

功能概述

- **高性能低功耗 8 位 MIC8S 内核**
 - 53 条指令，可配置 1T/2T/4T 指令周期
 - 8 级深度堆栈寄存器
- **2048x16bit 1K-Cycle FLASH 程序存储器**
- **128x8bit SRAM**
- **外设控制器**
 - 两个 8 位定时器 Timer0/2
 - 独立 8 位可编程预/后分频器(Timer2)
 - 一个 16 位定时器 Timer1
 - 内置预分频器
 - 支持外部时钟门控输入
 - 支持同步/异步时钟输入模式
 - 俘获/比较/PWM 模式(CCP)
 - 16 位外部输入俘获
 - 带死区控制 PWM 输出
 - 互补输出产生器(COG/PWM)
 - 上升/下降沿的死区控制
 - 相位控制，消隐控制以及硬件自动关闭
 - 12 位模数转换器(ADC)
 - IVREF/AVREF/VCC 可选参考
 - 内置 VCC/5 电源监控通道
 - 多路增益可编程(x1/8/16/32)的差分输入
 - 差分放大器支持轨对轨输入
 - 差分放大器出厂失调校准
 - 内部可校准 1.2V±1%参考电压源
 - 2x 模拟比较器
 - 最大 10mV 比较误差
 - 可选内部/外部比较器输入
 - 2x 6bit 或 1x 10bit DAC, 支持 DAC 外部输出
 - 可编程看门狗定时器 (WDT)，独立 32K 内部 RC 振荡器
- **特殊处理器功能**
 - 外部中断源与 I/O 电平变化中断
 - 2 路 80mA 大电流推挽驱动，用于驱动高速 PWM
 - 4 路 80mA 大电流扇入 LED 驱动
 - 内置上电复位电路 (POR) 与可编程低电压检测电路 (LVR)
 - 内部上电复位定时器(PWT)与晶振启动定时器(OST)
 - 内置±1%可校准 1MHz~32MHz RC 振荡器(粗调/细调)
 - 支持外部低速/高速晶振输入(最高到 20MHz)，支持晶振失效保护
- **封装: SOP16/14/8**
- **工作环境**
 - 工作电压: 2.0V ~ 5.5V
 - 工作温度: -40C~+85C
 - 待机功耗: 1uA@3.3V



8-bit MIC8S

Microcontroller with
2048x16 FLASH
Program Memory

LGT8F684A

Overview

Version 1.0.8

应用领域

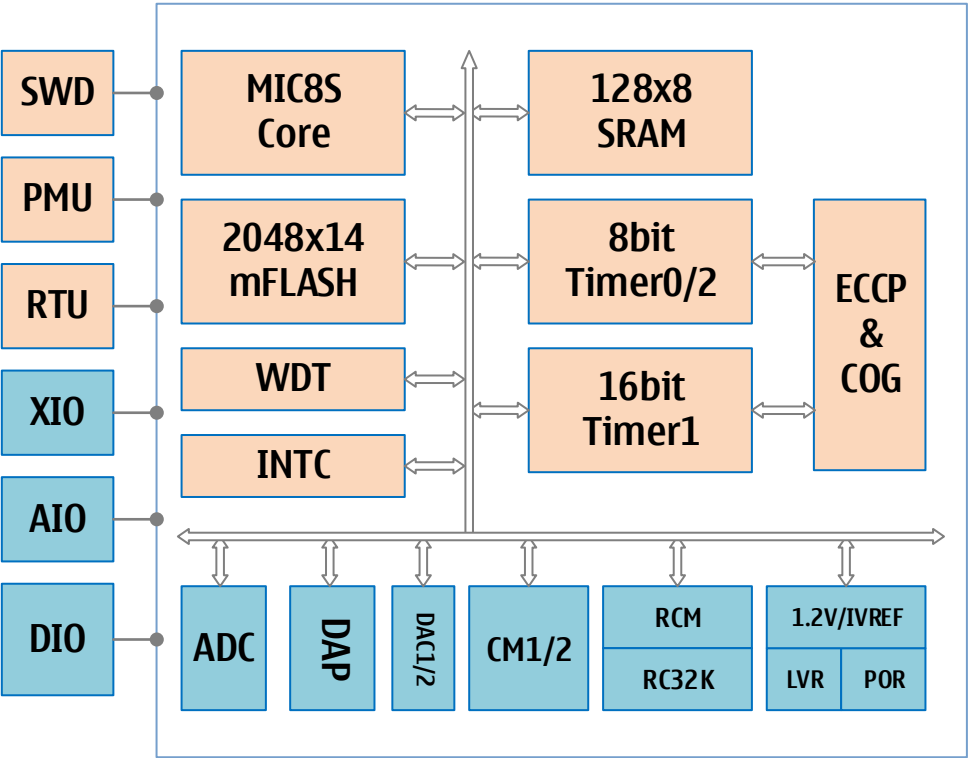
小家电

马达驱动

电池管理

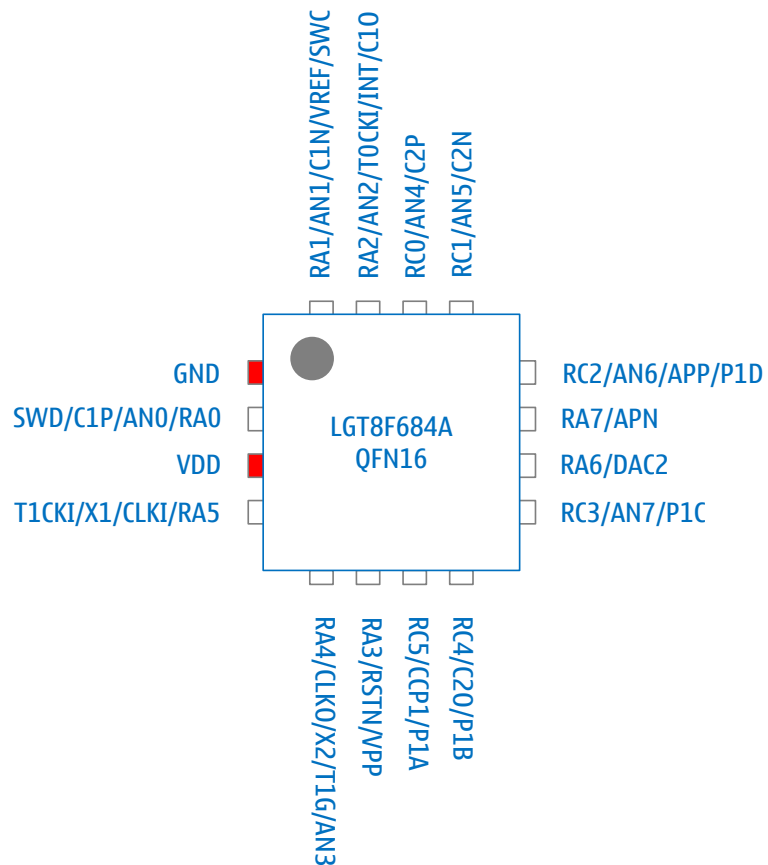
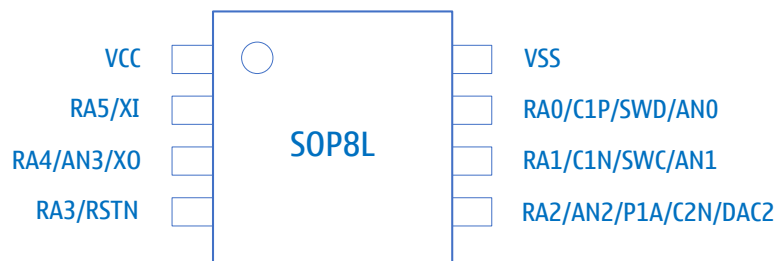
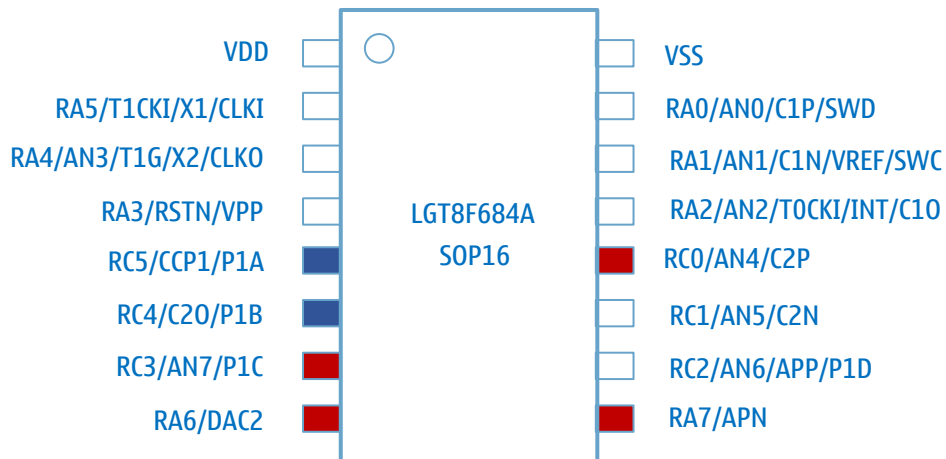
照明控制

系统框架



模块名称	模块功能
MIC8S	MIC8S 8 位微处理器内核
SWD	SWD 量产/测试接口
PMU	功耗管理单元
RTU	复位管理单元
XIO	晶振 I/O
AIO	模拟 I/O
DIO	数字 I/O
CM1/2	模拟比较器 1/2
ADC	12 位多通道模数转换器(ADC)
DAP	增益可编程差分放大器
Timer0/2	8 位定时器/计数器 0/2
Timer1	16 位定时/计数器 1
CCP	比较/俘获/PWM 产生单元
DAC1/2	6 位 DAC1/2 或者 10 位 DAC
RCM	1/8/16MHz 可校准 RC 振荡器，支持 2/16/32MHz 倍频输出模式
RC32K	32KHz 低功耗 RC 振荡器
1.2V/IVREF	1.2V±1% 可校准内部参考电压
INTC	中断控制器

封装定义



说明：SOP8L 基于 SOP16 封装的引脚功能，保留 SOP16 上半部分的引脚功能，并将 RC 组其中两个引脚（RC5，RC1）绑定到 RA2 上。在进行 RA2 端口配置时，请考虑到 RC5/RC1 端口的作用。

以下表格为 SOP8 与 SOP16 之间的 I/O 对应关系：

SOP8L	SOP16L
RA5	RA5
RA4	RA4
RA3	RA3
RA2	RA2+RC5+RC1
RA1	RA1
RA0	RA0

引脚说明

Pin.	GPIO	ADC	DAC	AC	DPA	Timer	CCP/PWM	IRQ	Power	MISC
1									VDD	
2	RA5					T1CKI		IOC		X1/CLKI
3	RA4	AN3				T1G		IOC		X2/CLKO
4	RA3							IOC	VPP	
5	RC5						CCP1/P1A			HD(*)
6	RC4			C20			P1B			HD(*)
7	RC3	AN7					P1C			HSD(*)
8	RA6		DAC2					IOC		HSD(*)
9	RA7				APN			IOC		HSD(*)
10	RC2	AN6			APP		P1D			
11	RC1	AN5		C2N						
12	RC0	AN4		C2P						HSD(*)
13	RA2	AN2	DAC1	C10		T0CKI		IOC/INT		
14	RA1	AN1/VREF		C1N				IOC		SWC
15	RA0	AN0		C1P				IOC		SWD
16									VSS	

说明：

HD: 80mA 推挽驱动

HSD: 80mA 高扇入驱动

重要说明

LGT8F684A 的编程下载需要复用芯片的部分引脚。当这些引脚被应用程序或者系统配置位改变了默认功能后，将会有可能**导致芯片锁死**。因此，建议这电路设计和程序开发期间注意和编程相关引脚的使用。

编程下载所复用的引脚为：

引脚	ISP 编程功能说明
RA3/VPP/RSTN	默认为 RSTN，即外部复位输入。编程器会首先拉低此引脚让芯片进入复位状态，然后通过 SWD 接口复位内核。最后释放 RSTN。在编程时，此引脚为编程提供 9.8V 左右的编程电压。
RA0/SWD	SWD 数据线，复位为输入默认，编程器外部上拉
RA1/SWC	SWC 时钟线，复位为输入默认，编程器外部上拉

在以下两种情况下，将会导致芯片锁死，锁死后将无法重复烧写：

1. 当软件通过寄存器 PCON[SWDD]位禁用 SWD 接口后，将导致芯片锁死；
2. 当 SWD 或 SWC 被设置为输出，并同时通过配置位将 RA3/RSTN 配置为 RA3 模式，将导致芯片锁死；

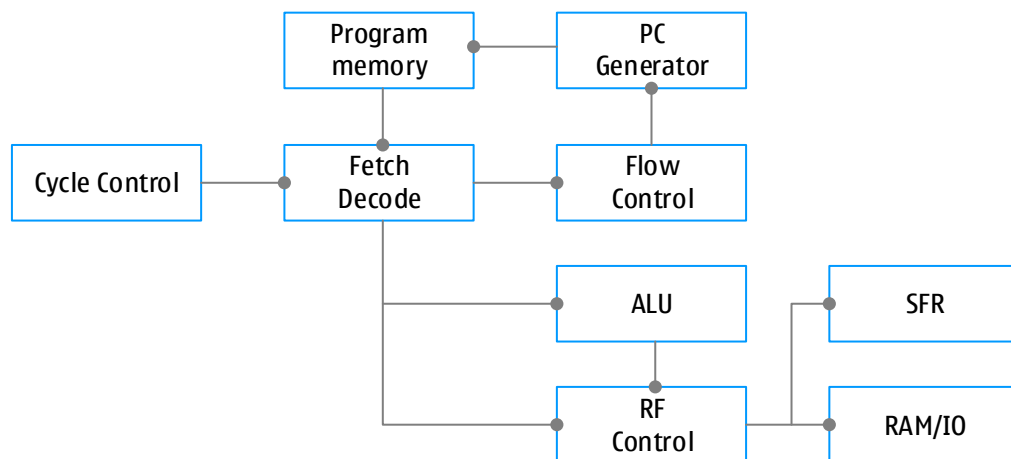
在实际应用中针对避免芯片锁死的使用建议：

1. 在没有非常必要的需求下，建议不要通过 SWDD 为禁用 SWD 接口；
2. 芯片的 RA3/RSTN/VPP 引脚在芯片编程时将提供 9.8V 左右的高压，因此需要在电路设计上设计编程接口时，需要注意编程高压不会对电路本身产生破坏性影响；
3. RA3/RSTN 引脚作为 I/O 功能 RA3 时，只能作为输入或开漏输出模式；因此建议在设计电路时将 RA3 分配为简单，独立的控制功能。等调试工作完成后，再将此 I/O 配置为所需数字功能；
4. 当 RA3/RSTN 被设置为 RA3 模式，并且软件将 RA0/SWD 或 RA1/SWD 设置为输出模式，在这种情况下，可以使用芯片其他 I/O 辅助设置的方式，将 RA0/1 设置为输入，实现解锁芯片，重复编程。
5. 如果应用需要将 RA3 设置位 RA3 I/O 模式，并且应用需要将 RA0 或者 RA1 设置位输出 I/O，在这种情况下，建议在设置 RA0/1 为输出之前，通过内部弱下拉电阻配合检查外部是否连接下载器，连接下载器后，RA0/1 将会被强上拉，因此可以通过判断是否连接了下载器。在确认没有连接下载器的情况下，可安全的将 RA0/1 设置为输出；
6. 当需要使用 RA0/1 的模拟功能时，如果是将 RA0/1 用于 ADC 输入或者 ADC 的外部参考输入，无需设置 ANS0/1，此时只有保证 RA0/1 为输入模式且关闭上下拉电阻即可。置位 ANS0/1 同样会禁止 RA0/1 的下载接口功能，因此建议在将 ANS0/1 置位前，参考第 5 点建议的方法。

MIC8S 内核

- 1T/2T/4T 可配置指令周期
- 42 条基本指令+11 条扩展指令
- 可配置分页/连续地址映射模式
- 零开销 LOOP 指令
- 支持程序空间的查表指令
- 直接以及间接寻址模式
- 自动递增/递减的间接数据寻址
- 基地址+4 位偏移量的间接数据寻址
- 支持软件/硬件中断源

综述



MIC8S 指令控制/执行流程图

MIC8S 采用哈佛总线构架，采用分离的指令和数据访问总线控制。与 RISC 构架具有丰富的通用工作寄存器不同，MIC8S 构架仅有一个工作寄存器(W)；但 MIC8S 的大部分指令可以直接访问低端 64 字节的空间，这部分空间包含了全部特殊功能寄存器以及部分 RAM 空间，因此 MIC8S 仍然可以高效的完成所有的运算以及控制指令。

除工作寄存器(W)外，MIC8S 还有几个与指令相关的特殊功能寄存器，这些寄存器协助指令系统完成灵活的数据以及程序寻址，这些寄存器包括：

INDF：间接寻址数据寄存器。对 INDF 读写访问的目标地址由 FSR 寄存器决定；

FSR：间接寻址目标地址或基地址，可以用于访问数据或程序空间；

LCR：循环计数器，实现零开销循环控制；LOOP 指令专用；

STATUS：系统状态寄存器，ALU 的执行结果状态，用于程序流程控制；

存储系统

程序存储空间

MIC8S 指令可直接寻址最大 8K 指令字空间, GOTO/CALL 指令宽度为一个指令字, 可直寻址 1K 指令字空间。LOOP 以及条件跳转指令 BRSZ/BRCZ/BRSC/BRCC 均为双指令字宽度, 可寻址 8K 指令空间。MIC8S 支持程序空间中的数据访问, 结合 PCH:FSR 寄存器, 可以实现全空间的程序数据访问以及灵活的数据查表功能。

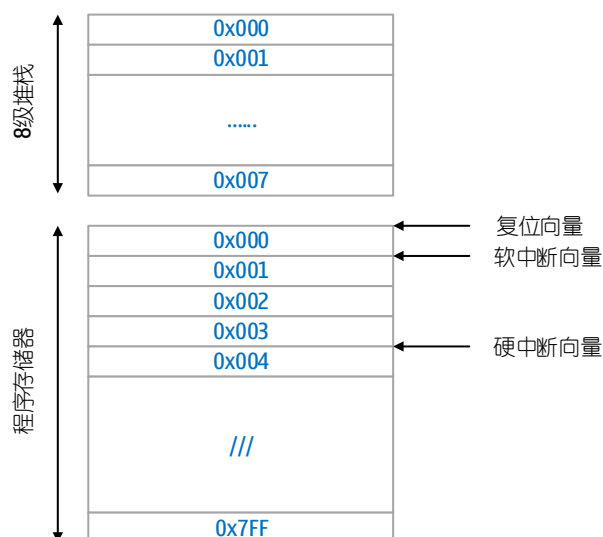
LGT8F684A 实现了 2Kx16 (0x0000 – 0x07FF) 的程序空间。超出此空间的地址访问, 将会回滚到 2K 范围内。系统上电/复位后, 内核从 0x0000 地址开始执行。LGT8F684A 实现了硬件/软件两种中断请求, 硬件中断的向量地址为 0x0004, 软件中断(INT 指令)的向量地址为 0x0001。

8 级深度堆栈寄存器

LGT8F684A 实现了一个 8 级深度的硬件堆栈单元。堆栈独立于程序以及数据空间。堆栈只能通过堆栈指针访问; 堆栈指针本身并不能通过指令直接访问。当系统执行 CALL 指令或者发生了有效的中断, 当前 PC 被压入堆栈。当系统执行了 RETURN/RETLW/RETFIE 指令后, 堆栈弹出最后压入的值(到 PC 指针)。在压栈和出栈操作过程中, PCHBUF 并不会受到影响。

堆栈的深度为 8 级, 当连续执行 8 次压栈操作后, 后续的压栈操作将会破坏堆栈中的数据, 最早压栈的数据将会从堆栈中弹出; 同样如果多次执行出栈的操作, 堆栈弹出最早的数据后, 将会循序返回之前的数据。这两种情况均会导致系统执行的不确定, 因此用户需要在使用时特别注意。

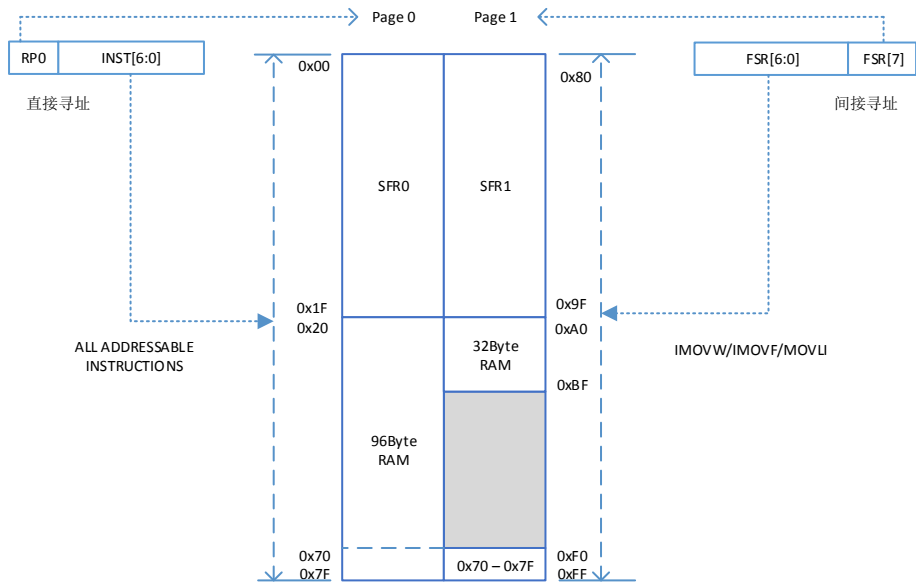
LGT8F684A 的程序/堆栈空间分布如下:



数据存储空间

MIC8S 使用分页模式管理内部数据空间。在分页地址映射模式下, MIC8S 最大支持 2 页的分页空间; 每个页面的大小为 128 字节。每页的低 64 字节被映射到寄存器空间, 高 64 字节被映射到 RAM 空间。系统实现可以自由将 IO 空间映射到寄存器页中, 也可以将没有实现的地址自动与第 0 页的空间重叠。在分页模式下, MIC8S 可以支持最大 128 字节的寄存器空间和 128 字节的 RAM 空间。

LGT8F684A 数据存储空间地址映射图:



MIC8S 实现了直接/间接两种寻址方式。在直接访问模式下，通过 STATUS 寄存器的 RP0 位选择访问的页面，页面内的偏移地址直接由指令给出。间接寻址模式使用 FSR 寄存器作为间接寻址的地址寄存器，通过读写 INDF 寄存器，实现间接数据访问。直接/间接模式都可以访问到所有的寄存器以及 RAM 空间。需要特别指出对 RAM 空间读操作访问，需要两个指令周期才能完成。另外，MIC8S 提供了 IMOVW/IMOVF，以及 MOVL 等专用指令，用于实现高效的间接访问操作。

LGT8F684A 实现了 128 字节的内部 RAM 空间，其中 96 字节映射到第一页的 0x20-0x7F 地址范围；另外 32 字节映射到第二页的 0xA0 - 0xBF 地址区间。对第二页地址空间 0xF0-0xFF 的访问，被映射到第一页的 0x70-0x7F 区间。因此，这里的 16 个字节在第一个和第二页都被映射到相同的物理单元，可用于实现快速的变量访问或者共享。

在直接寻址模式下，通过 STATUS 寄存器中的 RP0 位选择当前的页面，当前页面下的所有地址都可以通过指令直接寻址到，地址的低 7 位来自指令代码的最低 7 位。具体请参考文档最后关于指令集定义部分。

特殊功能寄存器 - 第 0 页

特殊功能寄存器: PAGE 0									
名称	地址	位定义							
INDF	0x00	FSR 间接寻址数据							
TMR0	0x01	Timer0 计数寄存器							
PCL	0x02	PC[7:0]							
STATUS	0x03	WPKF	RP1	RP0	T0	PD	Z	DC	C
FSR	0x04	FSR[7:0] 间接寻址地址寄存器							
PORTA	0x05	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
DC1LR	0x06	16 位 Timer1 占空比寄存器低 8 位							
PORTC	0x07	-	-	RC5	RC4	RC3	RC2	RC1	RC0
PR1LR	0x08	16 位 Timer1 定时周期寄存器低 8 位							
PR1HR	0x09	16 位 Timer1 定时器定时周期高 8 位							

PCHBUF	0x0A	-	-	-	-	PCH[11:8]			
INTCON	0x0B	GIE	PEIE	TOIE	INTIE	RAIE	TOIF	INTF	RAIF
PIR1	0x0C	CGIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF
DC1HR	0x0D	16 位 Timer1 占空比寄存器高 8 位							
TMR1L	0x0E	16 位 Timer1 计数器低 8 位							
TMR1H	0x0F	16 位 Timer1 计数器高 8 位							
T1CON	0x10	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
TMR2	0x11	Timer2 定时/计数寄存器							
T2CON	0x12	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
CCPR1L	0x13	俘获/比较/PWM 寄存器低 8 位							
CCPR1H	0x14	俘获/比较/PWM 寄存器高 8 位							
CCP1CON	0x15	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
PWM1CON	0x16	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
ECCPAS	0x17	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
WDTCN	0x18	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
CMCOM0	0x19	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
CMCON1	0x1A	C1DEM	C2OE	C1OE	T2ACS	C2HYS	C1HYS	T1GSS	C2SYNC
CMCON2	0x1B	CCM	C2EN	C1EN	FLTEN	C2P1	C2N1	C1P1	C1N1
APCON	0x1C	APEN	-	DPS1	DPS0	DNS1	DNS0	GA1	GA0
TCCR	0x1D	X2EN	-	T2CF	T2CS	T1CF	T1CS	-	-
ADRESH	0x1E	ADC 转换结果高位							
ADCON0	0x1F	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/DONE	ADON

特殊功能寄存器 - 第 1 页

特殊功能寄存器: PAGE 1									
名称	地址	位定义							
INDF	0x80	间接寻址数据寄存器							
OPTION_REG	0x81	RAPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
PCL	0x82	程序地址指针低 8 位							
STATUS	0x83	WKPF	RP1	RP0	T0	PD	Z	DC	C
FSR	0x84	FSR[7:0] 间接寻址地址寄存器							
TRISA	0x85	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
-	0x86								
TRISC	0x87	-	-	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
COGCR0	0x88								
COGCR1	0x89								
PCLATH	0x8A	-	-	-	-	PCHBUF[11:8]			
INTCON	0x8B	GIE	PEIE	TOIE	INTE	RAIE	TOIF	INTF	RAIF
PIE1	0x8C	CGIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE
COGPHR	0x8D								
PCON	0x8E	PPLP	SWDD	TMR0D	LVRE	DPSM1	DPSM0	PORF	BORF
OSCCON	0x8F	TSSM	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS

OSCTUNE	0x90	RCM 校准位							
ANSEL	0x91	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
PR2	0x92	定时器 2 定时周期寄存器							
WPDC	0x93	-	-	WPDC5	WPDC4	WPDC3	WPDC2	WPDC1	WPDC0
WPUC	0x94	-	-	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
WPUA	0x95	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
IOCA	0x96	IOCA7	IOCA6	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0
WPDA	0x97	WPDA7	WPDA6	WPDA5	WPDA4	WPDA3	WPDA2	WPDA1	WPDA0
ANSEL1	0x98	ANTM	-	-	-	-	-	ANS9	ANS8
VRCON1	0x99	DAC1EN	BGEN	VR15	VR14	VR13	VR12	VR11	VR10
COGAS	0x9A								
COGDBR	0x9B								
VRCON2	0x9C	DAC2EN	D10BEN	VR25	VR24	VR23	VR22	VR21	VR20
DACON	0x9D	CAPTM	-	-	-	DAC2S1	DAC2S0	DAC1S1	DAC1S0
ADRSEL	0x9E	ADC 转换结果高字节							
ADCON1	0x9F	CHS3	ADCS2	ADCS1	ADCS0	DIFS	ADSPD	VDS1	VDS0

STATUS – 状态寄存器

STATUS 寄存器包含了 ALU 的算术状态，复位标志以及分页模式下的页选择控制；STATUS 寄存器可以作为任意指令的目标地址，但如果指令本身会影响到 Z,DC 或者 C 标记位，那么指令对这三位的写将会被禁止。这些位的更新仅依赖于硬件逻辑。另外， \overline{TO}/PD 这两位为只读位，因此对指令直接对 STATUS 的操作的结果可能会与预期的结果有所不同。

例如，CLRf STATUS，这条指令将清除 STATUS 的高三位，设置 Z 位。指令执行后，STATUS 寄存器的值将为：000u u1uu (其中 u 是保持不变)。鉴于 STATUS 寄存器的特殊性，建议仅仅使用 BCF/BSF/SWAPF/MOVBWf 指令更改 STATUS 寄存器，因为这些指令本身不会对 STATUS 产生附加效果。关于指令对 STATUS 状态位的影响，请参考本手册“指令集速查表”部分。

STATUS – 状态寄存器								
地址: 0x03		默认值: 0001_1000						
Bit	7	6	5	4	3	2	1	0
STATUS	WKPF	GPA1	GPA0	\overline{TO}	\overline{PD}	Z	DC	C
R/W	R	R/W	R/W	R	R	R/W	R	R
Bit	Name	描述						
7	WKPF	唤醒标志位 1: 外部复位唤醒 0: 其他唤醒						
6:5	GPA1/0	通用寄存器位; 在分页地址映射模式下 GPA[0]作为页选择位(RP0) 1: 选择第 1 页 (0x80~0xFF) 0: 选择第 0 页 (0x00~0x7F)						
4	\overline{TO}	定时器溢出标志位 0: 看门狗溢出 1: 上电复位或者执行 CLRWDt/SLEEP 指令						

3	PD	待机标志位 0: SLEEP 指令 1: 上电复位或者执行 CLRWDT 指令
2	Z	零标志 0: 算术运算结果为 0 1: 算术运算结果非 0
1	DC	BCD 半进位/借位标志
0	C	进位/借位标志 0: 加法非进位, 减法借位 1: 加法进位, 减法非借位
备注: STATUS[4:3]复位值仅上电复位和 LVR 复位有效		

OPTION_REG – 外设配置寄存器

OPTION_REG 寄存器包含 Timer0/WDT 预分频控制, 外部中断 INT/RA2 触发沿控制, Timer0 相关控制以及全局上拉控制。

当系统进入深睡眠模式后, 只能通过外部中断引脚 RA2/INT 或者外部复位唤醒; 外部中断唤醒的电平也需要通过 INTEDG 位控制。

Timer0 与 WDT 共享一个预分频模块, 同一时间预分频模块只能分配给其中一个使用。当预分频分配给 WDT 后, Timer0 计数器为 1:1 的预分频模式。

MICS8S 内核实现了一个专用更新 OPTION_REG 寄存器的 OPTION 指令; OPTION 指令将工作寄存器 W 中的数据更新到 OPTIION 寄存器中。同时 OPTION_REG 寄存器被映射到第 1 页 0x81 地址, 因此除了可以通过 OPTION 指令访问, 也可以通过其他指令寻址。

OPTION_REG – 外设配置寄存器								
地址: 0x81					1111_1111			
Bit	7	6	5	4	3	2	1	0
OPTION	RAPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	RAPU	PORTA 全局上拉禁止控制 1 = PORTA 上拉被全局禁止						
6	INTEDG	外部中断/唤醒边沿选择 1: INT/RA2 的上升沿产生中断 0: INT/RA2 的下降沿产生中断						
5	TOCS	Timer0 时钟源选择 1: 选择 TOCKI 外部输入, Timer0 为计数器模式 0: 选择内部系统时钟, Timer0 为定时器模式						
4	TOSE	Timer0 时钟触发沿选择 1: TOCKI 的下降沿 0: TOCKI 的上升沿						
3	PSA	预分频器分配控制位 1: 预分频为 WDT 所有						

		0: 预分频为 Timer0 所有		
		预分频分频选择位		
		PS[2:0]	Timer0 Rate	WDT Rate
		000	1:2	1:1
		001	1:4	1:2
		010	1:8	1:4
		011	1:16	1:8
		100	1:32	1:16
		101	1:64	1:32
		110	1:128	1:64
		111	1:256	1:128

INDF – 间接寻址数据寄存器

INDF 寄存器并没有物理上对应的寄存器实现。INDF 寄存器用于间接寻址操作。

任何使用 INDF 作为目标寄存器的指令，间接访问由 FSR 寄存器指向的目标地址。间接读 INDF 寄存器本身将返回 0。通过 FSR 寄存器间接写 INDF 本身也是没有意义的。由于 LGT8F684A 仅仅实现了 0x00~0xFF 的地址空间，因此通过 FSR 寄存器可以寻址到整个内部数据空间。

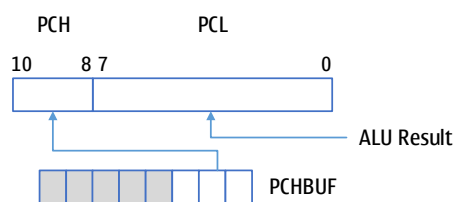
下面我们通过一个实例程序说明如何利用间接寻址进行高效的数据访问；实例程序使用间接寻址方式，清零 0x40 到 0x7F 之间的 RAM 区域：

	MOVLW	0x40	; initialize pointer
	MOVWF	FSR	; to RAM
NEXT	CLRF	INDF	; clear INDF register
	INCF	FSR	; inc pointer
	BTFSS	FSR, 7	; all done?
	GOTO	NEXT	; no, go on next
DONE			; yes

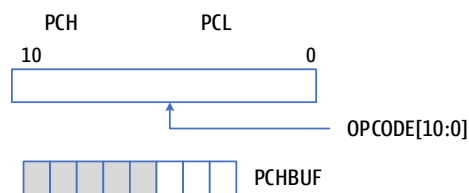
INDF – 间接寻址数据寄存器								
地址: 0x00					默认值: XXXX_XXXX			
Bit	7	6	5	4	3	2	1	0
INDF	INDF[7:0]							
R/W	R/W							
Initial	8'hXX							
Bit	Name	描述						
7:0	INDF	间接寻址数据寄存器，物理上并没有寄存器实现。读 INDF 将返回 FSR 寻址地址的数据。如果 FSR 指向 INDF 本身，将返回 0x00； 当使用 IMOVW/F 指令间接寻址时，不需要使用到 INDF 寄存器；						

PCL – PC 低字节寄存器

LGT8F684A 实现了 2K 指令字的程序空间，程序计数器(PC)为 11 位宽。PC 的低字节来自 PCL 寄存器。PCL 寄存器可读/写，PC 的高 3 位来自 PCHBUF。系统复位后，PC 的值根据系统模式分别指向不同的位置。下图说明了系统如何在指令的影响下更新 PC 值：

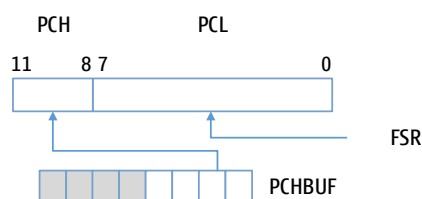


将PCL作为目标寄存器的指令

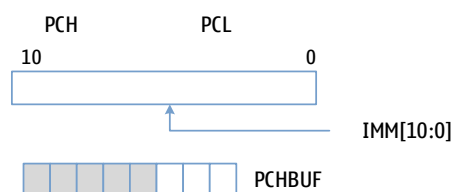


GOTO/CALL

MIC8S 同时也实现了几条与程序执行流程相关的扩展指令。此类扩展指令可以给直接访问程序空间实现查表以及直接程序流程控制带来方便。相关扩展指令对 PC 的影响如下图：



MOVWP



BRSZ/BRCZ/BRSC/BRCC/LOOP

需要注意的是，BRSZ/BRCZ/BRSC/BRCC/LOOP 这些指令为双指令字长度(28b)，除了这里列出的 5 条外，另外 MIC8S 还实现了 MOVL/MOVLT/MOVLI 这三条双指令字指令，MOVL 用于初始化 LOOP 指令的循环计数，MOVLT 指令用于设置查表指针，MOVLI 用于直接设置间接寻址寄存器 FSR。关于这类指令的详细定义，请参考本手册指令集速查表部分。

PCL – PC 低字节寄存器								
地址: 0x02					默认值: (参考运行模式介绍部分)			
Bit	7	6	5	4	3	2	1	0
PCL	PCL[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	PCL	PCL 实时反映当前运行的 PC 低字节的值； 程序可以通过直接修改这个寄存器更改当前指令的执行流程，更新 PCL 后，当前 PC 的值被更新为[PCH:PCL]						
备注：更新 PCL 前，请确认 PCH 的值为所需值。更新 PCH 不会改变 PC 的值。								

FSR – 间接寻址地址/基地址寄存器

FSR 寄存器用于辅助实现间接寻址操作。使用基本数据传输指令实现间接寻址，需要配合 INDF 寄存器完成数据的读/写访问。这部分细节请参考本章节中对 INDF 寄存器描述部分。使用扩展数据传输指令，可以直接与 FSR 配合使用，实现以下功能：

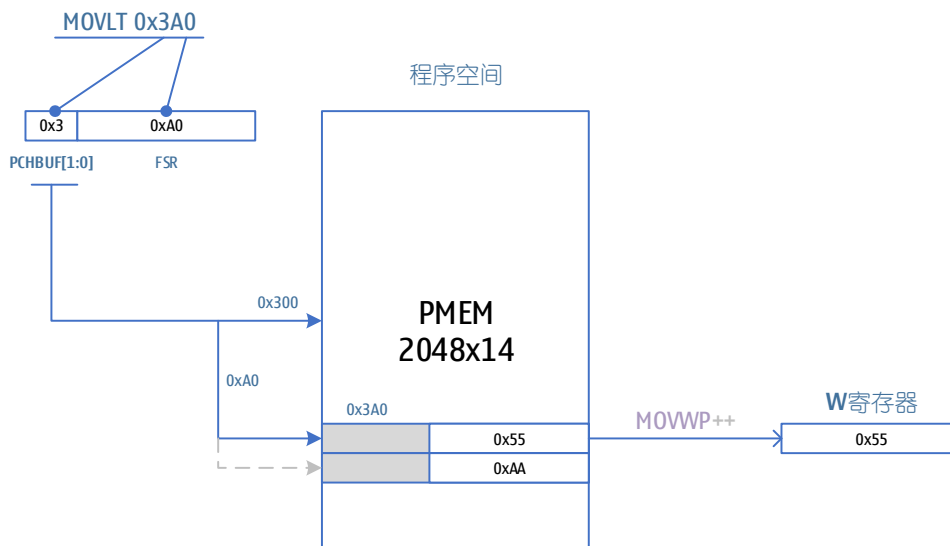
- 间接寻址数据/程序空间; (MOVWP/IMOVW/IMOVF)
- 自动 FSR 递增/递减的数据/程序空间访问; (MOVWP++/--, IMOVW/F++/--)
- 带 4 位偏移量的数据空间访问; (IMOVW/F +q/-q)

为方便此类间接寻址操作，MIC8S 实现了 MOVLT/MOVLI 指令，专用于初始化间接寻址相关的寄存器。

下面分别介绍以上几种扩展寻址的实现方式：

程序空间访问：

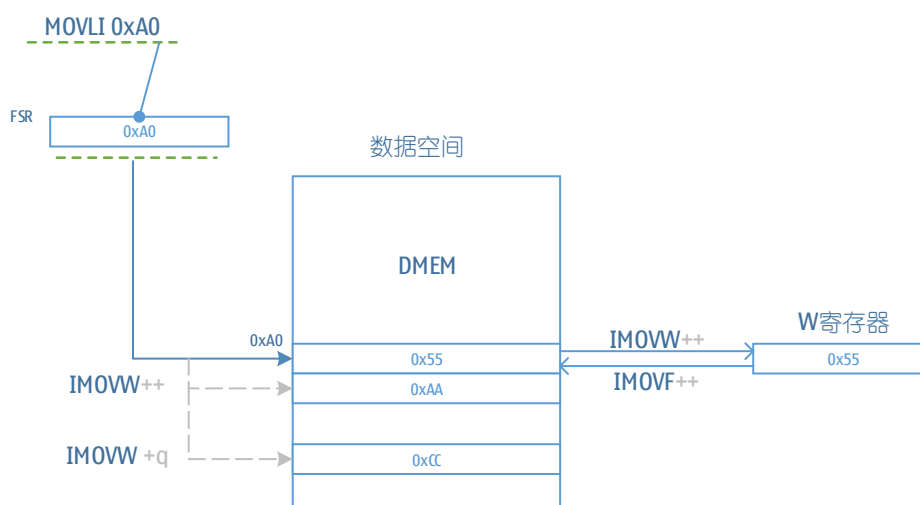
MIC85 基本指令集中没有直接访问程序空间的指令。使用基本指令集访问程序空间需要使用 RETLW 指令配合实现。扩展指令集中为此增加了专用于程序空间访问的 MOVWP 指令。MOVWP 指令访问的目标地址由 PCHBUF[3:0]以及 FSR 共同决定，可以实现对 LGT8F684A 整个 2K 指令字程序空间的访问。下图简单说明了 MOVWP 寻址程序空间的过程以及目标地址的构成方法：



如图所示，首先使用 MOVLT 指令直接给出程序空间的目标地址，MOVLt 的执行结果将使用指令所提供目标地址设置到 PCHBUF 以及 FSR 寄存器。然后执行 MOVWP 指令将{PCHBUF:FSR}指向的目标地址的数据的低字节读取出到 W 工作寄存器。如果使用 MOVWP++/--指令，执行执行完成后，自动将 FSR 的值递增/递减，为下个字节地址的访问完成准备工作。

扩展的间接数据空间寻址：

与访问程序空间原理相同，MIC85 实现了 IMOVW/F 指令专用于访问数据空间。基本指令集中实现的大部分指令都可以直接访问到所有数据以及寄存器空间。MIC85 实现的扩展访问方法作为一种辅助选择，在某些连续地址空间访问的场合，能提供更加有效的访问操作。



由于 LGT8F684A 只实现了 0x00~0xFF 地址范围的数据以及寄存器空间，因此我们使用 FSR 寄存

器就可以访问到所有数据空间。MOVLI 专用于初始化 FSR 寄存器。FSR 设置为需要访问数据的基地址，然后使用 IMOVW 指令将 FSR 指向的数据读出到 W 工作寄存器；或者使用 IMOVF 指令，将 W 工作寄存器中的数据写入到 FSR 指向的数据地址中。使用 IMOVW/F++/--可以在操作完成后自动递增/递减 FSR 的值。也可以使用 IMOVW/F +/-q 指令，指定一个相对于 FSR 的 4 位偏移地址进行读写。这种带偏移量的指令执行完成后，不会更新 FSR 寄存器。

FSR– 间接寻址地址/基地址寄存器								
地址: 0x04					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
FSR	FSR[7:0]							
R/W	R/W							
Initial	8'h00							
Bit	Name	描述						
7:0	FSR	间接寻址模式下，用于设置目标地址或者目标地址的基地址						
备注：可用于访问程序空间和数据空间，当访问程序空间时，FSR 和 PCH 配合产生目标地址								

LCR – 循环次数寄存器

LCR 寄存器专用于配合 LOOP 指令实现零开销的循环操作。LCR 寄存器设置 LOOP 指令循环的次数。LCR 寄存器并没有映射到寄存器空间中，只能通过 MOVLC 指令访问。LOOP 指令的执行流程如下：

首先使用 MOVLC 指令设置程序循环的次数，然后在需要循环操作的地址执行 LOOP 指令，LOOP 指令执行时对 LCR 寄存器递减，如果递减后 LCR 为零，则循环结束，程序继续执行 LOOP 后面的指令。否则将直接跳转到 LOOP 指令指向的程序地址开始执行。

下面的代码演示如何使用 LOOP 指令进行循环操作。实例代码从程序空间 0x340 出读取 32 字节的数据，并依次写入到 0x40 开始的 RAM 空间。这里因为 MOVWP 和 IMOVF 都要使用到 FSR 作为间接寻址寄存器，因此我们使用了 0x340 这个地址，程序空间与数据空间的 FSR 可以保持一致。

	MOVLT	0x340	; initialize pointer to PMEM[0x340], FSR=0x40
	MOVLC	0x20	; set loop counter to 32
NEXT	MOVWP++		; read program data to W, then FSR+=1
	IMOVF	-1	; write W to DMEM[FSR-1] consider FSR has increased
	LOOP	NEXT	; loop 32 times
DONE			; job is done

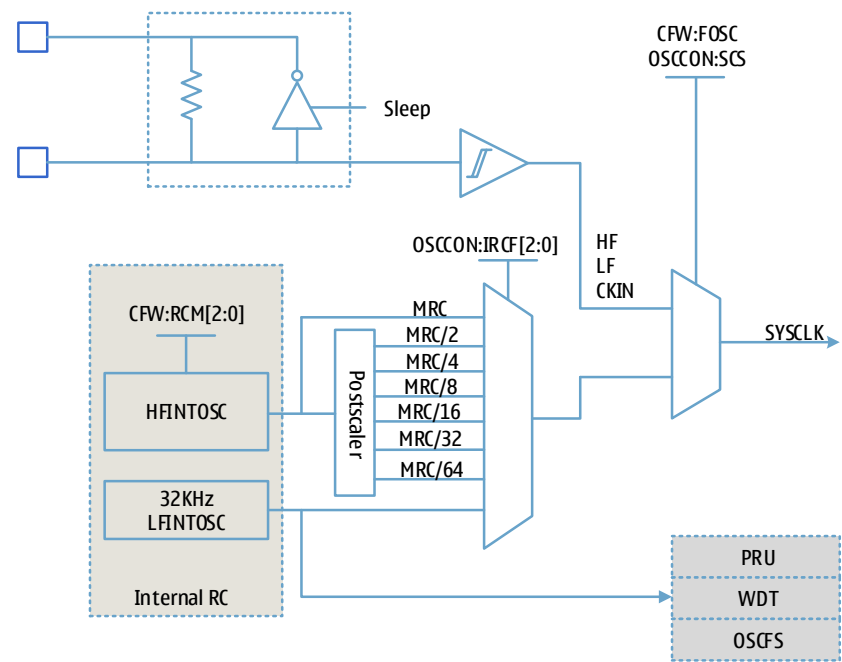
LCR – 循环次数寄存器								
地址: XX					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
LCR	LCR[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	LCR	循环次数寄存器，配合 LOOP 指令使用						
备注: LOOP 指令执行时，先判断 LCR 的值，如果为 0，则直接运行下一条指令；否则 LCR 的值减 1，程序跳转到 LOOP 指定的目标地址执行。LCR 寄存不寻址范围内，只能通过 MOVLC 访问								

系统时钟与复位

- 可校准内部主时钟 HFINTOSC 振荡器
 - 可配置为 1/8/16MHz 系统输出
 - 倍频模式可输出 2/16/32MHz 高速时钟
- 4T/2T/1T 可配置指令周期
- 32KHz 低功耗 LPRC 振荡器
- 外部低频/高频晶振支持
 - 晶振失效检测
- 看门狗定时器复位
- 可编程低压复位电路
- 外部复位输入
- 可配置启动时间

时钟模式

LGT8F684A 的系统时钟可选为内部或者外部时钟源。用户可以根据应用的功耗需求，选择适当的时钟源以达到合理的应用设计。下图说明了系统时钟支持的时钟源配置：



CFW: Configuration Words

系统时钟源可通过配置字设置为来自外部的高频或者低频晶振，外部直接时钟输入以及来自内部多种振荡器时钟源。下面的表格列出了系统支持的全部时钟配置模式。配置字相关的详细定义，请参考本手册“系统配置位”章节。

时钟模式	配置描述	其他说明
RCM	系统时钟来自内部高速 HFINTOSC 振荡器	可以通过 CFW:RCM 进一步配置系统时钟为 1/8/16MHz。系统运行时钟的频率还可以继续通过 OSCCON 寄存器的 IRCF 进行

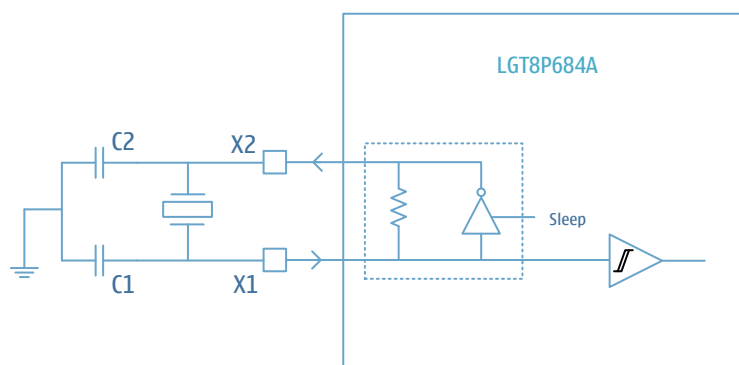
		预分频设置
RCK	系统时钟来自内部 32KHz LFINTOSC	
HFOSC	系统时钟来自外部高速晶振	也支持最高到 20MHz 的高频晶振。当系统工作频率高于 16MHz，建议系统提供高于 3.0V 的工作电压
LFOSC	系统时钟来自外部低速晶振	32.768KHz 常用晶振
CLKIN	直接外部时钟源输入(RA5)	

外部时钟源

时钟模式 HFOSC/LFOSC 以及 CLKIN 均为外部时钟输入模式。

HFOSC 外部高频晶振模式，支持外部 2M~20MHz 的外部晶振输入。对于需要精确定时的应用，可以根据需要选择合适的外部晶振。但需要注意，当系统时钟大于 16MHz，建议用户同时将系统保持在高于 3V 的工作电压。同时，对于频率较高的外部晶振，建议在晶振的两端接上 16pF~22pF 左右的电容，这样有利于晶振稳定起振。

LFOSC 外部低速晶振模式，支持外部 32KHz 范围的晶振输入。建议在外部晶振的两端接 16pF 左右的电容，这样便于晶振的稳定起振。



外部晶振与电容的选择：

外部晶振	外接电容
32KHz~4MHz	C1/C2 : 16pF
4MHz~20MHz	C1/C2 : 16pF~22pF

当系统时钟配置为外部晶振后，系统在上电/复位以及休眠模式唤醒后，会消耗额外的时间等待晶振稳定。当系统时钟为外部高速晶振，起振时间较快，不会对上电/复位以及休眠唤醒产生明显的影响；如果是外部低速晶振模式，系统需要额外消耗数百毫秒的时间等待晶振稳定。具体等待时间，请参考下面的数据：

晶振模式	附加稳定周期
LPOSC 模式	上电/复位： 休眠唤醒：
HPOSC 模式	上电/复位： 休眠唤醒：

CLKIN 外部有源时钟输入。在应用允许的情况下，可以直接从外部提供一个有源的时钟供系统工作。外部时钟信号直接从 **X1/RA5** 输入。当系统时钟模式配置为 **CLKIN** 模式，系统时钟从 **RA5** 输入，**RA 4** 引脚可以作为通用 **GPIO** 使用。

对于所有外部晶振输入模式，当系统进入深度休眠模式后，外部晶振将会被硬件自动关闭。用户可以通过寄存器配置选择在休眠模式中开启内部 **32KHz** 晶振，实现更为灵活的唤醒方式。此部分相关细节请参考本手册“功耗管理”部分。

内部时钟源

对于大部分对时钟精度没有特殊要求的应用，**LGT8F684A** 提供了丰富的内部时钟资源，用户可以根据需求，在功耗和性能之间做出合理的选择。**RCM/RCK** 均为内部时钟模式。

RCK 选择内部 **32KHz RC** 时钟作为系统时钟源。内部 **RC32K** 非校准时钟，当系统运行电压在 **2.0V** 到 **5.5V** 之前变化时，频率将有高于 **5%** 的变化。此 **RC32K** 仅仅提供一个低频的时钟，可以让系统运行在一个较低功耗下运行。如果应用需要较为精确的 **32.768K** 时钟源，请选择 **LFOSC** 模式；

RCM 选择内部高频可校准 **RC** 振荡器(**HFINTOSC**)作为主时钟源。在 **RCM** 模式下，用户可以使用配置位选择 **HFINTOSC** 的中心频率，**HFINTOSC** 输出频率可以在 **1M/8M/16M** 三个中心频率中选择。当 **HFINTOSC** 使能了倍频模式，将会产生一个对应的 **2M/16M/32M** 高频输出，这个高频输出主要是提供给定时计数器模块，以产生一个高速高精度的 **PWM** 输出。**HFINTOSC** 位可校准时钟源，芯片在出厂测试时，已将对应于 **1M/8M/16MHz** 三组中心频率的校准写入保留的系统配置空间，用于只需要通过 **CFW:RCM** 配置应用所需的时钟配置，硬件将自动完成对应频率的校准工作。晶振校准后，当系统供电电压在 **2.0V** 到 **5.5V** 的工作范围内变化时，可以保证 $\pm 2\%$ 的精度。

下面的表格列出 **RCM** 配置位与内部时钟源的对应关系：

CFW1: RCM[1:0]	HFINTOSC Frequency (RCM)
00	1MHz
01	8MHz
1X	16MHz

内部时钟源预分频与多路复用

系统时钟管理模块实现了一个主时钟预分频器，可以进一步对 **HFINTOSC** 的输出时钟进行预分频。预分频器的输出与内部 **32KHz RC** 时钟一起，经过多路复用，可以为系统运行提供更加丰富的选择。

主时钟预分频器为一个 **5** 位预分频单元，时钟源来自内部高频主时钟源(**HFINTOSC**)，经过预分频器，可以产生 **HFINTOSC/2**，**HFINTOSC/4**，...**HFINTOSC/64** 一共 **6** 种分频时钟输出。这六种时钟输出加上 **HFINTOSC** 时钟本身以及内部 **32K IRC**，一共 **8** 种时钟选择。这 **8** 种时钟通过内部多路复用器，在 **OSCCON:IRCF[2:0]**寄存器的控制下，选择内部系统时钟源(**INTOSC**)，

内部系统时钟(**INTOSC**)与来自外部的时钟源，最终通过 **OSCCON:SCS** 位或者 **CFW:FOSC** 配置位，选择最终用于驱动系统时钟工作的主时钟源。

OSCCON:SCS 的优先级高于 **CFW:FOSC** 配置位设置。当 **OSCCON:SCS=1** 时，系统被限制为只能工作于内部时钟源(**INTOSC**)，当 **OSCCON:SCS=0** 时，系统时钟源将由 **CFW:FOSC** 配置位决定。

OSCCON – 振荡器控制寄存器

OSCCON 寄存器控制着系统时钟以及频率选择。OSCCON 寄存器包括以下功能：

- 内部频率选择位(IRCF)
- 振荡器状态位(HTS/LTS)
- 系统时钟控制位(OSTS/SCS)

OSCCON– 振荡器控制寄存器								
地址: 0x8F					初始状态: 01100000			
Bit	7	6	5	4	3	2	1	0
	-	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS
R/W	-	R/W-1	R/W-1	R/W-1	R-1	R-0	R-0	R/W-0
Bit	Name	描述						
7	-	保留						
6:4	IRCF[2:0]	内部时钟选择位 111 = RCM 110 = RCM/2 (default) 101 = RCM/4 001 = RCM/64 000 = 31KHz IRC						
3	OSTS	晶振启动超时状态位 1 = 系统工作于由 FOSC 定义的外部时钟状态 0 = 系统工作于内部时钟状态						
2	HTS	内部高速振荡器(HFINTOSC)工作状态 1 = HFINTOSC 输出时钟已稳定 0 = HFINTOSC 输出时钟不稳定						
1	LTS	内部低速振荡器(LFINTOSC)工作状态 1 = LFINTOSC 输出时钟已稳定 0 = LFINTOSC 输出时钟不稳定						
0	SCS	系统时钟选择 1 = 系统时钟位内部时钟源(INTOSC) 0 = 系统时钟由 CFW:FOSC 配置位控制						

OSCTUNE – 内部主振荡器(HFINTOSC)校准寄存器

内部高频主振荡器 (HFINTOSC) 在出厂时已经进行了校准。用户仍然可以通过 OSCTUNE 寄存器对 HFINTOSC 的输出频率进行调整，以输出合适的系统工作频率。

OSCTUNE 的初始值与系统配置位 RCM 的设置相关，系统上电过程中，初始化模块将根据 RCM 的设置将 OSCTUNE 设置为对应中心频率的校准值。

OSCTUNE 是一个 8 位的可写寄存器，0x00 对应 HFINTOSC 最低输出频率，0xFF 对应 HFINTOSC 输出最高频率。设置 OSCTUNE 仅仅影响当前 HFINTOSC 的中心频率值，对 LFINTOSC 无影响。

OSCTUNE – HFINTOSC 校准寄存器								
地址: 0x90					初始状态: 出厂校准值			
Bit	7	6	5	4	3	2	1	0
	OSCTUNE[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	OSCTUNE	HFINTOSC 频率校准位 0x00 = 最小输出频率 0xFF = 最大输出频率						

双时钟启动模式

双时钟启动模式针对外部晶振作为主时钟的应用。双时钟启动模式通过最小化外部晶振启动到代码执行之间的时间, 获得更低的系统启动功耗。对于一些需要频繁在休眠模式与正常工作模式至今切换的应用, 双时钟启动模式可以避免外部晶振启动到稳定之间较长的等待时间, 从而降低了应用周期内的平均功耗。

双时钟启动模式主要是通过允许系统在唤醒之后, 首先使用内部时钟(INTOSC)执行一些指令, 完成必要检查共工作后, 迅速的再次进入休眠模式。

当系统配置为外部晶振工作模式时(LP/HS), 晶振启动定时器(OST)将会被使能。OST 定时器将一般为 1024 个振荡周期, 只要当振荡周期计数完成后, 系统才开始执行程序。双时钟启动模式通过在唤醒后使用内部振荡器执行程序的方式, 最小化 OST 对程序执行产生的影响。当 OST 计数完毕, OSCCON 寄存器的 OSTS 位被置位, 系统时钟切换至外部晶振继续工作。

双时钟启动模式配置

使能双时钟启动模式的配置如下:

1. 使能系统配置位 CFW:TSSM = 1
2. 设置 OSCCON:SCS = 0
3. 通过系统配置位 CFW:FOSC 配置系统时钟源为外部晶振

双时钟模式使能后, 系统在以下情况时, 将进入到双时钟启动模式:

1. 上电复位发生后, 在上电定时器超时后;
2. 从休眠模式唤醒后;

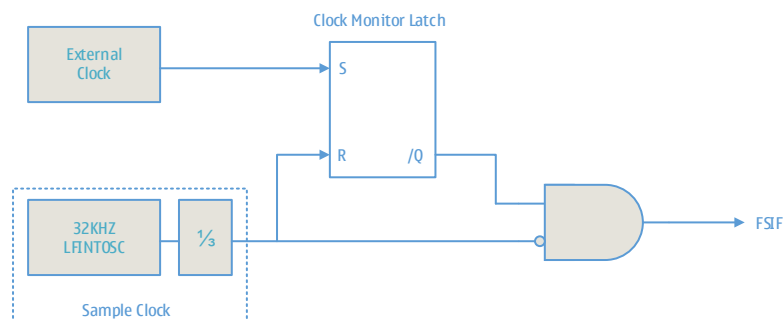
如果系统时钟源被配置为非外部晶振模式, 双时钟模式将会自动被禁止。

双时钟启动时序

1. 上电复位或者系统唤醒;
2. 系统使用由 OSCCON:IRCF 设置的内部时钟工作;
3. OST 使能并计数 1024 个周期
4. OST 计数超时;
5. OSCCO:OSTS 位被设置;
6. 系统时钟切换至外部时钟源

外部晶振失效监控

晶振失效监控(FSCM)允许系统在外部晶振故障后，仍然能够继续运行。FSCM 能够检测到 OST 超时后的晶振失效。FSCM 模式通过系统配置位 CFW:FSEN 使能。晶振失效监控仅对于外部晶振模式有效。FSCM 结构如下图所示：



失效保护检测

FSCM 模块通过在采样周期内检测晶振的输出变化来确定晶振是否正常工作。采样频率由内部 32KHz 低频 RC 振荡器经过分频得到。如上 FSCM 结构图所示，FSCM 内部主要检测功能由一个锁存器实现，外部晶振时钟在每一个下降沿置位锁存器输出；采样时钟在其上升沿复位锁存器。当在一个采样周期内，锁存器的输出持续位低电平，FSCM 将会报告一个时钟失效输出。

失效保护操作

当外部晶振发生了时钟失效，时钟管理单元将系统时钟切换到内部振荡器时钟。同时 FSCM 设置晶振失效中断标志位 OSFIF(PIR1 寄存器)。如果此时外设中断使能控制寄存器(PIE1)的 OSFIE 位使能了 FSCM 中断，OSFIF 将会产生一个有效的中断请求。系统在 OSFIF 之后将持续由内部时钟驱动，直到软件成功的重新启动外部晶振并手动切换至外部晶振。

外部晶振失效后，系统将切换至内部时钟源。内部时钟由 OSCCON 寄存器的 IRCF[2:0]决定。这样可以允许在时钟失效发生之前，配置一个合理的备用的内部时钟频率。

时钟失效使用注意事项

FSCM 被设计为检测晶振启动之后的失效状态。晶振启动中包含了一个完整的启动定时器超时计数(OST)。因此 OST 在复位以及唤醒后，都处于有效状态，因此 FSCM 需要在 OST 之后才能够正常工作。当 FSCM 被使能后，双时钟启动模式也同时被使能。因此系统将会在 OST 计数的同时执行程序代码。当用户执行完必须的检测工作后，需要通过检测 OSCCON:OSTS 位来确定外部晶振是否已经完成启动并正确的完成了时钟切换。

复位控制

LGT8F684A 支持以下 4 种复位控制：

1. 上电复位 (POR)
2. 看门狗溢出复位 (WDT)
3. 外部复位输入 (RSTN/RA3)
4. 低电压检测复位 (LVR)

系统中大部分寄存器在以上几种复位条件下，都会被复位到一个初始状态。但也有部分寄存器仅在 POR 复位或者 LVR 复位有效时才会被复位。此部分相关细节，请参考本章节后面寄存器与复位关系的表格描述。

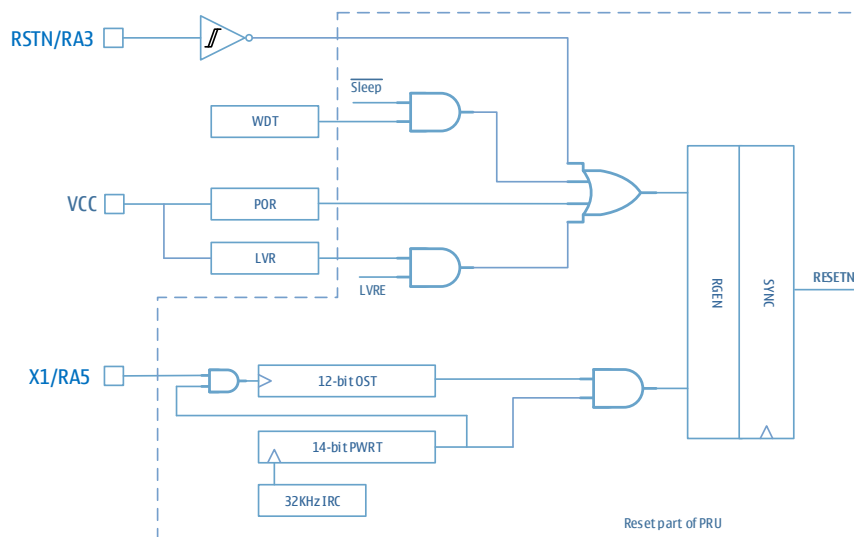
与看门狗正常的溢出复位不同，看门狗唤醒后，不会对寄存器产生影响，因为从休眠模式唤醒，一般是被处理为恢复到正常运行的状态。在系统默认状态下，WDT 是开启状态。用户可以通过 PCON 寄存器或者配置字对 WDT 的运行进行控制。看门狗具体实现，请参考本手册相关章节。

在系统默认状态下，RSTN/RA3 引脚为外部复位输入。用户可以通过配置字改变 RA3 的默认状态。关闭 RSTN/RA3 引脚的外部复位功能后，RSTN/RA3 可做为一个开漏输出的 I/O 使用。

低电压检测复位 (LVR) 是一个阈值可配置的低电压检测模块。用户可以根据应用环境需求，将 LVR 配置为适当的监控阈值，当系统供电 (VCC) 电压低于预设阈值时，LVR 将产生一个持续的复位信号，将系统强制为复位状态。LVR 模块默认是关闭的，用户可以通过 PCON 寄存器或者配置字控制 LVR 的开启和关闭。LVR 的检测阈值电压只能够通过配置字设置。

STATUS 寄存器中的 TO/PD 位会根据当前系统的复位情况而变化，用户可以通过这两位判定导致系统复位的原因。TO/PD 与复位源的对应关系，请参考本章后续介绍。

下图为系统复位的结构：



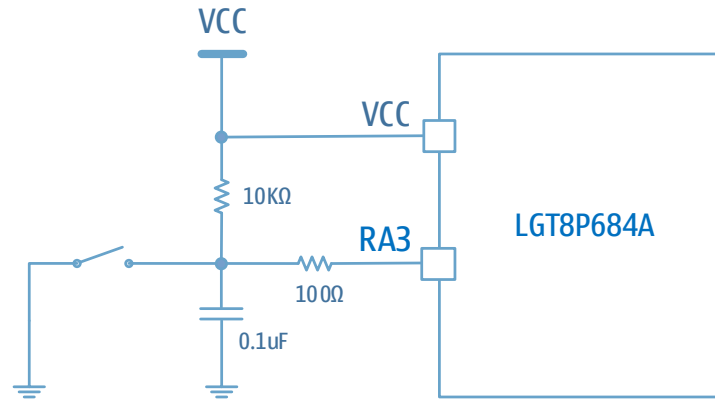
上电复位 (POR)

在系统上电过程中，在 VCC 电压上升到正常的工作电压范围之前，POR 电路将一直保持复位输出有效状态。内部 POR 电路的存在可以省去接在 RSTN/RA3 引脚上的外部 RC 复位电路，仅仅使用一个电阻将 RSTN/RA3 与 VCC 相连接即可。在系统工作的过程中，POR 也会持续监控 VCC 的电压变化，当 VCC 电压掉到 POR 阈值以下时，POR 电路产生有效的复位输出，将系统保持在复位状态。

当 LVR 使能后，系统的复位状态除了由 POR 控制，也会同时受到 LVR 电路的控制。LVR 相关特性请参考本手册电气特性部分。

外部复位输入 (RSTN/RA3)

LGT8F684A 的 RSTN/RA3 引脚默认为外部复位引脚，当此引脚工作在外部复位模式时，内部强制上拉到 VCC。如果需要外接复位电路，建议采用如下参考电路：



由于 RA3 同时也作为程序编程所需高压 VPP 供电引脚，为避免在正常工作时因 RA3 的电压高于 VCC 进入编程模式，一般建议在芯片外部将 RA3 与 VCC 直接通过一个电阻连接。

低压复位电路 (LVR)

低压复位电路的作用对于供电电压有特殊要求的应用非常重要，比如系统需要运行在较高的频率，而这个频率需要系统供电保持在一个较高的电压范围内。此时就需要通过开启 LVR 模块，实时检测供电电压的变化。当供电电压低于设置的阈值时，LVR 输出有效复位，系统保持为复位状态。否则，供电电压下降到无法保证系统在较高的频率运行时，将会因时序问题导致运行错乱。

LVR 默认为关闭状态。用户可以通过配置字开启 LVR 并设置复位阈值。当系统处于休眠模式时，如果系统使能了 LVR，LVR 在休眠模式下仍然会继续保持运行。为获得更低的休眠功耗，可以通过清零配置字的 PMOD 位，休眠控制模块在进入休眠模式前，自动关闭 LVR 模块。LVR 将会在系统被唤醒后自动使能。具体设置，请参考本手册系统配置位相关介绍。

LVR 复位与 POR 复位同属于最高级别的硬件复位，能够复位系统中大部分的寄存器，包括其他复位标志。

启动定时器 (SUT)

LGT8F684A 实现了一个可编程的启动时间定时器，SUT 在 PWRT 中实现，通过控制 PWRT 的定时器实现一个可配置的启动时间。用户可以通过配置字，选择合适的启动时间。

不同的启动时间设置用于特殊的工作环境，比如对于电源上电比较缓慢的应用，POR 在较低的电压点就会释放，如果此时系统进入工作状态，但由于电源上电缓慢，系统就会在一个较低的电压下工作，容易导致系统工作不稳定，造成功能错乱。此时需要将 SUT 配置为较大的时间，这样可以保证在电源上升到合适的电压值之前，系统一直处于复位状态。

另外，对于外部晶振应用，如果晶振本身起振较慢，也建议适当调整 SUT 的配置，保证在系统时钟源切换到外部晶振前，晶振处于稳定状态。

当然，对于要求系统快速启动的应用，也可以通过调整 SUT，达到应用的需要。

启动时间需要通过配置字设置，系统最终进入正常工作的时间除了与 SUT 相关，还与系统是否启用了外部晶振有关。当系统开启了外部晶振，会在 SUT 之后，附加一个额外的晶振启动时间(OST)。系统的启动时间与 PWRT 以及 OST 的关系，请参考如下列表：

FOSC Mode	SUT Settings	POR or LVR	RSTN or WDT
RCM RCK CLKIN	00	63ms	125us
	01	254ms	
	10	2ms	
	11	16ms	
HFOSC LFOSC	00	63ms + 2048*FOSC	
	01	254ms + 2048*FOSC	
	10	2ms + 2048*FOSC	
	11	16ms + 2048*FOSC	

寄存器复位状态

在不同复位模式下，寄存器被复位的情况稍有不同。系统中几乎所有寄存器都会被上电复位或者 LVR 复位。但对于外部复位或者看门狗复位，只有部分寄存器的值会受到影响。当系统从休眠模式唤醒后，内核恢复休眠前的指令处继续执行，因此基本上所有的寄存器状态都不会受到影响。寄存器初值与各种复位直接的关系，请参考下面的表格：

Register	Address	POR/BOR	WDT/RSTN
W	-	0000_0000	UUUU_UUUU
INDF	00h/80h	XXXX_XXXX	UUUU_UUUU
TMR0	01h	0000_0000	UUUU_UUUU
PCL	02h/82h	0000_0000	0000_0000
STATUS	03h/83h	0001_1000	000Q_QUUU
FSR	04h/84h	0000_0000	UUUU_UUUU
PORTA	05h	0000_0000	UUUU_UUUU
DC1R	06h	0000_0000	0000_0000
PORTC	07h	XX00_0000	XXUU_UUUU
PR1L	08h	1111_1111	1111_1111
PR1H	09h	1111_1111	1111_1111
PCH	0Ah/8Ah	XXXX_XX00	XXXX_XX00
INTCON	0Bh/8Bh	0000_0000	0000_0000
PIR1	0Ch	XXXX_0XX0	XXXX_0XX0
TMR1L	0Eh	0000_0000	UUUU_UUUU
TMR1H	0Fh	0000_0000	UUUU_UUUU
T1CON	10h	0000_0000	UUUU_UUUU
TMR2	11h	0000_0000	UUUU_UUUU
T2CON	12h	X000_0000	XUUU_UUUU
CCPR1L	13h	0000_0000	0000_0000
CCPR1H	14h	0000_0000	0000_0000
CCP1CON	15h	0000_0000	0000_0000
PWM1CON	16h	0000_0000	0000_0000

ECCPAS	17h	X000_0000	X000_0000
WDTCON	18h	0000_1000	0000_1000
CMCON0	19h	XX00_0000	XX00_0000
CMCON1	1Ah	0000_0010	0000_0010
CMCON2	1Bh	0000_0000	0000_0000
APCON	1Ch	0000_0000	0000_0000
TCCR	1Dh	0000_0000	0000_0000
ADRESH	1Eh	XXXX_XXXX	XXXX_XXXX
ADCON0	1Fh	0000_0000	0000_0000
OPTION_REG	81h	1111_1111	1111_1111
TRISA	85h	1111_1111	1111_1111
TRISC	87h	XX11_1111	XX11_1111
COGCR0	88h	0000_0000	0000_0000
COGCR1	89h	0000_0000	0000_0000
PIE1	8Ch	0000_0000	0000_0000
COGPHR	8Dh	0000_0000	0000_0000
PCON	8Eh	1000_0001	UUUU_UUUU
OSCCON	8Fh	0110_0000	UUUU_UUUU
OSCTUNE	90h	0000_0000	UUUU_UUUU
ANSEL	9Fh	1111_1100	1111_1100
PR2	92h	1111_1111	1111_1111
PDCR	93h	XX00_0000	XX00_0000
WPUC	94h	0000_0000	0000_0000
WPUA	95h	0000_0000	0000_0000
IOCA	96h	0000_0000	0000_0000
PDAR	97h	1111_1111	1111_1111
ANSEL1	98h	0XXX_XX00	0XXX_XX00
DACON	9Dh	0000_0000	0000_0000
ADRESL	9Eh	XXXX_XXXX	XXXX_XXXX
ADCON1	9Fh	0000_0000	0000_0000

Legend: U: unchanged, X: unimplemented, Q: value depends on condition

PCON – 功耗控制寄存器

PCON 主要用于控制系统低功耗模式，PCON 的最低两位为硬件复位标志位，指示当前的复位是上电复位(POR)还是低电压复位(LVR)，PCON 的复位状态位与 STATUS 寄存器中的 TO/PD 标记共同决定着系统中所有复位状态。

PCON 寄存器中也包含 WDT 和 LVR 的使能控制位。WDT 和 LVR 可以通过配置字控制。PCON 提供了一个方便的控制方法，用户可以使用软件对 WDT 或者 LVR 进行控制。

PCON 中包含了两位功耗模式控制位：DPSM1/0。DPSM 用于设置系统的休眠模式，用户需要根据应用的功耗控制需求设置 DPSM，随后执行 SLEEP 指令，内核将进入休眠模式。休眠模式的具体定义和配置使用方法，请参考本手册功耗管理相关章节。

PCON 中的 SWDD 位用于控制编程接口(SWD)。LGT8F684A 的编程接口本身就具有非常安全的数据保护算法。用户仍然可以通过此位禁用所有 SWD 接口的相关操作。

STATUS/PCON 复位状态:

POR	LVR	TO	PD	Conditions
0	1	1	1	上电复位
U	0	1	1	低压检测复位
U	U	0	U	看门狗复位
U	U	0	0	看门狗唤醒
U	U	U	U	外部复位
U	U	1	0	外部复位唤醒

说明: U = unchanged

PCON- 功耗控制寄存器								
地址: 0x8E					1000_0001			
Bit	7	6	5	4	3	2	1	0
	PPLP	SWDD	TMR0D	LVRE	DPSM1	DPSM0	POR	BOR
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	PPLP	程序存储器低功耗模式控制, 当系统运行在较低频率时, 可以通过开启低功耗模式降低系统运行功耗: 1 = 正常模式 0 = 低功耗模式						
6	SWDD	SWD 接口禁止位, 1: 禁止 SWD						
5	TMR0D	定时器 0 禁止位, 1: 禁止 Timer0						
4	LVRE	LVR 复位使能位, 1: 使能 LVR 的复位功能						
3:2	DPSM1/0	休眠模式控制位 00 : 仅关闭内核运行时钟, 用于快速唤醒 01 : 关闭 RCM, 系统时钟切换到 32K IRC 1x : 深睡眠模式, 关闭所有时钟源						
1	POR	上电复位标志位						
0	BOR	低压检测复位标志位						

功耗管理

- 休眠模式控制(DPSM)
- 外设中断以及 WDT 定时唤醒
- TMR2 异步模式定时唤醒
- 最低 1uA@3.3V 待机功耗

综述

LGT8F684A 实现了一个简单高效的功耗管理模块(PRU)，实现自动功耗控制。用户可以根据应用需求，选择合适的功耗管理模式，在满足应用功能以及性能的前提下，实现最为合理的功耗控制。

功耗管理模式主要由 PCON 寄存器以及其他相关配置字控制。LGT8F684A 支持多种时钟源，在应用的性能约束下，建议选择最合适的时钟配置，以获得更低的动态功耗。关于时钟源相关的配置，请参考本手册系统时钟相关章节。

LGT8F684A 的功耗管理模块可以对部分模块进行管理，也有部分模块的管理需要用户自行控制，比如模数转换器，模拟比较器等。如果在休眠模式下这些模块不需要工作，建议将其关闭，以避免不必要的功耗浪费。低压复位模块(LVR)可以通过 PCON 寄存器的 LVRE 位进行控制，也可以通过配置字的 LVRPM 位控制，功耗管理模块可以根据 LVRPM 位的配置，在进入休眠模式前自动关闭 LVR，并在唤醒后自动启动 LVR 模块。LVRPM 相关配置请参考本手册系统配置位章节。

功耗模式

软件使用 PCON 寄存器的 DPSM 位配置休眠模式，然后执行一条 SLEEP 指令进入休眠状态。进入休眠状态后，内核暂停在 SLEEP 指令之后的下一条指令上，其他模块的工作状态以及唤醒方式，与用户的配置有关，下面的表格详细列出休眠模式下模块的工作状况以及唤醒源：

DPSM[1:0]	Core	LVR	WDT	OSC	RCM	RC32K	Analog	Timers	唤醒源
00	X	(1)	(2)	(3)	√	√	(2)	(2)	外部复位 所有中断(4)
01	X	(1)	(2)	(3)	X	√	(2)	(2)	外部复位 所有有效中断(5)
1X	X	(1)	X	X	X	X	X	X	外部复位(6) 外部 INT/RA2(7) RA 引脚变化唤醒

(1): LVR 在休眠模式下的工作状态由 LVR 的配置位，PCON 的 LVRE 控制位以及配置位 LVRPM 共同决定。

如果用户没有通过 LVR 配置字或者 PCON 寄存器使能 LVR，LVR 将一直处于关闭模式。如果系统使能了 LVR，并且 LVRPM=0，LVR 将在进入休眠模式前自动被关闭，并在被唤醒后重新开启 LVR；

(2): 包括 ADC/AC/DAC 等模拟功能模块，在此模式下模块的工作情况取决于用户是否使能了该模块；

(3): OSC 包括 HFOSC/LFOSC 两种模式，只有在这两种模式下，此时 OSC 处于工作状态；

(4): 当用户使能了模块的中断功能；

(5): 在此模式下，系统时钟被切换到内部 32KHz IRC，基本上所有的模块都可以在此频率下正常工作。

需要特别注意，此时系统时钟已经改变，与系统时钟相关的模块运行状态与休眠前可能不一致；

(6): 外部复位可以将系统从休眠模式下唤醒，唤醒后系统也同时被复位。如果用户关闭了 RA3 的外部复位功能，RA3/RSTN 引脚将无法唤醒系统；

- (7): 系统支持 INT/RA2 电平唤醒, INT 的电平由 OPTION 寄存器 INTDGE 位控制, 上升沿对应高电平, 下降沿对应低电平;

休眠模式由 DPSM 控制, 主要分为三个级别:

1. **DPSM = 00**, 这是默认的休眠模式, 软件执行 sleep 指令后, 内核停止继续取指运行, PC 指针保持指向 SLEEP 之后的指令。PRU 模块关闭内核时钟, SRAM 以及程序存储器 OTP 进入待机模式。其他外设时钟在此模式下是正常工作的。用户可以根据需要, 关闭应用无关的模块以节省功耗。此模式可以通过系统中所有有效的中断唤醒。
2. **DPSM = 01**, 与第一种模式相比, 此休眠模式将系统时钟强制切换至内部 32KHz RC 振荡器。并关闭其他所有时钟源。内核停止运行, 内核时钟被关闭。其他外设仍然可以继续运行, 但此时外设的运行时钟也同时被切换至内部 32KHz RC 振荡器。用户需要特别注意此模式下时钟的变化。此模式可以通过系统中所有可用中断唤醒。用户应该根据需要, 关闭其他与应用无关的模块。此模式在获得相对较低的休眠功耗, 同时也可以支持外部引脚变化中断唤醒以及 WDT 定时唤醒。
3. **DPSM = 1x**, 当 **DPSM[1] = 1** 时, LGT8F684A 进入深睡眠模式。此模式下, PRU 关闭所有时钟, 包括内部 32KHz RC 振荡器。此模式可以通过以下几种方式唤醒:
 - a) 外部中断引脚(INT/RA2),
 - b) 外部复位
 - c) 引脚电平变化中断唤醒。

低功耗应用注意事项

为在实际应用中获得更低的休眠功耗, 需要在应用电路设计以及软件编程时对系统漏电做充分的分析, 避免一些不合理的电路配置产生的多余耗电。下面列出一些注意事项, 供设计过程参考:

1. 系统功耗(包括动态运行功耗以及休眠功耗)与系统的工作电压最为相关, 在应用允许的前提下, 能够降低工作电压是获得更低功耗最有效的办法。电压的控制往往被应用本身限制, 除了控制电压, 用户还可以考虑在满足应用性能需要的前提下, 降低系统的运行频率, 或者对系统的频率做动态的管理, 这样也可以获得更低的动态功耗;
2. 避免浮空的 I/O。I/O 浮空, 或者输入状态的 I/O 被外部驱动到一个中间电平, 都会因 I/O 内部电路的震荡产生比较大的漏电 (一般为几十到上百个 uA)。因此应该避免有浮空的 I/O。LGT8F684A 内部有可控的上拉下拉电阻, 用户可以通过寄存器控制将处于输入模式的 I/O 设置到一个合理的电平。
3. 作为输出的 I/O, 在进入休眠模式前, 应该根据 I/O 外部电路的情况, 将 I/O 驱动到合理的电平, 避免产生漏电路径。比如对于外部有上拉的 I/O, 在休眠模式下应将 I/O 也驱动到高电平, 避免上拉电阻对地产生漏电。对于外部没有上下拉的 I/O, 应尽量将 I/O 驱动到低电平, 因为对于推挽结构的 I/O, 驱动到高电平需要芯片内部消耗多余的电量。
4. 再进入低功耗模式前, 关闭所有模拟功能模块以及合理处理相关模拟输入。LGT8F684 内部比较器输入中集成了输入分压, 需要注意将其关闭。ADC 模块中的输入分压 VDO 默认情况下并没有关闭, 需要在进入休眠模式前通过 ADCON1 寄存器的 VDS 位将其关闭。
5. 对于生产代码大于或等于 1622 的芯片, DAC 模块默认使能了内部 1.2V 参考电压源。即使在应用没有使用 DAC1/2 模块的情况下, 也需要在休眠前通过 VRCON1[DBG]位关闭内部参考。

低功耗应用例程: 下面的例程示例休眠模式的应用。在进入休眠模式去, 首先打开 I/O 的上拉电阻, 避免浮空的 I/O 产生漏电; 通过 PCON 寄存器设置 DPSM 为深睡眠模式, 此模式下仅支持外部中断 INT 唤醒和外部复位唤醒。通过 OPTION 寄存器的 INTDGE 设置 INT 的唤醒电平为低电平。主循环 LOOP 翻转 RA4 引脚, 用于指示程序的运行状态。通过判断 RA5 的电平决定是否进入休眠模式。

这里需要注意：LGT8F684A 的 RA4 引脚默认为模拟 I/O（比较器输入），通过 ANSEL 寄存器禁止了该 I/O 的模拟功能后，RA4 为系统时钟输出功能。需要通过配置位将 RA4 的系统时钟输出功能禁用后，RA4 才可以作为 GPIO 使用。

	BANKSEL	ANSEL	; select bank 1
	CLRF	ANSEL	; disable analog function of I/O
	BSF	PCON, 3	; enable deep sleep mode (DPSM[1]=1)
	BCF	TRISA, 4	; RA4 as output (toggle indicator)
	BCF	OPTION_REG, 6	; INTEDG=0, low level wakeup on INT
	MOVLW	0x3F	;
	MOVWF	WPUA	; enable all pullup of PORTA
	BCF	OPTION_REG, 7	; global pullup enable
	BSF	ADCON1, 0	;
	BSF	ADCON1, 1	; disable interna VDO
	BANKSEL	PORTA	; select bank 0
	CLRF	PORTA	; clear I/O status
	MOVLW	0x10	; use to toggle RA4
LOOP	XORWF	PORTA	; toggle RA4
	BTFSS	PORTA, 5	;
	SLEEP		; goto sleep if RA5 = 0
	GOTO	LOOP	
END			; support by MPASM assembly

输入输出

- 内部可编程内部上拉电阻
- RA 组 I/O 电平变化中断
- RA 组 I/O 电平变化唤醒
- 2x80mA PWM 驱动 I/O
- 4x80mA 高扇入 LED 驱动 I/O

综述

LGT8F684A 最多支持 14 路可编程输入/输出端口(SOP16L 封装)。I/O 被分为 RA/RC 两组。当引脚的附加功能使能后，其中一些 I/O 将不能作为可编程输入/输出端口使用。

RA 的 I/O 都支持可编程的上下拉电阻。RC 组仅支持内部上拉，RA3/RSTN 引脚默认作为外部复位引脚，内部强制上拉。RA3/RSTN 只能作为输入或者开漏输出的 I/O 使用。

I/O 的输入/输出方向通过 TRISA/C 寄存器控制。当设置 TRISA/C 某位为 1，对应的 I/O 为输入模式（默认），清零 TRISA/C 位，相应的 I/O 被设置为输出模式。RA3 只能被设置为开漏输出的 I/O。

设置完成 I/O 的方向后，可以通过写 GPIO 寄存器设置 I/O 的输出状态，或者读 GPIO 寄存器获取当前 I/O 的状态变化。

TRISA/C 寄存器仅控制 I/O 的输入/输出方向。即便端口被设置为模拟功能。用户还是需要确保在 I/O 被设置为模拟功能使用时，TRISA/C 设置为输入模式。当 I/O 被配置为模拟端口后，读 GPIO 寄存器将固定返回 0。被设置为模拟功能的 I/O 也不能产生引脚电平变化中断。

I/O 初始化设置示例：

BANKSEL	PORTA	; select bank of PORTA
CLRF	PORTA	; init GPIO
BANKSEL	ANSEL	; select bank of ANSEL
CLR	ANSEL	; disable analog function
MOVLW	0x3C	;
MOVWF	TRISA	; set RA[1:0] as output

端口附加功能

LGT8F684A 的端口除 VCC/GND 之外，其他的 I/O 都支持可编程上下拉控制，引脚电平变化中断以及其他外设专属功能。下面章节将详细介绍 I/O 的这些附加功能。

模拟输入功能

LGT8F684A 内部集成丰富的模拟外设，可接受来自外部端口的模拟信号输入。ANSEL/1 寄存器用于控制比较器相关输入端口的状态。除 RC4/5 外，其他 IO 都具有模拟输入功能。上电复位后，大部分 IO 都处于默认的 GPIO 状态。软件可以通过 ANSEL/1 寄存器以及其他相关寄存器，使能相关 IO 的模拟输入功能，只有在使能了 IO 的模拟输入功能后，这些 IO 才能正确的作为模拟外设的模拟信号输入端口使用。

RA0/1 同时复用为编程控制接口(SWD/SWC)，默认状态下，RA0/1 作为 SWD 控制器的端口使用。用户可以通过 ANSEL 寄存器将 RA0/1 配置为模拟功能 I/O。此时 SWD 接口功能将被关闭。

当 I/O 被设置为模拟端口功能后，通过 PORTA/C 寄存器读到该端口的值将被固定为 0。但 ANSEL 位并

不会影响到该端口的输出功能。因此如果通过 **TRISA/C** 将端口设置为输出，同时又通过 **ANSEL** 将端口设置为模拟功能，这样可能会影响到数字以及模拟两种功能的正常运行，需要在使用时特别注意。

弱上拉/下拉电阻

RA 都具有可编程的内部上拉/下拉电阻。**RA** 下拉电阻可以通过 **PDAR** 寄存器控制。**RA/C** 每组 **IO** 的上拉电阻可以通过 **WPUA/WPUC** 寄存器单独控制。系统上电后，默认状态下所有的下拉电阻都处于禁用状态。**RA** 组端口(**RA3** 除外)的上拉被全局上拉控制位(**RAPU**)禁止。**RA3** 默认状态下为外部复位输入，内部强制上拉。关闭 **RA3** 的外部复位功能后，内部上拉可以通过 **PDAR/WPUA** 寄存器控制。**RC** 组端口的上下拉控制寄存器默认为无效状态，可以通过 **PDCR/WPUC** 对上/下电阻进行控制。

当 **I/O** 处于附加功能模式时，内部的上/下拉电阻将会被自动关掉。

引脚电平变化中断

RA 组每个 **I/O** 都可以通过 **IOCA** 寄存器分别单独配置为可产生引脚电平变化中断的端口。系统的引脚电平变化中断由 **INTCON** 寄存器的 **RAIE** 位控制。软件可以通过 **INTCON** 寄存器 **RAIF** 位判断是否为引脚电平变化而产生的中断。

引脚电平变化中断可用于唤醒控制。软件需要在中断复位中通过写零操作清除中断标记。

RC 组 **I/O** 没有引脚电平变化中断功能，也不能用于引脚电平变化唤醒。

其他外设功能

RA0/1 默认状态下作为芯片调试/量产的 **SWD** 接口。量产工具通过 **SWD** 接口烧写/验证程序空间中的数据。编程操作还需要通过 **RA3/VPP** 提供一个高压源。编程接口请参考相关文档。

RA4/5 可作为外部晶振的输入/输出端口。用户需要通过配置字设置时钟源(**FOSC**)模式为 **HFOSC/HFOSC**。当系统时钟源模式被设置为 **CLKIN**，系统时钟从 **RA5** 引脚灌入，**RA4** 可作为 **I/O** 使用。

RA4 在默认状态下为模拟功能。即使用户通过配置字设置 **RA4** 输出系统时钟。模拟功能具有最高的控制优先级。当通过 **ANSEL** 寄存器关闭了 **RA4** 的模拟功能后，**RA4** 可以作为系统时钟的输出端口。只有当同时禁用了 **RA4** 的模拟输入功能以及系统时钟输出功能，**RA4** 才可以作为一个 **GPIO** 使用。

RA3 默认为系统复位输入。在编程模式下，需要通过 **RA3** 灌入一个高压源用于编程。此时 **RA3** 的功能变为编程的 **VPP** 端口。

RA2 默认作为 **GPIO** 使用。用户可以通过相关寄存器将 **RA2** 设置为外部中断输入，**TMRO** 外部时钟源输入或者是模拟比较器的输出。

RC 组 **I/O** 默认状态下都为 **GPIO** 模式，软件可以通过使能相应的数字/模拟外设，开启 **RC** 组 **I/O** 的附加功能或者模拟输入 **I/O** 模式。

RC4/5 这两组 **I/O** 具有最高 **80mA** 的推挽驱动能力，一组互补的 **PWM** 输出被分配到 **RC4/5** 上。大电流驱动能力的 **PWM** 输出，可以在高速驱动大输入电容 **MOSFET** 时，仍然具有高品质的 **PWM** 信号。

RC0/3 以及 **RA6/7** 具有最高 **80mA** 的输出灌电流能力，可以用于驱动大功率 **LED**。当驱动功率大于 **100mA** 的 **LED** 时，可以考虑用 2 路或 4 路 **I/O** 并接驱动。

寄存器定义

PORTA – RA 端口数据寄存器

PORTA- RA 端口数据寄存器								
地址: 0x05					XXXX_XXXX			
Bit	7	6	5	4	3	2	1	0
	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	RA7..0	读/写的端口状态						

PORTC – RC 端口数据寄存器

PORTC- RC 端口数据寄存器								
地址: 0x07					XXXX_XXXX			
Bit	-	-	5	4	3	2	1	0
	-	-	RC5	RC4	RC3	RC2	RC1	RC0
R/W	-	-	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:6	-	Unimplemented						
5:0	RC5..0	读/写的端口状态						

TRISA – RA 端口方向控制寄存器

TRISA – RA 端口方向控制寄存器								
地址: 0x85					1111_1111			
Bit	7	6	5	4	3	2	1	0
	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	TRISA7..0	设置端口的输入/输出方向 1: 端口为输入 0: 端口为输出						

TRISC – RC 端口方向控制寄存器

TRISC – RC 端口方向控制寄存器								
地址: 0x87					XX11_1111			
Bit	7	6	5	4	3	2	1	0
	-	-	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
R/W	-	-	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:6	-	Unimplemented						
5:0	TRISC5..0	设置端口的输入/输出方向 1: 端口为输入 0: 端口为输出						

WPDA – RA 端口下拉控制寄存器

WPDA – RA 端口下拉控制寄存器								
地址: N/C					0000_X000			
Bit	7	6	5	4	3	2	1	0
	WPDA7	WPDA6	WPDA5	WPDA4	-	WPDA2	WPDA1	WPDA0
R/W	W/R	W/R	W/R	W/R	-	W/R	W/R	W/R
Bit	Name	描述						
7:0	WPDA7..0	端口弱下拉控制，RA3 不支持下拉电阻。 1: 使能端口下拉 0: 禁止端口下拉 其中 RA1 的下拉电阻需同时使能 IOCA1 位才可有效						

WPDC – RC 端口下拉控制寄存器

WPDC – RC 端口下拉控制寄存器								
地址: N/C					XX00_0000			
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-
Bit	Name	描述						
7:6	-	Unimplemented						
5:0	-	RC 组下拉无效，此寄存器请保持为 0						

WPUA – RA 端口上拉控制寄存器

WPUA – RA 端口上拉控制寄存器								
地址: 0x95					1111_1111			
Bit	7	6	5	4	3	2	1	0
	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	WPUA7..0	端口弱上拉控制 1: 使能端口上拉电阻 0: 禁止端口上拉电阻						

WPUC – RC 端口上拉控制寄存器

WPUC – RC 端口上拉控制寄存器								
地址: 0x95					XX00_0000			
Bit	7	6	5	4	3	2	1	0
	-	-	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
R/W	-	-	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:6	-	Unimplemented						

5:0	WPUC5..0	端口弱上拉控制 1: 使能端口上拉电阻 0: 禁止端口上拉电阻
-----	----------	---------------------------------------

IOCA – RA 端口电平变化中断控制寄存器

IOCR- 端口电平变化中断控制寄存器								
地址: 0x96					0000_0000			
Bit	7	6	5	4	3	2	1	0
	IOCA7	IOCA6	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	IOCA7..0	端口电平变化中断控制位 1: 使能端口电平变化中断 0: 禁止端口电平变化中断						

中断控制

- 软中断(INT 指令支持)
- 外部中断(RA2/INT)
- RA 引脚电平变化中断
- 比较器 1/2 中断
- ADC 转换完成中断
- Timer0/1/2 溢出中断
- 输入俘获中断
- COG 中断
- 外部晶振失效中断

综述

LGT8F684A 最多支持 12 种中断源。其中软中断由 INT 指令触发，执行 INT 指令后，PC 跳转到软中断向量地址(0x001)，并同时关闭全局中断使能位 GIE。其他中断为硬件中断，由不同的硬件模块触发。中断控制寄存器(INTCON)与外设中断请求寄存器(PIR1)记录了每种中断的请求标记。

每个模块的中断功能可以通过中断控制寄存器单独禁用。中断控制器寄存器(INTCON)中的 GIE 为全局中断使能控制位。只有通过设置 GIE 使能了全局中断，MIC8S 内核才会在执行过程中响应来自硬件或者软件的中断请求。

当系统响应了中断请求，将会执行如下的操作：

1. 硬件自动清零全局中断使能位 GIE，禁止响应后续的中断请求；
2. 当前 PC 作为返回地址被压入堆栈寄存器；
3. PC 载入中断对应的向量地址；

中断服务在执行到 RETFIE 指令后退出，RETFIE 指令同时也置位全局中断使能位 GIE。中断服务程序通过查询中断标志确定中断类型。中断状态位必须在中断得到响应后由软件清零，以避免中断被多次响应。

无论是否使能了模块本身的中断使能位或者 GIE 位，中断标记位都会正常工作。使用指令清零 GIE 位后，后续中断请求将不会被响应。后续发生的中断将保持请求状态直到下次 GIE 被使能，或者使用软件强制清除中断标记位。

外设中断控制的细节介绍，请参考本手册外设相关的章节。

中断向量

LGT8F684A 分别为软/硬中断分配了两个中断向量。当有效的中断请求发生后，当前 PC 被压栈，PC 被更新为中断对应的向量地址处开始执行。

LGT8F684A 中断向量分配如下：

中断类型	中断向量地址
软件中断(INT 指令)	0x001
硬件中断(外设请求)	0x004

软件中断具有较高的优先级。软件中断仅当程序执行了 INT 指令时被触发。当 INT 指令进入执行周期时发生了硬件中断，硬件中断将会被暂时忽略。内核首先相应软件中断请求。用户可以在软件中断服务程序中使能 GIE 控制位，以便及时响应其他硬件中断请求。

由于系统仅为硬件中断分配了一个向量地址，因此需要用户在中断服务中通过查询中断标记位区分中断源，并在中断服务中分别进行处理。

下面的代码示例如何使用 LGT8F684A 的中断功能：

		ORG	000h	; reset vector
000h		GOTO	MAIN	; address 0x000, jump to main code
001h	SIRQ	GOTO	SISR	; server routine for software interrupt
		ORG	0x004	; point to vector of hardware interrupt
004h		GOTO	HISR	; server routine for hardware interrupt
005h	MAIN			; main code start here
005h		BCF	OPTION_REG, 6	; interrupt on falling edge of RA2/INT pin
006h		MOVLW	90h	; enable GIE and INTIE
007h		MOVWF	INTCON	; and clear up interrupt flag
008h	LOOP	NOP		; main code
009h		GOTO	LOOP	; main loop
00Ah	SISR	NOP		; ISR of software vector
00Bh		RETFIE		; return from ISR
00Ch	HISR	NOP		; ISR of hardware vector
00Dh		RETFIE		; return from ISR
	END			

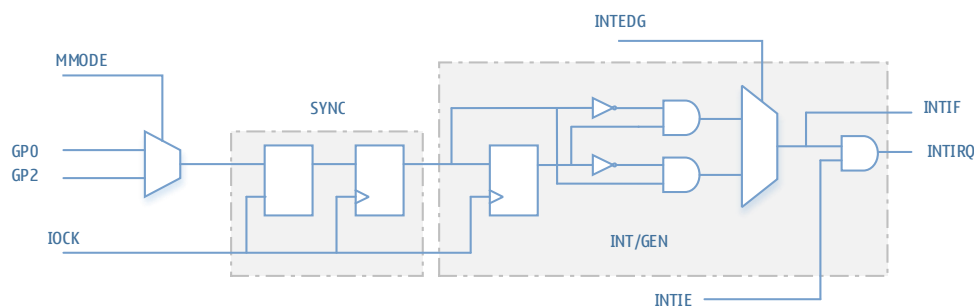
RA2/INT 外部中断

外部中断 **INT** 为边沿触发，触发边沿通过 **OPTION** 寄存器的 **INTEDG** 位设置。当 **RA2** 上发生有效的边沿的变化，**INTCON** 寄存器的 **INTIF** 位被置位。**RA2/INT** 上的中断功能可以通过 **INTCON** 寄存器的 **INTIE** 位禁止。**INTIF** 位必须在再次使能 **GIE** 位之前清除，否则会产生重复的中断。

RA2/INT 可用于休眠唤醒。作为唤醒功能，**RA2/INT** 支持可配置唤醒电平。唤醒电平可以通过 **INTEDG** 位控制，上升沿对应高电平，下降沿对应低电平。**RA2/INT** 可作为掉电模式的唤醒输入，需要使能 **RA2** 的外部中断功能，**RA2** 才可作为唤醒引脚。

当外部中断输入 **INT** 上发生电平变化后，由于 **I/O** 内部同步电路的存在，电平变化会被同步电路后延一个半周期，考虑到进入执行周期的指令不能被中断，因此外部引脚中断的响应时间最大会有 **4** 个指令周期。在最多 **4** 个指令周期后，中断向量代码被取指执行。

下图为外部中断输入同步与中断产生示意图：

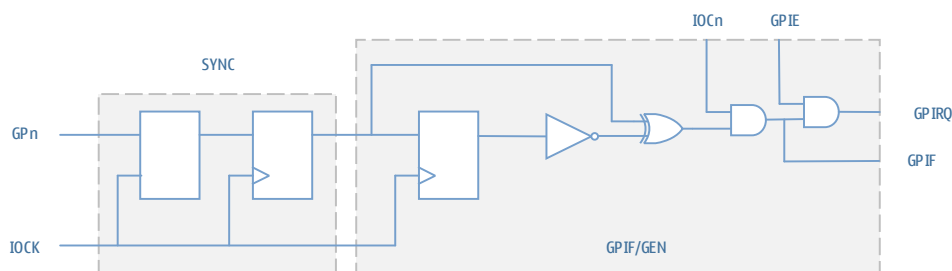


端口电平变化中断

LGT8F684A 的 RA 端口都支持电平变化中断。端口上任何类型的电平变化都会产生此类中断。端口电平变化中断可以通过 INTCON 寄存器的 RAIE 位独立控制。单个引脚的电平变化功能也可以通过 IOCA 寄存器单独控制。

与外部中断输入类似，GPIO 的输入都通过一个前端的同步电路，因此端口电平变化到内核响应该变化的中断请求，也需要最多 4 个指令周期。

端口电平变化中断电路示意图:



定时器 0 溢出中断

定时器 0 计数器(TMRO)溢出(FFh -> 00h)将会置位 INTCON 寄存器的 TOIF 中断标记位。如果此时使能了 TMRO 溢出中断(INTCON:TOIE)，并同时使能了全局中断 GIE，内核将响应该中断请求，执行硬件中断向量。

定时器 0 的溢出中断可以通过置位/清零 **INTCON** 寄存器的 **TOIE** 位实现使能/禁止控制。定时器 0 的相关细节，请查看本手册相关章节。

系统状态保存与恢复

系统响应中断请求后，当前 PC 被压栈保存。一般情况下，中断服务程序会需要使用到某些系统寄存器或者改变一些系统状态。在中断服务中更改这些状态既有可能会破坏主程序的执行环境，因此需要在执行中断服务前保存一些必要的系统状态(比如，W 以及 STATUS 寄存器)。这部分操作只能通过软件实现。

考虑到 **LGT8F684A** 为分页寻址，为便于数据保存和恢复，并且避免保存操作本身对统状态的影响(比如应该避免更改页寻址位)，应该将数据保存在地址被等同映射到每个页面的存储空间。**LGT8F684A** 将 128 字节的 **SRAM** 映射到 **0x20~0x7F** 以及 **0xA0~0xBF** 的区域，其中 **0x70~0x7F** 区域映射到第二页的 **0xF0~0xFF** 区域，非常适合保存临时数据。为避免和主程序应用分配的临时空间冲突，建议使用 **RAM** 空间 **0x70~0x7F** 作为保存系统状态的临时空间。

需要在中断服务中保存哪些数据，主要与中断服务程序的实现有关，用户需要详细评估中断服务可能会影响的系统状态。如果这些状态会影响到主程序的运行环境，必须在中断服务中保存，并在中断返回之前恢复。下面的程序片段示例如何在中断服务中保存和恢复系统状态：

W_TEMP	EQU	7Fh	; define W_TEMP register
STATUS_TEMP	EQU	7Eh	; define STATUS_TEMP register
	ORG	04h	; point to interrupt vector address
HISR	MOVWF	W_TEMP	; copy W to TEMP register
	SWAPF	STATUS, W	; swap STATUS to W

```

; using SWAP to avoid status affected
MOVWF STATUS_TEMP ; save STATUS to TEMP register
..... ;
(ISR) ; Insert user code here
..... ;
SWAPF STATUS_REG, W ; swap STATUS_TEMP into W
MOVWF STATUS ; restore STATUS register
SWAPF W_TEMP, F ; swap W_TEMP
SWAPF W_TEMP, W ; restore W register
RETFIE ; ISR return

```

寄存器定义

INTCON – 中断控制寄存器

INTCON/INTEN- 中断控制寄存器								
地址: 0x0B/0x8B					0000_0000			
Bit	7	6	5	4	3	2	1	0
	GIE	PEIE	TOIE	INTIE	RAIE	TOIF	INTIF	RAIF
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	GIE	全局中断使能控制位						
6	PEIE	外设中断使能控制 用于全局控制系统中除 TOIE/INTIE/RAIE 之外的其他外设中断, 只有在 PEIE 与 GIE 同时有效的前提下, 设置 CGIE, ADIE, CCP1IE, C2IE, C1IE, OSFIE, TMR2IE 以及 TMR1IE 才能够使能对应的中断请求功能						
5	TOIE	定时器 0 溢出中断使能控制位						
4	INTIE	外部中断使能控制位						
3	RAIE	RA 引脚电平变化中断使能控制位						
2	TOIF	定时器 0 溢出中断标记位						
1	INTIF	外部中断标记位						
0	RAIF	RA 引脚电平变化中断标记位						

PIE1 – 外设中断控制寄存器

PIE1 – 外设中断控制寄存器								
地址: 0x8C					0000_0000			
Bit	7	6	5	4	3	2	1	0
	CGIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	CGIE	COG 中断使能位						
6	ADIE	ADC 中断使能位						
5	CCP1IE	输入俘获中断使能位						
4	C2IE	比较器 2 中断使能位						

3	C1IE	比较器 1 中断使能位
2	OSFIE	外部晶振失效中断使能位
1	TMR2IE	定时器 2 中断使能位
0	TMR1IE	定时器 1 中断使能位

PIR1 – 外设中断标记寄存器

PIR1 – 外设中断标记寄存器								
地址: 0x0C					0000_0000			
Bit	7	6	5	4	3	2	1	0
	CGIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	CGIF	COG 模块中断标志位						
6	ADIF	ADC 转换完成中断标志位						
5	CCP1IF	输入俘获中断标志位						
4	C2IF	比较器 2 中断标志位						
3	C1IF	比较器 1 中断标志位						
2	OSFIF	外部晶振失效中断标志位						
1	TMR2IF	定时器 2 溢出中断标志位						
0	TMR1IF	定时器 1 溢出中断标志位						

看门狗定时器

- 时钟来自内部 32KHz LFINTOSC
- 独立 16 位预分频器
- 8 位预分频器(与 TMR0 共享)
- 1ms~268s 超时周期
- 灵活的软件/配置位控制
- 定时唤醒支持

综述

看门狗定时器由一个片内集成的独立 32KHz RC 振荡器驱动，不要任何外围器件的辅助即可工作。即使在休眠模式下，看门狗定时器仍然可以工作。在正常模式下，看门狗定时器溢出将会产生系统复位。而在休眠模式下，看门狗溢出可将系统从休眠模式中唤醒，系统被唤醒后从休眠之前的状态继续运行，不会产生系统复位。

在默认状态下，看门狗定时器是被使能的。用户可以通过配置字或者 WDTCON 寄存器的 SWDTEN 位进行控制。即使在配置字禁用 WDT 的情况下，仍然可以通过 WDTCON 的 SWDTEN 寄存器将其使能。

看门狗定时器正常模式下超时时间为 18MS。超时时间与温度以及系统供电电压也有关系。WDT 超时时间与电压变化的关系，请参考本手册电气特性部分。

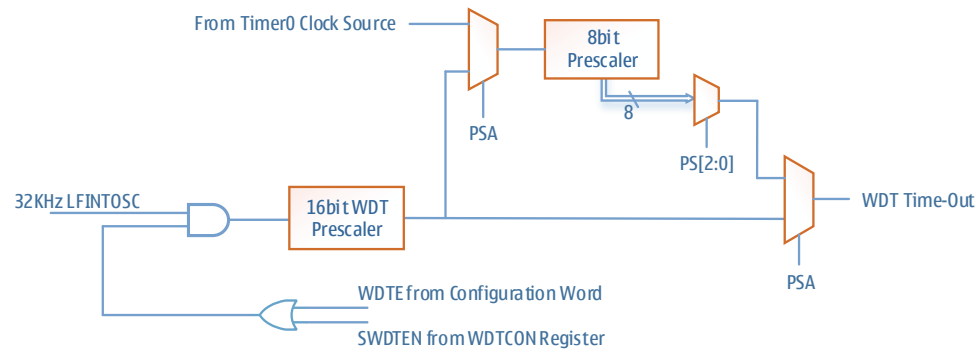
看门狗定时器与 TMR0 共享一个 8 位预分频器。使用预分频配置，最大可以产生正常模式 128 倍的超时时间。预分频的配置通过 OPTION 寄存器的 PS 位设置。OPTION 寄存器的定义请参考本手册 MIC8S 内核部分的介绍。除了与 TMR0 共享的 8 位预分频器，WDT 还有一个独立的 16 位预分频器以及一个 4 位的预分频选择，可以实现最多 12 级最大 65536 倍的超时时间。详细配置请参考 WDTCON 寄存器定义部分。

软件执行 CLRWDT 或者 SLEEP 指令可以清零 WDT 定时器以及预分频器(如果预分频器被分配给 WDT)。清零 WDT 完成喂狗操作。WDT 超时后，STATUS 寄存器中的 T0 位被清零。软件可以通过 STATUS 寄存器的 T0 /PD 位判定系统复位是否来自 WDT 超时。

Conditions	WDT STATUS
WDTE=0	Cleared
CLRWDT issued	
SLEEP issued	
Oscillator Failed Detected	
Wakeup + System Clock = CLKIN/RCM/RCK	
Wakeup + System Clock = HFSOC/LFOSC	Cleared until the end of OST

Note: 当系统工作在外部晶振模式，OST 定时器被使能。在 OST 计时期间，WDT 模块保持在复位状态，当 OST 超时后，WDT 才开始启动计数。

看门狗定时器结构图：



寄存器定义

OPTION_REG – 外设配置寄存器

OPTION_REG – 外设配置寄存器								
地址: 0x81					初始值: 1111_1111			
Bit	7	6	5	4	3	2	1	0
	RAPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	RAPU	RA 组 I/O 全局上拉禁止控制，低有效						
6	INTEDG	外部中断/唤醒边沿选择						
5	TOCS	Timer0 时钟源选择						
4	TOSE	Timer0 时钟触发沿选择						
3	PSA	预分频器分配控制位 1: 预分频为 WDT 所有 0: 预分频为 Timer0 所有						
2:0	PS[2:0]	预分频频选择位						
		PS[2:0]	Timer0 Rate		WDT Rate			
		000	1:2		1:1			
		001	1:4		1:2			
		010	1:8		1:4			
		011	1:16		1:8			
		100	1:32		1:16			
		101	1:64		1:32			
		110	1:128		1:64			
		111	1:256		1:128			

WDTCON – WDT 控制寄存器

WDTCON – WDT 控制寄存器								
地址: 0x18					初始值: XXX0_1000			
Bit	7	6	5	4	3	2	1	0
	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
R/W	-	-	-	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:5	-	Unimplemented						
4:1	WDTPS	WDT 独立预分频周期选择位						
		WDTPS[3:0]		预分频因子				
		0000		1:32				
		0001		1:64				
		0010		1:128				
		0011		1:256				
		0100		1:512 (默认设置)				
		0101		1:1024				
		0110		1:2048				
		0111		1:4096				
		1000		1:8192				
		1001		1:16384				
		1010		1:32768				
		1011		1:65536				
		1100		Reserved				
		1101		Reserved				
		1110		Reserved				
		1111		Reserved				
0	SWDTEN	WDT 软件使能/禁止控制位 如果系统配置字的 WDTE=1, WDT 将时钟被使能, 不受 SWDTEN 的控制 当系统配置字 WDTE=0 是, 软件可以通过 SWDTEN 使能或者关闭 WDT						

定时计数器 0

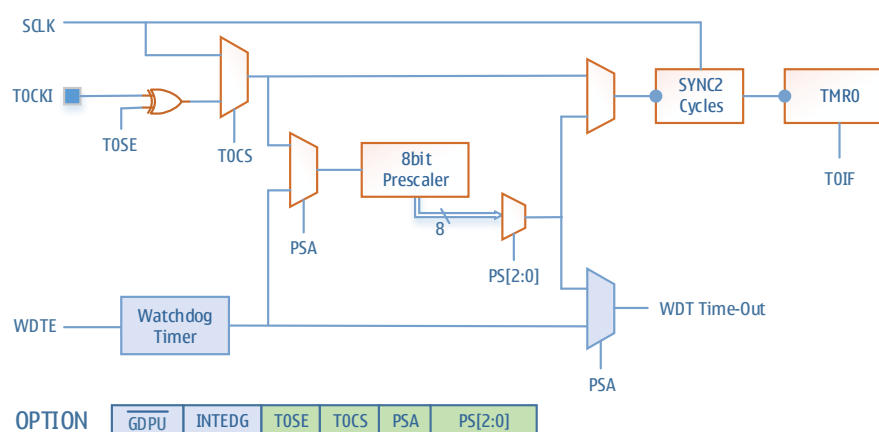
- 8 位定时/计数寄存器(TMR0)
- 8 位预分频器(与 WDT 共享)
- 可编程内部/外部时钟源
- 可编程外部时钟边沿选择
- 溢出中断

综述

TMR0 模块支持两种工作模式：8 位定时器/8 位计数器。

TMR0 计数溢出时产生溢出中断，软件通过 INTCON 寄存器的 TOIF 位查询 TMR0 的中断标志。因为在休眠模式下 TMR0 的接口时钟被关闭，无法产生溢出中断，因此无法支持休眠唤醒。

TMR0 框架图：



定时器模式

当工作于定时器模式，TMR0 的定时器在每个指令周期递增(无预分频)，定时器模式通过清零 OPTION 寄存器的 TOMS 位使能。当对 TMR0 寄存器进行写操作，在写操作发生的接下来两个周期内，TMR0 寄存器被禁止递增。如果需要补充写周期后的定时误差，可以在写 TMR0 之前对将要写的值继续调整。

计数器模式

将 OPTION 寄存器的 TOMS 位置 1，TMR0 工作于计数器模式。在计数器模式下，TMR0 寄存器在 TOCKI 的边沿驱动下递增计数。TOCKI 的边沿通过 OPTION 寄存器的 TOSE 位设置。

可编程预分频器

8 位可编程预分频器为 TMR0 和 WDT 共享。同一时刻预分频器只能被分配给两者之一使用。当 OPTION 寄存器的 PSA 被设置为 1，预分配器被分配给 WDT；否则分配给 TMR0 使用。由于 PSA 位在系统复位后默认为 1，所以预分频器在系统默认的状态下，是被分配给 WDT 使用。

对于 TMR0 模式而言，预分频器支持 8 级分频配置。比例从 1:2 到最大 1:256。分频系数通过 OPTION 寄存器的 PS[2:0]配置。当预分频器被分配给 WDT 时，TMR0 的预分配系数为默认 1:1。

预分频器参数不能够直接的读写访问。当分配给 **TMRO** 后，任何对 **TMRO** 寄存器的写操作都会同时复位预分频器。当预分频器被分配给 **WDT** 后，**CLRWDT** 指令将会同时清零 **WDT** 与预分频器。

因为预分频器可以被分配给 **TMRO** 或者 **WDT**，因此在改变预分频配置时，可能会产生非正常的系统复位。下面的程序代码示例如何在 **WDT** 与 **TMRO** 之前正确安全的切换预分频器。

预分频器从 **TMRO** 切换至 **WDT**:

```

BANKSEL    TMRO           ; select bank 0
CLRWDT           ; Clear WDT
CLRF        TMRO          ; clear WDT and prescaler

BANKSEL    OPTION_REG     ; select bank 1
BSF        OPTION_REG, PSA ; assign prescaler to PSA
CLRWDT           ; clear WDT again

MOVLW      b'1111_1000    ; mask prescaler"
ANDWF      OPTION_REG, W   ; bits
IORLW      b'0000_0101    ; set WDT prescaler
MOVWF      OPTION_REG,    ;

```

预分频器从 **WDT** 切换至 **TMRO**:

```

CLRWDT           ; clear WDT and prescaler

BANKSEL    OPTION_REG     ;
MOVLW      b'1111_0000    ; mask TMRO select and
ANDWF      OPTION_REG, W   ; prescaler bits
IORLW      b'0000_0011    ; set perscaler to 1:16
MOVWF      OPTION_REG     ;

```

TMRO 中断处理

TMRO 在计数溢出(FFh 到 00h)时，**TOIF** 为在溢出中断发生时。无论系统有没有使能 **TOIE**，此中断标记位仍然会被置位为 **1**。在休眠模式下，**TMRO** 工作时钟被关闭，无法产生用于休眠唤醒中断的信号。

TMRO 的使能与禁止

TMRO 默认状态是被使能的。但默认的 **OPTION_REG:TOCS=1** 将 **TMRO** 的时钟源选择为来自外部的 **TOCKI** 引脚。因此如果应用不需要 **TMRO**，并且 **TOCKI** 也没有特定的时钟信号，**TMRO** 的计数器也等同于停止运行状态，此时不关闭 **TMRO** 也不会带来明显的多余功耗。但如果 **TOCKI** 有比较高频的信号变化，没有关闭的 **TMRO** 将会带来可观的功耗损失。此时可以通过置位 **PCON** 寄存器的 **TMROD** 位禁止 **TMRO**。

定时计数器 1

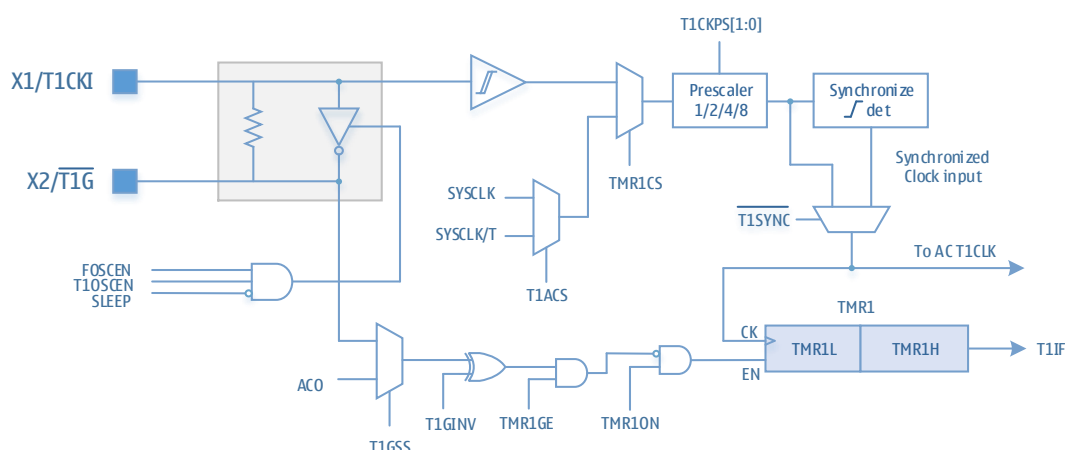
- 16 位定时/计数寄存器(TMR1H:TMR1L)
- 可编程内部或者外部时钟源
- 3 位独立预分频
- 可选外部低频晶振输入(LFOSC)
- 同步/异步模式
- TMR1 计数门控支持：比较器或/T1G 引脚
- 计数溢出中断
- 溢出唤醒(外部时钟模式或异步模式)
- 比较器输出与 TMR1 时钟同步

综述

TMR1 为一个 16 位的递增计数器，计数寄存器可以通过 TMR1H:TMR1L 访问。写 TMR1H:TMR1L 将直接更新 TMR1 计数寄存器。

当使用内部时钟作为 TMR1 时钟源，TMR1 工作于定时器模式；当配置为外部时钟源时，TMR1 可以工作作为定时器或者计数器模式。

TMR1 模块结构图：



说明：

1. SYSCLK 为系统时钟，SYSCLK/T 为指令周期。指令周期与系统时钟的关系通过配置字 TCYC 设置，默认为 4T 配置，此时指令周期为系统时钟的 1/4。详细介绍请参考本手册系统配置位相关部分。

时钟源

TMR1 的时钟源通过 T1CON 寄存器的 TMR1CS 位选择。TMR1 的时钟源配置请查考下面表格：

Clock Source	TMR1CS	T1ACS
SYSCLK/T	0	0
SYCLK	0	1
T1CKI pin	1	X

内部时钟源

当 TMR1 选择了内部时钟源，TMR1 计数寄存器在时钟的上升沿递增。计数器递增的频率由内部时钟频率以及预分频器设置共同决定。

外部时钟源

当选择外部时钟作为 TMR1 时钟源，TMR1 可工作于定时器或计数器模式。

作为定时器，TMR1 在外部时钟输入 T1CKI 的上升沿递增计数。计数模式时钟可配置为同步于系统时钟或者独立于系统时钟异步运行。

对于需要特殊频率的应用，比如常用的 32.768KHz 外部晶振，可以选择外部晶振作为 TMR1 的外部时钟源。通过 T1CON 寄存器的 T1OSCEN 位可以使能外部 LFOSC 作为 TMR1 的外部时钟输入。

需要注意，T1OSCEN 模式只在系统时钟模式为 LFOSC 或者内部 RCM/RCK 时工作。当系统时钟配置为 HFOSC/CLKIN 模式时，应避免使用 T1OSCEN 模式。当使能了外部晶振，在上电复位或者从休眠模式唤醒时，都需要考虑到 OST 晶振启动时间延时。

预分频器

TMR1 有一个 2 位的预分频器，可以实现对计数时钟的 1/2/4/8 分频。T1CON 寄存器的 T1CKPS 位用于设置预分频系数。预分计数器本身不能够被直接访问。更新 TMR1 计数寄存器(TMR1H/TMR1L)时会清零预分频计数器。

异步计数模式

当 T1CON 寄存器的/T1SYNC 为被置 1，外部时钟输入处于非同步模式。定时器的计数与内部系统时钟异步进行。此时 TMR1 可以在休眠模式下仍然继续运行。因此不同于 TMR0，TMR1 的溢出中断可用于唤醒休眠模式。

因此 TMR1 计数器工作于异步模式，在读写 TMR1 计数寄存器时，需要特别注意：

首先，在异步模式下访问 TMR1 寄存器有专属的同步逻辑以保证访问的完整有效。但必须考虑到，16 位的 TMR1 计数寄存器由两个 8 位的寄存器组合而成，而读取才做也必须分两次操作，因此有可能在读取高低字节的间隔中，计数器发生溢出，此时读到的数据将不能反映当前计数器的状态。

对于写操作，建议用户在写 TMR1 计数寄存器之前将 TMR1 关闭。写操作与计数器的递增操作可能会导致最终写入不可预知的结果。

门控模式

TMR1 的门控输入可选为来自引脚/T1G 或者模拟比较器的输出。门控模式实现用 TMR1 测量外部事件。外部事件可以是来自引脚/T1G 或者是来自模拟比较器反映的电平变化。可以利用 TMR1 的这个功能实现简单的 Delta-Sigma ADC 转换以及其他多种模拟量测试应用。

TMR1 的门控输入的极性可以通过 T1CON 寄存器的 T1GINV 为设置为反向，用于实现测量高有效或者低有效的信号变化。

TMR1 门控模式的应用，请参 LGT8F684A 相关应用手册。

溢出中断

当 TMR1 计数寄存器递增至 FFFFh，并从 FFFFh 回滚到 0000h 时，TMR1 溢出中断标志位 PIR1:T1IF 被

置位。当系统同时也使能了 TMR1 中断控制位(PIE1)以及全局中断控制位(GIE)，内核将相应 TMR1 的溢出中断请求，PC 跳转至硬件中断向量。

TMR1 溢出中断在中断服务程序中通过清理 T1IF 位实现。同时 TMR1 计数寄存器也应该在下次使能中断之前清零。

TMR1 支持在休眠模式工作。此时需要将 TMR1 设置为异步计数模式。在这种模式下，TMR1 的时钟源来自于外部晶振(LFOSC)或者外部时钟输入(CLKIN)。TMR1 支持唤醒需要使能溢出中断。当计数器发生溢出时，将系统从休眠模式唤醒，继续执行指令。如果使能了全局中断 GIE，内核将会相应溢出中断请求。

比较器输出同步

TMR1 计数器时钟源可用于同步比较器输出。此功能通过比较器的 CMCON1 寄存器 CMSYNC 为使能。

当使用比较器输出作为 TMR1 的门控输入，比较器需要与 TMR1 同步，这样可以确保 TMR1 能够完整的记录比较器的输出变化。关于此部分的更多信息，请参考比较器相关章节。

寄存器定义

T1CON – 定时器 1 控制寄存器

T1CON- 定时器 1 控制寄存器								
地址: 0x10					0000_0000			
Bit	7	6	5	4	3	2	1	0
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	T1GINV	TMR1 门控信号极性反向控制 1: TMR1 门控高有效 0: TMR1 门控低有效						
6	TMR1GE	TMR1 门控使能，当 TMR1ON 为 1 时，此为设置有效 1: TMR1 在门控有效时工作 0: TMR1 的工作不受门控信号控制						
5:4	T1CKPS	TMR1 输入时钟预分频选择位 11: 1/8 分频 10: 1/4 分频 01: 1/2 分频 00: 1/1 分频						
3	T1OSCEN	LFOSC 使能位 当系统时钟为 LFOSC 模式或内部 RCM/RCK 模式，此为设置为 1 将外部晶振作为 TMR1 的计数时钟						
2	T1SYNC	TMR1 外部输入时钟同步控制位。仅当 TMR1 时钟源为外部时钟有效 1: 异步时钟模式 0: 同步时钟模式						
1	TMR1CS	TMR1 时钟源选择位 1: 时钟输入来自外部 T1CKI						

		0: 时钟来自内部 SYSCLK 或 SYSCLK/T, 参考 CMCON1 T1ACS 位
0	TMR1ON	TMR1 使能控制 1: TMR1 使能 0: TMR1 停止工作

TMR1H – TMR1 计数寄存器高字节

TMR1H – TMR1 计数寄存器高字节								
地址: 0x0F					0000_0000			
Bit	7	6	5	4	3	2	1	0
TMR1H	TMR1H[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	TMR1H	TMR1 计数寄存器高字节						

TMR1L – TMR1 计数寄存器低字节

TMR1L – TMR1 计数寄存器低字节								
地址: 0x0E					0000_0000			
Bit	7	6	5	4	3	2	1	0
TMR1L	TMR1L[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	TMR1L	TMR1 计数寄存器低字节						

TCCR – 定时器时钟控制寄存器

TCCR – 定时器时钟控制寄存器								
地址: 0x1D					0X00_00XX			
Bit	7	6	5	4	3	2	1	0
TCCR	X2EN	-	T2CF	T2CS	T1CF	T1CS	-	-
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	X2EN	倍频时钟使能控制, 1=使能倍频时钟						
6	-	Unimplemented						
5	T2CF	定时器 2 倍频模式标志位, 为 1 表示定时器 2 倍频设置有效						
4	T2CS	定时器 2 倍频时钟使能位。1=设置定时器 2 工作于倍频模式						
3	T1CF	定时器 1 倍频模式标志位, 为 1 表示定时器 1 倍频设置有效						
2	T1CS	定时器 1 倍频模式使能位。写 1 设置定时器 1 工作于倍频模式						
1:0	-	Unimplemented						

PR1LR – Timer1 定时周期寄存器低 8 位

PR1LR – Timer1 定时周期寄存器低 8 位								
地址：0x08					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	PR1LR							
R/W	R/W							
Bit	Name	描述						
7:0	PR1LR	Timer1 定时周期寄存器低 8 位						

PR1HR – Timer1 定时周期寄存器高 8 位

PR1HR – Timer1 定时周期寄存器高 8 位								
地址：0x09					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	PR1HR							
R/W	R/W							
Bit	Name	描述						
7:0	PR1HR	Timer1 定时周期寄存器高 8 位						

DC1LR – Timer1 占空比寄存器低 8 位

DC1LR – Timer1 占空比寄存器低 8 位								
地址：0x06					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	DC1LR							
R/W	R/W							
Bit	Name	描述						
7:0	DC1LR	Timer1 占空比寄存器低 8 位						

DC1HR – Timer1 占空比寄存器高 8 位

DC1HR – Timer1 占空比寄存器高 8 位								
地址：0x0D					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	DC1HR							
R/W	R/W							
Bit	Name	描述						
7:0	DC1HR	Timer1 占空比寄存器高 8 位						

定时计数器 2

- 8 位定时/计数寄存器(TMR2)
- 8 位定时比较寄存器(PR2)
- 可编程预分频器(1:1, 1:4, 1:16)
- 可编程后分频器(1:1 to 1:16)
- TMR2 与 PR2 比较匹配中断

综述

TMR2 的计数时钟来自执行指令时钟。计数时钟输入到 TMR2 内部的预分频器。预分频器支持 1/1, 1/4 以及 1/16 分频配置。预分频的输出时钟用于驱动 TMR2 计数。

TMR2 计数值与 PR2 的值实时进行比较。当 TMR2 计数从 00 逐次增加到与 PR2 匹配后, 系统将发生一下两种状态更新:

- TMR2 计数值被清零, 重新开始下一次递增周期
- TMR2 后分频器递增

TMR2 与 PR2 比较匹配的事件输出给 TMR2 的后分频器。后分频器分频系数从 1:1 到 1:16 可选。TMR2 后分频器的输出用于产生 T2IF 中断标志。软件可通过 PIR1 寄存器访问到 T2IF 标志位。

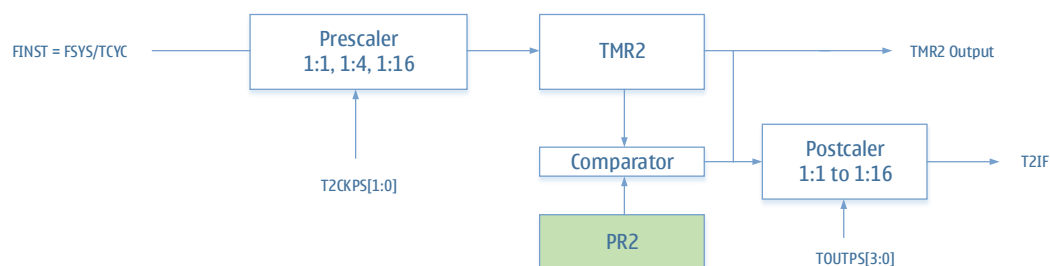
TMR2 寄存器与 PR2 寄存器支持软件读写访问。在发生任何种类的复位后, TMR2 寄存器被复位到 00, PR2 寄存器被复位为 0xFF。

TMR2 通过设置 T2CON 寄存器的 TMR2ON=1 使能; 清零 TMR2ON 将禁止 TMR2 模块的任何功能。

TMR2 预分频器由 T2CON 寄存器的 T2CKPS 位控制。TMR2 的后分频系数由 T2CON 寄存器的 TOUTPS 位控制。预分频以及后分频器在以下情况发生时, 将清零分频计数器:

- 写 TMR2 寄存器
- 写 T2CON 寄存器
- 任何种类的复位(上电复位, 外部复位, 看门狗复位以及 LVD 低电压复位)

TMR2 框架图:



寄存器定义

T2CON - 定时器2 控制寄存器

T2CON- 定时器2 控制寄存器								
地址: 0x12					X000_0000			
Bit	7	6	5	4	3	2	1	0
	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	-	Unimplemented						
6:3	TOUTPS	TMR2 输出后分频系数选择位; TOUTPS[3:0] = 0000 : 1/1 0001 : 1/2 0010 : 1/3 1101 : 1/14 1110 : 1/15 1111 : 1/16						
2	TMR2ON	TMR2 使能控制位; 1 = 使能, 0 = 禁止						
1:0	T2CKPS	TMR2 计数时钟预分频器系数选择位; T2CKPS[1:0] =						
		T2CKPS	指令时钟计数模式			高速时钟计数模式		
		00	1/1			不可用		
		01	1/4			1/1		
		1X	1/16			1/4		

TCCR - 定时器时钟控制寄存器

TCCR- 定时器时钟控制寄存器								
地址: 0x1D					0X00_00XX			
Bit	7	6	5	4	3	2	1	0
TCCR	X2EN	-	T2CF	T2CS	T1CF	T1CS	-	-
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	X2EN	倍频时钟使能控制, 1=使能倍频时钟						
6	-	Unimplemented						
5	T2CF	定时器2 倍频模式标志位, 为 1 表示定时器2 倍频设置有效						
4	T2CS	定时器2 倍频时钟使能位。1=设置定时器2 工作于倍频模式						
3	T1CF	定时器1 倍频模式标志位, 为 1 表示定时器1 倍频设置有效						
2	T1CS	定时器1 倍频模式使能位。写 1 设置定时器1 工作于倍频模式						
1:0	-	Unimplemented						

增强型比较/俘获/PWM 模块(ECCP)

- 俘获来自引脚的外部事件
- 比较匹配触发外部事件
- 产生频率和占空比可调的 PWM 信号
- PWM 增强模式支持死区控制的半桥/全桥驱动

综述

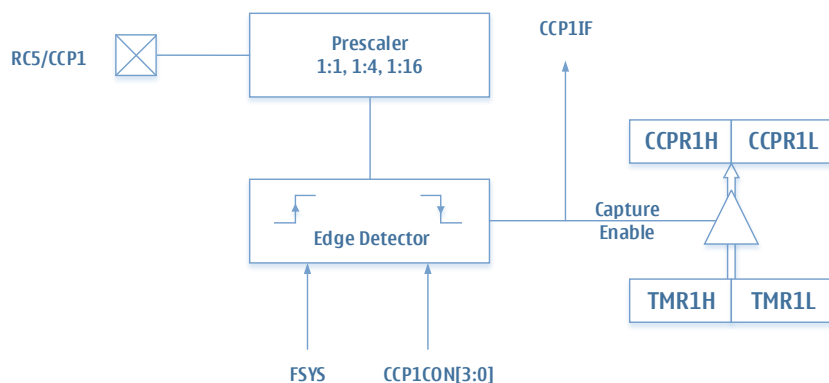
增强型捕获、比较及脉冲宽度调制模块（简称为 ECCP 模块）是一个用来计时和控制各种事件的外设。**Capture** 模式下，可以对外部事件进行计时。**Compare** 模式下，当预设的计时器溢出时可以触发外部事件。**PWM** 模式下，可以产生频率和占空比可调的 PWM 信号。

Capture 模式

Capture 模式下，当 CCP1 引脚上的事件发生时，定时器 1 的 16 位计数值 TMR1 会被捕获到 CCPRH:CCPRL 中去。该事件为以下四种当中的一种，由 CCPCON 寄存器中的 CCPM[3:0] 位来选择，包括：

- 任一下降沿
- 任一上升沿
- 第 4 个上升沿
- 第 16 个上升沿

当捕获产生后，位于 PIR1 寄存器中的相应中断请求标志位 CCIF 被置位。该标志位必须由软件来清零。在 CCPRH 和 CCPRL 寄存器的值被读走之前，如果发生新的捕获，原来的捕获值讲会被新的捕获值覆盖掉。在俘获模式下，须设置其相应的 TRIS 控制位来配置 CCP1 引脚作为输入。俘获模式工作原理如下图所示：



定时器 1 模式选择

Capture 模式下，定时器 1 须工作在定时模式或同步计数模式。异步计数模式下，捕获可能工作不正常。

软件中断

当 Capture 模式发生改变时，可能会产生一次假的捕获中断。用户需要保持位于 PIE1 寄存器的 CCP1IE 位为零来避免此假中断。此外，还需要在改变 Capture 模式后清零位于 PIR1 寄存器的 CCP1IF 中断标志位。

CCP 预分频器

Capture 模式下，位于 CCP1CON 寄存器的 CCP1M[3:0]设定了 4 种预分频系数。当 CCP 模块被关掉，或者没有工作在 Capture 模式下，此预分频器被清零。任一复位也将清零此预分频器。当切换 Capture 预分频系数时，预分频器不会被清零，因此可能会产生假的中断。为避免这种非预期的操作，在改变预分频系数前，可通过清零 CCP1CON 寄存器来关掉 CCP 模块。

Compare 模式

在 Compare 模式下，16 位的 CCPR1 寄存器会持续和 TMR1 寄存器进行比较，当比较匹配发生时，CCP 模块会：

- 翻转 CCP1 输出
- 置位 CCP1 输出
- 清零 CCP1 输出
- 产生一个特定的事件触发
- 产生一个软件中断

这些操作是由位于 CCP1CON 寄存器的 CCP1M[3:0]位来选择控制的。所有比较模式都会产生一个中断。用户须清零相应的 TRIS 位来配置 CCP1 引脚为输出引脚。

定时器 1 模式选择

在比较模式下，定时器 1 须工作在定时模式或同步计数模式。异步计数模式下，比较可能工作不正常。

软件中断模式

当选择产生软件中断模式时 (CCP1M[3:0]=1010b)，CCP 模块失去对 CCP1 引脚的控制（见 CCP1CON 寄存器）。

特殊事件触发器

当选择特殊事件触发模式时 (CCP1M[3:0]=1011b)，CCP 模块将会：

- 复位定时器 1
- ADC 使能时启动一次 ADC 转换

CCP 模块失去对 CCP1 引脚的控制（见 CCP1CON 寄存器）。

当 TMR1H, TMR1L 寄存器对与 CCPR1H, CCPR1L 寄存器对发生比较匹配时，CCP 模块的特殊事件触发输出会立即产生。TMR1H, TMR1L 寄存器对会在接下来的定时器 1 时钟上升沿发生复位。这就允许 CCPR1H, CCPR1L 寄存器对为定时器 1 有效提供 16 位可编程周期寄存器。

PWM 模式

PWM 模式可在 CCP1 引脚上产生一个脉冲宽度调制信号。占空比、周期和分辨率由以下寄存器来决定：

- PR2
- T2CON
- CCPR1L
- CCP1CON

PWM 模式下，CCP 模块可在 CCP1 引脚上产生高达 10 位分辨率的 PWM 输出。CCP1 引脚与 PORT 数据锁存是复用的，该引脚的 TRIS 位必须被清零来使能 CCP1 引脚输出驱动。需要注意的是，清零 CCP1CON 寄存器将会放弃 CCP1 输出对 CCP1 引脚的控制。

PWM 周期

PWM 周期由定时器 2 的 PR2 寄存器决定。PWM 周期由下面的公式来计算：

$$\text{PWM Period} = [(PR2) + 1] * 4 * T_{osc} * (\text{TMR2 Prescale Value})$$

当 TMR2 等于 PR2 时，下一个计数周期会发生以下三件事：

- TMR2 被清零
- CCP1 引脚被置位（例外：若 PWM 占空比为 0%，引脚将不会被置位）
- PWM 占空比值由 CCPR1L 锁存到 CCP1H

PWM 占空比

PWM 占空比由写入多个寄存器的 10 位值来决定：CCPR1L 寄存器和 CCP1CON 寄存的 CCP1[1:0] 位。CCPR1L 为高八位，CCP1[1:0] 为低两位。CCPR1L 和 CCP1[1:0] 任何时刻均可写入。直到当前周期结束时（即 PR2 和 TMR2 发生匹配时），CCPR1L 才会被锁存到 CCP1H。

PWM 脉冲宽度由下面的公式来计算：

$$\text{Pulse Width} = (\text{CCPR1L:CCP1[1:0]}) * T_{osc} * (\text{TMR2 Prescale Value})$$

PWM 占空比由下面的公式来计算：

$$\text{Duty Cycle Ratio} = (\text{CCPR1L:CCP1[1:0]}) / ((PR2+1)*4)$$

CCPR1H 用来作为 PWM 占空比的双缓冲寄存器。双缓冲可以保证 PWM 无缝操作。

定时器 2 的 8 位寄存器 TMR2，和 2 位内部系统时钟计数器，或 2 位预分频器，一起组成 10 位的时基。当定时器 2 的预分频器设置为 1:1 时，即采用 2 位系统时钟计数器作为低两位。当 10 位时基与 CCPR1H 和 CCP1[1:0] 发生匹配时，CCP1 引脚被清零。

PWM 分辨率

对于给定的周期，分辨率决定了可能的占空比周期数。10 位分辨率有 1024 个连续的占空比周期，8 位分辨率有 256 个占空比周期。

当 PR2 被设置为 255 时，拥有最大的 PWM 分辨率，即 10 位。分辨率是 PR2 寄存器的函数，如下面公式所示：

$$\text{Resolution} = \log[4 * (PR2 + 1)] / \log_2 \text{ bits}$$

下面两个表格为 PWM 频率和占空比的示例。

PWM 频率和分辨率 (Fosc = 20 MHz)

Frequency (KHz)	1.22	4.88	19.53	78.13	156.25	312.5
TMR2 Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x0F
Max Resolution	10	10	10	8	7	6

PWM 频率和分辨率 (Fosc = 8 MHz)

Frequency (KHz)	1.23	4.90	19.61	76.92	153.85	333.3
TMR2 Prescale	16	4	1	1	1	1
PR2 Value	0x65	0x65	0x65	0x19	0x0C	0x05
Max Resolution	8	8	8	6	5	4

休眠模式下的操作

睡眠模式下，定时器 2 的 TMR2 寄存器不会增加，状态也不会改变。若 CCP1 引脚已驱动，它将会继续保持该状态。当芯片被唤醒后，TMR2 会从之前的状态继续。

复位的影响

任一复位将会把所有端口强制为输入模式，CCP 模块的寄存器回到复位值。

PWM 操作的配置

当配置 CCP 模块工作在 PWM 模式时，可按照以下流程来执行：

1. 置位相应的 TRIS 位来禁止 PWM 引脚（CCP1）的输出驱动；
2. 加载 PR2 寄存器来设置 PWM 周期；
3. 加载 CCP1CON 寄存器为适合的值来设置 CCP 模块为 PWM 模式；
4. 加载 CCP1L 寄存器和 CCP1CON 寄存器的 CCP1 位来设置 PWM 占空比；
5. 配置和启动定时器 2，包括：
 - 清零位于 PIR1 寄存器的 TMR2IF 中断标志位
 - 加载位于 T2CON 寄存器的 T2CKPS 位来设置定时器 2 的预分频系数
 - 置位位于 T2CON 寄存器的 TMR2ON 位来使能定时器 2
6. 一个新的 PWM 周期开始后，使能 PWM 输出，即：
 - 等待定时器 2 溢出（即位于 PIR1 寄存器的 TMR2IF 位置位）
 - 清零相应的 TRIS 位来使能 CCP1 引脚的输出驱动

PWM 增强模式

PWM 增强模式可产生一路高达 10 位分辨率的 PWM 信号输出至 4 个不同的引脚上，可通过下面四种不同的 PWM 输出模式来实现：

- 单路 PWM
- 半桥 PWM
- 全桥 PWM，正向模式
- 全桥 PWM，反向模式

要选择 PWM 增强模式，位于 CCP1CON 寄存器的 PWM1M 位必须正确设置。

PWM 输出是和 IO 引脚复用的，并指定在 P1A，P1B，P1C 和 P1D 引脚上。PWM 的极性是可配置的，通过设置位于 CCP1CON 寄存器的 CCP1M 位来选择。

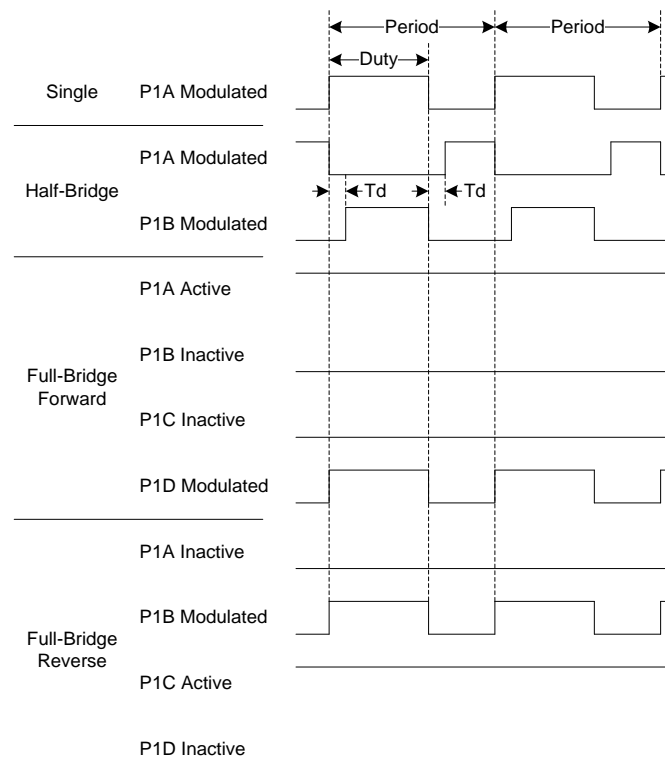
下面表格为 PWM 增强模式下引脚的分配情况。

PWM 模式	PWM1M	P1A	P1B	P1C	P1D
单路	00	Yes	No	No	No
半桥	10	Yes	Yes	No	No
全桥正向	01	Yes	Yes	Yes	Yes
全桥反向	11	Yes	Yes	Yes	Yes

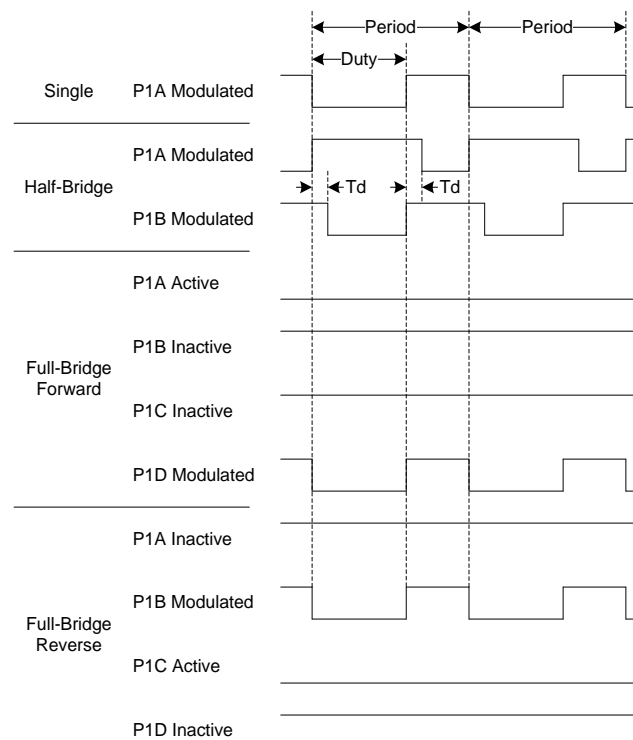
下面表格为 PWM 增强模式下引脚的有效电平状态

PWM 模式	PWM1M	CCP1M[1]=0	CCP1M[1]=1	CCP1M[0]=0	CCP1M[0]=1
单路	00	P1A 为高有效	P1A 为低有效	-	-
半桥	10	P1A 为低有效	P1A 为高有效	P1B 为高有效	P1B 为低有效
全桥正向	01	P1A 和 P1C 为高有效	P1A 和 P1C 为低有效	P1B 和 P1D 为高有效	P1B 和 P1D 为低有效
全桥反向	11	P1A 和 P1C 为高有效	P1A 和 P1C 为低有效	P1B 和 P1D 为高有效	P1B 和 P1D 为低有效

下图为 PWM 增强模式下的输出波形（高有效）



下图为 PWM 增强模式下的输出波形（低有效）



上面两图中，Period 为 PWM 的周期，Duty 为 PWM 的占空比，Td 为死区时间。

半桥模式

半桥模式下，有两个输出引脚来驱动推拉负载。一路 PWM 信号由 P1A 引脚输出，另一路互

补的 PWM 信号由 P1B 引脚输出。这个模式可用在半桥应用上，也可用在全桥应用上，其四个电源开关由两路 PWM 信号来调制。

半桥应用当中，所有的电源开关都通过 PWM 频率来调制。通常情况下，电源开关的关闭比打开需要更多的时间。如果上下电源开关在同一时间切换（一个打开，另一个关闭），两个开关可能会在短时间内同时处于打开状态，直到其中一个完全关闭。在这段短时间内，两个开关之间会通过一段很大的电流（直通电流），造成半桥短路。为了避免在切换过程中产生这种潜在的破坏性直通电流，通常在每个开关打开前加上一段延时，以保证另一个开关已经完全关闭。

半桥模式下，可编程的死区延时用来防止在半桥器件之间产生直通电流。死区延时发生在信号由非有效状态切换至有效状态期间。位于 PWM1CON 寄存器的 PWMDC[6:0]位，用来设置输出驱动有效前的 PWM 周期数。若 PWMDC 的值大于占空比的周期数，相应的输出在整个 PWM 周期内都会保持无效。

因 P1A 和 P1B 输出是和 IO 端口输出复用的，相应的 TRIS 位必须设置为低，来配置 P1A 和 P1B 输出到相应的引脚上。

全桥模式

全桥模式下，4 个引脚都被用作输出。

正向模式下，P1A 引脚被驱动至有效状态，P1D 引脚被调制，而 P1B 和 P1C 会被驱动至无效状态。反向模式下，P1C 引脚被驱动至有效状态，P1B 引脚被调制，而 P1A 和 P1D 会被驱动至无效状态。

因 P1A, P1B, P1C 和 P1D 输出是和 IO 端口输出复用的，相应的 TRIS 位必须设置为低，来配置 P1A, P1B, P1C 和 P1D 输出到相应的引脚上。

全桥模式下方向的改变

全桥模式下，位于 CCP1CON 寄存器的 PWM1M[1]位被用来控制正向和反向模式。当软件改变这个方向控制位的值后，方向模式会在接下来的 PWM 周期改变。

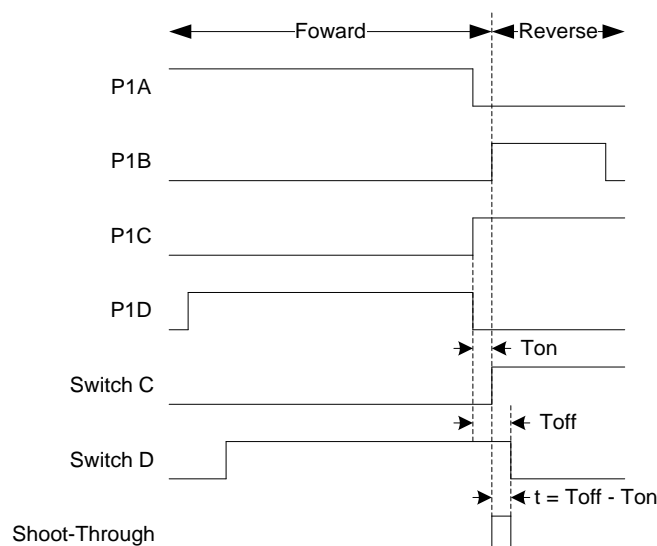
软件改变位于 CCP1CON 寄存器的 PWM1M[1]位后，方向改变就会启动。在当前 PWM 周期溢出前的 4 个定时器 2 周期内，将会发生下面的操作：

- 调制输出（P1B 或 P1D）会变成无效状态；
- 相应的未被调制输出（P1C 或 P1A）会切换至相反状态；
- PWM 调制会在接下来的周期开始时恢复。

全桥模式下，不会产生死区延时。因在一个时刻，只有一路输出是被调制的，通常情况下死区延时是不需要的。当下面两个条件同时存在时，需要死区延时：

- 1) 当输出的占空比为 100%或接近 100%，PWM 输出方向发生改变；
- 2) 电源开关包括电源器件和驱动电路中，关闭时间大于开启时间。

下图为占空比接近 100%，PWM 的方向由正向改变为反向时的实例。在 t1 时刻，P1A 和 P1D 输出变成无效，而 P1C 输出变成有效。因关闭时间大于开启时间，电源器件 C 和 D 之间会产生直通电流，持续时间为 t。当 PWM 的方向由反向改变为正向时，会在器件 A 和 B 之间发生同样的现象。



如果实际应用当中需要在高占空比时改变 PWM 的方向，有以下两个可能的解决方案来消除直通电流：

- 1) 在改变方向前一个 PWM 周期减小 PWM 占空比；
- 2) 使用关闭速度快于开启速度的开关驱动。

启动事项

无论使用哪种 PWM 模式，硬件须在 PWM 输出引脚上使用合适的外部上拉或下拉电路。当控制器复位被释放后，所有的 IO 引脚都处于高阻状态。外部电路须保持电源开关器件处于 OFF 状态，直到控制器驱动 IO 引脚至合适的电平状态。

位于 CCP1CON 寄存器的 CCP1M[1:0]位可用来选择 PWM 输出信号的有效驱动电平极性。在 PWM 引脚输出驱动使能之前，PWM 的输出极性须先设定好。在 PWM 引脚输出驱动使能后，不建议再改变极性配置。

PWM 模块被初始化之后，P1A, P1B, P1C 和 P1D 输出有可能不在合适的状态。此时使能 PWM 引脚输出有可能会对应应用电路造成损坏。因此建议在一个完整的 PWM 周期后再使能 PWM 输出驱动。当位于 PIR1 寄存器的 TMR2IF 位被置位后，表明当前 PWM 周期结束，下一个 PWM 周期开始。

增强 PWM 自动关闭模式

PWM 模式支持自动关闭，当关闭事件发生时，将会禁止 PWM 输出。自动关闭模式可以放置 PWM 输出引脚至预设状态。这可以用来放置 PWM 损坏应用电路。

位于 ECCPASR 寄存器的 ECCPAS[2:0]位用来选择自动关闭源。自动关闭事件可由下面来产生：

- INT 引脚上的“0”电平
- 比较器 1
- 比较器 2
- 软件设置 ECCPASE 位

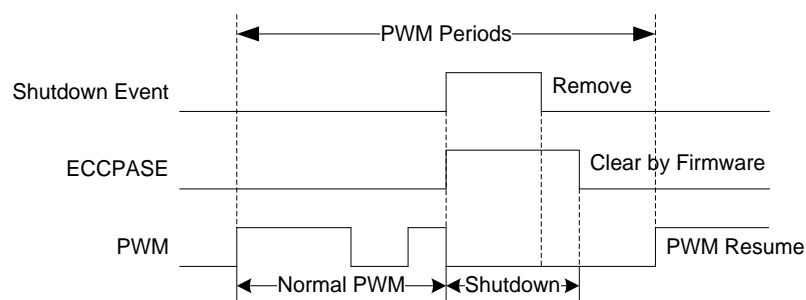
位于 ECCPASR 寄存器的 ECCPASE 位用来表明自动关闭的状态。当 ECCPASE 位为“0”时，PWM

输出为正常状态。当 **ECCPASE** 位为“1”时，PWM 输出位自动关闭状态。

当自动关闭事件发生时，会发生下面两件事：

- 1) **ECCPASE** 位被置位。**ECCPASE** 位会保持为高，直到软件把它清零，或者发生自动重启；
- 2) **CCP1IF** 位被置位。**CCP** 中断使能时，会产生 **CCP** 中断。
- 3) **PWM** 引脚输出立刻被切换至自动关闭预设状态。**PWM** 输出引脚被分为 **P1A/P1C** 和 **P1B/P1D** 两组，每组的自动关闭状态可由位于 **ECCPASR** 寄存器的 **PSSAC** 和 **PSSBD** 位来选择。每组引脚可切换至以下三种状态中的一种：
 - 驱动为逻辑高电平
 - 驱动为逻辑低电平
 - 高阻状态

下图为自动关闭操作示例图



自动关闭条件是电平驱动信号，非边沿驱动信号。当电平保持不变时，自动关闭状态可持续。

当自动关闭条件保持时，软件写 **ECCPASE** 位为无效操作。

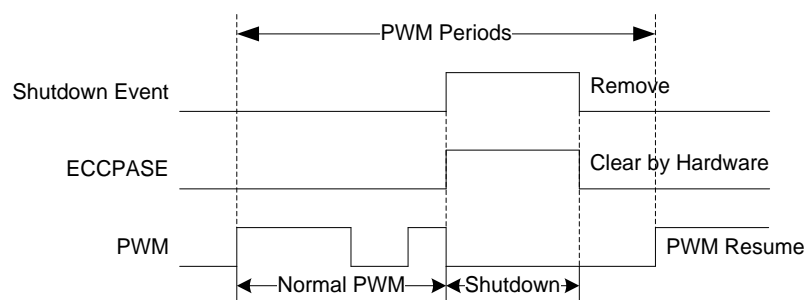
当自动关闭条件移除后，**PWM** 会重启（软件或自动重启），**PWM** 信号输出会在接下来的 **PWM** 周期开始重启。

自动重启模式

增强 **PWM** 模式可配置 **PWM** 信号自动重启，当自动关闭条件移除后，**PWM** 信号可自动重新启动输出。置位位于 **PWM1CON** 寄存器的 **PRSEN** 位，即可使能自动重启功能。

自动重启被使能后，自动关闭条件仍有效时，**ECCPASE** 位将会保持置位。当自动关闭条件移除时，**ECCPASE** 位会被硬件清零，**PWM** 恢复常规操作。

下图为自动重启操作示例图



寄存器定义

CCP1CON - CCP 控制寄存器

CCP1CON - CCP 控制寄存器								
地址: 0x15					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
	PWM1M1	PWM1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:6	PWM1M	PWM 输出配置 当 CCP1M[3:2] = 00, 01, 10 时: P1A 配置为 Capture/Compare 输入; P1B, P1C 和 P1D 配置为 IO 端口 当 CCP1M[3:2] = 11 时 11: 全桥输出反向模式; P1B 调制; P1C 有效; P1A 和 P1D 无效 10: 半桥输出: P1A 和 P1B 调制, 带死区控制; P1C 和 P1D 配置为 IO 端口 01: 全桥输出正向模式; P1D 调制; P1A 有效; P1B 和 P1C 无效 00: 单路输出; P1A 调制; P1B, P1C 和 P1D 配置为 IO 端口						
5:4	DC1B	PWM 占空比低两位 Capture 模式: 未使用 Compare 模式: 未使用 PWM 模式: 作为 PWM 占空比的低两位, 与 CCPR1L 组成 10 位占空比						
3:0	CCP1M	ECCP 模式选择位 1111: PWM 模式; P1A 和 P1C 输出极性反向; P1B 和 P1D 输出极性反向 1110: PWM 模式; P1A 和 P1C 输出极性反向; P1B 和 P1D 输出极性不变 1101: PWM 模式; P1A 和 P1C 输出极性不变; P1B 和 P1D 输出极性反向 1100: PWM 模式; P1A 和 P1C 输出极性不变; P1B 和 P1D 输出极性不变 1011: Compare 模式; 匹配时触发特殊事件 (CCP1IF 位被置位, 复位 TMR1, ADC 模块使能时开启一次 ADC 转换) 1010: Compare 模式; 匹配时产生软件中断 (CCP1IF 被置位, CCP1 引脚不受影响) 1001: Compare 模式; 匹配时清零输出 (CCP1IF 被置位) 1000: Compare 模式; 匹配时置位输出 (CCP1IF 被置位) 0111: Capture 模式; 每 16 个上升沿 0110: Capture 模式; 每 4 个上升沿 0101: Capture 模式; 每个上升沿 0100: Capture 模式; 每个下降沿 0011: 保留 0010: Compare 模式; 匹配时翻转输出 (CCP1IF 被置位) 0001: 保留 0000: Capture/Compare/PWM 关闭 (复位 ECCP 模块)						

PWM1CON – PWM 控制寄存器

PWM1CON – PWM 控制寄存器								
地址：0x16					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	PRSEN	PWMDC6	PWMDC5	PWMDC4	PWMDC3	PWMDC2	PWMDC1	PWMDC0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	PRSEN	PWM 重启使能控制 1：自动关闭发生后，当关闭事件移除时，ECCPASE 位被硬件自动清零；PWM 自动重启 0：自动关闭发生后，须通过软件清零 ECCPASE 位来重启 PWM						
6:0	PWMDC	PWM 延时计算控制 PWM 信号从非有效状态转换至有效状态期间的死区延时,用 PWMDC 个系统时钟周期来表示						

ECCPASR – ECCP 自动关闭控制寄存器

ECCPASR – ECCP 自动关闭控制寄存器								
地址：0x17					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	ECCPASE	ECCP 自动关闭使能控制状态 1：自动关闭已经发生；ECCP 输出处于自动关闭预设状态 0：ECCP 正常输出						
6:4	ECCPAS	ECCP 自动关闭源选择控制 111：INT 引脚为低电平；或比较器 2 输出改变；或比较器 1 输出改变 110：INT 引脚为低电平；或比较器 2 输出改变 101：INT 引脚为低电平；或比较器 1 输出改变 100：INT 引脚为低电平 011：比较器 2 输出改变；或比较器 1 输出改变 010：比较器 2 输出改变 001：比较器 1 输出改变 000：自动关闭被禁止						
3:2	PSSAC	P1A 和 P1C 自动关闭预设状态控制 1x：高阻 01：驱动为高电平 00：驱动为低电平						
1:0	PSSBD	P1B 和 P1D 自动关闭预设状态控制 1x：高阻 01：驱动为高电平 00：驱动为低电平						

CCPR1L – CCP 寄存器低字节

CCPR1L – CCP 寄存器 1 低字节								
地址：0x13					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	CCPR1L7	CCPR1L6	CCPR1L 5	CCPR1L 4	CCPR1L 3	CCPR1L 2	CCPR1L 1	CCPR1L 0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	CCPR1L	CCP 寄存器 1 低字节 Capture 模式：捕获发生时，所记录 TMR1L 的值 Compare 模式：发生比较时，与 TRM1L 进行比较的值 PWM 模式：产生 PWM 时，16 位可调占空比低八位的值						

CCPR1H – CCP 寄存器高字节

CCPR1H – CCP 寄存器 1 高字节								
地址：0x14					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	CCPR1H7	CCPR1H6	CCPR1H 5	CCPR1H 4	CCPR1H 3	CCPR1H 2	CCPR1H 1	CCPR1H 0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	CCPR1H	CCP 寄存器 1 高字节 Capture 模式：捕获发生时，所记录 TMR1H 的值 Compare 模式：发生比较时，与 TRM1H 进行比较的值 PWM 模式：产生 PWM 时，16 位可调占空比高八位的值						

TCCR – 定时器时钟控制寄存器

TCCR – 定时器时钟控制寄存器								
地址：0x1D					0X00_00XX			
Bit	7	6	5	4	3	2	1	0
TCCR	X2EN	-	T2CF	T2CS	T1CF	T1CS	-	-
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	X2EN	倍频时钟使能控制，1=使能倍频时钟						
6	-	Unimplemented						
5	T2CF	定时器 2 倍频模式标志位，为 1 表示定时器 2 倍频设置有效						
4	T2CS	定时器 2 倍频时钟使能位。1=设置定时器 2 工作于倍频模式						
3	T1CF	定时器 1 倍频模式标志位，为 1 表示定时器 1 倍频设置有效						
2	T1CS	定时器 1 倍频模式使能位。写 1 设置定时器 1 工作于倍频模式						
1:0	-	Unimplemented						

互补信号发生器(COG)

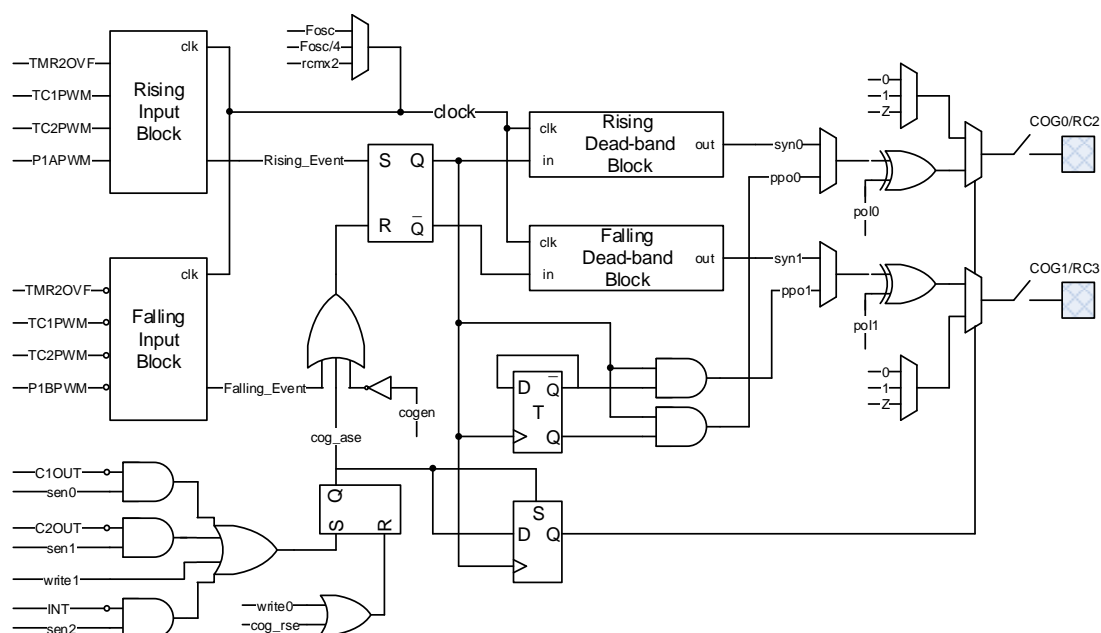
- 两种输出模式：同步 PWM 和 Push-Pull 输出
- 可选时钟源
- 独立可选的上升事件源
- 独立可选的下降事件源
- 独立可选的边沿或电平事件触发方式
- 独立的输出使能和输出极性选择
- 独立的上升和下降相位延迟
- 独立的上升和下降死区时间
- 自动关闭 PWM 输出控制：
 - 独立可选的关闭源
 - 自动重启使能
 - 自动关闭输出状态控制

综述

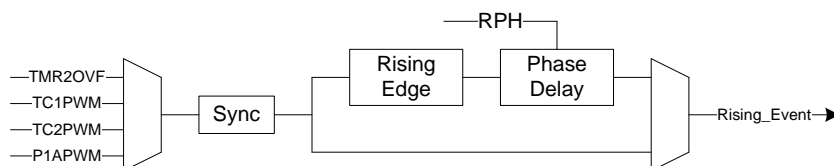
互补输出产生器模块（简称为 **COG** 模块）的主要作用是转换单路 PWM 信号为两路互补的 PWM 信号，亦可利用两个单独的输入事件来产生单一或互补的 PWM 输出。

COG 模块所输出 PWM 信号的频率和占空比由上升事件输入和下降事件输入来决定。上升事件发生的速率决定了 **COG** 模块输出 PWM 信号的频率，上升事件和下降事件的时间间隔决定了 PWM 信号的占空比。上升事件和下降事件可为相同的事件源，也可不相同。这些事件源可与 **COG** 模块的时钟同步或异步。**COG** 模块的时钟输入是可选的，用来产生相位延迟和死区时间。

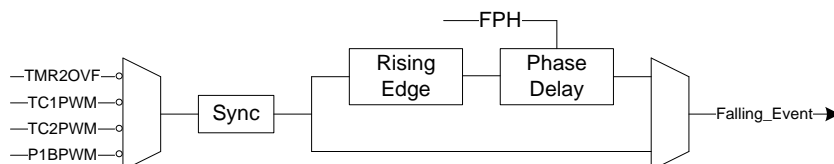
COG 模块的框架图



下图为上升事件输入模块的框架图。



下图为下降事件输入模块的框架图。



输入模式

COG 模块对输入信号由两种采样模式：同步采样和异步采样。当输入信号与 COG 模块时钟为同一个时钟域时，可设置位于 COGCR0 寄存器的 COGIMD 位为 0，上升和下降输入模块会对输入信号进行同步采样；当输入信号与 COG 模块时钟不属于同一个时钟域时，须设置位于 COGCR0 寄存器的 COGIMD 位为 1，上升和下降输入模块会对输入信号进行异步采样，即会对输入信号采用二级触发器同步后再采样。

输出模式

COG 模块的输出信号由两种模式：Synchronous PWM 模式和 Push-Pull 模式。由位于 COGCR0 寄存器的 COGOMD 位来选择，当设置 COGOMD 位为 0 时，输出采用 Synchronous PWM 模式；当设置 COGOMD 位为 1 时，输出采用 Push-Pull 模式。

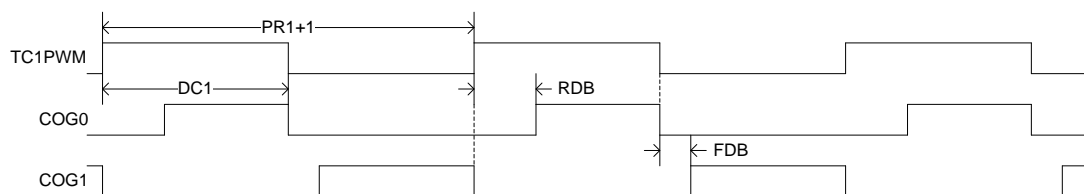
同步 PWM 模式

在 Synchronous PWM 模式下，COG 模块通过上升和下降事件源来产生两路输出互补的 PWM 信号波形。最简单的配置是，上升和下降事件源是同一个信号，即有着一定频率和占空比的 PWM 信号。COG 模块把这个单路的 PWM 输入转换成两路互补的 PWM 输出。两路 PWM 输出信号的频率和占空比与原单路输入的 PWM 信号相匹配。每一路输出信号由关至开的转变都可以相对于另一路信号由开至关的转变产生延迟，从而在每次 PWM 的转变后都会产生一定的时间间隔，即没有哪一路输出是有效的。这个时间间隔即被称为“死区时间”，在“死区控制”一节会有更详细的介绍。

下图为一种典型的工作波形，由单路的 TC1PWM 输入信号产生，并带有死区时间。其中，TC1PWM 信号由 16 位定时计数器 1 产生，其周期由 16 位寄存器 PR1(PR1H:PR1L)决定，占空比由 16 位寄存器 DC1(DC1H:DC1L)决定，如下公式所示：

$$\text{Period} = [(PR1H:PR1L) + 1] * T_{osc}$$

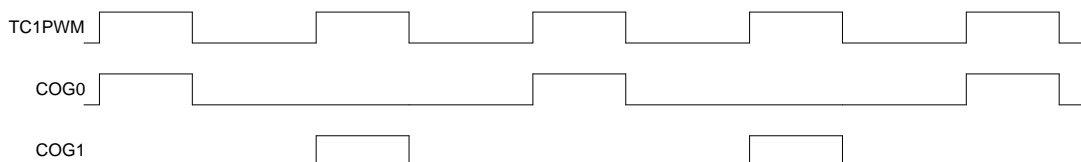
$$\text{Pulse Width} = [(DC1H:DC1L)] * T_{osc}$$



Push-Pull 模式

在 Push-Pull 模式下, COG 模块产生一路 PWM 信号, 在两个 COG 输出引脚上周期性地交替输出。从上升输入事件开始输出有效, 直到下降输入事件后输出终止。每一个上升事件开始一个新的周期, 并把 PWM 信号输出到前一个周期未用到的引脚上。

下图为一种典型的工作波形, 由单路的 CCP1 输入信号产生。



时钟源

COG 模块的时钟是各个定时器的基准时钟, 此定时器包括: 上升和下降死区时间定时器; 上升和下降事件相位延迟时间定时器。

COG 模块的时钟源可通过 COGCR1 寄存器的 COGCS[1:0]控制位从以下时钟中选择:

- 1) 高速时钟, 即内部 RCM 时钟的两倍频时钟;
- 2) 指令时钟 ($F_{osc}/4$);
- 3) 系统时钟 (F_{osc});

可选事件触发源

COG 模块通过各个独立可选的事件源的不同组合来产生互补波形。事件源可分为两类: 上升事件源和下降事件源。

有效的上升事件源由 COGCR1 寄存器的 COG0RS[2:0]控制位来选择, 有效的下降事件源由 COGCR1 寄存器的 COG1RS[2:0]控制位来选择。

下表格为上升事件源选择控制表格:

COG0RS[2]	COG0RS[1]	COG0RS[0]	上升事件源
0	X	X	保留
1	0	0	定时计数器 2 溢出
1	0	1	TC1PWM
1	1	0	TC2PWM
1	1	1	P1A

下表格为下降事件源选择控制表格:

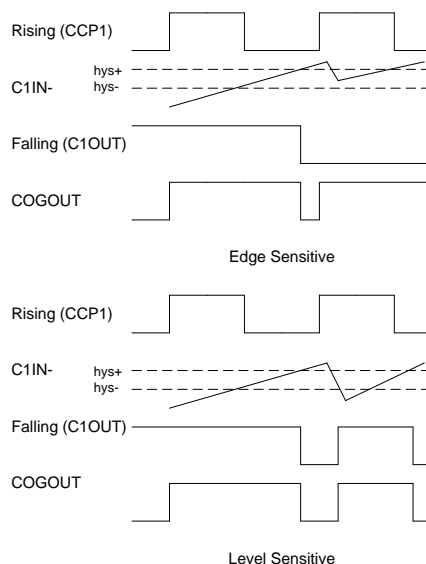
COG1RS[2]	COG1RS[1]	COG1RS[0]	下降事件源
0	X	X	保留
1	0	0	定时计数器 2 溢出
1	0	1	TC1PWM
1	1	0	TC2PWM
1	1	1	P1B

边沿和电平触发方式

事件输入检测可设置为电平或边沿触发, 由 **COGCRO** 寄存器的 **COGIMD** 位来选择。当 **COGIMD** 位为高时, 采用边沿检测, 为低时, 采用电平检测。

通常情况下, 由周期性信号源所驱动的事件采用边沿检测, 而由目标电路的电压门限所引起的事件采用电平检测。注意下面两种情形:

- 1) 第一种情形, **COG** 输出信号的周期由占空比为 **50%** 的输入时钟决定, 占空比由比较器所反馈的电压电平决定。如果时钟输入采用电平敏感, 占空比小于 **50%** 时将会产生不可预期的操作。
- 2) 第二种情形, 跟第一个实例类似, 只是占空比接近 **100%**。反馈比较器由高至低的转换会关闭 **COG** 的输出驱动, 但很快又会被周期信号打开输出驱动。如果时钟低电平时间足够短, 比较器的输入信号可能不会越过迟滞区到达低电位, 因而不能使比较输出发生变化。比较器输出持续为低, 缺少由高至低的变化来触发边沿检测, 从而使得 **COG** 输出保持为驱动常开的状态。如下图所示:



上升事件

上升事件会开启 **PWM** 输出为有效电平。上升事件模块的输出信号由低至高的转换即产生上升事件。当上升事件的相位延迟和死区时间均设置为 **0** 时, **COG0** 会立即输出。否则, **COG0** 会延时输出。上升事件源会触发以下动作:

- 开启上升事件相位延迟计数器 (使能时)
- 相位延迟后, 清零 **COG1** 输出信号
- 开始死区时间延时 (使能时)

- 死区延迟后，置位 COG0 输出信号

下降事件

下降事件会终止 PWM 输出为有效电平。下降事件模块的输出信号由低至高的转换即产生下降事件。当下降事件的相位延迟和死区时间均设置为 0 时，COG1 会立即输出。否则，COG1 会延时输出。下降事件源会触发以下动作：

- 开启下降事件相位延迟计数器（使能时）
- 相位延迟后，清零 COG0 输出信号
- 开始死区时间延时（使能时）
- 死区延迟后，置位 COG1 输出信号

输出控制

当禁止或者刚使能 COG 模块时，互补驱动的状态被配置为，COG0 处于驱动无效状态和 COG1 处于驱动有效状态。

每一路 COG 输出引脚都有独立的输出使能控制。输出使能由位于 COGCR0 寄存器的 COGOE0 和 COGOE1 位来选择。当输出使能控制被清零时，COG 模块的输出不再施加于其相应引脚上。当输出使能控制被置位时，优先值或 PWM 波形被输出至相应引脚上。输出使能控制位与位于 COGCR0 寄存器的 COG 模块的使能位 COGEN 无关。当 COGEN 位为被清零，且关闭控制处于无效时，PWM 波形的复位状态电平输出至 COG 输出引脚上。当关闭控制有效，COGEN 位被清零时，关闭状态的优先电平输出至 COG 输出引脚上。

极性控制

每一路 COG 的输出极性都可以独立选择。输出极性由位于 COGCR0 寄存器的 COGPOLO 和 COGPOL1 位来选择。当输出极性控制位被置位时，相应的输出为低电平有效。当输出极性控制位被清零时，相应的输出为高电平有效。但是，极性控制不会影响关闭状态下的优先电平。

死区控制

死区控制可以提供不相叠加的 PWM 互补输出信号，防止外部电源切换时产生直通电流。COG 模块包含两个死区时间计数器。一个用于上升事件死区时间控制，另一个用于下降事件死区时间控制。COG0 输出信号的死区时间由位于 COGDBR 寄存器的 COG0DBR[3:0]位来设置。COG1 输出信号的死区时间由位于 COGDBR 寄存器的 COG1DBR[3:0]位来设置。当 COG0DBR 或 COG1DBR 的值为 0 时，死区延迟被禁止。当 COG0DBR 或 COG1DBR 的值非 0 时，死区时间计数器工作在 COG 模块时钟下，并计数所需的周期数。

上升事件死区是在 COG1 信号失活和 COG0 信号激活之间增加延时。当上升事件模块输出有效时，上升事件死区计数器开始计数。

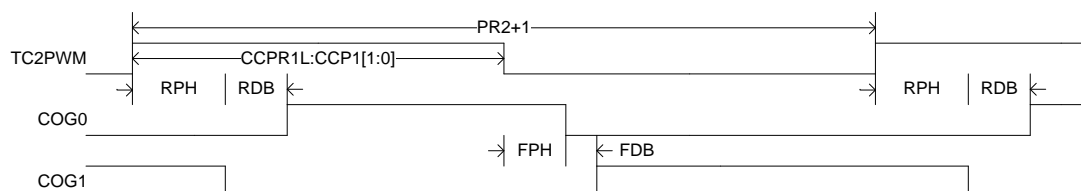
下降事件死区是在 COG0 信号失活和 COG1 信号激活之间增加延时。当下降事件模块输出有效时，下降事件死区计数器开始计数。

死区可能会发生交叠，有两种交叠情形：上升至下降的交叠和下降至上升的交叠。在上升至下降的交叠中，上升事件死区计数器还未完成计数，而下降事件已经发生。在这种情形下，**COG0** 的有效驱动时间会被压缩，死区时间会因下降事件死区时间的插入而延长。当延长的死区时间结束后，**COG1** 输出有效驱动。

在下降至上升的交叠中，下降事件死区计数器还未完成计数，而上升事件已经发生。在这种情形下，**COG1** 的有效驱动时间会被压缩，死区时间会因上升事件死区时间的插入而延长。当延长的死区时间结束后，**COG0** 输出有效驱动。

相位延迟

上升事件和下降事件后均可插入相位延迟。当往 **COGPHR** 寄存器的 **COGRPH[3:0]** 和 **COGFPH[3:0]** 写入非 0 值时，则会插入相位延迟。如下图所示，**COG** 操作是在 **TC2PWM** 信号的基础上插入相位延时。从上升事件信号到真实执行该事件时的延时即为相位延时，其计算方式和死区时间一样。其中，**TC2PWM** 信号由定时计数器 2 产生，其产生方式请参考 **ECCP** 模块中 **PWM** 模式章节的描述。



当相位延迟所设置的值为 0 时，相位延迟被禁止，上升和下降事件信号忽视相位延迟计数器而直接通过。

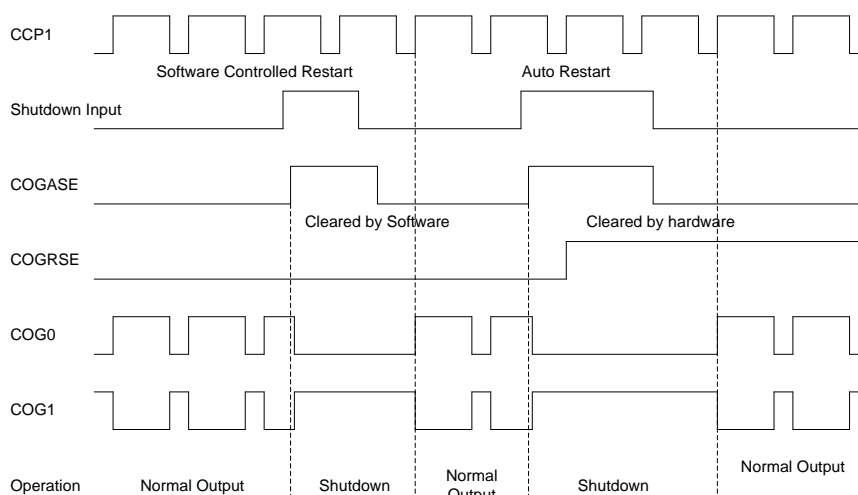
自动关闭控制

自动关闭是一种及时覆盖 **COG** 的输出电平至特定电平值的有效方法，从而能够及时安全地关闭后续的电路。

自动关闭的状态能够被自动清除，也能够一直保持直到被软件清除。在这两种情形下，自动关闭下的优先值会保持有效，直到自动关闭被清除后的第一个上升事件发生时才失效。

有以下两种机制来进入关闭状态：软件产生或外部输入

当设置位于 **COGASR** 寄存器的 **COGASE** 位为 1 时，**COG** 输出信号会强制进入关闭状态。当自动重启未使能时，关闭状态会一直保持，直到 **COGASE** 位被清零后的第一个上升事件发生。当自动重启使能后，自动关闭输入控制消失后，**COGASE** 位会被自动清零，**COG** 操作也会在第一个上升事件发生后恢复。如下图所示。



当故障条件产生时，外部自动关闭输入控制可以提供一种最快的方式来安全的暂停 COG 操作。当任一所选的自动关闭输入有效时，COG 的输出会立即变换成所设定的电平值，而不会引入软件延时。

任一输入源的组合都可以选作为自动关闭的条件。自动关闭输入源包括：

- 模拟比较器 1 的输出 **C1OUT**（低电平有效）
- 模拟比较器 2 的输出 **C2OUT**（低电平有效）
- 外部 **INT** 引脚（低电平有效）

自动关闭输入源都可被独立选用，由位于 **COGASR** 寄存器的 **COGASS[2:0]** 位来控制。自动关闭输入源属于电平敏感，非边沿敏感，当自动关闭输入源的电平一直保持低时，自动关闭状态可一直保持，除非自动关闭控制被禁止。

引脚优先电平

当自动关闭有效时，驱动 COG 输出引脚的电平由位于 **COGASR** 寄存器的 **PSSCOG0[1:0]** 和 **PSSCOG1[1:0]** 来分别控制。**PSSCOG0** 控制 **COG0** 所在引脚的优先电平值，**PSSCOG1** 控制 **COG1** 所在引脚的优先电平值。它们的优先电平值均有 3 种：

- 强制为低电平
- 强制为高电平
- 高阻状态

需注意的是，所选的优先电平值不受极性控制的影响。

自动关闭后重启

当自动关闭发生后，有两种方式来恢复 COG 的操作：软件控制或自动重启。

当位于 **COGASR** 寄存器的 **COGRSE** 位为 0 时，软件必须清零 **COGASE** 位来让 COG 模块从自动关闭状态中恢复。**COGASE** 位被清零后，COG 模块会在第一个上升事件发生后恢复操作。清零自动关闭状态，要求所有所选的自动关闭输入控制均失效，否则 **COGASE** 位会一直置位而不会被清零。

当位于 COGASR 寄存器的 COGRSE 位被置位时，COG 模块将会自动从自动关闭状态中恢复。当所有所选的自动关闭输入控制均失效，第一个上升事件发生后，会自动清零 COGASE 位，COG 模块恢复操作。

COGASE 位和 COGRSE 位均位于 COGASR 寄存器的最高位，读写它们时需注意：

- 当 COGASE 位为 0 时，写 COGASR 寄存器的最高位为 1 时，只会置位 COGASE 位
- 当 COGASE 位为 1 时，写 COGASR 寄存器的最高位为 1 时，会置位 COGASE 位和 COGRSE 位
- 写 COGASR 寄存器的最高位为 0 时，会同时清零 COGASE 位和 COGRSE 位
- 当 COGASE 位或 COGRSE 位为 1 时，读 COGASR 寄存器的最高位时，读到的值为 1
- 当 COGASE 位和 COGRSE 位均为 0 时，读 COGASR 寄存器的最高位时，读到的值为 0

休眠时操作

休眠模式下，当 COG 模块的时钟，上升事件源和下降事件源均处于有效时，COG 模块可继续工作。非深度睡眠模式下，高速时钟可继续有效，并选作 COG 模块的工作时钟。

寄存器定义

COGCR0 – COG 控制寄存器 0

COGCR0 – COG 控制寄存器 0								
地址：0x88					默认值：0000_0000			
Bit	7	6	5	4	3	2	1	0
	COGEN	COGOE1	COGOE0	COGPOL1	COGPOLO	COGCMD	COGIMD	COGOMD
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	COGEN	COG 模块使能控制 1：COG 模块被使能 0：COG 模块被禁止						
6:5	COGOE	COG 输出使能控制 11：COG1 输出，COG0 输出 10：COG1 输出，COG0 不输出 01：COG1 不输出，COG0 输出 00：COG1 不输出，COG0 不输出						
4:3	COGPOL	COG 输出极性控制 11：COG1 为低电平有效，COG0 为低电平有效 10：COG1 为低电平有效，COG0 为高电平有效 01：COG1 为高电平有效，COG0 为低电平有效 00：COG1 为高电平有效，COG0 为高电平有效						
2	COGCMD	COG 输入同步控制 1：输入同步采样使能 0：输入同步采样关闭						
1	COGIMD	COG 输入触发控制						

		1: 边沿触发 0: 电平触发
0	COGOMD	COG 输出模式控制 1: Push-Pull 输出模式 0: Synchronous PWM 输出模式

COGCR1 – COG 控制寄存器 1

COGCR1 – COG 控制寄存器 1								
地址: 0x89					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
	COG1RS2	COG1RS1	COG1RS0	COG0RS2	COG0RS1	COG0RS0	COGCS1	COGCS0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:5	COG1RS	COG 下降事件源选择控制 111: ECCP 的 P1A 输出的反向 110: 定时器 2 的 PWM 输出的反向 101: 定时器 1 的 PWM 输出的反向 100: 定时器 2 的溢出的反向 011: INT 引脚输入的反向 010: CCP1 引脚输入的反向 001: 比较器 2 的输出的反向 000: 比较器 1 的输出的反向						
4:2	COG0RS	COG 上升事件源选择控制 111: ECCP 的 P1B 输出 110: 定时器 2 的 PWM 输出 101: 定时器 1 的 PWM 输出 100: 定时器 2 的溢出 011: INT 引脚输入 010: CCP1 引脚输入 001: 比较器 2 的输出 000: 比较器 1 的输出						
1:0	COGIMD	COG 模块时钟源选择控制 1x: 高速时钟, 即内部 RCM 时钟的两倍频时钟 01: 指令时钟 00: 系统时钟						

COGASR – COG 自动关闭控制寄存器

COGASR – COG 自动关闭控制寄存器								
地址: 0x9A					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
	COGASE COGRSE	COGASS2	COGASS1	COGASS0	PSSCOG11	PSSCOG10	PSSCOG01	PSSCOG00
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

Bit	Name	描述
7	COGASE/C OGRSE	COG 自动关闭/自动重启使能控制状态 详细描述见自动关闭后重启章节
6:4	COGASS	COG 自动关闭源选择控制 111: INT 引脚为低电平; 或比较器 2 输出改变; 或比较器 1 输出改变 110: INT 引脚为低电平; 或比较器 2 输出改变 101: INT 引脚为低电平; 或比较器 1 输出改变 100: INT 引脚为低电平 011: 比较器 2 输出改变; 或比较器 1 输出改变 010: 比较器 2 输出改变 001: 比较器 1 输出改变 000: 自动关闭被禁止
3:2	PSSCOG1	COG1 自动关闭预设状态控制 1x: 高阻 01: 驱动为高电平 00: 驱动为低电平
1:0	PSSCOG0	COG0 自动关闭预设状态控制 1x: 高阻 01: 驱动为高电平 00: 驱动为低电平

COGDBR – COG 死区控制寄存器

COGDBR – COG 死区控制寄存器								
地址: 0x9B					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
	COG1DBR3	COG1DBR2	COG1DBR1	COG1DBR0	COG0DBR3	COG0DBR2	COG0DBR1	COG0DBR0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:4	COG1DBR	COG1 死区延时控制 COG 下降事件死区延时, 为 COG1 信号由非有效状态切换至有效状态时的延时, 延时的时间为 COG1DBR 个 COG 模块时钟周期。						
3:0	COG0DBR	COG0 死区延时控制 COG 上升事件死区延时, 为 COG0 信号由非有效状态切换至有效状态时的延时, 延时的时间为 COG0DBR 个 COG 模块时钟周期。						

COGPHR – COG 相移控制寄存器

COGPHR – COG 相移控制寄存器								
地址: 0x8D					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
	COG1PHR3	COG1PHR2	COG1PHR1	COG1PHR0	COG0PHR3	COG0PHR2	COG0PHR1	COG0PHR0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

Bit	Name	描述
7:4	COG1PHR	COG1 相位延时控制 COG 下降事件相位延时，为 COG1 信号的相位延时，延时的时间为 COG1PHR 个 COG 模块时钟周期。
3:0	COG0PHR	COG0 相位延时控制 COG 下降事件相位延时，为 COG0 信号的相位延时，延时的时间为 COG0PHR 个 COG 模块时钟周期。

模拟比较器 (CM1/2)

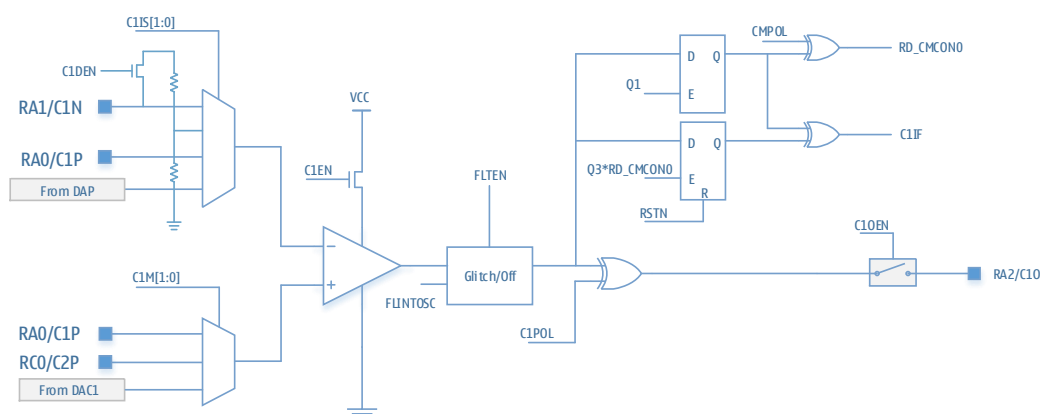
- 双比较器 CM1/CM2
- 低于 10mV 的比较精度
- 多路内部/外部比较器输入
- 独立/组合作模式
- 比较器输出直接到端口
- 可编程输出极性控制
- 支持休眠唤醒
- CM2 输出可作为 TMR1 门控
- CM2 输出与 TMR1 同步
- 独立可编程内部参考电压(DAC1/2)
- 输出迟滞控制
- 可选内部 150us 数字滤波

综述

比较器接收来自正端和负端的模拟输入信号，当正端输入电平大于负端输入电平时，比较器输出高电平。反之则输出低电平。如果正端/负端的输入电平非常相近，比较器的输出可能会出现高低振荡，为避免这种情况，一般比较器输出都有一个迟滞或者数字滤波器，通过使能比较器输出的迟滞电路或者数字滤波，可以获得一个非常稳定的比较结果。

LGT8F684A 内部集成了两个具有不同输入结构的模拟比较器单元(CM1/2)。CM1/2 可以分别独立工作，也可以将他们的输入组合起来，实现不同的并联工作结构。比较器的工作模式由 CMCON2:CCM 位控制，在不同的工作模式下，比较器的输入选择控制信号也将不同。

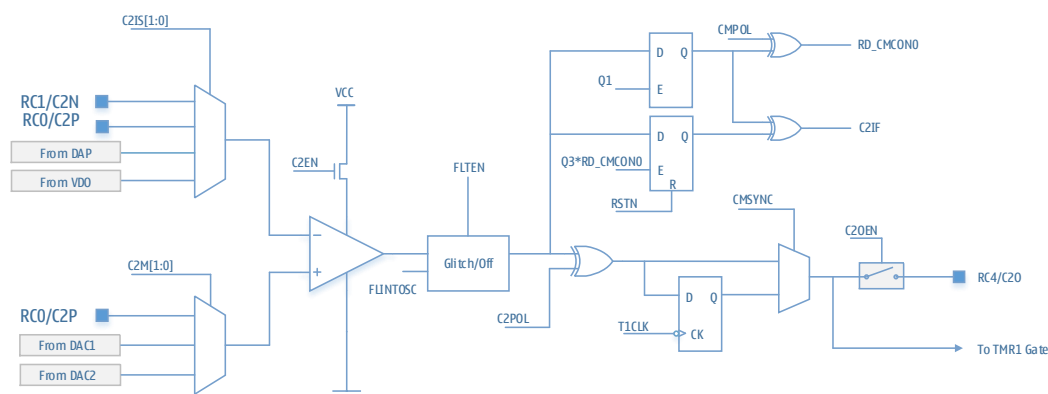
比较器 CM1 结构图：



比较器 CM1 的正端除了专用的 C1P 输入，还可以选择为来自比较器 CM2 的 C2P，这样的配置可以实现 CM1/2 的组合并联工作模式；CM1 的正端输入还可以选择来自内部 6 位 DAC1 的输出。DAC1 的参考电压源可以选择为 VCC/IVREF 或者外部 AVREF 参考。用户可以根据应用需求选择合适的参考电压配置。

CM1 的负端输入有 4 中不同的选择，可以来自 CM1 专用的 C1N，或者 C1N 经过内部 1/5 的分压电路；也可以来自 CM1 的 C1P 或者是来自内部差分放大器的输出。由于差分放大器(DAP)有更加灵活的输入选择，所以这种结构也能够 CM1 的应用带来更自由的空间。DAP 相关的细节，请参考本手册数模转换器(ADC)相关章节的介绍。

比较器 CM2 结构图：



CM2 的正端输入可以为外部专用的 C2P 引脚，也可以选择为内部 6bit DAC1/2。此 DAC2 与 DAC1 具有相同的参考电源结构，可以选择为 VCC，内部 1.2V IVREF 或者来自外部的 AVREF 引脚。CM2 负端有四种输入，分别为：来自 CM2 负端专用输入 C2N，来自 CM2 正端专用输入 C2P，来自内部差分放大器或者来自一个专用的分压模块(VDO)。VDO 是一个多路输入的 1/5 分压模块，多路输入可以选择为 VCC 以及来自芯片外部的 2 路模拟 I/O 输入。VDO 相关细节，请参考本手册模数转换器 ADC 相关章节。

CM2 的输出部分与 CM1 相比，多了与 TMR1 时钟同步的输出控制，以及可以将输出作为 TMR1 的门控时钟。TMR1 门控时钟的使用请参考本手册对定时器 1 介绍的章节。

比较器控制

比较器 CM1/2 支持两种工作模式，工作模式可以通过 CMCON2 寄存器的 CCM 位配置。默认 CCM=0 为组合控制模式。组合控制模式通过设置 CMCON0 的 CM[2:0]位配置比较器 CM1/2 的输入/输出连接方式。当 CCM=1 时，CM1/2 工作在独立控制模式。在独立控制模式下，CM1/CM2 为两个完全独立的模块。用于可以通过寄存器分别配置每个比较器的输入/输出连接方式。在比较器进入正常工作之前，需要首先保证比较器工作所需的环境已经正确的设置：

- 使能模拟输入 I/O 的模拟功能
- 将模拟输入 I/O 的数字功能设置为输入
- 配置好必要的内部参考源
- 比较器输出到 I/O 的功能使能后，相应的 I/O 不能开启模拟输入功能
- 比较器输出滤波/迟滞的正确设置

下面将分别介绍比较器在这两种工作模式下的配置和使用方法：

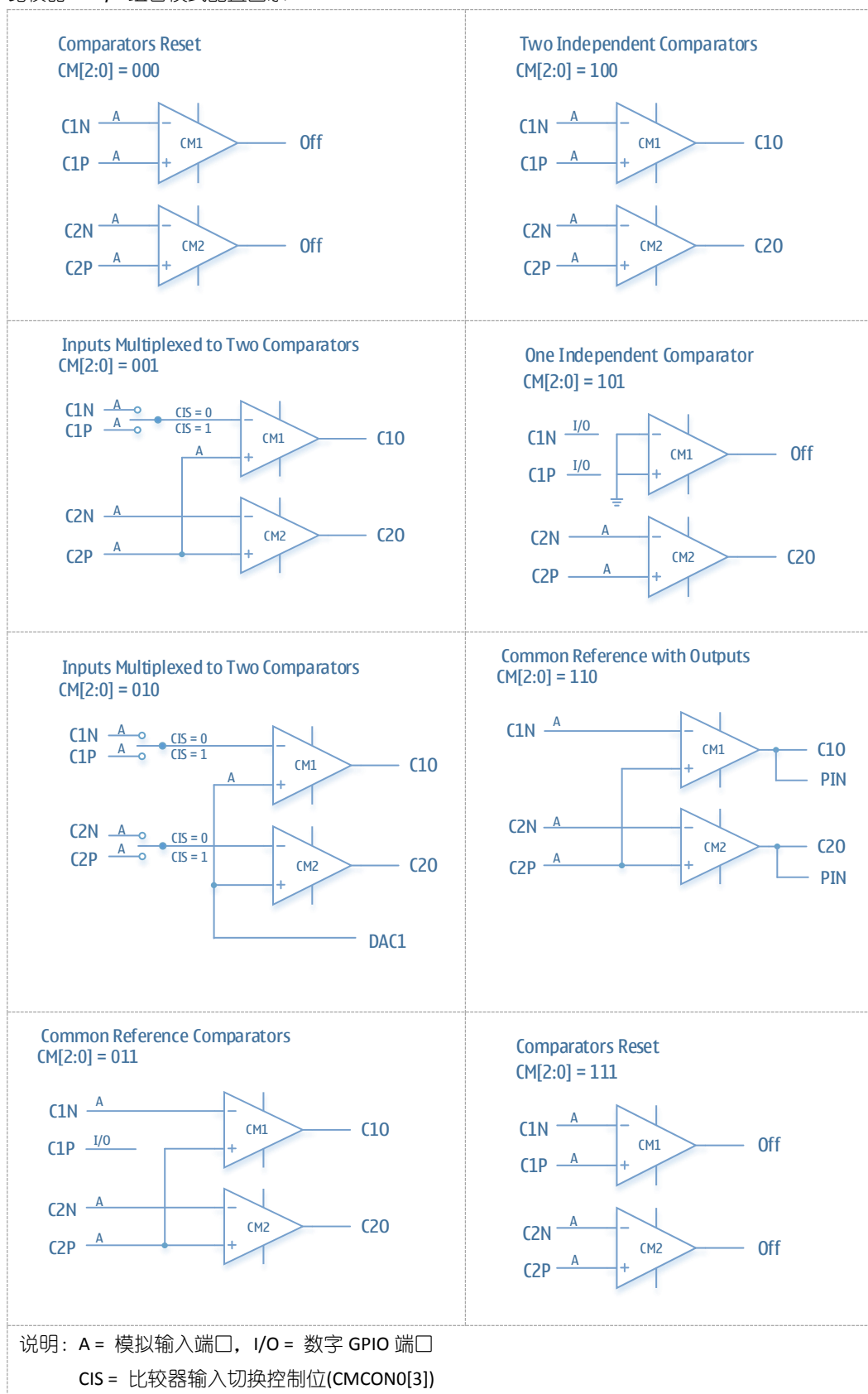
组合控制模式

在组合模式下，CM1/2 一共支持 8 种子工作模式。通过设置 CMCON0 寄存器的 CM 位，即可完成特定工作模式的配置。不同工作模式的拓扑结构，请参考下面的图例。

当端口被设置为模拟输入功能（通过 ANSEL/1），读该端口将固定返回 0。被配置为模拟功能的 I/O，其对应 TRIS 位也设置为 1，禁止端口的驱动电路。

在对比较器进行配置的过程中，建议首先关闭比较器的中断功能，避免当比较器模式发生变化时产生的非正常比较器输出翻转和中断。

比较器 CM1/2 组合模式配置图示：



独立控制模式

当 CMCON0 寄存器的 CM 位设置位 1，比较器 CM0/1 为独立控制模式。在此模式下 CM0/1 为两个完全独立的比较器模块。可以单独使能或者关闭比较器，为每一个比较器独立设置输入源。独立工作模式非常类似于组合控制模式下的模式 CM[2:0] = 100 配置，所不同的是独立控制模式可以自由的设置比较器输入源以及控制比较器输出到端口(C10/C20)。

在独立控制模式下，比较器的使能由 CMCON2 寄存器的 C2EN/C1EN 控制，置位 C2EN/C1EN 将会使能对应的模拟比较器单元 CM2/CM1。比较器输入源的选择设置由 CMCON0 以及 CMCON2 寄存器的低 4 位共同控制，详细的设置如下：

输入选择		控制位	设置描述
CM1	反向输入端	[C1N1, CM0]	00 输入选择引脚 C1N
			01 输入选择引脚 C1N/5 分压电路
			10 输入选择引脚 C1P
			11 输入选择内部差分放大器输出
	正向输入端	[C1P1, CM1]	00 输入选择引脚 C1P
			01 输入选择引脚 C2P
			10 输入选择内部 DAC1 输出
			11 或者内部 10 位 DAC 输出
CM2	反向输入端	[C2N1, CM2]	00 输入选择引脚 C2N
			01 输入选择引脚 C2P
			10 输入选择内部差分放大器输出
			11 输入选择内部 VDO 分压器输出
	正向输入端	[C2P1, C1S]	00 输入选择引脚 C2P
			01 输入选择内部 DAC2 输出
			10 输入选择内部 DAC1 输出
			11

在独立控制模式下，可以通过 CMCON1 寄存器的 C10E/C20E 位将比较器的输出直接连接到芯片的端口上，比较器的输出滤波通过 COMCON1 寄存器的 LFTEN 位使能。详细定义请参考本章节寄存器定义部分。

比较器相关端口控制

比较器的输出状态，可以通过 CMCON0 寄存器的 C1OUT/C2OUT 位获得。比较器的输出也可以直接连接到外部端口(RA2/RC4)。根据不同的工作模式，正确设置比较器输出控制，并不能保证比较器输出与端口相连接，还需清零对应端口设置为输出状态才可以最终将比较器结果输出到外部端口上。

CMCON0 的 C1INV/C1NV 位用于设置比较器的输出极性。将比较器的输出反向，等效于将比较器的正负输入端交换。下面的表格列出比较器在不同状态下的输出结果：

Input Conditions	CxINV	CxO
CxN > CxP	0	0
CxN < CxP	0	1
CxN > CxP	1	1
CxN < CxP	1	0

比较器中断

当比较器的输出产生变化，外设中断寄存器 **PIR1** 的比较器中断标记位 **CMIF** 被置 **1**。此标记位必须通过软件写零清除。如果外设中断控制寄存器 **PIE1** 的 **CMIE** 位被置 **1**，同时使能了全局中断标记位 **GIE**，系统将响应比较器中断请求，执行中断向量。

休眠唤醒

在系统进入休眠模式前使能了模拟比较器，在进入休眠模式后，比较器会仍然处于工作模式。此时比较器会增加休眠功耗。如果我们不需要使用比较器唤醒系统，可以使用 **CMON** 位关闭比较器，减小休眠模式的功耗。

比较器的输出中断可以将处于休眠模式的系统唤醒。为开启比较器的休眠唤醒功能，需要置位 **PIE1** 寄存器的 **CMIE** 位，使能比较器中断。如果使能了全局中断标记为 **GIE**，系统给唤醒后将立刻响应比较器中断请求，执行对应的中断向量。

TMR1 比较器输出门控

此功能可以使用定时器 **TMR1** 测量模拟信号发生变化的时间。将此功能与电容配合，可以被设计为多种非常实用的测量电路。清零 **CMCON1** 寄存器的 **T1GSS** 位，将 **TMR1** 的门控输入设置为比较器的输出。**TMR1** 的工作将直接被比较器的输出状态控制。**TMR1** 相关设置请参考“定时计数器 1”有关门控输入的介绍。

使用比较器作为 **TMR1** 的门控时，同时建议通过 **CMSYNC** 位使能比较器与 **TMR1** 的同步功能。这样可以确保计数器正确的检测到比较器的变化。

输出迟滞滤波

比较器输出端内部支持一个可控的迟滞电。用户可以通过 **CMCON1** 寄存器的 **CMHYS** 位使能迟滞电路。迟滞电路可以消除比较器状态变化过程的不稳定状态，达到输出滤波功能。

建议用户在使用比较器时，打开迟滞电路，获得一个稳定的比较器输出。

迟滞电路位于比较器模拟输出与数字输出之间。当比较器正端的输入电压 V_{IN+} 大于 $(V_{IN-} + V_{H+})$ 时，比较器 **COU** 输出为高；当 V_{IN+} 电压小于 $(V_{IN-} - V_{H-})$ 时，比较器输出低。迟滞电路避免了当比较器正端电压接近负端电压时，电路本身带来的抖动。

寄存器定义

CMCON0 – 比较器控制寄存器 0

CMCON0- 比较器控制寄存器 0								
地址: 0x19					XX000_0000			
Bit	7	6	5	4	3	2	1	0
	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
R/W	R/W	R/W	R/W	R/W	R/W	W/R	R/W	W/R
Bit	Name	描述						
7	C2OUT	比较器 CM2 输出标志位						
6	C1OUT	比较器 CM1 输出标志位						
5	C2INV	比较器 CM2 输出极性控制位						
4	C1INV	比较器 CM1 输出极性控制位						
3	CIS/C2P0	组合模式下输入源选择位，独立模式下为 C2P0						
2	CM2/C2N0	组合模式下为模式选择位，独立模式下为 C2N0						
1	CM1/C1P0	组合模式下为模式选择位，独立模式下为 C1P0						
0	CM0/C1N0	组合模式下为模式选择位，独立模式下为 C1N0。CM[2:0]以及[CxP1, CxP0], [CxN1, CxN0]的定义请参考本章节关于比较器工作模式的相关描述。						

CMCON1 – 比较器控制寄存器 1

CMCON1- 比较器控制寄存器 1								
地址: 0x1A					0000_0010			
Bit	7	6	5	4	3	2	1	0
	C1DEM	C2OE	C1OE	T1ACS	C2HYS	C1HYS	T1GSS	C2SYNC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W/R	W/R
Bit	Name	描述						
7	C1DEM	C1P 内部 1/5 分压通道使能控制位						
6	C2OE	CM2 比较器输出到端口使能控制						
5	C1OE	CM1 比较器输出到端口使能控制						
4	T1ACS	定时器 TMR1 计数时钟选择位 1 = 计数时钟为系统时钟 0 = 计数时钟为指令周期						
3	C2HYS	CM2 比较器迟滞功能使能控制，1=使能；0=禁止						
2	C1HYS	CM1 比较器迟滞功能使能控制，1=使能；0=禁止						
1	T1GSS	定时器 TMR1 门控输入源选择 1 = 外部 T1G 输入作为 TMR1 门控 0 = 比较器 CM2 输出作为 TMR1 门控						
0	C2SYNC	比较器 CM2 输出同步控制 1 = 比较器输出与 TMR1 时钟同步 0 = 比较器输出为异步模式						

CMCON2 – 比较器控制寄存器 2

CMCON2- 比较器控制寄存器 2								
地址: 0x1B					0000_0000			
Bit	7	6	5	4	3	2	1	0
	CCM	C2EN	C1EN	FLTEN	C2P1	C2N1	C1P1	C1N1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W/R	W/R
Bit	Name	描述						
7	CCM	比较器控制模式配置位 1 = 独立控制模式 0 = 组合控制模式						
6	C2EN	独立模式下，用于使能比较器 CM2						
5	C1EN	独立控制模式下用于使能比较器 CM1						
4	FLTEN	比较器输出数字滤波使能控制						
3	C2P1	独立模式下 CM2 正向输入源选择高位						
2	C2N1	独立模式下 CM2 反向输入源选择高位						
1	C1P1	独立模式下 CM1 正向输入源选择高位						
0	C1N1	独立模式下 CM1 反向输入源选择高位						

ANSEL – 端口模拟功能控制寄存器

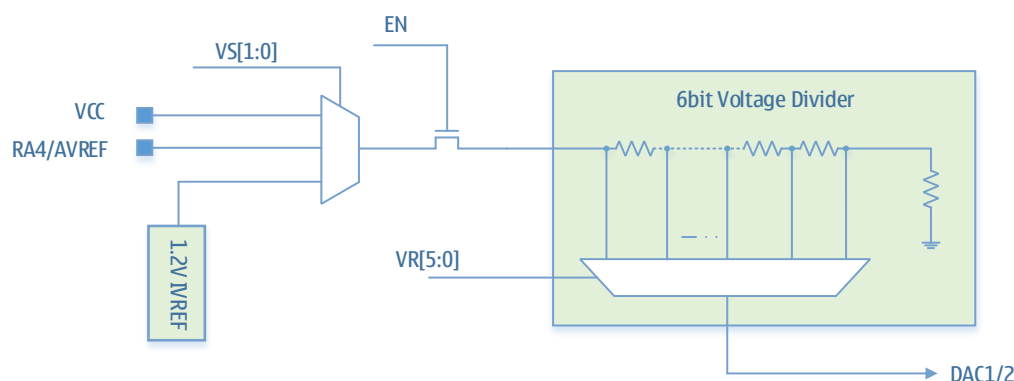
ANSEL- 端口模拟功能控制寄存器								
地址: 0x91					1111_1100			
Bit	7	6	5	4	3	2	1	0
	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
R/W	R/W	R/W	R/W	R/W	W/R	R/W	R/W	W/R
Bit	Name	描述						
7	ANS7	RC3/AN7 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						
6	ANS6	RC2/AN6 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						
5	ANS5	RC1/AN5 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						
4	ANS4	RC0/AN4 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						
3	ANS3	RA4/AN3 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						
2	ANS2	RA2/AN2 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						
1	ANS1	RA1/AN1 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						
0	ANS0	RA0/AN0 模拟输入模式控制, 1 = 模拟模式, 0 = 数字模式						

数模转换器(DAC1/2)

- 2 路 6 位数模转换输出
- 可组合为 1 路 10 位数模转换输出
- DAC 输出可作为模拟比较器参考输入
- 支持 DAC 输出到 I/O 端口
- 可选 VCC/AVREF/IVREF 分压电源

综述

LGT8F684A 内部集成两个完全相同的 6 位可编程数模转换器(DAC1/2), 其中 DAC2 可以配置为一个 10bit 的组合型 DAC, 产生更加精细的 DAC 输出。DAC1/2 的电源可以选择为来自系统工作电源, 内部 1.2V 基准电压源或者来自芯片外部的 RA1/AVREF 输入。DAC1/2 的输出可选择作为内部比较器 CM1/2 的输入参考, 也可以之间输出至芯片的外部引脚上作为外部参考使用。当 DAC 输出至外部引脚时, 不能直接用于驱动负载, 需要通过电压跟随器或其他类似的驱动电路。DAC1/2 内部结构如下图所示:



寄存器定义

DACON – DAC 控制寄存器

DACON- DAC 控制寄存器								
地址: 0x9D					0000_0000			
Bit	7	6	5	4	3	2	1	0
	CTM	-	-	-	DAC2S1	DAC2S0	DAC1S1	DAC1S0
R/W	R/W	-	-	-	R/W	W/R	R/W	W/R
Bit	Name	描述						
7	CTM	测试模式, 正常工作请设置为零						
6:4	-	Unimplemented						
3	DAC2S1	DAC2 电压源选择位 1						
2	DAC2S0	DAC2 电压源选择位 0。[DAC2S1, DAC2S0] = 00: 电压源选择系统工作电压 VCC 01: 电压源选择为外部输入 RA1/AVREF 1X: 电压源选择为内部 1.2V 参考电压						
1	DAC1S1	DAC1 电压源选择位 1						
0	DAC1S0	DAC1 电压源选择位 0。[DAC1S1, DAC1S0] = 00: 电压源选择系统工作电压 VCC						

		01: 电压源选择为外部输入 RA1/AVREF 1X: 电压源选择为内部 1.2V 参考电压
--	--	--

VRCON1 – DAC1 控制寄存器

VRCON1– DAC1 控制寄存器								
地址: 0x99					0000_0000			
Bit	7	6	5	4	3	2	1	0
	DAC1EN	DBGC	VR15	VR14	VR13	VR12	VR11	VR10
R/W	R/W	W/R	W/R	W/R	R/W	W/R	R/W	W/R
Bit	Name	描述						
7	DAC1EN	DAC1 使能控制位, 1 = 使能 DAC1						
6	DBGC	DAC1/2 内部 1.2V 基准使能控制位 对于芯片生产代码小于 1622 的版本, 此位可忽略; 对于生产代码大于或等于 1622 的版本, 此位含义如下: DBGC = 0: 使能内部 1.2V 参考电压 (默认) DBGC = 1: DAC 不使用内部 1.2V 参考。 由于默认 DBGC=0 ; 因此内部 1.2V 参考是处于工作状态, 在低功耗模式下, 软件可以自行关闭, 获得更低的功耗。						
5:0	VR1[5:0]	DAC1 输出电压控制位。VR1[5:0] = 000000 = VIN/64 000001 = 2*VIN/64 111110 = 63*VIN/64 111111 = VIN 注: VIN 为 DAC1 输入电压源						

VRCON2 – DAC2 控制寄存器

VRCON2– DAC2 控制寄存器								
地址: 0x9C					0000_0000			
Bit	7	6	5	4	3	2	1	0
	DAC2EN	D10BEN	VR25	VR24	VR23	VR22	VR21	VR20
R/W	R/W	W/R	W/R	W/R	R/W	W/R	R/W	W/R
Bit	Name	描述						
7	DAC2EN	DAC2 使能控制位, 1 = 使能 DAC2						
6	D10BEN	10 位组合型 DAC 使能						
5:0	VR2[5:0]	DAC2 输出电压控制位。VR2[5:0] = 000000 = VIN/64 000001 = 2*VIN/64 111110 = 63*VIN/64 111111 = VIN 注: VIN 为 DAC2 输入电压源						

12bit 模数转换器(ADC)

- INL: $\pm 1.5\text{LSB}$, DNL: $\pm 1.5\text{LSB}$
- 8 通道外部 ADC 输入
- 内置多路输入 1/5 分压电路(VDO)
- 内置多路输入差分放大器(DAP)
- 可编程差分放大增益(x1/8/16/32)
- 内部 $1.2\text{V} \pm 1\%$ 可校准参考电源(IVREF)
- 差分放大器出厂失调校准
- VCC/IVREF/AVREF 可选参考电源

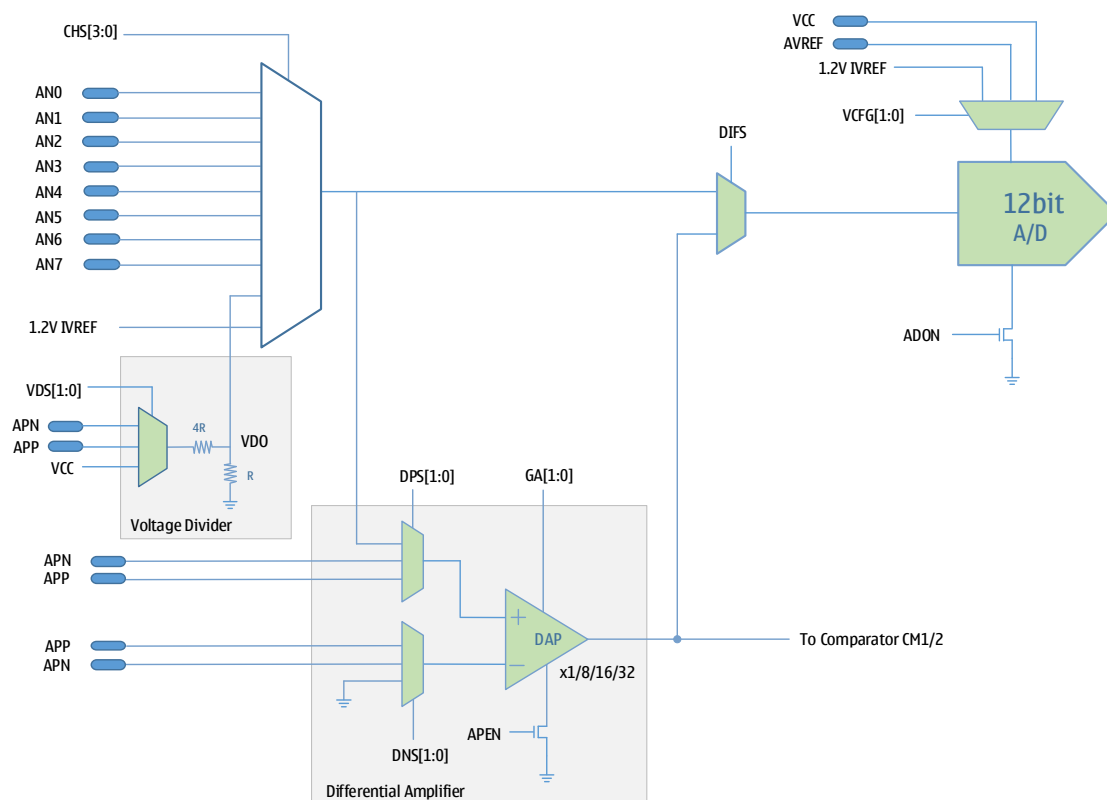
综述

LGT8F684A 内部集成一个高精度的 12 位模数转换器, 可以将模拟输入高速高精度的转换为 12 位的数字信号。ADC 支持多路模拟输入, 内部采用一个模拟多路复用开关将选中的模拟通道与 ADC 内部的采样保持电路连接。ADC 量化器采用主次逼近算法将采样保持电路的模拟信号转换为一个 12 位的二进制结果。在转换周期结束后, 转换的数字信号被存储在 ADC 结果寄存器中(ADRESL/ADRESH)。

ADC 的参考基准可以根据应用需要在三种内、外部输入源之间选择, 包括系统工作电源 VCC, 来自外部引脚输入的参考电压, 以及来自内部 1.2V 基准电压源。

由于 ADC 转换的精度对参考基准以及系统工作电源有决定性的关系, 因此一些高精度的 ADC 应用中, 应该对芯片的工作电源采取有效的滤波处理。同时对来自外部的模拟输入也需要在应用允许的范围内, 在靠近芯片输入引脚的位置设置较小的滤波电容(pF 级)。

ADC 内部结构示意图:



差分放大器(DAP)

单端模拟输入信号无法抑制来自系统电源以及周围环境的共模干扰，因此对于大部分小信号模数转换应用，都无法

成差分放大器的输入源选择，放大增益选择；APCON 的 APEN 位用于使能或者关闭差分放大器。简单的通过 ADC 直接转换。LGT8F684A 内部集成一个出厂校准的可编程增益差分放大器，非常高效的解决了以上问题。差分放大器采用轨对轨输入结构，可接受系统工作电源范围内的输入信号，内部增益支持 x1/x8/x16/x32。用户可以根据输入信号的级别自由的选择。

另外，差分放大器不仅仅可以与 ADC 协同工作，也可以完全独立运行。差分放大器的输入不仅可以来自 ADC 内部的多路复用输出，也可以选择直接来自外部引脚。差分放大器的输出可以供 ADC 转换，也可以输出给比较器 CM1/2，实现更加高效的电压、电流监测和保护。

软件通过 APCON 寄存器完成对差分放大器的设置。可以通过 APCON 完

放大器设置完成后，可以通过置位 ADCON1 寄存器的 DIFS 位，将放大器的输出与 ADC 的采样保持电路连接，使用 ADC 实现对放大信号的测量。

APCON 以及相关寄存器位的定义，请参考本章节寄存器定义部分。

1/5 多路输入分压器(VDO)

对于一些常用的电压测量，由于采用比较低的内部或者外部参考，不能用于测试高于参考电压的电位。同样，为实现 ADC 的内部电源电压测量通道，我们必须将电源电压分压至一个合理的电位，然后才能使用 ADC 进行准确的转换。为此，我们在 LGT8F684A 内部集成了一个具有固定 1/5 分压的多路输入分压器，可以满足最高到系统工作的电源电压的直接测量。

多路输入分压器的输入可以通过 ADCON1 寄存器的 VDS1/0 位进行选择控制。分压输入可以在外部引脚输入 RC2/APP, RA7/APN 以及系统工作电源三者之间选择。当选择系统工作电源时，可以通过设置 ADC 的模拟输入多路复用，选中 VDO 作为 ADC 当前采用输入通道，完成对电源电压的测量。

系统上电复位后，VDO 将默认选择 RA7/APN 作为输入通道。如果此时 RA7/APN 作为其他功能使用，VDO 将呈现位一个对地的下拉电阻。另外，在进入休眠模式后，默认的 VDO 设置也将会带来额外的漏电，因此，需要通过将 VDS1/0 设置位 11 的方式，关闭 VDO 模块。这里需要特别注意。

ADC 工作相关配置

在使用 ADC 之前，需要首先完成一下系统准备工作：

- 端口模式配置
- ADC 输入通道选择
- ADC 参考电压选择
- ADC 转换时钟选择
- 中断相关控制设置
- ADC 转换结果格式设置

端口相关配置

ADC 常用与转换模拟输入信号。因此在转换之前需要将输入 I/O 设置位模拟工作模式。ANSEL/1 寄存器用于开启端口的模拟输入通道。但 ANSEL/1 只负责开启端口的模拟输入通道，并不会控制端口的数字输入输出方向；需要通过 TRISA/C 寄存器将对应的端口设置为输入状态后，端口才能正确的表现外部模拟输入的电平变化。

ADC 输入通道选择

ADC 支持多路模拟输入，内部通过一个模拟多路复用器将多路输入复用到 ADC 的采样保持电路。因此在启动 ADC 转换器，首先需要设置 ADC 转换的当前输入通道。LGT8F684A 的 ADC 多路复用内部一共支持 10 路输入信号，其中 0~7 通道对应芯片的外部输入端口 AN0~7；通道 8, 9 为内部模拟输入专用。通道 8 连接内部 1/5 分压器(VD0)；通道 9 连接内部的 1.2V 参考基准。ADCON0 寄存器的 CHS2-0 以及 ADCON1 寄存器的 CHS3 组合为 CHS[3:0]共同实现对 ADC 转换当前输入通道的选择。

除了 ADC 专用的多路复用输入通道，ADC 还可以转换来自内部差分放大器(DAP)输出；通过将 ADCON1 寄存器的 DIFS 位设置为 1，将差分放大器的输出接入 ADC 的转换输入，可以实现对差分放大器输出电压的测量。

ADC 转换参考基准

ADCON0 寄存器的 VCFG[1:0]用于选择 ADC 转换的参考电压基准。参考基准可以在系统工作电源电压，外部 RA1/AVREF 输入电压以及内部 1.2V 基准之间选择。

ADC 转换时钟选择

LGT8F684A 内部 ADC 最高支持 2MHz 的转换时钟频率。高速的转换速率将会对 ADC 的转换精度产生影响。当用于主要较高的转换精度时，可通过降低 ADC 的转换时钟频率实现。ADC 推荐的转换时钟为 128KHz~512KHz 左右。可以根据应用的系统时钟频率，通过 ADC 时钟与分频器，选择合适的转换时钟。

ADC 采样逐次逼近的量化算法，转换 1 个位需要 1 个 ADC 转换周期，12 位的结果一共需要 13 个转换周期才能完成，加上系统采用转换完成信号的同步时钟，ADC 一次转换需要 15 个 ADC 转换时钟周期。

ADCON1 寄存器的 ADCS 位用于选择 ADC 的转换时钟，转换时钟可以在系统工作时钟频率的 2 分频到 64 分频之间选择。详细定义，请参考本章节寄存器定义部分。

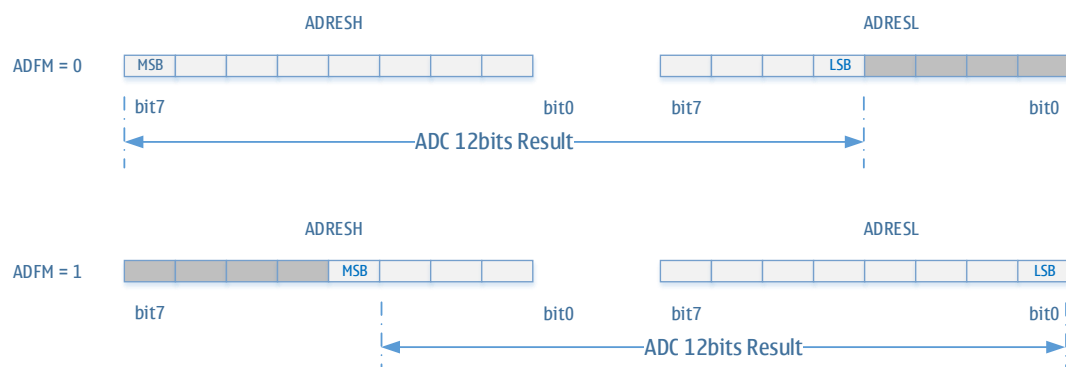
相关中断设置

ADC 在转换完成后，可以产生一个转换完成中断请求。ADC 转换完成中断的标志位位于 PIE1 寄存器的 ADIF 位。ADC 中断使能控制位为 PIE1 寄存器的 ADIE 位。ADIF 标志位必须在中断复位程序中由软件清零。

ADIF 位在每次转换完成的同一周期产生。即使没有使能 ADC 的中断功能，ADIF 位仍然会置位。当系统使能了 ADC 的中断请求功能，并且系统也通过 GIE 位使能了全局中断请求，ADC 转换完成后，将会中断系统指令的正常运行流程，转向执行硬件中断服务程序。用户需要在硬件中断服务程序中通过 PIE1 寄存器判断中断发生的设备，根据中断源的不同进行对应的处理。

转换结果格式

ADC 的转换结果位一个 12 位的二进制数据。被分为两个部分保存到两个 8 位的寄存器(ADRESH/L)。ADCON0 寄存器的 ADFM 位用于设置 12 位结果在两个 8 位寄存器中的存放格式。ADFM 的配置与对应的 ADC 结果格式如下图所示：



ADC 操作流程

启动转换

ADCON0 寄存器的 ADON 位用于使能 ADC 模块；ADON=1 后，系统开启给 ADC 供电，ADC 进入等待转换开始模式。此后，需要通过设置 ADCON0 寄存器的 GO/DONE 位开启一次转换流程。需要注意的时，开启 ADC 转换流程与 ADC 使能不可同时设置。同时设置的开启转换动作无效。

转换完成识别

ADC 完成一次转换后，将会进行如下状态变化：

- 将 ADCON0 的 GO/DONE 位清零
- 设置 ADIF 标志位
- 将 ADC 的转换结果更新到 ADRESH/L 寄存器

自动事件驱动

ECCP 模块的特定事件可用于驱动 ADC 的周期转换；此转换过程无需软件干预。当触发开始后，ADCON0 寄存器的 GO/DONE 位被置位，定时器 TMR1 的计数器也被同时清零。

需要注意的是，自动触发事件并不能保证 ADC 的转换时序的完整，这需要用户负责设置。

ADC 转换流程实例

; ADC 转换实例配置			
DATAH	equ	0x70	
DATAL	equ	0x71	; define temple variable
	BSF	STATUS, RP0	; select bank 1
	MOVLW	0x10	; ADCS = 001
	MOVWF	ADCON1	
	BSF	TRISA, 0x2	; set RA2 to input
	BSF	ANSEL, 0x2	; set RA2 to analog
	BCF	STATUS, RP0	; select bank 0
	MOVLW	0x89	; set ADFM=1, select AN2 as current channel
	MOVWF	ADCON0	; and enable ADC module
START	NOP		
	BSF	ADCON0, 0x1	; start ADC convert
CLOOP	NOP		
	BTFSZ	ADCON0, 0x1	; wait for coverting done
	GOTO	CLOOP	; continue waiting
	MOVF	ADRESH, W	; store ADC high byte
	MOVWF	DATAH	; to template memory (we only need high 8bits)
	CALL	adc_function	; call data process
	GOTO	START	; start another convert cycle
	END		;

寄存器定义

ADCON0 – ADC 控制寄存器 0

ADCON0– ADC 控制寄存器 0								
地址: 0x1F					0000_0000			
Bit	7	6	5	4	3	2	1	0
	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/DONE	ADON
R/W	R/W	R/W	R/W	R/W	R/W	W/R	R/W	W/R
Bit	Name	描述						
7	ADFM	ADC 转换结果格式设置。1 = 右对齐；0 = 左对齐						
6	VCFG1	ADC 参考电压配置位 1						
5	VCFG0	ADC 参考电压配置位 0。VCFG[1:0] = 00：选择系统工作电源作为参考电压 10：选择外部参考输入 RA1/AVREF 作为参考电压 X1：选择内部 1.2V 基准源作为参考电压						
4:2	CHS[2:0]	ADC 输入通道选择位，配合 ADCON1 寄存器的 CHS3 位共同选择 ADC 转换的当前输入通道；CHS[3:0] = 0000：选择外部输入 AN0 0001：选择外部输入 AN1 0111：选择外部输入 AN7 1000：选择内部 1/5 分压器 1001：选择内部 1.2V 基准电压源						
1	GO/DONE	ADC 转换开始以及转换完成标志位，写 1 启动一次 ADC 转换。此位为 0 表示 ADC 转换完成，ADC 处于空闲状态。						
0	ADON	ADC 使能控制位，1=使能 ADC 模块；0=关闭 ADC 模块						

ADCON1 – ADC 控制寄存器 1

ADCON1– ADC 控制寄存器 1								
地址: 0x9F					0000_0000			
Bit	7	6	5	4	3	2	1	0
	CHS3	ADCS2	ADCS1	ADCS0	DIFS	ADSPD	VDS1	VDS0
R/W	R/W	R/W	R/W	R/W	R/W	W/R	R/W	W/R
Bit	Name	描述						
7	CHS3	ADC 输入通道选择最高位。配合 ADCON0 的 CHS2:0 使用。具体信息请参考 ADCON0 寄存器定义部分。						
6:4	ADCS[2:0]	ADC 转换时钟配置位。ADCS[2:0] = 000/011：转换时钟为 FSYS/2 001：转换时钟为 FSYS/8 010：转换时钟为 FSYS/32 100：转换时钟为 FSYS/4 101：转换时钟为 FSYS/16						

		110: 转换时钟为 FSYS/64 111: 转换时钟为 FSYS/32 其中 FSYS 为系统工作频率
3	DIFS	差分放大器输出到 ADC 采用通道使能位 1 = ADC 采样差分放大器输出 0 = ADC 采用 ADC 多路复用输出
2	ADSPD	ADC 低速低功耗模式。当 ADC 工作在低速模式时，可以通过设置此位或者更低的 ADC 运行功耗。
1:0	VDS[1:0]	内部 1/5 分压电路输入源选择。VDS[1:0] = 00: 选择 RA7/APN 作为分压电路输入电压 01: 选择 RC2/APP 作为分压电路输入电压 10: 选择系统工作电源 VCC 作为输入电压 11: 关闭分压电路模块(VDO)

ADRESH – ADC 转换结果高位

ADRESH– ADC 转换结果高位								
地址: 0x1E					XXXX_XXXX			
Bit	7	6	5	4	3	2	1	0
	ADRESH[7:0]							
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Bit	Name	描述						
7:0	ADRESH	ADC 转换结果高位，具体含义请参考本章节关于转换结果格式的介绍						

ADRESL – ADC 转换结果低位

ADRESL– ADC 转换结果低位								
地址: 0x9E					XXXX_XXXX			
Bit	7	6	5	4	3	2	1	0
	ADRESL[7:0]							
R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Bit	Name	描述						
7:0	ADRESL	ADC 转换结果低位，具体含义请参考本章节关于转换结果格式的介绍						

APCON – 差分放大器控制寄存器

APCON– 差分放大器控制寄存器								
地址: 0x1C					0000_0000			
Bit	7	6	5	4	3	2	1	0
	APEN	-	DPS1	DSP0	DNS1	DNS0	GA1	GA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	APNE	差分放大器使能控制。1 = 使能差分放大器						
6	-	Unimplemented						
5:4	DPS1:0	差分放大器正向输入源选择; DPS[1:0] =						

		00: 选择 ADC 多路复用输出 01: 选择 RCA7/APN 10: 选择 RC2/AN6/APP 11: 关闭正端输入
3:2	DNS1:0	差分放大器反向输入源选择; DNS[1:0] = 00: 选择 RC2/AN6/APP 01: 选择 RA7/APN 10: 选择系统地 GND 11: 关闭正端输入
1:0	GA[1:0]	差分放大器增益选择; GA[1:0] = 00: x1 01: x8 10: x16 11: x32

ANSEL – 端口工作模式选择寄存器

ANSEL- 端口工作模式选择寄存器								
地址: 0x91					1111_1100			
Bit	7	6	5	4	3	2	1	0
	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
Bit	Name	描述						
7	ANS7	RC3/AN7 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
6	ANS6	RC2/AN6 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
5	ANS5	RC1/AN5 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
4	ANS4	RC0/AN4 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
3	ANS3	RA4/AN3 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
2	ANS2	RA2/AN2 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
1	ANS1	RA1/AN1 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
0	ANS0	RA0/AN0 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						

ANSEL1 – 端口工作模式选择寄存器 1

ANSEL1- 端口工作模式选择寄存器 1								
地址: 0x98					0000_0000			
Bit	7	6	5	4	3	2	1	0
	ANTM	-	-	-	-	-	ANS9	ANS8
R/W	R/W	-	-	-	-	-	R/W	R/W
Bit	Name	描述						
7	ANTM	ADC 输入通道测试模式, 正常工作请设置位 0						
6:2	-	Unimplemented						
1	ANS9	RA7/APN 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						
0	ANS8	RA6/DAC2 端口工作模式控制; 1 = 模拟 I/O; 0 = 数字 I/O						

系统配置位

LGT8F684A 包含两个独立的配置字,用于设置系统以及外设的运行模式。本章主要介绍配置字的定义,配置字以及编程信息,请参考相关资料。

配置字 1: 系统时钟相关配置

Bit13							Bit0	
CF1EN	N/C	RSTE	WDTE	SUT[1:0]	TSSM	OSCO	RCM[1:0]	FOSC[2:0]
位定义								
FOSC[2:0]	系统时钟模式配置字 000/111: HFINTOSC 模式, 内部高速 RC 110: RCK, 32KHz 内部 RC 时钟 101: HFOSC, 外部高速晶振模式 100: CLKIN, 外部时钟输入模式(RA5) 011: LFOSC, 外部低速(32.768KHz)晶振模式 Others: Unimplemented							
RCM[1:0]	内部 HFINTOSC 时钟模式配置字, 选择时钟输出中心频率 00: 1MHz 01: 8MHz 1X: 16MHz							
OSCO	端口 RA4 输出系统时钟							
TSSM	双速启动模式							
SUT[1:0]	启动时间配置字 11: 2ms 10: 150us 01: 254ms 00: 63ms							
WDTE	使能看门狗定时器							
RSTE	使能 RA3 作为外部复位输入							
CF1EN	配置字使能控制; 0 = 配置字 0 有效, 1 = 配置字 0 无效							

配置字 2: 系统运行相关配置

Bit13							Bit0	
CF2EN	N/C	PPLP	OSCFSEN	TCYC[1:0]	DPSM[1:0]	PINRM	LVDPM	LVDT[2:0]
位定义								
LVDT[2:0]	LVR 复位阈值配置字 111: 关闭 LVR 000: 1.8V 001: 2.0V 010: 2.2V 011: 2.4V							

	100: 2.6V 101: 3.6V 110: 4.0V Others: Unimplemented
LVDPM	LVR 休眠模式配置字 1: 休眠模式下不关闭 LVR 0: 休眠模式下关闭 LVR, 唤醒后使能
PINRM	读 I/O 来源控制, 仅对于 BCF/BSF 指令有效 1: 读 PORT 来自 PORT 寄存器 0: 读 PORT 来自外部引脚输入
DPSM[1:0]	休眠模式控制 00: IDLE 模式 01: 省电模式 1X: 深睡眠模式
TCYC[1:0]	内核指令周期配置字 1X: 4T, 指令周期等于 4 个系统时钟周期 01: 2T, 指令周期等于 2 个系统时钟周期 00: 1T, 指令周期等于 1 个系统时钟周期
OSCFSEN	外部晶振失效保护使能
PPLP	内部程序存储器低功耗模式; 1 = 正常工作模式, 0 = 低功耗模式 当系统工作在低于 2MHz 频率, 可以使能此模式获得更低的功耗
CF2EN	配置字 1 有效控制; 1 = 配置字 1 无效, 0 = 配置字 1 有效

指令集速查表

指令名称	操作	功能描述	指令字	状态位	周期
基本算术运算指令					
SUBWF	SUBWF F ¹ , d ²	$[W/F] = (F) - W$	1	C/DC/Z	1
DECF	DECF F, d	$[W/F] = (F) - 1$	1	Z	1
ADDWF	ADDWF F, d	$[W/F] = (F) + W$	1	C/DC/Z	1
COMF	COMF F, d	$[W/F] = \text{complement } F$	1	Z	1
INCF	INCF F, d	$[W/F] = (F) + 1$	1	Z	1
SUBLW	SUBLW k ⁴	$W = k - W$	1	C/DC/Z	1
ADDLW	ADDLW k	$W = k + W$	1	C/DC/Z	1
ADCWF	ADCWF F, d	$[W/F] = (F) + W + C$	1	C/DC/Z	1
SBCWF	SBCWF F, d	$[W/F] = (F) - W + C$	1	C/DC/Z	1
DAA	DAA	Decimal correct for W after ADDX	1	Z	1
DAS	DAS	Decimal correct for W after SUBX	1	-	1
CLRF	CLRF F	$(F) = 0$	1	Z	1
CLRW	CLRW	$W = 0$	1	Z	1
基本逻辑运行指令					
IORWF	IORWF F, d	$[W/F] = (F) W$	1	Z	1
ANDWF	ANDWF F, d	$[W/F] = (F) \& W$	1	Z	1
XORWF	XORWF F, d	$[W/F] = (F) \wedge W$	1	Z	1
RRF	RRF F, d	$[W/F] = \{C, F[6:0]\}$	1	C	1
RLF	RLF F, d	$[W/F] = \{F[6:0], C\}$	1	C	1
SWAPF	SWAPF F, d	$[W/F] = \{F[3:0], F[7:4]\}$	1	-	1
BCF	BCF F, b ³	$F[b] = 0$	1	-	1
BSF	BSF F, b	$F[b] = 1$	1	-	1
ANDLW	ANDLW k	$W = k \& W$	1	Z	1
XORLW	XORLW k	$W = k \wedge W$	1	Z	1
IORLW	IORLW k	$W = k W$	1	Z	1
基本流程控制指令					
DECFSZ	DECFSZ F, d	$[W/F] = (F) - 1$ IF(Z) PC = PC + 2	1	-	1/2
INCFSZ	INCFSZ F, d	$[W/F] = (F) + 1$ IF(Z) PC = PC + 2	1	-	1/2
BTFSZ	BTFSZ F, b	IF(!F[b]) PC = PC + 2	1	-	1/2
BTFS	BTFS F, b	IF(F[b]) PC = PC + 2	1	-	1/2
GOTO	GOTO k	PC = k	1	-	2
CALL	CALL k	PC+1 -> STACK, PC = k	1	-	2
RETURN	RETURN	PC <- STACK	1	-	2
RETFIE	RETFIE	GIE = 1 PC <- STACK	1	-	2

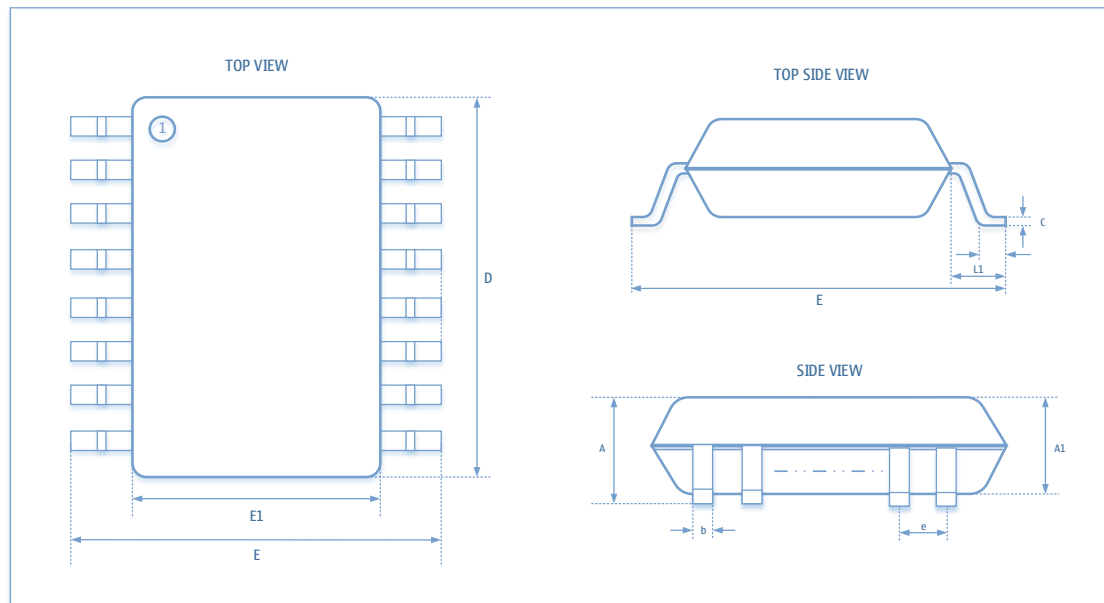
RETLW	RETLW k	W = k, PC <- STACK	1	-	2
基本数据传输指令					
MOVWF	MOVWF F	(F) = W	1	-	1
MOVF	MOVF F, d	[W/F] = (F)	1	Z	1
MOVLW	MOVLW k	W = k	1	-	1
其他辅助指令					
NOP	NOP	NO operation	1	-	1
OPTION	OPTION	OPTION = W	1	-	1
TRIS	TRIS F	IOSTA/B = W, (F=5/6)	1	-	1
SLEEP	SLEEP	Sleep mode	1	TO/PD	1
CLRWDT	CLRWDT	Clear WDT	1	TO/PD	1
INT	INT	PC+1 -> STACK PC = 0x2, GIE = 0	1	-	2
扩展数据传输指令					
MOVLT	MOVLT k	PCH = k[9:8] FSR = k[7:0]	2	-	2
MOVLC	MOVLC k	LCR = K[7:0]	2	-	2
MOVLI	MOVLI k	FSR = k[7:0]	2	-	2
MOVWP	MOVWP	W = PMEM[PCH:FSR]	1	-	2
	MOVWP++	W=PMEM[PCH:FSR] FSR++	1	-	2
	MOVWP--	FSR— W = PMEM[PCH:FSR]	1	-	2
IMOVW	IMOVW	W = DMEM[FSR]	1	-	1/2
	IMOVW++	W=DMEM[FSR] FSR++	1	-	1/2
	IMOVW--	FSR— W=DMEM[FSR]	1	-	1/2
	IMOVW +q ⁵	W=DMEM[FSR+q]	1	-	1/2
	IMOVW -q	W=DMEM[FSR-q]	1	-	1/2
IMOVF	IMOVF	DMEM[FSR] = W	1	-	1
	IMOVF++	DMEM[SFR] = W SFR++	1	-	1
	IMOVF--	SFR— DMEM[FSR] = W	1	-	1
	IMOVF +q	DMEM[FSR+q] = W	1	-	1
	IMOVF -q	DMEM[FSR-q] = W	1	-	1
扩展流程控制指令					
LOOP	LOOP k	IF(LCR!=0) { LCR--, PC = k}	2	-	2
BRSZ	BRSZ k	IF(Z==1) PC = k	2	-	2
BRCZ	BRCZ k	IF(Z==0) PC = k	2	-	2
BRSC	BRSC k	IF(C==1) PC = k	2	-	2
BRCC	BRCC k	IF(C==0) PC = k	2	-	2

说明:

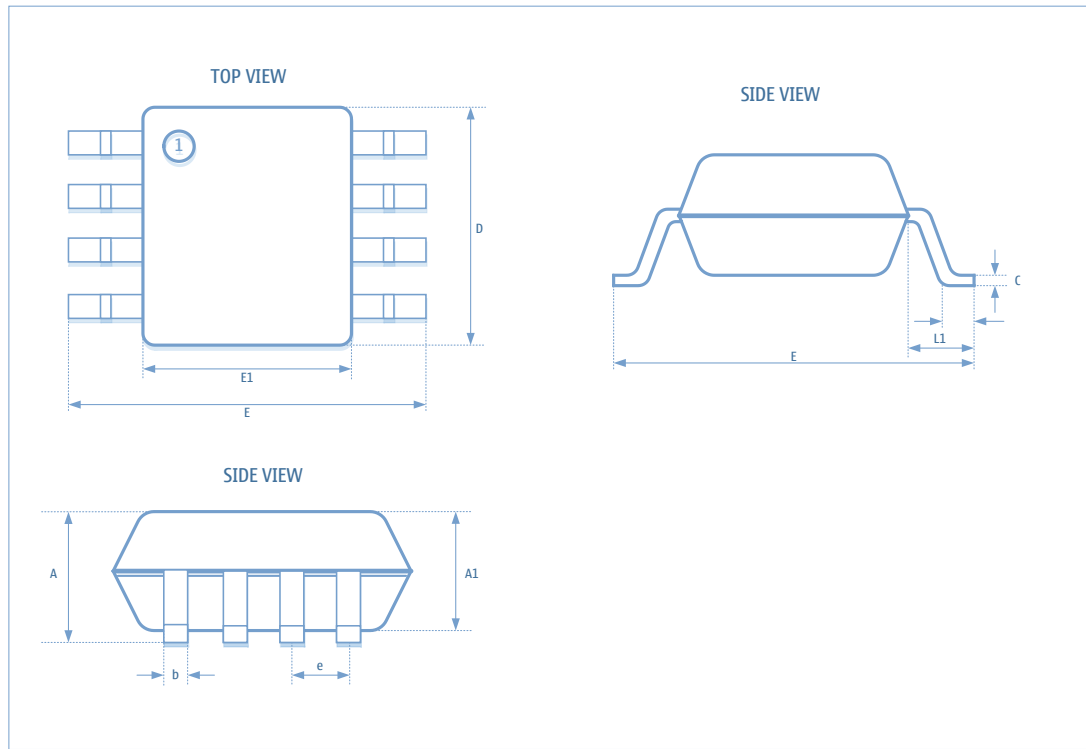
1. 指令集中的 **F** 为一个 **6** 位宽度的立即数，用于指定 **IO/RAM** 的地址；
2. 指令集中的 **d** 为一个 **1** 位宽度的立即数，当 **d=0**（默认）时，指令执行的结果回写到 **W** 工作寄存器，否则写入到 **F** 指定地址；
3. 指令集中的 **b** 为一个 **3** 位的立即数，用于指定访问数据的位地址；
4. 指令集中的 **k** 为一个立即数，根据指令不同，**k** 的宽度分为 **8/10** 位不等，**8** 位宽度一般是数据，**10** 位为程序目标地址；
5. 指令集中的 **q** 为一个 **4** 位的立即数，指定间接寻址模式的地址偏移量
6. **PMEM** 表示程序空间，**DMEM** 代表数据寄存器空间

封装参数

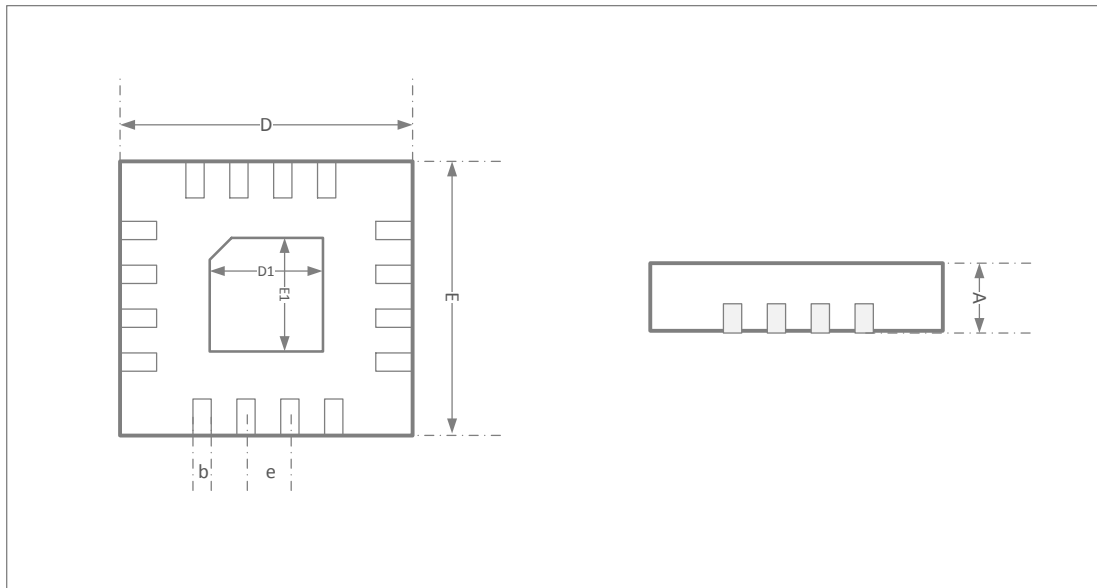
SOP16L 封装



Item	Min.	Typ.	Max.	Unit
A	1.36	1.66	1.96	mm
A1	1.30	1.50	1.70	mm
b	0.30	0.40	0.55	mm
e	-	1.27BSC	-	mm
E	5.70	6.00	6.30	mm
E1	3.75	3.95	4.15	mm
D	9.70	9.90	10.30	mm
C	0.15	0.25	0.35	mm
L	0.45	0.65	0.85	mm
L1	-	1.04BSC	-	mm

SOP8L 封装

Item	Min.	Typ.	Max.	Unit
A	1.35	1.55	1.75	mm
A1	1.25	1.40	1.65	mm
b	0.38	-	0.51	mm
e	-	1.27BSC	-	mm
E	5.80	6.00	6.20	mm
E1	3.80	3.90	4.00	mm
D	4.80	4.90	5.00	mm
C	0.17	-	0.25	mm
L	0.45	0.60	0.80	mm
L1	-	1.04BSC	-	mm

QFN16 封装

Symbols	DIMENSIONS MILLIMETERES		
	MIN	TYP.	MAX
A	0.70	0.75	0.80
b	0.18	0.23	0.28
D	2.95	3.00	3.03
D1	-	1.60BSC	-
E	2.95	3.00	3.03
E1	-	1.60BSC	-
e	-	0.50BSC	-

版本历史

V1.0.8 2016/11/14	增加了 QFN16 封装定义 修改了部分寄存器说明
V1.0.7	更新了 VRCON1 寄存器中对于 DBG_C 位的描述， 请参考 DAC 章节中关于 VRCON1 寄存器的描述部分 此修改针对芯片生产代码大于或等于 1622 的量产版本，对于没有 mark 的芯片， 生产代码位于芯片的背面
V1.0.6	更新了寄存器描述的错误 移除 RC 组下拉控制功能 增加了 SOP8L 封装定义
V1.0.5 2016-04-27	更新了 ADC 中 VDS 输入源的错误
V1.0.4 2016-01-30	更新了 COG 章节相关的寄存器定义 增加了 Timer1 章节 DC1L/H, PR1L/H 寄存器定义
V1.0.3 2016-01-19	修改寄存器描述的错误 1. 增加了 COG 章节 2. 更新了 ADC/DAP 输入信号
V1.0.2 2015-12-30	修改寄存器描述的错误
V1.0.1 2015-11-04	规范寄存器名称定义
V1.0.0 2015-10-23	初始版本