发送中断, I2C 中断, CCP 中断, 低电压中断, OP1 中断, OP2 中断, CCADC 中断, ADC 中断, PWM1 中断, PWM2 中断, OSC 异常中断, I2C 异常中断等。

0xfa3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR1	RCIP	TXIP	-	-	-	TMR2IP	TMR1IP	TRM0IP
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	1	1	-	-	-	1	1	1

0xfa2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR1	RCIF	TXIF	-	-	-	TMR2IF	TMR1IF	TRM0IF
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

0xfa1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE1	RCIE	TXIE	-	-	-	TMR2IE	TMR1IE	TRM0IE
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

0xfa0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR2	-	PBIP	PAIP	INT2IP	INT1IP	INT0IP	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-
初始值	-	1	1	1	1	1	-	-

0xf9f	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR2	-	PBIF	PAIF	INT2IF	INT1IF	INT0IF	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-
初始值	-	0	0	0	0	0	-	-

0xf9e	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE2	-	PBIE	PAIE	INT2IE	INT1IE	INT0IE	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-

初始值	-	0	0	0	0	0	-	-
-----	---	---	---	---	---	---	---	---

0xf9d	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR3	CCADCIP	ADIP	CCP3PIF-	CCP2PIF	CCP1PIP	CCP4PIP	-	LVDIP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
初始值	1	1	1	1	1	1	-	1

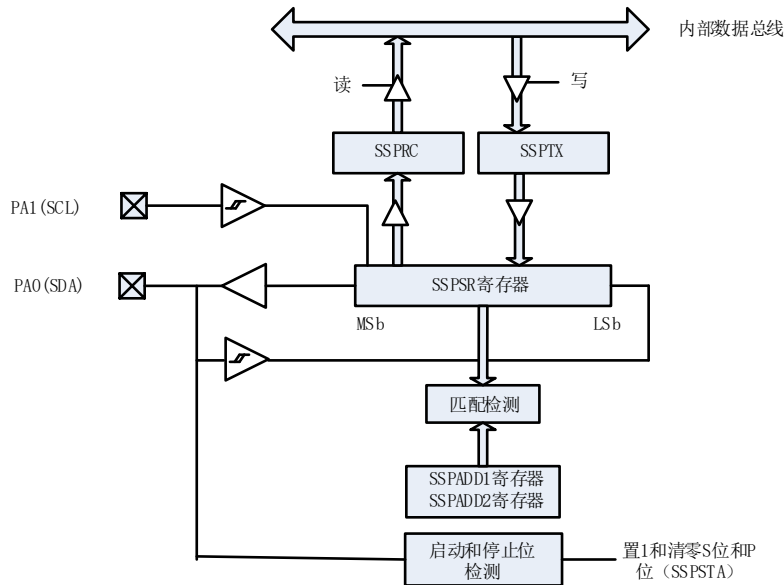
0xf9c	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR3	CCADCIF	ADIF	CCP3PIF	CCP2PIF	CCP1PIF	CCP4PIF	-	LVDIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
初始值	0	0	0	0	0	0	-	0

0xf9b	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE3	CCADCIE	ADIE	CCP3PIE	CCP2PIE	CCP1PIE	CCP4PIE	-	LVDIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
初始值	0	0	0	0	0	0	-	0

9 I2C 控制器

I2C 模块仅支持从动模式，有两个引脚用于数据传输：

- 串行时钟（SCL）-- PA1
- 串行数据（SDA）-- PA0



I²C 模式框图

I2CCON0、I2CCON1 和 I2CSTAT 是 I2C 模式的控制寄存器和状态寄存器。I2CCON0 和 I2CCON1 寄存器是可读写的。I2CSTAT 的低六位是只读的，而高两位是可读写的。

SSPSR 是用来将数据移入或移出的移位寄存器。SSSRC 是接收缓冲寄存器，用于数据字节的读出，I2CTX 是发送缓冲寄存器，用于数据字节的写入。在 I2C 从动模式下配置 SSP 时，SSPADD 寄存器将保存器件的地址。SSPADD 寄存器将保存从器件的地址。在主机模式下配置 I2C 时，I2CADD 的低 7 位保存波特率发生器的重载值。

接收数据时，SSPSR 和 I2CRC、SSPTX 共同构成一个双重缓冲接收器。当 SSPSR 接收到一个完整的字节后，该字节被送入 SSSRC 寄存器，同时将中断标志位 I2C_RCIF 置 1；同时如果 SSPTX 有数据，则可以立即进入发送数据模式，同时将中断标志位 I2C_TXIF 置 1。在发送过程中，SSPTX 与 SSPSR 构成双重缓冲发送。

I²C 相关寄存器:

0xfc6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CSTAT	SMP1	SMP0	D_A	P	S	R_W	BFRC	BFTX
R/W	R/W	R/W	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

BIT[7:6] SMP – 变化率控制位

在主控或从动模式下:

11: 标准速度模式下禁止变化率控制 (100kHz 和 1MHz)

00: 高速模式下使能变化率控制 (400kHz)

BIT[5] D_A – 数据/地址标志位

1: 表示上一个接收或发送的字节是数据

0: 表示上一个接收或发送的字节是地址

BIT[4] P – 停止位

1: 表示上一次检测到停止位

0: 表示上一次未检测到停止位

注: 当复位或 SSPEN 被清零时, 该位被清零。

BIT[3] S – 启动位

1: 表示上一次检测到启动位

0: 表示上一次未检测到启动位

BIT[2] R_W – 读/写信息位 (仅用于 I2C 模式)

1: 读

0: 写

注: 该位用来保存在最近一次地址匹配后的 $\overline{\text{R/W}}$ 位信息。该位

仅在从地址匹配开始到下一个启动位、停止位或非 ACK 位之间有效。

BIT[1] BFRC – 接收缓冲器满状态位

1: SSPSR 已满 (不包括 ACK 位和停止位)

0: SSPSR 为空 (不包括 ACK 位和停止位)

BIT[0] BFTX – 发送缓冲器满状态位

1: SSPBUF 已满

0: SSPBUF 为空

0xfc5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CCON0	WCOL	I2COV	I2CEN	-	I2CM3	I2CM2	I2CM1	I2CM0

R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] WCOL – 写冲突检测位

1: 正在发送前一个字时, 又有数据写入 I2CTX 寄存器 (必须用软件清零)

0: 未发生冲突

在接收模式 (主控或从动模式) 下:

在此忽略。

BIT[6] I2COV – 接收溢出指示位

在接收模式下:

1: I2CRC 寄存器仍在保存前一个字节时, 又接收到一个新的字节 (必须用软件清零)

0: 无溢出

在发送模式下:

此位被忽略。

BIT[5] I2CEN – 同步串口使能位

1: 使能并将 SDA 和 SCL 引脚配置为 I2C 串口引脚

0: 禁止并将上述引脚配置为 I/O 端口引脚

注: 当使能该位时, 必须将 SDA 和 SCL 引脚正确地配置为输入引脚或输出引脚

BIT[3:0] I2CM3:I2CM0 – I2C 模式选择位

0111: I2C 从动模式, 10 位地址

0110: I2C 从动模式, 7 位地址

0xfc4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CCON1	GCEN	ACKSTAT	ACKGET	ACKEN	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
初始值	0	0	0	0	0	0	0	0

BIT[7] GCEN – 广播呼叫使能位 (仅用于从动模式)

1: 当 SSPSR 接收到广播呼叫地址 (0000h) 时允许中断

0: 禁止广播呼叫地址

BIT[6] ACKSTAT – 从器件发送状态时, 接收到的主机的应答状态位

1: 收到来自主器件的 NACK 应答

0: 收到来自主器件的 ACK 应答

BIT[5] ACKGET – 从器件发送状态时，表示接收到的主机的应答状态位；ACKSTAT 表示接收的具体应答

1: 未收到来自主器件的应答，需要软件清 0

0: 收到来自主器件的应答

BIT[4] ACKEN – 从器件接收状态时，返回主机应答状态位

1: 表示下一个 I2CRX 接收后，从器件发送应答位 NACK

0: 表示下一个 I2CRX 接收后，从器件发送应答 ACK

注：对于 ACKEN、RCEN、PEN、RSEN 和 SEN 位来说，如果 I²C 模块不处于空闲模式，这些位不能被置 1（或支持后台操作），并且不能对 SSPBUF 进行写操作（或者禁止写 SSPBUF）。

0xfc9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CADDR0	I2CADDR0							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7: 0] I2CADDR0

在从动模式下：

I2C 器件地址（7 位或者 10 位，根据 I2CM 定义）

0xfc8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CADDR1	I2CADDR1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7: 0] I2CADDR1

在从动模式下：

I2C 器件地址（7 位另外一个地址或者 10 位高四位地址，根据 I2CM 定义）

0xfc7	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CADDRMASK	I2CADDRMASK							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7: 0] I2CADDRMASK

在从动模式下:

I2C 器件地址 7 位器件地址位匹配定义

0xfcb	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CTX	I2CTX							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7: 0] I2CTX

与 SSPSR 构成接收双发送 buffer

0xfca	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CRC	I2CRC							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7: 0] I2CRC

与 SSPSR 构成接收双接收 buffer

0xfc3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CIE	-	-	-	I2C_FLOATIE	I2C_STOPIE	I2C_STARTIE	I2C_TXIE	I2C_RCIE
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[4] I2C_FLOATIE-I2C 总线低时间超过 25ms, 则产生中断

1: 使能

0: 禁止

BIT[3] I2C_STOPIE-I2C STOP 信号产生中断

1: 使能

0: 禁止

BIT[2] I2C_STARTIE-I2C START 信号产生中断

1: 使能

0: 禁止

BIT[1] I2C_TXIE-I2C 当 I2CTX 寄存器写入到 SSPSR 时, 产生发送中断

1: 使能

0: 禁止

BIT[0] I2C_RCIE-I2C 当 SSPSR 写入 I2CRC 寄存器时, 产生接收中断

1: 使能

0: 禁止

0xfc2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CIF	-	-	-	I2C_FLOATIF	I2C_STOPIF	I2C_STARTIF	I2C_TXIF	I2C_RCIF
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[4] I2C_FLOATIF-I2C 总线浮空标志

1: 中断, 写 0 清 0

0: 没有中断

BIT[3] I2C_STOPIF-I2C STOP 中断标志

1: 中断, 写 0 清 0

0: 没有中断

BIT[2] I2C_STARTIF-I2C START 中断标志

1: 中断, 写 0 清 0

0: 没有中断

BIT[1] I2C_TXIF-I2C 当 I2CTX 发送中断标志

1: 中断, 写 I2CTX 数据

0: 没有中断

BIT[0] I2C_RCIF-I2C 接收中断标志

1: 中断, 读取 I2CRC 寄存器

0: 没有中断

注: 总线浮空中断产生时间为 31.965ms。

0xfc1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CIP	-	-	-	I2C_FLOATIP	I2C_STOPIP	I2C_STARTIP	I2C_TXIP	I2C_RCIP
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

I2CIP, 中断优先级控制寄存器

0xfc9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CADDR2	I2CADDR2							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7: 0] I2CADDR2

在从动模式下:

I2C 器件地址 (7 位或者 10 位, 根据 I2CM 定义)

0xfc9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CADDR3	I2CADDR3							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7: 0] I2CADDR3

在从动模式下:

I2C 器件地址 (7 位或者 10 位, 根据 I2CM 定义)

0xfc9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACKDLY	ACKDLYON	ACKDLY						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] ACKDLYON

1: 使能 ACKDLY 功能

0: 关闭 ACKDLY 功能

BIT[6:0] ACKDLY:

$$T_{ACKDLY} = ACKDLY * T_{I2C}$$

9.1 从动模式

在从动模式下, 一旦使能了 I2C 从动模块, 它就会等待启动条件出现。启动条件出现后, 就会向 SSPSR 寄存器移入 8 位数据。在时钟信号 (SCL) 的上升沿采样所有的输入位。在第 8 个时钟 (SCL) 脉冲的下降沿, 寄存器 SSPSR<7:1> 的值会和 I2CADD0 地址寄存器的值进行比较。如果地址匹配, 并且 RCBF 位和 I2COV 位都被清零, 会发生下列事件:

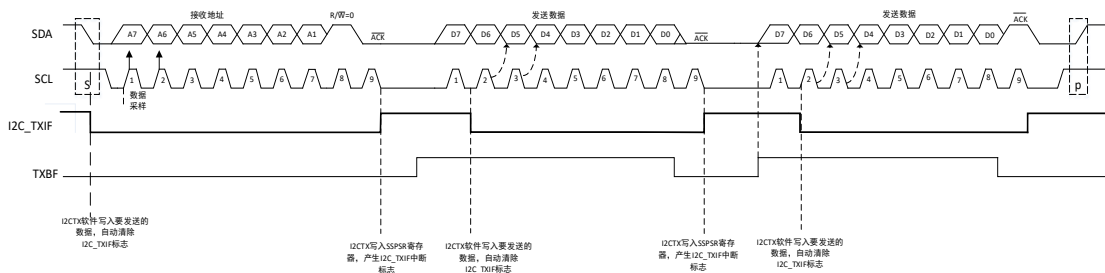
- 1) SSPSR 寄存器的值被载入 I2CRC 寄存器。
- 2) 缓冲器满标志位 RCBF 置 1。
- 3) 产生 ACK 脉冲。
- 4) 在第 9 个 SCL 脉冲下降沿, I2C_RCIF 被置 1 (如果允许中断, 则产生中断)。

当地址字节的 R/W 位为 0 且地址匹配时, SSPSTAT 寄存器的 R/W 位清零。接收的地址被装入 I2CRC 寄存器, 且 SDA 信号保持低电平 (ACK)。

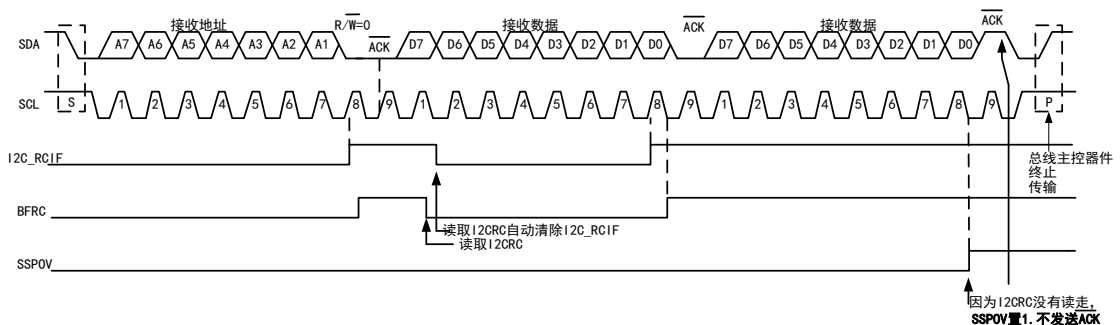
当接收的地址字节的 R/W 位为 1 且地址匹配时，SSPSTAT 寄存器的 R/W 位置 1。接收到的地址被装入 I2CRC 寄存器。ACK 脉冲在第 9 位上发送。从器件可以采用 SSPTX 提前准备好发送数据，并进行发送（可以在 I2C 配置时，写入一个 SSPTX 寄存器），当发送时被装入了 SSPSR 寄存器。8 个数据位在 SCL 时钟输入的下沿被移出。

来自主接收器的 ACK 脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。如果 SDA 数据信号线为高电平（无 ACK），那么表示数据传输已完成。在这种情况下，如果从器件锁存了 ACK，将复位从动逻辑（复位 SSPSTAT 寄存器），同时从器件监视下一个启动位的出现。如果 SDA 线为低电平（ACK），则必须将下一个要发送的数据装入 I2CTX 寄存器。

每个发送数据都会产生一个 I2C_TXIF 中断，并且。I2C_TXIF 位必须用软件清零，I2CSTAT 寄存器用于确定字节的状态。I2C_TXIF 位在第 8 个时钟脉冲的下沿被置 1。



I2C 从动模式的发送时序 (1) (SEN=0)



I2C 从动模式的接收时序

9.2 支持广播呼叫地址

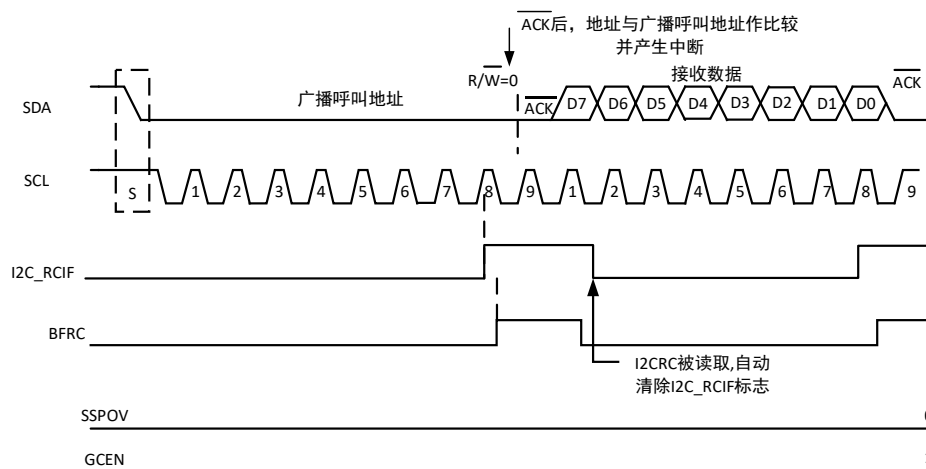
在 I²C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答信号来响应。同时支持两个地址选择，并支持 MASK 屏蔽选择。

广播呼叫地址是由 I²C 协议为特定目的保留的 8 个地址之一。它由全 0 组成，且 R/W=0。

当使能广播呼叫使能位 (GCEN) (I2CCON2<7>置 1) 时，即可识别广播呼叫地址。检测到启动位后，8 位数据会被移入 I2CRC，同时将该地址与 I2C_ADDR0 进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

如果与广播呼叫地址匹配，I2CTX 的值将被传输到 SSPBUF，BF 标志位（第 8 位）置 1，并且在第 8 位（ACK 位）的下降沿 I2C_TXIF 中断标志位置 1。

当响应中断时，可以通过读取 I2CRC 的内容来检查中断源。该值可以用于判断是特定器件的地址还是一个广播呼叫地址。



从动模式广播呼叫地址时序

10 CRC16

输入 CRC16IN 开始 CRC16 计算，在一个时钟周期后输出值为 CRC16H: CRC16L。

采用 CRC16-CCINT (CRC16-CCITT $x^{16}+x^{12}+x^5+1$)。

与 CRC16 相关寄存器:

0xf3c	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRC16I N	CRC16I N7	CRC16I N6	CRC16IN 5	CRC16IN 4	CRC16IN 3	CRC16IN 2	CRC16IN 1	CRC16IN 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0]

CRC16IN7: CRC16IN0 – CRC16 输入位

0xf3b	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRC16 H	CRC16H 7	CRC16H 6	CRC16H 5	CRC16H 4	CRC16H 3	CRC16H 2	CRC16H 1	CRC16H 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0]

CRC16H7: CRC16H0 – CRC16 输出高位

0xf3a	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRC16 L	CRC16L 7	CRC16L 6	CRC16L 5	CRC16L 4	CRC16L 3	CRC16L 2	CRC16L 1	CRC16L 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0]

CRC16L7: CRC16L0 – CRC16 输出低位

11 通用同步/异步收发器 (USART)

通用同步/异步收发器 (Universal Synchronous Asynchronous Receiver Transmitter, USART) 模块是两个串行 I/O 模块之一。(通常, USART 也被称为串行通信接口或 SCI)。AD1845 的 USART 支持全双工异步通信。

相关寄存器说明:

0xfad	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXSTA	CSRC	TX9	TXEN	SYNC	TXPARITY	BRGH	TRMT	TX9D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
初始值	0	0	0	0	0	0	1	0

BIT[7] CSRC – 时钟源选择位

异步模式:

忽略。

同步模式:

1: 主控模式 (时钟来自内部 BRG)

0: 从动模式 (时钟来自外部时钟源)

BIT[6] TX9 – 9 位发送使能位

1: 选择 9 位发送

0: 选择 8 位发送

BIT[5] TXEN – 发送使能位

1: 使能发送

0: 禁止发送

BIT[4] SYNC – USART 模式选择位

1: 同步模式

0: 异步模式

BIT[3] TXPARITY – 使能硬件自动奇偶校验

1: 自动填充发送第 9 位校验位

0: 校验位由 TXD9 填充

BIT[2] BRGH – 高波特率选择位

异步模式:

1: 高速

0: 低速

同步模式：

在此模式下未使用。

BIT[1] TRMT – 发送移位寄存器状态位

1: TSR 空

0: TSR 满

BIT[0] TX9D – 发送数据的第 9 位

该位可以是地址/数据位或奇偶校验位。

0xfac	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCSTA	SPEN	RX9	SREN	CREN	BGRM	FERR	OERR	RX9D
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
初始值	0	0	0	0	0	0	0	x

BIT[7] SPEN – 串口使能位

1: 使能串口（将 RX/DT 和 TX/CK 引脚配置为串口引脚）

0: 禁止串口（保持在复位状态）

BIT[6] RX9 – 9 位接收使能位

1: 选择 9 位接收

0: 选择 8 位接收

BIT[5] SREN – 单字节接收使能位

异步模式：

忽略。

同步主控模式：

1: 使能单字节接收

0: 禁止单字节接收

此位在接收完成后清零。

同步从动模式：

忽略。

BIT[4] CREN – 连续接收使能位

1: 使能接收器

0: 禁止接收器

BIT[3] BGRM – UART 时钟选择位

BIT[2] FERR – 帧错误位

1: 帧错误(可以通过读 RCREG 寄存器刷新并接收下一个有效字节)

0: 无帧错误

BIT[1] OERR – 溢出错误位

1: 溢出错误 (可以通过清零 CREN 位清除)

0: 无溢出错误

BIT[0] RX9D – 接收数据的第 9 位

该位可以是地址/数据位或奇偶校验位，必须由用户固件计算得到。

0xfab	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UARTCON0	INVMODE	PARITY	ODD_ EVEN	BOUD_ CLK	HALF_ DUPLEX	DLSB	STOP_ BIT1	STOP_ BIT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] INVMODE – 数据电平反向接受发送使能位

异步:

1: 使能

0: 禁止

同步:

忽略

BIT[6] PARITY – 接收奇偶校验位

1: 表示有奇数个 1 (奇校验)

0: 表示有偶数个 1 (偶校验)

BIT[5] ODD_EVEN – 设定发送、接收奇偶校验位功能

1: 奇校验

0: 偶校验

BIT[4] BOUD_CLK – 波特率发生器时钟源选择位

1: 外部时钟

0: 内部系统时钟

BIT[3] HALF_DUPLEX – 单线半双工使能位

1: 使能

0: 禁止

BIT[2] DLSB – 数据位高低位选择位

1: 高位在前 (MSB)

0: 低位在前 (LSB)

BIT[1:0] STOP_BIT[1:0] – 停止位个数设置位

00: 1 位停止位

01: 2 位停止位

10: 3 位停止位

11.1 波特率发生器 (BRG)

BRG 是一个 8 位的发生器，受 BRGH(TXSTA<2>)位控制。波特率计算公式如下：

BOUD_CLK	BGRM	BGRH	波特率公式
0	0	0	$F_{OSC}/[64*(SPBRG-1)]$
0	0	1	$F_{OSC}/[16*(SPBRG-1)]$
0	1	0	$F_{OSC}/[4*(SPBRG-1)]$
0	1	1	$F_{OSC}/[2*(SPBRG-1)]$
1	0	0	$BCLK/[64*(SPBRG-1)]$
1	0	1	$BCLK/[16*(SPBRG-1)]$
1	1	0	$BCLK/[4*(SPBRG-1)]$
1	1	1	$BCLK/[2*(SPBRG-1)]$

针对工作在异步模式下，工作频率 F_{osc} 为 16MHz，采用 8 位 BRG，目标波特率为 9600bps 的器件：

$$\text{目标波特率} = F_{osc}/(64([SPBRG]+1))$$

求解 SPBRG：

$$\begin{aligned} X &= ((F_{OSC}/\text{目标波特率})/64)-1 \\ &= ((16000000)/9600/64)-1 \\ &=[25.042]=25 \end{aligned}$$

$$\begin{aligned} \text{计算得到的波特率} &= 16000000/(64(25+1)) \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{误差} &= (\text{波特率计算结果}-\text{目标波特率})/\text{目标波特率} \\ &= (9615-9600)/9600=0.16 \text{ 误差} \end{aligned}$$

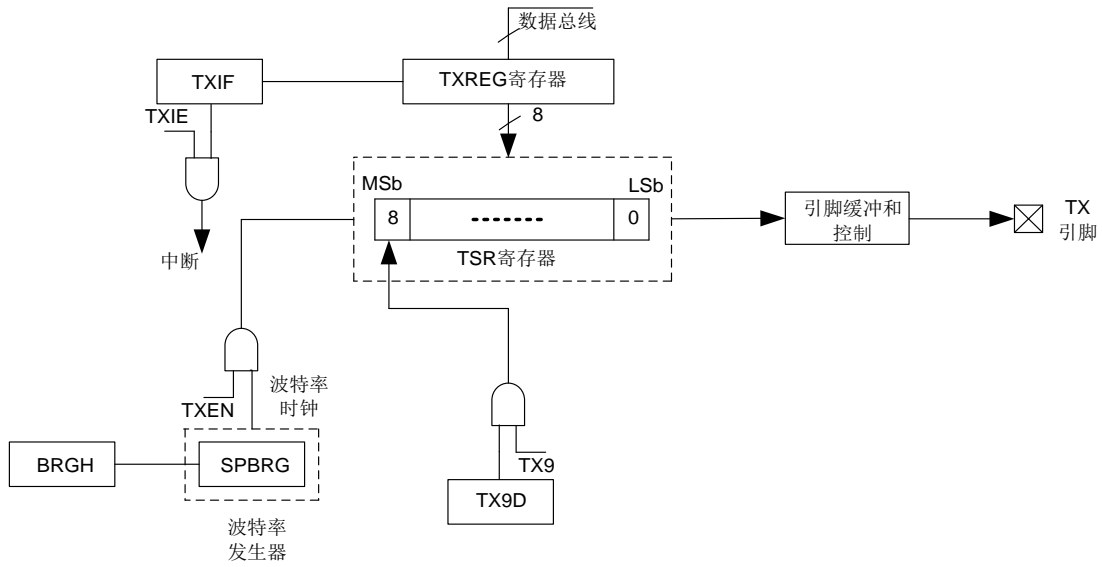
11.2 异步发送器

发送器的核心是发送（串行）移位寄存器（Transmit Shift Register, TSR）。移位寄存器从读/写发送缓冲寄存器 TXREG 中获取数据。TXREG 寄存器中的数据由软件装入。直到前一次装入的停止位已被发送，才会向 TSR 寄存器装入新数据。一旦停止位发送完毕，TXREG 寄存器中的新数据（如果有的话）就会被装入 TSR。

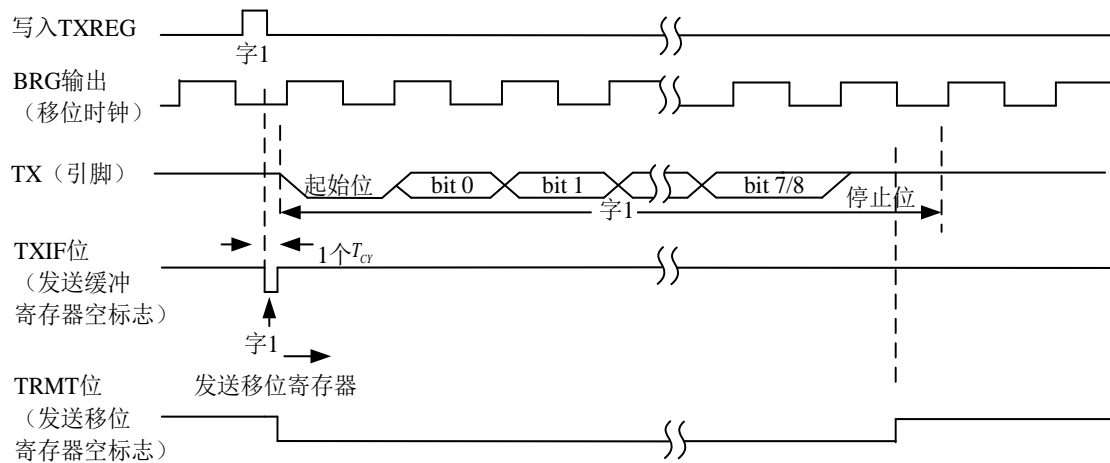
一旦 TXREG 寄存器向 TSR 寄存器传输了数据（在 1 个 T_{CY} 内发生），TXREG 寄存器就为空，同时标志位 TXIF (PIR1<6>) 置 1。可以通过将中断使能位 TXIE (PIE1<6>) 置 1 或清零来使能/禁止该中断。不管 TXIE 的状态如何，只要中断发生，TXIF 就会置 1 并且不能用软件清零。TXIF 不会在 TXREG 装入新数据时立即被清零，而是在装入指令后的第二个指令周期复位。因此在 TXREG 装入新数据后立即查询 TXIF，会返回无效结果。标志位 TXIF 指示的是 TXREG 寄存器的状态，而另一个位 TRMT (TXSTA<1>) 则指示 TSR 寄存器的状态。TRMT 是只读位，它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断均无关联，因此要确定 TSR 寄存器是否为空，用户只能对此位进行轮询。

设置异步发送的操作步骤如下：

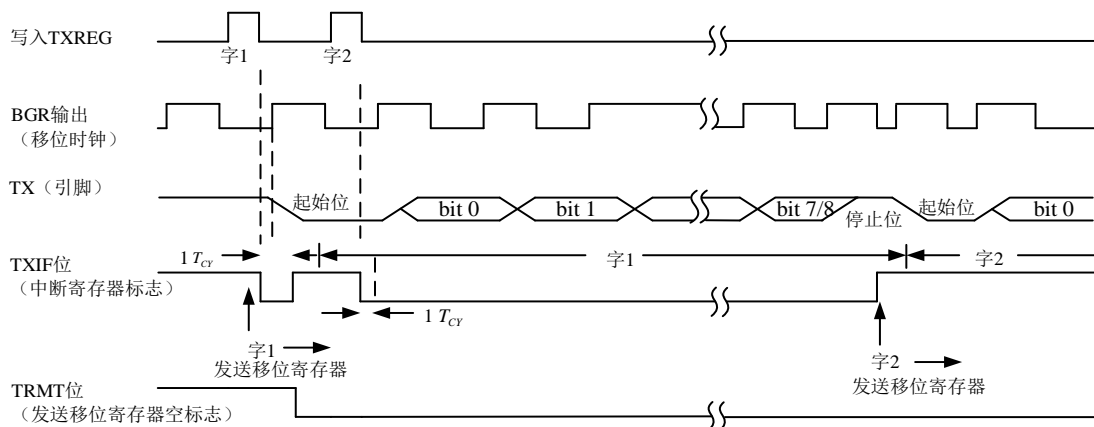
- 1) 初始化 SPBRG，设置合适的波特率。按需要将 BRGH 位置 1 或清零，以获得目标波特率。
- 2) 如果需要中断，将使能位 TXIE 置位。
- 3) 若需要发送 9 位数据，将发送位 TX9 置 1。发送的第 9 位可以是地址位也可以是数据位。
- 4) 通过将 TXEN 位置 1 使能发送，此操作同时也会将 TXIF 位置 1。
- 5) 如果选择发送 9 位数据，应该将第 9 位数据装入 TX9D 位。
- 6) 将数据装入 TXREG 寄存器（启动发送）。
- 7) 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。



UART 发送原理图



异步发送



异步发送（背对背）

11.3 异步接收

如 UART 接收原理图所示，在 RX 引脚上接收数据，并驱动数据恢复电路。数据恢复电路实际上是一个以 16 倍波特率为工作频率的高速移位器，而主接收串行移位器的工作频率等于比特率或 FOSC。此模式通常用于 RS-232 系统。

设置异步接收的操作步骤如下：

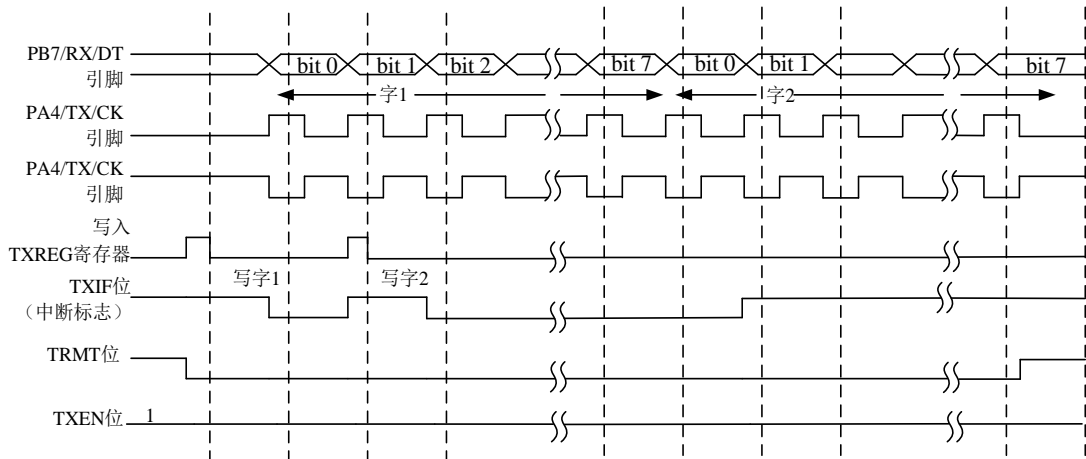
- 1) 初始化 SPBRG，设置合适的波特率。按需要将 BRGH 位置 1 或清零，以获得目标波特率。
- 2) 如果需要中断，将使能位 RCIE 置位。
- 3) 若需要发送 9 位数据，将发送位 RX9 置 1。
- 4) 通过将 CREN 位置 1，使能接收。
- 5) 当接收完成时标志位 RCIF 将被置 1，此时如果使能位 RCIE 已值 1，还将产生一个中断。
- 6) 读 RCSTA 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中是否发生了错误。
- 7) 通过读 RCREG 寄存器来读取接收到的 8 位数据。
- 8) 如果发生错误，通过将使能位 CREN 清零来清除错误。
- 9) 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

前一次装入数据的最后一位发送完成后，才会向TSR 寄存器装入新数据。一旦最后一位发送完成，就会将TXREG 寄存器中的新数据（如果有的话）装入TSR。

一旦TXREG 寄存器向TSR 寄存器传输了数据（在1个TCY 内发生），TXREG 寄存器就为空，同时标志位TXIF（PIR1<6>）被置1。可以通过将中断使能位TXIE（PIE1<6>）置1 或清零来使能/ 禁止该中断。TXIF 的设置与使能位TXIE 的状态无关，且不能用软件清零。只有在新数据写入TXREG 寄存器时，TXIF 才会复位。TXIF 表示的是TXREG 寄存器的状态，而另一个标志位TRMT（TXSTA<1>）则表示TSR 寄存器的状态。TRMT是只读位，它在TSR 寄存器为空时被置1。TRMT 位与任何中断均无关联，因此要判断TSR 寄存器是否为空，用户只能对此位进行轮询。TSR 寄存器并未映射到数据存储器中，因此用户不能直接访问它。

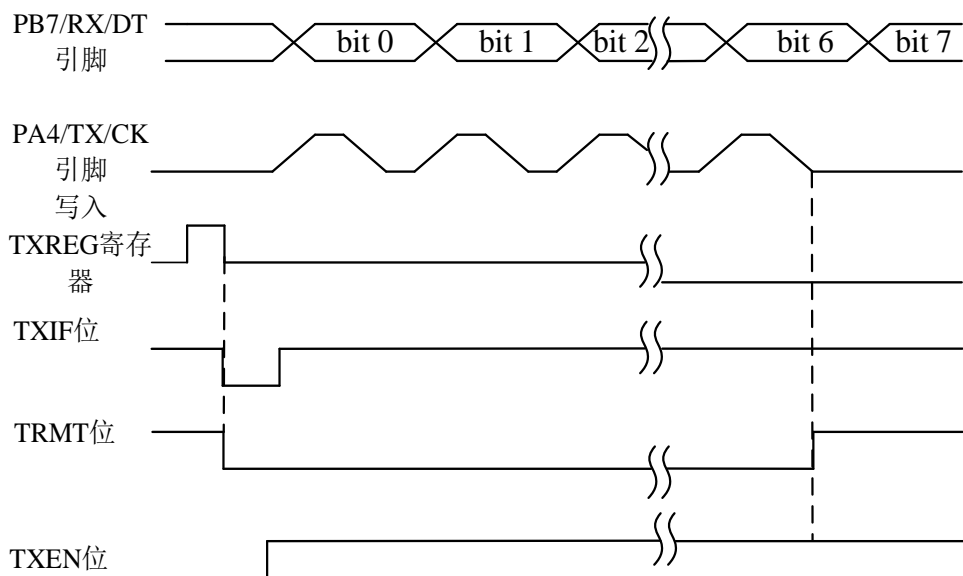
设置同步主控发送操作的步骤如下：

- 1) 初始化SPBRGH:SPBRG 寄存器，设置合适的波特率。按需要将BRG16 位置1 或清零，以获得目标波特率。
- 2) 通过将SYNC、SPEN 和CSRC 位置1，使能同步主控串口。
- 3) 若需要中断，将中断使能位TXIE 置1。
- 4)若需要发送9 位数据，将TX9 位置1。
- 5)将TXEN 位置1，使能发送。
- 6)如果选择发送9位数据，将第9位数据装入TX9D位。
- 7)将数据装入TXREG 寄存器，启动发送。
- 8)若想使用中断，请确保将INTCON 寄存器中的GIE 和PEIE 位（INTCON<7:6>）置1。



注：同步主控模式，SPBRG=0，连续发送两个8位字节

同步发送时序



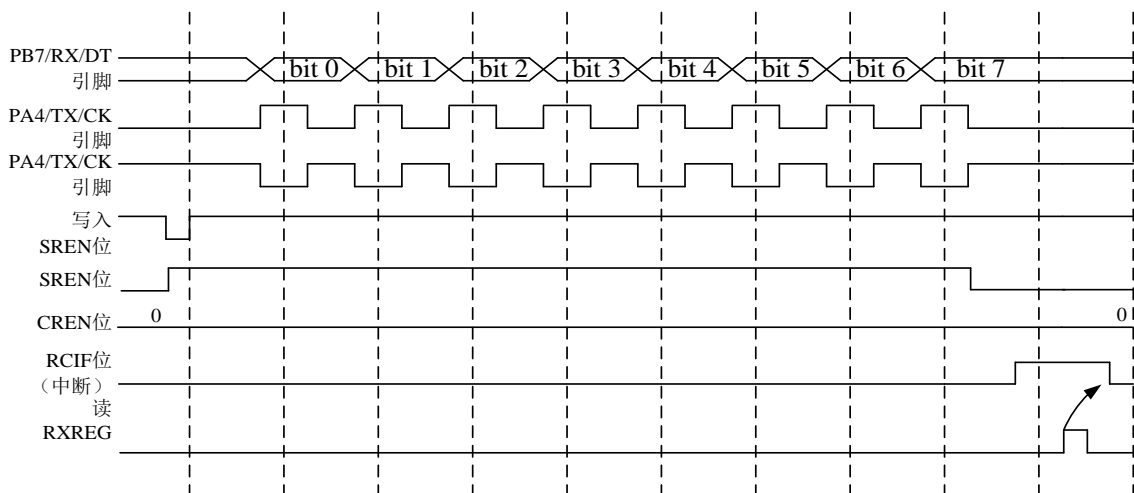
同步发送时序（有TXEN位控制）

11.4.2 同步主控接收

一旦选择了同步模式，只要将单字节接收使能位SREN（RCSTA<5>）或连续接收使能位CREN（RCSTA<4>）置1，即可使能接收。在时钟的下降沿采样RX引脚上的数据。如果将使能位SREN置1，则只接收单个字。如果将使能位CREN置1，则会连续接收数据，直到将CREN位清零。如果两个位均被置1，则CREN具有优先权。

设置同步主控接收操作的步骤如下：

- 1) 初始化SPBRGH:SPBRG 寄存器，设置合适的波特率。按需要将BRG16 位置1 或清零，以获得目标波特率。
- 2) 通过将SYNC、SPEN 和CSRC 位置1，使能同步主控串口。
- 3)确保将CREN 和SREN 位清零。
- 4)若需要中断，将中断使能位RCIE 置1。
- 5)若需要接收9 位数据，将RX9 位置1。
- 6)若需要单字节接收，将SREN 位置1 ；若需要连续接收，将CREN 位置1。
- 7)当接收完成时中断标志位RCIF 将置1，此时如果中断使能位RCIE 已置1，还将产生一个中断。
- 8)读RCSTA 寄存器获取第9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
- 9)通过读RCREG寄存器来读取接收到的8位数据。
- 10)如果发生错误，将CREN 位清零以清除错误。
- 11)若想使用中断，请确保将INTCON寄存器中的GIE和PEIE 位 (INTCON<7:6>) 置1。



注：时序图说明了SREN=1并且BRGH=0时的同步主控模式

主控模式同步接收的时序（由SREN位控制）

11.5 同步从动模式

将CSRC (TXSTA<7>) 位清零可进入同步从动模式。此模式与同步主控模式的区别在于移位时钟由CK 引脚上的外部时钟提供(主控模式中由内部时钟提供)。这使得器件能在任何低功耗模式下发送或接收数据。

11.5.1 同步从动发送

除了休眠模式以外,同步主控、从动模式的工作方式是完全相同的。如果向TXREG 写入两个字,然后执行SLEEP 指令,则将发生以下事件:

- a) 第一个字立即传送到TSR 寄存器进行发送。
- b) 第二个字仍保留在TXREG 寄存器中。
- c) 不会将标志位TXIF 置1。
- d) 当第一个字移出TSR 后, TXREG 寄存器将把第二个字送入TSR, 同时将标志位TXIF 置1。
- e) 如果中断使能位TXIE 已置1, 中断将把器件从休眠状态唤醒。如果使能了全局中断, 程序则会跳转到中断向量处执行。

设置同步从动发送操作的步骤如下:

- 1) 通过将SYNC 和SPEN 位置1 并将CSRC 位清零使能同步从动串口。
- 2) 将CREN 和SREN 位清零。
- 3) 若需要中断, 将中断使能位TXIE 置1。
- 4) 若需要发送9 位数据, 将TX9 位置1。
- 5) 将使能位TXEN 置1 使能发送。
- 6) 如果选择发送9位数据, 将第9位数据装入TX9D位。
- 7) 将数据装入TXREG 寄存器, 启动发送。
- 8) 若想使用中断, 确保将INTCON 寄存器中的GIE和PEIE 位(INTCON<7:6>) 置1。

11.5.2 同步从动接收

除了休眠模式、空闲模式以及在从动模式下忽略SREN位以外，同步主控和同步从动模式的工作方式完全相同。如果在进入休眠或空闲模式前将CREN 位置1，使能接收，那么在该低功耗模式下可以接收到一个数据字。接收到该字后，RSR 寄存器将把数据传输到RCREG寄存器，如果中断使能位RCIE 已置1，产生的中断将把器件从低功耗模式唤醒。如果使能了全局中断，程序则会跳转到中断向量处执行。

设置同步从动接收操作的步骤如下：

- 1)通过将SYNC 和SPEN 位置1 并将CSRC 位清零使能同步从动串口。
- 2)若需要中断，将中断使能位RCIE 置1。
- 3)若需要接收9 位数据，将RX9 位置1。
- 4)将使能位CREN 置1，使能接收。
- 5)当接收完成时，RCIF 位将被置1。如果使能位RCIE 已置1，还将产生一个中断。
- 6)读RCSTA 寄存器获取第9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
- 7)通过读RCREG寄存器来读取接收到的8位数据。
- 8)如果发生错误，将CREN 位清零以清除错误。
- 9)若想使用中断，请确保将INTCON 寄存器中的GIE 和PEIE 位（INTCON<7:6>）置1。

12 8×8 硬件乘法器和 16/16 硬件除法器

AD1845 包含一个 8x8 硬件乘法器。该乘法器可执行无符号运算并产生一个 16 位运算结果，该结果存储在—对乘积寄存器 PRODH:PRODL 中。该乘法器执行的运算不会影响状态寄存器中的任何标志。

通过硬件执行乘法运算只需要 1 个指令周期。硬件乘法器具有更高的计算吞吐量并减少了乘法算法的代码长度，从而可在许多先前仅能使用数字信号处理器的应用中使用 AD1845 器件。

AD1845 包含一个 16/16 硬件除法器。该除法器可执行无符号运算并产生一个 16 位的商和余数，该结果存储在—对商寄存器余数寄存器中。该除法器器执行的运算不会影响状态寄存器中的任何标志。

通过硬件执行除法运算只需要 9 个指令周期。硬件除法器具有更高的计算吞吐量并减少了除法算法的代码长度，从而可在许多先前仅能使用数字信号处理器的应用中使用 AD1845 器件。

被除数高 8 位寄存器

0xf4f	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVIDENDH								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

被除数低 8 位寄存器

0xf4e	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVIDENDL								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

除数高 8 位寄存器

0xf4d	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVISORH								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

除数低 8 位寄存器

0xf4c	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVISORL								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

商高 8 位寄存器

0xf4b	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVISORH								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

商低 8 位寄存器

0xf4a	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVISORL								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

余数高 8 位寄存器

0xf49	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
REMAINDERH								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

余数低 8 位寄存器

0xf48	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
REMAINDERL								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

0xf47	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIVCON		-	-	-	DIV0	COMPLETE	BUSY	START
R/W		-	-	-	R	R	R/W	R/W
初始值		0	0	0	0	0	0	0

BIT[3] DIV0 – 除数为 0 标志

- 1: 为 0
- 0: 不为 0

BIT[2] COMPLETE– 除法完成标志

- 1: 完成
- 0: 运行中

BIT[1] BUSY– 除法运算忙标志

- 1: 忙
- 0: 完成

BIT[0] START– 除法运算启动控制位

- 1: 启动，硬件自动清 0
- 0: 不启动

13 固定参考电压 (FVR)

0xF36	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FVRICON0	-	-	-	-	-	FVROUTEN	FVRPGA	FVREN
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	0	0	0

BIT[2] FVROUTEN – FVR 输出使能

1: 使能 FVR 输出

0: 禁止 FVR 输出

BIT[1] FVRPGA – FVR 电压放大 1 倍选择位

1: 使能 FVR 电压放大

0: 禁止 FVR 电压放大

BIT[0] FVREN – FVR 使能

1: 使能 FVR

0: 禁止 FVR

0xF35	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FVRICON1	-	-	-	FVR_SEL4	FVR_SEL3	FVR_SEL2	FVR_SEL1	FVR_SEL0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4:0] FVR_SEL4:0 – FVR 电压选择位

.....

01001: 1.0v

01010: 1.1v

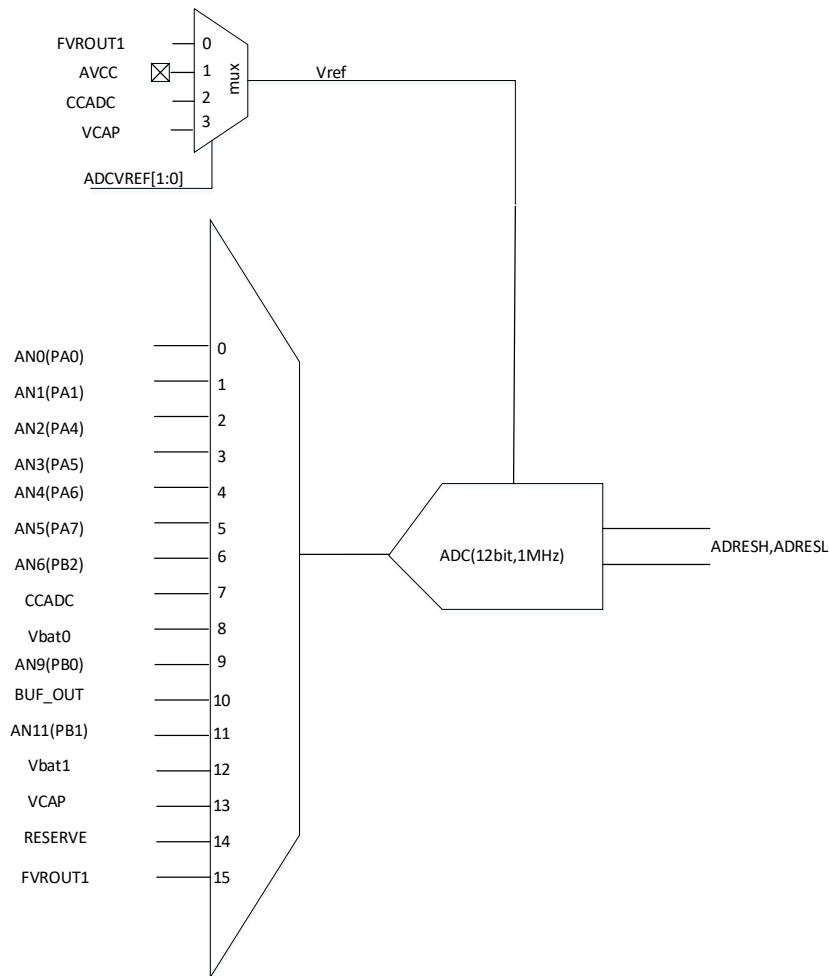
01011: NTC(此通道电压和 IC 温度负相关)

.....

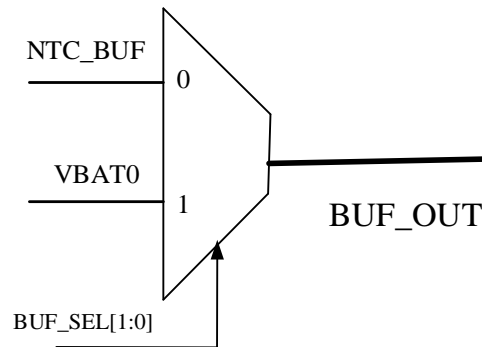
01111: VCAP

14 A/D 转换器

AD1845 包含一个 16 通道输入的 12 位 ADC，能够将一个模拟输入转换成 12 位数字信号。在根据需要配置好 A/D 模块之后，必须在转换开始之前对选定的通道进行采样。采集时间一结束，即可启动 A/D 转换。当 A/D 转换完成之后，转换结果被装入 ADRESH:ADRESL 寄存器对，GO/DONE 位被清零且 A/D 中断标志位 ADIF 被置 1。



A/D 结构图



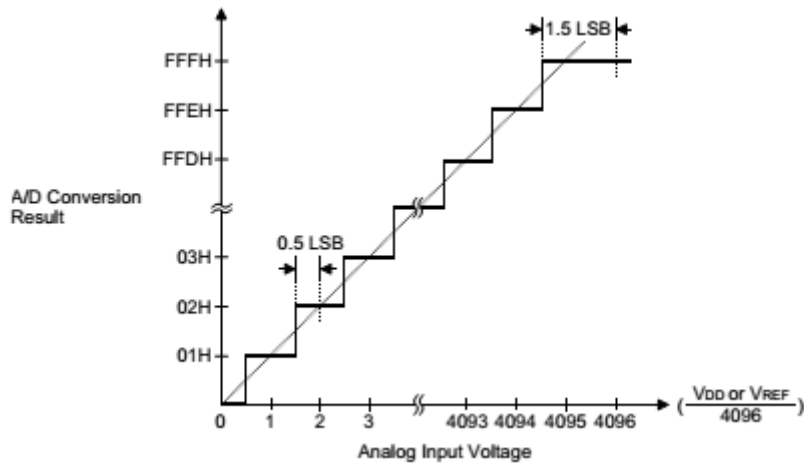
NTCBUF 结构图

注:需要采集 NTCBUF、VBAT0 通道的电压时, 根据 BUF_OUT 的结构图, 在 FVR1CON2 寄存器中使能以及选取相应的信号, 进行采集。

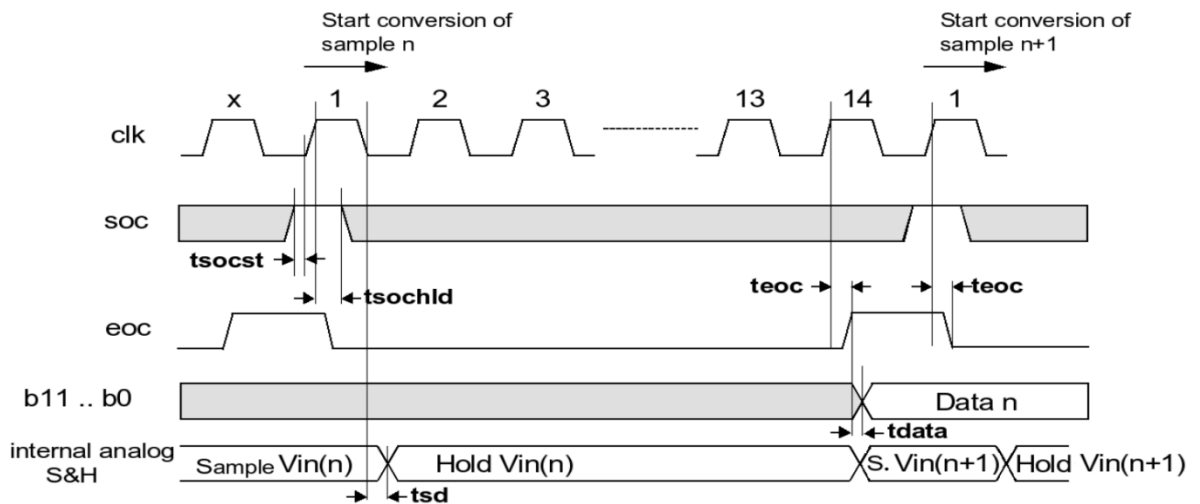
执行 A/D 转换步骤:

- 1) 配置 A/D 模块
 - 选择参考电压 (通过 ADCON1[7:6]寄存器)
 - 选择 A/D 输入通道 (通过 ADCON0[5:2]寄存器)
 - 选择 A/D 采集时间 (通过 ADCON1[5:3]寄存器)
 - 选择 A/D 转换时间 (通过 ADCON1[2:0]寄存器)
 - 使能 A/D 模块 (通过 ADCON0[0]寄存器)
- 2) 需要时, 配置 A/D 中断
 - 清零 ADIF 位
 - 将 ADIE 位置 1
 - 将 GIE 位置 1
- 3) 如果需要, 需等待所需的采集时间。
- 4) 启动转换: _____
 - 将 GO/DONG 位置 1 (ADCON0[1])
- 5) 等待 A/D 转换完成, 通过以下两种方法之一判断转换是否完成:
 - 查询 GO/DONE 位是否被清零
 - 或
 - 等待 A/D 中断
- 6) 读取 A/D 结果寄存器 (ADRESH:ADRESL), 需要时将 ADIF 位清零。

7) 如需再次进行 A/D 转换, 返回步骤 1 或步骤 2。



A/D 转换方式



A/D 转换时序图

与 A/D 相关的寄存器说明:

0xfa8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON0	ADFM	ADSP	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[5:2]

CHS3:CHS0 – 模拟通道选择位

0000: 通道 0

0001: 通道 1

0010: 通道 2

0011: 通道 3
 0100: 通道 4
 0101: 通道 5
 0110: 通道 6
 0111: 通道 7
 1000: 通道 8
 1001: 通道 9
 1010: 通道 10

 1111: 通道 15

BIT[1] GO/DONG – A/D 转换状态位
 当 ADON=1 时:
 1: A/D 转换正在进行
 0: A/D 空闲

BIT[0] ADON – A/D 模拟使能位
 1: 使能 A/D 转换器模块
 0: 禁止 A/D 转换器模块

0xfa7	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON1	ADVREF1	ADVREF0	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:6] ADVREF1:ADVREF0 – A/D 参考电压选择位
 00: FVR
 01: AVCC
 10: CCADC
 11: VCAP

BIT[5:3] ACQT2:ACQT0 – A/D 采集时间选择位
 111: 15 TADCLK
 110: 13 TADCLK
 101: 11 TADCLK
 100: 9 TADCLK

011: 7 TADCLK

010: 5 TADCLK

001: 3 TADCLK

000: 0 TADCLK

BIT[2:0] ADCS2:ADCS0 – A/D 转换时钟选择位

110: FOSC/64

101: FOSC/16

100: FOSC/4

011: CLK32K

010: FOSC/32

001: FOSC/8

000: FOSC/2

0xfa1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE1	RCIE	TXIE	-	-	-	TMR2IE	TMR1IE	TMR0IE
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

BIT[6] ADIE – A/D 中断允许位

1: 允许 A/D 中断

0: 禁止 A/D 中断

0xfa2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR1	RCIF	TXIF	-	-	-	TMR2IF	TMR1IF	TMR0IF
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	0	0	-	-	-	0	0	0

BIT[6] ADIF – A/D 中断标志位

1: 一次 A/D 转换完成（必须用软件清零）

0: A/D 转换未完成

0xfa3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPR1	RCIP	TXIP	-	-	-	TMR2IP	TMR1IP	TMR0IP
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
初始值	1	1	-	-	-	1	1	1

BIT[6] ADIP – A/D 中断优先级

1: 高优先级

0: 低优先级

0xF37	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FVRICON2	-	BAT1SEL	BAT1EN	BAT0EN	BUFSEL2	BUFSEL1	BUFSEL0	BUFEN
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	0	0	0	0	0	0	0

BIT[0] BUFEN – BUF_OUT 使能

1: 使能 BUF_OUT

0: 禁止 BUF_OUT

BIT[3:1] BUFSEL2:BUFSEL0 – BUF_OUT 通道选择

001: NTCBUF

010: VBAT0

100: VBAT1

BIT[4] BAT0EN – BAT0 使能

1: 使能 BAT0

0: 禁止 BAT0

BIT[5] BAT1EN – BAT1 使能

1: 使能 BAT1

0: 禁止 BAT1

BIT[6] BAT1SEL – BAT1 电压的输出

1: 1/8VBAT1

0: 1/4VBAT1

注：默认 BAT0 是 1/4VBAT0 电压输出，如需要采集 BAT0 电压时，则电压要采集 4 次的累加值。

15 ADC 电荷计数器- Sigma-delta 型计量标准的专用型模数转换器

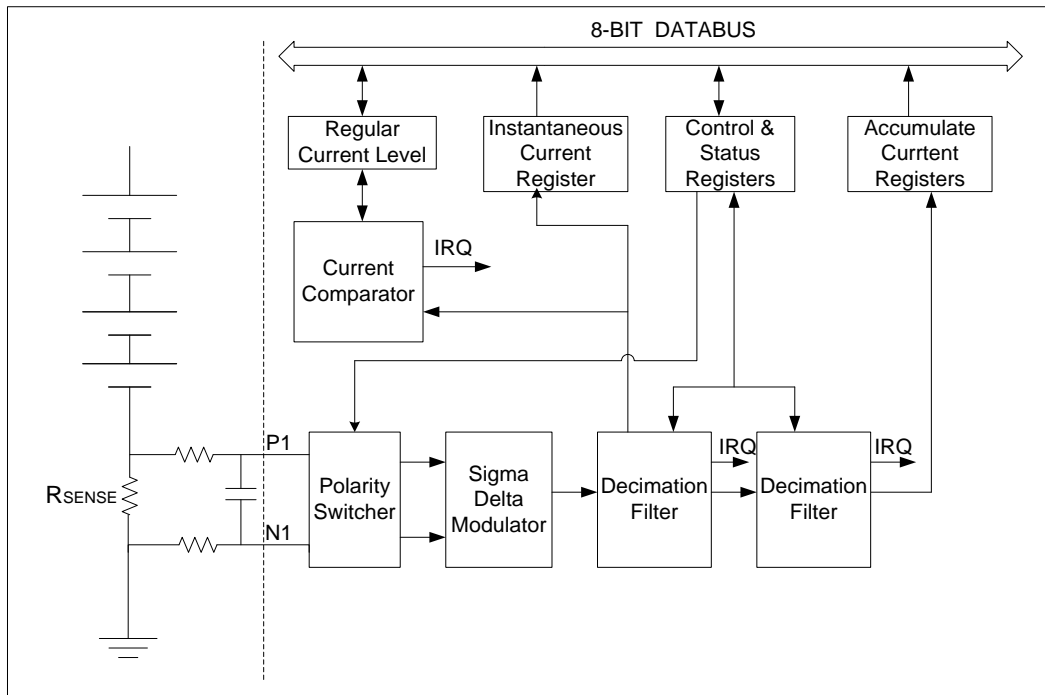
15.1 特性

- 采样系统电荷计数器
- 经过低功耗优化的Sigma-delta模数转换器
- 3.9毫秒内输出瞬时电流
 - 13位的分辨率（包括符号）
 - 瞬时电流转换完成后产生中断
- 累加电流输出
 - 可编程转换时间：128/256/512/1000毫秒
 - 18位的分辨率（包括符号）
 - 瞬时电流转换完成后产生中断
- 正常电流检测模式-
 - 可编程采样间隔时间：256/512/1000/2000毫秒
- 可编程输入电压范围 $\pm 100/200$ 毫伏
 - 允许测量 $\pm 20/40\text{A} @ 5\text{M}\omega$
- 切换输入极性消除偏移

15.2 概述

AD1845的Sigma-delta型计量标准专用模数转换器（CC-ADC）对电荷计数器进行了优化，通过采样流过外部感应电阻的充电/放电电流，CC-ADC追踪流入和流出电池的电流。

图21-1. 电荷计数器框图



CC-ADC在可编程电压范围内，需要对动态范围和外部检测电阻RSENSE之间进行权衡。

在正常转换模式下CC-ADC提供了两个不同的输出值，分别是瞬时电流和累加电流。采用低分辨率进行模数转换，瞬时电流的转换时间短。累加电流是采用高精度电流测量方法的输出结果。

CC-ADC提供一种特殊的常规电流检测模式，当CPU充电/放电电流小的时候，CPU工作在省电模式中执行超低功耗操作。在CPU运行期间，切换输入信号极性能够消除偏移。以上特性可以消除CC-ADC的内部偏移。

15.3 普通操作

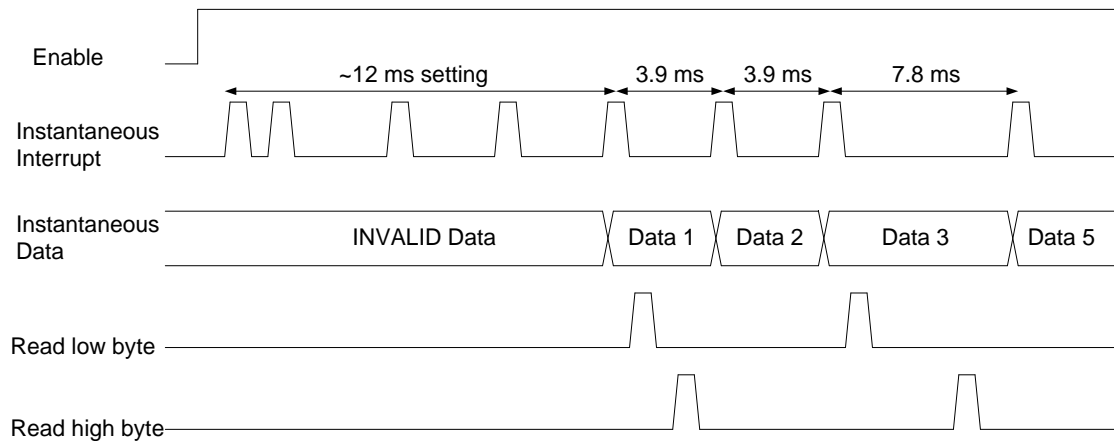
使能CC-ADC，CC-ADC持续测量外部感应电阻RSENSE上的电流。在普通转换模式下，CC-ADC输出两个数据转换结果：

- 瞬时转换结果
- 累加转换结果

瞬时电流转换时间固定为3.9毫秒(典型值)，电流输出值与输入值紧密相连。使能中断，则在每次瞬时电流转换完成后会产生中断。当发生中断时，瞬时电流

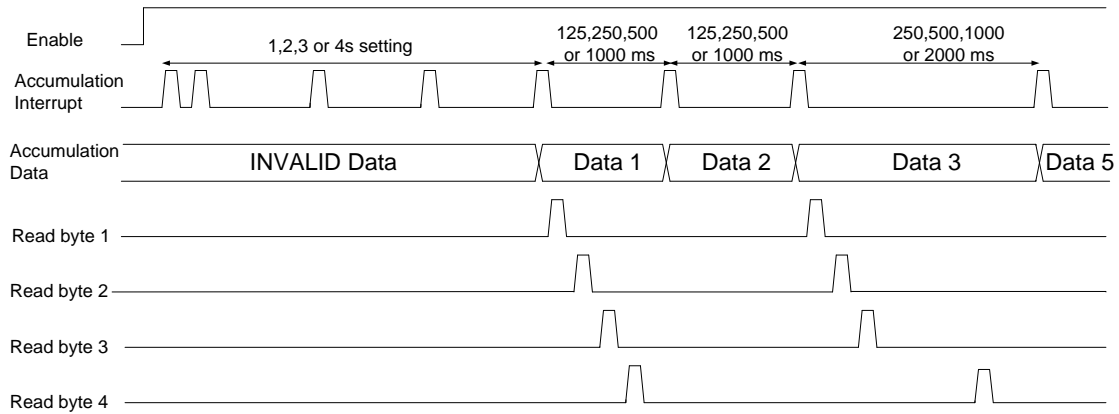
寄存器CADICL和CADICJH同时更新转换数据。为避免丢失转换数据，在中断发生期间，必须在3.9毫秒的时序窗口内读取低字节和高字节数据。低字节寄存器数据被读取，在高字节数据被读取前，瞬时电流寄存器停止更新数据，并且停止中断。图21-2为瞬时电流转换图，如图所示由于在有限周期内没有完成DATA3读取，导致DATA4消失。

图21-2. 瞬时电流转换



累加电流输出是高分辨率、高精度的输出结果，其可编程转换时间由CADCSRA中的CADAS位选择。转换值是在一个转换周期内，测量平均电流的准确测量值。使能中断，每当新的累加电流转换完成后，CC-ADC就会产生中断。当中断发生时，转换值将在累加电流寄存器CADAC0,CADAC1, CADAC2和CADAC3中同时更新。为避免丢失转换数据，必须在选定的转换周期内读取所有字节数据。低字节寄存器数据被读取，在最高字节数据被读取前，瞬时电流寄存器停止更新数据，并且停止中断。图21-3为累加电流转换示例，如图所示由于在有限周期内没有完成DATA3读取，导致DATA4消失。

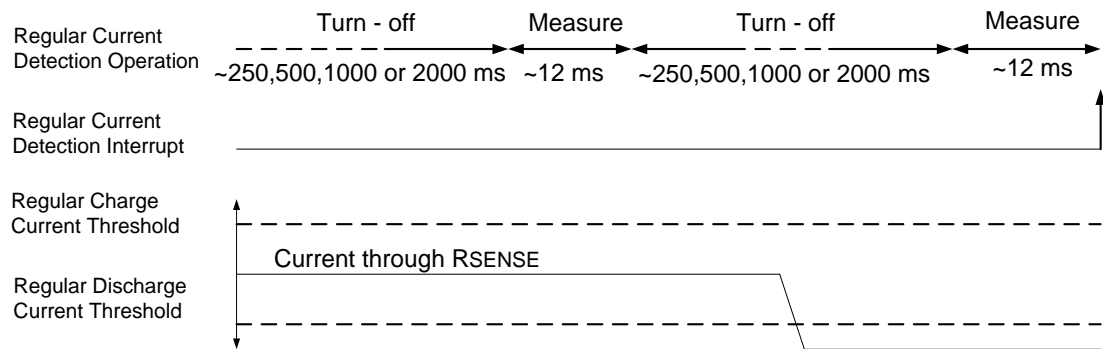
图21-3. 累加电流转换



15.4 常规电流检测

通过设置CADCSRA寄存器中的CADSE位，CC-ADC将进入特殊的常规电流检测采样模式，在这种模式下，CC-ADC以常规采样间隔时间进行一次瞬时电流转换，并且累加电流寄存器自动停止更新。配置CADCSRA寄存器中的CADSI位，可以控制采样间隔时间。每次转换完成后，转换结果与CADRCC/CADRDC寄存器中指定的常规充电/放电阈值进行比较。使能中断，电压高于或低于规定限制值，将会产生常规电流检测中断。图21-4显示常规电流检测模式

图21-4. 常规电流检测模式(CADSE=1)



15.5 切换极性消除偏移

CC-ADC通过切换极性消除内部偏移。在选定的时间间隔内切换采样输入信号的极性，将在CC-ADC的输出端消除内部电压偏移。转换极性可防止CC-ADC随时间累积造成偏移误差。

15.6 配置和使用

当CC-ADC正在转换时，CPU可以进入睡眠模式并等待中断。给电荷计数添加转换数据后，CPU可以再次回到睡眠状态，这样减少了CPU的工作量，大多数时间内CPU处于低功耗模式，降低了整体功耗。在使用CC-ADC之前，必须单独使能带隙电压基准。CADEN清零，CC-ADC则不消耗能量。因此当电荷计数器或常规电流检测功能未使能时，建议关闭CC-ADC。在断电模式下，CC-ADC被自动禁用。

设置CADEN比特位，使能CC-ADC后，由于前四个转换信息中不包含有用信息，因此这四个转换信息被忽略掉。同样的，清除CADSE位，或改变CADPOL、CADVSE位后，均可忽略前四个转换信息。转换时间和采样间隔由超低功率RC振荡器控制，并且转换时间和采样间隔的大小取决于RC振荡器的实际频率。为了获得准确的电荷计数结果，需要计算实际的转换时间。

15.7 寄存器描述

15.7.1 CADCSRA – CC-ADC 控制/状态寄存器 A

0xf77	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADCSRA	CADEN	CADPOL	CADUB	CADAS1	CADAS0	CADSI1	CADSI0	CADSE
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] CADEN: CC-ADC 使能

CADEN 位清零，CC-ADC 被禁用，任何正在进行的转换将被终止。CADEN 位设置为 1，CC-ADC 持续测量外部感应电阻 RSENSE 上的压降。在断电模式下，CC-ADC 总是被禁用，需要注意的是，带隙电压基准必须单独使能。

BIT[6] CADPOL: CC-ADC 极性

CADPOL 用于改变 Sigma Delta 调制解调器的输入采样极性，CADPOL 设置为 1，输入采样极性为负；CADPOL 设置为 0，输入采样极性为正。

BIT[5] CADUB: CC-ADC 更新忙碌

CC-ADC 运行的时钟域与 CPU 的时钟域不同，无论何时向 CADCSRA、CADCSRC、CADRCC 或 CADRDC 写入新值，新值必须同步到 CC-ADC 时钟域，在同步期间，这些寄存器的后续写入值将被阻塞，同步一个寄存器会阻碍其他寄存器的更新。同步这些寄存器中

的任何一个寄存器，CADUB 读为 1，当两个寄存器都没有被同步时，CADUB 读为 0。

BIT[4:3] CADAS1:0: CC-ADC 累加电流选择

CADAS 选择累加电流输出的转换时间，如表 21-1 所示。

表 21-1. CC-ADC 累加电流转换时间

CADAS1:0	CC-ADC 累加电流转换时间	CC-ADC clock Cycles
00	128ms	4096
01	256ms	8192
10	512ms	16384
11	1s	32768

BIT[2:1] CADSI1:0: CC-ADC 电流采样间隔

CADSI 确定常规电流检测的电流采样间隔，如表 21-2 所示。

表 21-2 CC-ADC 常规电流采样间隔

CADAS1:0	CC-ADC 常规电流采样间隔	CC-ADC clock Cycles
00	256ms (+ sampling time)	8192 (+ sampling time)
01	512ms (+ sampling time)	16384 (+ sampling time)
10	1s (+ sampling time)	32768 (+ sampling time)
11	2s (+ sampling time)	65536 (+ sampling time)

此时，LIRC 分频 LVDIVOSC 应设置为 8 分频

• BIT[0] CADSE: CC-ADC 采样使能

CADSE 为 1，正在进行的 CC-ADC 转换被中止，CC-ADC 进入常规电流检测模式。

15.7.2 CADCSRB – CC-ADC 控制/状态寄存器 B

0xf76	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADCSRB	-	CADACIE	CADRCIE	CADICIE	-	CADACIF	CADRCIF	CADICIF
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[6] CADACIE: CC-ADC 累加电流中断使能

CADACIE 设置为 1，状态寄存器的第 I 位设置为 1，则使能 CC-ADC 的累加电流中断。

BIT[5] CADRCIE: CC-ADC 常规电流中断使能

CADRCIE 设置为 1，状态寄存器的第 I 位设置为 1，则使能 CC-ADC 的常规电流中断。

BIT[4] CADICIE: CC-ADC 瞬时电流中断使能

设置为 1，状态寄存器的第 I 位设置为 1，则使能 CC-ADC 的瞬时电流中断。

BIT[2] CADACIF: CC-ADC 累加电流中断标志

累加电流转换完成后，CADACIF 被置 1。CADACIE 设置为 1，执行 CC-ADC 累加电流中断操作。对 CADACIF 写入逻辑“0”可清除 CADACIF 标志。

BIT[1] CADRCIF: CC-ADC 常规电流中断标志

如果最后一次 CC-ADC 转换结果的绝对值大于或等于常规充电/放电电流级寄存器设置的比较值，则 CADRCIF 置 1。正值与常规充电电流标准值比较，负值与常规放电电流标准值比较。CADRCIE 设置为 1，执行 CC-ADC 常规电流中断操作。对 CADRCIF 写入逻辑“0”可清除 CADRCIF 标志。

BIT[0] CADICIF: CC-ADC 瞬时电流中断标志

CC-ADC 瞬时电流转换完成后，CADICIF 位置 1。CADICIE 设置为 1，执行 CC-ADC 瞬时电流中断操作。对 CADICIF 写入逻辑“0”可清除 CADICIF 标志。

15.7.3 CADCSRC – CC-ADC 控制/状态寄存器 C

0xf75	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADCSRC	RCCCNT[3:0]				-	-	CADBUFEN	CADVSE
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
初始值	0	1	0	1	0	0	0	0

BIT[7:4] RCCCNT: CC-ADC 模式工作计数器，要求大于等于 5（用于在常规电流模式设定次数之后才产生 RCC 中断下防止误触发）

BIT[1] BBUFFER: CC-ADC 内部基准使能

设置 CADVSE，使能内部电压缩放。使能内部电压缩放后，CC-ADC 内部参考值除以

2, 影响输入电压范围和步长。表 21-3 显示 CADVSE 设置的输入电压范围和转换步长。

BIT[0] CADVSE: CC-ADC 电压缩放使能

设置 CADVSE, 使能内部电压缩放。使能内部电压缩放后, CC-ADC 内部参考值除以

2, 影响输入电压范围和步长。表 21-3 显示 CADVSE 设置的输入电压范围和转换步长。

表21-3. CADVSE设置输入电压范围和转换步长

CADVSE	Voltage	Step-size CADAC	Step-size CADIC
0	±200mV	1.67 μV	53.7 μV
1	±100mV	0.84 μV	26.9 μV

假如采集到的CCADC值为1000, 采样电阻为20m Ω 此时对应电流为
 $1000 * 53.7 \mu V / 20 m \Omega = 2.685 mA$ (选择模式CADVSE=0)

15.7.4 CADICH and CADICL – CC-ADC 瞬时电流

CADICH (CC-ADC 瞬时转换结果高 8 位寄存器)

0xf73	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADICH	CC-ADC 瞬时转换结果高 8 位寄存器							
R/W	R/W							
初始值								

CADICL(CC-ADC 瞬时转换结果低 8 位寄存器)

0xf72	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADICL	CC-ADC 瞬时转换结果低 8 位寄存器							
R/W	R/W							
初始值								

CC-ADC瞬时转换完成, 转换结果保存在这两个寄存器中。CADIC[15:0]表示二进制补码格式的转换结果, [12:0]是13位精度ADC的转换结果(包括符号), [15:13]是符号扩展位。

读取CADICL, 直到CADICH被读取之前, CC-ADC瞬时电流寄存器不进行更

新操作。读取CADICL，CADICH的读取值与CADICL一致。转换完成后，两个寄存器必须在下一次转换完成之前被读取，否则将丢失数据。

15.7.5 CADAC3, CADAC2, CADAC1 and CADAC0 – CC-ADC 累加电流

CADAC3

0xf71	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADAC3	CADAC[31:24]							
R/W	R/W							
初始值	0							

CADAC2

0xf70	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADAC2	CADAC[23:16]							
R/W	R/W							
初始值	0							

CADAC1

0xf6F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADAC1	CADAC[15:8]							
R/W	R/W							
初始值	0							

CADAC0

0xf6E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADAC0	CADAC[7:0]							
R/W	R/W							
初始值	0							

CADAC3、CADAC2、CADAC1和CADAC0寄存器包含累积电流的测量结果，测试结果以二进制补码格式保存。[17:0]是18位ADC结果（包括符号），[31:18]是符号扩展位。当CADAC0被读取，直到CADAC3被读取之前，CC-ADC累加电流寄存器不进行更新操作。读取CADAC0、CADAC1、CADAC2、CADAC3，四个寄存器的读取值一致。转换完成后，四个寄存器必须在下一次转换之前被读取，否则将丢失数据。

15.7.6 CADRCC – CC-ADC 常规充电电流

CADRCC								
0xf6D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADRCC	CADRCC[7:0]							
R/W	R/W							
初始值	0							

CC-ADC的常规充电电流寄存器决定了常规充电电流检测的阈值大小。CC-ADC瞬时电流转换结果为正，其值大于或等于常规充电电流标准值，则CC-ADC的常规电流中断标志置1。CADRCC寄存器的值规定常规充电电流标准值的低8位数值，该值以二进制补码格式保存，常规充电电流标准值的最高有效位一直为0。表21-4给出了常规充电电流标准值的可编程范围。

表21-4. 常规充电电流标准值的可编程范围（1）

		Step Size/uV	Minimum/mA	Maximum/mA
Voltage (uV)		53.7/26.9	0	13696/6848
Current (mA)	$R_{SENSE}(SENSE) = 1m\Omega$	53.7/26.9	0	13696/6848
	$R_{SENSE}(SENSE) = 5m\Omega$	10.7/5.37	0	2740/1370
	$R_{SENSE}(SENSE) = 10m\Omega$	5.37/2.685	0	1370/685

注：表21-4的数值包括CADVSE置0和置1的两种情况。

例如：CADPOL=0，充电电流对应的电压为正值，CADVSE=0， $R_{SENSE}=1m\Omega$ 时，CADRDC中存储的值记为D,则设置的充电电流为： $I=D*53.7\mu V/(1m\Omega)=Dc*53.7mA$,当充电电流大于 I 时，CADRCIF置位并触发相应中断。

15.7.7 CADRDC – CC-ADC 常规放电电流

CADRDC 寄存器

0xf6C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CADRDC	CADRDC[7:0]							
R/W	R/W							
初始值	0							

CC-ADC常规放电电流寄存器决定了常规放电电流检测的阈值大小。CC-ADC瞬时电流转换结果为负，其绝对值大于或等于常规放电电流标准值，CC-ADC常规电流中断标志置1。CADRDC寄存器的值规定常规充电电流标准值的低8位数值，该值以二进制补码格式保存，常规放电电流标准值的最高有效位一直为1。表21-5给出了常规放电电流标准值的可编程范围。

表21-5. 常规放电电流标准值的可编程范围

		Step Size/uV	Minimum/mA	Maximum/mA
Voltage (uV)		53.7/26.9	0	13696/6848
Current (mA)	$R_{SENSE}(SENSE) = 1m\Omega$	53.7/26.9	0	13696/6848
	$R_{SENSE}(SENSE) = 5m\Omega$	10.7/5.37	0	2740/1370
	$R_{SENSE}(SENSE) = 10m\Omega$	5.37/2.685	0	1370/685

注：表21-5的数值包括CADVSE置0和置1的两种情况。

CC-ADC常规放电电流寄存器不影响CC-ADC转换完成中断标志置1。

例如：CADPOL=0，放电电流对应的电压为负值，CADVSE=0， $R_{SENSE}=1m\Omega$ 时，CADRCC中存储的值记为D,其补码记为Dc(D反码加1),则设置的放电电流为： $I=Dc*53.7\mu V/(1m\Omega)=Dc*53.7mA$,当放电电流大于 I 时，CADRCIF置位并触发相应中断。

16 系统低功耗 SLEEP 模式

当 SMCR 寄存器中 SE 为 1 时，执行 SLEEP 指令可使 MCU 进入 SLEEP 低功耗工作模式，具体工作模式由 SMCR 的 SM2~SM0 控制。

AD1845 内部存在一个 PWST 寄存器，当系统进入 Power-off 之前，会保存当前一个状态，等系统重新 Power-on 后，可查看该寄存器内容，然后做相应动作。

与低功耗模式相关寄存器:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWST	PWOFFS T7	PWOFFS T6	PWOFFS T5	PWOFFS T4	PWOFFS T3	PWOFFS T2	PWOFFS T1	PWOFFS T0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] PWOFFST7:PWOFFST0 – 记录系统进入 Power-off 前的状态

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMCR	-	-	-	-	SM2	SM1	SM0	SE
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3:1] SM2:SM0 – 休眠模式选择

000: IDLE 模式，空闲模式

011: Power Save 模式，低功耗模式

010: Reserved

100: Power Off 模式，电源关闭模式

BIT[0] SE – 休眠模式使能位

1: 使能休眠模式,硬件自动清零

0: 禁止休眠模式

不同模式下工作的模块

模块	睡眠模式			
	ACTIVE	IDLE	POWER SAVE	POWER OFF
OSC16M	√	√		
OSC1M	√	√	√	√
CPU	√			
OTP	√	√		
SRAM	√	√		
Timer0	√	√	√	

Timer1	√	√	√	
Timer2	√	√		
CCP	√	√		
I2C	√	√		
UART	√	√		
WDT	√	√	√	√
External Interrupt	√	√	√	√
PAIF,PBIF	√	√	√	√
LDO	√	√	√	
BGR	√	√	√	
LVT	√	√	√	
POR	√	√	√	√
IO	√	√	√	√
RESET	√	√	√	√

睡眠模式可以唤醒的事件

模式	外部中断	键盘中断	WDT	Timer0	Timer1	Timer2	CCP	RESET
IDLE	√	√	√	√	√	√	√	√
Power Save	√	√	√	√	√			√
Power Off	√	√	√					√

17 LVD 检测

AD1845 的 LVD 电压有级（2.0V/2.2V/2.4V/2.6V/3.0V/3.6V），可通过 LVDCON[2:0]或配置位 LVT[2:0]进行配置。电压检测电路有一定的回滞特性，通常回滞电压为0.05V左右，例如，如果选择了3.6V 的 LVD电压，则当电源电压下降到约3.6V时LVD 复位有效，而电压需要上升到约 3.65V时LVR复位才会解除。

0xfd4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDCON	-	LVTST	LVDM1	LVDM0	LVT3	LVT2	LVT1	LVT0
R/W	-	R	R	R/W	R/W	R/W	R/W	R/W
初始值	-	0	0	0	0	1	0	1

BIT[6] LVTST – 内部参考电压稳定标志位

1: 表示电压检测逻辑在检测到指定的电压范围时，产生中断标志

0: 表示电压检测逻辑在检测到指定的电压范围时，不会产生中断标志，并且 LVD 中断不被使能

BIT[5:4] LVDM1, LVDM0 电压比较中断

00: 禁止电压比较器

01: VCC 低于阈值电压产生中断

10: VCC 高于阈值电压产生中断

11: VCC 高于阈值电压保护 PWM 的输入

BIT[3:0] LVD3~0 VCC 电压阈值选择

0: 2.0V

1: 2.2V

2: 2.4V

3: 2.6V

4: 2.8V

5: 3.0V

6: 3.2V

7: 3.4V

8: 3.6V

9: 4.0V

10: 4.1V

11: 4.2V

12: 4.3V

13: 4.4V

14 : 4.5V

15 : 4.6V

18 OTP 操作描述

OTP 可以自编程的地址空间从 0000 到 7FFFH, IAPADDR[10: 0] (由 IAPADDRH 高 3 位和 IAPADDRL 组成) 对应 OTP 的地址 0H。IAPDATH[13: 8]和 IAPDATL[7: 0]组成 14 位数据。

OTP 写入数据操作步骤:

- (1) 写入前 VIN 电压供电为 9.2V
- (2) 对 IAPTRIG 连着写入 0x18,0x01,0Xc0 入到 IAP mode。注意: 如果往 IAPTRIG 写入其他任何值, 则进入 IAPLOCK 状态。进行重新上电来解除锁定, 解除后需要重新开始进行 IAP 写操作。
- (3) 设置 IAPADDRH (高位地址) 和 IAPADDRL (低位地址), 设置数据要写入 OTP 对应 0000H-7FFFH 地址内的某个地址。
- (4) 设置 IAPDATH (高位数据) 和 IAPDATL (低位数据), 即把需要写入数据寄存器的数据暂放在 IAPDAT 寄存器中。
- (5) CPU 将 IAPDAT[13: 0]数据写入 IAPADDRH 和 IAPADDRL 对应的 OTP 地址中。
- (6) 如果要继续写, 则回到步骤 (2) 开始。如果要退出写入操作, 则到步骤 (7)。
- (7) 对 IAPTRIG 写入 0, 便可以进行写操作。

18.1 OTP 的相关寄存器

IAPTRIG

0xf2f	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPTRIG	IAPTRIG[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

IAPCTRL

0xf2e	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPCTRL	-	-	-				LOCK	PG

R/W	-	-	-				R/W	R/W
初始值	-	-	-				0	0

BIT[1] LOCK – IAP 锁定状态位

1: 锁定

0: 未锁定

BIT[0] PG – 写入

1: 写入 (需要 100us)

0: 未写入

IAPADDRH

0xf2d	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPADDRH	-	-				IAPADDR10	IAPADDR9	IAPADDR8
R/W	-	-				R/W	R/W	R/W
初始值	-	-				0	0	0

BIT[5:0] IAPADDRH[2:0] – IAP 地址高三位

IAPADDRL

0xf2c	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPADD	IAPA	IAPA	IAPA	IAPA	IAPA	IAPA	IAPA	IAPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] IAPADDRL[7:0] – IAP 地址低八位

IAPDATH

0xf2b	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPDATH			IAPDAT[13: 8]					
R/W			R/W	R/W	R/W	R/W	R/W	R/W
初始值			0	0	0	0	0	0

BIT[7:0] IAPDATH[7: 0] – IAPDAT[13:8]数据高六位

IAPDATL

0xf2a	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPDATL	IAPDAT[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] IAPDATL[7: 0] – IAPDAT[7: 0]数据低八位

IAPWAIT

0xf29	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPWAIT	IAPWAIT[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **IAPWAIT[7: 0]** – IAP 的写数据延时位

19 电气特性

19.1 绝对极限参数 over operating free-air temperature range.

参数	描述	最小值	最大值	单位
VCCIN	VCCIN 管脚输入电压	-0.3	6.0	V
VCC	REG20 LDO Supply Voltage		2.0	V
VBAT	BAT to VSS	-0.3	5.0	V
VIOD	Open-drain I/O to VSS(SDA,SCL,OWI)	-0.3	6.0	V
VIGP	其他管脚(GPIO)到地	-0.3	6.0	V
TA	工作温度	-40	85	°C
TSTG	储存温度	-50	125	°C

19.2 电源特性

TA = -40°C to 85°C; typical values at TA = 25°C and VCCIN = VBAT = 3.7 V.

参数	描述	测试条件	最小值	典型值	最大值	单位
VPOR	POR 开启电压(VCCIN)	TA = 25°C, ICCIN ≤ ICC	2.05	2.15	2.2	V
VHYS	Power-On Reset 迟滞电压		115			mV
ICC	正常工作模式	IC 处于 NORMAL MODE 状态	84	85	87	μA
IIDE	一级休眠模式	IC 处于 IDLE MODE 状态	10	12	14	μA
ISLP	二级休眠模式	IC 处于 SLEEP MODE 状态	1.9	2	2.5	μA
ISHP	运输模式	IC 处于 SHIPPING MODE 状态			<1	μA

19.3 直流特性

参数	描述	测试条件		最小值	典型值	最大值	单位
VDD	工作电压	—	fSYS=16MHz	2.6	—	5.5	V
	工作电压	—	fSYS =1MHz	2.6	—	5.5	V
IDD	工作电流	5.0V	fSYS =16MHz	—	1.45	1.7	mA
	工作电流	5.0V	fSYS =1MHz	—	0.40	0.42	mA
VIL	I/O 口低电平输入电压	5.0	—	—	0.45	1.3	V
VOL	I/O 口低电平输出电压	5.0	—	—	0.003	—	V
VOH	I/O 口高电平输出电压	5.0	—	—	5.0	—	V
IOL	I/O 口灌电流	5.0	—	19.5	20.7	21.1	mA
IOL	I/O 口拉电流	5.0	—	9.19	9.22	9.22	mA
RPH	I/O 口上拉电阻	5.0	PULL RES IS 30K	28.3	30	30	KΩ
RDH	I/O 口下拉电阻	5.0	LOW RES IS 30K	31.8	32.9	34	KΩ

RPH	I/O 口上拉电阻	5.0	PULL RES IS 190K	190	190	195	KΩ
RDH	I/O 口下拉电阻	5.0	LOW RES IS 300K	338	340	342	KΩ

19.4 交流特性

参数	描述	测试条件		最小值	典型值	最大值	单位
fSYS	系统时钟(OSCF)	5.0V	OTP_OSCM_16M、 FOSC_DIV1	—	16	—	MHz
	系统时钟(OSCF)	5.0V	OTP_OSCM_1M、FOSC_DIV1	—	1	—	
tRSTD	WDT 复位时间	5.0V	WDTEN_ON、TWDT_111	—	32	—	ms
			WDTEN_ON、TWDT_110	—	3.8	—	
			WDTEN_ON、TWDT_101	—	256	—	
			WDTEN_ON、TWDT_100	—	64	—	

19.5 ADC 电气特性

参数	描述	测试条件		最小值	典型值	最大值	单位
工作电压	---	5.0V	---	2.6	5.0	5.5	V
转换电流	---		---	1.25	1.3	1.46	mA
输入电压	---		---	0	---	VDD	V
INL	Integral Non-linearity		AVCC=5.0V	---	-4/+2	---	LSB
DNL	Differential Non-linearity		AVCC=5.0V	---	±1	---	LSB

19.6 CCADC(Instant-Current&Coulomb-Counter)电气特性

TA = -40°C to 85°C; typical values at TA =25°C and VCCIN = VBAT = 3.7V.

参数	描述	测试条件	最小值	典型值	最大值	单位
VSR	VSRPI to VSRNI 差分输入电压范围	VSR = VSRPI - VSRNI	-0.2		0.2	V
tCV(CC)	Coulomb-Counter 转换时间	单次转换	0.128		1	S
RCC	Coulomb-Counter 分辨率			18		Bits
tCV(IC)	Instant-Current 转换时间	单次转换		3.9		mS
RIC	Instant-Current 分辨率			16		Bits
VOS(SR)	输入 Offset		-5	10	5	μV
RSR	检流电阻 R-Sense		5		20	mΩ
ILK(SR)	输入漏电流				0.1	μA

19.7 LVR 电气特性

参数	描述	测试条件		最小值	典型值	最大值	单位
IQ	LVD Quiescent Current	5.0V	---		10		uA
VLVR	Low Voltage Reset Voltage/ [迟滞 50mV 左右]	LVT20V	LVR 使能	-5%	2.0	+5%	V
		LVT21V			2.1		
		LVT22V			2.2		
		LVT24V			2.4		
		LVT26V			2.6		
		LVT27V			2.7		
		LVT29V			2.9		
		LVT30V			3.0		
		LVT31V			3.1		
		LVT33V			3.3		
		LVT36V			3.6		
		LVT37V			3.7		
		LVT38V			3.8		
		LVT41V			4.1		
LVT42V	4.2						
LVT43V	4.3						

19.8 POR 电气特性

参数	描述	测试条件		最小值	典型值	最大值	单位
tpor	POR Trigger Voltage	---	-40~85°C	---	1.5	---	V

19.9 BOR 电气特性

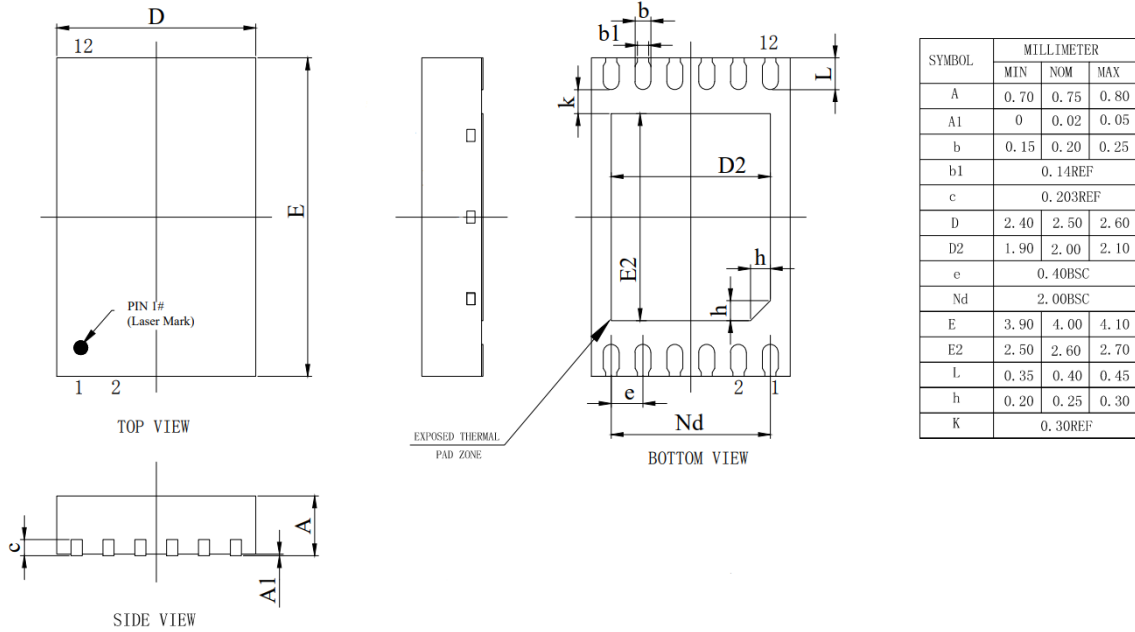
参数	描述	测试条件		最小值	典型值	最大值	单位
VBOR	BOR Trigger Voltage	---	-40~85°C	---	1.7	---	V
IQ	Power Supply Current			---	0.5	1	uA

19.10 FVR 电气特性

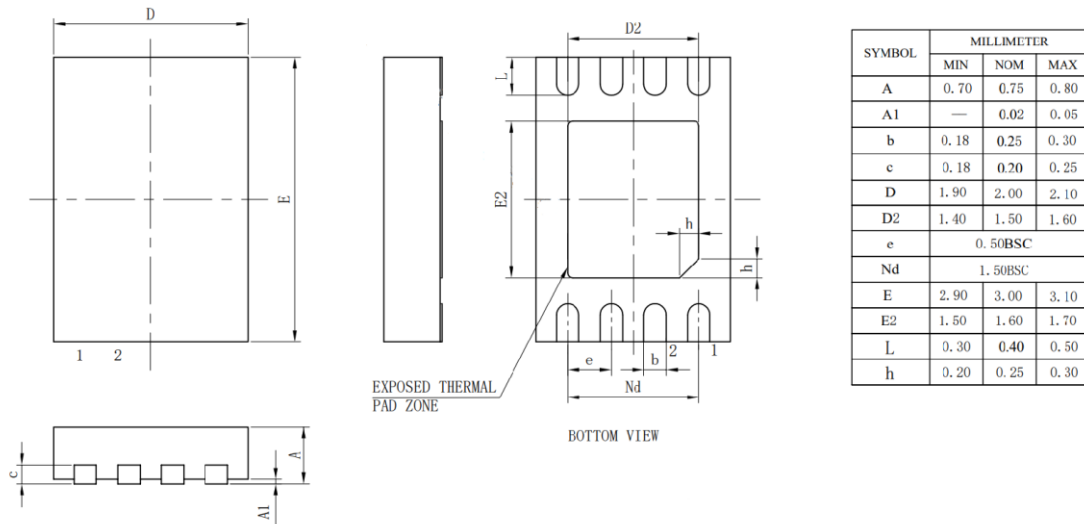
参数	描述	测试条件		最小值	典型值	最大值	单位
FVR	---	3.3-5.0	Vref=1V	0.993	1.0	1.006	V
		3.3-5.0	Vref=1*2V	1.986	2.0	2.012	V

20 封装尺寸

DFN12



DFN8



21 订购信息

丝印信息

现行ADUC的单片机表面印有一栏信息：产品代码和日期码。

	Marking	B845 1 20 1 x
		Device code Year Week SeriesNo Internal Usage
		Year: 0: 2020; 1: 2021
		Week: 01:第1周; 23:第23周 SeriesNo: 序列号0-Z, 1:当前周别第2个工单

标签信息

货品内外包装上粘贴的标签上包含：产品名称，封装信息，芯片批号，丝印信息，出货日期及包装数量。

	产品名称	Part No: AD1845D12-E0XTR Package: DFN12 Lot No: RCLJL2000 Marking: B8451351 Date: 2021-10-13 QTY: 5000pcs
	封装信息	
	芯片批号	
	丝印信息	
	出货日期	
	包装数量	

采购信息

AD1845			
产品名称	封装信息	工作温度	包装方式及数量
AD1845D8-E0XTR	DFN8, 绿色封装	-40-85°C	Tape & Reel 5000/reel
AD1845D12-E0XTR	DFN12, 绿色封装	-40-85°C	Tape & Reel 5000/reel